User guide

Sep 09, 2026

Contents

1 An overview of Routing Active Testing **4**

2 Getting started **5**
- 2.1 Logging in to Routing Active Testing . . . 5
- 2.2 Introduction to Routing Active Testing . . . 6
- 2.3 Guided tour of the main menu . . . 7
- 2.4 Introduction to measurement tasks . . . 12
- 2.5 Getting started with HTTP and DNS measurements . . . 12
- 2.6 Getting started with IPTV measurements . . . 13
- 2.7 Getting started with remote packet capture and traffic analysis . . . 14
- 2.8 Getting started with network performance measurements . . . 15
- 2.9 Getting started with your own Speedtest server . . . 16
- 2.10 Getting started with IP telephony (SIP and VoIP) measurements . . . 17

3 Managing your account **18**
- 3.1 Introduction to accounts . . . 18
- 3.2 Changing account settings . . . 18
- 3.3 Administering users and permissions . . . 18
- 3.4 Administering Test Agent registration users . . . 20
- 3.5 Setting up IPTV channels . . . 21
- 3.6 Setting up SIP accounts . . . 22
- 3.7 Setting up Ping hosts . . . 24
- 3.8 Setting up TWAMP reflectors . . . 28
- 3.9 Setting up an IP lookup table . . . 31
- 3.10 Setting up Y.1731 MEPs . . . 33
- 3.11 Setting up network devices . . . 35
- 3.12 Configuring Speedtest . . . 38
- 3.13 Configuring SLA (Service Level Agreement) thresholds . . . 41
- 3.14 Setting up alarms . . . 42
- 3.15 Changing the report logo . . . 51
- 3.16 Changing user settings . . . 52
- 3.17 Creating API tokens . . . 54

4 Test Agents **55**
4.1 Introduction to Test Agents . . . 55
4.2 Installing Test Agents . . . 70
4.3 Configuring Test Agents from the Routing Active Testing GUI . . . 165
4.4 Configuring Test Agents from the local console . . . 204
4.5 Further technical information on Test Agents . . . 232

5 Plugins **242**
5.1 Description of plugins . . . 242

6 Licensing **245**
6.1 Licensing and streams in Routing Active Testing . . . 245

7 Tests and monitors **252**
7.1 Introduction to tests and monitors . . . 252
7.2 Building tests . . . 254
7.3 Options for running tests . . . 258
7.4 The Tests view . . . 259
7.5 View showing an individual test . . . 260
7.6 Building monitors . . . 263
7.7 The Monitoring view . . . 265
7.8 View showing an individual monitor . . . 267
7.9 Reports on tests and monitors . . . 269
7.10 Creating templates . . . 273
7.11 Applying tags . . . 283
7.12 Icons used for tests and monitors . . . 286

8 Task types **286**
8.1 Overview of measurement task types . . . 286
8.2 Common test and monitor parameters . . . 287
8.3 Listing of task types supporting IPv6 . . . 288
8.4 Network performance testing . . . 290
8.5 IPTV and OTT video testing . . . 321
8.6 HTTP and DNS testing . . . 337
8.7 SIP testing . . . 347
8.8 Wi-Fi network testing . . . 351
8.9 Mobile network testing . . . 355
8.10 Ethernet service activation testing . . . 359
8.11 Transparency testing . . . 376
8.12 Reflector-based testing . . . 402
8.13 Security testing . . . 454
8.14 Utilities for testing . . . 470
8.15 Dynamic plugins . . . 471

9 Metrics in Routing Active Testing **473**
9.1 Introduction . . . 473
9.2 Resolution of Routing Active Testing measurement data . . . 473
9.3 Delay variation (DV), jitter . . . 474
9.4 Mean Opinion Score (MOS) . . . 474
9.5 Errored Seconds (ES) metric: Method of calculation . . . 475
9.6 Severely Errored Seconds (SES) . . . 477
9.7 Unavailable Seconds (UAS) . . . 477
9.8 SLA (Service Level Agreement) . . . 478

10 Applications **478**

10.1 Introduction . . . 478
10.2 Remote packet capturing . . . 479
10.3 Speedtest . . . 482
10.4 Setting up a Test Agent to act as proxy in tests . . . 486

11 Alarms **487**
11.1 Introduction to alarms . . . 487
11.2 Activating alarms for a monitor . . . 488
11.3 Alarm dashboard . . . 490

12 Sharing and collaboration **491**
12.1 Sharing Test Agents . . . 491
12.2 Sharing templates . . . 496
12.3 Sharing test and monitor results: Introduction . . . 500
12.4 Sharing test and monitor results between accounts . . . 500
12.5 Sharing test and monitor results using URLs . . . 504

13 Using a proxy **505**
13.1 Using a proxy . . . 505

14 Definitions and technical notes **507**
14.1 Abbreviations . . . 507
14.2 Bridge, bridging . . . 510
14.3 DSCP/DiffServ and IP Precedence . . . 511
14.4 Layer 2 Ethernet frame sizes . . . 512
14.5 MPEG basics . . . 512
14.6 MPEG metrics . . . 512
14.7 MPEG rate vs. MPEG transport rate . . . 513
14.8 Routing Active Testing server . . . 515
14.9 Priority Code Point (PCP) . . . 516
14.10 SNMP . . . 516
14.11 TCP implementation in Routing Active Testing . . . 517
14.12 Theoretical maximum throughput with Routing Active Testing Test Agents . . . 517
14.13 VLAN . . . 520

15 Release notes **521**
15.1 Release notes, Routing Active Testing software version 3.0.0 . . . 521
15.2 Release notes, Routing Active Testing software version 3.1.0 . . . 522
15.3 Release notes, Routing Active Testing software version 3.2.0 . . . 525
15.4 Release notes, Routing Active Testing software version 3.3.0 . . . 527
15.5 Release notes, Routing Active Testing software version 3.4.0 . . . 529
15.6 Release notes, Routing Active Testing software version 4.0.0 . . . 530
15.7 Release notes, Routing Active Testing software version 4.1.0 . . . 531

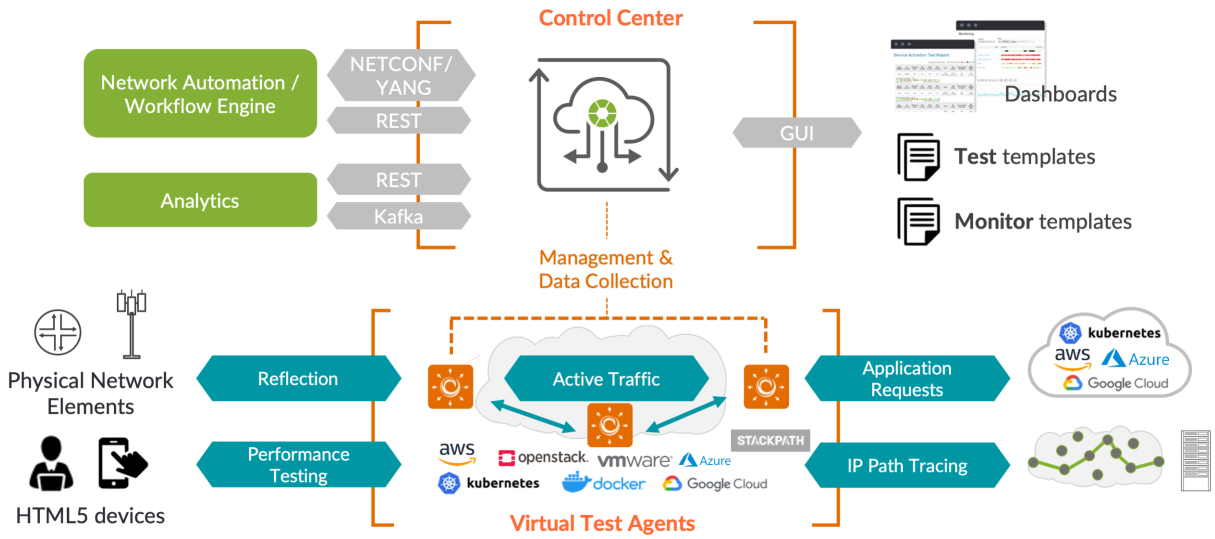
16 Sales and support **532**
16.1 General sales and support information . . . 532

1 An overview of Routing Active Testing

Routing Active Testing, formerly known as Netrounds, is a programmable, active test and service assurance platform for physical, hybrid and virtual networks. The automation capabilities of Routing Active Testing enable communication service providers to reduce manual efforts required for network testing and assurance, significantly decreasing operational costs and improving operating margins, as well as nearly eliminating capital expenditures associated with using traditional hardware test and measurement equipment.

The core component of Routing Active Testing is a unifying multi-tenant **Control Center**, which provides a user-friendly web GUI where operations staff can set up and run on-demand tests and continuous monitoring, as well as view both real-time and aggregated result metrics. Control Center is offered as a SaaS solution at https://app.netrounds.com, but it can also be deployed on-premise. Control Center has REST and NETCONF & YANG APIs which enable external OSS and NFV orchestrators to easily automate tests and monitoring. In the SaaS solution, the REST API browser is found at https://app.netrounds.com/rest.

Control Center remotely controls software-based, traffic-generating active **Test Agents**, whose capabilities include: measurement of network performance (UDP, TCP, Y.1731, TWAMP, Path trace, UDP loopback) and internet performance (HTTP, Ping, Speedtest), IPTV and OTT video, VoIP telephony and SIP, Wi-Fi, and remote packet inspection.



Test Agents may be placed in strategic locations across your network for continuous quality monitoring. They may also be installed on demand for more temporary purposes, such as activation testing of newly deployed services, or troubleshooting. Test Agents are available in a number of formats, and they are all remotely updated and maintained through Control Center.

The present documentation is the main support documentation for Routing Active Testing.

- Further product documentation is available at https://www.juniper.net/documentation/product/us/en/routing-active-testing.
- At https://support.juniper.net/support/requesting-support you can file support tickets and requests for new features in Routing Active Testing.
- If you have an on-premise installation of Control Center, you can download Control Center and various other Routing Active Testing software at https://support.juniper.net/support/downloads.

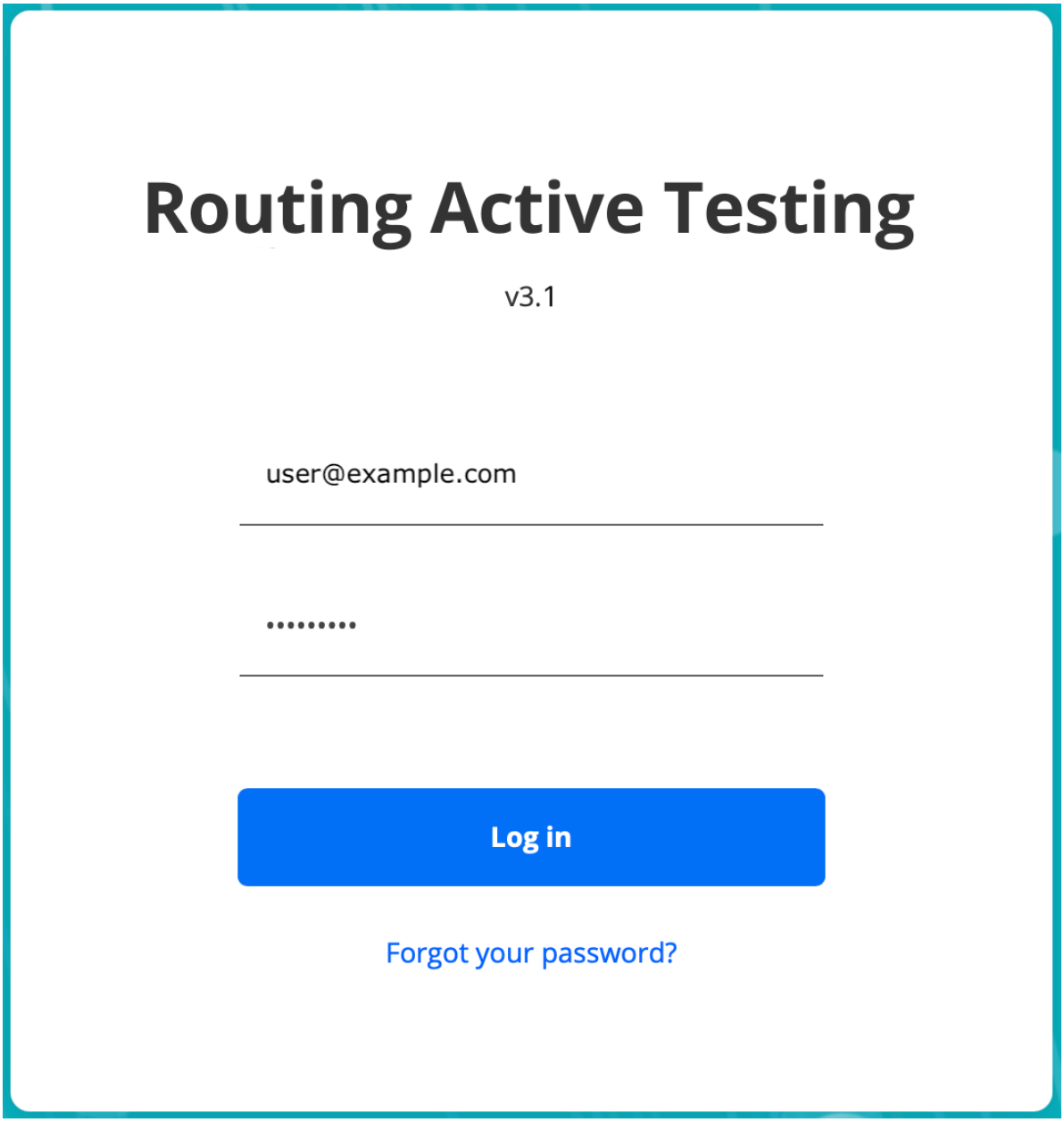
The status of the Routing Active Testing SaaS solution can be inspected at https://status.paa.juniper.net.

2 Getting started

2.1 Logging in to Routing Active Testing

Once you have a Routing Active Testing user created, you log in to Routing Active Testing as follows:

- In the dialog shown below, identify yourself by an email address and sign in by entering your password:

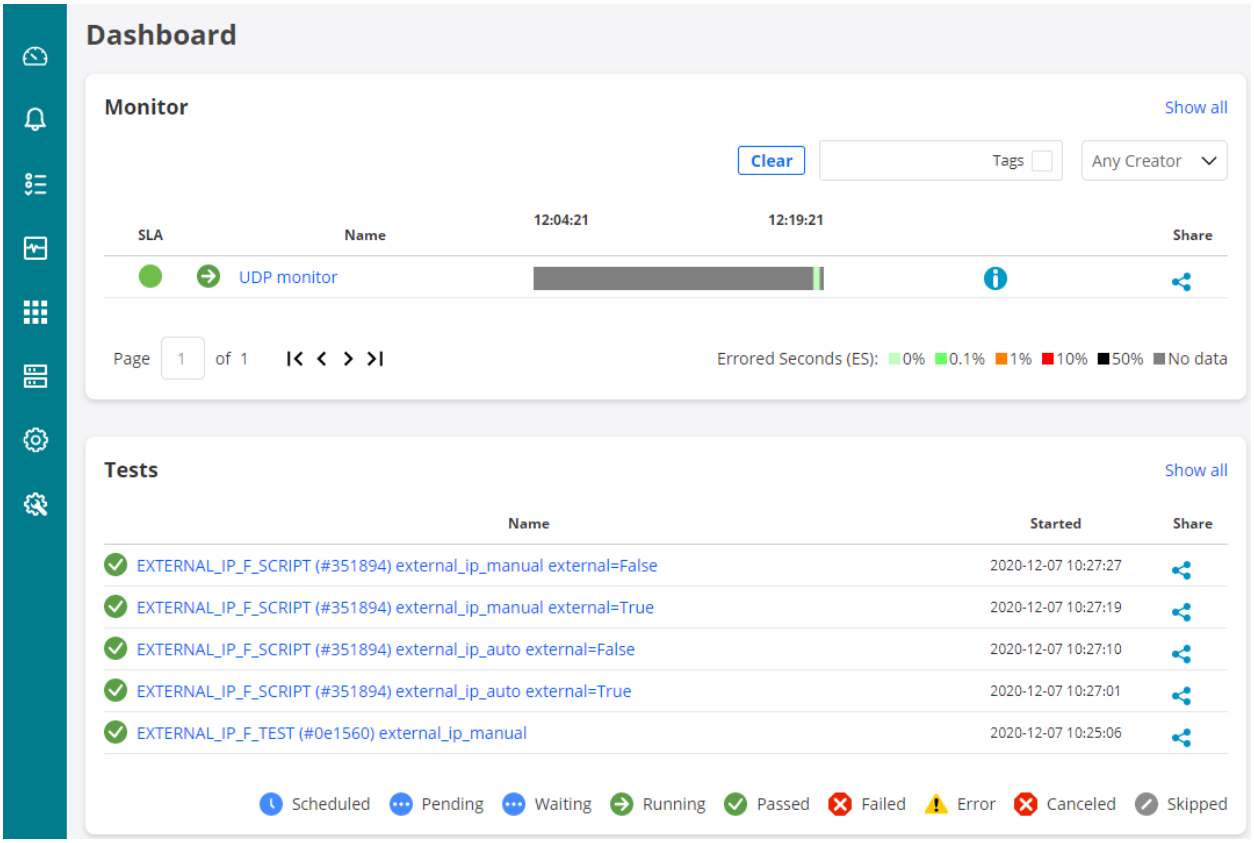


Alternatively, in an on-premise installation of Routing Active Testing, you can log in as a user created on an LDAP server and perform the authentication against that server. This requires configuration which is described in the Installation Guide, chapter “LDAP Authentication”. The login dialog reads “Username/email” in this case.

On successful login you are taken to the Routing Active Testing Dashboard, as shown on *this page* (page 6).

2.2 Introduction to Routing Active Testing

Once you have your account set up, you should see the Routing Active Testing Dashboard:



The Dashboard is where your latest tests and monitors will be displayed.

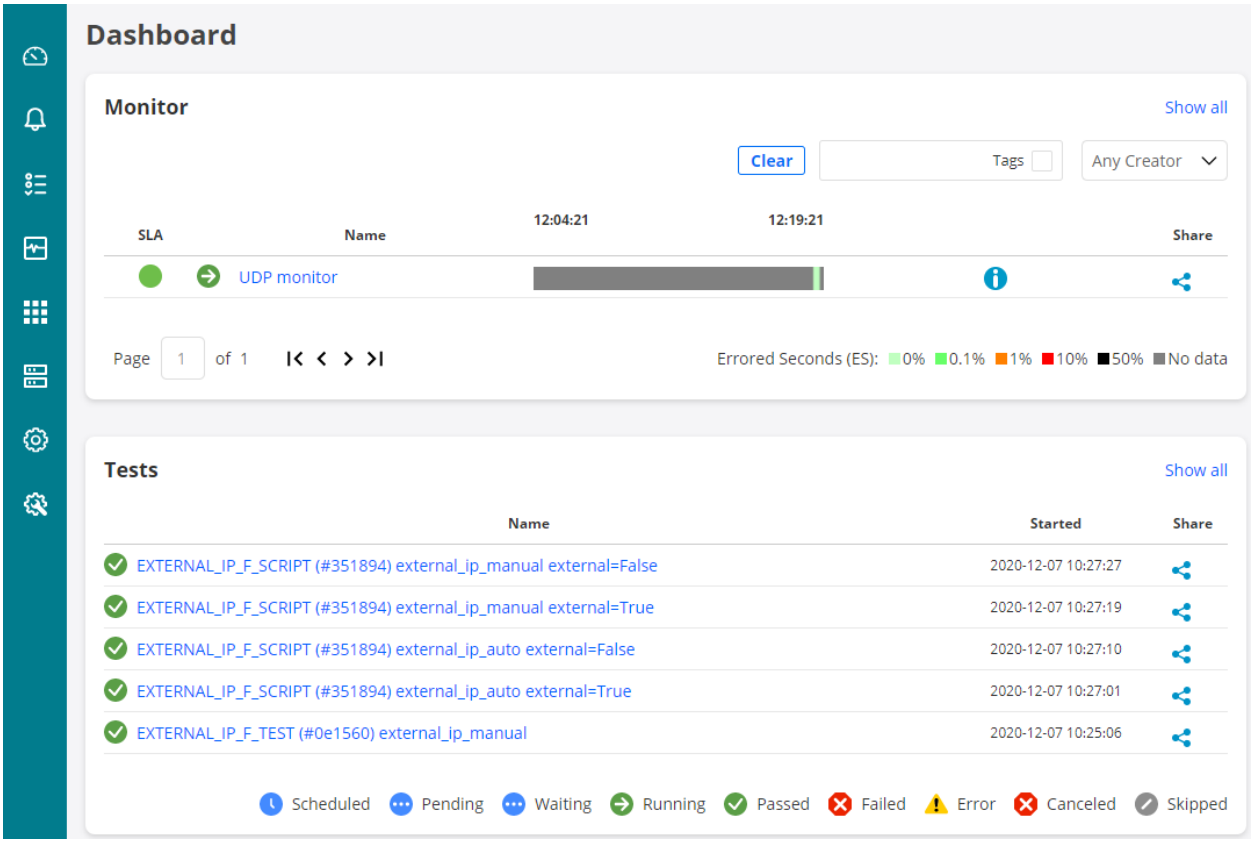
To start monitoring and troubleshooting with all the available features in Routing Active Testing, using your own active traffic generating Test Agents, you can download Test Agent software and install them on your own x86-based hardware or on virtualization platforms. Read more *here* (page 70).

For guidance on preparations for specific measurements supported by Routing Active Testing, see the *Getting started pages* (page 5) in this section.

2.3 Guided tour of the main menu

2.3.1 Dashboard

The Dashboard gives an overview of the most recent monitors and tests in Routing Active Testing.



2.3.1.1 Monitors

- An *SLA* (page 478) indicator and an *errored seconds (ES)* (page 475) bar give direct visual feedback on measured quality. The default time interval for both is 15 minutes; you can select a different time interval at top right, in which case an additional SLA indicator will appear in a column on the far left.

Note: Even if the number of monitors in the system is very large, the 15-minute SLA will not slow down the presentation on the dashboard since this SLA is precomputed. SLAs for other time intervals are not, however, something which you need to keep in mind when working with large monitor sets.

- To view log messages for the monitor, hover over the "i" icon. (You may have to change the History interval setting to see this button.) In the callout that appears, click the "Show details" link to view the full message log, as explained on *this page* (page 268).
- If an error has occurred while executing the monitor, this is indicated as an "!" icon. Hover over that icon to learn more about the error.

2.3.1.2 Tests

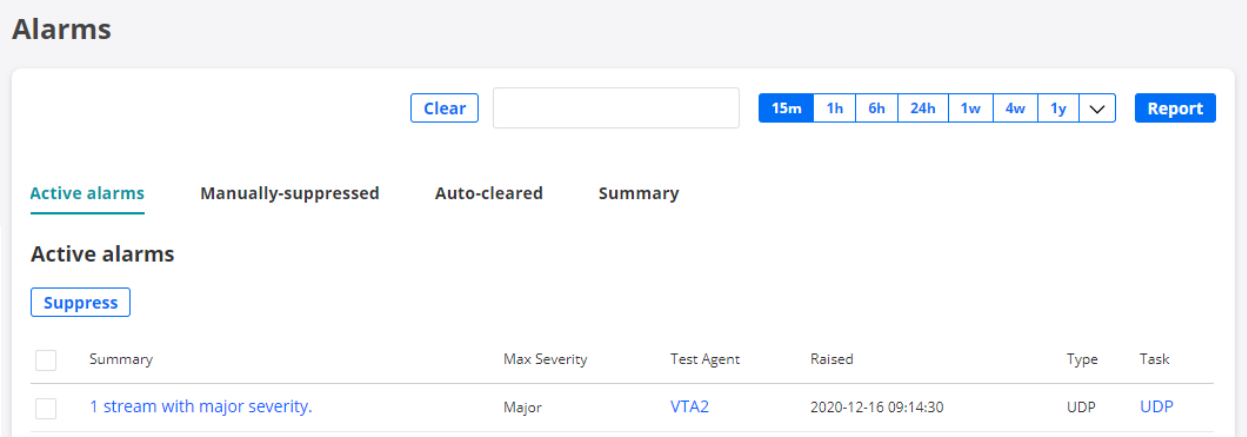
- The start time and a pass/fail indicator are displayed.

The remaining links on the left-side bar take you to further screens, as detailed below.

2.3.2 Alarms

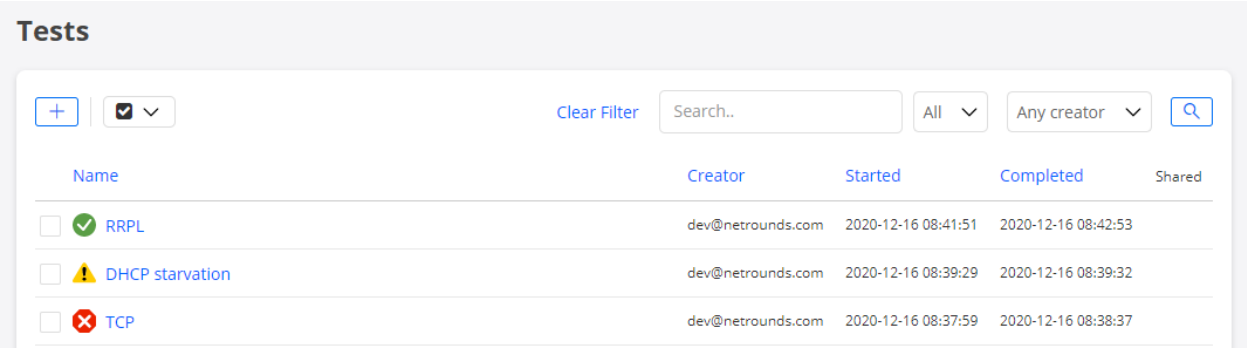
This screen displays *alarms* (page 487) defined for monitors in Routing Active Testing.

How to set up alarms is explained on *this page* (page 42).



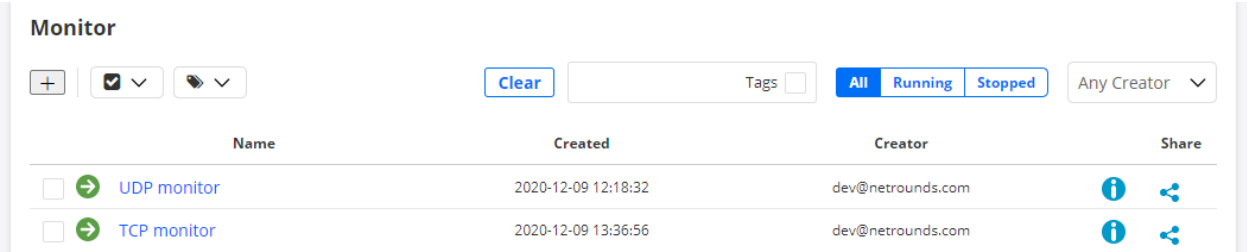
2.3.3 Tests

This screen shows all defined tests. A filter is provided for finding a specific test. To set up a new test, click the Tests button on the left-side bar and select New Test Sequence. Read more *here* (page 254) about how to do this.



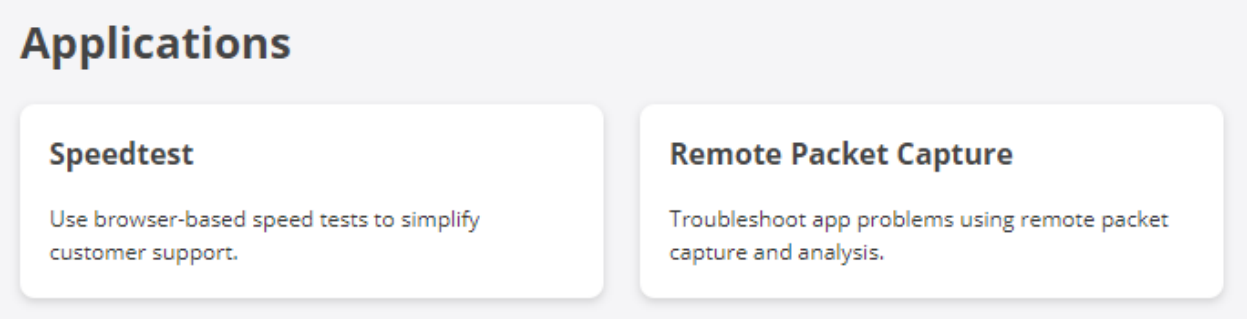
2.3.4 Monitoring

This screen displays all defined monitors. Here, too, a filter is provided for locating a specific monitor. To set up a new monitor, click the Monitoring button on the left-side bar and select New Monitor. Go to *this page* (page 263) for full instructions on how to build monitors.



2.3.5 Apps menu

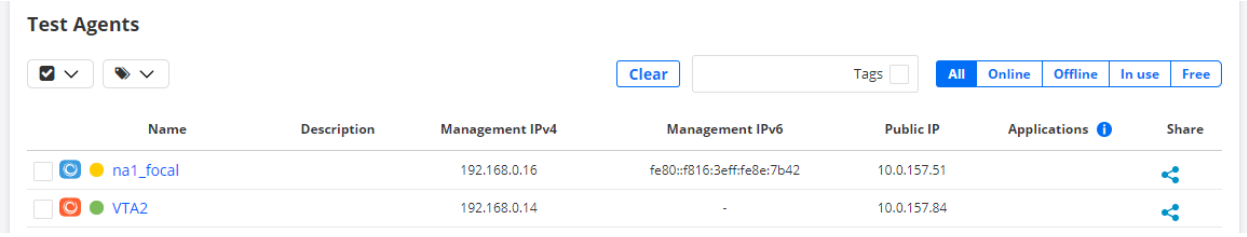
From here you access the *Speedtest* (page 482) and *Remote packet capture* (page 479) features.



2.3.6 Test Agents

2.3.6.1 Interface info tab

Under Test Agents, you will find all your registered Test Agents.



Under Test Agents shared with me, any *shared* (page 491) Test Agents will show up.

The colored dot immediately to the left of the Test Agent name indicates the current status of the Test Agent:

	Green: Online and ready, currently not in use
	Yellow: Online and currently in use
	Gray: Offline

2.3.6.2 License info tab

Here the license and stream information for your Test Agents is displayed:

- License type
- Number of available streams and number of streams currently in use.

Name	License	No. of streams	Used streams	Available streams	Share
na1_focal	Unlimited	8800	2	8798	
VTA1	SW-Test Agent Medium	100	2	98	
VTA2	Unlimited	8800	0	8800	

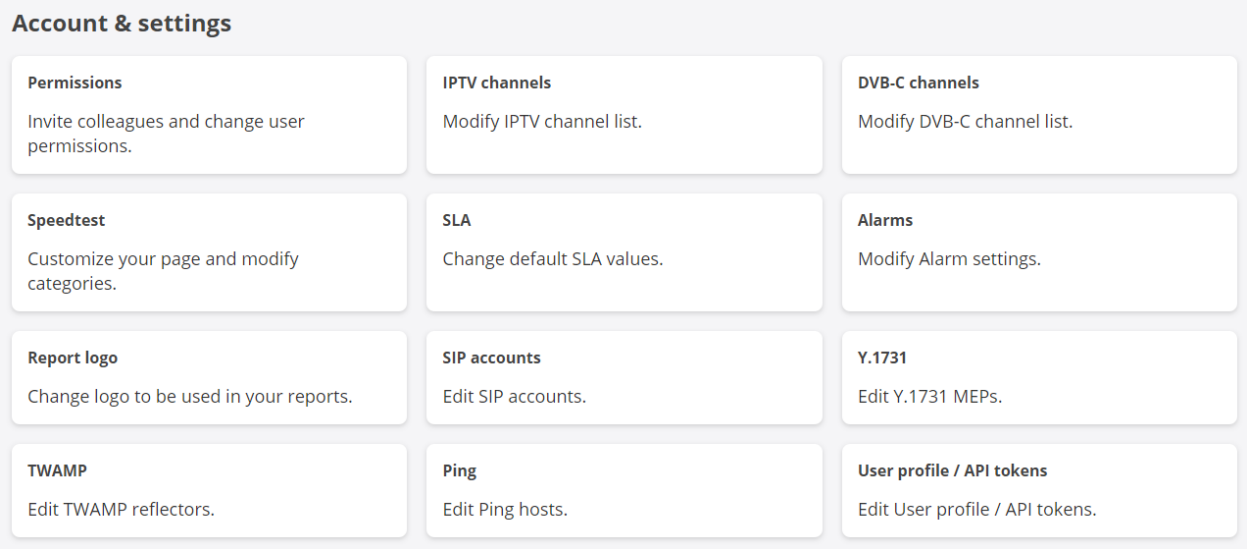
At the bottom, under the heading Licenses, are displayed the types of license connected to your account. Please refer to *this page* (page 245) for more detailed information.

Test Agent Licenses

Type	No. of licenses	Used licenses	Available licenses
SW-Test Agent Mini	10000	0	10000
SW-Test Agent Small	10000	0	10000
SW-Test Agent Medium	10000	1	9999
SW-Test Agent Large	10000	0	10000
SW-Test Agent 8800	10000	0	10000
SW-Agent Mini	10000	0	10000
SW-Agent Small	10000	0	10000
SW-Agent Medium	10000	0	10000
Unlimited	10000	2	9998
IPTV	-	Not used	Available
TWAMP	-	Not used	Available

2.3.7 Account

Read more *here* (page 18) about the various items under Account & Settings.



2.3.8 Plugins

Read more *here* (page 242) about plugins.

2.3.9 Docs

Clicking this button takes you to the contents page of the present documentation.

2.3.10 Top bar: “Bell” button

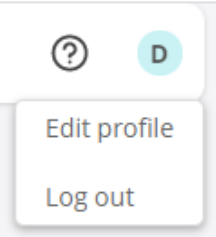


This button will light up if a notification is received; for example, if an *alarm* (page 490) is triggered or if someone *shares a Test Agent* (page 491) to your account.

2.3.11 Top bar: User name

Click your Routing Active Testing user name on the top bar in order to:

- edit your Routing Active Testing user profile
- switch to a different Routing Active Testing account (if you have access to several accounts)
- log out from Routing Active Testing.



2.4 Introduction to measurement tasks

In the Getting started chapter we will introduce a subset of the available measurement tasks just to get you started.

For the full range of measurement tasks, see the *Task types* (page 286) chapter.

2.5 Getting started with HTTP and DNS measurements

To get started with HTTP and DNS measurements in Routing Active Testing, please follow these simple steps.

2.5.1 Download a Test Agent

In this step we will download, install, and configure a Test Agent.

- Under Test Agents in the Routing Active Testing user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for Test Agents given *here* (page 70).

By default, once connected and powered up, all Test Agents will communicate directly with the Routing Active Testing server over an encrypted connection:

- In the cloud server case, to https://login.paa.juniper.net using:
  - Test Agent Appliance: TCP port 443
  - Test Agent Application: TCP port 6800
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.5.2 Start your first test

Follow the instructions below to start a simple HTTP test.

1. Log in to your account in the Routing Active Testing user interface.
2. On the left-side bar, click the Tests button and select New Test Sequence.
3. Select the HTTP & DNS task category.
4. Select the HTTP task.
5. Under Clients, select one of the Test Agents you have installed.
6. Under URL, specify what URL to request from the client Test Agent.
7. Give the test a name, and click Start.
8. Done; you should see measurement results starting to come in very soon.

To learn more, read the page *Introduction to tests and monitors* (page 252).

A full treatment of HTTP and DNS testing is found on the following pages:

- *HTTP* (page 339)
- *DNS* (page 345)

2.6 Getting started with IPTV measurements

2.6.1 Download a Test Agent

In this step we will download, install, and configure a Test Agent.

- Under Test Agents in the Routing Active Testing user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for Test Agents given *here* (page 70).

By default, once connected and powered up, all Test Agents will communicate directly with the Routing Active Testing server over an encrypted connection:

- In the cloud server case, to https://login.paa.juniper.net using:
  - Test Agent Appliance: TCP port 443
  - Test Agent Application: TCP port 6800
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

However, as an alternative setup, one Test Agent may act as proxy for all other Test Agents inside your network. This is useful in situations where the other Test Agents cannot easily obtain an Internet connection, for example if the IPTV network is isolated and cannot reach the Internet-based Routing Active Testing cloud servers.

Please read *here* (page 486) for more information about setting up Routing Active Testing using a proxy.

2.6.2 Add IPTV channels

Add some channels to your channel list. Read more on the page *Adding and configuring IPTV channels* (page 21).

2.6.3 Connect to your network and start your first test

Follow the instructions below to start a simple IPTV test.

1. Log in to your account in the Routing Active Testing user interface.
2. On the left-side bar, click the Tests button and select New Test Sequence.
3. Select the IPTV & OTT video task category.
4. Select the IPTV MPEG test.
5. Under Clients, select one of the Test Agents you have installed.
6. Under Channels, select channels from the ones you added to the IPTV channel list.
7. Give the test a name, and click Start.

You should now start getting IPTV MPEG measurement results, and it should only take a few seconds until you can judge the quality of your IPTV stream at the point in your network where you have connected the Test Agent.

To learn more, read the page *Introduction to tests and monitors* (page 252).

A full treatment of IPTV testing is found on the following pages:

- *IPTV MPEG* (page 322)
- *IPTV MPEG inline* (page 324)
- *IPTV channel zapping time* (page 327)
- *OTT testing: HTTP Live Streaming (HLS)* (page 329)

2.6.4 Typical setup

For IPTV measurements, the most common setup is to connect the Test Agent to a network port, as if it were a set-top box. Typically one is connected where the IPTV signal enters the network, and another somewhere else in the network where customers are having problems.

2.7 Getting started with remote packet capture and traffic analysis

To get started using Routing Active Testing for traffic analysis (RPCAP), please follow these simple steps.

2.7.1 Download a Test Agent

In this step we will download, install, and configure a Test Agent.

- Under Test Agents in the Routing Active Testing user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for Test Agents given *here* (page 70).

By default, once connected and powered up, all Test Agents will communicate directly with the Routing Active Testing server over an encrypted connection:

- In the cloud server case, to https://login.paa.juniper.net using:
  - Test Agent Appliance: TCP port 443
  - Test Agent Application: TCP port 6800
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.7.2 Perform your first remote packet capture

Once the Test Agent has connected to your Routing Active Testing account, it will show up in the Test Agents view. Read more *here* (page 479) on how to perform your first remote packet capture.

2.8 Getting started with network performance measurements

2.8.1 Download Test Agents

In this step we will download, install, and configure Test Agents. Since network performance measurements are made between Test Agents, you need to download and install two of them. Test Agents include active traffic generators for generating traffic within your network.

- Under Test Agents in the Routing Active Testing user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for Test Agents given *here* (page 70).

By default, once connected and powered up, all Test Agents will communicate directly with the Routing Active Testing server over an encrypted connection:

- In the cloud server case, to https://login.paa.juniper.net using:
  - Test Agent Appliance: TCP port 443
  - Test Agent Application: TCP port 6800
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.8.2 Connect the Test Agents to your network and start your first test

Once you have downloaded and installed two Test Agents and registered them to your Routing Active Testing account, you are ready to start testing. The Test Agents should appear in the Test Agents view.

Follow the instructions below to start a simple TCP throughput test.

1. Log in to your account in the Routing Active Testing user interface.
2. On the left-side bar, click the Tests button and select New Test Sequence.
3. Select the TCP/UDP performance task category.
4. Select the TCP test.
5. Under Server, select one of the Test Agents you have installed.
6. Under Clients, select the other Test Agent.
7. Give the test a name, and click Start.

You should see measurement results within just a few seconds.

Note: The connection will be initiated from the Test Agent you selected as client to the Test Agent selected as server. Therefore, if one agent is placed behind a NAT router, make sure you select that one as client.

To learn more, read the page *Introduction to tests and monitors* (page 252).

A full treatment of network performance testing is found on the following pages:

- *TCP* (page 291)
- *Multisession TCP* (page 294)
- *UDP* (page 299)
- *VoIP UDP* (page 303)

- *Multicast UDP* (page 305)
- *TCP throughput test according to RFC 6349* (page 312)
- *QoS policy profiling* (page 316)

2.9 Getting started with your own Speedtest server

2.9.1 Download a Test Agent

In this step we will download, install, and configure a Test Agent.

- Under Test Agents in the Routing Active Testing user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for Test Agents given *here* (page 70).

By default, once connected and powered up, all Test Agents will communicate directly with the Routing Active Testing server over an encrypted connection:

- In the cloud server case, to https://login.paa.juniper.net using:
  - Test Agent Appliance: TCP port 443
  - Test Agent Application: TCP port 6800
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.9.2 Activate the Speedtest responder

Once the Test Agent has connected to your Routing Active Testing account, it will appear in the Test Agents view.

- Click the Test Agent and select the Applications tab.
- Enable the Speedtest option. Read more *here* (page 38).

2.9.3 Run your first Speedtest

Once the Speedtest application has been activated, you are ready to start testing. Follow the instructions below.

1. Log in to your account in the Routing Active Testing user interface.
2. Click Apps.
3. Select Speedtest.
4. Click Go to public Speedtest page in the top right corner.
5. Run a test from your browser towards your Test Agent.
6. The result is stored in your Routing Active Testing account. To see the result, go to Apps > Speedtest.

Full details are found *here* (page 482).

2.10 Getting started with IP telephony (SIP and VoIP) measurements

To get started with SIP-based IP telephony measurements, please follow these simple steps.

2.10.1 Download Test Agents

In this step we will download, install, and configure Test Agents. Since SIP-based calls are made between Test Agents, you need to download and install two of them. Test Agents have built-in SIP user agents that can be used to initiate and receive SIP-based phone calls.

- Under Test Agents in the Routing Active Testing user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for Test Agents given *here* (page 70).

By default, once connected and powered up, all Test Agents will communicate directly with the Routing Active Testing server over an encrypted connection:

- In the cloud server case, to https://login.paa.juniper.net using:
  - Test Agent Appliance: TCP port 443
  - Test Agent Application: TCP port 6800
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.10.2 Configure SIP test accounts

The next step is to configure SIP accounts for use in the tests.

- From the main menu, navigate to Account > SIP accounts, and add at least two SIP accounts according to this guide: *Setting up SIP accounts* (page 22).

2.10.3 Connect the Test Agents to your network and start your first test

Once you have downloaded and installed two Test Agents and registered them to your Routing Active Testing account, and you have your SIP accounts set up, you are ready to start testing. The Test Agents should appear in the Test Agents view.

Follow the instructions below to start a simple SIP-based test.

1. Log in to your account in the Routing Active Testing user interface.
2. On the left-side bar, click the Tests button and select New Test Sequence.
3. Select the SIP task category.
4. Select the SIP test.
5. Under Hub, select one of the Test Agents you have installed, and select a SIP account to use.
6. Under Clients, select the other Test Agent, and select another SIP account.
7. Give the test a name, and click Start.
8. Done; you should see measurement results within a few seconds.

To learn more, read the page *Introduction to tests and monitors* (page 252).

A full treatment of SIP testing is found on the following page:

- *SIP* (page 348)

3 Managing your account

3.1 Introduction to accounts

During installation of Routing Active Testing, an *account* will have been created as well as one or several *users* which all belong to that account (multitenancy). In order to use Routing Active Testing, you log in to your account as one of these users. When a Test Agent registers with the system, it is registered to an account and becomes available to users in that account.

The account holds a large number of settings and assets that are common to all users in the account. These include SLA thresholds, alarms, and inventory items needed for certain measurements (for example, TWAMP reflectors). For details, explore the remaining pages in this chapter.

3.2 Changing account settings

3.2.1 How to change or reset your Routing Active Testing user password

To change or reset your user password, log in to Routing Active Testing, do one of the following:

- Access your personal profile by clicking your Routing Active Testing user name on the top bar, then click the Change password button.
- Alternatively, on the login screen, click the Forgot your password? link. You will receive an email with a link for changing your password.

Note: If your user is managed by an LDAP server, its password too is managed on that server and cannot be changed here. See the Installation Guide for further details on configuring LDAP user management in Control Center.

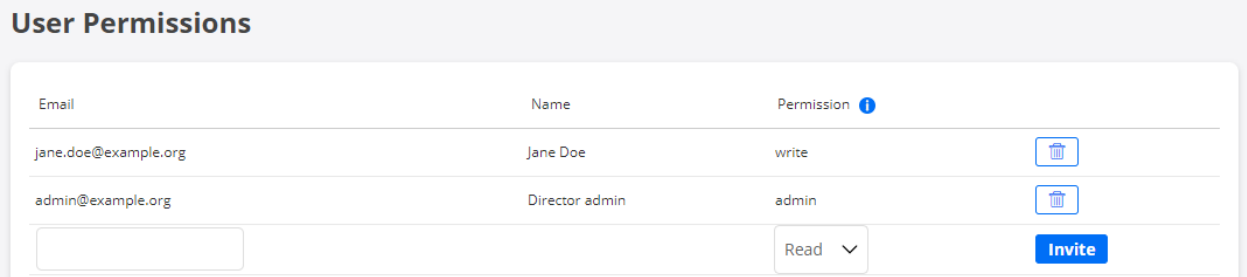
3.2.2 How to update the contact information for your Routing Active Testing account

Your contact information is part of your Routing Active Testing profile.

To update your profile, click your Routing Active Testing user name on the top bar, then select Edit profile and make the desired changes.

3.3 Administering users and permissions

Under Account > Permissions you manage users associated with a Routing Active Testing account, and their privileges. You can invite new users, remove users, and change permissions for each user individually.



To invite a new user, enter the user's email address and the permissions the user should have, then click the Invite button. Routing Active Testing will now send a welcome email to the address given.

The permission levels are *Admin*, *Write*, and *Read*. Over and above this, there are also *Test Agent registration* users which are only permitted to download and register Test Agents and nothing else. Such users are managed on a different page: read more *here* (page 20).

The permission hierarchy is *Admin* > *Write* > *Read*. Anything granted at one permission level is also granted at higher permission levels.

3.3.1 User permissions required for Routing Active Testing operations

All *viewing* of elements in the GUI can be done with **Read** permissions.

Read permissions are also sufficient for a few other operations, as detailed below.

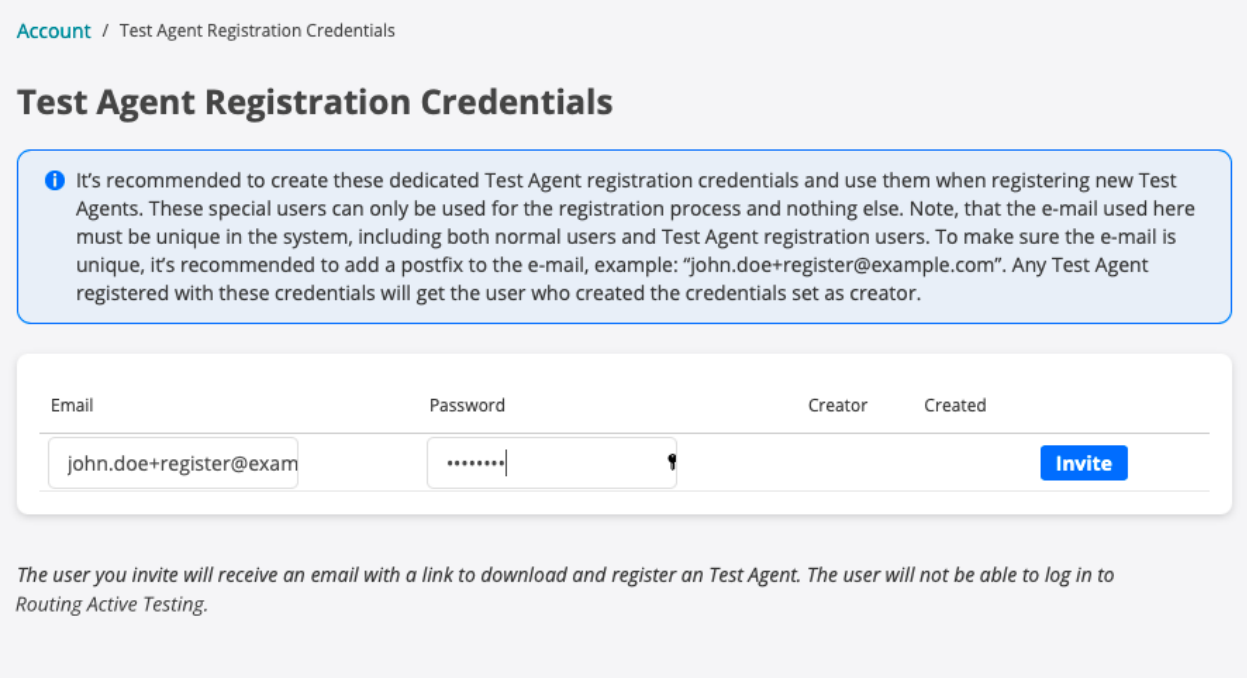
For other operations that affect the state of Routing Active Testing (edit, start/stop, delete, etc. in various contexts), **Write** permissions are required for the most part. However, some operations require **Admin** permissions. The following applies:

- *Users*
  - Edit profile, change password for user currently logged in: **Read**
  - Invite new users and set their permissions: **Admin**
- *Account setup*
  - SIP channels, Y.1731 MEPs, IPTV channels, TWAMP reflectors: **Write**
- *Test Agents*
  - Register Test Agent: **Admin** or **Register only**
  - Assign license to Test Agent: **Read**
  - Release license from Test Agent: **Admin**
  - Edit Test Agent properties: **Write**
  - Share Test Agents: **Admin**
  - Update Test Agent software: **Admin**
- *Tests*
  - Create, start, stop, rerun, delete tests: **Write**
  - Work with templates: **Write**
  - Share tests: **Admin**
- *Monitors*
  - Create, edit, start, stop, delete monitors: **Write**

  - Work with templates: **Write**
  - Share monitors: **Admin**
- *Applications*
  - Packet capture: **Write**
  - Speedtest: **Write**
- *Favorites* (add, remove): **Write**
- *Share measurement results:* **Admin**
- *Reports* (create, edit, delete): **Write**
- *Tags* (add to item, remove from item, delete): **Write**
- *Alarms* (all operations): **Write**

3.4 Administering Test Agent registration users

Under Account > Test Agent Registration users you manage special users that can only be used to download and register a new Test Agent. On this page you can invite new Test Agent registration users and remove such users.



We recommend creating these dedicated users for the purpose of registering new Test Agents.

- To do so, enter an email address and a password for the user, then click the Invite button.

Note that the email address must be unique in the Routing Active Testing system among Test Agent registration users as well as normal users. To make sure the email address is unique, it is a good idea to add a postfix to the address, as in "john.doe+register@example.com". Any Test Agent registered with these credentials will have the user who created the credentials set as creator.

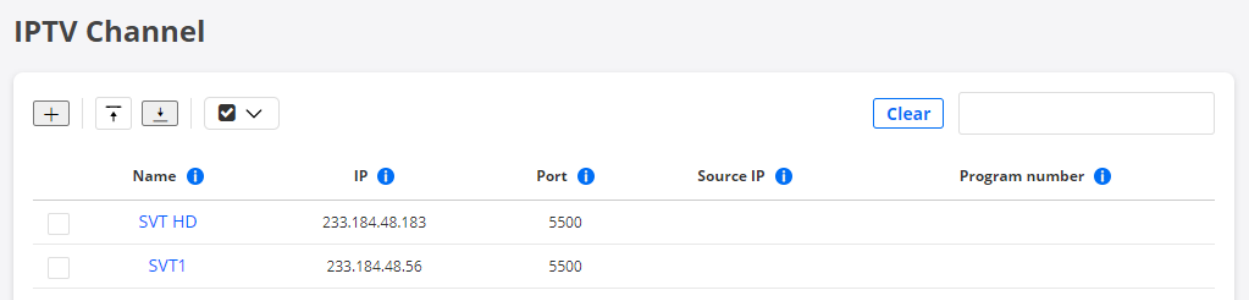
Note also that the person sharing the registration user needs to communicate the password to the recipient separately. For security reasons, the password is not sent in the email.

- To remove a Test Agent registration user, click the trash can button next to the user.

3.5 Setting up IPTV channels

Configuration of IPTV channels is done under Account > IPTV channels. Here you define the IPTV channels available in your network and their corresponding multicast addresses. All channels in the resulting channel plan can then be selected in the user interface when you set up an IPTV task.

- To add an IPTV channel manually, click the Add button.



The following parameters are *mandatory*:

- Name: IPTV channel name. Shown when starting IPTV measurements.
- IP: The IPv4 or IPv6 multicast address of the IPTV channel.
- Port: The UDP destination port used in the IPTV multicast stream. Routing Active Testing will filter on this port when measuring.

The remaining parameters are *optional*:

- Source IP: If a multicast source address is specified, IGMPv3 (for IPv4) or MLDv2 (for IPv6) will be used for joining the channels.
- Program number: This field is used when a transport stream contains more than one TV channel (program). A transport stream with more than one program is referred to as an MPTS (Multi Program Transport Stream). The program number points to one of the channels in the MPTS stream. Therefore, if you need to monitor several channels in the MPTS, you will typically add several channels with the same multicast address, but with different program numbers.

3.5.1 Uploading your channel plan as a CSV file

Alternatively, instead of entering IPTV channels manually in the user interface, you can upload a list of channels as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one channel per line, defined according to the following syntax:

```
channel_name, multicast_ip, udp_port, source_ip, program_number
```

The **source_ip** and **program_number** fields are optional.

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

> **Warning:** Uploading a new channel plan will overwrite all existing channels. Items not included in the new uploaded list will be removed, which will cause all current and historical measurement data for those IPTV channels to be deleted.

3.5.2 Exporting channels to a CSV file

You can export the list of defined IPTV channels to a CSV file.

- Click the down-arrow button at the top of the view.



A CSV file is now exported to the Downloads folder on your machine.

The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.6 Setting up SIP accounts

Configuration of SIP accounts is done under Account > SIP accounts.

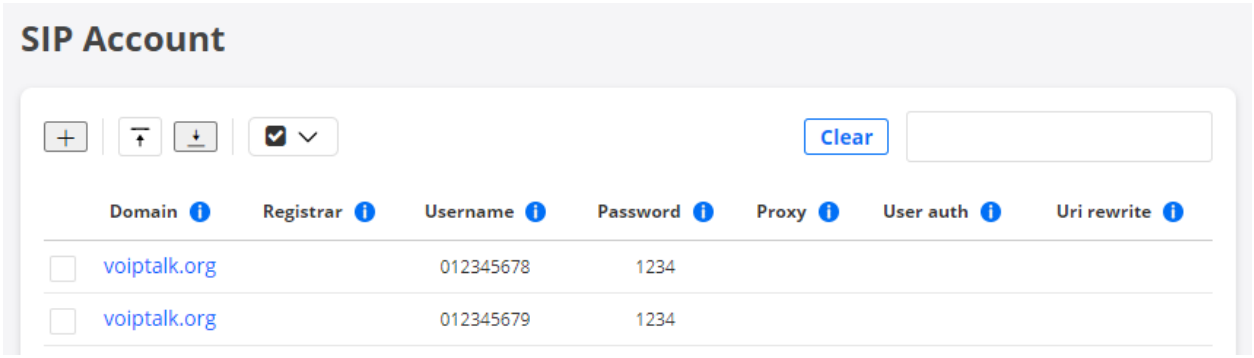
SIP testing is performed by the user-agent on a Test Agent placing SIP calls to other Test Agents and measuring call setup time (among other parameters). When the call is terminated, the call hangup time is measured.

Read more about SIP testing *here* (page 347).

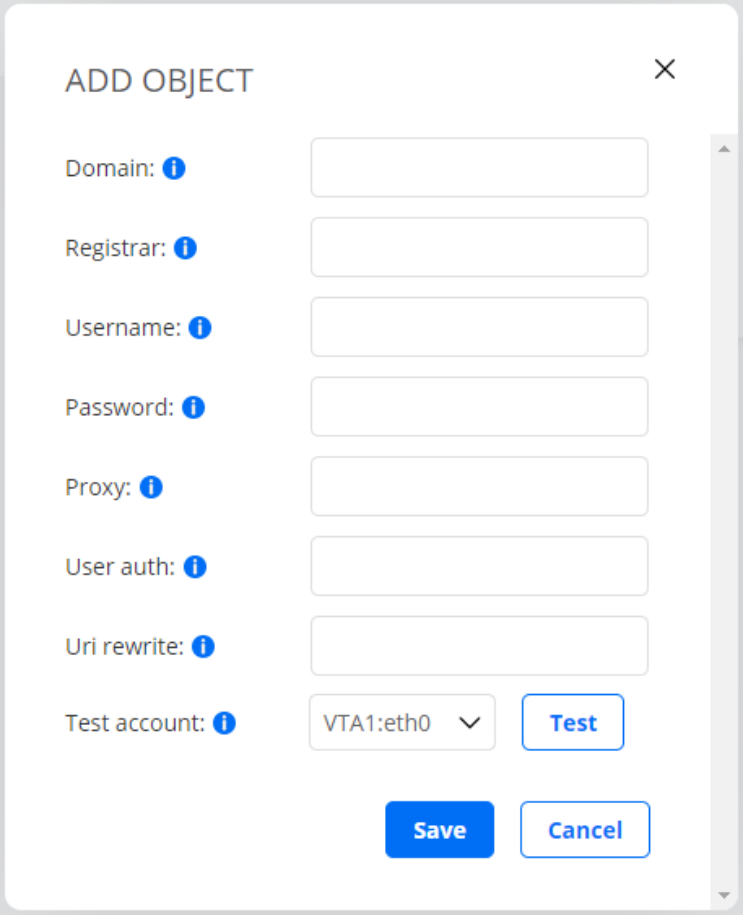
To do this testing, you need to have at least two SIP accounts registered. The SIP account used for a particular Test Agent is selected when the test or monitor is started, so each SIP account can be used for any Test Agent.

The SIP tool supports overwritten URIs. This means that the tool handles the situation where a SIP proxy server changes the original SIP URI during a call.

- To set up a new SIP account manually, click the Add button:



- Fill in the account credentials:



- Domain: The name of the SIP domain.
- Registrar: *(Optional)* The domain name or IP address of the SIP registrar. If empty, the domain name will be used.
- User name: The user name needed for registration.
- Password: The password needed for registration.
- Proxy: *(Optional)* The proxy server to use.
- User auth: *(Optional)* The user ID needed for authentication.
- URI rewrite: *(Optional)* The rewritten user ID of a SIP account used as caller (client). Some SIP servers will rewrite the user name part of the caller's URI. In that case the rewritten user name must be specified here, so that the hub Test Agent can correctly identify the incoming call. The format of the input may be either `username` alone or `username@domain`.
- Test account: Here you can test the SIP account. Select a Test Agent interface in the drop-down box, then click the Test button.

After testing the SIP account, click the Save button to save the configuration.

3.6.1 Uploading SIP accounts as a CSV file

Alternatively, instead of entering SIP accounts manually in the user interface, you can upload a list of SIP accounts as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one SIP account per line, defined according to the following syntax:

```
domain, registrar, user_name, password, proxy, user_auth, uri_rewrite
```

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

Warning: Uploading a new SIP account list will overwrite all existing SIP accounts. Items not included in the new uploaded list will be removed, which will cause all current and historical measurement data associated with those SIP accounts to be deleted.

3.6.2 Exporting SIP accounts to a CSV file

You can export the list of defined SIP accounts to a CSV file.

- Click the down-arrow button at the top of the view.



A CSV file is now exported to the Downloads folder on your machine.

The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.7 Setting up Ping hosts

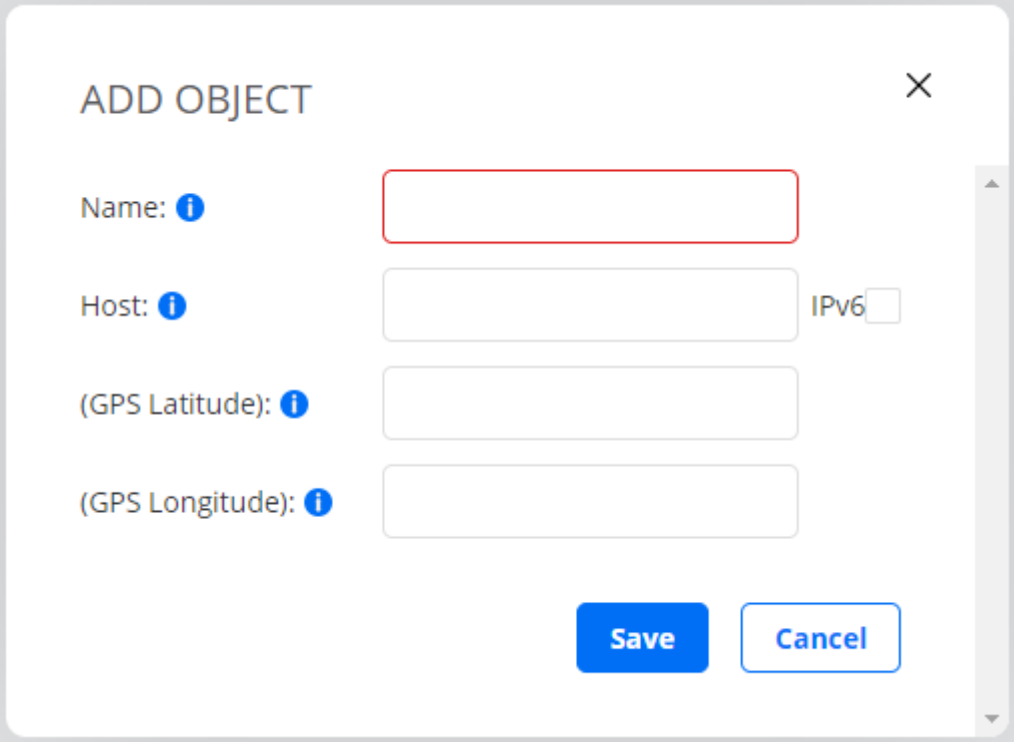
Ping testing requires at least one Ping host, towards which a Test Agent (acting as initiator) sends test traffic. Ping hosts usually reside in third-party equipment; however, Test Agents also have Ping host functionality and can be added to the Routing Active Testing host inventory in the same way.

Adding Ping hosts is done under Account > Ping. Here you define the hosts that are available in your network and their corresponding host names or IP addresses. These Ping hosts are then used as test points in Ping tests and monitors, as described on the page *Ping measurements* (page 430).

- To add a Ping host manually, click the Add button.



This dialog appears:



- Name: The name of the Ping host.
- Host: Host name or IP address of the Ping host. If you enter an IPv6 address, you also need to select the IPv6 checkbox.
- GPS Latitude: *(Optional)* Latitude of Ping host according to WGS 84. Expressed as a decimal number between –85.06 and +85.06, where a negative number means “south”.
- GPS Longitude: *(Optional)* Longitude of Ping host according to WGS 84. Expressed as a decimal number between –180 and +180, where a negative number means “west”.

The GPS coordinates are handled in the same way as for Test Agents; see *this page* (page 196).

Finish by clicking the Save button.

3.7.1 Tagging hosts

You can apply *tags* to Ping hosts in order to identify them as having a specific property or belonging to a specific subset.

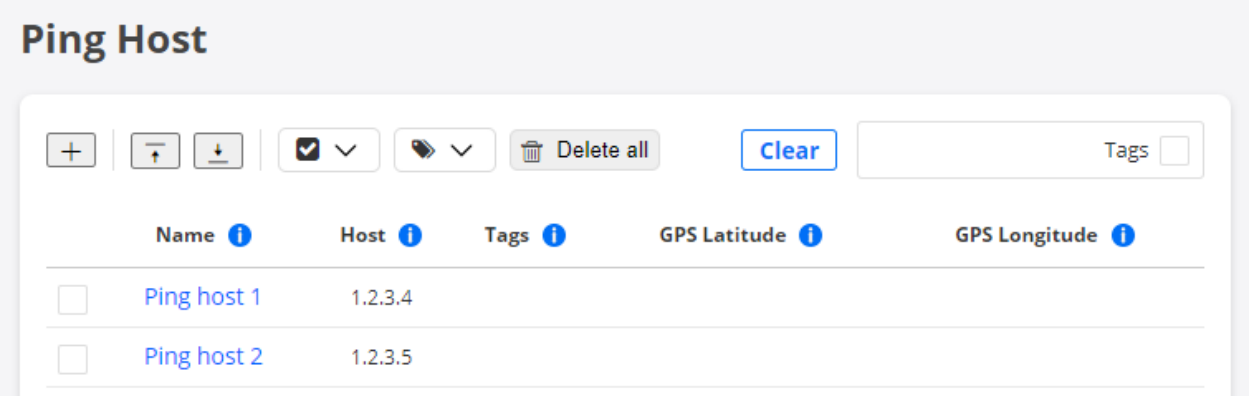
How to work with tags is covered on *this page* (page 283).

In addition, when importing hosts from a CSV file (see *below* (page 26)), you can apply tags to hosts directly in that file, without any user interface actions.

3.7.2 Ping host view

The view shows all Ping hosts known to Routing Active Testing. The information displayed is the same as described above for the "add" dialog, with the addition of this column:

- Tags: Tags assigned to the Ping host.



3.7.3 Importing a list of Ping hosts from a CSV file

Alternatively, instead of entering Ping hosts manually in the user interface, you can upload a list of Ping hosts as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one Ping host per line, defined according to the following syntax:

```
name, host, tags, gps_lat, gps_long
```

The **tags** field is optional. If several tags are given in the **tags** string, they must be separated by commas.

Below is an example CSV file where a single Ping host is defined:

```
"name","host","tags","gps_lat","gps_long"
"Host","192.168.0.1","tag,tag with spaces",65.58,22.15
```

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

- Click the Import button to start the import procedure.

An import dialog appears which also summarizes the requirements on the CSV file syntax.

- Either drag and drop your CSV file to the dashed area in the dialog, or click the Browse button to browse the file system for the file.
- Under Select file encoding, select the encoding used in the CSV file. The UTF-8 format is recommended for a file that contains international characters.

- Then click Import. The Ping Hosts view is now populated with the CSV file content, alongside any Ping hosts previously defined there.

3.7.4 Exporting Ping hosts to a CSV file

You can export the list of defined Ping hosts to a CSV file.

- Click the down-arrow button at the top of the view.



- In the dialog that opens, click Export.

A CSV file is now exported to the Downloads folder on your machine.

The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.7.5 Editing Ping hosts

You can edit a Ping host by clicking its name in the Ping Hosts view or by clicking its "edit" icon on the far right. The dialog that appears is the same as when *adding* (page 24) a Ping host.

3.7.6 Deleting Ping hosts

To delete one or several Ping hosts, do the following:

- Select the box on the far left for each host you wish to delete.
- Click the trash can button that appears at the top of the view.



Warning: This action permanently removes the selected Ping hosts from your account and cannot be undone. All historical results associated with these hosts are also removed. You will be prompted to confirm that this is your intention.

You can delete all Ping hosts by clicking the Delete all button at the top:



Warning: This action permanently removes all Ping hosts from your account and cannot be undone. All historical results associated with all hosts are also removed. You will be prompted to confirm that this is your intention.

3.8 Setting up TWAMP reflectors

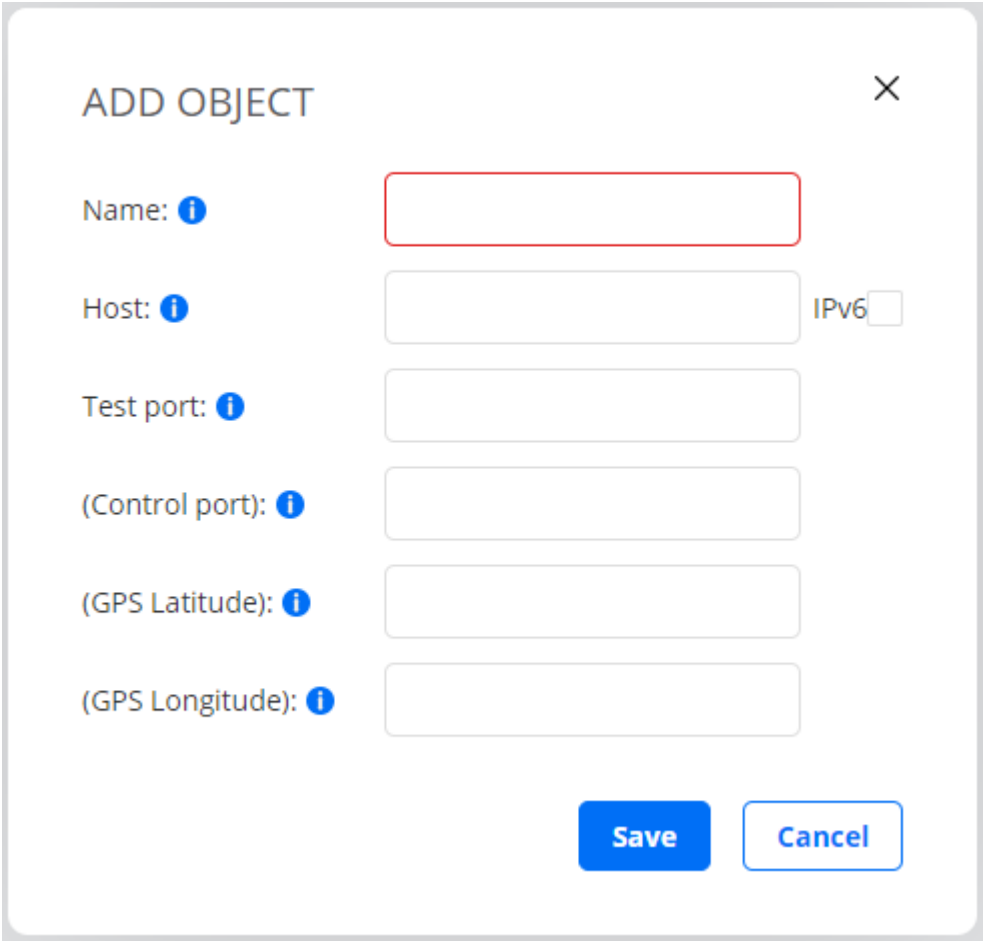
TWAMP testing requires at least one TWAMP reflector, towards which a Test Agent (acting as initiator) sends test traffic. TWAMP reflectors usually reside in third-party equipment; however, Test Agents also have TWAMP reflector functionality and can be added to the Routing Active Testing reflector inventory in the same way.

Adding TWAMP reflectors is done under Account > TWAMP. Here you define the reflectors that are available in your network and their corresponding host names or IP addresses. These TWAMP reflectors are then used as test points in TWAMP tests and monitors, as described on the page *TWAMP measurements* (page 416).

- To add a TWAMP reflector manually, click the Add button.



This dialog appears:



- Name: The name of the TWAMP reflector.
- Host: Host name or IP address of the TWAMP reflector. If you enter an IPv6 address, you also need to select the IPv6 checkbox.
- Test port: UDP port number for the TWAMP reflector. Note: A TWAMP task engaging this reflector may in fact end up using a different port, as explained *here* (page 418).
- Control port: *(Optional)* Control port number for the TWAMP reflector. Leave empty to disable the control protocol (i.e. to use TWAMP Light). Not applicable for Test Agent TWAMP reflectors.

- Source port: *(Optional)* If you specify this port, the TWAMP sender will use it as source port in the test session. If you leave this blank, a random unused port will be picked as source port.
- GPS Latitude: *(Optional)* Latitude of TWAMP reflector according to WGS 84. Expressed as a decimal number between –85.06 and +85.06, where a negative number means “south”.
- GPS Longitude: *(Optional)* Longitude of TWAMP reflector according to WGS 84. Expressed as a decimal number between –180 and +180, where a negative number means “west”.

The GPS coordinates are handled in the same way as for Test Agents; see *this page* (page 196).

Finish by clicking the Save button.

3.8.1 Tagging reflectors

You can apply *tags* to TWAMP reflectors in order to identify them as having a specific property or belonging to a specific subset.

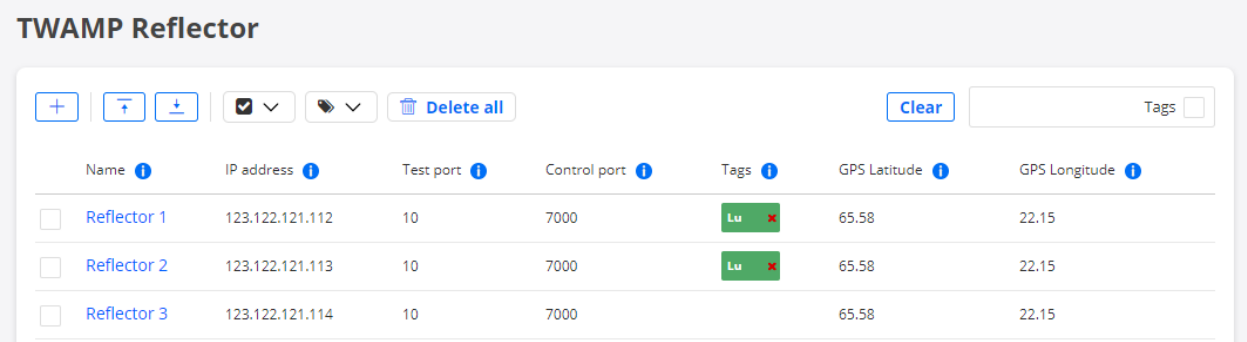
How to work with tags is covered on *this page* (page 283).

In addition, when importing reflectors from a CSV file (see *below* (page 29)), you can apply tags to reflectors directly in that file, without any user interface actions.

3.8.2 TWAMP reflectors view

The view shows all TWAMP reflectors known to Routing Active Testing. The information displayed is the same as described above for the “add” dialog, with the addition of this column:

- Tags: Tags assigned to the TWAMP reflector.



3.8.3 Importing a list of TWAMP reflectors from a CSV file

Alternatively, instead of entering TWAMP reflectors manually in the user interface, you can upload a list of reflectors as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one TWAMP reflector per line, defined according to the following syntax:

```
name, host, port, ctrl_port, src_port, tags, gps_lat, gps_long
```

The **tags** field is optional. If several tags are given in the **tags** string, they must be separated by commas.

Below is an example CSV file where a single TWAMP reflector is defined:

```
"name","host","port","ctrl_port","src_port","tags","gps_lat","gps_long"
"Reflector","192.168.0.1",10,7000,,"tag,tag with spaces",65.58,22.15
```

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

- Click the Import button to start the import procedure.

An import dialog appears which also summarizes the requirements on the CSV file syntax.

- Either drag and drop your CSV file to the dashed area in the dialog, or click the Browse button to browse the file system for the file.
- Under Select file encoding, select the encoding used in the CSV file. The UTF-8 format is recommended for a file that contains international characters.
- Then click Import. The TWAMP Reflectors view is now populated with the CSV file content, alongside any TWAMP reflectors previously defined there.

3.8.4 Exporting TWAMP reflectors to a CSV file

You can export the list of defined TWAMP reflectors to a CSV file.

- Click the down-arrow button at the top of the view.



- In the dialog that opens, click Export.

A CSV file is now exported to the Downloads folder on your machine.

The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.8.5 Editing TWAMP reflectors

You can edit a TWAMP reflector by clicking its name in the TWAMP Reflectors view or by clicking its “edit” icon on the far right. The dialog that appears is the same as when *adding* (page 28) a TWAMP reflector.

3.8.6 Deleting TWAMP reflectors

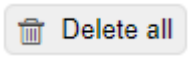
To delete one or several TWAMP reflectors, do the following:

- Select the checkbox on the far left for each reflector you wish to delete.
- Click the trash can button that appears at the top of the view.



Warning: This action permanently removes the selected TWAMP reflectors from your account and cannot be undone. All historical results associated with these reflectors are also removed. You will be prompted to confirm that this is your intention.

You can delete all TWAMP reflectors by clicking the Delete all button at the top:



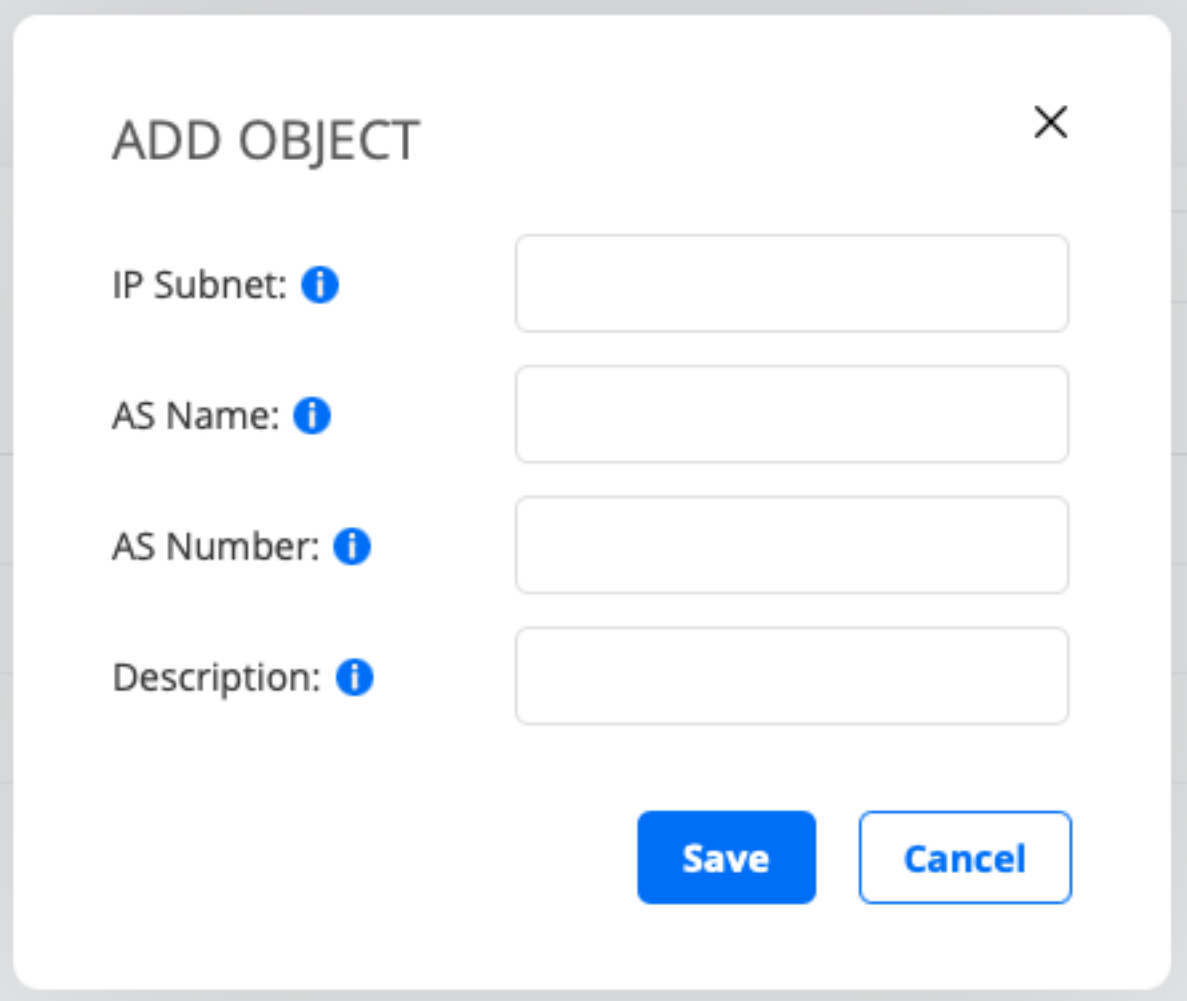
Warning: This action permanently removes all TWAMP reflectors from your account and cannot be undone. All historical results associated with all reflectors are also removed. You will be prompted to confirm that this is your intention.

3.9 Setting up an IP lookup table

Under Account > IP Lookup Table, you can set up a lookup table which translates IP subnets into the names and numbers of autonomous systems (ASes). This information makes it possible to display AS information alongside IP addresses and domain names in the *Path trace* (page 440) presentation.

Note that one AS typically corresponds to many IP subnets.

- To add an IP lookup table entry manually, click the Add button.



- IP subnet: IP subnet to look up: for example, 192.168.0.0/24. Mandatory field.
- AS name: Name of the autonomous system to which the IP subnet belongs. Optional field.
- AS number: Autonomous System Number, ASN. This is a globally unique number associated with the autonomous system. Optional field.
- Description: Plain-text description of the IP subnet. Optional field.

3.9.1 Uploading a IP lookup table as a CSV file

Alternatively, instead of entering the IP lookup table manually in the user interface, you can upload a table as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one channel per line, defined according to the following syntax:

```
ip_subnet, as_name, as_number, description
```

All fields except `ip_subnet` are optional.

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

Warning: Uploading a new IP lookup table will overwrite all existing rows in the table. Items not included in the new uploaded list will be removed.

3.9.2 Exporting the IP lookup table to a CSV file

You can export the IP lookup table to a CSV file.

- Click the down-arrow button at the top of the view.



A CSV file is now exported to the Downloads folder on your machine.

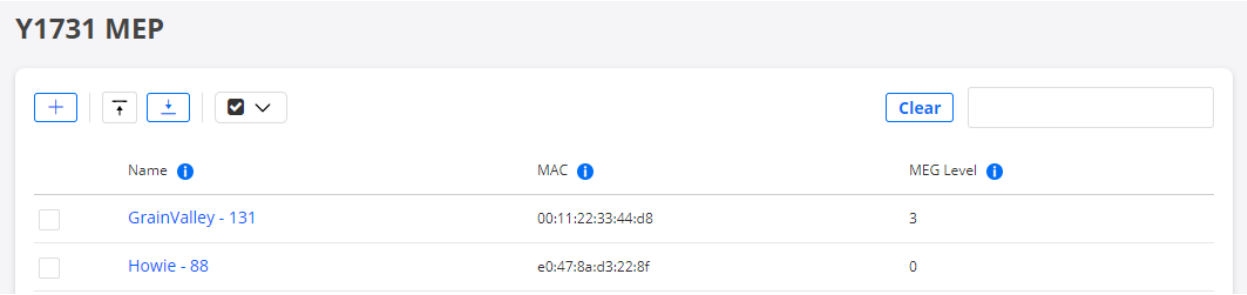
The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.10 Setting up Y.1731 MEPs

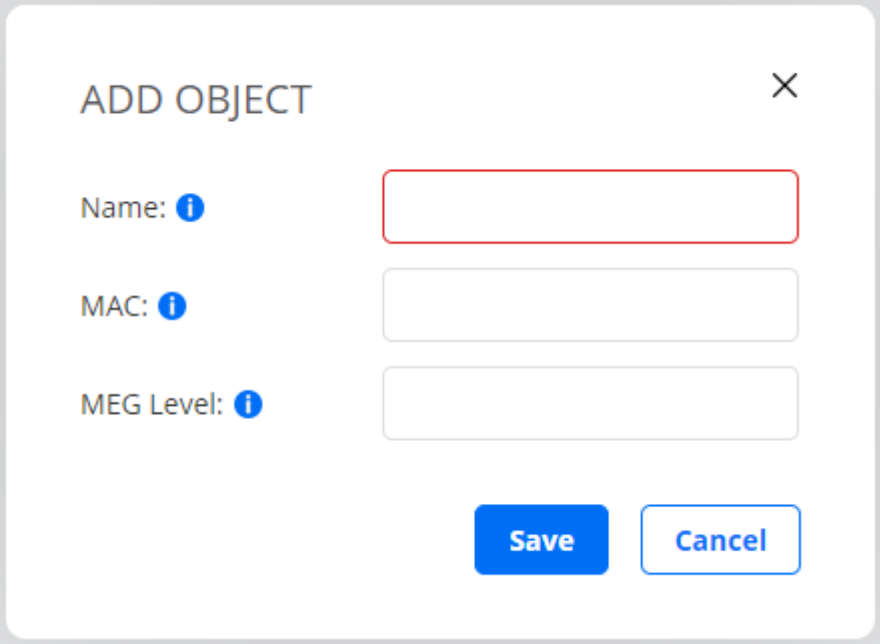
The configuration of ► ITU-T G.8013/Y.1731 MEPs (Maintenance End Points) is done under Account > Y.1731. Here you define the MEPs that are available in your network and their corresponding MAC addresses. Y.1731 testing requires at least one MEP to which a Test Agent can send test traffic.

The MEPs defined here are used as test points in *Y.1731 testing* (page 402).

- To manually set up a new Y.1731 MEP, click the Add button.



This dialog appears:



- Name: The name of the Y.1731 MEP.
- MAC: The MAC address of the Y.1731 MEP.
- MEG Level: The MEG (Maintenance Entity Group) level of the Y.1731 MEP.

Finish by clicking the Save button.

3.10.1 Uploading a Y.1731 MEP list as a CSV file

Alternatively, instead of entering MEPs manually in the user interface, you can upload a list of MEPs as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one MEP per line, defined according to the following syntax:

```
name, mac, meg_level
```

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

Note: Uploading a Y.1731 MEP list will overwrite all existing Y.1731 MEPs in your account. Items that are not included in the new uploaded list will be removed, which will cause all current and historical measurement data related to those Y.1731 MEPs to be deleted.

3.10.2 Exporting Y.1731 MEPs to a CSV file

You can export the list of defined Y.1731 MEPs to a CSV file.

- Click the down-arrow button at the top of the view.



A CSV file is now exported to the Downloads folder on your machine.

The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.10.3 Editing Y.1731 MEPs

You can edit a Y.1731 MEP by clicking its name in the Y.1731 MEP view or by clicking its “edit” icon on the far right. The dialog that appears is the same as when *adding* (page 33) a Y.1731 MEP.

3.10.4 Deleting Y.1731 MEPs

To delete one or several Y.1731 MEPs, do the following:

- Select the box on the far left for each MEP you wish to delete.
- Click the trash can button that appears at the top of the view.



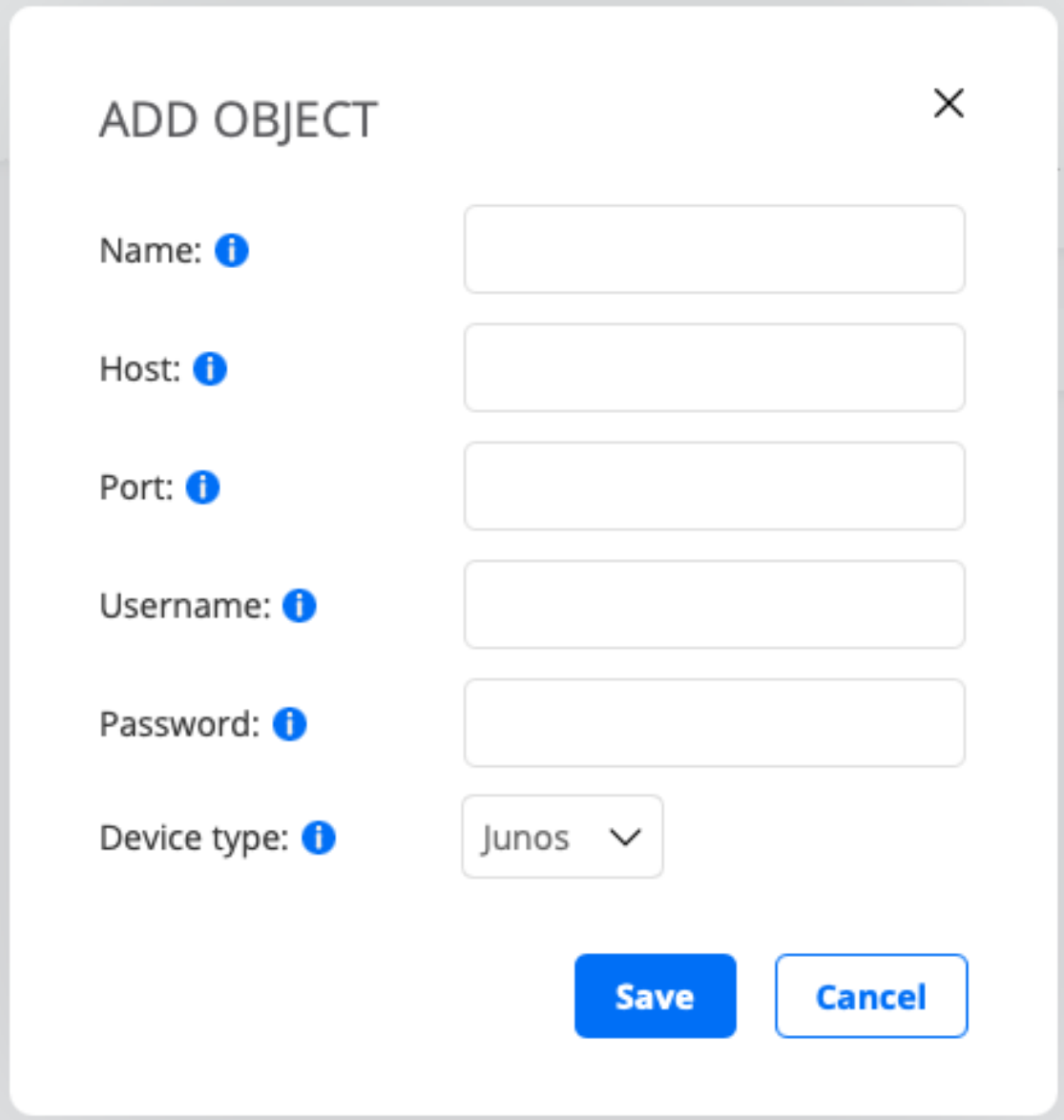
Warning: This action permanently removes the selected Y.1731 MEPs from your account and cannot be undone. All historical results associated with these MEPs are also removed. You will be prompted to confirm that this is your intention.

3.11 Setting up network devices

Network devices are an umbrella term for various devices in a network that Test Agents in Routing Active Testing can interact with. Currently, the only type of network devices supported are Junos devices which are engaged in one of the “Junos” tasks (such as Junos TWAMP or Junos HTTP).

- To add a network device manually, click the Add button.

 This dialog appears:



- Name: The name of the network device.
- Host: The hostname of the network device.
- Port: The port to connect to on the network device.
- Username: Username for logging in to the network device.
- Password: Password for logging in to the network device.

Note: The password cannot be retrieved later on through the Network devices view, nor by performing a *CSV export* (page 37).

- Device type: The type of network device. Currently the only type supported is Junos devices.
- Tags: Tags currently assigned to the network device. See *this section* (page 37).

Finish by clicking the Save button.

3.11.1 Tagging network devices

You can tag a network device in the same way as a monitor. For example, you may want to tag a network device with hardware model or location information. See *this page* (page 283) for detailed instructions.

3.11.2 Network devices view

The Network devices view lists all network devices defined in the system. It displays the configuration parameters entered in the dialog *above* (page 35), along with any tags attached to each device.

3.11.3 Importing a list of network devices from a CSV file

Alternatively, instead of entering network devices manually in the user interface, you can upload a list of network devices as a comma-separated file with extension `.csv`.

- Click the up-arrow button at the top of the view.



- In the dialog that opens, select the desired CSV file.

The CSV file must have one network device per line, defined according to the following syntax:

```
name, hostname, port, username, password, tags
```

The **tags** field is optional. If several tags are given in the **tags** string, they must be separated by commas.

Below is an example CSV file where a single network device is defined:

```
"name","hostname","port","username","password","tags"
"My Network Device","192.168.0.1",830,"johndoe","mypassword","tag,tag with spaces"
```

For full CSV file syntax requirements, see this page.

Please make sure that the file is encoded in UTF-8 format. The encoding can be specified in the text editor settings.

You can edit a network device which is already in the inventory by uploading it in a CSV file with the same name but with modified parameters. If you leave the password blank, the password will remain unchanged.

Note: Uploading a list of network devices will overwrite all existing network devices in your account. Items that are not included in the new uploaded list will be removed, which will cause all current and historical measurement data related to those network devices to be deleted.

3.11.4 Exporting network devices to a CSV file

You can export the list of defined network devices to a CSV file.

- Click the down-arrow button at the top of the view.



A CSV file is now exported to the Downloads folder on your machine.

The exported file will adhere to the specifications given on the CSV file syntax page, and it will be encoded in UTF-8.

3.11.5 Editing network devices

You can edit a network device by clicking its name in the Network devices view. The dialog that appears is the same as when *adding* (page 35) a network device.

3.11.6 Deleting network devices

To delete one or several network devices, do the following:

- Select the box on the far left for each network device you wish to delete.
- Click the trash can button that appears at the top of the view.



Warning: This action permanently removes the selected network devices from your account and cannot be undone. All historical results associated with these network devices are also removed. You will be prompted to confirm that this is your intention.

3.12 Configuring Speedtest

For general information about the Speedtest function in Routing Active Testing and the presentation of Speedtest results, see *this page* (page 482).

Configuration of Speedtest is done under Account > Speedtest. The following settings are available:

3.12.1 General tab

- WebSocket port: HTTP listen port used by the Test Agent Speedtest server for listening to incoming WebSocket connections from the Speedtest web page. Default: 80. Please note that this uses HTTP, not HTTPS, for performance reasons.
- Category label: Label displayed next to the drop-down box which is by default labeled Category on the Apps > Speedtest page as well as on the public Speedtest page. You can change from "Category" to, for example, "Location", "Access type", or "Internet service". You can then configure multiple categories to choose from on the Categories tab (see *Categories tab* (page 39) below).

Note: Be aware that web browsers normally block connections to ports commonly used by other protocols, so that these ports must be avoided. Examples: Ports blocked by Firefox; Ports blocked by Chrome.

- Max parallel tests: The maximum number of parallel Speedtest tests that can be running towards a server. Default: 1. This limit prevents the Test Agent from being overloaded with too many tests at the same time so that low throughput values are obtained. After the maximum number of tests is reached, the server will not accept a new test until some currently running test has finished, and a message will notify the user of this.
- Test duration (s): Duration of the downstream and upstream tests in seconds. Default: 10 s in each direction. Max: 60 s. It makes little sense to set this lower than 10 s, since not enough data would then be gathered to yield a reliable test result.

Warning: Increasing the test duration can potentially cause issues, such as the browser crashing.

- Allow public report: Governs whether to allow users to download a PDF report on the test once it has finished.
- Allow social sharing: Governs whether to allow users to share test results to social networks.
- Show shared Test Agents: Governs whether to allow Test Agents shared to this account to be used for tests. If this setting is disabled, shared Test Agents will be hidden when you start a new Speedtest.
- Show full description: Governs whether to allow users to see the four test steps and how they work on the public Speedtest page. If this option is not selected, users will not see this information.
- IP access filter: Limits the access to the Speedtest service for a specific address range. The access filter is a comma-separated string of IP addresses or address ranges. *Example:* “192.168.1.10-192.168.1.50, 192.168.1.75”. This filter is typically used to make sure that tests are not being run towards your Test Agent by other customers/users not residing within your network.
- Language: Select the user interface language for Speedtest. Default: English.

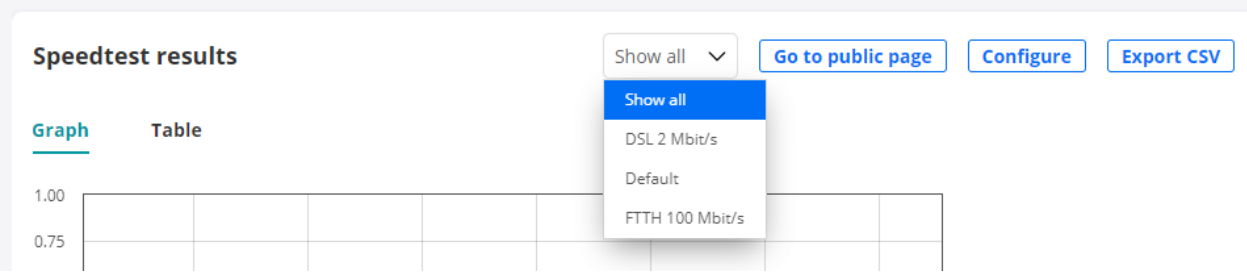
Confirm your settings by clicking the Save config button.

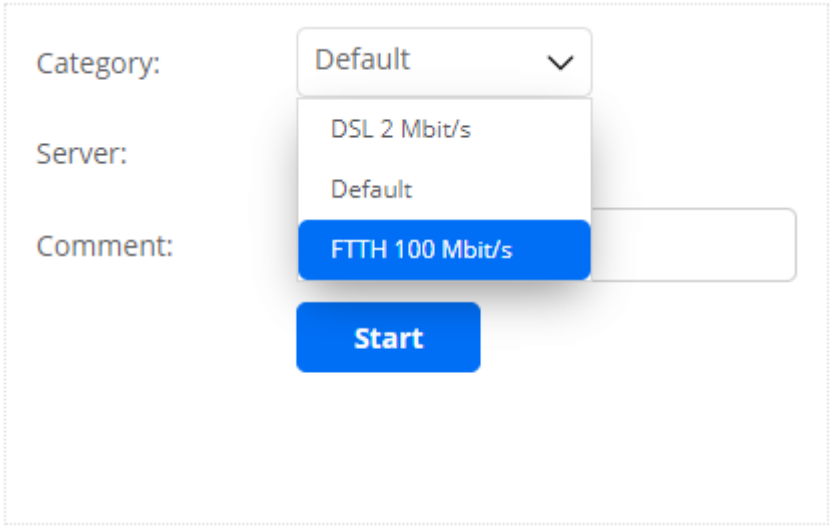
3.12.2 Categories tab

When users perform a test, they can select a category for it.

For example, you might define “Internet service” as category type on the General tab, and then define categories like “DSL 2 Mbit/s” and “FTTH 100 Mbit/s” on the Categories tab. Each category can have different fail thresholds for throughput, loss, and round-trip delay. In this way, users can conveniently indicate what type of Internet service they are using when running their test. Later on, support technicians can then easily review, say, all tests run for “DSL 2 Mbit/s”.

The defined category type and categories are displayed both in the Routing Active Testing Speedtest presentation and on the public Speedtest page:





If no category is selected by the user, the “Default” category will be applied to the test.

3.12.3 Logo tab

You can upload a logotype to display on the Speedtest page that is seen by your users.

Supported image formats are JPEG, PNG, and GIF. The maximum image size is 400 × 200 pixels, and the maximum image file size is 100 kB.

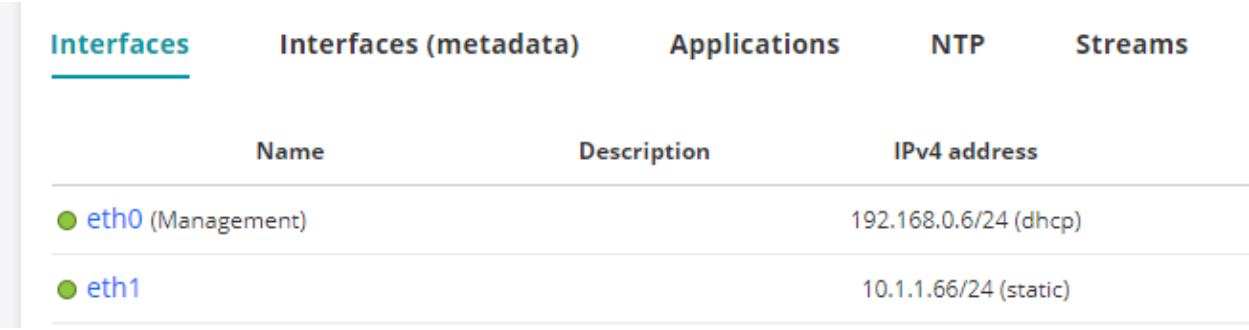
3.12.4 Social logo tab

You can also upload a logotype to display when sharing results to social networks.

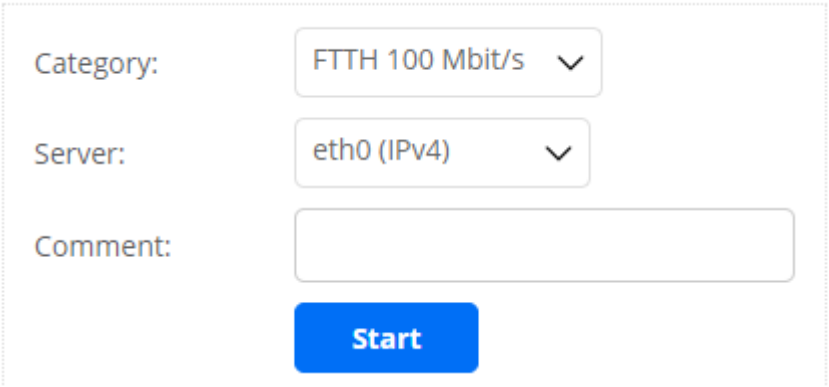
For the image to be displayed correctly it needs to be 250 × 250 pixels, in PNG format, and less than 100 kB in size. Note that social networks might cache old images, and changes made to the logo will then take some time to appear.

3.12.5 Naming Test Agent interfaces taking part in Speedtest

By default, Test Agent interfaces will be displayed on the public Speedtest page in the format `<Test Agent name>:<interface name>`, for example, `MyTestAgent:eth0`. If you want to the interface to appear under a custom name, go to the Test Agent settings, click the interface, and enter the desired name in the Description field for that interface:



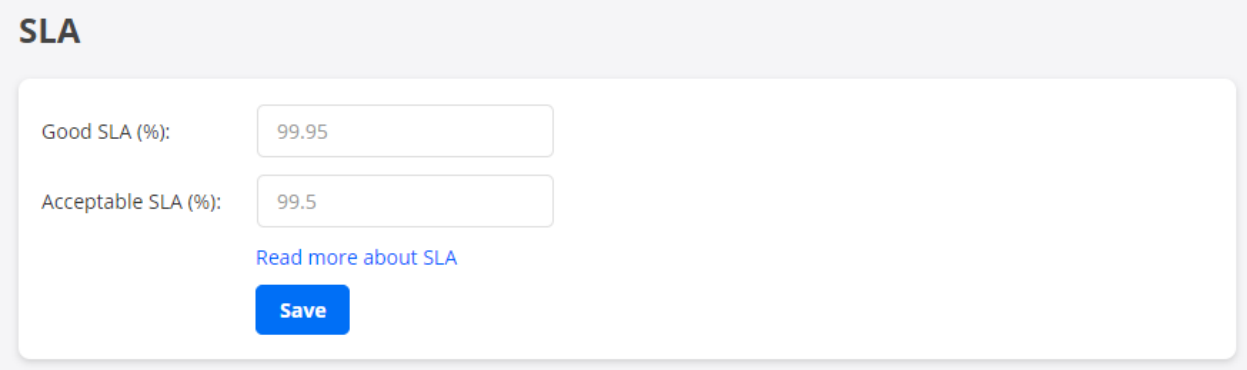
This name will from now appear under Server on the public Speedtest page:



3.13 Configuring SLA (Service Level Agreement) thresholds

Configuration of SLA (Service Level Agreement) thresholds is done under Account > SLA in the main menu. SLAs thresholds are used in Routing Active Testing to give a quick indication of whether the network or service you are monitoring exhibits good, acceptable, or bad performance.

You can set a global default value for SLA thresholds. This value should of course tally with the actual operator SLA or with other agreed SLA levels in order to be a reliable metric.



Note that you can override this default SLA threshold when setting up or editing a monitor.

Good SLA: Threshold for "good" fulfillment of the service level agreement (green SLA icon). By default this requirement is (100 – ES) ≥ 99.95%, or in other words, the percentage of errored seconds must not exceed 0.05%.

Acceptable SLA: Threshold for "acceptable" fulfillment of the service level agreement (orange SLA icon). The default requirement is (100 – ES) ≥ 99.5%, that is, the percentage of errored seconds must not exceed 0.5%.

If the percentage of errored seconds is higher than the Acceptable SLA threshold, the SLA fulfillment is categorized as **Bad** (red SLA icon). With the default settings, this means that (100 – ES) < 99.5%, i.e. ES > 0.5%.

Read more about SLAs *here* (page 478).

3.14 Setting up alarms

Alarm and *SNMP* (page 516) configuration is done under Account > Alarms. Specifically, the items configured are:

- SMNP managers
- Email lists
- Alarm templates
- Automatic alarm suppressions

3.14.1 SNMP Manager settings

Routing Active Testing can send SNMP traps to an SNMP manager whenever the SLA is violated during a monitoring session. Routing Active Testing supports SNMP version 2c as well as the more secure SNMP version 3.

Note: The SNMP traps, regardless of SNMP version, are sent from the host where Control Center is installed (in the SaaS case, from https://app.netrounds.com), by default using the standard port for SNMP traps (UDP port 162). The port can be changed. No traps are sent directly from Test Agents.

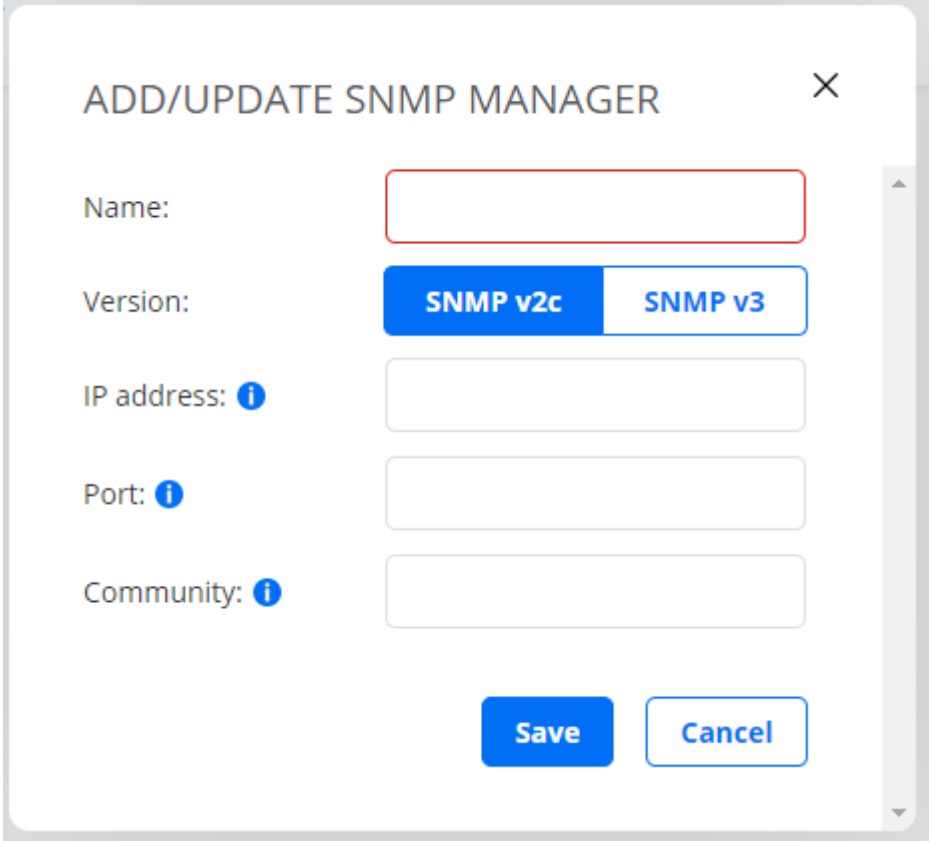
The Routing Active Testing alarm MIB can be downloaded from https://www.juniper.net/documentation/us/en/software/active-assurance/mibs/paragon-active-assurance-mibs.zip.

When *setting up a monitor* (page 263), you select which SNMP manager should receive alarm traps.

Here is how to set up a new SNMP manager:

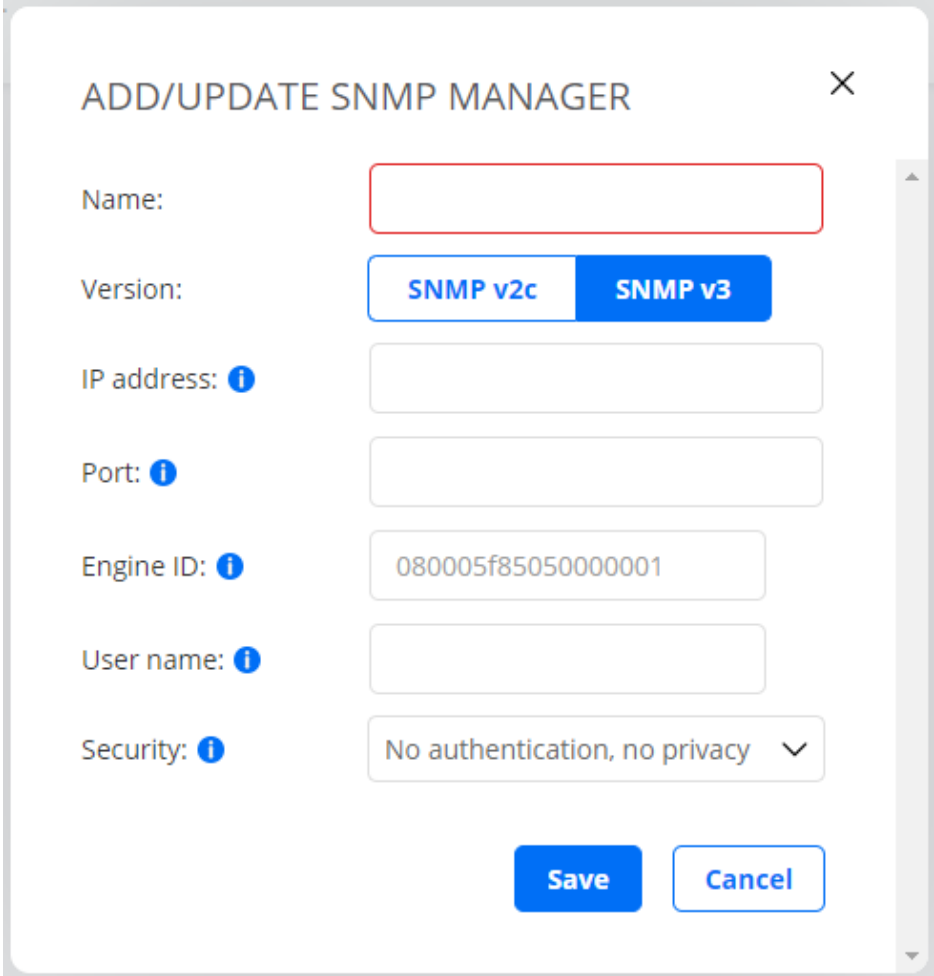
- Select the SNMP Manager tab.
- Click the button Add Manager.
- In the dialog that appears, select SNMP version: v2c or v3.

Further parameters for SNMP version v2c are as follows:



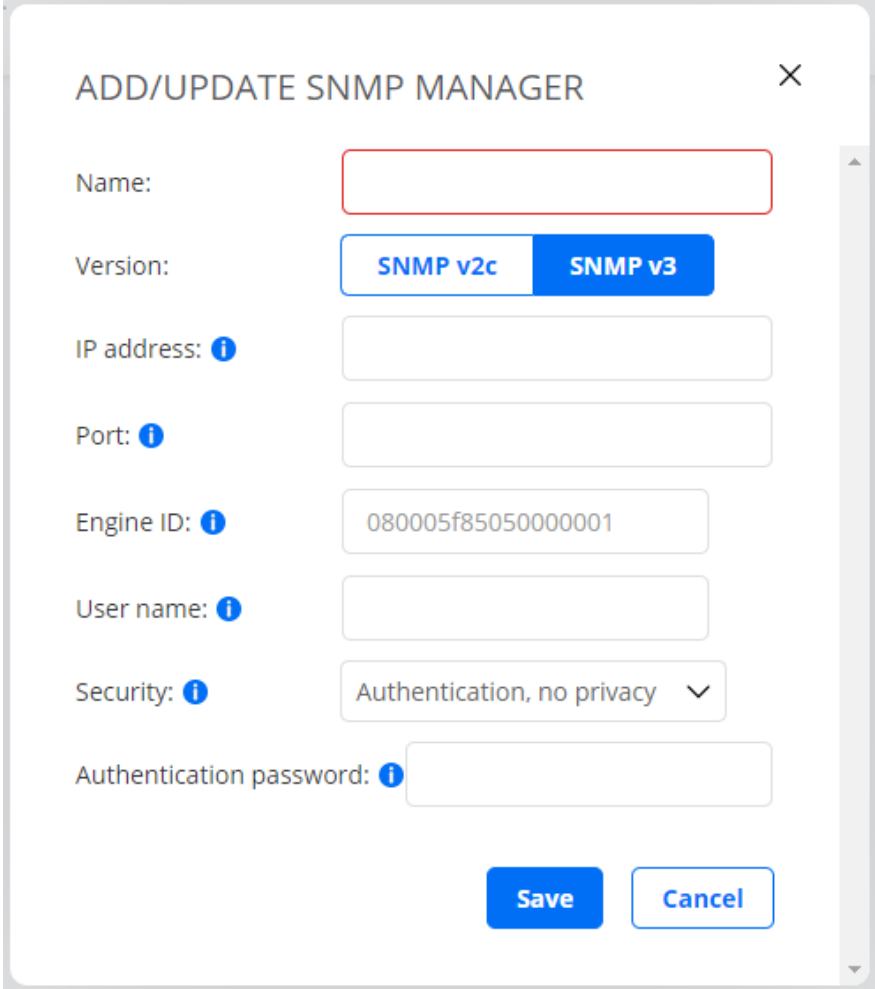
- IP address: The IP address of your SNMP manager (trap sink).
- Port: Port on which to send SNMP traps (trap sink).
- Community: The community string used for authentication.

For SNMP version 3, a couple more parameters need specifying:

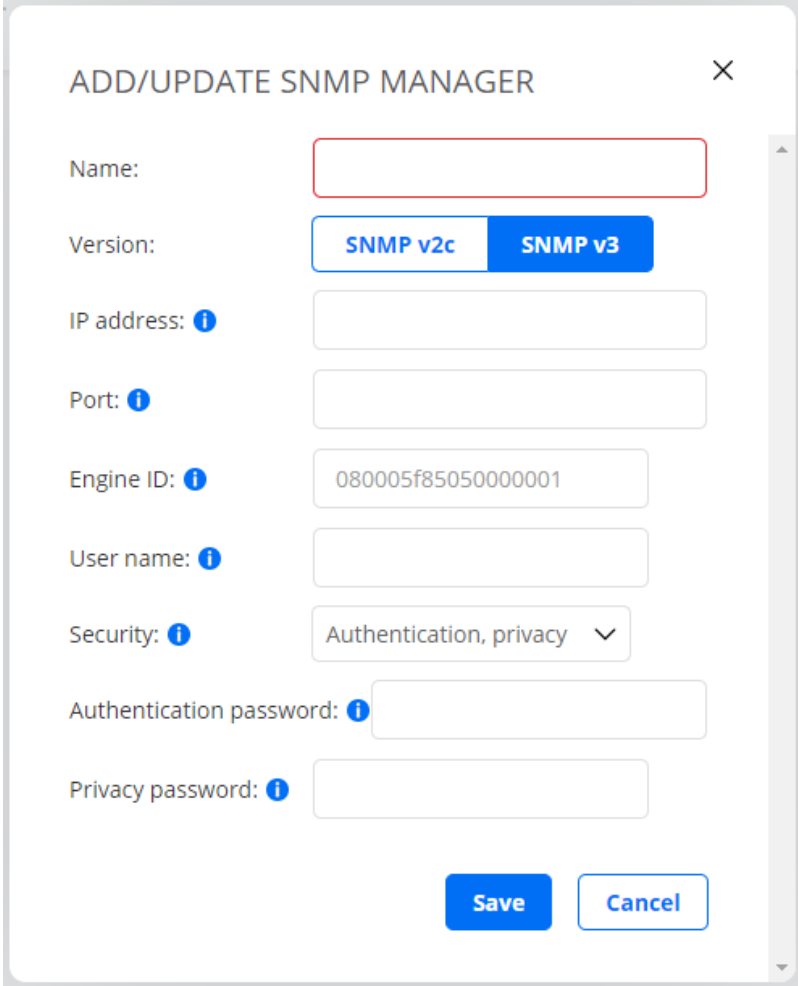


- IP address: The IP address of your SNMP manager (trap sink).
- Port: Port on which to send SNMP traps (trap sink).
- Engine ID: The engine id to use. This identity should be the same in Routing Active Testing and in your SNMP manager.
- User name: The user name to supply for authentication.
- Security: The security level to use when sending traps. SNMPv3 defines three levels of security:
  - *No authentication, no privacy*
  - *Authentication, no privacy*
  - *Authentication, privacy*

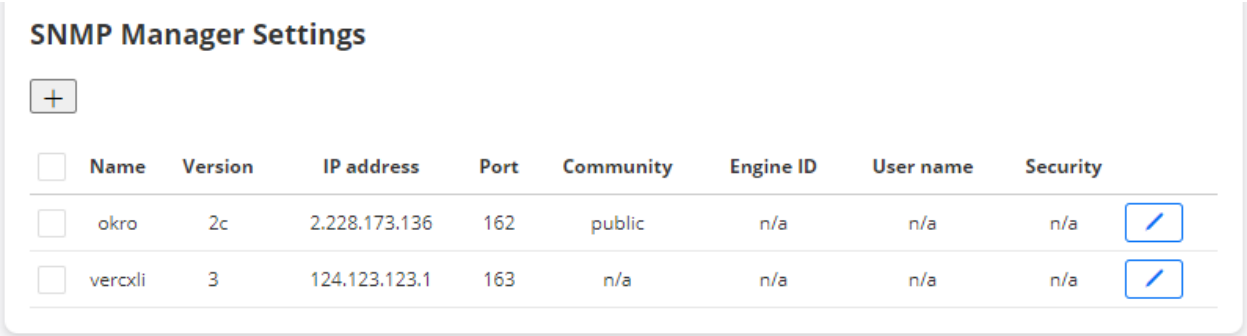
If you select the security level *Authentication, no privacy*, you need to provide a password for authentication. The password must be at least 8 characters in length. The authentication protocol is MD5.



If you select the security level *Authentication, privacy*, you need to provide a password for privacy (= encryption of the SNMP trap). The password must be at least 8 characters in length. The privacy protocol is DES.



The picture below shows an example of SNMP manager settings for the two SNMP versions supported.



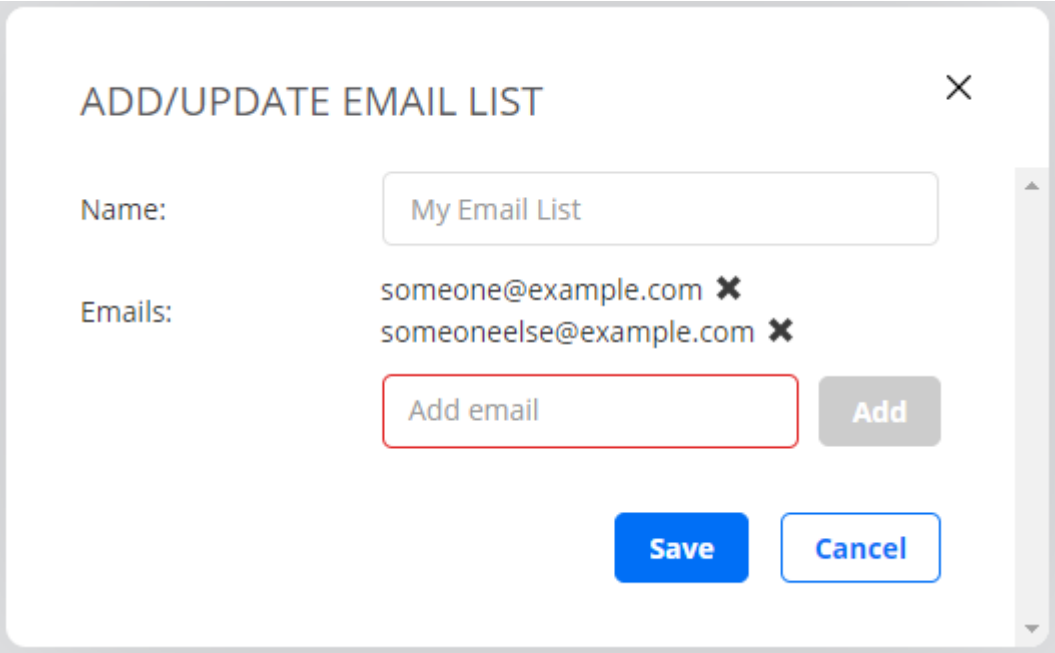
3.14.2 Email settings

As an alternative to SNMP traps, you can have alarms sent by email. You set up an email list in the alarm settings and point to that list when setting up a monitor. An email will then be sent to all email addresses on the list whenever one of the following happens:

- an alarm is raised (some SLA threshold violated), or
- the severity level of the alarm changes, or
- the alarm is cleared (ES level back below SLA threshold).

To set up an email list:

- Select the Email tab.
- Click the button Add Email List.



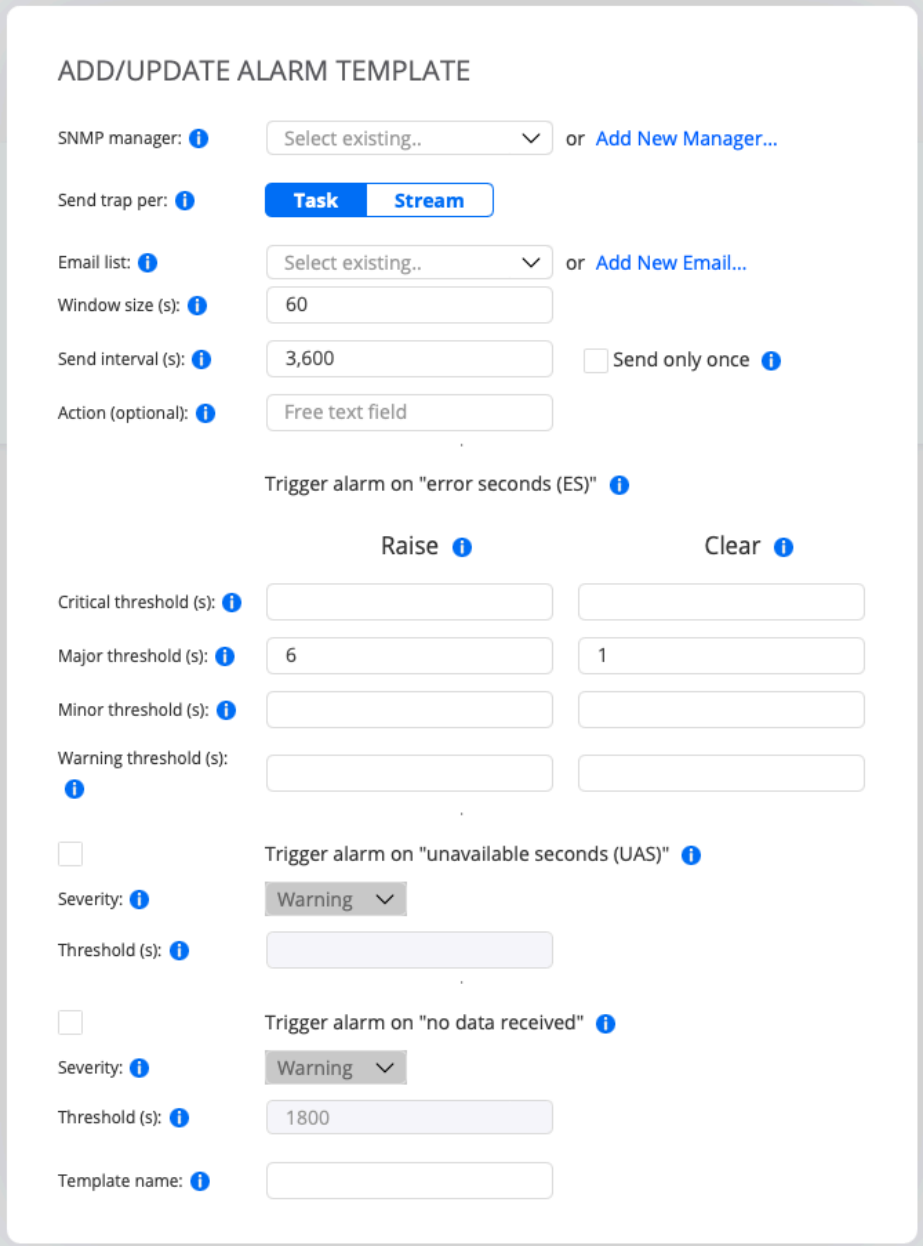
- Name: Enter a name for the email list.
- Emails: Enter one email address at a time, clicking the Add button after each. Click the cross next to an added email address in order to remove it.

3.14.3 Alarm templates

Alarm templates are used to predefine alarm conditions.

To set up such a template:

- Select the Template tab.
- Click the button Add Template and configure the parameters as described below.



- SNMP manager: SNMP manager to send alarms to. If you want to set up a new SNMP manager, read about the settings *here* (page 42).
- Send trap per: Choose whether to send one SNMP trap per monitor task or one per stream in the monitor.
- Email list: Email list to send alarms to. If you want to set up a new email list, read about the settings *here* (page 47).

Note: You can select either an SNMP manager, or an email list, or one of each.

3.14.3.1 Trigger alarm on errored seconds

The thresholds in this section are used to evaluate whether to trigger an alarm for excessive violation of SLA criteria, and if so which severity the alarm should have.

For each severity level, an alarm is *raised* if the number of errored seconds reaches the Raise threshold within the sliding time window, and the alarm is *cleared* if the number of errored seconds drops below the Clear threshold within the window.

- Window size (s): The size of the sliding window used when calculating the errored second count to be compared to the SLA thresholds.
- Send interval (s): The interval between resending of alarms if no change has occurred in severity. Select the Send only once checkbox if you want to send the alarm only once, disabling the Send interval setting. Default: 3600 seconds = 1 hour.
- Send only once: Only send the alarm once instead of at an interval.
- Critical threshold (s): Raise and clear thresholds for alarms with severity "Critical". No defaults.
- Major threshold (s): Raise and clear thresholds for alarms with severity "Major". Defaults: Raise 6 ES, Clear 1 ES.
- Minor threshold (s): Raise and clear thresholds for alarms with severity "Minor". No defaults.
- Warning threshold (s): Raise and clear thresholds for alarms with severity "Warning". No defaults.

Note: If the monitor contains multiple tasks, these criteria are applied to each task separately, not to the monitor as a whole. One task in a monitor can thus trigger an alarm even if no thresholds are exceeded for the other tasks.

Example of alarm triggering: If Window size is set to 60, and the threshold for raising a critical alarm (Critical threshold/Raise) is 10, this means that a critical alarm will be raised if 10 or more errored seconds have occurred during the last 60 seconds.

To avoid frequent on/off toggling of alarms, you should set the Clear threshold lower than the Raise threshold for a given severity (even though it is possible to set them equal).

- Action (optional): This is a free-text field where you can suggest an action to be performed when an alarm is raised. The Action text will be included in each sent email or SNMP trap. One way to use this field is to add a link to a test (for example, `https://<host IP>/<account>/testing/12345`) which the recipient of the alarm should run in order to troubleshoot the problem.

3.14.3.2 Trigger alarm on "unavailable seconds (UAS)"

This section defines an alarm which is triggered when some task reports a specified number of *unavailable seconds (UAS)* (page 477).

This type of alarm is optional; select the checkbox next to the heading to enable it.

- Trigger alarm on "unavailable seconds (UAS)": Enables triggering of an alarm if a given number of unavailable seconds occur. This must be enabled for the settings below to apply.
- Severity: Select a severity for the alarm. One of: "Warning", "Minor", "Major", or "Critical".
- Threshold (s): The length of an unavailability period that should trigger an alarm.

3.14.3.3 Trigger alarm on “no data received”

This section defines an alarm which is triggered when some task in a monitor has not delivered any data for a specified length of time.

This type of alarm is optional; select the checkbox next to the heading to enable it.

- Trigger alarm on “no data received”: Enables triggering of an alarm if no data is received. This must be enabled for the settings below to apply.
- Severity: Select a severity for the alarm. One of: “Warning”, “Minor”, “Major”, or “Critical”.
- Threshold (s): Time to wait for a monitor task to resume delivery of data before raising an alarm. Default: 1800 seconds = 0.5 hours.

3.14.3.4 Alarm template name

- Template name: Enter a name for the alarm template here.

3.14.3.5 Using alarm templates

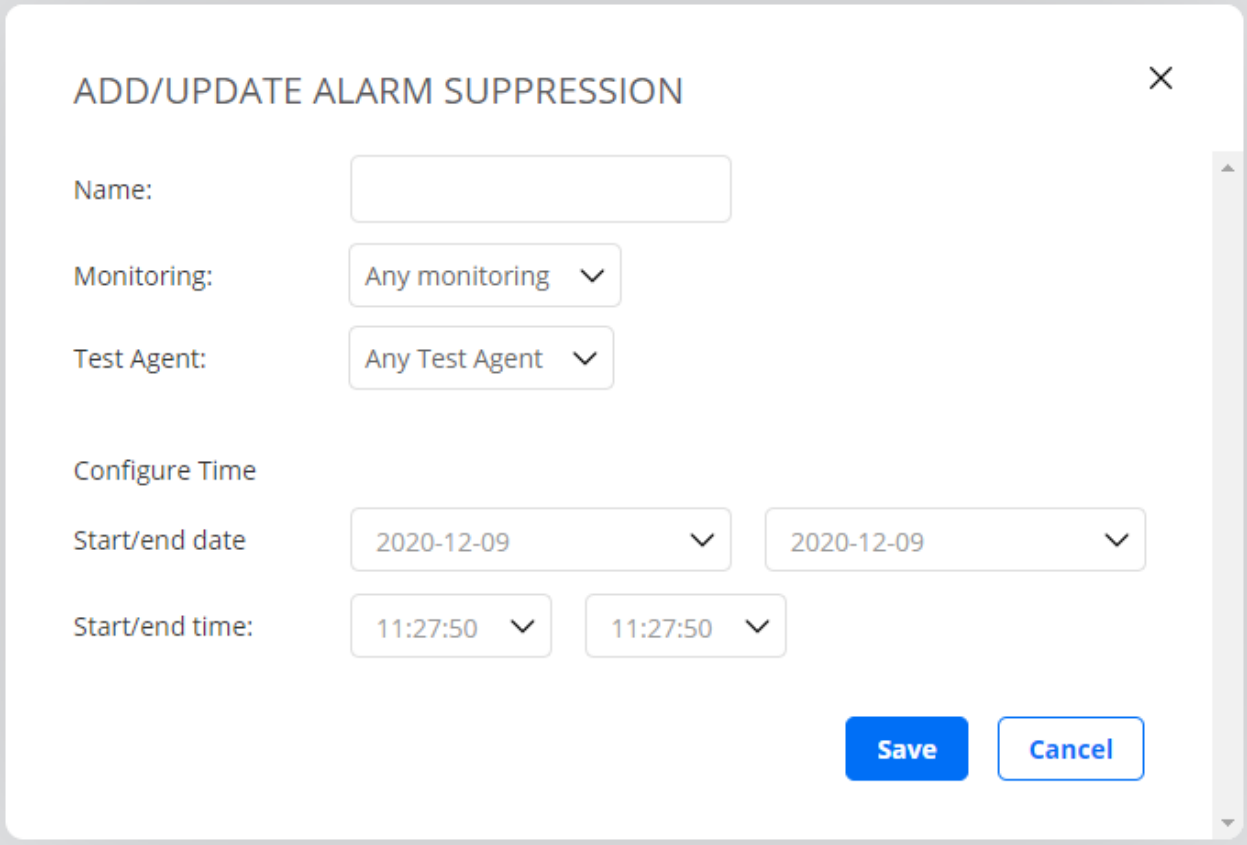
After you have created an alarm template, you can use it to configure alarms when creating a new monitor. This is described *here* (page 488).

3.14.4 Suppressing alarms

It is possible to *suppress* an alarm during a specified time interval. No alarms will then be sent even if thresholds are exceeded. This function is useful if a monitoring session or a Test Agent is temporarily affected by maintenance or other planned work.

To set up an alarm suppression:

- Select the Suppression tab.
- Click the button Add Suppression and configure the parameters as described below.

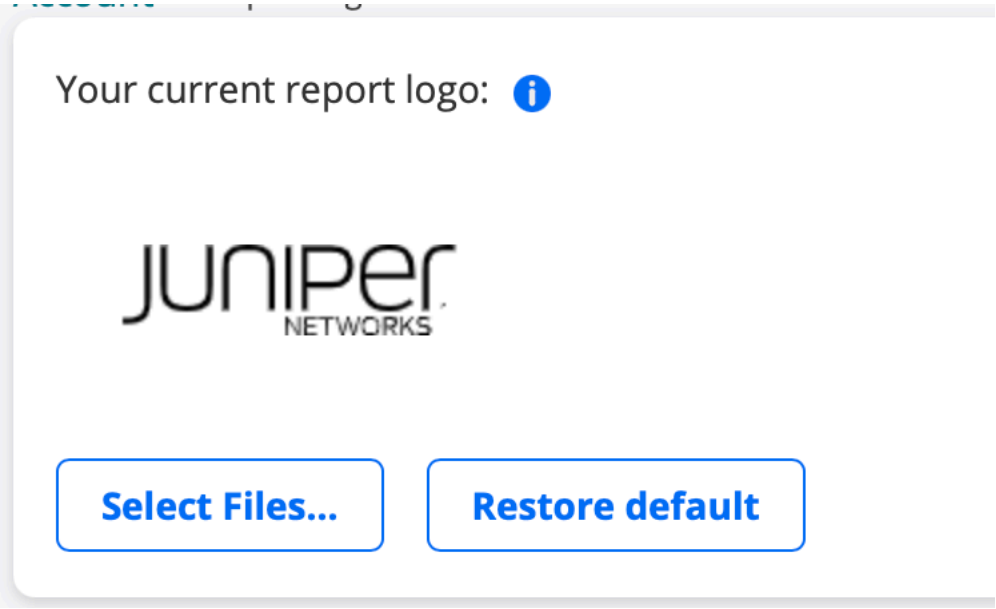


- Name: Enter a name for the alarm suppression.
- Monitor: The monitors the suppression should apply to.
- Test Agent: The Test Agents the suppression should apply to.
- Start/end date, Start/end time: The time interval during which the suppression will be in effect (for example, a maintenance window).

Finish by clicking Save. The suppression is now activated and will suppress alarms during the specified period.

3.15 Changing the report logo

Under Account > Report logo, you can upload your own logotype for displaying in test and monitoring reports.

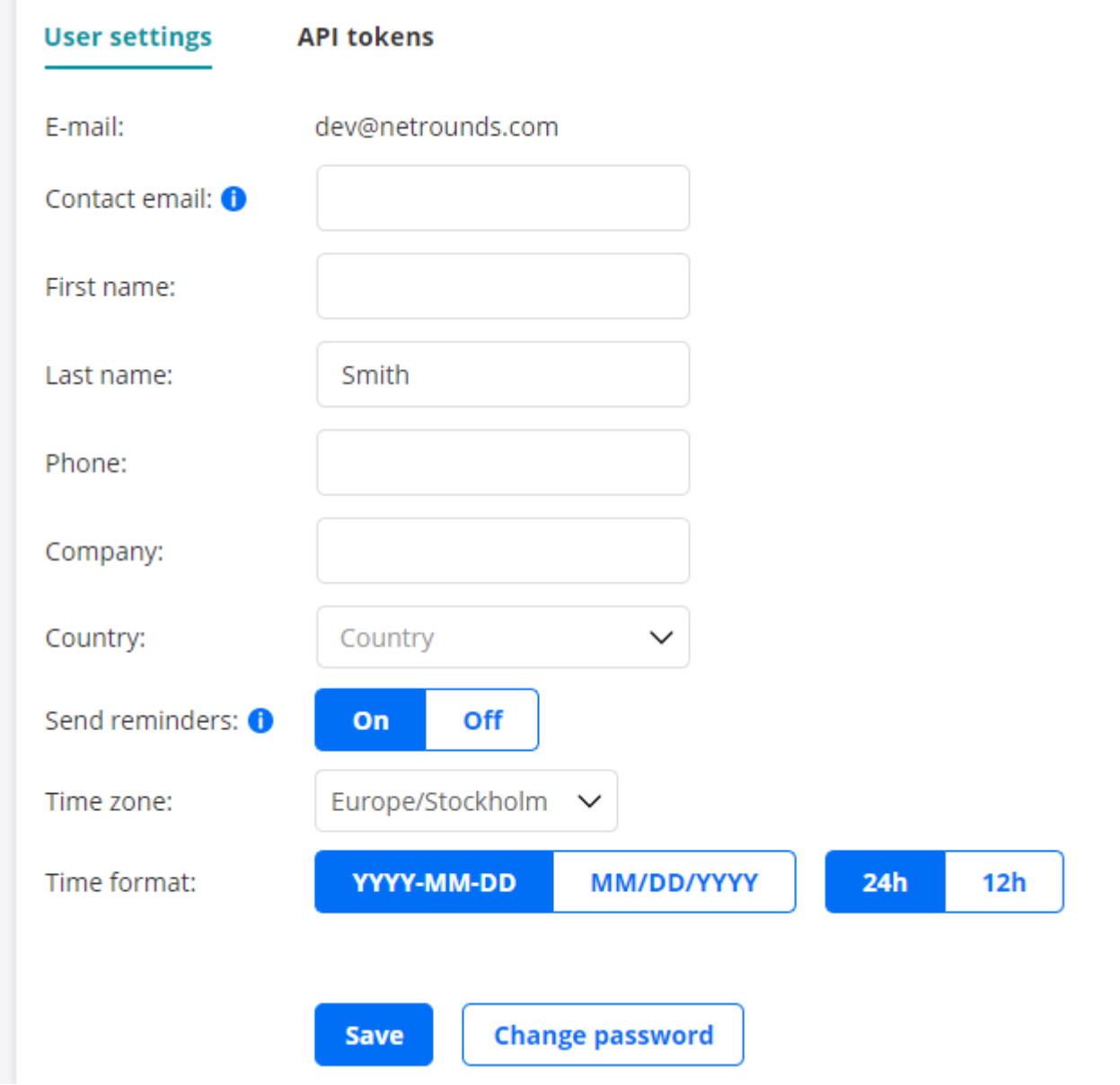


- Click the Select new logo button, and point to the relevant image file. Supported image formats are JPEG, PNG, and GIF. The image size must not exceed 400 × 250 pixels or 100 kB.
- To remove the current logo, click the Delete logo button.

3.16 Changing user settings

You can change various settings for your Routing Active Testing user, including the time zone and the date/time format, by editing your Routing Active Testing user profile.

- Click your user name in the top right corner of the screen, and select Edit profile.



- Contact email, First name, Last name, Phone, Company, Country: Here you can edit personal information. The Contact email address allows you to have email sent to a different address than the one used for registration.
- Send reminders: This setting governs whether you will be sent email reminders if case of low user activity. This settings applies to such reminders only, and not to alarms or periodic monitor reports.
- Time zone: Select the time zone where you currently reside. The default is UTC.
- Time format: Select date and time formats as desired.

Click the Save button to save your settings.

You can also change your Routing Active Testing password here by clicking the Change password button. Follow the instructions in the dialog that appears.

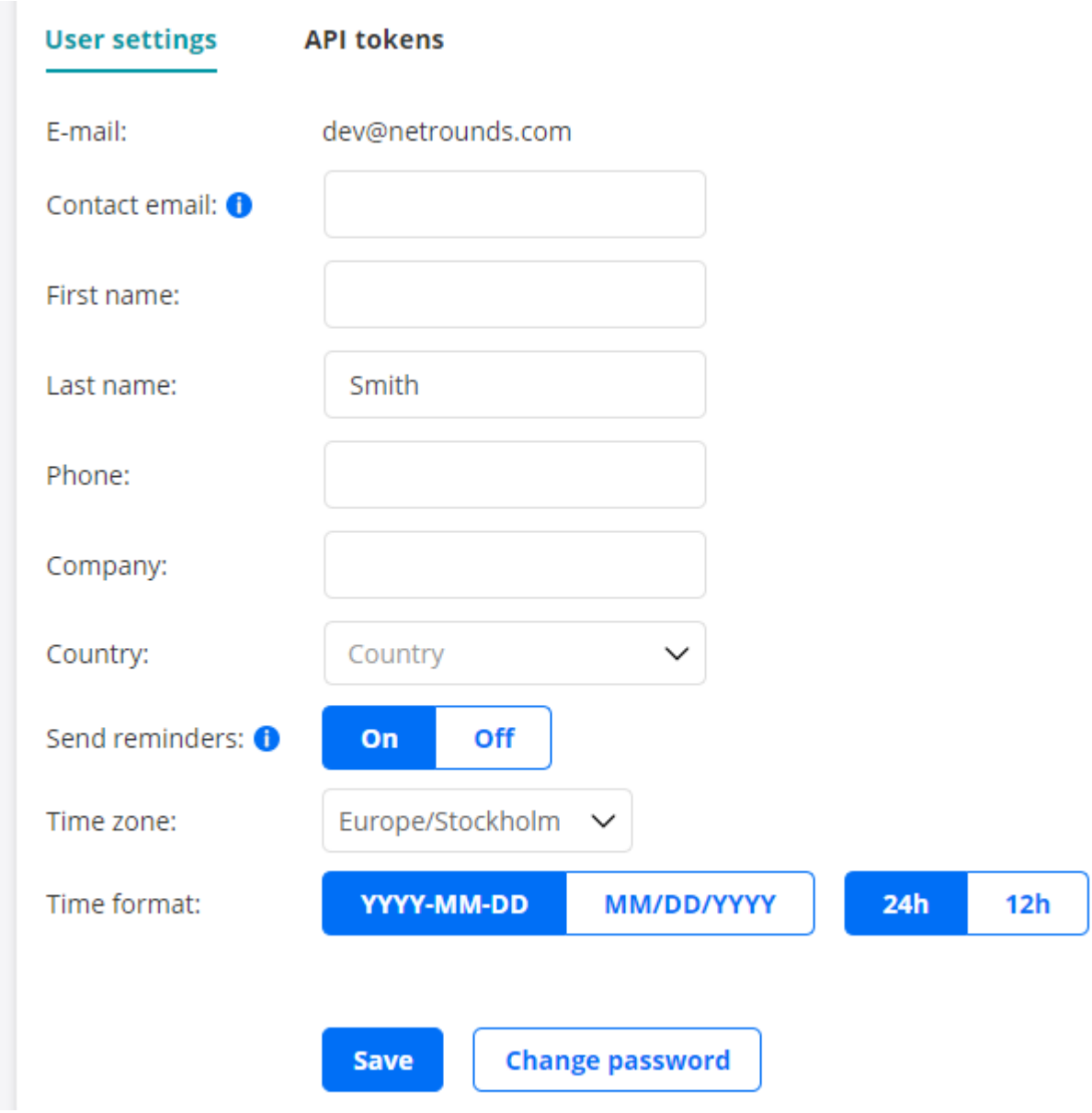
Note: If your user is managed by an LDAP server, its password too is managed on that server and cannot be changed

here. See the Installation Guide for further details on configuring LDAP user management in Control Center.

3.17 Creating API tokens

If you are going to perform orchestration of Routing Active Testing tasks through the REST API, you need to create an authorization token for that API. This is done as follows:

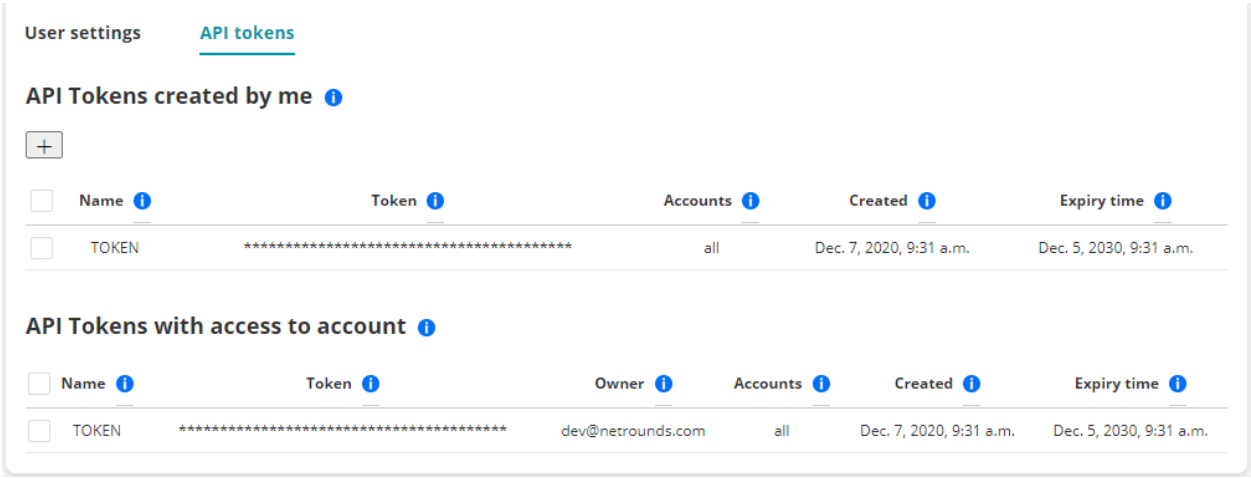
- Click your user name in the top right corner of the screen, and select Edit profile.



- Select the API tokens tab.
- Click the Add button.



- A new token is generated and added to the list API tokens created by me (multiple tokens can be defined). Name it as desired in the Name column.



Note: The value of the API token is visible only at creation time. Therefore you must note it down to be able to use the token later on.

You will need to include this token in all code calling the REST API, as well as when calling the REST API through the API browser provided. See the document "REST API Orchestration Guide" for further information.

If you are not an admin user, you will only see a list of API token objects that you have created yourself. (What is shown is metadata such as creation time, not the token string itself.) However, if you are an admin user you will see a second table here named API tokens with access to account, listing *all* API tokens defined by users in the account. You can delete any tokens here as appropriate when permissions are to be revoked.

4 Test Agents

4.1 Introduction to Test Agents

4.1.1 Test Agent types

Test Agents are measurement points in Routing Active Testing, deployed at arbitrary locations in your network. Test Agents consist of software capable of generating, receiving, and analyzing network traffic.

Test Agents come in the following main varieties: *Test Agent Appliance* and *Test Agent Application*.

- A **Test Agent Appliance** is a full-fledged Test Agent equipped with all measurement capabilities described in this documentation. The Test Agent Appliance is integrated with an optimized Debian Linux OS. The appliance is delivered by Juniper Networks in the form of software which can be installed:
  - on the Juniper NFX150 Network Services Platform
  - on selected Juniper ACX routers
  - on customer-provided x86 hardware
  - as a virtual machine (Virtual Test Agent, vTA) on a hypervisor or public cloud provider.

  In the past, Netrounds also delivered Test Agent Appliances preinstalled on specific hardware. This is no longer offered, but existing devices are still supported.
- A **Test Agent Application** is a Test Agent with functionality detailed *below* (page 57). It consists of software and can be packaged and delivered by Juniper Networks in two ways:
  - Regular **Test Agent Application:** Consists of software downloaded by the customer and is installed as an application on a Linux computer.
  - **Test Agent Cloud-Native Network Function (TA CNF or cTA):** The Test Agent Application can optionally run as a container in any environment that supports it, for example, in routers.

This documentation mostly uses the term "Test Agent", with no suffix. This refers to any Test Agent, insofar as the statement being made is applicable to all Test Agent types. If this is not the case (for example, when discussing a measurement task type not supported by Test Agent Applications), "Test Agent" is simply short for "Test Agent Appliance". Where necessary for the sake of clarity, the intended variety of Test Agent is spelled out.

Note on older naming: Test Agent Appliances were previously referred to as "Probes".

4.1.1.1 Basics of Test Agent Appliances

How to install the Test Agent on an NFX150 is described on *this page* (page 77).

Installation on a PC hard disk, or on a USB memory for temporary use of the PC, is covered *here* (page 72).

Virtual Test Agents are installed in virtualized environments, such as OpenStack, VMware and VirtualBox. In this way a Test Agent can be used as a virtual network function (VNF) in ETSI NFV MANO (Network Function Virtualization Management & Orchestration). Virtual Test Agents can be configured using cloud-init, which means they can be automatically registered with Control Center. For further information, see *this page* (page 102).

A preinstalled Test Agent is simply plugged physically into your network and managed from Control Center. How to get started with a preinstalled Test Agent is covered on *this page* (page 70).

4.1.1.2 Basics of Test Agent Applications

Test Agent Applications are designed for installation on top of a Linux system. It can optionally run as a container in any environment that supports it, for example, in routers. The containerized application is in a position to approximate very closely the performance of other applications running on the same virtual machine. How to install a Test Agent Application is covered *here* (page 159).

The feature set of Test Agent Applications compared to Test Agent Appliances is laid out *here* (page 57).

In certain situations, you need to configure settings on the host OS to get full functionality out of Test Agent Applications. For specifics, see these pages:

- *PCP* (page 516)
- *Ping* (page 431)

4.1.1.3 Configuring Test Agents

There are two ways to configure Test Agent Appliances in Routing Active Testing:

- From the Routing Active Testing GUI. This is covered *here* (page 165).
- From the Test Agent local console, accessed from the command line. The local console is described *here* (page 204).

These interfaces are partly overlapping with respect to their functionality. *Registration* (page 205) of Test Agents, however, must be done from the local console.

Test Agent Applications have no configurable settings other than registration credentials.

4.1.2 Test Agent capabilities

4.1.2.1 Capabilities of Test Agents

The table below gives an overview of the capabilities of Test Agents.

Some features require additional hardware; this is indicated in footnotes.

The measurement features themselves are covered *here* (page 478) (Speedtest, packet capture) and *here* (page 286) (all other features in the table).

Note that certain features recur in multiple tasks; for example, TCP is also used in the Multisession TCP, RFC 6349, and QoS policy profiling tasks.

Feature	TA Appliance	TA Application	No. of TA interfaces needed
TCP	Yes	Yes	2
UDP	Yes	Yes	2
IPTV MPEG	Yes	Yes	1
OTT video (HLS)	Yes	Yes	1
Netflix Speedtest	Yes	Yes	1
HTTP	Yes	Yes	1
DNS	Yes	Yes	1
SIP	Yes	No	2
Wi-Fi network testing	Yes[3]	No	1
Mobile network testing	Yes[4]	No	1
Ethernet service activation	Yes	No	2
Transparency tests	Yes	No	2
Y.1731	Yes	No	1
TWAMP incl. HW timestamping	Yes	Yes	1
TWAMP reflector	Yes	Yes	1
Junos TWAMP	No	Yes	1
Ping	Yes	Yes	1
BWPing	Yes	No	1
Path trace	Yes	Yes	1
UDP loopback	Yes	No	1
Security tests[1]	Yes	No	1 or 2 depending on test
Speedtest[2]	Yes	No	1
Packet capture	Yes	No	1

Test Agents on Juniper equipment

- The Test Agent Appliance running on the NFX150 supports the normal range of features as laid out in the table above (“TA Appliance” column).
- The Test Agent Application running on ACX7000 Family routers supports the following features:
  - UDP
  - TCP
  - HTTP
  - DNS
  - Ping
  - Path trace
  - IPTV
  - OTT video

[3] Requires Wi-Fi network card; details *here* (page 353).
[4] Requires mobile network modem; details *here* (page 355).
[1] These also include IGMP join/leave and Multicast group limit (found in the IPTV category in the GUI).
[2] TCP throughput testing initiated from a web browser.

4.1.2.2 Configuring Test Agents

There are two ways to configure Test Agent Appliances in Routing Active Testing:

- From the Routing Active Testing GUI. This is covered *here* (page 165).
- From the Test Agent local console, accessed from the command line. The local console is described *here* (page 204).

These interfaces are partly overlapping with respect to their functionality. *Registration* (page 205) of Test Agents, however, must be done from the local console.

Test Agent Applications have no configurable settings other than registration credentials.

4.1.2.3 Using Test Agent Applications for 5G core network testing

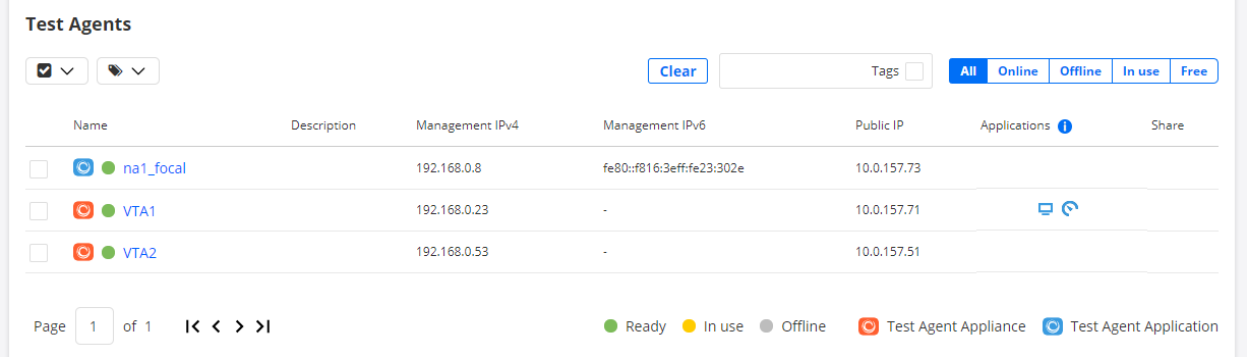
Test Agent Applications are capable of connecting to a 5G core network and running tests and monitors to measure the performance of that network. See *this page* (page 358) for further information.

This feature is currently not available on Test Agent Appliances.

4.1.3 Test Agents view

The Test Agents view shows all Test Agents registered to your account. The list is divided into pages with 15 Test Agents on each page.

In the Shared with me section at the bottom, any Test Agents *shared* (page 491) to your account from other accounts will appear.



The Download button at the top of the view is used to download Test Agent software. See *these pages* (page 70) for more information.

By default, the Test Agents view shows the Interface info tab. Most of this page describes the contents of that tab. For the License info tab, skip to *here* (page 62).

4.1.3.1 Columns in the Test Agents view

Name column

The first icon on the left indicates the type of Test Agent:

	Icon used for Test Agent Appliance
	Icon used for Test Agent Application
	Icon used for generic Test Agent (see below)

Note: *Generic Test Agents.* When creating a Test Agent via the REST or NETCONF/YANG API, we cannot know beforehand which type it is: Test Agent Appliance or Test Agent Application. This becomes clear only after the Test Agent has registered. Therefore, in the meantime, the Test Agent is represented by a generic, *gray* icon in the Test Agent view. Upon registration, the icon will change color to orange (for a Test Agent Appliance) or blue (for a Test Agent Application).

The colored dot icon immediately to the left of the Test Agent name indicates the current status of the Test Agent:

	*Green:* Online and ready, currently not in use
	*Yellow:* Online and currently in use
	*Gray:* Offline

You can *filter* the Test Agents view with respect to Test Agent status:



Selecting the option Online means that only Test Agents marked with a green or yellow dot will appear, while selecting the option Offline means that only Test Agents marked with a red dot will appear.

Clicking the Test Agent name takes you to a new screen showing Test Agent properties, divided into a number of tabs. See *this page* (page 64).

Description column

Plain-text description of the Test Agent entered in the heading of the Test Agent properties view: see *this page* (page 64).

Management IP column

This column shows the IP address of the Test Agent management interface (this is by default “eth0” for a Test Agent Appliance).

When two addresses are given, in the format “address_1 (address_2)”, the first is the local network IP address used by the Test Agent, for example behind a NAT router or a firewall, and the second is the public IP address seen by Control Center. If only one address is given, the local and public addresses are the same.

Applications column

This column uses icons to show which applications have been enabled on this Test Agent. The enabling is done on the Applications tab of the Test Agent properties, as detailed *here* (page 188).

These are the icons used:

	Speedtest
	Use of Test Agent as proxy
	Live remote packet capture

Share column

For Test Agent Appliances, this column displays an icon indicating the sharing status of the Test Agent. All about sharing of Test Agents is covered on *this page* (page 491).

These icons occur in the main listing, Test Agents:

	This Test Agent is not currently shared to any other account.
	This Test Agent is currently shared to at least one other account.

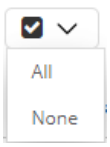
This icon is used in the Shared with me listing:

	This Test Agent is being shared to this account from some other account.

4.1.3.2 Actions applicable to Test Agents

You can *select* one or several Test Agents by selecting the checkboxes on the far left in the Test Agents listing.

- Click this button at the top to select or deselect all Test Agents.



- You can then perform *tagging* actions by clicking this button. See *this page* (page 67) for further instructions on using tags.



- When at least one Test Agent is selected, this additional button appears. You can use it to update the software of the selected Test Agents, or of all Test Agents.



4.1.3.3 Filtering the Test Agents view

The filter controls are found at the top of the view.



Filtering of Test Agents by *status* (online/offline) has already been covered *here* (page 60).

You can also filter Test Agents by *name*, by typing characters into the "Search" text box, or by *tag*, by selecting the Tags checkbox and then typing or selecting tag names as explained *here* (page 68).

4.1.3.4 License info tab of the Test Agents view

The License info tab displays license and stream information for the Test Agents.

- Name column: Holds the same information as on the Interface info tab.
- License column: Shows the type of license assigned to the Test Agent.
- No. of streams column: Shows the total number of streams granted by the license.
- Used streams column: Shows the number of streams currently in use.
- Available streams column: Shows the number of unused streams currently available.
- Share column: Shows the same information as on the Interface info tab, but the icons are not clickable.

Name	License	No. of streams	Used streams	Available streams	Share
na1_focal	Unlimited	8800	2	8798	
VTA1	SW-Test Agent Medium	100	2	98	
VTA2	Unlimited	8800	0	8800	

Read more about licensing and streams on *this page* (page 245).

At the bottom, under the heading Licenses, are displayed the types of license connected to your account.

- Type: License type.
- No. of licenses: Total number of licenses purchased.
- Used licenses: Number of licenses currently in use.
- Available licenses: Number of unused licenses currently available.

Test Agent Licenses

Type	No. of licenses	Used licenses	Available licenses
SW-Test Agent Mini	10000	0	10000
SW-Test Agent Small	10000	0	10000
SW-Test Agent Medium	10000	1	9999
SW-Test Agent Large	10000	0	10000
SW-Test Agent 8800	10000	0	10000
SW-Agent Mini	10000	0	10000
SW-Agent Small	10000	0	10000
SW-Agent Medium	10000	0	10000
Unlimited	10000	2	9998
IPTV	-	Not used	Available
TWAMP	-	Not used	Available

Finally, at bottom right, another table gives the maximum number of *streams* allowed in the account, the number of streams currently in use, and the number of streams available.

Total streams for Account

Total streams allowed	100000
Used	4
Available	99996

Again, please refer to the page on *licensing and streams* (page 245) for more detailed information, especially on the counting of streams for various task types.

4.1.4 Test Agent properties view

This page deals with the multi-tab property view that is displayed when you click a Test Agent in the Test Agents view.

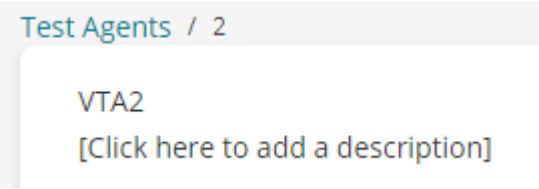
Much of this is covered on other pages of the documentation, and the present page links to those.

The set of property tabs displayed differs between Test Agent types, as laid out in the table below. “Yes” with an asterisk (*) means that further conditions must be met in order for the tab to be visible, as detailed in the sections that follow.

Tab	TA Appliance	TA Application
Interfaces	Yes	Yes
Applications	Yes*	No
NTP	Yes*	No
Streams	Yes	Yes
License	Yes	Yes
Utils/Unregister	Yes	Yes
GPS Location	Yes	No
Platform Information	Yes	No

4.1.4.1 Heading at top of view

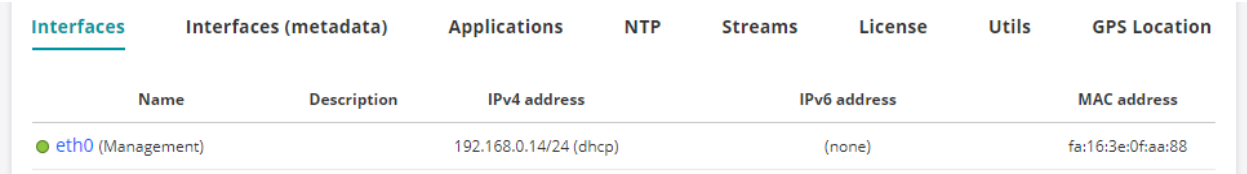
The heading consists of the Test Agent name followed by a row in smaller type with a plain-text description. You can edit both the name and the description by clicking on the item in question.



If you change the Test Agent name, it will be updated in any tests and monitors that use the Test Agent.

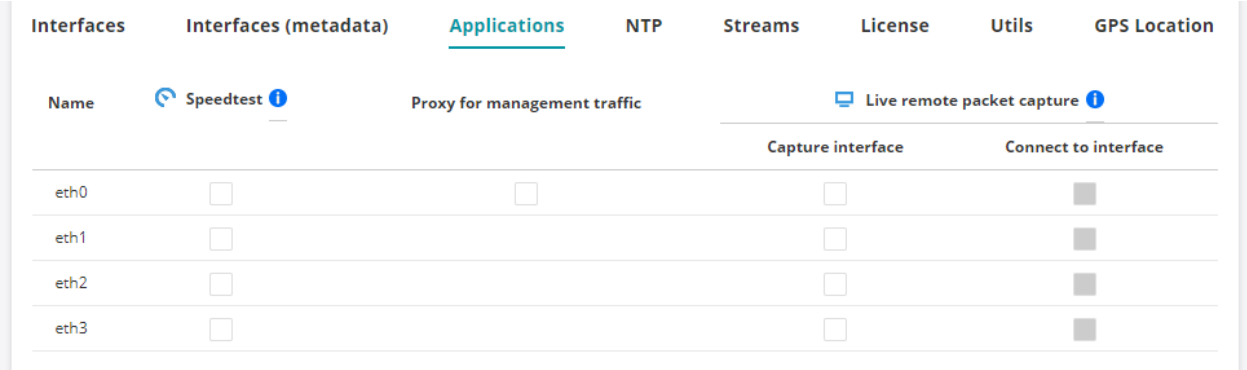
4.1.4.2 Interfaces tab

See *this page* (page 166), which covers everything about Test Agent interface configuration.



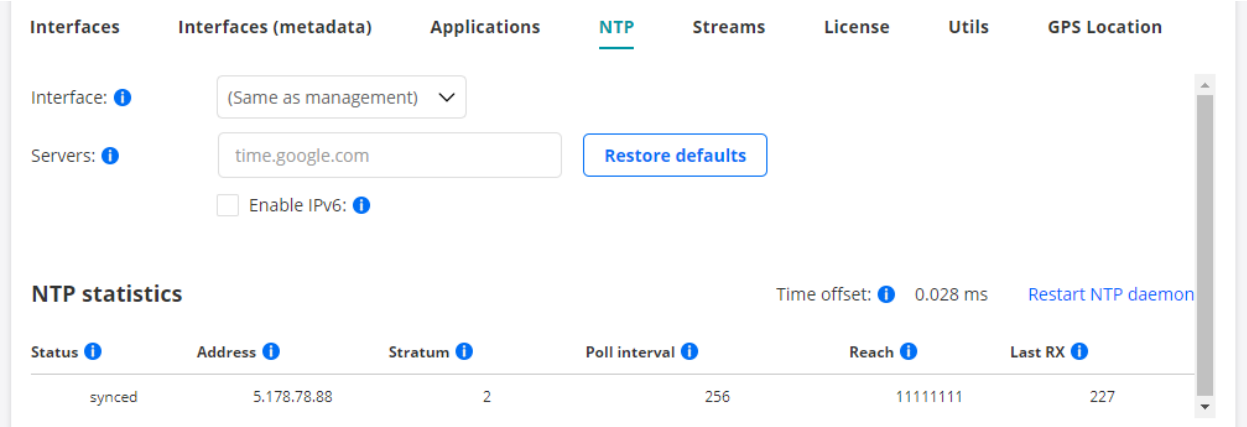
4.1.4.3 Applications tab

This tab appears only for Test Agents that are currently online. On this tab you can enable and disable the various *applications* available in Routing Active Testing. For further information, see *this page* (page 188).



4.1.4.4 NTP tab

This tab appears only for Test Agents that are currently online. It deals with NTP server settings and is covered in full *here* (page 189).

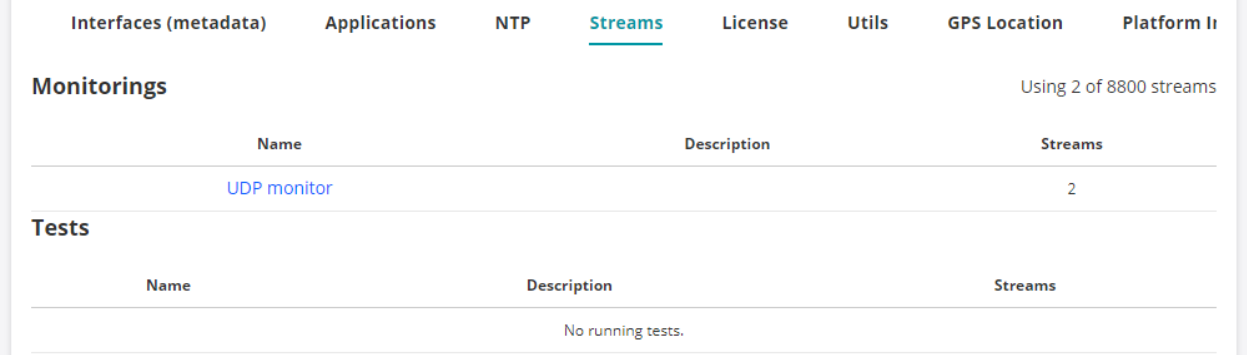


4.1.4.5 Streams tab

This tab lists all streams that are being used by monitoring sessions, tests, applications, and shares.

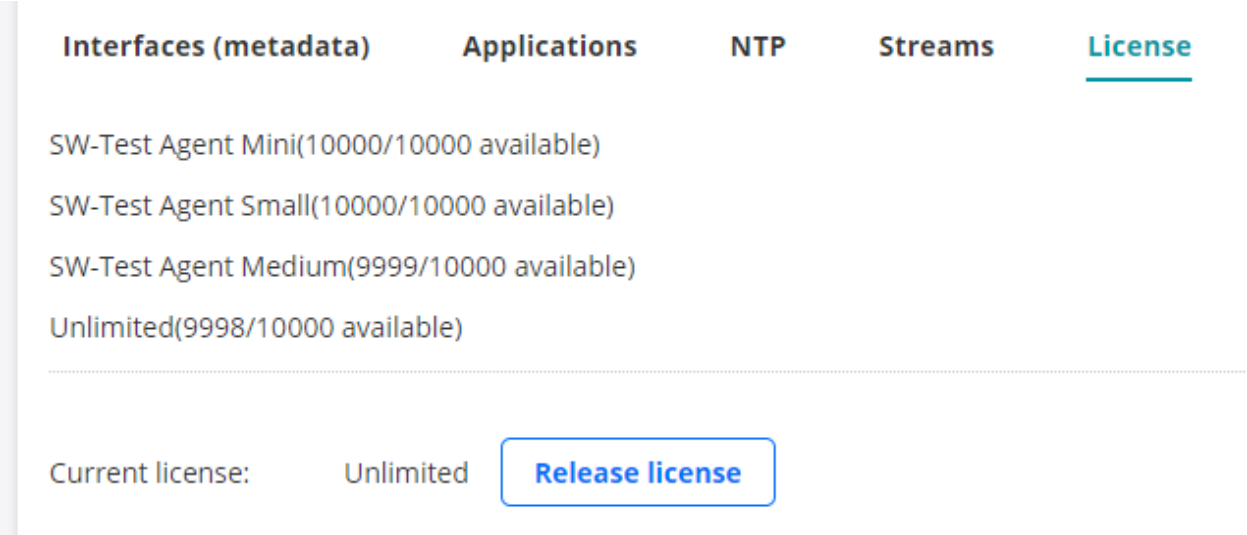
At the top of the tab, the total number of streams currently in use is indicated.

Read more *here* (page 245) about the number of streams consumed by testing of various services and by applications.

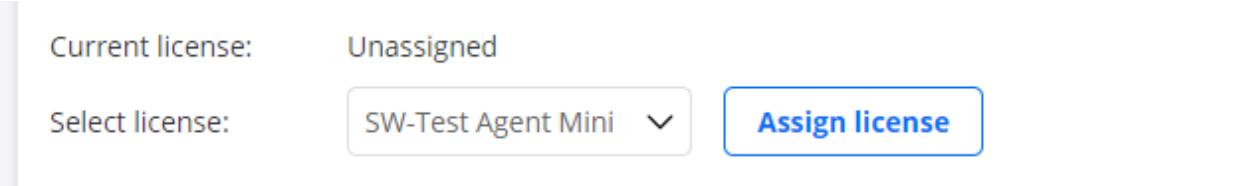


4.1.4.6 License tab

This tab shows the pool of licenses currently available to Test Agents in your account. It also shows the license currently assigned (if any) to this Test Agent:



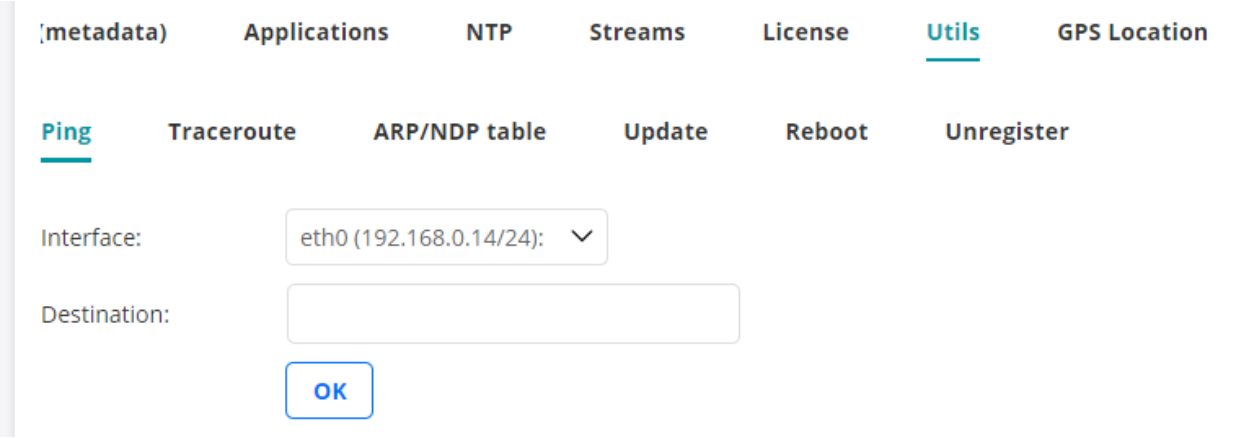
If no streams are currently in use, you can release that license by clicking the Release license button. When the Test Agent has no license assigned, the tab looks like this instead:



To assign a license, pick a license type from the drop-down box, then click the Assign license button.

4.1.4.7 Utils tab

This tab holds some useful utilities for troubleshooting and management. See *this page* (page 193) for full coverage.



4.1.4.8 GPS Location tab

This tab holds geographical coordinates for the Test Agent. See *this page* (page 196) for more information.

4.1.4.9 Platform Information tab

This tab holds information about the Test Agent's platform such as BIOS version, memory, and operating system. See *this page* (page 197) for more information.

4.1.5 Using tags to group and manage Test Agents

Tags simplify management of Test Agents in the Test Agents view. Tags provide a means to group Test Agents into categories, so that you can quickly select relevant devices on which to start a new test or monitoring session. This feature is particularly helpful if your fleet of Test Agents is large.

A tag can consist either of a *key* alone (for example, "location") or of a *key* with an accompanying *value* (for example, "location:sweden"). A colon is used as separator between key and value. Tags may consist of up to 50 characters, which may be lowercase letters or digits.

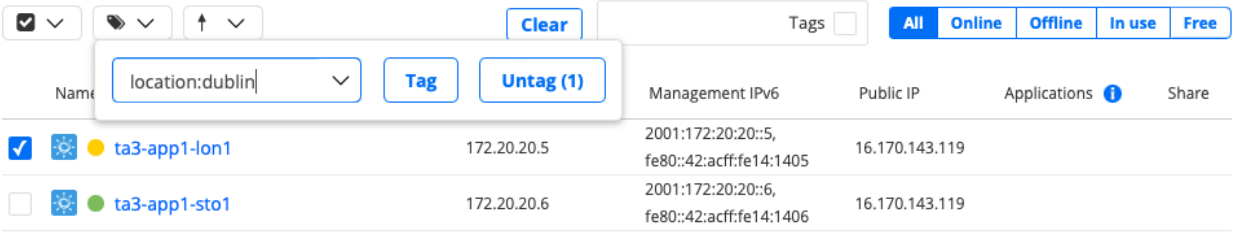
In the Test Agents view it is possible to:

- add a tag to selected Test Agents
- remove a tag from ("untag") selected Test Agents.

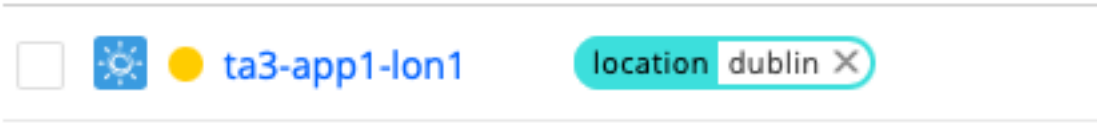
Note that the tags themselves, once created, cannot be deleted from the Control Center GUI.

4.1.5.1 Adding tags to Test Agents

- First select the Test Agents that you want to tag.
-  Click the Tags button at the top of the page.
- You can either create a new tag or select an existing tag:
  - To create a new tag, type the desired string into the box, then click the Tag button.
  - To select an existing tag, click the down arrow and select the desired tag from the drop-down box (optionally, you can type the first few characters of the tag name to match the name). Then click the Tag button. See the screenshot below:



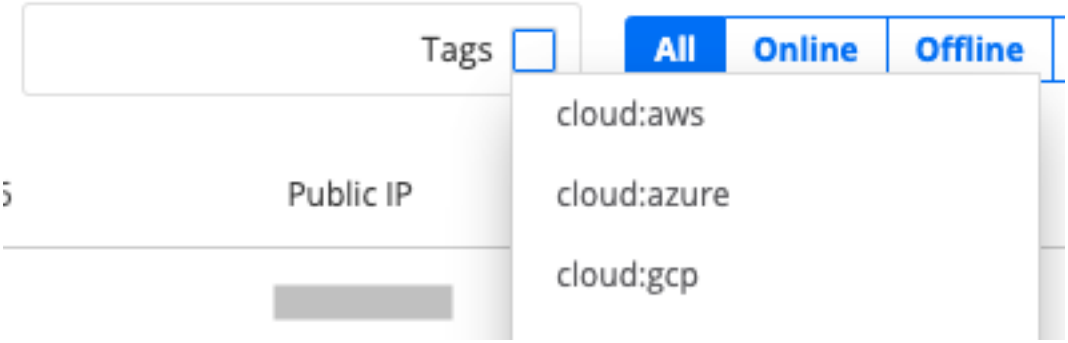
Each tag will show up in a box next to the Test Agent name.



It is possible to add several tags to the same Test Agent.

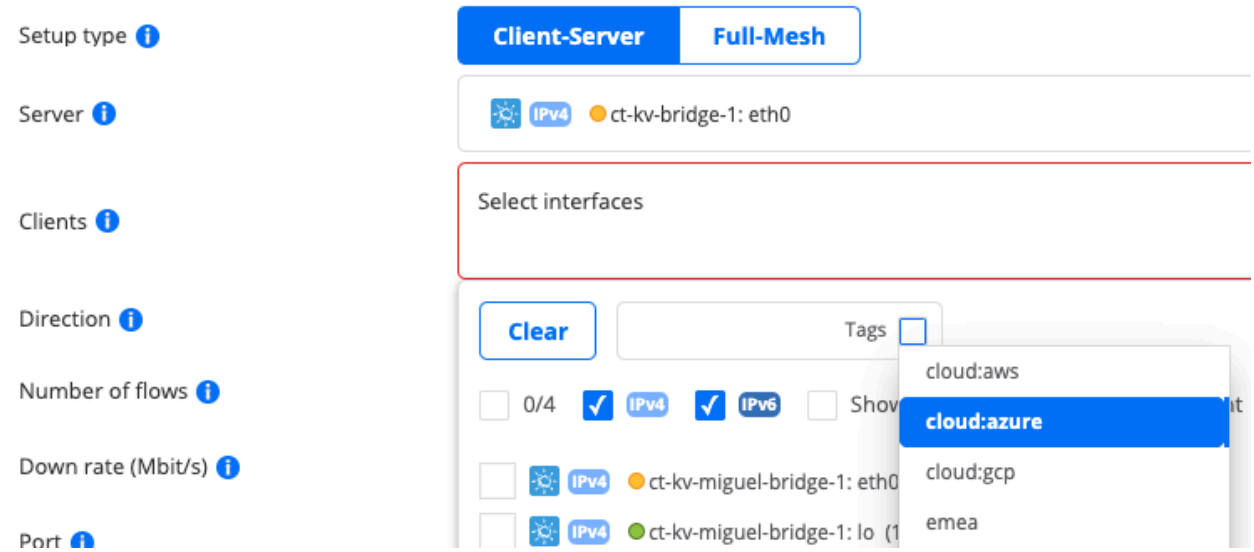
4.1.5.2 Using tags to filter Test Agents

- Select the Tags checkbox in the search field, and a drop-down list appears holding all defined tags.



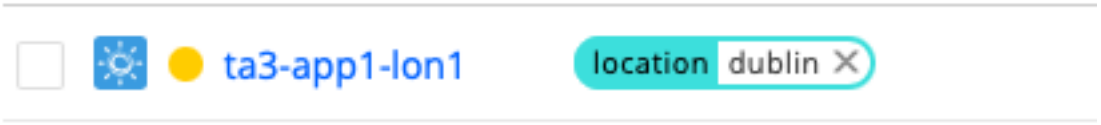
- Select one or several tags to filter the Test Agents view on these tags. The Test Agents having *all* the selected tags will then be displayed.
- To display the full list again, just deselect all tag names in the Tags drop-down in the search field.

Filtering on tags can also be applied when you select interfaces during creation of new tests and monitors. Just select the Tags option, then select the name of the tag(s) to filter on. See the example below.

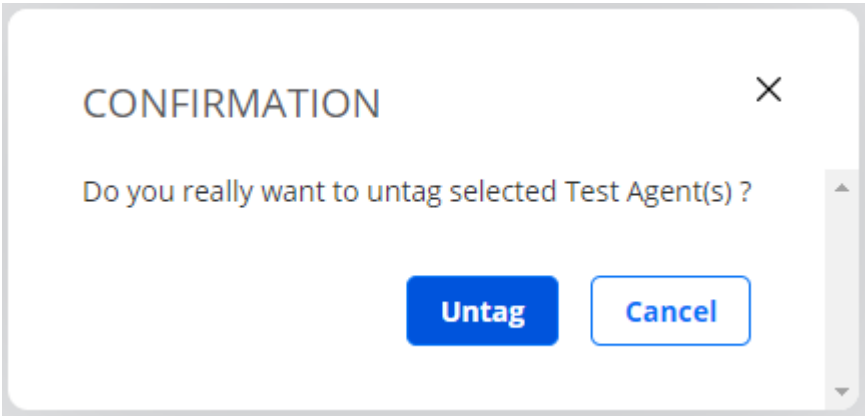


4.1.5.3 Untagging Test Agents

The simplest way to remove a tag from a Test Agent is to click the cross on the tag:

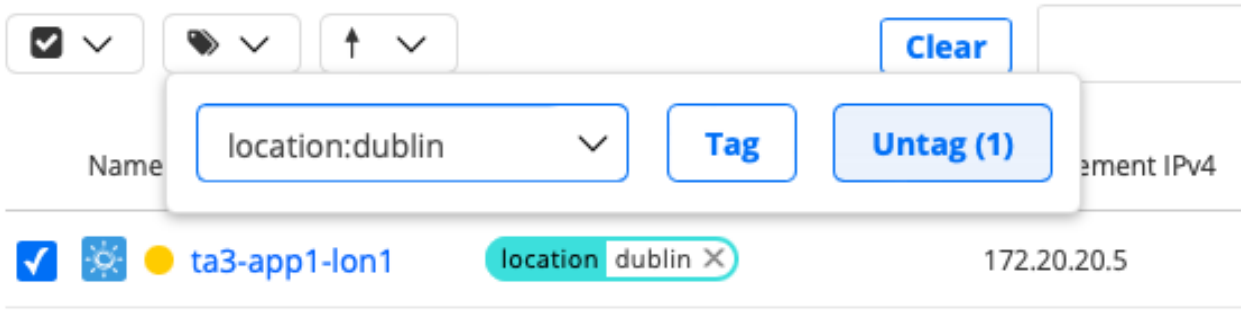


- You are prompted to confirm this action. Click Untag.



Alternatively, you can do the following:

- First select the Test Agents that you want to untag.
-  Click the Tags button at the top of the page.
- Then select the name of the tag to remove (“location:dublin” in the example below), and click the Untag button.



The tag is now removed from the selected Test Agents.

4.2 Installing Test Agents

4.2.1 Getting started with preinstalled Test Agents

Note: Test Agents are no longer delivered preinstalled on hardware. However, existing devices continue to be supported, and the present page is kept for reference.

The delivery will have contained the ordered combination of hardware devices, each with Test Agent software installed. Follow the steps below to prepare the Test Agents for use.

In some situations below, it is necessary to use the local administration console of the Test Agent. How to access that console is described *here* (page 204).

- Unpack the Test Agents. Note that each Test Agent has multiple network interfaces, making it possible to use dedicated ports for management, testing and monitoring.
- Each Test Agent must establish a secure connection to the Routing Active Testing server before you can use it through the management interface and port. All remote accessing and administration of your Test Agents are done through the Routing Active Testing server once the Test Agents have been registered and have come online.
  - If you are using your own on-premise Routing Active Testing server, you need to configure the Test Agent with the server address and port. This is done through the Test Agent local administration console. Use *this function* (page 217) to set the login server to the Control Center host IP and the login server port to 6000.
- Connect the Test Agent's "eth0" interface (the default management interface) to an Ethernet access port on your network.
  - If you are using an on-premise Routing Active Testing server, make sure that the path between the Test Agent's management interface and Control Center is not blocked on port 6000.
  - If you are using a cloud server account, make sure that the Test Agent has internet connectivity.
- Once the Test Agent is powered on, it will request a DHCP address on the "eth0" interface.
  - If your network does not offer DHCP addresses, you need to specify a static IP address on the Test Agent. This is done through the Test Agent local console. Follow *these instructions* (page 210). If you want to use a physical interface other than "eth0" or a VLAN interface for Test Agent management, this is likewise done from the local console; again, details are found *here* (page 210).
- Registration (on-premise server only): If you are using an on-premise Routing Active Testing server, register the Test Agent to the previously created Routing Active Testing account on the server. To this end, use the Register option in the Test Agent local console as explained *here* (page 205). (With a cloud server, the registration will be automatic, as explained *below* (page 71).)

- After an IP address has been assigned, the Test Agent will connect to your specific Routing Active Testing account using the encryption keys that were preloaded into the Test Agent before delivery.
- If you have a firewall in place, make sure that the Test Agent is allowed to establish an outgoing session towards the Routing Active Testing server:
  - in the cloud server case, to https://login.paa.juniper.net using TCP port 443 (Test Agent Appliance) or TCP port 6800 (Test Agent Application);
  - in the on-premise server case, to the host IP using TCP port 6000.

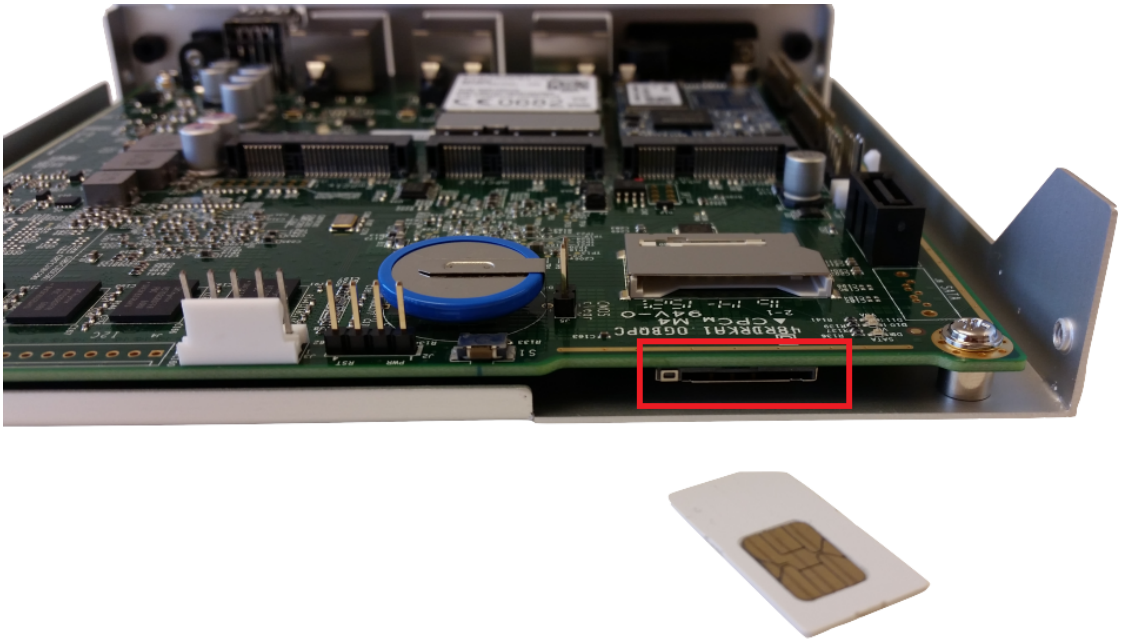
  You do not need to allow any incoming connections, since the Routing Active Testing server will communicate with the Test Agent using the reverse direction on the same TCP session.
- If you are making use of the Routing Active Testing cloud server, the Test Agent will now automatically register with the server and appear in your Routing Active Testing account.
- Log in to your account in the Routing Active Testing GUI by entering your email address and password.
- Check that the Test Agent is online by accessing the Test Agents view from the main menu. Here, an inventory of all your Test Agents is shown. Your Test Agent should have a green indicator next to it, signifying that it is online. See *this page* (page 59) for full coverage of the Test Agents view.
- A valid license is assigned to the preinstalled Test Agent by Routing Active Testing prior to delivery. To inspect this license, click the Test Agent and go to the License tab. Should no valid license exist after all, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

The preinstalled Test Agent is now ready to use.

4.2.1.1 Installing a SIM card in HW Medium Mobile hardware

The photo below shows where to insert a SIM card in the “HW Medium Mobile” Test Agent hardware platform equipped with an LTE chip.

First remove the four screws at the side of the box to detach the cover.



Note: The SIM card must have its PIN code disabled, as there is no way to communicate a PIN from the Routing Active Testing user interface to the Test Agent.

4.2.2 Installing Test Agent Appliances: Introduction

You can install a Test Agent Appliance:

- on an NFX150 Network Services Platform from Juniper. See *this page* (page 77).
- on a PC Engines APU2 hardware server. See *this page* (page 89).
- on a Supermicro hardware server. See *this page* (page 93).
- on a fitlet2 hardware server. See *this page* (page 99).
- on x86 hardware of your own to permanently use that hardware as a Test Agent. See *this page* (page 75).
- on a USB memory for temporary use of a PC where it is inserted. This is suitable for temporary measurements and troubleshooting. See *this page* (page 73) for further instructions.
- on various virtualization platforms. See *this page* (page 102).

The hardware used (including the network interface cards) determines the number of available network interfaces and the link performance. Minimum hardware requirements are stated in the Test Agents datasheet, which is found at https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000684-en.pdf.

Note: No existing or underlying operating system is needed on your hardware – the Test Agent download package includes an optimized real-time Linux operating system.

4.2.2.1 Prerequisites

- A wired interface is required for *registering* (page 205) Test Agents. Registration cannot be done over Wi-Fi. This means that a hardware device equipped only with Wi-Fi interfaces must be extended with at least one wired interface for the purpose of registration. Once the Test Agent is registered, it can be *configured to be managed* (page 210) over a Wi-Fi interface.
- If you have a firewall in place on the device or in the network, make sure that the Test Agent is allowed to establish an outgoing session towards the Routing Active Testing server: in the cloud server case, to https://login.paa.juniper.net using TCP port 443 (Test Agent Appliance); in the on-premise server case, to the host IP using TCP port 6000 (default). You do not need to open any incoming connections, since the Routing Active Testing server will communicate in the reverse direction on the same TCP session that the Test Agent initiated.
  - Alternatively, a Test Agent may communicate with the Routing Active Testing (cloud) server via an HTTP proxy or via another Test Agent acting as proxy. This makes it possible to do testing in networks that do not otherwise allow any external connections. See *this page* (page 505) for instructions.

4.2.3 Installing a Test Agent Appliance as a bootable image on a USB memory stick

This page explains how to install a Test Agent Appliance as a bootable image on a USB memory. You can then boot a PC from that USB memory and temporarily transform the PC into a Routing Active Testing measurement device.

In the instructions given below, the installation procedure is done using a GUI application named Etcher. If you are skilled in working with bootable disk images, you can alternatively manage without this program. You can then download the `img.gz` file mentioned below and transfer the embedded disk image to a USB memory using the Unix/Linux/Mac OS X command `dd`.

Warning: Be careful to write to the correct disk; otherwise you might wipe your entire PC.

4.2.3.1 Installation procedure

- Download the Test Agent installer from Control Center:
  - Click Test Agents in the main menu.
  - Click the Download button in the top right corner.
  - In the dialog that appears, click the link RAW disk image (.img.gz) to download.

Note: The disk image is compressed using the GZIP compression format. If you are not using Etcher (see below), you will need to decompress the image before flashing it. To uncompress the archive, use one of the following tools:

- 7-Zip (Windows)
- The Unarchiver (Mac)
- `gzip` command line utility (Linux)

4.2.3.2 Writing the image to a USB memory stick using Etcher

Warning: Any existing content on the USB memory will be erased.

- Download Etcher and install it.
- Insert a USB memory stick with at least 4 GB of free space into your PC.
- Open **Etcher**, and from your hard disk select the Test Agent `.img.gz`.
- Select the USB memory stick you wish to write your image to.
- Review your selections and click **Flash!**

4.2.3.3 Using the USB memory stick

- Insert the USB memory into a PC with USB boot support.
- Turn on the PC.
- Access the BIOS boot menu. In many cases you can do that by pressing one of the keys F8, F11, or F12 during start-up.
- Make sure the USB memory comes *before* the PC hard disk in the boot sequence.
- Select USB boot from the BIOS boot menu.
- The boot process takes about 20 seconds. When the login prompt is shown, log in as user “admin” with password “admin”.

A text-based menu will now appear:



- Make sure the PC is connected to the Internet using a wired interface. You can check the link status by selecting Show interface status; read more about that *here* (page 209).
- You can also change IP addressing for management of the Test Agent by selecting Configure management. By default, the Test Agent uses DHCP on the first detected interface (normally called “eth0”). If your network does not offer DHCP addresses, you need to specify a static IP address on the Test Agent; how to do this is explained *here* (page 210). The same page also explains how to specify a different interface for Test Agent management.
- If your PC has multiple wired interfaces, and you are unsure which interface is eth0, unplug all interfaces but one and select Show interface status to see which interface has a link detected. It is helpful to label each interface with a sticker for easy reference when cabling on-site.

Before you can use the Test Agent, you also need to *register* it with the Routing Active Testing server. How to do this is explained *here* (page 205).

After successful installation, the Test Agent will check for new software packages to ensure that it is up to date. This process may take several minutes to finish. After completing the update check, the Test Agent should automatically appear in your Routing Active Testing account.

- Click Test Agents on the main menu. This view shows all Test Agents in your account. Verify that your Test Agent is found here and has a green dot next to it, signifying that it is online.

If the Test Agent does not have a green dot, check that it has a *license* assigned (this is normally done automatically upon registration):

- Click the Test Agent in the Test Agents view, then click the License tab.

- If the Test Agent does not have a license, you need to assign one as explained *here* (page 191).

4.2.4 Installing a Test Agent Appliance on your own hardware

This page explains how to install Test Agent Appliance software on x86 hardware of your own.

4.2.4.1 Writing the Test Agent image to a USB stick

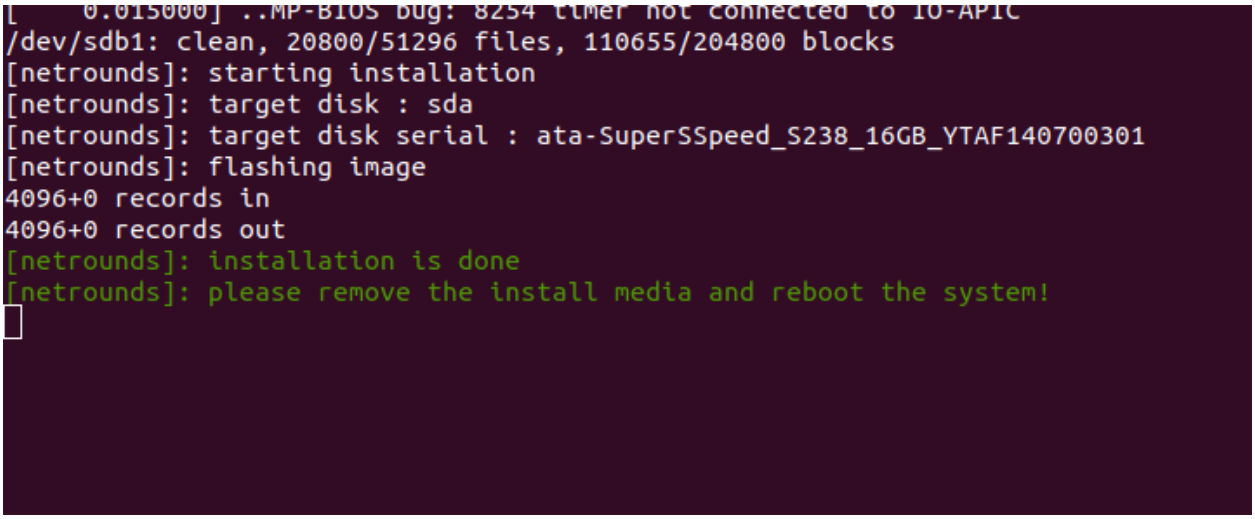
- Follow the instructions on the page *Installing a Test Agent Appliance as a bootable image on a USB memory stick* (page 73), up to and including the section "Writing the image to a USB memory stick using Etcher".

4.2.4.2 Installing the Test Agent on your hardware

- Insert the USB stick with the Test Agent image into a USB port on your hardware device.
- Access the BIOS boot menu.
- Make sure the USB memory comes before the hard disk in the boot sequence.
- Select "USB boot" from the BIOS boot menu.
- The boot process takes about 20 seconds. When the login prompt is shown, log in as user "admin" with password "admin".
- Go to Utilities.
- Select Install to disk.

> **Warning:** Any existing content on the disk will be erased.

When the procedure has finished, the screen will look something like this:



You now need to reboot the hardware. Be sure to remove the USB stick so that the hardware boots from the installed image on the hard disk.

The hardware will now boot up as a Test Agent.

- When the login prompt is shown, log in with user “admin” and password “admin”. A text-based menu will now appear, as shown below.



- Make sure the hardware is connected to the Internet using a wired interface. You can check the link status by selecting Show interface status; read more about that *here* (page 209).
- You can also change IP addressing for management of the Test Agent by selecting Configure management. By default, the Test Agent uses DHCP on the first detected interface (normally called “eth0”). If your network does not offer DHCP addresses, you need to specify a static IP address on the Test Agent; how to do this is explained *here* (page 210). The same page also explains how to specify a different interface for Test Agent management.

If your hardware device has multiple wired interfaces and you are unsure which interface is eth0, unplug all interfaces but one and select Show interface status to see which interface has a link detected. It is helpful to label each interface with a sticker for easy reference when cabling on-site.

- Before you can use the Test Agent, you also need to *register* it with the Routing Active Testing server. How to do this is explained *here* (page 205).

After successful installation, the Test Agent will check for new software packages to ensure that it is up to date. This process may take several minutes to finish. After completing the update check, the Test Agent should automatically appear in your Routing Active Testing account.

- Click Test Agents on the main menu. This view shows all Test Agents in your account. Verify that your Test Agent is found here and has a green dot next to it, signifying that it is online.

If the Test Agent does not have a green dot, check that it has a *license* assigned (this is normally done automatically upon registration):

- Click the Test Agent in the Test Agents view, then click the License tab.
- If the Test Agent does not have a license, you need to assign one as explained *here* (page 191).

.. include:: ../../custom_roles.rst

4.2.5 Installing Test Agent Appliance on the Juniper Networks® NFX150 Network Services Platform

4.2.5.1 Overview

This document describes how to install a Test Agent Appliance on the NFX150-C-S1 Network Services Platform. The supported version of Junos OS is 22.1R1.

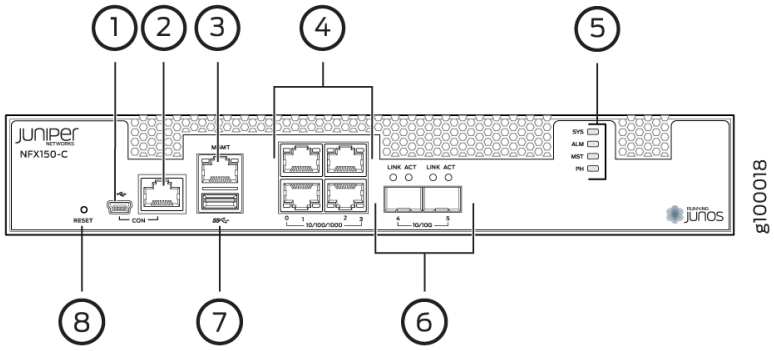
The NFX150 Network Services Platform provides multiple network connection options (including 4G/LTE) and an open, standards-based architecture. As a secure, universal customer premises equipment (uCPE) device, the NFX150 can support and manage multiple Juniper and third-party VNFs. For more information, go here: https://www.juniper.net/us/en/products-services/sdn/nfx-series/nfx150/

By default all command examples shown here are executed in the Junos OS CLI, which you can start by logging in to the NFX150 and running the command `cli`.

4.2.5.2 Physical layout of the NFX150

All NFX150 models have four 10/100/1000BaseT interfaces and two SFP/SFP+ interfaces, as well as a dedicated management interface.

In addition to this, the NFX150-S1 models have a module slot that can provide an additional eight 1 Gbps interfaces; these are currently not used by the Test Agent.



Legend:

1/2 – Serial port for NFX150 management.

3 – Dedicated management port (`fxp0`). Used for management of the NFX150 itself, but also possible to use from VNFs.

4 – General 10/100/1000BaseT Ethernet ports. Intel i350 NIC with SR-IOV support. Internally called `heth-0-0` to `heth-0-3`.

6 – General SFP/SFP+ ports. Intel X553 NIC. Internally called `heth-0-4` and `heth-0-5`.

4.2.5.3 Initial configuration

In this section, for convenience, we reproduce the instructions found here: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/initial-configuration-nfx150.html

Connecting to the device

- Login as root:

```
FreeBSD/amd64 (Amnesiac) (ttyu0)

login: root
```

- Start the CLI:

```
root@:~ # cli
root>
```

Inspecting Junos OS version and hardware information

To display the Junos OS version:

```
show version | grep Junos:
```

If the Junos OS version is something other than 22.1R1, you need to update the firmware to that version as described in the next section.

To display information about chassis hardware:

```
show chassis hardware
```

Verify that the chassis description is “NFX150-C-S1”. This is the chassis supported.

4.2.5.4 Updating the NFX150 firmware (Junos OS)

Please refer to this page: https://www.juniper.net/documentation/en_US/junos/topics/reference/command-summary/request-vmhost-software-add-nfx-series.html

An alternative method is to install from USB according to the following document: https://kb.juniper.net/InfoCenter/index?page=content&id=KB32971

Enabling basic connectivity

Here you will change the root password and optionally enable root SSH access.

1. Ensure that the NFX150 device is powered on.
2. Connect to the console port:
    1. Plug one end of the Ethernet cable into the console port on your NFX150 device.
    2. Connect the other end of the Ethernet cable to the RJ-45-to-DB-9 serial port adapter shipped with your device.

3. Connect the RJ-45-to-DB-9 serial port adapter to the serial port on the management device. Use the following values to configure the serial port: Baud rate – 9600; Parity – N; Data bits – 8; Stop bits – 1; Flow control – None.

Note: Alternatively, you can use the USB cable to connect to the mini-USB console port on the device. To use the mini-USB console port, you must download the USB driver from the following page and install it on the management device: https://www.juniper.net/support/downloads/junos.html

---

3. Use any terminal emulation program, such as HyperTerminal, to connect to the device console. The CLI displays a login prompt.
4. Log in as root and enter the password `juniper123`. If the software completes booting before you connect to the console, you might need to press the Enter key for the prompt to appear:

Note: Starting with Junos OS Release 18.1R2 or later, the root password is not configured for initial configuration of the NFX150 devices.

```
login: root
password: juniper123
```

5. Start the CLI:

```
root@:~ # cli
root@>
```

6. Enter configuration mode:

```
root@> configure
[edit]
root@#
```

7. Change the password for the root administration user account:

```
[edit]
root@# set system root-authentication plain-text-password
New password: password
Retype new password: password
```

8. Enable SSH service for the root user:

```
[edit]
root@# set system services ssh root-login allow
```

9. *(Optional)* Enable the Internet connection for devices connected on LAN by setting the DNS IP:

```
[edit]
root@# set access address-assignment pool junosDHCPPool family inet dhcp-
↪attributes name-server
dns-server-ip
```

10. Commit the configuration:

```
[edit]
root@# commit
```

Configuring LAN access on the NFX150 management port

The port labeled `MGMT` can be set up for remote management of the NFX150 device. This port is by default configured with a virtual interface called `fxp0` which by default has the static IPv4 address 192.168.1.1. For further configuration of the device it is suitable to set up the `fxp0` interface to give access to a Control Center, so that the device can download images and configuration files from there.

Below is an example of how to remove the existing static address and configure the virtual interface to use DHCP instead:

```
root> configure
root# delete interfaces fxp0 unit 0 family inet address 192.168.1.1/24
root# set interfaces fxp0 unit 0 family inet dhcp
root# commit
```

For more information on configuring the in-band management interface, refer to https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/nfx150-configuring-inband-management-interface.html

Configuring a suitable virtualization mode for the Test Agent

To be able to dedicate as many CPUs as possible to the Test Agent VNF, you need to configure the device in a custom mode where the Layer 2 and Layer 3 infrastructure is disabled. The operation requires the device to be rebooted for the changes to be applied. Run the command below and then type “yes”.

```
root> configure
root# set vmhost mode custom paa layer-2-infrastructure offline
root# set vmhost mode custom paa layer-3-infrastructure offline
root# commit
root> request vmhost mode paa
```

Use this command to get more information about mode and settings:

```
root> show vmhost mode
```

Note that provisioning the interfaces to the VNF is done as part of configuring the VNF.

4.2.5.5 Configuring NTP

The NFX150 has an NTP client and server running. You need to configure NTP on the NFX150 as described here: https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/ntp-configuring-nfx.html

Further, NTP needs to be configured as usual for the Test Agent. This can be done either from *Control Center* (page 189) or from the Test Agent *local console* (page 214). Follow the links for instructions.

4.2.5.6 Configuring the Test Agent VNF

The management interface for the TA is `eth1`, which is bridged to the management port of the NFX150. The remaining interfaces are mapped to the physical ports of the NFX150.

The Test Agent VNF can be configured either manually or by means of a cloud-config template. The two methods are covered in turn below.

Configuring the Test Agent VNF manually

1. Ensure the NFX150 is in compute mode with L2 and L3 infrastructure disabled:

```
set vmhost mode custom netrounds layer-2-infrastructure offline
set vmhost mode custom netrounds layer-3-infrastructure offline

request vmhost mode netrounds
```

   The system will prompt for a reboot if necessary.

2. Download the Test Agent image:

```
file copy https://<NCC address>/static/test_agent/paa-test-agent<current version␣
↪number>.qcow2
  /var/public/
```

3. Create a directory for the VNF:

```
file make-directory /var/public/test-agent-vnf
```

4. Make a copy of the Test Agent image so that the original can be reused to restore the initial state:

```
file copy /var/public/paa-test-agent<current version number>.qcow2
  /var/public/test-agent-vnf/paa-test-agent.qcow2
```

5. Allocate hugepages:

```
edit
set system memory hugepages page-size 2 page-count 512
```

6. Create the VNF and set the image to be used:

```
edit virtual-network-functions testagent
set image /var/public/test-agent-vnf/paa-test-agent.qcow2
```

7. Pin the vCPU on the NFX-150-C-S1:

```
set virtual-cpu 0 physical-cpu 2
set virtual-cpu 1 physical-cpu 3
set virtual-cpu count 2
set virtual-cpu features hardware-virtualization
set virtual-network-functions testagent emulator physical-cpu 1
set virtual-network-functions testagent iothread 1 physical-cpu 1
set iothread count 1
```

8. In the NFX150 default configuration, interfaces are mapped to the vmhost. Remove these mappings so that they are free to use by the Test Agent:

```
delete vmhost virtualization-options
```

9. Map all physical interfaces to the VNF as SR-IOV interfaces:

```
set interfaces eth2 mapping interface heth-0-0 virtual-function
set interfaces eth3 mapping interface heth-0-1 virtual-function
set interfaces eth4 mapping interface heth-0-2 virtual-function
set interfaces eth5 mapping interface heth-0-3 virtual-function
```

(continues on next page)

(continued from previous page)

```
set interfaces eth6 mapping interface heth-0-4 virtual-function
set interfaces eth7 mapping interface heth-0-5 virtual-function
```

10. Allocate memory to the VNF. (“1048576” = 1 GB is what is available on the NFX150-C-S1.)

```
set memory size 1048576
set memory features hugepages page-size 2
```

11. You are done. Commit the config. This should start up the VM:

```
commit
```

12. Access the Test Agent using the local console:

```
run request virtual-network-function console testagent
```

13. Register the Test Agent according to the instructions *here* (page 205).

Using the cloud config template to configure the Test Agent

As part of the Test Agent release, Juniper Networks also provides a base configuration and some template files for configuring the Test Agent VNF on the NFX150 device. This can be used to load the VNF configuration as well as the base for a config-drive used to set up the Test Agent; for example, to register and configure the management interface.

Before downloading the files, you need to ensure the NFX150 is in compute mode with L2 and L3 infrastructure disabled:

```
set vmhost mode custom netrounds layer-2-infrastructure offline
set vmhost mode custom netrounds layer-3-infrastructure offline

request vmhost mode netrounds
```

The system will prompt for a reboot if necessary.

Below are step-by-step instructions on how to set up the Test Agent. You can download the required files from the Control Center either using a script from root console or using the NFX150 CLI. The two possibilities are covered in separate subsections.

Downloading files using a script

To download the downloader script from Control Center, use the following command in the root shell of the NFX150.

```
curl --insecure https://<Control Center address>/static/test_agent/config-data/paa-
↪get-files.sh --output paa-get-files.sh
```

Replace `<Control Center address>` with the address to the Control Center where you have an account.

Then execute the script to download the files needed for installation as follows:

```
sh paa-get-files.sh <Control Center address> <version>
```

Again, replace `<Control Center address>` with the address to the Control Center where you have an account, and replace `<version>` with the Test Agent version available for download from that Control Center.

After the download is complete, proceed to apply the configuration in the NFX150 CLI as explained *here* (page 84).

Downloading files using the NFX150 CLI

1. Download the Test Agent image to the device:

```
file copy https://<Control Center address>/static/test_agent/paa-test-agent_
↪<current version
  number>.qcow2 /var/public/
```

   Replace `<Control Center address>` with the address of the Control Center where you have an account. `<current version number>` is the version number of the Test Agent hosted on the Control Center; a direct link to this can be found by clicking Test Agents on the navigation bar, then clicking the Download button.

2. Set up a directory on the NFX150 file system where the Test Agent image will be stored:

```
file make-directory /var/public/test-agent-vnf
```

3. Make a copy of the Test Agent image. This lets you reuse the original if you want to redo the setup with a clean Test Agent. The copy is stored in the newly created directory.

```
file copy /var/public/paa-test-agent<current version number>.qcow2
  /var/public/test-agent-vnf/paa-test-agent.qcow2
```

4. Download the Test Agent VNF base configuration:

   Choose a template matching the NFX150 device and configuration you are after. This is the config of the VNF on the NFX150 device. The config contains CPU pinning and mapping of interfaces to the Test Agent VNF. The following configs are available (please note again that Test Agent installation is currently supported only on the NFX150-C-S1 device):

Config file	Target device	CPUs	Memory (GiB)
TA-NFX150-C-S1_1-core.conf	NFX150-C-S1	1 (CPU2)	1
TA-NFX150-C-S1_2-core.conf	NFX150-C-S1	2	1
TA-NFX150-S1.conf	NFX150-S1	4	4

   The 2 CPU configuration is recommended.

   In the following example the config file `TA-NFX150-C-S1_2-core.conf` is downloaded:

```
file copy https://<Control Center address>/static/test_agent/config-data/TA-
↪NFX150-C-S1_2-core.conf
  /var/public/test-agent-vnf/
```

5. Create a config-data directory:

   In order to pass cloud-config to the Test Agent, we need to set up a directory that will be mounted in the Test Agent and picked up by cloud-init as cloud-config. This drive setup is not tied to any specific Test Agent VNF instance, so you do not need to repeat these steps if you set up a new Test Agent on the device later on.

   Create the directory where the config data will be stored in the NFX150 file system:

```
file make-directory /var/public/ta-config-drive
```

   Then download the cloud-init user-data template specific to Junos OS and a static metadata file that will be used to pass config to the Test Agent VNF instance:

```
file copy https://<Control Center address>/static/test_agent/config-data/user-
↪data.template
  /var/public/ta-config-drive/
file copy https://<Control Center address>/static/test_agent/config-data/meta-data
  /var/public/ta-config-drive/
```

Applying the configuration

It is now time to apply the configuration to the device and spin up the Test Agent.

Enter configuration mode on the NFX150 CLI:

```
configure
```

Load the configuration that you have downloaded:

Note: If you are configuring some other device than the *-C-S1 with 2 CPUs, the file name will be different.

```
load set /var/public/test-agent-vnf/TA-NFX150-C-S1_2-core.conf
```

You can inspect what has been configured so far by running

```
show | compare
```

Before you commit the configuration you need to set the required parameters that will be part of the cloud config. Below, in order to make the configuration commands shorter, we edit the configuration at the `parameters` level of the config. This is where the registration information and management interface configuration are set.

Navigate to the `parameters` subelement:

```
edit virtual-network-functions testagent config-data source template cloud-config␣
↪parameters

[edit virtual-network-functions testagent config-data source template cloud-config␣
↪parameters]
   root#
```

Configure the required parameters:

The file `user-data.template` holds descriptions of all available parameters. You can show these by running

```
run file show /var/public/ta-config-drive/user-data.template
```

Setting the required registration information

Run these commands:

```
set name <Test Agent name>
set server <Control Center server address>
set email <email>
set password <password>
set account <account>
```

Values enclosed in angle brackets <> should be replaced with information specific to your account. For example, naming the Test Agent "NFX-demo" would be done by typing:

```
set name NFX-demo
```

Setting the management interface configuration

The default management interface on the NFX150 device is preset to `eth1` in the provided VNF config. Therefore, if you do not set any cloud-config parameters in the CLI, the Test Agent will use `eth1` with DHCP as default address config. Below are some examples of how to configure this differently:

Suppose you want to use `eth2` for management and configure a static address. While editing the config at `virtual-network-functions testagent config-data source template cloud-config parameters`, run these commands:

```
set management_interface eth2
set management_address_type static
set management_ip 10.0.0.10/27
set management_dns 10.0.0.1
set management_default_gateway 10.0.0.1
```

Refer to the template file for the complete set of available config parameters.

Committing the configuration

Before committing the config it is a good idea to verify that the config is correct. Do this by running

```
show
```

Once you are satisfied with the configuration, commit it:

```
commit
```

This will apply the configuration and start the Test Agent VNF. The Test Agent should then take the config passed from the CLI and register to the configured Control Center.

4.2.5.7 Fetching system information

You can inspect the current system configuration of the NFX150 with the command

```
show configuration virtual-network-functions testagent
```

This command can be used to display information about CPU configuration:

```
show system visibility cpu
```

Here is how to view information on memory:

```
show system visibility memory
```

For further information, see https://www.juniper.net/documentation/en_US/junos/topics/reference/command-summary/show-system-visibility-cpu-nfx-series.html

4.2.5.8 Managing VNFs on the NFX150 device

How to start, stop, restart, and get console access to VNFs is covered on this page:

https://www.juniper.net/documentation/en_US/junos/topics/topic-map/managing-vnfs-nfx-series.html

Note: Whenever you modify the configuration of the Test Agent VNF, you need to restart it for the changes to take effect.

4.2.5.9 Console access to VNFs

To get a serial console for a VNF, use the following command:

```
request virtual-network-function console <vnf-name>
```

For the Test Agent, this becomes

```
request virtual-network-function console testagent
```

(as mentioned in the section on *manual VNF configuration* (page 81)). Continue according to *this page* (page 209).

Note: The `request virtual-network-function console` command is supported only for root login over SSH.

4.2.5.10 Junos OS upgrade/recovery

If the Test Agent installation fails, we recommend reinstalling Junos OS from USB according to the document "Performing a USB upgrade on NFX150" which is found here: https://kb.juniper.net/InfoCenter/index?page=content&id=KB32971

By contrast, performing a factory reset is not recommended.

4.2.5.11 Interface mapping

In the provided configuration example and templates, all physical ports of the NFX150 are made available to the Test Agent VNF as shown in the table below.

Test Agent interface	Physical port	Internal name
eth0	Internal management	–
eth1	Shared with management port	fxp0
eth2	Port 1 (1 Gbps)	heth-0-0
eth3	Port 2 (1 Gbps)	heth-0-1
eth4	Port 3 (1 Gbps)	heth-0-2
eth5	Port 4 (1 Gbps)	heth-0-3
eth6	Port 5 (10 Gbps)	heth-0-4
eth7	Port 6 (10 Gbps)	heth-0-5

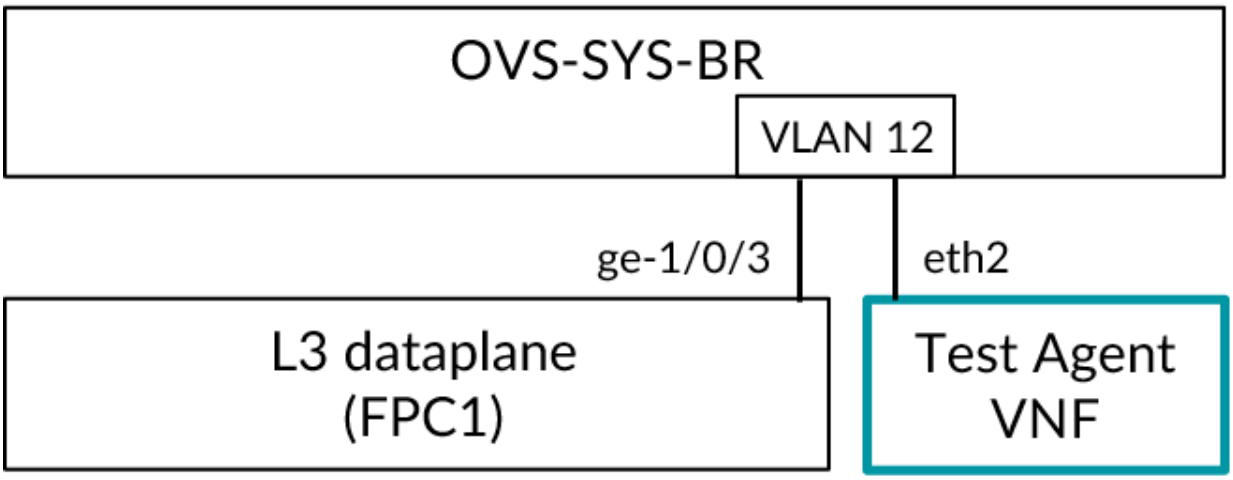
Further information on how to configure the interface mapping on the NFX150 can be found here: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/interface-mapping-nfx150.html

4.2.5.12 Updating the Test Agent software version

No special steps are needed when updating a Test Agent running on this platform. The procedure is the same as on all other platforms and is handled from Control Center. See *this page* (page 165).

4.2.5.13 Connecting the Test Agent to the Layer 3 dataplane

If Layer 3 traffic is being routed via the NFX150, it can be useful to connect a virtual interface on the Test Agent to the L3 dataplane. For this purpose, a virtio interface is mapped to a VLAN and attached to the Test Agent. One of the `ge-1/0/x` interfaces is configured on the same VLAN with an IP address that the Test Agent can use as a gateway. The image below shows an example.



1. Create the VLAN:

```
set vmhost vlans vlan12 vlan-id 12
```

2. Attach an interface from FPC1 to the OVS bridge. On the NFX150, `ge-1/0/0` is already attached, but any of the other `ge-1/0/x` interfaces can also be attached.

```
set vmhost virtualization-options interface ge-1/0/3
```

3. Add a logical interface to the interface and map it to the VLAN:

```
set interfaces ge-1/0/3 vlan-tagging
set interfaces ge-1/0/3 unit 12 vlan-id 12
```

4. Add the gateway address:

```
set interface ge-1/0/3 unit 12 family inet address 192.168.12.1/24
```

5. Add the interface to the `trust` security zone:

```
set security zones security-zone trust interfaces ge-1/0/3.12
```

6. Map the virtio interface to the VLAN (we are assuming here that the VNF is named `testagent`):

```
set virtual-network-function testagent interfaces eth2 mapping vlan members vlan12
```

7. Commit.
8. From Control Center, configure `eth2` on the Test Agent with an address from the 192.168.12.0/24 subnet and gateway 192.168.12.1.

4.2.5.14 Removing the Test Agent VNF from the NFX150 device

All of the following steps are executed from the NFX150 CLI.

1. First stop the instance:

```
request virtual-network-function stop testagent
```

2. Enter configuration mode and delete the config for the VNF:

```
root> configure
Entering configuration mode

[edit]
root# delete virtual-network-functions testagent

[edit]
root# commit
```

3. If you want to set up a new Test Agent later on, it is important to *replace the image file* that was used since the image will otherwise hold the state of the deleted Test Agent:

```
file copy /var/public/paa-test-agent_<current version number>.qcow2
  /var/public/test-agent-vnf/paa-test-agent.qcow2
```

   Then follow the instructions under *Applying the configuration* (page 84) to set up a new Test Agent.

4.2.5.15 Known platform limitations

There are currently certain limitations on the testing that is possible on the NFX150-C-S1 platform.

- The single-core architecture limits the throughput, as detailed on *this page* (page 237).
- The number of VLANs is limited to 31 for 1 Gbit/s interfaces and to 63 for 10 Gbit/s interfaces.
- Forwarding through a bridge is not supported.
- The IPTV inline task type, using the configuration described in the present documentation, is not supported as it relies on a bridge.
- 802.1x authentication is not supported on the NFX150 management interface (`fxp0`). It is however supported on the test interfaces, and does not require any special configuration of the NFX150. How to configure 802.1x authentication on the Test Agent is explained *here* (page 170).
- Transparency tests are limited on test interfaces, as detailed below. The tests are all supported on the NFX150 management port (`virtio`).
- Testing with jumboframes is not possible on the NFX150 management port. It is supported on the test interfaces.
- Separate IP addresses are needed for management of the NFX150 and of the Test Agent.

Details on transparency tests

Test type	Sender				Receiver			
	ethX		ethX.x (VLAN interf.)		ethX		ethX.x (VLAN interf.)	
	1G	10G	1G	10G	1G	10G	1G	10G
Custom VLAN	Fail	Fail	Pass	Pass	Fail	Fail	Pass	Pass
Ethertypes	Pass	Pass	Pass	Pass	[1]	[1]	[1]	[1]
MAC address limit	Fail	Fail	Fail	Fail	Pass	Pass	Pass	Pass
VLAN	Fail[2]	Fail[2]	N/A	N/A	Fail[2]	Fail[2]	N/A	N/A
Custom Ethertypes	Pass	[3]	Pass	[3]	[4]	[4]	[4]	[4]
DSCP remapping	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
IPv6	Pass	Pass	Pass	Pass	Fail[5]	Fail[5]	Fail[5]	Fail[5]
IP	Pass	Pass	Pass	Pass	Fail[6]	Fail[6]	Fail[6]	Fail[6]
Multicast	Pass	Fail[7]	Pass	Fail[7]	Fail[8]	Fail[8]	Fail[8]	Fail[8]
Ethernet control protocols	Pass	Fail[9]	Pass	Fail[9]	Fail[10]	Fail[10]	Fail[10]	Fail[10]

4.2.5.16 Reference performance

See *this page* (page 237).

4.2.6 Installing Test Agent Appliance on a PC Engines APU2 hardware server

4.2.6.1 Introduction

This page explains how to build a Routing Active Testing Test Agent on a PC Engines (► https://www.pcengines.ch) APU2 hardware server. The page lists the required hardware and describes how to assemble it and how to install the Test Agent software.

A quick summary of the procedure is as follows:

- Get the hardware components.
- Assemble the hardware components.
- Install the Test Agent software.

Note: The procedure given here must be followed strictly; otherwise, errors may result.

[1] Not CDP, Cisco VTP/DTP, or CGMP.
[2] All QinQ tests pass.
[3] Passes only when source MAC matches that of sending interface.
[4] Passes only when destination MAC matches that of receiving interface.
[5] Fails on MLD, MLDv2, neighbor solicitation, router solicitation, DHCP solicitation.
[6] Fails all multicast.
[7] STP and LLDP fail.
[8] Fails STP, LLDP, Ethernet configuration test protocol and MPLS (except MAC 01:00:5E:80:00:01).
[9] Link Aggregation Control Protocol fails.
[10] Only EAP test passes.

4.2.6.2 Step 1: Obtaining the hardware components

The hardware server to be built is PC Engines APU2.

The hardware consists of the following components (auxiliary items such as screws and thermal paste are left out of the tables). Order the components from your preferred hardware supplier.

Required components

These components make up the basic APU2 hardware server.

Qty	Component	Brand	Model/Version
1	System board	PC Engines	apu2e2[1]
1	Enclosure	PC Engines	case1d2u
1	Power brick adapter	PC Engines	ac12veur3 (EU), ac12vus2 (US), ac12vuk2 (UK)
1	SSD	PC Engines	msata16g: SSD M-Sata 16GB MLC, Phison S11 controller (for APU)

Optional components

Wi-Fi card and antenna

Qty	Component	Brand	Model/Version
1	Wi-Fi modem	Intel	Intel Dual Band Wireless-AC 8265, model 8265 NGWMG
1	Adapter	(generic component)	M.2 2230 to MiniPCIe adapter
1	Antenna cable	(generic component)	MHF4 to RP-SMA (female), 4 or 5 inches (10 to 11 cm)
1	Antenna	(generic component)	Wi-Fi antenna, MHF4 2.4/5g (20 cm)

4.2.6.3 Step 2: Assembling the hardware

Necessary tools: Phillips-head and flat-head screwdrivers, scissors, tweezers.

- Rip off the head spreader kit from the box and open it.
- There is one aluminum heat spreader in the bag and two blue heat conductive pads. We will need only one heat conductive pad for the assembly. Keep the other one for future use.
- Open the chassis. You should find a small bag inside the chassis containing eight screws, two plugs and four rubber legs.
- Make sure to use the four bigger screws for the board and the four smaller ones for the chassis.
- The heat spreader should be placed 35 mm from the left and 10 mm from the bottom. *(In the video, a special fixture is used to achieve precise placement of the heat spreader. This item is not mandatory but is recommended if you are building a large batch. It can be purchased at the PC Engines web store. Use a ruler if you do not have the fixture.)*
- Take the board out of the bag and remove the two hex screws from the serial port. If you do not have a hex bit, use your fingers and/or pliers.
- Turn the board upside down and place it on the bubble foil for protection.

[1] Note on naming syntax: (a) “apu2” is the board and product name; (b) “e” is the revision; (c) “2” indicates the amount of RAM used.

- There are two heat conductive pads on the strip. We will need only one.
- Peel the blue foil from one side and place the pad on the CPU. Use tweezers to peel the foil from the other side of the pad.
- Now it is time to mount the board into the chassis. Make sure that the conductive pad is placed firmly against the heat spreader.
- Use the four larger screws to fasten the board to the chassis. Use a Phillips-head screwdriver (the one with the cross). The two screws near the connector panel are somewhat tricky, because they need to be inserted into a very narrow space.
- Install the two hex screws, one on each side of the serial port.
- Install the SSD into the mSata slot. There are three slots, but the SSD will work only in the mSata slot. The mSata slot is the one to the right on the board (with the connector panel facing upwards).
- Push the drive into the slot at an angle, and then down. The drive should clip in.
- Make sure that the two metal latches hold the SSD firmly in place. Use a toothpick or screwdriver to bend them. If the SSD is not firmly placed, it may fall out.
- If you want to install a Wi-Fi card, do so now. Follow the installation instructions for that component *below* (page 91).
- Now close the chassis using the four small screws.
- There are two small plugs to plug the antenna holes. If you have not installed Wi-Fi on your system, install these plugs.
- If you do not have a wall mount, you can install the four rubber legs that come with the chassis.

Step 2a: Installing a Wi-Fi card and antenna

- Connect the Intel Wi-Fi card to the M2 adapter.
- Attach the M2 adapter with the Wi-Fi card on it to the slot farthest from the mSata slot.
- Gently connect the antenna cables to the Wi-Fi card module.
- Slide the other end of each cable (the rubber antenna connector) through the case lid.
- Tighten the nut from the outside of the lid to firmly fasten the rubber antenna connector onto it.
- Screw the rubber antennas onto their respective connectors.

4.2.6.4 Step 3: Installing the Test Agent

Prerequisites

In this step you may need to access the BIOS on the PC Engines server. To this end, you need to connect to the server via a serial interface, so you require:

- An RS-232-to-USB adapter.
- A terminal emulator such as PuTTY (for Windows) or Terminal (for macOS).

The bit rate 38400 bit/s is required when first connecting to the PC Engines server.

- In PuTTY, set the bit rate under Connection > Serial > Speed (baud).
- In Terminal as well as in Linux, use the `cu` command with the syntax

```
cu -l /dev/<serial device to be used> -s 38400
```

Instructions

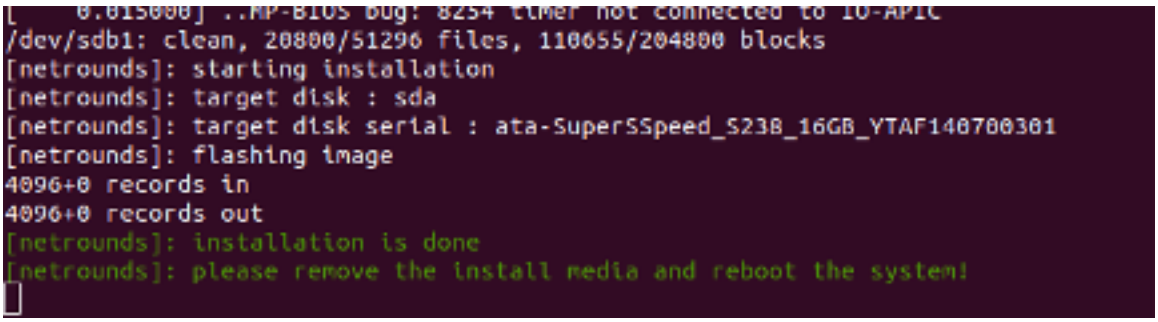
1. On a PC with internet access, log in to your account on your Routing Active Testing Control Center server.
2. In the left-hand pane, click Test Agents.
3. Click the Download button in the top right corner.
4. Click the link “RAW disk image (.img.gz)” to download that Test Agent disk image.
5. The next step is to write the disk image to a USB memory stick using the Etcher utility (which exists in Windows, macOS, and Linux versions).

- Download the appropriate version of Etcher from ► https://etcher.io and install it.
- Insert a USB memory stick with at least 4 GB of free space into your PC.
- Open Etcher, and from your hard disk select the Test Agent disk image (netrounds-test-agent_<version number>.img.gz).
- Select the USB memory stick in order to write the disk image to it.
- Review your selections and click Flash!

1. Once you have burned the Test Agent disk image to the USB memory, insert it into a USB port on the PC Engines hardware server.
2. The BIOS will by default boot from USB. However, if the BIOS boot priority has been changed, you need to access the BIOS and restore the default settings. This is described in the steps below.

- Connect to the PC Engines hardware server using a terminal emulator (see *Prerequisites* (page 91) above). Note that the bit rate 38400 bit/s must be used in this step.
- Before accessing the BIOS, change the bit rate to 115200 bit/s.
- Access the BIOS boot menu, and make sure the USB memory comes before the hard disk in the boot sequence.
- Change the bit rate back to 38400 bit/s.
- Select USB boot from the BIOS boot menu.

1. The boot process takes about 20 seconds. When the login prompt is shown, log in as user “admin” with password “admin”.
2. Go to Utilities.
3. Select Install to disk.
4. When the procedure has finished, the screen will look like this:



1. Power down the PC Engines hardware server.
2. Remove the USB memory stick, then boot up the server without the USB memory stick connected and confirm that the Test Agent is installed.

4.2.7 Installing Test Agent Appliance on a Supermicro hardware server

4.2.7.1 Introduction

This page explains how to build a Routing Active Testing Test Agent on a Supermicro (► https://www.supermicro.com) hardware server. The document lists the required hardware as well as some optional expansions, and describes how to install the BIOS and the Test Agent software.

A quick summary of the procedure is as follows:

- Get the hardware components.
- Assemble the hardware components.
- Flash the BIOS.
- Install the Test Agent software.
- *(If applicable:)* Install optional hardware components.

Note: The procedure given here must be followed strictly; otherwise, errors may result. For example, if you install an extra network interface card before installing the Test Agent software, interface names may be assigned randomly to ports.

4.2.7.2 Step 1: Obtaining the hardware components

The hardware consists of the following components (auxiliary items such as screws and thermal paste are left out of the tables). Order the components from your preferred hardware supplier.

Required components

These components make up the basic Supermicro hardware server.

Qty	Component	Brand	Model/Version
1	Motherboard	Supermicro	X11SSH-LN4F
1	BIOS	Supermicro	2.0b
1	CPU	Intel	Intel Xenon E3-1240v6 3.7/4.1 GHz Quadcore HT, 8 MB
			Part number: BX80677E31240V6
1	Heat sink	Supermicro	SNK-P0046P
2	RAM	Kingston	Kingston 4 GB DDR4 ECC 2400 MHz UDIM
			Part number: KVR24E17S8/4
1	HDD	Supermicro	SDD-DM016
1	Power supply	Supermicro	PWS-203-1h
1	Chassis	Supermicro	Superchassis 510-203B
2	Fan	Nidec	W40S-A5

Required configuration

- The IPMI port must be sealed. IPMI must be turned off for the LAN ports.
- Auto power-on must be enabled in BIOS.
- PXE boot must be disabled.

Optional components

You can install either of the network interface cards listed in this section (but not both).

10 Gbit network interface card

Qty	Component	Brand	PID	Model/Version
1	4-port 10 Gbit network interface card	Supermicro	HW-NIC20-P	Supermicro Intel X710 4-port SFP+ 10GbE PCIe 3.0 x8 LP AOC-STG-i4S
1	Network riser	Supermicro		RSC RR1UE16

SFP+ transceivers

Qty	Component	Brand	PID	Model/Version
1	SFP+ transceiver	Intel	HW-SFP+L-P	Intel SFP+ LR Optics 10GBASE-LR/1000BASE-LX
1	SFP+ transceiver	Intel	HW-SFP+S-P	Intel SFP+ SR Optics 10GBASE-SR/1000BASE-SX

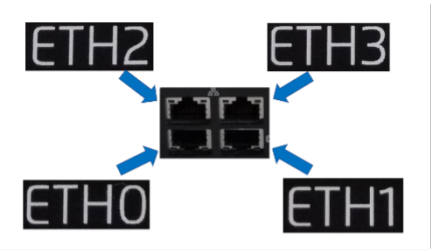
4-port 1 Gbit network interface card

Qty	Component	Brand	PID	Model/Version
1	4-port 1 Gbit network interface card	Supermicro	HW-NIC1-P	Supermicro Intel AOC-SGP-I4
1	Network riser	Supermicro		RSC RR1UE16

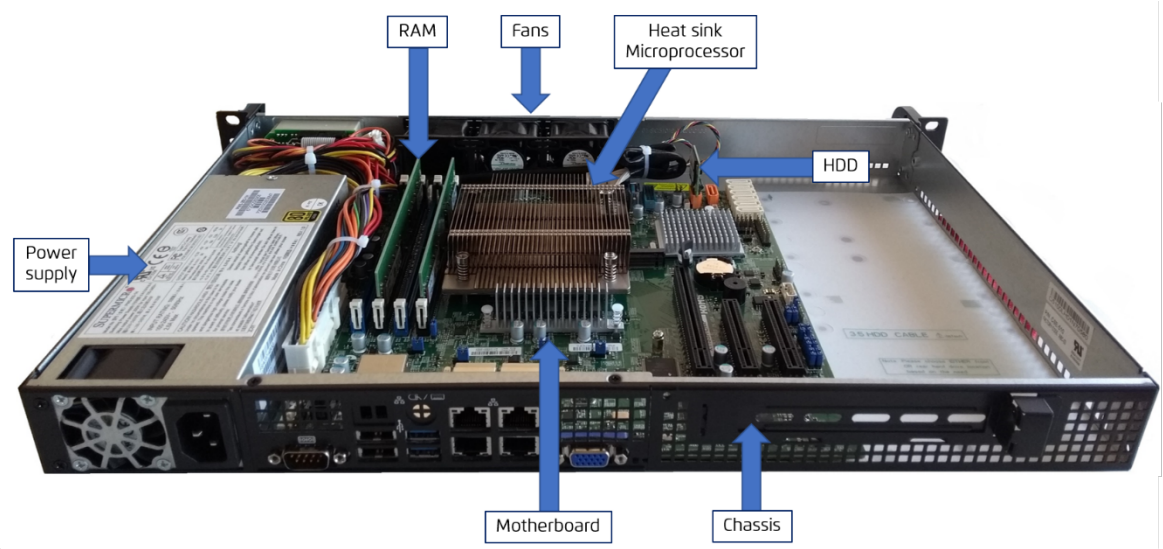
4.2.7.3 Step 2: Assembling the hardware

How to assemble the hardware components is not described in detail here and is assumed to be known. However, note the following:

- Thermal paste must be applied between the CPU and heat sink during installation of the CPU.
- Any expansions must be installed at the end of the procedure (see *Step 5* (page 97)).
- The assignment of interfaces to ports appears from Figure 1. Label the ports according to this image.



The image below shows a Supermicro hardware server with all components in *Step 1* (page 93) installed.



4.2.7.4 Step 3: Installing the correct BIOS version

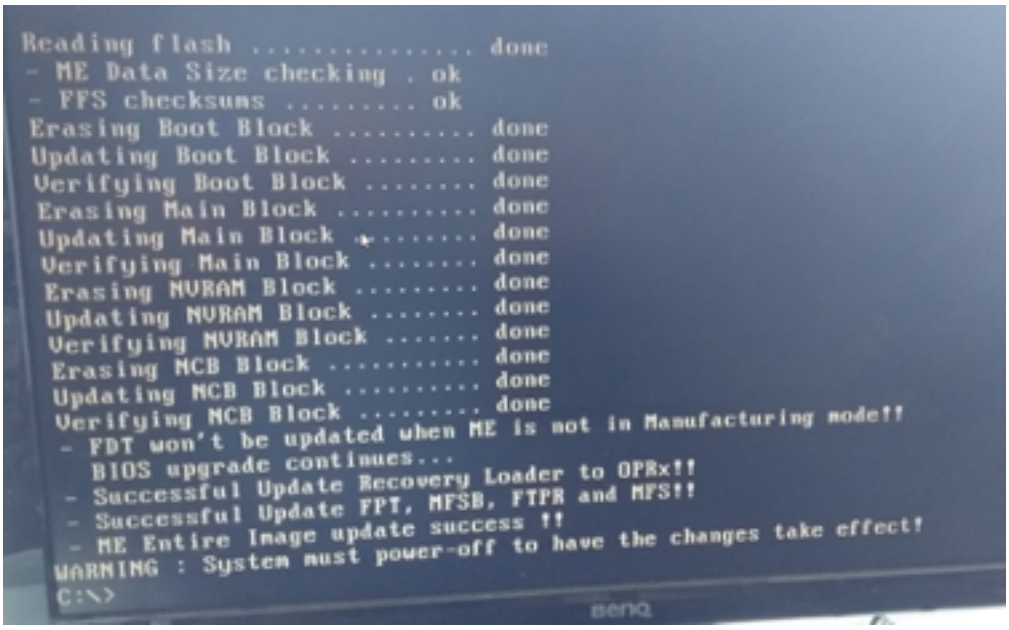
The BIOS version to be used on the Supermicro X11SSH-LN4F motherboard is version 2.0b. This is the BIOS version used in testing Routing Active Testing, so it is necessary to use this version in order to ensure performance. Please contact Juniper Networks in order to obtain the correct BIOS.

Then do as follows to update the BIOS to the correct version:

- The `Readme.txt` file provided by Supermicro has instructions on how to create a bootable FreeDOS USB. Follow these instructions.
- Once this is done, here is how to update the BIOS (for the version 2.0b):
- Start up the Supermicro with
  - a keyboard plugged in
  - a bootable USB with BIOS (created in *Step 1* (page 93)) connected
  - a VGA monitor connected
- When the “Supermicro” screen shows up, press F11 to enter the booting menu.
- Select the USB memory stick (usually identified here by the USB vendor’s name).
- Select Default (the only option available).
- Select “FreeDOS Live CD only”.
- Execute the following commands to see the correct file to use:

```
C:                          // (Press Enter)
dir                         // Displays files
flash.bat X11SSH7.727       // Flashes the BIOS for this specific model
```

1. When the `flash.bat` command finishes, it displays the following output:



1. After the flashing procedure, the Supermicro server may restart twice before starting up as normal.

4.2.7.5 Step 4: Installing the Test Agent

1. On a PC with internet access, log in to your account on your Routing Active Testing Control Center server.
2. In the left-hand pane, click Test Agents.
3. Click the Download button in the top right corner.
4. Click the link “RAW disk image (.img.gz)” to download that Test Agent disk image.
5. The next step is to write the disk image to a USB memory stick using the Etcher utility (which exists in Windows, macOS, and Linux versions).

- Download the appropriate version of Etcher from ► https://etcher.io and install it.
- Insert a USB memory stick with at least 4 GB of free space into your PC.
- Open Etcher, and from your hard drive select the Test Agent disk image (`netrounds-test-agent_<version number>.img.gz`).
- Select the USB memory stick in order to write the disk image to it.
- Review your selections and click Flash!

1. Once you have burned the Test Agent disk image to the USB memory, insert it into a USB port on the Supermicro hardware server.
2. Access the BIOS boot menu.
3. Make sure the USB memory comes before the hard drive in the boot sequence.

4. Select USB boot from the BIOS boot menu.
5. The boot process takes about 20 seconds. When the login prompt is shown, log in as user “admin” with password “admin”.
6. Go to Utilities.
7. Select Install to disk.
8. When the procedure has finished, the screen will look like this:

```
[    0.815008] ..MP-BIOS bug: 8254 timer not connected to IO-APIC
/dev/sdb1: clean, 20800/51296 files, 110655/204800 blocks
[netrounds]: starting installation
[netrounds]: target disk : sda
[netrounds]: target disk serial : ata-SuperSSpeed_S238_16GB_YTAF140700301
[netrounds]: flashing image
4096+0 records in
4096+0 records out
[netrounds]: installation is done
[netrounds]: please remove the install media and reboot the system!
```

1. Power down the Supermicro hardware server.
2. Remove the USB memory stick, then boot up the server without the USB memory stick connected and confirm that the Test Agent is installed.

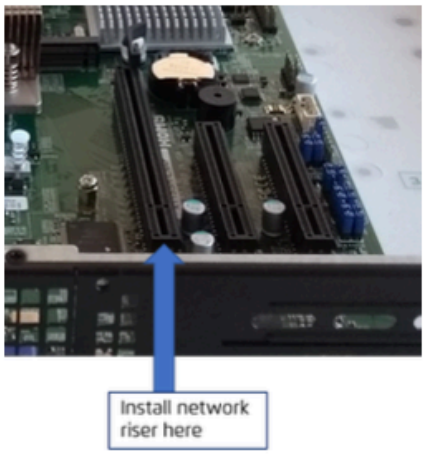
4.2.7.6 Step 5: Installing optional components

Once the Test Agent software is installed, you can install one of the hardware expansions (10 Gbit network interface card, 4-port 1 Gbit network interface card).

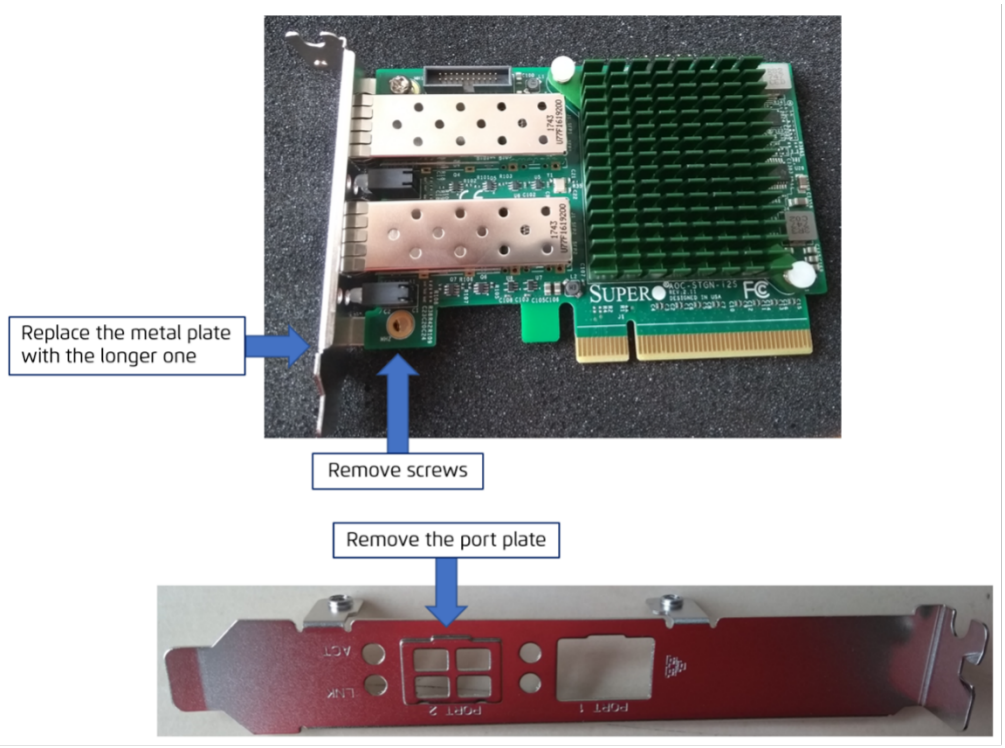
Note: This must be left until after the Test Agent software has been installed. If you install an extra network interface card before the Test Agent software, interface names may be randomly assigned to ports (for example, “eth0” to a port on the motherboard, “eth1” to a port on the extra card, etc.).

If you want to install an extra network interface card, you also need to install a network riser in the slot indicated in the following image:





For the network interface card itself, you need to exchange the metal plate at the front for a longer one as shown in the figure below:



Finally, this picture shows a Supermicro hardware server with network riser and 10 Gbit network interface card mounted:



4.2.8 Installing Test Agent Appliance on a fitlet2

This page explains how to install a Routing Active Testing Test Agent on a Compulab (► https://www.compulab.com) fitlet2 hardware server.

4.2.8.1 Obtaining the hardware components

Some Juniper resellers may be able to resell Compulab. Please ask your local Juniper reseller.

Alternatively, you can order preinstalled Test Agents

- directly from Compulab: Email to sales@fit-iot.com
- from a Compulab reseller: See https://fit-iot.com/web/resellers/

In your order, refer to the SKUs listed *below* (page 100).

Required components

fitlet2 is equipped with 2x1GE interfaces (Intel i211) with support for hardware timestamping.

Two hardware models are recommended: *cost-optimized* and *performance-optimized*. Their respective features appear from the table below.

- The cost-optimized model is comparable to the *PC Engines APU2* (page 89).
- The performance-optimized model is recommended for advanced use cases.

Feature	Cost-optimized	Performance-optimized
CPU	Intel Apollo Lake CPU	Intel Apollo Lake CPU
	Atom x5-E3930 (2 cores)	Atom x7-E3950 (4 cores)
RAM	4 GB	8 GB
Storage	32 GB SSD	32 GB SSD
Chassis	Standard	Standard
Temperature range	Commercial	Commercial
Power supply	Standard	Standard

Optional components

Two expansions are available: Wi-Fi and 2×1GE. Either of these can be used, but not both at the same time.

Wi-Fi expansion

- Intel 8260AC
- 2×2 802.11ac for 2.4 and 5 GHz
- Up to 867 Mbps
- 2×RP-SMA antennas

2×1GE expansion

- Two additional 1GE interfaces
- Intel i211
- Provides a total of 4×1GE interfaces

4.2.8.2 SKUs

These SKUs refer to fitlet2 units with Test Agent software preinstalled, which is indicated by the suffix "-XJUN".

SKUs *without* this suffix refer to fitlet2 units without Test Agent software.

Cost-optimized model

Expansion	SKU
None	FITLET2-CE3930-D4-M32S-XJUN
Additional 2×1GE interfaces	FITLET2-CE3930-D4-M32S-FLAN-XJUN
Additional Wi-Fi	FITLET2-CE3930-D4-M32S-WI8260-XJUN

Performance-optimized model

Expansion	SKU
None	FITLET2-CE3950-D8-M32S-XJUN
Additional 2x1GE interfaces	FITLET2-CE3950-D8-M32S-FLAN-XJUN
Additional Wi-Fi	FITLET2-CE3950-D8-M32S-WI8260-XJUN

4.2.8.3 Manually installing a Test Agent image on a fitlet2

These instructions apply if you have purchased a fitlet2 without Test Agent software preinstalled (see SKUs *here* (page 100)) and you want to install the Test Agent image manually, or if you want to reset a fitlet2 to the factory default state.

- Make sure the BIOS is set to boot in legacy boot mode. Make the following settings in the BIOS menu:
  - Advanced > CSM Configuration > CSM Support : **[Enabled]**
  - Advanced > CSM Configuration > Boot option filter : **[Legacy only]**
  - Advanced > CSM Configuration > Network : **[Legacy]**
  - Advanced > CSM Configuration > Storage : **[Legacy]**
  - Advanced > CSM Configuration > Video : **[Legacy]**
  - Advanced > CSM Configuration > Other PCI devices : **[Legacy]**
  - Boot > Boot Mode Selection : **[Legacy]**
  - Press the F4 key to **Save & Exit**.

- Prepare a USB memory stick with a Test Agent image, then boot and install according to this instruction on *this page* (page 73).

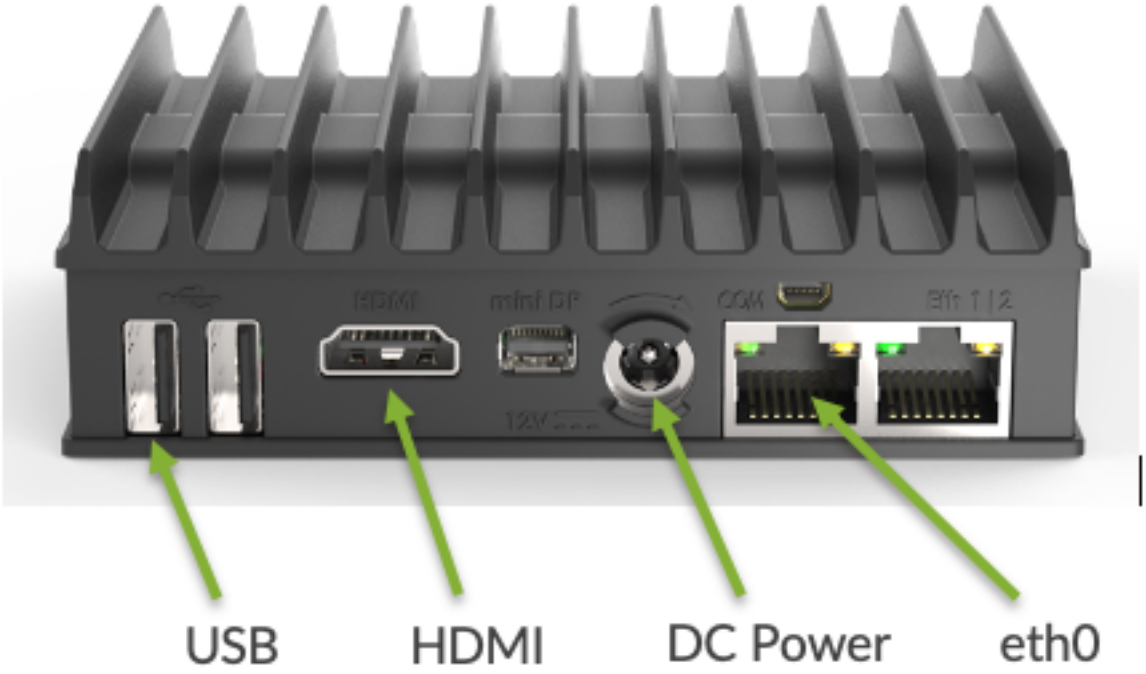
Known issues

- The RS232 serial console does not work with the fitlet2 device.
- During boot, the errors “Invalid GROUP operation” and “Failed to find module ‘autofs4” are briefly displayed. They can be safely ignored.

4.2.8.4 Getting started with the Test Agent on fitlet2

Once the Test Agent is installed on the fitlet2, here is how to get started using it:

- Connect an HDMI display, a USB keyboard, and power supply to the fitlet2 device.
- Connect eth0 to a network. This is the Ethernet connector next to the DC power connector.
- Start the fitlet2 by pressing the power button. After 1–2 minutes the login page should appear on the screen.
- Register the Test Agent with a Control Center following the *regular procedure* (page 205).



4.2.8.5 Interface mapping

eth0 is the default management interface for a Test Agent configured for DHCP.

The physical ports of the fitlet2 map to interfaces in Routing Active Testing as laid out in the tables below.

No expansion installed

Physical port	Interface name
1	eth0
2	eth1

Wi-Fi expansion installed

Physical port	Interface name
1	eth0
2	eth1
Expansion	wlan0

2×1GE expansion installed

Physical port	Interface name
1	eth0
2	eth1
Expansion	eth2, eth3

4.2.9 Installing a Test Agent Appliance on virtualization platforms: Introduction

This page gives an overview of how you can deploy a Test Agent on some popular hypervisor platforms. The various Test Agent images are provided either directly from Routing Active Testing or from your NFV orchestration partner. You upload and boot the Test Agent from the virtualization platform, and the Test Agent will automatically connect to Control Center.

Full information on how to do the deployment is found on separate pages:

- *AWS* (page 105)
- *Azure* (page 110)
- *Google Cloud* (page 116)
- *OpenStack* (page 122)
- *Oracle Cloud* (page 129)
- *VirtualBox* (page 141)
- *VMware* (page 148)

A generic description of the virtual Test Agent VNF is given *here* (page 158).

Requirements on Test Agent disk image format differ between platforms, as detailed below.

- **Amazon Web Services (AWS):** For this you need a Test Agent AMI (Amazon Machine Image), a special type of virtual appliance used to create a virtual machine within the Amazon Elastic Compute Cloud ("EC2").
- **Azure:** For this you need a Test Agent software disk image in VHD format, which can be deployed in the Microsoft Azure cloud computing service.
- **Google Cloud (GCP):** For this you need a Test Agent image in format `*.gcp.tar.gz`, which can be deployed on the Google Cloud platform.
- **OpenStack/KVM:** For this you need a Test Agent software disk image in RAW or QCOW2 format. You upload the Test Agent image to OpenStack and launch an instance of it, either through the OpenStack Horizon user interface (with or without a Heat Orchestration Template, HOT) or by means of a Python script using OpenStack Python APIs. HTTP Get or config-drive is used to transfer user data into the virtual machine running the vTA to provide day-0 configuration and information on how to connect to Control Center (domain name/IP address and port).
- **Oracle Cloud:** For this you need a Test Agent software disk image in QCOW2 or VMDK format, which can be deployed on the Oracle Cloud platform.
- **VirtualBox:** For this you need a Test Agent software disk image in OVA format. User data is not supported in VirtualBox, which means that you have to register the Test Agent manually to Control Center using the console once the Test Agent has booted.
- **VMware/vSphere:** For this you need a Test Agent software disk image in OVA format. You upload the Test Agent image to vSphere and start a Test Agent vApp in VMware. Configuration of user data is done via the vSphere web client or directly in the OVF file.

It may be noted that the RAW format can be converted to a variety of other formats using appropriate tools. If you have specific questions on this topic, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support. The Test Agent RAW image may be seen as a replacement for the ISO image which was provided in certain older (Netrounds) product versions.

4.2.9.1 Format of "cloud-config" metadata

On certain virtualization platforms, including AWS and OpenStack, "cloud-config" metadata must be entered specifying among other things how to connect to Control Center. The format of this metadata is given below.

Basic configuration:

Note: The `#cloud-config` and `paa_test_agent` lines must be present, and all of the remaining lines must be indented.

```
#cloud-config
paa_test_agent:
  name: MyvTA                                  # vTA name
  email: john.doe@example.com                  # NCC user email
  password: secret                             # NCC user password
  account: theaccount                          # NCC account (short name, found in NCC␣
↪URL)
  server: <login-server>:<port>                # NCC host and port
                                               # Default == SaaS: login.netrounds.
↪com:443
                                               # On-prem default port is 6000, e.g. my.
↪server.me:6000
```

(continues on next page)

(continued from previous page)

```
                                          #   Note: With an IPv6 server address␣
↪the whole string
                                          #         including port must be in␣
↪double quotes
  admin_password: secret                  # Admin user password. Use null to␣
↪disable.
  root_password: secret                   # Menu root shell access password. Use␣
↪null to disable.
                                          #
  management_interface: eth0              # Test Agent management interface
  management_mtu: 1500                    # MTU on management interface
  management_address_type: dhcp           # Can be "dhcp" or "static"

  ## Set the following if using "static" above:
  # management_ip: 192.168.1.2/24
  # management_dns: 8.8.8.8, 8.8.4.4
  # management_default_gateway: 192.168.1.1
  # management_ntp: time.google.com

  ## Set the following if using an HTTP proxy:
  # http_proxy: myproxy.lan
  # http_proxy_port: 80
  # http_proxy_auth_type: none              # Can be "none" or "basic"
  # http_proxy_username: johndoe
  # http_proxy_password: secret

  ## Note: IPv6 management requires special config, see separate documentation
  # management_enable_ipv6: False
  # management_ntp_allow_ipv6: False
  # management_address6_type: none          # Can be "dhcp", "slaac", or "static"

  ## Set if "static". Note: Use CIDR format for IP
  # management_ip6: 2001:db8:85a3::8a2e:370:7334/64
  # management_dns6: 2001:4860:4860::8888, 2001:4860:4860::8844
  # management_default_gateway6: <gateway IP address>
```

The parameters `management_ip` … `management_ntp` are required only if `management_address_type` is "static".

The parameters `http_proxy` … `http_proxy_auth_type` are required only if the vTA is connecting to the server through an HTTP proxy.

The parameters `http_proxy_username` and `http_proxy_password` are required only if `http_proxy_auth_type` is "basic".

The parameters `management_enable_ipv6` … `management_default_gateway6` are required only if IPv6 is used on the vTA management connection.

The parameters `admin_password` and `root_password` expect Unicode strings. The passwords can be text or a salted hash. They are optional. If these parameters are absent, the system will revert to its default behavior.

4.2.9.2 Interface assignment in a hypervisor

When a vTA is provisioned in a hypervisor, its interfaces are numbered in ascending order according to the hypervisor PCI IDs. That is, the interface with the lowest PCI ID will be called eth0, the next will be called eth1, etc.

Later on, however, an interface will keep its name even if its PCI ID changes, or if other interfaces are added or removed.

4.2.9.3 Hardware requirements

The compute resources available to the virtual Test Agent will impact its performance. Hardware requirements for virtual Test Agents are stated in the Test Agents datasheet; it is found at https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000684-en.pdf.

4.2.9.4 Notes on hypervisor resource management

Most hypervisors have a resource manager for sharing the underlying hardware among all host VMs. These resources are basically CPU, storage I/O, and network bandwidth. For more deterministic behavior, SR-IOV and CPU pinning can be considered, but that is not required for the Test Agent.

Please be aware that the resource manager may throttle some parameters, such as network bandwidth. To give just one example, the VM might think it has access to a 1 Gbit/s port, whereas the throttling only allows 80 Mbit/s throughput.

4.2.10 Installing Test Agents on virtualization platforms

4.2.10.1 Deploying an AMI Test Agent in AWS

Introduction

This page explains how to deploy a Routing Active Testing Test Agent in Amazon EC2 (Elastic Compute Cloud) by launching an AWS instance on which to run a Routing Active Testing AMI (Amazon Machine Image).

Prerequisites

Control Center account

You need an account in a Routing Active Testing Control Center in order to access it: either the one belonging to the Routing Active Testing SaaS solution or one installed on-premise in your organization. If you do not already have a Routing Active Testing account, please contact your Juniper partner or your local Juniper account manager or sales representative.

Routing Active Testing AMI for Test Agent

An Amazon Machine Image (AMI) is a special type of virtual appliance used to create a virtual machine within the Amazon Elastic Compute Cloud ("EC2"), which is part of Amazon Web Services. The AMI serves as the basic unit of deployment for services delivered using EC2.

A public AMI for Routing Active Testing is available under Community AMIs. The next chapter tells how to obtain and configure this AMI.

Launching an AWS instance

This chapter tells how to launch an AWS instance on which to run the Routing Active Testing AMI.

Be aware that the AMI is shared to a specific geographical *region* within EC2. Therefore you need to know what region that is and make sure you access the same region.

Logging in to Amazon EC2

- Go to https://aws.amazon.com/ec2.
- Click the button Get started with Amazon EC2.
- Sign in to your AWS account:
- Click Services on the top bar.
- Click Compute, then click EC2. You are taken to the EC2 Management Console.

Choosing an AMI

- In the left-hand pane, under Images, click AMIs.
- In the drop-down which by default reads “Owned by me”, select Public images.
- Enter the search term “juniper-paa”.
- Enter the search term “Owner = 830665132438”. This is the owner account that signed the AMI.
- You will find an AMI whose name begins with “Juniper-PAA-TA-Appliance”. Note the ID of this AMI.
- Check the box next to this AMI, and click the Launch Instance from AMI button.

You are taken to the Launch an instance page.

- Under Name and tags, enter a name for the instance. There is no need to add any tags for the instance.
- Under Application and OS Images, enter “juniper-paa” in the search field. You will find the Test Agent AMI in the Community AMIs category. Verify that this AMI has the same ID as the one found in the previous search.
- Click the Select button next to the Test Agent AMI. You are now taken back to the Launch an instance page.

Choosing an AWS instance type

A large number of AWS instance types will typically appear in this list. Which one to choose depends on the performance needed when running the AMI. We recommend an Amazon EC2 C5 instance for the Test Agent.

Selecting a key pair

Under Key pair (login) you can select a public/private key pair for connecting securely to your AWS instance via SSH. If you have such a private key, select it here. You can also create a new key pair. If you prefer to do without a key pair, select the option Proceed without a key pair.

Configuring network settings

The security group selected here must allow outgoing traffic on ports that the vTA needs in order to communicate with Control Center. Specifically, for SaaS, TCP port 443; for an on-premise installation, TCP port 6000. In addition, UDP port 123 needs to be open to permit NTP time sync.

The security group must also allow traffic on all ports needed for the testing you intend to do with the vTA.

Configuring storage

The recommendation here is to have at least 2 GB of storage. Add a suitable volume.

Configuring user data

It is highly recommended that you enter the cloud-init config for the vTA as user data, as explained below. Alternatively, you can configure this after launching the instance by connecting to the vTA via SSH and navigating the vTA console interface (see the *Troubleshooting* (page 109) section).

- Expand the Advanced details section at the bottom of the page.
- Under User data, provide the cloud-init config for the vTA by pasting it into the box.

The basic cloud-init config is as shown below. Text in angle brackets < > needs to be replaced by the proper strings. Note that lines with parameter settings must be indented as shown. Lines where the default value is kept can be omitted.

```
#cloud-config
paa_test_agent:
  name: <vTA name>
  email: <Routing Active Testing user email address>
  password: <Routing Active Testing password>
  account: <Routing Active Testing account name>
  server: <Routing Active Testing server> (default: login.paa.juniper.net:443)
  management_interface: eth1 (default: eth0)
  management_address_type: dhcp | static (default: dhcp)
```

The following parameters are required only if management_address_type is “static”:

```
management_ip: <management IP address>/<prefix>
management_dns: <DNS server IP address>[,<DNS server IP address>]
management_default_gateway: <gateway IP address>
management_ntp: <NTP server IP address or host name> (default: time.google.com)
```

The following parameters are required only if the vTA is connecting to the server through an HTTP proxy:

```
http_proxy: <proxy server>
http_proxy_port: <proxy port>
http_proxy_auth_type: none | basic (default: none)
```

The following parameters are required only if http_proxy_auth_type is “basic”:

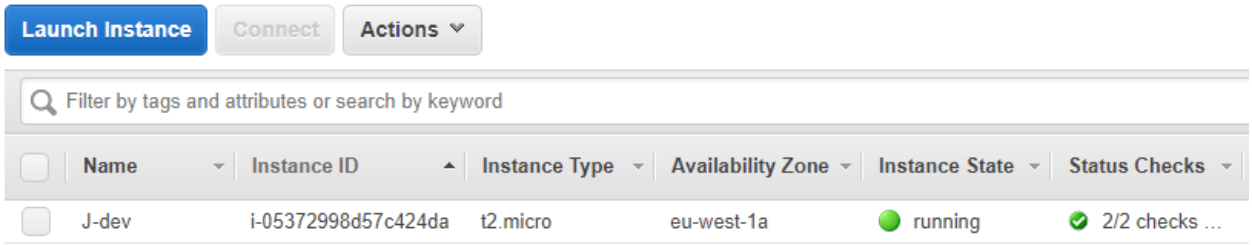
```
http_proxy_username: <proxy authorization user name>
http_proxy_password: <proxy authorization password>
```

The remaining settings can be left as-is.

Reviewing your instance settings and launching your instance

- In the Summary pane on the right, check that all settings for the AWS instance are appropriate. Then click Launch instance.

Your instance should now appear under Instances in the EC2 Management Console. After it has started up, Instance State will be “running”:



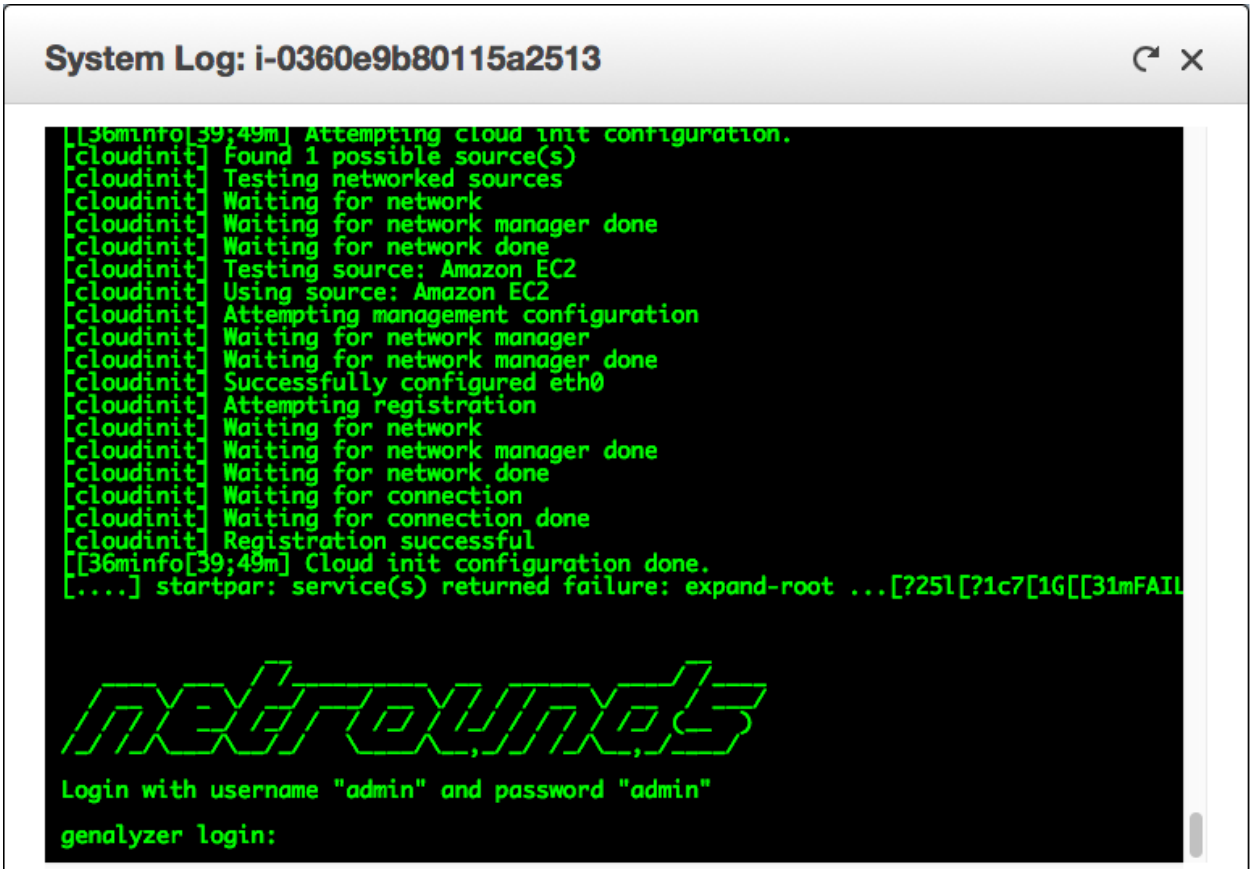
The Test Agent will now automatically register with Control Center and will then appear in the Control Center web GUI under Test Agents. Check for the AWS instance name in that view to verify that the Test Agent has registered.

Verifying successful Test Agent configuration

To verify that the cloud-init configuration of the vTA instance has been successful and that you have access to the Test Agent user interface, proceed as follows:

- Select the Routing Active Testing AMI in the AMI view.
- Click the Actions button and select Instance Settings > Get System Log.

The log should look something like this:



Troubleshooting

If the vTA does not show up in Control Center, it is useful to open its local console to investigate the cause of the problem. This requires that you supplied an SSH public key when creating the AWS instance (see *this section* (page 108)).

- In the Instances view, inspect the public IP address of the instance.
- At a command prompt, type:

```
ssh -i <private_key> admin@<instance_public_ip>
```

where `<private_key>` is the name of the file holding your SSH private key and `<instance_public_ip>` is the public IP address of the vTA instance.

You are now taken to the Test Agent admin menu:



The functionality found here is described *here* (page 209). The following functions are particularly helpful:

- Utilities > Ping for checking that the vTA has a working internet connection.
- Utilities > Troubleshoot connection for verifying that the Routing Active Testing management connection is working.
- Utilities > Root shell for leaving the local console and going to the Linux prompt.

4.2.10.2 Deploying a VHD virtual Test Agent image in Azure

Introduction

This page explains how to deploy a virtual Test Agent from Routing Active Testing as a virtual machine in Microsoft Azure.

Prerequisites

Control Center account

You need an account in a Routing Active Testing Control Center in order to access it: either the one belonging to the Routing Active Testing SaaS solution or one installed on-premise in your organization. If you do not already have a Routing Active Testing account, please contact your Juniper partner or your local Juniper account manager or sales representative.

vTA image

The VNF vTA image is provided either by Juniper's partners or directly from Juniper.

The vTA image for Azure is provided in VHD format.

Once you have your vTA image, you need to upload it to your Azure environment and deploy it. This can be done either through the Azure web GUI or from the Azure CLI. Both procedures are described in this document:

- *Uploading and deploying a vTA image through the Azure web GUI* (page 111)
- *Uploading and deploying a vTA image through the Azure CLI* (page 114)

Uploading and deploying a vTA image through the Azure web GUI

Signing in to Azure

- Go to https://azure.microsoft.com. You will be redirected to a URL associated with your location. The description that follows deals with the English-language version of the GUI.
- Sign in to your Azure account.
- You should find a heading Azure services, under which there is an item Storage accounts.

Creating a storage account

- Click the Storage accounts icon. This opens a view listing whatever storage accounts you have already defined.
- Click + Create to create a new storage account.
- Make the appropriate choice under Resource group (we are assuming here that some resource group has already been created).
- Under Storage account name, enter a name for the storage account.
- Under Region, select your geographical region.

The remaining settings can be left as-is.

- Click the Review + create button at the bottom.
- After validation has passed, click Create.

Your storage account is now created. Click the Go to resource button to bring up an overview page for the account.

Creating a storage container (blob)

- On the left-side bar of the account overview page, scroll down to Data management > Blob inventory.
- Click the button Add your first inventory rule.
- Under Rule name, enter a name for the inventory rule.
- Under Container, click the Create new link. In the box that appears, name the container, then click OK.
- Under Object type to inventory, select Blob.
- Under Blob types, check Page blobs.
- For the other settings, the defaults can be kept here as well.
- Then click Save. Your container is now created.

Uploading the Test Agent VHD file to the storage container

The next step is to upload your Test Agent VHD file to the storage container you just created.

- On the left-side bar of the account overview page, under Data storage, click Containers.
- Click the storage container in the list of containers.
- Click Upload.
- Under Files, select your Test Agent VHD file.
- Expand Advanced.

Important: Under Blob type, select Page blob.

- Keep the defaults for the remaining settings.
- Click Upload.

The upload will take some time as the Test Agent VHD file is approximately 2 GiB in size.

Creating an image

- In the search field at the top, type “images”. If an Images icon turns up, click it and you will be taken to the Images view. Alternatively, this view may appear right away.
- Click + Create.
- Make the appropriate choice under Resource group.
- Under Name, enter a name for the image.
- Under Region, select your geographical region.
- Under OS type, select Linux.
- Under Storage blob, browse to select the VHD file you uploaded in *this section* (page 112). Browse all the way to the VHD file, click the file, then click Select.
- Leave the remaining settings unchanged.
- Click Review + create.
- After validation has passed, click Create.

A message “Your deployment is complete” will appear when the image has been created.

Creating a virtual machine

In this section we will create a virtual machine (VM) in which to run the vTA.

- In the search field at the top, type “images”.
- Click the Images item that turns up.
- In the Images view, click the image you just created.
- Click + Create VM at the top.
- Under Project details, make the appropriate selections.

- Under Instance details, do the following:
  - Enter a name for the virtual machine.
  - Under Size, select “Standard_D2_v5”, or whatever is suitable for your use case.
  - Add att least 8 GB data disk.
- Under Administrator account, you need to provide an SSH key pair to be able to log in to the Test Agent admin menu later on:
  - Set Authentication type to “SSH public key”.
  - Enter an arbitrary string under Username. This setting cannot be left undefined, but it is not used when logging in to the Test Agent.
  - Under SSH public key, paste your SSH public key.
- Under Inbound port rules, make the following settings:
  - Set Public inbound ports to “Allow selected ports”.
  - Under Select inbound ports, select “SSH (22)”.
- Under Licensing type, select “Other”.
- Keep the defaults for all other settings.
- Click Review + create, then Create.

A message “Your deployment is complete” will appear when the virtual machine has been created.

Logging in to the Test Agent

Here is how to log in to the Test Agent via SSH. This is needed in order to register the Test Agent with the Routing Active Testing system, and it is also useful for troubleshooting:

- In the search field at the top, type “virtual”.
- Click the Virtual machines item that turns up. The Virtual machines view is displayed.
- Select the virtual machine created for the Test Agent.
- Note down the Public IP address of the virtual machine.
- At a command prompt, type:

```
ssh -i <id_rsa> admin@<vm_public_ip>
```

where `id_rsa` is the name of the file holding your SSH private key and `vm_public_ip` is the virtual machine’s public IP address.

You are now taken to the Test Agent admin menu:



Here you can register the Test Agent with the Routing Active Testing system as described *here* (page 205). Upon registration, the Test Agent will be visible in Control Center.

Again, please note that initialization of the Test Agent with user data using cloud-init cannot be done through the web GUI. The Azure CLI must be used for this purpose; see the section *Creating a virtual machine* (page 112) above. ** bort

The other functionality found in the console is described *here* (page 209). The following functions are particularly helpful:

- Utilities > Ping for checking that the vTA has a working internet connection.
- Utilities > Troubleshoot connection for verifying that the Routing Active Testing management connection is working.

Uploading and deploying a vTA image through the Azure CLI

A different way to create and manage Azure resources is through the Azure CLI. In this section we indicate how to use the CLI to perform the operations done via the web GUI *here* (page 111).

Full documentation of the Azure CLI is found here: https://docs.microsoft.com/en-us/cli/azure

Creating a storage account

Below we show:

- how to create a resource group (this is assumed to exist in the web GUI in the *section dealing with that GUI* (page 111))
- how to create a storage account within the resource group
- how to create access keys. Access keys are used to authenticate applications when they make requests to the Azure storage account. They are needed for some of the operations that follow.

```
# Create resource group
az group create --location northeurope --name paa

# Create storage account
az storage account create --name paastorage --resource-group paa --location␣
↪northeurope
export AZURE_STORAGE_ACCOUNT=paastorage
```

(continues on next page)

(continued from previous page)

```
# Get access key
az storage account keys list --resource-group paa --account-name paastorage -o table
export AZURE_STORAGE_KEY=<one of the keys from the above command>
```

Creating a storage container (blob)

```
# Create storage container
az storage container create --name paacontainer --account-name paastorage --account-
↪key AZURE_STORAGE_KEY
```

Uploading the Test Agent VHD file to the storage container

The VHD file you have downloaded from Control Center is named `paa-test-agent_<version number>.vhd`. This is provided as the `--file` attribute. The `--name` attribute specifies what the VHD file is to be called in Azure.

```
# Upload VHD
az storage blob upload --container-name paacontainer --file paa-test-agent_<version␣
↪number>.vhd --name test-agent.vhd --type page --account-name paastorage --account-
↪key AZURE_STORAGE_KEY
```

Creating a virtual machine

When creating a virtual machine for running the vTA, you need to use the `--admin-username` option to specify an admin user and the `--ssh-key-value` option to supply your public SSH key in a file (assumed to be named `id_rsa.pub` below).

```
# Create VM
az vm create --resource-group paastuff --name paavta --os-type Linux --image https://
↪paastorage.blob.core.windows.net/paacontainer/test-agent.vhd --use-unmanaged-disk --
↪storage-account paastorage --boot-diagnostics-storage paastorage --custom-data user-
↪data.yaml --admin-username <username> --ssh-key-value id_rsa.pub
```

The attribute `--custom-data` is used to initialize the Test Agent with a Routing Active Testing cloud-init config in a YAML file (`userdata.yaml`). Note that this cannot be done through the web GUI. The YAML file has the following format:

```
#cloud-config
paa_test_agent:
  name: MyTAA
  email: myuser@email.com
  password: mypassword
  account: myaccount
```

An additional line `server:` can be included in the YAML file to specify a server different from the Routing Active Testing SaaS server (which is the default).

Provided that correct credentials are given here, the vTA will register automatically with the Routing Active Testing system and appear in the list of Test Agents in the Control Center GUI.

4.2.10.3 Deploying a virtual Test Agent image in Google Cloud

Introduction

This page explains how to deploy a Virtual Test Agent from Routing Active Testing as a virtual machine in Google Cloud.

Prerequisites

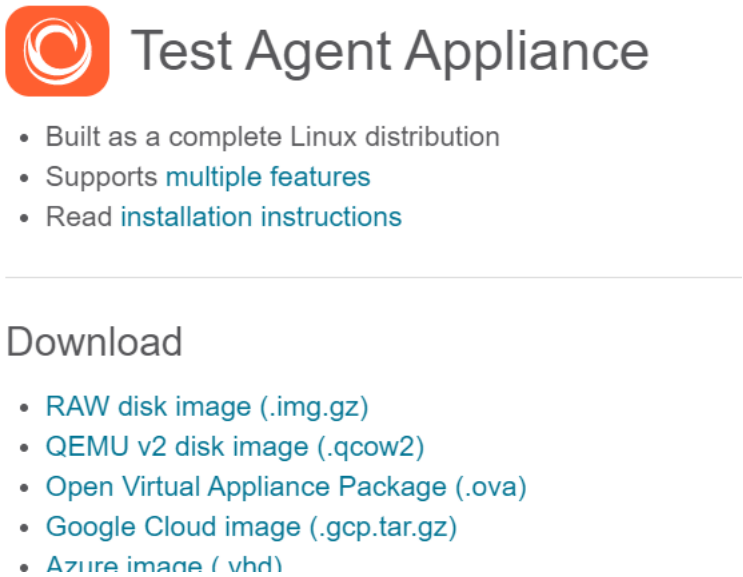
Control Center account

You need an account in a Routing Active Testing Control Center in order to access it: either the one belonging to the Routing Active Testing SaaS solution or one installed on-premise in your organization. If you do not already have a Routing Active Testing account, please contact your Juniper partner or your local Juniper account manager or sales representative.

vTA image

The vTA image is provided either by one of Juniper's partners or directly by Juniper.

The vTA image for Google Cloud is available for download in Control Center. It has the file extension .gcp.tar.gz:



Once you have your vTA image, you need to upload it to your Google Cloud environment and deploy it.

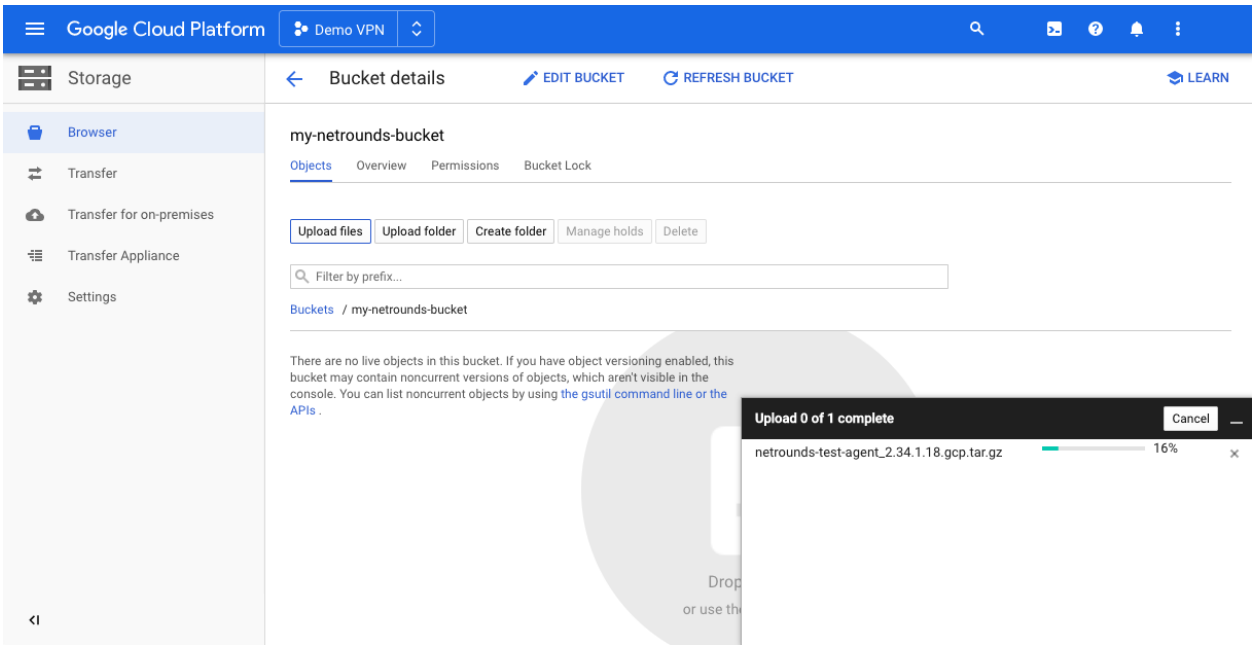
Uploading and deploying a vTA image through the Google Cloud web GUI

Signing in to Google Cloud

- Go to https://cloud.google.com.
- Sign in to your Google account if you are not signed in already.

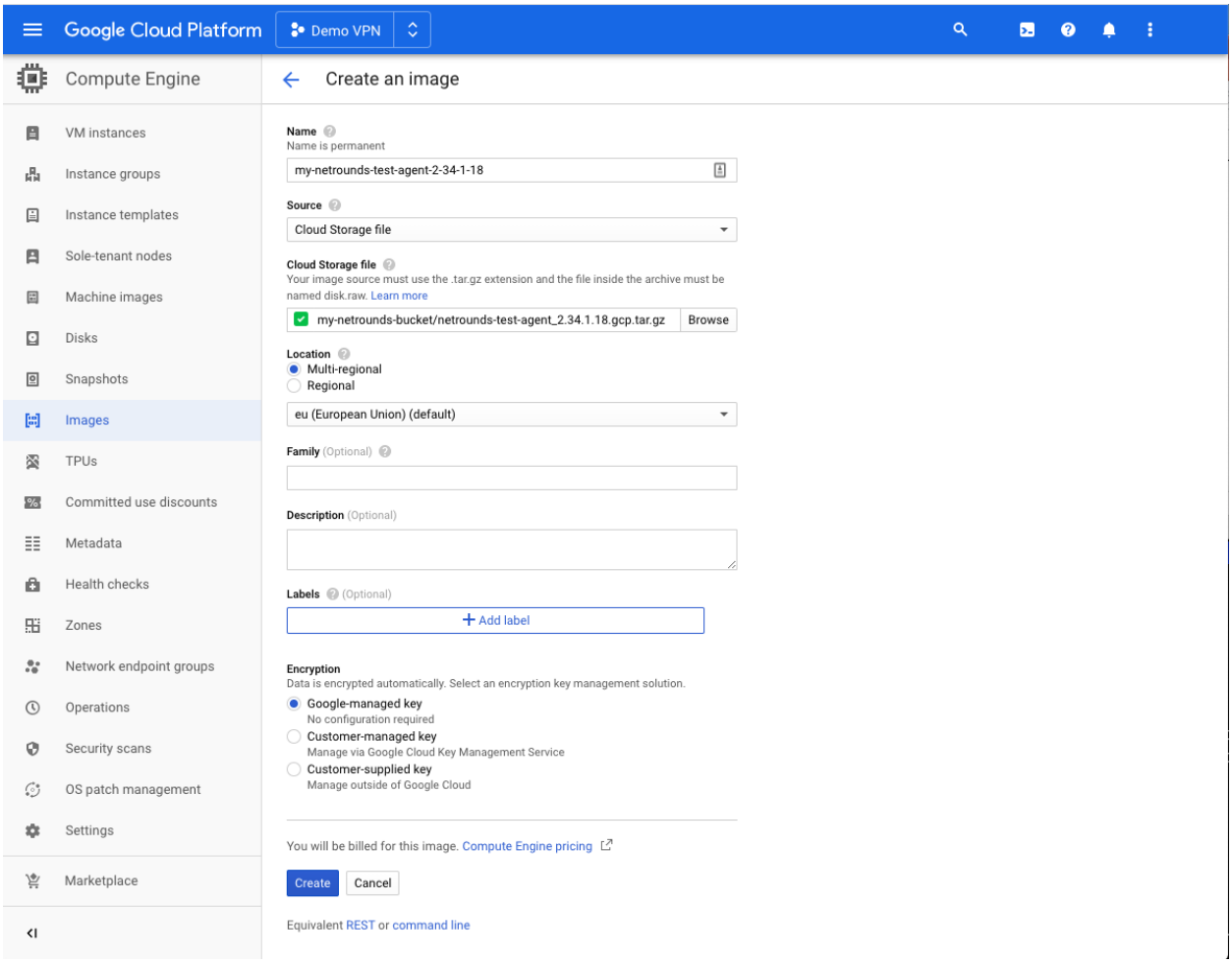
Uploading a Test Agent image to Google Cloud

- In your GCP console, create a storage bucket to host the Test Agent image file, and then click Upload files to upload the image to this bucket.



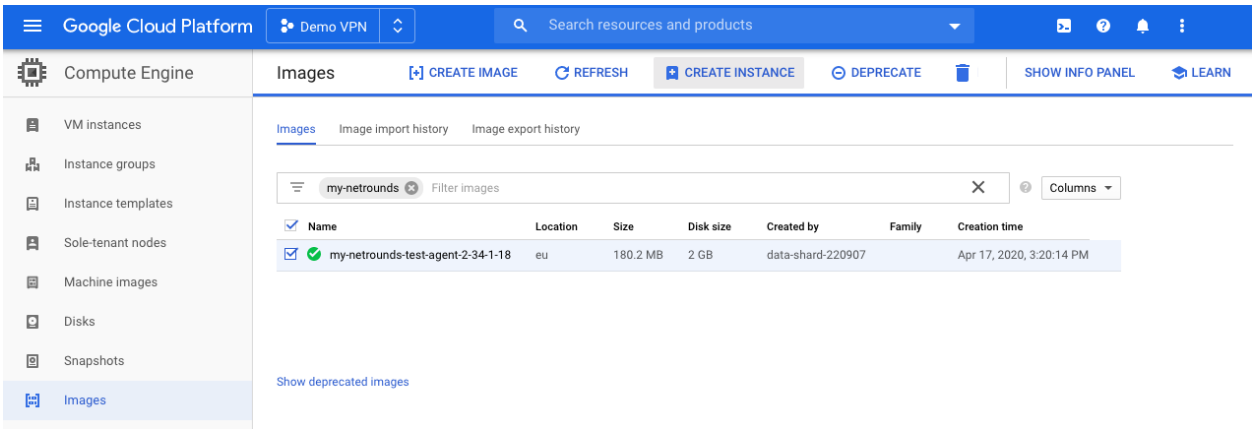
Registering a compute image

Register a new compute image by referring to the file in the cloud storage bucket (Cloud Storage File in the screenshot below). It will take a minute or two to create the image.

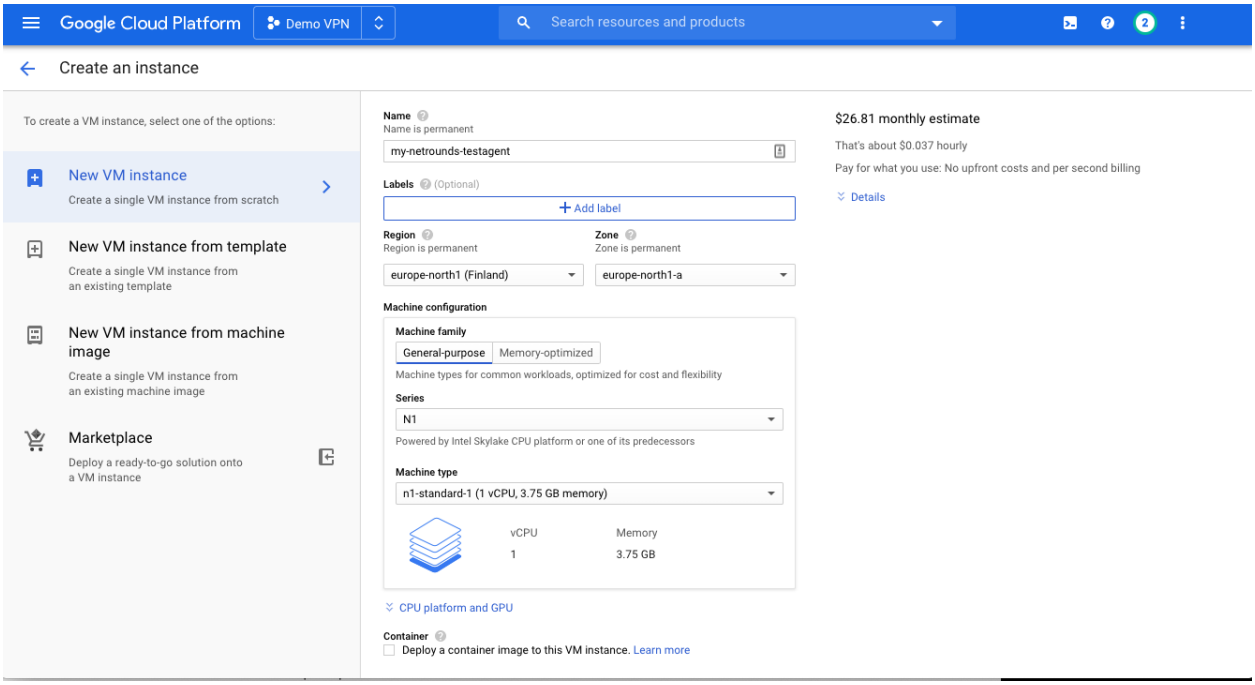


Creating a VM instance

We can now launch compute instances based on this image:

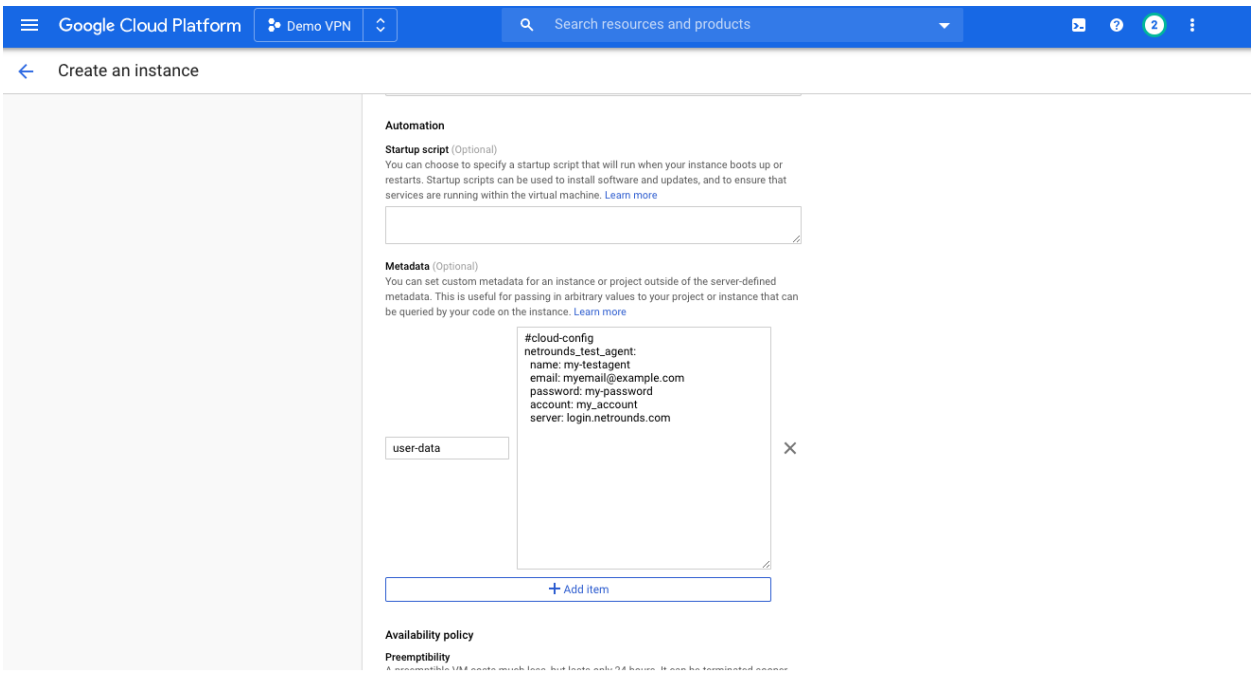


The default Machine type is set to “n1-standard-1”, which works well with the Test Agent, but both smaller and larger instances are supported as well. All machine types available today should work fine with the Test Agent.

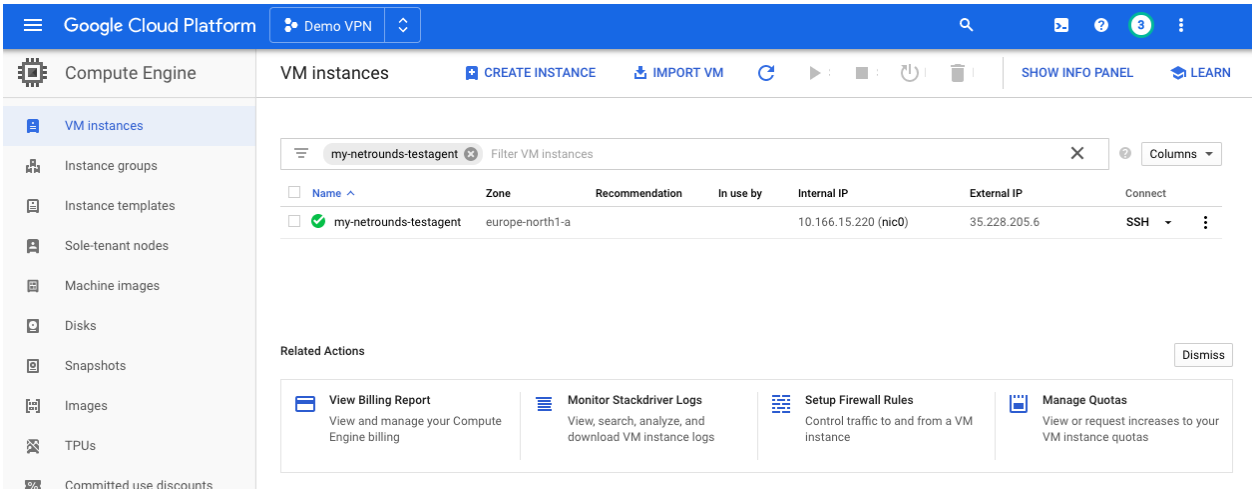


Under advanced settings it is possible to specify Metadata. This may be used to provide cloud-config to register the Test Agent with Control Center.

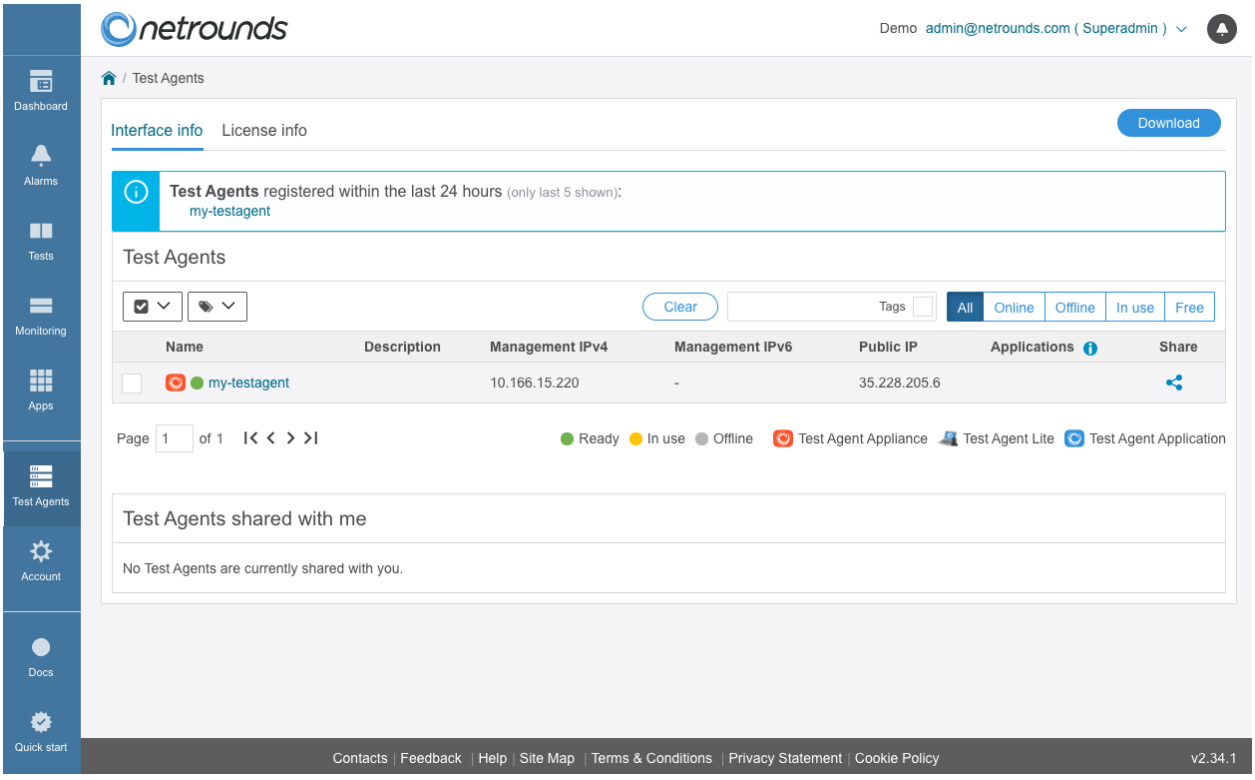
The user data should be provided as value to the key “user-data”:



Once created and started, the instance will appear in the VM instances view:

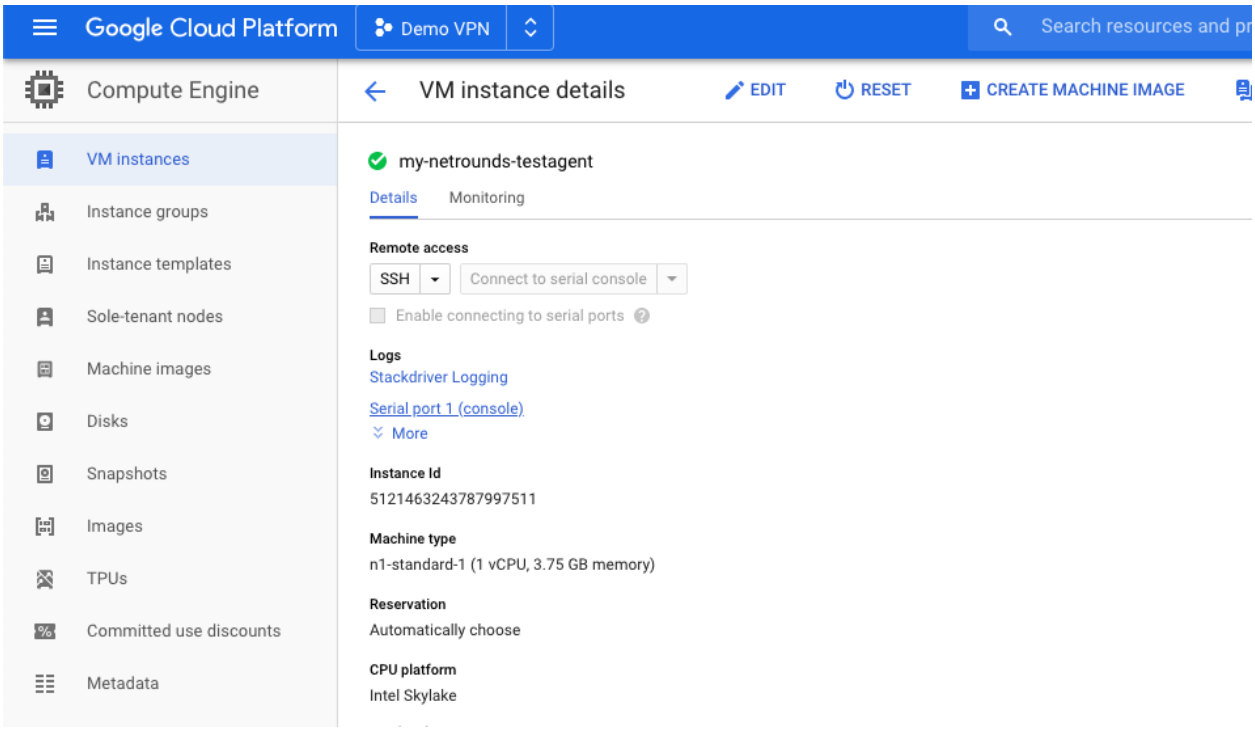


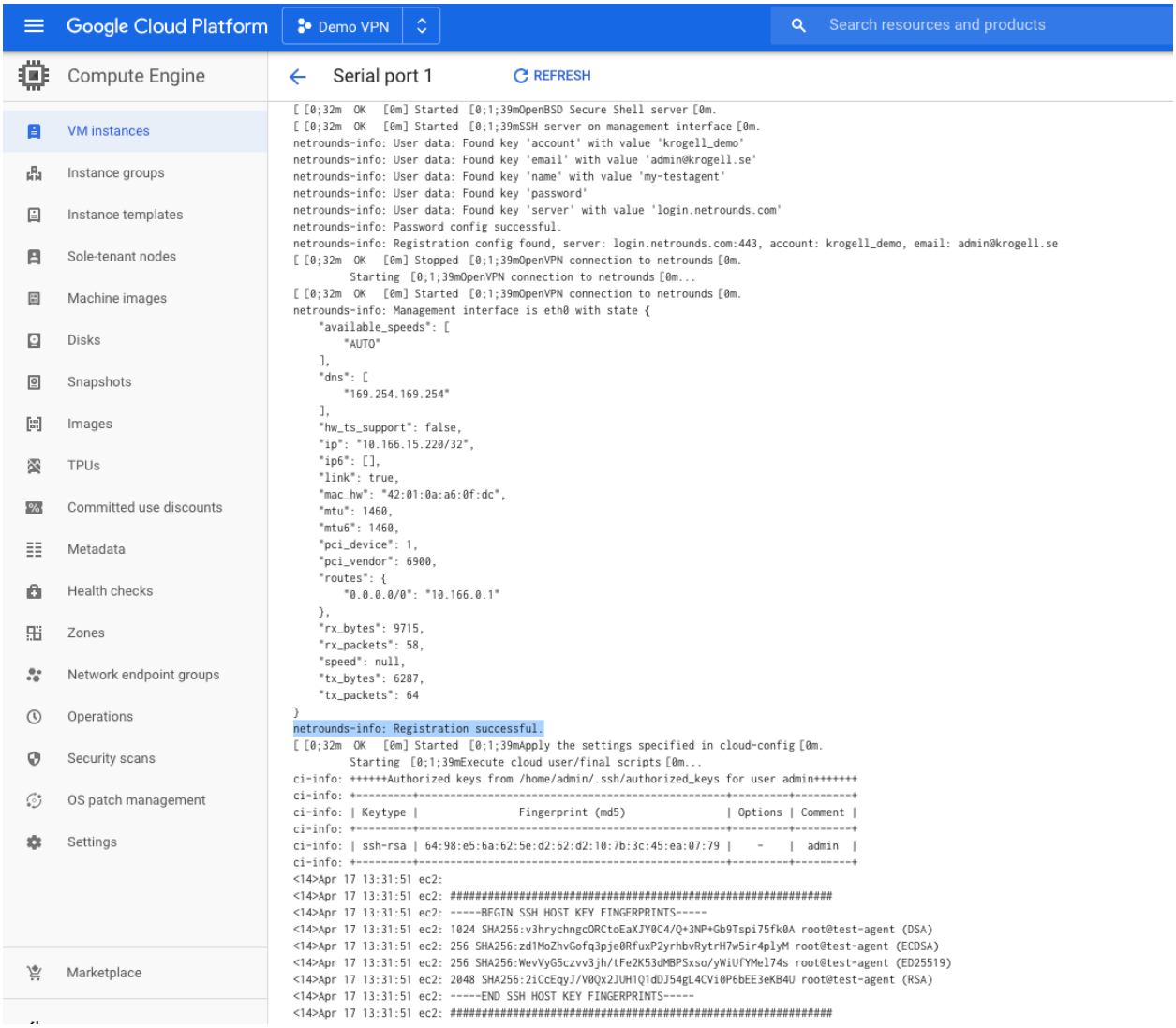
It should also appear in Control Center as a new Test Agent:



Troubleshooting

The instance serial log has details about boot-up and registration status:





Using the interactive serial console

As an alternative to using cloud-config to register the Test Agent with Control Center, it is possible to enable the interactive serial console (as explained here: https://cloud.google.com/compute/docs/instances/interacting-with-serial-console) and then use the regular serial CLI to manage the Test Agent directly in your web browser.

4.2.10.4 Deploying a virtual Test Agent in OpenStack

Introduction

This page explains how to deploy virtual Test Agents (vTAs) from Routing Active Testing in OpenStack and how to control these from Routing Active Testing Control Center.

In ETSI NFV terminology, these vTAs are likewise referred to as Virtual Test Agents, and the centralized controller is referred to as the combination of Test Controller (TC) and Test Result Analysis Module (TRAM).

This guide assumes that you have a basic knowledge of OpenStack and Network Function Virtualization (NFV), and that you have your own OpenStack environment in which to launch the vTA images.

Prerequisites

Control Center account

You need an account in a Routing Active Testing Control Center in order to access it: either the one belonging to the Routing Active Testing SaaS solution or one installed on-premise in your organization. If you do not already have a Routing Active Testing account, please contact your Juniper partner or your local Juniper account manager or sales representative.

vTA image

The VNF vTA image is provided either by Juniper's partners or directly from Juniper.

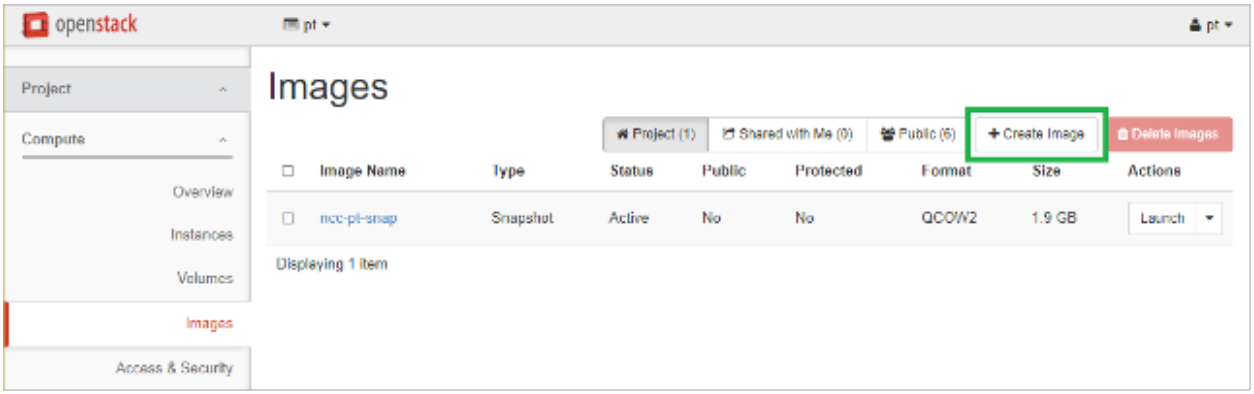
The vTA image for OpenStack is provided in either raw or QCOW2 format.

Launching the vTA image in OpenStack

Uploading the vTA image to OpenStack

First you need to upload the vTA image to your NFV/OpenStack platform. In OpenStack's Horizon GUI, do the following:

- On the navigation bar on the left, expand Project, and under Compute, click Images.
- Click the Create Image button. Fill out the dialog that appears, and create the image.



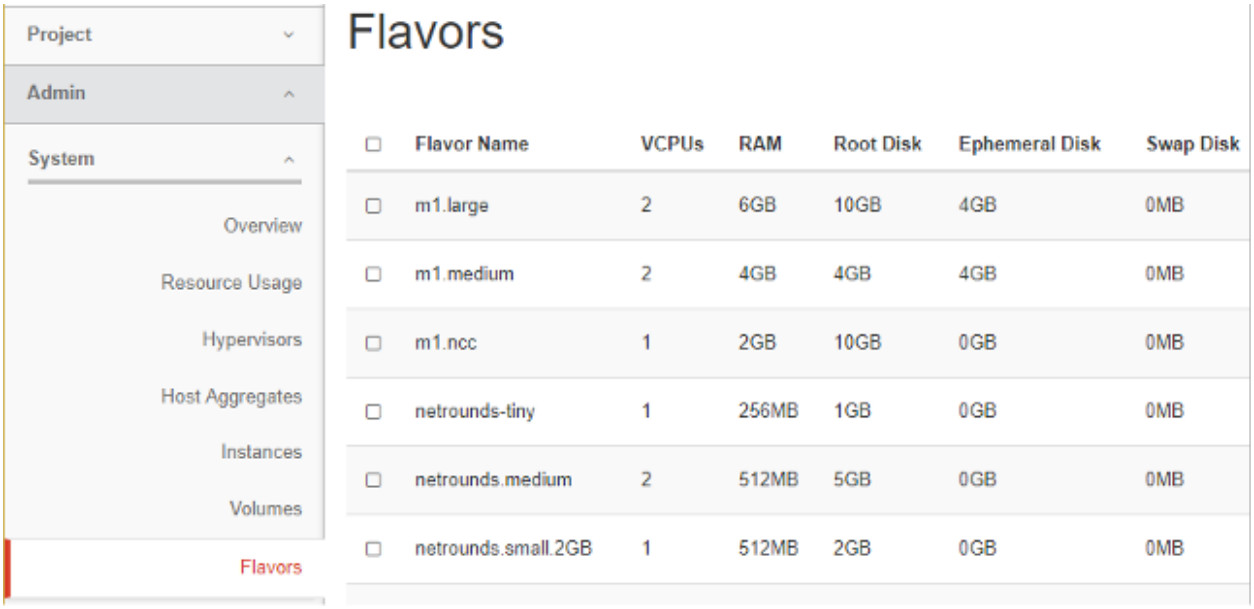
Creating a flavor for the vTA image

Flavors in OpenStack are virtual hardware templates, defining RAM allocation, block storage, and number of CPU cores. In the Horizon GUI, flavors are shown and created under Admin > System > Flavors.

The minimum recommended flavor for a vTA is:

- 1 vCPU
- 512 MB RAM
- 2 GB block storage

See below for some examples of flavors, including one called “netrounds.small.2GB” which corresponds to the above specifications.



Launching an instance of the vTA image

The final step is to launch an instance of the vTA image in OpenStack. Things will look a bit different depending on whether you are using a HEAT Orchestration Template (HOT), which is essentially a virtual machine descriptor, or whether you are specifying the details when launching the vTA. See below for instructions on how to launch the vTA:

- manually – see *this section* (page 124)
- using HOT automation – see *this section* (page 126)
- using the Python OpenStack API – see *this section* (page 128).

Once you have launched the vTA image, the vTA will register with Control Center and appear in its web GUI under Test Agents. Check for the vTA name in that view to verify that the vTA has registered. You can then initiate tests and monitoring sessions on the vTA from Control Center.

Launching a vTA image manually in Horizon

Here is how to deploy a vTA manually, without using a HEAT Orchestration Template.

- On the navigation bar, select Project > Compute > Instances. Then click Launch instance.
- Under Details, enter a name for the vTA instance.
- Under Source, select the image to boot the vTA instance from. In general, the default volume settings can be kept.
- Under Flavor, select a suitable flavor for the vTA. Please refer back to *this section* (page 123).
- Under Networks, select whatever is most appropriate in your case.
- The Network Ports section can be skipped.
- Under Security Groups, the group selected must satisfy the following:

  - The vTA must be able to establish an outgoing session towards its Control Center: in the cloud server case, to https://login.paa.juniper.net using TCP port 443; in the on-premise server case, to the host IP using TCP port 6000 (default).
  - UDP port 123 must be open to permit NTP time sync.
  - Traffic must be allowed on all ports needed for the testing you intend to do with the vTA.
- The Key Pair section can be skipped, as SSH is not in use for vTAs.
- Under Configuration, in the Customization Script box, you need to enter "cloud-config" metadata specifying among other things how to connect to Control Center. The format of this metadata is given *below* (page 125).
- The Server Groups, Scheduler Hints, and Metadata sections can be skipped.
- Finish by clicking the Launch Instance button at the bottom of the dialog.

A vTA image can also be deployed from the command line, but that method is not explained further in this introductory guide.

Format of "cloud-config" metadata

The vTA supports only #cloud-config metadata. Its format is as shown below.

Note: The `#cloud-config` and `paa_test_agent` lines must be present, and that all of the remaining lines must be indented.

Basic configuration:

```
#cloud-config
paa_test_agent:
  name: MyvTA                                  # vTA name
  email: john.doe@example.com                  # NCC user email
  password: secret                             # NCC user password
  account: theaccount                          # NCC account
  server: login.paa.juniper.net:443            # Login server
  management_interface: eth0                   # Test Agent management interface
  management_address_type: dhcp                # Can be "dhcp" or "static"
```

The following parameters are required only if `management_address_type` is "static":

```
## Set the following if using "static" above:
# management_ip: 192.168.1.2/24               # Management IP address
# management_dns: 8.8.8.8, 8.8.4.4            # DNS server IP address(es)
# management_default_gateway: 192.168.1.1     # Gateway IP address
# management_ntp: time.google.com             # NTP server IP address or host name
```

The following parameters are required only if the vTA is connecting to the server through an HTTP proxy:

```
## Set the following if using an HTTP proxy:
# http_proxy: myproxy.lan                     # Proxy server
# http_proxy_port: 80                         # Proxy port
# http_proxy_auth_type: none                  # Can be "none" or "basic"
```

The following parameters are required only if http_proxy_auth_type is "basic":

```
# http_proxy_username: johndoe                # Proxy authorization user name
# http_proxy_password: secret                 # Proxy authorization password
```

The following parameters are required if IPv6 is to be used for Test Agent management connections:

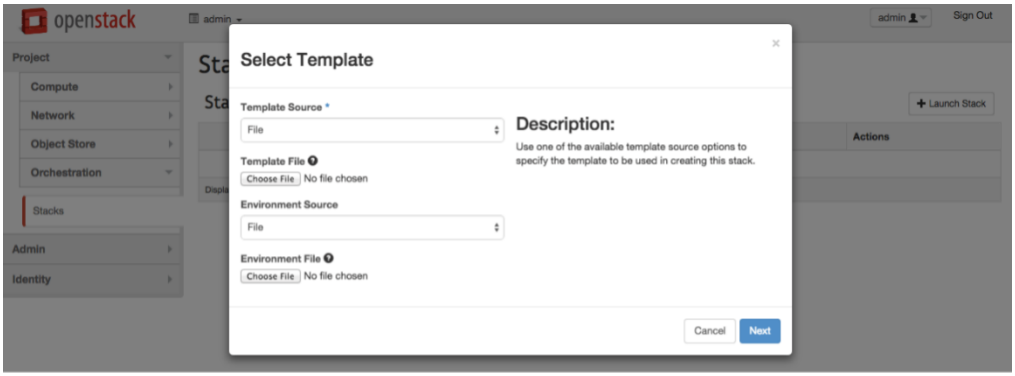
```
## Note: IPv6 management requires special config, see separate documentation
# management_enable_ipv6: False
# management_ntp_allow_ipv6: False
# management_address6_type: none              # Can be "dhcp", "slaac", or "static"

## Set if "static". Note: Use CIDR format for IP
# management_ip6: 2001:db8:85a3::8a2e:370:7334/64
# management_dns6: 2001:4860:4860::8888, 2001:4860:4860::8844
# management_default_gateway6: <gateway IP address>
```

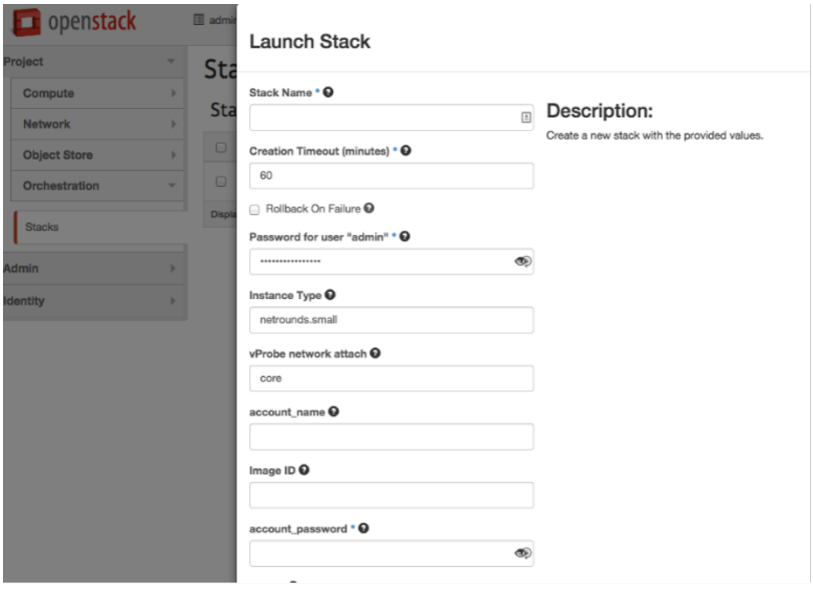
Launching a vTA image using a Heat Orchestration Template in Horizon

To deploy a vTA image in OpenStack using a Heat Orchestration Template (HOT):

- On the navigation bar, select Project > Orchestration > Stacks.
- Click the Launch Stack button on the far right. Select the HOT file to be used. For an example of such a file, see *this section* (page 127).



The template will ask for input to be specified, as shown in the following screenshot:



Some of these fields have default values, as defined in the HOT file, while others are variables in the template.

After a vTA image has been booted up, it must receive its instance-specific userdata/metadata so that it can properly configure the network interfaces and call back home to Control Center. The vTA supports two ways to provide this instance-specific data:

1. Requesting the data from a service running at http://169.254.269.254 on the compute host. This is similar to the approach taken by an EC2 instance running in AWS.
2. Reading the data from a special configuration drive that is attached to the vTA instance when it boots. This is referred to as the "config-drive method".

The format of the instance-specific userdata/metadata is given in *this section* (page 125).

Using either of the above methods, it is possible to automate connecting the vTA to Control Center.

Note: You need to ensure that the vTA can establish an outgoing session towards its Control Center: in the cloud server case, to https://login.paa.juniper.net using TCP port 443; in the on-premise server case, to the host IP using TCP port 6000 (default). Alternatively, the vTA may connect via an HTTP proxy. In addition, UDP port 123 needs to be open to permit NTP time sync.

Example of HOT file

An example of a HOT file is provided below. Replace the [text in square brackets] with your input.

```
heat_template_version: 2015-07-15

description: Heat template to deploy a single vTA in OpenStack
parameters:
        account_name:
                type: string
                description: Routing Active Testing account name to be used for␣
↪autoregistration
        email:
                type: string
                description: User name (email address) for Routing Active Testing␣
↪account
        account_password:
                type: string
                hidden: true
                description: Routing Active Testing account password to be used for␣
↪autoreg
        vTA_name:
                type: string
                description: Name to display in Routing Active Testing inventory
                default: vTA1
        image_id:
                type: string
                label: Image ID
                description: Image to be used for compute instance
                default: [Replace with vTA image name in your OpenStack env]
        instance_type:
                type: string
                label: Instance Type
                description: Type of instance (flavor) to be used
                default: [Replace with flavor that suits vTA]
        vTA_network:
```

(continues on next page)

(continued from previous page)

```
                type: string
                label: vTA network attach
                default: [Replace with network that vTA attaches to]
                description: vTA test interface attached to this network
resources:
        vTA:
                type: OS::Nova::Server
                properties:
                        key_name: Operations
                        image: { get_param: image_id }
                        flavor: { get_param: instance_type }
                        networks:
                        - network: management
                        - network: { get_param: vTA_network }
                        user_data_format: RAW
                        user_data:
                                str_replace:
                                  template: |
                                    email: $email
                                    password: $password
                                    account: $account_name
                                    server: login.paa.juniper.net:443
                                    name: $vTA_name
                                  params:
                                    $email: { get_param: email }
                                    $password: { get_param: account_password }
                                    $account_name: { get_param: account_name }
                                    $vTA_name: { get_param: vTA_name }
```

Launching a vTA image using the Python OpenStack API

It is possible to use the OpenStack Nova and Keystone Python APIs to automatically launch the vTA image from a Python script:

```
def create_instance(net, vta_name, os_password):
#connect using credentials
creds = get_nova_creds(os_password)
nova = nvclient.Client(**creds)

#get image, flavor, network/nics
image = nova.images.find(name="vTA_cloudinit_image")
flavor = nova.flavors.find(name="netrounds.small.2GB")

network = nova.networks.find(label=net)
nics = [{'netid': network.id}]

# Create instances; requires cloudinit on vTA images
instance = nova.servers.create(name=vta_name, image=image,\
flavor=flavor, key_name="default", nics=nics,\
userdata=open("./userdata.txt")
```

The cloud-init file (user-data) in `userdata.txt` above has this content:

```
#cloud-config
paa_test_agent:
```

(continues on next page)

(continued from previous page)

```
email: [email you use when logging in to Routing Active Testing]
password: [Routing Active Testing login password]
account: [Routing Active Testing account]
server: [Control Center address; for SaaS: login.paa.juniper.net:443]
name: [Name of vTA to appear in Control Center inventory]
```

Note again that the `#cloud-config` and `paa_test_agent` lines must be present.

Please contact Juniper support at https://support.juniper.net/support/requesting-support to get example Python code.

4.2.10.5 Deploying a virtual Test Agent in Oracle Cloud

Introduction

This page describes how to deploy a Routing Active Testing Test Agent in Oracle Cloud.

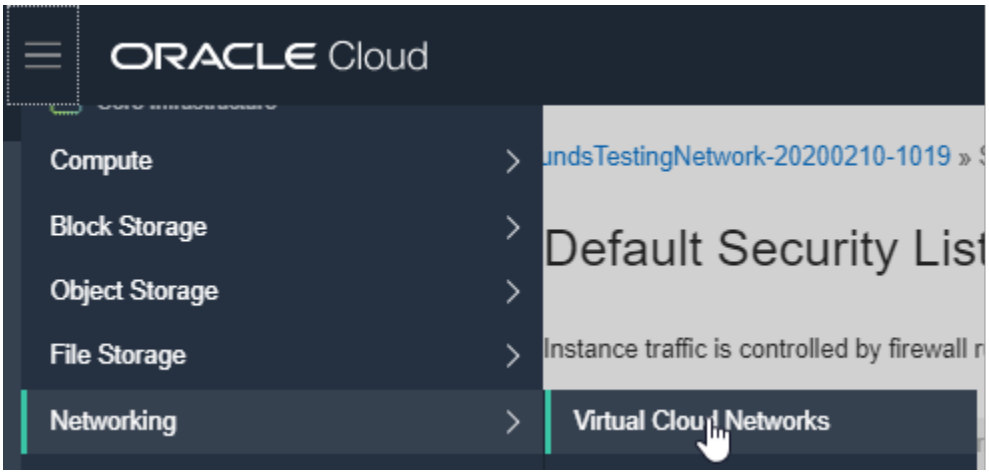
Prerequisites

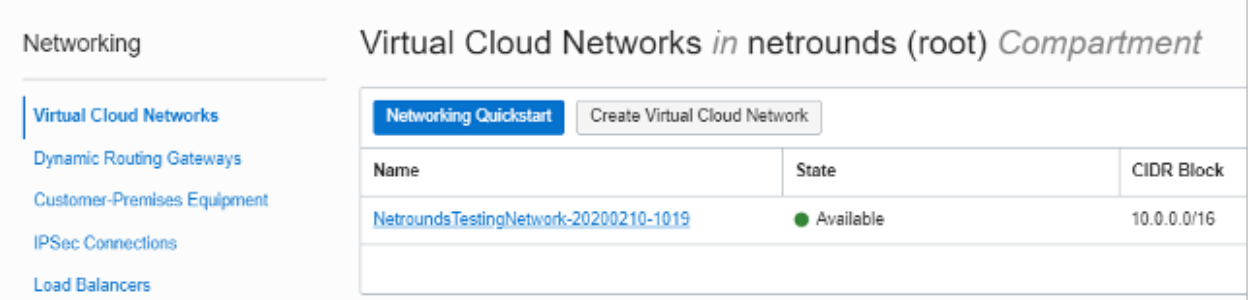
Oracle Cloud uses QEMU as its underlying virtualization platform and thus accepts images in QCOW2 and VMDK format.

A Test Agent image in QCOW2 format is available in Control Center under Test Agents > Download. If you prefer VMDK, you can easily convert the QCOW2 image to that format, or you can extract it from the OVA provided in the same dialog.

Further prerequisites are as follows:

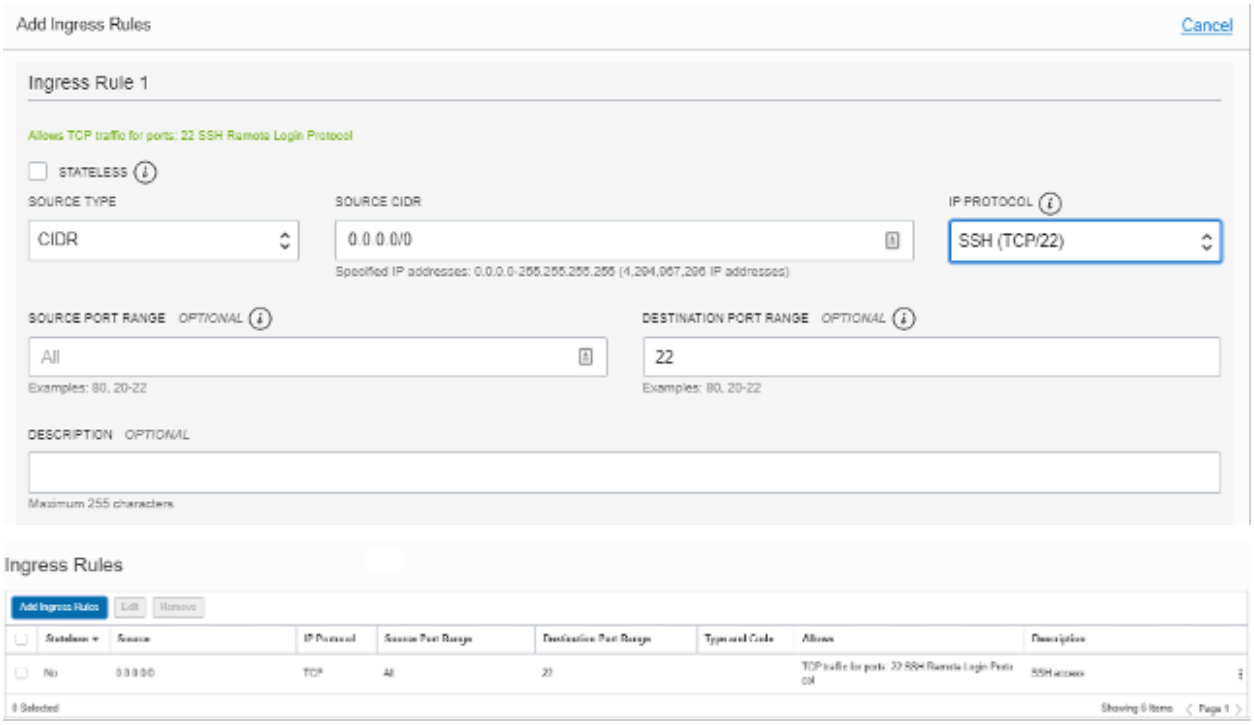
- An account in Oracle Cloud.
- A Test Agent boot disk made available in Oracle Cloud as a custom image.
- A defined virtual cloud network resource in Oracle Cloud. The network needs at least SSH access from the desired source IP address. This configuration can be done under the selected virtual cloud network under Security List: see the following screenshots.







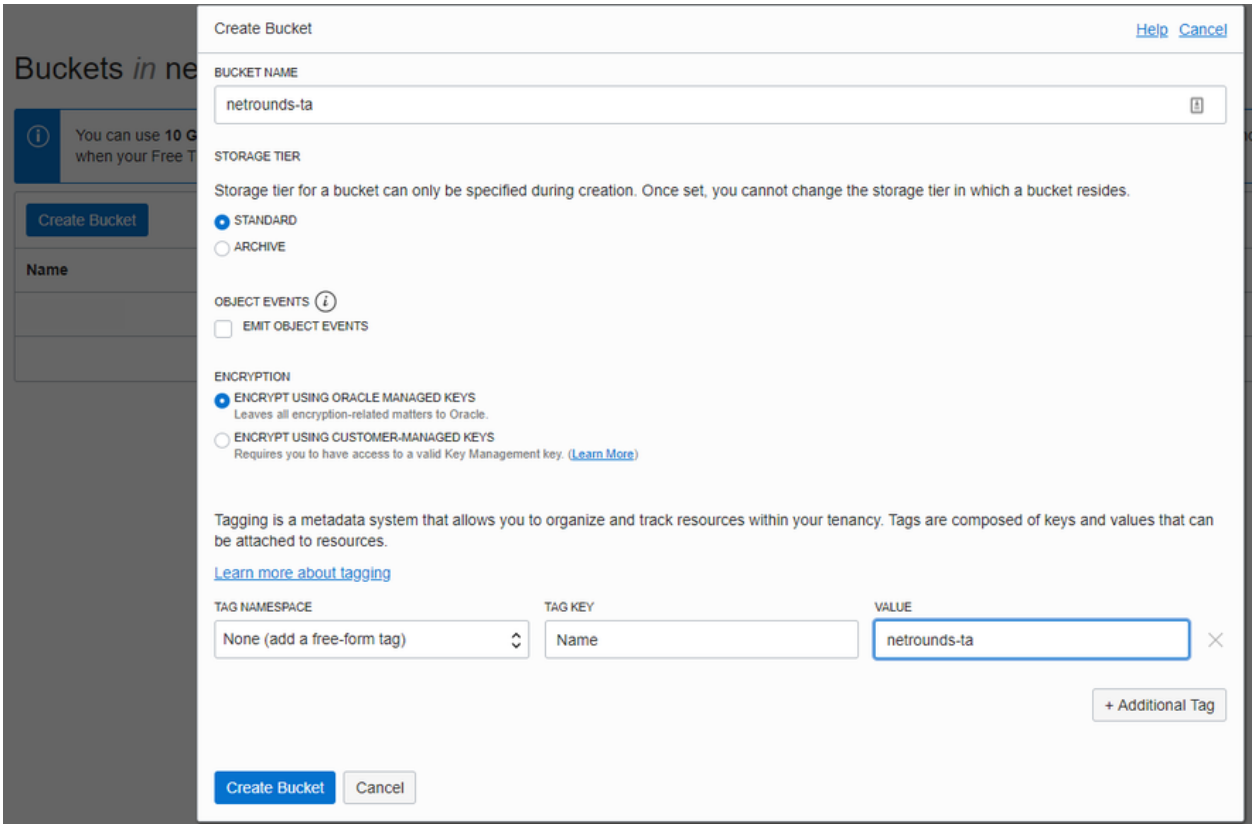
Note: For security reasons, please restrict the range of sources from which SSH access is enabled. Do not accept SSH access from any source address.



Deploying the Test Agent in Oracle Cloud

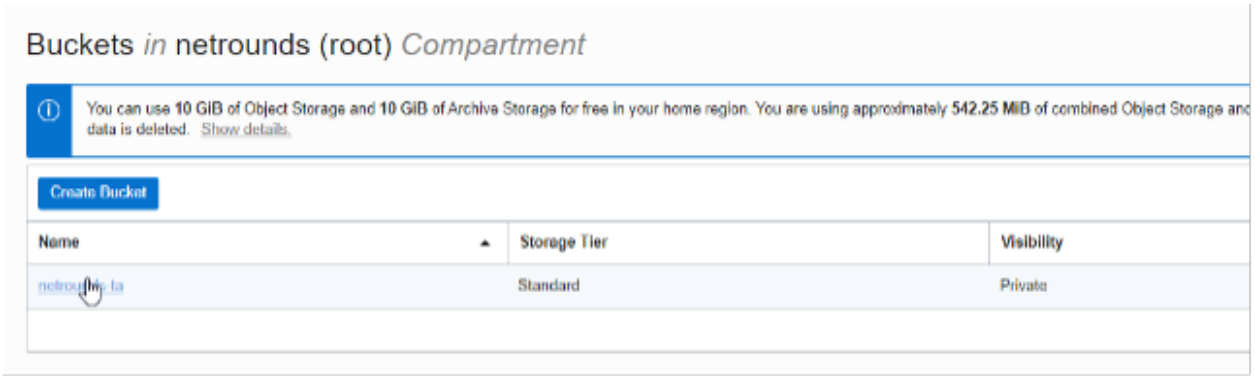
Creating an Oracle bucket

- Go to the Oracle Cloud console.
- Select the compartment “netrounds (root)”.
- Go to Object Storage in that compartment.
- Create a new bucket to upload the Test Agent disk image to. Make settings as shown in the screenshot below, then click Create Bucket.

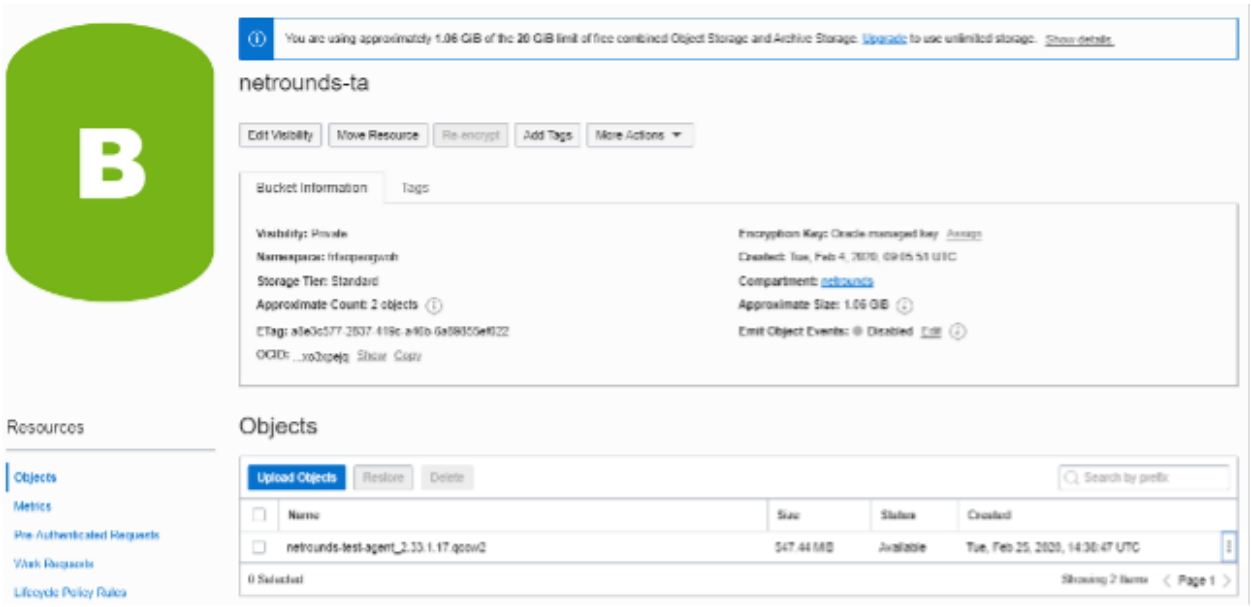


Uploading the image to your bucket

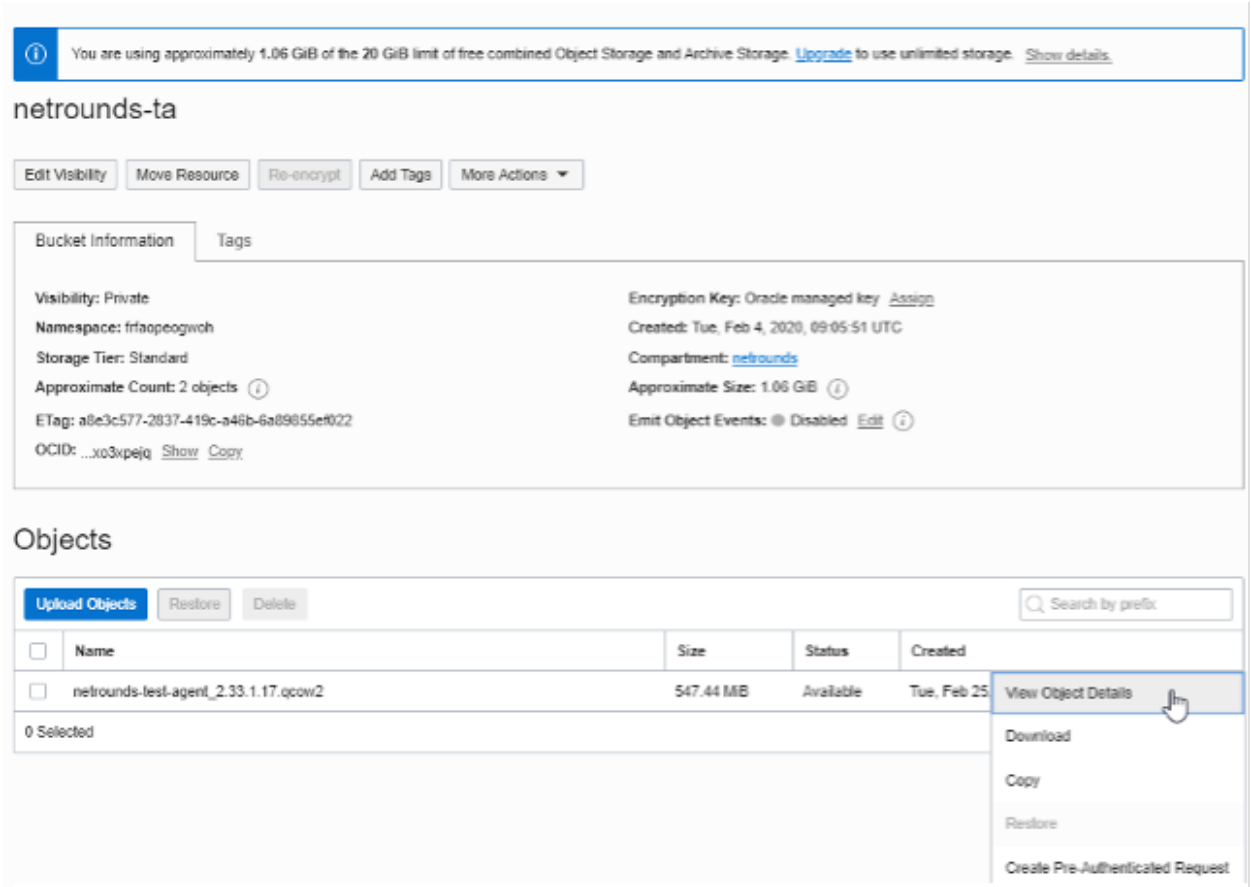
- Click the bucket name in the list of buckets.



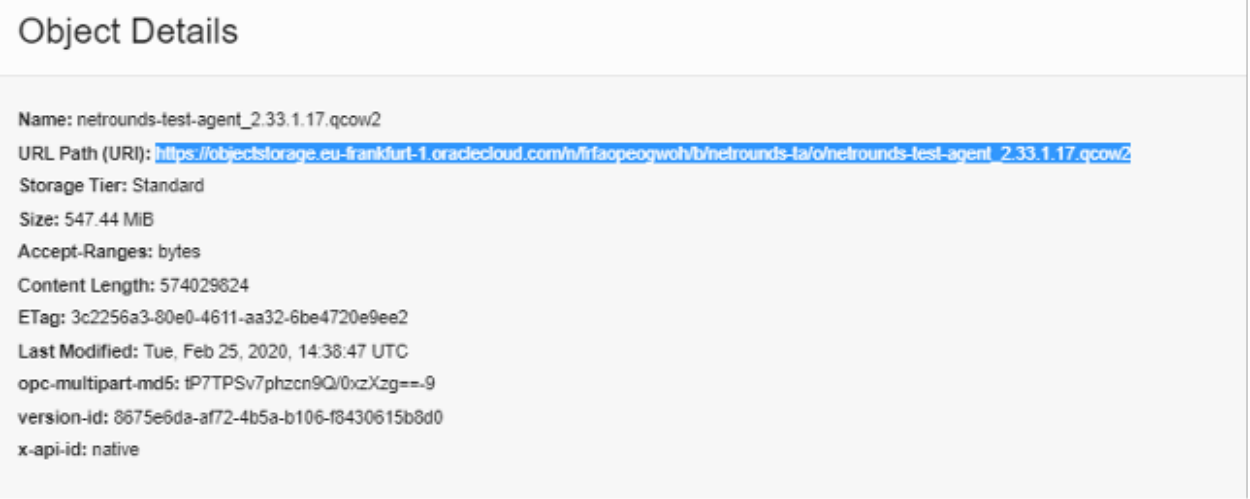
- Upload the QCOW2 image. For more information, see the Oracle Cloud online documentation at https://docs.oracle.com/en-us/iaas/Content/Compute/Tasks/importingcustomimagelinux.htm. In the example below, the QCOW2 image has been used. The VMDK format can be easily obtained either by converting the QCOW2 image or by uncompressing the OVA offered in Routing Active Testing Control Center.



- Click the three-dots button for the image (on the far right).
- Select View Object Details to get the *object storage ID (URL path)*.



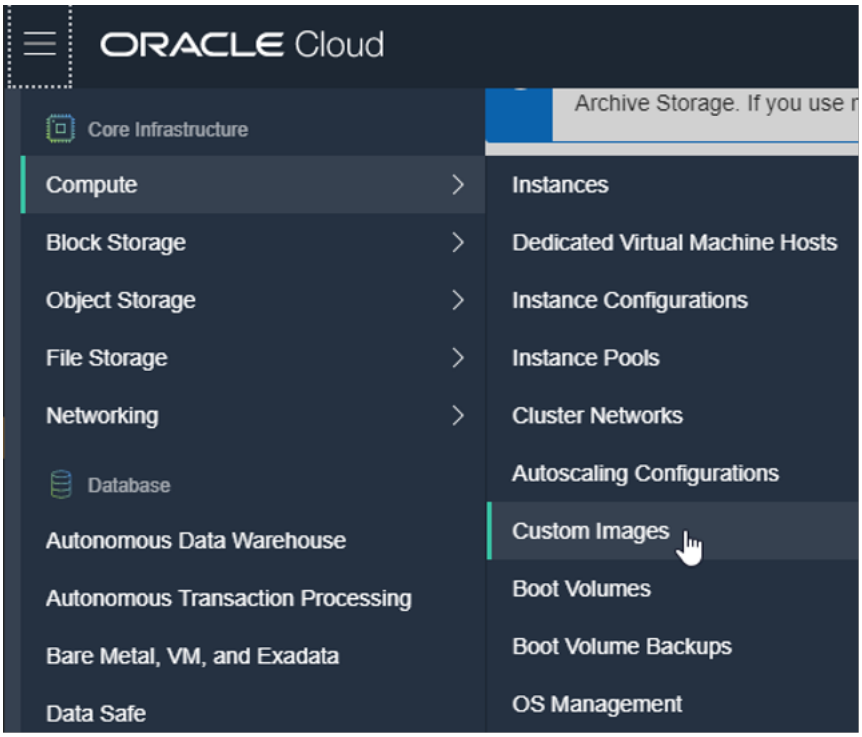
- Copy URL Path (URI). It will be needed in the next section *below* (page 134).



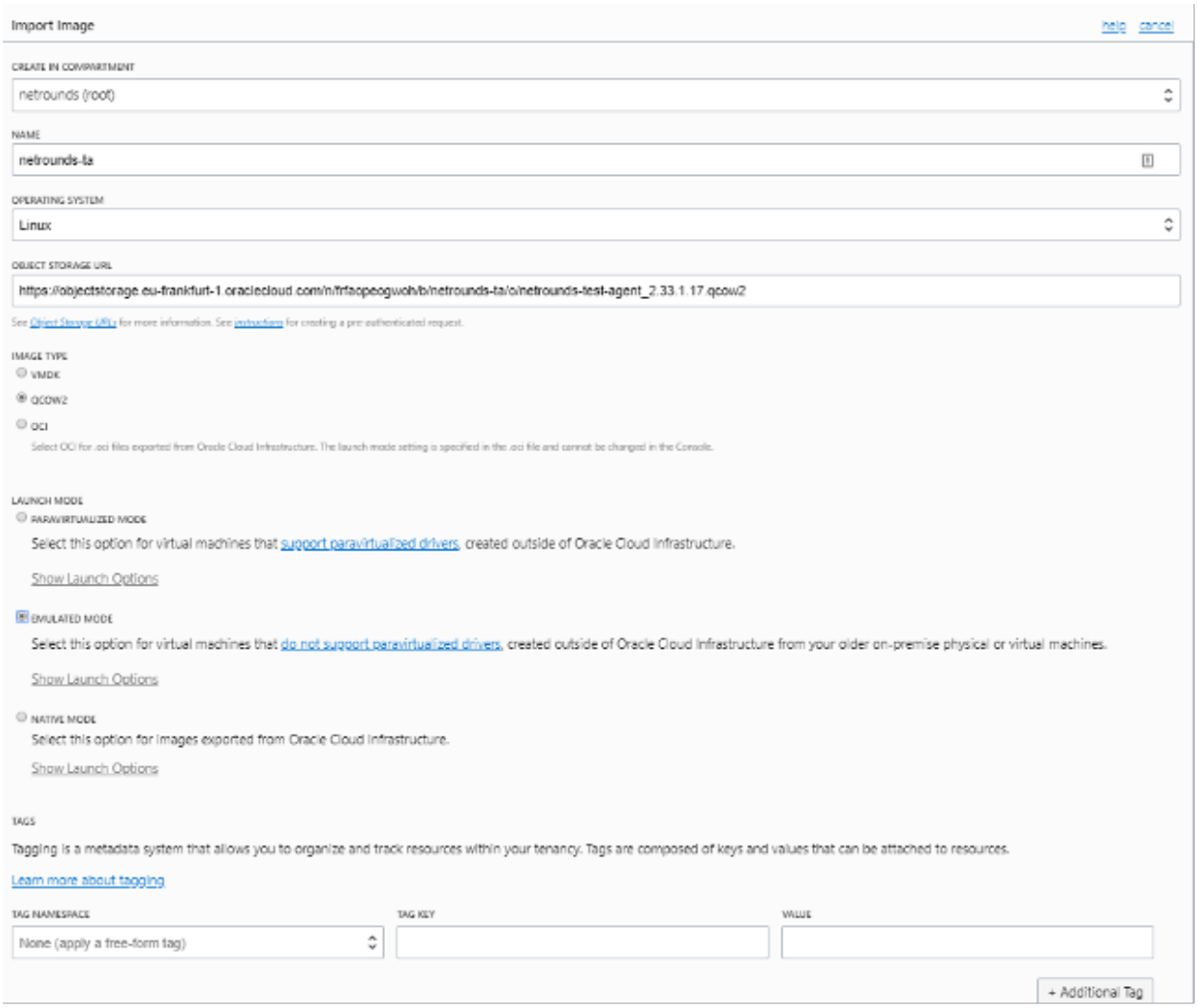
Creating a Test Agent custom image

This process is needed to be able to boot virtual Test Agents from the Routing Active Testing image uploaded in the previous section.

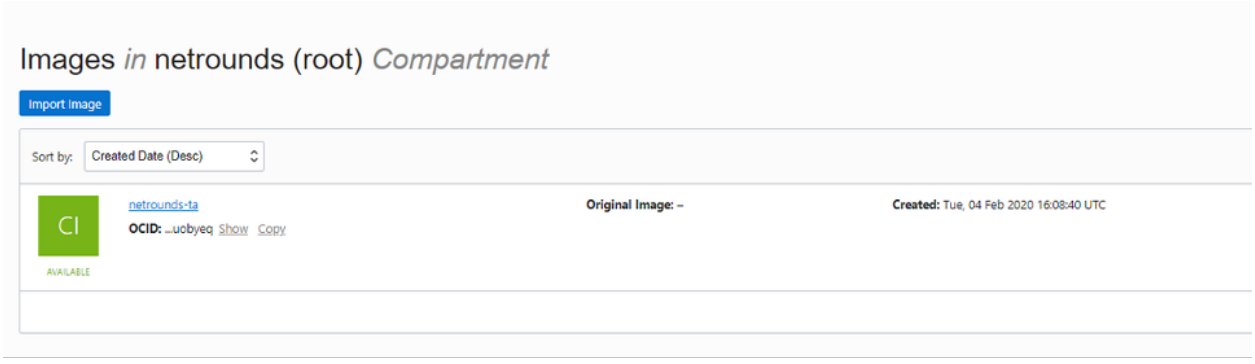
- Go to the Oracle Cloud menu and select Custom Images.



Now it is time to import the disk image you have uploaded as a bootable disk. Note that the bootable disk will be available in the compartment you have uploaded to. Here we will use the object storage ID to point to the disk.



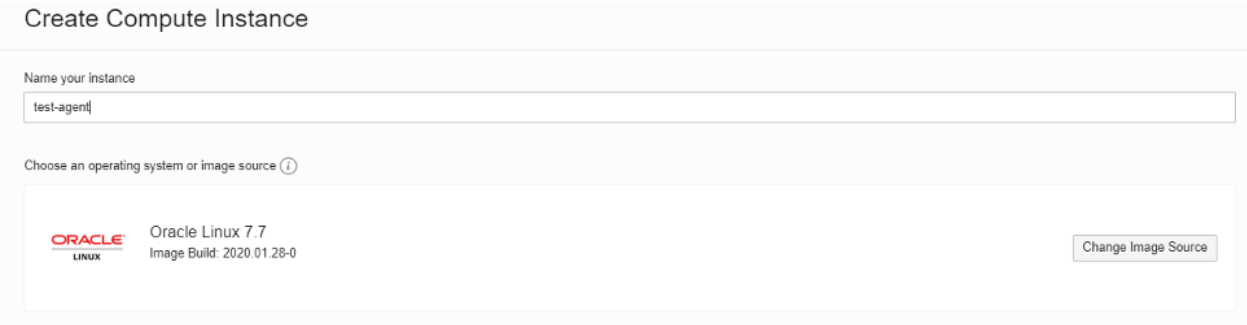
Once you have created the image, it will be displayed in your inventory:



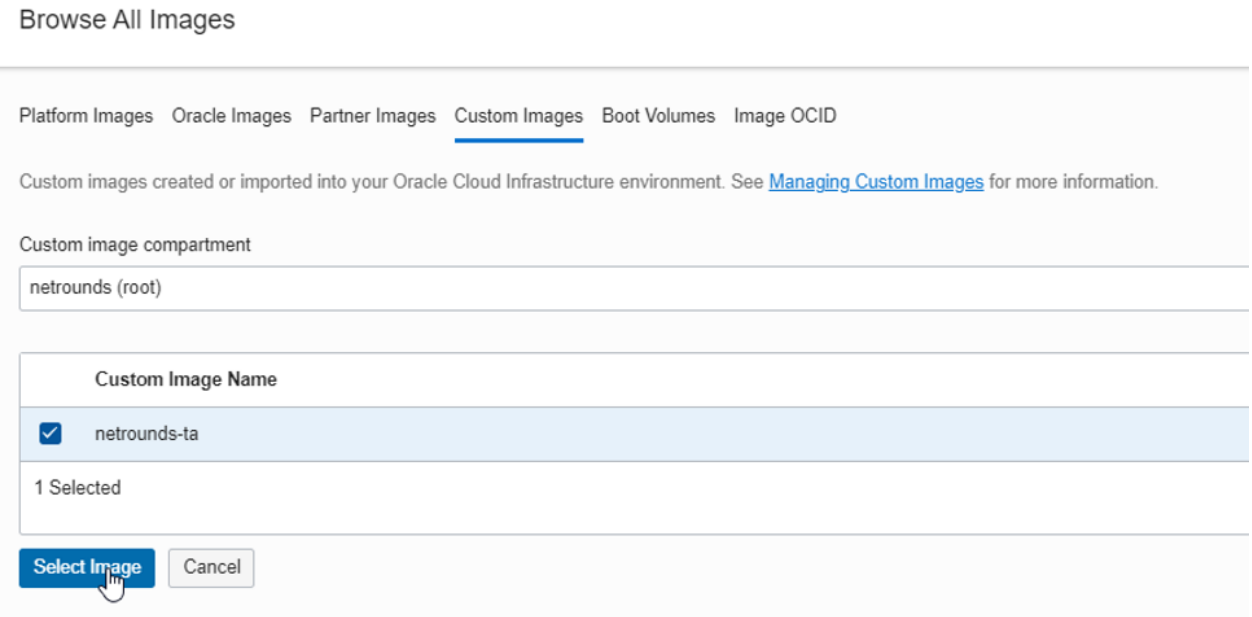
Deploying the Test Agent from the custom image

Important: At this point, if no network has been previously created in your Oracle Cloud, a default network will be created (for example, 10.0.0.0/16). This will be displayed in the configuration template (Virtual cloud network in *this step* (page 137) below). You will need to choose where to attach the image.

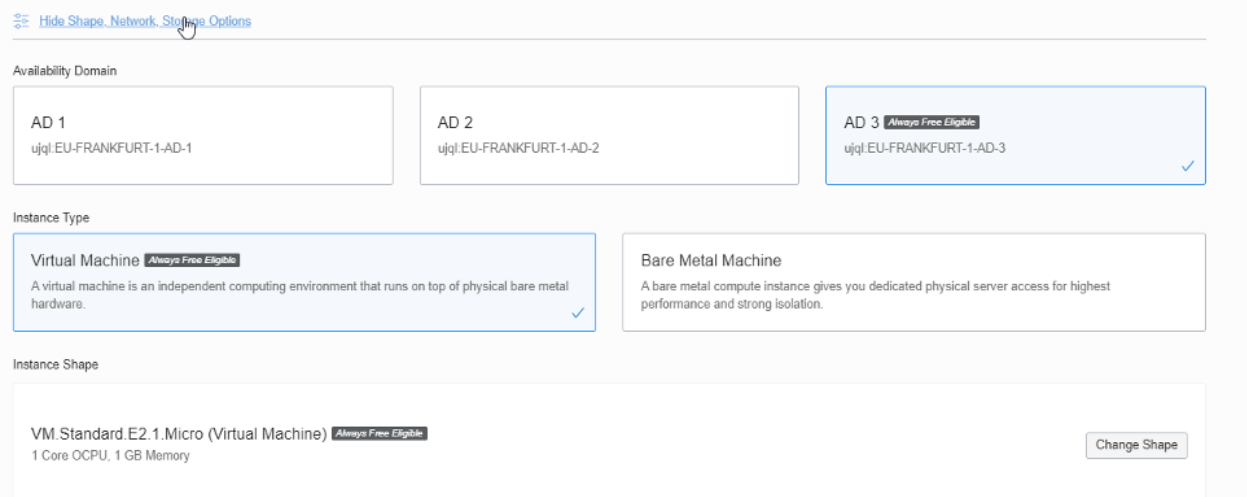
- Go to Instance > New Instance > Create New Instance.
- Enter a name for the instance.



- Click the Change Image Source button and select the Custom Images option. At this point the uploaded image (below, "netrounds-ta") should be displayed as a selectable item.



- Click Show Shape, Network, and Storage Options to configure things such as:
  - Availability domain
  - Instance type: Virtual machine or bare metal instance (a cost will be applied for the latter).
  - Instance shape: *The Test Agent has been tested with the default instance shape.* The available virtual machine shapes can be found at https://docs.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm#vm-standard.



Here are some examples of instance shapes:

X5-based shapes availability is limited to monthly universal credit customers existing on or before November 9, 2018, in the US West (Phoenix), US East (Ashburn), and Germany Central (Frankfurt) regions.

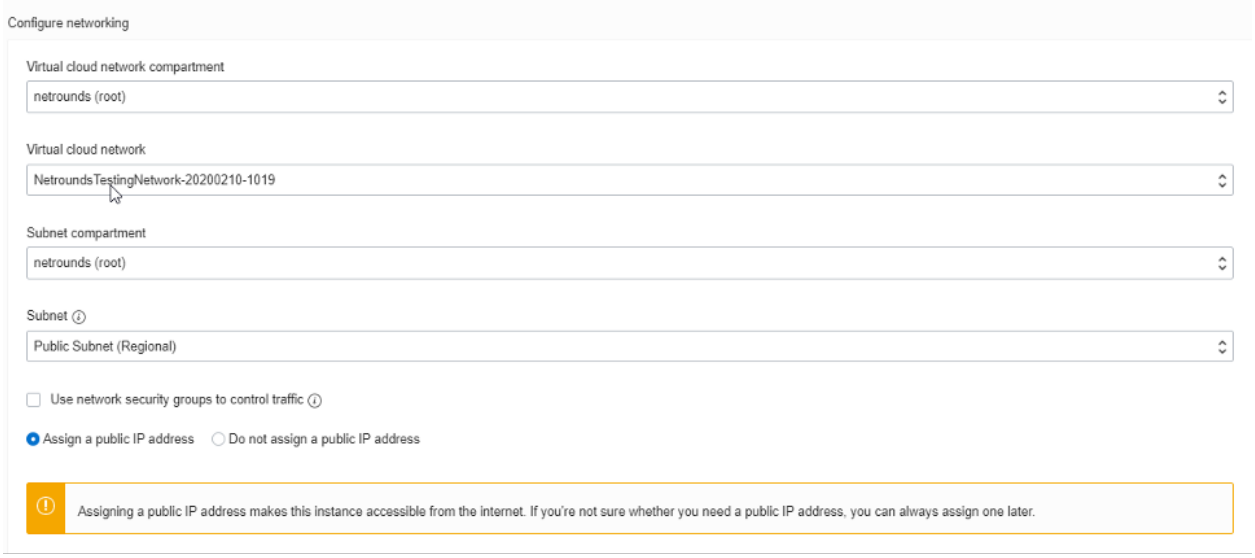
- **VM.Standard.B1:** X6-based standard compute. Processor: 2.2 GHz Intel Xeon E5-2699 v4.
- **VM.Standard2:** X7-based standard compute. Processor: 2.0 GHz Intel Xeon Platinum 8167M.
- **VM.Standard.E2.1.Micro:** E2-based standard compute with AMD CPUs. Processor: 2.0 GHz AMD EPYC 7551.
- **VM.Standard.E2:** E2-based standard compute. Processor: 2.0 GHz AMD EPYC 7551.

Shape	OCPU	Memory (GB)	Local Disk (TB)	Max Network Bandwidth	Max VNICs Total: Linux	Max VNICs Total: Windows
VM.Standard1.1	1	7	Block Storage only	600 Mbps	2	1
VM.Standard1.2	2	14	Block Storage only	1.2 Gbps	2	1

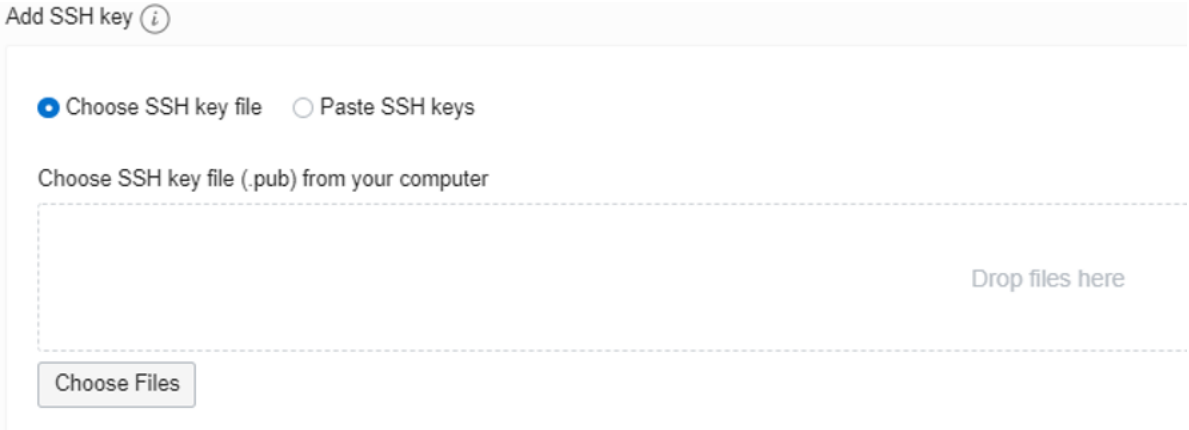
Note: The Test Agent should be assigned resources according to the specifications for production environments:

- 2 vCPUs
- 4 GB RAM
- 2 VNICs, 1 for management and 1 for testing
- Test Agent image disk size: 2 GB

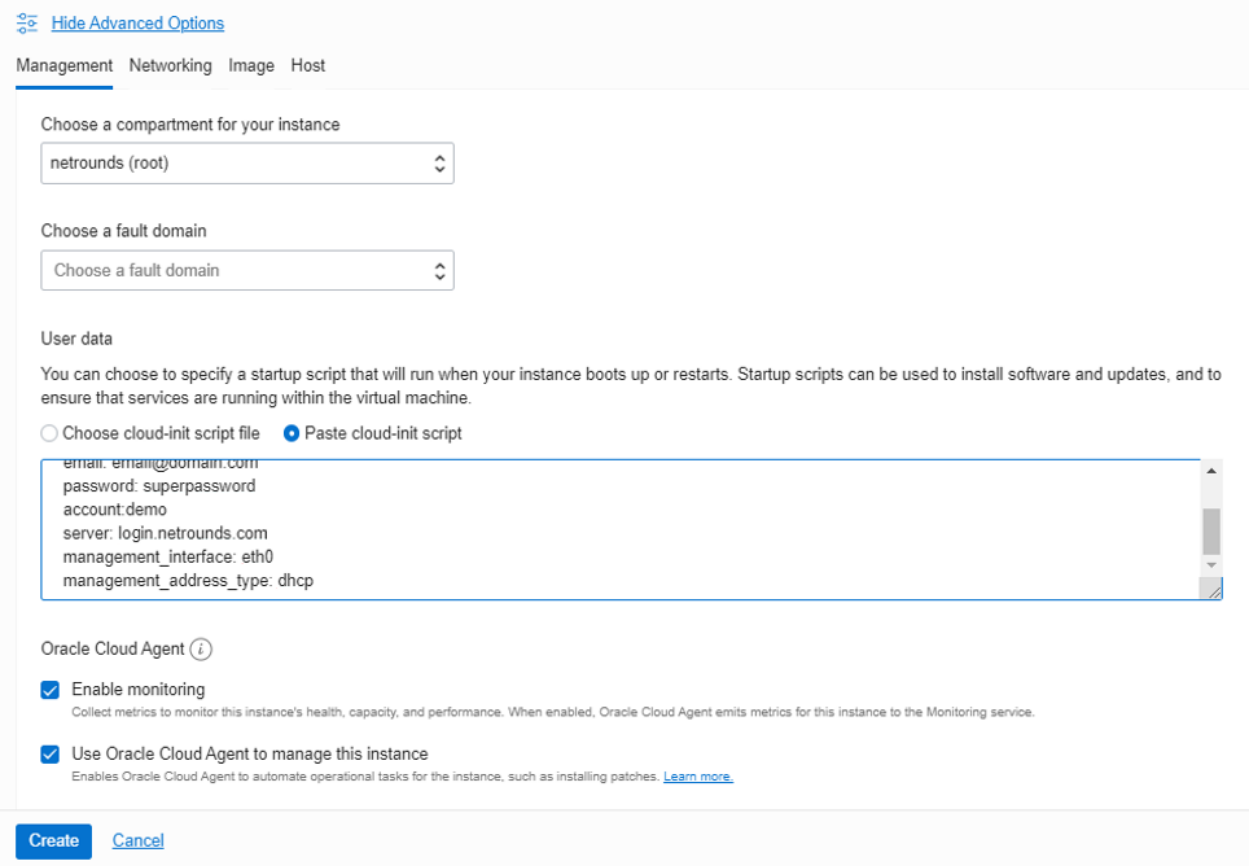
- Configure other basic settings such as the following (the rest are optional):
  - Virtual cloud network compartment: The compartment which the virtual machine will belong to.
  - Virtual cloud network: The virtual cloud network needs to be defined prior to deployment. If no cloud network has been defined, Oracle Cloud offers you to set up a virtual cloud network with the default security group. The default security group allows ingress SSH traffic to the network from any location and egress traffic without any restrictions.
  - Use network security groups to control traffic: On top of the security group, other policies can be applied to the specific virtual machine. These policies need to be defined prior to deployment.
  - Assign a public IP address: Here you assign a public IP to associate with the Test Agent.



- Choose an SSH key:



- *Optional:* If you want to use cloud-config to register the Test Agent, go to Advanced Options > Management and paste the cloud-config data in the User data section:

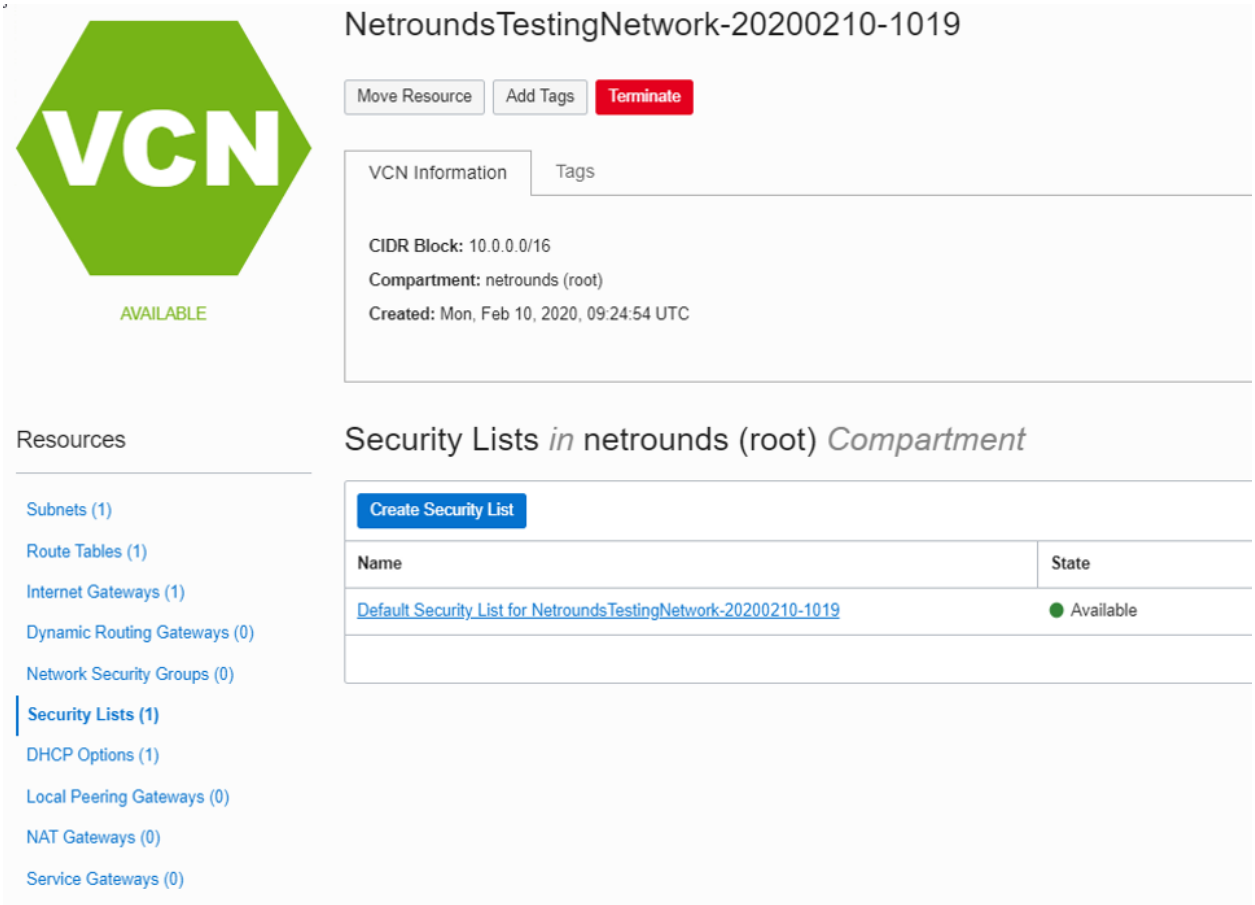


- Finally, click the Create button.

Example of network

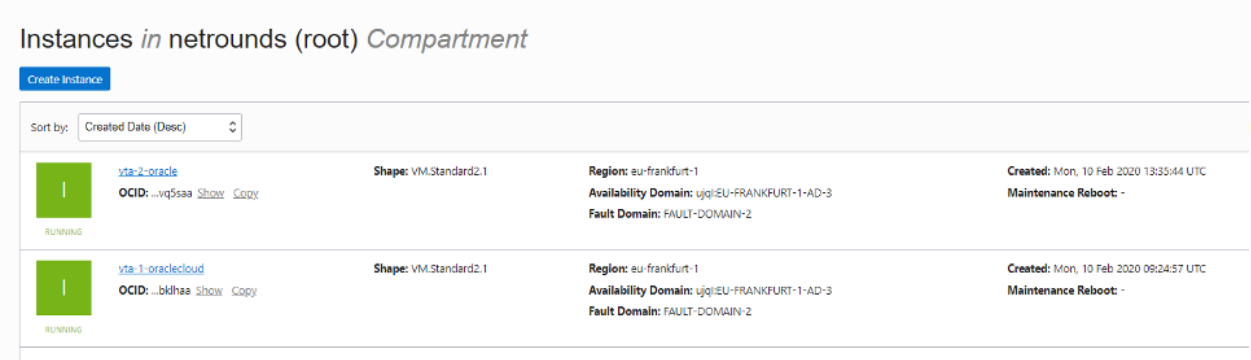
Below is an example of what the network the Test Agent is attached to may look like.

The compartment should be the same when deploying the virtual machine.



Displaying of running Test Agents

The Instances service will display successfully deployed Test Agent instances as follows:



Connecting to the Test Agent's serial console

Once the Test Agent is online, if a public IP address has been assigned to it, the serial console will be reachable using port 22 on that address. Remember that the security list needs to allow an SSH connection from the source IP addresses, as explained in the *Prerequisites* (page 129) section. The public SSH key must be assigned to the Test Agent at booting time. To log in to the Test Agent console, use the admin user, and the Test Agent menu will be displayed.

4.2.10.6 Deploying a virtual Test Agent in VirtualBox

Introduction

This page explains how to start a virtual Test Agent (vTA) from Routing Active Testing in VirtualBox and how to establish contact between the vTA and a Routing Active Testing Control Center residing outside VirtualBox.

Prerequisites

Control Center account

You need an account in a Routing Active Testing Control Center in order to access it: either the one belonging to the Routing Active Testing SaaS solution or one installed on-premise in your organization. If you do not already have a Routing Active Testing account, please contact your Juniper partner or your local Juniper account manager or sales representative.

vTA image

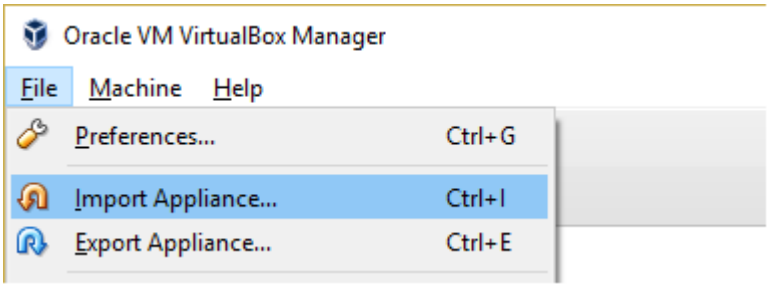
The VNF vTA image is provided either by one of Juniper's partners or directly by Juniper. If you do not already have the VNF vTA image, you can download it from the Control Center web GUI (whether SaaS or on-premise).

The vTA image is in OVA (OVF/VMDK) format and is packaged using the OVF Tool, which uses a SHA1 checksum. The OVF file specifies version VMX-09, since that is the lowest version which has the requisite functionality. The OVF file also specifies 512 MB RAM and 2 GB block storage for the vTA.

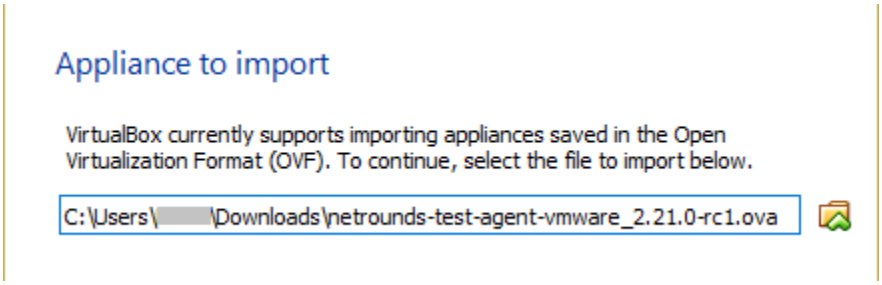
Setting up a virtual Test Agent in VirtualBox

Here is how to load the vTA from the OVA image:

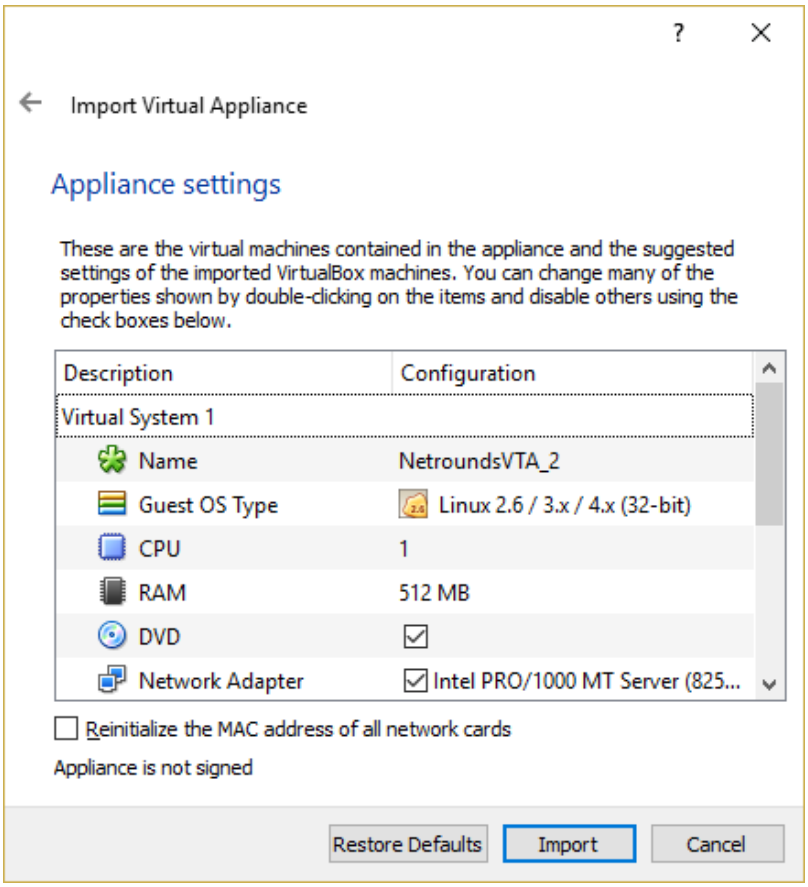
- In VirtualBox Manager, select the virtual machine named "vTA".
- From the File menu, select Import Appliance.



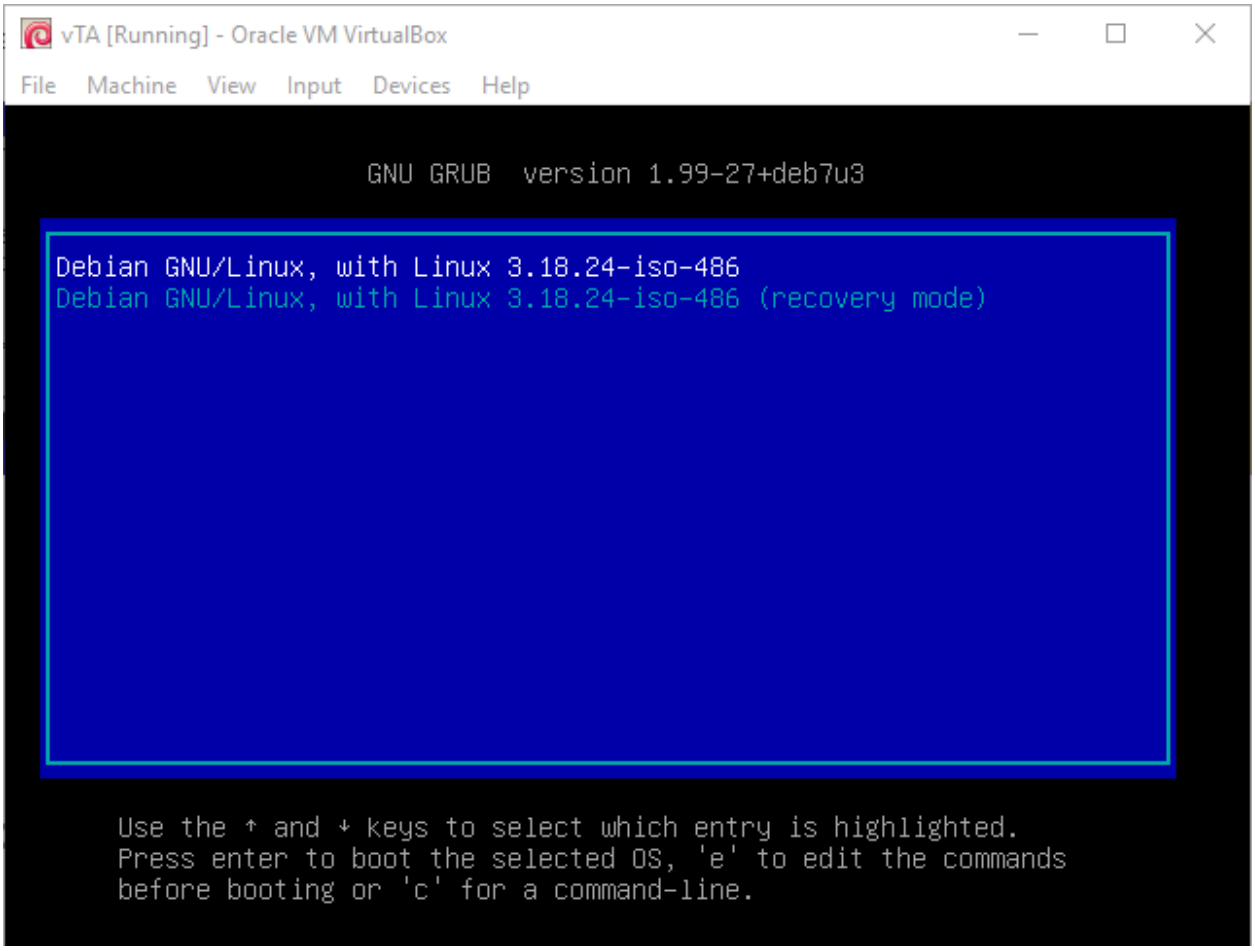
- Browse to select the OVA image file (*.ova). Then click Next.



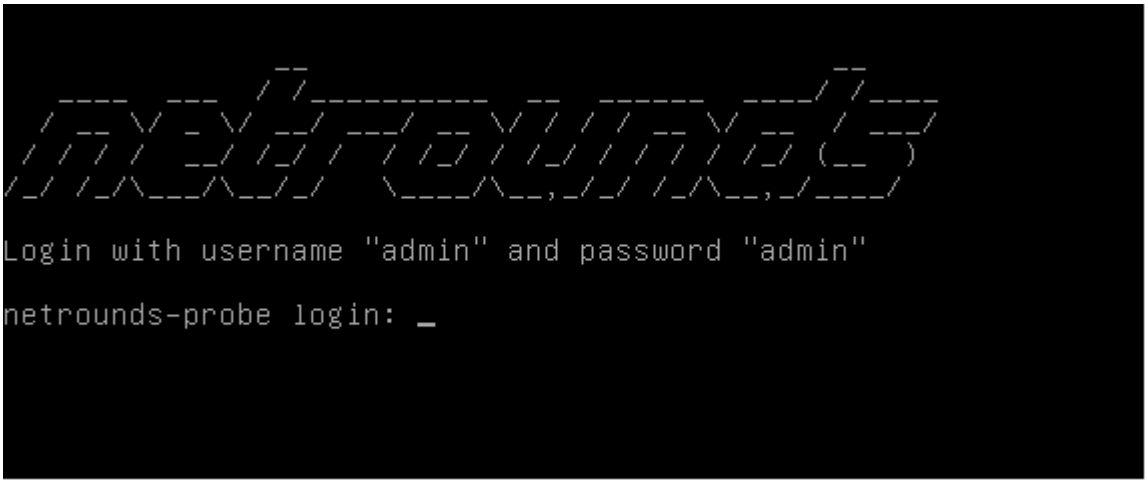
- Change the appliance settings if necessary. The default settings are however a good starting point for testing.
- Then click Import to go ahead with importing the OVA image.



Now start the virtual machine named “vTA” in VirtualBox Manager. You will be prompted to log in to the virtual machine. Normally the topmost item should be selected in the screenshot that follows:



- Log in to the Test Agent using the credentials “admin”/”admin” as indicated below.



The Test Agent local console menu now appears. Proceed to register the Test Agent with a Control Center as described *here* (page 205).

How to establish contact with Control Center

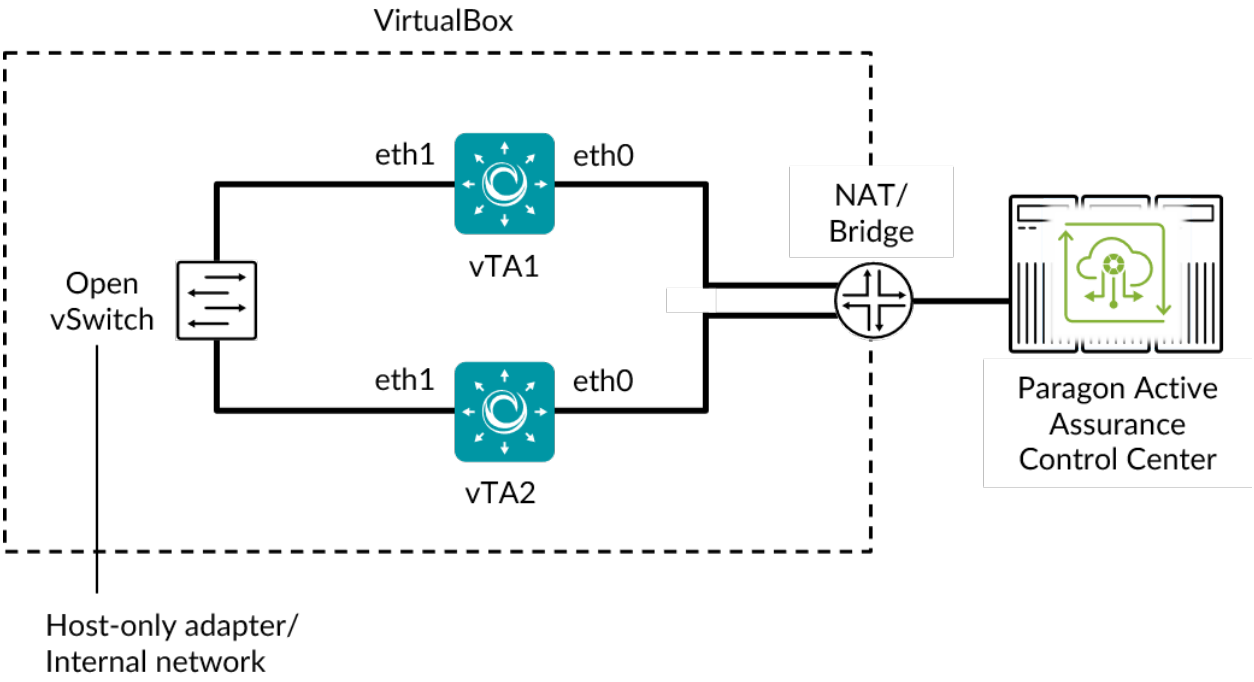
This section explains how to establish contact between virtual Test Agents deployed in VirtualBox and a Control Center residing outside VirtualBox.

For communication outside VirtualBox, either NAT or a bridge can be used. In the following, the use of NAT is assumed. A bridge may however be preferable if a more transparent setup is needed: for example, to permit communication in both directions between vTAs in VirtualBox and other Test Agents installed elsewhere.

Internally in VirtualBox, the following setup is recommended for each vTA:

- The vTA has one network adapter (labeled “eth0” in the diagram below) attached to NAT for communication with Control Center. This is the default setting for the network adapter which is predefined for the vTA.
- The vTA has another network adapter (labeled “eth1” in the diagram) attached to a VirtualBox host-only Ethernet adapter. This connection is used for communication between vTAs in the course of testing. How to configure this is covered *here* (page 145) and *here* (page 147).

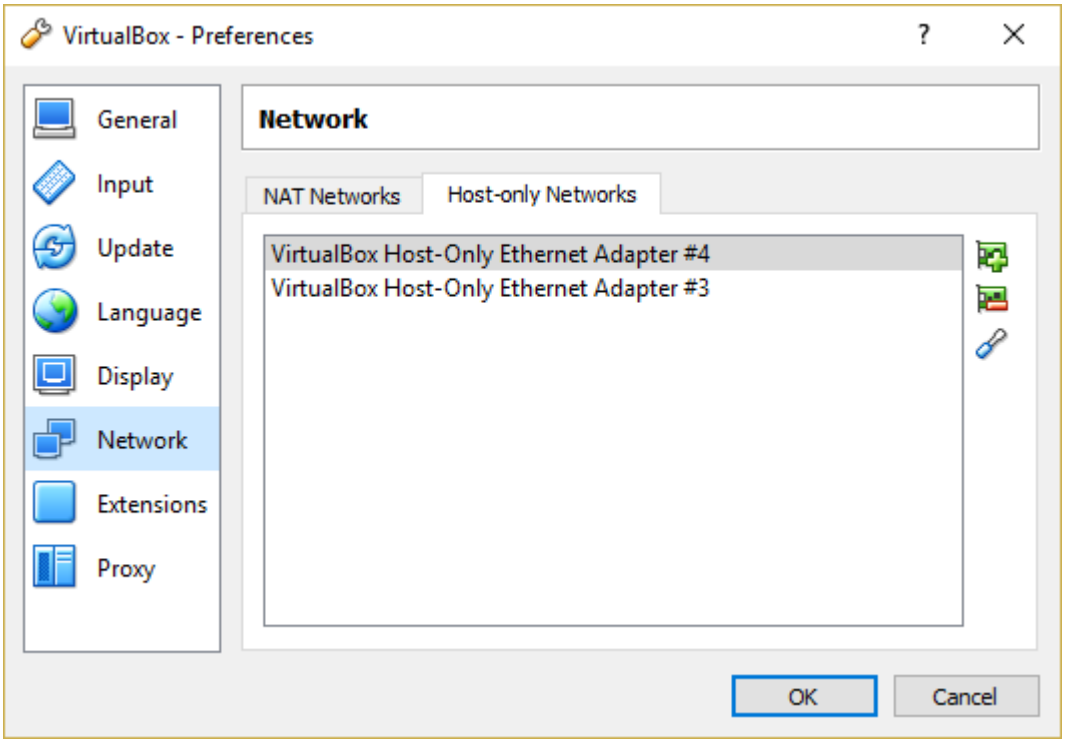
An overview of the setup is given in the diagram below.



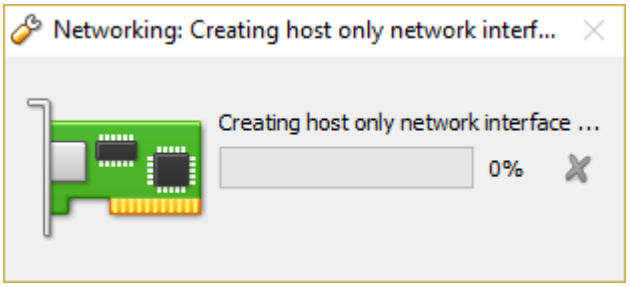
A VirtualBox host-only Ethernet adapter allows you to access the vTA virtual machines directly from the host OS, which is convenient especially in a lab environment, where security is not a primary concern. VirtualBox also offers the option Internal Network, which does not permit any direct connection between the virtual machines and the host (but is otherwise similar to Host-only Adapter). You may wish to use the Internal Network option if you want an entirely closed network inside VirtualBox, not accessible from external hosts. See *this section* (page 147) for instructions on how to connect the vTA to an Internal Network adapter.

Setting up a VirtualBox host-only Ethernet adapter

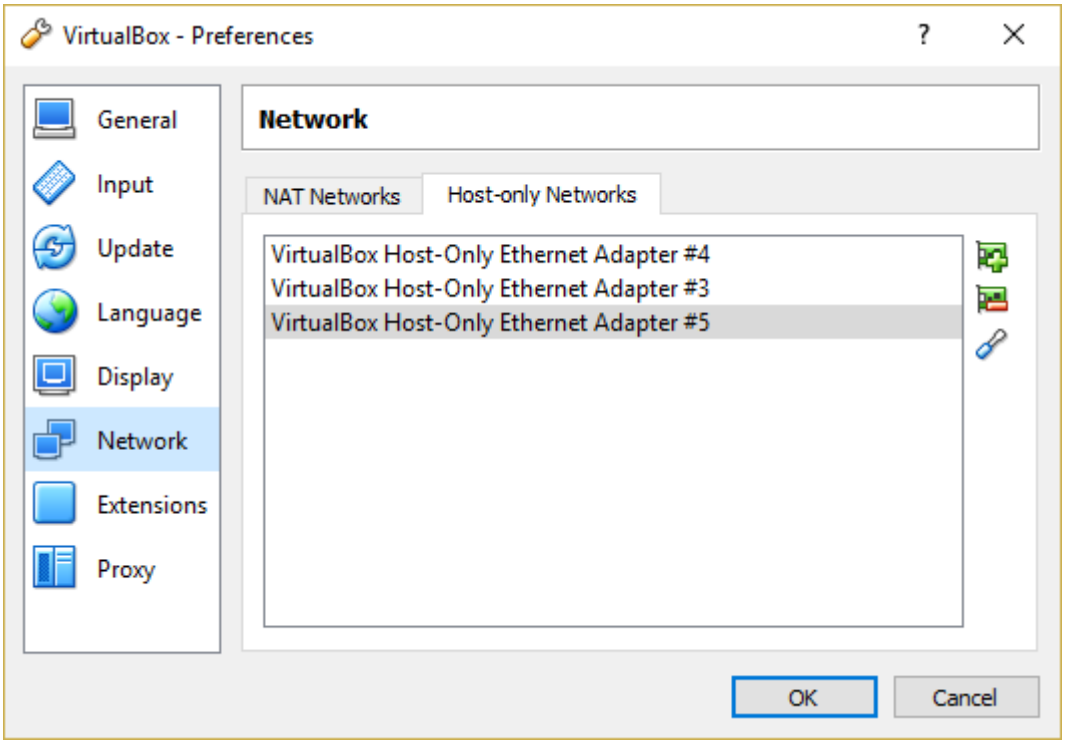
- From the File menu, select Preferences.
- Select Network, and click the Host-only Networks tab.



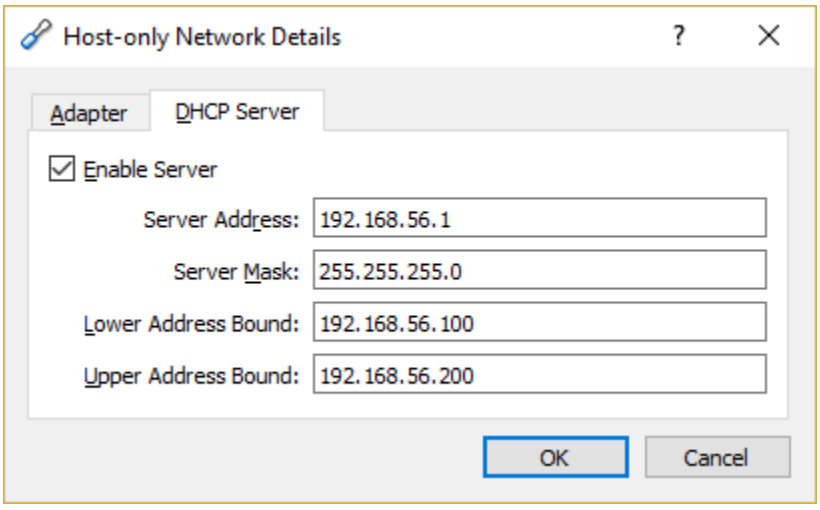
- Click the  ("New") button.
- Wait for this procedure to finish:



- A new VirtualBox host-only Ethernet adapter will now appear in the listing:



- Double-click the new Ethernet adapter.
- In the dialog that appears, select the DHCP Server tab.
- Check the Enable Server box.
- Enter Server Address and Server Mask, and set lower and upper address bounds. An example is shown in the screenshot below.

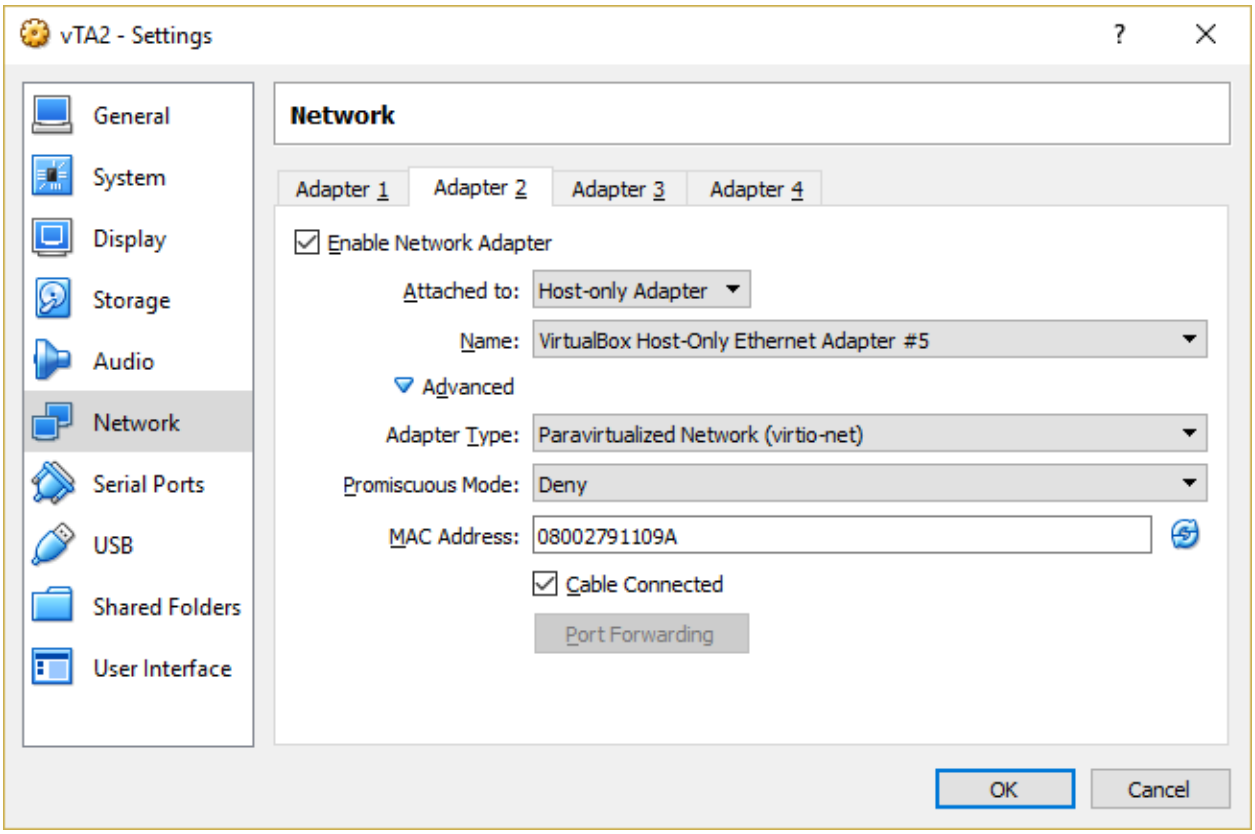


Connecting vTAs to a host-only Ethernet adapter

For each vTA in a subset that should be able to communicate with each other, do the following:

- Power off the vTA if it is currently running.
- With the vTA selected in the main window, click Settings.
- In the left-hand pane, select Network.
- The Adapter 1 tab defines the default network adapter which is attached to NAT. This network adapter can be left as-is.
- Click the Adapter 2 tab. Here we will define a second network adapter for the vTA and attach it to the host-only adapter created in *this section* (page 145).
- Check the Enable Network Adapter box.
- Under Attached to, select Host-only Adapter.
- Under Name, select the adapter that you created in *this section* (page 145).
- Expand the Advanced section.
- Under Adapter Type, select “Paravirtualized Network (virtio-net)”.
- Make sure that Cable Connected is checked.
- Finish by clicking OK.

Refer to the screenshot below.

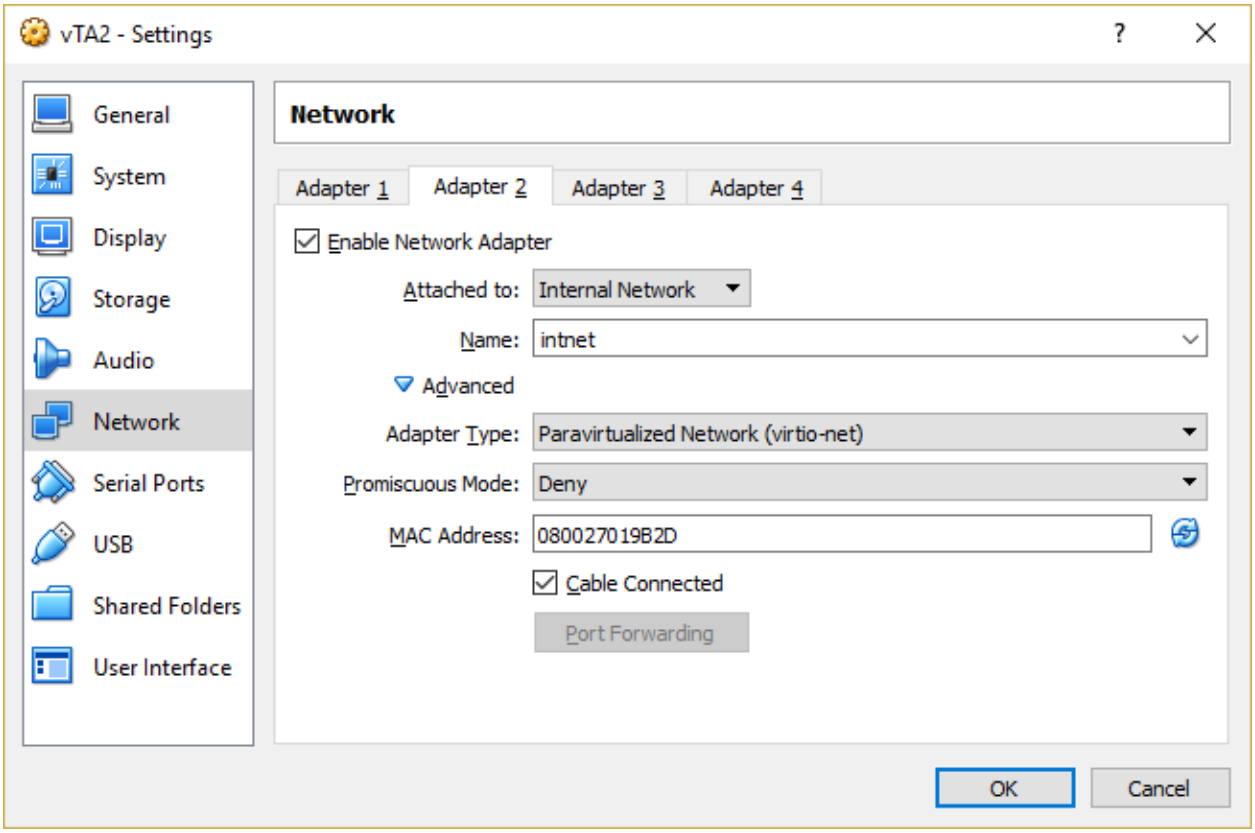


Connecting vTAs to an Internal Network Ethernet adapter

If you are using an Internal Network adapter, here is how to set up the vTAs:

- Navigate to the vTA network adapters as described *here* (page 147).
- Check the Enable Network Adapter box.
- Under Attached to, select Internal Network.
- Expand the Advanced section.
- Under Adapter Type, select “Paravirtualized Network (virtio-net)”.
- Make sure that Cable Connected is checked.
- Finish by clicking OK.

Refer to the screenshot below.



4.2.10.7 Deploying a virtual Test Agent in VMware

Introduction

This page explains how to start a virtual Test Agent from Routing Active Testing as a vApp on a VMware virtual machine.

Prerequisites

Control Center account

You need an account in a Routing Active Testing Control Center in order to access it: either the one belonging to the Routing Active Testing SaaS solution or one installed on-premise in your organization. If you do not already have a Routing Active Testing account, please contact your Juniper partner or your local Juniper account manager or sales representative.

vTA image

The VNF vTA image is provided either by one of Juniper's partners or directly by Juniper.

The vTA image for VMware is provided in OVA (OVF/VMDK) format and is packaged using the OVF Tool which uses a SHA1 checksum. The OVF file specifies version VMX-09, since that is the lowest version which has the required functionality.

The OVF file also specifies 512 MB RAM and 2 GB block storage for the vTA.

Uploading and deploying a vTA image

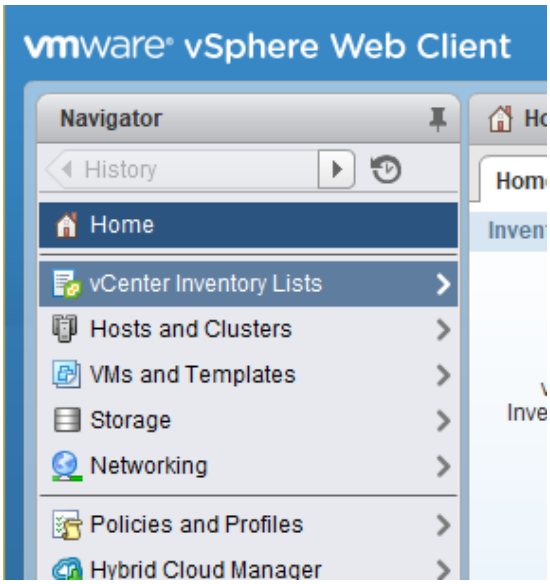
Once you have your vTA image, you need to upload it to your VMware environment and deploy it. This can be done either via the VMware vSphere Client or with the OVF Tool.

The supplying of Routing Active Testing user data is done in the process of this deployment.

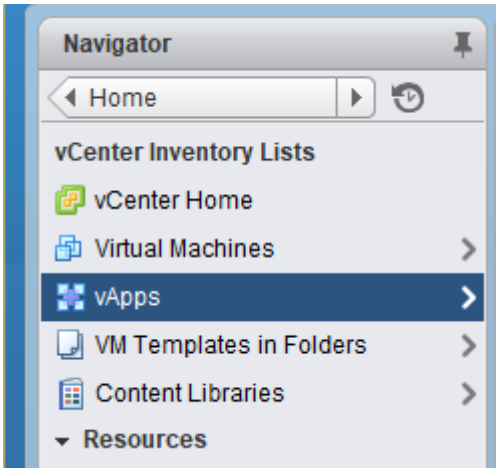
Uploading and deploying a vTA image via vSphere Client

This is possible only in Windows and iOS. If you are using a different operating system, you need to use the method in *this section* (page 155) instead.

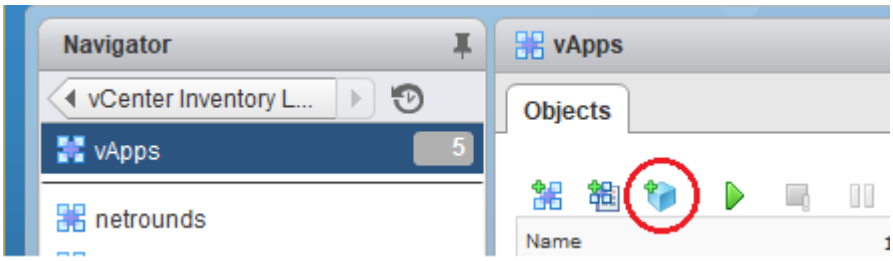
- Log in to vSphere Client.
- In vSphere Client, navigate to vCenter Inventory Lists.



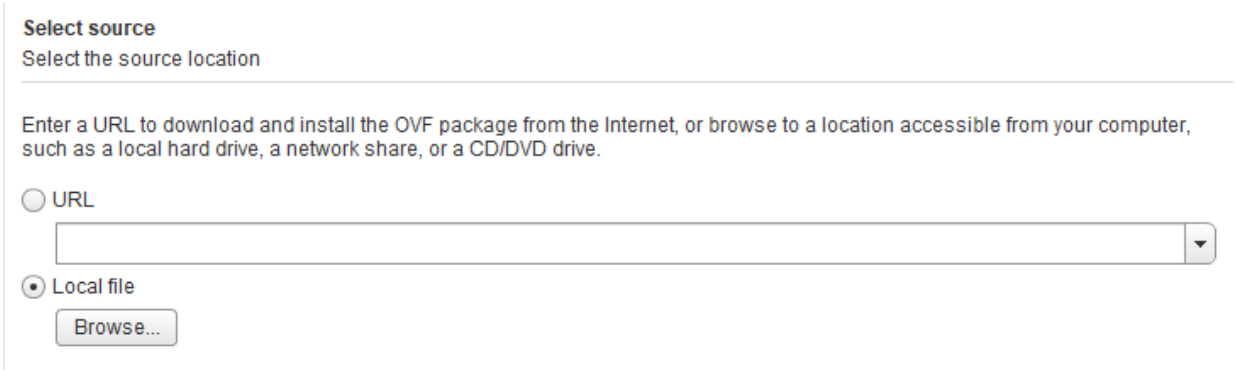
- Select vApps.



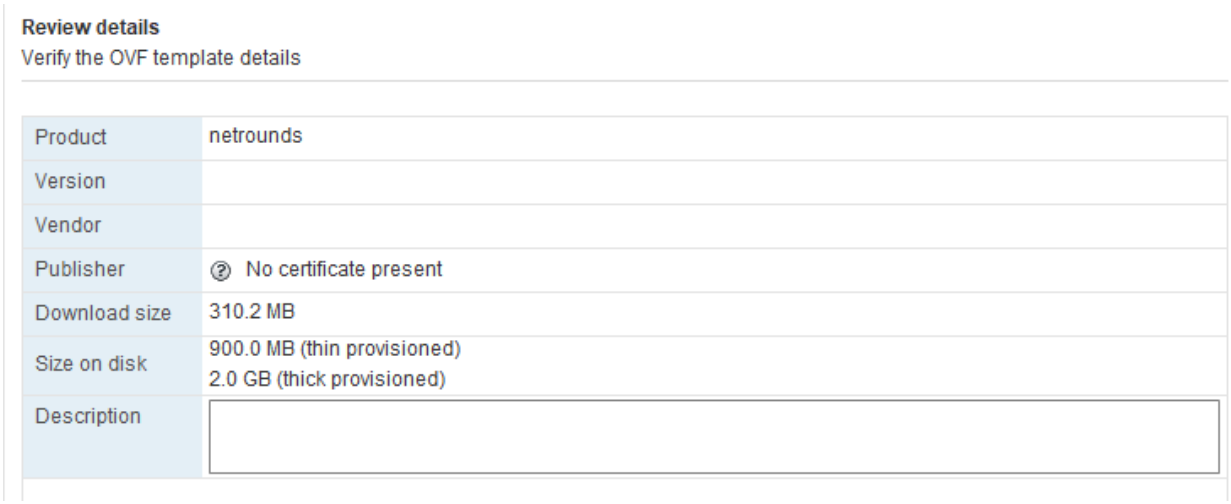
- Click the Deploy OVF template button (circled in the screenshot below).



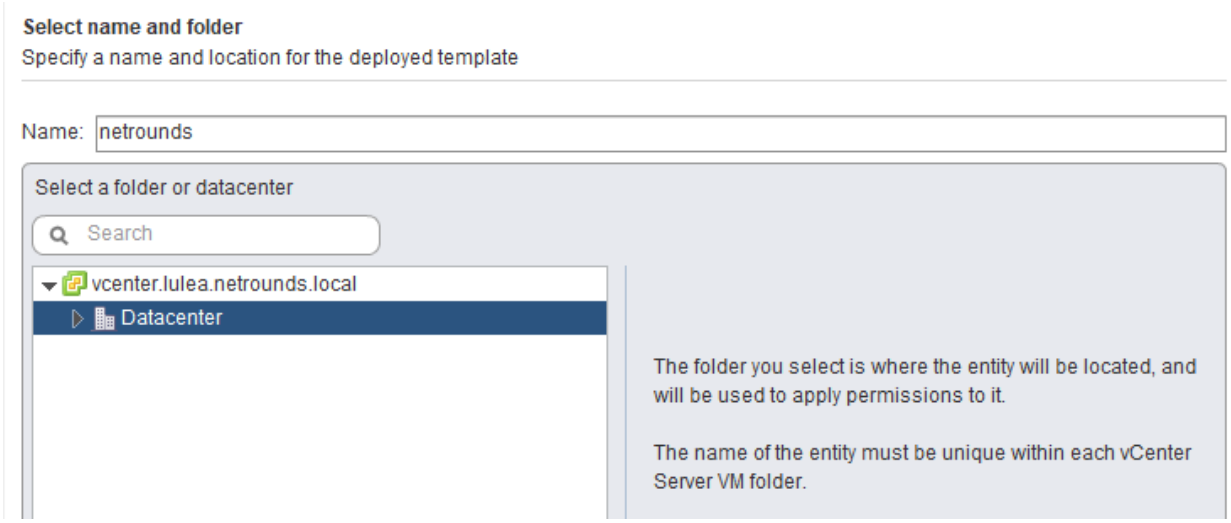
- In the wizard that appears, select Local file and browse to select your OVA/OVF file. Then click Next.



- On the Review details screen, click Next.



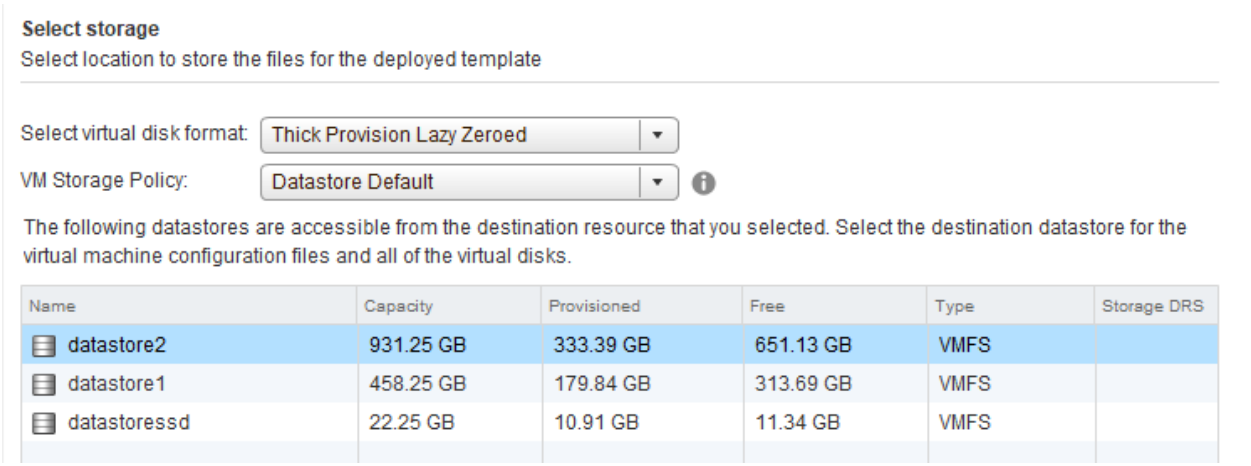
- On the Select name and folder screen, Name is predefined as “netrounds”. Select a *folder* or *datacenter* as exemplified in the screenshot below. Then click Next.



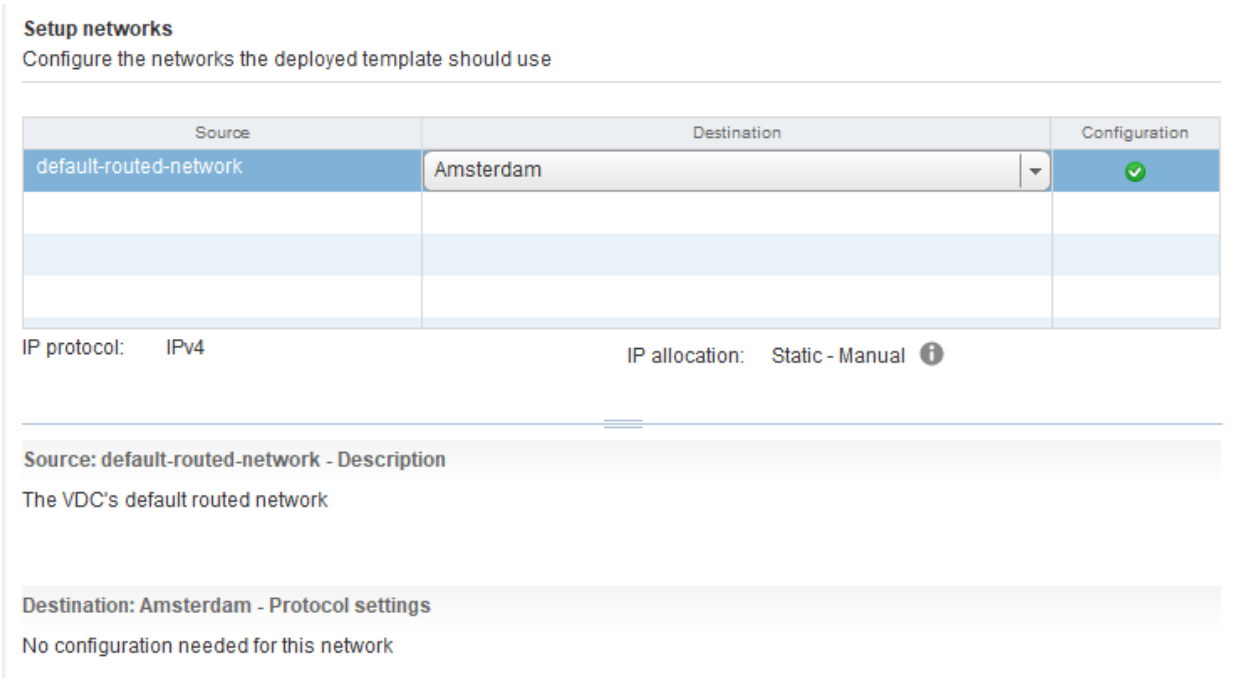
- On the Select a resource screen, select where to run the deployed template. Click Next.



- On the Select storage screen, the settings can be left as-is. Select a datastore in which to store the files for the deployed template. Click Next.



- On the Setup networks screen, edit the configuration if necessary; otherwise, no action is required here. Continue by clicking Next.



- On the Customize template screen, you need to fill in your Routing Active Testing cloud-init config (“user-data”) in base64-encoded format.

The cloud-init config is as shown below. Replace the values as appropriate. Note that lines with parameter settings must be indented as shown. Lines where the default value is kept can be omitted.

```
#cloud-config
paa_test_agent:
  name: MyvTA                              # vTA name
  email: john.doe@example.com              # NCC user email
  password: secret                         # NCC user password
  account: theaccount                      # NCC account (short name, found in NCC URL)
  server: <login-server>:443               # NCC host and port (default == SaaS)
                                           #   Note: With an IPv6 server address the
                                           #   whole string including port must be in
```

(continues on next page)

(continued from previous page)

```
                                        #   double quotes
admin_password: secret                  # Admin user password. Use null to
                                        #   disable.
root_password: secret                   # Menu root shell access password. Use
                                        #   null to disable.
                                        #
management_interface: eth0              # Test Agent management interface
management_mtu: 1500                    # MTU on management interface
management_address_type: dhcp           # Can be "dhcp" or "static"

## Set the following if using "static" above:
# management_ip: 192.168.1.2/24
# management_dns: 8.8.8.8, 8.8.4.4
# management_default_gateway: 192.168.1.1
# management_ntp: time.google.com

## Set the following if using an HTTP proxy:
# http_proxy: myproxy.lan
# http_proxy_port: 80
# http_proxy_auth_type: none            # Can be "none" or "basic"
# http_proxy_username: johndoe
# http_proxy_password: secret

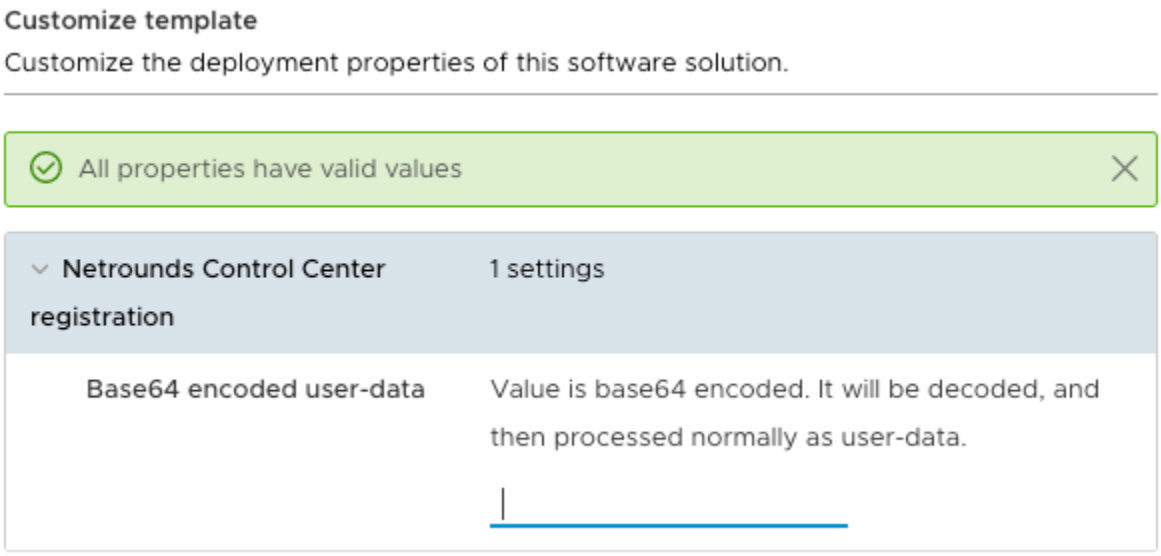
## Note: IPv6 management requires special config, see separate documentation
# management_enable_ipv6: False
# management_ntp_allow_ipv6: False
# management_address6_type: none        # Can be "dhcp", "slaac", or "static"

## Set if "static". Note: Use CIDR format for IP
# management_ip6: 2001:db8:85a3::8a2e:370:7334/64
# management_dns6: 2001:4860:4860::8888, 2001:4860:4860::8844
# management_default_gateway6: <gateway IP address>
```

- In Linux you can use the base64 command to do the encoding:

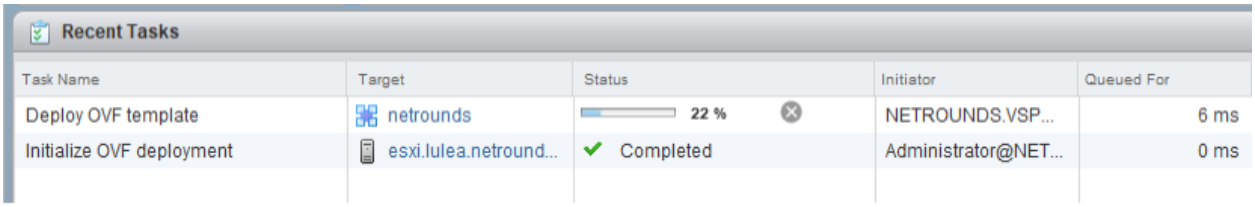
```
base64 <user-data file name>.txt
```

- Then click Next.



- Finally, on the Ready to complete screen, review your settings. Then click Finish.

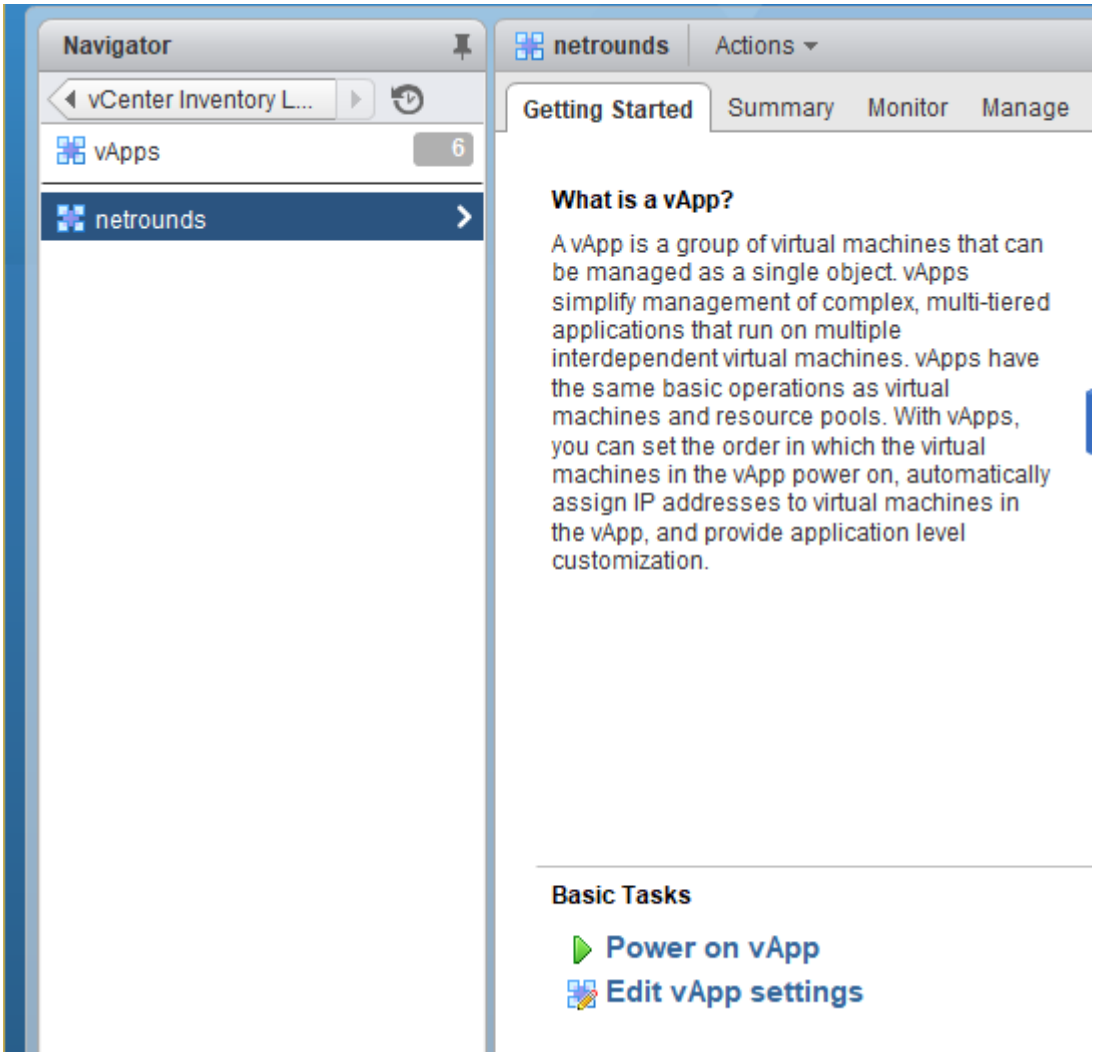
The OVF template will now be deployed in VMware. This will take a non-trivial amount of time; the progress of the deployment is shown in the Recent Tasks pane in vSphere Web Client:



Powering on the vTA

For the vTA to come online, you must power it on. In vSphere Client, do the following:

- From the Home screen in vSphere Client, navigate to vCenter Inventory Lists > vApps, and select your vApp (by default named “netrounds”).



- Click the Power on vApp link to power on the “netrounds” vApp. This powers on the NetroundsVTA virtual machine as well.

The vTA will now register with Control Center and appear in its web GUI under Test Agents. Check for the vTA name in that view to verify that the vTA has registered.

Uploading and deploying a vTA image with OVF tool

With OVF Tool the procedure for vTA deployment is as follows:

- You first need to configure vTA user data in the OVF file. To this end, uncompress the OVA file, which among other things contains the OVF.
- Open the OVF file in your text editor of choice and scroll down to the `ovf:ProductSection` tag:

```
<ovf:ProductSection ovf:class="" ovf:instance="" ovf:required="true">
<ovf:Info>Information about the installed software</ovf:Info>
<ovf:Category>Routing Active Testing Control Center registration</ovf:Category>
...
```

Inside that tag you will find several `ovf:Property` tags, each of which controls one data entry, identified by its `ovf:key`. To change the value of a Property, edit its `ovf:value`:

```
...
<ovf:Property ovf:key="netrounds.http_proxy" ovf:password="false" ovf:type="string"
ovf:userConfigurable="true" ovf:value="">
    <ovf:Label>Test Agent HTTP-Proxy server</ovf:Label>
    <ovf:Description>The address to a HTTP-Proxy. This is optional</ovf:Description>
</ovf:Property>
...
```

- When you are done configuring vTA user data, compress the files back into an OVA again (use the tar format and then replace the file extension).
- You are now ready to upload your vTA image with OVF Tool. The vTA image also needs to be powered on in the same command. Use this syntax:

```
$ ovftool --acceptAllEulas "--datastore=DATASTORE" "--network=NETWORK" --powerOn file.
↪ova vi://USER:PASSWORD@SERVER/DATACENTER/host/HOST
```

Here, each CAPITALIZED word should be replaced with the appropriate value, and `file.ova` is the name of your OVA file.

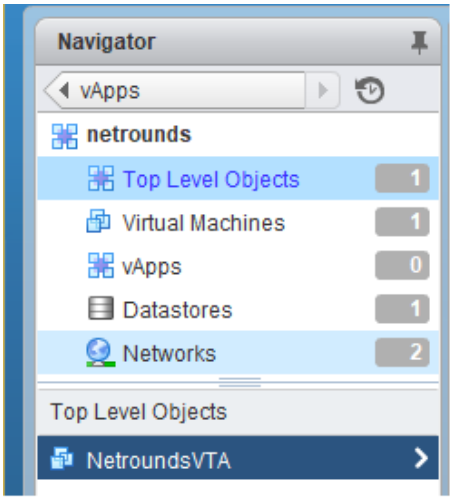
Example:

```
content_copy zoom_out_map
$ ovftool --acceptAllEulas "--datastore=datastore1" "--network=VM Network" --powerOn␣
↪vTA.ova vi://admin@netrounds.vsphere:mypassword@vcenter.lulea.netrounds.local/
↪Datacenter/host/esxi.lulea.netrounds.local
```

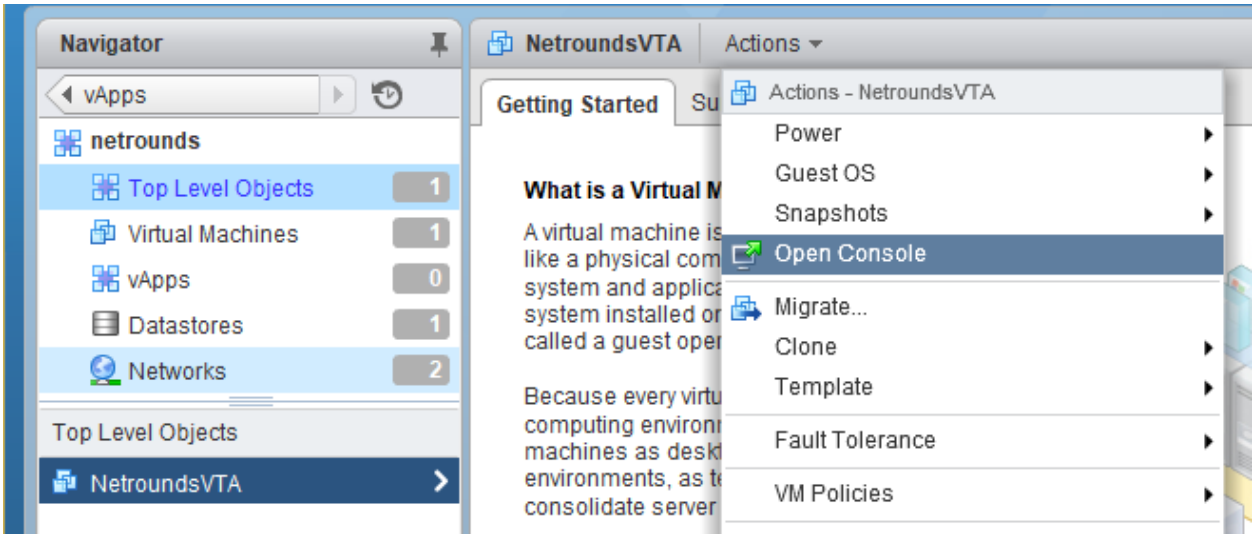
Troubleshooting

If the vTA does not show up in Control Center, it is useful to open the vTA's local console to investigate the cause of the problem.

- From the Home screen in vSphere Client, navigate to vCenter Inventory Lists > vApps, and select your vApp (by default named "netrounds").
- Click the "netrounds" vApp once more in the left-hand navigation pane, then click Top Level Objects.
- Click the "NetroundsVTA" object.



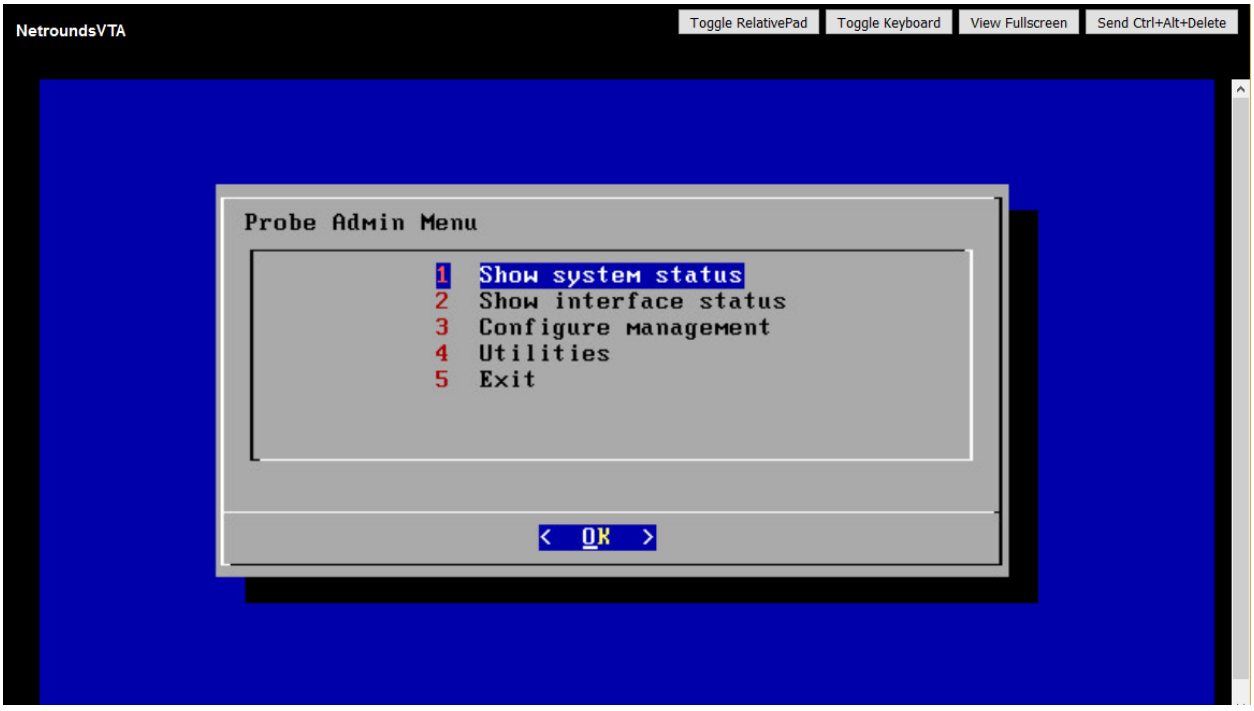
- In the right-hand pane, at the top, open the Actions menu and select Open Console.



- The console opens on a new tab. If you do not see the prompt shown below, click the Send Ctrl+Alt+Delete button in the top right corner.



- Log in using the credentials indicated. You are taken to the top-level Probe Admin Menu:



The functionality in this menu is described *here* (page 209). The following functions are particularly helpful:

- Utilities > Ping for checking that the vTA has a working internet connection.
- Utilities > Troubleshoot connection for verifying that the Routing Active Testing management connection is working.

4.2.10.8 Description of the vTA VNF and its requirements

The vTA VNF is capable of running in a plain, "vanilla" environment using a standard cloud configuration and orchestration based on one of the platforms mentioned in this documentation. There might be some limitations in terms of performance and also some minor limitations in terms of jitter and delay accuracy depending on your infrastructure and how heavily loaded it is. However, for early proof-of-concepts and evaluations, this should not be a major issue. To obtain line rate packet generation and optimal usage of your specific hypervisor environment, an integration project is required.

The vTA VNF consists of a single standalone VNF. However, the VNF must be able to connect and communicate securely with Control Center, which is not a VNF. Control Center is readily available in the public cloud (in addition to private cloud installations), something which simplifies test and evaluation projects.

Interfaces trust the natural OS bootstrap order in terms of how they are identified.

The performance is dependent on the underlying hardware. The more powerful the hardware, the higher the performance. For a 3 GHz quad-core processor, achievable performance is up to 10 Gbit/s using five concurrent unidirectional TCP streams.

The minimum recommended specification is: 1 vCPU, 512 MB RAM, and 2 GB of block storage.

It is assumed that a generic VNF manager which is not part of the Routing Active Testing solution does the instantiation, scaling, and termination of the vTA VNF.

The vTA VNF needs to register with Control Center to receive commands. For public cloud Control Center scenarios, the VNF needs connectivity to the Internet from the eth0 interface. For plug-and-play configuration of the VNF, DHCP should be used for IP addressing of the vTA's interfaces, as well as for assignment of an available DNS server.

The VNF will resolve the Control Center address and initiate an outbound connection using TCP. To successfully connect and authenticate itself to the correct Routing Active Testing account, the VNF needs to have credentials provided to it during initialization. The VNF supports cloud-init and config-drive for this purpose for its day-zero configuration. Once the VNF has connected to Control Center, it can be controlled either via a web browser or through the Routing Active Testing cloud API to start monitoring user experience KPIs, conduct a service turn-up test, or perform on-demand troubleshooting tests. The connection is an encrypted OpenVPN connection.

The vTA VNF also requires synchronization to an NTP server in order to achieve accurate delay and jitter measurements. By default, Test Agents will synchronize their internal clock to time.google.com, a service provided by Google; however, any NTP server (internal or external) can be used.

Rescaling of the VNF again needs to be handled by a generic VNF manager. For example, if the available connection bandwidth is increased, the VNF might need to be scaled up to be able to push enough bandwidth through the link for testing purposes.

4.2.11 Installing a Test Agent Application

This page describes how to install a Test Agent Application under Linux. The Test Agent Application download includes all software needed for installation on top of your existing operating system.

The Test Agent Application can be run either in a Docker container or as a native app in Linux. The two possibilities are covered below.

Note that a Test Agent Application always needs a counterpart to be able to run performance and stability tests: either another Test Agent Application, installed on a separate PC or server, or a Test Agent of a different type. If you need assistance, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

Note: For installing Test Agent Applications on Juniper routers, please turn to the Junos® OS Evolved Software Installation and Upgrade Guide. This document is found in TechLibrary under “Junos OS Evolved”:

- Installing the Test Agent
- Using the service paa install command
- Uninstalling the Test Agent
- Checking the status of the Test Agent

4.2.11.1 Prerequisites

If you have a firewall in place on the device or in the network, make sure that the Test Agent Application is allowed to establish an outgoing session towards the Routing Active Testing server on TCP port 6800.

You do not need to open any incoming connections, since the Routing Active Testing server will communicate in the reverse direction on the same TCP session that the Test Agent Application initiated.

4.2.11.2 Installing a Test Agent Application in a Docker container

To install and run the Test Agent in a Docker container, you can use either the official Routing Active Testing Docker image, available on Docker Hub, or a Docker image of your own. We cover both options in what follows.

For general Docker documentation, go to docs.docker.com.

Getting the official Docker image

The official Docker image is available on Docker Hub at https://hub.docker.com/r/netrounds/test-agent-application. Instructions on how to register and start the Test Agent are given on that page.

Building your own custom Docker image

- To build your own Docker image for the Test Agent Application, you first need to download the release tarball from Control Center.
  - Click Test Agents in the main menu.
  - Click the Download button in the top right corner.
  - In the dialog that appears, download the Test Agent Application tarball.
- Unpack the tarball:

```
tar -xvzf paa-test-agent-application_<version_architecture>.tar.gz

# Remove the version from the directory name
mv paa-test-agent-application_<version> paa-test-agent-application/
```

- Create a custom Dockerfile in the directory where you unpacked the Test Agent tarball. An example is given below:

```
FROM debian:bullseye-slim

RUN apt-get update && \
    apt-get install -y ca-certificates iproute2 socat && \
    rm -rf /var/lib/apt/lists/*

COPY paa-test-agent-application/ /opt/paa-test-agent-application/
ENTRYPOINT ["/opt/paa-test-agent-application/paa-test-agent-application"]
```

- To build your own Docker container for the Test Agent Application, you use the `docker build` command:

```
docker build -t mycustom/paa-test-agent-application .
```

You can replace the name `mycustom...` with whatever you like. It is however used in the commands that follow below.

Running the custom container

- Register the Test Agent Application to Control Center if you have not already done so:

```
docker run --rm -v $(pwd):/config mycustom/paa-test-agent-application \
           register --config /config/agent.conf \
                    --ncc-host <Control Center hostname> \
                    --account <Control Center account> \
                    --email <Control Center user email> \
                    --password <Control Center user password> \
                    --api-token <API token for user in Control Center> \
                    --name <Test Agent name>
```

  The configuration file written by the Test Agent is now stored as `agent.conf` in the current directory.

- Then start the Test Agent with the configuration file:

```
docker run --network=host --cap-add=NET_ADMIN --device=/dev/net/tun -d \
           --privileged -v $(pwd):/config -v /var/run/netns:/var/run/netns \
           --log-opt max-size=10m --log-opt max-file=2 <Test Agent docker image> \
           --config /config/agent.conf -A [-T]
```

Note: If you want the Test Agent to use hardware timestamping, you need to add the `-T` option.

Note: If you want to do *5G core network testing* (page 358), you need to remove the `-A` option from the `register-run` command and add the option *–enable-interface-daemon* to that command.

The options `--cap-add=NET_ADMIN`, `--device=/dev/net/tun`, and `-D` are necessary for 5G RAN interfaces to work.

The `--privileged` option and the `/var/run/netns:/var/run/netns` argument allow the Test Agent to use network namespaces to route the traffic independently.

The `-A` option causes the Test Agent to detect all available namespaces and their interfaces.

The `--log-opt` option is used to limit the size and number of the logfiles produced by the Test Agent (these files can be inspected with the command `docker logs <container ID>`, which is useful for container management generally). Limiting the logfiles in some way is necessary since otherwise they will grow without limit. Above, the maximum logfile size is set to 10 MB and the maximum number of logfiles to 2, with log rotation applied to the data.

See the help output for more information about the command line arguments:

```
docker run --rm mycustom/paa-test-agent-application --help
```

4.2.11.3 Installing a Test Agent Application as a native app in Linux

Downloading the application

See *above* (page 160).

Installing the application

```
# Unpack the Test Agent Application tarball
tar -xvzf paa-test-agent-application_<version_architecture>.tar.gz

# Move the contents of the directory to a more permanent location
sudo mv paa-test-agent-application_<version> /opt/paa-test-agent-application

# Set root as owner and group of the new directory
sudo chown -R root:root /opt/paa-test-agent-application
```

Registering the application

To connect the Test Agent to Control Center, run the command below.

This command will register the Test Agent Application to Control Center and write its credentials to a file `/etc/paa-test-agent-application.conf`. The Test Agent Application will be tied to your Routing Active Testing account through an encrypted and secure connection, and it will appear in the Test Agents view in Control Center with an "offline" icon next to it.

If Control Center does not have a valid SSL certificate, add the `--ssl-no-check` option to the command. Note that this is not recommended for security reasons.

```
sudo /opt/paa-test-agent-application/paa-test-agent-application register \
    --config /etc/paa-test-agent-application.conf \
    --ncc-host <Control Center hostname> \
    --account <Control Center account> \
    --email <Control Center user email> \
    --password <Control Center user password> \
    --api-token <API token for user in Control Center> \
    --name <Test Agent name>
```

If you specify the Control Center user email as a command line argument, but do not specify a password, the Test Agent will prompt for the Control Center user password. For example:

```
sudo /opt/paa-test-agent-application/paa-test-agent-application register \
    --config /etc/paa-test-agent-application.conf \
    --ncc-host app.netrounds.com \
    --account myaccount \
    --email me@example.com \
    --name "My Test Agent"
Enter password for dev@netrounds.com in PAA Control Center:
```

If you do not specify an email address either, the Test Agent will prompt for the API token of the user in Control Center:

```
sudo /opt/paa-test-agent-application/paa-test-agent-application register \
    --config /etc/paa-test-agent-application.conf \
```

(continues on next page)

(continued from previous page)

```
    --ncc-host app.netrounds.com \
    --account myaccount \
    --name "My Test Agent"
Enter API token for user in PAA Control Center:
```

Note that, for enhanced security, the two methods just mentioned are the preferred ways to pass the user credentials during registration. However, in scenarios where other restrictions prevent an interactive registration of the Test Agent, you can instead pass credentials as command arguments `--password <user password>` and `--api-token <user API token>`, as indicated in the beginning of this section.

Make sure you escape any special characters in the password correctly in your shell.

Starting the application

To start up the registered Test Agent, run this command:

```
/opt/paa-test-agent-application/paa-test-agent-application \
    --config /etc/paa-test-agent-application.conf -A
```

Optionally, an explicit command `run` can be given here:

```
/opt/paa-test-agent-application/paa-test-agent-application run \
...
```

The `-A` option causes the Test Agent to detect all available namespaces and their interfaces.

On being started, the Test Agent will appear as "ready" in Control Center.

If Control Center does not have a valid SSL certificate, add the `--ssl-no-check` option to the above command. Again, note that this is not recommended for security reasons.

4.2.11.4 Registering and starting the application in one command

A special command `register-run` is provided which first registers the Test Agent with Control Center and then starts it. Its syntax combines that of the `register` and `run` commands given above.

For a Test Agent in a Docker container, use this:

```
docker run --network=host --cap-add=NET_ADMIN --device=/dev/net/tun -d \
           --privileged -v $(pwd):/config -v /var/run/netns:/var/run/netns \
           --log-opt max-size=10m --log-opt max-file=2 <Test Agent docker image> \
           register-run --config /config/agent.conf -A [-T]   \
                        --ncc-host <Control Center hostname> \
                        --account <Control Center account> \
                        --email <Control Center user email> \
                        --password <Control Center user password> \
                        --api-token <API token for user in Control Center> \
                        --name <Test Agent name>
```

Note: If you want to do *5G core network testing* (page 358), you need to remove the `-A` option from the `register-run` command and add the option *–enable-interface-daemon* to that command.

For a Test Agent installed as a native app in Linux, use this:

```
sudo /opt/paa-test-agent-application/paa-test-agent-application register-run \
    --config /etc/paa-test-agent-application.conf -A \
    --ncc-host <Control Center hostname> \
    --account <Control Center account> \
    --email <Control Center user email> \
    --password <Control Center user password> \
    --api-token <API token for user in Control Center> \
    --name <Test Agent name>
```

Note: If the registration cannot be done interactively, the `register-run` command should not be used since it would require storing user credentials in start-up scripts. In such a situation it is better to perform a two-step start-up where you first register the Test Agent and then start it.

Automatic application start

To have `systemd` start a previously registered Test Agent Application automatically on boot, add the following to `/etc/systemd/system/paa-test-agent-application.service`:

```
[Unit]
Description=Routing Active Testing Test Agent Application
After=network.target

[Service]
ExecStart=/opt/paa-test-agent-application/paa-test-agent-application \
    --config /etc/paa-test-agent-application.conf

[Install]
WantedBy=default.target
```

If Control Center does not have a valid SSL certificate, add the `--ssl-no-check` option to the ExecStart command.

Reload the `systemd` units, then enable and start the Test Agent Application:

```
systemctl daemon-reload
systemctl enable paa-test-agent-application.service
systemctl start paa-test-agent-application.service
```

Check if the Test Agent Application service started:

```
systemctl status paa-test-agent-application.service
```

The application's status icon should turn green in the Test Agents view, meaning that the application is ready to use.

Using the Test Agent Application

Log in to Control Center with your user credentials. If you have successfully registered and started the Test Agent, it should now be listed as online in the Control Center web interface, and you can start using it for measurements.

Troubleshooting

- Check that no firewalls are blocking the connection to Control Center on TCP port 6800.
- Check that Control Center has a valid SSL certificate or that you have added the `--ssl-no-check` option to the agent command.
- Check the logs from the agent. If you are using `systemd` to start the agent, the logs will be available using the command `journalctl -u test-agent-application`.
- To get more detailed logs, enable debug logging by adding the `--log-level DEBUG` option to the Test Agent Application start command.
- If you are still having problems, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

4.2.12 Making changes to the machine where the Test Agent is installed

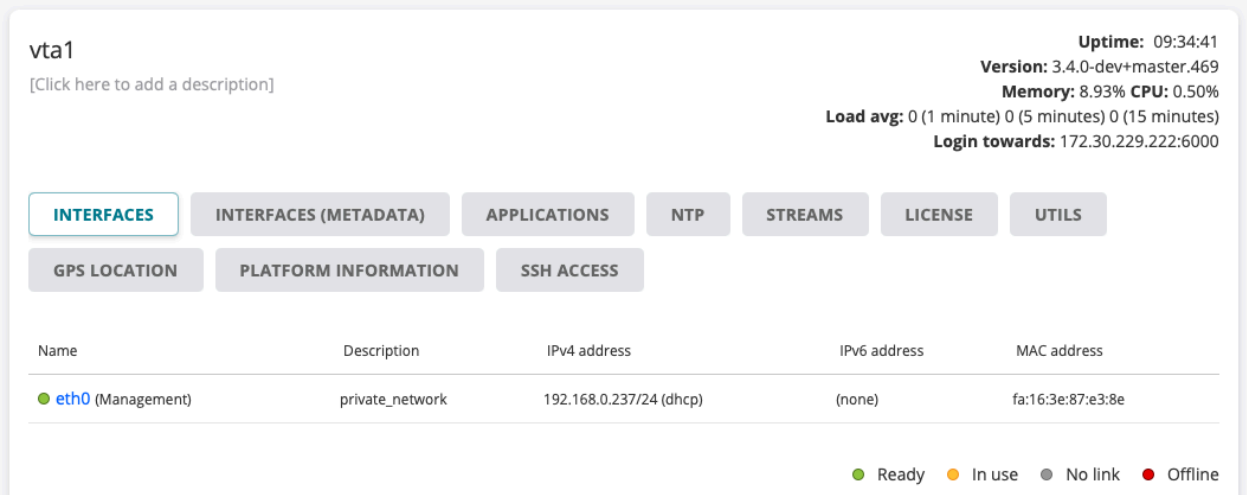
If you want to install an additional network interface card or other hardware on the machine where your Test Agent is running, be sure to first shut down the Test Agent. The Test Agent does not support hot swapping.

The same thing applies in a virtual environment. If you want to add extra network interfaces to the virtual machine where the Test Agent runs, you need to shut down and restart the Test Agent.

4.3 Configuring Test Agents from the Routing Active Testing GUI

4.3.1 Viewing Test Agent status and properties

To inspect the current status, current activity, and various properties of a Test Agent, click Test Agents on the main menu, then click the Test Agent in the listing that appears. The screenshot below shows an example.



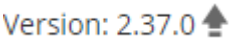
This page explains the status information found at top right on this screen. For coverage of the tabbed dialog beneath it, turn to the *other pages in this section* (page 165).

4.3.1.1 Test Agent status information (*top right*)

Uptime: The time the Test Agent has been online since logging in.

If the Test Agent is offline, a string "**`OFFLINE` since: <date and time>**" is displayed instead to help you understand why this is so; you can correlate that date and time with known actions and events.

Version: Version of Test Agent software. If newer software is available, an up arrow is shown next to the version number:



You can then click the arrow in order to upgrade the Test Agent software to the latest version.

Note: The upgrade will briefly impact performance as the Test Agent will reboot soon after initiating the upgrade procedure.

Memory: Memory currently in use on the Test Agent.

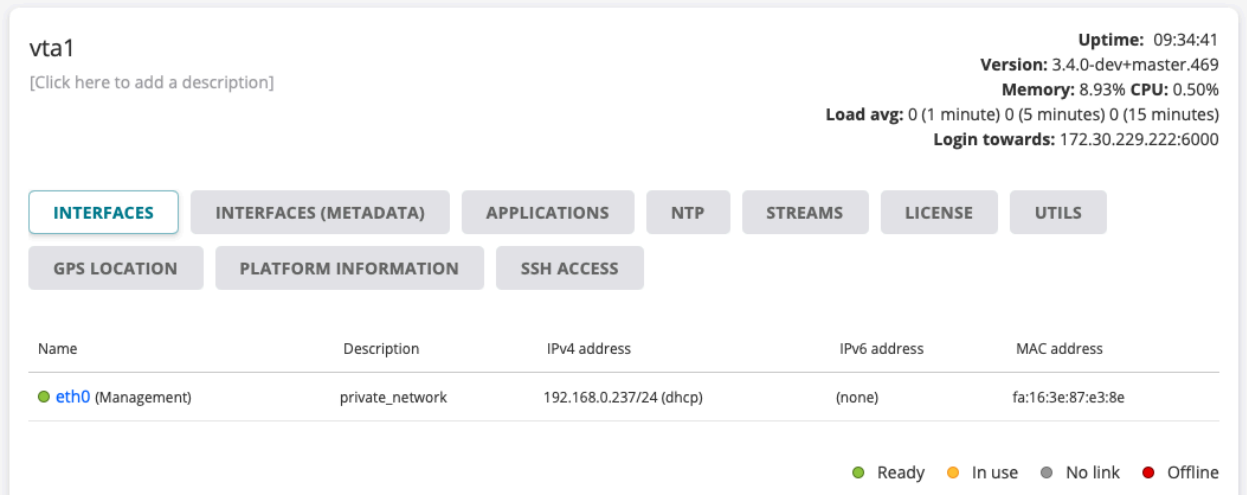
CPU: Current CPU load on the Test Agent.

Load avg: Average CPU load over the last 1 min, 5 min, and 15 min periods.

Login towards: Usually, this is the login server to which the Test Agents are connected. However, if a proxy is used, the proxy address is shown here instead.

4.3.2 Test Agent interface configuration

The Interfaces tab of the Test Agent configuration dialog lists the interfaces to the Test Agent.



4.3.2.1 Types of Test Agent interfaces

Test Agent interfaces are of one of the following kinds:

- *Physical:* A "normal" physical network interface that exists in hardware in the Test Agent.
- *VLAN:* A Virtual LAN interface which has a parent interface. A VLAN interface is created on a parent interface as described *here* (page 175).
- *Bridge:* The "virtual interface" part of a network bridge. The bridge connects a set of "child" (bridged) interfaces. A bridged interface cannot be the management interface, nor can it be used for NTP. Bridges are created as described *here* (page 176).
- *Mobile:* A special interface that uses a USB 4G dongle. Mobile interfaces have some special settings specific to the mobile network, but few other network-related settings since these are not exposed. See *this page* (page 178).
- *Wi-Fi:* A special interface that uses a Wi-Fi card. Wi-Fi interfaces have some settings specific to the Wi-Fi network. See *this page* (page 182).

In a virtual Test Agent running in a hypervisor, all interfaces are of course virtual. The assignment of interface names for a virtual Test Agent is described *here* (page 105).

4.3.2.2 Naming of Test Agent interfaces

For a Test Agent Application in an environment with multiple namespaces, interface names follow the syntax `<namespace>/<interface>`. For example, "ns1/eth0" means "interface eth0 in namespace ns1". Read more about Test Agent Application namespace awareness *here* (page 240).

In other situations, interface names consist solely of an interface designation such as "eth0" or "wlan0".

4.3.2.3 Status of Test Agent interfaces

The status of each interface is indicated by a colored dot to the left of the interface name.

	*Green:* Online and ready, currently not in use
	*Yellow:* Online and currently in use
	*Gray:* No Ethernet link
	*Red:* Offline

4.3.2.4 Properties of Test Agent interfaces

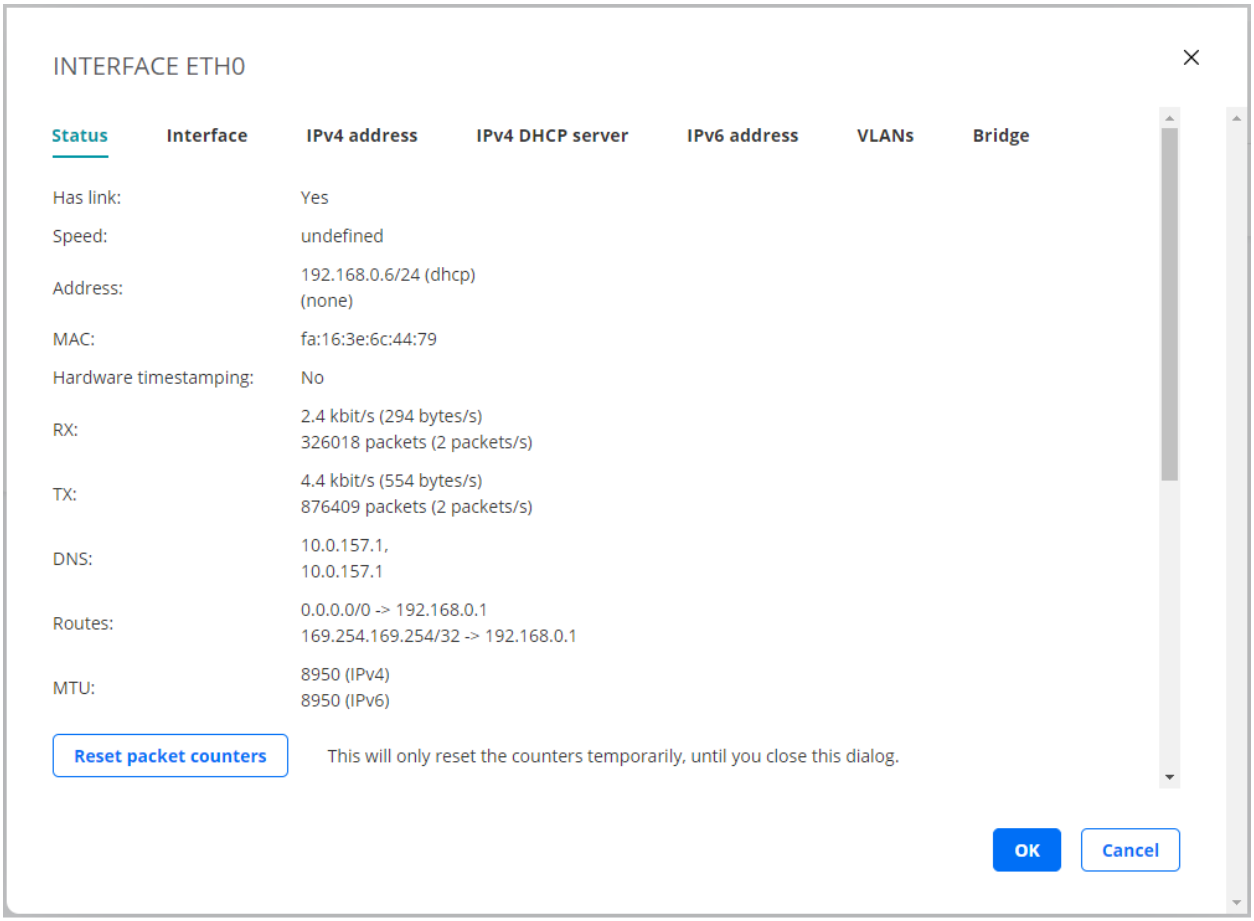
Provided that the Test Agent is not offline or shared, you can click each of its interfaces (for example, "eth0") to display the configuration options for that interface. These include IP/MAC addresses, DHCP, VLANs, and bridges. A new tabbed dialog appears for the interface; its contents are gone through in the subsections that follow.

Note: If the Test Agent has a mobile interface, it is named "usb0". This type of interface has a different set of configuration options, including mobile-specific ones. See *Mobile interface configuration details* (page 178).

For an introduction to using mobile interfaces, see also the *page on mobile network measurements* (page 355).

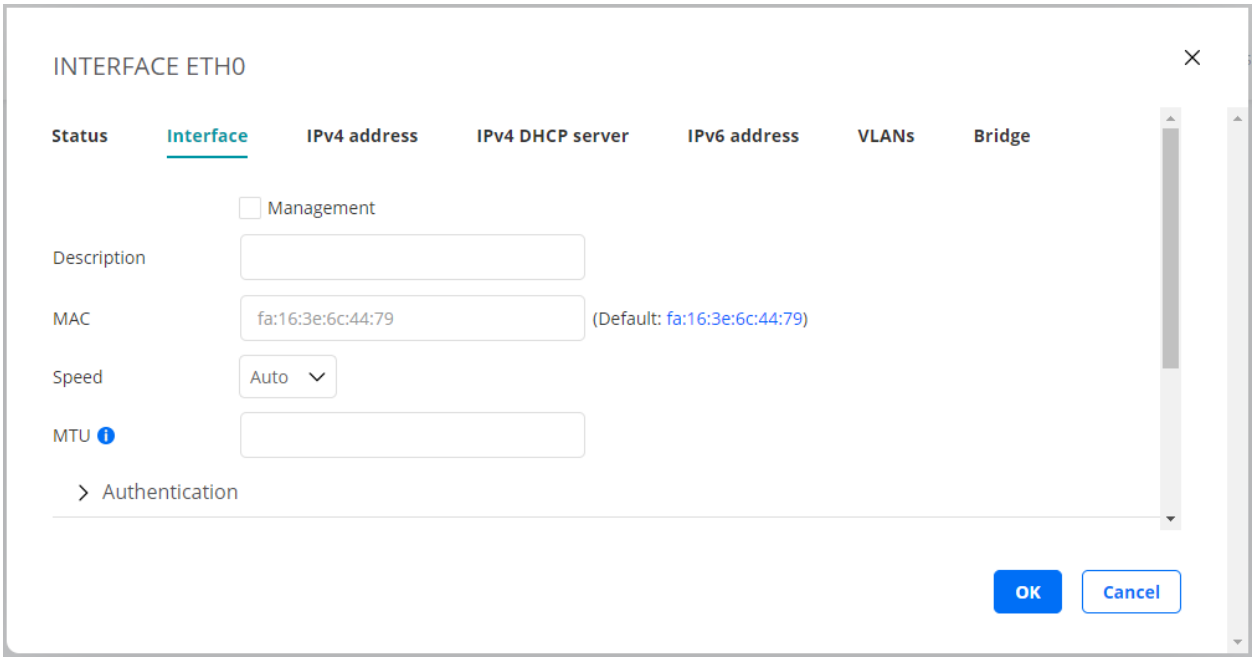
Status tab

The Status tab of the interface configuration dialog shows read-only information: link status, interface, IP/MAC address, RX/TX data rates, DNS server IP address, and the routes for the interface. You can reset packet counters by clicking the Reset packet counters button.



Interface tab

On this tab you can configure the link properties of the interface.



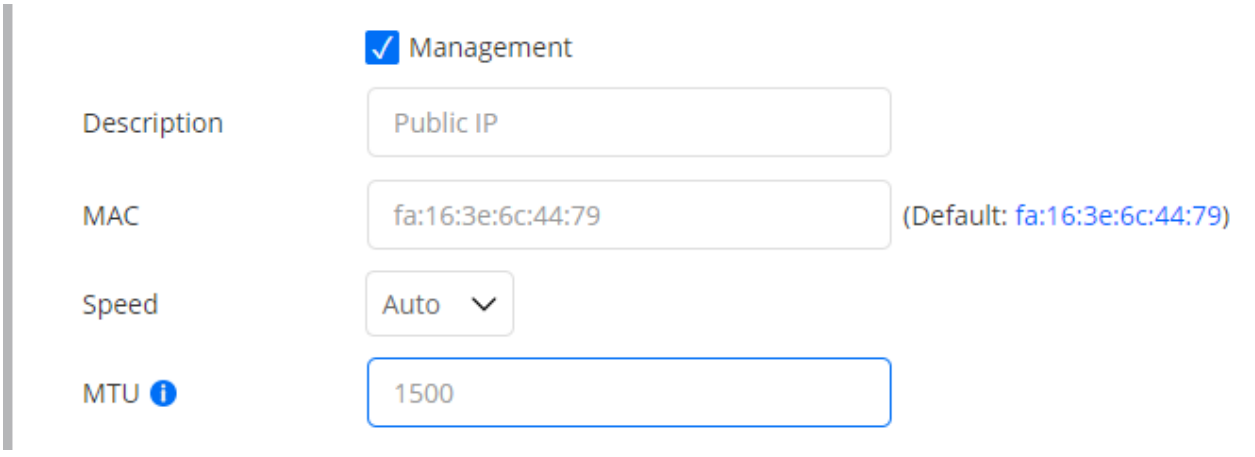
Management: Check the box if this interface is to be used as Routing Active Testing internal management interface for connection to the Routing Active Testing server. Management can be done over either IPv4 or IPv6. By default, “eth0” is selected as management interface. Note: Be careful when changing the management interface, as the Routing Active Testing server might lose contact with the Test Agent if you accidentally configure this interface incorrectly.

Description: Plain-text description of the interface.

MAC: Changeable MAC address. In the screenshot above, the MAC is changed to mimic a set-top box.

Speed: The link speed and duplex settings of the interface. Default: Auto.

MTU: The Maximum Transmission Unit size on the interface. Range: 1280 … 9216 bytes. Note: The MTU set here is only an initial value. Leaving this field empty means that the MTU will not be set. It may be changed at any time to a value specified in a DHCP lease or a Router Advertisement (RA).



Authentication

Here you can optionally set parameters for 802.1X authentication.

- Authentication type: The 802.1X authentication type used in the network. One of:
  - *EAP-TLS:* Extensible Authentication Protocol - Transport Layer Security (EAP-TLS), an IETF open standard that uses the Transport Layer Security (TLS) protocol. Defined in ► IETF RFC 5216. EAP-TLS is the original, standard wireless LAN EAP authentication protocol.
  - *EAP-TTLS/MSCHAPv2:* EAP Tunneled Transport Layer Security (EAP-TTLS), an EAP protocol that extends TLS. Defined in ► IETF RFC 5281.
  - *PEAPv0/EAP-MSCHAPv2:* Protected EAP, a protocol that encapsulates EAP within a potentially encrypted and authenticated Transport Layer Security (TLS) tunnel.
  - *None:* Open network, no security.

The parameters vary depending on the choice made under Authentication type. The following parameters occur:

- Anonymous identity: Used in EAP-TTLS and PEAP to allow the authenticator to choose the correct authentication server to process the credentials.
- Identity: The client's identity in the network.
- Password: The client's password for the network.
- CA certificate: Certificate Authority certificate, PEM encoded. The input takes the following form:

```
-----BEGIN CERTIFICATE-----
(base 64 encoded DER)
-----END CERTIFICATE-----
```

- Client certificate: Client certificate, PEM encoded. The input takes the following form:

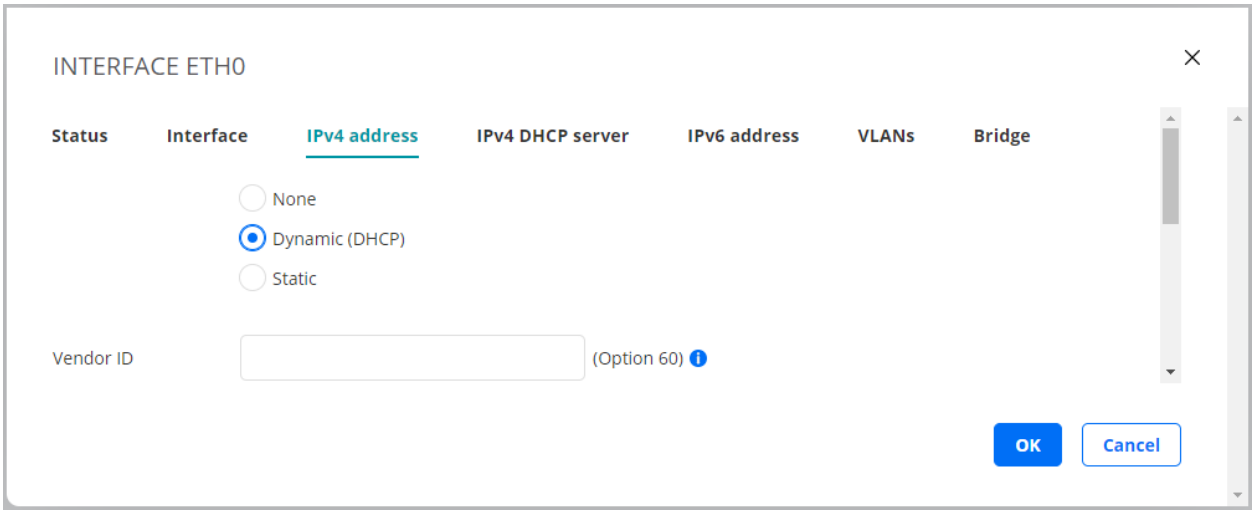
```
-----BEGIN CERTIFICATE-----
(base 64 encoded DER)
-----END CERTIFICATE-----
```

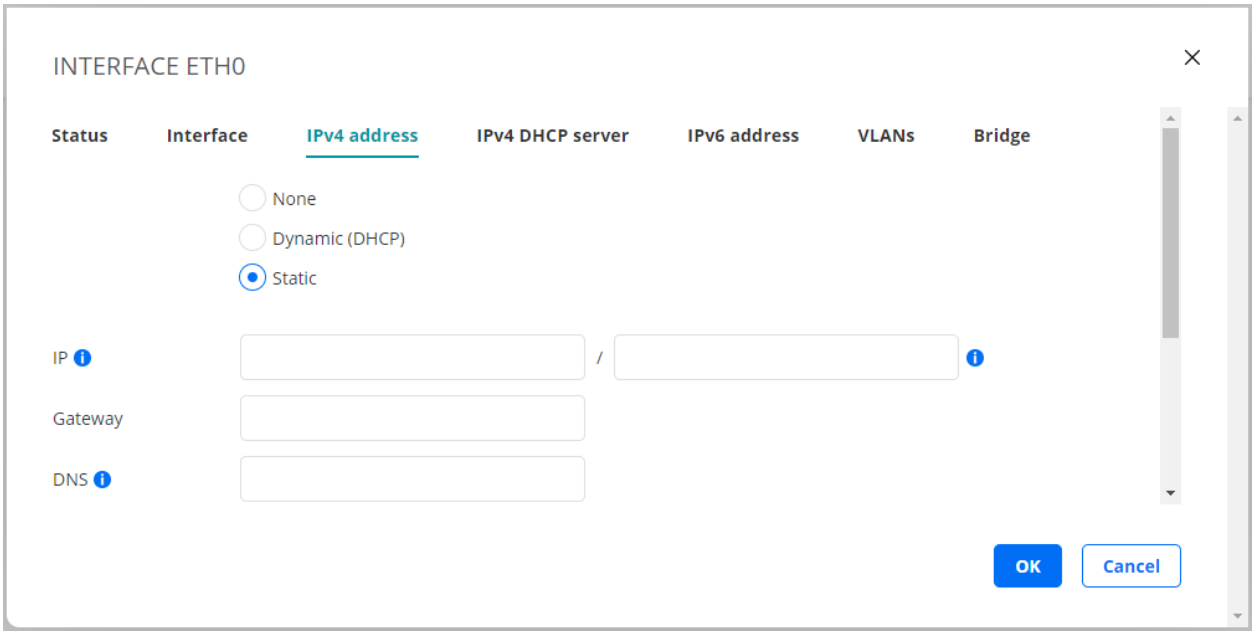
- Private key: Unencrypted key. The input takes the following form:

```
-----BEGIN PRIVATE KEY-----
(base 64 encoded DER)
-----END PRIVATE KEY-----
```

IPv4 address tab

On this tab you configure IPv4 address assignment for the interface. The options are None, Dynamic (DHCP), and Static.





Dynamic (DHCP)

An IPv4 address is assigned by a DHCP server.

Vendor ID (Option 60): This option is used to optionally identify the vendor type and configuration of a DHCP client.

Vendors may choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration.

See ► IETF RFC 2132 for details.

Static

A static IPv4 address is assigned to the Test Agent.

IP: The IP address is specified in the IP/prefix length format. Example: “192.168.1.5/24”.

IPv4 addresses can be specified using two different notations:

- `192.168.0.5/255.255.255.0` (Netmask)
- `192.168.0.5/24` (Prefix length)

For consistency among all IP addresses, Routing Active Testing expects the prefix length notation. For example, prefix length /24 is the same as 255.255.255.0. Full information is given in the following conversion table:

Netmask	Prefix length	Netmask	Prefix length
255.255.255.255	/32	255.254.0.0	/15
255.255.255.254	/31	255.252.0.0	/14
255.255.255.252	/30	255.248.0.0	/13
255.255.255.248	/29	255.240.0.0	/12
255.255.255.240	/28	255.224.0.0	/11
255.255.255.224	/27	255.192.0.0	/10
255.255.255.192	/26	255.128.0.0	/9
255.255.255.128	/25	255.0.0.0	/8 (Class A)
255.255.255.0	/24 (Class C)	254.0.0.0	/7
255.255.254.0	/23	252.0.0.0	/6
255.255.252.0	/22	248.0.0.0	/5
255.255.248.0	/21	240.0.0.0	/4
255.255.240.0	/20	224.0.0.0	/3
255.255.224.0	/19	192.0.0.0	/2
255.255.192.0	/18	128.0.0.0	/1
255.255.128.0	/17	0.0.0.0	/0
255.255.0.0	/16 (Class B)		

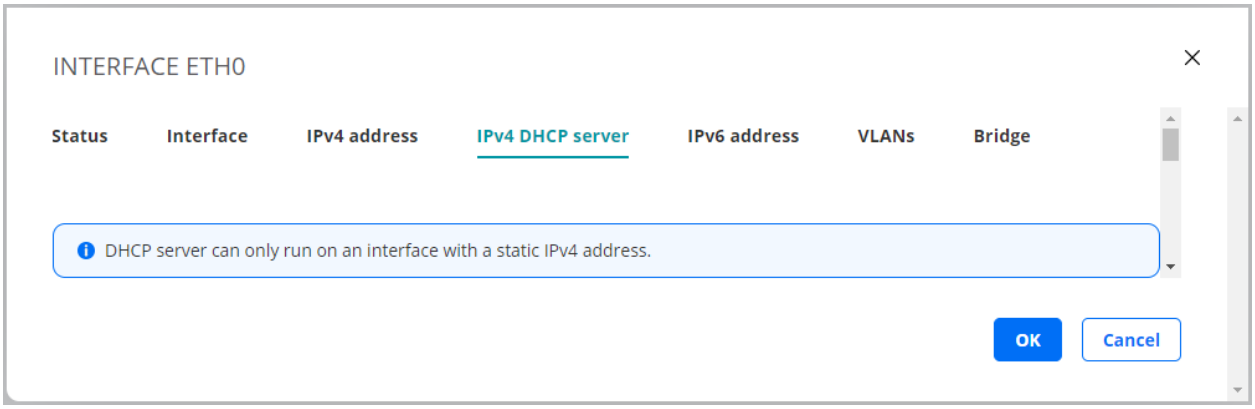
Gateway: The IP address of the default gateway in your network.

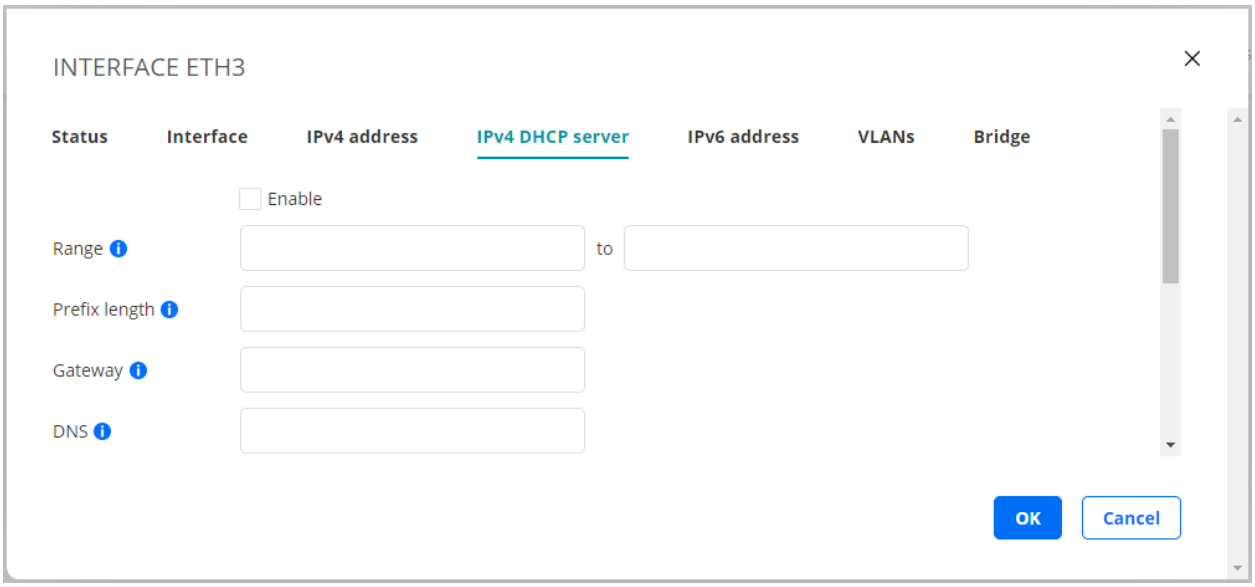
DNS: DNS server address(es). Multiple servers can be specified using a comma-separated list. Example: “192.168.1.1, 10.0.0.1”.

At least one DNS server is required for the Test Agent management interface as the Test Agent resolves the Routing Active Testing server host name (for the public cloud server, https://login.paa.juniper.net) in order to set up the encrypted management connection to the Routing Active Testing server. For an on-premise Routing Active Testing server with a static IP address, no DNS lookup is of course needed. For other interfaces, a DNS server is required only for certain tests such as HTTP, DNS and Ping, again if the target address is entered as a host name.

IPv4 DHCP server tab

On this tab you can activate a DHCP server on the interface. To use the function you must have the interface configured with a static IPv4 address.





Enable: Check this box to enable the DHCP server.

Range: The IP address interval from which the DHCP server will assign addresses to clients.

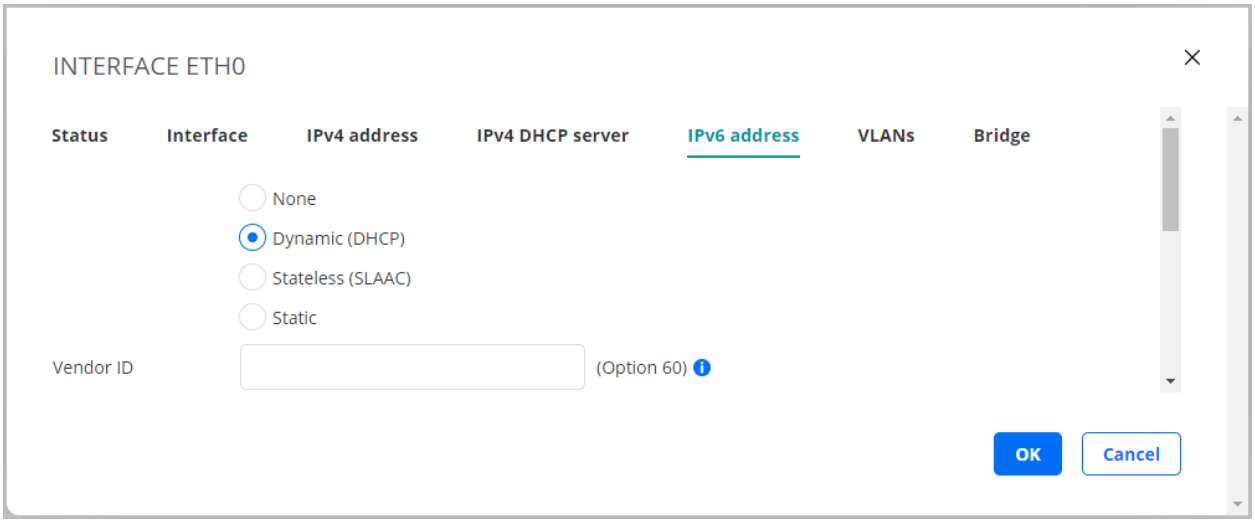
Prefix length: The length of the address prefix. See the IP address description above for more information about the prefix length format.

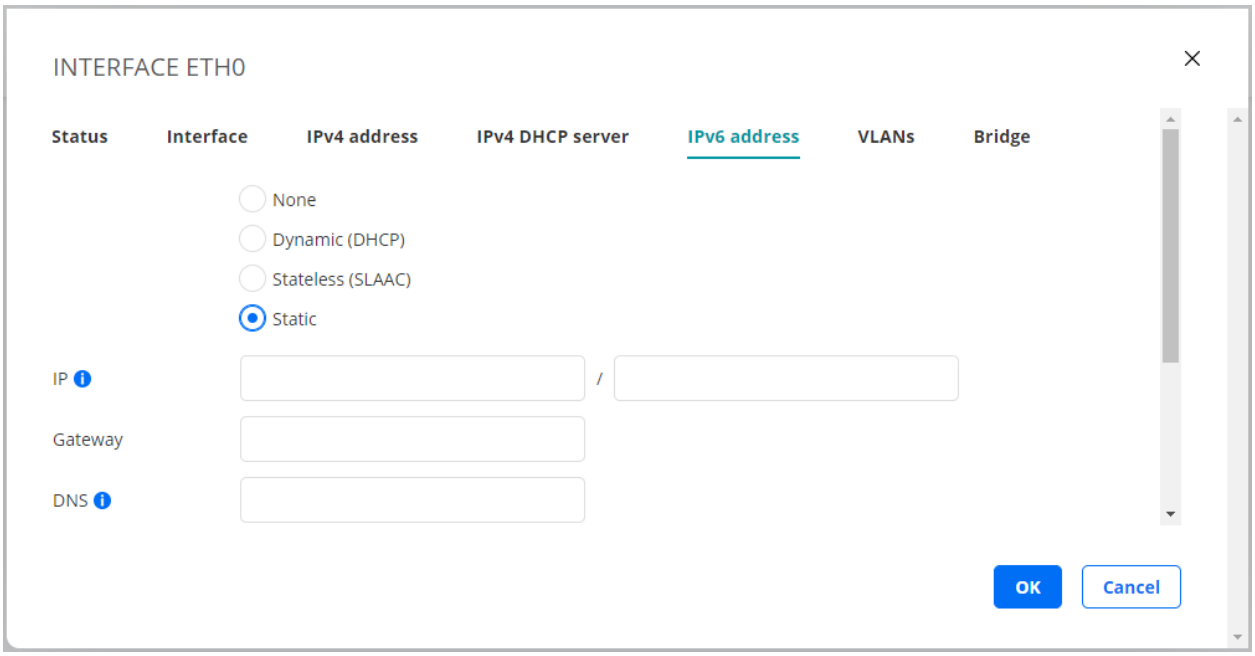
Gateway: The default gateway to send to clients.

DNS: The DNS server to send to clients.

IPv6 address tab

This tab is where you configure IPv6 address assignment for the interface. The options are None, Dynamic (DHCP), Stateless (SLAAC), and Static.





Dynamic (DHCP)

An IPv6 address is assigned by a DHCP server.

Vendor ID (Option 60): This option is used to optionally identify the vendor type and configuration of a DHCP client.

Vendors may choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration.

See ► IETF RFC 2132 for details.

Stateless (SLAAC)

The Test Agent self-assigns an IPv6 address by means of Stateless Address Autoconfiguration (SLAAC).

SLAAC is described in ► IETF RFC 4862.

DNS: DNS server address(es). Multiple servers can be specified using a comma-separated list.

Static

A static IPv6 address is assigned to the Test Agent.

IP: The IPv6 address is specified in the IP/prefix length format. The format used by |paa-product| for IPv6 addresses is "0123::4567:89ab:cdef:0123/64".

For IPv6 addresses only the prefix length notation is available, since typing 128-bit netmasks would be very tedious. You thus need to type the IPv6 address in full or collapsed form followed by the prefix length.

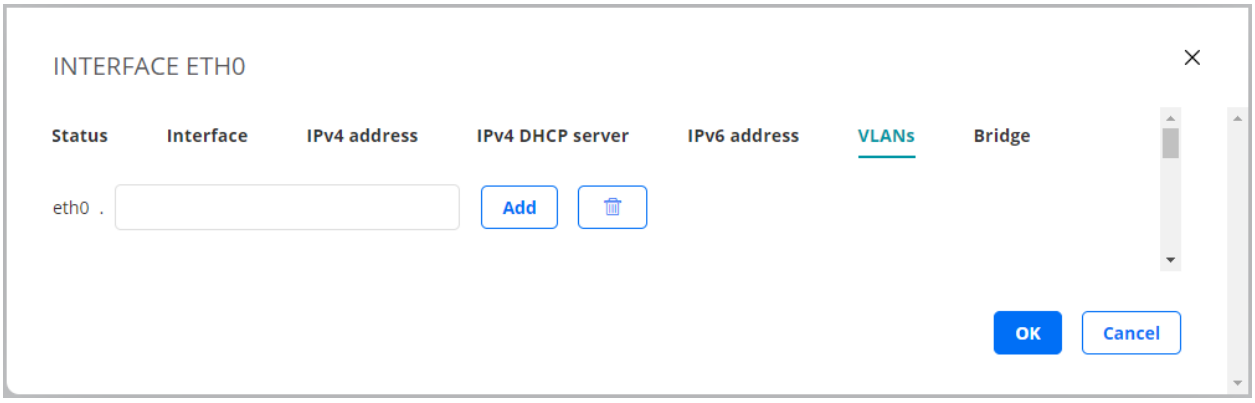
Note that the deprecated *site-local* IPv6 addresses cannot be used. These have the unicast prefix "fec0::/10". See ► IETF RFC 3879 for further information.

Gateway: The IPv6 address of the default gateway in your network.

DNS: DNS server address(es). Multiple servers can be specified using a comma-separated list.

VLANs tab

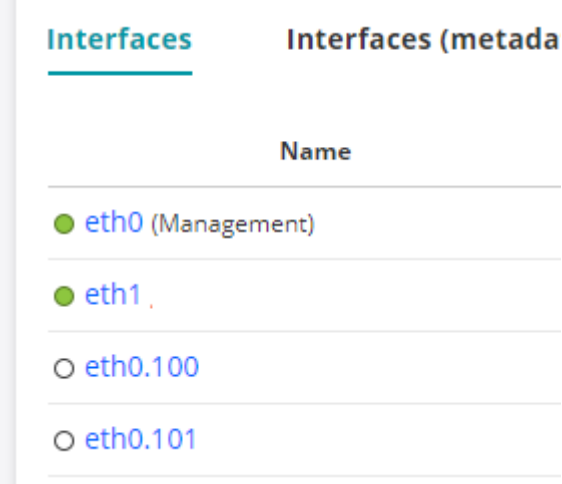
On this tab you can create VLANs on an interface. Test Agents support VLAN according to IEEE 802.1Q. The screenshot shows an example where VLANs can be added on the physical interface "eth0".



If a bridge exists, you can add VLANs on top of that bridge. Type the VLAN identifier in the box, then click the Add button.



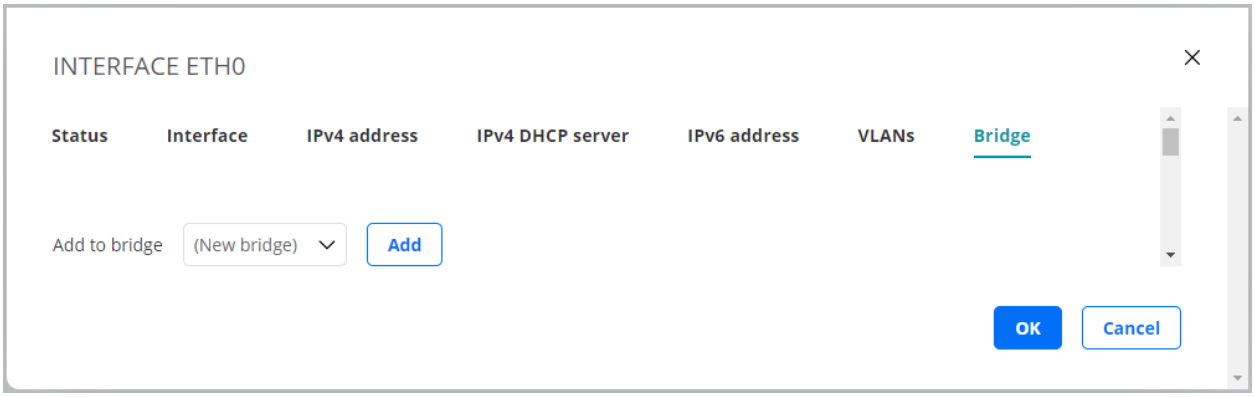
On the top-level Interfaces tab, VLANs will appear as shown below:



The MAC address of a VLAN is by default inherited from its parent interface. You can manually set a different MAC address for the VLAN on the *Interface* (page 168) tab of its properties dialog. Click the VLAN link to access that dialog.

Bridge tab

On this tab you can add a bridge to and remove a bridge from an interface.



To a bridge one or more interfaces can be connected, as shown in the screenshot below. Use the left and right arrow buttons to configure which interfaces should be joined by the bridge.



On the top-level Interfaces tab, a bridge will appear as shown in the following screenshot. Here, the bridge is called “br0”, and it bridges the interfaces “eth1” and “eth2”. VLANs have also been added to this bridge.



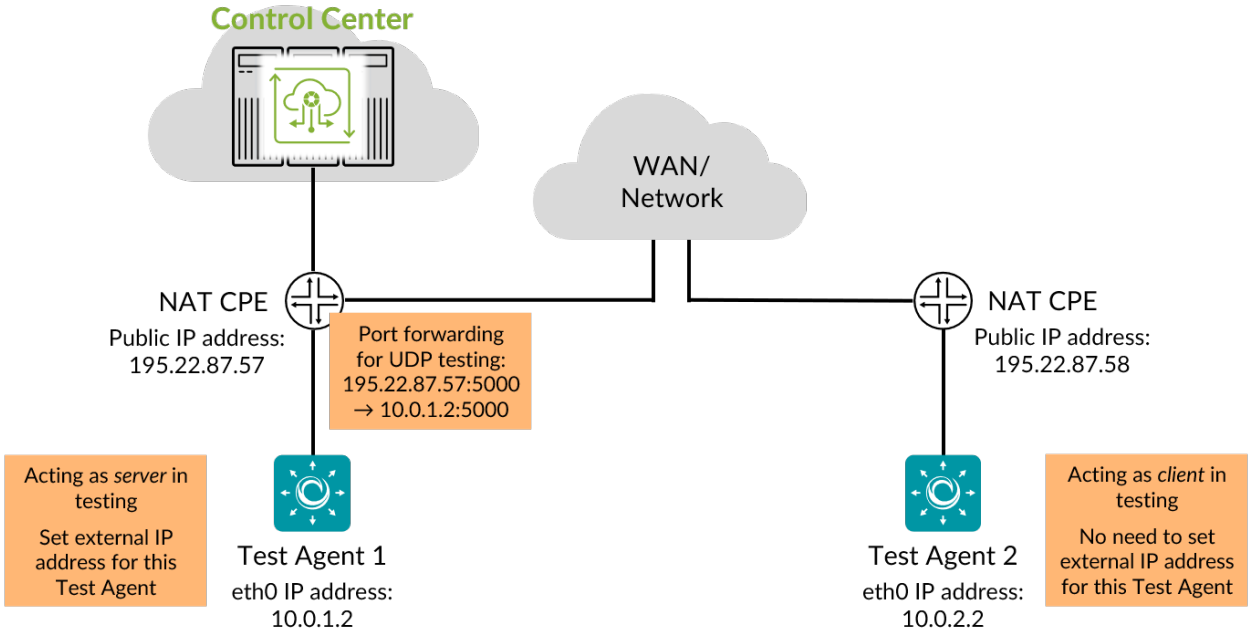
4.3.3 Test Agent interface metadata configuration

The Interfaces (metadata) tab of the Test Agent configuration dialog lets you configure an external IP address for a Test Agent interface.

This allows a Test Agent to act as a server in measurements even if it is behind a NAT router. (The party initiating the measurement is termed “client”, whereas the other party is called “server”.) Other Test Agents will then use the public address of the router for communicating with the server Test Agent behind NAT. A prerequisite for this to work is that the necessary port forwarding has been set up between the NAT router and the server Test Agent. The port forwarding setup needs to be handled outside Routing Active Testing, and the details are not dealt with here.

For a Test Agent that is not behind NAT, this setting is not applicable. Consequently, you need to use this setting only if both client and server are behind NAT (otherwise, simply pick the Test Agent that is behind NAT to act as client).

An example of port forwarding and use of an external IP address is shown in the diagram below. In this example, UDP is tested on port 5000. Other tasks may use different ports.



An external IP address can be set for both IPv4 and IPv6. In each case, the address assignment can be either automatic or manual.

- *Automatic:* The IP address used by the management interface when connecting to Control Center will be assigned as external IP address. This option can only be used for the management interface.
- *Manual:* Here you enter an external IP address manually in the External IP field.

Once configured, the external IP address will appear as a separate interface item when *configuring tasks and monitors* (page 255), with "(external)" as postfix.

Note: The automatic address assignment only works for the IP version currently in use on the management interface. For example, if management (on eth0) is currently done over IPv4, and you set External IPv6 address to Automatic on eth0, the address assignment will fail if you try to run some task over IPv6 on eth0. The workaround is to set an IPv6 address manually for the eth0 interface.

4.3.3.1 Note on defunct "Use public address" option

Previous versions of Routing Active Testing (Netrounds) had an option Use public address in the Test Agent interface configuration dialog (Interfaces tab > IPv4 address tab). Selecting this option was largely equivalent to configuring an external IP address as "Automatic" on the management interface in the current version. However, the old and new features differ in the following ways:

- The new feature works for both Test Agent Appliance and Test Agent Application, while the old one was restricted to the Appliance.
- The new feature is (independently) applicable to all interfaces, whereas the old one was limited to the management interface.
- The new feature allows you to choose between private and external IP address for each measurement task, unlike the old feature which applied globally to anything the interface was used for.
- The new feature allows you to manually specify an external IP address, an option which was previously absent.
- The new feature applies to both IPv4 and IPv6, while the old one was for IPv4 only.

4.3.4 Mobile interface configuration details

The Mobile support in Routing Active Testing enables you to run tests and monitoring sessions over a mobile interface using a preinstalled mPCIe 4G modem in a Test Agent provided by Routing Active Testing.

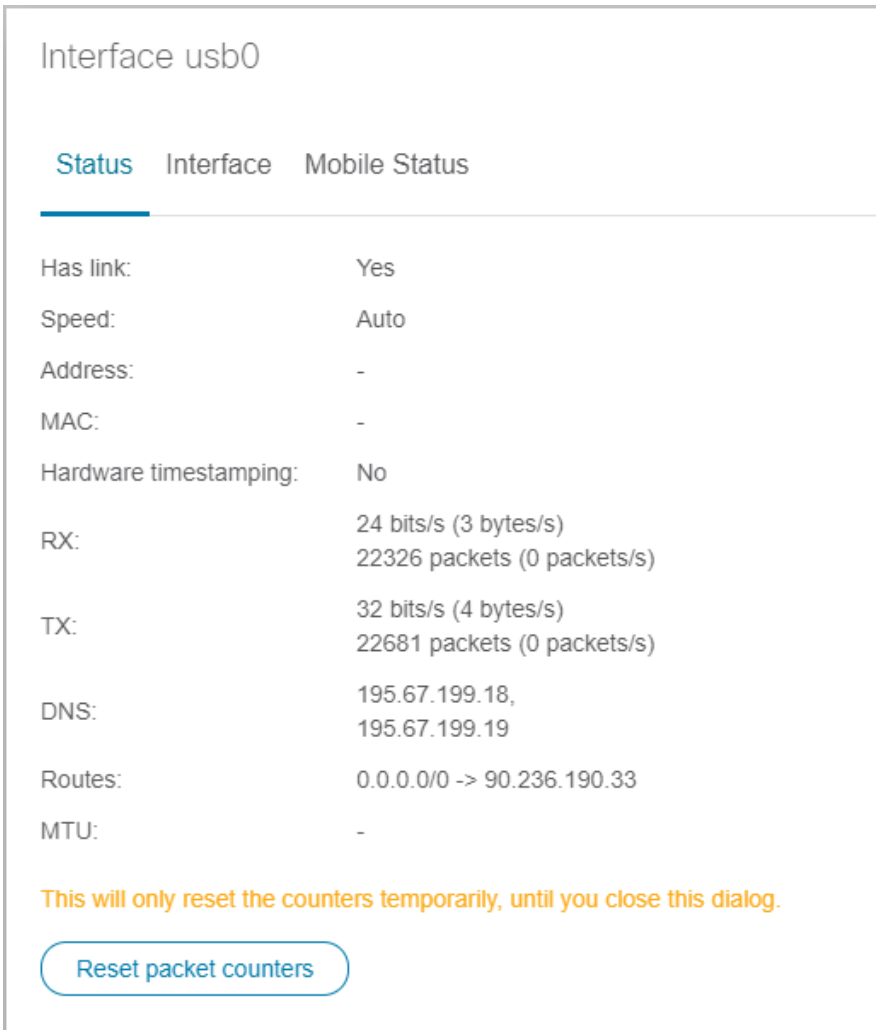
Besides running normal data traffic over this interface, it is also possible to log mobile network parameters such as signal strength and cell network location information.

For details on normal network interface configuration, see *Test Agent interface configuration* (page 178).

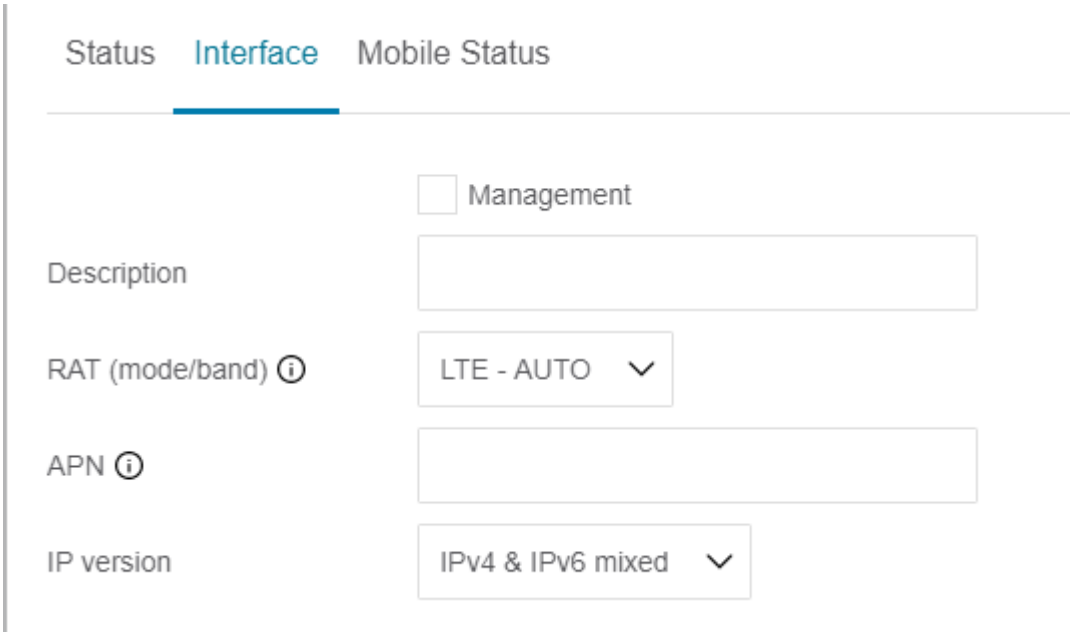
4.3.4.1 Mobile interface dialog

- Click Test Agents on the main menu.
- Click the Test Agent of interest.
- Click the Test Agent's mobile network interface ("usb0").

In the dialog that appears, the Status tab shows read-only information. You can reset the counting of packets by clicking the Reset packet counters button. Note that this will reset the packet counters only temporarily, until you close the dialog.

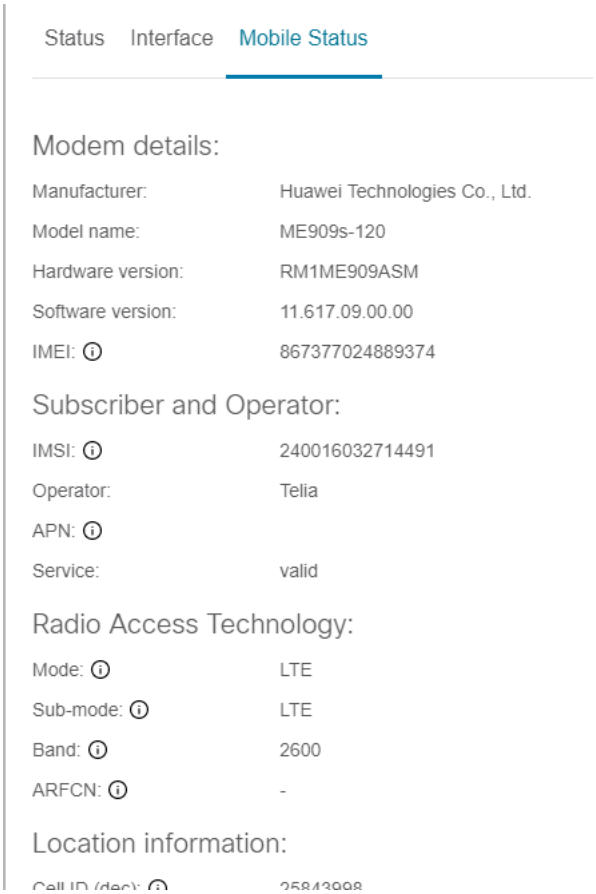


The Interface tab looks as follows:



- Management: Check this box if the mobile interface is to be used for Test Agent management. Compare the page on *Test Agent interface configuration* (page 166) in general.
- RAT (mode/band): Here you select what radio access technology (RAT) and frequency band the Test Agent should preferably use. “AUTO” means that the choice of RAT and/or band will be done automatically.
- APN: Access Point Name for connecting to the mobile network.

The Mobile Status tab, finally, also shows read-only information, including some mobile network measurements (some of which are visible in the screenshot). See the section below for details on the terminology used.



4.3.4.2 Terminology

Here is a list of terms used in conjunction with the Mobile interface.

Some of the terms are specific to a certain type of network access technology such as GSM, WCDMA or LTE.

- IMSI: International Mobile Subscriber Identity.
- IMEI: International Mobile Station Equipment.
- APN: Access Point Name.
- RAT mode: Radio Access Technology mode.
- RAT sub-mode: Radio Access Technology sub-mode.
- ARFCN: Absolute Radio-Frequency Channel Number.
- CI: Cell ID as a decimal number.
- eNB+CI [LTE only]: eNB + Cell ID [LTE Only]. For LTE the Cell ID, or more correctly E-UTRAN Cell Identifier, is composed of two values, eNB ID (20 bits) and Cell ID (8 bits).
- RNCID + CI [WCDMA only]: RNCID + Cell ID [WCDMA only]. In WCDMA the Cell ID, or more correctly UTRAN Cell Identifier, is composed of two values, RNC ID (12 bits) and Cell ID (16 bits).
- RAC: Routing Area Code [GSM/WCDMA only].
- LAC: Location Area Code (LAC) [GSM/WCDMA only].

- TAC: Tracking Area Code (TAC) [LTE only].
- PLMN: Public Land Mobile Network.
- PCI: Physical Cell ID [LTE only].
- BSIC: Base Station Identity Code.
- SC: Scrambling Code.
- RSSI: Received Signal Strength Indication.
- RSRP: Reference Signal Received Power.
- RSCP: Received Signal Code Power.
- RSRQ: Reference Signal Received Quality.
- Ec/Io: Chip energy interfering co-channel.
- SINR: Signal to Interference + Noise Ratio.

4.3.4.3 Related topics

- Use the *Mobile logger* (page 355) task to collect mobile network measurements.
- Use the *Mobile switcher* (page 357) task to change mobile interface parameters.

4.3.5 Wi-Fi interface configuration details

The Wi-Fi support in Routing Active Testing enables you to run tests and monitors over a Wi-Fi interface. For details on supported hardware, see *here* (page 351).

Besides running normal data traffic over this interface, it is also possible to log Wi-Fi network performance data and parameters, such as Tx/Rx data rates and modulation coding scheme usage.

You set up the Wi-Fi network connection in the Wi-Fi interface configuration dialog. (Wi-Fi interface configuration cannot be done via the local console.) During execution of a test or monitor you can switch to a different Wi-Fi network (or a different access point within the same network) using the *Wi-Fi switcher* (page 354) task. This action will overwrite the existing Wi-Fi configuration.

Configuring VLANs on top of a Wi-Fi interface is not supported.

Using a Test Agent to bridge an Ethernet network to Wi-Fi is not supported.

For details on normal network interface configuration, see *Test Agent interface configuration* (page 166).

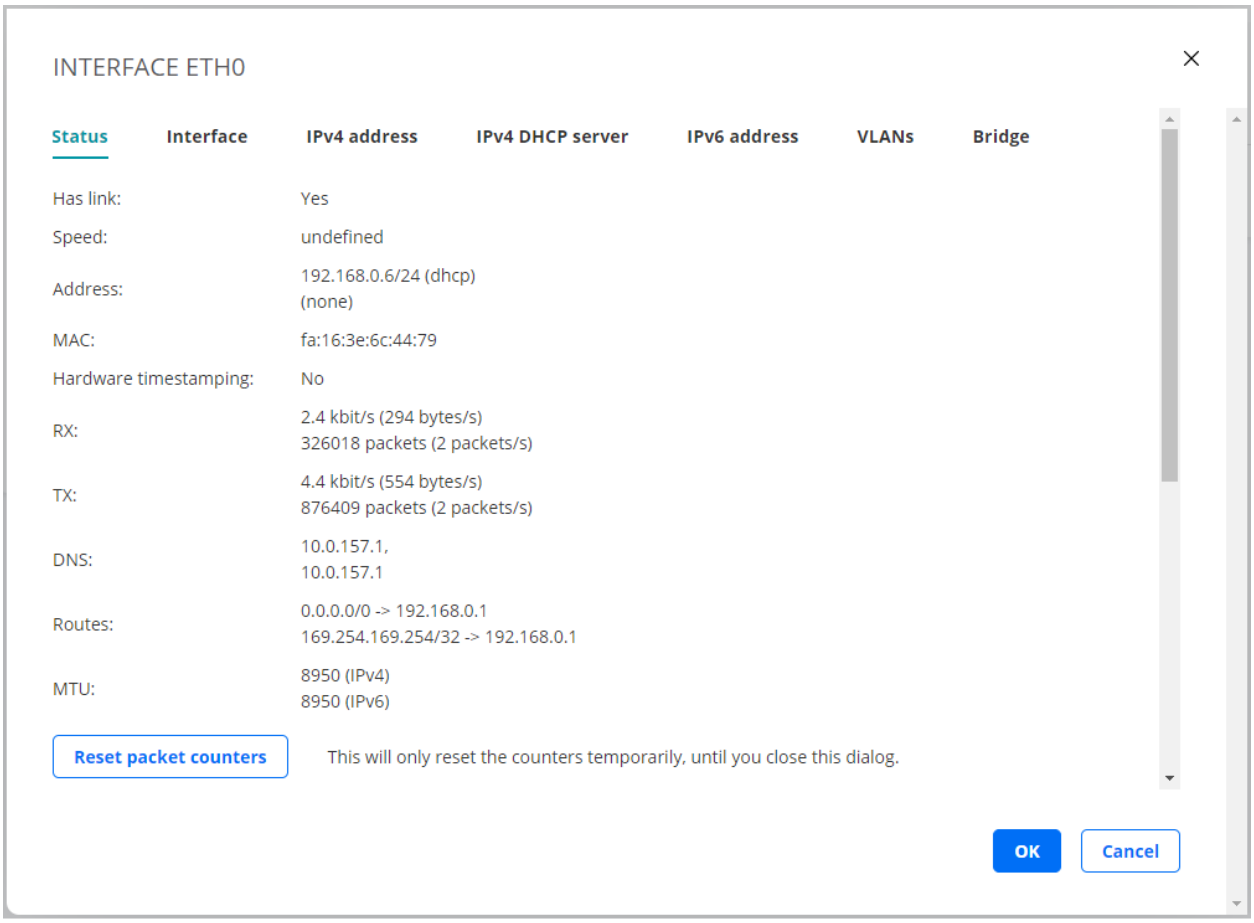
4.3.5.1 Wi-Fi interface dialog

- Click Test Agents on the main menu.
- Click the Test Agent of interest.
- Click the Test Agent's Wi-Fi network interface ("wlan0").

The dialog that appears contains tabs as detailed below.

Status tab

This tab shows read-only information. You can reset the counting of packets by clicking the Reset packet counters button. Note that this will reset the packet counters only temporarily, until you close the dialog.

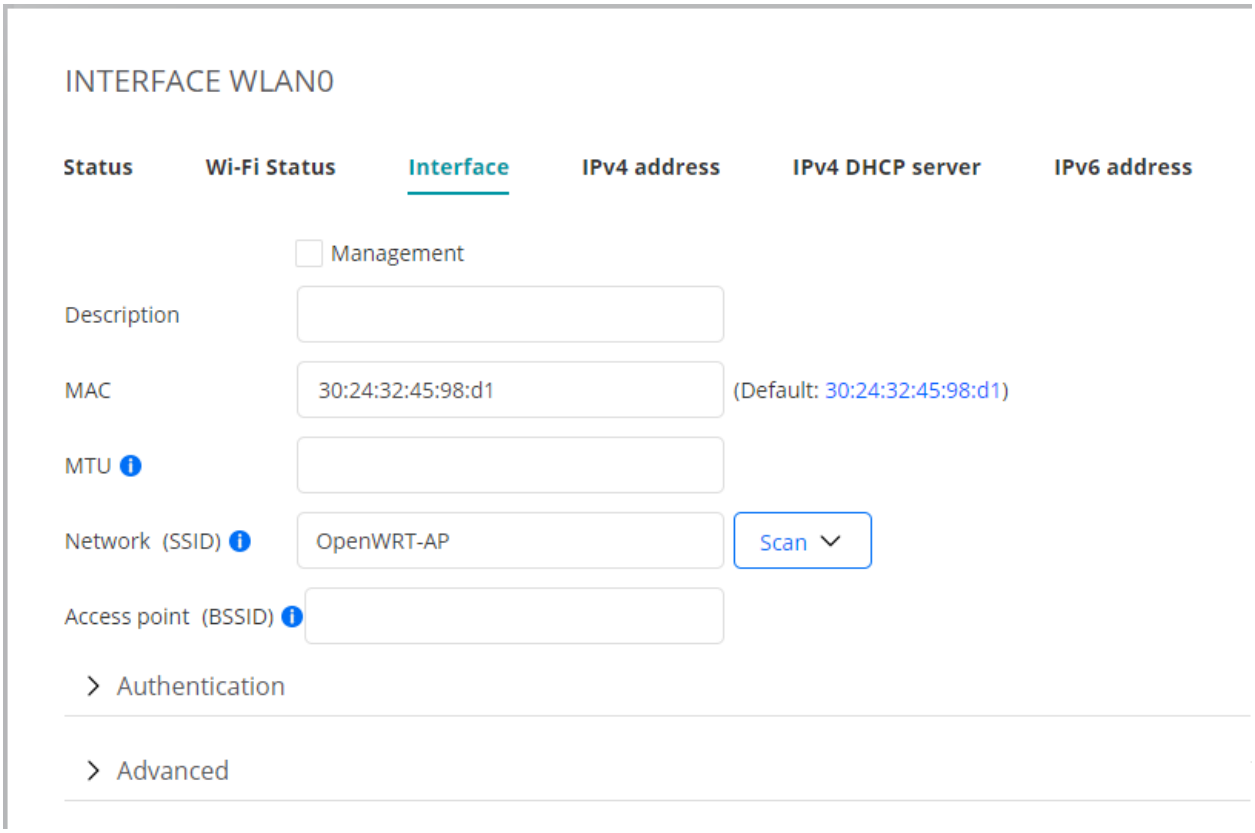


Wi-Fi Status tab

This tab, too, is read-only and shows a variety of data on Wi-Fi configuration and performance.


INTERFACE WLAN0
Status
Wi-Fi Status
Interface
IPv4 address
IPv4 DHCP server
IPv6 address
General:
BSSID: 00:0d:b9:ce:00:00
Channel: 40
TX power: 22 dBm
Inactive time: 0 ms
TX/RX:
TX retries: 756
TX failed: 0
Beacon loss: -
Beacon RX: 0
RX drop misc: 32

Interface tab



- Management: Select this box if the Wi-Fi interface is to be used for Test Agent management. Compare the page on *Test Agent interface configuration* (page 166) in general.
- Description: Free-text field for describing the Wi-Fi interface.
- MAC: Changeable MAC address for the Wi-Fi interface.
- MTU: Maximum Transmission Unit (in bytes) on the Wi-Fi interface.
- Scan for networks: Clicking the Scan button starts a Wi-Fi network scan, which populates this list with the available Wi-Fi networks found. This is the same scan performed in the *Wi-Fi scan* (page 352) task type. Click a row (network) in the list to select it. Its parameters are then copied into the fields that follow below.
  - If you select the option Select access point, access points in Wi-Fi networks will be listed individually in the list (one entry is given for each access point and corresponding BSSID). This is useful if you want to connect to a specific access point instead of allowing roaming between all access points in the network. If you clear the Select access point box, only one list entry is displayed for each Wi-Fi network (SSID only).
- Network (SSID): Service Set Identifier (that is, the name) of the Wi-Fi network.
- Access point (BSSID): Basic Service Set Identifier of the access point. This is typically the access point's MAC address. A network has an SSID that all access points share, and each access point has a unique BSSID.
- Authentication type: The 802.1X authentication type used in the Wi-Fi network. One of:
  - *EAP-TLS:* Extensible Authentication Protocol - Transport Layer Security (EAP-TLS), defined in ► IETF RFC 5216, an IETF open standard that uses the Transport Layer Security (TLS) protocol. EAP-TLS is the original, standard wireless LAN EAP authentication protocol.

- *EAP-TTLS/MSCHAPv2:* EAP Tunneled Transport Layer Security (EAP-TTLS), an EAP protocol that extends TLS. Defined in ► IETF RFC 5281.
- *PEAPv0/EAP-MSCHAPv2:* Protected EAP, a protocol that encapsulates EAP within a potentially encrypted and authenticated Transport Layer Security (TLS) tunnel.
- *WPA Personal:* Also referred to as WPA-PSK (pre-shared key) mode; designed for home and small office networks and does not require an authentication server.
- *None:* Open network, no security.

Authentication

The parameters vary depending on the choice made under Authentication type. The following parameters occur:

- Cipher:
  - AUTO: The cipher type is automatically selected by the Wi-Fi card.
  - CCMP: Counter Mode Cipher Block Chaining Message Authentication Code (Counter Mode CBC-MAC) Protocol.
  - TKIP: Temporal Key Integrity Protocol.
- Anonymous identity: Used in EAP-TTLS and PEAP to allow the authenticator to choose the correct authentication server to process the credentials.
- Identity: The client's identity in the Wi-Fi network.
- Password: The client's password for the Wi-Fi network.
- CA certificate: Certificate Authority certificate, PEM encoded. The input takes the following form:

```
-----BEGIN CERTIFICATE-----
(base 64 encoded DER)
-----END CERTIFICATE-----
```

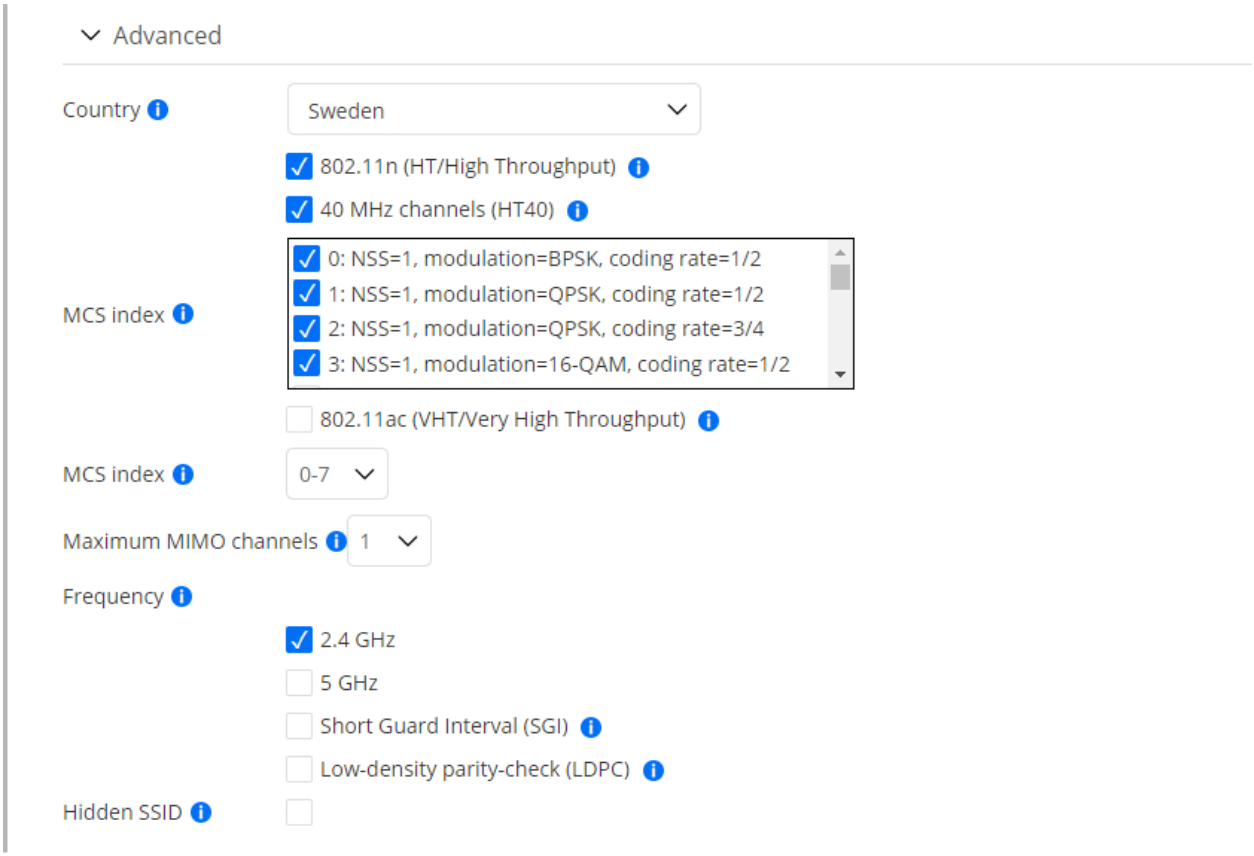
- Client certificate: Client certificate, PEM encoded. The input takes the following form:

```
-----BEGIN CERTIFICATE-----
(base 64 encoded DER)
-----END CERTIFICATE-----
```

- Private key: Unencrypted key. The input takes the following form:

```
-----BEGIN PRIVATE KEY-----
(base 64 encoded DER)
-----END PRIVATE KEY-----
```

Advanced



- Country: Regulatory country in which the Wi-Fi network resides. You need to specify this in order to ensure that you do not violate any regulatory requirements, as allowed frequencies, output power, and channel width all vary between countries. The Wi-Fi card and driver in the Test Agent will handle this automatically provided that your country is correctly entered here.

Among the Wi-Fi standards, IEEE 802.11g is always supported. The other standards can optionally be disabled in the settings below.

- 802.11n (HT/High Throughput): Enable 802.11n high-throughput amendment, increasing throughput from 54 Mbit/s (802.11ag) to theoretically 600 Mbit/s.
  - 40 MHz channels: Select this to allow 40 MHz channels. If this is not selected, only 20 MHz channels are allowed.
  - MCS index: Select the allowed subset of Modulation and Coding Scheme indexes. The MCS index determines the number of spatial streams (“NSS”), the type of modulation, and the coding rate (proportion of the data stream that is made up of non-redundant data).
- 802.11ac (VHT/Very High Throughput): Enable 802.11ac Very High Throughput amendment, enabling theoretical Gigabit speeds.
  - MCS index: Modulation and Coding Scheme index. Range: 0-7 … 0-9. Default: 0-9.
  - Maximum MIMO channels: The maximum number of MIMO (Multiple Input Multiple Output) spatial streams. Range: 1 … 4. Default: 4.
- Frequency: Select which Wi-Fi frequency bands to allow. By default, both are allowed.
  - 2.4 GHz: Allow the 2.4 GHz frequency band.

  - 5 GHz: Allow the 5 GHz frequency band.

- Short Guard Interval (SGI): The guard interval is intended to prevent interference between information symbols due to the multipath effect. Such interference will degrade the Wi-Fi signal. If the multipath effect in the environment is not too serious, the short guard interval can be used to improve throughput. Select this box to use the short guard interval 400 ns instead of the default 800 ns.
- Low Density Parity Check (LDPC): Select this to use the low density parity check (LDPC) error-correcting code, which can provide a performance gain.
- Hidden SSID: Select this to use a Wi-Fi probing frame that enables connecting to networks with hidden SSID. This can make connecting to a network with visible SSID considerably slower.

IPv4 Address tab

This tab is the same as for a regular, wired interface; it is covered *here* (page 170).

IPv6 Address tab

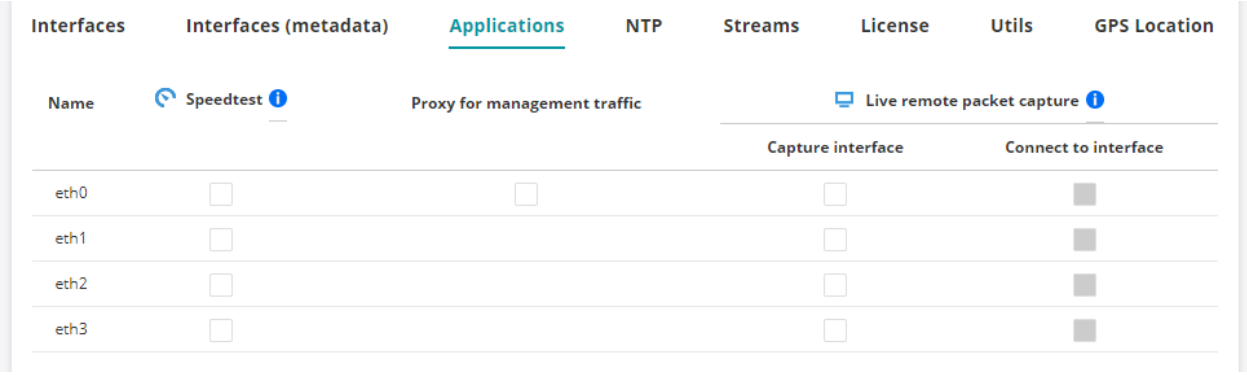
This tab is the same as for a regular, wired interface; it is covered *here* (page 174).

4.3.5.2 Related topics

- Use the *Wi-Fi logger* (page 353) task to collect Wi-Fi network measurements.
- Use the *Wi-Fi switcher* (page 354) task to change Wi-Fi interface parameters.
- Use the *Wi-Fi scanner* (page 352) task to scan for Wi-Fi networks.

4.3.6 Configuration of applications

On the Applications tab of the Test Agent configuration dialog, you can enable and disable the following Routing Active Testing applications: Proxy, Speedtest, and Live remote packet capture.



Speedtest: Browser-based speed and throughput testing of end-user connections. The Speedtest result page is found under Apps in the main menu. Read more *here* (page 482).

Proxy: Proxy functionality for Routing Active Testing management, used when some of your Test Agents do not have direct Internet access. Read more *here* (page 486).

Live remote packet capture: Real-time remote packet capture on the Test Agent. Once collected, this data can be accessed using Wireshark.

Note: The remote live capture process is stopped automatically after 24 hours, if not manually stopped before then.

In Wireshark, when you add the remote interfaces, use the IP address of the selected connect interface as host and 2002 as port. The default connect interface is the management interface.

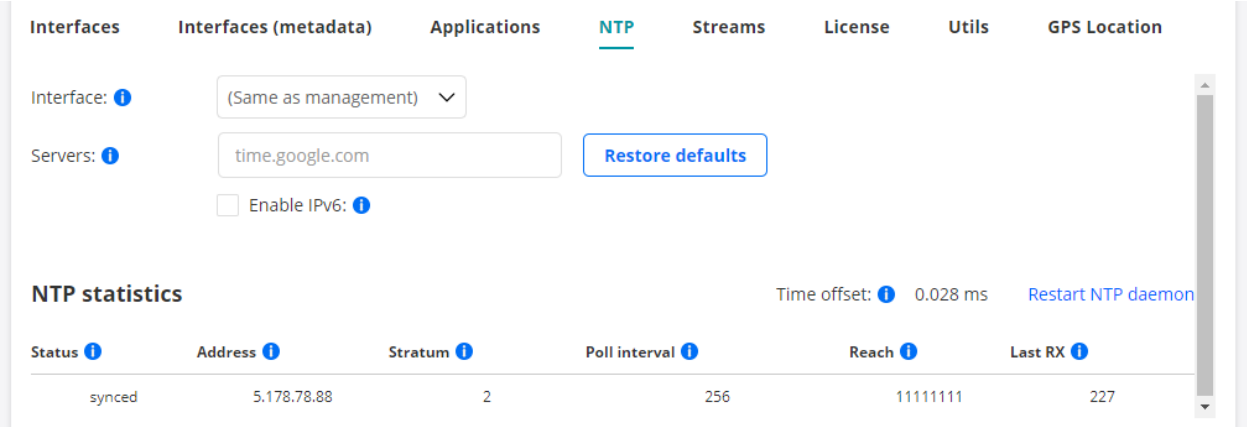
You can only select one remote interface at a time for capturing in Wireshark.

Routing Active Testing also has another packet capture function, *Remote Packet Capture* (page 479) found under Apps in the main menu. When using this function, the Test Agent stores the captured data locally, which you can later download from the Routing Active Testing web interface without the aid of Wireshark.

4.3.7 Test Agent NTP configuration

The NTP tab of the Test Agent configuration dialog deals with NTP server settings.

Many task types in tests and monitoring sessions measure delay and/or *jitter* (page 474). These measurements are dependent on an accurate clock. By default, Test Agents will synchronize their internal clock to time.google.com. This is a service provided by Google, and the terms and conditions for the service can be found at https://developers.google.com/terms/. You can however use any other NTP server (internal or external). In general, the closer the Test Agents are to the NTP server used, the more accurate the measurements will be.



Interface: The Test Agent network interface used for time synchronization.

Servers: A comma-separated list of NTP servers and/or NTP pools (IPv4/6 addresses or domain names). The default is to use the NTP server time.google.com; any other NTP server or pool can be specified either as an IP address or as a host name. Up to four servers will be used from each pool.

If you are using one Test Agent as a *proxy* (page 486) for your other Test Agents, you can have these other Test Agents synchronize their clocks using the proxy Test Agent. You then select the IP address of the proxy Test Agent as NTP server. The proxy Test Agent itself may for example synchronize to time.google.com.

If you are running the Test Agent in AWS, it is a good idea to use Amazon Time Sync Service, which provides a highly accurate reference clock by means of a fleet of redundant satellite-connected and atomic clocks in each Amazon region. To access this service, set Server to 169.254.169.123.

Enable IPv6: Enable IPv6 support for NTP.

4.3.7.1 NTP statistics

Time offset: The current time offset to the configured NTP server.

The NTP sync is normally very fast and reliable, and typical time offsets are smaller than 1 ms. The conditions below will therefore be rare (but need to be mentioned):

- If the time offset is larger than 100 ms, the Test Agent is not allowed to perform any testing.
- UDP one-way delay measurements pose special requirements on synchronization. If delays above 300 s or below -0.3 ms are measured, a clock synchronization error event will be triggered. The UDP task will still start and will produce other measurements, but no one-way delay will be obtained while this event is active.
- If a question mark "?" is shown for the time offset, this is because the Test Agent does not currently have a connection to the NTP server.

Clicking the Restart NTP daemon button restarts and resets the Test Agent's NTP client. Doing this will not usually improve the synchronization, however.

Status: Status of NTP source.

- *synced:* The Test Agent is currently synchronized to this source.
- *combined:* This is another acceptable source which is combined with the currently selected source.
- *not_combined:* This is another acceptable source which is excluded by the combining algorithm.
- *unreachable:* The connectivity to this source has been lost, or its packets do not pass all tests. It is also shown at start-up, until at least three samples have been gathered from the source.
- *time_error:* This source is judged to be a falseticker (that is, its time is inconsistent with a majority of other sources).
- *too_variable:* The time of this source appears to have too much variability.

Address: Address of NTP server.

Stratum: How close to a real-time reference the clock source is. The value zero means the NTP server is not reachable. The strata range from 1 through 15.

- Stratum 1 indicates a computer that has a true real-time reference directly connected to it (e.g. GPS or atomic clock); such computers are expected to be very close to real time.
- Stratum 2 computers are those which are synchronized against a stratum 1 server; stratum 3 computers are synchronized against a stratum 2 server, and so on.
- A large value like 10 indicates that the clock is so many hops away from a reference clock that its time is fairly unreliable.

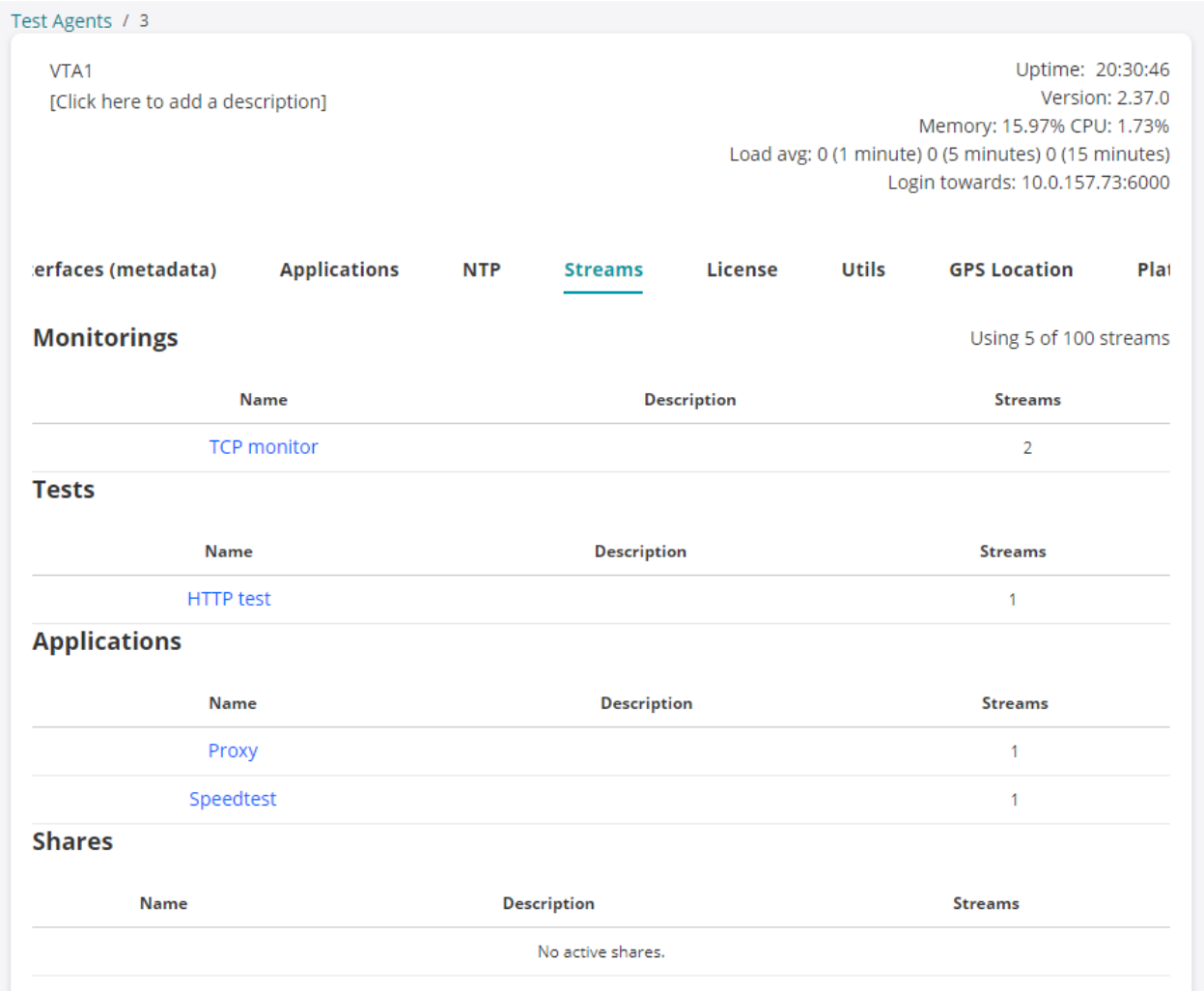
Poll interval: This shows the interval (in seconds) at which the source is being polled.

Reach: This shows the source's reachability register, printed as a binary number in string format. The register has 8 bits and is updated on every received or missed packet from the source. A value of `11111111` indicates that a valid reply was received for all of the last eight transmissions.

Last RX: This shows how long ago (in seconds) the last sample was received from the source.

4.3.8 Current activity on Test Agents

The Streams tab of the Test Agent configuration dialog shows the jobs that are running on the Test Agent (monitoring sessions, tests, applications, and shares) and the number of streams being consumed by each job. The streams are connected to the licensing in Routing Active Testing, as detailed *here* (page 245).



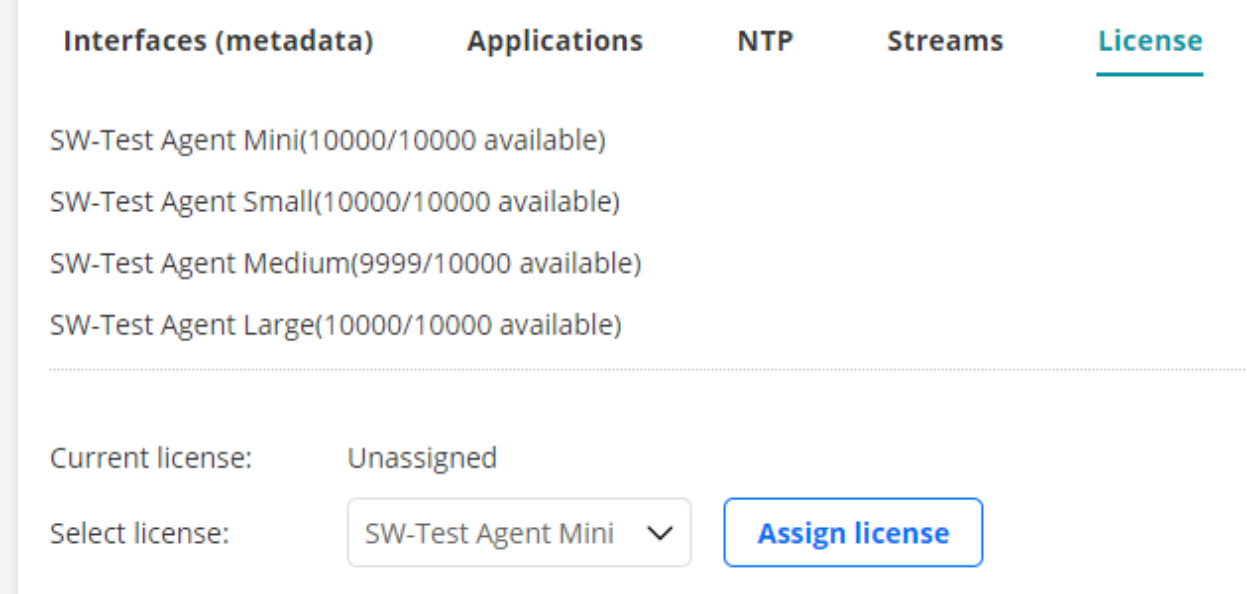
Running as well as completed jobs can also be viewed on the Tests and Monitoring screens accessed from the main menu.

4.3.9 Test Agent license management

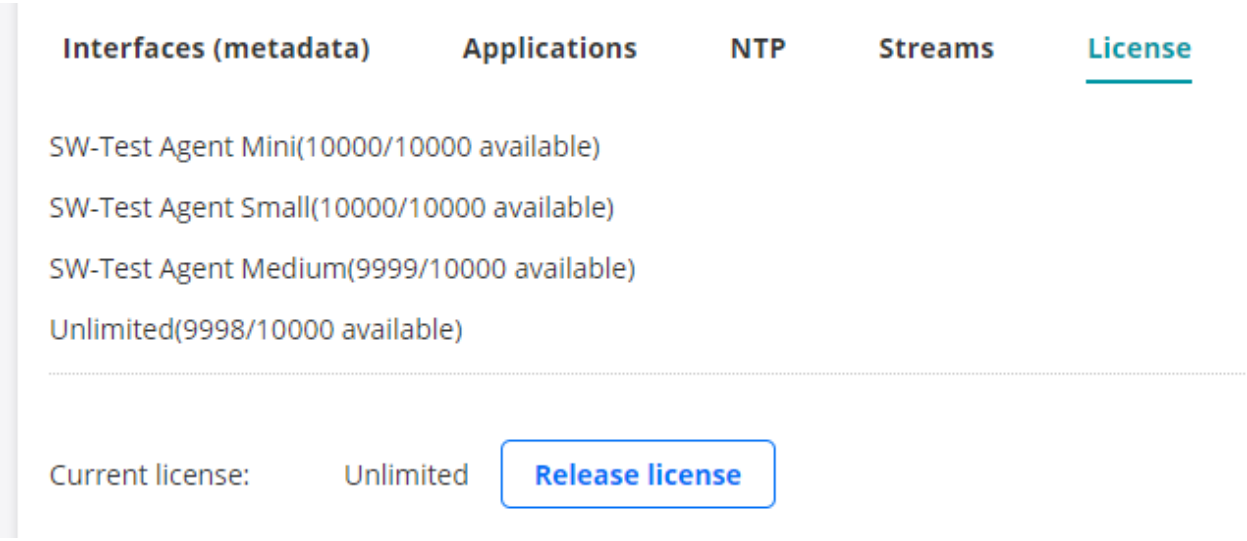
The License tab of the Test Agent configuration dialog deals with Test Agent license management. Here you can assign licenses to and unassign licenses from all Test Agents (except preinstalled ones, which have licenses assigned at delivery).

Read more about licensing on *this page* (page 245).

The following screenshot shows a Test Agent with an unassigned license:



- Select an appropriate license for the Test Agent, then click Save. The assigned license will appear immediately in the dialog:

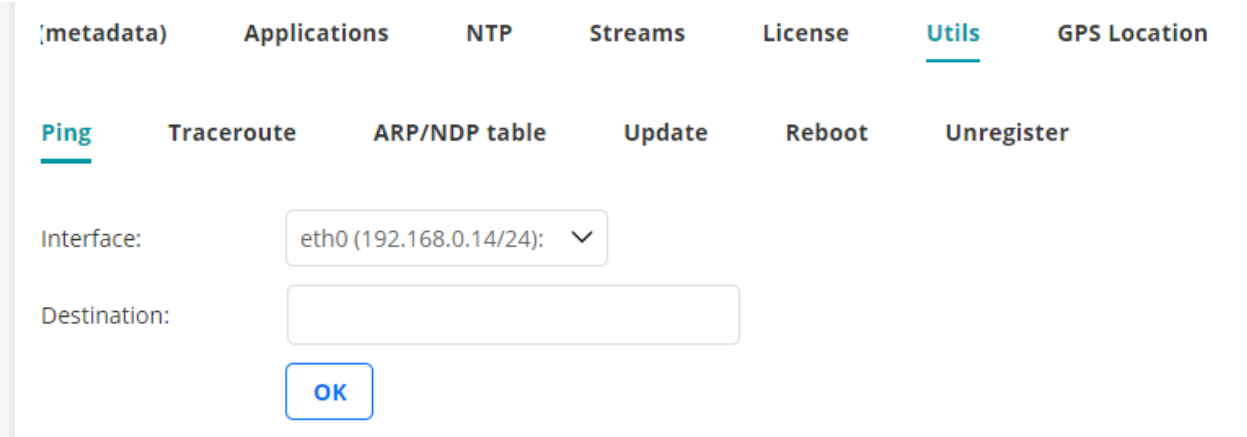


License information for all of your Test Agents is found on the License info tab of the Test Agents screen accessed from the main menu.

Name	License	No. of streams	Used streams	Available streams
na1_focal	Unlimited	8800	2	8798
VTA1	SW-Test Agent Medium	100	2	98
VTA2	Unlimited	8800	0	8800

4.3.10 Test Agent utilities

The Utils tab of the Test Agent configuration dialog holds some useful utilities for troubleshooting and management.



4.3.10.1 Ping tab

From here you can run ICMP Ping directly from the Test Agent towards a destination address. This is useful when troubleshooting Test Agent connectivity, for instance the management connection to the Routing Active Testing server. The function supports both IPv4 and IPv6.

Interface: The interface to use.

Destination: The destination IP address.

Example

```
PING www.google.se (173.194.71.94) from 192.168.1.73 : 56(84) bytes of data.
64 bytes from 173.194.71.94: icmp_req=1 ttl=45 time=39.0 ms
64 bytes from 173.194.71.94: icmp_req=2 ttl=45 time=38.0 ms
64 bytes from 173.194.71.94: icmp_req=3 ttl=45 time=38.0 ms
64 bytes from 173.194.71.94: icmp_req=4 ttl=45 time=38.0 ms
--- www.google.se ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 1502ms
rtt min/avg/max/mdev = 38.004/38.283/39.086/0.483 ms
```

4.3.10.2 Traceroute tab

From here you can run Traceroute directly from the Test Agent towards a destination address. The function is useful when troubleshooting Test Agent connectivity. Both IPv4 and IPv6 are supported.

Interface: The interface to use.

Destination: The destination IP address.

Example

```
traceroute to www.google.se (74.125.143.94), 30 hops max, 60 byte packets
1  192.168.1.1  1.227 ms  1.396 ms  1.688 ms
2  90.224.86.1  16.251 ms  16.233 ms  16.210 ms
3  80.91.250.43  31.519 ms  31.504 ms  31.481 ms
```

(continues on next page)

(continued from previous page)

```
4  213.248.93.198  32.972 ms  32.954 ms  32.932 ms
5  216.239.43.122  31.686 ms  31.666 ms  31.643 ms
6  209.85.254.31  32.268 ms  31.493 ms  31.353 ms
7  72.14.236.159  39.974 ms  39.851 ms  39.750 ms
8  72.14.233.170  39.535 ms  38.830 ms  38.736 ms
9  * * *
10  74.125.143.94  39.245 ms  38.925 ms  38.877 ms
```

4.3.10.3 ARP/NDP table tab

This tab displays the Test Agent ARP/NDP table on a selected interface.

Interface: The interface to display.

Example

```
192.168.1.1 dev eth0 lladdr a4:b1:e9:bd:f6:3c REACHABLE
```

4.3.10.4 Update tab

Here you can manually initiate a Test Agent software update. Note that the Test Agent will normally be restarted following the update.

4.3.10.5 Reboot tab

From this tab you can reboot the Test Agent.

4.3.10.6 Unregister tab

From here you can unregister the Test Agent from the Routing Active Testing system.

Note that an unregistered Test Agent cannot be re-registered. If you have accidentally unregistered one of your Test Agents, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

4.3.10.7 Move to other Control Center tab

Note: For obvious reasons, this feature is only applicable if you have several Routing Active Testing systems deployed, each having its own Control Center.

On the Move to Other Control Center tab you can move a Test Agent to another Control Center instance (called the "new Control Center" below). You do this by providing credentials for the Test Agent which enable it to register with the new Control Center:

- Test Agent name: Name of the Test Agent in the new Control Center.
- Host: Server hosting the new Control Center. Check the IPv6 box in order to use IPv6 when communicating with this Control Center.

- Port: Server port to which the Test Agent should connect in the new Control Center. If you do not specify a port, the Test Agent will connect to port 6000.
- Account, Username, Password: Credentials for logging in to the new Control Center.

When you are done entering this information, click the button Move to other Control Center. You are prompted to confirm this action. The Test Agent will then be moved out of your account in the current Control Center and transferred to the specified account in the new Control Center.

Note: You cannot move a Test Agent if it is acting as a proxy for management traffic for other Test Agents (see *this page* (page 486)).

4.3.10.8 Move to JRD tab

Note: For obvious reasons, this feature is only applicable if you are migrating your system to JRD.

On the Move to JRD tab you can move a Test Agent to an JRD instance (called the "new JRD" below). You do this by providing the secret for the Test Agent which enable it to register with the new JRD:

- Host: Server hosting the new JRD. Check the Validate SSL box in order to validate SSL certificates over HTTPS.
- Port: Server port to which the Test Agent should connect in the new JRD. If you do not specify a port, the Test Agent will connect to the default port 6800.
- Secret: Secret for authenticating to the new JRD.
- Ignore warnings Ignore pre-check warnings.
- Proxy address: HTTP Proxy address.
- Proxy port: HTTP Proxy port.
- Proxy auth type Toggle HTTP Proxy authentication.
- Proxy username: HTTP Proxy basic authentication username.
- Proxy password: HTTP Proxy basic authentication password.

When you are done entering this information, click the button Move to JRD. You are prompted to confirm this action. The Test Agent will then be moved out of your account in the current Control Center and transferred to the specificed new JRD.

Note: Moving a Test Agent if it is acting as a proxy for management traffic for other Test Agents could cause service disuption(see *this page* (page 486)).

4.3.10.9 Scheduling a move for a Test Agent that is offline

You can schedule a move for a Test Agent that is currently offline, by entering details exactly as explained above. The move will be performed once the Test Agent comes online. While the Test Agent is offline, you can cancel the move at any time.

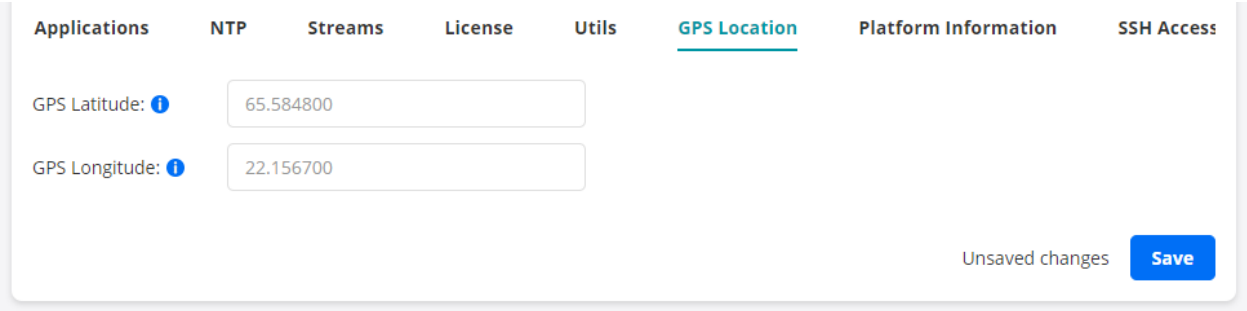
4.3.10.10 Advanced

It is possible to migrate the Test Agent configuration (including tests and monitors) to the new Control Center instance prior to performing the move, so that the Test Agent will retain the same identity as in the current Control Center. For this to work, it is imperative that the Test Agent names in the source and target instances match.

Such a migration cannot be done through the Control Center GUI; for assistance in these matters, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

4.3.11 Test Agent GPS location

On the GPS Location tab of the Test Agent configuration dialog you can enter geographical coordinates for the Test Agent.



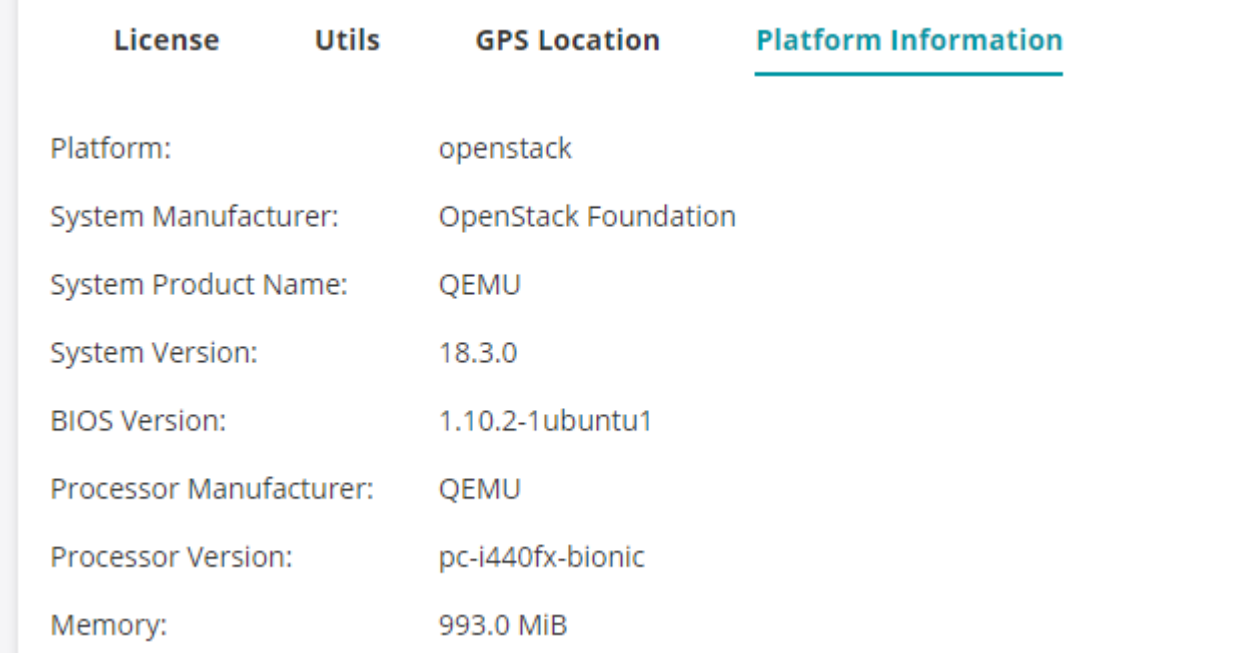
Latitude and longitude are to be given according to the Web Mercator/Pseudo-Mercator projection, which is based on the WGS 84 (World Geodetic System 1984) coordinate system. See this web page: ► https://epsg.io/3857.

- GPS Latitude: Test Agent latitude according to WGS 84. Expressed as a decimal number between –85.06 and +85.06, where a negative number means “south”.
- GPS Longitude: Test Agent longitude according to WGS 84. Expressed as a decimal number between –180 and +180, where a negative number means “west”.

Note: The coordinates must be given in decimal degrees. They cannot be given in degrees, minutes and seconds, nor can they contain letters such as “N” or “W”.

4.3.12 Test Agent Platform Information

On the Platform Information tab of the Test Agent configuration dialog you can view details on the Test Agent platform. As the items found here are self-explanatory, no further commentary on them is provided here.



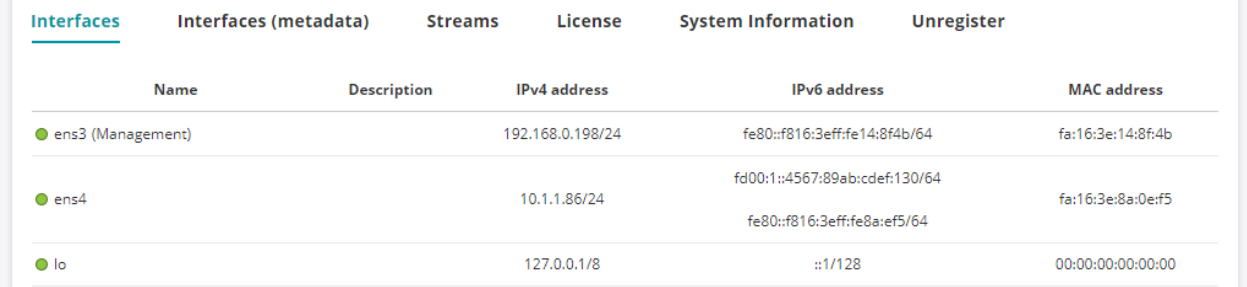
4.3.13 Managing Test Agent Applications

Clicking a Test Agent Application opens a dialog displaying some of the properties of the device.

Test Agent Applications have a limited set of functionality compared to Test Agent Appliances, and there are no configuration options. The dialog is divided into tabs as shown below.

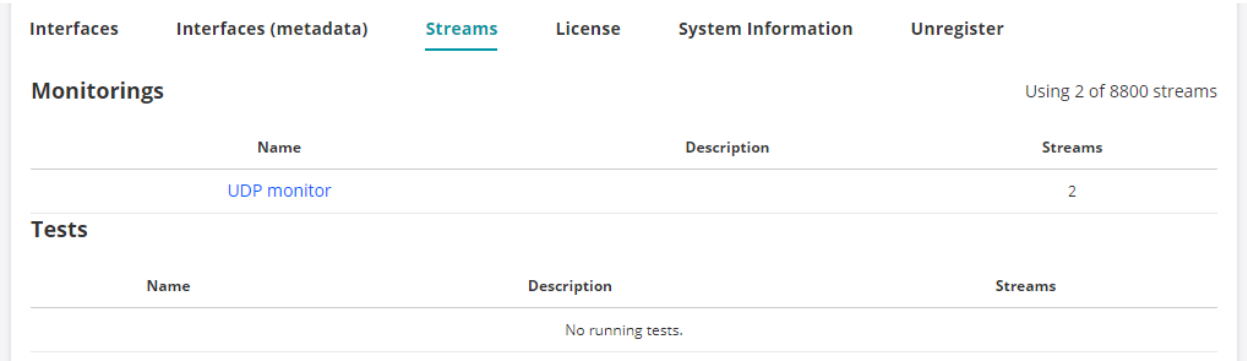
4.3.13.1 Interfaces tab

This tab shows all interfaces on the computer where the Test Agent is running.



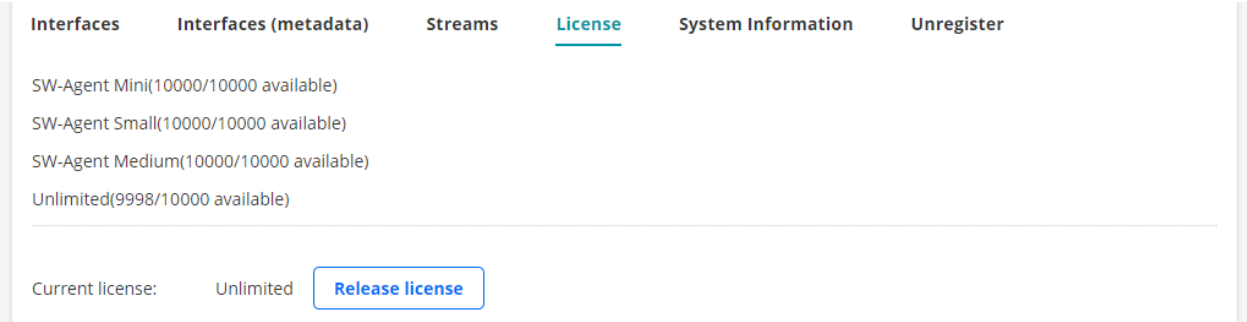
4.3.13.2 Streams tab

This tab lists all streams that are being used by monitors and tests currently running on the Test Agent.



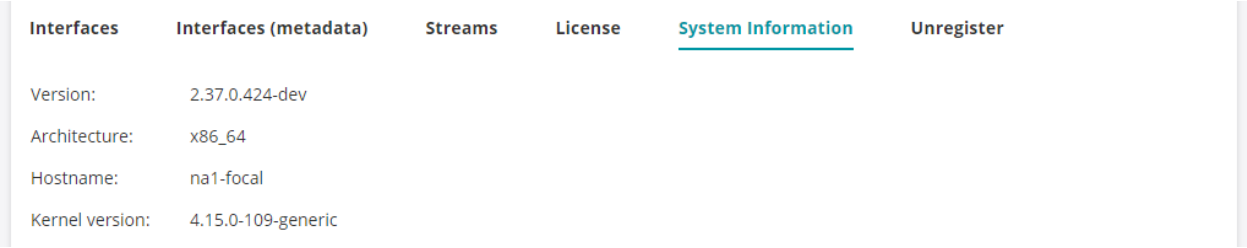
4.3.13.3 Licenses tab

If no streams are currently in use, you can click the Release license button to release the license installed on the Test Agent.



4.3.13.4 System Information tab

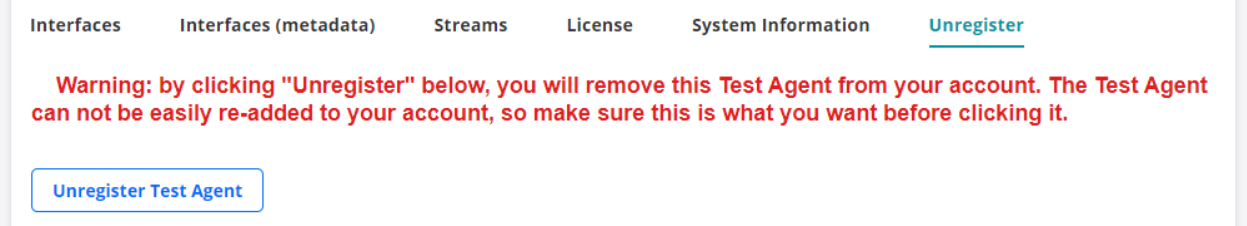
This tab displays the Test Agent version and some information on the system where it is running.



4.3.13.5 Unregister tab

This tab holds a button for unregistering the Test Agent.

Note that to re-register a Test Agent Application, you need to download the software and reinstall it on your computer. Therefore, do not unregister such a Test Agent unless you want to remove it permanently from your Routing Active Testing account.

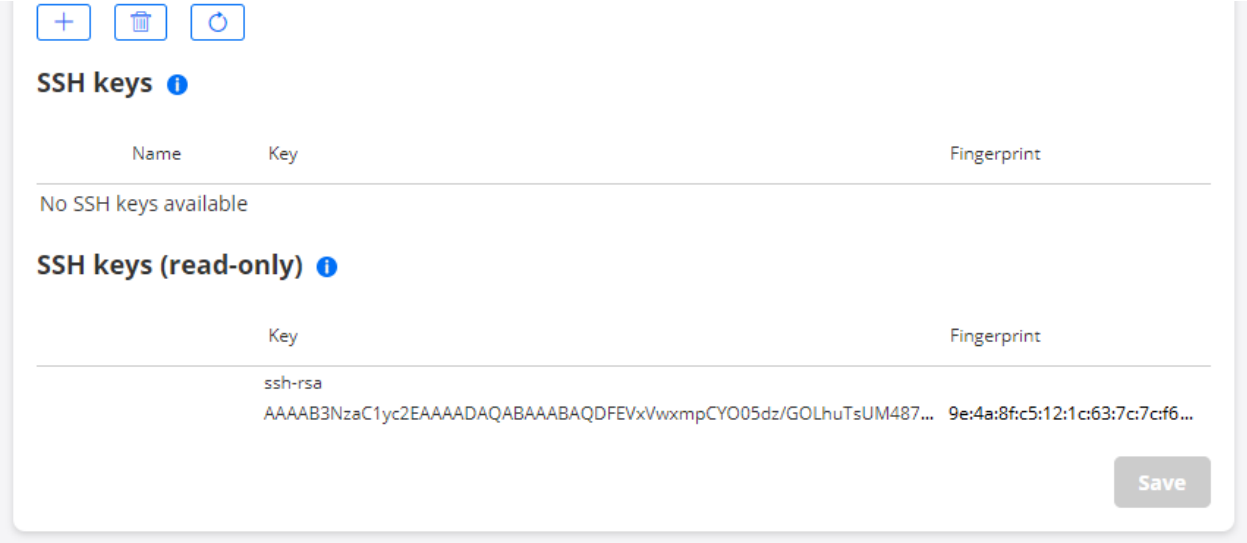


4.3.14 Test Agent SSH keys

The SSH Access tab deals with SSH keys.

You can log in to a Test Agent via SSH by uploading an SSH public key to the Test Agent and then use your corresponding private key to set up an encrypted connection. The public key can be uploaded either from Control Center or from the Test Agent local console.

The tab has two sections:



SSH Keys: These SSH keys are managed by Control Center and are pushed to the Test Agent.

You add SSH keys by clicking the plus-sign button and entering the key. After the key has been installed on the Test Agent, SSH is automatically activated on that Test Agent once it comes online.

A *yellow* dot to the left of the SSH key name means that the SSH key has not yet been transferred to the Test Agent. Once the key has been transferred, the dot turns *green*.

You can delete an SSH key by selecting its checkbox, then clicking the trash can button. If you delete all keys, SSH is automatically deactivated on the Test Agent, unless one or more read-only SSH keys are left (see below). As the latter are not manageable from Control Center, the SSH capability will remain activated in that case.

SSH keys pushed from an API (REST or NETCONF/YANG) also end up in this category, so that they are manageable from Control Center.

Note: Any SSH key with one or more options specified will be rejected by the Test Agent, as will keys containing leading or trailing whitespace such as spaces or newlines. Such keys will nevertheless appear in the Control Center GUI, but will never become "green".

SSH Keys (read-only): These SSH keys are managed locally on the Test Agent.

These keys are ones entered in the Test Agent local console, as described *here* (page 226), or via cloud-config (cloud-init). They cannot be deleted from Control Center.

4.3.14.1 How to generate and use SSH keys

Below is an example of how to generate an SSH key pair (`key_file` and `key_file.pub`).

```
$ ssh-keygen -t rsa -f key_file -b 4096 -C "test-agent-key"
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key_file.
Your public key has been saved in key_file.pub.
The key fingerprint is:
SHA256:w6gKxgzBDOKFDBmKp9+3Y3a437JOIUiM3/oNQac/a2s test-agent-key
The key's randomart image is:
```

(continues on next page)

(continued from previous page)

```
+---[RSA 4096]----+
|Bo..             |
|Xo.o             |
|++o o . .        |
| + o + =         |
|o   o * S        |
|+. . o + o       |
|.+. + o.+        |
|.. . o==E+       |
|  .  o+OB+.      |
+----[SHA256]-----+

$ ls key_file*
key_file  key_file.pub
```

Now copy the contents of `key_file.pub` and use it to create a public key either from Control Center or from the Test Agent local console.

Finally, use the private key to connect to the Test Agent, for example:

```
$ ssh admin@10.0.150.100 -i key_file
```

4.3.15 Test Agent RAN configuration

The RAN tab of the Test Agent configuration dialog lets you configure a gNodeB (base station) and UEs (user equipment) in a 5G RAN (radio access network) for a Test Agent.

4.3.15.1 Adding a gNodeB

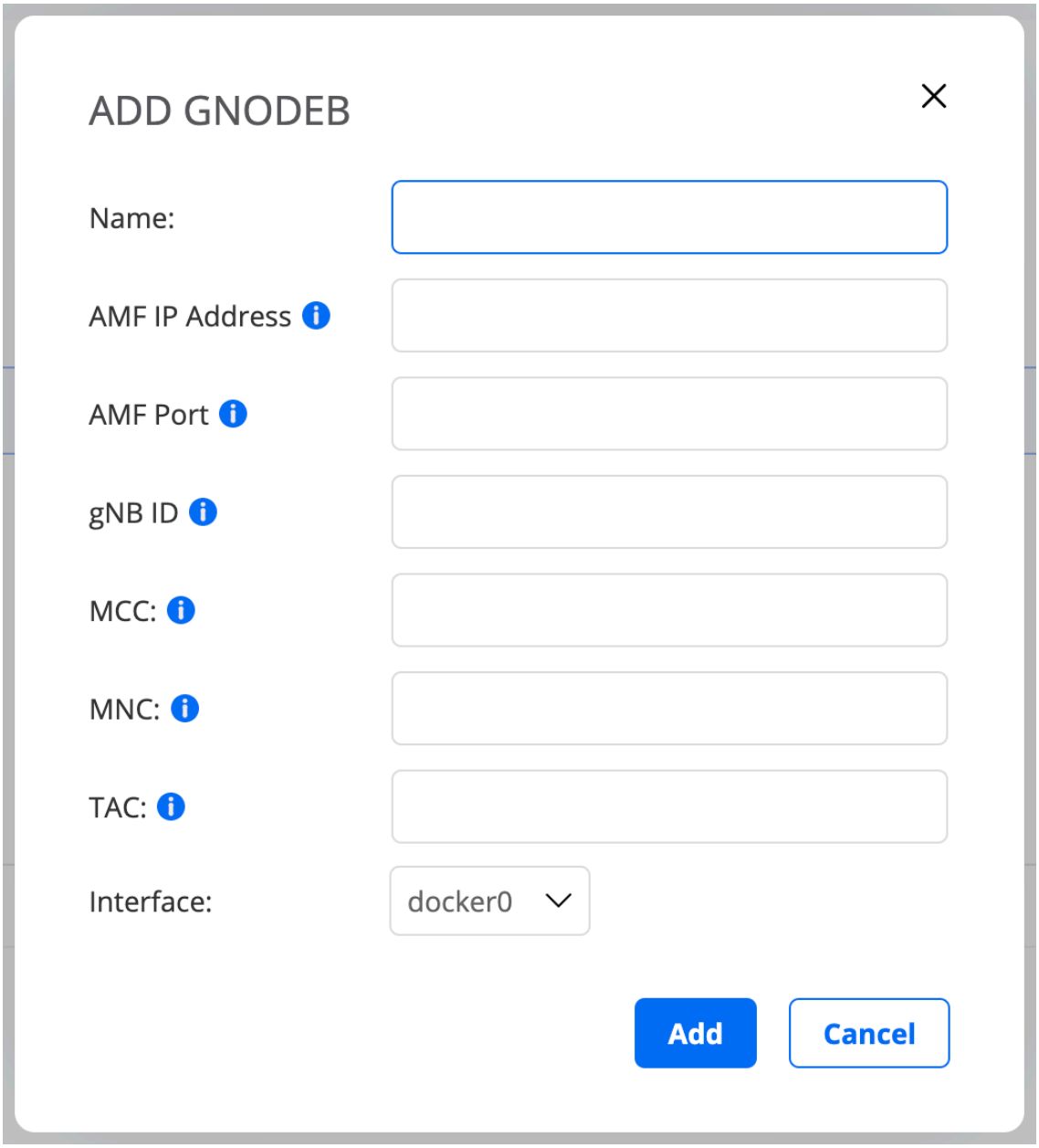
Note: It is currently not possible to configure more than one gNodeB for a Test Agent.

To create a gNodeB:

- Click the Add button below the gNodeB heading.



A dialog appears with the following settings:



- Name: Enter a name for the gNodeB.
- AMF IP Address: Address to the 5G core Access and Mobility Management Function (AMF) to connect to.
- AMF Port: AMF port to connect to. The default is 38412.
- gNB ID: 28-bit unique gNodeB identifier within the PLMN (public land mobile network).
- MCC: Mobile Country Code, 3-digit country identifier.
- MNC: Mobile Network Code, 2-digit or 3-digit carrier identifier.
- TAC: Tracking Area Code, 16-bit unique identifier for a tracking area within the PLMN.
- Interface: Interface used to connect to the 5G core AMF.

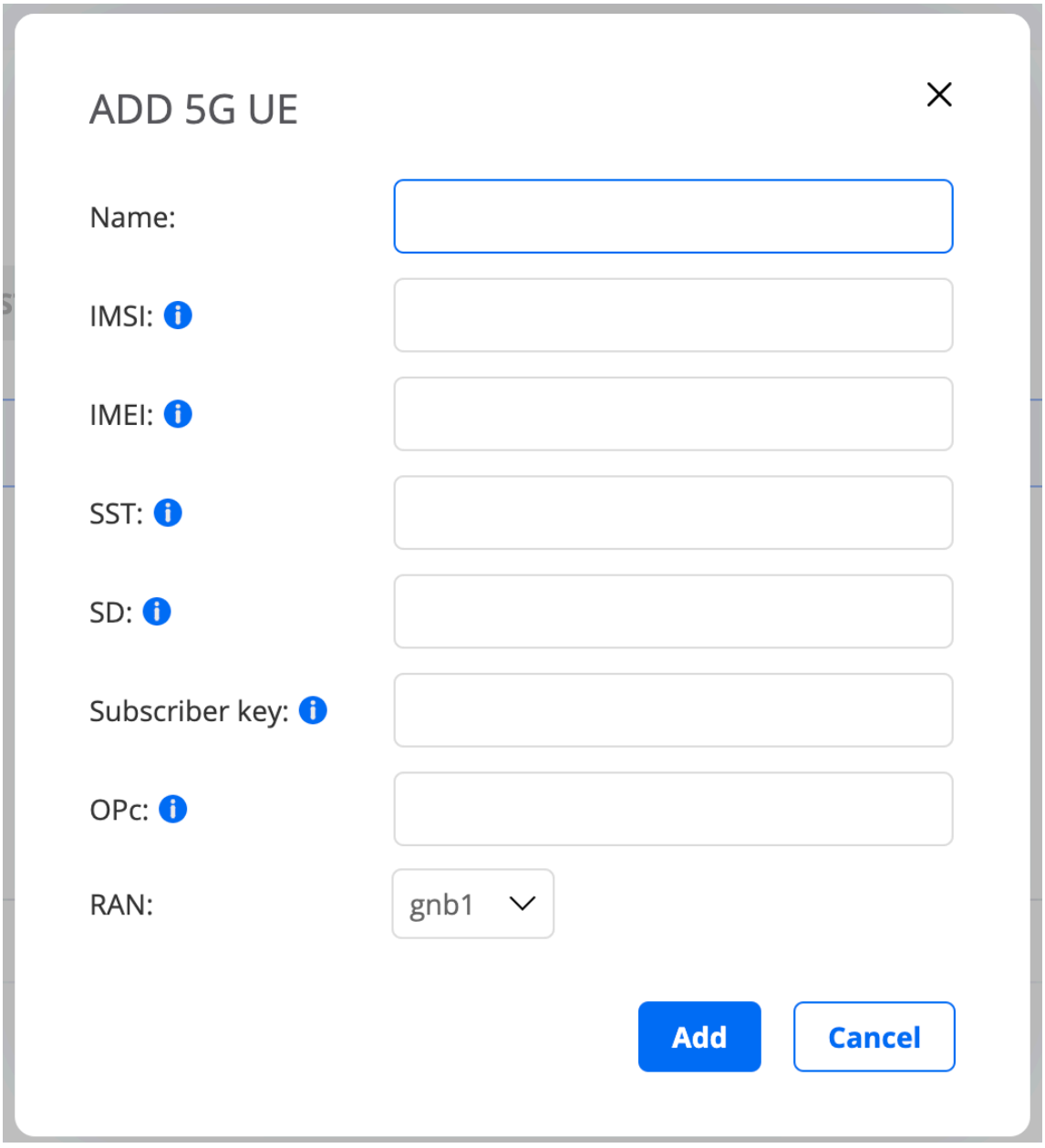
4.3.15.2 Adding a 5G UE

You can create multiple UEs, each of which is connected to the gNodeB configured. For each new UE:

- Click the Add button below the UE heading.



A dialog appears with the following settings:



- Name: Enter a name for the UE.
- IMSI: International Mobile Subscriber Identity, 15-digit unique identifier of the end user.
- IMEI: International Mobile Equipment Identity, 15-digit unique identifier for 3GPP devices.

- SST: 8-bit slice/service type.
- SD: 24-bit slice differentiator.
- Subscriber key: 128-bit key used for subscriber authentication.
- OPc: Operator code (en)crypted, 128-bit key derived from the Operator Key and used for authentication by all subscribers of a particular operator.
- RAN: The gNodeB to which this UE should connect (currently, only one can be defined).

Note: If you modify the UE configuration later on, this may intermittently impact the operation of UEs previously configured. Specifically, that will happen if you add a new UE or change the slice configuration (SST and SD parameters) of an existing UE. In this situation the gNB needs to be restarted, since available slices need to be negotiated between the gNB and the 5G core during session establishment. All other UEs connected to the gNB will then also have to be restarted. Any tests and/or monitors running on these UEs will therefore be interrupted and may have to be rerun/restarted.

4.3.15.3 Deleting a gNodeB and UEs

To delete a gNodeB or UE:

- Check the checkbox next to the item.
- Click the trash can button that appears.



To delete all UEs, click the Delete all button under the UE heading.

4.4 Configuring Test Agents from the local console

4.4.1 Accessing the local console of the Test Agent

Accessing a physical Test Agent locally means that you must either attach a keyboard and display or connect a serial cable. The latter method is required for certain preinstalled Test Agents. Virtual Test Agents, on the other hand, can be accessed through the console provided in the virtualized environment. Detailed instructions follow below.

Regardless of access method and Test Agent type, you log in as user “admin” using “admin” as password.

Note: The keyboard layout for all local console access is “standard American”. Please keep this in mind if you are using a different keyboard layout for local management.

4.4.1.1 Preinstalled Test Agent on HW Medium or HW Small

These devices can only be accessed through their serial port. A null modem serial cable (9-pin D-Sub, DB-9) therefore needs to be connected to the Test Agent.

The Test Agent adapts to the data transfer capabilities of the terminal emulator. The Test Agent first tries its default 38,400 baud data rate, which is recommended for communicating with the Test Agent. If the terminal emulator instead uses a 9,600 or 115,200 baud rate, the Test Agent will usually detect this automatically and switch to the correct rate. It might however be necessary to send a BREAK for the Test Agent to detect the new baud rate. See the documentation for your terminal emulator for information on how to send a BREAK.

The rest of the Test Agent serial port configuration is "8N1" (8 bits, no parity, 1 stop bit).

4.4.1.2 Preinstalled Test Agent on HW Medium Plus or HW Large

To these devices you can connect a standard USB keyboard and VGA display.

Alternatively, these devices can be accessed via their serial port, as described *above* (page 205).

4.4.1.3 Test Agent installed on NFX150

Give the following command in the NFX150 CLI:

```
request virtual-network-function console testagent
```

4.4.1.4 Test Agent installed on custom hardware

Connect a standard USB or PS/2 keyboard and VGA display to the device where the Test Agent is running. Should this not be possible, access through the serial port is a fallback option here as well if the device has one.

4.4.1.5 Virtual Test Agent

Use the standard console on your virtualization management server to access the virtual machine where the virtual Test Agent is running.

4.4.2 Registering a Test Agent from the local console

This page describes the process for registering a Test Agent. The description is applicable to all user-installed Test Agents, i.e. ones installed from a disk image or OVA package, as well as Virtual Test Agents (vTAs).

Note that vTAs also support initialization via cloud-init; see *this page* (page 103).

Yet another possibility is to preconfigure Test Agent registration details offline. This is covered on a *separate page* (page 208).

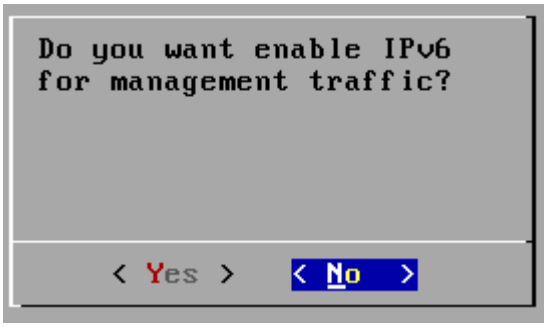
We recommend that the registration of Test Agents be handled by a "Test Agent registration" user, which has no other privileges. How to create such a user is described *here* (page 20).

4.4.2.1 Registration process

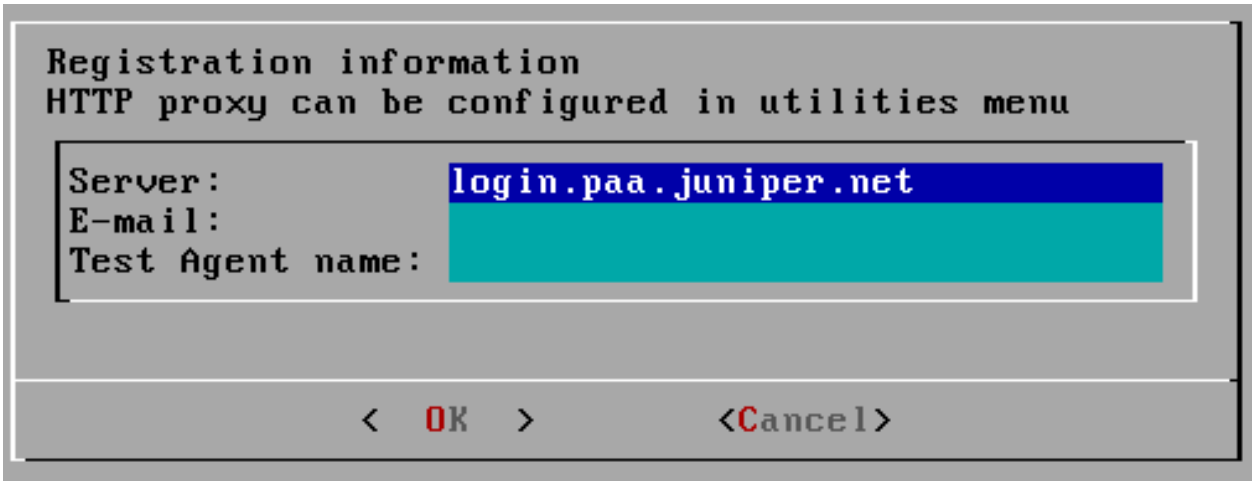
- Use the text-based menu and navigate to Register.



You will be asked whether you want to enable Test Agent management over IPv6. If enabled, this is configured as described *here* (page 210). The registration process itself is the same regardless of your choice here.



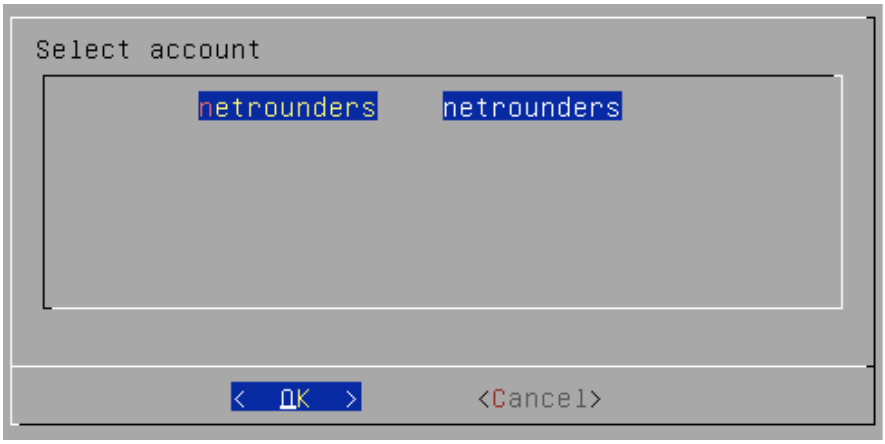
- Under Server, you should normally use the host name or IP address of the Routing Active Testing server.
  - In the cloud server case, this is https://login.paa.juniper.net. For this server alone, the Test Agent will connect to port 443 (Test Agent Appliance) or 6800 (Test Agent Application; configurable).
  - In the on-premise server case, enter the server address. If you do not specify a port, the Test Agent will connect to port 6000. You can use a different port by specifying it explicitly: ":<port number>".
  - If you want to register the Test Agent via *another Test Agent used as proxy* (page 486), you need to point to the static IP address of the proxy Test Agent, with port 443 specified. This is possible with Test Agent Appliances only.
- Under E-mail, enter the email address that serves as user name for your Routing Active Testing account or your Test Agent registration user.
- Under Test Agent name, enter a name for the Test Agent. This name is what will be shown in the Test Agents view.



- Enter the password for your Routing Active Testing account or Test Agent registration user.



- Select your Routing Active Testing account. If you are a member of several accounts, all of these will be shown in a list.



The message “Registration successful” should now appear. The Test Agent will then be visible in the Test Agents view for your Routing Active Testing account: `https://<Control Center host IP>/<your account>/genalyzer`.

Note on reusing Test Agent names

The following applies if a "Test Agent registration" user is used for registering Test Agents. With an admin user, the requirement below does not apply.

If you want to reuse the name of a Test Agent that you have registered, then *unregistered* (page 194), you must have the following set in `/etc/netrounds/netrounds.conf`:

```
REGISTER_ONLY_ALLOW_REPLACE=True
```

After editing this file, you need to restart all services:

```
sudo ncc services restart
```

4.4.3 Preconfiguring registration details for Test Agents

This function is primarily intended for use in the process of installing Test Agent software on hardware devices. It can also be used with preinstalled Test Agents; however, in what follows, the installation scenario is described.

In certain situations, especially when you are going to deploy a large number of Test Agents, it may be convenient to first preconfigure the Test Agents with registration details without being dependent on network connectivity, and do the actual registration only later. You can do this using the Offline registration utility in the Test Agent local console.

Follow these steps:

- At the outset, the Test Agent should have no network connectivity.
- Boot the Test Agent from a USB stick holding the Test Agent image (as also described on *this page* (page 75)):
  - Insert the USB stick with the Test Agent image into a USB port on your hardware device.
  - Access the BIOS boot menu.
  - Make sure the USB memory comes before the hard disk in the boot sequence.
  - Select "USB boot" from the BIOS boot menu.
  - The boot process takes about 20 seconds. When the login prompt is shown, log in as user "admin" with password "admin".
- In the text-based menu, navigate to Utilities > Offline registration.

A screen with a text field appears. Here you need to enter a cloud-init configuration with the same contents and syntax as for virtual Test Agents. It is described on *this page* (page 103). The configuration you enter here will remain in the dialog the next time you open it.

- If you do not specify a `name` for the Test Agent, a name will be generated from the MAC address of the Test Agent's management interface.
- If you do not specify `management_address_type`, it will default to DHCP.

This completes the cloud-init configuration.

- Next, go to Utilities and select Install to disk.
- After the installation has completed, remove the USB stick.

Then, at some later time:

- Connect the Test Agent to a network, and reboot the Test Agent. The Test Agent will now register: cloud-init will pick up the configuration just entered, configure the Test Agent management interface and register the Test Agent with Control Center. Upon successful registration, the configuration file is deleted.

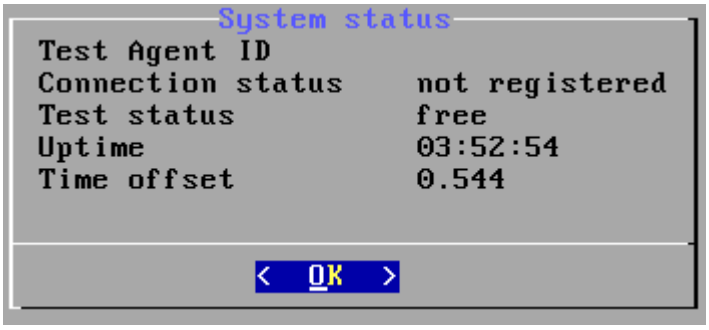
For a registered Test Agent, the Offline registration menu item changes to Unregister, and you can select this to remove the Test Agent from the system.

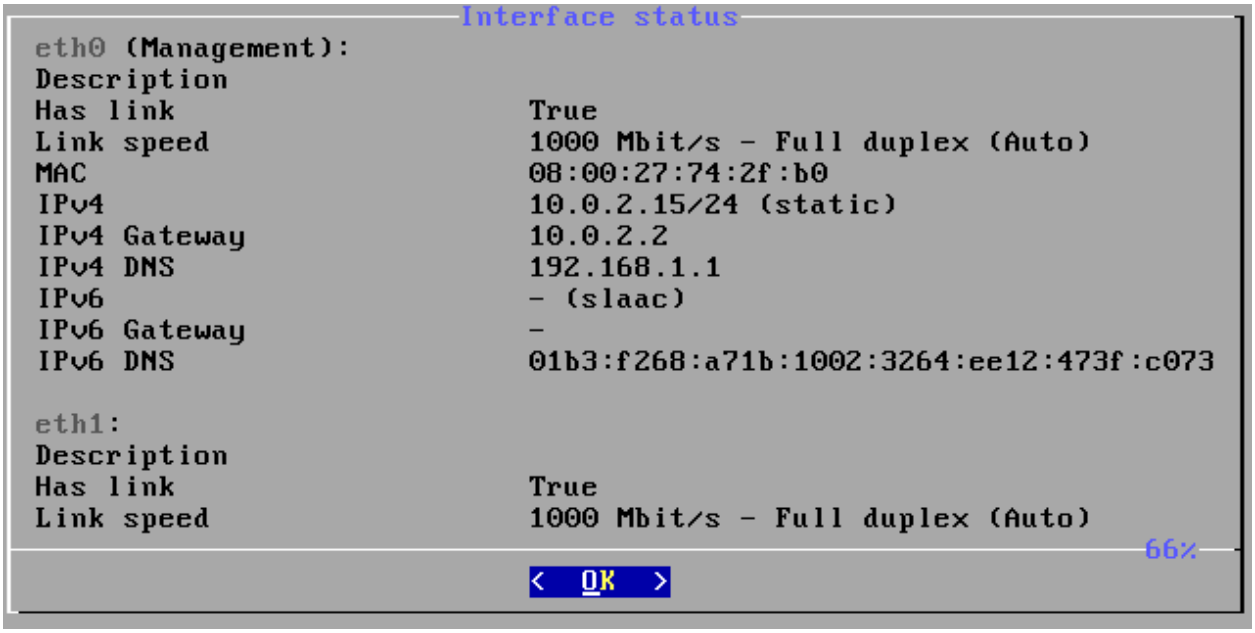
4.4.4 Configuring a Test Agent from the local console

4.4.4.1 Show system status/Show interface status

To inspect the current status of the Test Agent and its interfaces, navigate to Show system status and Show interface status respectively in the text-based menu.







The System status screen shows the Test Agent ID. This ID is the unique identifier of a Test Agent, which simplifies correlation with the Test Agents you see in the GUI (the URL of the Test Agent ends with that identifier: in the above example, `https://<Control Center host IP>/<your account>/genalyzer/55`).

Connection status can have the following values: *not registered*, *connecting*, or *logged in*.

- *Not registered:* The Test Agent is not registered to the Routing Active Testing account.
- *Connecting:* The Test Agent is trying to connect to the Routing Active Testing server.
- *Logged in:* The Test Agent is connected and logged in to the Routing Active Testing server.

Test status is one of the following: *free*, *updating*, or *in use*.

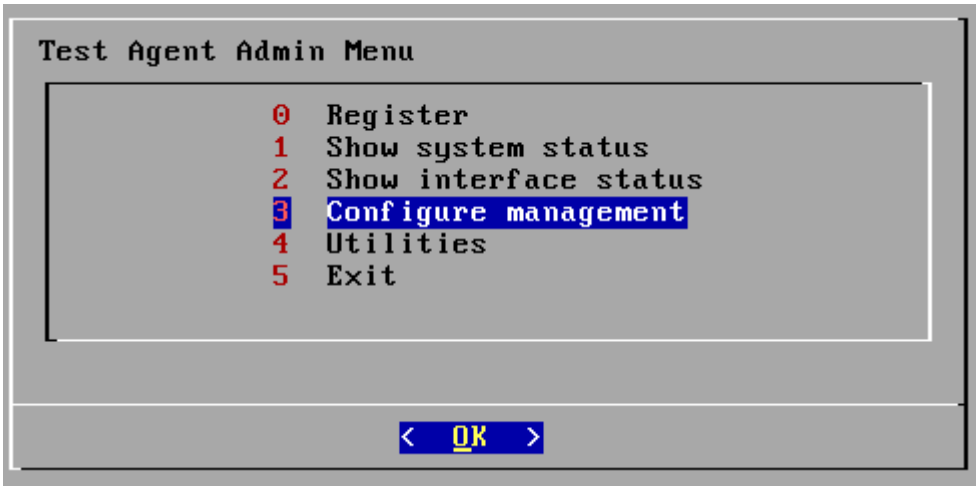
- *Free:* The Test Agent is idle.
- *Updating:* The Test Agent is downloading and installing the latest firmware. Note: Do NOT power off the Test Agent during updates.
- *In use:* The Test Agent is performing tests or monitoring.

The Interface status screen shows the status of all available interfaces of the Test Agent. In the example shown, “eth0” is the management interface with link and IPv4 as well as IPv6 addresses, while “eth1” has no IP.

4.4.4.2 Configuring the management interface on a Test Agent

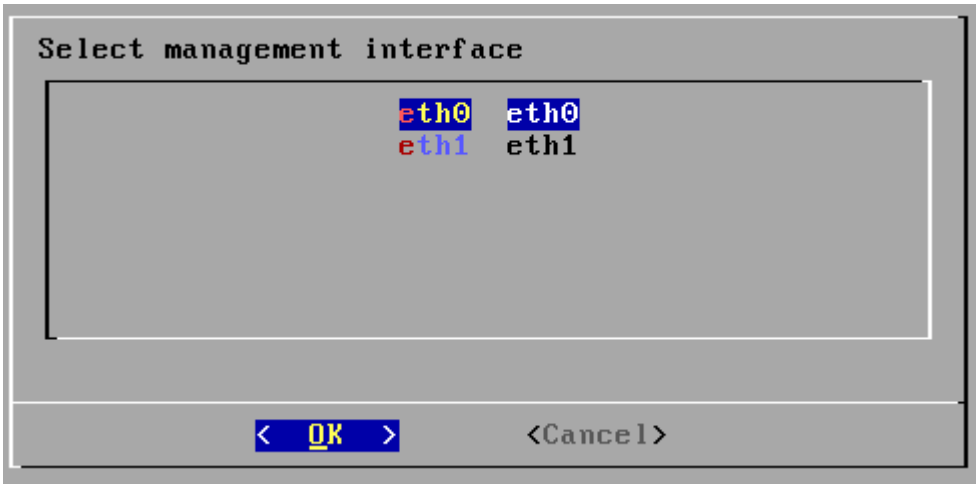
For configuration of Test Agent management over a mobile network interface, see *this page* (page 228).

- In the text-based menu, navigate to Configure management.

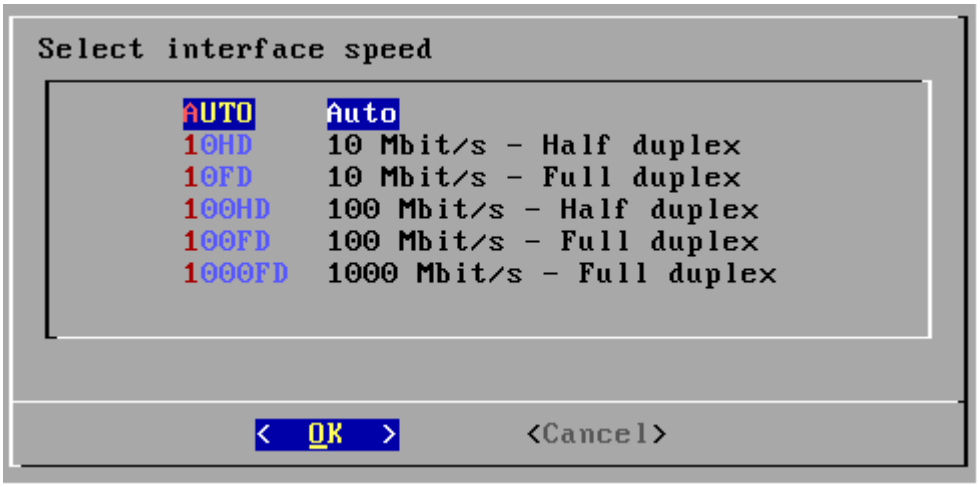


- Select which interface should handle management traffic to and from the server (preferably, “eth0”).

Configuration of the other interfaces is done via the Routing Active Testing server; see *Test Agent interface configuration* (page 166).

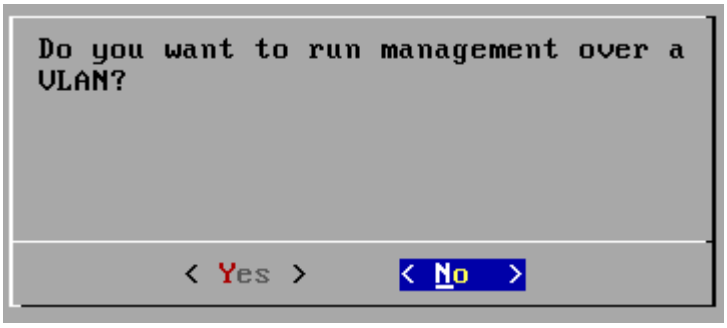


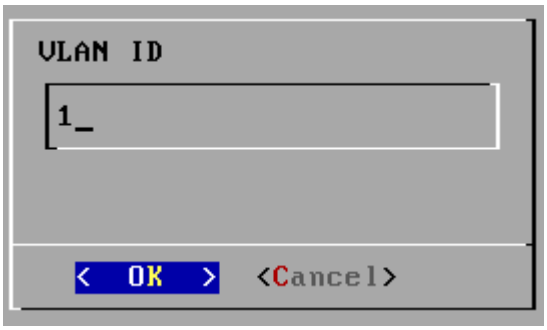
- Then select interface speed and duplex on the management port.



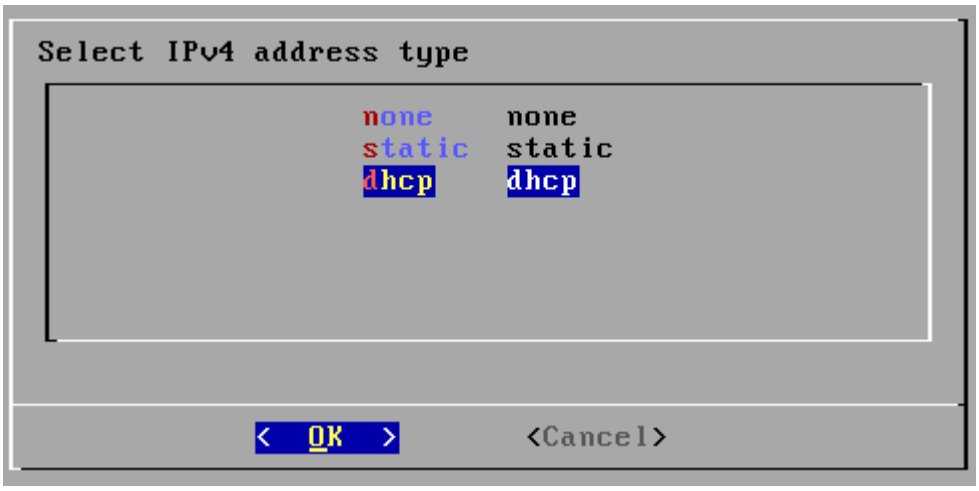
- You are also asked whether to use a VLAN tag for the management interface. Select Yes or No. If you select Yes,

enter the VLAN ID, for example “1” as in the screenshot.

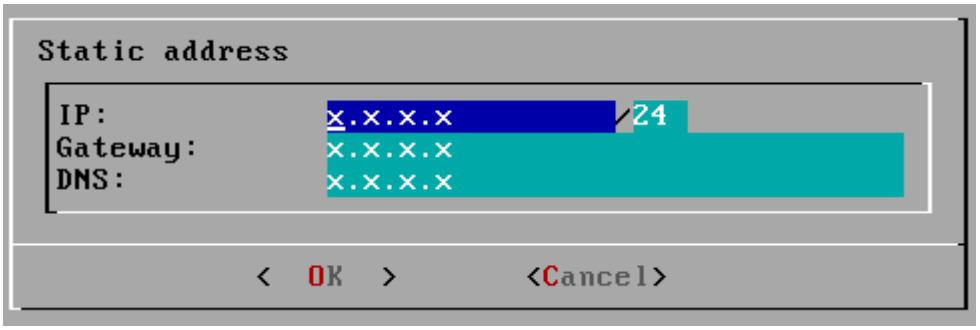




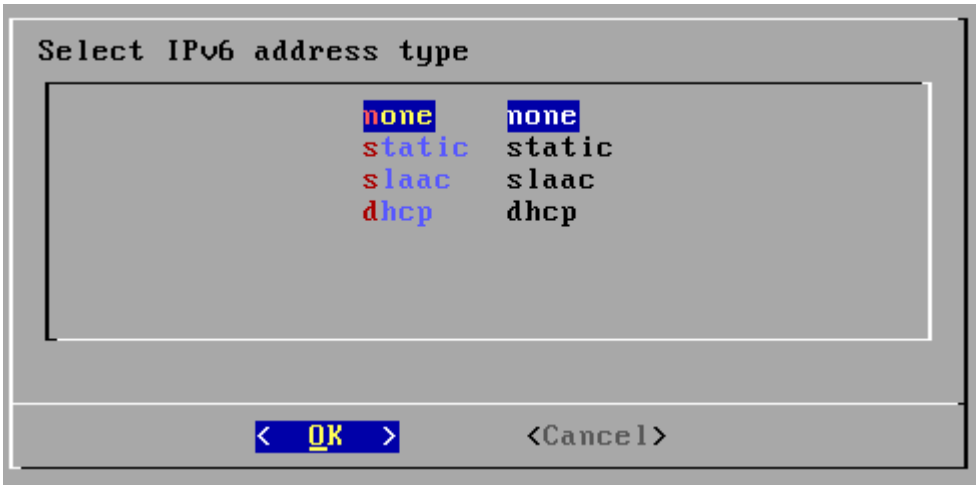
- Next, select IPv4 address type: DHCP or static. Alternatively, you can choose not to configure an IPv4 address type by selecting “none”.



- If you selected “DHCP” as address type, no more configuration is needed for IPv4. On the other hand, if you selected “static” as address type, you need to configure IP, Network mask, Gateway, and DNS in the format shown in the screenshot below.



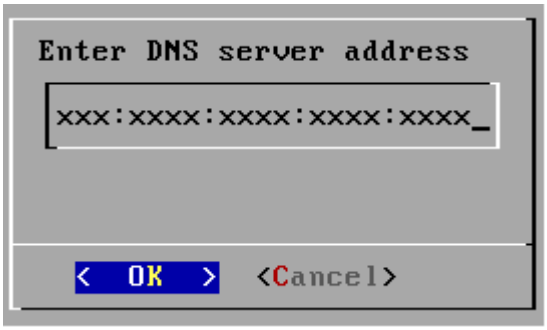
- Now if you enabled IPv6 for management when *registering* (page 205) the Test Agent, the next screen will prompt you to select IPv6 address type. This is one of “static”, “slaac”, or “dhcp”. If you don’t want to configure an IPv6 address, select “none”. (If IPv6 is not enabled, the configuration will skip to the *NTP server setting* (page 214).)



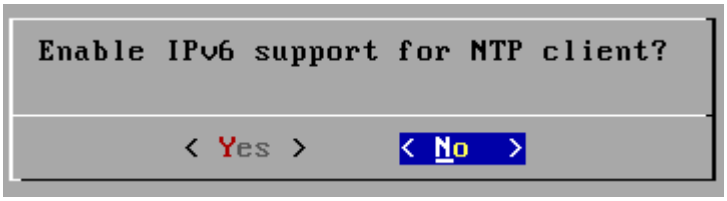
- Again, for DHCP no more configuration is needed.
- For “static”, you need to configure IP, Gateway, and DNS in the format shown in the screenshot.



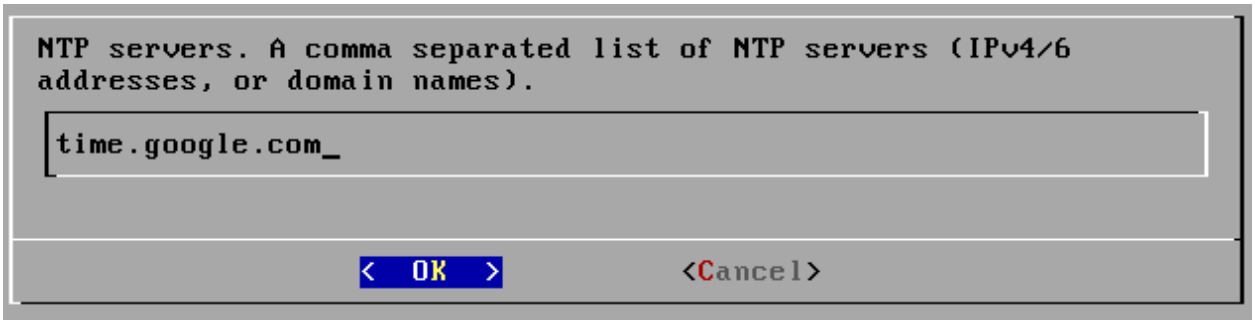
- For SLAAC, enter the DNS server address:



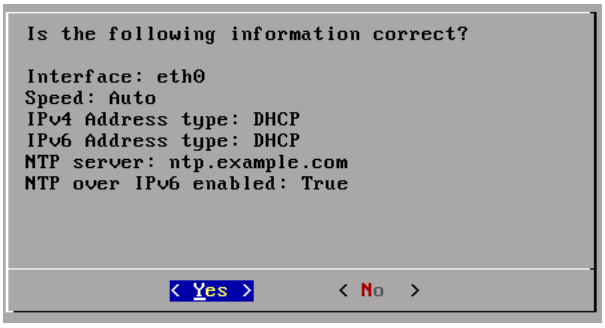
- Next, you are asked whether to enable IPv6 support for the NTP client. If you do, you can specify an IPv6 address to the NTP server.



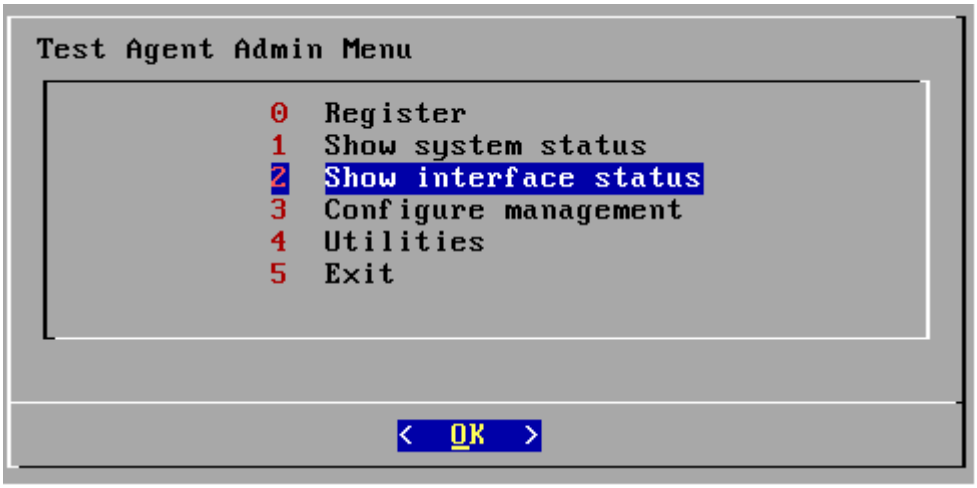
- The NTP servers setting follows. You can use multiple NTP servers and/or NTP pools, specified as a comma-separated list. An NTP server can be specified either as an IP address or as a hostname. An NTP pool must be specified as a hostname. Up to four servers will be used from each pool. If you are running the Test Agent in Amazon WS, it is a good idea to use Amazon Time Sync Service, as explained *here* (page 189). To access this service, enter the IP address 169.254.169.123.



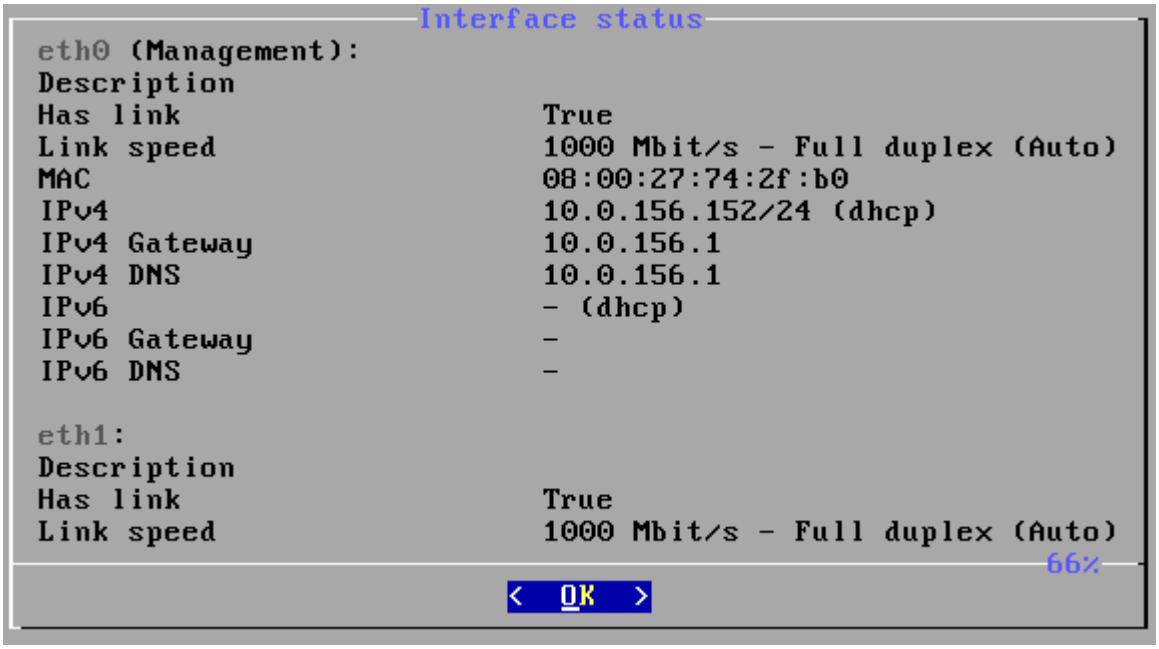
- Finally, verify that the settings you have entered are correct. For example, Interface = “eth0.100” means VLAN ID 100 on the “eth0” Ethernet interface.



- To check the configuration, navigate to Show interface status.

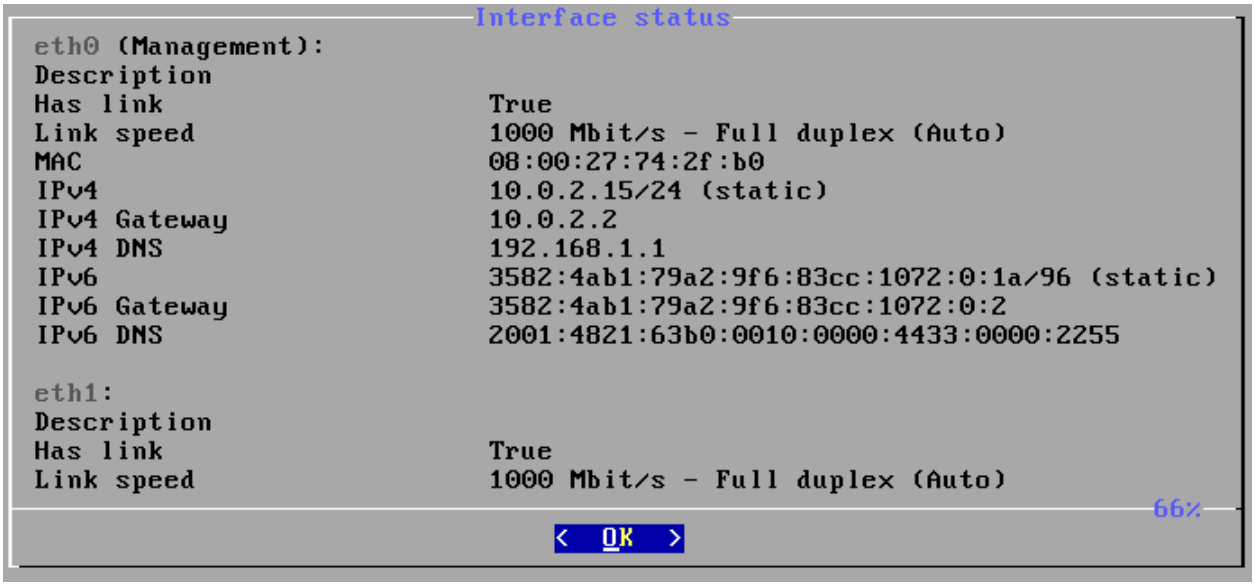


Example 1: “eth0” configured as management interface with no VLAN tag and address type = DHCP for both IPv4 and IPv6.

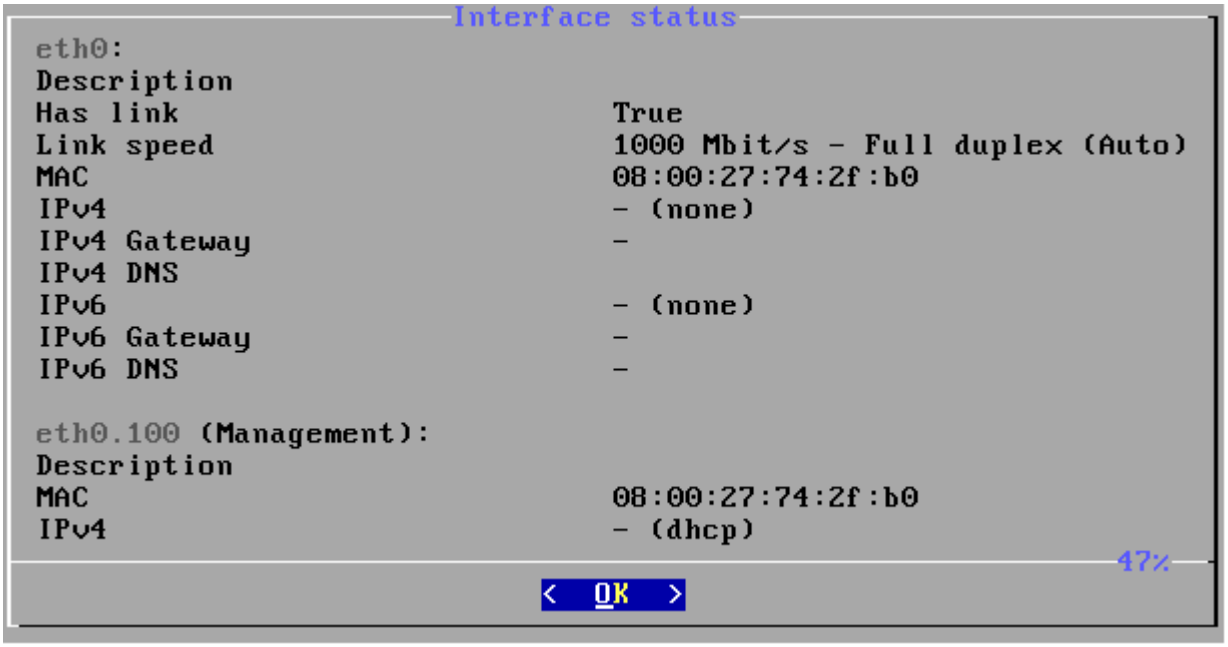


(When a percentage figure appears at bottom right, you can scroll up and down using the Page Up and Page Down keys, or their equivalents, to view the full contents of the screen. Only the topmost part is shown here.)

Example 2: “eth0” configured as management interface with no VLAN tag and address type = static for both IPv4 and IPv6.

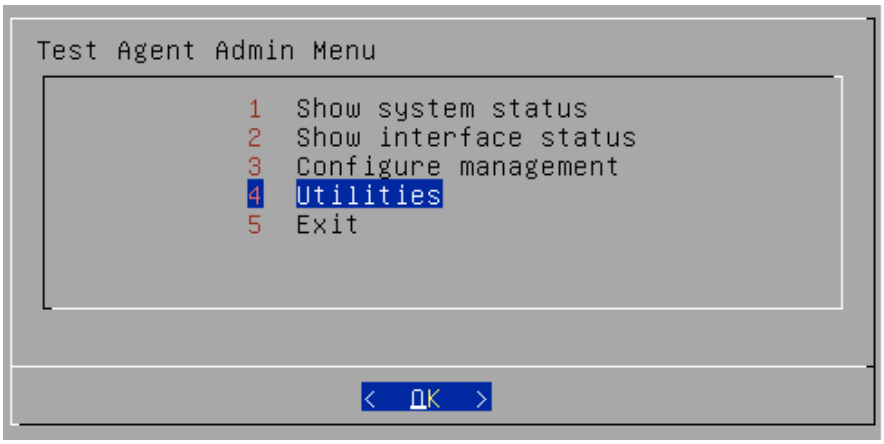


Example 3: “eth0” configured as management interface with VLAN tag and IPv4 address type = DHCP.



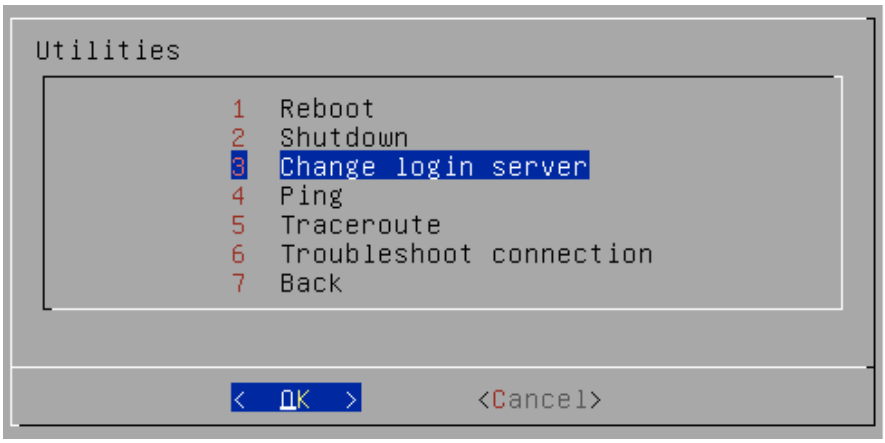
4.4.4.3 Using Test Agent utilities

- To access the Test Agent utilities, navigate to Utilities in the text-based menu.

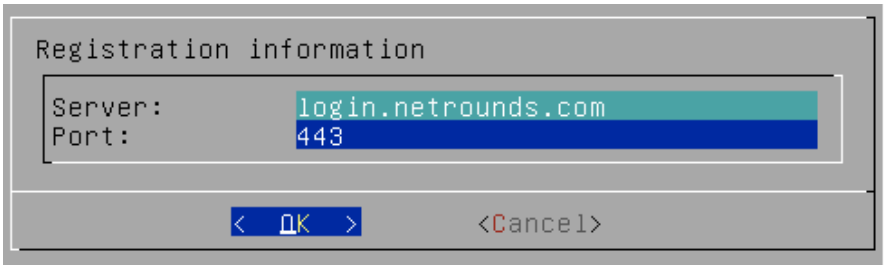


Changing login server

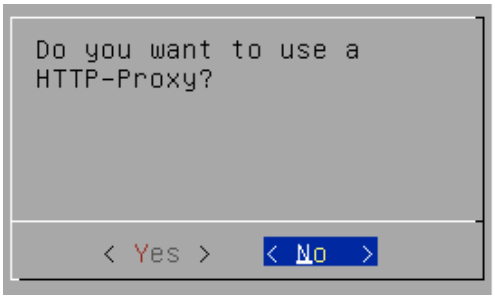
- To use the Routing Active Testing proprietary proxy function or a standard HTTP (web) proxy, navigate to Change login server. The concept of Test Agent proxy is described *here* (page 486).



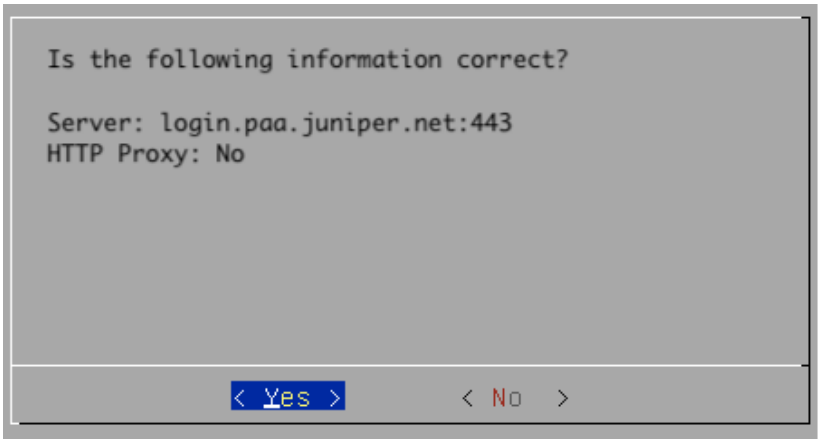
- To be able to use the Test Agent proxy function, you need to change the login server here to point to the static IP address of the proxy Test Agent.



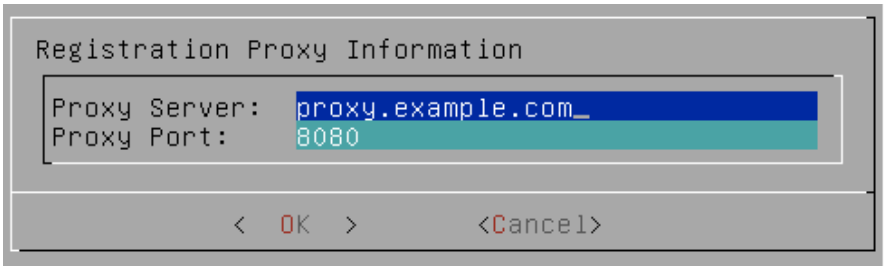
- Select No if you are not using a standard HTTP proxy.



- Check that the settings are correct, then select Yes.

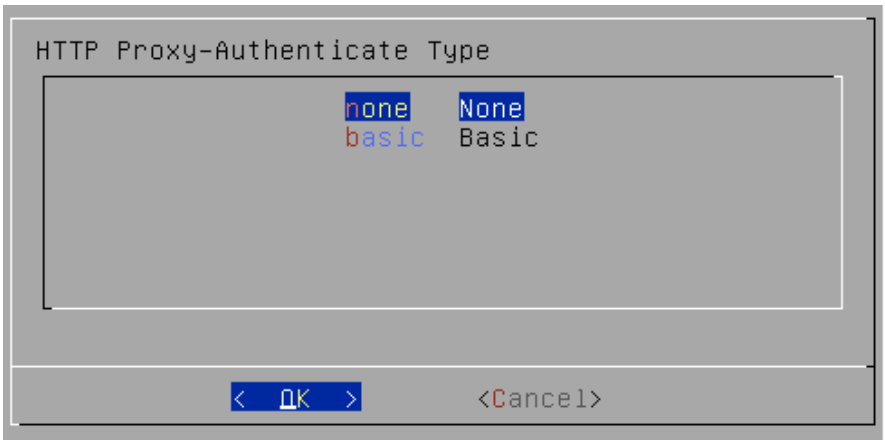


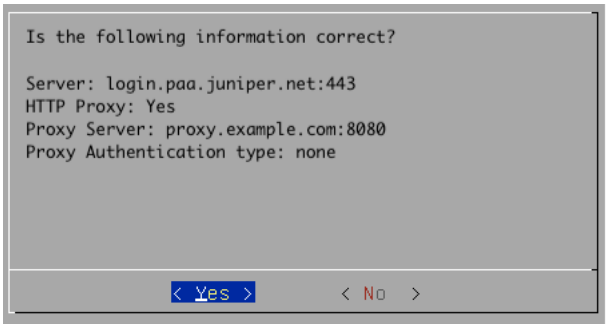
- To use an HTTP proxy, answer Yes to the question “Do you want to use an HTTP proxy?” above, and proceed to fill in the proxy address.



- For HTTP proxy authentication, the options are none and basic. Make the appropriate selection.

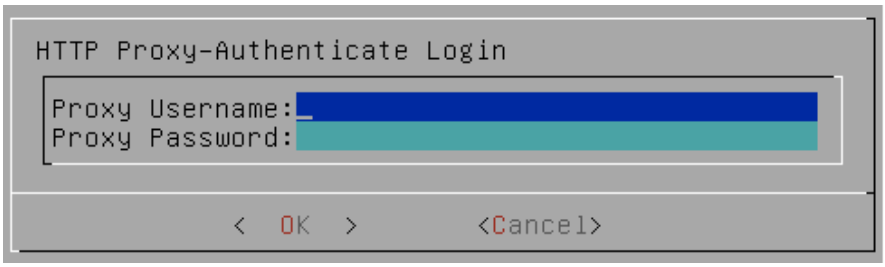
No authentication

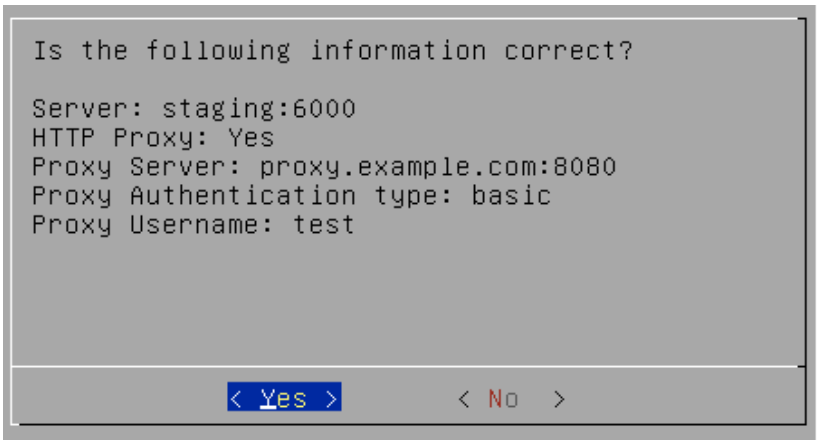




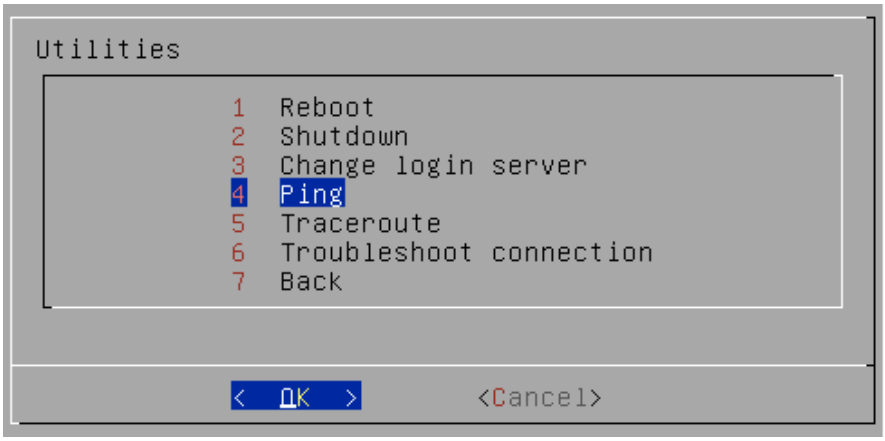
Basic authentication

- Fill in the requested credentials.

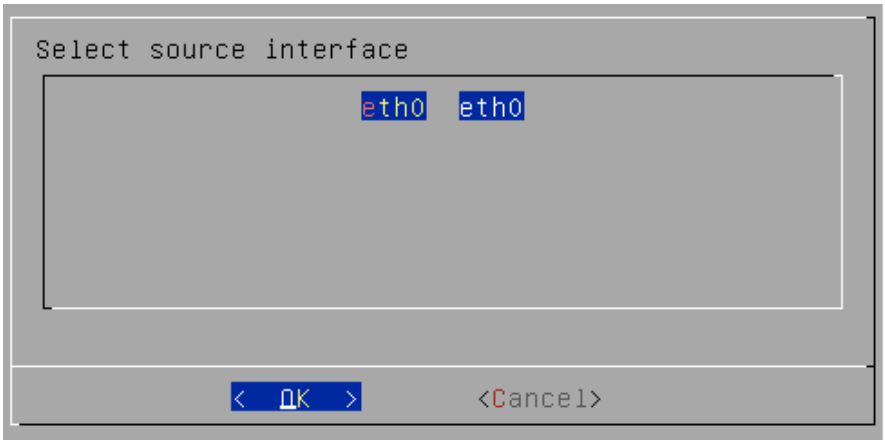


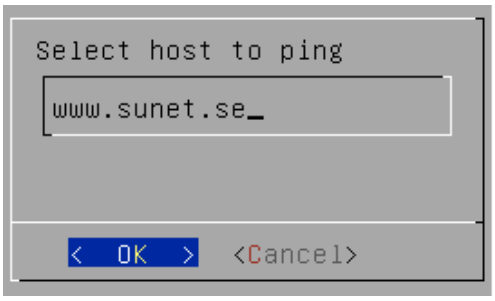


Ping

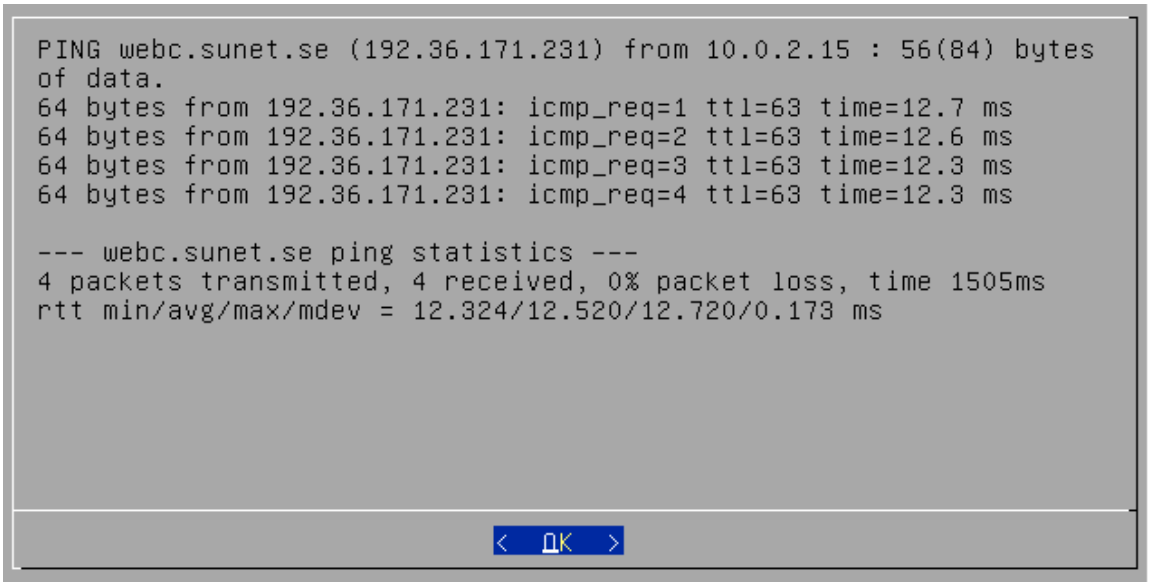


- Select interface and enter the host to ping.

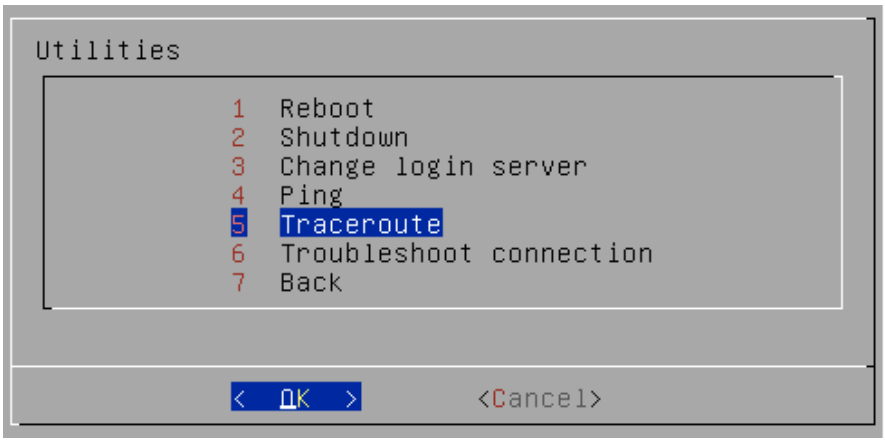




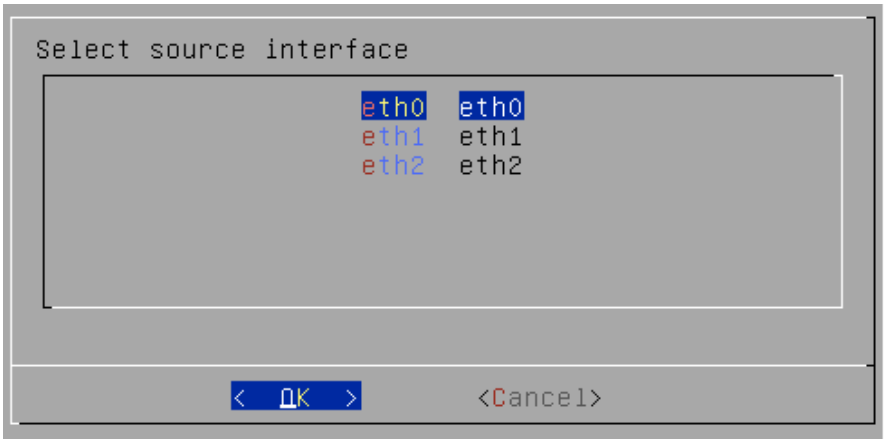
The Ping session is displayed as follows:

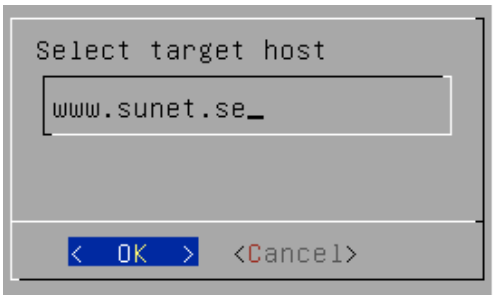


Traceroute

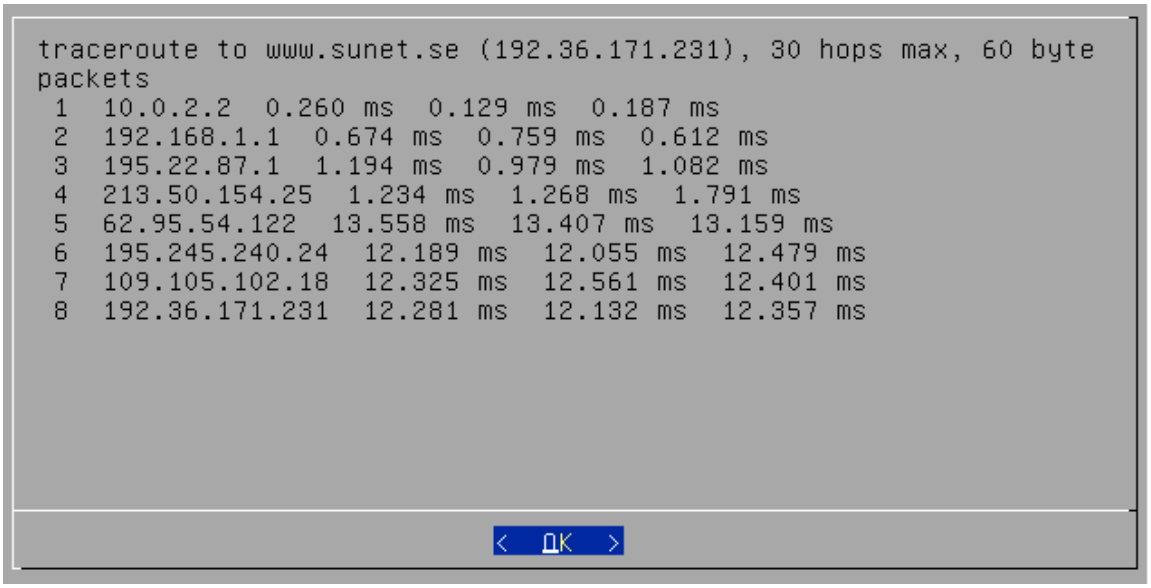


- Select interface and enter the host to traceroute.



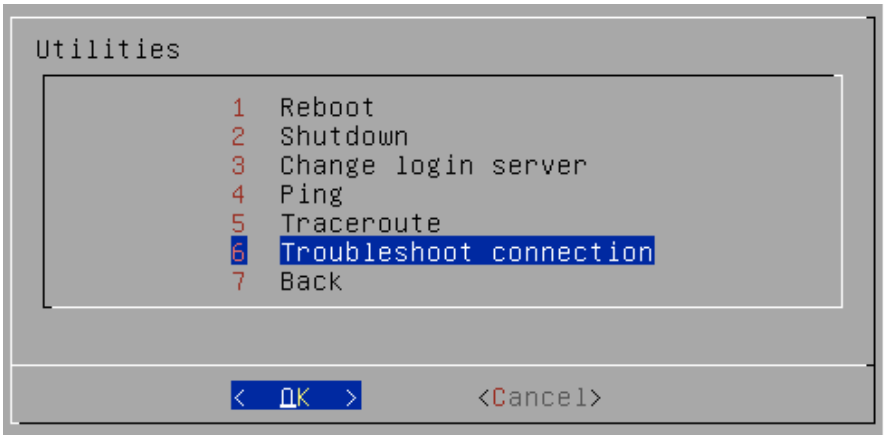


The Traceroute session is displayed as follows:

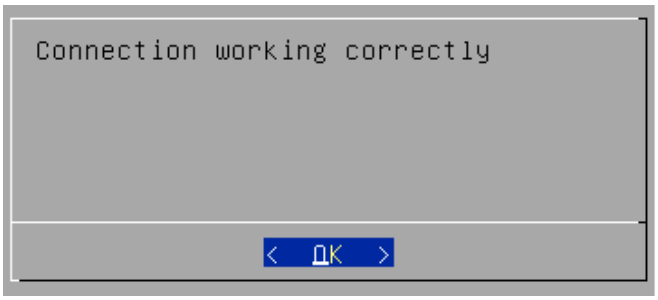


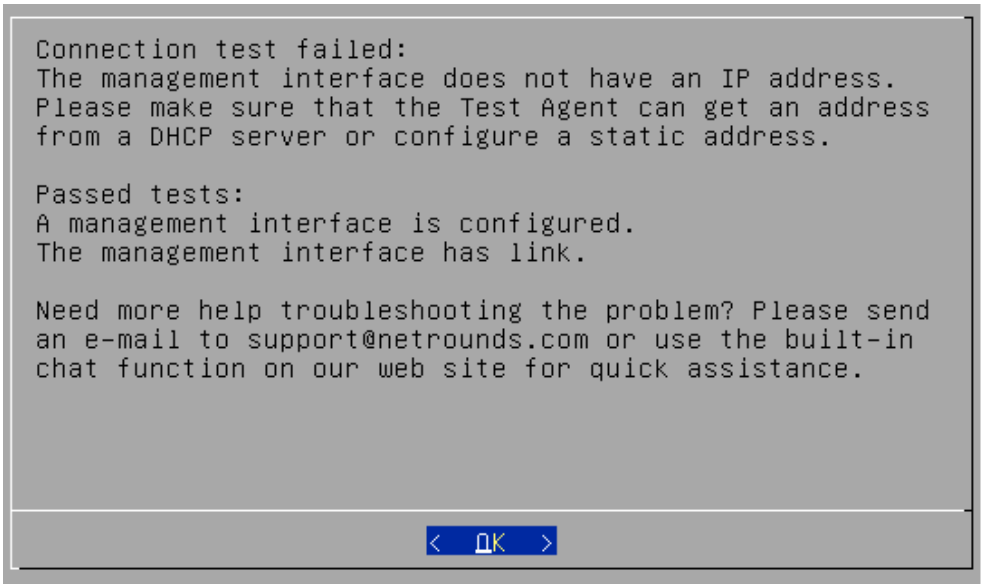
Troubleshoot connection

This function tests the management connection to verify that it is working.



The outcome is a pass/fail test result. An example of each is shown in the screenshots below.



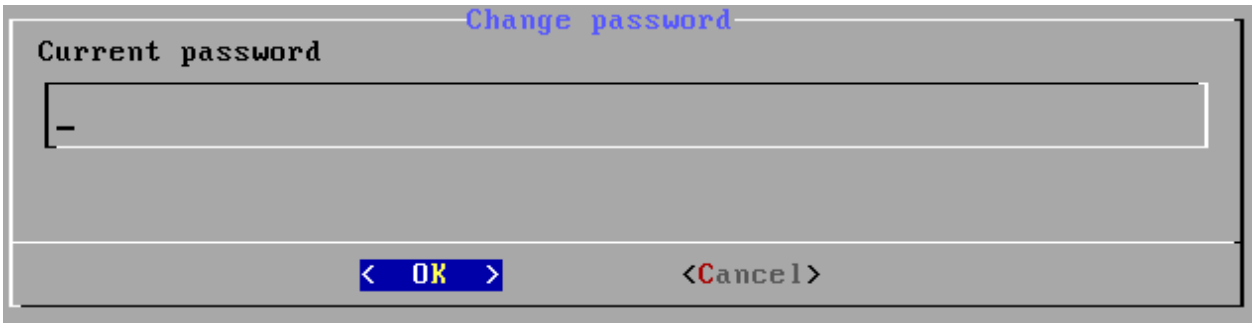


Change admin password

Here you can change the Test Agent admin password from the default (which is “admin”).

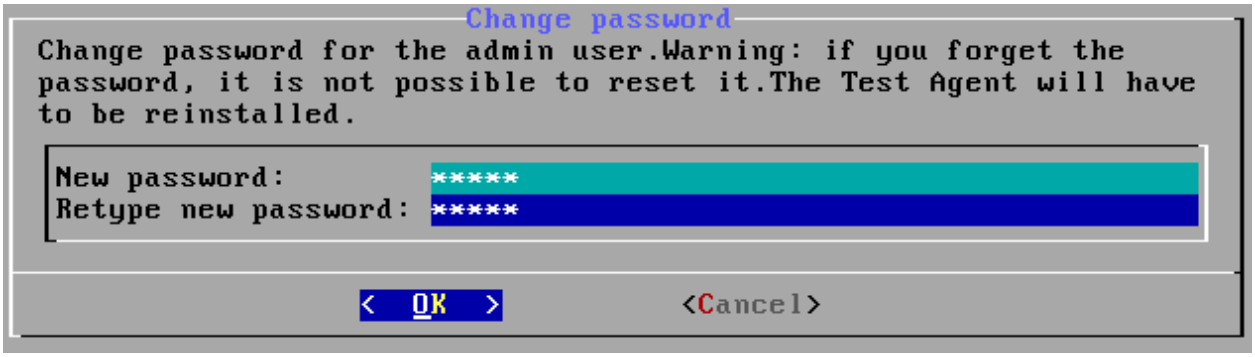


- Type the current password.

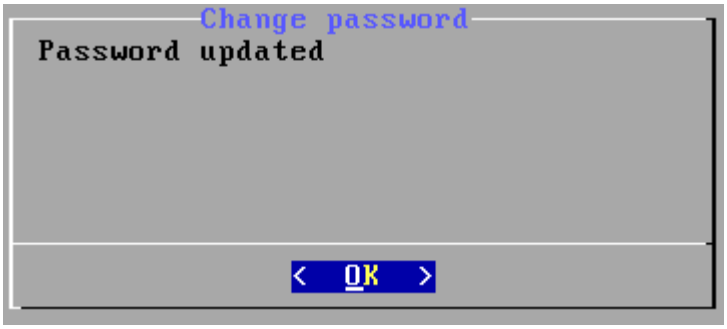


- Type your new password twice, then select OK.

Warning: Take care not to forget this password, as it cannot be reset. If the password has been lost, the only way to make the Test Agent operational again is to reinstall it.



Provided that the two entries match, the admin password will be changed and a confirmation will be given as shown below.



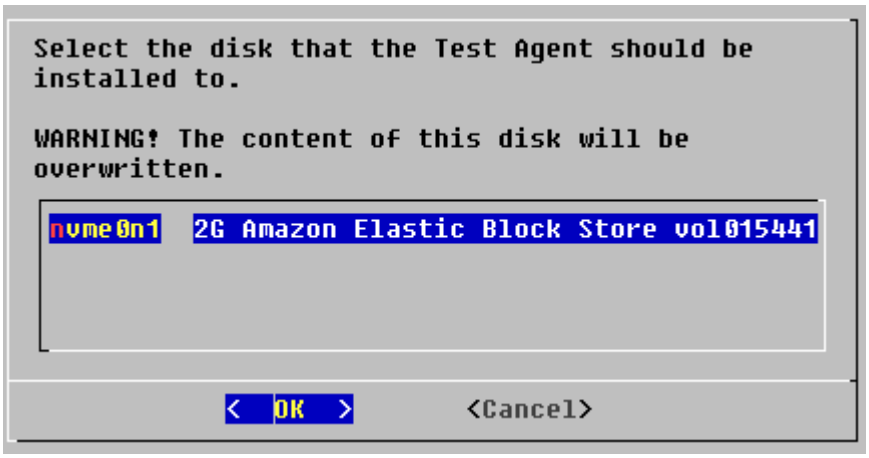
Install to disk

This utility lets you install the Test Agent to a different disk.



A list of available disks is displayed.

Warning: If you go ahead with the installation, any content on the selected disk will be overwritten.

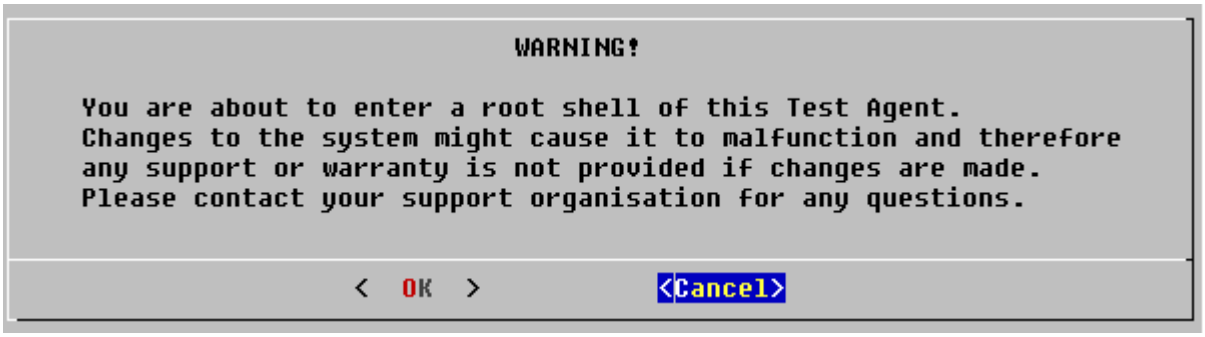


Root shell

This option allows you to enter a root shell of the Test Agent.



Warning: Be aware that changes made in the root shell may cause the Test Agent to malfunction.

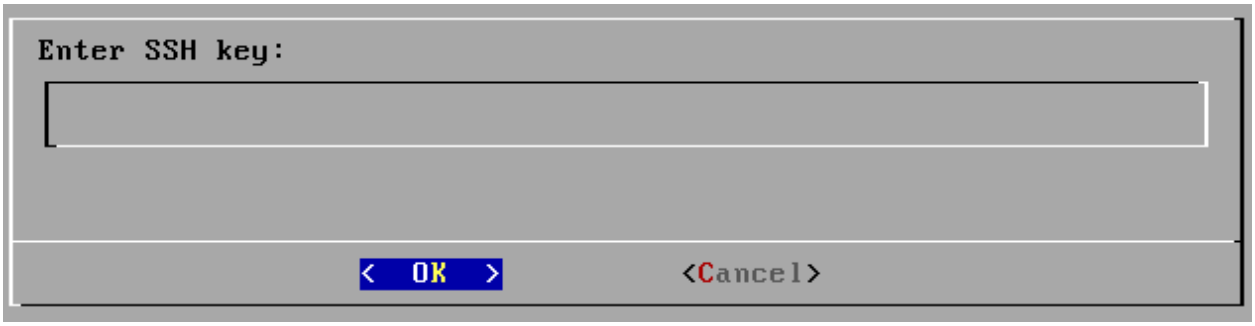


Manage SSH authorized keys

Here you can add SSH public keys to the Test Agent. Using the corresponding private key you can then log in to the Test Agent via SSH.

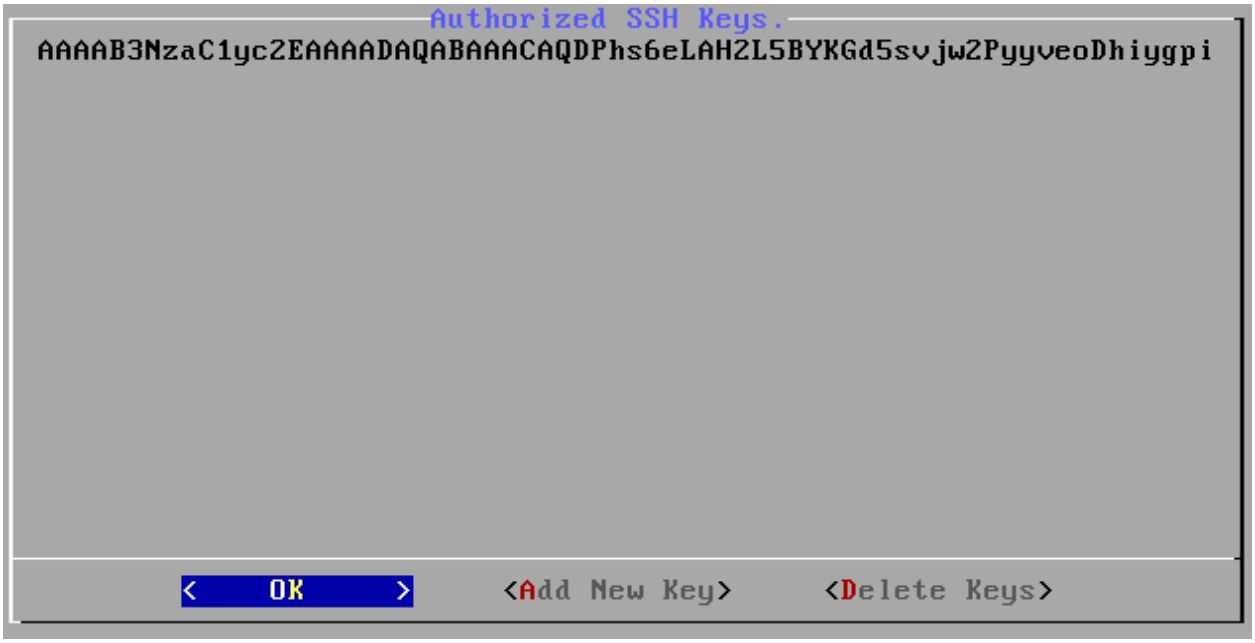


- When first selecting this menu item, you are asked whether to add a key. Select Yes.
- An example of how to generate an SSH key pair is given *here* (page 200).
- Enter the SSH key.

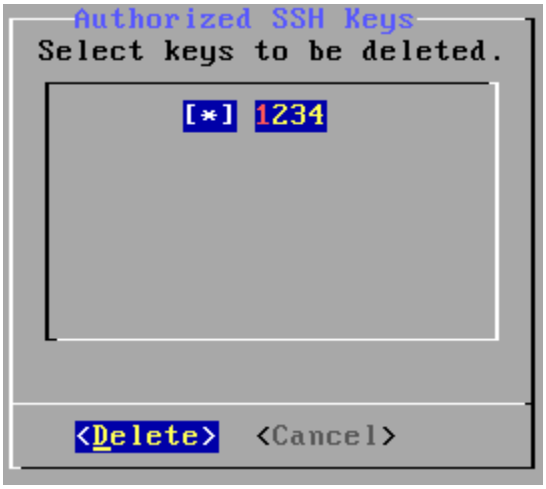


Note: The SSH key must not have any options specified, nor any leading or trailing whitespace such as spaces or newlines. Any such keys will be rejected by the Test Agent.

The dialog will now display a list of keys added (each key is truncated at the end of the line). Once a key has been installed on the Test Agent, SSH is automatically activated on that Test Agent.



There are also options to add another key or delete keys. In the latter dialog, press Space to select the keys you want to delete:



If you delete all SSH keys on a Test Agent, SSH is automatically deactivated on that Test Agent.

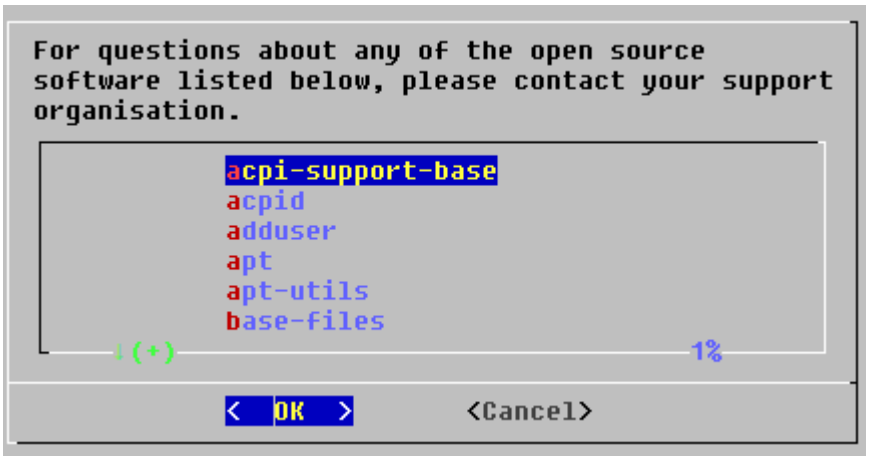
SSH keys can alternatively be pushed to Test Agents from Control Center; see *this page* (page 199). Note that keys added from the local console *cannot* be managed from Control Center but can only be displayed there.

Licenses

Here is displayed license information for open-source software used by Test Agents.



The software components are listed in alphabetical order. Select a component to view its license information.

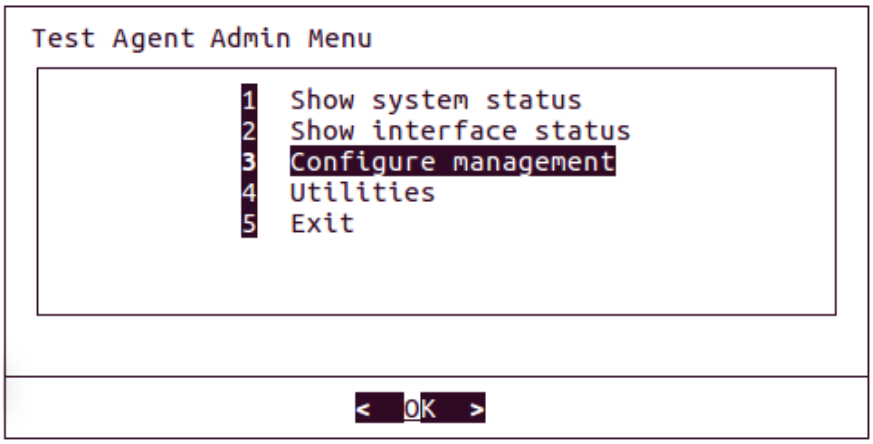


4.4.5 Configuring management over a mobile interface on a Test Agent

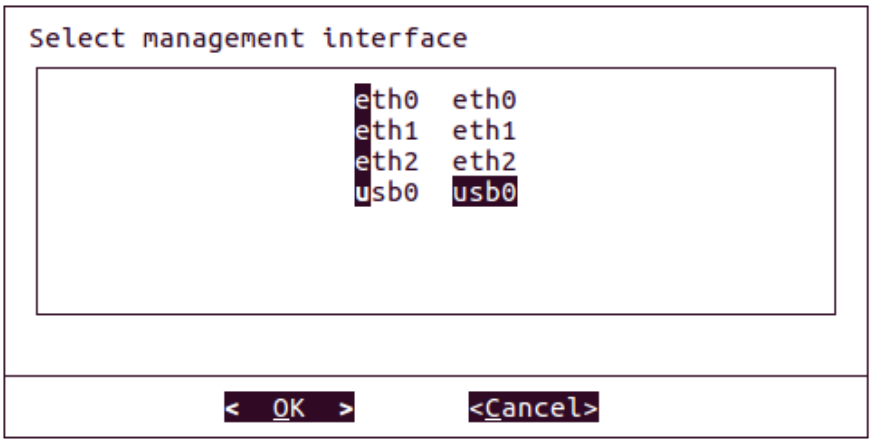
This page describes configuring Test Agent management over a mobile interface. This is different from configuring management over an Ethernet interface, described on *this page* (page 209).

Note: Management over a mobile interface is available only for preinstalled Test Agents built on HW Medium hardware and equipped with an LTE chip (“HW Medium Mobile”). Such Test Agents are no longer available for purchase.

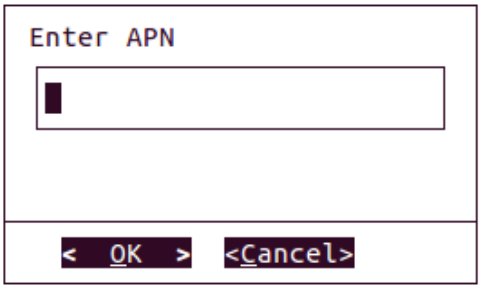
- In the text-based menu, navigate to Configure management.



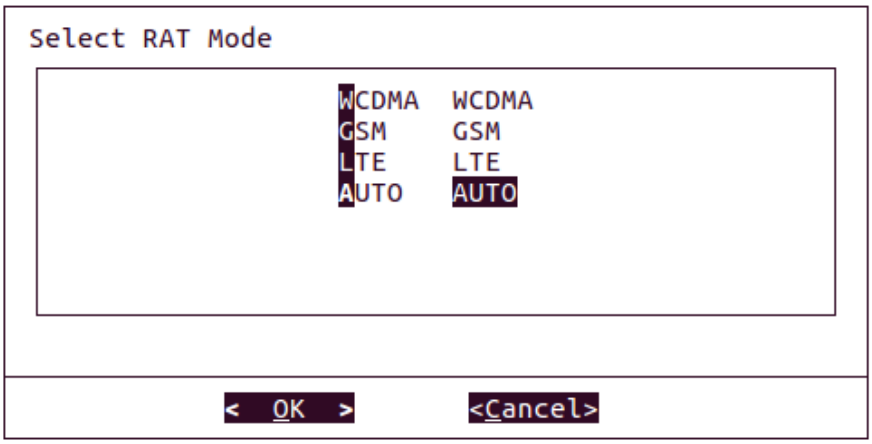
- Select which interface should handle management traffic to and from the server (here, “usb0”). Configuration of the other interfaces is done via the Routing Active Testing server; see *Test Agent interface configuration* (page 166).



- Enter the **APN** (Access Point Name) for the mobile subscription you are using.



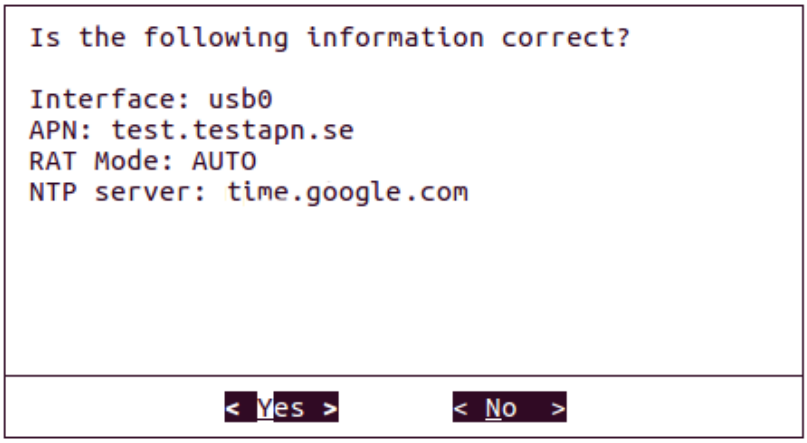
- Select the **RAT** (Radio Access Technology) to be used over the radio interface.



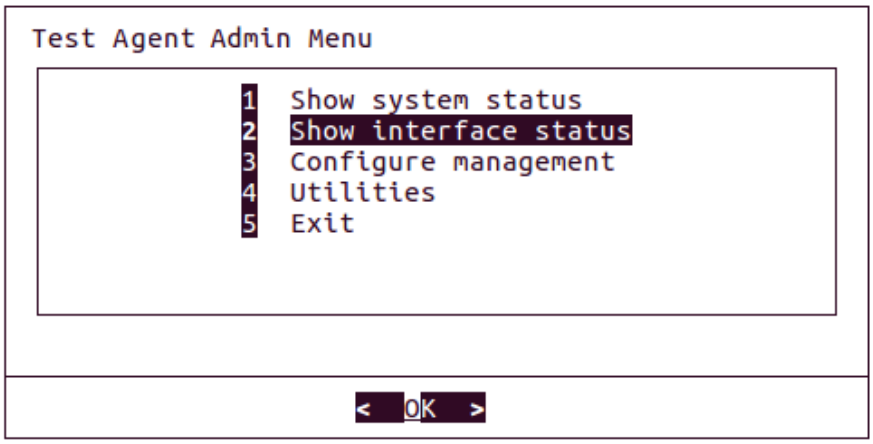
- The NTP server setting follows. You can use either the default NTP server, time.google.com, or a local one (specified as an IP address or host name).



- Finally, confirm all settings for the management port.



- To check the configuration, navigate to Show interface status.



4.4.6 Resetting a Test Agent to its default configuration

This page describes how to revert the Test Agent configuration to the factory default, i.e. perform a software reset. As the Test Agents do not have a hardware button for factory reset, you need to do the reset according to this step-by-step guide.

All steps are done via the local console unless otherwise indicated. The fine points of navigating the local console are covered *here* (page 209).

4.4.6.1 Resetting the interface configuration

- In the text-based menu, navigate to Configure management.
- Select “eth0” as management interface and AUTO for interface speed.
- When asked whether to use VLAN, select No.
- Select DHCP as address type.
- Use the default NTP server setting, time.google.com.
- Finally, confirm the settings for the management port.

All interfaces are now reset to the factory default, with management on eth0 using DHCP.

4.4.6.2 Resetting the NTP server

- To reset the NTP server to the Routing Active Testing default, you have to factory reset the interface configuration. Follow the instructions above.
- Alternatively, you can do this from the Web user interface. In the Test Agents view, click the relevant Test Agent and navigate to the *NTP tab* (page 189) of its configuration dialog. Click Restore defaults.

4.4.6.3 Restoring the login server and the HTTP proxy

(Here, the Routing Active Testing server is assumed to reside in the public cloud.)

- Navigate to Change login server.
- Under Server, enter "https://login.paa.juniper.net". Under Port, enter 443 for a Test Agent Appliance, and 6800 (the default port) for a Test Agent Application.
- When asked whether to use a standard HTTP proxy, select No.
- Finally, confirm your settings.

4.5 Further technical information on Test Agents

4.5.1 Test Agent hardware specifications

These specifications concern hardware units that are no longer available for purchase but are still supported in Routing Active Testing. The hardware intended as a replacement for these is the Juniper NFX150 Network Services Platform and Juniper ACX routers.

4.5.1.1 HW Small

This is a small, portable device with three Gigabit Ethernet ports. It has no fan or other moving parts and is therefore easy to move around and to deploy basically anywhere, even at an end-user site.

See *this page* (page 239) for a description of the front panel LEDs on the HW Small device.

Operating system	Debian distribution with real-time Linux kernel
Processor	1 × AMD Geode LX 800, 500 MHz
CPU clock speed	500 MHz (single)
RAM	256 MB
Storage	2 GB CF (flash)
Network interfaces	3 × 10BASE-T/100BASE-TX (RJ45)
Ports	1 × serial RS-232 + 2 × USB
Voltage	18 V (6 W), 230 V network adapter included, 1.8 m (6 ft) cord
Dimensions (H × D × W)	30 × 157 × 168 mm (1.2 × 6.2 × 6.6 in)
Weight	800 g (1.8 lb)

4.5.1.2 HW Medium

This is a small, portable device with three Gigabit Ethernet ports. It has no fan or other moving parts and is therefore easy to move around and to deploy basically anywhere, even at an end-user site.

This hardware also exists in a variant "HW Medium Mobile" which is additionally equipped with an LTE chip (Huawei ME909s mPCI card) for mobile network connectivity. It is otherwise identical to the regular HW Medium hardware.

Operating system	Debian distribution with real-time Linux kernel
Processor	1 × AMD Bobcat T40E, 1 GHz dual core
RAM	2 GB DDR3
Storage	16 GB SSD
Network interfaces	3 × 10BASE-T/100BASE-TX/1000 BASE-T (RJ45)
Ports	1 × serial RS-232 + 2 × USB
Voltage	12 V DC (6–12 W), 230 V network adapter included, 1.5 m (5 ft) cord
Dimensions (H × D × W)	30 × 157 × 168 mm (1.2 × 6.2 × 6.6 in)
Weight	800 g (1.8 lb)

4.5.1.3 HW Medium Plus

This is a mini rack server with four Gigabit Ethernet ports, with options for more Ethernet ports. It is most commonly placed in the distribution or core network, or in the data center.

Operating system	Debian distribution with real-time Linux kernel
Processor	1 × Intel Core i3, 2.9 GHz dual-core
RAM	4 GB
Storage	2 TB HDD
Network interfaces	4 × 10BASE-T/100BASE-TX/1000BASE-T (RJ45)
	2 × 10BASE-T/100BASE-TX/1000BASE-T (RJ45, old versions)
	Optional:
	4 × 10BASE-T/100BASE-TX/1000BASE-T (RJ45)
Ports	1 × VGA, 2 × USB
Voltage	220 V
Dimensions (H × D × W)	43 × 290 × 437 mm (1.7 × 11.4 × 17.2 in)
Weight	5 kg (11 lb)
Operating temperature range	5°C … 35°C
Non-operating temperature range	–40°C … 60°C
Operating relative humidity range	8% … 90% (non-condensing)
Non-operating relative humidity range	5% … 95% (non-condensing)

4.5.1.4 HW Large

This is a mini rack server with four Gigabit Ethernet ports and two 10G SFP+ ports, with options for more Ethernet ports. It is most commonly placed in the distribution or core network, or in the data center.

Operating system	Debian distribution with real-time Linux kernel
Processor	1 × Intel Xeon E3, 3.2 GHz quad-core
RAM	4 GB
Storage	2 TB HDD
Network interfaces	4 × 10BASE-T/100BASE-TX/1000BASE-T (RJ45)
	2 × 10BASE-T/100BASE-TX/1000BASE-T (RJ45, old versions)
	Optional:
	4 × 10BASE-T/100BASE-TX/1000BASE-T (RJ45)
	2 × 10GBASE-SR or 10GBASE-LR (SFP+)
	4 × 10GBASE-SR or 10GBASE-LR (SFP+)
Ports	1 × VGA, 2 × USB
Voltage	220 V
Dimensions (H × D × W)	43 × 290 × 437 mm (1.7 × 11.4 × 17.2 in)
Weight	5 kg (11 lb)
Operating temperature range	5°C … 35°C
Non-operating temperature range	–40°C … 60°C
Operating relative humidity range	8% … 90% (non-condensing)
Non-operating relative humidity range	5% … 95% (non-condensing)

4.5.2 Preinstalled Test Agent performance characteristics

Below is an overview of performance characteristics of the various hardware models on which Test Agents could be preinstalled in pre-3.0 versions of Routing Active Testing. (This hardware is still supported, but is no longer offered for sale.)

Performance numbers for Test Agents installed on Juniper Networks® NFX150 are found on *this page* (page 237).

4.5.2.1 UDP performance

HW Medium, unidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 2 flows
64	27	40
1518	860	949
9018	801	900

HW Medium, bidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 2 flows
64	14	20
1518	430	474
9018	400	450

HW Medium Plus, unidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 2 flows
64	209	209
1518	937	931
9018	947	941

HW Medium Plus, bidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 2 flows
64	104	104
1518	468	466
9018	473	471

HW Large (1G), unidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 4 flows
64	316	486
1518	937	937
9018	947	941

HW Large (1G), bidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 4 flows
64	158	243
1518	468	468
9018	473	471

HW Large (10G), unidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 4 flows
64	350	500
1518	8599	9800
9018	8910	8910

HW Large (10G), bidirectional

Frame size (bytes)	Rate (Mbit/s), 1 flow	Rate (Mbit/s), 4 flows
64	175	250
1518	4300	4900
9018	4455	4455

4.5.2.2 Other performance numbers

Performance category	HW Small	HW Medium	HW Medium Plus	HW Large
Unidirectional TCP, multiple sessions	Line rate	Line rate	Line rate	Line rate
	100 Mbit/s	1000 Mbit/s	1000 Mbit/s	10 Gbit/s
IPTV MPEG measurement	90 Mbit/s	300 Mbit/s	900 Mbit/s	3 Gbit/s
Equivalent number of SD channels at 4 Mbit/s	22	75	225	750
Traffic generation performance (packets per second)	19,500 pps	78,000 pps	390,000 pps	390,000 pps
Concurrent TCP sessions	500	500	1000	2000
Accuracy	About 1 ms	About 1 ms	About 1 ms	About 1 ms

4.5.3 Performance of Test Agent running on Juniper Networks® NFX150

This page describes expected performance for a Test Agent running as a VNF on an NFX150-C-S1 using one pinned CPU and 1 GB memory. The precision for latency measurements is expected to be better than 1 ms for 99.97% of the time in this configuration.

4.5.3.1 Network performance: UDP (single-stream)

Frame size (bytes)	Direction	Rate (Mbit/s)
64	Bidirectional	30
64	Down	30
64	Up	30
1518	Bidirectional	1500
1518	Down	3000
1518	Up	3000
9018	Bidirectional	5000
9018	Down	9900
9018	Up	9900

4.5.3.2 Network performance: TCP

Single-stream

Frame size (bytes)	Direction	Rate (Mbit/s)
1518	Bidirectional	1500
1518	Down	6000
1518	Up	4000
9018	Bidirectional	4000
9018	Down	9900
9018	Up	9900

4 streams

Frame size (bytes)	Direction	Rate (Mbit/s)
1518	Bidirectional	1000
1518	Down	3000
1518	Up	2000

4.5.3.3 Reflector-based testing: TWAMP sender

Single-stream towards single reflector

Frame size (bytes)	Rate (Mbit/s)
87	15
512	100
9018	100

Multiple-stream towards multiple reflectors

Frame size (bytes)	PPS	No. of streams
87	50	150
87	10	300

4.5.3.4 Reflector-based testing: UDP loopback

Packet size (bytes)	Rate (Mbit/s)
64	10
128	20
256	35
512	75
1024	150
1518	220

4.5.4 Performance of Test Agent running on Juniper Networks® ACX7000 Family routers

This page describes expected performance for a Test Agent running on an ACX7000 Family router.

The precision for latency measurements is normally better than 5 ms, but exceptions may occur for example during routing convergence or under other significant system load.

To ensure that traffic bound for the Routing Active Testing Test Agent doesn't overwhelm the router, this traffic occupies its own DDOS queue, and the bandwidth is limited to 140 Mbit/s for the ACX7100 and the ACX7509 routers and to 40 Mbit/s for the ACX7024 router.

The measurement tasks supported on ACX7000 Family routers are listed *here* (page 58).

4.5.4.1 Network performance: UDP (single-stream, bidirectional)

Router model	Frame size (bytes)	Rate (Mbit/s)	Rate (pps)
ACX7024	64	1	1953
ACX7024	1518	40	3293
ACX7100	64	5	9765
ACX7100	1518	140	11528
ACX7509	64	5	9765
ACX7509	1518	140	11528

4.5.4.2 Network performance: TCP (single-stream, bidirectional)

Router model	Rate (Mbit/s)
ACX7024	40
ACX7100	140
ACX7509	140

4.5.4.3 Concurrent streams

Router model	No. of streams
ACX7024	50
ACX7100	100
ACX7509	100

4.5.5 HW Small hardware LED indicators

This page describes how to interpret the LED indicators on the front panel of the HW Small hardware unit.

LED	Meaning
Left	A green light indicates that the Test Agent hardware is powered on.
Middle	A *steady* green light indicates that the Test Agent is connected to the server. A *flashing* green light indicates that the Test Agent is not connected to the server.
Right	A green light indicates that the Test Agent is busy testing, or that its software is being updated.

No other Test Agent hardware models have LED indicators.

4.5.6 Test Agent Application namespace awareness

Namespaces are a feature of the Linux kernel that partitions kernel resources such that different sets of processes see different sets of resources.

In Routing Active Testing versions prior to 2.36.0, Test Agent Applications ran in one namespace, seeing only the resources in that specific namespace. In order to utilize all resources in the system, it was necessary to install one Test Agent Application in each namespace.

In the current version of Routing Active Testing, the Test Agent Application is equipped with a knowledge of all namespaces and all of their resources, and it can run tests and monitors in any namespace. Therefore, a system with *n* namespaces can now be covered with a single Test Agent Application rather than *n* Test Agents. Note: This requires that the Test Agent Application be started as root.

In the Control Center GUI, namespaces are visible in the names of Test Agent Application interfaces, as detailed *here* (page 167).

4.5.7 Test Agents: Frequently asked questions

4.5.7.1 What is the measurement accuracy of Test Agents?

Thanks to the Test Agent being an appliance, the accuracy of its measurements is very good. A Test Agent has full control of the underlying hardware on which you install it, and the appliance makes sure that no unknown or unnecessary processes are consuming CPU cycles. This results in the highest possible precision in all measurements. The Routing Active Testing implementation also ensures that network packet processing always gets top priority in the CPU.

4.5.7.2 What is the data rate of the control traffic from a Test Agent to the Routing Active Testing server?

The bandwidth of the control traffic to and from the server is typically just a few kbit/s during execution of tests or monitoring sessions. When a Test Agent is in idle mode, there is virtually no traffic towards the Routing Active Testing server.

4.5.7.3 What happens if the management link between a Test Agent and the Routing Active Testing server is lost?

All your Test Agents will continue their activities even if the connection to the Routing Active Testing server is temporarily lost. The Test Agent will store all measurements locally for up to one hour and upload them to the Routing Active Testing server at a later time when the connection is restored. If the connection stays down for more than one hour, all measurements from the time when the connection dropped and onward will be lost.

If you are running a distributed and automated test, it will be prematurely stopped if one of the Test Agents becomes unavailable during the test.

4.5.7.4 What is meant by the message "Time offset is too high" received when I try to start a test or monitoring session?

Test Agents synchronize their internal clocks using NTP (Network Timing Protocol). This is needed for one-way delay measurements.

The sender Test Agent adds a timestamp in the packet, and the receiver Test Agent compares this timestamp to its own time. Thanks to the clocks in the two Test Agent being synchronized, it is possible to calculate the delay.

However, if a Test Agent has for example just been connected and powered on, it may take a few minutes until the clock has acquired the requisite accuracy. In this case, therefore, Routing Active Testing issues a warning rather than allowing the test or monitoring session to start with insufficient accuracy.

This is particularly important for real-time services where delay is important, such as VoIP and videoconferencing. For UDP and SIP, the maximum allowed deviation of the Test Agent's internal clock is 4 ms.

TCP-based services have less stringent timing requirements, so the maximum clock deviation in Routing Active Testing is set to 100 ms in this case.

4.5.7.5 Do Test Agents require calibration?

No calibration is needed for Test Agents. Calibration is most relevant for hardware-based components with extremely high accuracy (in the order of nanoseconds).

4.5.7.6 If I use a USB-based Test Agent on my laptop, will it affect my previous installation?

No. For a USB-based Test Agent, all required software is embedded in the USB flash memory, including the operating system as well as all required test tools.

The USB-based Test Agent uses your laptop hardware only temporarily. The laptop boots based on the contents on the USB, and as long as the laptop remains turned on, your laptop will act as a Test Agent.

Once you remove the USB and restart the laptop using a normal hard disk boot, your laptop will revert to its normal state.

4.5.7.7 Is the USB-based Test Agent dependent on the previous OS on the PC/laptop?

No, it does not matter what operating system you have on your computer prior to using it as a USB-based Test Agent.

When you download and create your bootable USB memory, a custom OS is also included. The computer then boots this OS using your downloaded USB contents. You might even use a USB Test Agent on a PC that does not have a hard disk installed.

4.5.7.8 Is it possible to install a downloadable Test Agent on Apple hardware?

No, this is currently not supported.

5 Plugins

5.1 Description of plugins

5.1.1 Introduction

A plugin consists of software which Test Agents use to collect measurements in a network. As of Routing Active Testing version 3.1.0, these plugins are visible in the Control Center GUI.

Plugins are currently used mainly by Test Agent Applications; on Test Agent Appliances, only the Netflix plugin can currently run. Nonetheless, “Test Agent” will be used as a shorthand in what follows.

Note: If you have followed the installation instructions, Control Center is already equipped with all plugins needed by Test Agents. However, it is also possible to upload further plugins to Control Center, as mentioned in the section *Custom plugins* (page 244) below.

When a Test Agent is configured in a test or monitor to run a particular plugin, the Test Agent will download it from Control Center, then configure it for measurement as specified in the test or monitor. The setup of tests and monitors in the Routing Active Testing GUI still works the same way, and the plugin system integrates seamlessly with it.

5.1.2 Notes on product versions

The concept of plugins was introduced overtly in Routing Active Testing 3.1.0.

In Routing Active Testing versions prior to 3.1.0, the plugins were bundled with the Test Agent Application core; however, from version 3.1.0 onward, the Test Agent downloads all plugins dynamically from Control Center.

5.1.3 How do plugins work?

A plugin comes in a file with the extension `.nap`. This file contains the plugin executable and any dependencies that are required to run it.

When a test or monitor is started, the Test Agent will receive a measurement configuration that tells it to start running the required plugin or plugins (in this paragraph, we assume there is only one). If the Test Agent does not have the plugin locally, it will start by downloading the plugin from Control Center over its management connection. Once the download has completed, the Test Agent will start the plugin executable as a child process, and the plugin will then be assigned its configuration parameters. Finally, the plugin will start producing metrics and events, which are uploaded to Control Center.

Each plugin has an equivalent *task* (for example, TCP) associated with it in the Control Center GUI.

When a task is run as part of a test or monitor, Control Center will split the task configuration into multiple *measurement* configurations, one for each instance of the plugin that should run on the Test Agents specified in the task configuration (that is, one measurement configuration for each Test Agent interface involved).

5.1.4 Plugin versions

Since the plugins are not tied to a particular Control Center version, and since it is possible to upload new plugins at any time, plugins need their own versioning scheme.

Note: The versions of official Routing Active Testing plugins will follow the Control Center version. However, this is not a general requirement.

This also means that a single Control Center can have multiple plugin versions available. However, to keep metrics consistent within each account, only a single version of a given plugin can be enabled in an account at a time. This also lets the account admin pick the plugin version to use for all Test Agents and decide when to upgrade the plugins.

5.1.4.1 Setting the enabled plugin version for an account

To edit the current enabled plugin version, use the plugin CLI in the Control Center shell:

```
export ACCOUNT_SHORT_NAME=myaccount
export PLUGIN_VERSION=http@3.1.1

ncc plugins edit enabled-version --account $ACCOUNT_SHORT_NAME $PLUGIN_VERSION
```

where `$ACCOUNT_NAME` is the account short name and `$PLUGIN_VERSION` contains the plugin name (here, “http”) and version separated by a `@` delimiter.

For further information about this command, type

```
ncc plugins edit enabled-version --help
```

You can also do bulk update operations applying to all plugins and/or all accounts:

This command updates a specified plugin to its latest version for all accounts in the database:

```
ncc plugins edit enabled-version --all-accounts $PLUGIN_VERSION
```

This command updates all plugins to their latest versions for all accounts in the database:

```
ncc plugins edit enabled-version --all-accounts --all-plugins
```

5.1.4.2 Plugin platforms

In addition to multiple plugin versions, Routing Active Testing has support for multiple plugin *platforms* for each version.

A `.nap` plugin package is specific to a single platform, for example x86_64. This means that a single plugin version can consist of multiple `.nap` packages, one for each platform. Below is an example of what the hierarchy may look like:

- Plugin xyz
  - Plugin version 2
    - Platform A
    - Platform B
  - Plugin version 1
    - Platform A

Note that there is no requirement that all plugin versions support all plugin platforms.

5.1.5 Pre-bundled plugins

All plugins for a release of Routing Active Testing are found in a separate Debian package that will have been installed in Control Center if the instructions in the Installation Guide have been followed.

This means that in a default installation, you do not need to upload any plugins manually.

5.1.6 Custom plugins

It is also possible to upload new plugins that are not tied to a particular release of Control Center or of Test Agents.

Currently all plugins are written by Juniper. The company's aim is to authorize partners and other external parties to write plugins; however, this will be realized at a later date.

5.1.6.1 Uploading custom plugins

Again, please note that you do not normally need to upload any plugins as all the plugins you need are installed along with Control Center.

However, if you receive a new plugin version from Juniper that is not part of the normal release cycle, you will need to upload it to Control Center.

To upload a plugin, you must be a superuser or have access to the Control Center shell so that you can use the CLI.

As mentioned *above* (page 244), the `.nap` file for the plugin contains a single plugin platform. This means that you will need to upload one file for each platform you want to use.

Use this command:

```
sudo -u netrounds ncc plugins upload $FILE
```

where `$FILE` is the `.nap` file to upload.

Note: The command requires `sudo` in order to successfully store the plugin file on disk in the `plugins` directory.

For further information about using the above command, type

```
ncc plugins --help
```

5.1.6.2 Enabling a plugin

To enable a separately uploaded plugin, set its version to the current enabled version as mentioned *above* (page 243).

Note: **Important:** When a separately uploaded plugin is enabled, its corresponding task appears not in its usual place in the GUI but in the category Dynamic plugins. See *this page* (page 471) for more information.

5.1.6.3 Disabling a plugin

Note: You can only disable a plugin that has been uploaded separately. Plugins that are preinstalled with Control Center cannot be disabled.

To disable a plugin in all accounts:

```
ncc plugins edit disable myplugin --all-accounts
```

To disable a plugin in a specific set of accounts:

```
ncc plugins edit disable myplugin --accounts account_a,account_b
```

6 Licensing

6.1 Licensing and streams in Routing Active Testing

This page describes how the licensing and its enforcement work for Routing Active Testing accounts. It also describes the *stream* concept and how it is applied in the context of licensing.

6.1.1 Licensing of Test Agents

Each Test Agent must be assigned a *license* before it can be used. To see whether a Test Agent has a license, proceed as follows:

- Click Test Agents on the main menu.
- Locate the Test Agent of interest in the list, and click it.
- Go to the License tab.
- If the Test Agent already has a license, this will be indicated here. If the Test Agent does not have a license, you need to assign one. See *here* (page 66) for details.

A Test Agent is activated when it is first registered with your Routing Active Testing account. Deactivating a Test Agent from your account must be done manually.

The total number of Test Agents registered to a Routing Active Testing account can never exceed the total number of licenses permitted in the account. This is to avoid a scenario where a large number of Test Agents are registered, but where the number of purchased/available licenses is low. In other words, if you try to register a Test Agent to your account, but you do not have a free license to assign to it, the registration will not succeed.

For nearly all task types, the license allows an arbitrary number of Test Agents to concurrently run the task type in question.

6.1.2 Streams

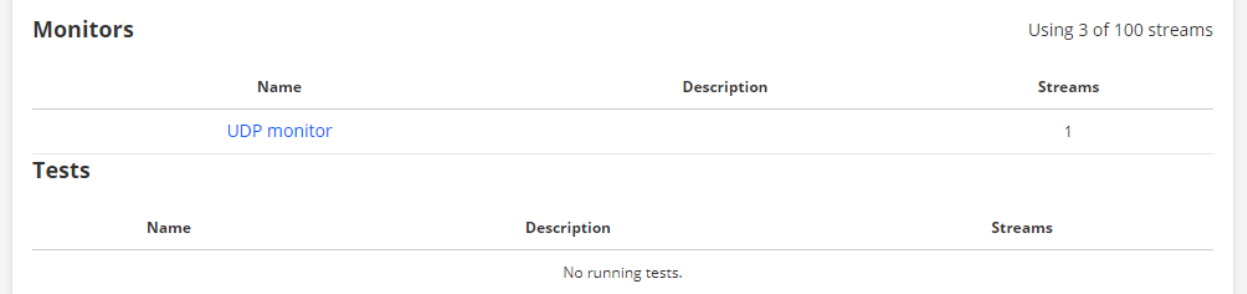
The concept of *stream* in Routing Active Testing is an abstract one used for licensing purposes. It is basically something which a Test Agent consumes in executing tests and monitors.

Routing Active Testing offers several types of license, each relating to streams in its own way:

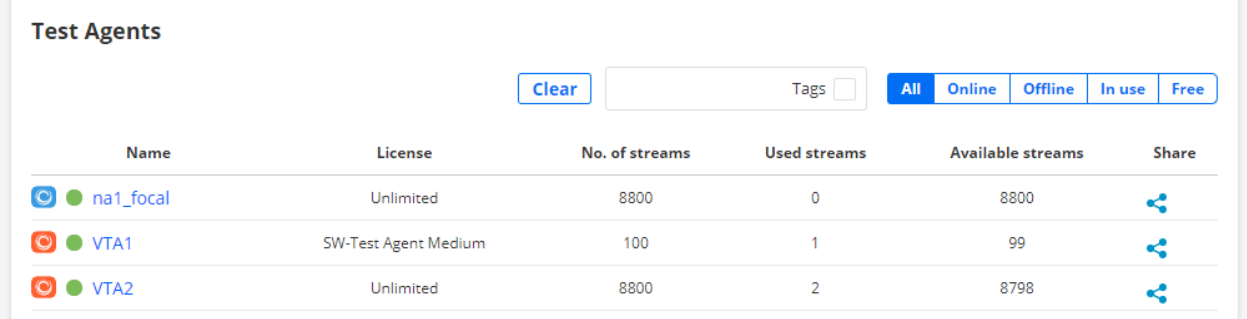
- One license type grants the right to concurrently use a given number of streams in an instance of **Control Center**. This license is relevant for on-premise customers. The license will enforce that the total number of streams for all tests and monitors does not exceed the limit. If that happens, a warning will be raised, and the test or monitor will not be started.
- Another license type caps the number of streams for each Control Center instance, while also limiting the number of streams for each **account** in a Control Center. This license is relevant for both SaaS and on-premise customers.
- A third license type specifies the number of streams that are possible to run on a **single Test Agent**. There are different options here, including an “unlimited” license, which does not limit the number of streams at all.

There are two ways to inspect how many streams your Test Agents are consuming, and what they are being used for:

- *Method 1:* Navigate to the Streams tab for a Test Agent. For each test, monitor, and application that the Test Agent is executing, the number of streams consumed is shown.



- *Method 2:* In the Test Agents view, click the tab named License info.



How the execution of various tasks and applications translates into stream consumption is detailed in the following subsections. Here, when talking about streams being "used" or "consumed" we are always referring to licensing, not to actual streams (or connections, sessions, calls) being set up.

For task types that can be part of a monitor, the stream consumption is the same for both monitors and tests.

Links in headings below point to task and application descriptions.

6.1.2.1 Stream consumption: TCP/UDP performance tasks

UDP and TCP

This subsection applies to the basic UDP and TCP tasks.

- Setup type = "Client-Server": For each pair of Test Agent interfaces involved in the task, one stream is used by each Test Agent in the pair. The Direction setting does not matter (that is, whether only one stream A -> B is set up or two streams A -> B, B -> A are set up between the interfaces).
  - *Example:* If three Test Agent interfaces on different Test Agents are selected as Clients, then the Server Test Agent will use 3 streams, and each Client Test Agent will use 1 stream. This holds no matter how Direction is set (Down, Up, or Bidirectional).
- Setup type = "Full-Mesh": With n Clients selected, $n - 1$ streams are used on each Client, so the total number of streams used on all Clients is $n \times (n - 1)$.
  - *Example:* If four Clients take part in the measurement, each Client will use 3 streams, and together they will use a total of 12 streams.

Multicast UDP

The same rules apply as for *regular UDP* (page 247) with Setup type = Client-Server.

Multisession TCP

On both Server and Client, one stream is used, regardless of the number of TCP sessions between Server and Client. The Direction setting does not matter either.

VoIP UDP

- Setup type = “Client-Server”: On the Server, the number of licensing streams consumed equals (number of Clients) × (number of VoIP streams). On each Client, the number of licensing streams consumed equals the number of VoIP streams.
  - *Example:* If the number of Clients is 3 and Number of streams (that is, VoIP streams) = 2, then 6 licensing streams are used on the Server, and 2 licensing streams are used on each Client.
- Setup type = “Full-Mesh”: With n Clients and m VoIP streams, $m \times (n - 1)$ licensing streams are used on each Client, so the total number of licensing streams used on all Clients is $m \times n \times (n - 1)$.
  - *Example:* If four Clients take part in the measurement and Number of streams = 2, then each Client will use $2 \times 3 = 6$ streams, and together they will use a total of $2 \times 4 \times 3 = 24$ streams.

Junos TCP

One stream used for each Junos device that the Client (Test Agent interface) connects to.

Junos UDP

One stream used for each Junos device that the Client (Test Agent interface) connects to.

RFC 6349 TCP throughput test

One stream used for each Test Agent interface taking part, regardless of task settings.

QoS policy profiling

One stream used for each Test Agent interface taking part, regardless of task settings.

6.1.2.2 Stream consumption: IPTV & OTT video

IPTV MPEG

On each Client, one stream is used for each IPTV channel. Whether SPTS (Single Program Transport Stream) or MPTS (Multiple Program Transport Stream) is used to deliver the IPTV channels does not matter.

IPTV MPEG inline

One stream used for each IPTV channel joined by the customer’s set-top box.

OTT - HLS

One stream used on each Client.

IPTV channel zapping

One stream used on each Client, regardless of the number of Channels selected.

Netflix Speedtest

One stream used on each Client, regardless of the number of OCAs engaged.

6.1.2.3 Stream consumption: HTTP and DNS

HTTP

One stream used on each Client (Test Agent interface).

Junos HTTP

One stream used for each Junos device that the Client (Test Agent interface) connects to.

DNS

One stream used on each Client (Test Agent interface).

6.1.2.4 Stream consumption: SIP

- On the Hub, one stream is used for each Client.
- On each Client, one stream is used.

The value of Number of calls per test cycle does not affect stream consumption.

6.1.2.5 Stream consumption: Wi-Fi network testing

Mobile logger

One stream used for each Test Agent interface logged.

Mobile switcher

One stream used on the Test Agent.

Wi-Fi scan

One stream used on the Test Agent.

Wi-Fi logger

One stream used on the Test Agent.

Wi-Fi switcher

One stream used on the Test Agent.

6.1.2.6 Stream consumption: Security testing

One stream used on each of Customer and ISP. (Some security tasks only have a subset of these roles.)

6.1.2.7 Stream consumption: Ethernet service activation testing

One stream each used on Sender and Receiver.

6.1.2.8 Stream consumption: Transparency testing

One stream each used on Sender and Receiver.

6.1.2.9 Stream consumption: Reflector-based testing

- *Y.1731* (page 402) tasks: On each Client, one stream used for each MEP entered.
- *TWAMP/TWAMP Light* (page 416) task: On each Sender, one stream used for each Reflector and Test Agent Reflector entered.
- *TWAMP Reflector* (page 424) task: One stream used for each Reflector (Test Agent interface) entered.
- *Junos TWAMP* (page 425) task: One stream used for each Junos device that the Client (Test Agent interface) connects to.
- *Ping* (page 430) task (Ping to multiple defined hosts): On each Client, one stream used per Host entered.
- *BWPing* (page 433) task: One stream used. (Only one Sender and one Host can be selected in each task.)
- *Junos ICMP* (page 436) task: One stream used for each Junos device that the Client (Test Agent interface) connects to.
- *Path trace* (page 440) task: The number of streams used is equal to the value of the Max TTL parameter.
- *UDP loopback* (page 448) task: One stream used. (Only one Client and one Host can be selected in each task.)

6.1.2.10 Stream consumption: Applications

Speedtest

Enabling Speedtest on a Test Agent causes the Test Agent to use a number of streams equal to Max parallel tests in the Speedtest configuration. Note that the TCP sessions setting does not affect stream use.

Packet capture

- Live packet capture: One stream used on the Test Agent.
- Non-live packet capture: No streams used on the Test Agent; this feature is “free”.

6.1.3 Stream consumption for multi-task tests and monitors

If a test or monitor comprises multiple tasks – whether executed concurrently, or (for tests) sequentially in steps, or a combination of both –, stream use on each Test Agent engaged by that test or monitor is calculated as the sum of the streams used for each task. Those streams always remain in use on the Test Agent for the entire duration of the test (i.e. this also applies for a test where the Test Agent is idle in some steps).

6.1.3.1 Example 1

Test Agent TA1 participates in the first two steps of a test.

- Step 1:
  - TCP, Client-Server: One interface acting as Client (1 stream used)
  - UDP, Client-Server: One interface acting as Client (1 stream used)
- Step 2:
  - IPTV: One interface receiving one channel (1 stream used)

The test also has a third step which does not involve Test Agent TA1.

For the entire duration of the test, Test Agent TA1 will consume 3 streams, from the start of Step 1 up until the end of Step 3.

6.1.3.2 Example 2

Test Agent TA2 participates in a security test with ten steps.

This will consume 10 streams on Test Agent TA2 for the full duration of this test.

6.1.4 Users in Routing Active Testing

A *user* is defined as a named user within a Routing Active Testing account with permission to log in to the application. Each named user is counted towards the user limit for an account, whether the user is currently logged in or not.

The Routing Active Testing licensing model is *not* based on counting logged-in users, so the above has no bearing on Test Agent stream use as described in the *Streams* (page 246) section of this page.

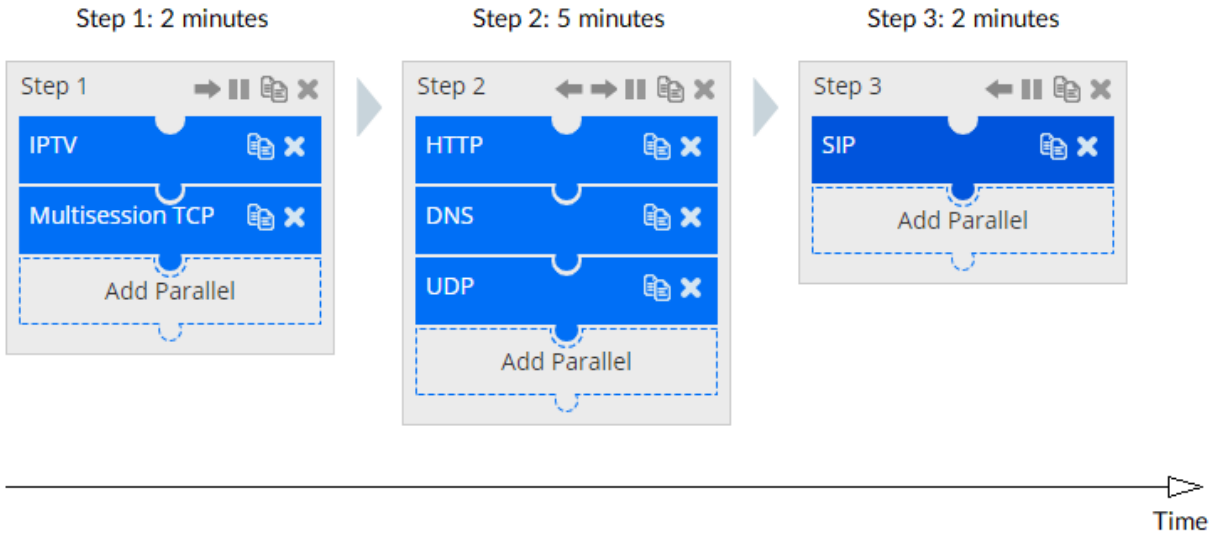
7 Tests and monitors

7.1 Introduction to tests and monitors

Measurements in Routing Active Testing are conducted mainly in *tests* and *monitors*. (Besides these, Test Agents can also run a number of *applications*. They are covered *here* (page 478).)

7.1.1 Definitions

A **test** consists of one or several *steps*, which are executed sequentially. Each step has a specified, finite duration, and entails running one *measurement task* (*task* for short) or multiple tasks concurrently. Both of these properties are illustrated in the example below.

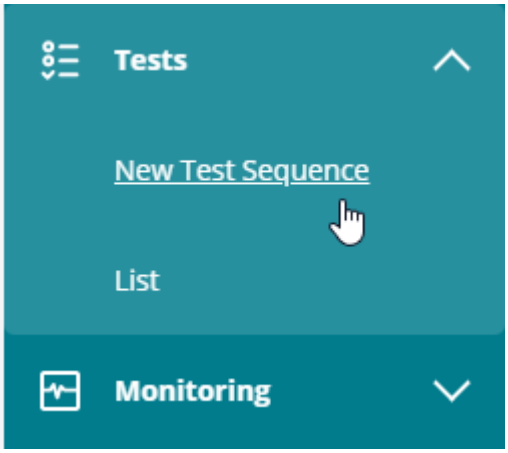


A **monitor** is built up in the same way as one step in a test: multiple *tasks* can be run in parallel. See the picture below for an example. However, a monitor cannot be made up of multiple steps, since a monitoring session has indefinite duration.



7.1.2 Creating tests and monitors

The simplest way to set up a new test or monitor is to click the relevant plus-sign button on the left-side bar holding the main menu:



You are taken to the setup screen for tests or monitors. For further guidance, consult one of these pages:

- *Building tests* (page 254)
- *Building monitors* (page 263)

The above pages use TCP and UDP as examples. Other test and monitoring task types are set up similarly. For advice on parameter settings in individual task types, see the pages dealing with each task type.

7.1.3 Prerequisites

The prerequisites for creating both tests and monitors are an active Routing Active Testing account and a number of registered Test Agents (the precise number depends on the kind of testing to be performed). In some cases, Test Agent Applications can be used; their functionality is however limited. See *this page* (page 57) for full details.

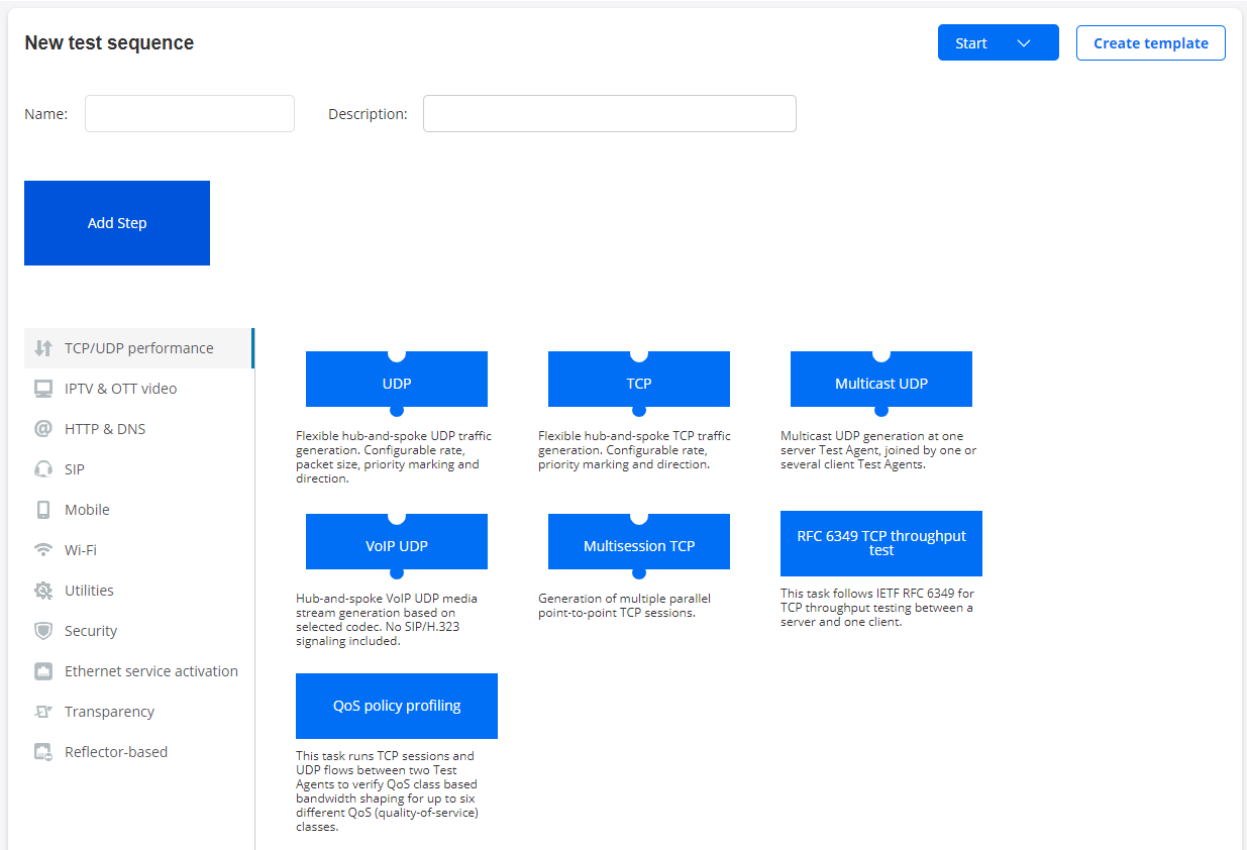
If you haven't yet installed your own Test Agents, please consult the installation guides found *here* (page 70).

7.2 Building tests

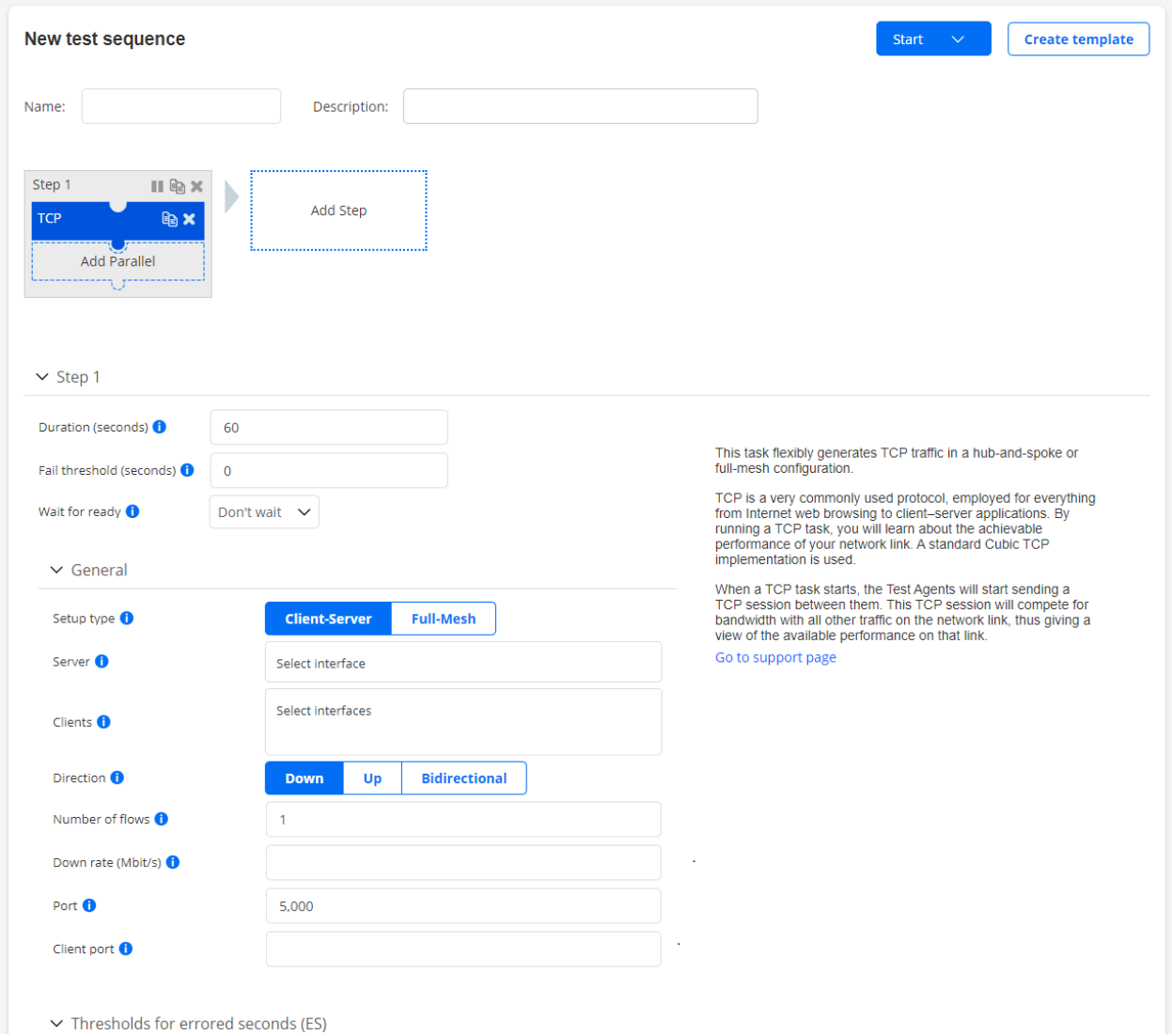
When you create a new test on the New test sequence screen, the test is initially empty.

At the top level, a test consists of a number of *steps* which are executed in sequence. To add a step to a test:

- First select a test task category on the left, for example, TCP/UDP performance.
- Then click the jigsaw puzzle piece or box representing the desired test task. In the example below, TCP is selected.



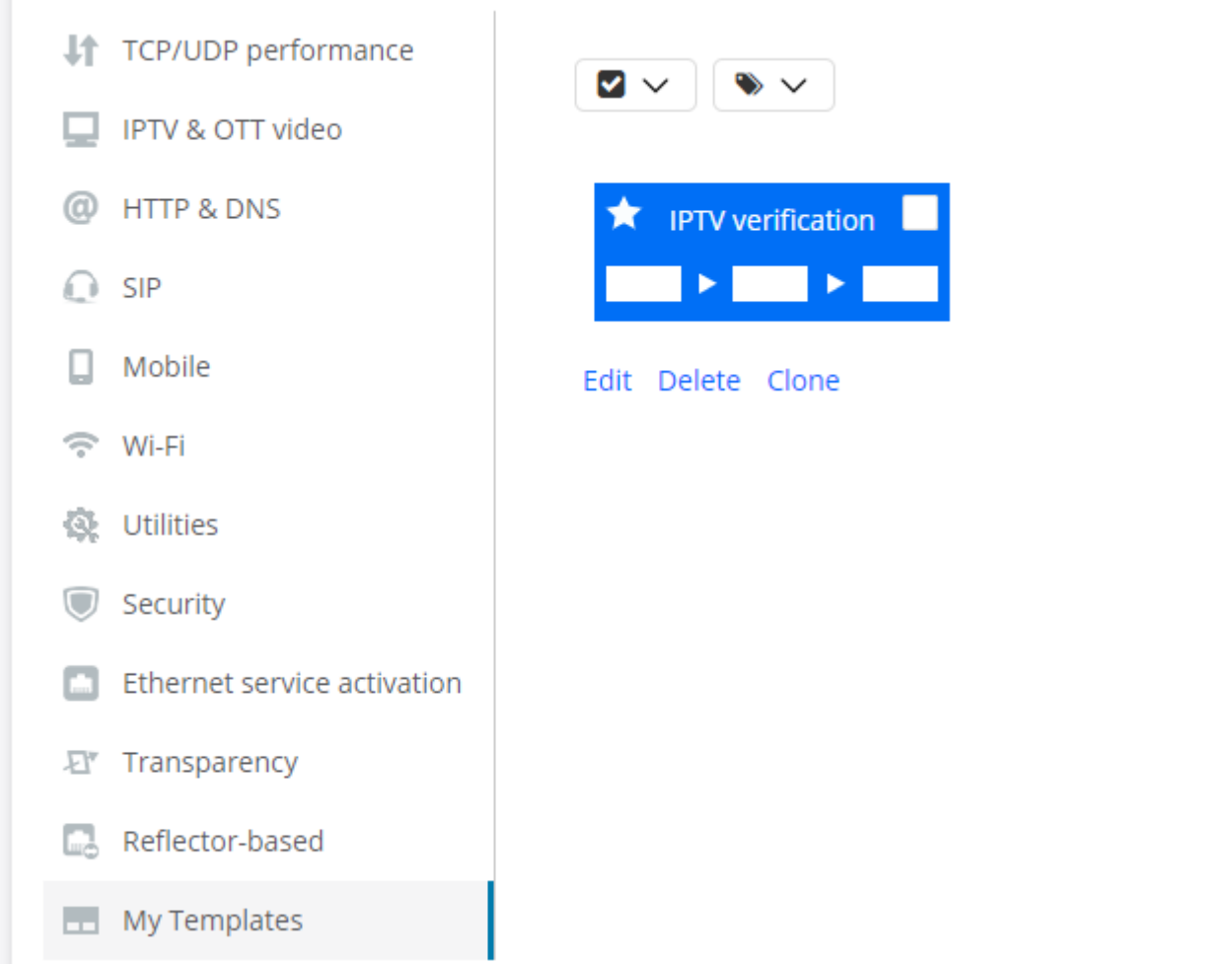
- Fill in the parameters for the test, as exemplified below. Hover over the “i” symbol for a parameter to view a tooltip explaining it. You can also click the “i” symbol to view a more detailed parameter description in this support documentation (a new copy of it will open on a new tab in your web browser).



You can apply various filters to the Test Agent interfaces appearing above (in the Server and Clients boxes). Hiding some of these may be useful if you have a large number of Test Agents:

- IPv4: Show IPv4 interfaces.
- IPv6: Show IPv6 interfaces.
- Show offline: Show interfaces of Test Agents that are offline.
- Show management: Show management interfaces.
- Show external IP: Show external IP addresses of interfaces (configured as described *here* (page 177)). They will appear as separate items.

If test *templates* (page 273) have been defined, you can build each test step from a template instead. To this end, click the My Templates category on the left, then select the desired template.



7.2.1 Putting test tasks in parallel

A step can contain multiple *tasks* that are executed in parallel. Note, however, that only a subset of task types can run concurrently with another. Those that can are represented by puzzle pieces on the New test sequence screen; those that cannot are represented by boxes. (There are also certain task types which require exclusive access to the Test Agent, so that no other tests can be assigned to the Test Agent. This is noted for each task type to which the limitation applies.)

To add another task to the step defined:

- Click the empty puzzle piece Add Parallel in the currently selected step.
- Pick a task in the same way as above. Selectable task types are limited to those allowed in the parallel construct.

The recommended maximum number of parallel tasks in a test step is three.

7.2.2 Putting test steps in sequence

You can also extend the test by adding more steps. Each step will then run to completion before the next step is begun.

- To add a step, click the empty box Add Step on the right. Proceed to select a test task as described above. Again, if you like, you can specify multiple tasks to be run in parallel in this step.

In this way you can continue to build your test sequence.

The recommended maximum number of steps in a test is 30.

The maximum number of steps allowed is strongly dependent on the number of parallel tasks in each step and the resource requirements of the task types involved.

7.2.3 Cloning an element in a test sequence

 You can duplicate a task or an entire step in the test sequence by clicking its Clone button.

- Cloning a *task* adds a duplicate of the task (with the same parameter settings) in parallel in the same step.
- Cloning a *step* adds a duplicate of the entire step to the right of the original, again with the same parameter settings for the step and for all tasks contained in it.

7.2.4 Moving a step in a test sequence

 You can move a step one position to the *left* in the test sequence by clicking its left-arrow button.

 You can move a step one position to the *right* in the test sequence by clicking its right-arrow button.

7.2.5 Removing an element from a test sequence

 You can remove a task or a step from a test sequence by clicking its Remove button.

7.2.6 Starting a test

When you are done setting up the test, it is ready to be started.

- Click the Start button (top right) to start the test.

There are some further options here, accessed by clicking the arrow in the right-hand part of the Start button. Regarding these options, see *Options for running tests* (page 258).

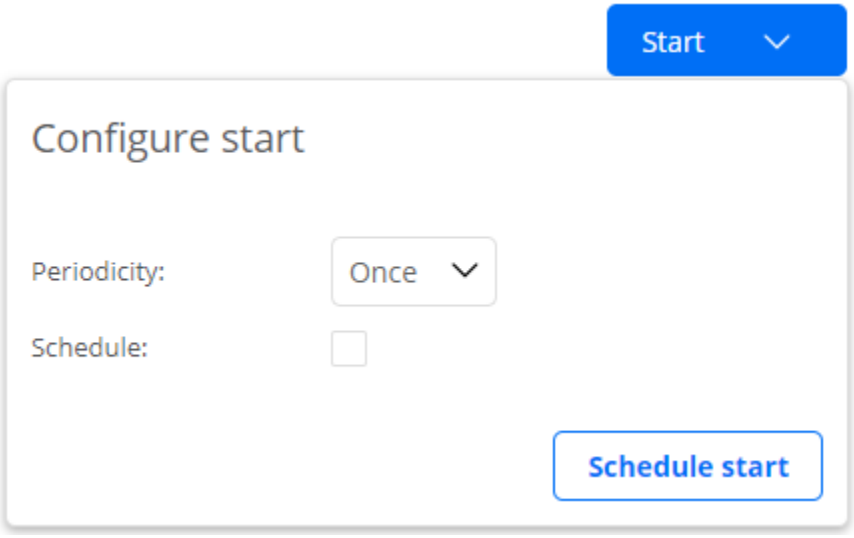
Once the test has been started, it will appear in the Tests view as well as in the overview on the Dashboard.

After a test has been started, it cannot be edited further. To create editable and reusable test templates, use the Create template function. See *this page* (page 273).

7.3 Options for running tests

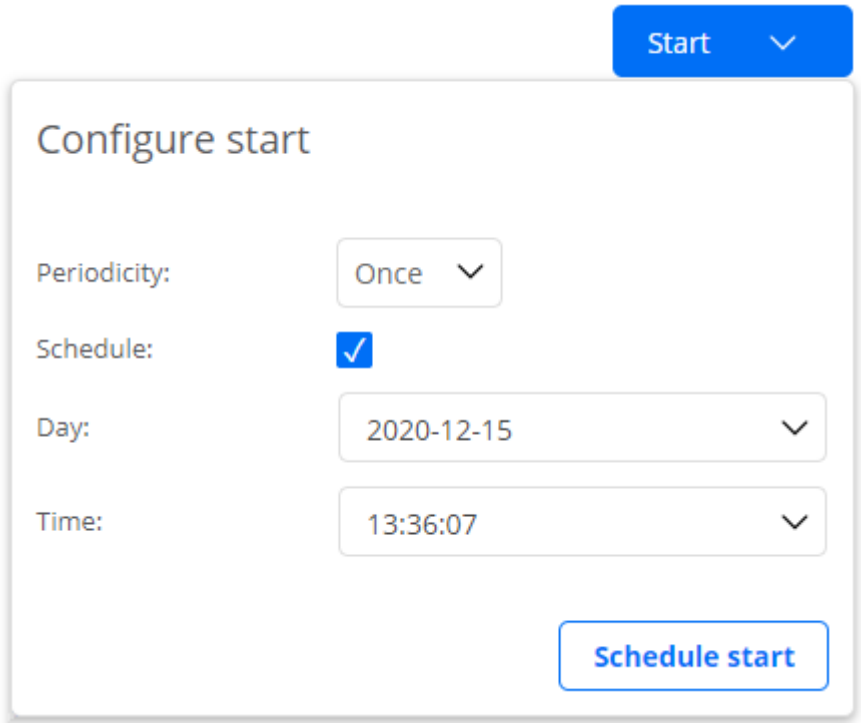
- To simply start a test right away and run it once, click the Start button in the top right corner of the New test sequence screen.

- If you want to run a test periodically, click the arrow in the right-hand part of the Start button, and make your choice under Periodicity (“Hourly”, “Weekly”, etc.). Then click the Start periodic test button.



Note: If you make a test periodic, it will be *converted into a monitor* and will appear in the Monitoring view instead of the Tests view.

- You can also schedule a test to start at a specified future time. To this end, click the arrow on the Start button, then select the Schedule checkbox and enter a date and time. If you want to run the test only once, select “Run once” under Periodicity, then click the Schedule start button.



- The periodicity and scheduling options can be combined. For example, you can schedule a test to be run once every hour, starting tomorrow at 2 p.m. Again, such a test will be converted into a monitor.
- If the test contains steps that you're not interested in at this time, you can skip them. To skip a step, click its Skip Step button. This button and the step label will turn orange and the box enclosing the step will be colored white, signifying that the step will be skipped over in the execution.



- There is also the Rerun option in the *view showing an individual test* (page 260).

7.4 The Tests view

The Tests view lists all tests currently defined in the Routing Active Testing system. Up to 20 tests are listed on one page; if more tests are defined, the view will be split into multiple pages, which you browse by means of page links at the bottom of the view.

For each test the following is indicated:

- Session: Name of the test. Click the name to view the test setup and a summary of its execution. To the left of the name is an icon indicating the current status of the test; for explanations, see *Icons used for tests and monitors* (page 286).
- Creator: The name of the person who created the test.

- Started: Date and time when the test was started.
- Completed: Date and time when the test was completed.
- Shared: Icon indicating the share status for the test. How to share tests with others is explained on the page *Sharing test and monitoring results* (page 500).

7.4.1 Inspecting an individual test

The name of each test in the Tests view is a clickable link. Clicking a test takes you to a new view which details the execution of that test, while also allowing you to perform some further user actions. See the page *View showing an individual test* (page 260).

7.4.2 Multi-select function

You can select a test by selecting its checkbox on the far left. You can select all tests by clicking the checkmark box at the top of the list and selecting “All”. Select “None” in the same box to clear all selections.

When at least one test is selected, a button labeled with a trash can appears. Click this button to delete all selected tests.



(This is a shortcut for clicking each test individually, then clicking the Delete button in the *test-specific view* (page 260).)

7.4.3 Filtering the Tests view

At the top of the Tests view are some filtering controls. You can filter tests with respect to their name, their current status, and their creator. This is handy especially if a large number of tests have been defined in the system.



- Specify filtering criteria as desired, then click the Filter button to apply them. The view will now show only tests matching the criteria.
- To remove the filtering, click the Clear button.

7.4.4 Creating a new test

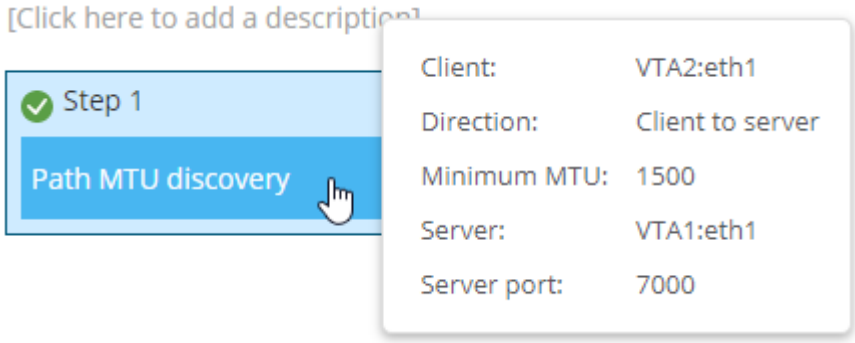
- To set up a new test, click Tests on the left-side bar and select New Test Sequence, or click the Create new button in the Tests view. All about this is covered on the page *Building tests* (page 254).

7.5 View showing an individual test

Clicking a test in the Tests view takes you to a new view dedicated to that test.

At the top, immediately below the heading with the test name, is a diagram showing the steps of the test. It has the same look as in the test builder.

- Hover the mouse pointer over a test step to view parameter settings for that step in a tooltip.



- Click a step to view details on the execution and outcome of that step further down the page.

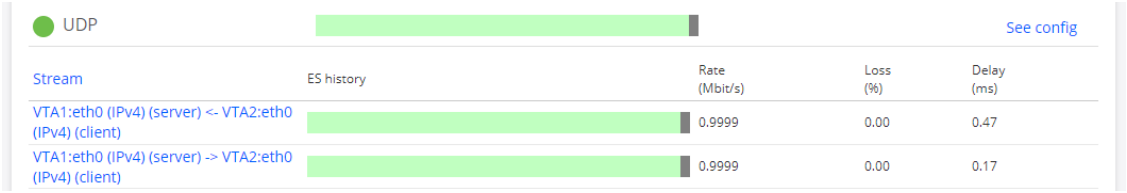
  What exactly is shown in these details depends on the nature of the test. Here are some examples:

  – For some tests, selected parameter settings are shown in a table.

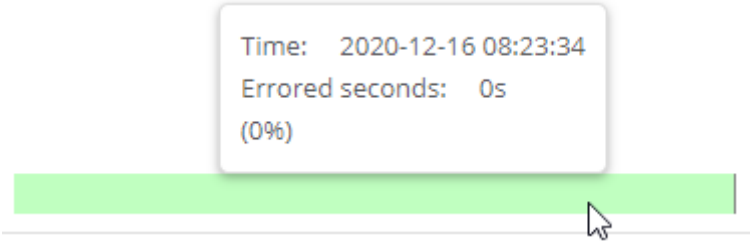
Configuration

Name	Value
Client->Server MTU	1500
TCP implementation	Linux TCP stack as in RFC 793, 1122, 2001 with NewReno and SACK extensions
Number of TCP streams	2
Socket buffer size	4.0 KB

  – For a service performance test such as TCP or UDP, a table like the one below is displayed.



  – The ES history bar shows the history of *errored seconds* (page 475) over the duration of the test. Hover the mouse pointer over the bar to identify time instants and view overall errored second statistics.



  – The table will also in many cases hold various result metrics. Several tables may appear, each dealing with a different aspect of the test.



  - You can click the item on the left in a table (for example, a client or stream) to call up a new window with more detailed test results. These results are formatted the same way as in a test report (see the page *Reports on tests and monitors* (page 269)) and are simply a subset of such a report.

- Where relevant, a timestamped event log appears (for the test in its entirety).

```
2015-09-10 14:02:56: Per socket buffer limit on probes: 160.0 KB
2015-09-10 14:02:56: Total socket buffer limit on probes: 33146.4 KB
2015-09-10 14:02:56: Getting path MTU for direction customer2:eth1.10 (IPv4) -> corenew:eth1.10 (IPv4).
2015-09-10 14:02:57: Got 1500 bytes MTU.
2015-09-10 14:02:57: Measuring base RTT.
2015-09-10 14:03:10: Got 0.28 ms RTT.
2015-09-10 14:03:10: Calculated BDP: 0.02 KB
2015-09-10 14:03:10: Using 0.32 KB BDP
2015-09-10 14:03:10: Needed memory on probes: 0.02 MB
2015-09-10 14:03:10: Free memory on corenew:eth1.10 (IPv4): 1927.94 MB
2015-09-10 14:03:10: Free memory on customer2:eth1.10 (IPv4): 199.69 MB
2015-09-10 14:03:10: Starting streams.
2015-09-10 14:12:36: Collecting statistics.
2015-09-10 14:12:45: Passed:
```

- For security tests, a listing of operations is provided, with a pass/fail outcome indicated for each operation.

```
2020-12-16 08:41:52: Testing connectivity from VTA1:eth1 (IPv4) to VTA2:eth1 (IPv4).
2020-12-16 08:41:53: Connectivity is working.
2020-12-16 08:41:53: Starting listeners. The test will take about 60 seconds.
2020-12-16 08:42:53: Passed: Test passed, no router redundancy protocols received at customer port.
```

7.5.1 Additional functionality in this view

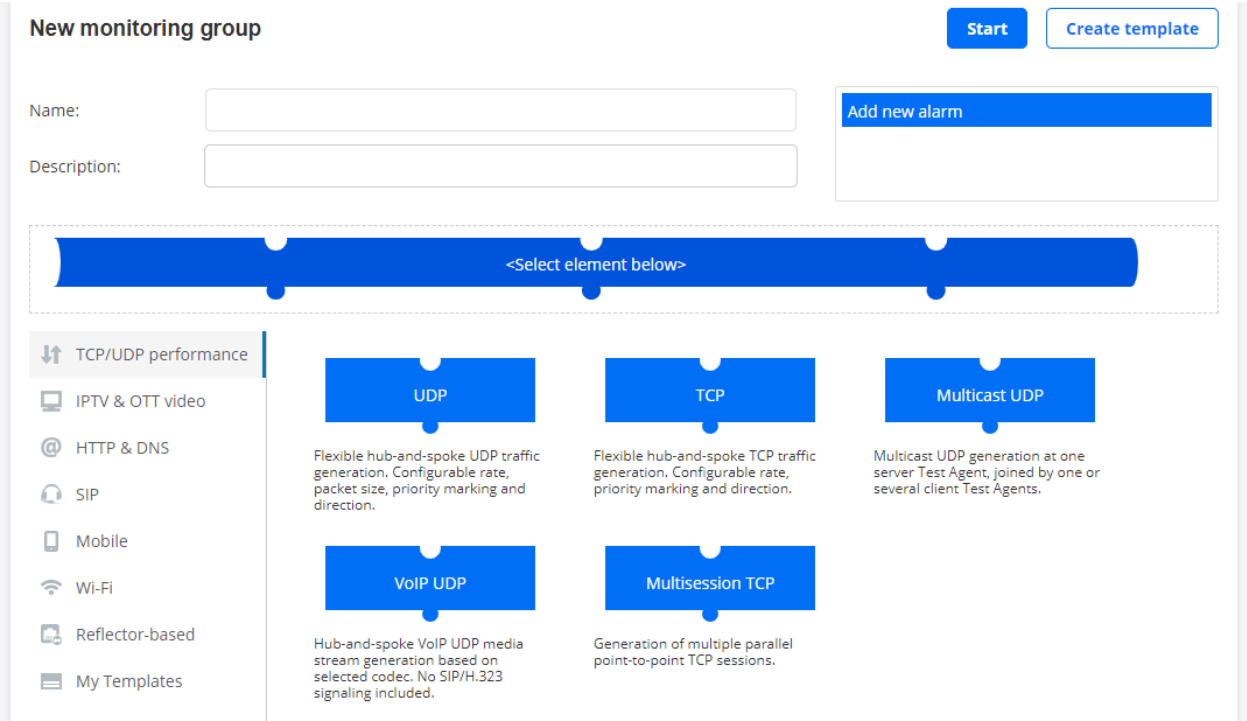
- Rerun button: Click to run this test once more. If you just click Rerun, the data from the previous execution is not affected. However, if you click the down-arrow and select Rerun & overwrite, the data from the previous execution is deleted and replaced by the data from the new one.
- Delete button: Clicking this button will delete the test.

Warning: This action cannot be undone, and you are prompted to confirm it.

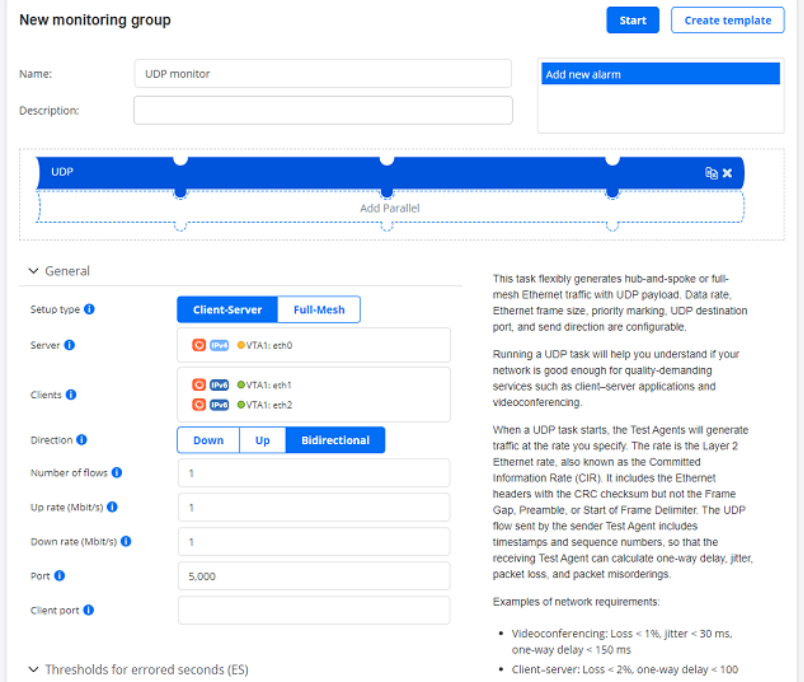
- Report button: Generate a report on this test. See the page *Reports on tests and monitors* (page 269).
- Export button: Export test results in comma-separated or plain-text format (the choice of format depends on the task type). Again, see the page *Reports on tests and monitors* (page 269).

7.6 Building monitors

- To create a new monitor, click the puzzle piece that represents the desired function.



- Proceed to fill in the desired parameters, as exemplified for UDP below. Hover over the “i” symbol for a parameter to view a tooltip explaining it. You can also click the “i” symbol to view a more detailed parameter description in this support documentation (a new copy of it will open on a new tab in your web browser).



Name the monitoring. Optionally, you can also add a description.

Select the Test Agents that should act as server and client(s).

Set UDP stream bit rates.

Optionally, set thresholds for errored seconds.

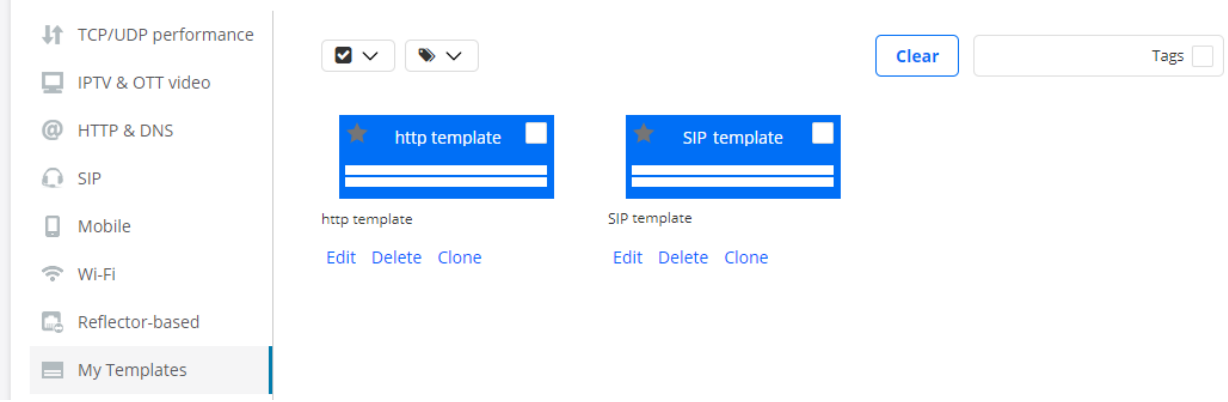
You can apply various filters to the Test Agent interfaces appearing above (in the Server and Clients boxes). Hiding some of these may be useful if you have a large number of Test Agents:

- IPv4: Show IPv4 interfaces.
- IPv6: Show IPv6 interfaces.
- Show offline: Show interfaces of Test Agents that are offline.
- Show management: Show management interfaces.
- Show external IP: Show external IP addresses of interfaces (configured as described *here* (page 177)). They will appear as separate items.

You can add further functions to monitor concurrently with the first one.

- To add one more function, click the empty puzzle piece labeled Add Parallel and repeat the above procedure.

If monitor *templates* (page 273) have been defined, you can build your monitor from a template instead. To this end, click the My Templates category on the left, then select the desired template.



Note that you can also configure alarm notifications via email or SNMP traps. This is done in the top section under Add new alarm. See the page *Activating alarms for a monitor* (page 488) for more information.

7.6.1 Starting a monitor

- Now click the Start button to start the monitor. Results will appear in the Monitoring view as well as in the Dashboard view.

7.7 The Monitoring view

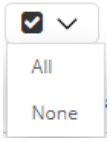
The Monitoring view lists all monitors currently defined in the Routing Active Testing system. Up to 25 monitors are listed on one page; if more monitors are defined, the view will be split into multiple pages, which you browse by means of page links at the bottom of the view.

- Name: Name of the monitor. Click the name to view a summary of its execution. To the left of the name is an icon indicating the current status of the monitoring session; for explanations, see the page *Icons used for tests and monitors* (page 286).
- Tags: Tags applied to the monitor. See the page *Applying tags to monitors and templates* (page 283).
- Created: Date and time when the monitor was created.
- Creator: The user who created the monitor.
- Share: Icon indicating the share status for the monitor. How to share monitors with others is explained on the page *Sharing test and monitoring results* (page 500).

The name of each monitor in the view is a clickable link. Clicking a monitor takes you to a new view which details the execution of that monitor, while also allowing you to perform some further user actions. See the page *View showing an individual monitor* (page 267).

7.7.1 Multi-select function

You can apply a number of functions to monitors that you have selected by selecting their checkboxes on the far left. You can select all monitors by clicking the checkmark button at the top and selecting “All”. Select “None” in the same box to clear all selections.



When at least one monitor is selected, a number of new buttons appear (and the Tag button, which is always visible, is enabled):

	Tag button: See the page *Applying tags to monitors and templates* (page 283).
	Play button: Click this button to start a monitor that is currently stopped.
	Stop button: Click this button to stop a monitor that is currently running.
	Trash can button: Click this button to delete all selected monitors. (This is a shortcut for clicking each monitor individually, then clicking the Delete button on the screen that follows.)

7.7.2 Searching (filtering) the Monitoring view

At the top of the Monitoring view are some controls for searching. As search criteria you can specify current status and creator, as well as enter a string to match in the monitor name. If you select the Tags checkbox, you can also specify tags that monitors should have applied to them. The search function is handy for picking out monitors with desired properties, especially if a large number of monitors have been defined in the system.



7.7.3 Applying tags to monitors

See the page *Applying tags to monitors and templates* (page 283).

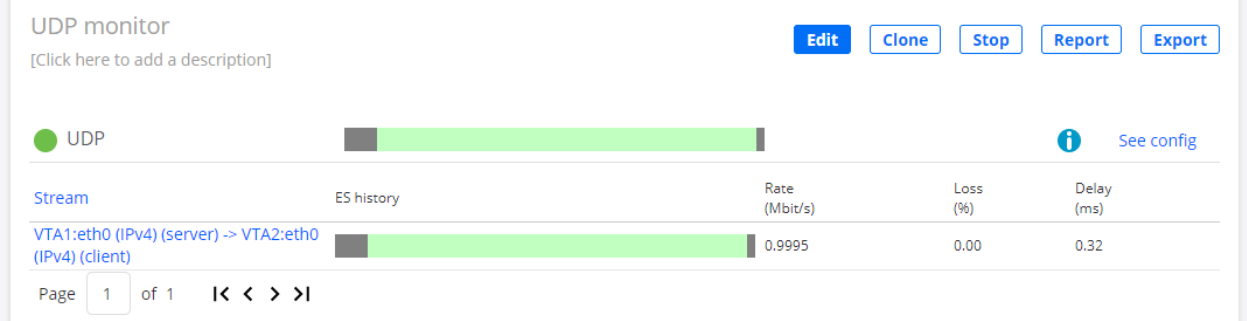
7.7.4 Creating a new monitor

- Either click Monitoring on the left-side bar and select New Monitor, or click the Create new button in the Monitoring view. See the page *Building monitors* (page 263).

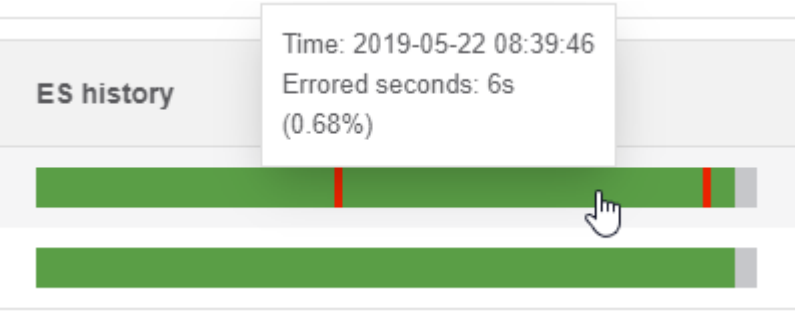
7.8 View showing an individual monitor

Clicking a monitor in the Monitoring view takes you to a new view dedicated to that monitor.

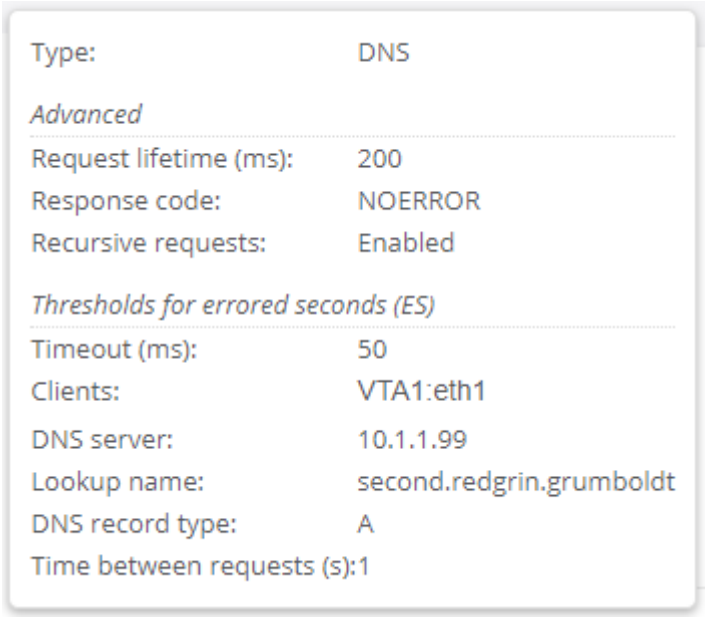
One or several tables like the one below are displayed:



- The ES history bar shows the history of *errored seconds* (page 475) over the interval specified under Time interval at the top of the page.
  - Hover the mouse pointer over the ES history bar to identify time instants and view overall errored second statistics.

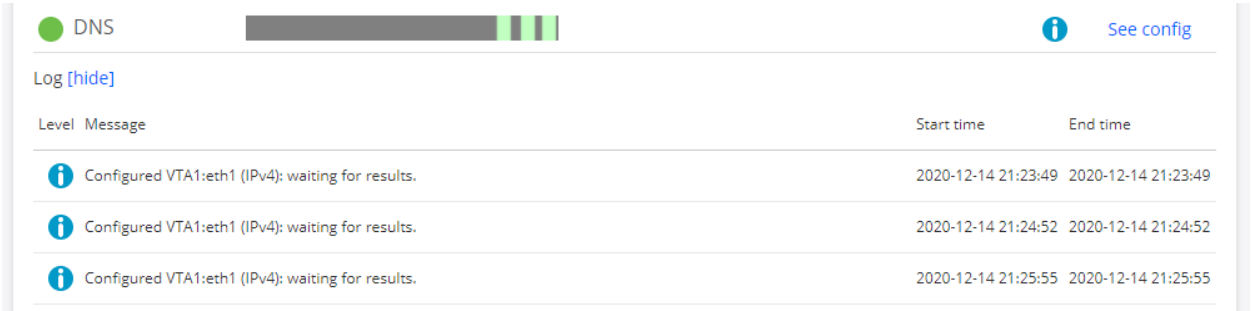


- The table will also in many cases hold various result metrics, as in the above example.
- You can click the item on the left in a table (for example, a client or stream) to call up a new window with more detailed monitoring results. These results are formatted the same way as in a test report (see the page *Reports on tests and monitors* (page 269)) and are simply a subset of such a report.
- Hover the mouse pointer over the See config link to view parameter settings for the monitor in a tooltip.



- The *colored dot* indicates the SLA fulfillment level. For the color coding, please refer to the page *SLA (Service Level Agreement)* (page 478).


Click this button to show event log messages for the monitoring session. These messages are by default hidden.



7.8.1 Additional functionality in this view

- Start/Stop button: Click to start the monitor if it is not currently running, or conversely to stop the monitor if it is being executed.
- Edit button: Edit the monitor in the monitor builder. A monitor can be edited even if it is running. After you save changes to the monitor, it will continue to execute but now using the new setup. Note, however, that data from any parts of the monitor that you abandon is lost and cannot be recovered later on. For example, if the monitor originally targeted three IPTV channels and you drop one of them, the data collected for that channel is lost. It is a good idea to export monitoring data collected so far before you edit a monitor; see Export button below.
- Clone button: Create a copy of this monitor. You are taken to the monitor builder with the settings of this monitor filled in.
- Report button: Generate a report on the execution of this monitor. See the page *Reports on tests and monitors* (page 269).
- Export button: Export monitoring results in comma-separated format. Again, see the page *Reports on tests and monitors* (page 269) for details.

- Delete button: Clicking this button will delete the monitor.

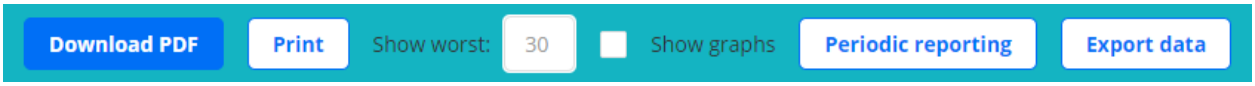
> **Warning:** This action cannot be undone, and you are prompted to confirm it.

7.9 Reports on tests and monitors

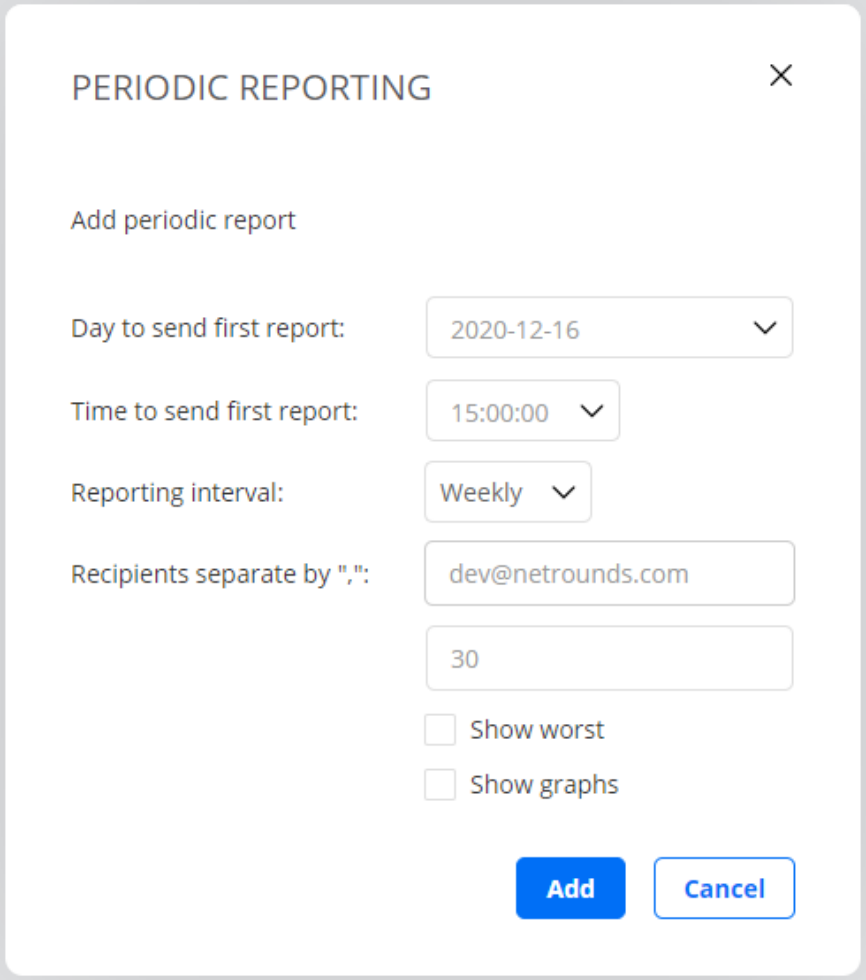
In Routing Active Testing you can easily create reports on tests and monitors, showing a result summary as well as details of the test or monitoring session. These reports can be automatically emailed to you daily, weekly or monthly in order to help you gain insight into the health of your network and to point out potential problem nodes. This page explains all result views and reporting options.

- Tests view: Simply click the test you want to create a report on.
- Monitoring view: Click the monitor you want to create a report on. Then make a selection under Time interval to specify the time interval to be covered by the report. You can set an arbitrary "from–to" interval by clicking the down-arrow button.
- Click the Report button (top right) to open the report in a new window.

The controls at the top of the report window are as follows:

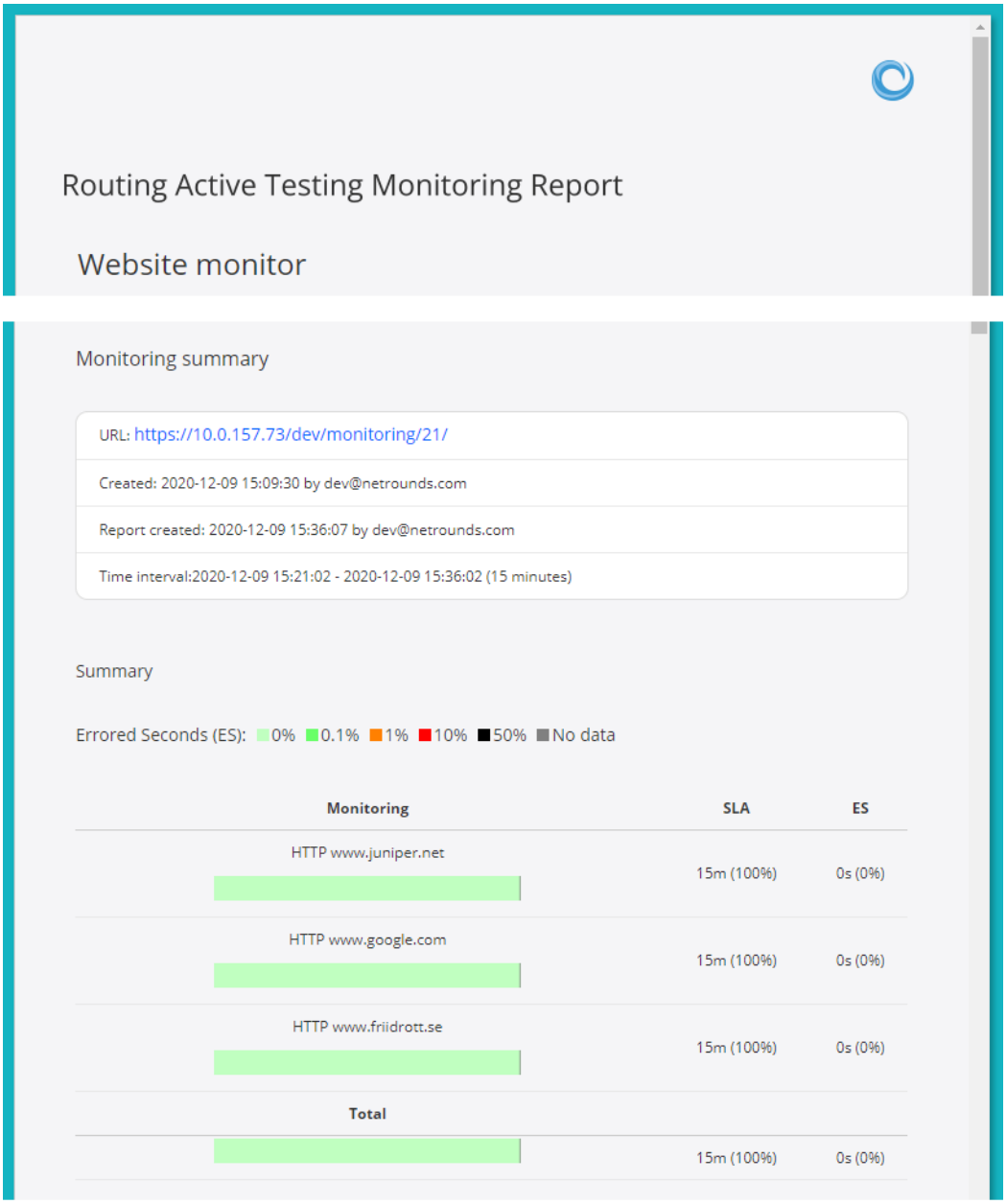


- Download PDF button: Download and create a PDF of the report.
- Print button: Send the report to a printer.
- Show worst: For each task in a test or monitor, you can specify how many measurement results to show, ranked by the number of errored seconds with the worst on top. The scope of a measurement result is task-dependent; to give one example, for HTTP it is the result obtained for one client. The default number is 5.
- Show graphs: Check this box to display graphs in the report.
- Periodic reporting button: For monitors, you can configure the Routing Active Testing server to send a report periodically at user-specified intervals.



- Export data button: Click this button to create a zip file with all test results.
  - Each result table for the test or monitor is exported to an individual comma-separated file with extension `.csv`.
  - For tests whose output consists of an event log, that log is reproduced in a plain-text file with extension `.txt`.

Below are some extracts from a report:



Routing Active Testing Monitoring Report

Website monitor

Monitoring summary

URL: https://10.0.157.73/dev/monitoring/21/

Created: 2020-12-09 15:09:30 by dev@netrounds.com

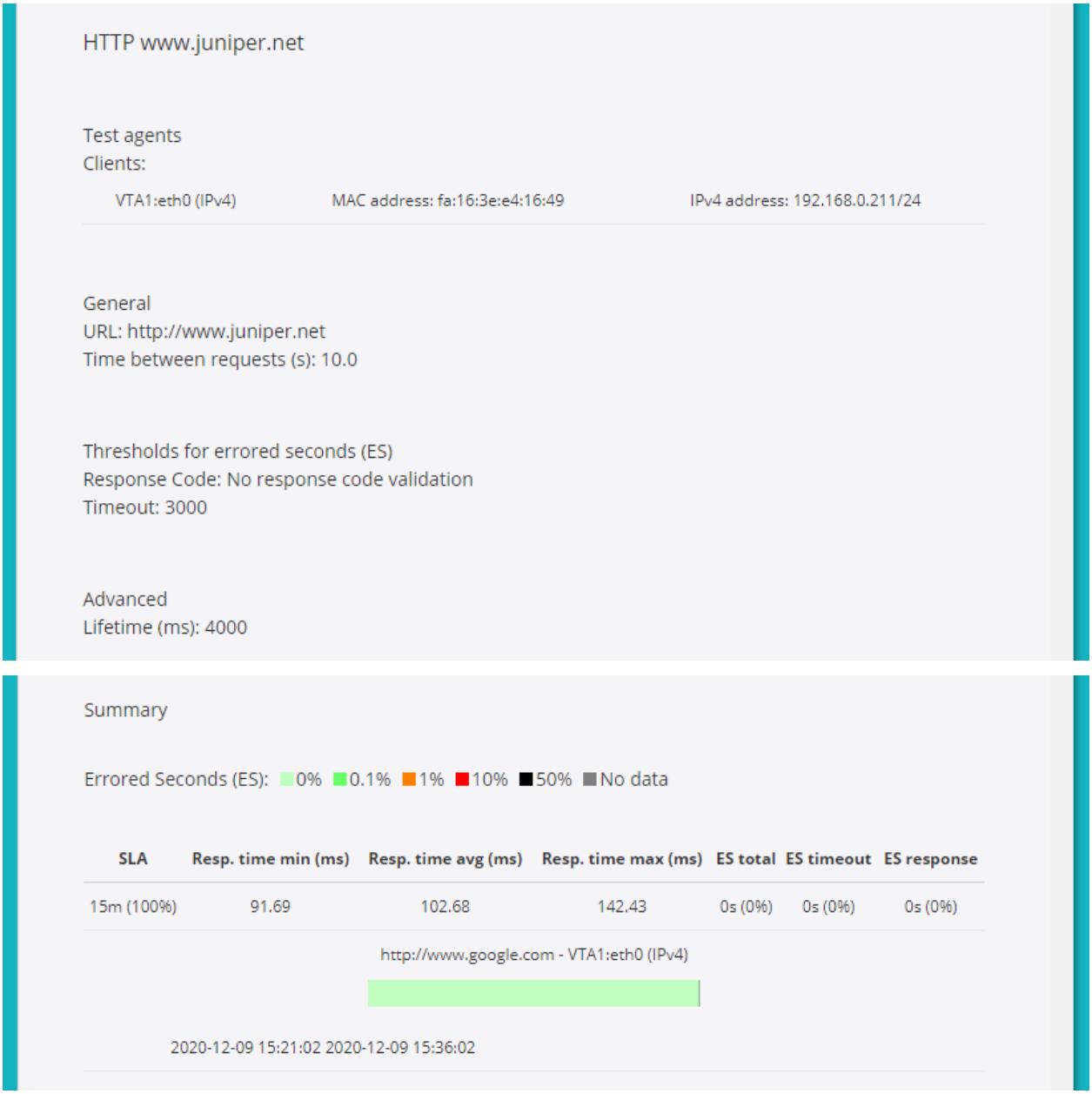
Report created: 2020-12-09 15:36:07 by dev@netrounds.com

Time interval:2020-12-09 15:21:02 - 2020-12-09 15:36:02 (15 minutes)

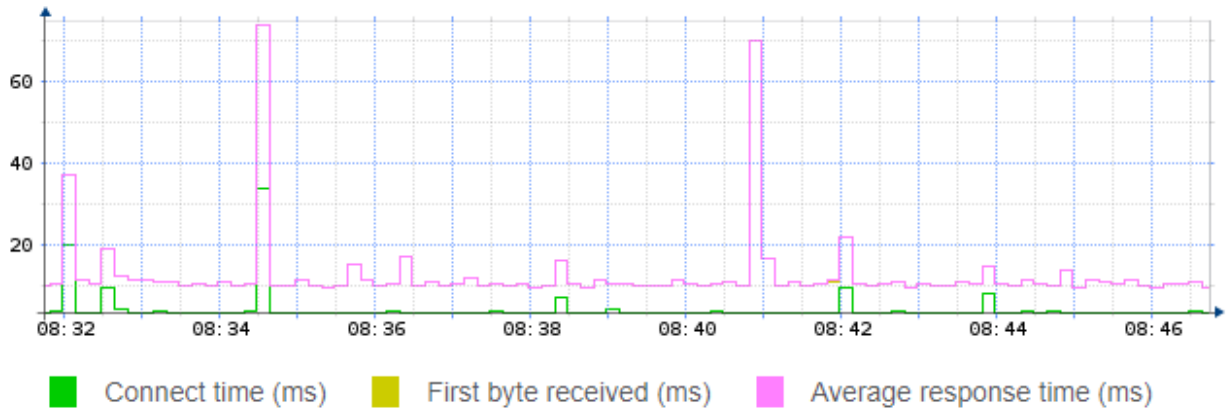
Summary

Errored Seconds (ES): 0% 0.1% 1% 10% 50% No data

Monitoring	SLA	ES
HTTP www.juniper.net	15m (100%)	0s (0%)
HTTP www.google.com	15m (100%)	0s (0%)
HTTP www.friidrott.se	15m (100%)	0s (0%)
Total	15m (100%)	0s (0%)



Here is a sample of what the graphs may look like:



7.10 Creating templates

The template feature in Routing Active Testing is a very powerful tool for creating complex entities that can be reused as building blocks in tests and monitors. Just like a test, a test template can comprise a multi-step sequence, where several tasks can run concurrently in each step. A monitor template consists of a single step, again optionally with several concurrent tasks.

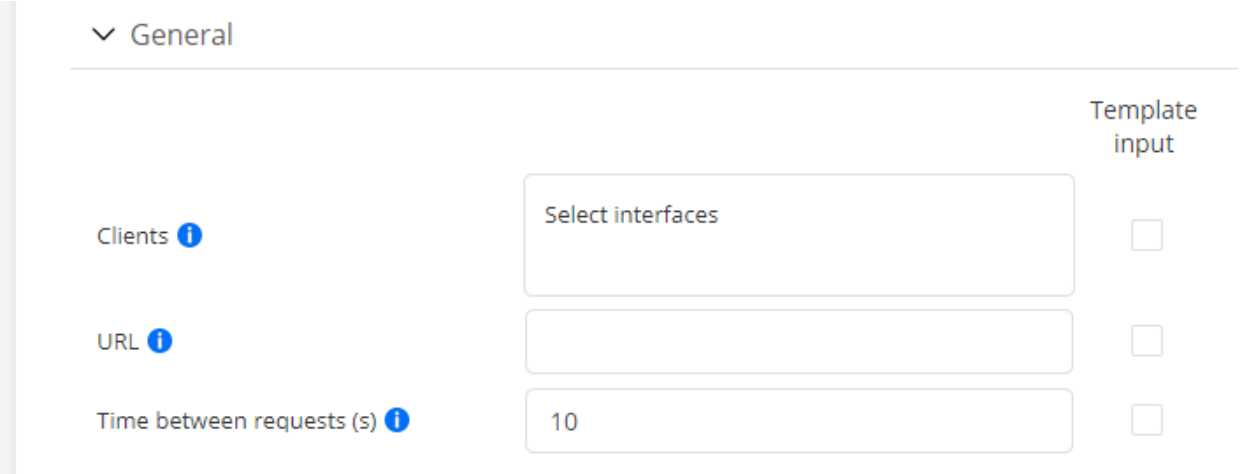
Moreover, rather than setting fixed values for parameters, you can leave parameters in a template to be defined at runtime. This further adds to the flexibility of templates in that you can reuse them on different occasions without having to edit them.

7.10.1 Creating templates for tests

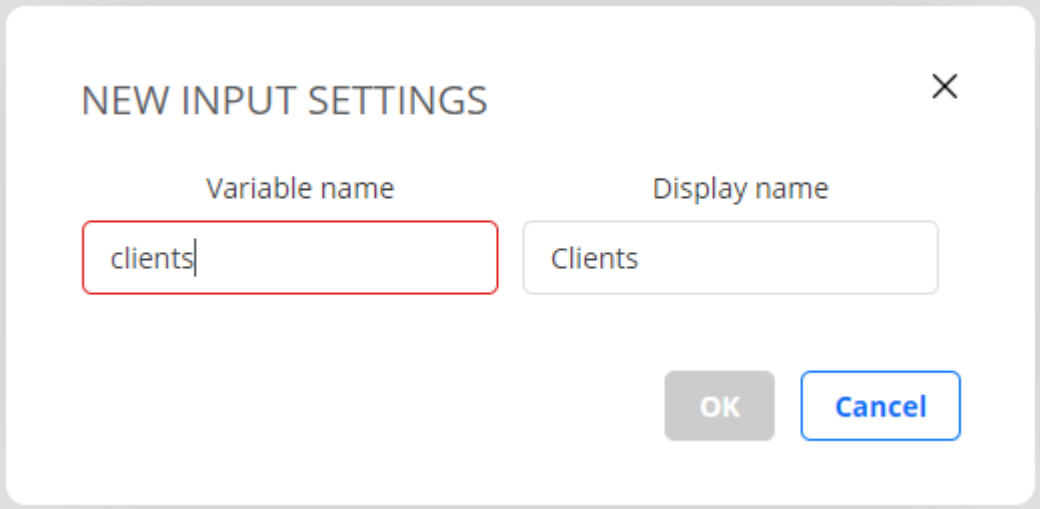
- Start by creating a new test, then click Create template.



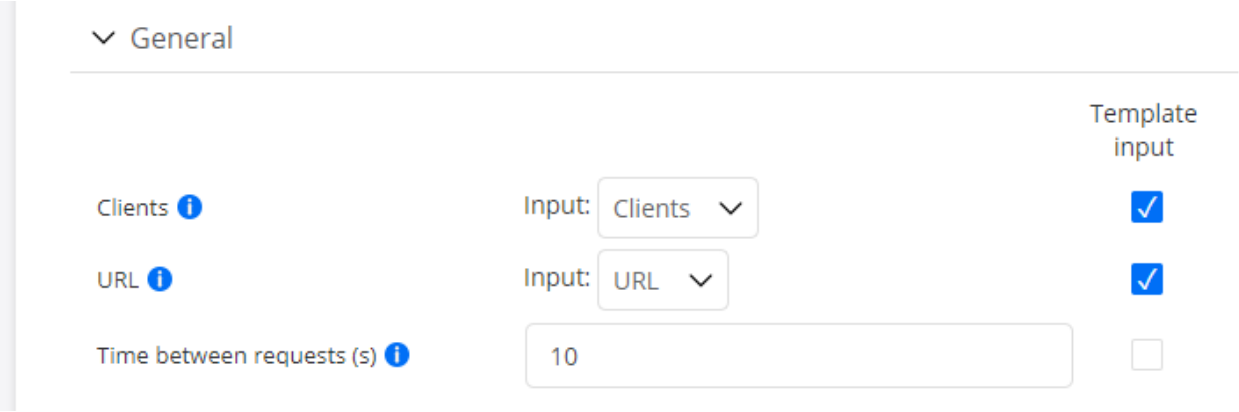
The procedure is very similar to building tests, but when setting up individual tasks, a new option Template input is provided for each parameter. This option determines whether the parameter is fixed or variable in the template. The parameters you specify as template inputs are left to be defined at the time of running the test.



- For all parameters that you want to leave to be defined at runtime, select the Template input checkbox.
- Then, for each of these parameters, select “Create new” in the Input box. You can accept the default Display name for the template input (identical with the parameter name), or you can if you wish name it differently. (The Variable name field is primarily of interest in the context of orchestration: it is the name used to refer to the parameter in orchestration APIs. See the Routing Active Testing orchestration guides for further details.)

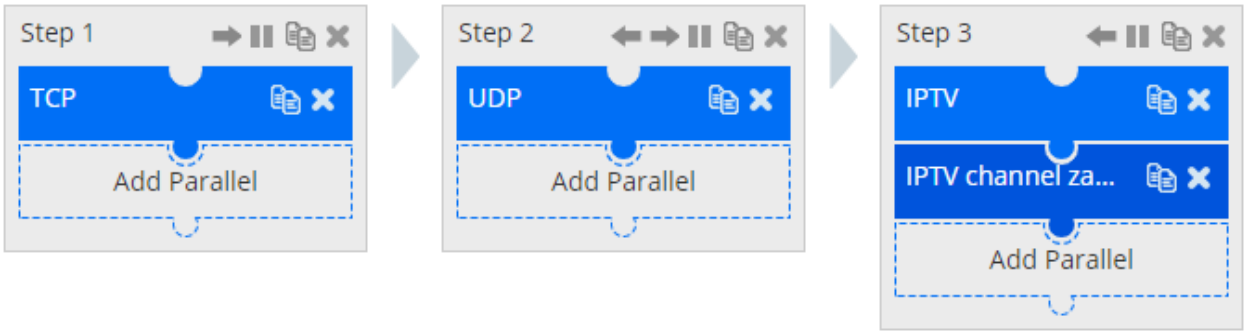


In the screenshot below, Clients and URL are treated as template inputs.



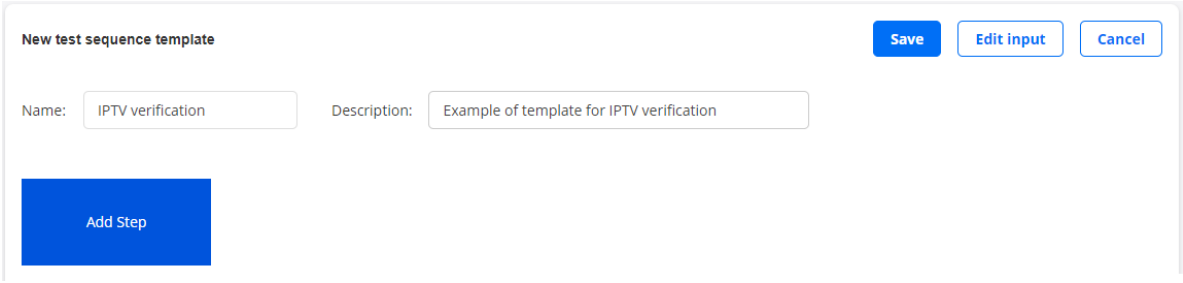
7.10.1.1 Procedure for building a template

Below we build a simple example of a template, designed to validate a broadband connection for IPTV. The template consists of three steps, as shown in the following screenshot:

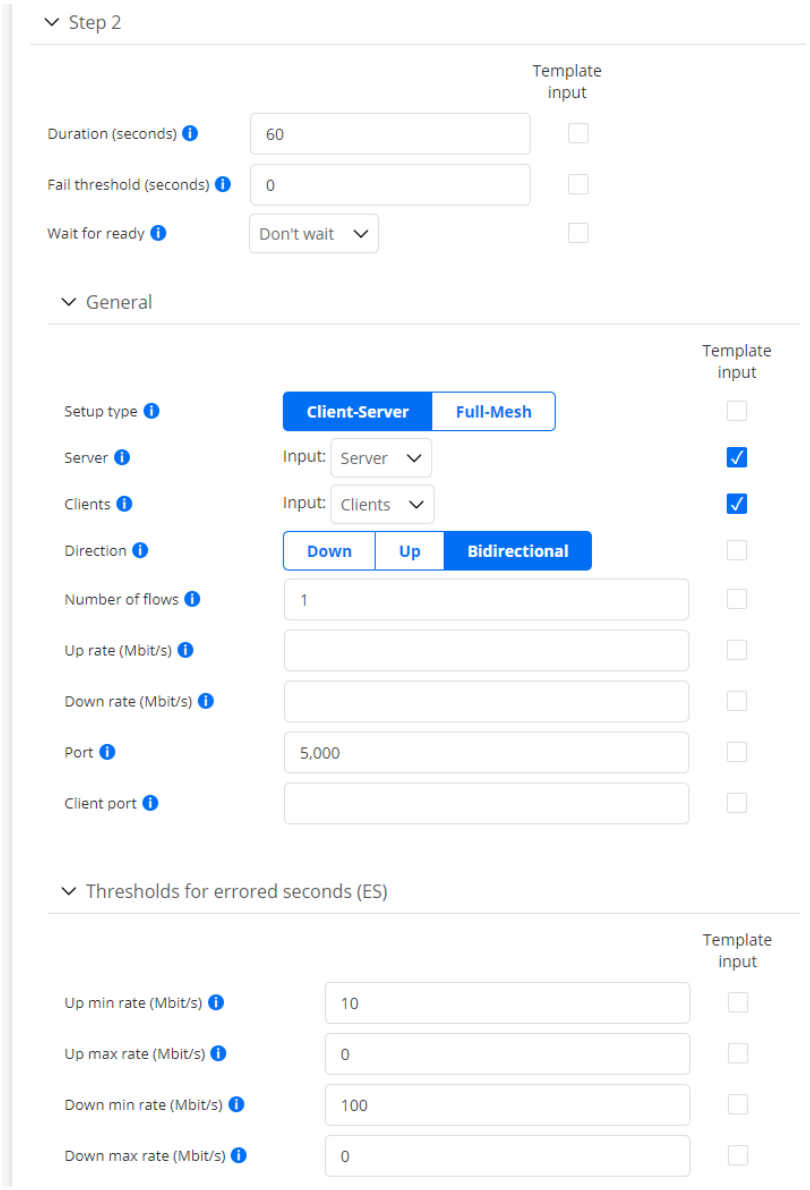


Proceed as follows:

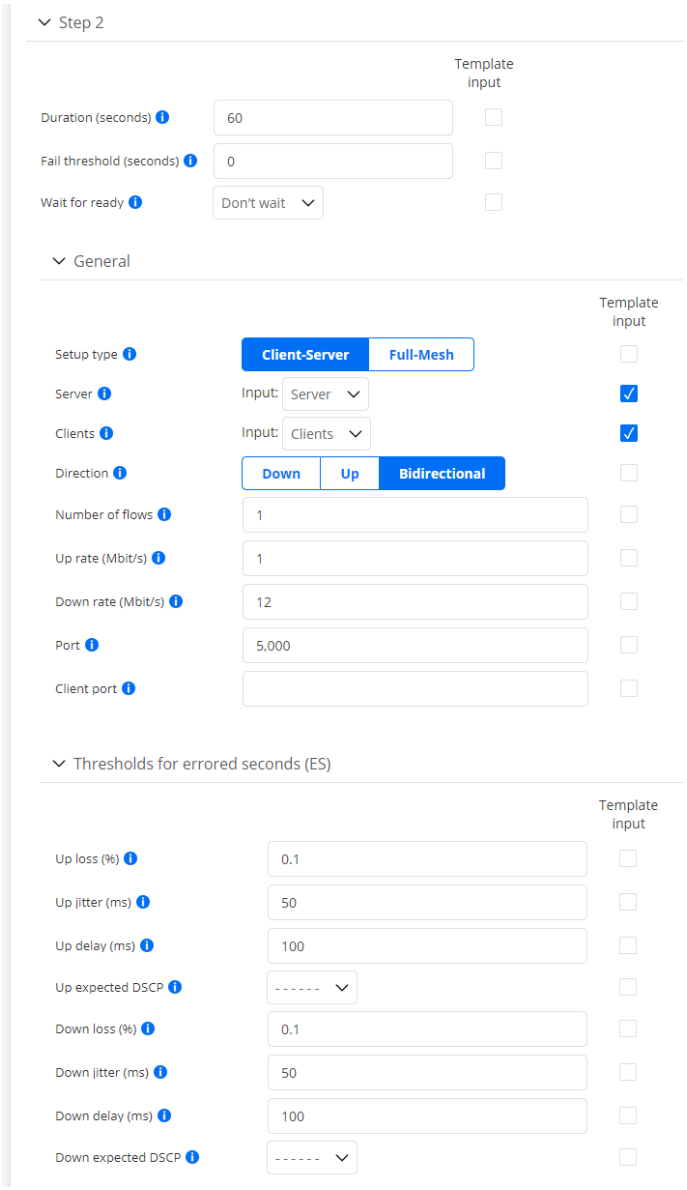
1. Name the template, and optionally enter a description of it:



2. Add a TCP throughput task for testing the throughput, and set the parameters and Template input options according to the screenshot below.

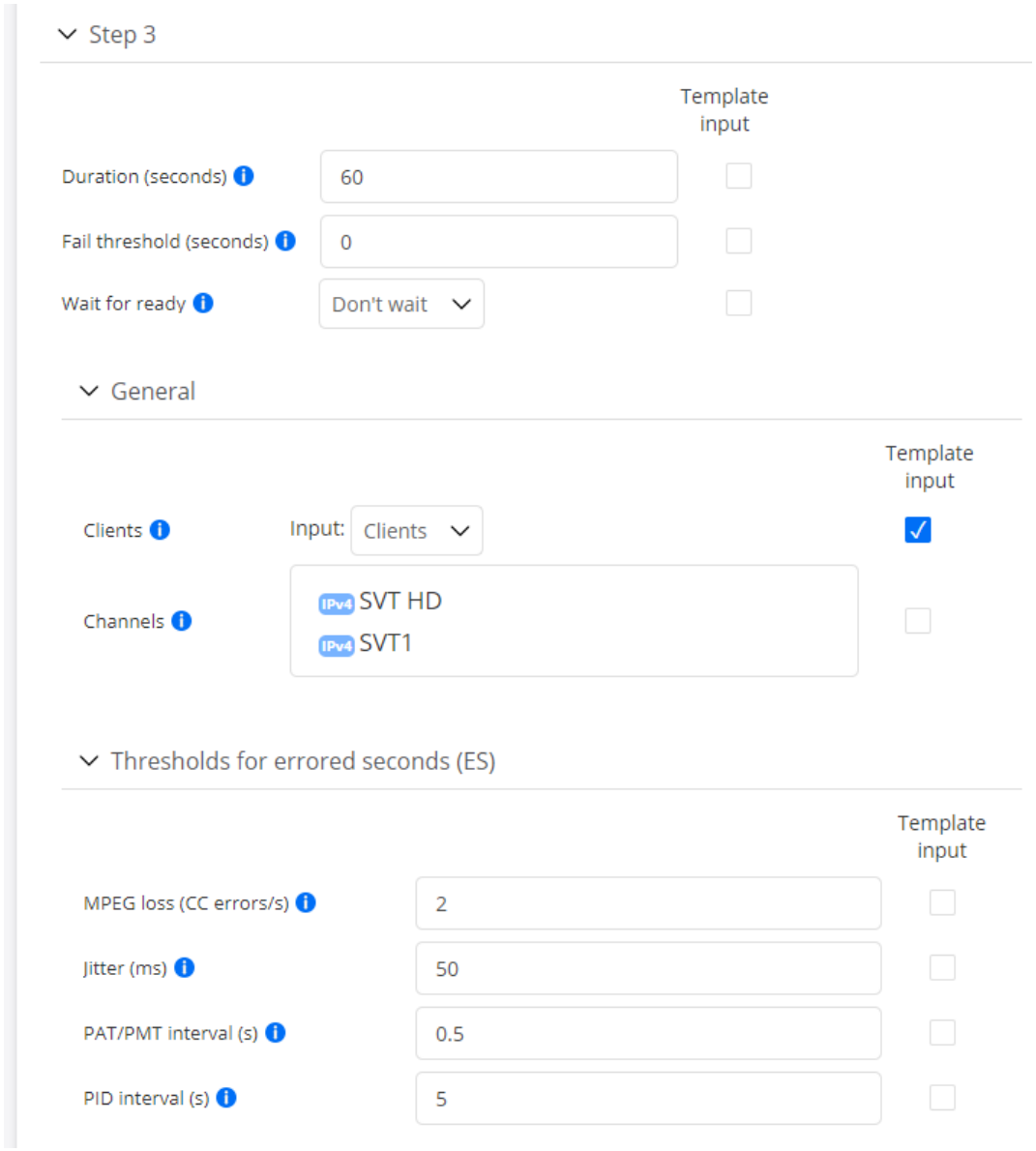


3. Add a UDP task as a second step in order to as ascertain the jitter and loss characteristics of the connection. Configure the task as shown below.

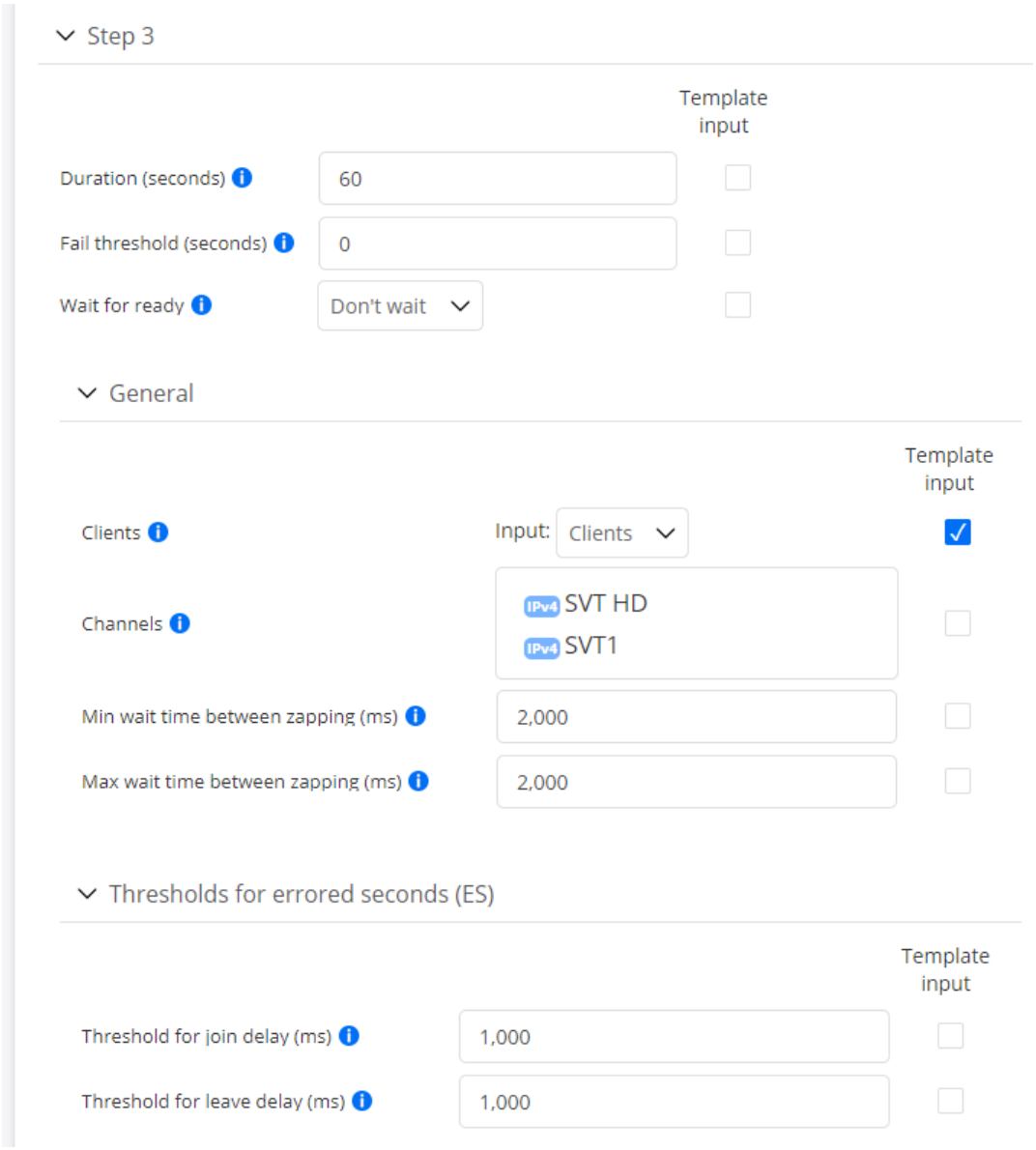


4. Add a third and final step with an IPTV task and an IPTV channel zapping task in parallel. This step checks the TV characteristics. Configure the step as shown below.

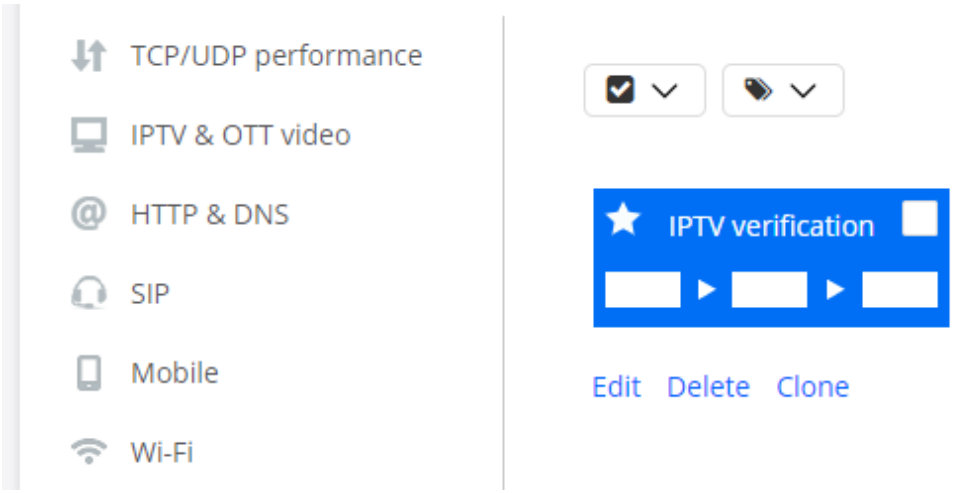
   IPTV:



IPTV channel zapping:



5. Click the Save button at the top. The new template now appears in the My Templates section when you start *defining a new test* (page 254):



7.10.2 Creating templates for monitors

Monitor templates are similar to one step of a test template.

Like test templates, monitor templates offer the Template input option for each parameter, allowing you to postpone the definition of parameters until runtime.

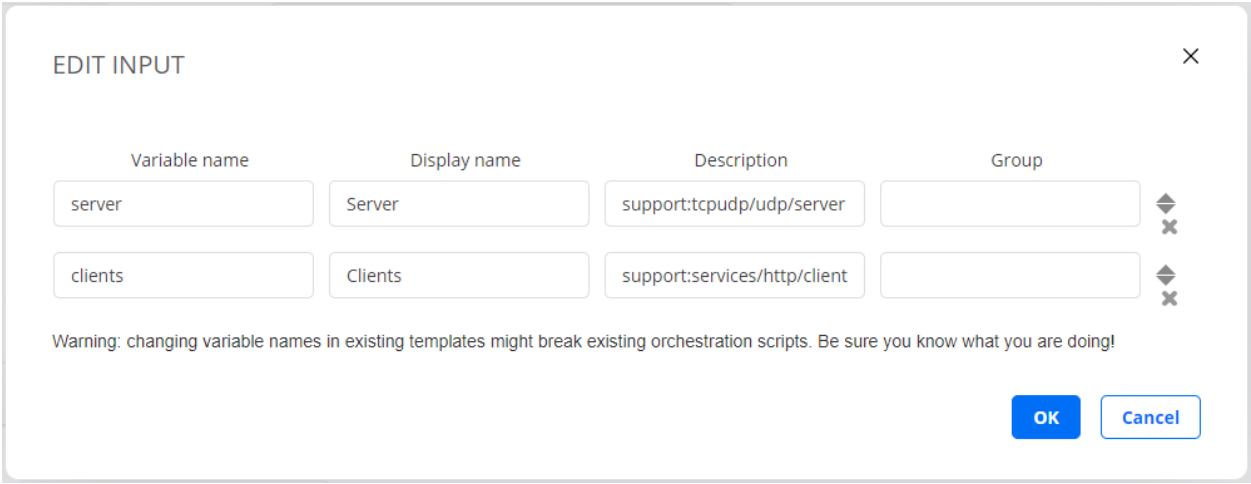
- Start by creating a new monitor, then click Create template.
- From here onward, follow the procedure for adding a step to a test template, optionally defining several tasks in parallel. See above (and in particular *Step 3* (page 276)).
- When you are done, click the Save button at the top. The new template now appears in the My Templates section when you start *defining a new monitor* (page 263).

7.10.3 Editing template input

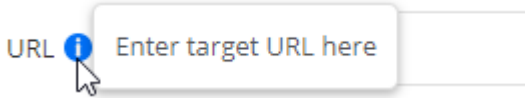
You can modify the properties and handling of template input parameters as follows:

- Click the Edit input button at the top of the screen.

A dialog appears listing all parameters marked as template inputs. An example is shown below.

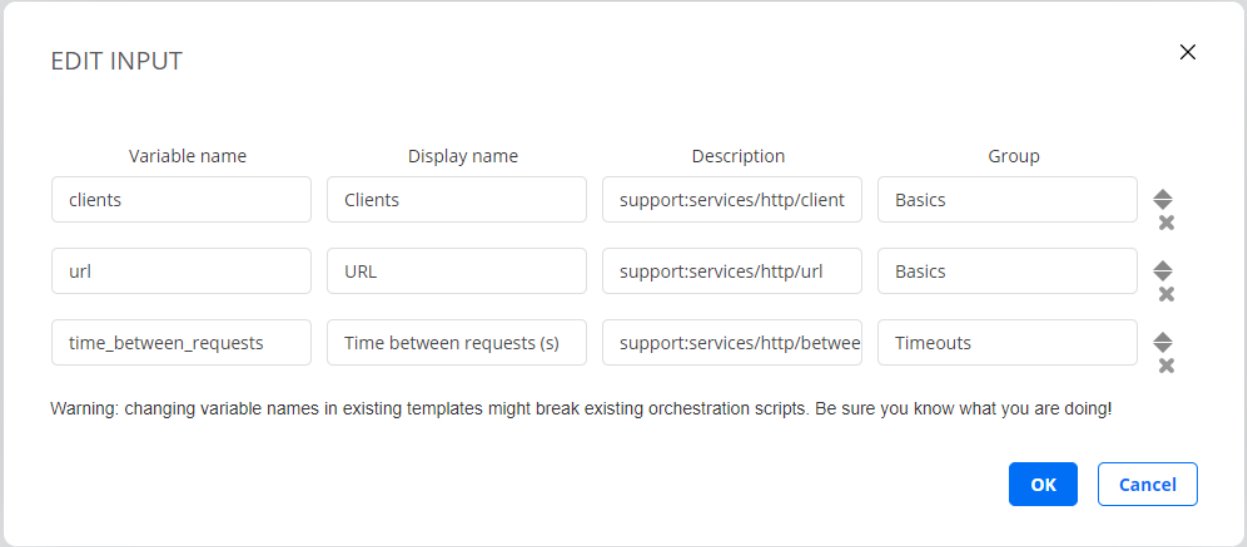


- Variable name: Again, this is the name of the parameter used in orchestration. Note: Be careful about changing this; any existing orchestration scripts which use the current name will stop working unless they too are updated.
- Display name: Here you can change the parameter name displayed in the Control Center web GUI.
- Description: This field by default contains a pointer to the description of the parameter in the present documentation. Clicking the "i" symbol next to the field takes you to that description, and hovering over the "i" shows a short version in a tooltip. If you wish you can replace this with a custom descriptive string. When you create a test based on this template, your custom description will then be shown as a tooltip when you hover over the "i". Below is an example.

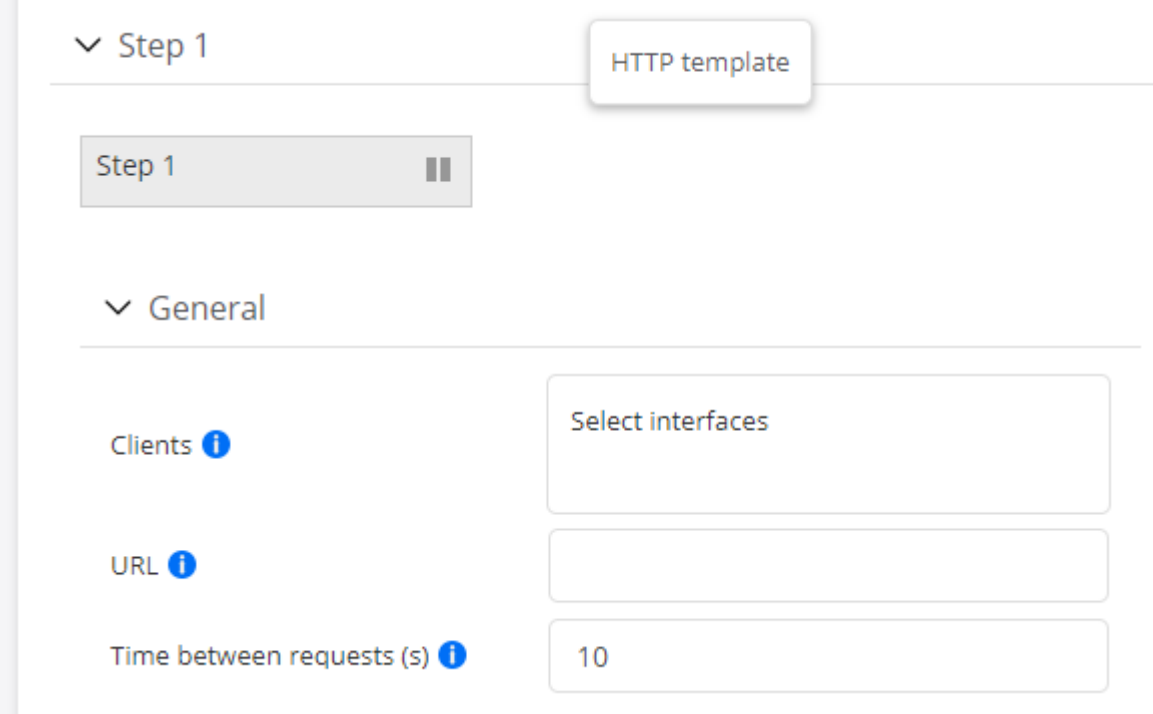


- Groups: You can organize all or some of your template inputs into *groups* of your own design in tests based on this template. These groups will then replace the default ones in the user interface. Enter the same group label for all inputs that you want to keep together in a group. Again, an example will serve to illustrate this.

This setup



will result in the following user interface for creating a test:



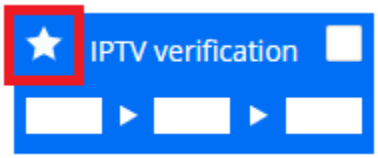
From the Edit input dialog you can also do the following, using the controls on the right:

- Click the cross for a template input to remove it from the list of inputs (that is, to hard-code this parameter into the template instead).
- Click the up or down arrow for a template input to move it up or down in the list of inputs shown in the user interface.

When you are done editing template inputs, finish by clicking the Save button at the top.

7.10.4 Marking templates as favorites

For both test and monitor templates, you can click a template's star icon to designate it as a favorite.



That template will then appear under the Favorites menu at the top of the left-side bar with the task categories.



The Favorites menu gives you quicker access to the templates that you use most frequently, especially if a large number of templates have been defined.

7.10.5 Propagating changes in templates to monitors

After you edit and save a monitor template, you need to do the following to apply the changes to any monitor (whether running or stopped) that is based on the template:

- Click the monitor of interest in the Monitoring view.
- Click the Edit button.
- Click the Save button.

This will update the monitor with the modified template.

Note: It is not sufficient to stop and restart the monitor.

(The above procedure is not applicable to tests since you cannot edit tests after creating them.)

7.11 Applying tags

Tags defined in Routing Active Testing can be applied to monitors, as well as to:

- templates
- Test Agents
- TWAMP reflectors
- network devices.

For example, you can tag a monitor with the same tag as a subset of Test Agents that are going to run the monitor. This feature is particularly helpful if you have a large number of monitors and templates defined.

A tag can consist either of a *key* alone (for example, “location”) or of a *key* with an accompanying *value* (for example, “location:sweden”). A colon is used as separator between key and value. Tags may consist of up to 50 characters, which may be lowercase letters or digits.

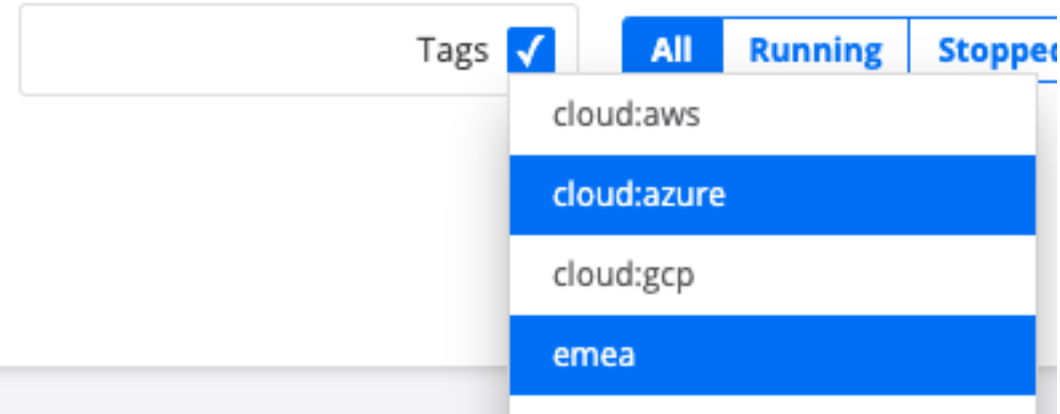
Tags are applied in the same way to all items listed above. The description that follows deals with applying tags to monitors in the Monitoring view, but it is equally applicable to the other item types (with obvious minor adjustments because of the differing screen layout).

In the Monitoring view it is possible to:

- *add* a tag to selected monitors;
- *remove* a tag from (“untag”) selected monitors.

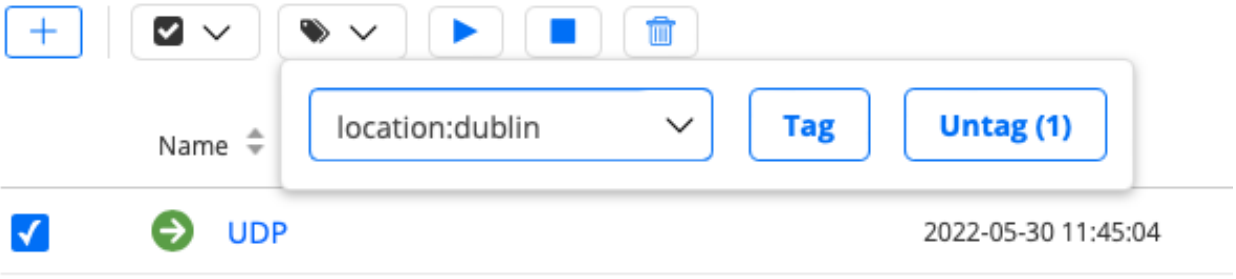
Note that the tags themselves, once created, cannot be deleted from the Control Center GUI.

In the search box, in addition to the text search function, you can filter the view on a selected subset of tags. The monitors having *all* the selected tags will then be displayed.

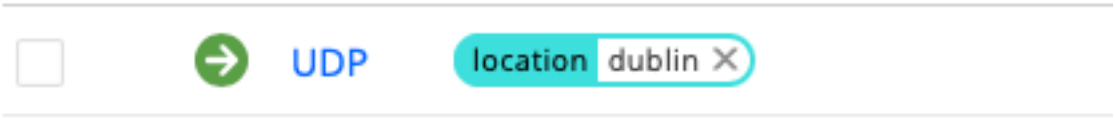


7.11.1 Adding tags to monitors

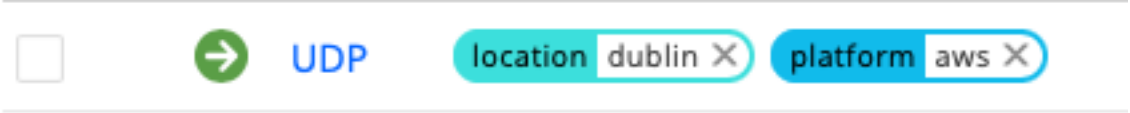
- First select the monitors that you want to tag.
-  Click the Tags button at the top of the page.
- You can either create a new tag or select an existing tag:
  - To create a new tag, type the desired string into the box, then click the Tag button.
  - To select an existing tag, click the down arrow and select the desired tag from the drop-down box (optionally, you can type the first few characters of the tag name to match the name). Then click the Tag button. See the screenshot below:



Each tag will show up in a box next to the monitor name.

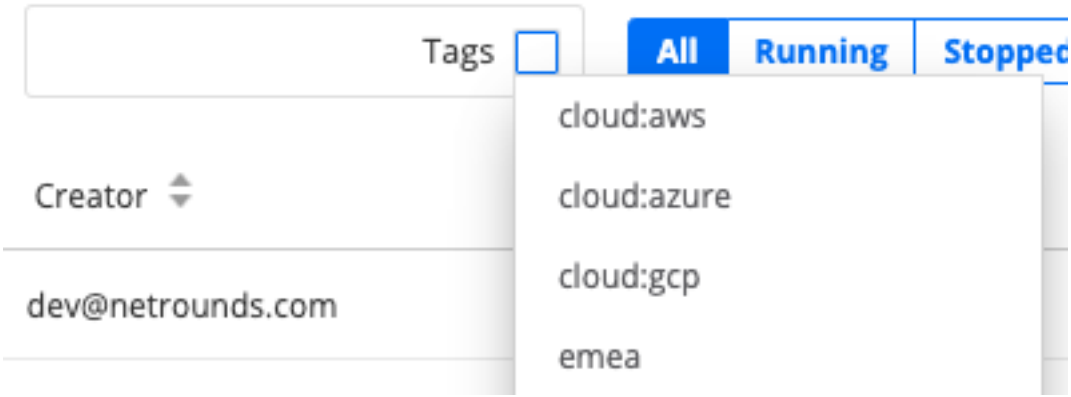


It is possible to add several tags to the same monitor, as shown below:



7.11.2 Using tags to filter monitors

- Select the Tags checkbox in the search field, and a drop-down list appears holding all defined tags.



- Select one or several tags to filter the Monitoring view on these tags.
- To display the full list again, just deselect the tag names in the search field.

7.11.3 Untagging monitors

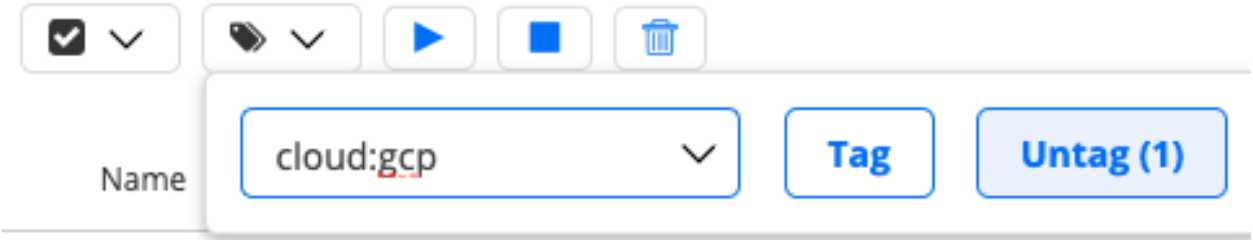
The simplest way to remove a tag from a monitor is to click the cross on the tag:



- You are prompted to confirm this action. Click OK.

Alternatively, you can do the following:

- First select the monitors that you want to untag.
-  Click the Tags button at the top of the page.
- Then select the name of the tag to remove (“cloud:gcp” in the example below), and click the Untag button.



The tag is now removed from the selected monitors.

7.12 Icons used for tests and monitors

7.12.1 Icons for tests

	The test is scheduled to be run at a future time.
	The test is pending or waiting to be run.
	The test is running.
	The test has completed and passed the test criteria.
	The test has completed but failed the test criteria.
	The test has stopped prematurely because of an error.
	The test has been canceled manually.
	The test has been skipped.

7.12.2 Icons for monitors

	A monitoring session is running.
	A monitoring session has been stopped manually.

8 Task types

8.1 Overview of measurement task types

The following table lists the measurement task categories found in the Control Center web GUI and indicates which tasks are available for use in tests and monitors respectively.

It should be noted that any test can be made *periodic* (page 258), in which case it will be run as part of a monitor. What we mean by a monitor here, however, is one created as such in the web GUI.

Task category	Test	Monitor
Network performance (page 290)	Yes	[1]
IPTV and OTT video (page 321)	Yes	Yes
HTTP and DNS (page 337)	Yes	Yes
SIP (page 347)	Yes	Yes
Mobile network (page 355)	Yes	[2]
Wi-Fi network (page 351)	Yes	[3]
Ethernet service activation (page 359)	Yes	No
Transparency (page 376)	Yes	No
Reflector-based (page 402)	Yes	[4]
Security (page 454)	Yes	No
Utilities (page 470)	Yes	No

8.2 Common test and monitor parameters

Here is a list of parameters that are set not on the *task* level, but rather on the *test/test step* or *monitor* level.

For parameters related to specific *tasks*, see the pages dealing with these tasks.

8.2.1 Test step parameters

A few parameters are set on the *test step* level rather than for the task. (A test step may contain multiple tasks executed in parallel.)

- Duration (seconds): Duration of this test step. Min: 30 s. Max: 604,800 s (= 1 week). Default: 60 s.
- Fail threshold (seconds): The maximum number of *errored seconds (ES)* (page 475) that may occur without triggering a fail for this test step. Default: 0.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: "Don't wait", i.e. zero wait time.

8.2.2 Advanced test parameters

- Delayed start (s): *(Tests only)* Time by which to delay the start of the task within a test step. Min: 10 s. Max: Equal to Duration minus 30 s. Default: 0, meaning that no delay is introduced.

[1] All tasks can be run in monitors except *TCP throughput test according to RFC 6349* (page 312) and *QoS policy profiling* (page 316).
[2] The *Mobile logger* (page 355) task can be used in monitors, while the *Mobile switcher* (page 357) task is available in tests only.
[3] The *Wi-Fi logger* (page 353) task can be used in monitors, while the *Wi-Fi switcher* (page 354) task is available in tests only.
[4] All tasks can be run in monitors except *BWPing* (page 433).

8.2.3 Monitor parameters

Since monitors are meant to be run for an extended period of time, it does not make much sense for them to be configurable with parameters like delayed start or a fixed duration.

- SLA Good: Threshold for good fulfillment of service level agreement. Default: 99.95%.
- SLA Acceptable: Threshold for acceptable fulfillment of service level agreement. Default: 99.5%.

8.3 Listing of task types supporting IPv6

The table below lists the task types for which IPv6 is supported. This information is also given on the pages dealing with each task, but a summary is provided here for convenience.

Task categories where IPv6 support is not relevant (such as Wi-Fi) are left out of the table.

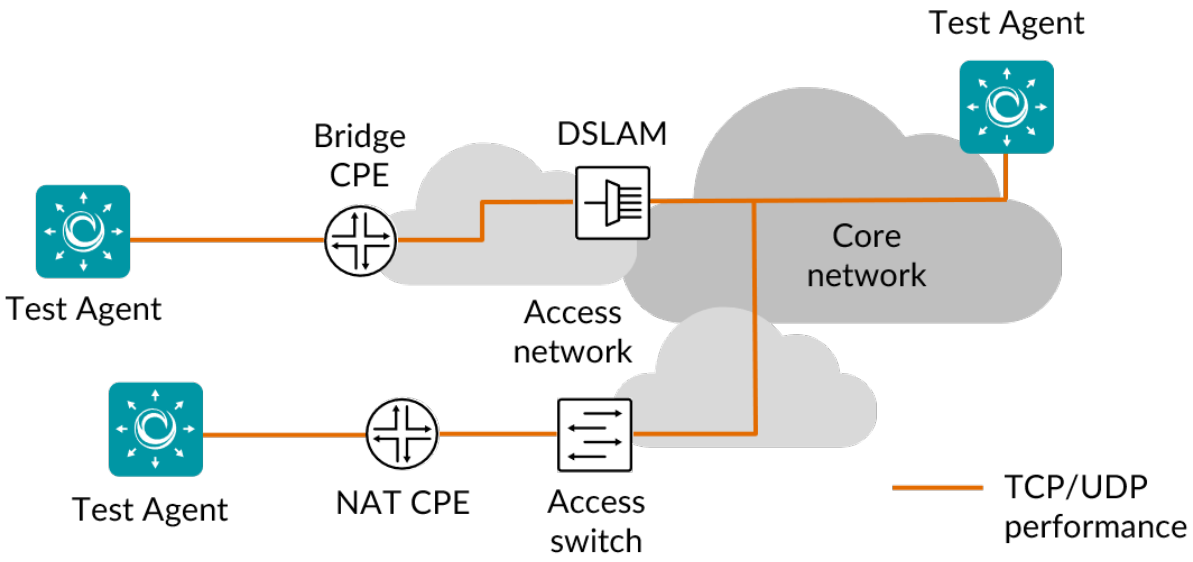
Task category	Tasks supporting IPv6
Network performance	UDP TCP Multicast UDP VoIP UDP
IPTV and OTT video	IPTV MPEG IPTV channel zapping
HTTP and DNS	DNS HTTP
SIP	–
Ethernet service activation	–
Transparency	L2 Ethernet control protocols L2 transparency – Custom Ethertype L2 transparency – Custom VLAN L2 transparency – Ethertypes L2 transparency – IP L2 transparency – IPv6 L2 transparency – MAC address limit L2 transparency – Multicast L2 transparency – VLAN
Reflector-based	ETH-DM ETH-LB ETH-SLM Ping BWPing TWAMP full TWAMP Light Path trace UDP loopback
Security	–

8.4 Network performance testing

8.4.1 Introduction to TCP/UDP performance testing

Test Agents include a powerful active traffic generator tool that can send TCP and UDP traffic to other Test Agents, enabling you to test and troubleshoot your network connections.

The receiving Test Agents at the other end of the network link will calculate packet loss, jitter, and minimum/average/maximum delay, as well as determine if your network has the expected performance and quality for applications such as IP telephony, Citrix, and videoconferencing.



For TCP and UDP traffic generation, Routing Active Testing supports both point-to-point and hub-and-spoke setups. In a hub-and-spoke setup, all selected Routing Active Testing "clients" will exchange traffic with the Test Agent selected as "server". For UDP, TCP and VoIP UDP, Routing Active Testing also supports the full-mesh setup, where the system automatically generates connections between all selected Routing Active Testing clients.

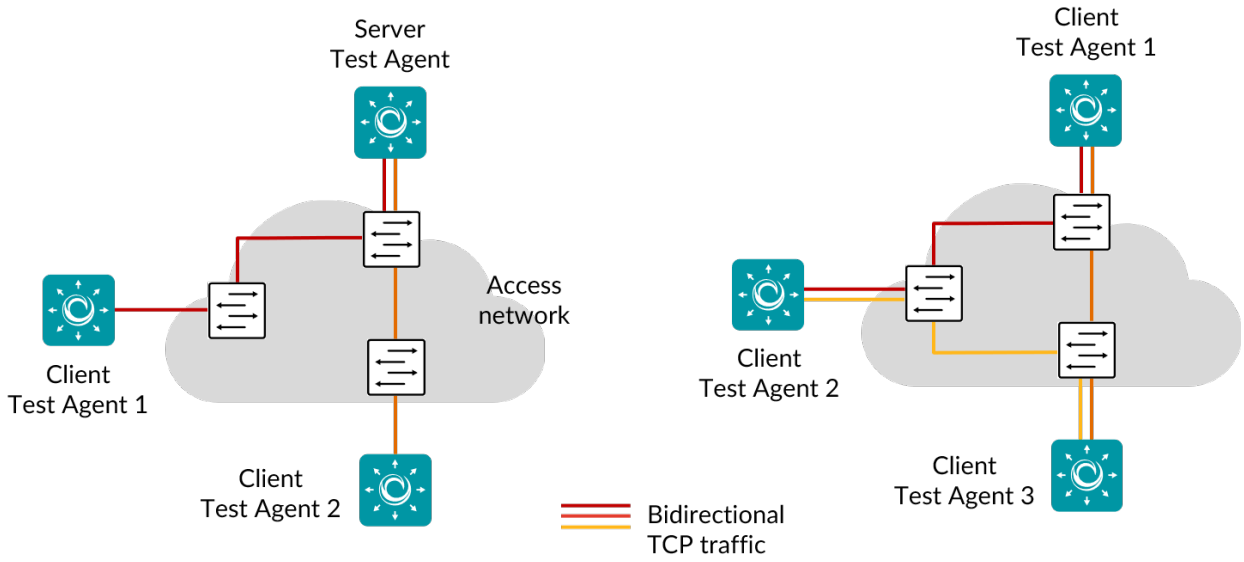
TCP and UDP traffic is initiated by the clients, and traffic from the server back to the clients is sent on the same ports. This makes it possible to send traffic through a NAT router, as long as the Test Agent placed behind the NAT is designated as a client.

The performance metrics in tests affect fail/pass criteria, and in monitoring sessions they are compared to the thresholds in the SLA.

For details on individual TCP/UDP performance test tasks, see the following pages:

- *TCP* (page 291)
- *Multisession TCP* (page 294)
- *UDP* (page 299)
- *VoIP UDP* (page 303)
- *Multicast UDP* (page 305)
- *TCP throughput test according to RFC 6349* (page 312)
- *QoS policy profiling* (page 316)

8.4.2 TCP



The pictures show hub-and spoke (left) and full-mesh (right) TCP traffic generation, used for performance and throughput measurements.

TCP is a very commonly used protocol, employed for everything from Internet web browsing to client–server applications. By running a TCP task, you will learn about the achievable performance of your network link. Routing Active Testing uses a standard Cubic TCP implementation; for more information, see the page on *TCP implementation* (page 517) in Routing Active Testing.

Client Test Agents can be placed behind NAT, since traffic will be initiated from the clients to the server.

When a TCP task starts, the Test Agents will start sending a TCP session between them. This TCP session will compete for the available bandwidth with all other traffic on the network link, thus giving a view of the available performance on that link.

The size of the TCP packet actually sent is determined by the OS/TCP algorithm, just as for a regular user. This is done to faithfully emulate the end-user experience running TCP-based applications.

This task works with both IPv4 and IPv6.

8.4.2.1 Prerequisites

To run TCP measurements you need to have at least two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then add a TCP task to your test or monitor and fill in the mandatory parameters below:

8.4.2.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Setup type: Select how to set up the measurement: "Client-Server" or "Full-Mesh". When Client-Server is selected, certain parameters take an "Up" or "Down" prefix to indicate the direction of transmission, whereas for Full-Mesh no such prefix is needed. Default: Client-Server.
- Server: Test Agent interface that is going to act as server. If a NAT router or firewall is present, the server must be located on the outer (public) side.
- Clients: Test Agent interfaces that will participate in the TCP measurement and exchange traffic with the server. The clients can be placed behind NAT, since traffic will be initiated from the clients to the server.
- Number of flows: Number of TCP sessions. If more than one flow is specified, the Client port setting is ignored, and all client ports will be ephemeral. Min: 1. Max: 64. Default: 1. Note: Even if multiple flows are used, a single set of statistics is shown with aggregate values picked for each parameter.
- Direction: One of: Down (from server to clients), Up (from clients to server), or Bidirectional (in both directions at the same time).
- Up rate (Mbit/s): Upstream (client-to-server) target rate for TCP. The rate specified is TCP goodput, thus including only the TCP payload. If this field is left blank, TCP will not use any rate limitation and will use TCP congestion control. Min: 0.01 Mbit/s. Max: 1,000 Gbit/s.
- Down rate (Mbit/s): Downstream (server-to-client) target rate for TCP. The rate specified is TCP goodput, thus including only the TCP payload. If this field is left blank, TCP will not use any rate limitation and will use TCP congestion control. Min: 0.01 Mbit/s. Max: 1,000 Gbit/s.
- Port: TCP server port to which clients will send traffic. Range: 1 … 65535. Default: 5000.
- Client port: *(Optional)* TCP client port from which clients will send traffic. If this is omitted, the client will select a port. Range: 1 … 65535.

Thresholds for errored seconds (ES)

- Up/Down min rate (Mbit/s): Minimum data rate thresholds. If the TCP data rate (goodput) during one second is lower than this value, an errored second is triggered. If a threshold is left undefined, no errored seconds will be generated for that direction. Min: 0 Mbit/s. Default: 0 Mbit/s.
- Up/Down max rate (Mbit/s): Maximum data rate thresholds. If the TCP data rate (goodput) during one second is higher than this value, an errored second is triggered. If a threshold is left undefined, no errored seconds will be generated for that direction. Min: 0 Mbit/s. Default: By default this field is left blank.

TCP retransmissions cannot cause errored seconds directly, since Routing Active Testing does not count TCP retransmissions. However, as retransmissions will lower the achieved throughput (goodput), you can obtain an indirect measure of them by setting a suitable threshold for the up and down rates.

Advanced

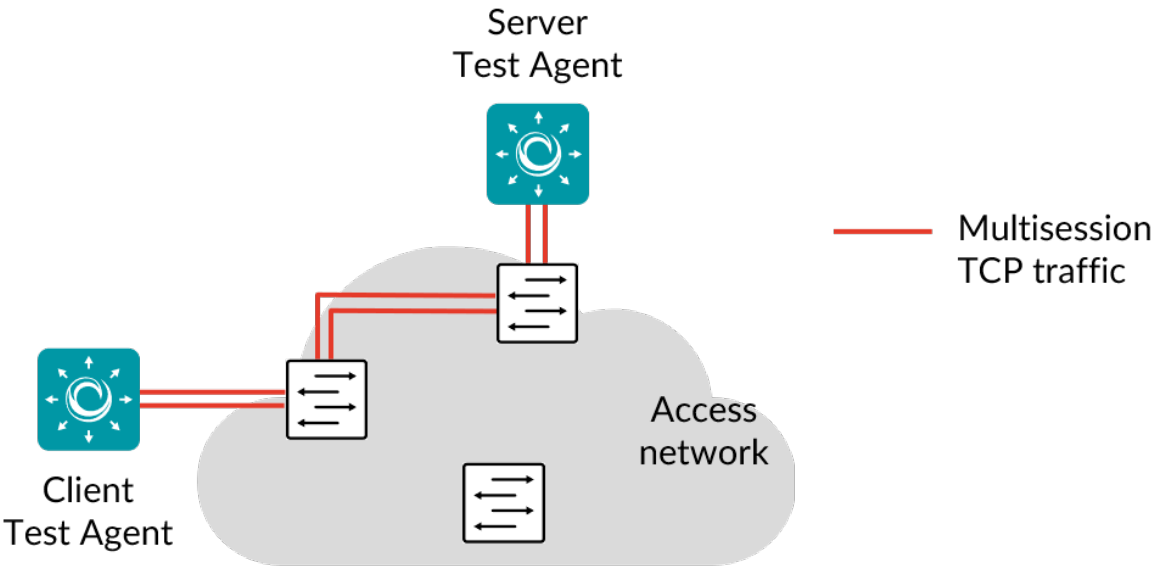
- Up/Down DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: "0 / IPP 0".
- Up/Down VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See *this page* (page 516). Range: 0 … 7. Default: 0. Note: When a Test Agent Application attempts to configure PCP settings in outgoing IP packets, it cannot be guaranteed that the settings are indeed carried through. This is because the Test Agent Application does not control the host it is running on and its interface configurations.
- Socket send buffer (bytes): The socket send buffer size in bytes. *Optional.* Min: 1 byte. Max: 10,000,000 bytes. Default: By default this field is left blank, and Linux will then use a dynamic buffer that is adjusted according to usage and available memory.
- Socket receive buffer (bytes): The socket receive buffer size in bytes. *Optional.* Min: 1 byte. Max: 10,000,000 bytes. Default: By default this field is left blank, and Linux will then use a dynamic buffer that is adjusted according to usage and available memory.
- Client connect delay (s): Configurable delay in seconds until the client initiates a TCP connection towards the server. *Optional.* Default: By default this field is left blank, and no delay is applied.
- Proxy server: If set, the specified server and port will be used as TCP proxy.
- Proxy server port: Port to use as TCP proxy port. Range: 1 … 65535.

8.4.2.3 Result metrics

- **Rate (Mbit/s):** Received TCP data rate (goodput).
- **ES:** Aggregated errored seconds, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.4.3 Multisession TCP

This task generates TCP traffic in the form of a specified number of parallel TCP sessions. The picture below shows point-to-point multisession TCP:



By running a Multisession TCP task, you will gain insight into the performance (available bandwidth) of your network. Hundreds of sessions will often be initiated when using peer-to-peer software or when browsing certain websites with many objects.

Routing Active Testing uses a standard Cubic TCP implementation. For more information, see the page on *TCP implementation* (page 517) in Routing Active Testing.

When a Multisession TCP task starts, the Test Agent will start the selected number of TCP sessions. There is usually no risk in running five or ten simultaneous sessions during production hours; however, if the number of simultaneous sessions goes into the hundreds or thousands, you might overload NAT routers or simple firewalls, or even slow down Internet accesses. Such tests should therefore preferably be conducted outside of production hours.

A client Test Agent can be placed behind NAT, since traffic will be initiated by the client towards the server.

This task works only with IPv4.

8.4.3.1 Prerequisites

To run multisession TCP measurements you need to have two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then add a Multisession TCP task to your test or monitor and fill in the mandatory parameters below:

8.4.3.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Server: Test Agent interface that is going to act as server. If a NAT router or firewall is present, the server must be located on the outer (public) side.
- Client: Test Agent interface that will participate in the TCP measurement and exchange traffic with the server. The client can be placed behind a NAT, since traffic will be initiated by the client towards the server.
- Port: TCP server port to which client will send traffic. Range: 1 … 65535. Default: 5000.
- Connections: The number of parallel TCP sessions that will be set up. Min: 1. Max: 10,000. Default: 10.
- Direction: One of: Down (from server to client), Up (from client to server), or Bidirectional (in both directions at the same time).

Thresholds for errored seconds (ES)

- Number of connections (down direction): Threshold for triggering an errored second for the server-to-client direction. An ES is indicated if the number of connections falls below this value. Min: 0. Max: 10,000. Default: 10.
- Rate (down direction): Threshold for triggering an errored second for the server-to-client direction. An ES is indicated if the data rate (goodput) drops below this value. Min: 0.
- Number of connections (up direction): Threshold for triggering an errored second for the client-to-server direction. An ES is indicated if the number of connections falls below this value. Min: 0. Max: 10,000. Default: 10.
- Rate (up direction): Threshold for triggering an errored second for the client-to-server direction. An ES is indicated if the data rate (goodput) drops below this value. Min: 0.

8.4.3.3 Result metrics

- **Rate (Mbit/s):** Total data rate (goodput) measured for all TCP sessions combined.
- **Connected:** Number of connected TCP sessions.
- **Active:** Number of active TCP sessions.
- **Disconnects:** Number of TCP session disconnects.
- **ES total (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES rate (%):** Errored second percentage for data rate (up and down directions aggregated).
- **ES connected (%):** Errored second percentage for number of TCP connections (up and down directions aggregated).
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

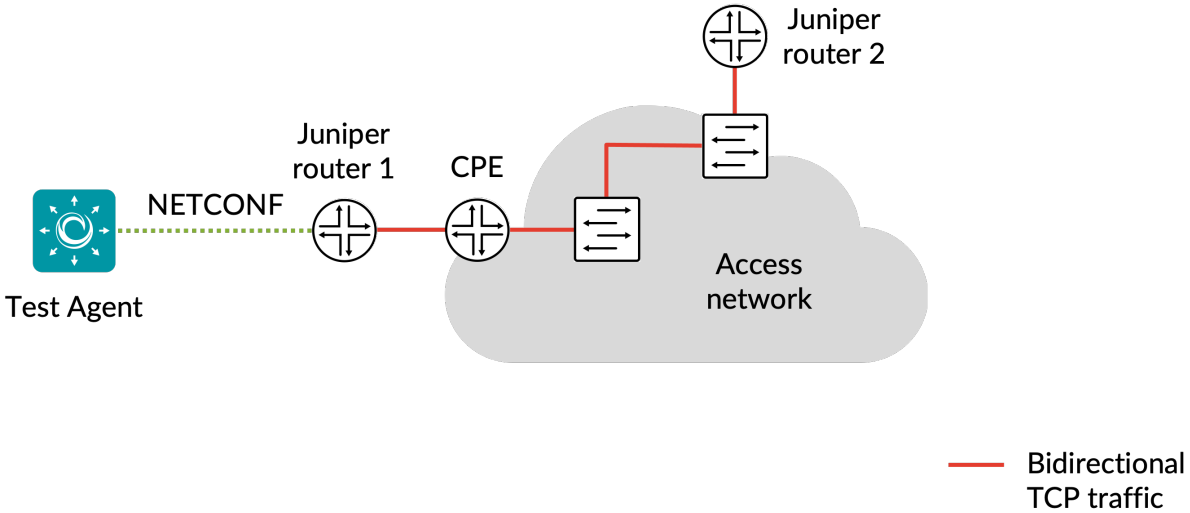
8.4.4 Junos TCP

In this task, a Test Agent Application connects to one or several Junos devices (defined as *network devices* (page 35) in Routing Active Testing) via the NETCONF protocol and accesses TCP sessions running on these devices (these sessions have not been configured in Routing Active Testing). The Test Agent collects measurement results from the TCP sessions, evaluates errored second thresholds, and reports all results back to Control Center.

Each TCP session running on a Junos device is identified as a "test" belonging to an "owner" (specified as `services rpm probe <owner> test <test>` in the Junos CLI). The test and owner are shown along with the results in Control Center so that you can correlate them with the TCP sessions on the Junos device.

Note: A Test Agent Application is required for this task; the Test Agent Appliance does not support it. The Junos TCP task is also different from *regular TCP* (page 291) in that the Test Agent does not itself conduct the measurements.

IPv6 is supported in the communication between the Test Agent and the Junos device.



8.4.4.1 Prerequisites

To perform Junos TCP measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

As regards each targeted Junos device, the following holds:

- The functionality has been verified to work on the following Juniper device models, but it might also work on other devices:
  - vMX
  - MX204
- The functionality has been verified for Junos versions 18.3–20.2. Junos Evolved is not supported.
- There must be network connectivity from the Test Agent to the device (default TCP port: 830).
- You must have a user account on the device to be able to log in to it and retrieve measurement data. How to create a user account is described *here* (page 298).
- The Junos device must be configured as a *network device* (page 35) in the Routing Active Testing inventory.
- The probe of Junos device 1 (see diagram) must be configured with the correct probe type:

```
set services rpm probe <owner> test <test> probe-type tcp-ping
```

- The probe of Junos device 1 must also be configured with the target address:

```
set services rpm probe <owner> test <test> target <address_type> <address>
```

  Example (IPv4):

```
set services rpm probe owner1 test t1 target address 192.168.0.1
```

  Example (IPv6):

```
set services rpm probe owner1 test t1 target inet6-address 2070::1
```

- Junos device 2 (see diagram) must have a TCP server port configured. Run this command in edit mode:

```
set services rpm probe-server tcp port <port number>
```

- To start Junos TCP, the probe of Junos device 1 (see diagram above) must be configured with the port number of Junos device 2 (see preceding bullet):

```
set services rpm probe <owner> test <test> destination-port <port number>
```

- The Junos device must be running TCP measurements when you execute the Junos TCP task. The relevant Junos documentation is found here.

Once you have finished the above preparations, you can add a Junos TCP task to your test or monitor and fill in the mandatory parameters as shown below.

8.4.4.2 Junos device configuration

This section is about suitable configuration of the Junos device. This needs to be done directly on the device and cannot be performed in the Routing Active Testing task. For details, please refer to Junos device documentation.

Creating a user account on a Junos device

The user account should have limited permissions and can be created as follows:

```
configure
set system login user <username> class read-only
set system login user <username> authentication plain-text-password
New password: <password>
commit
```

Configuring TCP history size

The configuration parameter `services rpm probe <owner> test <test> history-size` in the Junos device specifies how many historical probe results are stored for each owner and test. (Regarding these concepts, see the *introductory section* (page 296) above.)

The Test Agent will fetch new probe results from this result set every 5 seconds. To have some margin for communication delays between the Test Agent and the Junos device, we recommend that you configure `history-size` to correspond to at least 1 minute. For example, with `probe-interval` set to 10 seconds, we recommend a `history-size` of at least 6.

8.4.4.3 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Client: Test Agent interface that will connect to the Junos device.
- Network devices: Junos devices that are running TCP tasks and will be accessed by the Test Agent.
- Filter based on owner: Only collect results from the specified owner as configured on the Junos device. If you leave this blank, results will be collected from all owners.
- Filter test session: Only collect results from the specified test as configured on the Junos device. If you leave this blank, results will be collected from all tests.

Thresholds for errored seconds (ES)

- RTT threshold (ms): Round-trip time threshold for triggering an errored second. If the round-trip time exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Jitter threshold (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.

Note: Loss will always trigger an errored second.

Advanced

- Collection interval (s): The interval at which results are collected from the Junos devices. Min: 5s Max: 300s Default: 5s.
- Session timeout (s): The time after which idle sessions will be removed.

8.4.4.4 Result metrics

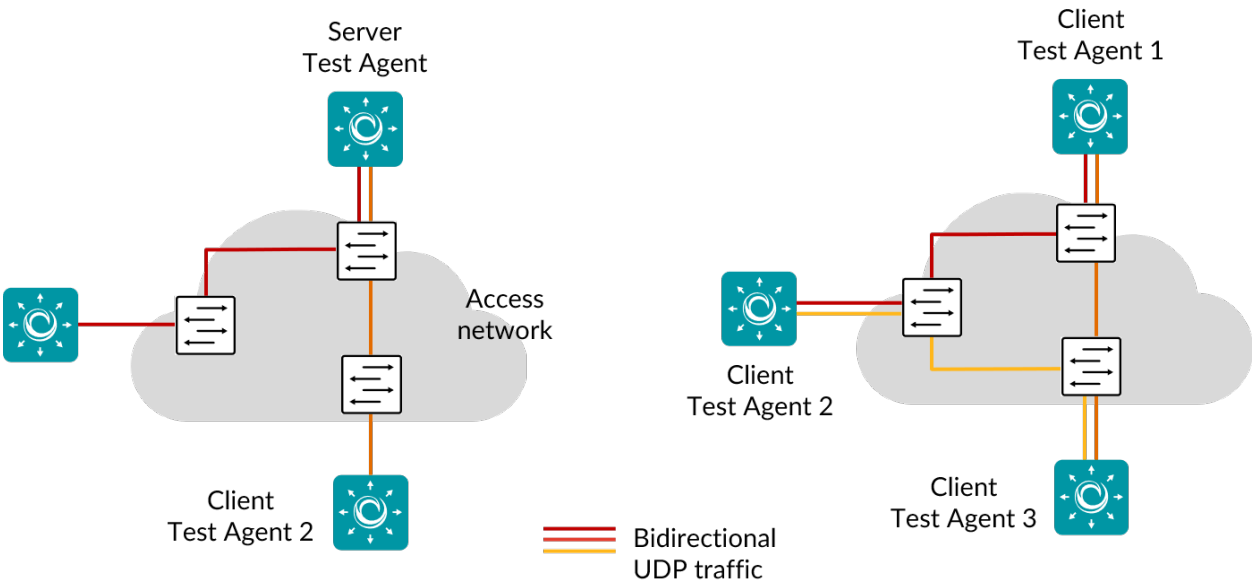
- **RTT (ms):** Delay between the transmission of a probe and the arrival of its response.
- **Round-trip jitter (ms):** Difference between the current round-trip time and the previous measurement.
- **Round-trip interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in IETF RFC 1889, calculated for the full round-trip.

Note: Jitter and interarrival jitter are calculated differently from what is called "delay variation" in other tasks.

- **Loss (%):** Round-trip packet loss in percent.
- **ES (s):** Aggregated number of errored seconds (ES), taking into account all types of error.
- **ES loss (s):** Number of errored seconds caused by loss.
- **ES RTT (s):** Number of errored seconds caused by round-trip time.
- **ES jitter (s):** Number of errored seconds for caused by jitter.

8.4.5 UDP

The pictures show hub-and spoke (left) or full-mesh (right) UDP traffic generation between two or more Test Agents for measuring link performance:



Running a UDP task will help you understand if your network is good enough for quality-demanding services such as client–server applications and videoconferencing.

When a UDP task starts, the Test Agents will generate traffic at the rate you specify. The rate is the Layer 2 Ethernet rate, also known as the Committed Information Rate (CIR). It includes the Ethernet headers with the CRC checksum but not the Frame Gap, Preamble, or Start of Frame Delimiter. The UDP flow sent by the sender Test Agent includes timestamps and sequence numbers, so that the receiving Test Agent can calculate one-way delay, jitter, packet loss, and packet misorderings.

Examples of network requirements:

- *Videoconferencing:* Loss < 1%, jitter < 30 ms, one-way delay < 150 ms
- *Client–server:* Loss < 2%, one-way delay < 100 ms

Such requirements can be immediately expressed as thresholds, set individually for each UDP task.

This task works with both IPv4 and IPv6.

8.4.5.1 Prerequisites

To run UDP measurements you need to have at least two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then add a UDP task to your test or monitor and fill in the mandatory parameters below:

8.4.5.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Setup type: Select how to set up the measurement: “Client-Server” or “Full-Mesh”. When Client-Server is selected, certain parameters take an Up/Down prefix, whereas for Full-Mesh this prefix is absent. Default: Client-Server.
- Server: Test Agent interface that is going to act as server. If a NAT router or firewall is present, the server must be located on the outer (public) side.
- Clients: Test Agent interfaces exchanging UDP traffic with the server. The clients can be placed behind NAT, since traffic will be initiated from the clients to the server.
- Number of flows: Number of UDP flows. If more than one flows is specified, the Client port setting is ignored and all client ports will be ephemeral. Min: 1. Max: 64. Default: 1.

Note: Even if multiple flows are used, a single set of statistics is shown with aggregate values picked for each parameter: average rate, average loss, sum of misorderings, minimum delay, average delay, maximum delay, maximum jitter, and sum of ES.

Note: Using multiple flows may lower loss figures. This is because distributing data across several flows may lead to fewer misorderings (misordered packets are counted as lost). For example, suppose we are using a single flow and packets arrive in the order 0, 2, 1, 3, 4. This sequence contains one misordering. Now suppose that we instead use two flows with packets 0, 1, 4 arriving in one flow and packets 2, 3 in the other. In this case we have no misorderings. The same tendency towards elimination of misorderings will prevail generally.

- Direction: One of: Down (from server to clients), Up (from clients to server), or Bidirectional (in both directions at the same time).
- Up rate (Mbit/s): Client-to-server target data rate. Min: 0.01 Mbit/s. Max: 1,000 Gbit/s.
- Down rate (Mbit/s): Server-to-client target data rate. Min: 0.01 Mbit/s. Max: 1,000 Gbit/s.
- Rate (Mbit/s): Client-to-client target data rate when running in Full-mesh. Min: 0.01 Mbit/s. Max: 1,000 Gbit/s.
- Port: UDP server port to which clients will send traffic. Range: 1 … 65535. Default: 5000.
- Client port: *(Optional)* UDP client port from which clients will send traffic. If this is omitted, the client will select a port. Range: 1 … 65535.

Thresholds for errored seconds (ES)

- Up/Down loss (%): Packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Max: 100%. Default: 0%.
- Up/Down jitter (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and reflector exceeds this value during one second, an ES will be indicated. Min: 1 ms. Max: 1000 ms. No default.
- Up/Down delay (ms): One-way delay threshold for triggering an errored second. If the delay between server and reflector exceeds this value during one second, an ES will be indicated. Min: 1 ms. Max: 1000 ms. No default.
- Up/Down expected DSCP: The expected *Differentiated Services Code Point or IP Precedence* (page 511) that the IP packets will have at the receiving side. If the received DSCP value does not match, an ES will be indicated. By default, no DSCP validation is done (------ selected in drop-down box).

Thresholds for severely errored seconds (SES)

- Up/Down loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). Min: 0%. No default.
- Up/Down jitter (ms): Jitter (delay variation) threshold for triggering a severely errored second. Min: 0 ms. No default.
- Up/Down delay (ms): One-way delay threshold for triggering a severely errored second. Min: 0 ms. No default.

Advanced

- Don't fragment datagram: Controls whether the datagram that exceeds the MTU is allowed to be fragmented. Be aware that enabling fragmentation (by specifying the "No" option) may cause performance degradation both in the network and in the sending or receiving Test Agents.

  Note: This setting has no effect on IPv6 packets.

- Up/Down Ethernet frame size (bytes): Size of Layer 2 Ethernet frame for the flow. See *this page* (page 512). Min: 64 bytes for IPv4; 84 bytes for IPv6. Max: 9018 bytes. Default: 1518 bytes.
- Up/Down DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: "0 / IPP 0".
- Up/Down VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See *this page* (page 516). Range: 0 … 7. Default: 0. Note: When a Test Agent Application attempts to configure PCP settings in outgoing IP packets, it cannot be guaranteed that the settings are indeed carried through. This is because the Test Agent Application does not control the host it is running on and its interface configurations.
- Socket send buffer (bytes): Send socket buffer (in bytes) used for the flow. This is used by the kernel to temporarily store packets before sending. Streams with higher rate need a larger buffer. *Optional.* Min: 2048 bytes. Max: 10,000,000 bytes. Default: By default this field is left blank. Routing Active Testing will then calculate the buffer size based on rate and Ethernet frame size. *Example:* For a 10 Mbit/s flow with Ethernet frame size 1518 bytes, a send buffer of size 14,544 bytes is used by default.
- Socket receive buffer (bytes): Receive socket buffer (in bytes) used for the flow. This buffer is used by the kernel to temporarily store incoming packets. Streams with higher rate need a larger buffer. The socket buffer also limits the burst that the receiver is able to receive without packet loss. *Optional.* Min: 2048 bytes. Max: 10,000,000 bytes. Default: By default this field is left blank. Routing Active Testing will then calculate the buffer size based on rate and Ethernet frame size. *Example:* For a 10 Mbit/s flow with Ethernet frame size 1518 bytes, a receive buffer of size 95,312 bytes is used by default.

8.4.5.3 Result metrics

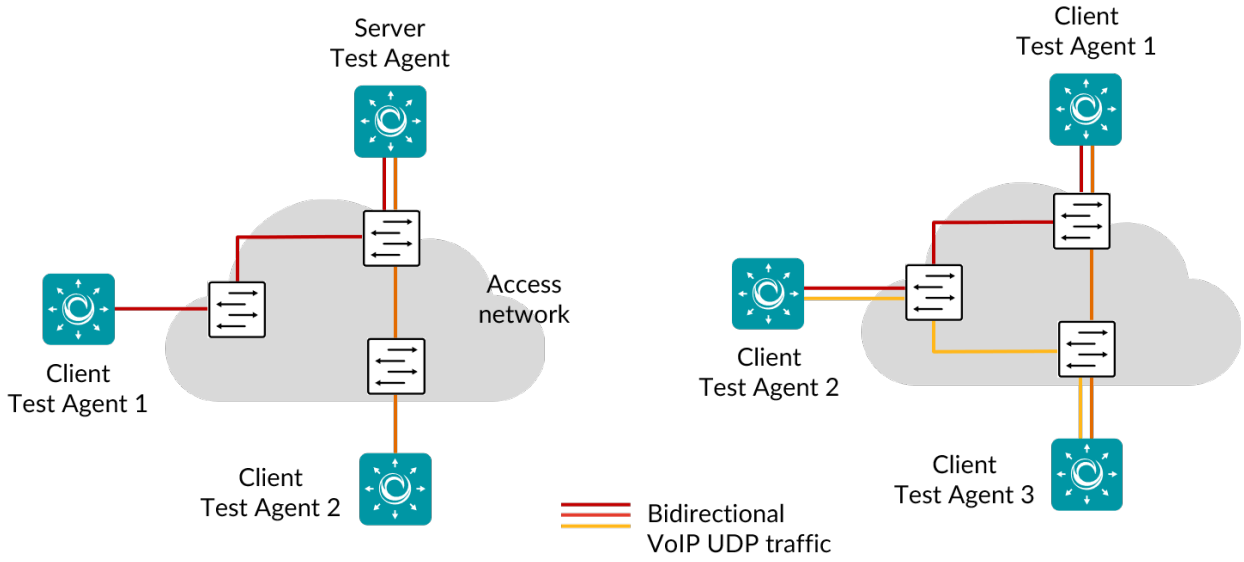
- **Rate (Mbit/s):** Ethernet rate of the UDP flow.
- **Loss (%):** Packet loss in percent.
- **Misordered (packets):** Number of misordered packets.
- **Delay min (ms):** Minimum one-way delay.
- **Delay average (ms):** Average one-way delay.
- **Delay max (ms):** Maximum one-way delay.

Note: The one-way delay metrics are only reported if certain requirements on NTP clock synchronization are met. Read more *here* (page 190).

- **Jitter (ms):** *Jitter (delay variation)* (page 474).
- **Received packets (packets):** Number of received packets.
- **Lost packets (packets):** Number of lost packets.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES loss (%):** Errored second percentage for packet loss.
- **ES delay (%):** Aggregated errored second percentage, taking into account delay and delay variation.
- **ES DSCP (%):** Accumulated errored second percentage for DSCP.
- **Severely errored seconds (%):** Aggregated severely errored second (SES) percentage, taking into account delay and delay variation.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.4.6 VoIP UDP

This task generates VoIP media flows over UDP, using a selected voice codec, and measures voice quality. The testing can be done in a hub-and-spoke (left) or in a full-mesh (right) configuration, as shown in the picture below.



By running a VoIP UDP task, you will be able to measure how your network influences the quality of VoIP traffic. An objective quality score on the MOS scale is calculated for VoIP based mainly on network jitter and packet loss.

When a VoIP UDP task starts, the Test Agents will generate UDP traffic with a fixed frame size and bit rate, matching the codec you have selected. For example, for G.711 the frame size is 218 bytes, and the bit rate is 87.2 kbit/s.

No SIP or H.323 signaling is captured.

This task works with both IPv4 and IPv6.

8.4.6.1 Prerequisites

To run VoIP UDP measurements you need to have at least two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then add a VoIP UDP task to your test or monitor and fill in the mandatory parameters below:

8.4.6.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Setup type: Select how to set up the measurement: "Client-Server" or "Full-Mesh". Default: Client-Server.
- Server: Test Agent interface that is going to act as server. If a NAT router or firewall is present, the server must be located on the outer (public) side.
- Clients: Test Agent interfaces that will participate in the VoIP UDP measurement and exchange VoIP-like traffic with the server. Client Test Agents can be placed behind NAT, since traffic will be initiated by the clients towards the server.
- Number of flows: Number of VoIP flows. Min: 1. Max: 64. Default: 1.

Note: Even if multiple flows are used, a single set of statistics is shown with aggregate values picked for each parameter: average rate, average loss, sum of misorderings, minimum delay, average delay, maximum delay, maximum jitter, and sum of ES.

Note: Using multiple flows may lower loss figures. This is because distributing data across several flows may lead to fewer misorderings (misordered packets are counted as lost). For example, suppose we are using a single flow and packets arrive in the order 0, 2, 1, 3, 4. This sequence contains one misordering. Now suppose that we instead use two flows with packets 0, 1, 4 arriving in one flow and packets 2, 3 in the other. In this case we have no misorderings. The same tendency towards elimination of misorderings will prevail generally.

- Codec: Voice codec used. The following voice codecs are supported:
  - G.711: Frame size 218 bytes, bit rate 87.2 kbit/s (default)
  - G.723: Frame size 82 bytes, bit rate 21.9 kbit/s
  - G.729: Frame size 78 bytes, bit rate 31.2 kbit/s
  - GSM EFR: Frame size 89 bytes, bit rate 35.6 kbit/s
- Port: UDP destination port for VoIP flows. Range: 1 … 65535. Default: 5000.

Thresholds for errored seconds (ES)

- MOS: Mean Opinion Score threshold for triggering an errored second. See *this page* (page 474), which also details how the MOS is calculated. Range: 1 … 5. Default: 4.
- Up/Down expected DSCP: The expected Differentiated Services Code Point or IP Precedence at the receiving side. By default, no DSCP validation is done (------ selected in drop-down box).

Advanced

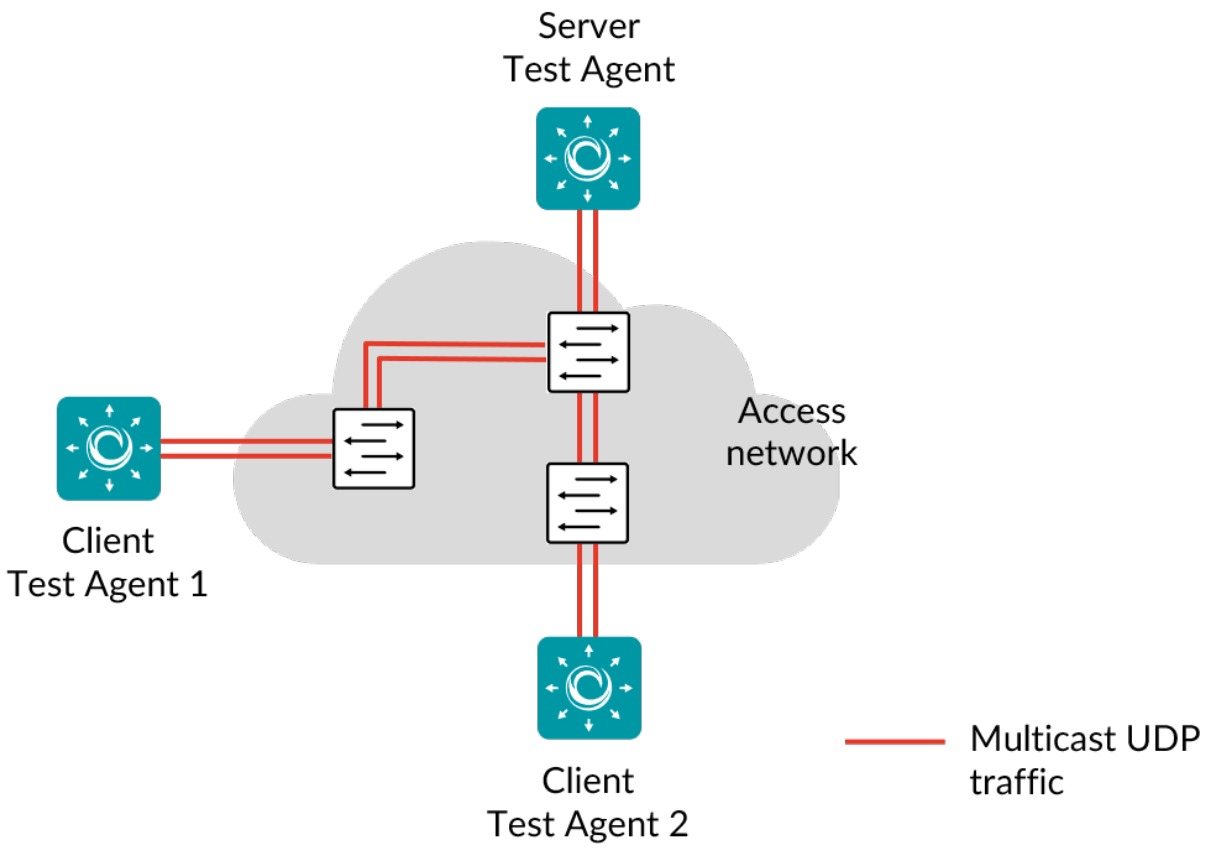
- Up/Down DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: "0 / IPP 0".
- Up/Down VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See *this page* (page 516). Range: 0 … 7. Default: 0.

8.4.6.3 Result metrics

- **Rate (Mbit/s):** Ethernet rate of the VoIP UDP flow.
- **Loss (%):** Packet loss in percent.
- **Misordered (packets):** Number of misordered packets.
- **Delay min (ms):** Minimum one-way delay.
- **Delay average (ms):** Average one-way delay.
- **Delay max (ms):** Maximum one-way delay.
- **Jitter (ms):** *Jitter (delay variation)* (page 474).
- **Received packets (packets):** Number of received packets.
- **Lost packets (packets):** Number of lost packets.
- **MOS:** Estimated voice quality MOS, calculated from network metrics.
- **ES (%):** Aggregated errored second (ES) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.4.7 Multicast UDP

This task generates multicast UDP traffic by means of a server Test Agent, which is joined by a number of client Test Agents.



Multicast is commonly used as a transport mechanism for services like IPTV, as well as for updating many PCs at the same time. By running a Multicast UDP task, you will learn how well your network handles multicast traffic in terms of end-to-end delay, jitter, and packet loss.

When a Multicast UDP task starts, the server Test Agent will start generating traffic towards the destination multicast address you have specified. The client Test Agents will then try to join that multicast address and, if successful, calculate one-way delay, jitter, packet loss, and packet misorderings for the received network flow.

This task works with both IPv4 and IPv6.

8.4.7.1 Prerequisites

To run Multicast UDP measurements you need to have at least two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then add a Multicast UDP task to your test or monitor and fill in the mandatory parameters below:

8.4.7.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Server: Test Agent interface that will generate multicast traffic.
- Clients: Test Agent interfaces that will join the multicast traffic sent by the server. Client Test Agents can be placed behind NAT, since traffic will be initiated from the clients to the server.
- Rate (Mbit/s): Bit rate of multicast flow in Mbit/s.
- Multicast address: Multicast address to use. Default: 239.1.1.1.
- Port: UDP destination port of multicast flow. Range: 1 … 65535. Default: 5000.

Thresholds for errored seconds (ES)

- Loss (%): Packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Max: 100%. Default: 0%.
- Delay (ms): One-way delay threshold for triggering an errored second. If the delay from server to clients exceeds this value during one second, an ES will be indicated. Min: 0%. No default.
- Jitter (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and clients exceeds this value during one second, an ES will be indicated. Min: 0%. No default.
- Expected DSCP: The expected *Differentiated Services Code Point or IP Precedence* (page 511) that the IP packets will have at the receiving side. If the received DSCP value does not match, an ES will be indicated. By default, no DSCP validation is done (------ selected in drop-down box).

Thresholds for severely errored seconds (SES)

- Loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). Min: 0%. Max: 100%. No default.
- Delay (ms): One-way delay threshold for triggering a severely errored second. Min: 0%. No default.
- Jitter (ms): Jitter (delay variation) threshold for triggering a severely errored second. Min: 0%. No default.

Advanced

- Ethernet frame size (bytes): Size of Layer 2 Ethernet frame for the network flow. See *this page* (page 512). Min: 64 bytes for IPv4; 84 bytes for IPv6. Max: 9018 bytes. Default: 1518 bytes.
- DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: "0 / IPP 0".
- VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See *this page* (page 516). Range: 0 … 7. Default: 0. Note: When a Test Agent Application attempts to configure PCP settings in outgoing IP packets, it cannot be guaranteed that the settings are indeed carried through. This is because the Test Agent Application does not control the host it is running on and its interface configurations.

8.4.7.3 Result metrics

- **Rate (Mbit/s):** Ethernet rate of the UDP flow.
- **Loss (%):** Packet loss in percent.
- **Misordered (packets):** Number of misordered packets.
- **Delay min (ms):** Minimum one-way delay.
- **Delay average (ms):** Average one-way delay.
- **Delay max (ms):** Maximum one-way delay.
- **Jitter (ms):** *Jitter (delay variation)* (page 474).
- **Received packets (packets):** Number of received packets.
- **Lost packets (packets):** Number of lost packets.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES loss (%):** Errored second percentage for packet loss.
- **ES delay (%):** Aggregated errored second percentage, taking into account delay and delay variation.
- **ES DSCP (%):** Accumulated errored second percentage for DSCP.
- **Severely errored seconds (%):** Aggregated severely errored second (SES) percentage, taking into account delay and delay variation.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

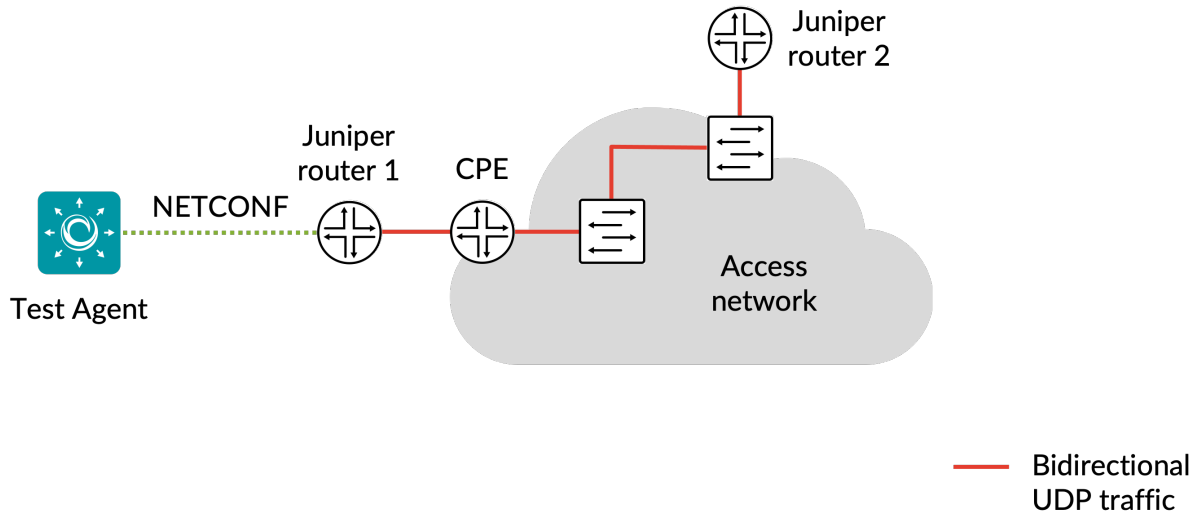
8.4.8 Junos UDP

In this task, a Test Agent Application connects to one or several Junos devices (defined as *network devices* (page 35) in Routing Active Testing) via the NETCONF protocol and accesses UDP sessions running on these devices (these sessions have not been configured in Routing Active Testing). The Test Agent collects measurement results from the UDP sessions, evaluates errored second thresholds, and reports all results back to Control Center.

Each UDP session running on a Junos device is identified as a "test" belonging to an "owner" (specified as `services rpm probe <owner> test <test>` in the Junos CLI). The test and owner are shown along with the results in Control Center so that you can correlate them with the UDP sessions on the Junos device.

Note: A Test Agent Application is required for this task; the Test Agent Appliance does not support it. The Junos UDP task is also different from *regular UDP* (page 299) in that the Test Agent does not itself conduct the measurements.

IPv6 is supported in the communication between the Test Agent and the Junos device.



8.4.8.1 Prerequisites

To perform Junos UDP measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

As regards each targeted Junos device, the following holds:

- The functionality has been verified to work on the following Juniper device models, but it might also work on other devices:
  - vMX
  - MX204
- The functionality has been verified for Junos versions 18.3–20.2. Junos Evolved is not supported.
- There must be network connectivity from the Test Agent to the device (default TCP port: 830).
- You must have a user account on the device to be able to log in to it and retrieve measurement data. How to create a user account is described *here* (page 310).
- The Junos device must be configured as a *network device* (page 35) in the Routing Active Testing inventory.
- The Junos device must be running UDP measurements when you execute the Junos UDP task. The relevant Junos documentation is found here.

Once you have finished the above preparations, you can add a Junos UDP task to your test or monitor and fill in the mandatory parameters as shown below.

8.4.8.2 Junos device configuration

This section is about suitable configuration of the Junos device. This needs to be done directly on the device and cannot be performed in the Routing Active Testing task. For details, please refer to Junos device documentation.

Creating a user account on a Junos device

The user account should have limited permissions and can be created as follows:

```
configure
set system login user <username> class read-only
set system login user <username> authentication plain-text-password
New password: <password>
commit
```

Configuring UDP history size

The configuration parameter `services rpm probe <owner> test <test> history-size` in the Junos device specifies how many historical probe results are stored for each owner and test. (Regarding these concepts, see the *introductory section* (page 308) above.)

The Test Agent will fetch new probe results from this result set every 5 seconds. To have some margin for communication delays between the Test Agent and the Junos device, we recommend that you configure `history-size` to correspond to at least 1 minute. For example, with `probe-interval` set to 10 seconds, we recommend a `history-size` of at least 6.

8.4.8.3 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Client: Test Agent interface that will connect to the Junos device.
- Network devices: Junos devices that are running UDP tasks and will be accessed by the Test Agent.

Thresholds for errored seconds (ES)

- RTT threshold (ms): Round-trip time threshold for triggering an errored second. If the round-trip time exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Jitter threshold (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.

Note: Loss will always trigger an errored second.

Advanced

- Filter based on owner: Only collect results from the specified owner as configured on the Junos device. If you leave this blank, results will be collected from all owners.
- Filter test session: Only collect results from the specified test as configured on the Junos device. If you leave this blank, results will be collected from all tests.
- Collection interval (s): The interval at which results are collected from the Junos devices. Min: 5s Max: 300s Default: 5s.
- Session timeout (s): The time after which idle sessions will be removed.

8.4.8.4 Result metrics

- **RTT (ms):** Delay between the transmission of a probe and the arrival of its response.
- **Round-trip jitter (ms):** Difference between the current round-trip time and the previous measurement.
- **Round-trip interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in IETF RFC 1889, calculated for the full round-trip.

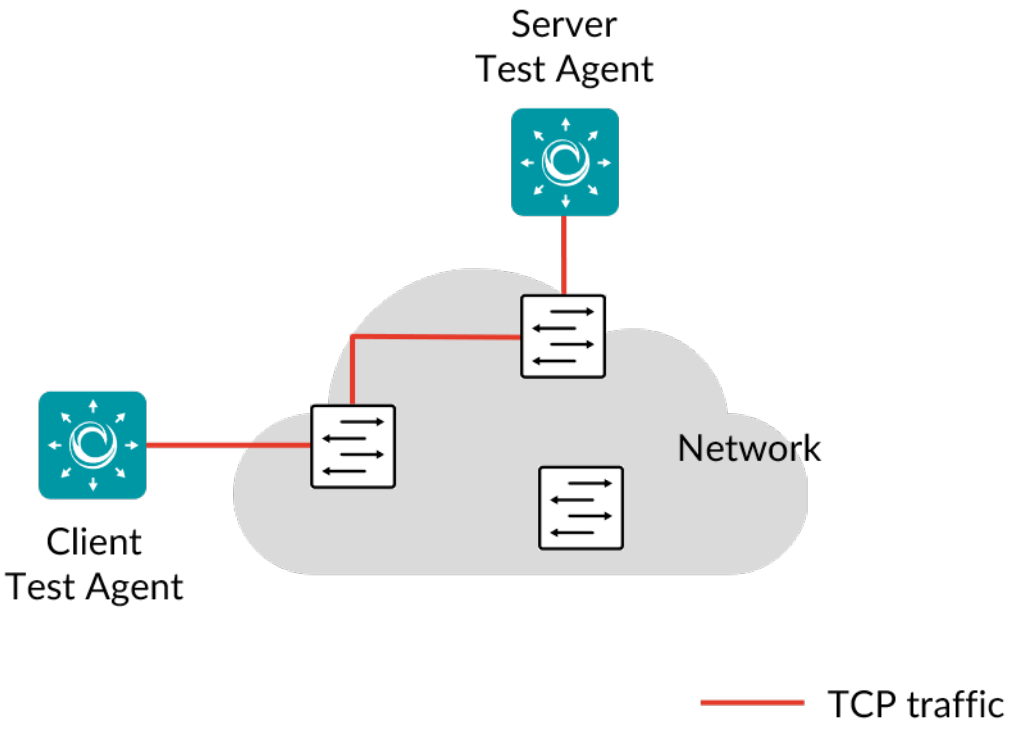
Note: Jitter and interarrival jitter are calculated differently from what is called "delay variation" in other tasks.

- **Loss (%):** Round-trip packet loss in percent.
- **ES (s):** Aggregated number of errored seconds (ES), taking into account all types of error.
- **ES loss (s):** Number of errored seconds caused by loss.
- **ES RTT (s):** Number of errored seconds caused by round-trip time.
- **ES jitter (s):** Number of errored seconds for caused by jitter.

8.4.9 TCP throughput test according to RFC 6349

This task follows ► IETF RFC 6349, which is a framework for TCP throughput testing. The RFC describes a practical methodology for measuring end-to-end TCP throughput in an IP network. The objective of the RFC was to provide a better throughput indication which takes the user experience into account.

The TCP throughput test is conducted between two Test Agents, one acting as server and the other acting as client, as illustrated below.



The test is executed in the following steps:

Step 1: Path MTU (Maximum Transmission Unit) is measured in the selected traffic direction or directions. This is accomplished by running a TCP session that measures the path MTU. If the measured MTU is lower than what is currently configured on that interface, the test will end with an error. You then need to adjust the interface's MTU setting. Note that the measurement can never return an MTU size larger than what is currently configured, even if larger MTUs are otherwise supported in the network path.

Step 2: The baseline RTT (round-trip time) is measured. This is accomplished by sending ICMP echo requests (pings) from the client to the server. Based on the measured RTT and the supplied bottleneck bandwidth (BB) values, we can calculate the bandwidth-delay product BDP = BB × RTT for each traffic direction.

> The BDP is the minimum sending and receiving socket buffer size required to reach the bottleneck bandwidth (minus overheads) with the given RTT value. The calculated buffer size is then multiplied by 20 (40 for bidirectional tests); this is needed as a safety margin, and also because the Linux kernel's buffer usage is not very efficient. The larger of the buffer sizes thus obtained is used for both directions.
>
> The bottleneck bandwidth is the Layer 2 capacity of the network path tested. This rate includes everything up to the Ethernet header.

Step 3: A number of TCP sessions are started with appropriate socket buffer settings as determined in step 2. You can request a specific number of TCP sessions; otherwise the number is automatically calculated based on the BDP. Max: 20 sessions.

The buffer size calculated in step 2 is the combined buffer size for all flows, so if multiple flows are used, each flow will use a part of that buffer. You can also request a specific buffer size setting per flow. If the bottleneck bandwidth cannot theoretically be reached with the given number of flows and buffer size, then the test will end with an error.

During this phase the ping measurement is still running, and RTT statistics are collected.

Step 4: The measurement is stopped, and statistics are collected. The buffer delay percentage shows how much the RTT increased while loading the TCP connection: (RTT – RTT_baseline) / RTT_baseline × 100. If there is a substantial increase, the socket buffer size calculated in step 2 might not be sufficient. In that case it is advisable to rerun the test with a manually adjusted buffer size value.

The **TCP transfer time ratio** shows the ratio of actual to ideal download time for a hypothetical file. This should be the same as the ratio between the theoretically achievable throughput and the measured combined TCP throughput. The theoretically possible throughput (TCP goodput) is calculated from the given BB by subtracting the overheads.

The **TCP efficiency** is the ratio of total received bytes to total sent bytes. A value less than 100% here indicates TCP retransmissions.

The test **fails** if the measured rates are lower than the rate thresholds, or the TCP efficiency is lower than the threshold set, or the transfer time ratio is higher than the threshold set.

Note that socket buffer sizes cannot be arbitrarily large. First, there is a system limit on the buffer size that can be allocated to a single TCP socket. Second, there is a system limit on the total buffer size that all TCP sessions can use. Finally, there is the limitation of the physical memory in the Test Agents. The test prints the system limits to the logs. If any limit is violated, the test ends with an error.

This task works only with IPv4.

Note: This test requires exclusive access to the Test Agent, meaning that no other tests or monitors can be assigned to the Test Agent.

8.4.9.1 Prerequisites

To run a TCP throughput test you need to have two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then create a new TCP test and fill in the mandatory parameters below:

8.4.9.2 Parameters

General

- Server: The Test Agent interface that will act as server.
- Client: The Test Agent interface that will act as client.
- Server port: The server TCP port. Range: 1 … 65535. Default: 5000.
- Traffic direction: The direction(s) of TCP traffic. One of: Upstream (from client to server), Downstream (from server to client), or Bidirectional (in both directions at the same time). Default: Downstream.
- Bottleneck bandwidth from server to client (Mbit/s), Bottleneck bandwidth from client to server (Mbit/s): Both of these bandwidths are specified on the Ethernet level and include the CRC but not the Inter Frame Gap, Preamble, or Start of Frame Delimiter. On a 100 Mbit/s interface, the maximum throughput is around 98.7 Mbit/s. Range: 0.1 … 10,000 Mbit/s. No default.

- DSCP: The *Differentiated Services Code Point or IP Precedence* (page 511) to be used in IP packet headers. The available choices are listed in the drop-down box. Default: “0 / IPP 0”.
- VLAN priority (PCP): The *Priority Code Point* (page 516) to be used in the VLAN header. Range: 0 … 7. Default: 0.
- Test duration (seconds): The duration of this test step in seconds. Min: 30 s. Max: 600 s. Default: 60 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

Thresholds

- Rate (down) for pass criteria (Mbit/s): The server-to-client TCP rate threshold for passing the test. Range: 0.1 … 10,000 Mbit/s. No default.
- Rate (up) for pass criteria (Mbit/s): The client-to-server TCP rate threshold for passing the test. Range: 0.1 … 10,000 Mbit/s. No default.
- Transfer time ratio (down): Threshold for server-to-client transfer time ratio, i.e. the maximum allowed ratio of actual to ideal TCP transfer time. Range: 1 … 1000. No default.
- Transfer time ratio (up): Threshold for client-to-server transfer time ratio, i.e. the maximum allowed ratio of actual to ideal TCP transfer time. Range: 1 … 1000. No default.
- TCP efficiency down (%): Threshold for server-to-client TCP efficiency, i.e. the minimum required ratio of total received bytes to total sent bytes. Range: 0.1 … 100 %. No default.
- TCP efficiency up (%): Threshold for client-to-server TCP efficiency, i.e. the minimum required ratio of total received bytes to total sent bytes. Range: 0.1 … 100 %. No default.

Advanced settings

- Number of streams, optional: The number of TCP sessions to use. *(Optional.)* If not specified, the number will be calculated automatically. Range: 1 … 20. No default.
- Buffer size (KiB), optional: Socket buffer size to use for both sending and receiving. *(Optional.)* If not specified, the size will be set automatically. Range: 4 … 9765 KiB. No default.
- Max send rate down (Mbit/s), optional: Server-to-client maximum TCP send rate for the test. Range: 0.1 … 10,000 Mbit/s. No default.
- Max send rate up (Mbit/s), optional: Client-to-server maximum TCP send rate for the test. Range: 0.1 … 10,000 Mbit/s. No default.

8.4.9.3 Result metrics

- **MTU (bytes):** The actual measured path MTU in the selected traffic direction(s), i.e. the maximum frame size supported by all included network equipment on the specified path.
- **Baseline RTT (ms):** Baseline round-trip time.
- **RTT under load (ms):** Round-trip time under TCP load.
- **Buffer delay (%):** Buffer delay percentage, showing how much the round-trip time increased during the application of TCP load.

- **TCP throughput (Mbit/s):** TCP throughput achieved.
- **TCP transfer time ratio:** The ratio between actual and ideal TCP transfer time, or (equivalently) the theoretically possible TCP throughput and the measured TCP throughput.
- **TCP efficiency (%):** The ratio of total received bytes to total sent bytes. If this ratio is less than 100%, it indicates that TCP retransmissions have occurred.
- **Pass/fail** outcome of test.

The picture below shows an example of results obtained from an RFC 6349 TCP throughput test.

Configuration

Name	Value
Server-to-client MTU	8950
TCP implementation	Linux TCP stack as in RFC 793, 1122, 2001 with CUBIC and SACK extensions
Number of TCP streams	1
Socket buffer size	4.0 KiB

RTT results

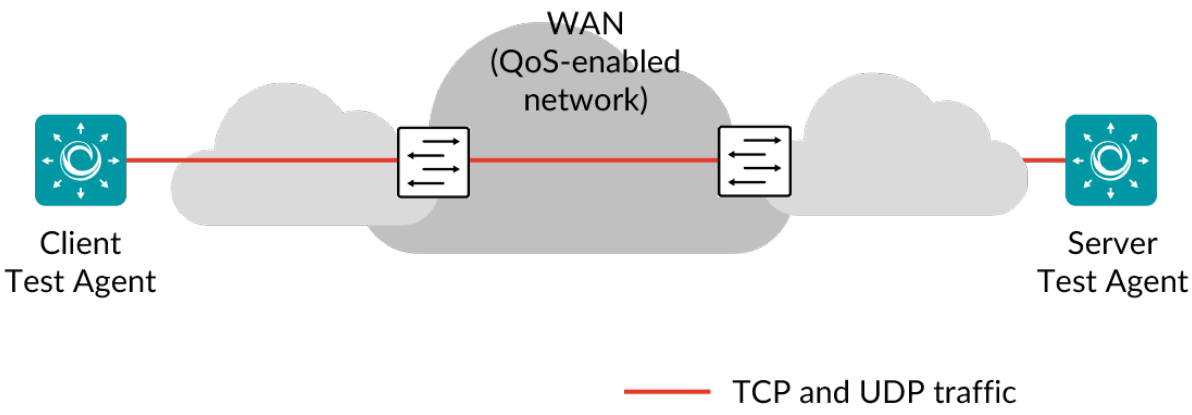
Baseline RTT (ms)	RTT under load (ms)	Buffer delay percentage
0.31	0.22	-29.35

TCP results

Direction	Throughput (Mbit/s)	TCP transfer time ratio	TCP efficiency percentage
VTA1:eth0 (IPv4) -> VTA2:eth0 (IPv4)	156.92	0.01	99.98

2019-06-11 10:13:14: Per-socket buffer limit on Test Agents: 9765.0 KiB
2019-06-11 10:13:14: Total socket buffer limit on Test Agents: 31735.2 KiB
2019-06-11 10:13:15: Getting path MTU for direction VTA1:eth0 (IPv4) -> VTA2:eth0 (IPv4).
2019-06-11 10:13:28: Got 8950 byte MTU.
2019-06-11 10:13:28: Measuring base RTT.
2019-06-11 10:13:40: Got 0.31 ms RTT.
2019-06-11 10:13:40: Calculated BDP: 0.04 KiB
2019-06-11 10:13:40: Using 0.75 KiB BDP
2019-06-11 10:13:40: Required memory on Test Agents: 0.01 MiB
2019-06-11 10:13:40: Free memory on VTA1:eth0 (IPv4): 383.25 MiB
2019-06-11 10:13:40: Free memory on VTA2:eth0 (IPv4): 385.14 MiB
2019-06-11 10:13:40: Starting streams.
2019-06-11 10:14:54: Collecting statistics.
2019-06-11 10:14:54: Passed:

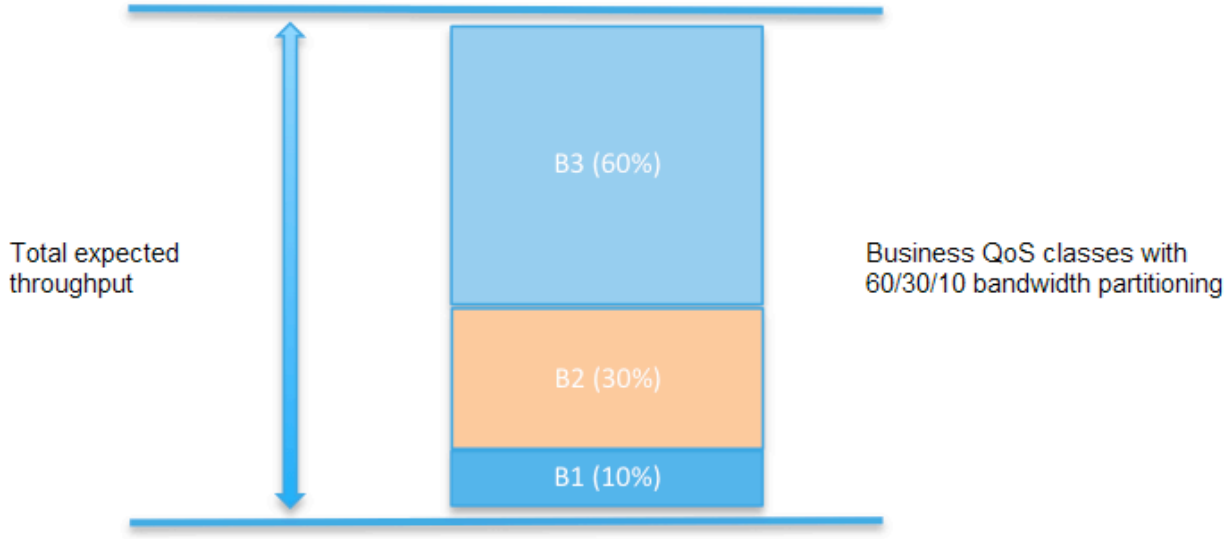
8.4.10 QoS policy profiling



This task runs TCP sessions and UDP flows between two Test Agents to verify QoS class based bandwidth shaping for up to six different QoS (quality-of-service) classes.

You can specify the total expected throughput between the Test Agents, and also its expected distribution among the QoS classes for which throughput is important. Throughput is measured as TCP bandwidth. Each measured rate is allowed to deviate from the expected value by a specified percentage. If any of these deviation thresholds is exceeded, the test fails.

The picture below shows an example with three QoS classes.



For QoS classes where delay is more important than throughput, you can measure UDP buffer delay instead of TCP bandwidth. The critical metric here is the difference (increase) in UDP buffer delay when TCP sessions in other QoS classes are running concurrently. A permissible UDP buffer delay deviation is specified for each class.

When the QoS policy profiling test starts, the UDP flows are started first and left running for 10 seconds. During this time, the average one-way delay is measured. Then the TCP sessions are also started, and one-way delay is measured once more for the UDP flows. All flows are then left running for the specified duration.

The difference between the UDP buffer delay values with and without the TCP traffic must not exceed the specified UDP delay deviation, otherwise the test fails.

The test also fails if any of UDP packet loss, (absolute) delay, or jitter goes above its respective threshold.

The DSCP and PCP (VLAN priority) value can be configured for each class.

The UDP flows are running at 50 kbit/s rate with configurable packet size.

When the TCP sessions are started, the results for the first 2–3 seconds are skipped in order to give the TCP sessions time to stabilize.

This task works only with IPv4.

8.4.10.1 Prerequisites

To run QoS policy profiling, you need to have two Test Agents installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Note that this task requires exclusive access to the Test Agents; no other tasks can be assigned to the Test Agents while it is executing.

Create a new test with a QoS policy profiling task and fill in the mandatory parameters below:

8.4.10.2 Parameters

All deviations are calculated as `(measured - expected) / expected * 100%`.

General

- Server: The Test Agent interface that will act as server.
- Client: The Test Agent interface that will act as client.
- Traffic direction: Direction of TCP/UDP traffic. One of: Upstream (from client to server) or Downstream (from server to client). Default: Downstream.
- Test duration (seconds): The duration of this test step in seconds. Min: 20 s. Max: 300 s. Default: 60 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: "Don't wait", i.e. zero wait time.

Aggregate

- Total expected TCP rate (Mbit/s): The total expected TCP payload rate for all TCP sessions combined. Range: 0.1 … 10,000,000 Mbit/s. No default.
- Max deviation from expected TCP rate (%): The allowed deviation from the expected total TCP rate, expressed as a percentage. Range: 0 … 100%. No default.

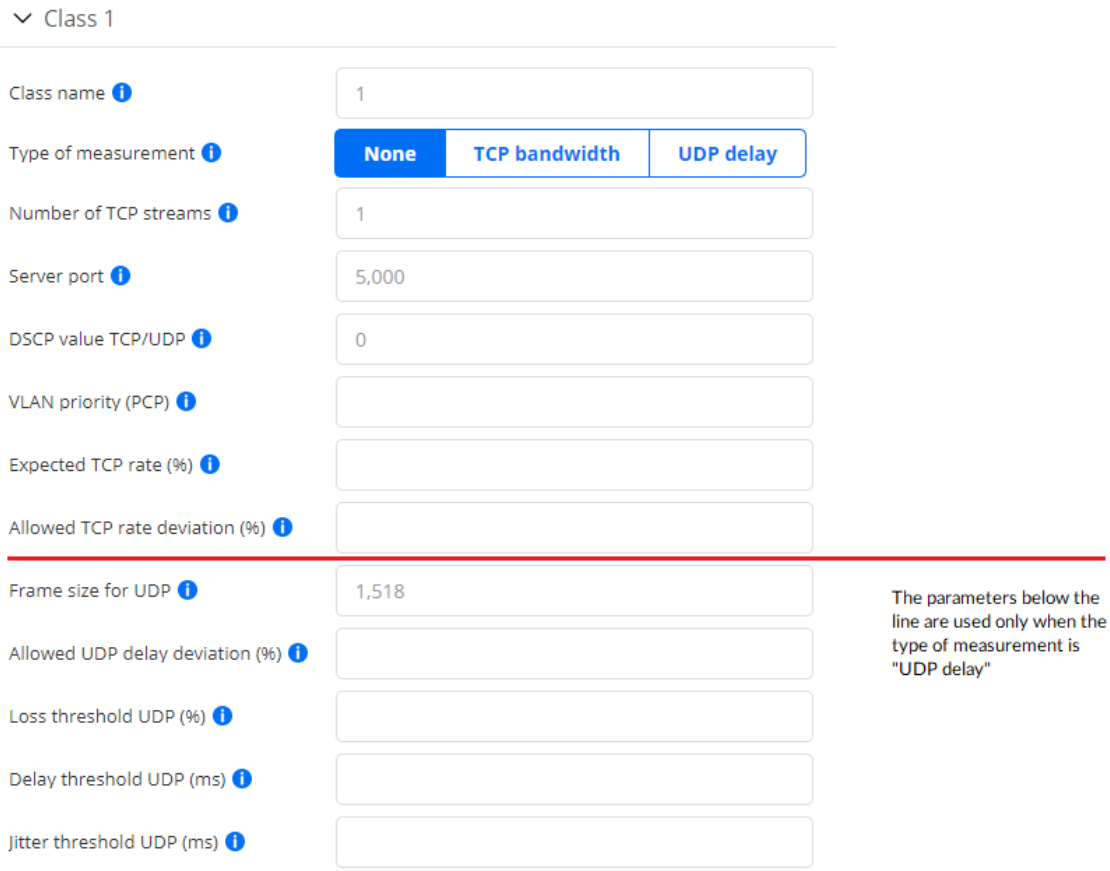
Advanced

- Pause NTP on client: Pause the time synchronization during test. This might reduce time sync problems that can occur if the test is overloading the Test Agent management interface. Value: Yes or No. Default: No.

Class 1 ... Class 6

- Class name: The name of the QoS class. Default names are simply "1" … "6".
- Type of measurement: Bandwidth (TCP) or delay (UDP). Default: None.
- Number of TCP streams: Number of TCP network flows. Range: 0 … 15. Default: 1.
- Server port: The TCP or UDP server port. Range: 1 … 65535. Default: 5000.
- DSCP value TCP/UDP: *Differentiated Services Code Point* (page 511) used for the TCP or UDP flow. Range: 0 … 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN header. Range: 0 … 7. No default.
- Expected TCP rate (%): The expected combined TCP rate in this class as a percentage of the total rate. Range: 0 … 100%. No default.
- Allowed TCP rate deviation (%): The allowed deviation from the expected TCP rate. Range: 0 … 100%. No default.
- Frame size for UDP: *Ethernet frame size* (page 512) for the UDP flow. Range: 64 … 1518 bytes. Default: 1518 bytes.

- Allowed UDP delay deviation (%): The allowed difference in UDP delay with concurrent TCP in other QoS classes as compared to the situation with no TCP. Range: 0 … 1000%. No default.
- Loss threshold UDP (%): Maximum loss allowed for the UDP flow. Range: 0 … 100%. No default.
- Delay threshold UDP (ms): Maximum (absolute) delay allowed for the UDP flow. Range: 0 … 1000 ms. No default.
- Jitter threshold UDP (ms): Maximum *jitter* (page 474) allowed for the UDP flow. Range: 0 .. 1000 ms. No default.



8.4.10.3 Result metrics – TCP bandwidth

- **Measured rate (Mbit/s):** TCP rate per class.
- **Measured rate (%):** TCP rate per class as percentage of the total rate.
- **Rate deviation (%):** The deviation from the expected TCP rate in percent.
- **Pass/fail** outcome for TCP bandwidth.

8.4.10.4 Result metrics – UDP delay

- **Loss (%):** The loss in the UDP flow before the TCP sessions are started.
- **Loss under load (%):** The loss in the UDP flow under TCP load.
- **Delay (ms):** The delay for the UDP flow before the TCP sessions are started.
- **Delay deviation (%):** The delay for the UDP flow under TCP load.
- **Jitter (ms):** The jitter for the UDP flow before the TCP sessions are started.
- **Jitter under load (ms):** The jitter for the UDP flow under TCP load.
- **Pass/fail** outcome for UDP delay.

The pictures below give some examples of results from a QoS policy profiling test.

The first screenshot shows measurements on a connection where three QoS classes share bandwidth according to a 60/30/10 partitioning.

TCP bandwidth results

Class	DSCP/PCP	Expected rate (Mbps)	Measured rate (Mbps)	Expected rate (%)	Measured rate (%)	Rate deviation (%)	Allowed rate deviation (%)	Allowed rate deviation (Mbps)	Result
B3	26	5.64	5.74	60.0	60.0	1.71	5.0	5.36 - 5.92	PASS
B2	18	2.82	2.87	30.0	30.0	1.71	5.0	2.68 - 2.96	PASS
B1	10	0.94	0.96	10.0	10.01	1.78	5.0	0.89 - 0.99	PASS
TOTAL		9.4	9.56			1.71	10.0	8.46 - 10.34	PASS

The second screenshot shows TCP bandwidth results for two QoS classes (“BC”, “BE”) and UDP delay results for a real-time QoS class (“RT”).

TCP bandwidth results

Class	DSCP/PCP	Expected rate (Mbps)	Measured rate (Mpbs)	Expected rate (%)	Measured rate (%)	Rate deviation (%)	Allowed rate deviation (%)	Allowed rate deviation (Mbps)	Result
BC	26	470.5	477.96	50.0	50.77	1.59	10.0	423.45 - 517.55	PASS
BE	10	470.5	463.44	50.0	49.23	-1.5	10.0	423.45 - 517.55	PASS
TOTAL		941.0	941.4			0.04	10.0	846.9 - 1035.1	PASS

UDP delay results

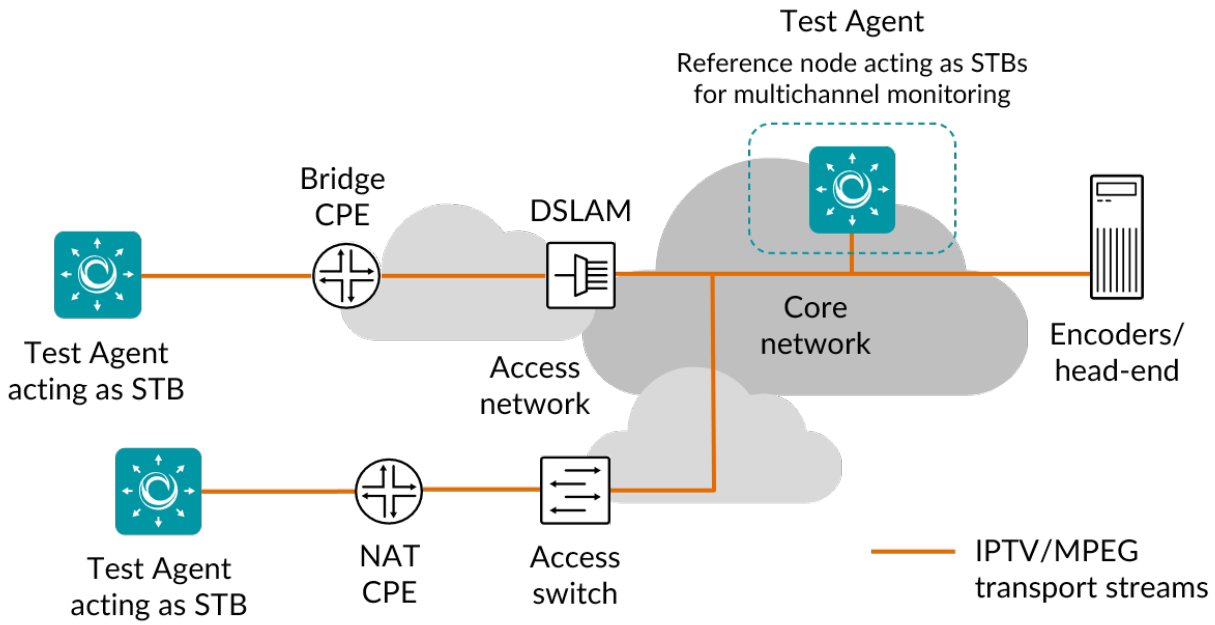
Class	DSCP	Loss (%)	Loss under load (%)	Delay (ms)	Delay under load (ms)	Delay deviation (%)	Allowed delay deviation (%)	Jitter (ms)	Jitter under load (ms)	Result
RT	46/0	0.0	0.0	0.29	22.36	7724.97	10.0	0.25	0.77	FAIL

Both measurement types have pass/fail criteria defined according to the above description.

8.5 IPTV and OTT video testing

8.5.1 Introduction to IPTV/MPEG testing

Routing Active Testing Test Agents are equipped with IPTV receivers that perform multichannel measurements at multiple locations inside your network. This makes it possible to visualize your IPTV quality from the head-end all the way to your customers. In this way, you can pinpoint where problems occur and take appropriate actions.



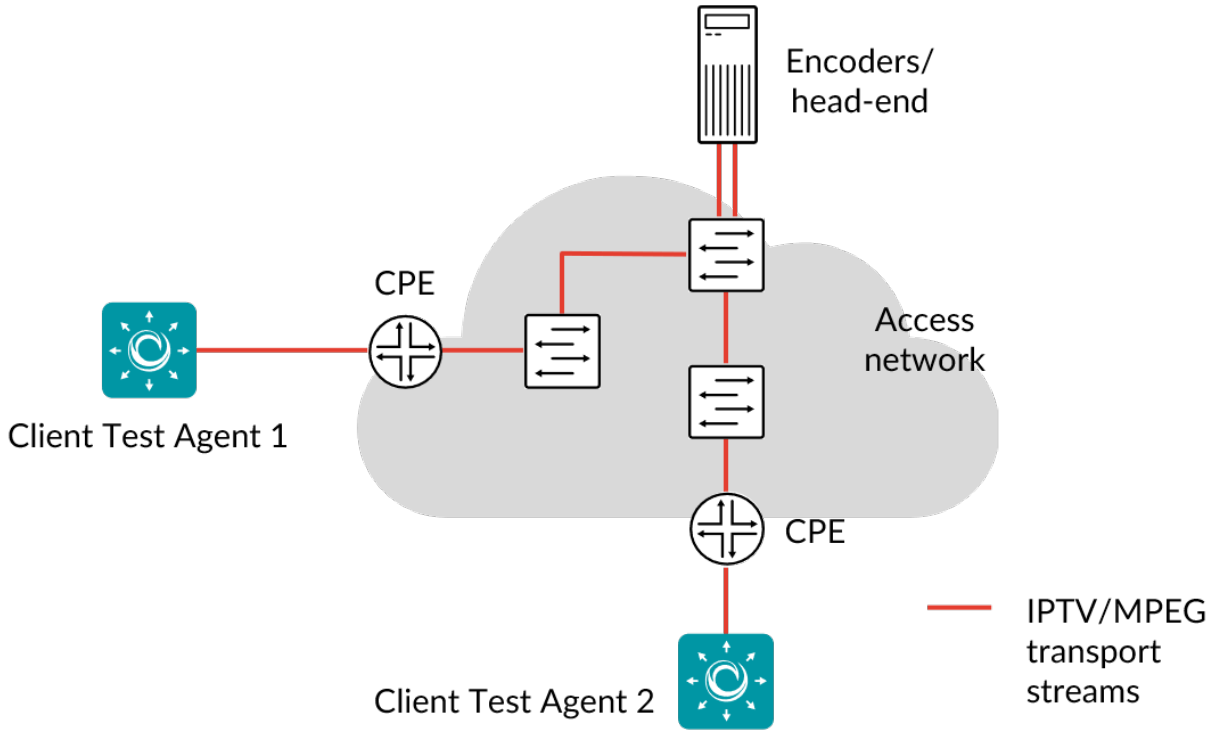
An IPTV receiver is an instance that receives IPTV multicast groups/channels (one or several). For example, if you are monitoring three channels on one interface and three channels on another interface (physical or logical), this means you are using two receivers.

Read more about the IPTV tests supported in Routing Active Testing on the following pages:

- *IPTV MPEG* (page 322)
- *IPTV MPEG inline* (page 324)

- *IPTV channel zapping time* (page 327)
- *IGMP channel join/leave* (page 332)
- *Multicast group limit* (page 334)

8.5.2 IPTV MPEG



Test Agents can receive one or several IPTV channels, measuring MPEG TS priority-1 parameters according to ► ETSI TR 101 290 (Measurement guidelines for DVB systems).

The IPTV MPEG task quickly gives you a view of IPTV channel quality at the points where you have connected the Test Agents. Using multiple Test Agents lets you monitor quality in different parts of your network. Routing Active Testing will measure and highlight MPEG loss, PCR jitter, rate, packet loss, continuity count (CC) errors, and any general problems with the MPEG stream.

When an IPTV MPEG task starts, the Test Agents will join the channels by sending IGMP join messages. Once they receive the MPEG streams, the Test Agents will continuously measure quality.

Note: Routing Active Testing does not decrypt any of the MPEG streams; its quality measurements are based on the unencrypted MPEG headers only.

It is possible to configure a threshold for the IPTV PAT/PMT receive interval, that is, define how frequently PAT and PMT information should be detected in the received MPEG stream. Note that this overrides the default frequency of two PAT/PMT packets per second, as specified in ETSI TR 101 290.

This task works with both IPv4 and IPv6.

8.5.2.1 Prerequisites

To run IPTV/MPEG measurements you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Also make sure that you have prepared Routing Active Testing with your *IPTV channel list* (page 21).

Then add an IPTV MPEG task to your test or monitor and fill in the mandatory parameters below:

8.5.2.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces on which you want to receive one or several IPTV channels.
- Channels: IPTV channels to monitor from the preset IPTV channel list. See *this page* (page 21). The maximum number of channels that can be monitored is 400.

Thresholds for errored seconds (ES)

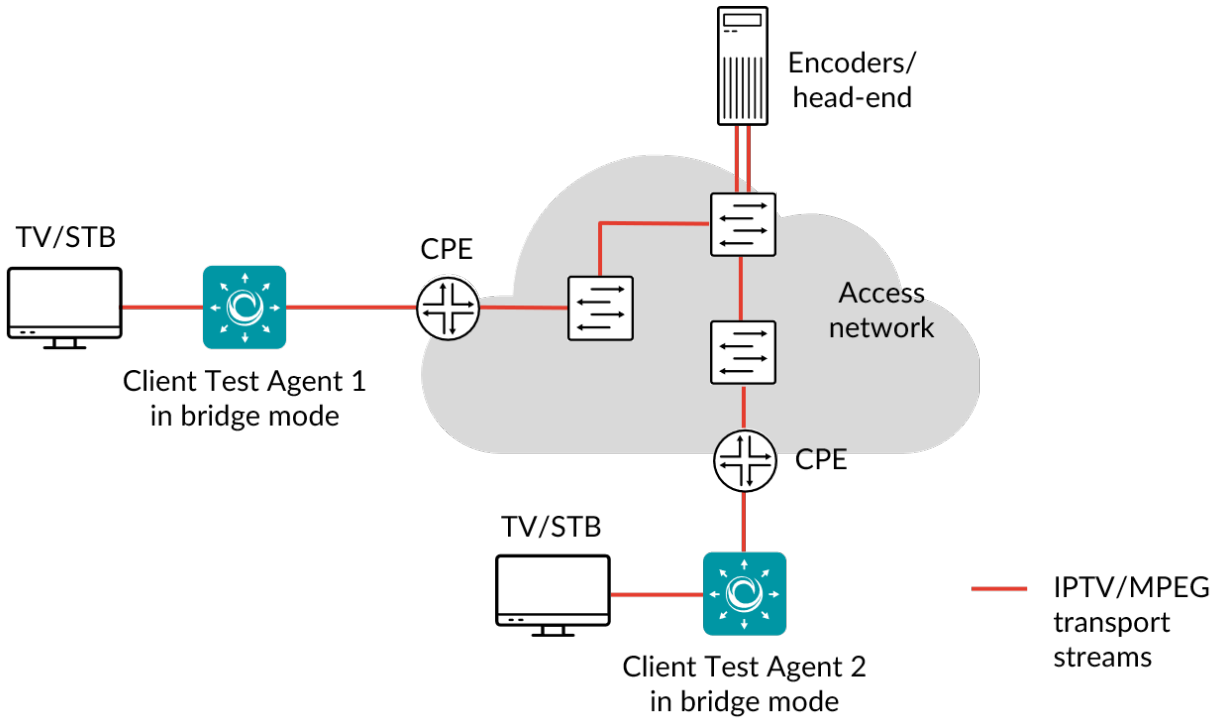
- MPEG loss (CC errors/s): Maximum tolerated MPEG packet loss (CC errors) per second. See *this page* (page 512). Min: 0 packets per second. Default: 2 packets per second.
- Jitter: Maximum tolerated PCR and RTP jitter (delay variation) in the received MPEG streams. See *this page* (page 474). Min: 0 ms. Default: 50 ms.
- PAT/PMT interval (s): Maximum tolerated interval between PAT/PMT transmissions. Min: 0.5 s. Max: 60 s. Default: 0.5 s. Note: PAT/PMT should be received every half-second on a program according to the standards.
- PID interval (s): Maximum tolerated interval between audio or video PIDs as specified by PMT. Min: 1 s. Max: 60 s. Default: 5 s. Note: On regular audio/video streams, a PID should be received every 5 seconds according to the standards.

8.5.2.3 Result metrics

- **Rate (Mbit/s):** The bit rate of the MPEG program stream.
- **Transport rate (Mbit/s):** The bit rate of the MPEG transport stream (MPEG-TS), that is, the rate of the MPEG stream including the overhead from the header of the Transport Stream packet. See *this page* (page 513) for further details.
- **MPEG loss:** MPEG packet loss, calculated from the Continuity_count_error counter in the MPEG stream.
- **PCR jitter (ms):** The jitter (delay variation) of the received MPEG stream. Calculated from the timestamps in the Program Clock Reference (PCR) field transmitted in the adaption layer of the MPEG transport stream.

- **RTP jitter, RTP loss, RTP misorders:** If the MPEG stream contains RTP headers, Routing Active Testing will calculate RTP jitter, loss, and misorderings, which are basically the same as the corresponding metrics for IP. Whether or not the MPEG stream contains RTP headers depends on the encoder at the head-end.
- **PAT errors:** A PAT error is triggered if a Program Allocation Table (PAT) is not received on a multicast group within PAT/PMT interval.
- **PMT errors:** A PMT error is triggered if a Program Map Table (PMT) is not received on a multicast group within PAT/PMT interval.
- **PID errors:** On regular audio/video streams, a frame should be received in every PID interval. If no frame is received within that interval, one PID error is generated for every second that elapses.
- **ES MPEG loss:** Number of errored seconds triggered by MPEG loss exceeding the MPEG loss threshold during one second.
- **ES jitter:** Number of errored seconds triggered by PCR jitter or RTP jitter exceeding the Jitter threshold.
- **ES invalid stream:** An aggregate of PAT, PMT, and PID errors. If any of these types of error is encountered during a second, it is marked as an "Invalid stream" errored second.
- **ES total:** Aggregated errored second percentage, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

8.5.3 IPTV MPEG inline



The IPTV MPEG inline task lets you monitor the quality of TV channels that customers are watching. When this task starts, Routing Active Testing will start listening to the IGMP signaling (IGMPv2 is supported, whereas IGMPv3 is not) and measure on the channels that the set-top box joins. Measurements include MPEG loss, PCR jitter and data rate; Routing Active Testing will also alert you about any general problems with the MPEG stream.

Note: Routing Active Testing will only measure on channels that are on its preconfigured IPTV channel list (see this page). If a set-top box joins a channel that is not present in the Routing Active Testing channel list, no measurement data will be obtained. Note also that Routing Active Testing does not decrypt any of the MPEG streams, but utilizes only the unencrypted MPEG headers for quality measurements.

It is possible to configure a threshold for the IPTV PAT/PMT receive interval, that is, define how frequently PAT and PMT information should be detected in the received MPEG stream. Note that this overrides the default frequency of two PAT/PMT packets per second, as specified in ► ETSI TR 101 290 (Measurement guidelines for DVB systems).

8.5.3.1 Prerequisites

To do IPTV inline measurements you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

To prepare for IPTV inline measurements, first create a bridge interface and connect it between the residential gateway (CPE) and the customer set-top box (STB).

Also, as noted above, make sure that you have configured Routing Active Testing with your *IPTV channel list* (page 21).

Then add an IPTV MPEG inline task to your test or monitor and fill in the mandatory parameters below:

8.5.3.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces on which to receive IPTV channels. Note: As remarked above, a prerequisite for this task is that the Test Agents have a *bridge interface* (page 176).

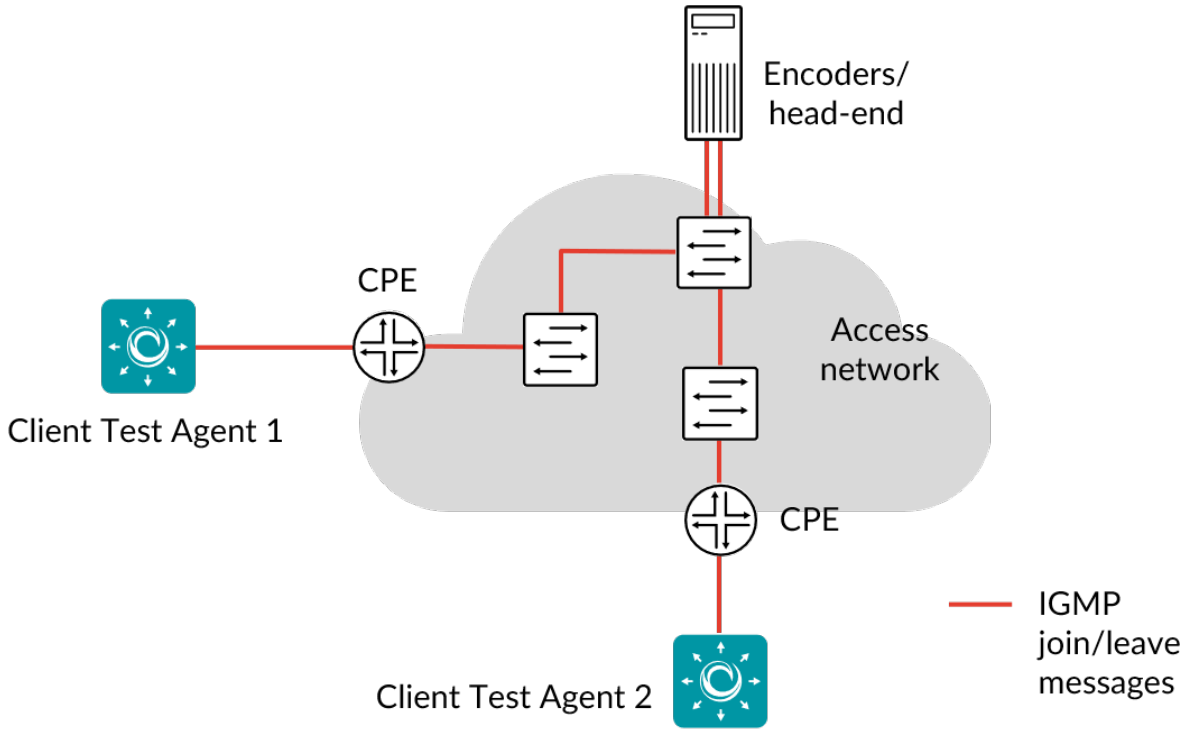
Thresholds for errored seconds (ES)

- MPEG loss (CC errors/s): Maximum tolerated MPEG packet loss (CC errors) per second. See *this page* (page 512). Min: 0 packets per second. Default: 2 packets per second.
- Jitter: Maximum tolerated PCR and RTP jitter (delay variation) in the received MPEG streams. See *this page* (page 474). Min: 0 ms. Default: 50 ms.
- PAT/PMT interval (s): Maximum tolerated interval between PAT/PMT transmissions. Min: 0.5 s. Max: 60 s. Default: 0.5 s. Note: PAT/PMT should be received every half-second on a program according to the standards.
- PID interval (s): Maximum tolerated interval between audio or video PIDs as specified by PMT. Min: 1 s. Max: 60 s. Default: 5 s. Note: On regular audio/video streams, a PID should be received every 5 seconds according to the standards.

8.5.3.3 Result metrics

- **Rate (Mbit/s):** The bit rate of the MPEG program stream.
- **Transport rate (Mbit/s):** The bit rate of the MPEG transport stream (MPEG-TS), that is, the rate of the MPEG stream including the overhead from the header of the Transport Stream packet. See *this page* (page 513) for further details.
- **MPEG loss:** MPEG packet loss, calculated from the Continuity_count_error counter in the MPEG stream. See *this page* (page 512).
- **PCR jitter (ms):** The jitter (delay variation) of the received MPEG stream. Calculated from the timestamps in the Program Clock Reference (PCR) field transmitted in the adaption layer of the MPEG transport stream.
- **RTP jitter, RTP loss, RTP misorders:** If the MPEG stream contains RTP headers, Routing Active Testing will calculate RTP jitter, loss, and misorderings, which are basically the same as the corresponding metrics for IP. Whether or not the MPEG stream contains RTP headers depends on the encoder at the head-end.
- **PAT errors:** A PAT error is triggered if a Program Allocation Table (PAT) is not received on a multicast group within PAT/PMT interval.
- **PMT errors:** A PMT error is triggered if a Program Map Table (PMT) is not received on a multicast group within PAT/PMT interval.
- **PID errors:** On regular audio/video streams, a frame should be received in every PID interval. If no frame is received within that interval, one PID error is generated for every second that elapses.
- **ES MPEG loss:** Number of errored seconds triggered by MPEG loss exceeding the MPEG loss threshold during one second.
- **ES jitter:** Number of errored seconds triggered by PCR jitter or RTP jitter exceeding the Jitter threshold.
- **ES invalid stream:** An aggregate of PAT, PMT, and PID errors. If any of these types of error is encountered during a second, it is marked as an “Invalid stream” errored second.
- **ES total:** Aggregated errored second percentage, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

8.5.4 IPTV channel zapping time



This task measures zapping times (in ms) when switching between different IPTV channels. An IPTV channel zapping consists of two IGMP messages: an IGMP join and an IGMP leave.

The IPTV channel zapping task lets you monitor and test channel change times, that is, how long it takes from the customer switching channels until the new channel is received.

When this task starts, the Test Agent starts zapping between the selected multicast channels. It waits a specified length of time for each zapping to complete (i.e. to receive traffic on the new channel). At the end of the task, the Test Agent reports the minimum, maximum, and average zapping times (the maximum will be limited to the timeout setting).

8.5.4.1 Prerequisites

To run IPTV channel zapping measurements you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Also make sure that you have prepared Routing Active Testing with your *IPTV channel list* (page 21).

Note: Please note that it does not make sense to run an IPTV channel zapping task on a channel where the same Test Agent is already running an *IPTV* (page 322) monitoring session on the same interface. (If this is the case, the Test Agent will not leave the channel on IGMP leave, since the IPTV monitor stipulates that the channel should be received continuously. In other words, because of the way multicast works, the IPTV monitor will interfere with the IPTV channel zapping.)

In your test or monitor, add an IPTV channel zapping time task and fill in the mandatory parameters below:

This task works with both IPv4 and IPv6.

8.5.4.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces to use as clients.
- Channels: IPTV channels to monitor from the preset IPTV channel list. See *this page* (page 21).
- Min wait time between zapping: The minimum time a client will wait between consecutive zappings. Each zapping is constituted by an IGMP join and an IGMP leave message. Min: 0 ms. Default: 2000 ms.
- Max wait time between zapping: The maximum time a client will wait between consecutive zappings. Min: 0 ms. Default: 2000 ms.

Note: If you set Min wait time… and Max wait time… differently, the wait time between zappings will be randomized within the specified interval.

Thresholds for errored seconds (ES)

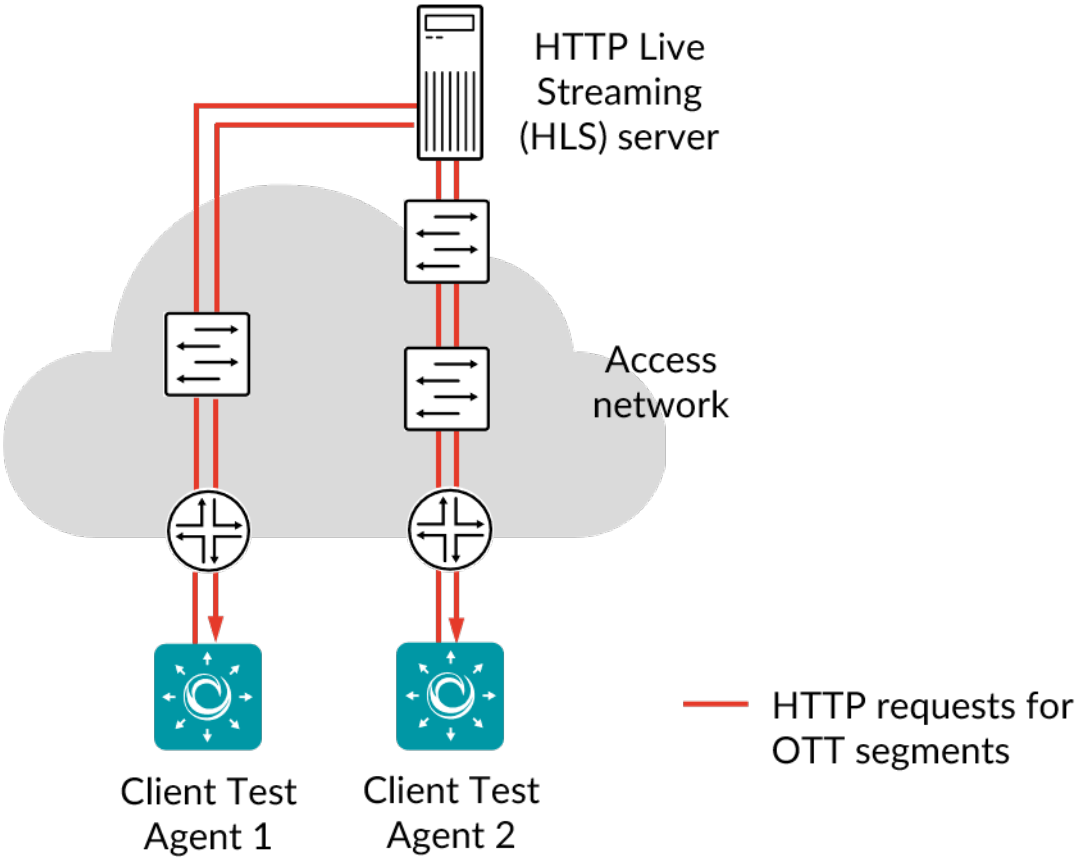
- Threshold for join delay: The join delay is the time from when the client issues an IGMP join message for a multicast group until the first packet is received for that multicast group. An errored second is triggered if this threshold is exceeded. The default value is set to 500 ms in accordance with ► ETSI TS 102 034 (2009-08). Min: 0 ms. Default: 500 ms.
- Threshold for leave delay: The leave delay is the time from when the client issues an IGMP leave message for a multicast group until the last packet is received for that multicast group. An errored second is triggered if this threshold is exceeded. The default value is set to 500 ms in accordance with ETSI TS 102 034 (2009-08). Min: 0 ms. Default: 500 ms.

8.5.4.3 Result metrics

- **Average join delay (ms):** Average delay from sending an IGMP join until the flow arrived.
- **Minimum join delay (ms):** Minimum delay from sending an IGMP join until the flow arrived.
- **Maximum join delay (ms):** Maximum delay from sending an IGMP join until the flow arrived.
- **Failed joins:** Number of failed IGMP joins. A join fails if the flow does not arrive before an IGMP leave is sent.
- **Average leave delay (ms):** Average delay from sending an IGMP leave until the flow was stopped.
- **Minimum leave delay (ms):** Minimum delay from sending an IGMP leave until the flow was stopped.
- **Maximum leave delay (ms):** Maximum delay from sending an IGMP leave until the flow was stopped.
- **Failed leaves:** Number of failed IGMP leaves. A leave fails if the flow is not stopped before a new IGMP join is sent.

- **ES join:** Number of errored seconds triggered by a failed IGMP join occurring during the second.
- **ES leave:** Number of errored seconds triggered by a failed IGMP leave occurring during the second.
- **ES total:** Aggregated errored second percentage, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

8.5.5 OTT testing: HTTP Live Streaming (HLS)



A Test Agent can measure user experience for adaptive video streaming using HTTP Live Streaming (HLS). On being told the URL of the video, the Test Agent will parse the manifest file and start downloading the video segments. The algorithm will adapt to current network conditions and select the highest possible quality (bit rate), while avoiding buffering.

OTT is an abbreviation for "Over The Top" and refers to the content being delivered on top of an ordinary Internet service, unlike IPTV, which runs as a separate service. HLS is one common OTT protocol, specified by Apple Inc. and supported by all Apple mobile devices as well as by the Safari browser.

The HLS protocol supports multiple qualities with different bit rates, called *variants*, for the same video, and the client can choose to download segments from the variant that best matches network and client performance.

This task works only with IPv4.

8.5.5.1 Prerequisites

To run HLS measurements you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

In you test or monitor, add an OTT - HLS task and fill in the mandatory parameters below:

8.5.5.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces to use as clients.
- URL: URL of the video stream. This can be either a playlist linking to other playlists for the different variants, or a playlist containing the segments. The file extension is `.m3u` or `.m3u8`.

Thresholds for errored seconds (ES)

- Playback rate (Mbit/s): An errored second is triggered if the playback rate drops below this threshold. Min: 0 Mbit/s. No default.
- Download rate (Mbit/s): An errored second is triggered if the download rate drops below this threshold. Min: 0 Mbit/s. No default.
- Selected rate (Mbit/s): An errored second is triggered if the rate of the variant selected by the Test Agent is below this threshold. Min: 0 Mbit/s. No default.
- Buffer size (seconds): An errored second is triggered if the duration of the data segment buffered in the Test Agent drops below this threshold. Min: 0 s. No default.

Advanced

- Buffer size (seconds): Target duration of buffered data. When the duration of the buffered data falls below this value, new segments will be downloaded. Min: 0 s. Default: 60 s.
- Initial buffering (seconds): The duration of the initial buffered data required before the playback starts. Min: 0 s. Default: 10 s.
- Loop: If set to Yes, the video stream will loop when the end of the playlist is reached. If set to No, the playback will stop, and errored seconds will be triggered. The "No" option is relevant mainly for live video streams which you expect never to end, and for which you want to trigger an alarm if that happens. Default: Yes.

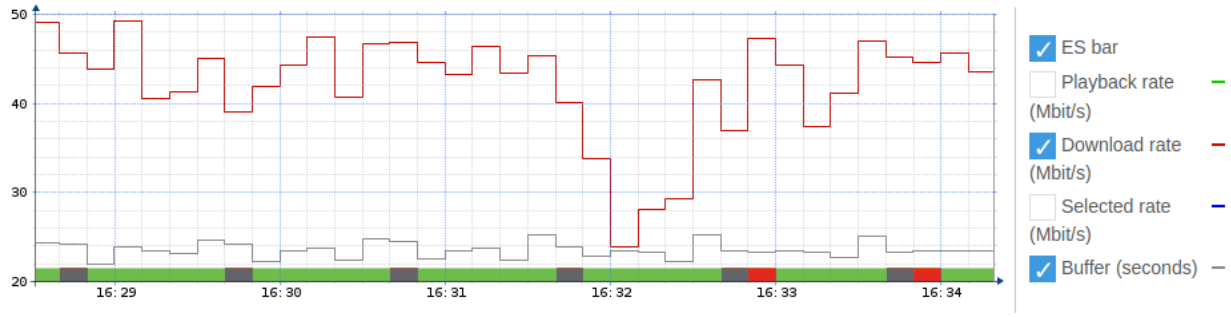
8.5.5.3 Result metrics

- **Playback rate (Mbit/s):** Actual data rate of the video stream.
- **Download rate (Mbit/s):** Download rate for the segments. The download is started when the amount of buffered data falls below the configured buffer size.
- **Selected rate (Mbit/s):** Data rate of the selected variant in the manifest file.
- **(Min) Buffer (s):** Actual buffer length in seconds.
- **ES playback rate (%):** Percentage of seconds during which the playback rate dropped below the Playback rate threshold.
- **ES download rate (%):** Percentage of seconds during which the download rate dropped below the Download rate threshold.
- **ES selected rate (%):** Percentage of seconds during which the selected rate dropped below the Selected rate threshold.
- **ES buffer underrun (%):** Percentage of seconds during which the buffered data dropped below the Buffer size threshold.
- **ES buffering (%):** Percentage of seconds during which the buffer became empty, so that rebuffering was needed.
- **ES total:** Aggregated errored second percentage, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

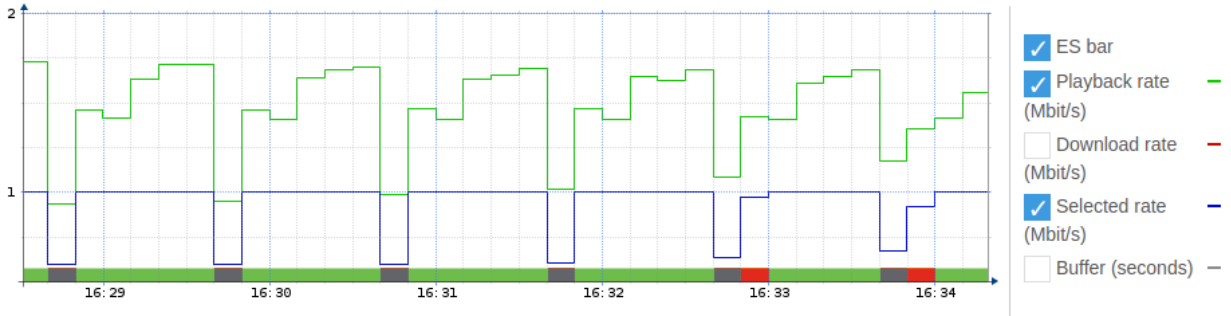
8.5.5.4 Example

Below is an example of OTT testing.

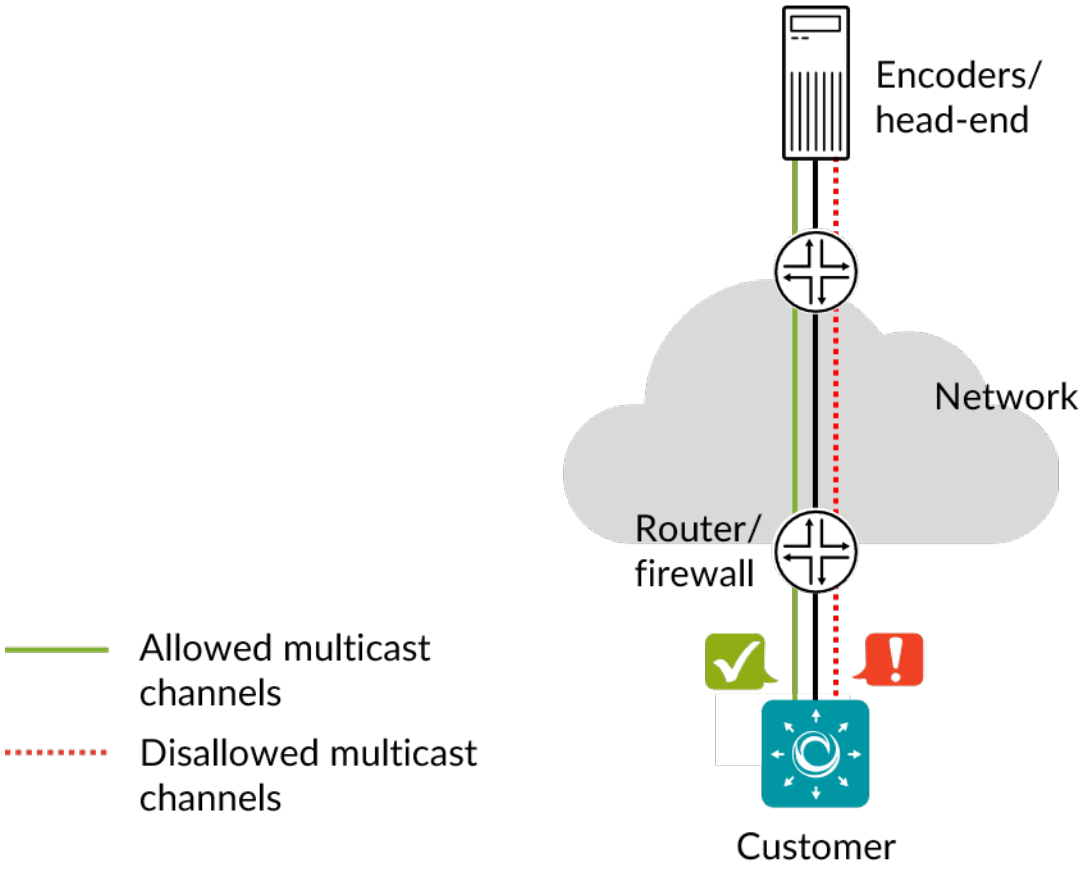
The first graph shows the downloading of OTT data (red) and the amount of data stored in the buffer (gray).



The second graph shows the actual playback rate (green) compared to the playback rate currently selected by the client (blue).



8.5.6 IGMP join/leave



This task checks if customers can join the allowed multicast channels and receive data on these channels. The task also verifies that customers do not receive multicast channels other than those specified.

This task works only with IPv4.

8.5.6.1 Impact

DoS

8.5.6.2 Test procedure

Customer joins the multicast channels, listens for packets for 10 seconds, then leaves.

8.5.6.3 Fail criteria

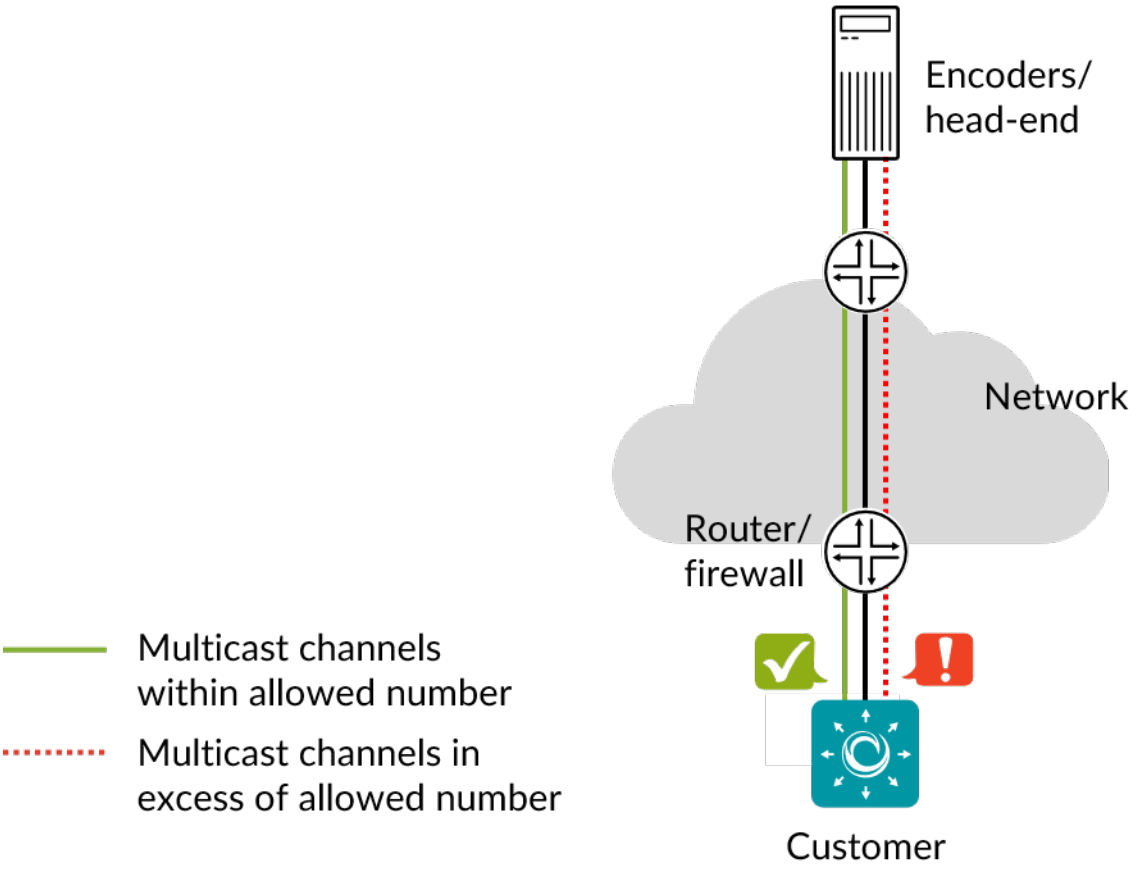
- No data is received on one of the allowed multicast channels.
- Data is received on one of the disallowed multicast channels.

8.5.6.4 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- Available groups: Multicast channels that Customer should be able to receive. Separated by commas. Default: 239.0.0.1, 238.0.0.2
- Unavailable groups: Multicast channels that Customer should not be able to receive. Separated by commas. Default: 238.0.5.1

8.5.7 Multicast group limit



This task is a security test that checks that a customer can only join a specified maximum number of multicast channels. If the customer attempts to join one more channel, then either the join should be ignored, or some channel previously joined should be disabled.

The task also checks that within the above limitation, the customer can receive the desired multicast channels without problems.

This task requires an external multicast source and works only with IPv4.

8.5.7.1 Impact

DoS

8.5.7.2 Test procedure

1. Customer joins the allowed number of multicast channels.
2. Customer tries to join one additional channel.

8.5.7.3 Fail criteria

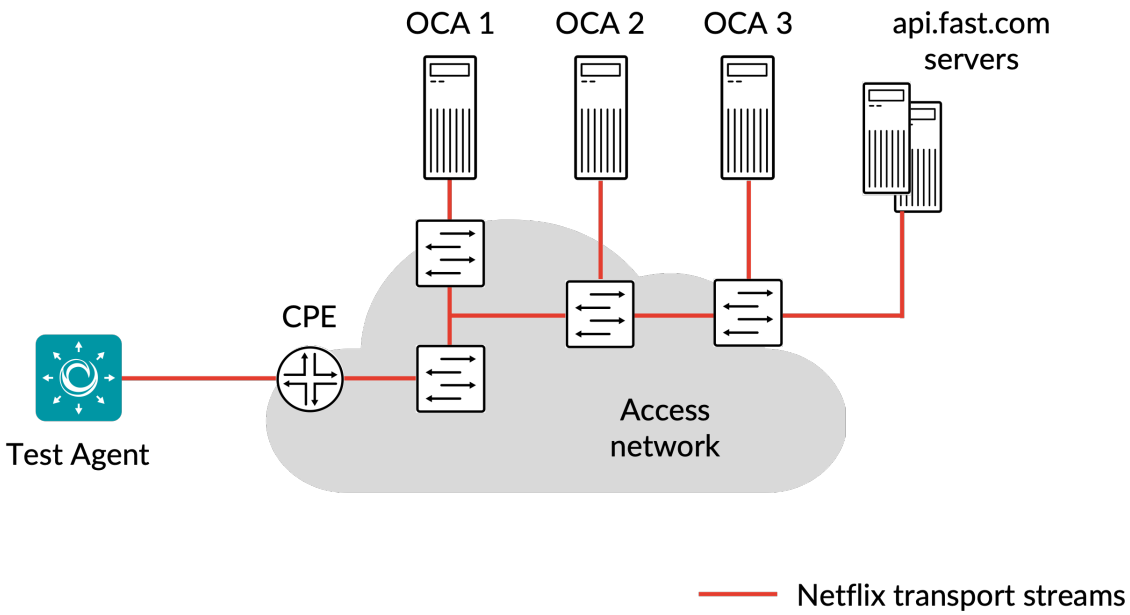
- In step 1, some channel is not received.
- In step 2, the total number of channels received exceeds the maximum number.

8.5.7.4 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- Available groups: Multicast channels to be joined during the test. Separated by commas. Default: 239.0.0.1, 238.0.0.2, 237.0.0.3, 236.0.0.4
- Maximum groups: The maximum number of multicast channels that a customer is allowed to receive concurrently. Default: 3.

8.5.8 Netflix Speedtest



This task instructs the Test Agent to download Netflix test segments over HTTPS from one or several OCAs (Open Connect Appliances) operated by Netflix. The bandwidth is measured via download or upload to OCA URLs provided by api.fast.com servers, and the bandwidth obtained is evaluated against errored second thresholds.

Each individual measurement for a retrieved set of OCA URLs is termed a *measurement cycle*.

The criterion for considering a measurement *stable* is as follows: All samples taken within a one-second sliding window must differ by no more than 10% from the average bandwidth measured within that sliding window. The stability check begins to operate only after Minimum duration, then continues to be applied (every 100 ms) until the stability criterion is met or the Maximum duration of the measurement expires.

The common test parameter *Delayed start* (page 287) is not available for Netflix.

8.5.8.1 Parameters

General

- Clients: Test Agents interfaces that will act as Netflix clients.
- Measurement cycle period: Time between two Netflix measurement cycles in minutes. Regarding the term “measurement cycle”, see the introduction above. Min: 1 min. Max: 60 min. Default: 15 min.
- Download or Upload: Bandwidth to report: upload or download bandwidth. Default: Download.

Advanced

- Minimum duration: Minimum duration for which the measurement should run before checking for throughput stabilization. Min: 1 s. Max: 3600 s. Default: 2 s.
- Maximum duration: Maximum duration for which the measurement should run and check for throughput stabilization before it expires. Min: 5 s. Max: 3600 s. Default: 30 s.
- Minimum concurrent OCAs: The minimum number of OCAs which will be used to make measurements concurrently. Min: 1. Max: 5. Default: 3.
- Maximum concurrent OCAs: The maximum number of OCAs which will be used to make measurements concurrently. Min: 1. Max: 5. Default: 5.

Thresholds for errored seconds (ES)

- Minimum bandwidth (Mbit/s): An errored second is triggered if the bandwidth drops below this threshold.
- Maximum latency (ms): An errored second is triggered if the latency goes above this threshold.

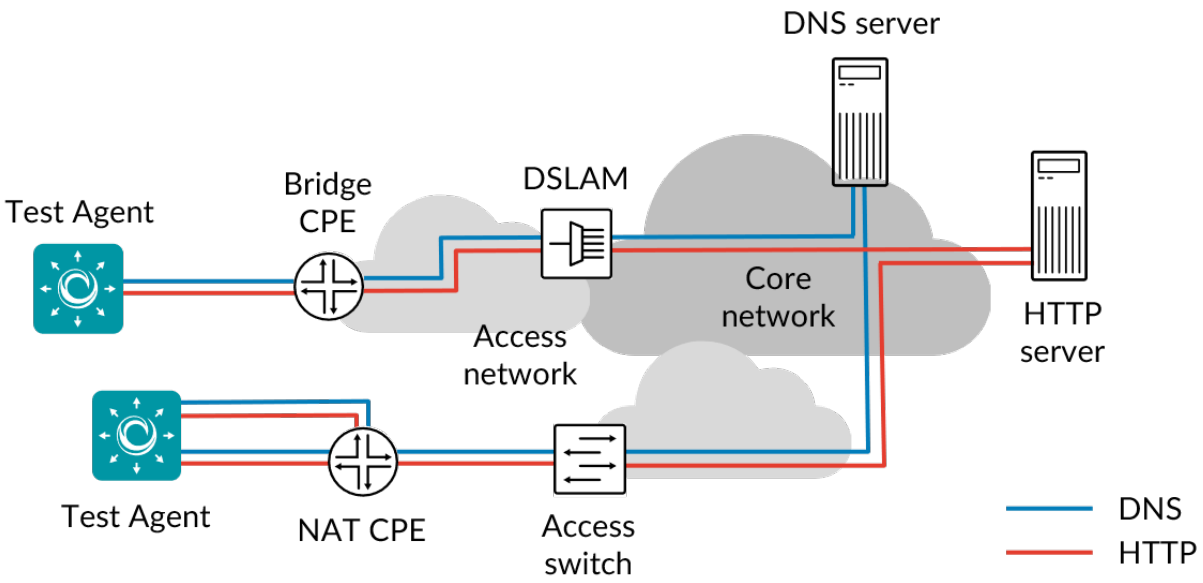
8.5.8.2 Result metrics

- **Bandwidth speed (Mbit/s):** Speed of data transmission to or from the Netflix servers.
- **ES bandwidth:** Number of errored seconds due to low bandwidth.
- **ES latency:** Number of errored seconds due to high latency.

8.6 HTTP and DNS testing

8.6.1 Introduction to HTTP and DNS testing

Test Agents have the capability to issue HTTP and DNS requests. These functions enable you to request web pages, verify response codes, and test DNS response times from distributed locations in your network. This in turn helps you to quickly locate the sources of possible problems with your network, web applications or servers, whether they are in the IPv4 or the IPv6 addressing space.



Generation of HTTP and DNS requests can be done in both point-to-point and hub-and-spoke setups.

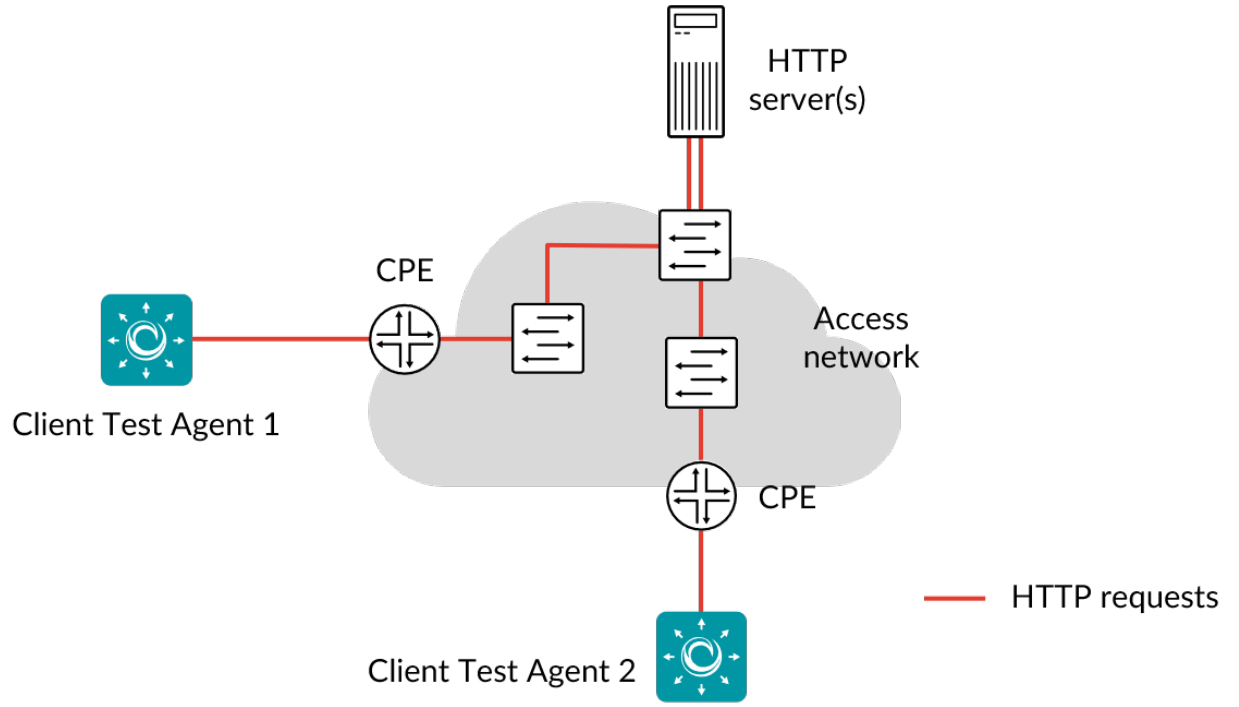
Read more about Routing Active Testing HTTP and DNS testing on the following pages:

- *HTTP* (page 339)
- *DNS* (page 345)

Test Agents can also collect data from HTTP sessions conducted by Junos devices. This is covered here:

- *Junos HTTP* (page 341)

8.6.2 HTTP



The HTTP task is used to test or monitor HTTP servers.

Running an HTTP task gives you a good overview of the performance of a website or web application, of the web server, and of the network between the web server and the Test Agent. You can request web pages and verify response codes from distributed locations inside or outside of your network.

When an HTTP task starts, the Test Agents will make an HTTP Get request towards the specified URL and fetch the response. No rendering is done of HTML pages, so no additional requests are made for linked resources, (images, CSS files, and so forth). Measured parameters include TCP connect time, time until first byte received, time until last byte received, and download speed.

Traffic is initiated by the Test Agents, and the HTTP server reciprocates by sending traffic to the Test Agents using the same ports. This setup makes it possible to run tests also when the Test Agents are located behind NAT.

HTTPS is supported, but no verification is done of the SSL certificate.

This task works with both IPv4 and IPv6.

8.6.2.1 Prerequisites

To run HTTP measurements you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

In your test or monitor, add an HTTP task and fill in the mandatory parameters below:

8.6.2.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces to use as clients.
- URL: URL that the Test Agents will request. Note: IPv6 addresses must be enclosed in brackets as per ► IETF RFC 2732. Example: `http://[2001:470::9c66]`.
- Time between requests (s): Time to wait between sending consecutive HTTP GET requests. Min: 0.01 s. Max: 3600 s. Default: 10 s.

Note: Take care not to set this parameter too low. If HTTP requests are sent at a higher rate than the web server can process them, requests will accumulate at the server and may overload it.

Thresholds for errored seconds (ES)

- HTTP response code: HTTP status code that must be matched in the HTTP response. If the HTTP response code does not match this one, an errored second is triggered. Default: "200 OK", the status code for a successful HTTP request.
- Timeout (ms): If no response to the HTTP request is obtained within this time, an errored second is triggered. Min: 1 ms. Max: 30,000 ms. Default: 3000 ms.
- Response content: The (case-insensitive) regular expression against which the HTTP response content will be validated. If the response content is larger than 100 KB, only the first 100 KB will be used in the match. The response content is decoded using the character set specified in the Content-Type HTTP header, or ISO 8859-1 if no encoding is specified. The Content-Type MIME must be "text/*", otherwise the matching will fail.

Advanced

- Request lifetime (ms): Maximum time the Test Agent will wait for an HTTP response before canceling the HTTP request. Min: 1 ms. Max: 30,000 ms. Default: 4000 ms.
- Proxy server: If set, the specified IP address will be used as HTTP proxy server.
- Proxy server port: Port to use as HTTP proxy port. Range: 1 … 65535. Default: 8080.
- Proxy authentication: Authentication type used by proxy server. If set, Proxy username and Proxy password should also be entered. Note: Currently there is an issue with NTLM authentication that causes Routing Active Testing to measure too low connect time.
- Proxy username: User name used for HTTP proxy authentication.
- Proxy password: Password used for HTTP proxy authentication.

8.6.2.3 Result metrics

- **Connect time (ms):** Time taken to set up a TCP connection to the web server (time from sending TCP SYN until receiving TCP ACK).
- **First byte received (ms):** Time from sending the HTTP Get request until the first byte of the response is received. For a dynamic website, the server side may take a while to generate the response.
- **Response time average (ms):** Average response time for the selected time period, that is, the average time taken to download the content from the URL.
- **Response time min (ms):** Minimum response time during the selected time period.
- **Response time max (ms):** Maximum response time during the selected time period.
- **Size (KiB):** Length of the HTTP response, including HTTP headers.
- **Rate (Mbit/s):** Download rate of the response. Calculated as the size of the response divided by the total response time.
- **ES timeout:** Number of errored seconds triggered because no HTTP response was obtained before the Timeout period expired.
- **ES response:** Number of errored seconds triggered by an invalid HTTP response or by the HTTP response code or content not matching the specified ones.
- **ES total:** Aggregated errored seconds, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

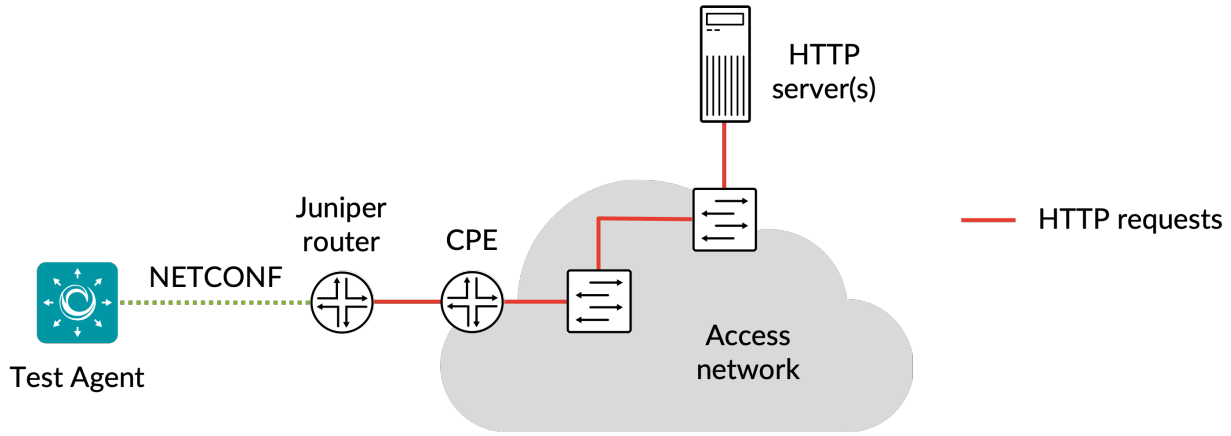
8.6.3 Junos HTTP

In this task, a Test Agent Application connects to one or several Junos devices (defined as *network devices* (page 35) in Routing Active Testing) via the NETCONF protocol and accesses HTTP sessions running on these devices (these sessions have not been configured in Routing Active Testing). The Test Agent collects measurement results from the HTTP sessions, evaluates errored second thresholds, and reports all results back to Control Center.

Each HTTP session running on a Junos device is identified as a "test" belonging to an "owner" (specified as `services rpm probe <owner> test <test>` in the Junos CLI). The test and owner are shown along with the results in Control Center so that you can correlate them with the HTTP sessions on the Junos device.

Note: A Test Agent Application is required for this task; the Test Agent Appliance does not support it. The Junos HTTP task is also different from *regular HTTP* (page 339) in that the Test Agent does not itself conduct the measurements.

IPv6 is supported in the communication between the Test Agent and the Junos device.



8.6.3.1 Prerequisites

To perform Junos HTTP measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

As regards each targeted Junos device, the following holds:

- The functionality has been verified to work on the following Juniper device models, but it might also work on other devices:
  - vMX
  - MX204
- The functionality has been verified for Junos versions 18.3–20.2. Junos Evolved is not supported.
- There must be network connectivity from the Test Agent to the device (default TCP port: 830).
- You must have a user account on the device to be able to log in to it and retrieve measurement data. How to create a user account is described *here* (page 343).
- The Junos device must be configured as a *network device* (page 35) in the Routing Active Testing inventory.
- The probe of the Junos device must be configured with the target address:

```
set services rpm probe <owner> test <test> target <address_type> <address>
```

  Example:

```
set services rpm probe owner1 test t1 target url http://192.168.0.1:3000
```

- The probe of the Junos device must also be configured with the correct probe type:

```
set services rpm probe <owner> test <test> probe-type <http-get/http-metadata-get>
```

- The Junos device must be running HTTP measurements when you execute the Junos HTTP task. The relevant Junos documentation is found here.

Once you have finished the above preparations, you can add a Junos HTTP task to your test or monitor and fill in the mandatory parameters as shown below.

8.6.3.2 Junos device configuration

This section is about suitable configuration of the Junos device. This needs to be done directly on the device and cannot be performed in the Routing Active Testing task. For details, please refer to Junos device documentation.

Creating a user account on a Junos device

The user account should have limited permissions and can be created as follows:

```
configure
set system login user <username> class read-only
set system login user <username> authentication plain-text-password
New password: <password>
commit
```

Configuring HTTP history size

The configuration parameter `services rpm probe <owner> test <test> history-size` in the Junos device specifies how many historical probe results are stored for each owner and test. (Regarding these concepts, see the *introductory section* (page 341) above.)

The Test Agent will fetch new probe results from this result set every 5 seconds. To have some margin for communication delays between the Test Agent and the Junos device, we recommend that you configure `history-size` to correspond to at least 1 minute. For example, with `probe-interval` set to 10 seconds, we recommend a `history-size` of at least 6.

8.6.3.3 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Client: Test Agent interface that will connect to the Junos device.
- Network devices: Junos devices that are running HTTP tasks and will be accessed by the Test Agent.
- Filter based on owner: Only collect results from the specified owner as configured on the Junos device. If you leave this blank, results will be collected from all owners.
- Filter test session: Only collect results from the specified test as configured on the Junos device. If you leave this blank, results will be collected from all tests.

Thresholds for errored seconds (ES)

- RTT threshold (ms): Round-trip time threshold for triggering an errored second. If the round-trip time exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Jitter threshold (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.

Note: Loss will always trigger an errored second.

Advanced

- Collection interval (s): The interval at which results are collected from the Junos devices. Min: 5s Max: 300s Default: 5s.
- Session timeout (s): The time after which idle sessions will be removed.

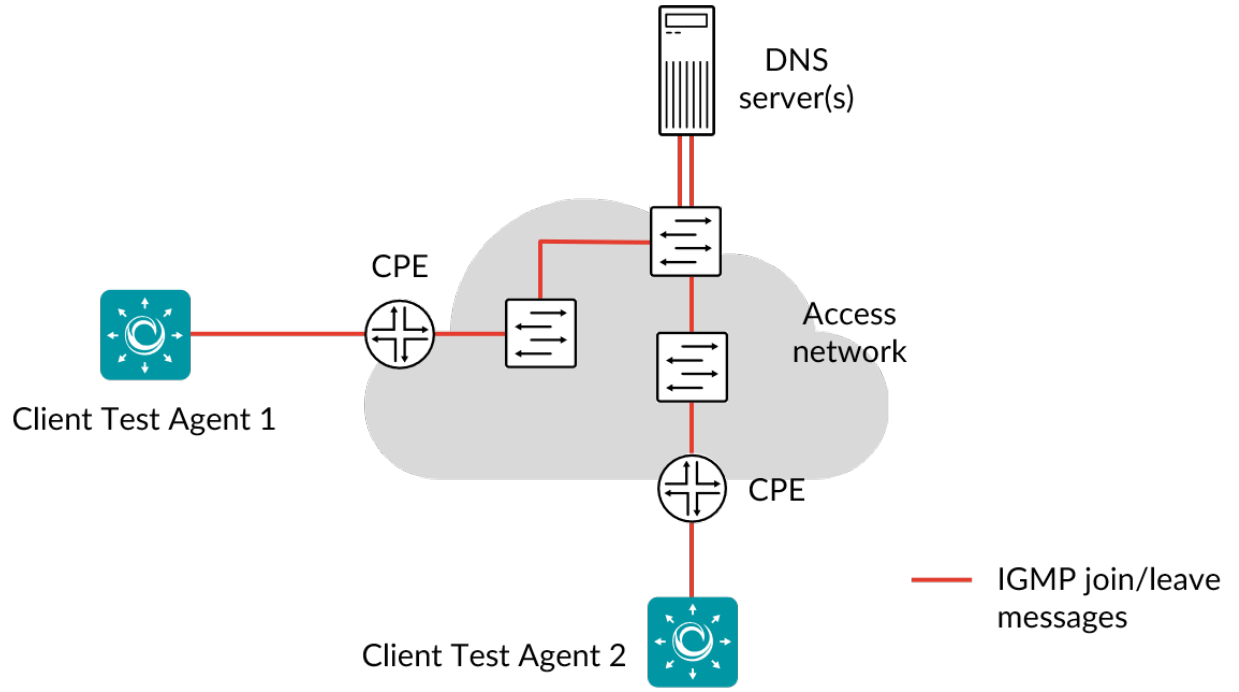
8.6.3.4 Result metrics

- **RTT (ms):** Delay between the transmission of a probe and the arrival of its response.
- **Round-trip jitter (ms):** Difference between the current round-trip time and the previous measurement.
- **Round-trip interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in IETF RFC 1889, calculated for the full round-trip.

Note: Jitter and interarrival jitter are calculated differently from what is called "delay variation" in other tasks.

- **Loss (%):** Round-trip packet loss in percent.
- **ES (s):** Aggregated number of errored seconds (ES), taking into account all types of error.
- **ES loss (s):** Number of errored seconds caused by loss.
- **ES RTT (s):** Number of errored seconds caused by round-trip time.
- **ES jitter (s):** Number of errored seconds for caused by jitter.

8.6.4 DNS



This task enables distributed testing and monitoring of your DNS servers.

Running a DNS task provides information about the response times of your DNS servers from different locations. High DNS response times translate into high response times for all services that use DNS to resolve IP addresses, such as web surfing.

When a DNS task starts, the Test Agents will send a request to resolve a lookup address, and collect statistics on response times.

DNS primarily uses User Datagram Protocol (UDP) on port number 53 to serve requests. DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server.

This task works with both IPv4 and IPv6.

8.6.4.1 Prerequisites

To run DNS measurements you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Traffic will be initiated by the Test Agents, and the DNS server will respond using the same ports. This setup makes it possible to run tests also when the Test Agents are located behind NAT.

In your test or monitor, add a DNS task and fill in the mandatory parameters below:

8.6.4.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces to use as clients.
- DNS server: DNS server to query and test. Leave this empty to use the interface default, which is usually the DNS you have been provided via DHCP.
- Lookup name: Domain name to look up, e.g. “example.com”. Lookups of this domain name will recur periodically.
- DNS record type: Type of DNS record to look for. The record types supported by Routing Active Testing are as follows:
  - A (IPv4 address; default)
  - AAAA (IPv6 address)
  - CNAME (canonical name)
  - MX (email)
  - PTR (pointer)
  - NS (name server)
  - SOA (start of authority)
  - TXT (text)
- Time between requests (s): Time to wait between consecutive DNS requests. Min: 0.01 s. Max: 3600 s. Default: 10 s.

Thresholds for errored seconds (ES)

- Timeout (ms): If no response to the DNS request is obtained within this time, an errored second for timeout will be indicated (unless the request lifetime expires; see below). Min: 1 ms. Max: 30,000 ms. Default: 50 ms.

Advanced

- Request lifetime (ms): Maximum time to wait for a response before the DNS request is canceled. If this time expires without a response, an errored second for lifetime expiry will be indicated. Min: 1 ms. Max: 30,000 ms. Default: 200 ms.
- Response code: Here you can specify an expected response code from the DNS server. If the actual response code does not match this one, a “Response” errored second is triggered. Possible response codes are: NOERROR, REFUSED, NXDOMAIN, SERVFAIL, and NOTAUTH. Default: NOERROR.
- Expected response: Here you can specify an expected response from the DNS server. If the actual response does not match this, a “Response” errored second is triggered. If the response consists of multiple answers, one of them must match.

- Recursive requests: Set the Recursion Desired flag in DNS requests. Default: Enabled.

8.6.4.3 Result metrics

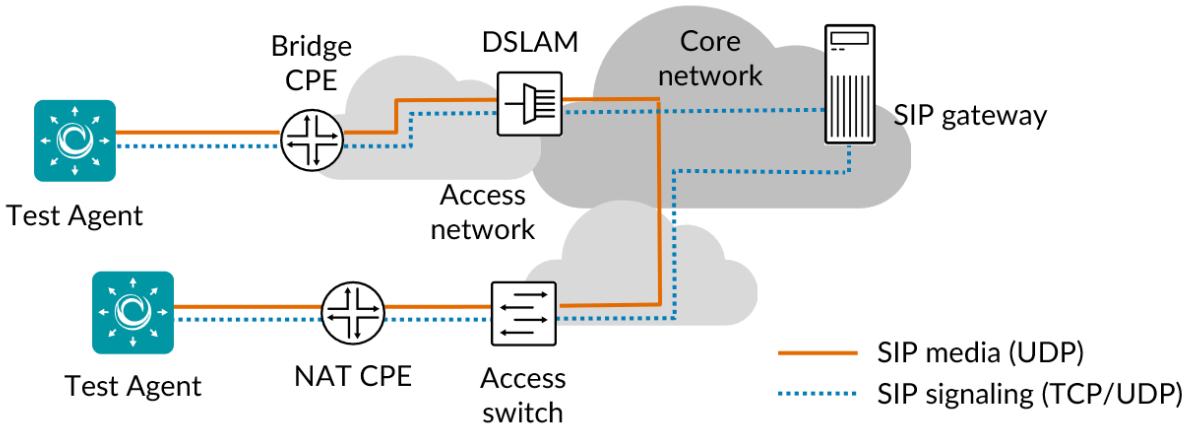
- **Response time average (ms):** Average response time during the selected time period, that is, the average time taken to receive an answer from the DNS server.
- **Response time min (ms):** Minimum DNS response time.
- **Response time max (ms):** Maximum DNS response time.
- **ES timeout:** Number of errored seconds triggered because no DNS response was obtained before the Timeout period expired (although a response did arrive within Request lifetime, if this has been set).
- **ES lifetime:** Number of errored seconds triggered because no DNS response was obtained before the Request lifetime period expired.
- **ES response:** Number of errored seconds triggered because the DNS response code differed from Response code or the response differed from Expected response.
- **ES total:** Aggregated errored seconds, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

8.7 SIP testing

8.7.1 Introduction to SIP testing

Test Agents in Routing Active Testing are equipped with SIP clients that are capable of setting up VoIP sessions between each other.

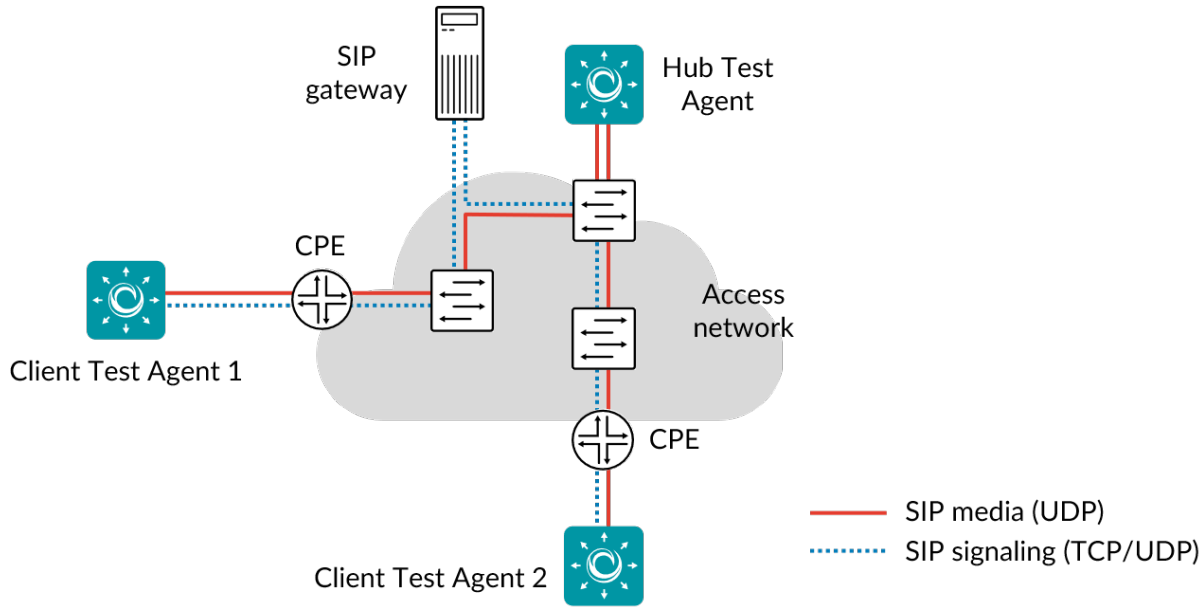
The SIP clients verify the availability and performance of SIP servers by measuring the completion time for various SIP operations (registration, invite, and so on). The clients also measure the performance and voice quality of VoIP sessions, thus making sure that the network and the service in general work adequately.



Read more about SIP testing with Routing Active Testing on *this page* (page 348).

8.7.2 SIP

This task measures response times in SIP signaling and the voice quality of RTP media flows.



SIP testing works in a hub-and-spoke topology with a passive hub and a number of Test Agents as clients. In each testing cycle, the clients will register and then set up a call towards the hub. The call lasts for a specified amount of time, after which the client terminates the call and unregisters. The next cycle then begins.

The SIP clients verify the availability and performance of SIP servers by measuring the completion time for various SIP operations. During the test cycles, each client measures completion times for SIP operations (register, invite, hang-up, and unregister). During VoIP calls, both the hub and the client measure the performance and quality of the VoIP session (rate, loss, misorderings, jitter) as well as voice quality MOS scores.

The Test Agents support execution of multiple concurrent SIP tests on different interfaces (one SIP test/account per interface).

The following audio codecs are supported: GSM Full Rate, G.711 A-law, and G.711 μ-law.

This task works only with IPv4.

8.7.2.1 Prerequisites

To run SIP measurements, you need to have at least two Test Agents installed. If you haven't already done the installation, consult this page: *Getting started with IP telephony (SIP and VoIP) measurements* (page 17).

Make sure that you have prepared the Routing Active Testing account with SIP accounts. Read more on this topic on the page *Setting up SIP accounts* (page 22).

It is possible to run the SIP test cycles without the Test Agents ever registering. The VoIP calls will then be set up directly towards the IP address of the hub Test Agent. It is also possible to perform only an initial registration; the Test Agents will then register with the SIP server once at the beginning, but in each test cycle they will only make calls, without unregistering and re-registering.

The number of calls in each test cycle is configurable. The calls will be made sequentially.

In your test or monitor, add a SIP task and fill in the mandatory parameters below.

8.7.2.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Hub: Test Agent interface that will act as hub for this task. The client Test Agents will make calls towards the hub.
  - SIP account: After selecting the hub, you are prompted to select what SIP account to associate with the hub. You can choose among the SIP accounts that are available under Account > SIP accounts.
- Clients: Test Agent interfaces that will make calls towards the hub.
  - SIP accounts: After selecting a client, you are prompted to select what SIP account to associate with that client. You can choose among the SIP accounts that are available under Account > SIP accounts.
- Registration during test cycles: This setting determines whether the Test Agents will do SIP registration during testing.
  - *Yes:* The Test Agents will unregister and re-register in each test cycle.
  - *Only once at the beginning:* The Test Agents will register only once at the beginning of the test. In subsequent cycles, only calls are made.
  - *No:* The Test Agents will never register. The client Test Agents will make SIP calls directly to the IP address of the hub Test Agent on the standard SIP port 5060. The hub will bind to this port, so in this mode only one instance of the SIP task can be running on a given hub interface.
- Number of calls per test cycle: Number of VoIP calls to make during a test cycle. If this is set to zero, only SIP registration and unregistration will be done in each test cycle. Default: 1.
- Time to keep a call/registration: Specifies how long to keep up a VoIP call before it is terminated and either a new call or unregistration is attempted. If no calls but only registration and unregistration are done during the test cycles, this parameter specifies how long to wait between registration and unregistration. Min: 1 s. Default: 10 s.

Thresholds for errored seconds (ES)

- SIP response time (ms): Maximum allowed completion time for SIP operations (registration, unregistration, invite, and hang-up). If any of these operations takes longer to complete, an errored second is triggered. Min: 1 ms. Default: 400 ms.
- MOS: Lowest allowed Mean Opinion Score during VoIP calls. If during a call the MOS value drops below this level, an errored second is triggered. Regarding MOS, see *this page* (page 474), which also details how the MOS is calculated. Range: 1 … 5. Default: 4.

Advanced

- DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in the IP packet headers, for SIP signaling as well as RTP media flows. See *this page* (page 511). Range: 0 … 63. Default: 0.
- Transport: Transport protocol to be used for SIP messages: UDP or TCP. Default: UDP.
- Codec: Audio codec to be used for RTP media flows. One of: GSM Full Rate, G.711 A-law (PCMA), or G.711 μ-law (PCMU). Default: GSM Full Rate ("GSM").

8.7.2.3 Result metrics

Signaling

- **Register (ms):** Average time taken to complete SIP Register operation.
- **Invite (ms):** Average time taken to complete SIP Invite operation.
- **Hangup (ms):** Average time taken to complete SIP Hangup operation.
- **Unregister (ms):** Average time taken to complete SIP Unregister operation.
- **ES total (%):** Aggregated errored second percentage, taking into account all types of error.
- **ES register (%):** Errored second percentage for SIP Register.
- **ES invite (%):** Errored second percentage for SIP Invite.
- **ES hangup (%):** Errored second percentage for SIP Hangup.
- **ES unregister (%):** Errored second percentage for SIP Unregister.

Media

- **Rate (Mbit/s):** SIP data rate.
- **Loss (%):** Packet loss in percent.
- **Jitter (ms):** *Jitter (delay variation)* (page 474).
- **Misorders (packets):** Number of misordered packets.
- **MOS:** Estimated voice quality Mean Opinion Score, calculated from network metrics.
- **ES total (%):** Errored second percentage, equal to errored seconds due to poor MOS.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

Call statistics

- **Calls:** Number of SIP calls.
- **Success rate:** SIP call success rate.
- **Blocked calls:** Number of SIP calls that were not successfully set up.
- **Dropped calls:** Number of SIP calls that were dropped due to no audio (RTP) flow being received for 4 seconds.

8.7.3 SIP server response codes

Below is a list of selected response codes that may be sent from a SIP server. A full list can be found in ► IETF RFC 3261, section 21.

8.7.3.1 1xx – Provisional responses

- **100 Trying:** This response indicates that the request has been received by the next-hop server and that some unspecified action is being taken on behalf of this call (for example, a database is being consulted).

8.7.3.2 4xx – Client failure responses

- **403 Forbidden:** The server understood the request, but is refusing to fulfill it.
- **404 Not Found:** The server has definitive information that the user does not exist at the domain specified in the Request-URI. This status is also returned if the domain in the Request-URI does not match any of the domains handled by the recipient of the request.
- **408 Request Timeout:** The server could not produce a response within a suitable amount of time, for example, if it could not determine the location of the user in time. The client may repeat the request without modifications at any later time.

8.8 Wi-Fi network testing

8.8.1 Introduction to Wi-Fi network testing

The Test Agent software has the capability to connect to Wi-Fi (WiFi) networks and to run tests and monitors over such networks. This requires a Test Agent with a Wi-Fi Network Interface Card (NIC).

If you are using a preinstalled Test Agent from Routing Active Testing, a specific hardware model with an mPCIe Wi-Fi card from Intel is required (“HW Medium Wi-Fi”). Such Wi-Fi-capable preinstalled Test Agents have the iwlwifi driver installed. Note: These preinstalled Test Agents are no longer available for purchase.

If you are using x86 hardware not provided by Routing Active Testing, you again need a Wi-Fi NIC made by Intel (iwlwifi driver).

Currently the cards supported are those available for version 4.14 of the Linux kernel.

The Wi-Fi cards provided by Routing Active Testing have been tested together with our hardware. If you use a different Wi-Fi card, measurement accuracy cannot be guaranteed since not all possible setups have been verified.

The following Wi-Fi standards are supported:

- IEEE 802.11g
- IEEE 802.11n
- IEEE 802.11ac

8.8.1.1 Disclaimer

Wi-Fi result metrics are affected by many factors beyond the control of Routing Active Testing, such as other traffic, interference from other signals, and materials used in building structures. The results can therefore vary widely and must be interpreted with caution.

8.8.1.2 Related topics

- Use the *Wi-Fi interface configuration dialog* (page 182) to configure the Wi-Fi card.
- Use the *Wi-Fi logger* (page 353) task to collect Wi-Fi network measurements.
- Use the *Wi-Fi switcher* (page 354) task to change Wi-Fi interface parameters.

8.8.2 Wi-Fi scan

This task scans for available Wi-Fi networks in the vicinity of the Test Agent and returns a list of the networks found.

The same procedure is performed (though not as a test or monitor task) when you click the Scan button in the Test Agent's *Wi-Fi interface configuration dialog* (page 182).

8.8.2.1 Parameters

- Interface: Test Agent Wi-Fi interface to use for scanning.

8.8.2.2 Result metrics

- **Network name (SSID):** Name (Service Set ID) of the Wi-Fi network.
- **BSSID:** Basic Service Set Identifier of an access point within the Wi-Fi network. This is typically the MAC address of the access point.
- **Frequency:** The Wi-Fi channel frequency in MHz.
- **Signal level:** Wi-Fi channel signal strength in dBm.
- **Flags:** These indicate, among other things, the authentication method in use and are reported by WPA supplicant. Full explanations of these flags are found in the ► WPA supplicant documentation.

8.8.3 Wi-Fi logger

This task lets you log Wi-Fi network parameters using a Test Agent equipped with a Wi-Fi card. For details on supported hardware, see *here* (page 351).

You can change the Wi-Fi interface and its settings dynamically while running a test using the *Wi-Fi switcher* (page 354) task.

For normal configuration, use the *Wi-Fi interface configuration settings* (page 182).

8.8.3.1 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent Wi-Fi interfaces to log.

Thresholds for errored seconds (ES)

For each of the following thresholds, Routing Active Testing will indicate an errored second if the quantity drops below the threshold during that second.

- Signal (dBm): Received Signal Strength Indication (RSSI). Min: −120 dBm. Max: −25 dBm. Default: −100 dBm.
- TX bitrate (Mbit/s): Theoretical maximum transmit data rate (from Test Agent to access point) in current conditions as reported by the Wi-Fi card. No default.
- RX bitrate (Mbit/s): Theoretical maximum receive data rate (from access point to Test Agent) in current conditions as reported by the Wi-Fi card. No default.
- TX retries: Percentage of transmit retries. No default.

8.8.3.2 Result metrics

- **Received Signal Strength Indication (dBm):** Received Signal Strength Indication.
- **TX bitrate (Mbit/s):** Theoretical maximum transmit data rate (from Test Agent to access point) reported by the Wi-Fi card.
- **RX bitrate (Mbit/s):** Theoretical maximum receive data rate (from access point to Test Agent) reported by the Wi-Fi card.
- **Tx MCS index:** Transmit Modulation Coding Scheme index (used from Test Agent to access point).
- **Rx MCS index:** Receive Modulation Coding Scheme index (used from access point to Test Agent).
- **Guard interval (ms):** Guard interval used.
- **Number of transmit MIMO streams:** Number of Multiple-Input Multiple-Output (MIMO) transmit streams.
- **Number of receive MIMO streams:** Number of Multiple-Input Multiple-Output (MIMO) receive streams.

- **TX retries:** Percentage of transmit retries.
- **ES (%):** Aggregated errored second percentage, taking into account all types of error.
- **ES Signal (%):** Errored second percentage for Signal (RSSI).
- **ES TX bitrate (%):** Errored second percentage for transmit data rate.
- **ES RX bitrate (%):** Errored second percentage for receive data rate.
- **ES TX retries (%):** Errored second percentage for transmit retries.

If the network changes while the Wi-Fi logger is running (for example, if you switch to a new access point), that change is recorded in the message log which also appears in the output.

8.8.4 Wi-Fi switcher

This task is a utility which lets you change Wi-Fi interface parameters while running a *Wi-Fi logger* (page 353) task. When this task executes, it modifies the existing configuration of the Wi-Fi interface (described on *this page* (page 182)).

Warning: Be careful if using this task in orchestration (that is, in a test or monitor template used by an orchestrator). When an orchestrator configures a Wi-Fi interface, it normally does so in the course of creating or modifying a Test Agent, and not using the Wi-Fi switcher. If the orchestrator runs a Wi-Fi switcher task, it will end up out of sync with Control Center. To restore sync in such a situation, you need to run a second Wi-Fi switcher task which undoes the configuration changes made by the first.

Note: All Wi-Fi interface parameters for which you want a specific value must be set explicitly in the Wi-Fi switcher task. It is not possible to keep the current parameter value by leaving the field blank.

8.8.4.1 Parameters

- Interface: The Test Agent Wi-Fi interface on which to change parameters.

Network

- Network name (SSID): New Wi-Fi network to switch to.
- BSSID: Basic Service Set Identifier of the access point to connect to in the new Wi-Fi network.

Detailed configuration

These parameters are the same as in the Wi-Fi interface configuration GUI, which is described *here* (page 182).

8.8.4.2 Result metrics

- Outcome of configuration switch: Success or failure
- Log detailing the configuration switch process

8.9 Mobile network testing

8.9.1 Introduction to mobile network testing (including configuration)

Certain Test Agents (preinstalled on HW Medium Mobile hardware) can measure network performance and user experience in mobile networks by means of a built-in mobile network interface.

Note: Such preinstalled Test Agents are no longer available for purchase.

This function is currently limited to European frequency bands and to the LTE (4G), WCDMA (3G), and GSM (2G) 3GPP standards.

8.9.1.1 Related topics

- Use the *Mobile interface configuration dialog* (page 178) to configure the 4G modem.
- Use the *Mobile logger* (page 355) task to collect mobile network measurements.
- Use the *Mobile switcher* (page 357) task to change mobile interface parameters.

8.9.2 Mobile logger

This task lets you log mobile network parameters using a Test Agent that has a mPCIe 4G modem in it. Besides 4G (LTE), this device also supports 3G (WCDMA) and 2G (GSM) mobile networks. Note that this only works with Test Agents specifically supplied from Juniper Networks with this extra hardware support.

You can change the mobile interface and its settings (APN, RAT, band) using the *Mobile switcher* (page 357) utility.

8.9.2.1 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent mobile interfaces to log.

Thresholds for errored seconds (ES)

For each of the following thresholds, Routing Active Testing will indicate an errored second if the quantity drops below the threshold during that second.

- RSSI (dBm): Received Signal Strength Indication (mobile technology independent measurement). Min: −120 dBm. Max: −25 dBm. Default: −100 dBm.
- RSRP (dBm): Reference Signal Received Power (LTE). Min: −140 dBm. Max: −44 dBm. Default: −95 dBm.
- RSCP (dBm): Received Signal Code Power (WCDMA). Min: −120 dBm. Max: −25 dBm. Default: −85 dBm.
- RSRQ (dB): Reference Signal Received Quality (LTE). Min: −19.5 dB. Max: −3 dB. No default.
- Ec/Io (dB): Per-chip signal-to-noise ratio (WCDMA). Min: −32 dB. Max: 0 dB. Default: −13 dB.
- SINR (dB): Signal to interference-plus-noise ratio (LTE). Min: −20 dB. Max: 30 dB. Default: 10 dB.

8.9.2.2 Result metrics

Generic

- **ES total (%):** Aggregated errored second percentage, taking into account all types of error.
- **ES RSSI (%):** Errored second percentage for RSSI.

LTE-specific

- **ES RSRP (%):** Errored second percentage for RSRP.
- **ES RSRQ (%):** Errored second percentage for RSRQ.
- **ES SINR (%):** Errored second percentage for SINR.

WCDMA-specific

- **ES RSCP (%):** Errored second percentage for RSCP.
- **ES Ec/Io (%):** Errored second percentage for Ec/Io.

8.9.3 Mobile switcher

This task is a utility which lets you change mobile interface parameters while running a *Mobile logger* (page 355) task.

8.9.3.1 Parameters

- Mobile interface: The Test Agent mobile interface on which to change parameters.
- APN: New APN to switch to. Leave blank to keep the current APN.
- Mode: Change the mobile device's choice of preferred radio access technology and (in the case of LTE) preferred frequency band. Note that regardless of this setting, the mobile network has the final word on what RAT and band the mobile device will use.
  - *Don't change:* No change from current setting.
  - *Auto:* No RAT or band preference.
  - *GSM – Auto:* The mobile device will prefer GSM as radio access technology.
  - *WCDMA – Auto:* The mobile device will prefer WCDMA as radio access technology.
  - *LTE – Auto:* The mobile device will prefer LTE as radio access technology. No preference regarding frequency band.
  - *LTE – 2600:* The mobile device will prefer LTE and the 2600 MHz band (LTE Band 7).
  - *LTE – 2100:* The mobile device will prefer LTE and the 2100 MHz band (LTE Band 1).
  - *LTE – 1800:* The mobile device will prefer LTE and the 1800 MHz band (LTE Band 3).
  - *LTE – 900:* The mobile device will prefer LTE and the 900 MHz band (LTE Band 8).
  - *LTE – 800:* The mobile device will prefer LTE and the 800 MHz band (LTE Band 5).

8.9.4 Introduction to 5G core network testing

Test Agent Applications are capable of connecting to a 5G core network and running tests and monitors to measure the performance of that network.

The tests and monitors can make use of any task types supported by Test Agent Applications (see the above table), while connecting to the 5G core network is done by means of emulating a UE and a gNodeB (in *this dialog* (page 201)) and setting up a `tun` (tunnel) interface for data transmission. In addition, a control channel is set up towards the 5G core AMF. When configuring a test or monitor, you select the `tun` interface in the configuration dialog.

You can set up multiple gNodeB-and-UE pairs and reconfigure the tunnel interface for any of these pairs using the *RAN switcher* (page 358) task.

When you run the RAN switcher, a set of 5G-specific KPIs are computed and displayed.

This feature is currently not available on Test Agent Appliances.

8.9.4.1 Limitations

The 5G core network testing feature currently has the following limitations:

- The tunnel interface names cannot be configured; they will always be in the format `uesimtun<n>`, where `<n>` starts at zero.
- The tunnel interface MTU is fixed at 1400 bytes.
- Only one gNodeB is supported. The number of UEs is limited by system capabilities.
- The open-source software used to emulate UEs and a gNodeB does not support namespaces. This will cause issues if the interface connecting back to the 5G core is in a different namespace from the emulated tunnel interface. The "namespace aware" option in the Test Agent CLI, mentioned *here* (page 161), cannot be used if the 5G core network testing feature is enabled in the CLI.
- Some KPIs are not available. For details, see this page.
- The KPI resolution is limited to milliseconds.
- IPv6 is not supported.
- At least 2 vCPUs are required for a Test Agent Application with the 5G feature enabled. A single vCPU is not sufficient.

8.9.5 RAN switcher

This task is a utility which lets you reconfigure the tunnel interface for a gNodeB-and-UE pair (detach + attach) while doing *5G core network testing* (page 358). In the process, 5G KPIs are fetched for the reconfigured interface.

Note: This task is unrelated to the other ("Mobile") tasks in this category.

8.9.5.1 Parameters

- RAN interface: Select the Test Agent `tun` interface (identifying an emulated gNodeB and UE) for which you want to reconfigure the tunnel interface.

8.10 Ethernet service activation testing

8.10.1 Introduction to Ethernet service activation testing

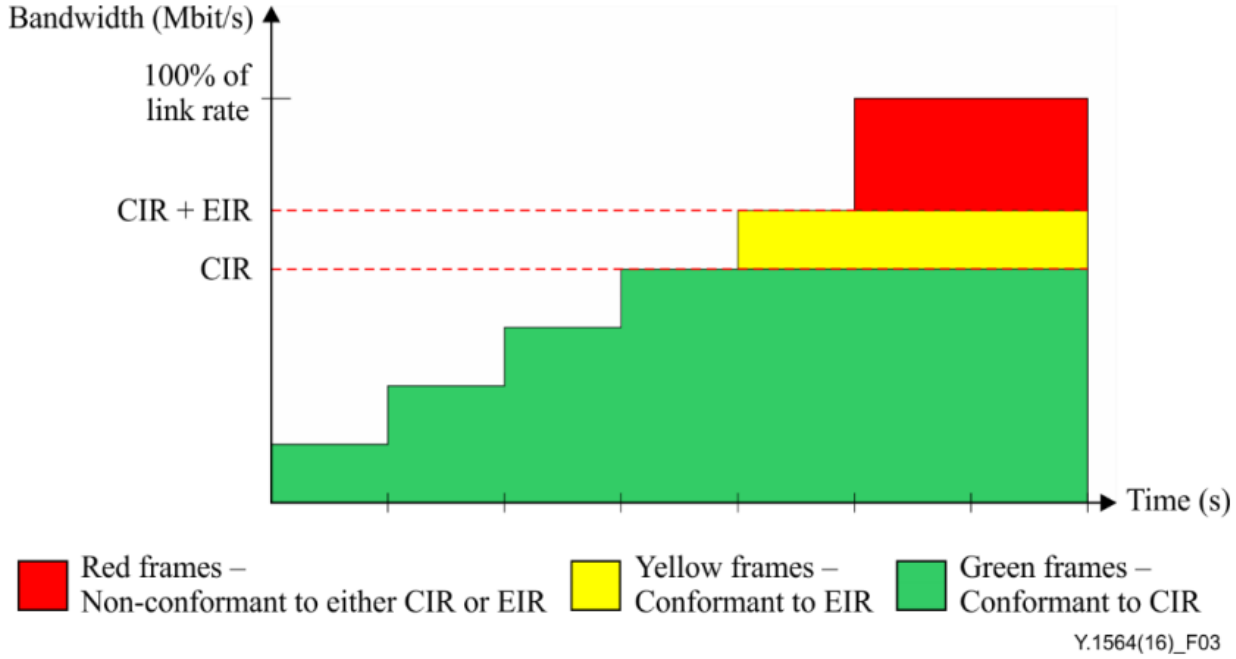
The purpose of Ethernet service activation tests is to validate carrier- or enterprise-grade Ethernet services according to ► ITU-T Recommendation Y.1564 as well as ► Metro Ethernet Forum technical specifications.

These tests go further than ► IETF RFC 2544, a benchmarking methodology for hubs, switches, and routers. Together, the tests constitute an out-of-service test methodology for assessing the configuration and performance of an Ethernet service prior to customer notification and delivery.

8.10.1.1 Compliance with ITU-T Y.1564

Routing Active Testing is compliant with ITU-T Recommendation Y.1564, "Ethernet Service Activation Test Methodology" (color-aware and non-color-aware), in verifying that the Ethernet connection complies with service acceptance criteria (frame loss ratio, frame delay, frame delay variation, Ethernet availability) for your Ethernet services, and that the Quality of Service (QoS) profile is correctly configured.

In particular, the Routing Active Testing tests verify that the traffic policing based on the concepts of *Committed Information Rate (CIR)* and *Excess Information Rate (EIR)* works as expected for colored as well as non-colored flows. The picture below, taken from the Y.1564 specification, illustrates the relationship between CIR, EIR, and the color coding of traffic.



The description of the task type for each test has a reference to the corresponding identification in ITU-T Y.1564.

To fully comply with ITU-T Y.1564 as well as ► Y.1563 (“Ethernet Frame Transfer and Availability Performance”), the Ethernet service activation tests must be complemented with *TCP/UDP performance* (page 290) tests. These send and receive multiple TCP sessions and UDP flows with differentiated QoS settings to make sure that packet loss, jitter, and maximum one-way delay are within your service acceptance criteria and meet your service availability requirements.

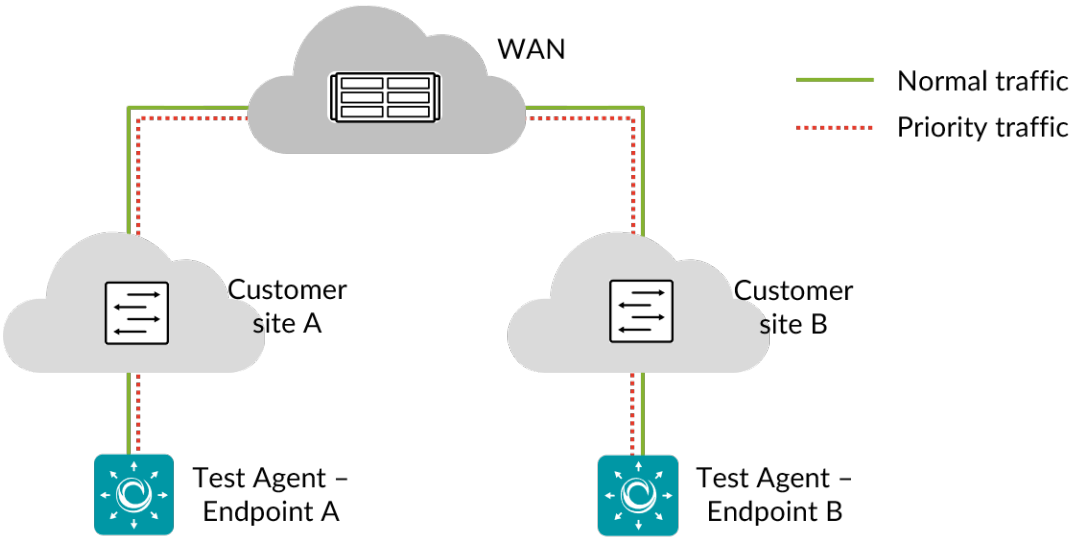
Therefore, for a full validation of an Ethernet-based service, first confirm that the service is correctly configured by running the Ethernet service activation tests, and then validate the quality of the services as delivered to the end-user by means of the TCP/UDP performance tests.

8.10.1.2 Compliance with MEF Carrier Ethernet 2.0

Routing Active Testing is also compliant with Metro Ethernet Forum Carrier Ethernet (CE) 2.0 services certification (E-Line, E-LAN, E-Tree, and E-Access).

8.10.1.3 Physical test setup

For Ethernet service activation tests, you need to install and register at least two Test Agents (acting as sender and receiver) and connect them to a switch or router port in your network. If you want to test more connections, you need to create separate tests for each connection.



Two interfaces are used on the Test Agents: one interface on each Test Agent is used as test interface, and the other (“eth0”) is used for management, maintaining an encrypted connection to the Routing Active Testing server. The management interface cannot be used for testing.

All tests are very fast to run, and they are suitable both for lab setups and for verifying Ethernet services end-to-end in a live network.

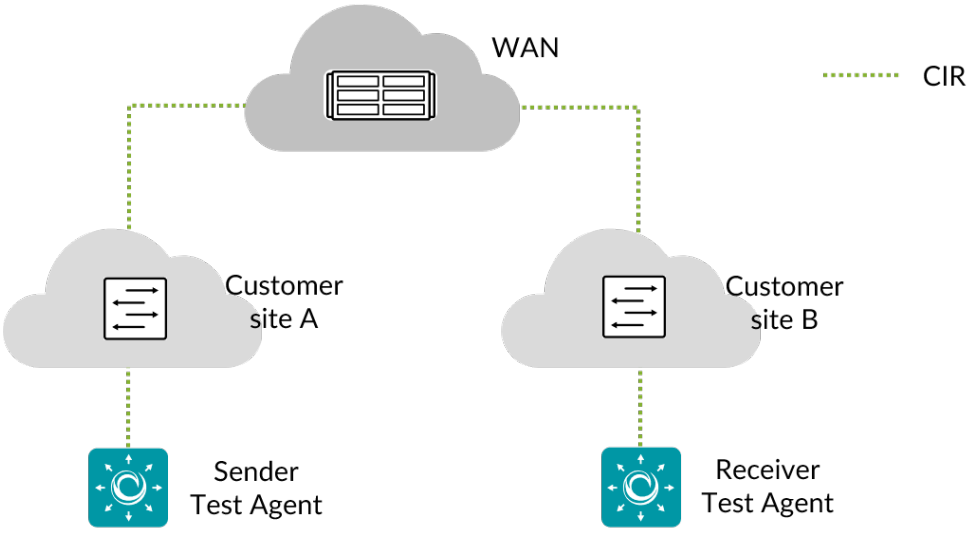
Ethernet service activation tests are supported for IPv4 only.

8.10.1.4 Further reading

For details on individual tasks handling Ethernet service activation tests, see the following pages:

- *Simple CIR validation test* (page 361)
- *Step load CIR test* (page 363)
- *EIR configuration test (color-aware)* (page 366)
- *EIR configuration test (non-color-aware)* (page 369)
- *Traffic policing test (color-aware)* (page 371)
- *Traffic policing test (non-color-aware)* (page 374)

8.10.2 Simple CIR validation test



This task verifies that an Ethernet service QoS profile has the expected behavior. It is assumed that the service has a QoS profile defined with the CIR parameter as well as maximum frame loss ratio, delay, and jitter values for conforming frames. For the generated flow the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

8.10.2.1 Reference

The test performed conforms to ► ITU-T Y.1564 section 8.1.2, test A.1 (“CIR configuration test: Simple CIR validation”).

8.10.2.2 Test procedure

- The sender Test Agent generates frames at a rate equal to CIR.
- The receiver Test Agent measures received rate, loss, delay, and jitter on the flow.

8.10.2.3 Fail criteria

- The test fails if any of the thresholds Max frame loss, Max frame delay, or Max frame jitter is exceeded.

8.10.2.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Test duration (s): Duration of the test in seconds. Min: 5 s. Max: 60 s. Default: 20 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

QoS profile

- CIR (Mbit/s): Committed Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.
- Max frame delay (ms): Maximum frame delay for conforming packets. Min: 0 ms. Max: 1000 ms. No default.
- Max frame jitter (ms): Maximum frame *jitter* (page 474) for conforming packets. Min: 0 ms. Max: 1000 ms. No default.

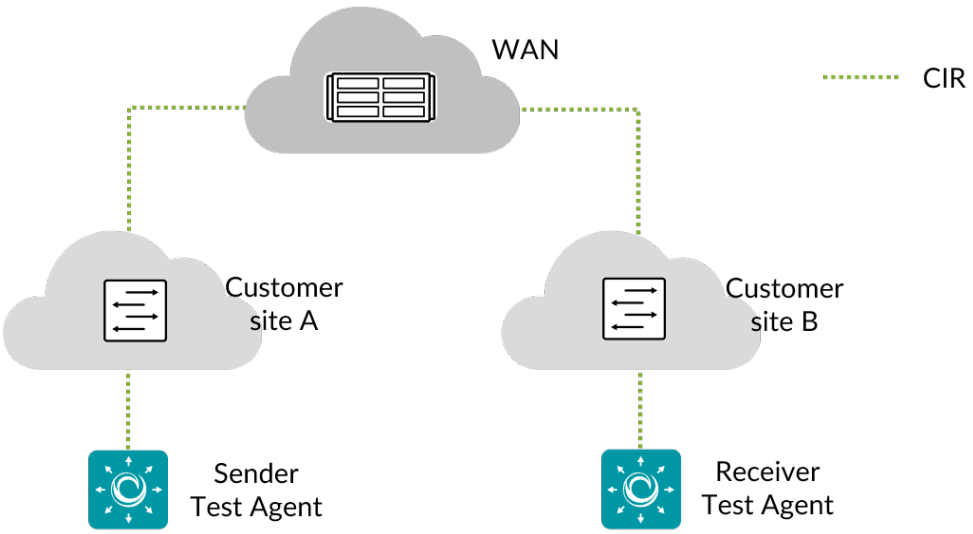
Traffic profile

- Frame size (bytes): *Ethernet frame size* (page 512) in bytes. Min: 64 bytes. Max: 1514 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5000.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5000.
- DSCP: *Differentiated Services Code Point* (page 511). Min: 0. Max: 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN header. Min: 0. Max: 7. Default: 0.

8.10.2.5 Result metrics

- **Rate (Mbit/s):** Ethernet data rate.
- **Loss (%):** Packet loss.
- **Delay (ms):** Average one-way delay.
- **Jitter (ms):** Jitter (delay variation).
- **Pass/fail** outcome of test.

8.10.3 Step load CIR test



This task verifies that an Ethernet service QoS profile behaves as expected as the bit rate is increased. It is assumed that the service has a QoS profile with the CIR parameter as well as the maximum frame loss ratio, delay, and jitter values for conforming frames. For the generated flow the following parameters need to be specified: frame size, load frame size, the source and destination UDP port, DSCP value, and VLAN priority (p-bits).

8.10.3.1 Reference

The test performed conforms to ► ITU-T Y.1564 section 8.1.2, test A.2 (“CIR configuration test: Step load CIR test”).

8.10.3.2 Test procedure

- The sender Test Agent starts by generating frames at a rate of 0.25 × CIR.
- The receiver Test Agent measures received rate, loss, delay, and jitter on the flow. If the last three metrics are below the maximum frame thresholds, the test is repeated at a higher rate. In each step, the rate is increased by the Load step size parameter. This procedure is iterated until CIR is reached.

8.10.3.3 Fail criteria

- The test fails if any of the thresholds Max frame loss, Max frame delay, or Max frame jitter is exceeded in any step.

8.10.3.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Load step size (Mbit/s): The amount by which the rate is increased in each step. No default.
- Test duration (s): Duration of the test in seconds. Min: 5 s. Max: 60 s. Default: 20 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

QoS profile

- CIR (Mbit/s): Committed Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.
- Max frame delay (ms): Maximum frame delay for conforming packets. Min: 0 ms. Max: 1000 ms. No default.
- Max frame jitter (ms): Maximum frame *jitter* (page 474) for conforming packets. Min: 0 ms. Max: 1000 ms. No default.

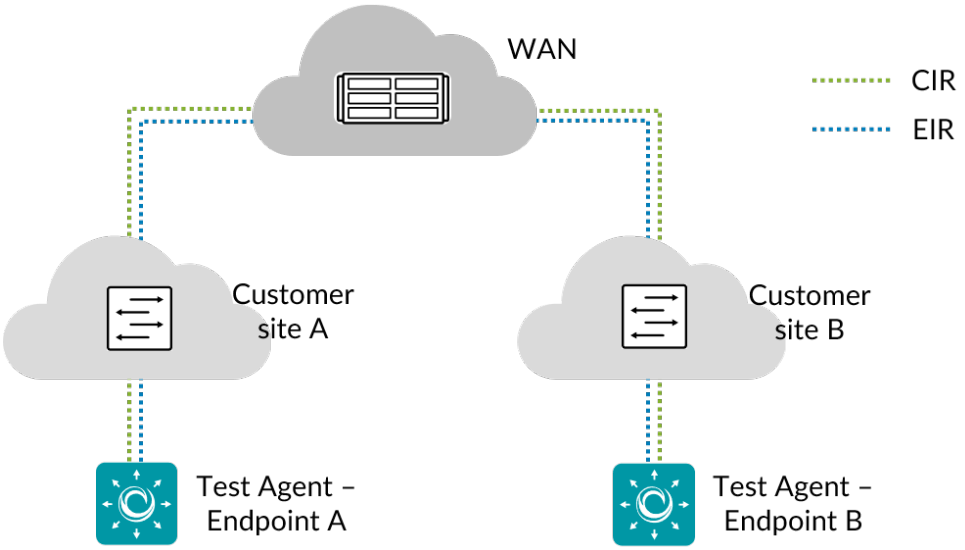
Traffic profile

- Frame size (bytes): *Ethernet frame size* (page 512) in bytes. Min: 64 bytes. Max: 1514 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5000.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5000.
- DSCP: *Differentiated Services Code Point* (page 511). Min: 0. Max: 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN header. Min: 0. Max: 7. Default: 0.

8.10.3.5 Result metrics

- **Rate (Mbit/s):** Ethernet data rate.
- **Loss (%):** Packet loss.
- **Delay (ms):** Average one-way delay.
- **Jitter (ms):** Jitter (delay variation).
- **Pass/fail** outcome of test.

8.10.4 EIR configuration test (color-aware)



This task verifies that a colored Ethernet service QoS profile has the expected behavior. It is assumed that the service has a QoS profile defined with CIR and EIR parameters as well as maximum frame loss ratio, delay, and jitter values for conforming frames. The test being color-aware means that one green and one yellow flow (two QoS classes) are present. In accordance with ITU-T Y.1564, the green flow is checked against CIR and the yellow flow against EIR. For both flows the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

8.10.4.1 Reference

The test performed conforms to ► ITU-T Y.1564 section 8.1.2, test B.1 (“EIR configuration test, colour aware”).

8.10.4.2 Test procedure

- The sender Test Agent generates green frames at a rate equal to CIR, and yellow frames at a rate equal to EIR. The combined rate should not exceed the link rate.
- The receiver Test Agent measures received rate, loss, delay, and jitter on both flows (green and yellow).

8.10.4.3 Fail criteria

- The test fails if any of the thresholds Max frame loss, Max frame delay, or Max frame jitter is exceeded for the green flow.

8.10.4.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Test duration (s): Duration of the test in seconds. Min: 5 s. Max: 60 s. Default: 20 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

QoS profile

- CIR (Mbit/s): Committed Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- EIR (Mbit/s): Excess Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.
- Max frame delay (ms): Maximum frame delay for conforming packets. Min: 0 ms. Max: 1000 ms. No default.
- Max frame jitter (ms): Maximum frame *jitter* (page 474) for conforming packets. Min: 0 ms. Max: 1000 ms. No default.

Traffic profile for CIR (green) flow

- Frame size (bytes): *Ethernet frame size* (page 512) in bytes. Min: 64 bytes. Max: 1518 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5000.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5000.
- DSCP: *Differentiated Services Code Point* (page 511) in IP packet headers. Range: 0 … 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN headers. Min: 0. Max: 7. Default: 0.

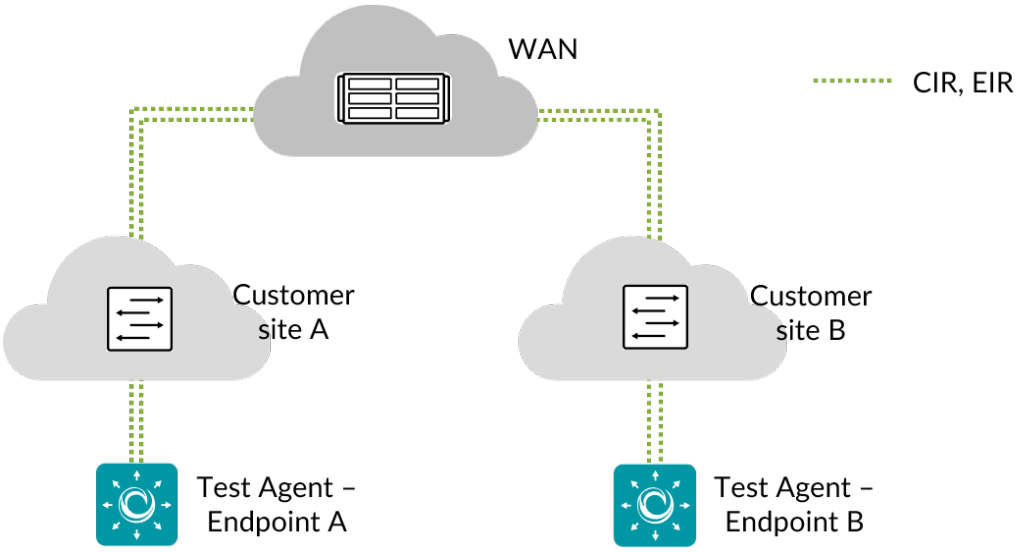
Traffic profile for EIR (yellow) flow

- Frame size (bytes): Ethernet frame size in bytes. Min: 64 bytes. Max: 1518 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5001.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5001.
- DSCP: Differentiated Services Code Point in IP packet headers. Range: 0 … 63. Default: 0.
- VLAN priority (PCP): Priority Code Point in VLAN headers. Min: 0. Max: 7. Default: 0.

8.10.4.5 Result metrics

- **Rate (Mbit/s):** Ethernet data rate.
- **Loss (%):** Packet loss.
- **Delay (ms):** Average one-way delay.
- **Jitter (ms):** Jitter (delay variation).
- **Pass/fail** outcome of test.

8.10.5 EIR configuration test (non-color-aware)



This task verifies that a non-colored Ethernet service QoS profile has the expected behavior. It is assumed that the service has a QoS profile defined with CIR and EIR parameters as well as maximum frame loss ratio, delay, and jitter values for conforming frames. The test being non-color-aware means that there is only one flow. For the generated flow the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

8.10.5.1 Reference

The test performed conforms to ► ITU-T Y.1564 section 8.1.2, test B.2 (“EIR configuration test, non-colour aware”).

8.10.5.2 Test procedure

- The sender Test Agent generates frames at a rate of CIR + EIR.
- The receiver Test Agent measures received rate, loss, delay, and jitter on the flow.

8.10.5.3 Fail criteria

- The test fails if the measured rate is less than CIR × (1 – Max frame loss / 100).

8.10.5.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Test duration (s): Duration of the test in seconds. Min: 5 s. Max: 60 s. Default: 20 s.

- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

QoS profile

- CIR (Mbit/s): Committed Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- EIR (Mbit/s): Excess Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.

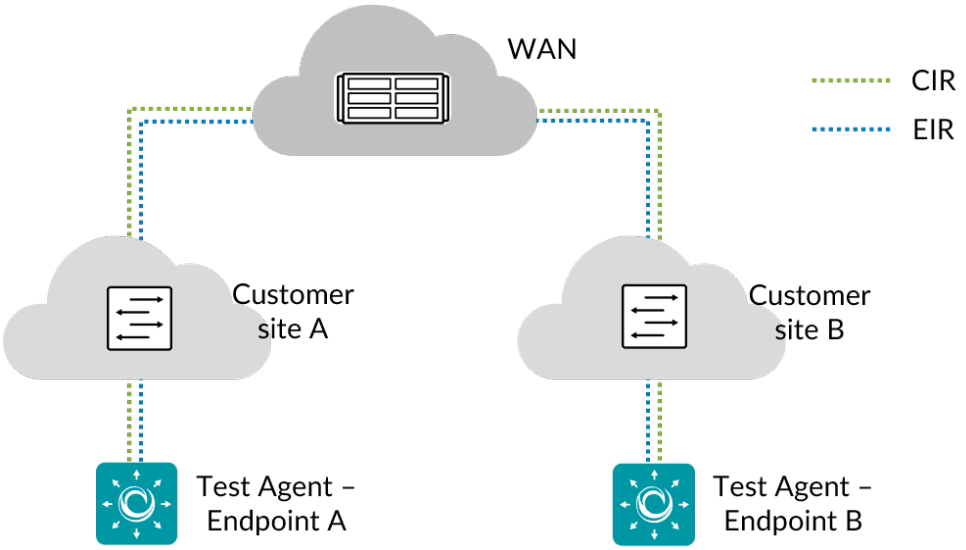
Traffic profile

- Frame size (bytes): *Ethernet frame size* (page 512) in bytes. Min: 64 bytes. Max: 1514 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5000.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5000.
- DSCP: *Differentiated Services Code Point* (page 511). Min: 0. Max: 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN header. Min: 0. Max: 7. Default: 0.

8.10.5.5 Result metrics

- **Rate (Mbit/s):** Ethernet data rate.
- **Loss (%):** Packet loss.
- **Delay (ms):** Average one-way delay.
- **Jitter (ms):** *Jitter (delay variation)* (page 474).
- **Pass/fail** outcome of test.

8.10.6 Traffic policing test (color-aware)



This task verifies that a colored Ethernet service QoS profile has the expected behavior. It is assumed that the service has a QoS profile defined with the CIR and EIR parameters as well as maximum frame loss ratio, delay, and jitter values for conforming frames. The test being color-aware means that one green and one yellow flow (two QoS classes) are present. In accordance with ITU-T Y.1564, the green flow is checked against CIR and the yellow flow against EIR. For both flows the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

8.10.6.1 Reference

The test performed conforms to ► ITU-T Y.1564 section 8.1.2, test C.1 (“Traffic policing test: Colour aware”).

8.10.6.2 Test procedure

- The sender Test Agent generates green frames at a rate equal to CIR, and yellow frames at a rate of 1.25 × EIR. (However, if EIR is less than 20% of CIR, the yellow frames are instead generated at a rate of EIR + 0.25 × CIR.) The combined rate (CIR + EIR) should not be higher than the link rate.
- The receiver Test Agent measures received rate, loss, delay, and jitter on both flows (green and yellow).

8.10.6.3 Fail criteria

- The test fails if any of the thresholds Max frame loss, Max frame delay, or Max frame jitter is exceeded for the green flow, or the total received rate is higher than 1.05 × (CIR + EIR).

8.10.6.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Test duration (s): Duration of the test in seconds. Min: 5 s. Max: 60 s. Default: 20 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

QoS profile

- CIR (Mbit/s): Committed Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- EIR (Mbit/s): Excess Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.
- Max frame delay (ms): Maximum frame delay for conforming packets. Min: 0 ms. Max: 1000 ms. No default.
- Max frame jitter (ms): Maximum frame *jitter* (page 474) for conforming packets. Min: 0 ms. Max: 1000 ms. No default.

Traffic profile for CIR (green) flow

- Frame size (bytes): *Ethernet frame size* (page 512) in bytes. Min: 64 bytes. Max: 1518 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5000.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5000.
- DSCP: *Differentiated Services Code Point* (page 511). Range: 0 … 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN header. Min: 0. Max: 7. Default: 0.

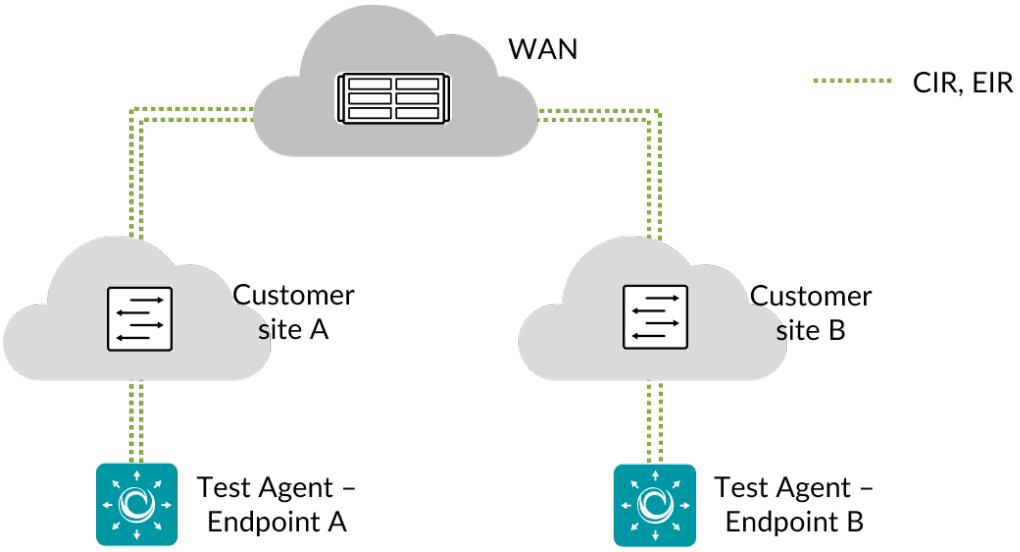
Traffic profile for EIR (yellow) flow

- Frame size (bytes): Ethernet frame size in bytes. Min: 64 bytes. Max: 1518 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5001.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5001.
- DSCP: Differentiated Services Code Point. Range: 0 … 63. Default: 0.
- VLAN priority (PCP): Priority Code Point in VLAN header. Min: 0. Max: 7. Default: 0.

8.10.6.5 Result metrics

- **Rate (Mbit/s):** Ethernet data rate.
- **Loss (%):** Packet loss.
- **Delay (ms):** Average one-way delay.
- **Jitter (ms):** Jitter (delay variation).
- **Pass/fail** outcome of test.

8.10.7 Traffic policing test (non-color-aware)



This task verifies that a non-colored Ethernet service QoS profile has the expected behavior. It is assumed that the service has a QoS profile defined with the CIR and EIR parameters as well as maximum frame loss ratio, delay, and jitter values for conforming frames. The test being non-color-aware means that there is only one flow and one QoS class. For the generated flow the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

8.10.7.1 Reference

The test performed conforms to ► ITU-T Y.1564 section 8.1.2, test C.2 (“Traffic policing test: Non-colour aware”).

8.10.7.2 Test procedure

- The sender Test Agent generates frames at a rate of CIR + 1.25 × EIR. (However, if EIR is less than 20% of CIR, the frames are instead generated at a rate of 1.25 × CIR + EIR.)
- The receiver Test Agent measures received rate, loss, delay, and jitter on the flow.

8.10.7.3 Fail criteria

- The test fails if the measured rate is higher than CIR + 1.05 × EIR or less than CIR × (1 – Max frame loss).

8.10.7.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Test duration (s): Duration of the test in seconds. Min: 5 s. Max: 60 s. Default: 20 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

QoS profile

- CIR (Mbit/s): Committed Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- EIR (Mbit/s): Excess Information Rate. Min: 0.1 Mbit/s. Max: 10,000 Mbit/s. No default.
- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.

Traffic profile

- Frame size (bytes): *Ethernet frame size* (page 512) in bytes. Min: 64 bytes. Max: 1518 bytes. Default: 512 bytes.
- Source UDP port: The source UDP port to use. Range: 1 … 65535. Default: 5000.
- Destination UDP port: The destination UDP port to use. Range: 1 … 65535. Default: 5000.
- DSCP: *Differentiated Services Code Point* (page 511). Min: 0. Max: 63. Default: 0.
- VLAN priority (PCP): *Priority Code Point* (page 516) in VLAN header. Min: 0. Max: 7. Default: 0.

8.10.7.5 Result metrics

- **Rate (Mbit/s):** Ethernet data rate.
- **Loss (%):** Packet loss.
- **Delay (ms):** Average one-way delay.
- **Jitter (ms):** *Jitter (delay variation)* (page 474).
- **Pass/fail** outcome of test.

8.11 Transparency testing

8.11.1 Introduction to transparency testing

Transparency testing consists of packet mangling and network transparency/QoS tests which verify:

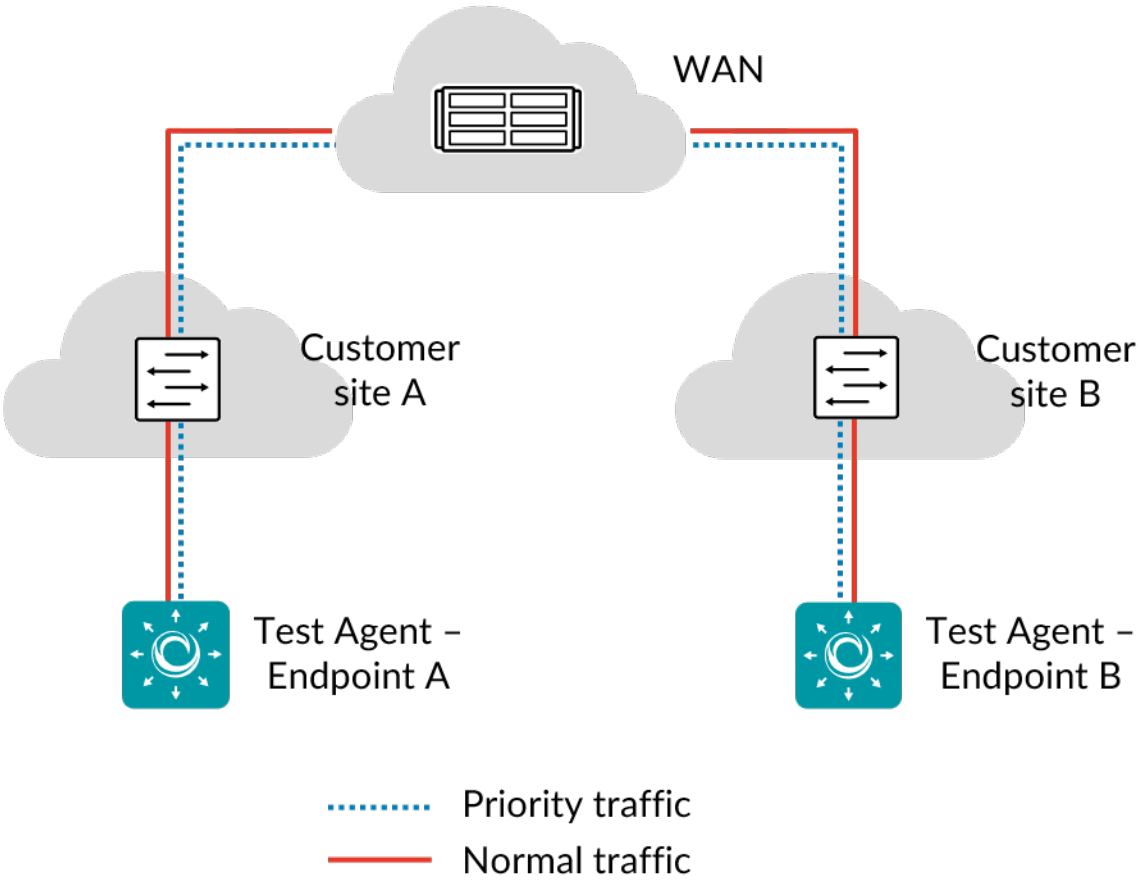
- that Layer 2 and Layer 3 services are transparent, i.e. that various packet types are received unchanged from the sender
- that the network passes various protocols/Ethertypes
- that the network preserves the QoS fields.

This task suite gives you a toolbox for in-depth verification of how the network affects various types of traffic and whether a point-to-point connection has the characteristics you would expect. All tests are designed according to best practice on how to verify the characteristics of point-to-point networks.

The tests are suitable both for lab setups and for end-to-end testing in a live network.

8.11.1.1 Physical test setup

To perform transparency tests, two Test Agents are needed in the setup, one acting as sender and the other as receiver. If you want to test more connections, you need to create separate tests for each connection.

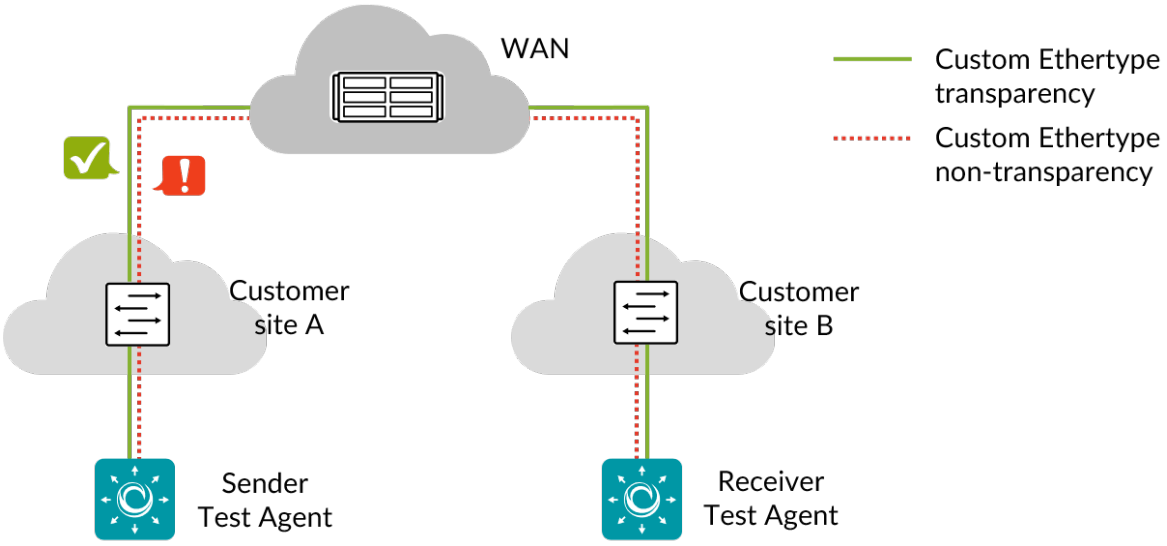


Usually, two interfaces are used on the Test Agents: one for testing, and the other (by default “eth0”) for management, maintaining an encrypted connection to the Routing Active Testing server. It is however possible to use the management interface also for testing. Note, however, that the management interface requires and always has an IP address. Many transparency tests, on the other hand, can be run without an IP address (all except DSCP remapping, Layer 4 destination port DSCP remapping, and Path MTU discovery), and if this is desired a different interface must be used.

Some transparency tests can only be executed on physical interfaces, while others can also run on VLAN interfaces. The generated traffic is either untagged or contains one or two VLAN headers, depending on the task type.

IPv6 is supported for transparency tests, except where otherwise noted for individual tests.

8.11.2 L2 transparency – Custom Ethertype



This task verifies Layer 2 transparency for a custom Ethertype, i.e. checks that the specified Ethertype passes through the network.

Subtypes of Ethertypes are supported. This is relevant for Metro Ethernet Forum Layer 2 Control Protocol tests, where (for example) Ethertype 0×8809 defines different protocols depending on subtype.

8.11.2.1 Test procedure

Five Ethernet frames are sent with the specified MAC addresses and Ethertype, and with dummy payload.

An expected outcome (frames passed or dropped) is specified for the test as a whole. By default the expected outcome is that the frames should pass the network.

8.11.2.2 Fail criteria

The test fails if some frame is treated differently from the expected outcome.

8.11.2.3 Limitations

This test can only run on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.2.4 Parameters

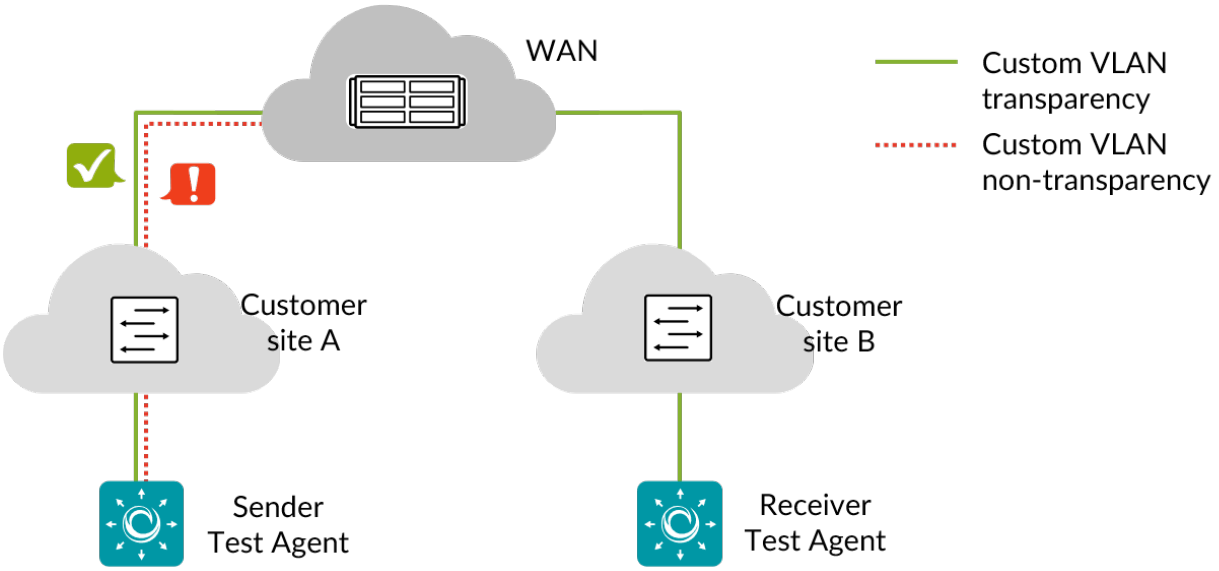
General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Source MAC (optional): Source MAC address in the format `11:22:33:aa:bb:cc`. If you leave this empty, the MAC address of the interface will be used.
- Destination MAC ranges: Comma-separated list of destination MAC address ranges, where each range has the format `11:22:33:aa:bb:cc[-11:22:33:aa:bb:ff]`.
- Ethertype: Ethertype in decimal format (e.g. `65534`) or in hexadecimal format (e.g. `0×1122`). Ethertype 0×8100 is not supported, nor are Ethertypes below 0×0600.
- Subtype: *(Optional)* Subtype for Slow Protocols, a comma-separated list of subtype ranges in decimal format or hexadecimal format (e.g. `0×01-0×03`). Subtype 0×01 = LACP, subtype 0×02 = LAMP, subtype 0×03 = Link OAM, subtype 0×04 = ESMC.
- Expected outcome for test: Expected outcome for all frames in the test: Pass or Drop. Default: Pass.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

8.11.2.5 Result metrics

- **Pass/fail** for the tested custom Ethertype per destination MAC address

8.11.3 L2 transparency – Custom VLAN



This task verifies VLAN transparency for a specific VLAN, that is, that packets with user-specified VLAN tag and *VLAN priority (PCP)* (page 516) are not modified by the network.

8.11.3.1 Test procedure

Five Ethernet frames with 802.1q tags are sent from the sender to the receiver, where they are validated according to the expected VLAN id and VLAN priority values. The frames are correctly created Ethernet frames with a VLAN header and an IP payload.

8.11.3.2 Fail criteria

The test fails if not all frames are received, or if the VLAN ID or PCP has been modified.

8.11.3.3 Limitations

This test can be run only on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.3.4 Parameters

General

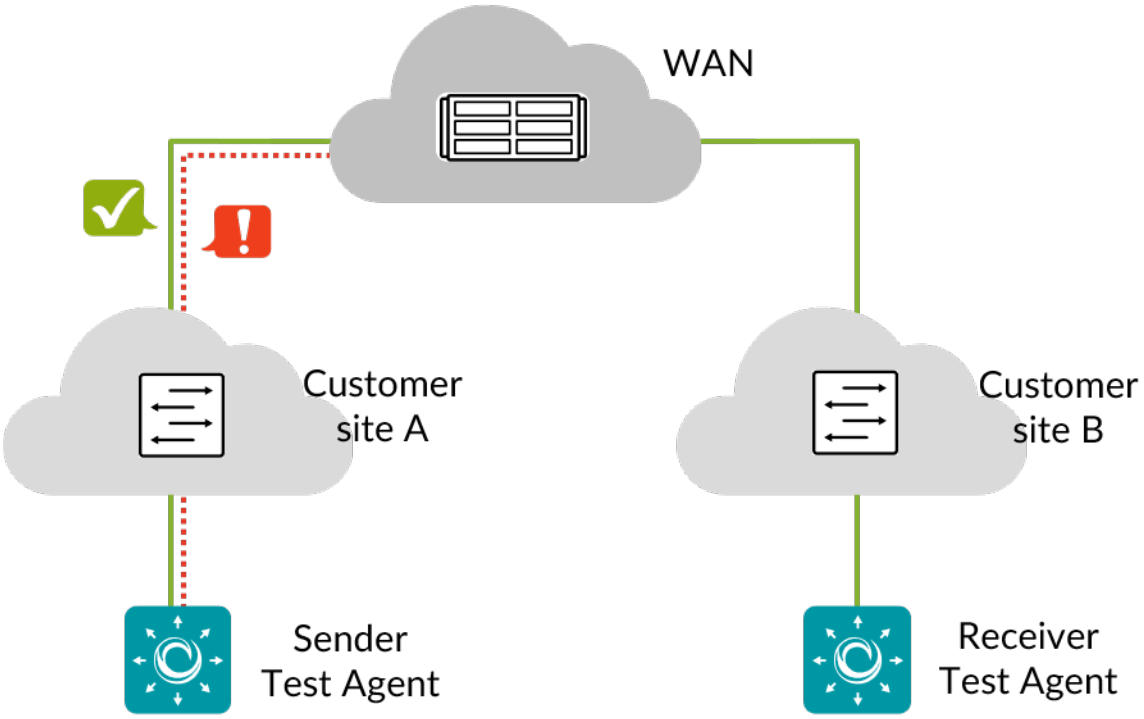
- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Outgoing VLAN id: The outgoing VLAN id of the Ethernet frame at the sender. Note: If you have selected a VLAN interface (e.g. “eth1.100”) under Sender, this field must be left empty or match the id of the selected interface (100 in the example just given).

- Outgoing VLAN priority (PCP): The value of the Priority field in the outgoing VLAN tag at the sender. Range: 0 … 7. No default.
- Frame size (bytes): The size of the outgoing frame, including the VLAN tag. Min: 64 bytes. Max: 1522 bytes. Default: 512 bytes.
- Expect drop: Specify whether the frame is expected to be dropped by the network (Yes or No). Default: No.
- Expected untagged: Specify whether the VLAN tag is expected to be stripped by the network (Yes or No). Default: No.
- Expect VLAN id: The expected VLAN id of the incoming frame at the receiver. Range: 1 … 4096. No default.
- Expected VLAN priority (PCP): The expected value of the Priority field in the incoming VLAN tag at the receiver. Range: 0 … 7. No default.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

8.11.3.5 Result metrics

- **Pass/fail** for the tested custom VLAN

8.11.4 L2 transparency – Ethernet control protocols



This task checks transparency for Ethernet control protocols:

- LACP, Link Aggregation Control Protocol
- EAPoL, Extensible Authorization Protocol over LAN
- MVRP, Multiple VLAN Registration Protocol

8.11.4.1 Test procedure

For each protocol to be tested, five frames are generated and sent.

8.11.4.2 Fail criteria

When the expected outcome for a protocol is set to Pass, the test fails for that protocol if less than four frames are received.

When the expected outcome for a protocol is set to Drop, the test fails for that protocol if at least one frame is received.

8.11.4.3 Limitations

This test can only run on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.4.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

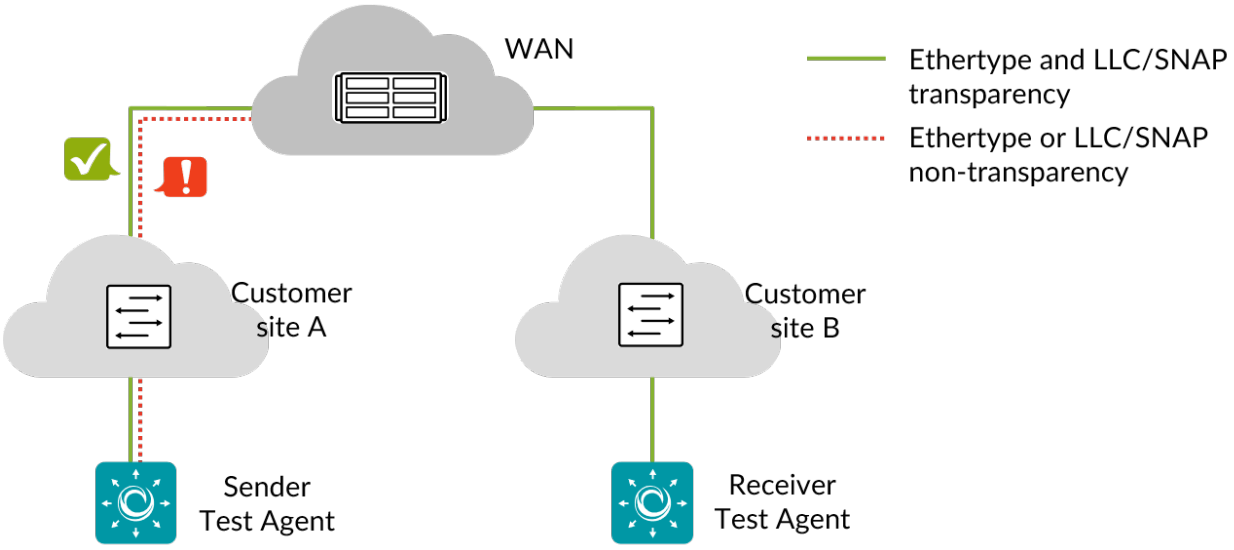
Advanced

- Expected outcome for LACP: Expected outcome for Link Aggregation Control Protocol. Default: Pass.
- Expected outcome for EAPoL: Expected outcome for Extensible Authorization Protocol over LAN. Default: Pass.
- Expected outcome for MVRP: Expected outcome for Multiple VLAN Registration Protocol. Default: Pass.

8.11.4.5 Result metrics

- **Pass/fail** for each tested protocol

8.11.5 L2 transparency – Ethertypes



This task verifies Layer 2 transparency for various Ethertypes and LLC/SNAP protocols, i.e. checks that the Ethertypes and protocols are passed through the network.

8.11.5.1 Test procedure

Five frames are sent in both directions on each protocol. Each frame is an Ethernet frame with correctly set MAC addresses and dummy payload.

8.11.5.2 Fail criteria

The test fails if any of the protocols listed below is not received according to the configured expected outcome. By default all protocols are expected to pass.

8.11.5.3 Limitations

This test can only run on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.5.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

Advanced

- Expected outcome for <Ethertype/protocol>: For each item, select Pass, Drop, or Don't test. Default: Pass.

8.11.5.5 Result metrics

- **Pass/fail** for selected Ethertypes and LLC/SNAP protocols

8.11.5.6 Reference section

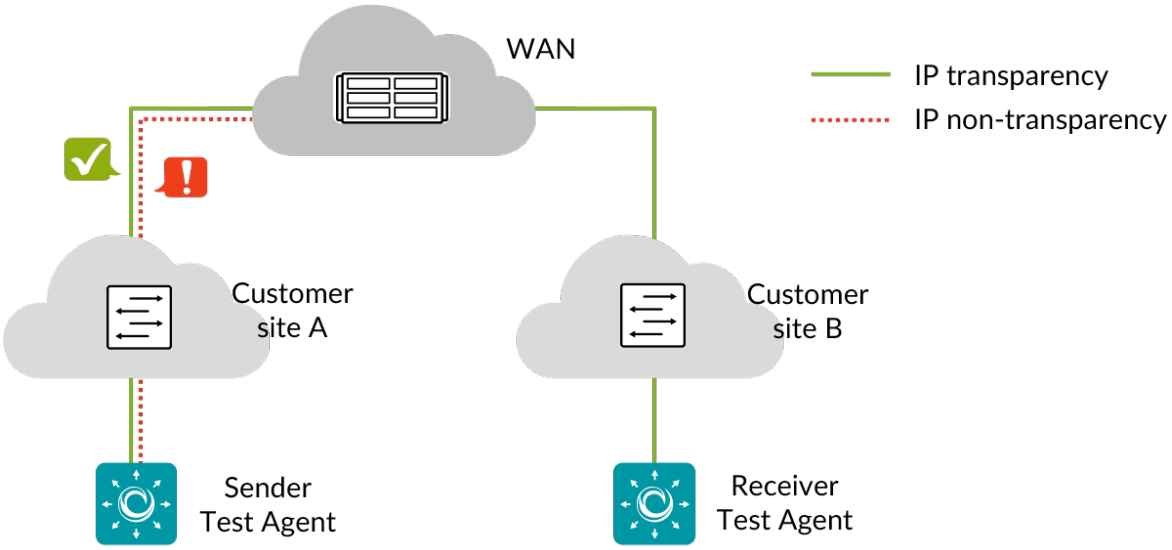
Ethertypes

Protocol	Ethertype
3COM Loop Detect	0×9003
3COM TCP-IP Sys	0×9002
3COM XNS Sys Mgmt	0×9001
AppleTalk	0×809b
AppleTalk ARP	0×80f3
CESoE	0×88d8
DEC LANBridge	0×8038
Frame Relay ARP	0×0808
GSMP	0×880c
IBM SNA Service	0×80d5
MPLS	0×8847
Novell 8137	0×8137
Novell 8138	0×8138
PPP	0×880b
PPPoE Discovery Stage	0×8863
PPPoE Session Stage	0×8864
Raw Frame Relay	0×6559
SNMP	0×814c

LLC/SNAP protocols

Protocol	Destination MAC	SAP	CTRL	OUI	PID
CDP	01:00:0c:cc:cc:cc	0×aa	0×03	0×0000c	0×2000
Cisco VTP	01:00:0c:cc:cc:cc	0×aa	0×03	0×0000c	0×2003
Cisco DTP	01:00:0c:cc:cc:cc	0×aa	0×03	0×0000c	0×2004
CGMP	01:00:0c:dd:dd:dd	0×aa	0×03	0×0000c	0×2001

8.11.6 L2 transparency – IP



This task verifies IP header integrity as well as IP multicast, checking in both cases that IP packets are not dropped in the network.

8.11.6.1 Test procedure

The test is divided into two parts.

IP header integrity

In this part of the test, 256 IP packets are sent with random source IP address 10.x.y.z, random destination IP address 10.xx.yy.zz, and a dummy payload. The ID, TTL, and TOS fields are set to the same value (in the range 0 … 255) for each packet.

IP multicast

In this part of the test, a number of multicast address ranges are tested. Note that the address 224.0.0.0 is excluded from the first range.

Multicast address range	No. of addresses in range
224.0.0.1 … 224.0.0.63	63
224.10.10.0 … 224.10.10.63	64
228.0.128.0 … 228.0.128.63	64
230.70.80.64 … 230.70.80.127	64
239.240.250.0 … 239.240.250.63	64

For each multicast address in a range, a packet is generated with correctly configured Ethernet and IP header. Each packet will carry the multicast address in the IP payload.

8.11.6.2 Fail criteria

IP header integrity

The test fails if any packet is dropped, or if the ID, TTL, or TOS fields do not match for some packet.

IP multicast

The test fails if any packet in a range is dropped, or if the destination addresses in the IP header and the IP payload do not match.

8.11.6.3 Limitations

This test can only run on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.6.4 Parameters

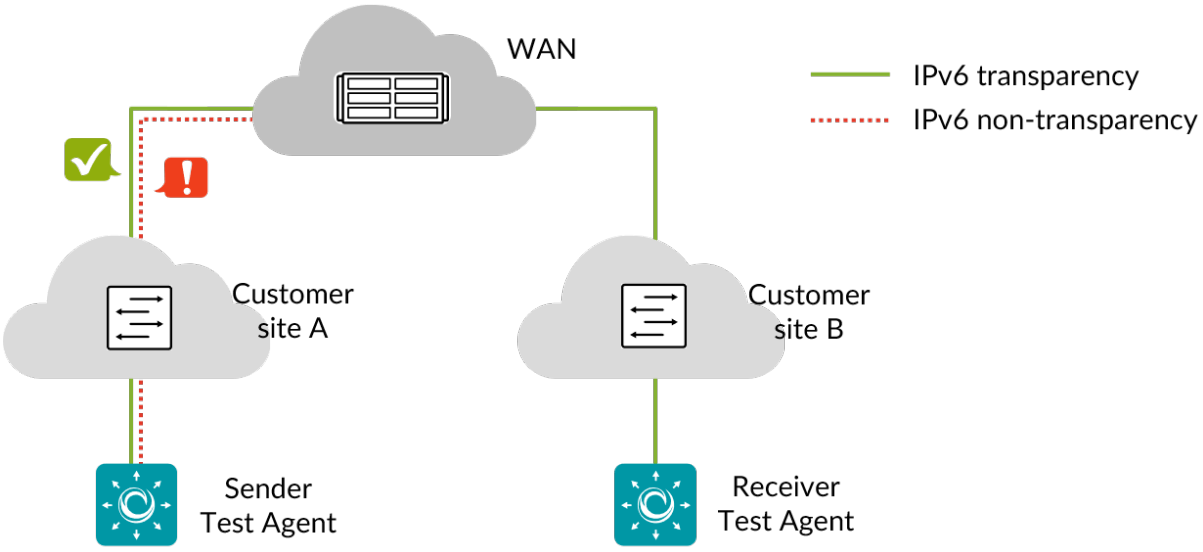
General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

8.11.6.5 Result metrics

- **Pass/fail** on IP header integrity and IP multicast

8.11.7 L2 transparency – IPv6



This task verifies IPv6 header integrity, i.e. checks that IPv6 packets are not dropped or blocked in the network.

This task also verifies transparency for a number of IPv6 protocols:

- Multicast Listener Discovery (MLD), Versions 1 and 2
- ICMPv6 with packet types
  - Router Solicitation
  - Router Advertisement
  - Neighbor Solicitation
  - Neighbor Advertisement
- ICMPv6 Echo
- DHCPv6 Solicit

The link is considered IPv6 transparent if the IPv6 packets go through. No verification is done of packet content.

8.11.7.1 Test procedure and fail criteria

The test is divided into two parts corresponding to the above description.

IPv6 header integrity

In this part of the test, 10 IPv6 packets are sent with dummy payload. The flow label, hop limit, traffic class fields, and UDP ports are set to the same value (in the range 0 … 255) for each packet.

The test fails if any packet is dropped, or if the flow label, hop limit, traffic class fields, or UDP ports do not match between sender and receiver.

IPv6 protocol transparency

In this part of the test, a number of IPv6 protocols are tested for transparency. For each protocol, one packet is generated with correctly configured Ethernet and IPv6 headers, and it is checked that the messages pass transparently. The protocols tested are:

Protocol	Messages
Multicast Listener Discovery Protocol, version 1 (MLD)	MLD query and report messages
Multicast Listener Discovery Protocol, version 2 (MLDv2)	MLD query and report messages
Neighbor Solicitation	Neighbor Solicitation messages
Neighbor Advertisement	Neighbor Advertisement messages
Router Solicitation	Router Solicitation messages
Router Advertisement	Router Advertisement messages
ICMPv6 Echo	ICMPv6 Echo messages
DHCPv6 Solicit	DHCPv6 Solicit messages

The test fails for a particular protocol if the message sent on that protocol does not pass transparently.

8.11.7.2 Limitations

This test can only be run on physical or VLAN interfaces (not bridges).

This test cannot be run through a routed (Layer 3) network.

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.7.3 Parameters

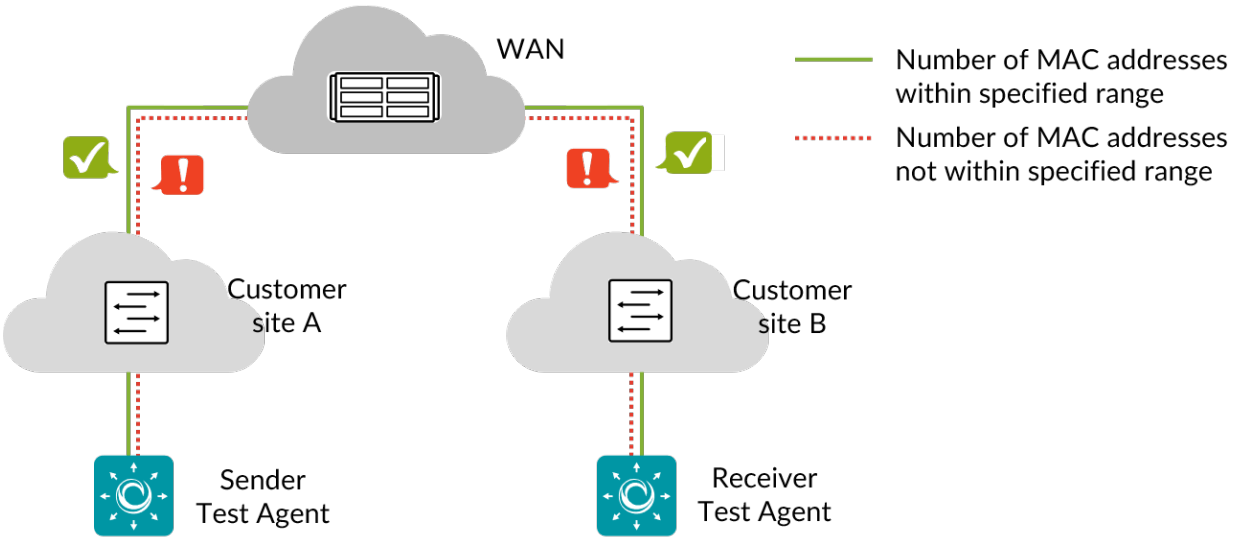
General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Expected outcome for <protocol>: For each protocol, select Pass, Drop, or Don't test. Default: Pass.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: "Don't wait", i.e. zero wait time.

8.11.7.4 Result metrics

- **Pass/fail** on IPv6 header integrity, overall and for each IPv6 protocol

8.11.8 L2 transparency – MAC address limit



This task checks:

- that a specified minimum number of MAC addresses are allowed from a customer port;
- that it is not possible to use more than a specified maximum number of MAC addresses.

8.11.8.1 Test procedure and fail criteria

If a minimum number of MAC addresses *min* has been specified, the sender generates traffic with *min* different source MAC addresses. For each MAC address, 10 frames are sent. The receiver measures how many addresses it received. The test fails if for any MAC address the packet loss is higher than the loss percentage allowed.

If a maximum number of MAC addresses *max* has been specified, the sender generates traffic with (*max* + 1) different MAC addresses. The test fails if the number of MAC addresses for which at least one packet is received is higher than *max* (i.e. equal to *max* + 1).

It is important to make sure that the switches/network devices are in a clean state when the test is started, so that they don't have any active MAC addresses in their tables at that point.

8.11.8.2 Limitations

This test can be run only on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.8.3 Parameters

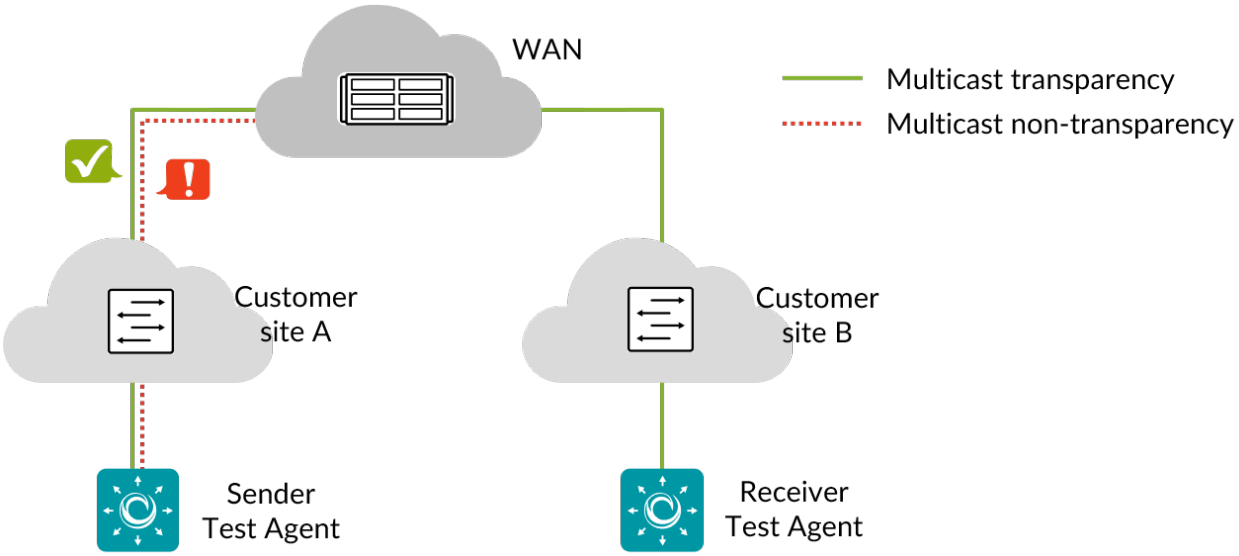
General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Minimum number of MAC addresses: The minimum number of source MAC addresses that must pass from sender to receiver. Min: 1. Max: 100.
- Maximum number of MAC addresses: The maximum number of source MAC addresses that are allowed to pass from sender to receiver. Min: 1. Max: 100. No default.
- Percent of loss allowed (%): Maximum packet loss percentage allowed for a given MAC address. Min: 0%. Max: 99%. Default: 0%.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

8.11.8.4 Result metrics

- **Pass/fail** on minimum number of MAC addresses
- **Pass/fail** on maximum number of MAC addresses

8.11.9 L2 transparency – Multicast



This task verifies transparency for Layer 2 multicast protocols, i.e. checks that the multicast packets are not dropped in the network. The tested protocols are: Spanning Test Protocol, a set of general multicast/broadcast protocols, and MPLS multicast.

8.11.9.1 Test procedure

Five Ethernet frames with dummy payload are sent on each protocol.

For each protocol, the outcome is compared to the expected outcome (Pass or Drop) set for it.

8.11.9.2 Fail criteria

- If the expected outcome for a protocol is set to Pass, the test fails for that protocol if less than four frames are received with the specified Ethertype and destination MAC.
- If the expected outcome for a protocol is set to Drop, the test fails for that protocol if at least one frame is received with the specified Ethertype and destination MAC.

Exception: For MPLS multicast, the receiver does not check the destination MAC address; it filters only for the source MAC address and Ethertype.

8.11.9.3 Protocols tested

Protocol	Destination MAC	Ethertype
STP	01:80:c2:00:00:00	LLC/SNAP
ARP	ff:ff:ff:ff:ff:ff	0×0806
RARP	ff:ff:ff:ff:ff:ff	0×8035
LLDP	01:80:c2:00:00:0e	0×88cc
Ethernet Configuration Test Protocol	cf:00:00:00:00:00	0×9000
IP global broadcast	ff:ff:ff:ff:ff:ff	0×0800
IP local broadcast	ff:ff:ff:ff:ff:ff	0×0800
MPLS multicast addresses	01:00:5e:80:00:01	0×8848 (see Note 1)
MPLS multicast addresses	01:00:5e:8d:dd:dd	0×8848
MPLS multicast addresses	01:00:5e:8f:ff:01	0×8848

Note 1: Ethertype 0×8848, formerly known as the “MPLS multicast codepoint”, is to be used only when an MPLS packet whose top label is upstream-assigned is carried in a multicast Ethernet frame.

Note 2: Ethernet frames with a value of 1 in the least significant bit of the first octet of the destination address are treated as multicast frames and are typically flooded to all points on the network. While frames with ones in all bits of the destination address (ff:ff:ff:ff:ff:ff) are sometimes referred to as broadcasts, Ethernet network equipment generally does not distinguish between multicast and broadcast frames.

8.11.9.4 Limitations

This test can only run on physical or VLAN interfaces (not bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.9.5 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

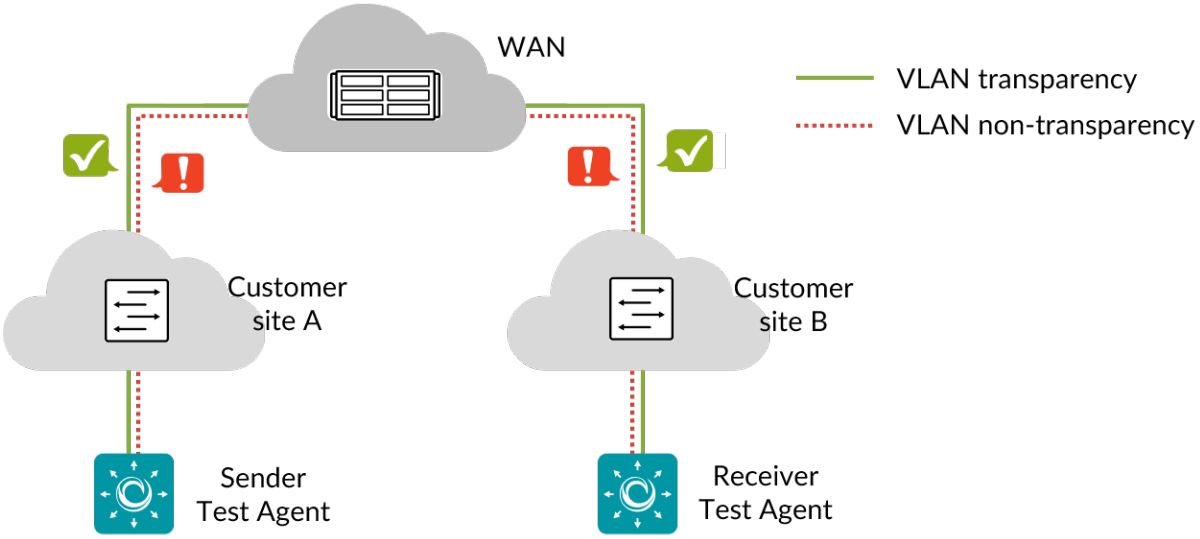
Advanced

- Expected outcome for <protocol>: For each protocol, select Pass, Drop, or Don’t test. Default: Pass.

8.11.9.6 Result metrics

- **Pass/fail** on STP
- **Pass/fail** on General multicast/broadcast protocols
- **Pass/fail** on MPLS multicast

8.11.10 L2 transparency – VLAN



This task verifies VLAN transparency, that is, that packets with user-specified VLAN tag, *VLAN priority (PCP)* (page 516), and *DSCP* (page 511) are not modified by the network.

8.11.10.1 Test procedure and fail criteria

VLAN transparency

In this part of the test, five frames are generated for each of the VLANs in the table below. The frames are correctly created Ethernet frames with a VLAN header and an IP payload.

The test fails if not all frames are received, or if any of the VLAN ID, p-bits, or IP DSCP is modified.

VLAN ID	p-bits	DSCP
0	0	0
12	4	20
99	3	15
150	6	30
517	5	25
1033	1	5
1434	2	10
1800	0	0
2047	7	35
4094	6	30
4095	7	35

Q-in-Q transparency

In this part of the test, Q-in-Q is tested with Ethertypes 0×8100, 0×88a8, 0×9100, and 0×9200 for the outer VLAN. For each Ethertype, a number of outer and inner VLAN ID combinations are used; for each such combination, five frames are generated. The frames are correctly created Ethernet frames with a Q-in-Q header and an IP payload.

The test fails if not all frames are received, or if any of the VLAN ID (inner/outer), p-bits, or IP DSCP is modified.

Outer VLAN ID	Outer p-bits	Inner VLAN ID	Inner p-bits	DSCP
0	0	4095	7	0
13	5	4083	2	25
100	4	3995	3	20
151	7	3944	0	35
518	6	3577	1	30
1034	2	3061	5	10
1435	3	2660	4	15
1801	1	2294	6	5
2048	0	2047	7	0
4095	6	1	7	35

8.11.10.2 Limitations

This test can only run on physical interfaces (not on VLANs or bridges).

Regarding use of the Test Agent management interface for this test, see *here* (page 377).

8.11.10.3 Parameters

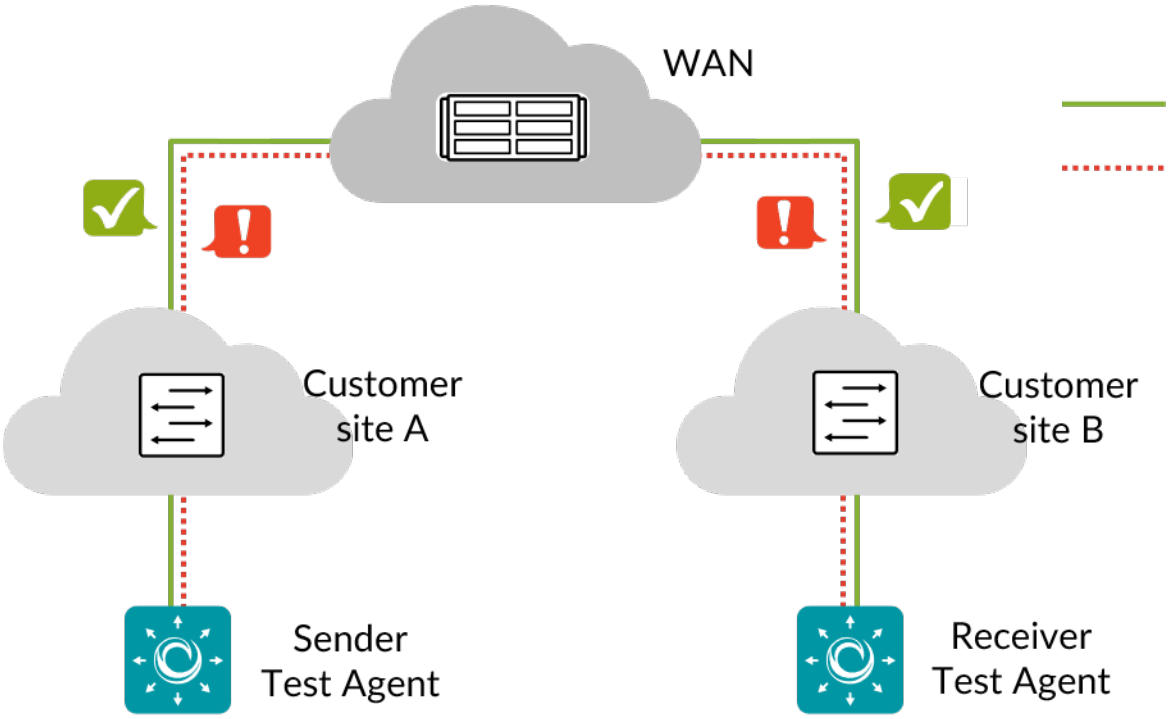
General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

8.11.10.4 Result metrics

- **Pass/fail** on VLAN transparency
- **Pass/fail** on Q-in-Q transparency

8.11.11 DSCP remapping



This task verifies the expected remapping of *Differentiated Services Code Point* (page 511) values between two points in your network.

8.11.11.1 Test procedure

UDP packets are sent from the sender Test Agent to the receiver Test Agent with different DSCP values, and Routing Active Testing checks if they are mapped correctly by the network.

8.11.11.2 Fail criteria

The test fails if any received DSCP differs from the expected DSCP value.

8.11.11.3 Limitations

IPv6 is not supported for this task.

8.11.11.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

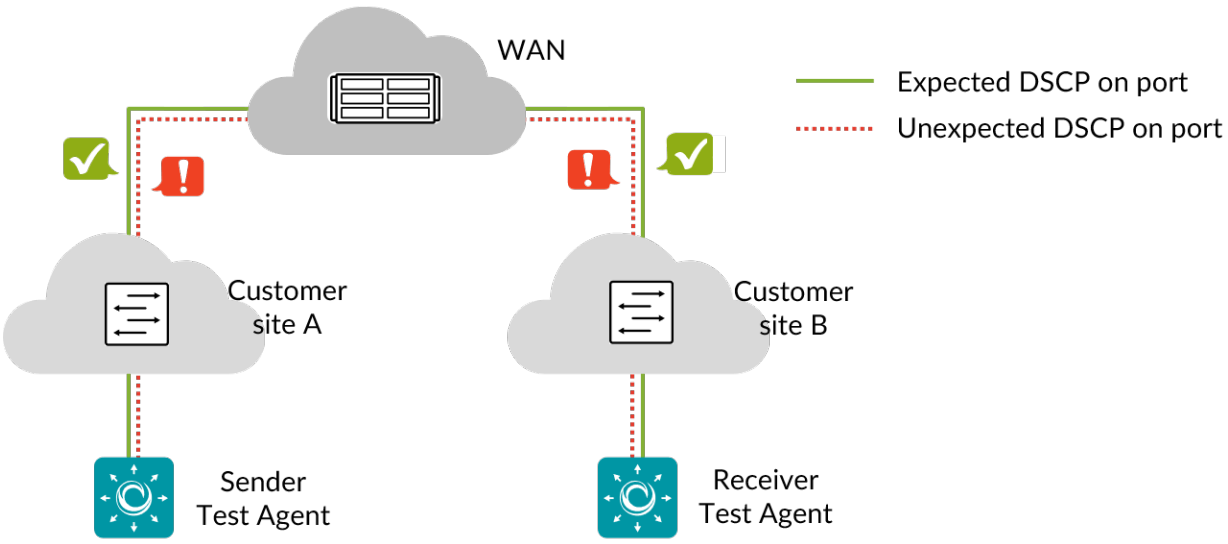
Advanced

- Expected result for DSCP <…>: For each DSCP, state the expected result of the DSCP remapping. Default: The default for all DSCP values is no change.

8.11.11.5 Result metrics

- **Pass/fail** for each tested DSCP

8.11.12 Layer 4 destination port DSCP remapping



This task verifies the expected *DSCP (Differentiated Services Code Point)* (page 511) remapping between two points in your network, with specific UDP or TCP destination ports indicated.

8.11.12.1 Test procedure

UDP or TCP packets are sent from the sender Test Agent to the receiver Test Agent with a specified destination port number, and it is checked how the DSCP is mapped by the network. This is repeated for each port in the specified port ranges.

8.11.12.2 Fail criteria

The test fails for a port range if the DSCP mapping for any port in that range differs from what is expected.

8.11.12.3 Limitations

IPv6 is not supported for this task.

8.11.12.4 Parameters

General

- Sender: The sender Test Agent interface.
- Receiver: The receiver Test Agent interface.
- Protocol: The protocol to use: UDP or TCP. Default: UDP.
- Port ranges: Port ranges, separated by commas (,). Default: “80-90” *(i.e. a single range)*.
- Sent DSCP: Sent DSCP or IP Precedence value. Default: “0 / IPP 0”.
- Expected DSCP: Expected received DSCP or IP Precedence value. Default: “0 / IPP 0”.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: “Don’t wait”, i.e. zero wait time.

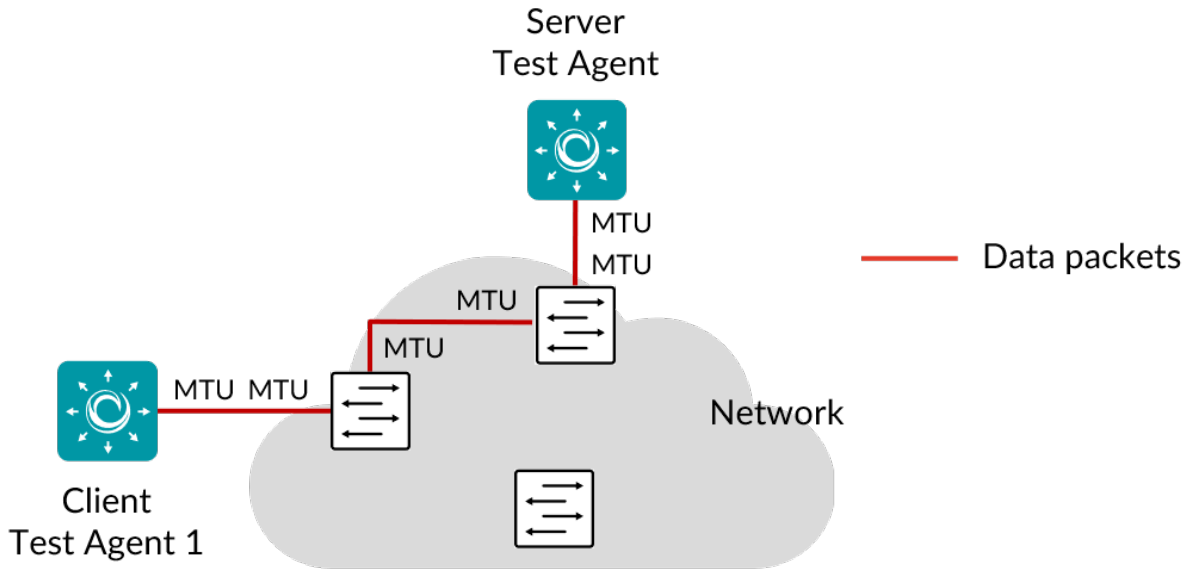
8.11.12.5 Result metrics

- **Pass/fail** for each tested port range

8.11.12.6 Remark

This test can alternatively be used simply to find out which ports are open between the two points in the network (by testing the full range of existing port numbers).

8.11.13 Path MTU discovery



This task determines the path MTU (Maximum Transmission Unit) between two Test Agents. It allows you to find out whether the MTU values configured in network elements are appropriate. This in turn is important in order to avoid packet fragmentation.

8.11.13.1 Test procedure and fail criteria

The algorithm starts by sending a UDP frame of size equal to the smaller of the MTUs configured on the server and client Test Agent interfaces (as described *here* (page 168)). If this frame is not received, the algorithm tries smaller frames according to a predetermined search pattern in order to determine the maximum acceptable frame size on the connection, that is, the path MTU.

The test fails if the path MTU is found to be smaller than Minimum MTU. If this is the case, you need to increase the MTU setting in network elements in the path, if possible.

8.11.13.2 Limitations

IPv6 is not supported for this task.

8.11.13.3 Parameters

General

- Server: The server Test Agent interface.
- Client: The client Test Agent interface.
- Direction: The direction of the test traffic: from client to server, or vice versa.
- Minimum MTU: The smallest acceptable MTU value. This cannot be set higher than the smaller of the MTUs configured on the Test Agent interfaces, or the test will end with an error. Default: 1500 bytes.
- Server port: The UDP server port to use on the client. Range: 1 … 65535. Default: 7000.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: "Don't wait", i.e. zero wait time.

8.11.13.4 Result metrics

- Calculated **path MTU** with **pass/fail** indication

8.12 Reflector-based testing

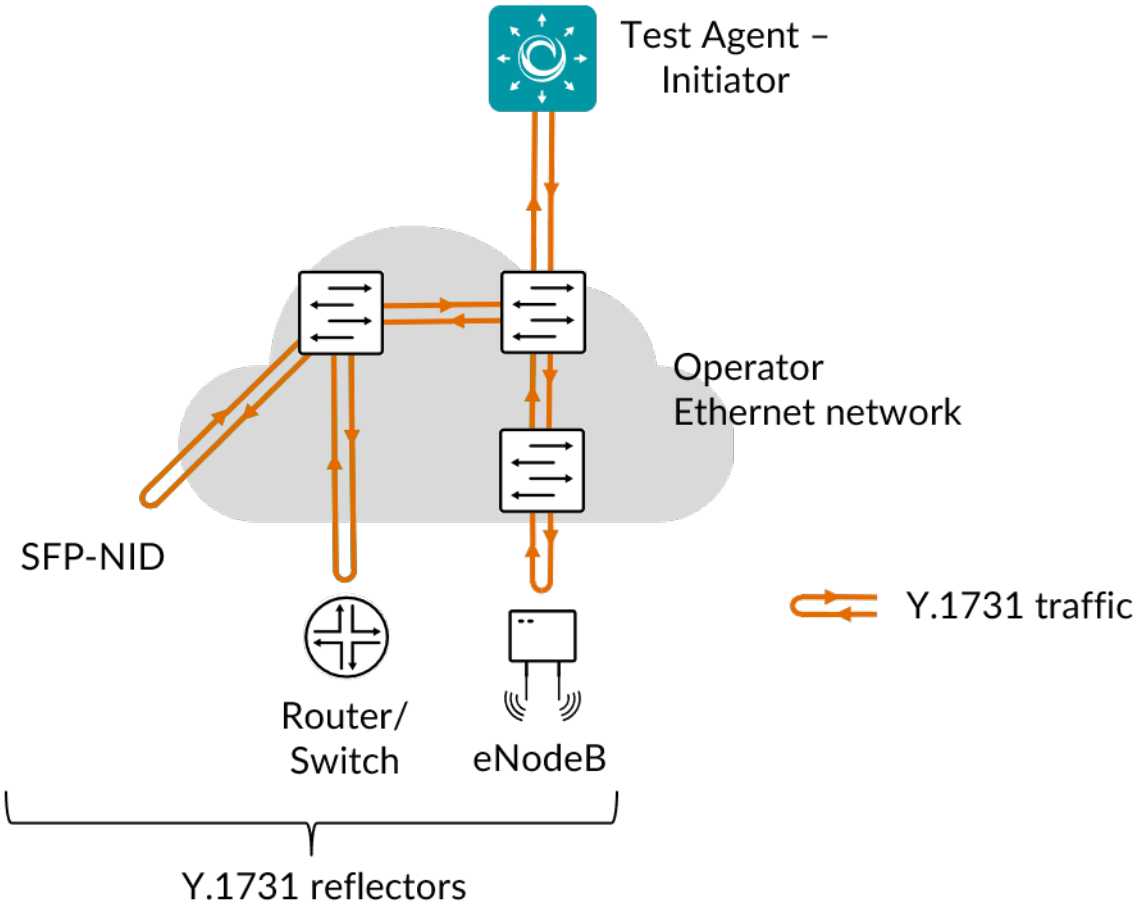
8.12.1 Introduction to Y.1731 testing

Routing Active Testing supports the following parts of ► ITU-T Recommendation G.8013/Y.1731:

- Fault management:
  - Ethernet loopback (ETH-LB)
- Performance monitoring:
  - Frame delay measurement (ETH-DM)
  - Synthetic loss measurement (ETH-SLM)

These are part of the Y.1731 OAM functions for Ethernet-based networks.

The ETH-LB function is also found in the IEEE 802.1ag (CFM, Connectivity Fault Management) standard.



Y.1731 defines a Layer 2 protocol, and therefore it requires Layer 2 connectivity between the Test Agent and the device you are testing towards. You can then send traffic from the Test Agent towards a Y.1731-capable device and have the traffic reflected back to the Test Agent, which measures loss, delay, and delay variation (jitter).

Terminology and definitions used on the Y.1731 pages of the documentation:

- **ME:** Maintenance Entity
- **MEG:** ME Group
- **MEL:** MEG Level
- **MEP:** MEG End Point
- **MIP:** MEG Intermediate Point

8.12.1.1 MEG level

MEG levels range from 0 to 7.

In the case where MEGs are nested, the OAM flow of each MEG has to be clearly identifiable and separable from the OAM flows of the other MEGs. In cases where the OAM flows are not distinguishable by the ETH layer encapsulation itself, the MEG level in the OAM frame distinguishes between the OAM flows of nested MEGs.

Eight MEG levels are available to accommodate different network deployment scenarios.

When customer, provider, and operator data path flows are not distinguishable based on the ETH layer encapsulations, the eight MEG levels can be shared amongst them to distinguish between OAM frames belonging to nested MEGs of

customers, providers, and operators. The default MEG level assignment amongst the customer, provider, and operator roles is as follows:

- *Operator role:* 0 … 2
- *Provider role:* 3 … 4
- *Customer role:* 5 … 7

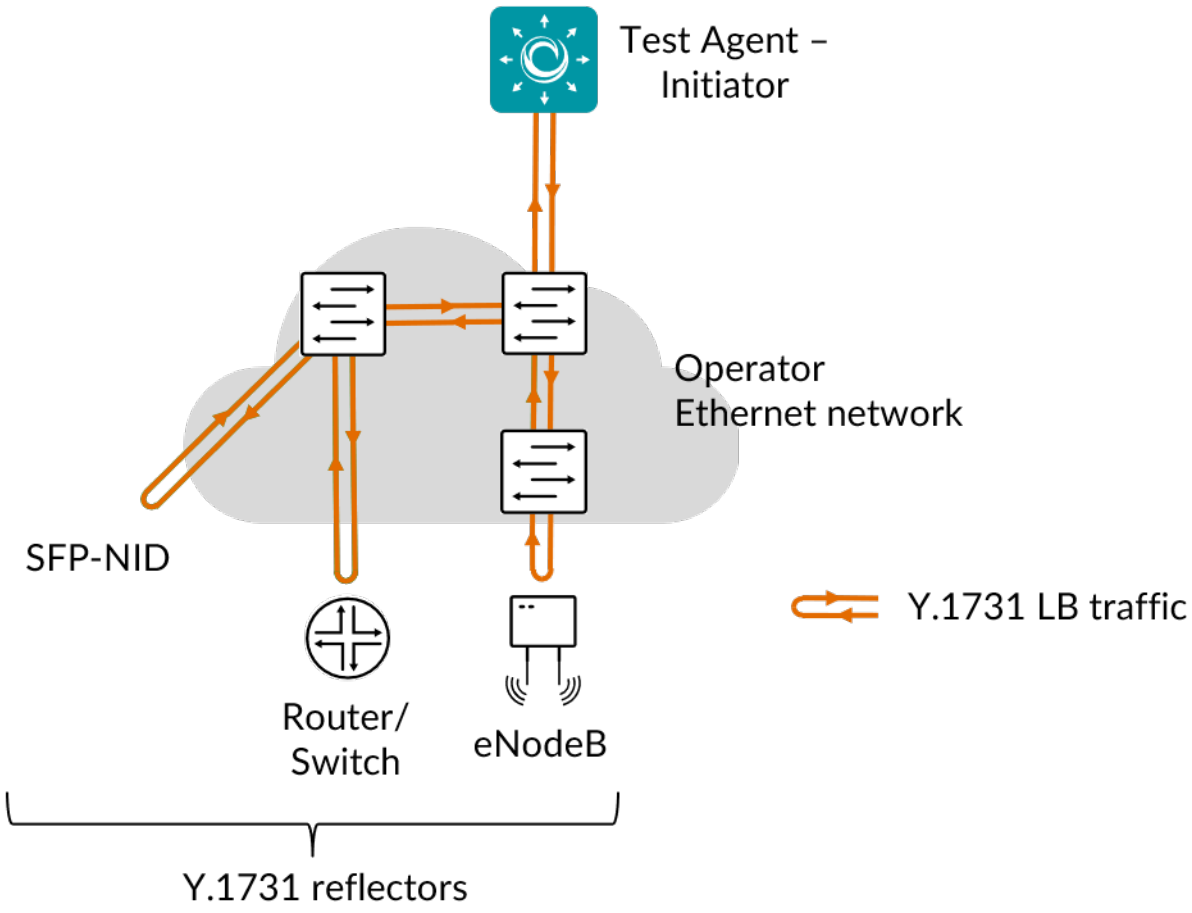
8.12.1.2 Related topics

- *Y.1731 Ethernet loopback (ETH-LB)* (page 404)
- *Y.1731 delay measurement (ETH-DM)* (page 407)
- *Y.1731 synthetic loss measurement (ETH-SLM)* (page 410)

8.12.2 Y.1731 Ethernet loopback (ETH-LB)

This task lets you send traffic from a Test Agent towards a Y.1731 ETH-LB capable device and have the traffic reflected back to the Test Agent, which measures two-way delay, loss, and delay variation (jitter). ETH-LB can be used for the following applications:

- Verifying bidirectional connectivity between a MEP and a MIP or between two MEPs.
- Performing a bidirectional in-service or out-of-service diagnostics test between a pair of peer MEPs. This includes verifying bandwidth throughput and detecting bit errors.



Y.1731 is a Layer 2 protocol, and Layer 2 connectivity is therefore required between the Test Agent and the device you are testing towards.

This task works with both IPv4 and IPv6.

8.12.2.1 Prerequisites

To run a Y.1731 ETH-LB measurement you need to have at least one Test Agent installed as well as one or several Y.1731-enabled devices in your network. See the installation guides found *here* (page 70) for instructions on how to deploy a new Test Agent. Regarding enabling of Y.1731 on your devices, consult your equipment vendor.

You also need to prepare a Y.1731 MEP list in Routing Active Testing, as explained on *this page* (page 33).

Then add an ETH-LB task to your test or monitor and fill in the mandatory parameters below:

8.12.2.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces that will act as initiating MEPs.
- MEPs (MAC addresses): List of reflector MEPs containing MAC addresses.
- Rate (Mbit/s): Rate at which clients will send frames in Mbit/s.
- Rate (packets/s): Rate at which the clients will send frames in packets/s. Min: 2 packets/s. Max: 1,000,000 packets/s.
- Frame size (bytes): Size of Layer 2 Ethernet frame. See *this page* (page 512). Min: 64 bytes. Max: 9018 bytes. Default: 1518 bytes.

Once Frame size is defined, changing one Rate parameter will cause the other to adjust automatically to agree with it.

Thresholds for errored seconds (ES)

- Loss (%): Packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Default: 0%.
- Delay (ms): Two-way delay threshold for triggering an errored second. If the delay between server and clients exceeds this value during one second, an ES will be indicated. Min: 1 ms. No default.
- Delay variation (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and clients exceeds this value during one second, an ES will be indicated. Min: 1 ms. No default.

Thresholds for severely errored seconds (SES)

- Loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). If the loss exceeds this value during one second, an SES will be indicated. Min: 0%. No default.
- Delay (ms): Two-way delay threshold for triggering a severely errored second. If the delay between server and clients exceeds this value during one second, an SES will be indicated. Min: 1 ms. No default.
- Delay variation (ms): Delay variation (jitter) threshold for triggering a severely errored second. If the delay variation between server and clients exceeds this value during one second, an SES will be indicated. Min: 1 ms. No default.

Advanced

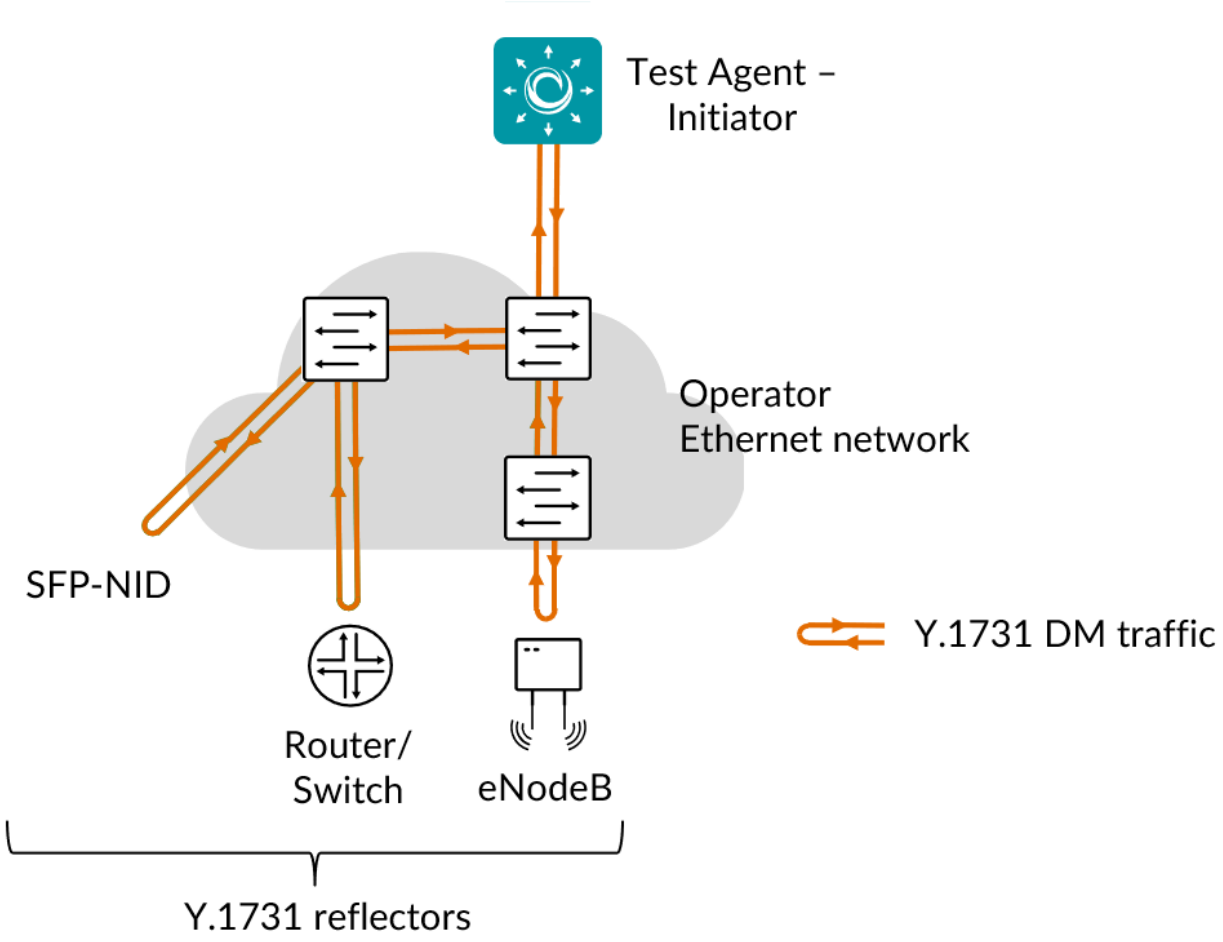
- VLAN priority (PCP): Priority Code Point to use in the VLAN header. See *this page* (page 516). Default: 0.

8.12.2.3 Result metrics

- **Rate (Mbit/s):** Actual rate at which clients sent Y.1731 frames.
- **Sent (packets):** Number of sent packets.
- **Received (packets):** Number of received packets.
- **Lost (packets):** Number of lost packets.
- **Loss (%):** Packet loss ratio.
- **Misorder (packets):** Number of misordered packets.
- **Min delay (ms):** Minimum two-way delay.
- **Average delay (ms):** Average two-way delay.
- **Max delay (ms):** Maximum two-way delay.
- **Delay variation (ms):** Delay variation (jitter).
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES loss (%):** Errored second percentage for packet loss.
- **ES delay (%):** Errored second percentage for delay.
- **ES delay variation (%):** Errored second percentage for delay variation.
- **SES (%):** Aggregated severely errored second (SES) percentage, taking into account all types of error.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.12.3 Y.1731 delay measurement (ETH-DM)

This task lets you send traffic from a Test Agent towards a Y.1731 ETH-DM capable device. Frame delay and frame delay variation measurements are collected by sending periodic frames with ETH-DM information to the peer MEP and receiving frames with ETH-DM information from the peer MEP during a proactive measurement session and/or diagnostic interval. Each MEP may perform frame delay and frame delay variation measurement.



Y.1731 is a Layer 2 protocol, and Layer 2 connectivity is therefore required between the Test Agent and the device you are testing towards.

This task works with both IPv4 and IPv6.

8.12.3.1 Prerequisites

To run a Y.1731 ETH-DM measurement, you need to have at least one Test Agent installed as well as one or several Y.1731-enabled devices in your network. See the installation guides found *here* (page 70) for instructions on how to deploy a new Test Agent. Regarding enabling of Y.1731 on your devices, consult your equipment vendor.

You also need to prepare a Y.1731 MEP list in Routing Active Testing, as explained on *this page* (page 33).

Then add an ETH-DM task to your test or monitor and fill in the mandatory parameters below:

8.12.3.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces that will act as initiating MEPs.
- MEPs (MAC addresses): List of reflector MEPs containing MAC addresses.
- Rate (Mbit/s): Rate at which clients will send frames in Mbit/s.
- Rate (packets/s): Rate at which clients will send frames in packets/s. Min: 2 packets/s. Max: 1,000,000 packets/s.
- Frame size (bytes): Size of Layer 2 Ethernet frame. See *this page* (page 512). Min: 64 bytes. Max: 9018 bytes. Default: 1518 bytes.

Once Frame size is defined, changing one Rate parameter will cause the other to adjust automatically to agree with it.

Thresholds for errored seconds (ES)

- Delay (ms): One-way delay threshold for triggering an errored second. If the delay between server and clients exceeds this value during one second, an ES will be indicated. Min: 1 ms. No default.
- Delay variation (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and clients exceeds this value during one second, an ES will be indicated. Min: 1 ms. No default.

Thresholds for severely errored seconds (SES)

- Delay (ms): One-way delay threshold for triggering a *severely errored second* (page 477). If the delay between server and clients exceeds this value during one second, an SES will be indicated. Min: 1 ms. No default.
- Delay variation (ms): Delay variation (jitter) threshold for triggering a severely errored second. If the delay variation between server and clients exceeds this value during one second, an SES will be indicated. Min: 1 ms. No default.

Advanced

- VLAN priority (PCP): Priority Code Point to use in the VLAN header. See *this page* (page 516). Default: 0.

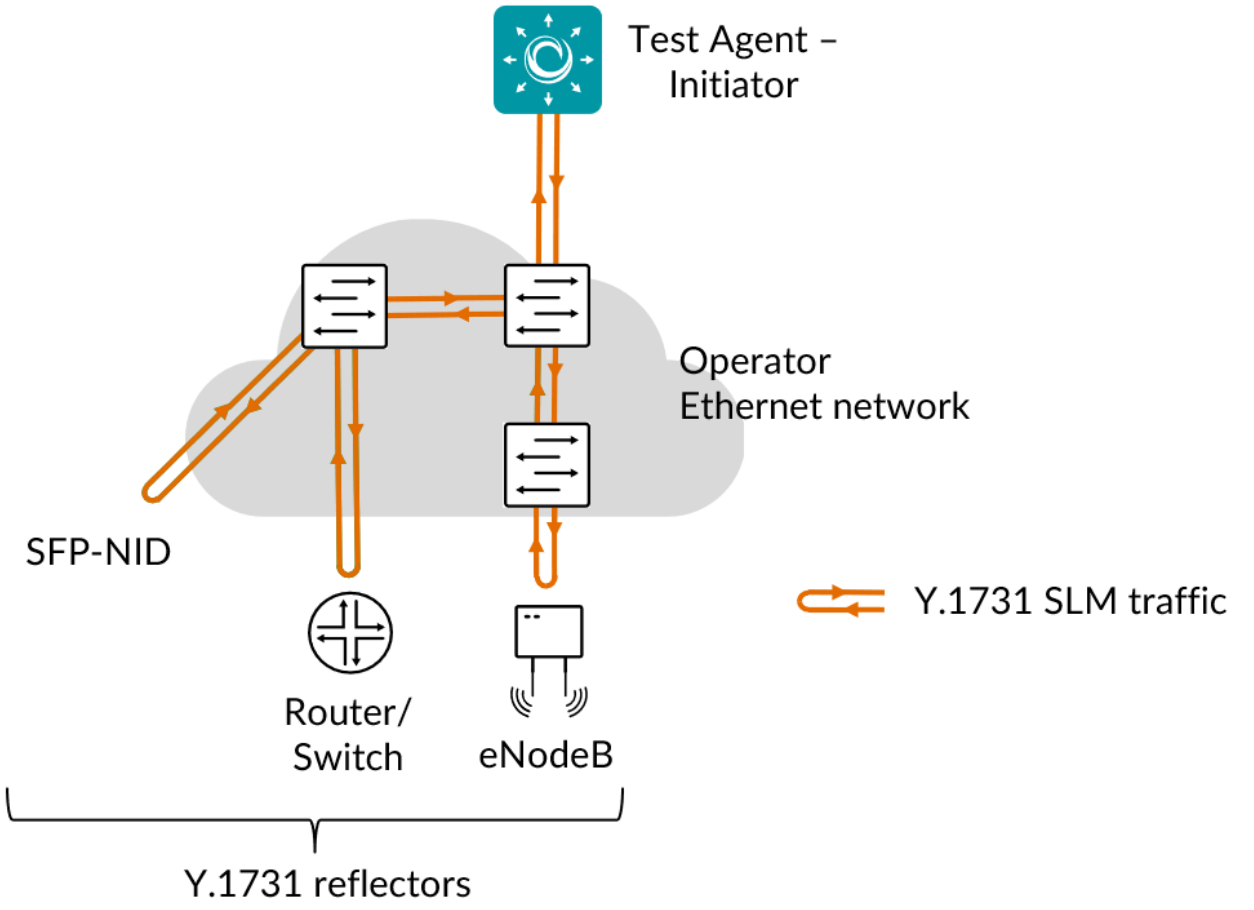
8.12.3.3 Result metrics

- **Rate (Mbit/s):** Actual rate at which clients sent Y.1731 frames.
- **Sent (packets):** Number of sent packets.
- **Received (packets):** Number of received packets.
- **Min delay near (ms):** Minimum one-way delay, near end.
- **Average delay near (ms):** Average one-way delay, near end.
- **Max delay near (ms):** Maximum one-way delay, near end.
- **Delay variation near (ms):** Delay variation, near end.
- **Min delay far (ms):** Minimum one-way delay, far end.
- **Average delay far (ms):** Average one-way delay, far end.
- **Max delay far (ms):** Maximum one-way delay, far end.
- **Delay variation far (ms):** Delay variation, far end.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES delay (%):** Errored second percentage for delay.
- **ES delay variation (%):** Errored second percentage for delay variation.
- **SES (%):** Aggregated severely errored second (SES) percentage, taking into account all types of error.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.12.4 Y.1731 synthetic loss measurement (ETH-SLM)

This task provides a mechanism for measuring frame loss using synthetic frames, rather than inspecting real customer data traffic.

ETH-SLM sends frames with ETH-SLM information to one or multiple peer MEPs and receives similar frames from the peer MEPs. Each MEP then performs frame loss measurements based on the counters added to the frames. Since a bidirectional service is defined as unavailable if either of the two directions is declared unavailable, ETH-SLM must facilitate each MEP to perform near-end and far-end synthetic frame loss measurements.



Y.1731 ETH-SLM is a Layer 2 protocol, and Layer 2 connectivity is therefore required between the Test Agent and the device you are testing towards.

This task works with both IPv4 and IPv6.

8.12.4.1 Prerequisites

To run a Y.1731 ETH-SLM measurement you need to have at least one Test Agent installed as well as one or several Y.1731-enabled devices in your network. See the installation guides found *here* (page 70) for instructions on how to deploy a new Test Agent. Regarding enabling of Y.1731 on your equipment, consult your equipment vendor.

You also need to prepare a Y.1731 MEP list in Routing Active Testing, as explained on *this page* (page 33).

Then add an ETH-SLM task to your test or monitor and fill in the mandatory parameters below:

8.12.4.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agent interfaces that will act as initiating MEPs.
- MEPs (MAC addresses): List of reflector MEPs containing MAC addresses.
- Rate (Mbit/s): Rate at which clients will send frames in Mbit/s.
- Rate (packets/s): Rate at which clients will send frames in packets/s. Min: 2 packets/s. Max: 1,000,000 packets/s.
- Frame size (bytes): Size of Layer 2 Ethernet frame. See *this page* (page 512). Min: 64 bytes. Max: 9018 bytes. Default: 1518 bytes.
- Source MEP ID: Source MEP ID which the Test Agent will use in the SLM request. Range: 0 … 8191. Default: 0.

Once Frame size is defined, changing one Rate parameter will cause the other to adjust automatically to agree with it.

Thresholds for errored seconds (ES)

- Loss (%): Packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Default: 0%.

Thresholds for severely errored seconds (SES)

- Loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). If the loss exceeds this value during one second, an SES will be indicated. Min: 0%. No default.

Advanced

- VLAN priority (PCP): Priority Code Point to use in the VLAN header. See *this page* (page 516). Default: 0.

8.12.4.3 Result metrics

Note: "Far-end" means towards the reflector MEP, and "near-end" means back towards the initiating MEP.

- **Rate (Mbit/s):** Actual rate at which clients sent Y.1731 frames.
- **Sent (packets):** Number of sent packets.
- **Received (packets):** Number of received packets.
- **Lost near (packets):** Number of lost packets, near-end.

- **Loss near (%):** Packet loss ratio, near-end.
- **Lost far (packets):** Number of lost packets, far-end.
- **Loss far (%):** Packet loss ratio, far-end.
- **ES (%):** Aggregated errored second (ES) percentage.
- **ES loss (%):** Errored second percentage for packet loss.
- **SES (%):** Aggregated severely errored second (SES) percentage.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

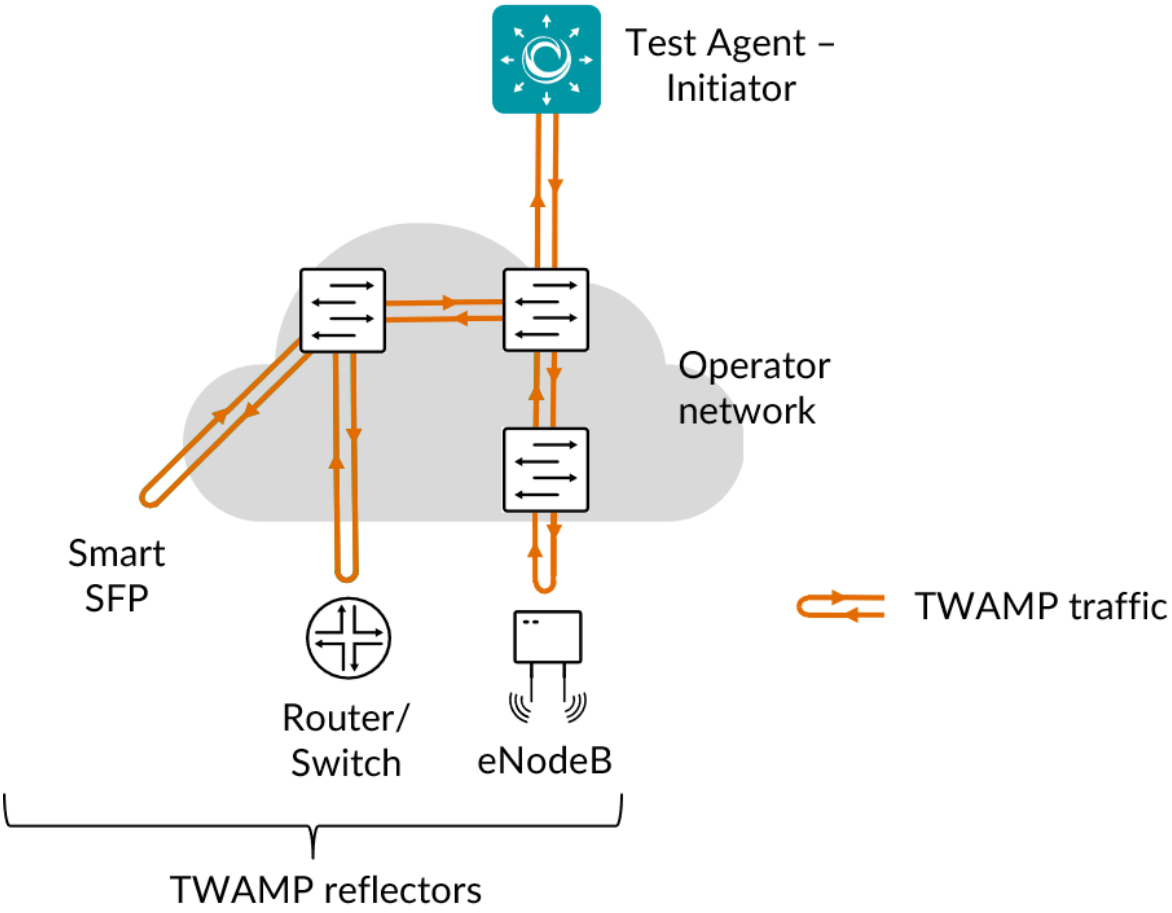
8.12.5 Introduction to TWAMP testing

Routing Active Testing supports the use of TWAMP and TWAMP Light for measuring two-way (and in part also one-way) loss and delay.

TWAMP is short for "Two-way Active Measurement Protocol" and is defined in ► IETF RFC 5357. TWAMP is based on OWAMP (One-way Measurement Protocol, ► IETF RFC 4656), to which it adds two-way measurement capabilities. Since TWAMP is a Layer 3 protocol, Layer 3 connectivity is required between the Test Agent and the target device.

The difference between TWAMP Light (defined in Appendix I of RFC 5357) and "regular" TWAMP is that the Light version does not require support for the TWAMP control protocol, which performs a handshake between initiator and reflector.

When used for TWAMP testing the Test Agent is typically placed in the core part of the network, or in the data center, and initiates UDP flows towards TWAMP-capable routers or other devices. These reflect the flows back to the Test Agent, which collects measurements and calculates various network KPIs.



The scenario just described is handled by the task *TWAMP/TWAMP Light* (page 416). The main benefit of this setup is that you can activation test, monitor, and troubleshoot your network end-to-end without the need for a dedicated test device at the customer site. This saves time and money. The limitation compared to using a Test Agent at both ends is that the testing capabilities are more restricted.

Although TWAMP is a two-way measurement protocol, it is still possible to measure one-way packet loss, that is, separate loss values for the forward and backward directions. This is possible since the reflector places its own sequence number in the packet. In fact, the loss values are always one-way, and the loss threshold always refers to one-way loss.

The formula used to calculate forward and backward packet loss is aligned with the ► ITU-T G.8013/Y.1731 standard. This has the consequence that duplicates and misorderings can give rise to negative loss values.

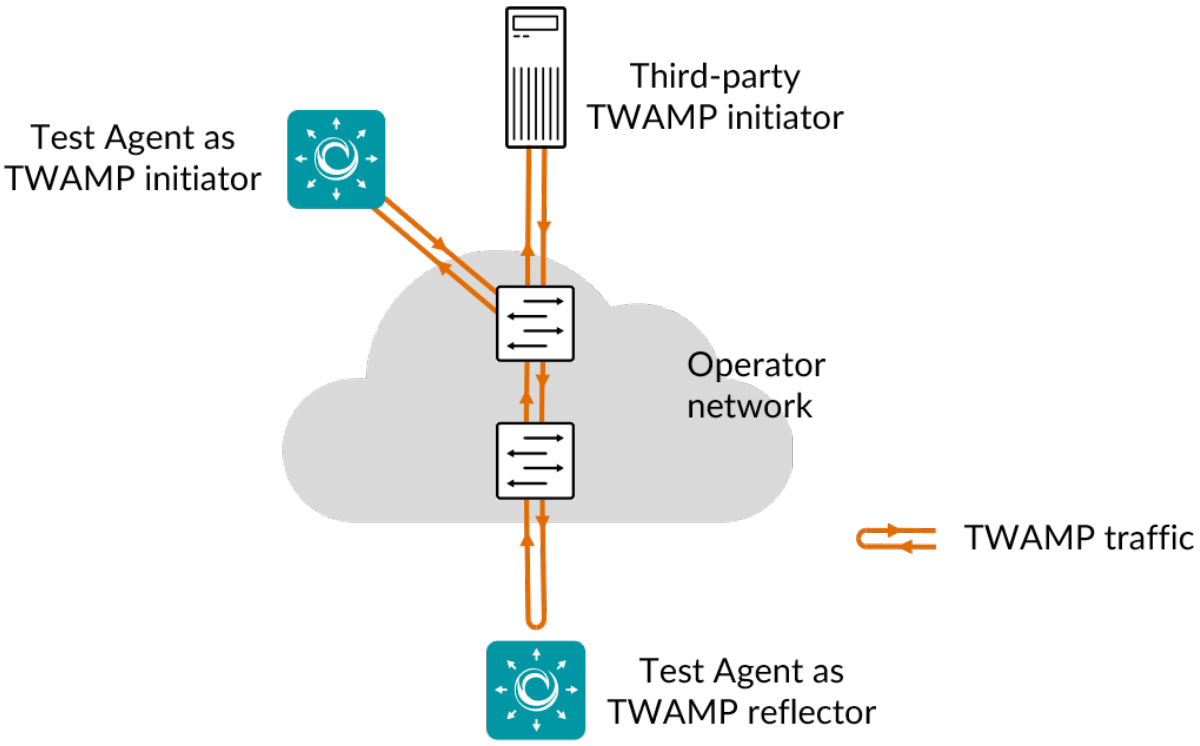
Thanks to the reflector timestamps it is also possible to measure one-way delay and delay variation. This is enabled by setting the Time sync parameter to Yes. However, for this measurement the sender Test Agent and the reflector must have their clocks synchronized. Round-trip delay is measured in this case, too; however, when time sync is enabled, the delay thresholds are applied to one-way delay values rather than to round-trip delay.

The default timestamping of TWAMP packets is software-based. Alternatively, hardware timestamping can be used for higher accuracy if your NIC supports it.

8.12.5.1 Test Agents as TWAMP reflectors

Test Agents can act not only as TWAMP senders but also as TWAMP reflectors. This is an option in the *TWAMP/TWAMP Light* (page 416) task. Further, a distinct task type called *TWAMP Reflector* (page 424) is provided where Test Agents reflect TWAMP traffic initiated by a third-party device.

Test Agent TWAMP reflectors support the TWAMP Light protocol only.



To illustrate the versatility of this feature it is instructive to give a summary of possible use cases:

"Normal" use case

A Test Agent TWAMP reflector is started using the *TWAMP/TWAMP Light* (page 416) task. Test Agents act as both TWAMP senders and TWAMP reflectors. External TWAMP reflectors can also be included in this setup.

All Test Agent TWAMP reflectors are forced to use the same port. If you want to differentiate port usage, follow the section *TWAMP reflectors on different ports* (page 416) below.

External TWAMP senders

To use third-party devices as TWAMP senders rather than Test Agents, use the *TWAMP Reflector* (page 424) task and configure the external senders to send traffic towards the Test Agent acting as reflector.

TWAMP reflectors on different ports

If you want to run multiple Test Agent TWAMP reflectors but with each one using a different port, you can start multiple *TWAMP Reflector* (page 424) tasks, each with a different port selected. Add all of these reflectors to the TWAMP inventory, and then direct Test Agent TWAMP senders towards them using the *TWAMP/TWAMP Light* (page 416) task.

8.12.5.2 Collecting TWAMP data from Junos routers

Test Agents can also collect measurements from TWAMP sessions running on Junos devices. This is covered *here* (page 425).

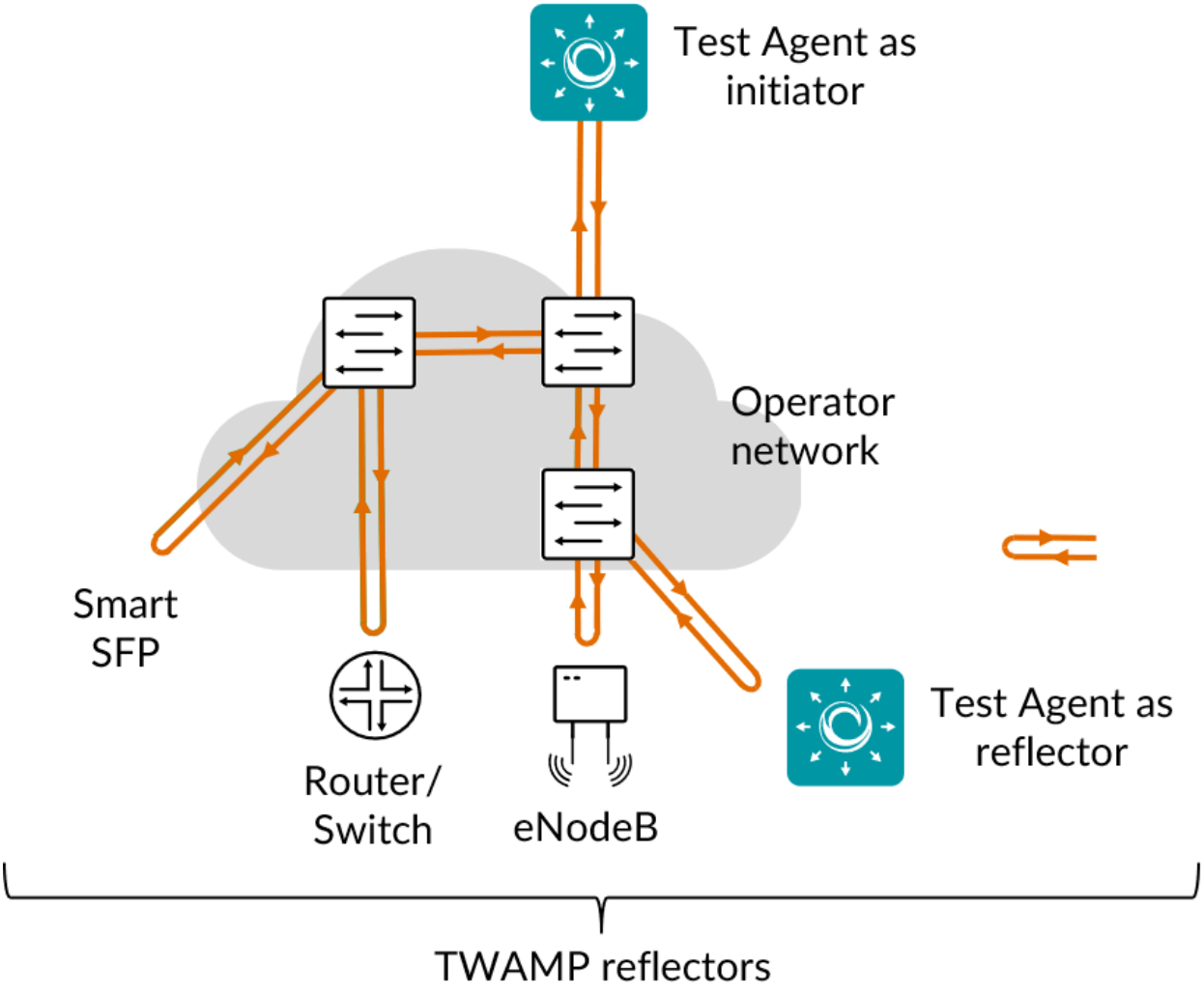
8.12.5.3 Related topics

- *Setting up TWAMP reflectors* (page 28)
- *Setting up TWAMP measurements* (page 416)

8.12.6 TWAMP/TWAMP Light

This task uses the Two-way Active Measurement Protocol (TWAMP) for measuring two-way (and in part also one-way) loss and delay. Both full TWAMP and TWAMP Light are supported. Full TWAMP includes the TWAMP control protocol, which performs a handshake between initiator and reflector, while TWAMP Light does not require the control protocol.

Either third-party devices or Test Agents can act as reflectors.



Both full TWAMP and TWAMP Light work with IPv4 as well as IPv6.

Test Agents acting as reflectors support TWAMP Light only.

8.12.6.1 Prerequisites

To perform TWAMP measurements, you need to install at least one Test Agent and have one or several TWAMP-enabled devices in your network acting as reflectors. The reflector devices may be either third-party devices or further Test Agents.

For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

To enable TWAMP in third-party equipment, please consult the documentation from your equipment vendor.

A further prerequisite for using third-party TWAMP reflectors is that you have *configured* (page 28) them in Routing Active Testing. To enable full TWAMP towards a reflector, a control port must have been set for the reflector.

To use Test Agents as reflectors in this task, you need to specify the Test Agent interfaces to be used as TWAMP Light reflectors in the Test Agent Reflectors parameter.

Once you have finished the above preparations, you can add a TWAMP/TWAMP Light task to your test or monitor, and fill in the mandatory parameters as shown below.

8.12.6.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Senders: Test Agent interfaces that will act as TWAMP senders.
- Reflectors: If you are going to use third-party TWAMP reflectors, select them here. Note: It is not certain that the port number specified for a reflector in the *inventory* (page 28) will in fact be used when the task executes. If this port is not available (for example, blocked by a firewall), the reflector may request a different port for the session, and Routing Active Testing will then accept this request and proceed with the session.
- Test Agent Reflectors: If you are going to use Test Agents as reflectors, list their interfaces here. The reflector tool is run separately on each interface.
- Test Agent Reflector Port: Port to be used by Test Agent reflectors. Default: 7000.
- Rate (Mbit/s): Rate at which the senders will send Ethernet frames in Mbit/s. Each Ethernet packet contains one frame. Max: 10,000 Mbit/s. No default.
- Rate (packets/s): Number of Ethernet frames the senders will send each second. Each Ethernet packet contains one frame. Minimum and maximum values correspond to those for Rate (Mbit/s) and depend on the Frame size setting. Min: 2 packets/s. Max: 1,000,000 packets/s. No default.

Changing one Rate parameter will cause the other to adjust automatically to agree with it, based on the Frame size setting.

- Time sync: Select “Yes” if the clocks of the Test Agent and the reflector are synchronized via NTP; otherwise, select “No”.
- In-band time sync: If NTP time synchronization is not available, select “Yes” to use in-band synchronization instead. With in-band synchronization, the sender Test Agent uses the timestamps from the TWAMP packets to estimate one-way delay. When starting a test or monitor, it may take about a minute to get good enough synchronization with this method. Please note that in-band synchronization might not be as exact as NTP. However, the algorithm indicates when it judges its estimates to be reliable (good enough convergence), and it also issues a warning whenever it thinks its estimates are currently not reliable.
- Hardware timestamping: If this is set to Yes, hardware timestamps from the Test Agent’s network interface card will be used for delay and jitter measurements. This requires support for hardware timestamping in the NIC. If this option is selected and the Test Agent NIC does not support it, an error message will be given, and the measurement will not start. By default, this parameter is set to No, and software timestamps provided by the Linux kernel are used instead. Regarding the two timestamping methods, see *below* (page 419).
- Reflector timestamping mode: one of STANDARD, FOLLOW-UP-HW, FOLLOW-UP-SW. FOLLOW-UP modes are only supported by Paragon TWAMP reflector plugin.

Timestamping methods

All timestamping for TWAMP in Routing Active Testing, whether originating from hardware or software, uses the SO_TIMESTAMPING interface provided by the Linux kernel. The definitions in this section are taken from ► https://www.kernel.org/doc/Documentation/networking/timestamping.txt.

Hardware timestamping in Routing Active Testing uses the following methods:

- SOF_TIMESTAMPING_TX_HARDWARE: Request tx timestamps generated by the network adapter. This flag can be enabled via both socket options and control messages.
- SOF_TIMESTAMPING_RX_HARDWARE: Request rx timestamps generated by the network adapter.

Software timestamping in Routing Active Testing uses the following methods; the best method available is selected:

- SOF_TIMESTAMPING_TX_SCHED: Request tx timestamps prior to entering the packet scheduler.
- SOF_TIMESTAMPING_TX_SOFTWARE: Request tx timestamps from the network interface driver.
- SOF_TIMESTAMPING_RX_SOFTWARE: Request rx timestamps when data enters the kernel. These timestamps are generated just after a device driver hands a packet to the kernel receive stack.

Thresholds for errored seconds (ES)

Note: The delay and delay variation thresholds refer to round-trip delay if no time synchronization is used (neither NTP nor in-band), and to one-way delay otherwise.

- Loss (%): Packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Max: 100%. Default: 0%.
- Delay (ms): Delay threshold for triggering an errored second. If the delay between server and reflector exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Delay variation (ms): Delay variation threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and clients exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Expected DSCP: The *Differentiated Services Code Point or IP Precedence* (page 511) that IP packets are expected to have on being received from the reflector device. If the received DSCP value does not match this, an ES will be indicated. By default, no DSCP validation is done (------ selected in drop-down box).

Thresholds for severely errored seconds (SES)

Note: The delay and delay variation thresholds refer to round-trip delay if no time synchronization is used (neither NTP nor in-band), and to one-way delay otherwise.

- Loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). Min: 0%. No default.
- Delay (ms): Delay threshold for triggering a severely errored second. Min: 0.001 ms. No default.
- Delay variation (ms): Delay variation threshold for triggering a severely errored second. Min: 0.001 ms. No default.

Advanced

Packet fragmentation

- Don't fragment packet: Controls whether the packet that exceeds the MTU is allowed to be fragmented. Be aware that enabling fragmentation (by specifying the "No" option) may cause performance degradation both in the network and in the sending or receiving Test Agents.

Note: This setting has no effect on IPv6 packets.

Percentiles

You can specify two *percentiles* for delay values. The *nth* percentile for a distribution means the smallest value which *n* percent of the data points does not exceed. Commonly used percentiles include the 90th and 99th, which informally mean "90% (99%) of the data points are below this value".

Note: The thresholds below refer to round-trip delay if no time synchronization is used (neither NTP nor in-band), and to one-way delay otherwise.

- First delay percentile (%): First delay percentile, e.g. 90.
- Threshold for first delay percentile (ms): Threshold for triggering an errored second based on the first delay percentile.
- SES threshold for first delay percentile (ms): Threshold for triggering a severely errored second based on the first delay percentile.
- Second delay percentile (%): Second round-trip delay percentile, e.g. 99.
- Threshold for second delay percentile (ms): Threshold for triggering an errored second based on the second delay percentile.
- SES threshold for second delay percentile (ms): Threshold for triggering a severely errored second based on the first delay percentile.

Note on percentile accuracy

The closer to 100% you set a percentile, the more measurement samples are required to reliably determine the corresponding delay value. For example, if you set a percentile to 99.99%, then you need 10,000 samples to get one sample above the threshold. Even then, the resulting metric is based on a single outlier, so for a more reliable result still more samples are needed.

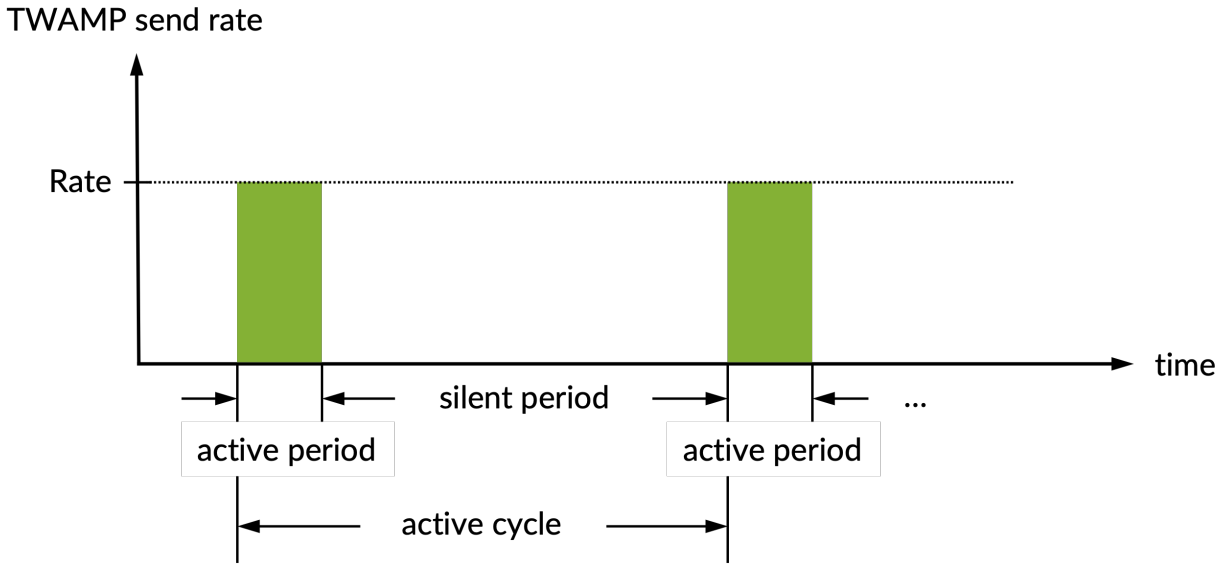
In the results, a warning is indicated for a computed percentile if there are not at least *two* samples above the percentage threshold.

Other

- Frame size (bytes): Size of Layer 2 Ethernet frame for the flow. See *this page* (page 512). Min: 87 bytes. Max: 9018 bytes. Default: 1518 bytes. If you change this setting, the Rate (packets/s) setting will adjust automatically, with Rate (Mbit/s) kept constant.
- DSCP: Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: "0 / IPP 0".
- VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See *this page* (page 516). Min: 0. Max: 7. Default: 0.
- Random padding: Use random or all zeroes as padding in TWAMP test traffic. If enabled, random padding is used. Default: Yes.
- UAS period length: The smallest number of consecutive severely errored seconds (SES) that will cause a period of unavailability to be detected. The UAS concept is described in more detail *here* (page 477). Normally, this parameter is fixed at 10 seconds, but for TWAMP it is configurable. Min: 1 s. Max: 300 s. Default: 10 s.
- Accept UDP checksum zero for IPv6: Some TWAMP reflectors sets the IPv6 UDP checksum to zero. If enabled, these packets are accepted anyway. Default: Yes.

8.12.6.3 Periodic streams

By default, Ethernet frames are sent continuously without a break at the rate specified. Alternatively, you can configure a *bursty* transmission pattern using the parameters in this section. The Test Agent will then repeat a cycle where it transmits at the rate specified by the Rate parameters for a given length of time, and then stays silent for the remainder of the cycle. See the diagram below.

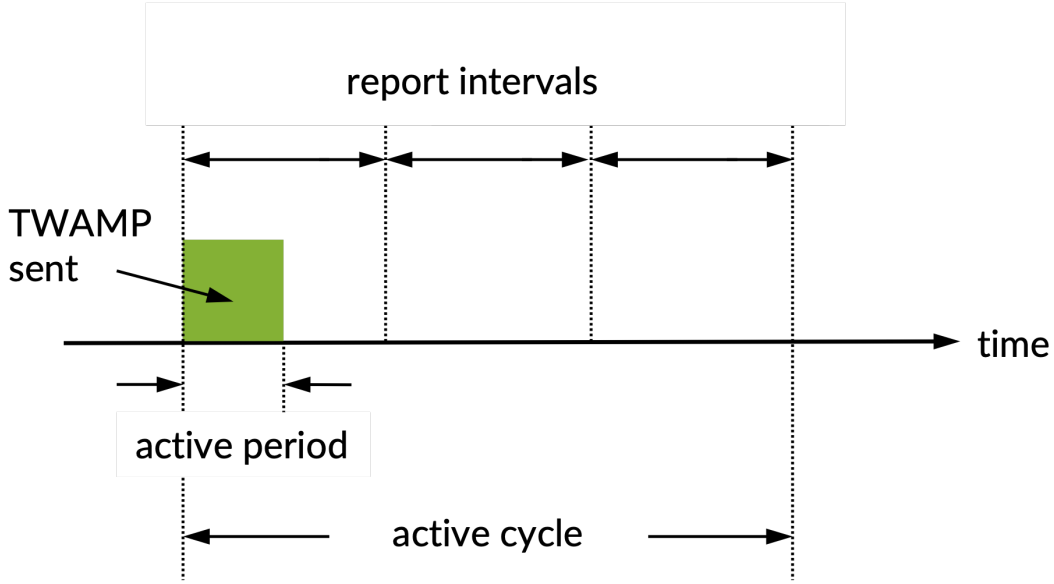


- Active period duration (ms): Length of the period in each cycle during which Ethernet frames are sent. Min: 1 ms. Max: 3,600,000 ms = 1 h.
- Active cycle (s): Length of the cycle starting with an active period and ending with a silent period. The cycle length must be at least equal to the length of the active period. Min: 1 s. Max: 604,800 s = 7 days.
- Report metrics during inactive periods: If enabled, reports metrics related to inactive periods of a periodic test. Default: No.

Whenever possible, active periods will be aligned to report intervals, so that an active period will always be contained in a single report. For this to be possible, however, the following conditions must be satisfied:

- The active period must be shorter than or equal to the report interval.
- The active cycle must be a multiple of the report interval (since otherwise the active period will drift and sometimes straddle two reports).

The diagram below gives an example where the active cycle is three times the report interval.



The report interval is

- always 10 s for monitors;
- for tests, equal to ceil(`test_length`) seconds, where `test_length` is the length of the test in *hours*. That is, for any short test, the report interval is 1 second; for tests more than an hour in length, the report interval increases by 1 second for each hour of testing.

8.12.6.4 Result metrics

Note: "Far-end" refers to the direction towards the reflector. "Near-end" means from the reflector back towards the sender Test Agent.

- **Rate (Mbit/s):** Rate at which the senders received TWAMP packets.
- **Min round-trip delay (ms):** Minimum round-trip delay.
- **Average round-trip delay (ms):** Average round-trip delay.
- **Max round-trip delay (ms):** Maximum round-trip delay.
- **Average round-trip DV (ms):** Average round-trip delay variation.
- **Received packets:** Number of packets received.
- **Round-trip loss (%):** Round-trip packet loss in percent.
- **Round-trip lost:** Number of lost packets, round-trip.

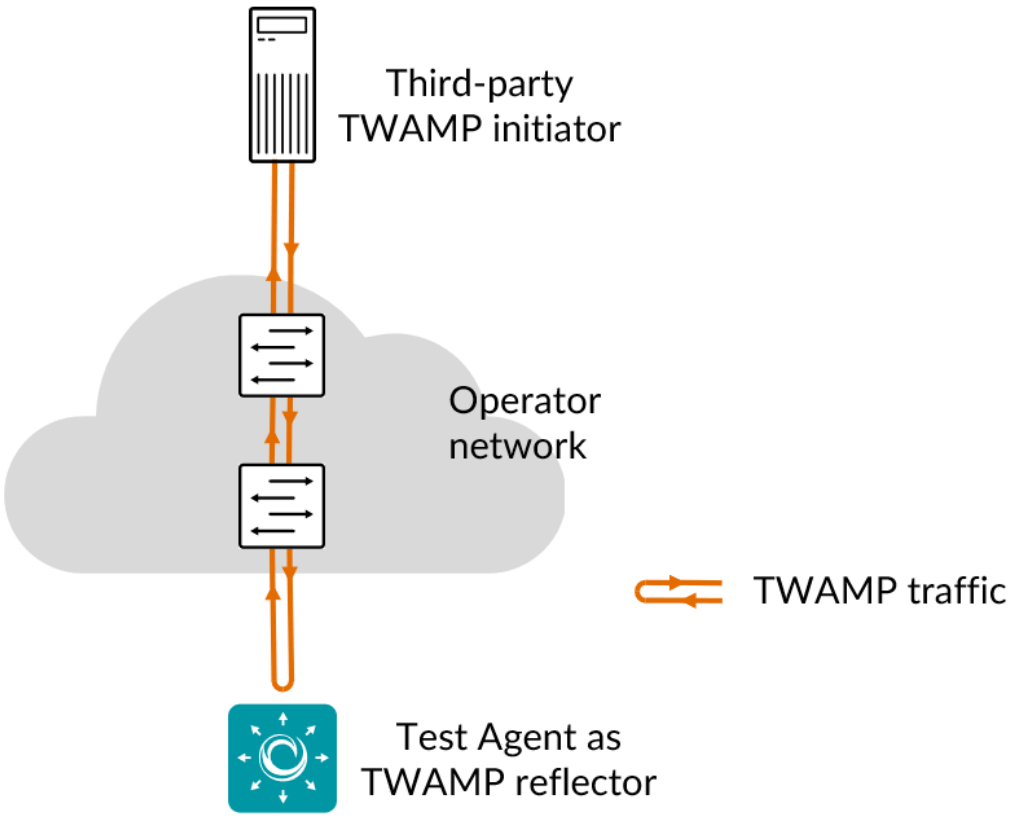
- **Far-end loss (%):** Packet loss in percent, far-end.
- **Far-end lost:** Number of lost packets, far-end.
- **Far-end misorders:** Number of misordered packets, far-end.
- **Min far-end delay (ms):** Minimum one-way far-end delay.
- **Average far-end delay (ms):** Average one-way far-end delay.
- **Max far-end delay (ms):** Maximum one-way far-end delay.
- **Far-end DV (ms):** Far-end delay variation.
- **Near-end loss (%):** Packet loss in percent, near-end.
- **Near-end lost:** Number of lost packets, near-end.
- **Near-end misorders:** Number of misordered packets, near-end.
- **Min near-end delay (ms):** Minimum one-way near-end delay.
- **Average near-end delay (ms):** Average one-way near-end delay.
- **Max near-end delay (ms):** Maximum one-way near-end delay.
- **Near-end DV (ms):** Near-end delay variation.
- **First round-trip delay percentile:** First delay percentile for round-trip delay.
- **Second round-trip delay percentile:** Second delay percentile for round-trip delay.
- **First far-end delay percentile:** First delay percentile for far-end delay.
- **Second far-end delay percentile:** Second delay percentile for far-end delay.
- **First near-end delay percentile:** First delay percentile for near-end delay.
- **Second near-end delay percentile:** Second delay percentile for near-end delay.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES loss (%):** Accumulated errored second percentage for packet loss.
- **ES delay (%):** Accumulated errored second percentage for two-way delay.
- **ES delay variation (%):** Accumulated errored second percentage for delay variation.
- **ES DSCP (%):** Accumulated errored second percentage for DSCP.
- **SES (%):** Aggregated severely errored second (SES) percentage, taking into account all types of error.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.12.7 TWAMP Reflector

The TWAMP Reflector task starts Test Agent TWAMP reflectors and collects basic metrics from these reflectors as they respond to third-party senders.

At present, the Test Agent TWAMP reflectors only support the TWAMP Light protocol. This means they have no control channel support.

Both IPv4 and IPv6 are supported by Test Agent TWAMP reflectors.



8.12.7.1 Prerequisites

To run a TWAMP Reflector task, you need to install at least one Test Agent and have one or several TWAMP-enabled devices in your network.

For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70). The procedure is different depending on what kind of hardware you have available. Regarding how to enable TWAMP on your equipment, please consult the documentation from your equipment vendor.

You also need to have third-party equipment acting as TWAMP senders.

8.12.7.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Test Agent Reflectors: Test Agent interfaces that will act as TWAMP reflectors.
- Test Agent Reflector Port: Port to be used by Test Agent TWAMP reflectors.

Note: Test Agent TWAMP reflectors will not show up in the TWAMP reflector inventory under Account. Rather, they are started and stopped dynamically along with the TWAMP Reflector task itself.

If you want to use Test Agent reflectors in a TWAMP/TWAMP Light task, we recommend that you point directly to these reflectors when configuring the task: see *this page* (page 416).

For *special use cases* (page 416), you might want to start the Test Agent TWAMP reflectors, add them manually to the TWAMP inventory as detailed on *this page* (page 28), and then use these reflectors in the TWAMP/TWAMP Light task.

Thresholds for errored seconds (ES)

- Rate (Mbit/s): Threshold rate for severely errored second.

8.12.7.3 Result metrics

- **Rate (Mbit/s):** Actual rate at which the senders sent TWAMP packets.
- **Received packets:** Number of packets received.
- **Received bytes:** Number of bytes received.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.

8.12.8 Junos TWAMP

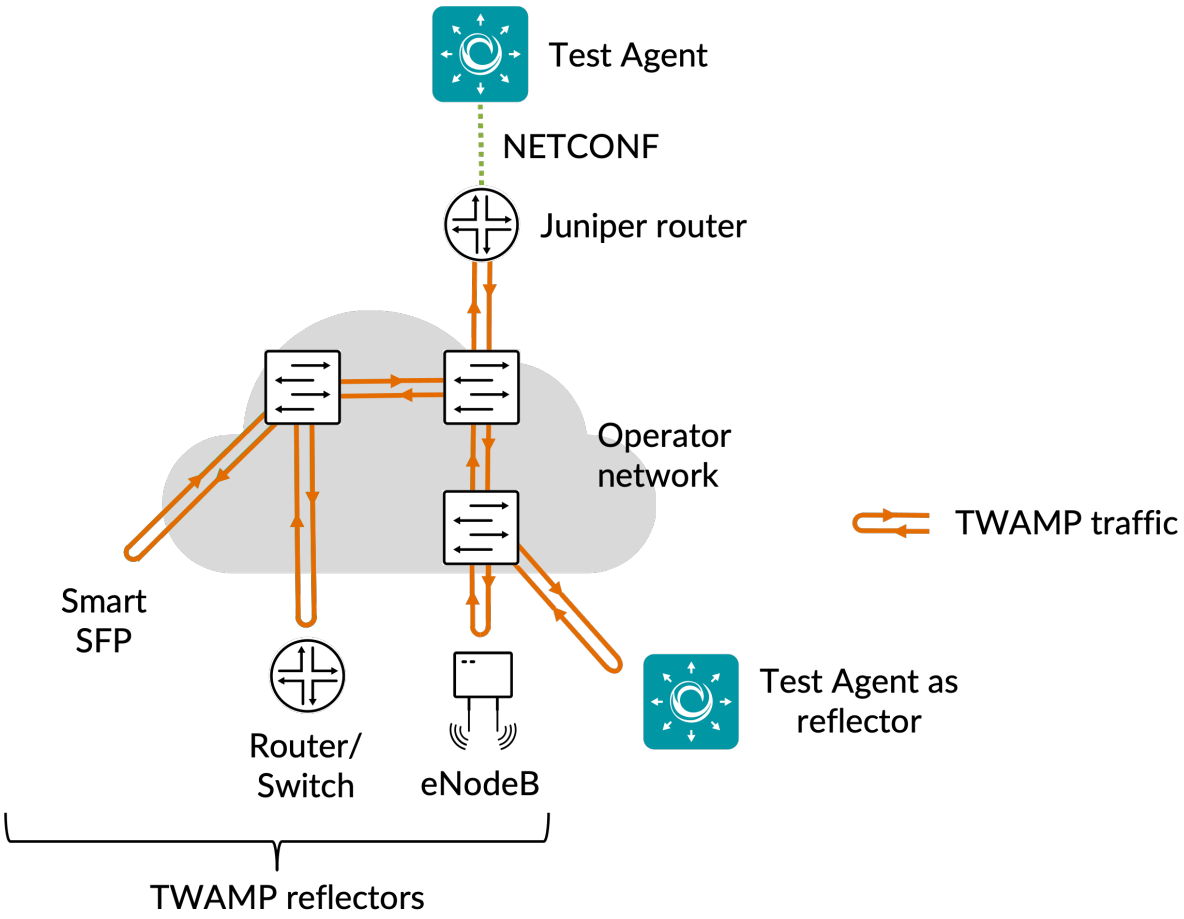
In this task, a Test Agent Application connects to one or several Junos devices (defined as *network devices* (page 35) in Routing Active Testing) via the NETCONF protocol and accesses TWAMP sessions running on these devices (these sessions have not been configured in Routing Active Testing). The Test Agent collects measurement results from the TWAMP sessions, evaluates errored second thresholds, and reports all results back to Control Center.

The TWAMP sessions running on the Junos devices are identified based on their configured "control connection" and "test session" (specified as `services rpm twamp client control-connection <control connection> test-session <test session>` in the Junos CLI). The names of the control connections and test sessions are shown along with the results in Control Center so that you can correlate them with the TWAMP sessions on the Junos device.

TWAMP sessions can optionally be run against other Test Agents acting as reflectors. Such Test Agents then need to be running a *TWAMP Reflector* (page 424) task.

Note: A Test Agent Application is required for this task; the Test Agent Appliance does not support it. The Junos TWAMP task is also different from the other TWAMP tasks in that the Test Agent does not itself conduct the measurements.

IPv6 is supported in the communication between the Test Agent and the Junos device. For the communication between TWAMP sender/client and reflector, Junos currently only supports IPv4.



8.12.8.1 Prerequisites

To perform Junos TWAMP measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

As regards each targeted Junos device, the following holds:

- The functionality has been verified to work on the following Juniper device models, but it might also work on other TWAMP-capable devices:
  - vMX
  - MX204

  - MX480
  - MX960
- The functionality has been verified for Junos versions 18.3–20.2.
- There must be network connectivity from the Test Agent to the device (default TCP port: 830).
- You must have a user account on the device to be able to log in to it and retrieve measurement data. How to create a user account is described *here* (page 427).
- The Junos device must be configured as a *network device* (page 35) in the Routing Active Testing inventory.
- The Junos device must be running TWAMP measurements when you execute the Junos TWAMP task. The relevant Junos documentation is found here.

Once you have finished the above preparations, you can add a Junos TWAMP task to your test or monitor and fill in the mandatory parameters as shown below.

8.12.8.2 Junos device configuration

This section is about suitable configuration of the Junos device. This needs to be done directly on the device and cannot be performed in the Routing Active Testing task. For details, please refer to Junos device documentation.

Creating a user account on a Junos device

The user account should have limited permissions and can be created as follows:

```
configure
set system login user <username> class read-only
set system login user <username> authentication plain-text-password
New password: <password>
commit
```

Configuring TWAMP history size

The configuration parameter `services rpm twamp client control-connection <control connection> history-size` in the Junos device specifies how many historical probe results are stored for each control connection and test session. (Regarding these concepts, see the *introductory section* (page 425) above.)

The Test Agent will fetch new probe results from this result set every 5 seconds. To have some margin for communication delays between the Test Agent and the Junos device, we recommend that you configure `history-size` to correspond to at least 1 minute. For example, with `probe-interval` set to 10 seconds, we recommend a `history-size` of at least 6.

8.12.8.3 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Client: Test Agent interface that will connect to the Junos device.
- Network devices: Junos devices that are running TWAMP tasks and will be accessed by the Test Agent.

Thresholds for errored seconds (ES)

- RTT threshold (ms): Round-trip time threshold for triggering an errored second. If the round-trip time exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Jitter threshold (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.

Note: Loss will always trigger an errored second.

Advanced

- Collection interval (s): The interval at which results are collected from the Junos devices. Min: 5s Max: 300s Default: 5s.
- Session timeout (s): The time after which idle sessions will be removed.

This functionality is offered in the Juniper API.

- Filter control connection: Only collect results from the specified control connection as configured on the Junos device (irrespective of test session). Only one control connection can be specified. If you leave this blank, results will be collected from all control connections.
- Filter test session: Only collect results from the specified test session as configured on the Junos device (irrespective of control connection). Only one test session can be specified. If you leave this blank, results will be collected from all test sessions.

If you filter on both control connection and test session, both of these must match.

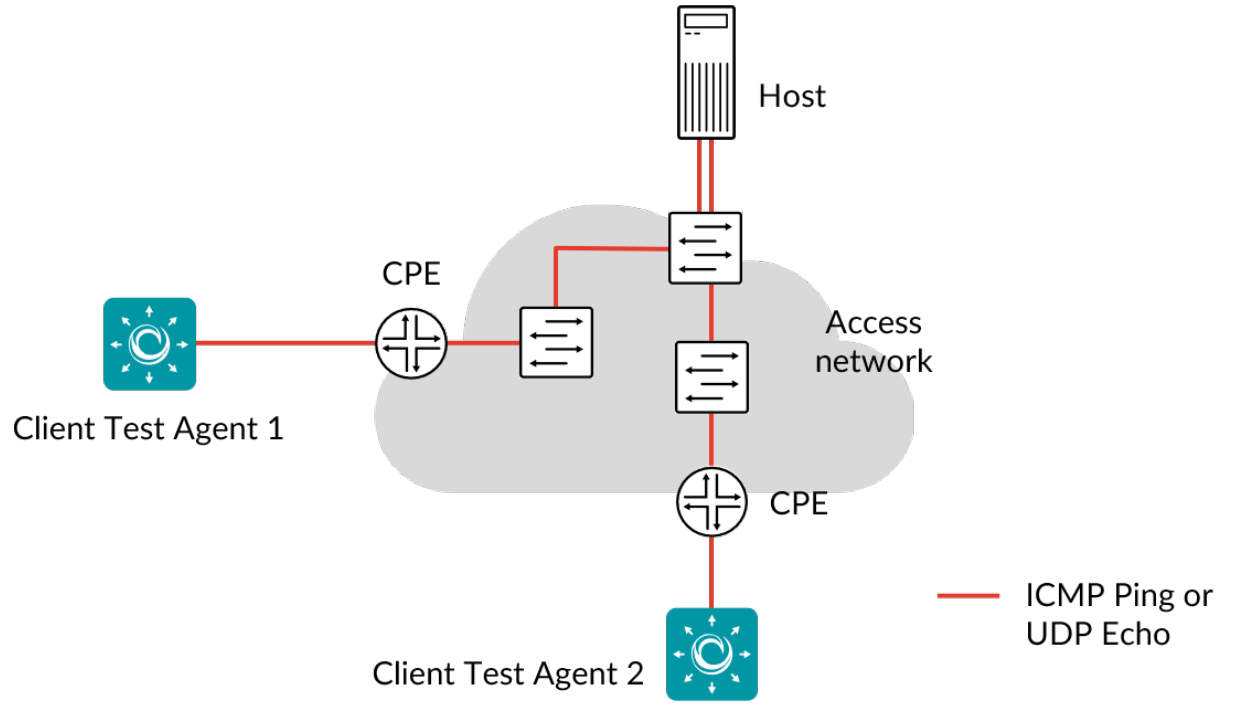
8.12.8.4 Result metrics

- **RTT (ms):** Delay between the transmission of a probe and the arrival of its response.
- **Egress jitter (ms):** Difference between the current egress delay and the previous measurement.
- **Ingress jitter (ms):** Difference between the current ingress delay and the previous measurement.
- **Round-trip jitter (ms):** Difference between the current round-trip time and the previous measurement.
- **Egress interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in ► IETF RFC 1889, calculated for the outbound direction (from Junos device to reflector).
- **Ingress interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in IETF RFC 1889, calculated for the inbound direction (from reflector back to Junos device).
- **Round-trip interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in IETF RFC 1889, calculated for the full round-trip.

Note: Jitter and interarrival jitter are calculated differently from what is called "delay variation" in other tasks.

- **Loss (%):** Round-trip packet loss in percent.
- **ES (s):** Aggregated number of errored seconds (ES), taking into account all types of error.
- **ES loss (s):** Number of errored seconds caused by loss.
- **ES RTT (s):** Number of errored seconds caused by round-trip time.
- **ES jitter (s):** Number of errored seconds for caused by jitter.

8.12.9 Ping



Ping is a computer network administration utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer. The hosts can reside inside or outside your network.

By using Ping you can find out if any of the IP hosts have a problem, and correlating Ping responses from different hosts helps you pin down where problems occur.

When you start Ping testing, the Test Agents will continuously send ICMP Ping messages towards the hosts you have specified and collect statistics on round-trip delay and packet loss.

The ICMP protocol is defined in ► IETF RFC 792.

Besides ICMP-based Ping, *UDP Echo* is also supported. This is a UDP-based echo service, where a server listens for UDP echo requests on a UDP port. When such a frame is received, its data is sent back in an answering frame ("echo respond"). The metrics are the same as those retrievable with ICMP Ping. The UDP Echo protocol is defined in ► IETF RFC 862.

This task works with both IPv4 and IPv6.

8.12.9.1 Prerequisites

To run Ping measurements in Routing Active Testing you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

In your test or monitor, add a Ping task and fill in the mandatory parameters below.

Prerequisites for Test Agent Applications

If you want to run Ping on a Test Agent Application, you need to do one of the following:

Either

- **Run as root:** Start the Test Agent Application using `sudo`. Docker is mostly run `sudo`, so if you are running the Test Agent in a Docker container, you probably will not need to do anything special.

or

- **Set CAP_NET_RAW capability** on the Test Agent Application executable. (You cannot set this capability on the *plugin* executable, since that is a Python script.) Be aware that the Test Agent will then have this capability all the time, and not just when using the Ping plugin.

```
sudo setcap cap_net_raw=ep paa-test-agent-application
```

The Ping plugin also needs Python 3.7 to run. In the Docker Test Agent deliverable this is included, so the Ping plugin will run fine from Docker. However, if the Test Agent Application is installed from a tarball, Ping will not work out-of-the-box unless Python 3.7 is present.

8.12.9.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agents to use as clients. These can be located behind NAT.

You can either enter hosts to ping manually or select them from an inventory prepared in Routing Active Testing. How to set up such an inventory is covered *here* (page 24).

- Hosts: Hosts: The hosts to ping as IP addresses or host names.
- Hosts inventory: The hosts to ping from your inventory of Ping hosts.
- Time between requests: Time to wait between consecutive Ping requests. Min: 0.01 s. Max: 3600 s. Default: 10 s.

Threshold for errored seconds (ES)

- Delay (ms): Maximum tolerated Ping delay. If this value is exceeded, an errored second will be triggered. Min: 1 ms. Max: 30,000 ms. Default: 1000 ms.
- DV (ms): Delay variation (jitter) threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and clients exceeds this value during one second, an ES will be indicated. Delay variation is measured every second as the difference between the longest and shortest round trip time of ping responses received during that second. Min: 0.0 ms. Max: 10,000.0 ms. Default: 500.0 ms.

Advanced

- Request lifetime (ms): Maximum time to wait for a Ping response before the Ping request is canceled. Min: 1 ms. Max: 30,000 ms. Default: 2000 ms.
- TTL: Time To Live, the number of router hops an ICMP packet is allowed to traverse through the IP network before it is discarded by a router. For each router hop, the TTL value is decremented by one, and when TTL reaches zero, the IP packet is discarded. Min: 1. Max: 255. Default: 64. See also ► IETF RFC 792.

  Examples of default TTL values:

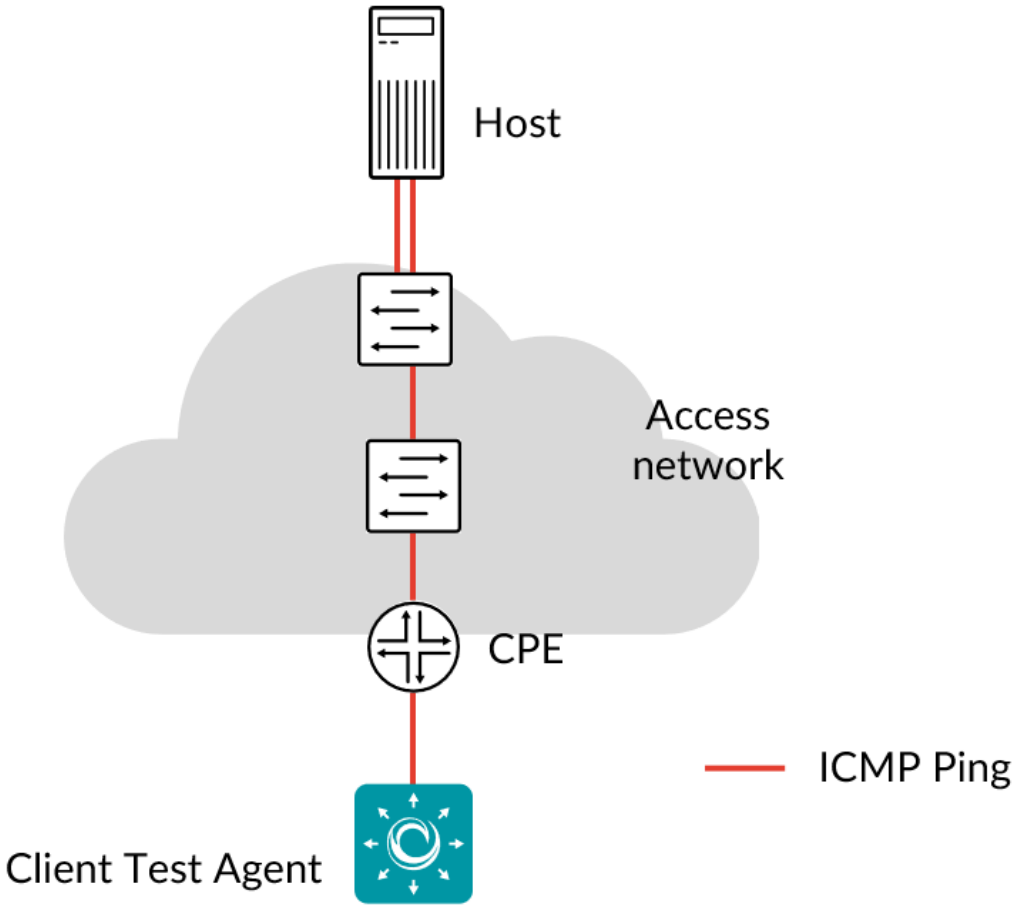
  - Linux 2.4 kernel: 255
  - MacOS X (10.5.6): 64
  - Windows 7: 128
  - Windows 10: 64
- Payload (bytes): Number of bytes carried in the Ping packet payload. Min: 0 bytes. Max: 65,000 bytes. Default: 56 bytes.
- DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). Default: "0 / IPP 0".
- Protocol: Network protocol to use for sending Ping packets: ICMP or UDP. Default: ICMP.

8.12.9.3 Result metrics

- **Average round-trip delay (ms):** Average round-trip delay in milliseconds during the selected time period.
- **Min round-trip delay (ms):** Minimum round-trip delay in milliseconds.
- **Max round-trip delay (ms):** Maximum round-trip delay in milliseconds.
- **Average round-trip DV (ms):** Average round-trip delay variation (jitter) in milliseconds.
- **Loss (%):** Percentage of Ping packets lost.
- **ES loss:** Number of errored seconds due to lost Pings (for which reply was not received): for example, if Time-To-Live (TTL) expired and the Ping packet was dropped by a router.
- **ES delay:** Number of errored seconds triggered because the Delay (Ping delay) threshold was exceeded.
- **ES response:** Number of errored seconds due to unresolved hosts or payload mismatch (sent and received Ping data did not match).
- **ES DV:** Number of errored seconds due to the jitter (delay variation, DV) threshold being exceeded.
- **ES:** Aggregated errored seconds, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES total**) %.

8.12.10 BWPing

This task measures bandwidth and response times between a Test Agent and a network device (router or switch) using the Internet Control Message Protocol (ICMP) echo request/echo reply mechanism. The tool is based on the BWPing software, which is available at ► bwping.sourceforge.net.



The primary purpose of the BWPing task is to achieve high data throughput in service activation testing or troubleshooting. It is especially useful when testing towards a host device that does not support *TWAMP* (page 413). Bandwidths up to 30 Mbit/s per task instance can be achieved. The BWPing task is run once, for a specified duration, and is therefore available only for use in tests (not in continuously executing monitors). A report with successive measurement results, produced at user-specified intervals, is output directly in the BWPing test view. Multiple flows for different QoS classes can be run concurrently in a BWPing test.

While based on ICMP, the BWPing task is fundamentally different from the regular *Ping* (page 430) task, which is intended for reachability testing or (with path trace) monitoring of round-trip times, and which does not provide data rate as a result metric.

The ICMP protocol is defined in ► IETF RFC 792.

This task works with both IPv4 and IPv6.

8.12.10.1 Prerequisites

To perform BWPing measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

In your test or monitor, add a BWPing task and fill in the mandatory parameters below:

8.12.10.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *Advanced settings* (page 287) common to all test tasks: Delayed start.

General

- Client: Test Agent interface that will act as sender (and receiver) of ICMP Ping packets.
- Host: IP address of device acting as reflector.
- Test duration (s): Duration of BWPing test.
- Report interval (s): Interval at which new result metrics are reported. For each new report, a new line is written to the result table. Min: 1 s. Max: 60 s. Default: 1 s.

Class 1 ... Class 6

A separate input parameter section is provided for each QoS class to be tested. These classes are distinguished by the DSCP value set for each.

- Class name: The name of the QoS class. Default names are simply "1" … "6". If you define your own class names, the headings "Class 1", etc., will change accordingly.

For each class, the following parameters can be defined:

- Rate (Mbit/s): Rate at which the sender will send Ethernet frames for this QoS class. Min: 0.1 Mbit/s. Max: 4000 Mbit/s. The combined rate for all classes taken together also cannot exceed 4000 Mbit/s. No default.
- Frame size (bytes): Size of Layer 2 Ethernet frame for the flow. See *this page* (page 512). Min: 64 bytes. Max: 9018 bytes. Default: 1518 bytes.
- DSCP: Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: "0 / IPP 0".
- Socket buffer size (bytes), optional: Socket buffer size to use for both sending and receiving. *(Optional.)* If this is not set, the BWPing software will configure the buffer sizes according to its own formula.

Thresholds (test fail criteria)

Note: These thresholds are applied to the final results of the test, that is, those reported on the "Total" line in the table. No errored seconds are computed based on intermediate test results.

- Rate threshold (Mbit/s): Ethernet rate threshold for test failure. Min: 0.1 Mbit/s. Max: 4000 Mbit/s. No default.
- Delay threshold (ms): Round-trip delay threshold for test failure. Min: 1 ms. Max: 1000 ms. No default.

- Delay variation threshold (ms): Round-trip delay variation threshold for test failure. Min: 1 ms. Max: 1000 ms. No default.

8.12.10.3 Result metrics

The results are output as periodic reports in tables with columns as listed below. A separate table is produced for each QoS class tested. Further tables are also given, showing the general test configuration as well as the configuration for each QoS class.

The periodic reports are cumulative, showing results from the start of the test up to the indicated point in time. The last row (with "Total" in the **Time** column) is the final test report spanning the entire test duration.

- **Time:** Time in seconds from start of test.
- **TX pkts:** Number of Ping packets transmitted.
- **RX pkts:** Number of Ping packets received.
- **Rate (Mbit/s):** Received Ethernet packet rate. Note: Although the BWPing software operates on the IP level, Routing Active Testing reports Ethernet packet rate to conform to the reporting for other tasks. – It should also be mentioned that the intermediate rates reported are sometimes lower than the configured rate. This is what the BWPing software reports, and it is shown without modification in Routing Active Testing. The most accurate rate is that for the entire test (last row in table, "Total"). As noted above, the rate for the entire test is also what Routing Active Testing compares to the rate threshold for test failure.
- **Min delay (ms):** Minimum round-trip delay.
- **Avg delay (ms):** Average round-trip delay.
- **Max delay (ms):** Maximum round-trip delay.
- **DV (ms):** Round-trip delay variation.

Again, note that all of these values apply to the whole test thus far, and not to the time elapsed since the previous report.

8.12.10.4 BSD license for BWPing

Copyright © 2016 Oleg Derevenetz <oleg.derevenetz@gmail.com>. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY OLEG DEREVENETZ "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

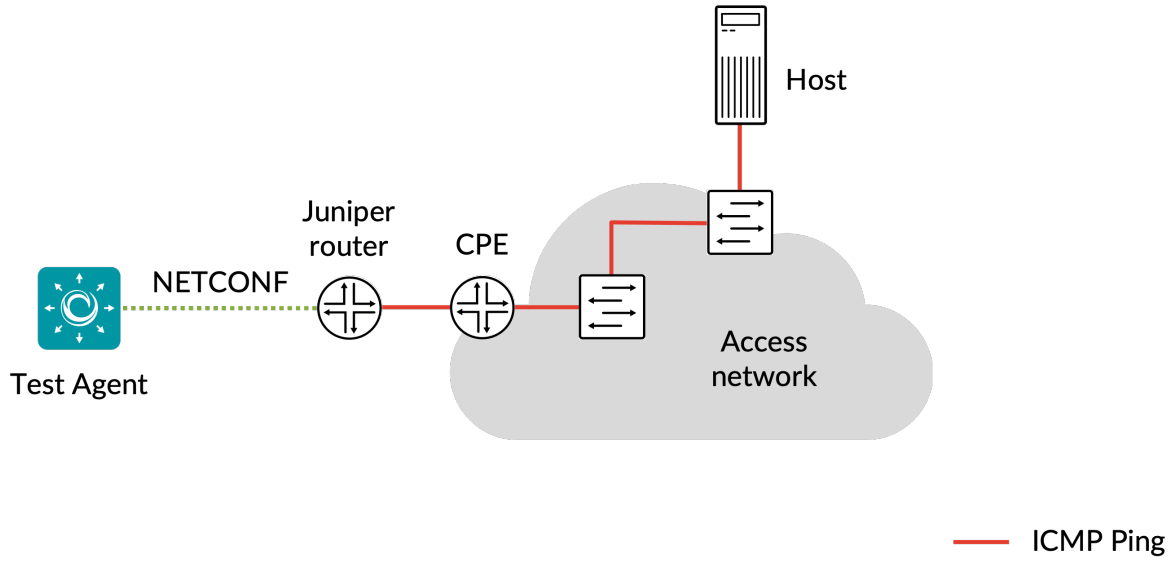
8.12.11 Junos ICMP

In this task, a Test Agent Application connects to one or several Junos devices (defined as *network devices* (page 35) in Routing Active Testing) via the NETCONF protocol and accesses ICMP sessions running on these devices (these sessions have not been configured in Routing Active Testing). The Test Agent collects measurement results from the ICMP sessions, evaluates errored second thresholds, and reports all results back to Control Center.

Each ICMP session running on a Junos device is identified as a "test" belonging to an "owner" (specified as `services rpm probe <owner> test <test>` in the Junos CLI). The test and owner are shown along with the results in Control Center so that you can correlate them with the ICMP sessions on the Junos device.

Note: A Test Agent Application is required for this task; the Test Agent Appliance does not support it. The Junos ICMP task is also different from *regular Ping* (page 430) in that the Test Agent does not itself conduct the measurements.

IPv6 is supported in the communication between the Test Agent and the Junos device.



8.12.11.1 Prerequisites

To perform Junos ICMP measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

As regards each targeted Junos device, the following holds:

- The functionality has been verified to work on the following Juniper device models, but it might also work on other devices:
  - vMX

  - MX204

- The functionality has been verified for Junos versions 18.3–20.2. Junos Evolved is not supported.
- There must be network connectivity from the Test Agent to the device (default TCP port: 830).
- You must have a user account on the device to be able to log in to it and retrieve measurement data. How to create a user account is described *here* (page 437).
- The Junos device must be configured as a *network device* (page 35) in the Routing Active Testing inventory.
- The probe of the Junos device must be configured with the target address:

```
set services rpm probe <owner> test <test> target <address_type> <address>
```

  Example (IPv4):

```
set services rpm probe owner1 test t1 target address 192.168.0.1
```

  Example (IPv6):

```
set services rpm probe owner1 test t1 target inet6-address 2001:0DB8::1
```

- The probe of the Junos device must also be configured with the correct probe type:

```
set services rpm probe <owner> test <test> probe-type <icmp-ping/icmp6-ping/icmp-
↪ping-timestamp>
```

- The Junos device must be running ICMP measurements when you execute the Junos ICMP task. The relevant Junos documentation is found here.

Once you have finished the above preparations, you can add a Junos ICMP task to your test or monitor and fill in the mandatory parameters as shown below.

8.12.11.2 Junos device configuration

This section is about suitable configuration of the Junos device. This needs to be done directly on the device and cannot be performed in the Routing Active Testing task. For details, please refer to Junos device documentation.

Creating a user account on a Junos device

The user account should have limited permissions and can be created as follows:

```
configure
set system login user <username> class read-only
set system login user <username> authentication plain-text-password
New password: <password>
commit
```

Configuring ICMP history size

The configuration parameter `services rpm probe <owner> test <test> history-size` in the Junos device specifies how many historical probe results are stored for each owner and test. (Regarding these concepts, see the *introductory section* (page 436) above.)

The Test Agent will fetch new probe results from this result set every 5 seconds. To have some margin for communication delays between the Test Agent and the Junos device, we recommend that you configure `history-size` to correspond to at least 1 minute. For example, with `probe-interval` set to 10 seconds, we recommend a `history-size` of at least 6.

8.12.11.3 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Client: Test Agent interface that will connect to the Junos device.
- Network devices: Junos devices that are running ICMP tasks and will be accessed by the Test Agent.
- Filter based on owner: Only collect results from the specified owner as configured on the Junos device. If you leave this blank, results will be collected from all owners.
- Filter test session: Only collect results from the specified test as configured on the Junos device. If you leave this blank, results will be collected from all tests.

Thresholds for errored seconds (ES)

- RTT threshold (ms): Round-trip time threshold for triggering an errored second. If the round-trip time exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Jitter threshold (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.

Note: Loss will always trigger an errored second.

Advanced

- Collection interval (s): The interval at which results are collected from the Junos devices. Min: 5s Max: 300s Default: 5s.
- Session timeout (s): The time after which idle sessions will be removed.

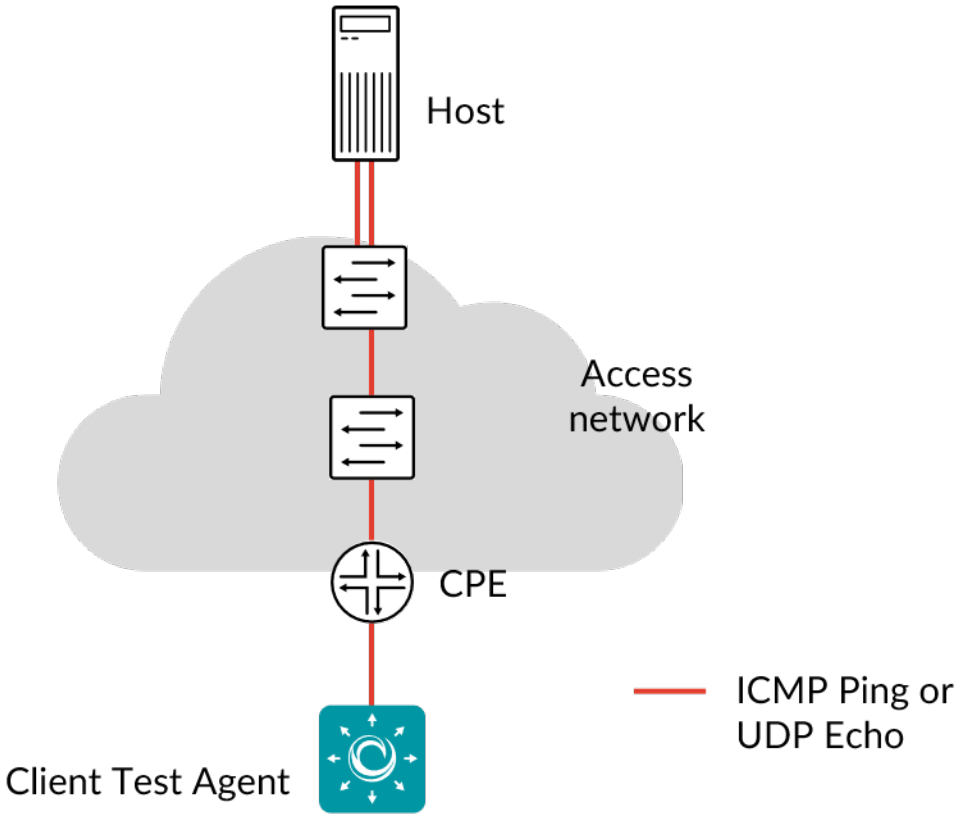
8.12.11.4 Result metrics

- **RTT (ms):** Delay between the transmission of a probe and the arrival of its response.
- **Round-trip jitter (ms):** Difference between the current round-trip time and the previous measurement.
- **Round-trip interarrival jitter (ms):** Estimate of the statistical variance of a packet's interarrival time as defined in IETF RFC 1889, calculated for the full round-trip.

Note: Jitter and interarrival jitter are calculated differently from what is called "delay variation" in other tasks.

- **Loss (%):** Round-trip packet loss in percent.
- **ES (s):** Aggregated number of errored seconds (ES), taking into account all types of error.
- **ES loss (s):** Number of errored seconds caused by loss.
- **ES RTT (s):** Number of errored seconds caused by round-trip time.
- **ES jitter (s):** Number of errored seconds for caused by jitter.

8.12.12 Path trace



When data is rerouted in a network, this often results in packet loss, reordering of packets, and jitter. It is important to keep track of how this affects user traffic, and to identify where between two endpoints a problem resides. For these tasks, a path tracing tool is useful.

The Path trace tool in Routing Active Testing continuously sends trains of ICMP and/or UDP Echo packets with increasing TTL, and measures the time it takes from sending a packet to receiving an ICMP control message back from each router. Any route changes are detected and recorded.

The Paris traceroute algorithm is used in order to minimize the risk of missing links, or detecting false links, in the presence of per-flow or per-packet load balancing (something which causes difficulties for classic traceroute). Read more about this algorithm here.

In the user interface, routes are visualized in a graph. The times of route changes are collected in a drop-down list, and selecting one of these time instants highlights the route taken at that time in the graph. Each router hop in the graph is labeled with a user-selected ES metric (delay, jitter, or loss), showing the quality experienced for that hop.

A key property of the Path trace tool is the continuous detection of network paths, which keeps the results up-to-date and relevant when the path changes.

The Path trace tool supports both IPv4 and IPv6.

8.12.12.1 Prerequisites

To run Path trace measurements in Routing Active Testing you need to have at least one Test Agent installed. If you haven't already done the installation, consult the installation guides found *here* (page 70).

In your test or monitor, add a Path trace task and fill in the mandatory parameters below:

8.12.12.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Client: Test Agent interface that will act as client, sending packets to the host.
- Host: Host to send packets to. Specified as IP address or host name.
- Rate (packets/s): Number of Ethernet frames per hop that the client will send each second. Each Ethernet packet contains one frame. Min: 0.1. Max: 50. Default: 2.
- Lifetime (ms): Lifetime of a packet. Lifetime of a packet before it is marked as lost. Min: 0.1. Max: 50. Default: 2.
- Evaluate thresholds for every hop: Here you select whether to evaluate ES and SES thresholds for every hop. Default: False.

Thresholds for errored seconds (ES)

- Loss (%): Packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Max: 100%. Default: 0%.
- Delay (ms): Delay threshold for triggering an errored second. If the delay between server and reflecting router exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- DV (ms): Jitter threshold for triggering an errored second. If the *jitter (delay variation)* (page 474) between server and clients exceeds this value during one second, an ES will be indicated. Min: 0.001 ms. Max: 1000 ms. No default.
- Expected DSCP: The *Differentiated Services Code Point or IP Precedence* (page 511) that IP packets are expected to have when arriving at the reflecting router. This is possible to check because each IP packet sent back from the reflector contains an ICMP packet, which in turn contains the modified version of the IP packet sent by the Test Agent. If the DSCP value in the latter packet does not match Expected DSCP, an ES will be indicated. By default, no DSCP validation is done (------ selected in drop-down box).

Thresholds for severely errored seconds (SES)

- Loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). Min: 0%. No default.
- Delay (ms): Delay threshold for triggering a severely errored second. Min: 0.001 ms. No default.
- DV (ms): Delay variation threshold for triggering a severely errored second. Min: 0.001 ms. No default.

Advanced

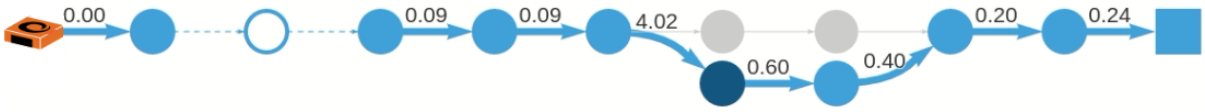
- Frame size (bytes): Size of Layer 2 Ethernet frame for the flow. See *this page* (page 512). Min: 82 bytes. Max: 9018 bytes. Default: 82 bytes.
- Max TTL: The maximum value of the Time-To-Live parameter. The same parameter also limits the number of flows that can be running in the Path trace tool per test or monitor. Min: 2. Max: 64. Default: 30.
- Max results: The maximum number of hops (sum taken over all routes) that can be processed per test or monitor. A warning will be shown if this maximum is reached. Min: 1. Max: 1000. Default: 128.
- DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). Default: "0 / IPP 0".
- Protocol: Network protocol to use for sending Ping packets: ICMP or UDP. Default: ICMP.
- UDP port: The UDP port to use. *(Visible only if UDP is selected under Protocol.)* Default: 7.
- Stable period: The number of seconds for which all packet trains must follow the same route in order for a stable state to be assumed. Min: 3. Max: 60. Default: 10.

8.12.12.3 Presentation

Overview

The presentation is divided into two tabs, Graph and Result List.

On the Graph tab, which is displayed by default, the route or routes taken from the client (Test Agent) to the host (destination) are presented in a graph showing the successive hops between intermediate routers. The client is represented by a Test Agent icon and the host by a square, while intermediate routers appear as circles. The IP address and DNS name of each router can be viewed in a tooltip.

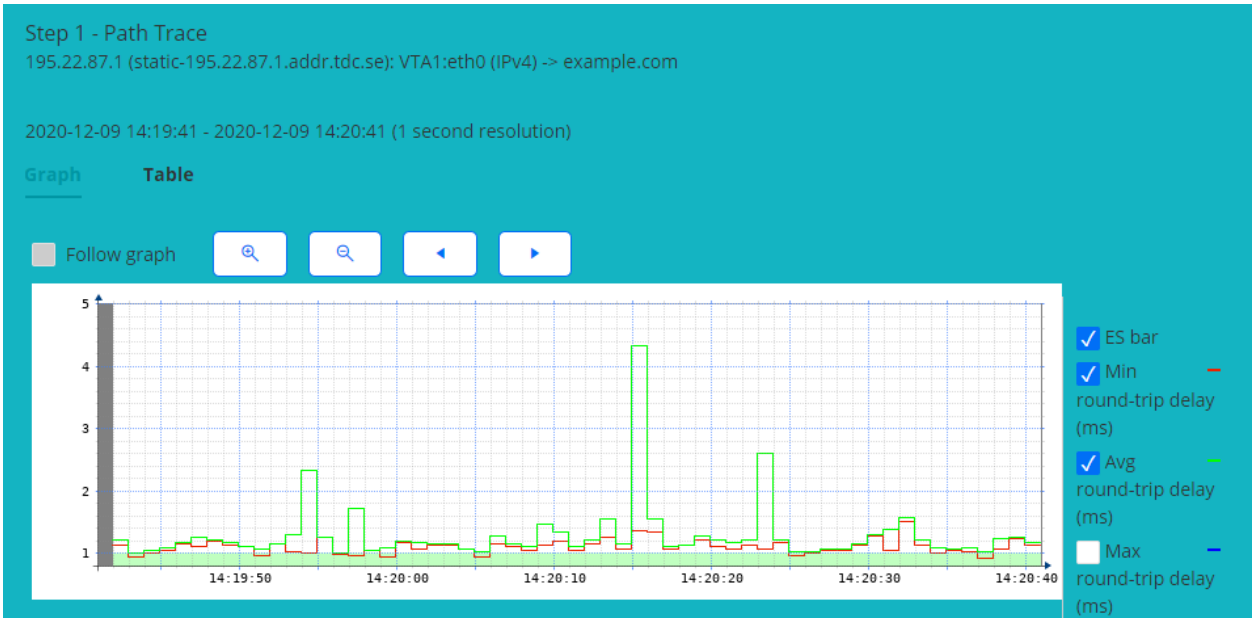


By default the graph shows the latest route taken, that is, the route taken as a result of the latest *reroute event* (page 444). This is also the route shown when you refresh the page in the browser.

Each hop is labeled with the value of a selected *metric* (page 445). The tooltip for a router, apart from the IP address and DNS name, also displays all metrics computed for the hop which terminates at that router.

The thickness of an arrow indicates how frequently the network path in question has been used: the thicker the arrow, the more often this path has been taken.

Clicking an arrow opens a separate window with detailed data for this hop:



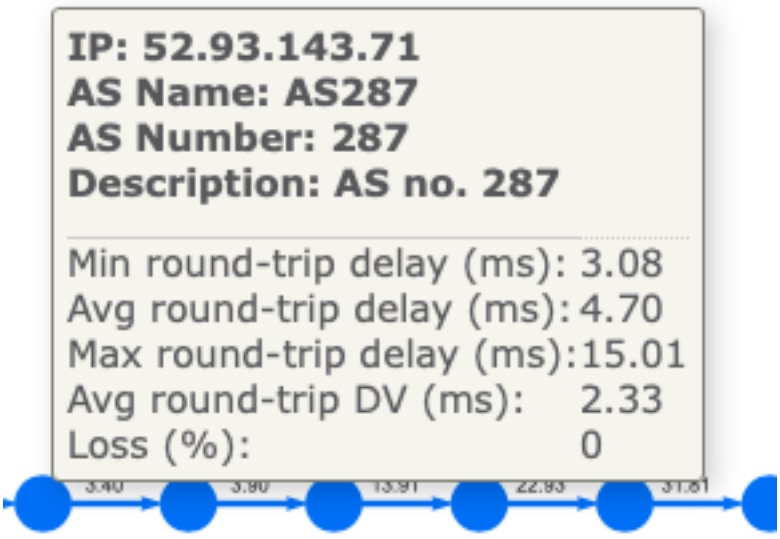
Clicking a router displays detailed data for the hop terminating at that router.

The Results List tab gives a listing of the router hops with errored second bars and metrics in a familiar Routing Active Testing format. See *below* (page 447).

Router tooltip

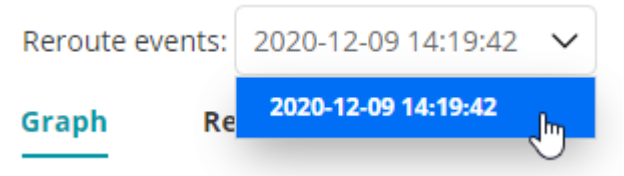
When you hover over a router in the graph, a tooltip appears showing

- the router IP address
- the router domain name
- the name and number of the AS (autonomous system) to which this router belongs (requires that an *AS inventory* (page 31) has been created in the account settings)
- for that router, all the metrics that can be selected for display in the graph (see *this section* (page 445)).

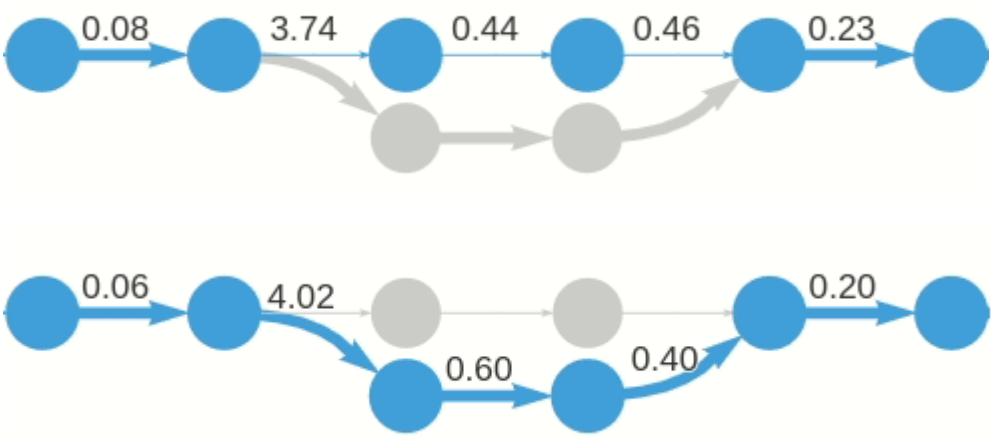


Reroute events box

This drop-down contains all reroute events that have occurred during the Path trace session. The events are listed in reverse chronological order with the most recent on top.



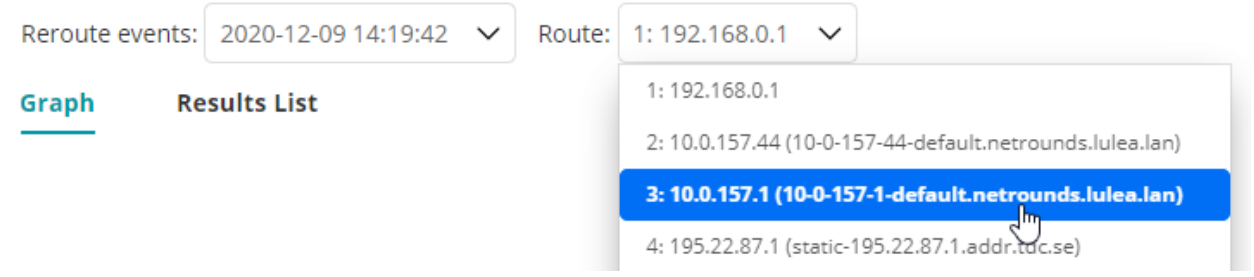
When you select a reroute event, the corresponding route in the graph is redrawn in blue (if it was not blue already). By contrast, routers and edges that are not part of the route currently selected are drawn in gray.



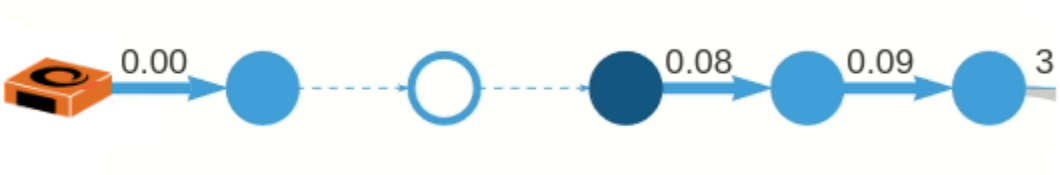
If you click a gray route segment, that segment will be redrawn in blue, and the Reroute events box will switch to the latest reroute event causing this segment to be used.

Route box

This drop-down lists the IP addresses and the DNS names (where resolved) of all intermediate routers that are part of the route currently selected.



When you select an IP address, the corresponding router in the graph is highlighted in dark blue. Conversely, if you click a router in the graph, the Route box is updated to show the IP address and DNS name of that router.



If the IP address of a router could not be determined, it is represented by an asterisk (*) in the drop-down box. In the graph, the router appears as an empty circle (see the above screenshot for an example). The edges adjacent to this router will be dashed and will not be labeled with any metrics. Clicking such edges has no effect, as there is no data to inspect.

Metric box

Here you select what metric to display in the graph. It is one of the following:

- Minimum round-trip delay (ms)
- Maximum round-trip delay (ms)
- Average round-trip delay (ms)
- Average round-trip DV (ms)
- Loss (%)

All of these are calculated over the time interval between the currently selected reroute event and the reroute event following it. The delay variation is computed at short intervals as described *here* (page 474), and an average is taken over these values.

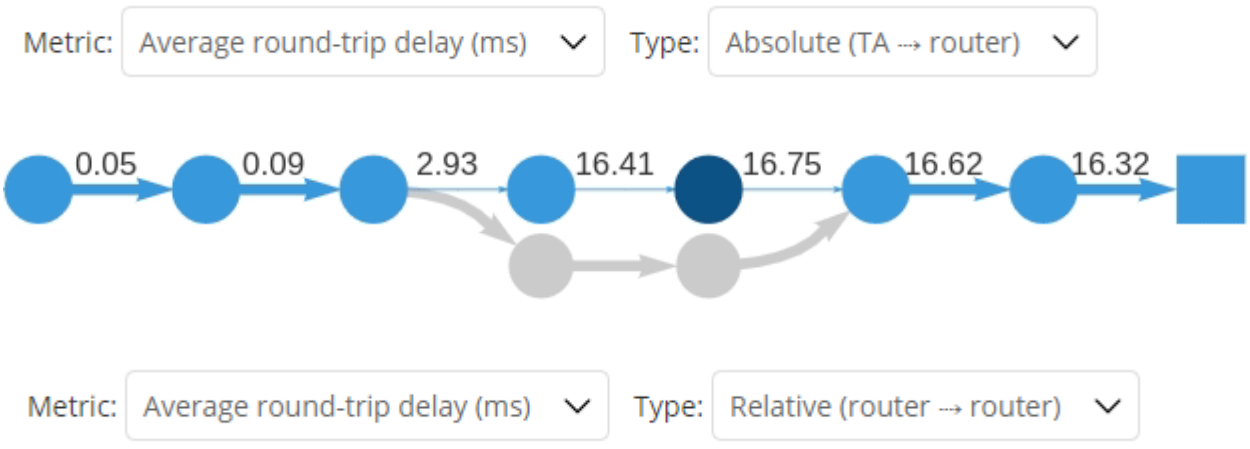
For a monitor, the inter-event time interval might be cut short by the Time interval display setting, for example, if you set the latter to 15 minutes.

Type box

The Path trace metrics can be calculated in two ways.

- *Absolute:* The metric shown for an edge in the graph is calculated all the way from the client (Test Agent) to the router where the edge terminates.
- *Relative:* The metric shown for an edge in the graph is calculated for that hop, that is, between the two routers it connects.

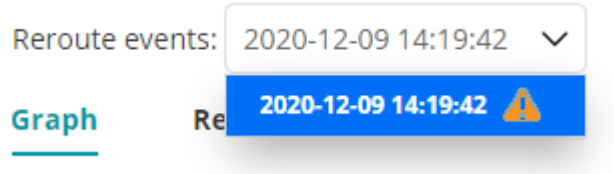
An example of each is shown below.





Presentation of failed routes

If a route has failed to reach its destination, that is, Host, the relevant reroute event is labeled with a warning icon:



In the graph, a warning icon is used to represent the destination:



Handling of unresponsive hops

When a hop stops responding (that is, no more packets are received from a router at a given distance = TTL), metrics are calculated for 60 more seconds for that hop. After this time, no more metrics are calculated. Later on, if and when a response is received from a router with the same address, the metrics calculation is resumed, and the route stays the same. If instead a different router responds at the given TTL, a route change is reported.

Modifying the appearance of the graph; graph controls

The graph can be panned and zoomed with the mouse in the usual way. Routers can also be rearranged by dragging. You can revert to the default appearance at any time by clicking the reset button; see below.

At top right are found the following buttons:



- *Reset graph:* This button restores the graph's default pan and zoom settings and moves all routers back to their default positions. The selections in the drop-down boxes and the graph contents dictated by these selections are not affected.



- *Full screen:* This button zooms the graph to full screen mode. Click the button once more to exit full screen mode.

Results list

The Results List tab shows an errored second bar and metrics for each router hop in a format familiar from elsewhere in Routing Active Testing.

- You select a route to present by making a selection in the Reroute events box.
- Selecting an item under Route will highlight that row in the list.
- The Metric box is disabled, since all available metrics are shown in the list.
- Under Type, you can choose between absolute and relative metrics just as in the graph. Read more *here* (page 445).

These metrics are displayed:

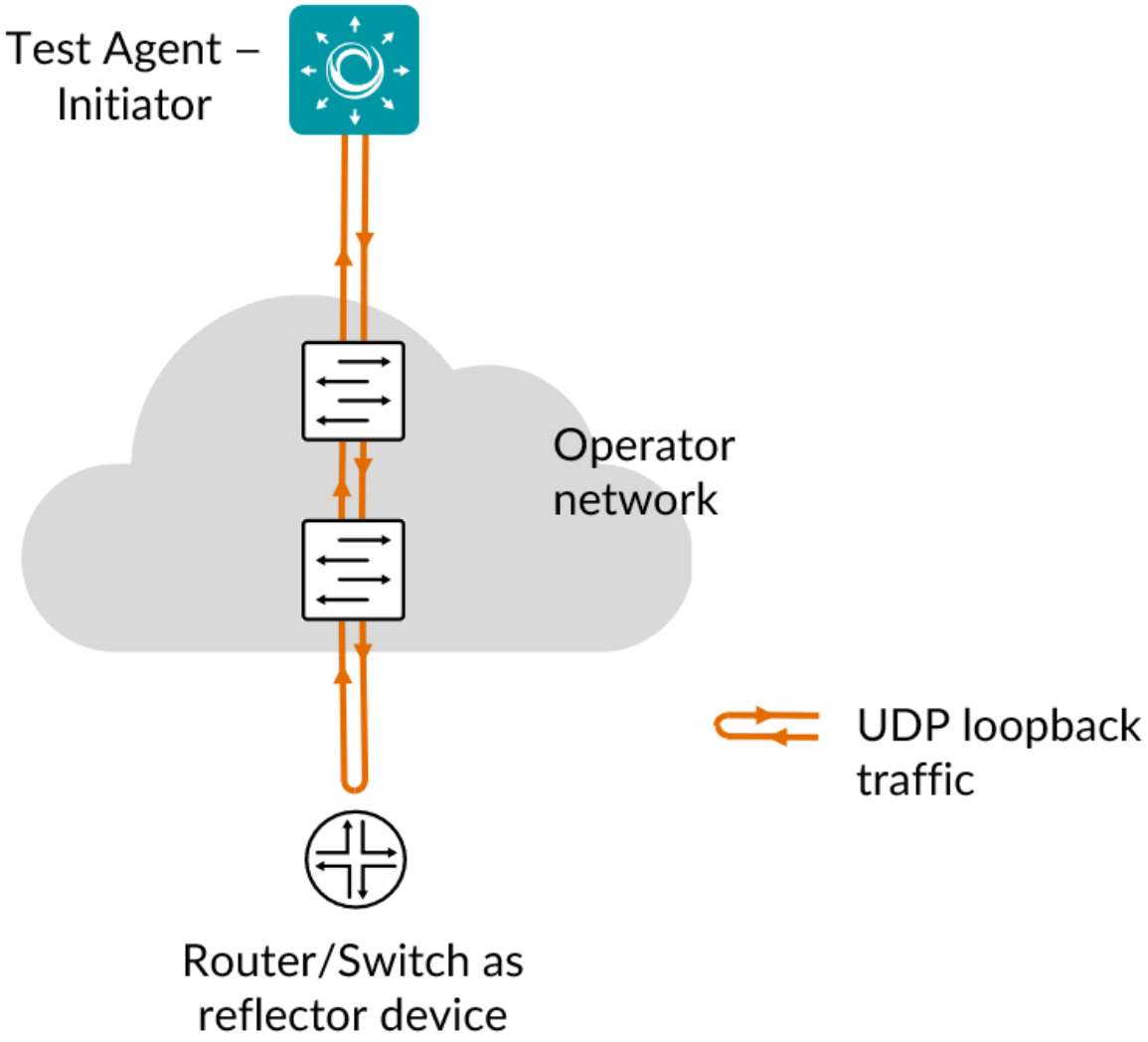
- **Round-trip delay, minimum/average/maximum (ms)**
- **Average round-trip DV (ms)**
- **Loss (%)**

For details on how these metrics are calculated, see the *Metric box* (page 445) section.



8.12.13 UDP loopback

This task pushes UDP packets from a Test Agent towards a network device (router/switch) acting as reflector. The reflector device loops each UDP packet back to the Test Agent.



The primary purpose of the UDP loopback task is to achieve very high data throughput in testing. To enable this, the reflector device needs to be configured in either of two special ways so that any UDP traffic can be looped back in hardware (in the forwarding plane). The possibilities are described *below* (page 449).

Note: Either of these special reflector configurations is *required* for the UDP loopback task to work.

With such a configuration in place, the reflector device does not create a new packet to send back; rather, it forwards the same packet it has received back to the Test Agent originating it. This is different from the procedures for *TWAMP* (page 413) or *UDP echo* (page 430), where the reflector does create a new packet which it sends back to the Test Agent.

This task works with IPv4 as well as IPv6.

8.12.13.1 Prerequisites

To perform UDP loopback measurements, you need to install at least one Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found *here* (page 70).

You also need to have a device (router or switch) available in your network that can serve as reflector of UDP packets.

Reflector configuration

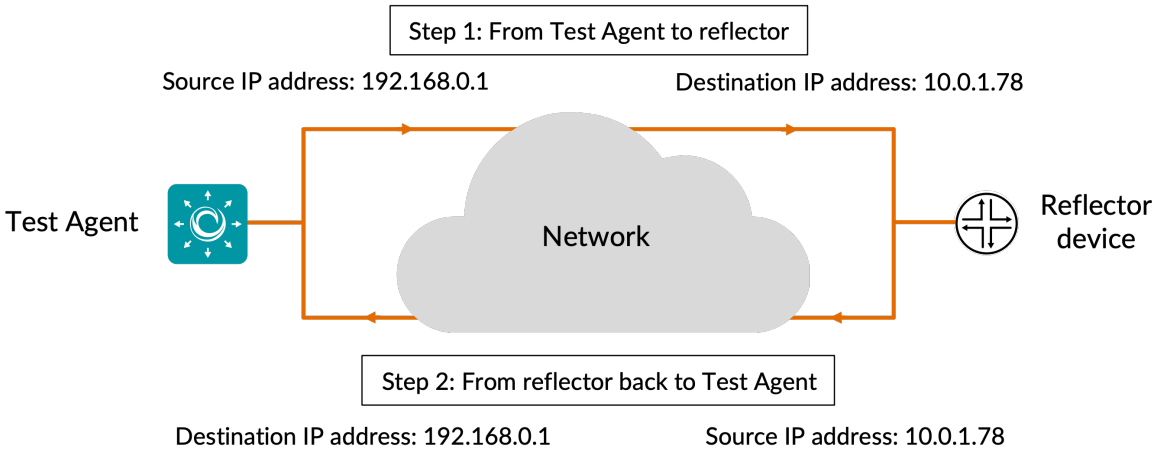
Either IP loopback configuration or NAT (network address translation) is required on the reflector device in order for that device to reflect the UDP traffic back to the Test Agent.

The use of Cisco equipment is assumed in the description that follows. Equipment from other manufacturers needs to be configured correspondingly.

Option 1: IP loopback configuration

Many types of network devices, such as routers, switches and smart SFPs, can be configured to receive an IP packet with a combination of source and destination IP address, switch the two addresses, and send back the modified IP packet.

The diagram below illustrates how the IP addressing in the packet changes in the course of transferring the packet from the Test Agent to the reflector device and back again.



In a Cisco router, configure reflector as follows:

```
! Specify the SLA ID to start the IP SLA session.
ip sla 1

! Specify the service performance type as IP and the destination IP address.
! Specify the target for the SLA session. The options are: service instance,␣
↪interface,
! vrf, and bridge-domain.
service-performance type ip dest-ip-addr 10.0.1.78 interface gi0/0/0 service instance␣
↪1

! Specify the number of interactions and the delay between iterations.
frequency iteration 1 delay 1

! Configure the loopback direction.
```

(continues on next page)

(continued from previous page)

```
loopback direction internal

! Specify the packet profile, defining the packets to be generated.
profile packet
! Specify the source IP address.
source-ip-addr 192.168.0.1
! Specify the VLAN ID that is populated in the outer VLAN tag of the packet.
outer-vlan 301
! Specify the period of time for which to send packets.
duration time 30000
```

In a Juniper router, configure reflector as follows:

```
# Start configuration editing
edit
set services monitoring rfc2544 tests test-name ref-ip mode reflect
set services monitoring rfc2544 tests test-name ref-ip family inet
# Specify the target inteface
set services monitoring rfc2544 tests test-name ref-ip test-interface et-0/0/0.0
# Specify the UDP port of the destination to be used in the UDP header for the␣
↪generated frames
set services monitoring rfc2544 tests test-name ref-ip source-udp-port 7777
# Specify the UDP port of the source to be used in the UDP header for the generated␣
↪frames
set services monitoring rfc2544 tests test-name ref-ip destination-udp-port 7777
# Specify the destination IPv4 address to be used in generated test frames
set services monitoring rfc2544 tests test-name ref-ip destination-ipv4-address 192.
↪168.0.1
# Specify the source IPv4 address to be used in generated test frames
set services monitoring rfc2544 tests test-name ref-ip source-ipv4-address 10.0.1.78
set services monitoring rfc2544 tests test-name ref-ip disable-signature-check
commit
# Exit from editing
exit
# Start the test
test services monitoring rfc2544 test ref-ip start
```

Option 2: NAT configuration

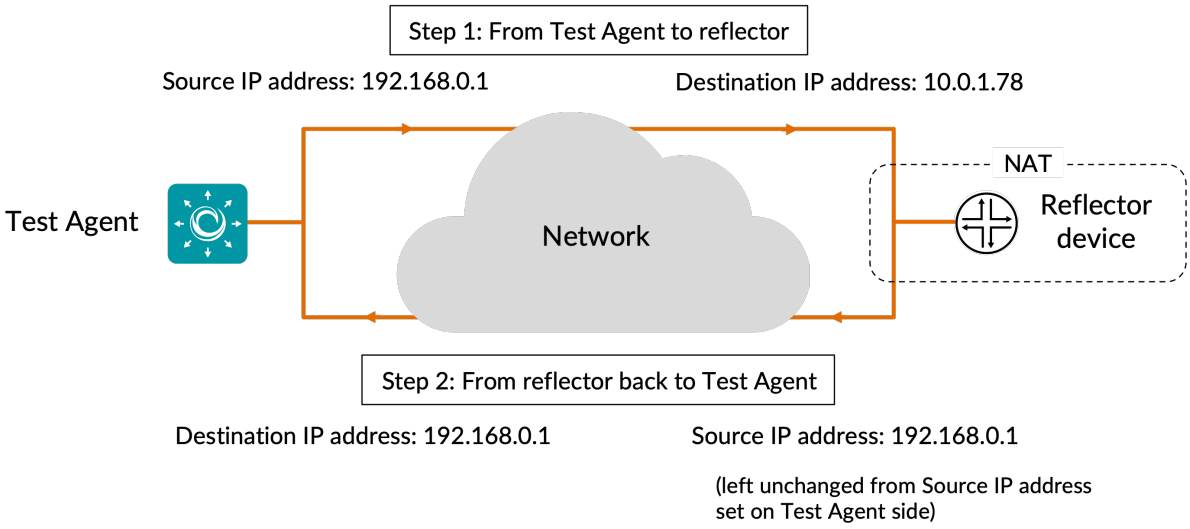
In a Cisco router, the network address translation takes place prior to packet forwarding. The WAN interface where the traffic enters the router (`GigabitEthernet X` in the example *below* (page 449)) needs to be defined as the NAT "outside" interface, and a loopback interface (`loopback Y` below) needs to be defined as the NAT "inside" interface. Network address translation is then set up using the `ip nat inside` command.

With this NAT setup, the following happens:

- The source IP address in the packet sent from Test Agent to reflector is retained as source IP address when the packet is reflected back to the Test Agent.
- The destination IP address in the packet sent from Test Agent to reflector is exchanged (according to the IP NAT table) for the Test Agent IP address when the packet is reflected back to the Test Agent.

In the reflection step, therefore, the source and destination addresses are the same.

The diagram that follows illustrates how the IP addressing in the packet changes in the course of transferring the packet from the Test Agent to the reflector device and back again.



Below is shown how to do the NAT configuration, which is mandatory:

```
! Create a "loopback Y" interface with an IP address and define it as "inside".
! In the above example, this IP address is 10.0.1.78 (i.e. reflector address)
interface loopback Y
     ip add <loopback interface IP address> 255.255.255.255
     ip nat inside

! Define the WAN interface as "outside"
interface GigabitEthernet X
     ip nat outside

! Create a NAT entry which translates the loopback address into the Test Agent's IP␣
↪address.
! This redirects traffic with the loopback address as destination to the Test Agent.
! In the above example, the Test Agent has IP address 192.168.0.1
ip nat inside source static <Test Agent IP address> <loopback interface IP address>
```

Routing configuration (always needed)

Routing configuration requirements are more variable, as explained further in the comments below. Announcing the loopback address in some way is however mandatory.

- In the case of IP loopback, the loopback address is the IP address of the interface where the IP loopback functionality has been turned on.
- In the NAT case, the loopback address is equal to the address of the loopback interface (named "Y" in the example above).

The configuration may look something like this:

```
! Tell the router how to send traffic to the Test Agent.
! This can be skipped if the Test Agent IP address is already available in routing␣
↪tables.
! In the above example, <next-hop IP address> is "GigabitEthernet X"
ip route <Test Agent IP address> 255.255.255.255 <next-hop IP address>

! For the Test Agent to reach the loopback address, the latter must be announced in␣
↪the
```

(continues on next page)

(continued from previous page)

```
! routing. Below, this is done via BGP (Border Gateway Protocol).

! Enter into BGP configuration with relevant AS (autonomous system) number
router bgp <AS number>
     ! Add loopback address to BGP to have it announced via this protocol
     network <loopback interface IP address> mask 255.255.255.255
```

When done configuring the router, add a UDP loopback task to your test or monitor, and fill in the mandatory parameters as shown below.

8.12.13.2 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Sender: Test Agent interface that will act as sender (and receiver) of UDP packets.
- Host: IP address of device acting as reflector. This is the address of the "outside" interface referred to in the *Reflector configuration* (page 449) section.
- UDP port: UDP destination port on reflector device. The same port will be used as source port on the Test Agent. Note: This port must not be selected in any other test or monitor task using the same Test Agent interface. If you are setting up multiple UDP loopback tasks to run on the same Test Agent interface, use a different UDP port in each task.
- Rate (Mbit/s): Rate at which the sender will send Ethernet frames in Mbit/s. The value is calculated based on packet rate and frame size. Each Ethernet packet contains one frame. Max: 10,000 Mbit/s. No default.
- Rate (packets/s): Number of Ethernet frames the sender will send each second. Each Ethernet packet contains one frame. Minimum and maximum values correspond to those for Rate (Mbit/s) and depend on the Frame size setting. No default.
- Frame size (bytes): Size of Layer 2 Ethernet frame for the flow. See *this page* (page 512). Min: 64 bytes for IPv4, 84 bytes for IPv6. Max: 9018 bytes. Default: 1518 bytes. If you change this setting, the Rate setting last edited will be kept constant, and the other will be adjusted automatically.

Changing one Rate parameter will cause the other to adjust automatically to agree with it.

Thresholds for errored seconds (ES)

- Rate (Mbit/s): Ethernet rate threshold for triggering an errored second. If the rate goes below this value during one second, an ES will be indicated. Max: 10,000 Mbit/s. No default.
- Loss (%): Round-trip packet loss threshold for triggering an errored second. If the loss exceeds this value during one second, an ES will be indicated. Min: 0%. Max: 100%. Default: 0%.
- Delay (ms): Round-trip delay threshold for triggering an errored second. If the round-trip delay exceeds this value during one second, an ES will be indicated. Min: 1 ms. Max: 1000 ms. No default.
- Delay variation (ms): Round-trip jitter threshold for triggering an errored second. If the round-trip *jitter (delay variation)* (page 474) exceeds this value during one second, an ES will be indicated. Min: 1 ms. Max: 1000 ms. No default.
- Expected DSCP: The *Differentiated Services Code Point or IP Precedence* (page 511) that IP packets are expected to have on being received from the reflector device. If the received DSCP value does not match this, an ES will be indicated. By default, no DSCP validation is done (------ selected in drop-down box).

Thresholds for severely errored seconds (SES)

- Loss (%): Packet loss threshold for triggering a *severely errored second* (page 477). Min: 0%. No default.
- Delay (ms): Delay threshold for triggering a severely errored second. Min: 1 ms. No default.
- Delay variation (ms): Delay variation threshold for triggering a severely errored second. Min: 1 ms. No default.

Advanced

- DSCP: Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 511). The available choices are listed in the drop-down box. Default: “0 / IPP 0”.
- VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See *this page* (page 516). Min: 0. Max: 7. Default: 0.

8.12.13.3 Result metrics

- **Received packets:** Number of Ethernet packets received.
- **Rate (Mbit/s):** Received Ethernet packet rate.
- **Min round-trip delay (ms):** Minimum round-trip delay.
- **Average round-trip delay (ms):** Average round-trip delay.
- **Max round-trip delay (ms):** Maximum round-trip delay.
- **Average round-trip DV (ms):** Average round-trip delay variation.
- **Lost packets:** Number of lost packets.
- **Loss (%):** Packet loss in percent.
- **Misorders:** Number of packet misorderings.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **ES rate (%):** Accumulated errored second percentage for received rate.

- **ES loss (%):** Accumulated errored second percentage for packet loss.
- **ES delay (%):** Accumulated errored second percentage for round-trip delay.
- **ES delay variation (%):** Accumulated errored second percentage for round-trip delay variation.
- **ES DSCP (%):** Accumulated errored second percentage for DSCP.
- **SES (%):** Aggregated severely errored second (SES) percentage, taking into account all types of error.
- **Unavailable seconds (%):** *Unavailable second (UAS)* (page 477) percentage.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

8.13 Security testing

8.13.1 Introduction to security testing

The security tests in Routing Active Testing are primarily designed for Layer 3 networks. However, all issues should be tested independently of what type of network design or topology is used. Our experience shows that the potential for configuration errors exists in any network and can give rise to security issues. Most of the security issues tested in Routing Active Testing cannot be mitigated by end-users; it is the network that must provide protection.

The Routing Active Testing security tests focus mainly on:

- **Man-in-the-middle (MITM) attacks:** The ability to eavesdrop and possibly change traffic without the customer being aware of it.
- **Denial-of-service (DoS) attacks:** The ability for one customer to affect the services of one or several other customers.
- **Abuse – Tracking of end-users (IP addresses):** The ability to identify a customer if there has been some incorrect usage of the services.

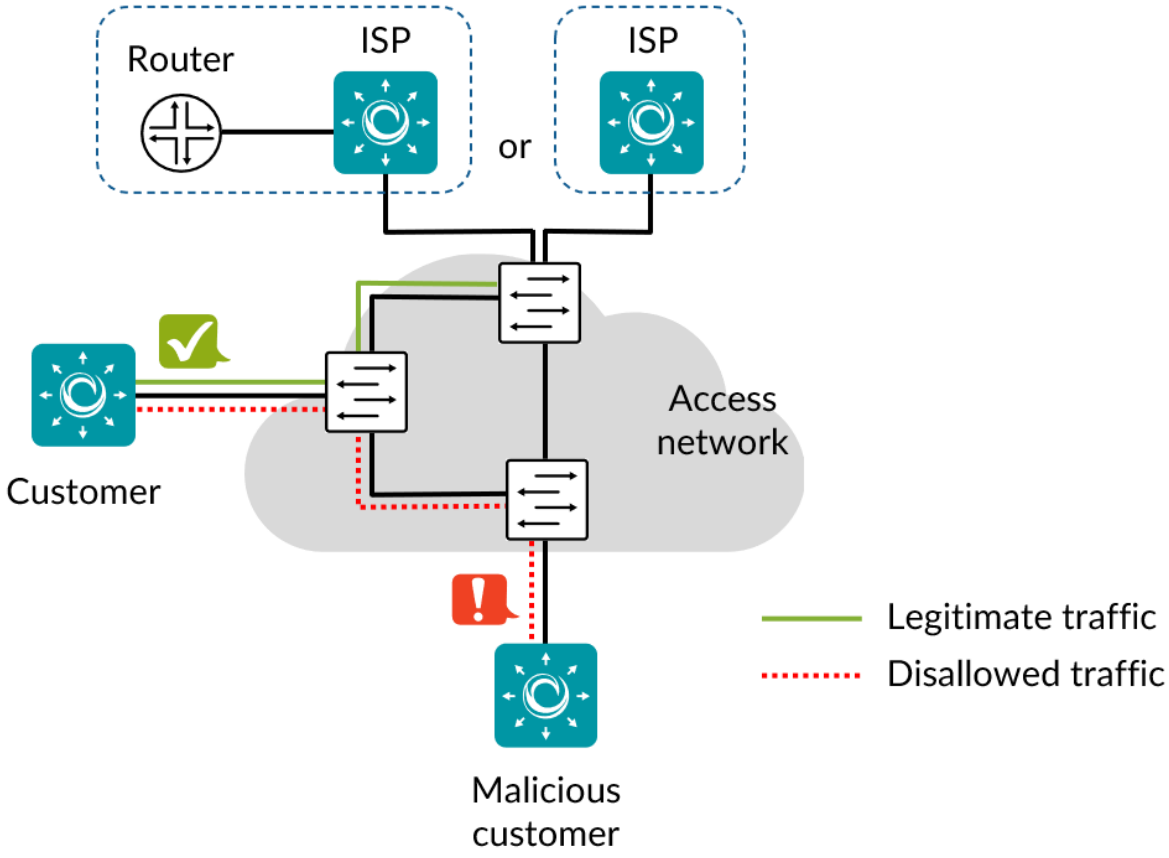
Routing Active Testing supports security testing on the IPv4 protocol.

To perform security tests, one or two Test Agents are needed. Two interfaces must be used on each Test Agent: one agent interface is used for testing, and the other ("eth0") is used to maintain the encrypted management connection to the Routing Active Testing cloud servers.

Test Agents play one of the following roles in security tests:

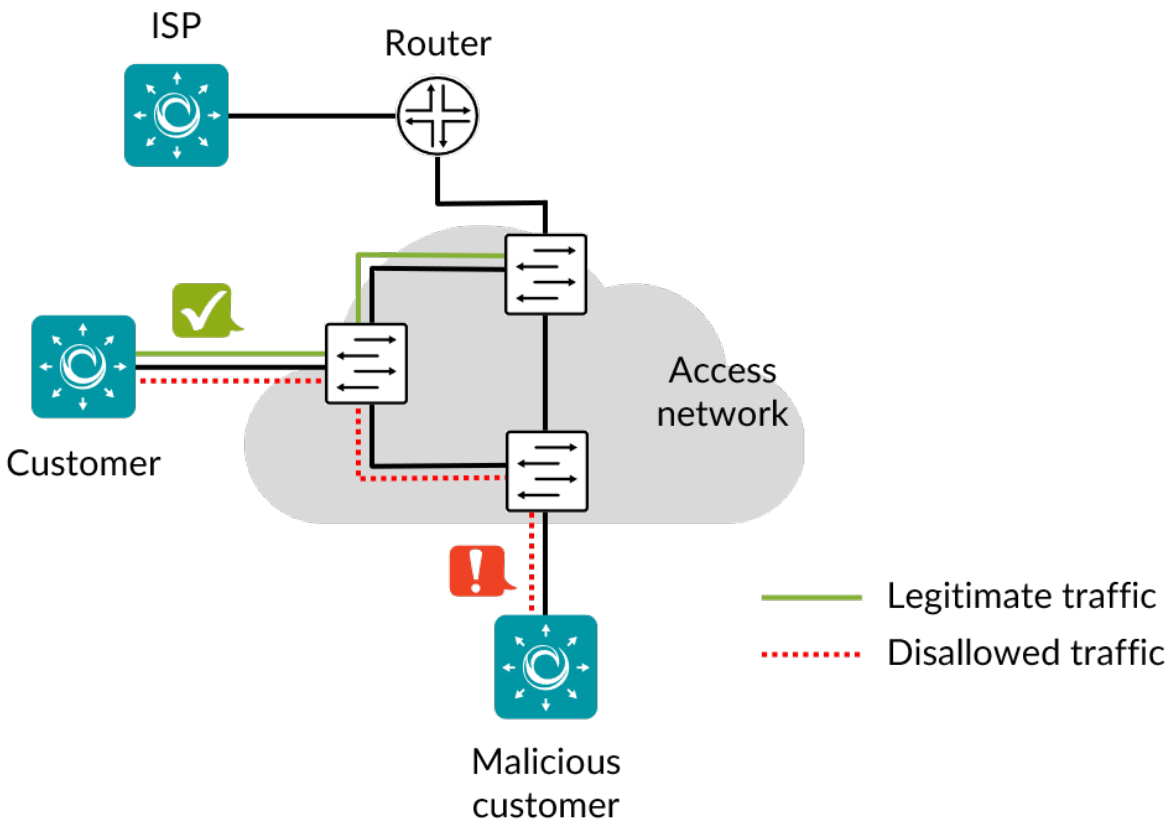
- **Customer:** A customer attempting a particular operation. Each Test Agent interface playing this role is connected as a standard customer to an access port.
- **ISP:** Internet Service Provider, placed in a trusted zone. Only one Test Agent interface plays this role.

The picture below shows an example of a test configuration.



It is most convenient to place the ISP Test Agent in the same Layer 2 network as customers, as this is required for some of the tests (though not for all of them; the requirement is noted for each task type to which it applies). This setup makes it possible to run all security tests using the same network configuration.

It is also possible to place the ISP Test Agent in the Layer 3 network, as depicted below.



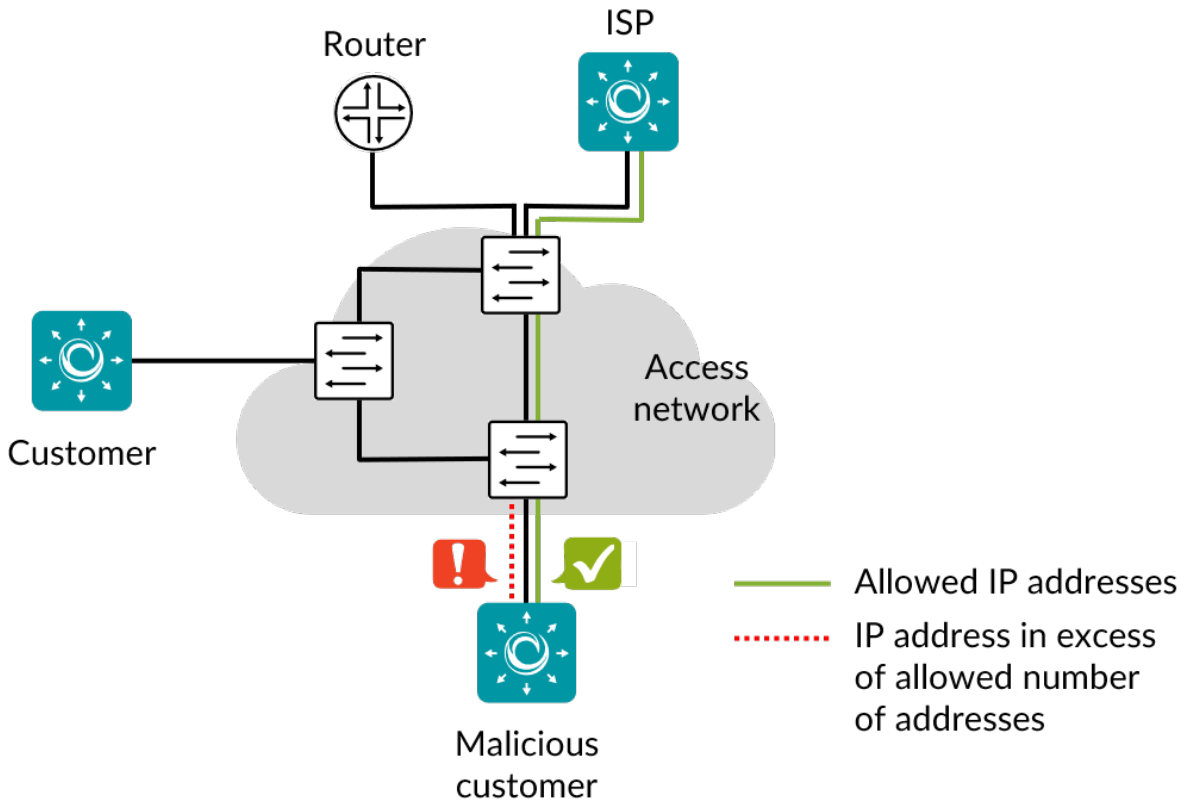
The description of each task type has a reference to the corresponding identification in SAVI.

A few of the tests require a DHCP server and a multicast sender, neither of which is normally provided by the ISP Test Agent.

- The DHCP server requirement can be met by *setting up the ISP Test Agent as a DHCP server* (page 173), or by using an external DHCP server connected to the network.
- A multicast sender can either distribute real multicast traffic, or it can consist of a lab setup. A multicast source must be available in the network for tests involving multicast/IGMP.

See *this page* (page 454) for an overview of all supported security features.

8.13.2 DHCP starvation



DHCP starvation is an attack that works by broadcasting vast numbers of DHCP requests with spoofed MAC addresses simultaneously, exhausting the DHCP server IP pool. This task checks that a customer can only obtain a limited number of IPv4 addresses, so that DHCP starvation is prevented. Customer takes the allowed number of addresses, then verifies that it cannot get one more.

The test will not detect if an old address is released.

A DHCP server is required for the DHCP starvation test.

8.13.2.1 References

The test performed conforms to SAVI section 3.1.2.

8.13.2.2 Impact

DoS

8.13.2.3 Test procedure

1. Customer verifies connectivity to ISP.
2. Customer takes the allowed number of IPv4 addresses.
3. Customer then sends another DHCP request.

8.13.2.4 Fail criteria

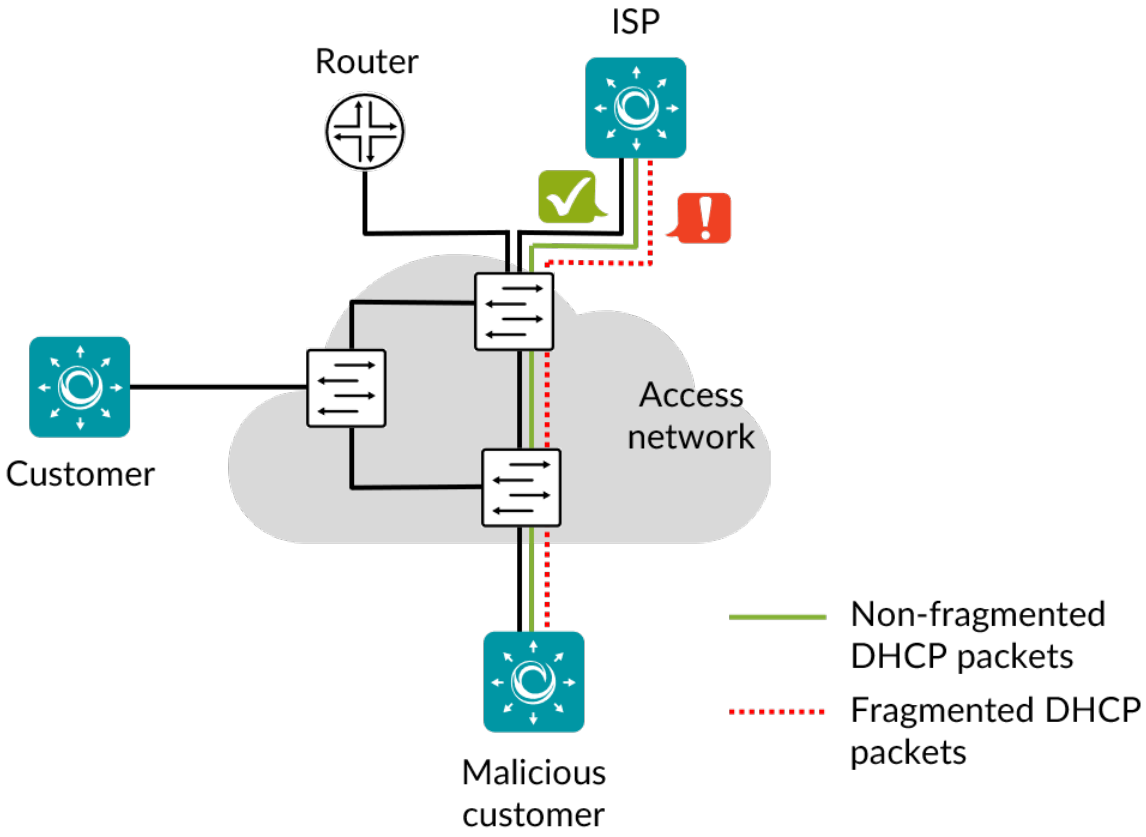
- Customer cannot obtain the allowed number of IPv4 addresses.
- Customer can obtain more than the allowed number of IPv4 addresses.

8.13.2.5 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- ISP: A Test Agent interface acting as a central node on a trusted port.
- Max addresses: The maximum number of IPv4 addresses a customer is allowed to hold. Default: 3.

8.13.3 Fragmented DHCP packets



This task checks that the switch drops fragmented DHCP packets before they reach the control plane. If fragmented packets are not dropped, they will consume resources at the switch's control plane upon reassembly. This fact can be exploited to launch a DoS attack causing the CPU to run out of cycles or filling up the packet buffers.

Since the control plane is normally in a controlled environment, the MTU is known. There is therefore no reason for packets to be fragmented, nor for packet reassembly to be needed.

8.13.3.1 References

The test performed conforms to SAVI section 3.1.2.

8.13.3.2 Impact

DoS

8.13.3.3 Test procedure

1. Customer sends a valid DHCP packet.
2. Customer sends DHCP packets fragmented into 40 byte and 104 byte fragments.

8.13.3.4 Fail criteria

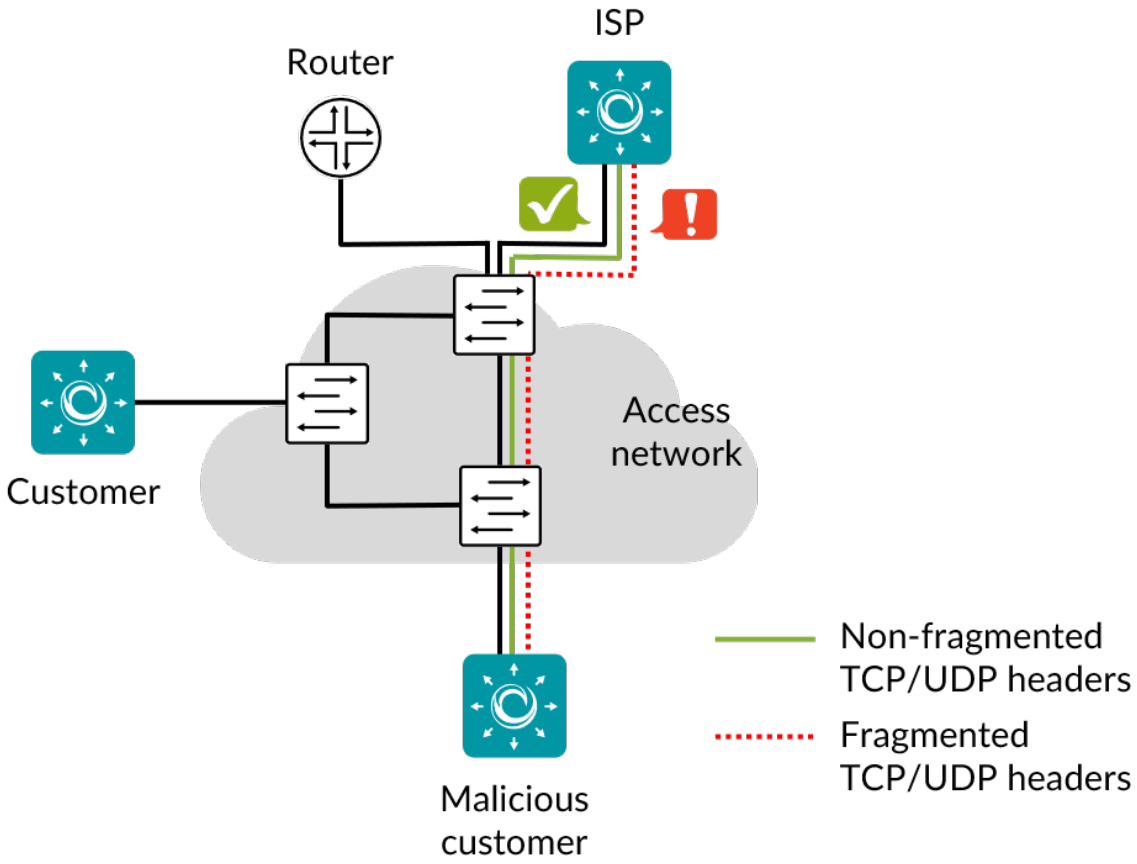
- ISP does not receive the valid DHCP packet.
- ISP receives any fragment of a fragmented packet.

8.13.3.5 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- ISP: A Test Agent interface acting as a central node on a trusted port. The test requires that the ISP reside in the same Layer 2 network as the customers.

8.13.4 Fragmented TCP/UDP headers



This task checks that the switch drops IPv4 and IPv6 packets with fragmented headers. By fragmenting TCP or UDP headers it is possible to bypass access lists which are based on information in those headers. The test verifies that packets with a small fragment offset are blocked/dropped.

8.13.4.1 References

The test performed conforms to SAVI section 3.1.2.

8.13.4.2 Impact

DoS, Abuse, Illegal access to content

8.13.4.3 Test procedure

1. Customer sends non-fragmented TCP and UDP packets to ISP.
2. Customer sends similar packets fragmented into 8 and 16 byte fragments.

8.13.4.4 Fail criteria

- ISP does not receive the non-fragmented packets.
- ISP receives all fragments of a fragmented packet.

8.13.4.5 Parameters

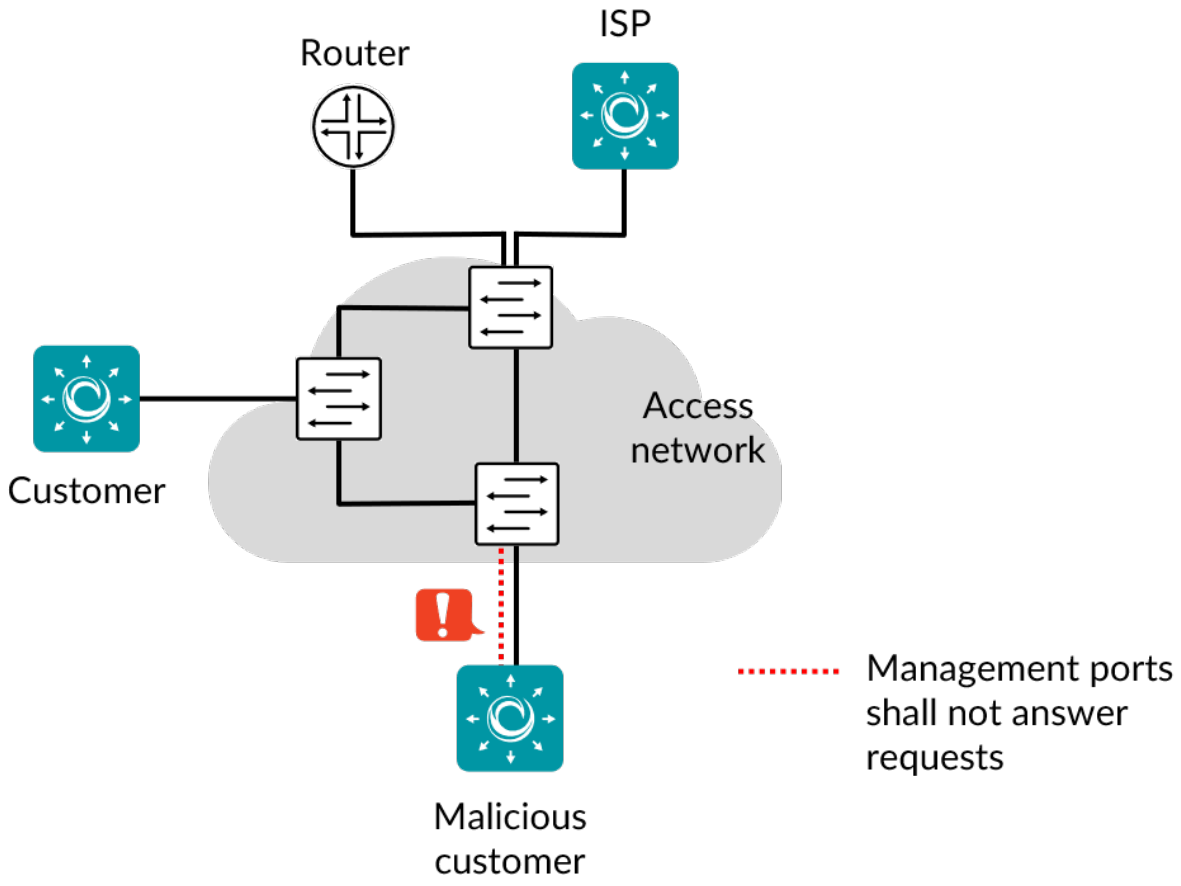
General

- Customer: A Test Agent interface acting as a customer.
- ISP: A Test Agent interface acting as a central node on a trusted port.

Advanced

- Source UDP/TCP port: Source UDP or TCP port for traffic sent from Customer to ISP. Range: 1 … 65535. Default: 41234.
- Destination UDP/TCP port: Destination UDP/TCP port for traffic sent from Customer to ISP. Range: 1 … 65535. Default: 24567.

8.13.5 Management protocol scanning



This task checks that management protocols are unavailable at customer ports and that users are prevented from interfering with equipment management. Network equipment must ignore incoming management traffic from customer ports.

8.13.5.1 References

The test performed conforms to SAVI section 3.1.7.

8.13.5.2 Impact

MITM, DoS, Abuse

8.13.5.3 Test procedure

1. Customer runs a TCP SYN scan for all addresses on standard ports for FTP, SSH, Telnet, HTTP, and HTTPS.
2. Customer attempts an SNMP Get, a Ping Request, and an NTP Get for all management addresses.

8.13.5.4 Fail criteria

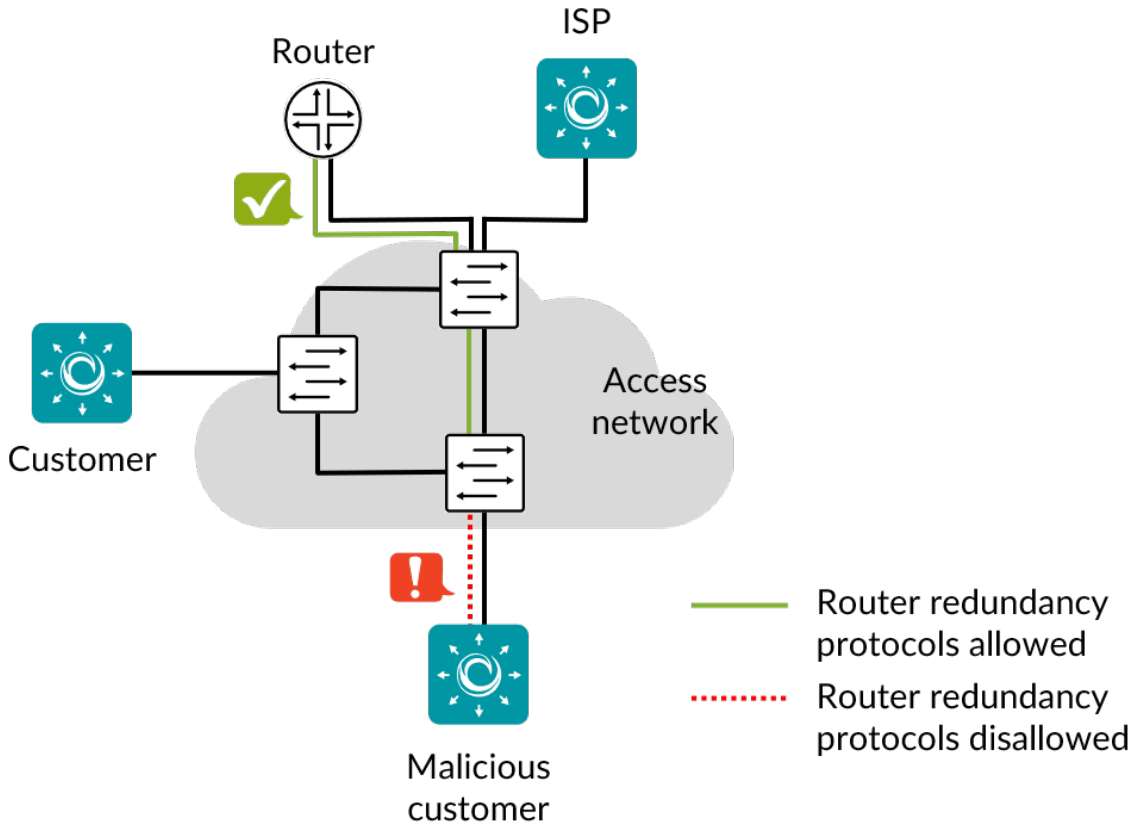
- One of the TCP ports is listening for traffic.
- Customer receives an answer to an SNMP Get, Ping Request, or NTP Get.

8.13.5.5 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- Management IPs: IP addresses used to manage equipment, separated by commas.

8.13.6 Router redundancy protocol listening



This task checks that router redundancy protocols are unavailable at customer ports. If such protocols are available, a malicious customer can sniff the protocols and then force other customers to point their default traffic route to the malicious customer, thus launching a man-in-the-middle (MITM) attack.

Note: For this task, routers must be present in the network. The test cannot be performed against switches only.

Tested protocols:

- VRRP/CARP
- GLBP
- HSRP

8.13.6.1 References

The test performed conforms to SAVI section 3.1.7.

8.13.6.2 Impact

MITM, DoS

8.13.6.3 Test procedure

Customer listens during 60 seconds for traffic on each of the above protocols.

8.13.6.4 Fail criteria

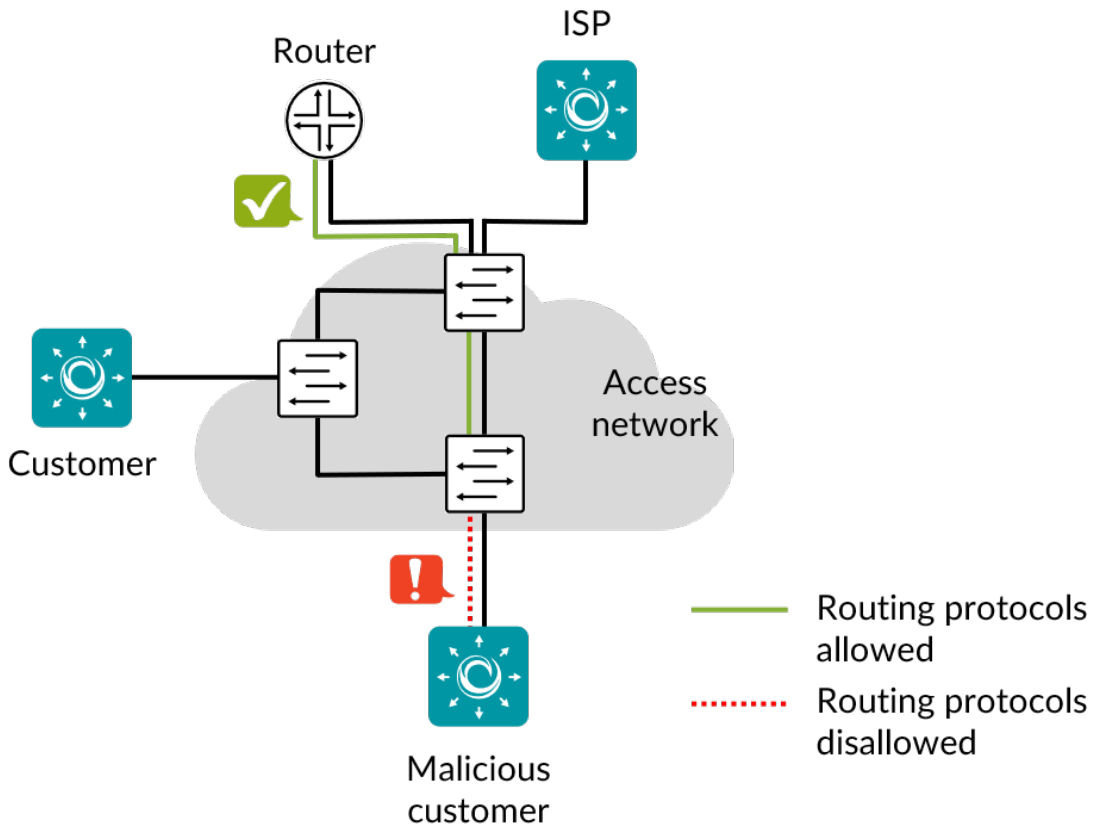
- A packet from any router redundancy protocol is received by Customer.

8.13.6.5 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- ISP: A Test Agent interface acting as a central node on a trusted port.

8.13.7 Routing protocols



This task checks that routing protocols are not available on customer ports. If such protocols are available, malicious customers can interfere with the router signaling and launch MITM and DoS attacks using the routing protocols.

Note: This task requires routers in the network. The test cannot be performed against switches only.

Tested protocols:

- BGP
- EIGRP
- IGRP
- IS-IS
- OSPF
- RIPv2

8.13.7.1 References

The test performed conforms to SAVI section 3.1.7.

8.13.7.2 Impact

MITM, DoS

8.13.7.3 Test procedure

Customer sends multicast join messages used by the above routing protocols and then listens during 60 seconds for traffic on each protocol.

8.13.7.4 Fail criteria

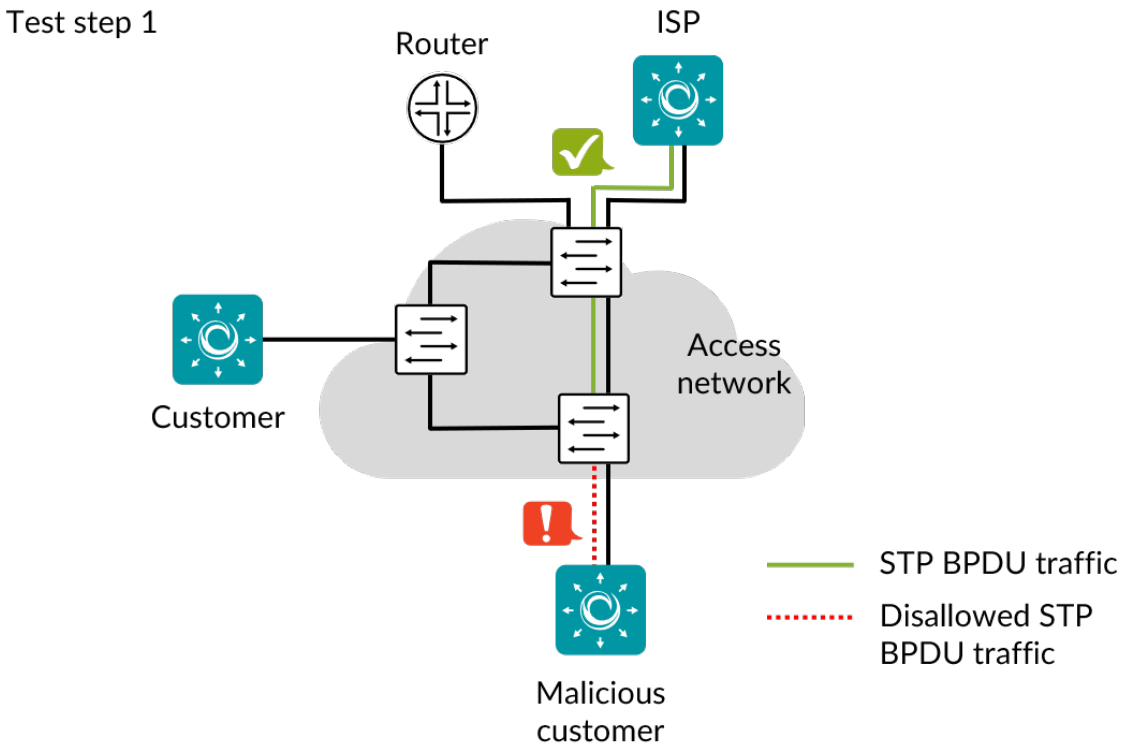
- A packet from any routing protocol is received at Customer.

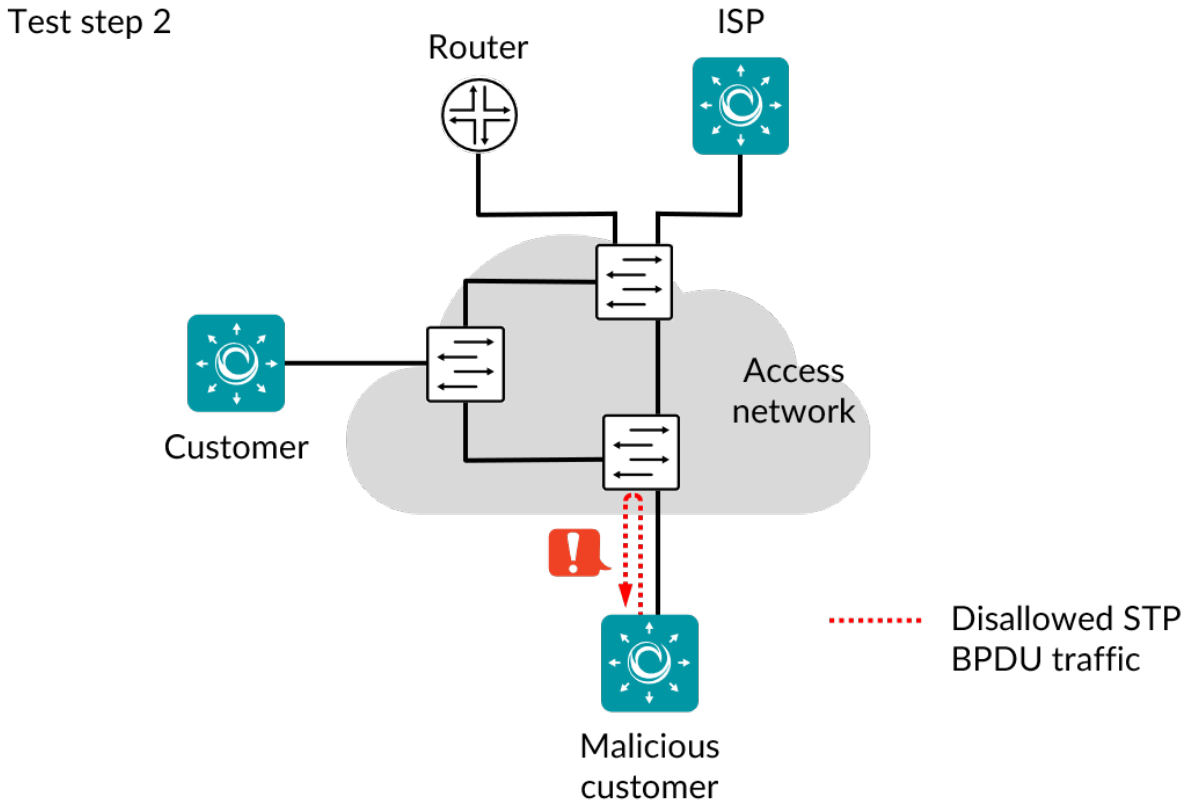
8.13.7.5 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- ISP: A Test Agent interface acting as a central node on a trusted port.

8.13.8 STP – Spanning Tree Protocol





This task checks that the Spanning Tree Protocol (STP) is not available on customer ports. If available, this protocol could be used to perform various attacks in the network, such as redirecting traffic or overloading devices.

No spanning-tree packets should be sent out on customer ports, and any spanning-tree packets received should be silently discarded.

8.13.8.1 References

The test performed conforms to SAVI section 3.2.3.

8.13.8.2 Impact

DoS, MITM

8.13.8.3 Test procedure

1. Customer listens for BPDU packets.
2. Customer sends BPDU packets (on STP, RSTP, PVST, and MSTP) and keeps listening on the interface if the switch responds.

8.13.8.4 Fail criteria

- An STP BPDU packet arrives at Customer.

8.13.8.5 Parameters

General

- Customer: A Test Agent interface acting as a customer.
- ISP: A Test Agent interface acting as a central node on a trusted port.

8.14 Utilities for testing

8.14.1 Delay

This task is a utility that pauses the execution of a test, delaying the test step that follows it by a configurable amount of time.

Delays are applicable only to tests, not to monitoring sessions.

8.14.1.1 Parameters

- Sleep time (s): Here you specify for how long to pause the test. Min: 1 s. Max: 500 s. Default: 60 s.
- Wait for ready: Time to wait before starting this test step. The purpose of inserting a wait is to allow all Test Agents time to come online and acquire good time sync. Min: 1 min. Max: 24 hours. Default: "Don't wait", i.e. zero wait time.

8.15 Dynamic plugins

8.15.1 Dynamic plugins

A plugin consists of software which Test Agents (currently, Test Agent Applications only) use to collect measurements in a network.

As explained in more detail on *this page* (page 242), Control Center is delivered with all plugins needed by Test Agents, and by default you will not see any tasks in the gui:*Dynamic plugins* category.

However, it is also possible to upload further plugins ("dynamically") to Control Center. If this is done, the plugin task will appear in the Dynamic plugins category instead of its usual category (for example, Network performance).

The configuration of a dynamic plugin task is similar to that of the corresponding regular task.

8.15.1.1 Parameters

See the *common parameters page* (page 287) for the following:

- Parameters that are set on the *test step* (page 287) level: Duration, Fail threshold, and Wait for ready.
- *SLA thresholds* (page 288) for *monitors*: SLA Good and SLA Acceptable.
- *Advanced settings* (page 287) common to all *test* tasks: Delayed start.

General

- Clients: Test Agents to use as clients.
- Plugin: Name of the plugin.

The remaining parameters will be unique to each plugin task.

8.15.2 System monitoring

This task monitors the system where the plugin is running, keeping track of (among other things) memory, CPU and disk usage.

System monitoring is a dynamic plugin which is not delivered with Routing Active Testing but is uploaded separately to Control Center.

8.15.2.1 Prerequisites

To run a System monitoring task you need to have at least one Test Agent installed, equipped with the System monitoring plugin. If you haven't already done the installation, consult the installation guides found *here* (page 70).

Then add a System monitoring task to your test or monitor and fill in the mandatory parameters below:

8.15.2.2 Parameters

General

- Clients: Test Agents to use as clients.

Thresholds for errored seconds (ES)

- Maximum memory usage percentage: An errored second will be indicated if the memory usage exceeds this percentage threshold. Default: 100%.
- Maximum CPU usage percentage: An errored second will be indicated if the CPU usage exceeds this percentage threshold. Default: 100%.
- Maximum disk space usage percentage: An errored second will be indicated if the disk space usage exceeds this percentage threshold. Default: 100%.
- Maximum ingress dropped packets: An errored second will be indicated if the number of ingress dropped packets exceeds this threshold. Default: 0.
- Maximum ingress dropped bytes: An errored second will be indicated if the number of ingress dropped bytes exceeds this threshold.` Default: 0.

Advanced

- Measurement period: Number of seconds between consecutive system monitoring measurements. Default: 1 s.
- Disk partition: Disk partition for which to track disk usage percentage. By default, this is the disk partition where the System monitoring plugin is running.

8.15.2.3 Result metrics

- **Memory usage percentage (%):** Percentage of available memory used.
- **CPU usage percentage (%):** Percentage of available CPU used.
- **Disk usage percentage (%):** Percentage of available disk used.
- **ES memory usage (%):** Errored second percentage for memory usage.
- **ES CPU usage (%):** Errored second percentage for CPU usage.
- **ES disk usage (%):** Errored second percentage for disk usage.
- **ES ingress dropped packets (%):** Errored second percentage for ingress dropped packets.
- **ES ingress dropped bytes (%):** Errored second percentage for ingress dropped bytes.
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **SLA:** *Service level agreement* (page 478) fulfillment: equal to (100 – **ES**) %.

9 Metrics in Routing Active Testing

9.1 Introduction

This chapter describes how certain metrics in Routing Active Testing are computed. It is not comprehensive as it does not deal with straightforward metrics such as packet loss and delay. The focus is on metrics that are non-trivial to calculate.

The chapter also goes into some specifics of the raw measurement data.

9.2 Resolution of Routing Active Testing measurement data

Time series of Routing Active Testing measurement data are stored in a round-robin database, where older data is progressively consolidated into lower resolutions.

9.2.1 Resolution for monitoring sessions

In each of your monitoring sessions, the Test Agents periodically collect measurement data in 10-second intervals, then compile the results into individual measurement reports and send them to the Routing Active Testing server. The measurement data can be said to have a “resolution” of 10 seconds.

In the database, the 10-second resolution is retained for data from the last 12 hours. Older data is aggregated as indicated in the following table:

Timeframe	Resolution
12 hours	10 seconds
2 days	1 minute
1 week	5 minutes
1 month	20 minutes
1 year	4 hours

9.2.2 Resolution for tests

Tests in Routing Active Testing use a higher resolution than monitoring sessions. For tests of duration up to 1 hour, the resolution is 1 second; for tests up to 2 hours in length, the resolution is 2 seconds; and so on for longer tests.

9.3 Delay variation (DV), jitter

Delay variation (DV), also termed *jitter*, arises when different packets take a different amount of time to travel from sender to receiver.

The jitter calculation for synthetic traffic in Routing Active Testing follows ► IETF RFC 3393. In short, Routing Active Testing calculates the difference between the maximum and the minimum measured delay within a specific interval, commonly one second.

Since the output of a video stream needs to be continuous, jitter forces a set-top box (STB) to buffer a certain amount of data. The more jitter there is, the more the STB needs to buffer. If the buffer runs empty, or runs full, the effect on IPTV quality will be the same as that of packet loss (pixelation, audio glitches, etc.).

The standard buffering requirement for coping with jitter is 50 ms. Modern STBs are often able to buffer much more than 50 ms of data, and jitter buffers can vary between vendors.

9.4 Mean Opinion Score (MOS)

The metric used for estimating voice quality is a Mean Opinion Score (MOS) based on the ITU-T E-model (► ITU-T Recommendation G.107). The inputs are network statistics such as speech codec usage, network delay, jitter, and packet loss. The modified E-model outputs an R-value, which is straightforwardly converted to a MOS value.

The MOS scale is laid out in the following table:

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible
3	Fair	Annoying
2	Poor	Very annoying
1	Bad	Impossible to communicate

9.4.1 Calculation of MOS in Routing Active Testing

The algorithm for calculating MOS in the VoIP task is given below. It follows the ITU-T E-model.

For SIP, the algorithm is the same, but the average delay (davg) is assumed to be 5 ms in this case.

The function takes the following arguments:

- loss = packet loss in %
- davg = average delay
- dmin = minimum delay
- dmax = maximum delay
- ie = equipment impairment factor
- bpl = packet-loss robustness factor (codec-specific)

```
float Stats::calc_mos(float loss, float davg, float dmin, float dmax, float ie, float␣
↪bpl)
{
    float mos, r, deff;


    deff = davg + 2 * (dmax - dmin) + 10;
    if (deff < 160)
    {
        r = 93.2 - deff / 40;
    }
    else
    {
        r = 93.2 - (deff - 120) / 12;
    }
    r -= ie + (95 - ie) * loss / (loss + bpl);
    if (r < 0)
    {
        mos = 1;
    }
    else
    {
        mos = 1 + 0.035 * r + 0.000007 * r * (r - 60) * (100 - r);
    }
    return mos;
}
```

9.5 Errored Seconds (ES) metric: Method of calculation

This page explains the fine points of how the errored seconds (ES) metric is calculated.

In each of your monitoring sessions, your Test Agents periodically collect measurement data in 10-second intervals, then compile the results into measurement reports and send them to the Routing Active Testing server. The measurement data can be said to have a resolution of 10 seconds.

Now suppose that within one 10-second interval, there were two seconds during which a Test Agent measured a high level of packet loss or delays. The measurement report will then indicate 2 out of 10 seconds in error = 20% errored seconds (ES). In the Routing Active Testing user interface, this is displayed as a red bar representing an ES percentage between 10% and 49%.

In the user interface, you can zoom in on a time interval, drilling progressively deeper into the details of the measurement history. Conversely, as you zoom out a graph, you might see the colors (= error levels) in the graph changing. This is because the measurement resolution changes along with the zoom setting, as detailed *here* (page 473). For instance, zooming out from the "last 15 minutes" to the "last 24 hours" will change the resolution from 10 seconds to 30 seconds. Two errored seconds, as in the above example, then no longer correspond to an ES percentage of 20% but rather to 6.7%.

Assume now that the packet loss is presented in a graph or table as 0.8% during a 10-second interval (measurement resolution = 10 seconds). This triggered an ES, although you set the ES threshold at 1% packet loss. How come?

The explanation is that the packet loss may have occurred in a burst rather than being evenly distributed over the 10 seconds. For example, you might have had 8% packet loss during a single second, which triggered an ES for that second, and no packet loss the rest of the time. The detailed table will indicate an ES percentage of 10% (one out of ten seconds), that is, the percentage of seconds when the packet loss was above your threshold.

Note: The image below is provided only to illustrate time intervals of increasing lengths; the data does not correspond to the numbers discussed above.



9.5.1 Errored second calculation for delay

Uniquely among metrics output by Test Agents, the delay metric is reported in the form of minimum, maximum, and average values. When calculating errored seconds for delay, what is compared to the ES threshold is the maximum delay. In other words, an errored second will be indicated if the delay exceeded the ES threshold at any point during the one-second interval.

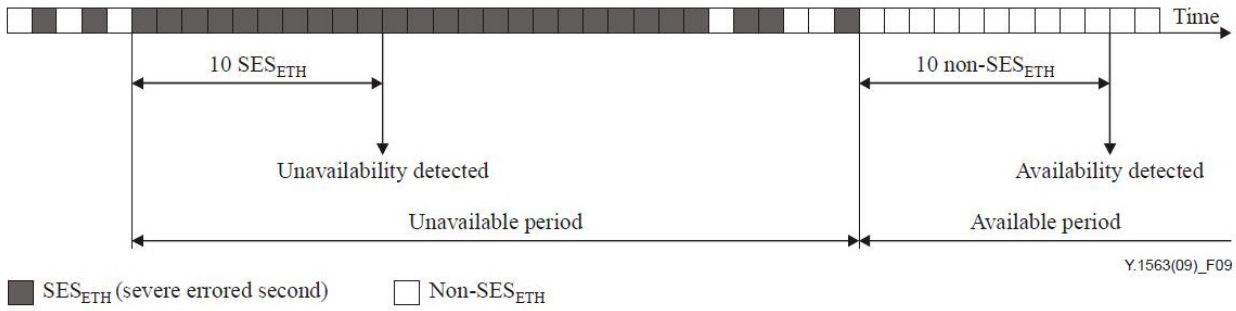
9.6 Severely Errored Seconds (SES)

The Severely Errored Seconds (SES) metric is very similar to the regular *Errored Seconds (ES)* (page 475) metric. A severely errored second occurs when the ratio of threshold violations during a one-second interval exceeds a certain predefined threshold, which should be higher than the ES threshold.

An SES should always be more severe than a regular ES; ► ITU-T Recommendation Y.1563, however, mentions only packet loss in connection with SES, and suggests an SES threshold of 50% for this metric. Routing Active Testing has its own user-configurable SES thresholds for packet loss, delay and jitter. SES statistics appear in the table view in the result report.

9.7 Unavailable Seconds (UAS)

The Unavailable Seconds (UAS) metric indicates during how many seconds of some interval the service can be considered to have been unavailable. ► ITU-T Recommendation Y.1563 defines UAS based on the concept of *Severely Errored Seconds (SES)* (page 477). A period of unavailability starts with 10 consecutive SES and ends with 10 consecutive non-*ES* (page 475). Those first 10 consecutive SES are part of the period of unavailability. See the diagram below, taken from Y.1563, for an illustration:



Routing Active Testing presents the total number of UAS, i.e. the total number of seconds during which the network has been in the unavailable state during the test or monitoring session that is currently running.

Note: For the TWAMP task, the SES threshold is configurable, as explained *here* (page 421). For all other tasks that produce the UAS metric, the SES threshold is fixed at 10 seconds as described above.

9.8 SLA (Service Level Agreement)

A service level agreement is an agreement between the service providers and the customer. SLAs commonly refer to measurements made to understand how the service is received by the customer, and they are of interest to the service provider and the customer alike. In Routing Active Testing, all metrics related to service degradation are transformed into *errored seconds (ES)* (page 475) by setting appropriate thresholds for the metrics.

The level of SLA fulfillment given in monitoring results is calculated as 100 – ES (%). With the default SLA thresholds in Routing Active Testing, we obtain the following:

- If 100 – ES ≥ 99.95%, that is, if the ES percentage is below 0.05%, the service level is classified as **SLA Good** (green SLA icon).
- If 99.5% ≤ 100 – ES < 99.95%, that is, if the ES percentage is in the range 0.05% … 0.5%, the service level is classified as **SLA Acceptable** (orange SLA icon).
- If 100 – ES < 99.5%, the service level is classified as **SLA Bad** (red SLA icon), and you should consider taking immediate action to locate and solve the problem. For example, an ES percentage of 1% equates to an SLA fulfillment of 99%, which is in the SLA Bad region.

These SLA icons are presented on the *dashboard* (page 7).

The SLA icons provide a quick way to understand if the service level is high enough, or if there are quality issues degrading the service, and if so what is causing these problems. SLA thresholds set in Routing Active Testing should of course correspond to what is set down in the actual SLA, or to other agreed SLA levels, in order for measurement results to be accepted by all parties. How to change the default SLA levels is described on *this page* (page 41).

Note: The coloring of the SLA icons is determined entirely by the above criteria and is thus independent of the color range used for errored seconds.

---

10 Applications

10.1 Introduction

Application in Routing Active Testing is an umbrella term for a variety of Test Agent activities that are neither tests nor monitors. They are grouped in a separate category in the GUI.

- *Remote packet capture* (page 479)
- *Speedtest* (page 482)
- *Test Agent as proxy* (page 486)

10.2 Remote packet capturing

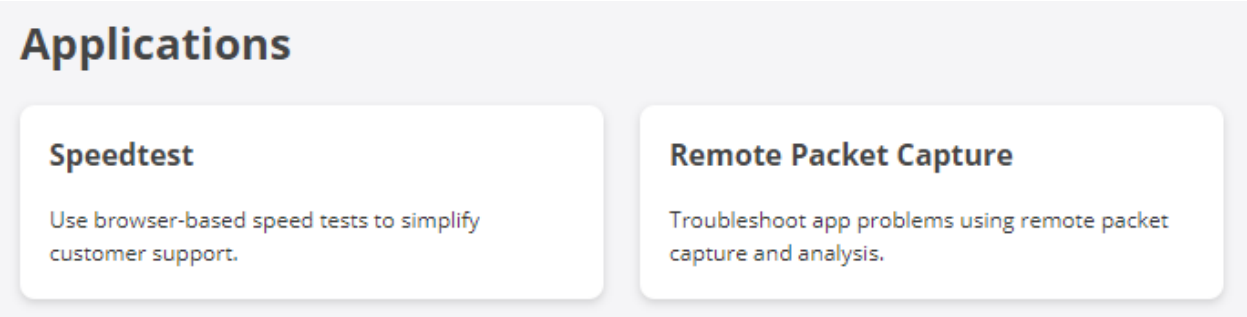
Routing Active Testing supports two ways of capturing traffic for packet-by-packet analysis using a packet analyzer such as Wireshark.

- **Non-live packet capture:** This method captures traffic on Test Agent interfaces; the traffic can subsequently be downloaded to your PC. The method is accessed from Apps in the main menu.
  - *Advantages:* Can be used to capture traffic behind NAT. Distributed captures can be easily triggered.
  - *Disadvantages:* No real-time capturing. Size of capture is limited.
- **Live packet capture:** This method captures traffic in real time by forwarding all traffic from the Test Agent directly to Wireshark. The method is accessed under Test Agents by clicking on a Test Agent, then clicking the Applications tab.
  - *Advantages:* You can capture much more traffic, since the traffic is not stored on the Test Agents, and you can track the capture in real time.
  - *Disadvantages:* Capture behind NAT is not supported (you need a direct connection to the IP address). Distributed captures are not as easy.

Read more about these capture methods below.

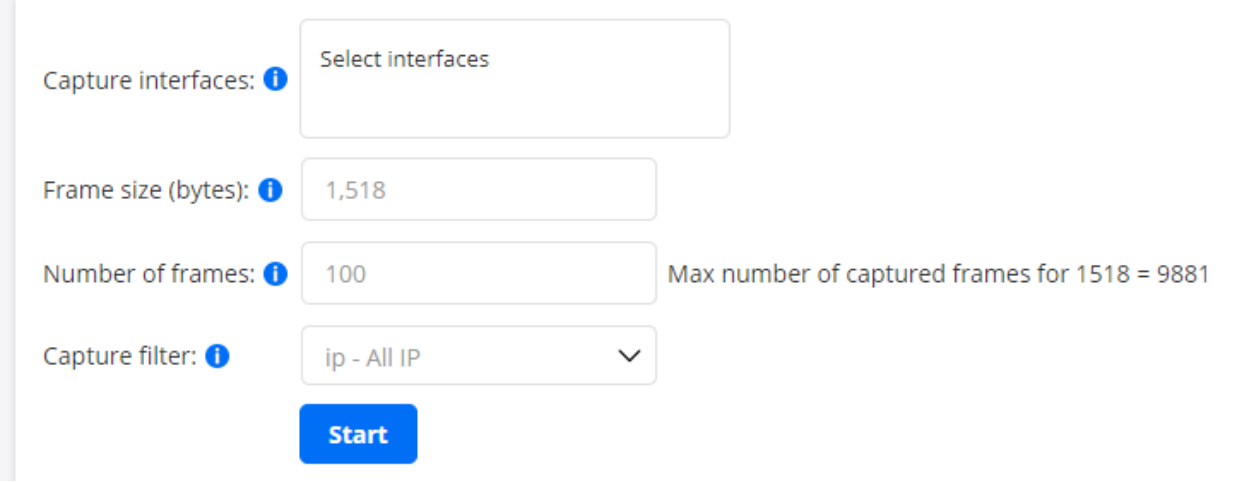
10.2.1 Non-live packet capture

Use this method to capture real user traffic on any of your Test Agent interfaces directly from your Routing Active Testing account.



You can start a capture on multiple interfaces in parallel, and you will see the number of captured packets updated live. When the specified number of packets have been captured, you can download the capture as a `.pcap` file. If the capture takes too long, you can cancel the capture at any time and still download the packets captured up until that point.

To configure this method, specify the parameters below, then start the capture by clicking the Start button.



- Capture interfaces: Select the Test Agent interfaces on which to perform the packet capture.
- Frame size (bytes): The maximum number of bytes to be captured of each packet. Default: 65,535.
- Number of frames: The maximum number of packets to be captured on each interface. The maximum depends on the size of each packet, since the total allocated memory is 15 MB.
- Capture filter: Only packets matching this filter will be captured. The tcpdump/Wireshark filter format is used.

The maximum size of the capture buffer is 15 MB. This means 245,760 packets with Packet size = 64 bytes, or 10,361 packets with Packet size = 1518 bytes. The higher you set Packet size, the fewer packets you will get. Also please note that the download of captured data may time out if the Test Agent management connection is too slow.

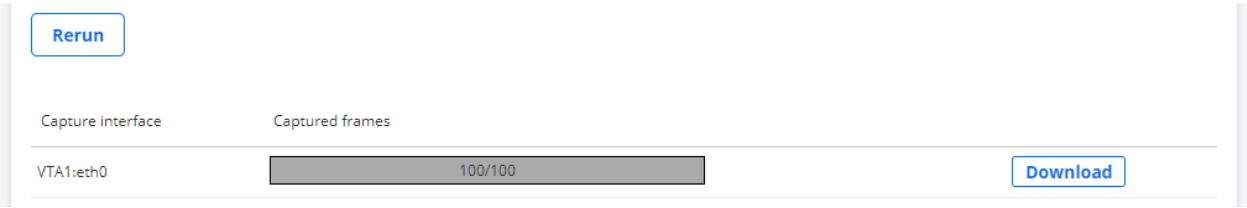
The packet capture filter follows the same format as the capture filters in Wireshark. For the syntax of these filters, refer to the Wireshark capture filters wiki.

Some useful predefined filters are available:

Filter	Function
ip	All IP
udp	All UDP
tcp	All TCP
icmp	All ICMP
udp port 53	DNS
tcp port 80	HTTP
port 5060 or port 5061	SIP
tcp port 143	IMAP only
udp port 161	SNMP only

You can also create your own capture filters.

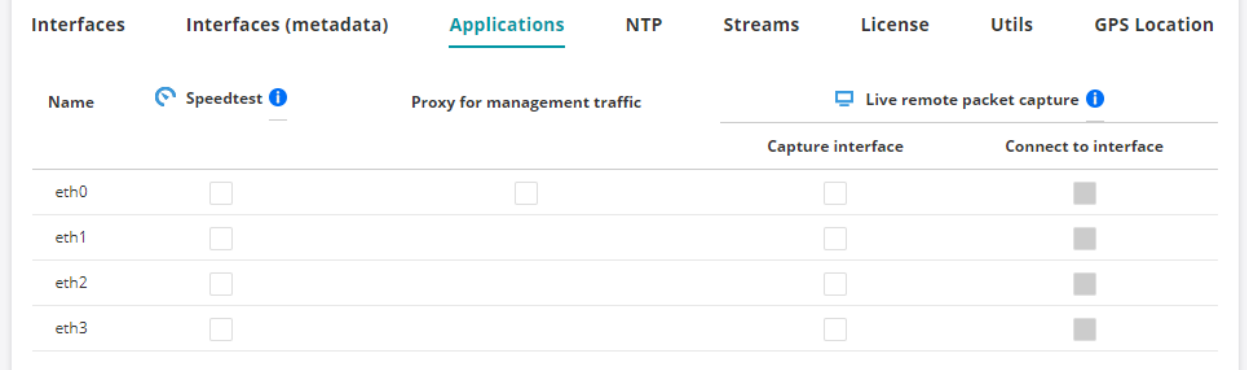
After the capture has finished, you have the option to download and open a `.pcap` file in Wireshark or in some other packet analyzer of your choice.



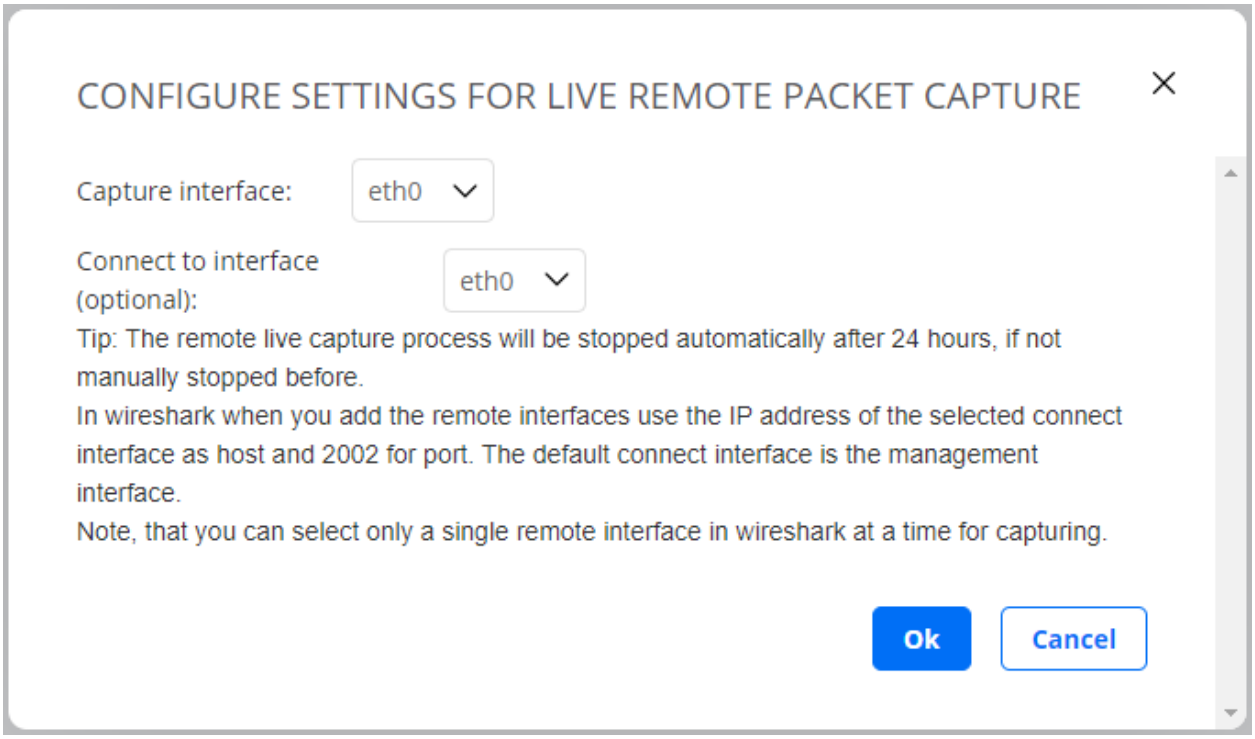
For security reasons, the captured data is not stored on the Test Agents, nor on the Routing Active Testing server, and is available only as long as you stay on the Remote Packet Capture page.

10.2.2 Live packet capture

- To enable the live capture in the menu, click the Test Agent in the Test Agents view, then select the Applications tab.
- Select a capture interface in the Capture interface column.



- A dialog appears where you also get to select which interface to connect to (the two can be different):



In Wireshark, when adding the remote interfaces, use the IP address of the selected connect interface as host, and use 2002 for port. The default connect interface is the management interface.

For more information on how to capture traffic from remote interfaces using Wireshark, go here and search for “remote interfaces”.

Note: Live packet capture requires the WinPcap library. It does not work with the Npcap library introduced as the

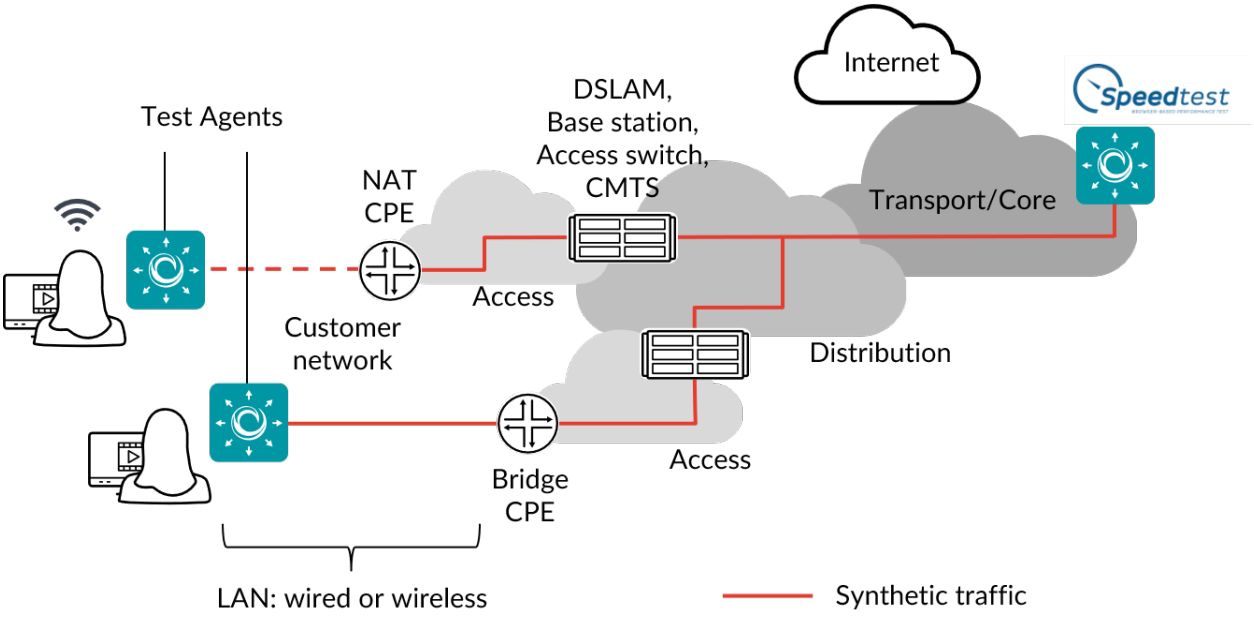
default for packet capture in Wireshark 3.0.

10.3 Speedtest

Speedtest (formerly called BBQ) is a browser-based throughput test or quality test between an end-user connection and a well-defined endpoint in your network – that is, a Test Agent.

Speedtest uses the WebSocket technology.

The main differences between Speedtest in Routing Active Testing and publicly available tools such as ► bredband-skollen.se or ► speedtest.net lie in where the servers are located and what you can do in addition to a Speedtest in order to spot and locate problems. One main advantage with Speedtest is that your customers can measure broadband speed in a controlled environment. This is illustrated in the picture below:



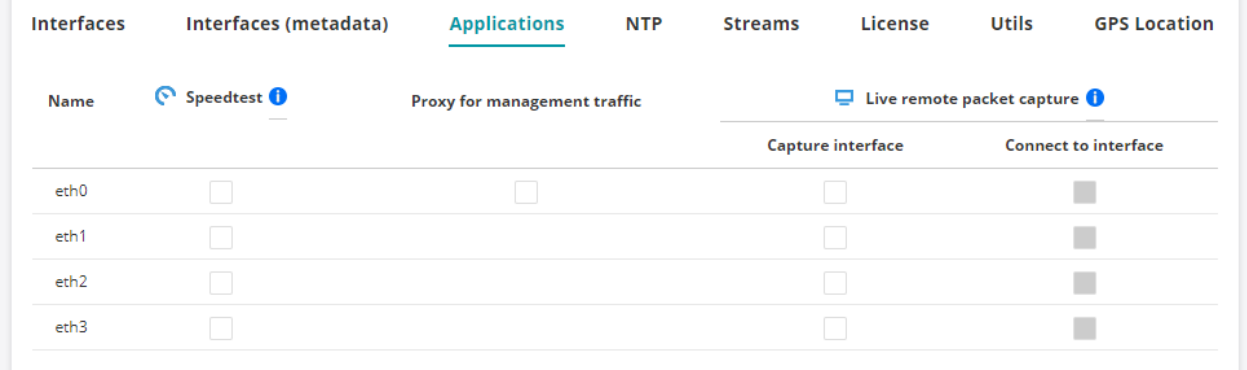
Another benefit of the Speedtest function in Routing Active Testing is that it supports additional and more advanced testing and troubleshooting features, suitable for use if the Speedtest indicates a network performance issue.

Below, the Speedtest function built into Routing Active Testing is described. It is also possible to set up a customizable web page user interface for Speedtest. That web page can be hosted on any web server (that is, not necessarily where Control Center resides). For further information, consult the document "Creating a Custom Speedtest Web Page", available at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

10.3.1 Prerequisites

To start using Speedtest, you need to enable it on at least one of the Test Agents that you have registered to your account. You find your Test Agents under Test Agents in the main menu.

- To enable Speedtest on the Test Agent, click the Test Agent in the Test Agents view, then select the Applications tab.
- In the Speedtest column, select the interfaces for which you want to enable Speedtest.



How to configure Speedtest is described on the page *Configuring Speedtest* (page 38).

You also need to make sure that all Speedtest Test Agents are able to receive traffic on the port selected as TCP destination port (this port must not be blocked by a firewall). Again, see the Configuring Speedtest page.

10.3.2 Running a Speedtest

To run a Speedtest in your web browser of choice, go to Apps > Speedtest and click the button Go to public page.

On clicking one of the buttons, you are taken to the public Speedtest page. Its URL is `https://<Control Center host IP>/<your account>/speedtest`.

If you have defined multiple categories in your Speedtest configuration, select one in the box that is by default labeled Category (you can choose to label the box differently in the configuration).

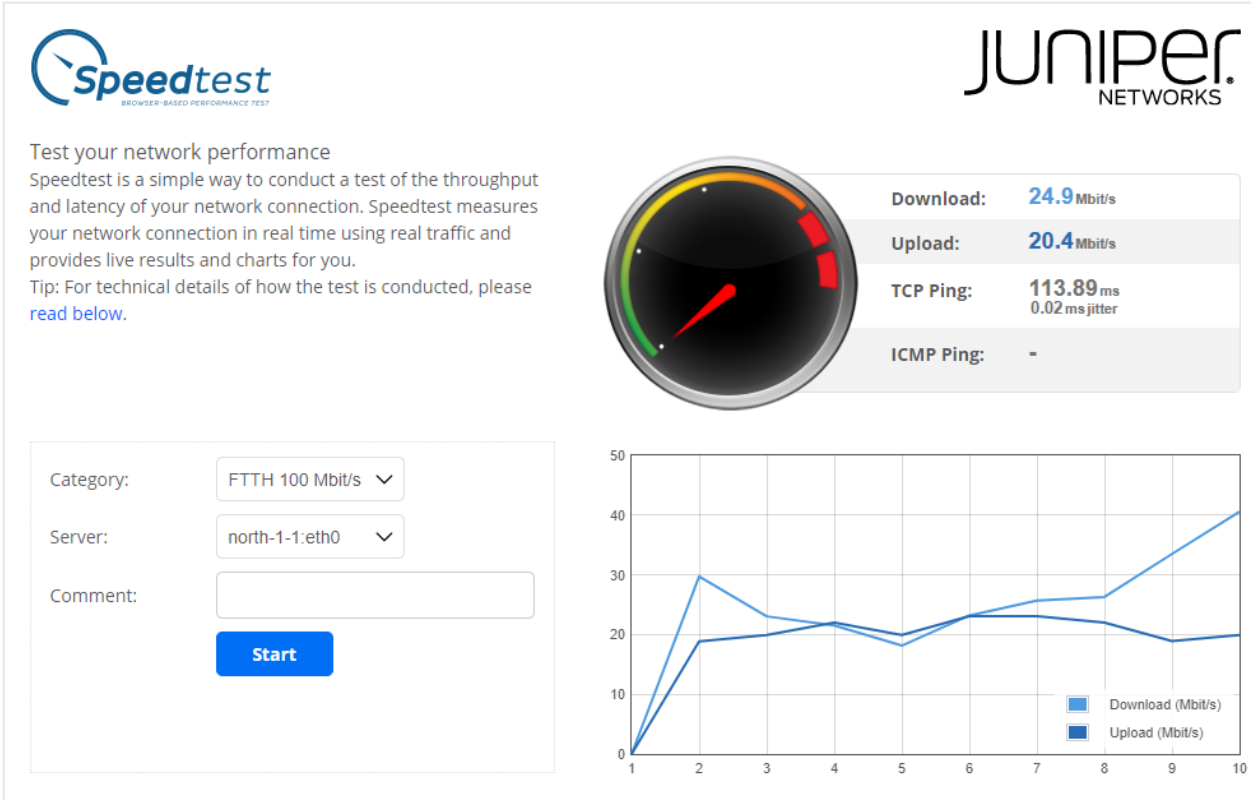
If Speedtest is enabled on more than one Test Agent, select which one to use under Server.

Then click the Start button on this page to start a Speedtest.

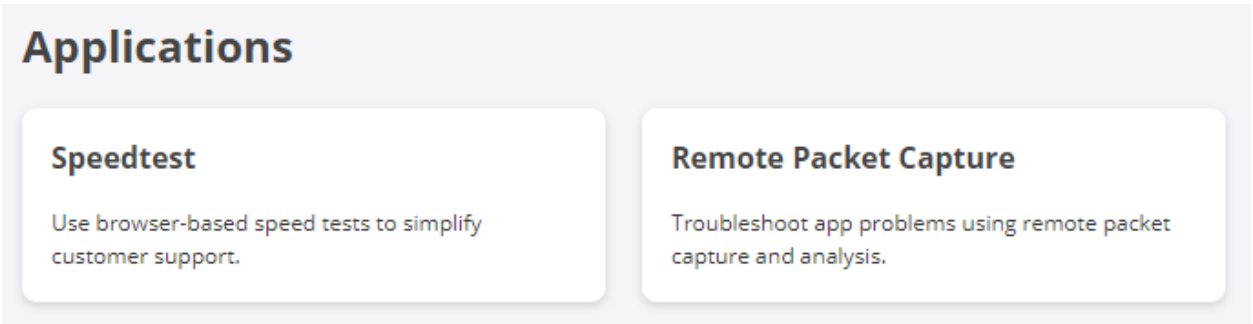
These are the steps performed during a Speedtest:

- Download: A number of parallel TCP sessions are set up to measure the receiving capacity towards your computer. The test will start with 5 sessions; more sessions are then added depending on the rate measured during the first 5 seconds. A total of 24 parallel sessions might be used.
- Upload: A number of parallel TCP sessions are set up to measure the sending capacity from your computer. The test will start with 5 sessions; more sessions are then added depending on the rate measured during the first 5 seconds. A total of 24 parallel sessions might be used.
- TCP Ping: A small amount of data is sent back and forth over a single TCP session to measure the round-trip delay.
- ICMP Ping: ICMP echo requests are sent from the server to measure round-trip delay and loss.

The public Speedtest page displays selected metrics:



More detailed information on the outcome of the test, and other tests that have been performed towards your account, can be viewed in the Routing Active Testing web GUI. Click Apps in the main menu:



Then click the Speedtest box to go to the Speedtest result pages:



10.3.3 TCP information

During both upload and download, some TCP information is obtained from Speedtests. The `tcp_info` struct in the Linux kernel on the Test Agent is sampled once every second. These samples are then aggregated and reported on the Speedtest result page under Details on the TCP info tab.

10.3.3.1 Download

- **Congestion window:** The congestion window is a TCP sender limitation on the number of packets allowed to be transmitted without acknowledgement. Each second the sum is taken of the congestion windows for all TCP sessions in the test. The max, min, and average presented in the results are based on these samples.
- **Packets in flight:** The number of transmitted packets waiting for acknowledgement from the receiver. Each second the sum is taken of the number of packets in flight for all TCP sessions in the test. The max, min, and average presented in the results are based on these samples.
- **RTT:** TCP round-trip time. The presented max and min values are taken over all samples obtained during the test.
- **RTT variance:** TCP round-trip time variance. The presented max and min values are taken over all samples obtained during the test.
- **Retransmissions:** Total number of retransmitted packets in all sessions during a download test.
- **Path MTU:** Path Maximum Transmission Unit determined in the course of the download test.
- **Max active sessions:** The maximum number of TCP sessions that were active concurrently during the test. A session is considered active if some data has been transmitted since the last sampling of tcp_info. Note, however, that this number is not itself found in the tcp_info struct.

10.3.3.2 Upload

For uploads, fewer measurements are obtained: **RTT** (max/min/avg), **Path MTU**, and **Max active sessions**. They are analogous to those obtained for the downlink; see above.

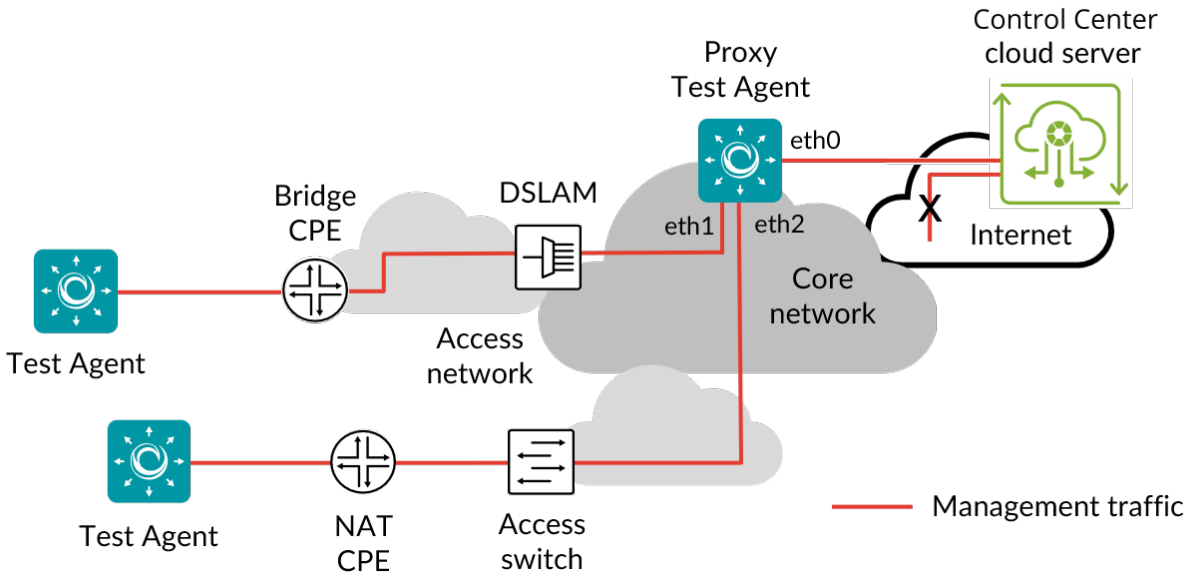
10.4 Setting up a Test Agent to act as proxy in tests

Using one of your Test Agents as a proxy makes it possible to do testing with Test Agents that would otherwise not be reachable from the Internet or from Routing Active Testing cloud servers. You might, for example, want to test and monitor an IP telephony network that does not allow any connection to the Internet.

The Routing Active Testing proxy enables a forwarding mechanism for the management traffic between the Test Agents and the Routing Active Testing cloud server in networks that do not have public IP addresses.

A proxy Test Agent always communicates with other Test Agents on the same port which it uses for communication with Control Center.

Only Test Agent Appliances can act as proxies and test via a proxy; Test Agent Applications do not have these capabilities.



This page assumes that the Routing Active Testing server resides in the public cloud, as the use of a proxy is relevant mainly in that case. It is however possible to use a proxy with an on-premise Routing Active Testing server as well.

To set up and use a Test Agent as a proxy, follow these steps:

1. Allow Internet access for one of the interfaces on one of your Test Agents. This Test Agent is commonly placed in the server hall or core network.
2. Enable proxy on that Test Agent (called “proxy Test Agent” from now on):
    - In the Routing Active Testing user interface, navigate to Test Agents, and click the relevant Test Agent in this view.
    - Go to the Applications tab, and select the checkbox Proxy for management traffic for the management interface. (See also *this page* (page 188).)

- We recommend that you configure a static IP address on the interface to which the other "internal" Test Agents connect, so that the address does not change unexpectedly as might happen when using DHCP.

3. Connect another interface ("eth1", etc.) of the proxy Test Agent to the internal network that does not allow Internet connections.
4. For each of the *other* Test Agents that you want to be able to connect to the Internet via the proxy Test Agent, do the following:
   - Log in directly to the Test Agent (that is, not via the Routing Active Testing GUI).
   - If the Test Agent is already registered with the Routing Active Testing cloud server, go to Utilities > Change login server and change the login server from https://login.paa.juniper.net to the IP address of the proxy Test Agent interface ("eth1", etc.) configured in step 2 (the internal address to the proxy). Under Port, enter the port used by the proxy Test Agent (e.g. 443).
   - If the Test Agent is not yet registered, do the registration according to *this page* (page 205).
5. *(Optional:)* Since the Test Agents need to have a working NTP synchronization, the NTP server likely has to be reconfigured. Do the following:
   - If you have your own NTP server, change the NTP server from time.google.com to your internal NTP server for all Test Agents, or:
   - Change the NTP server for all internal Test Agents from time.google.com to the IP address of the proxy Test Agent's "eth1" interface, and let the proxy Test Agent continue to synchronize to time.google.com.

See the page *Test Agent NTP configuration* (page 189) for more information about NTP.

The proxy functionality is now configured and ready to use. This allows the Test Agents to communicate with the Routing Active Testing cloud server via the proxy Test Agent, which forwards their traffic to the Routing Active Testing cloud.

11 Alarms

11.1 Introduction to alarms

Monitors in Routing Active Testing can be associated with *alarms*. Broadly speaking, an alarm is triggered when something is amiss with the monitor. The following types of alarm can be set:

- When the number of *errored seconds* (page 475) within a specified time window exceeds a defined threshold. In other words, this kind of alarm is raised when the level of *SLA* (page 478) fulfillment drops.
- When a task in a monitor stops delivering data. This is often caused by a Test Agent going offline unexpectedly.

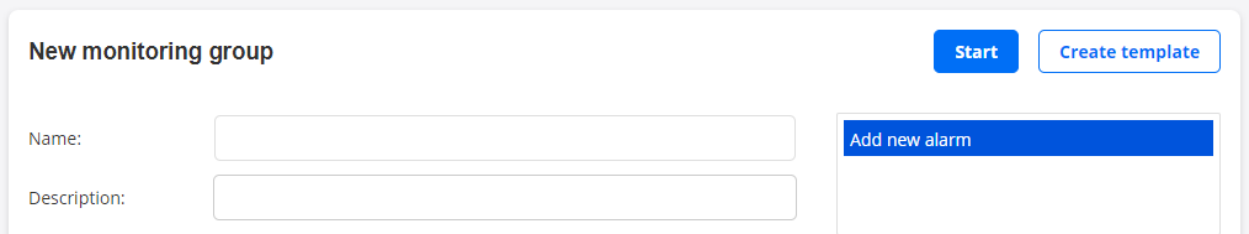
Alarms can be sent either as SNMP traps or by email. This and other aspects of alarm setup are configured in your Routing Active Testing account, as described *here* (page 42).

How to activate an alarm for a monitor is explained on *this page* (page 488).

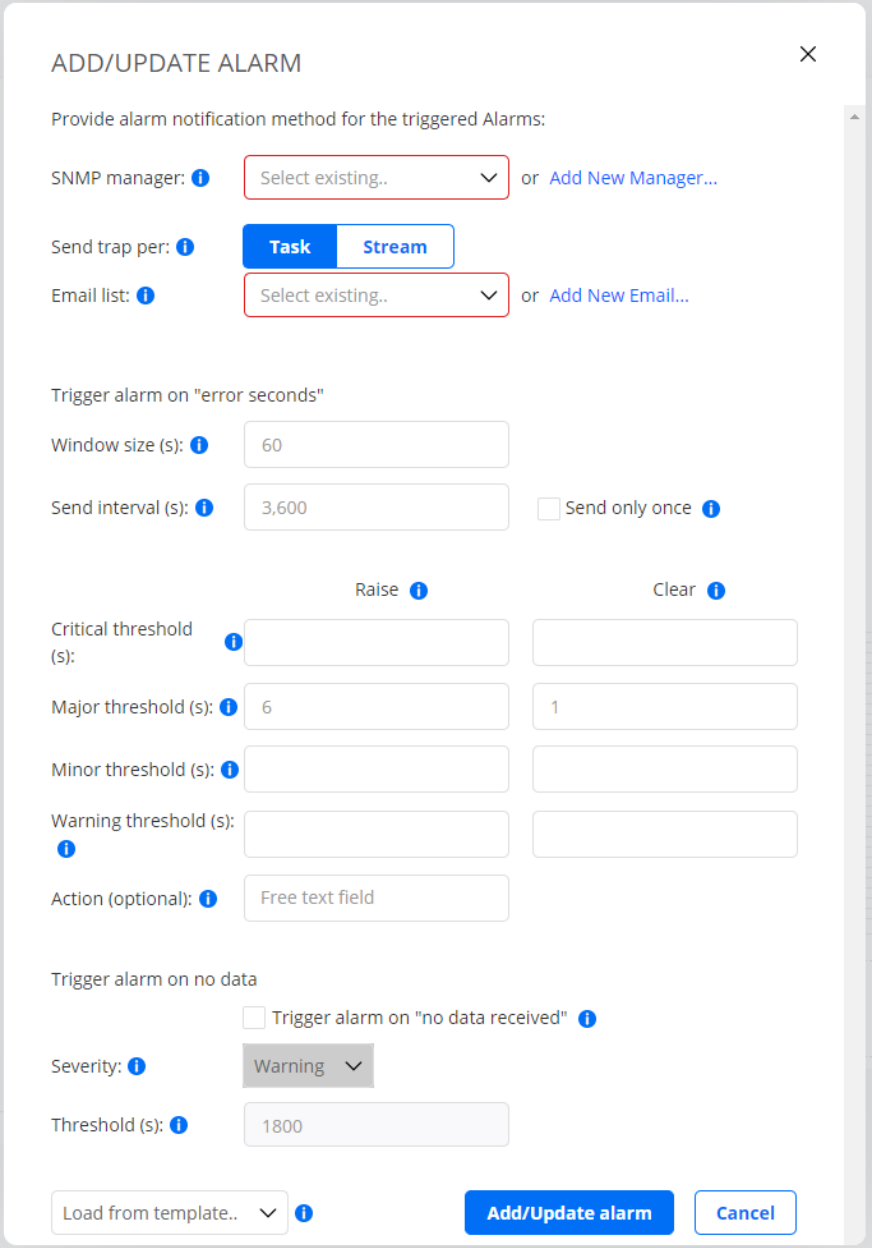
An overview of all alarms defined in your account is found on the *alarm dashboard* (page 490).

11.2 Activating alarms for a monitor

You can activate alarm handling for a monitor when creating or editing it, as mentioned on the page *Building monitors* (page 263).



Clicking Add new alarm opens the following dialog, which is mostly identical to the one for *setting up alarm templates* (page 47):

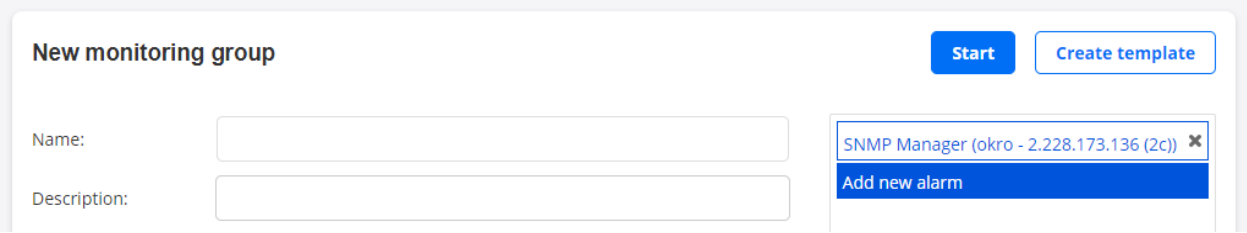


If you want to use a previously defined template for the alarm, select one of these in the Load from template box. This will populate the dialog with settings from the template. You can override any template settings simply by changing them in the dialog.

You can also proceed without selecting a template, filling in the dialog manually (the same steps as when creating a template). For the details, see the instructions on *setting up alarm templates* (page 47).

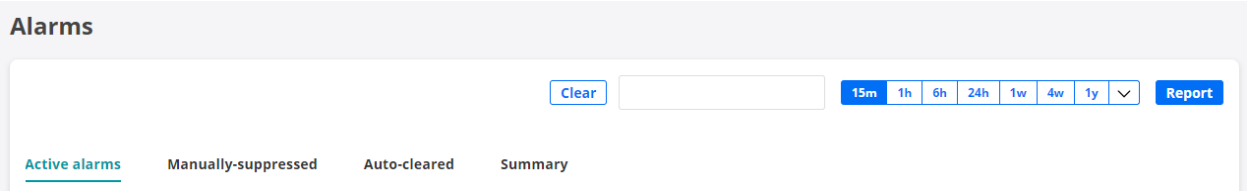
- To finalize the activation of an alarm, click Add/Update alarm.

Below is an example of an active SNMP manager.



11.3 Alarm dashboard

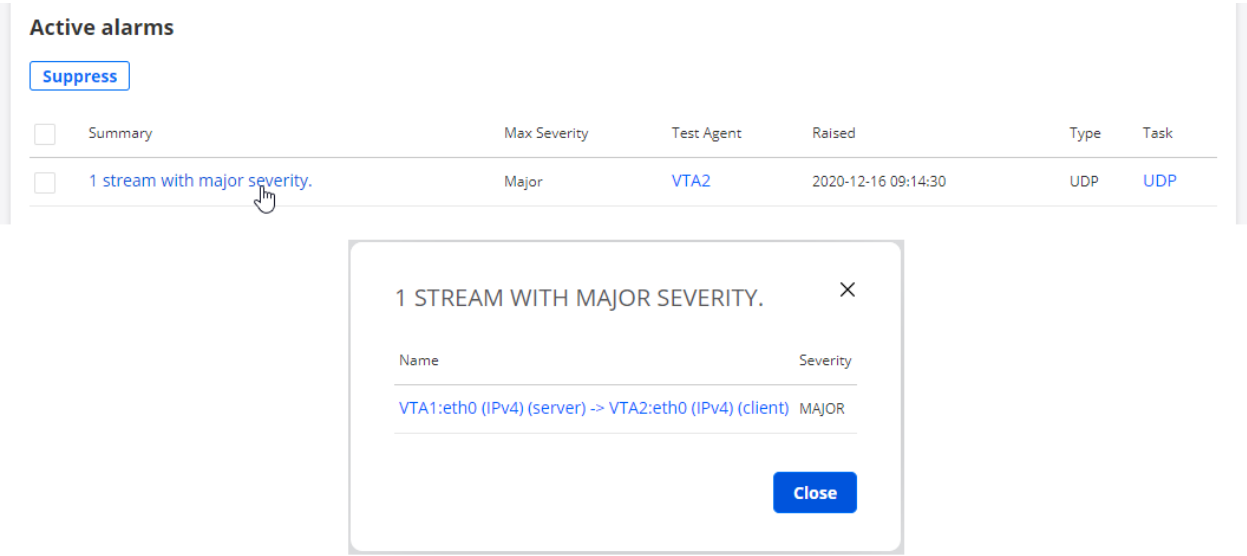
The Alarm dashboard is reached via Alarms on the main menu. All alarms are collected here: active alarms, manually suppressed active alarms, and automatically cleared alarms. The length of the history displayed can be changed under History interval (click the down arrow to set an arbitrary “from–to” interval).



Active alarms

This tab shows all active alarms from all active monitors which have an alarm configured. An active alarm is triggered when the errored second thresholds for the severity level of the alarm are exceeded.

If an alarm has been configured in the form of SNMP traps per stream, the presentation of the alarm will likewise differentiate streams. Below is an example.



Manually suppressed active alarms

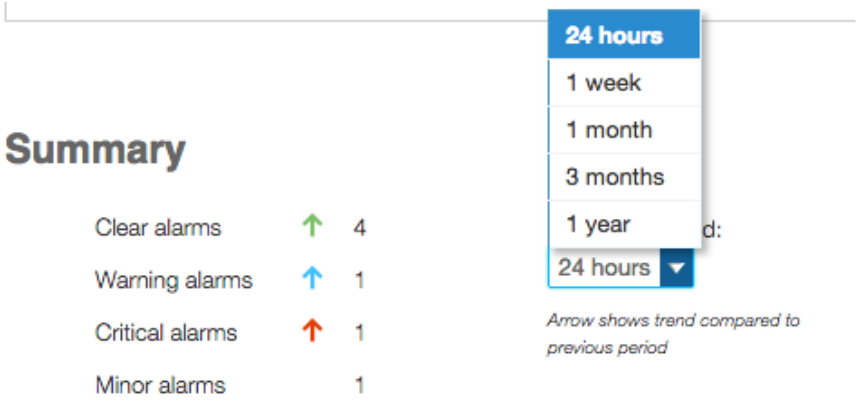
This tab shows all manually suppressed alarms. The suppressions are defined as described on the page *Setting up alarms* (page 50).

Automatically cleared alarms

On this tab all cleared alarms are collected. A cleared alarm means that the condition on the measured connection has reverted to normal.

Summary

This tab gives a summary of all alarms by severity during any of a number of predefined time periods (see screenshot below).



12 Sharing and collaboration

12.1 Sharing Test Agents

You can share a Test Agent with partners or business associates. The Test Agent will be available for use in tests and monitoring in the accounts you share it to.

The *packet capture* (page 479) function is disabled on shared Test Agents, and changing interface configurations is not possible.

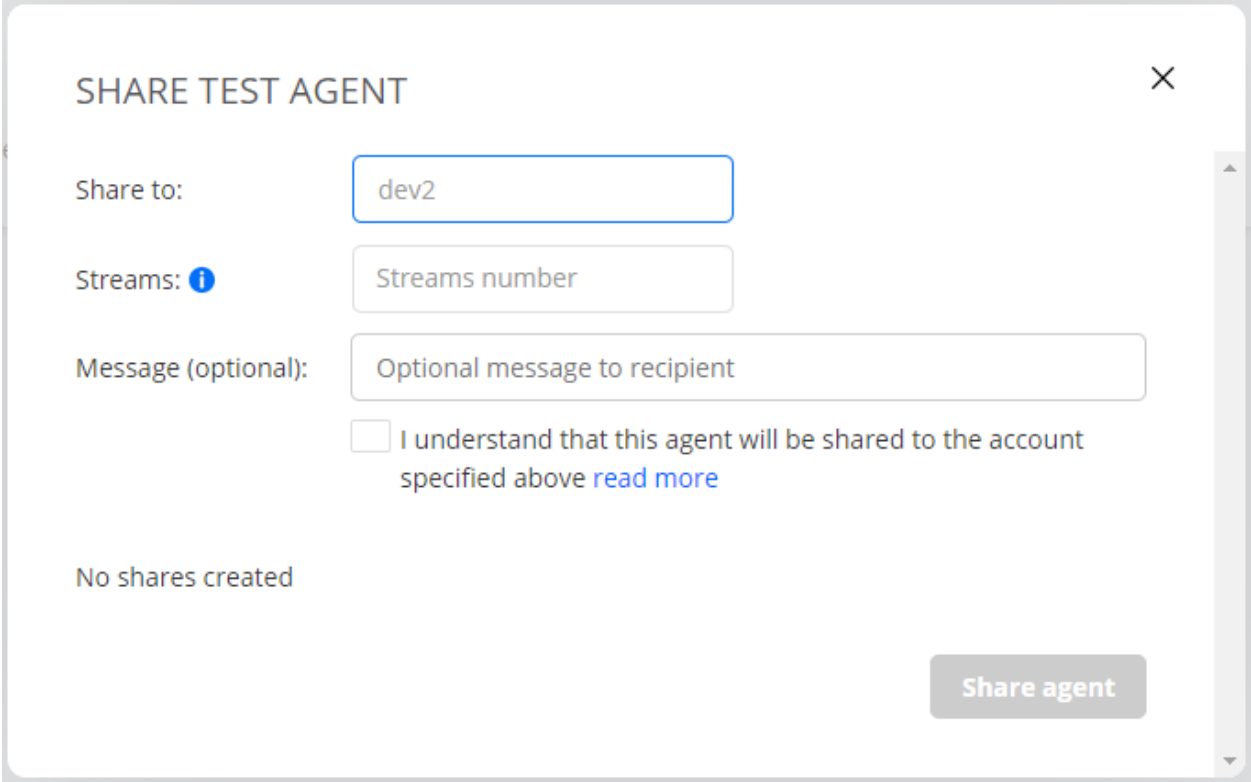
The person sharing a Test Agent is called the “sharer” on this page, and the person receiving a shared Test Agent is referred to as the “sharee”.

12.1.1 How to share a Test Agent

- Navigate to Test Agents in the main menu.


In the Share column, click the icon for the Test Agent you want to share.

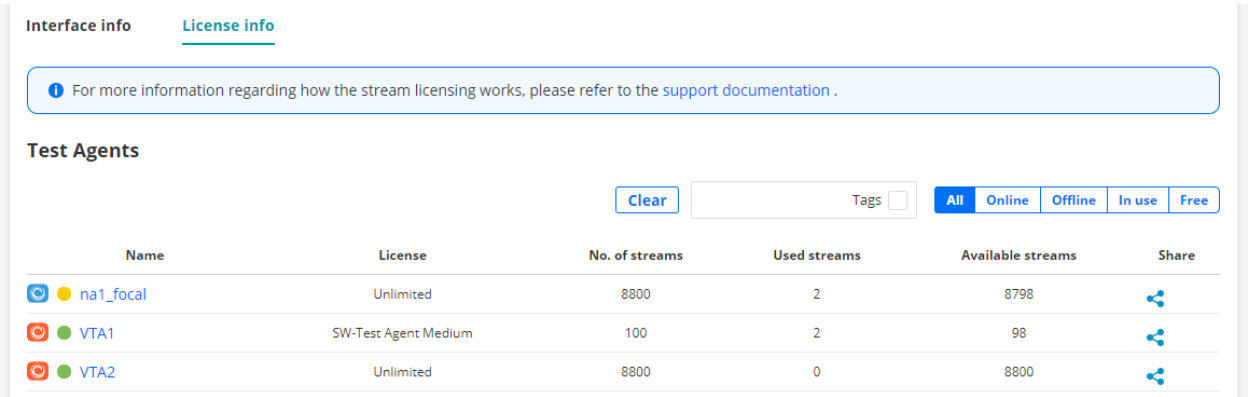
- In the dialog that appears, enter the account to share the Test Agent to and how many streams you want to assign to the shared Test Agent.
- Click the Share agent button.



You will be notified as to whether the sharing was successful or not.

The icon in the Shared column changes to a dark blue color:

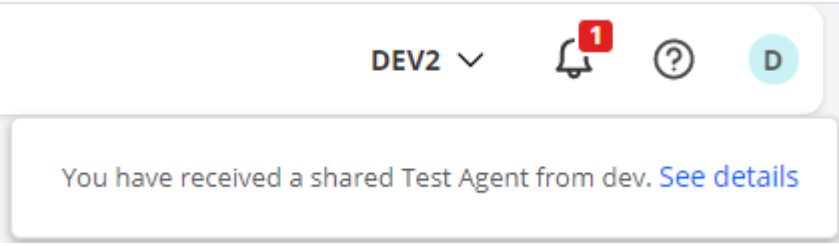
On the License info tab you can see how many streams are used (not only shared streams) and how many streams are available.



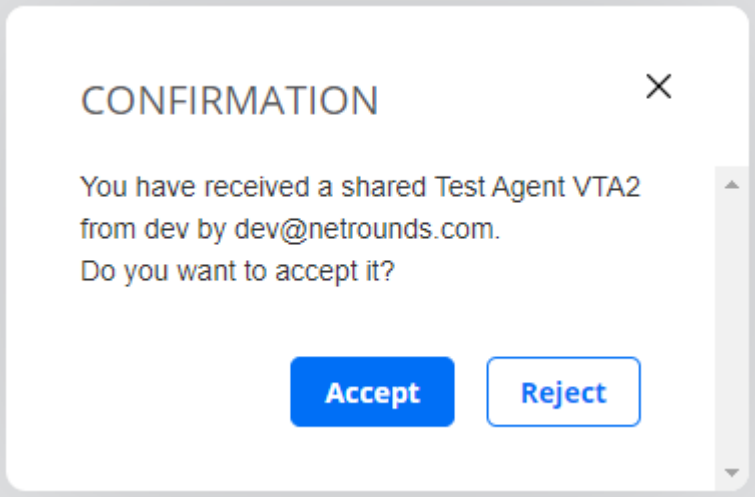
- Repeat this procedure if you want to share more Test Agents, or share the same Test Agent to multiple accounts.

12.1.2 Accepting a shared Test Agent

When a Test Agent is shared to an account, users of that account are notified by a digit appearing (or incrementing) on the top bar alarm bell. Clicking the alarm bell displays the following message:

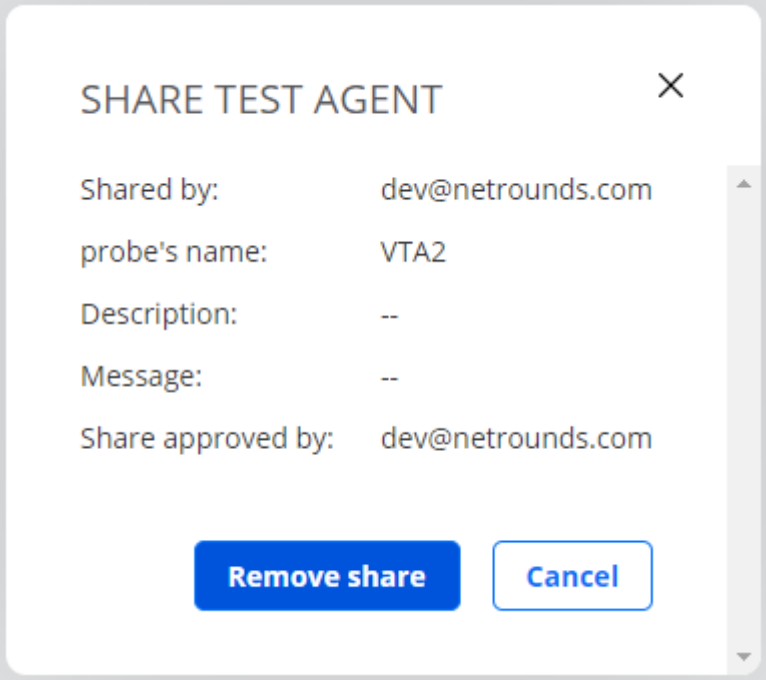


- Click the See details and choose whether to accept or reject the shared Test Agent:



If you accept the share, the Test Agent will be accessible on the Test Agents screen in the section Shared with me:

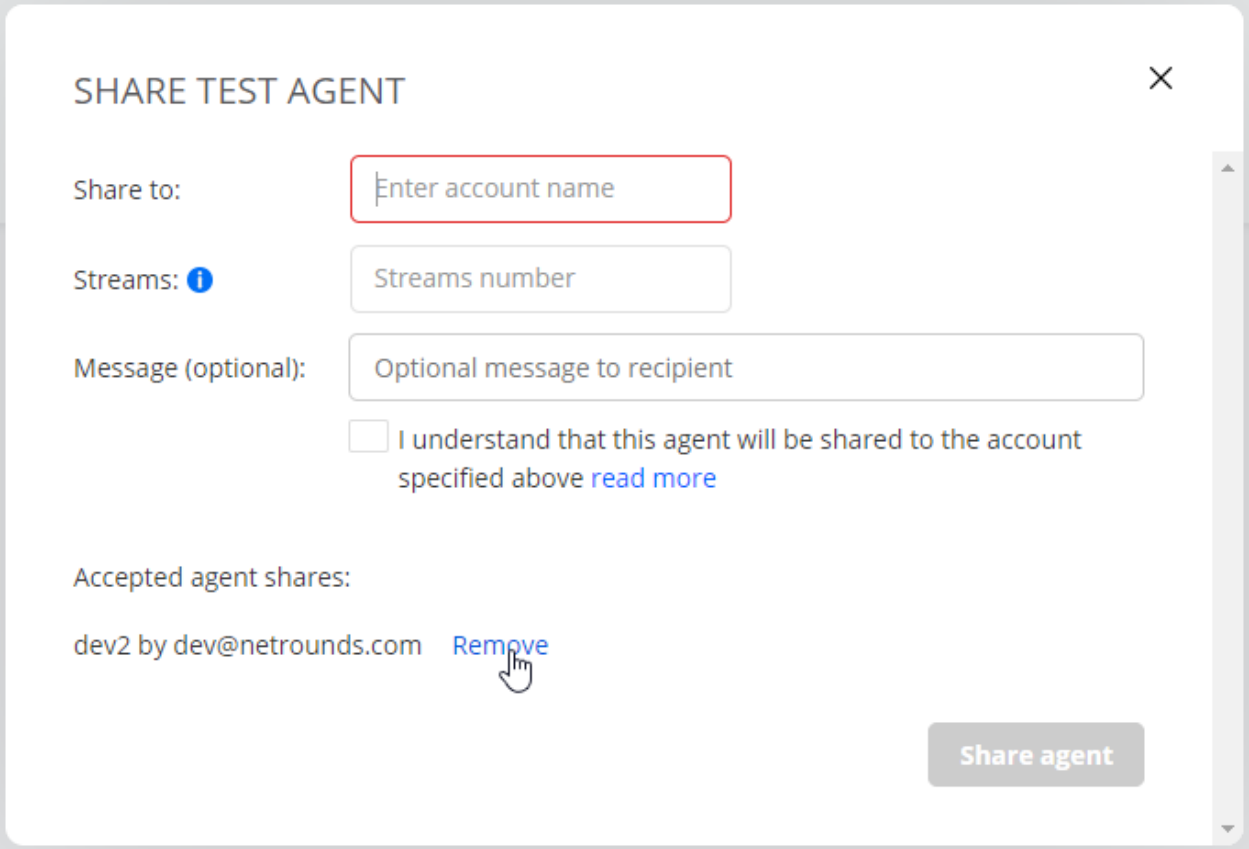
 Clicking the “shared from” icon displays sharing information:



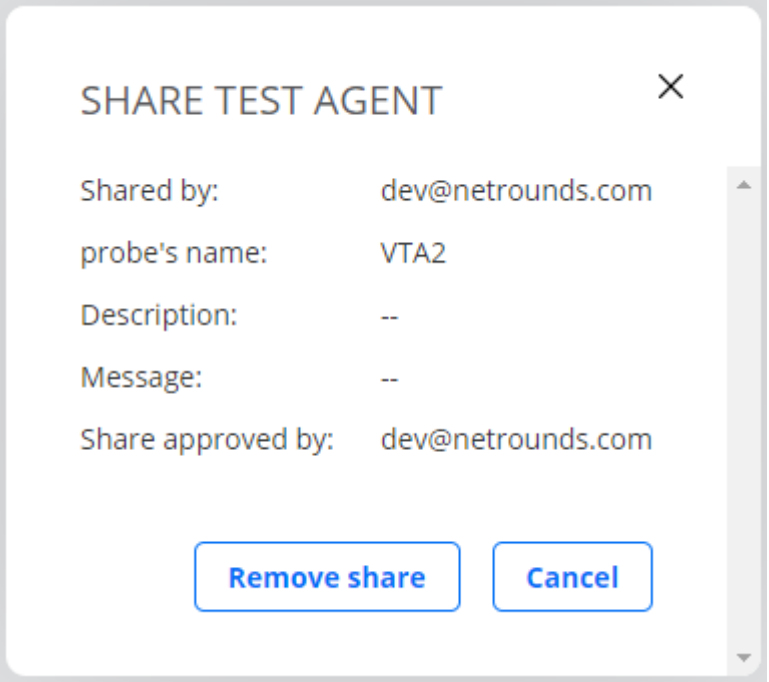
12.1.3 Removing a shared Test Agent

Both sharer and sharee can remove a share.

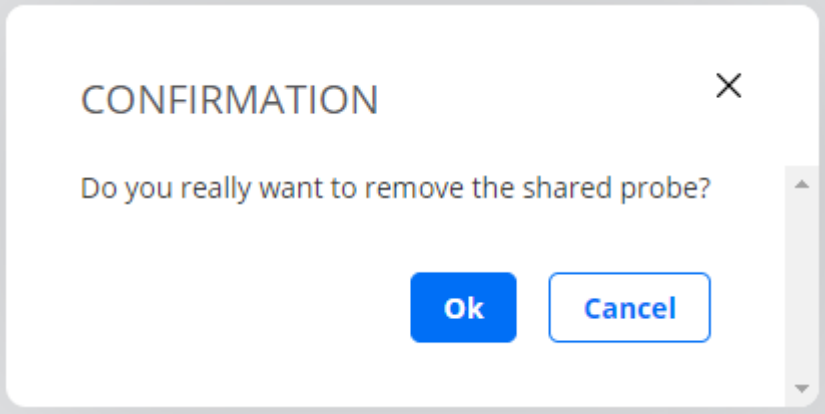
- The *sharer* clicks the  icon and then, in the dialog that appears, clicks the Remove link to the right of the Test Agent shared. See this screenshot:



- The *sharee* clicks the  icon and then, in the dialog that appears, the Remove share button:



In both cases, a confirmation dialog appears:



12.2 Sharing templates

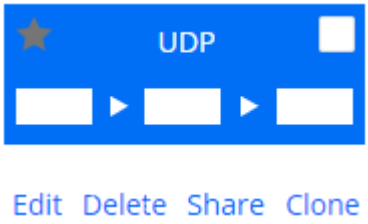
You can share a template with partners or business associates. The template will be available for use in tests in the accounts you share it to.

It is not possible to share templates with non-sharable components.

The person sharing a template is called the “sharer” on this page, and the person receiving a shared template is referred to as the “sharee”.

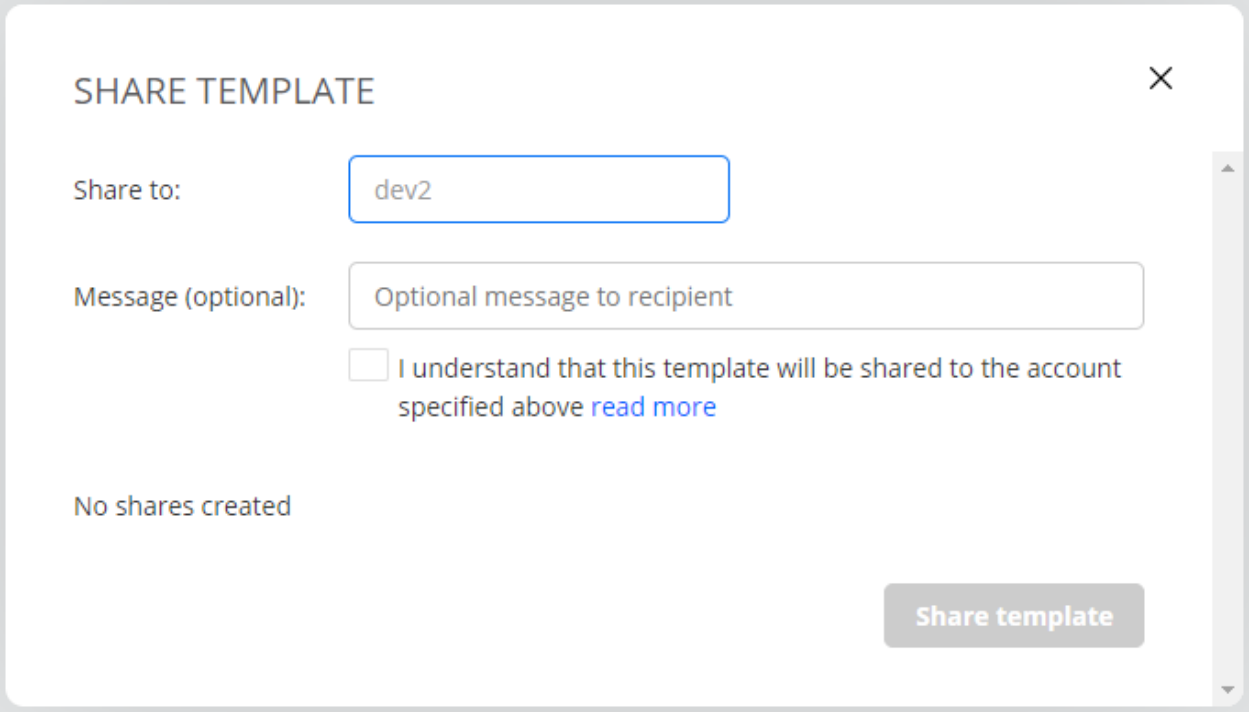
12.2.1 How to share a template

- On the left-side bar, click the Tests button and select New Test Sequence.
- Click My Templates.
- Click the Share link for the template you want to share.



Edit Delete Share Clone

- In the dialog that appears, enter the account to share the template to. You can also optionally enter a message for the recipient.
- Select the I understand… checkbox.
- Click the Share template button.

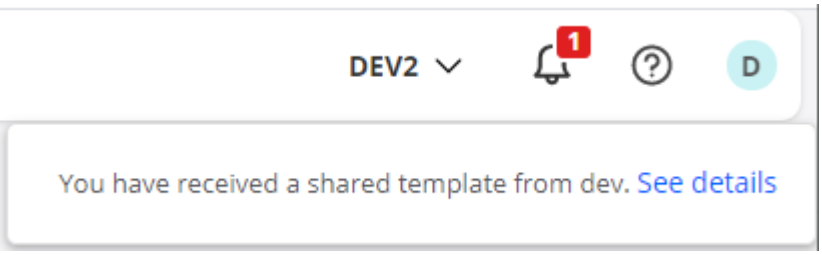


You will be notified as to whether the sharing was successful or not.

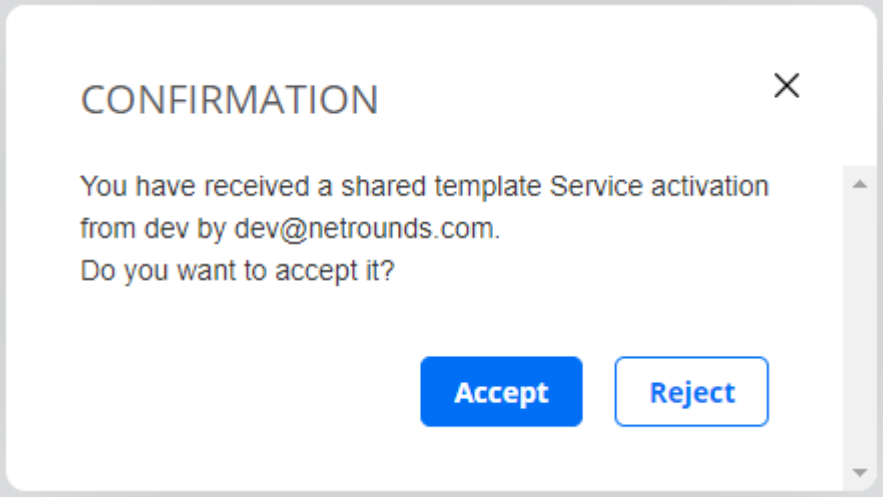
- Repeat this procedure if you what to share more templates, or share the same template to multiple accounts.

12.2.2 Accepting a shared template

When a template is shared to an account, users of that account are notified by a digit appearing (or incrementing) on the top bar alarm bell. Clicking the alarm bell displays the following message:

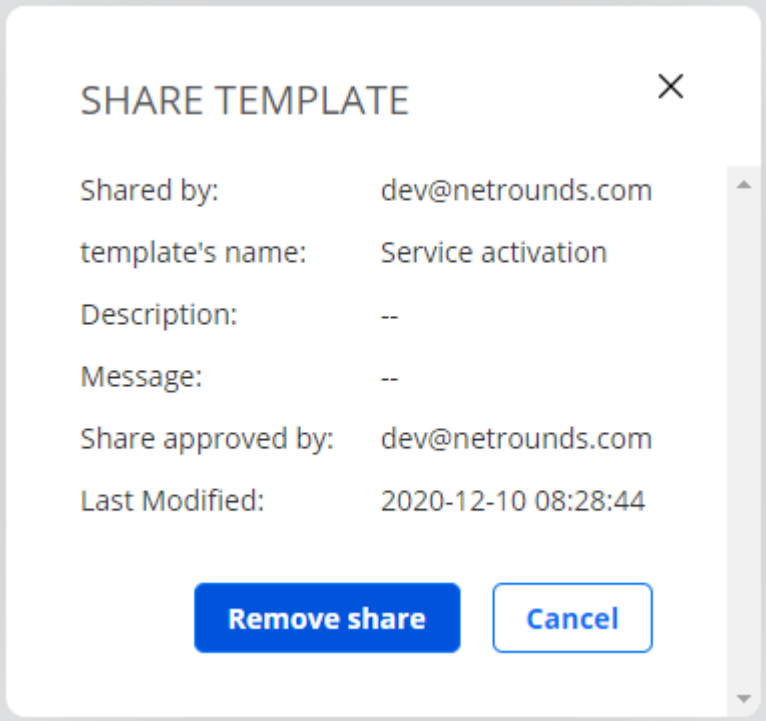


- Click the See details link and choose whether to accept or reject the shared template:



If you accept the share, the template will be accessible under Shared Templates when you create a new test.

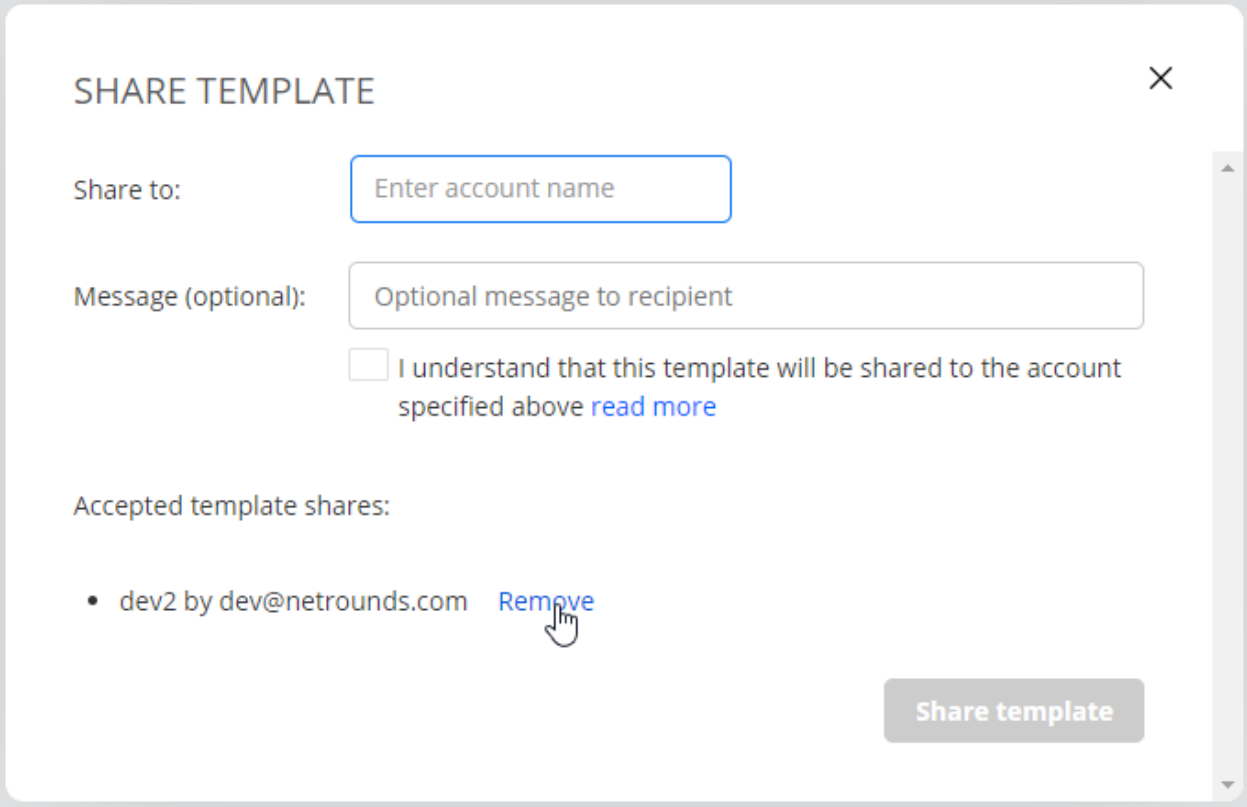
Clicking the Sharing info link displays sharing information:



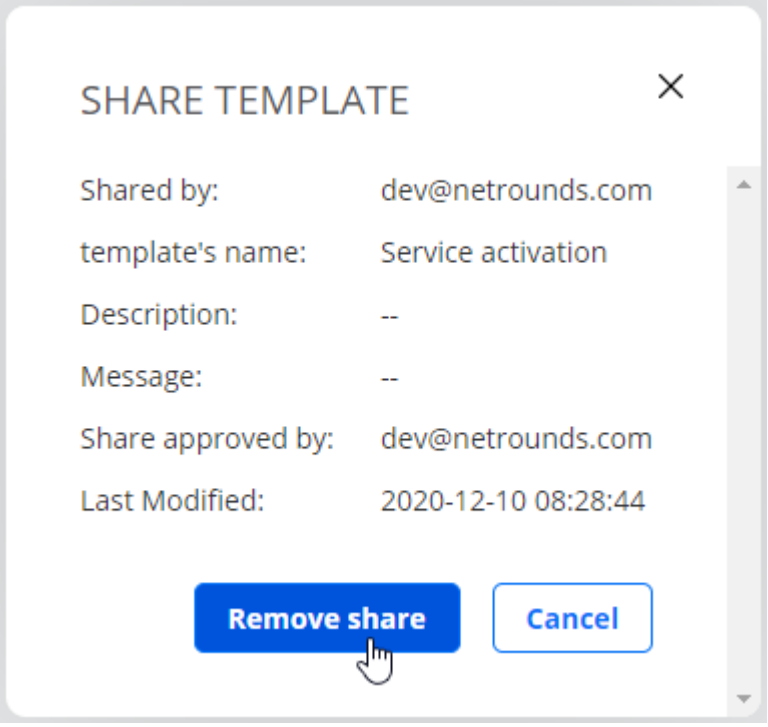
12.2.3 Removing a shared template

Both sharer and sharee can remove a shared template.

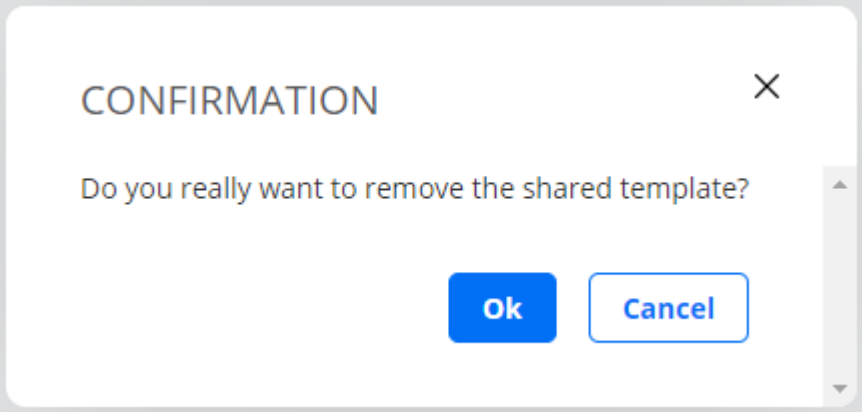
- The *sharer* enters the My Templates view, clicks the Share link below the template to be unshared, then clicks the relevant Remove link in the dialog that appears:



- The *sharee* enters the Shared Templates view, clicks the Sharing info link below the template to be unshared, then clicks the Remove share button in the dialog that appears:



In both cases, a confirmation dialog appears:



12.3 Sharing test and monitor results: Introduction

You can share specific live test and monitor results with partners and customers. This sharing can be done in two ways:

- *between accounts* (page 500) in Routing Active Testing. This method is similar to sharing Test Agents or templates.
- *using URLs* (page 504). This method is unique to measurement results and lets you share such results outside of Routing Active Testing.

The two methods are described on separate pages; please follow the links above.

12.4 Sharing test and monitor results between accounts

When you share test and monitor results to a different Control Center account, the results appear on that account's Routing Active Testing dashboard and Tests and Monitoring screens. The share and unshare procedures are the same for tests and monitors, so only sharing of monitoring results is described here. The account shared to only has read access to shared results.

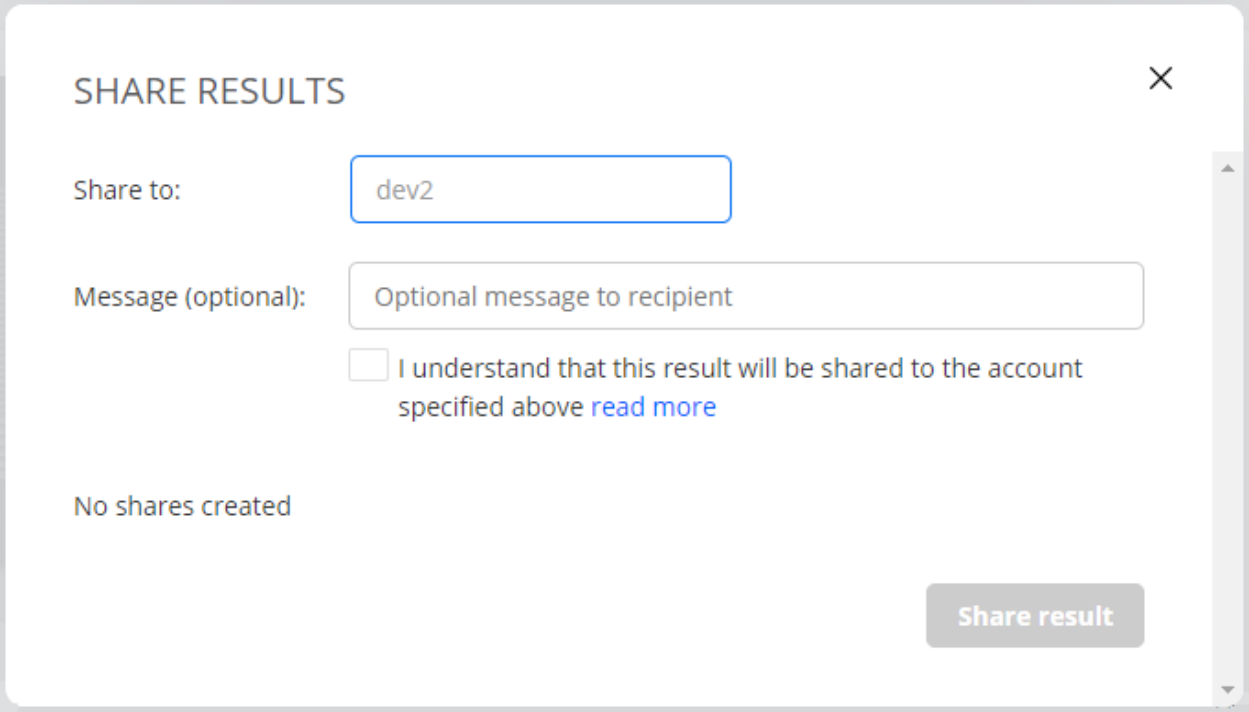
The person sharing results is called the "sharer" on this page, and the person receiving shared results is referred to as the "recipient".

12.4.1 How to share a monitor

- Navigate to Monitoring in the main menu.


In the Share column, click the icon for the monitor you want to share.

- Click the Share using account button.
- In the dialog that appears, enter the account to share to. Optionally, you can also enter a message to the receiver account.
- Check the I understand… checkbox.
- Click the Share result button.



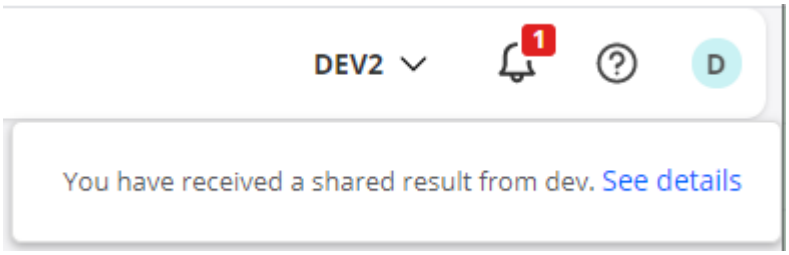
You will be notified as to whether the sharing was successful or not.


The icon in the Shared column changes to the symbol seen below:

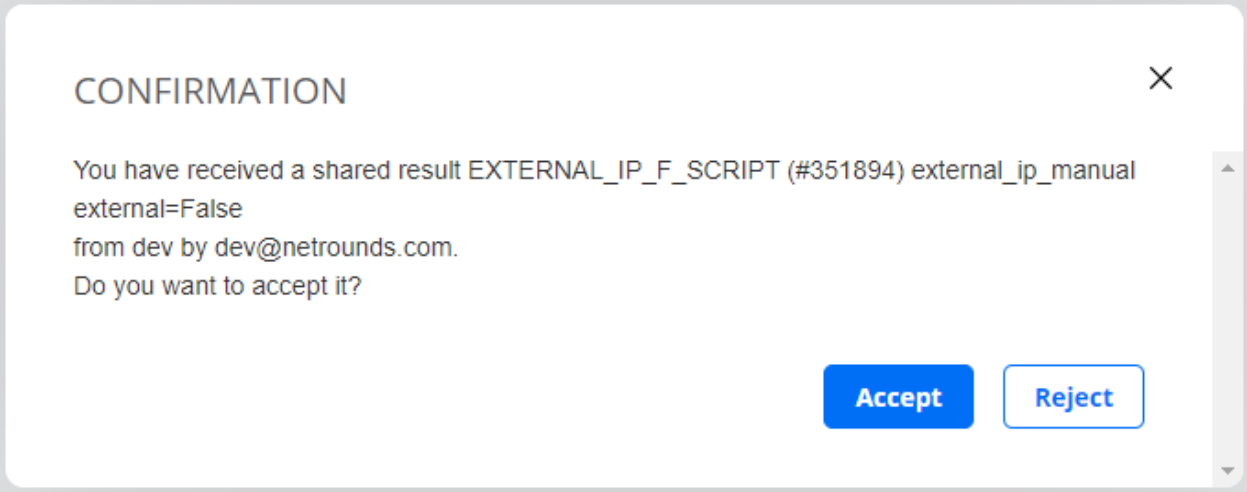
- Repeat this procedure if you want to share more monitors, or share the same monitor to multiple accounts.

12.4.2 Accepting a shared monitor

When a monitor is shared to an account, users of that account are notified by a digit appearing (or incrementing) on the top bar alarm bell. Clicking the alarm bell displays the following message:

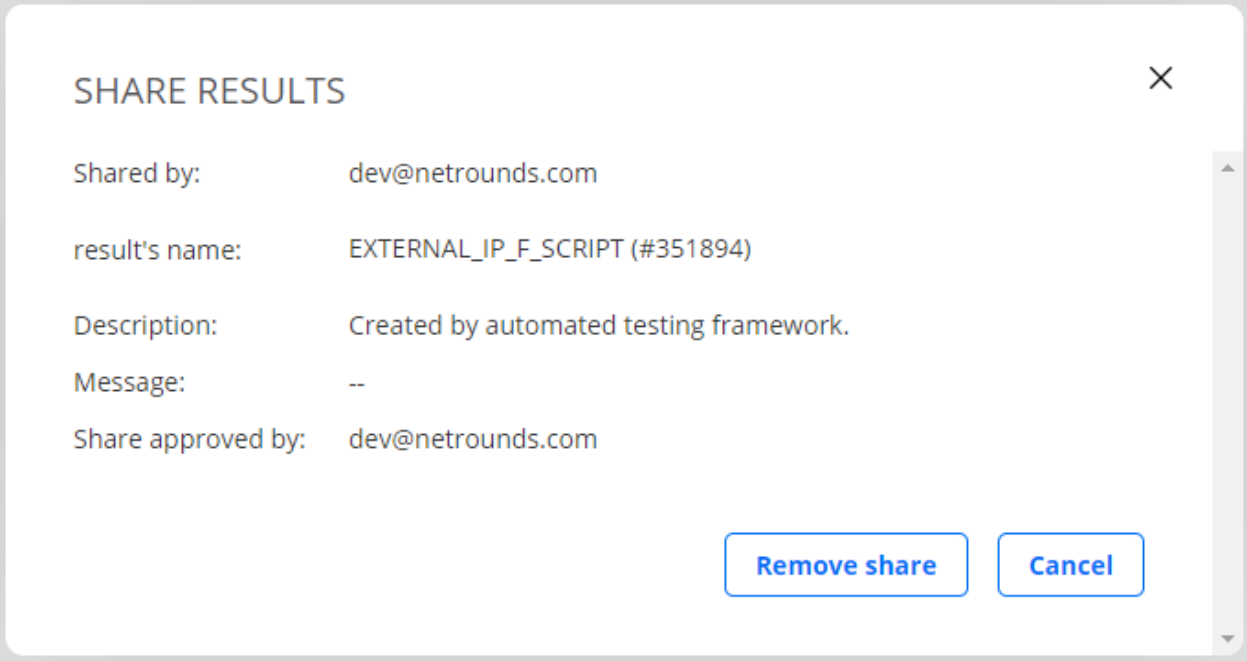


- Click the See details link and choose whether to accept or reject the shared template:



If you accept the share, the monitor will be accessible in the list under Monitoring.

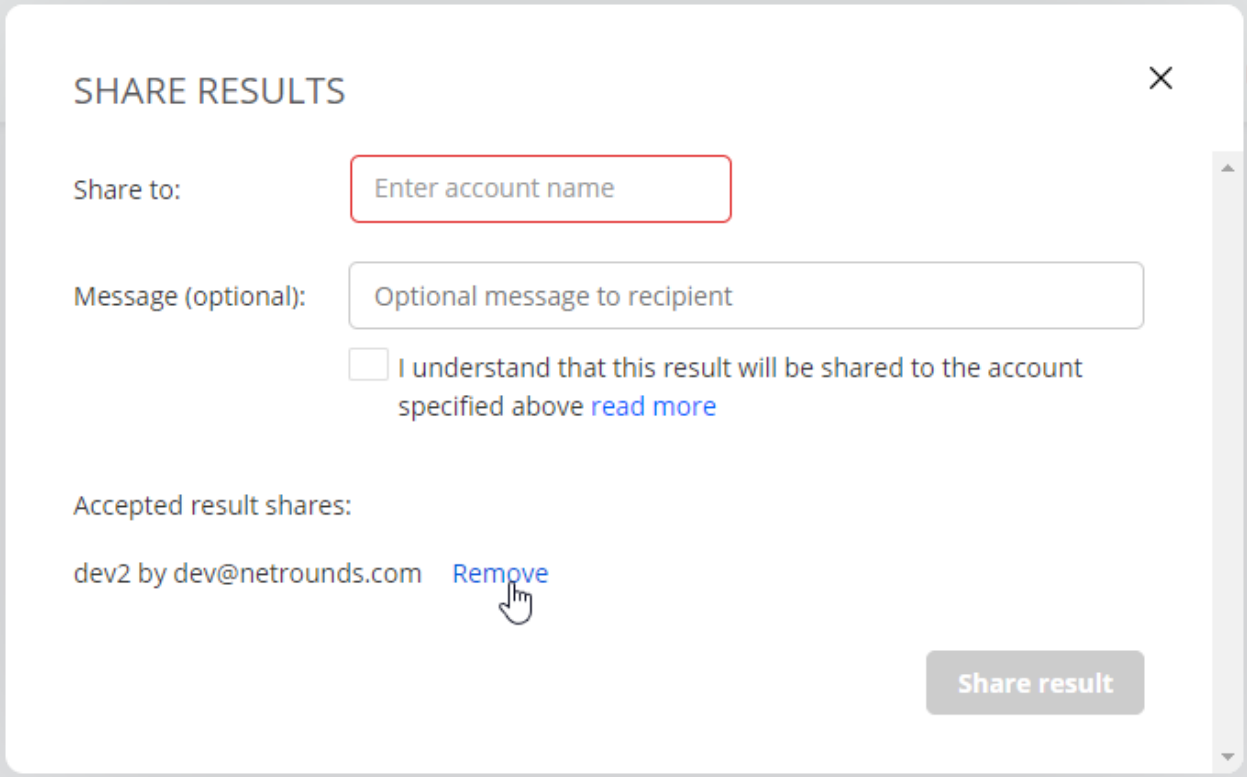
 Clicking the “shared from” icon for the shared monitor displays sharing information:



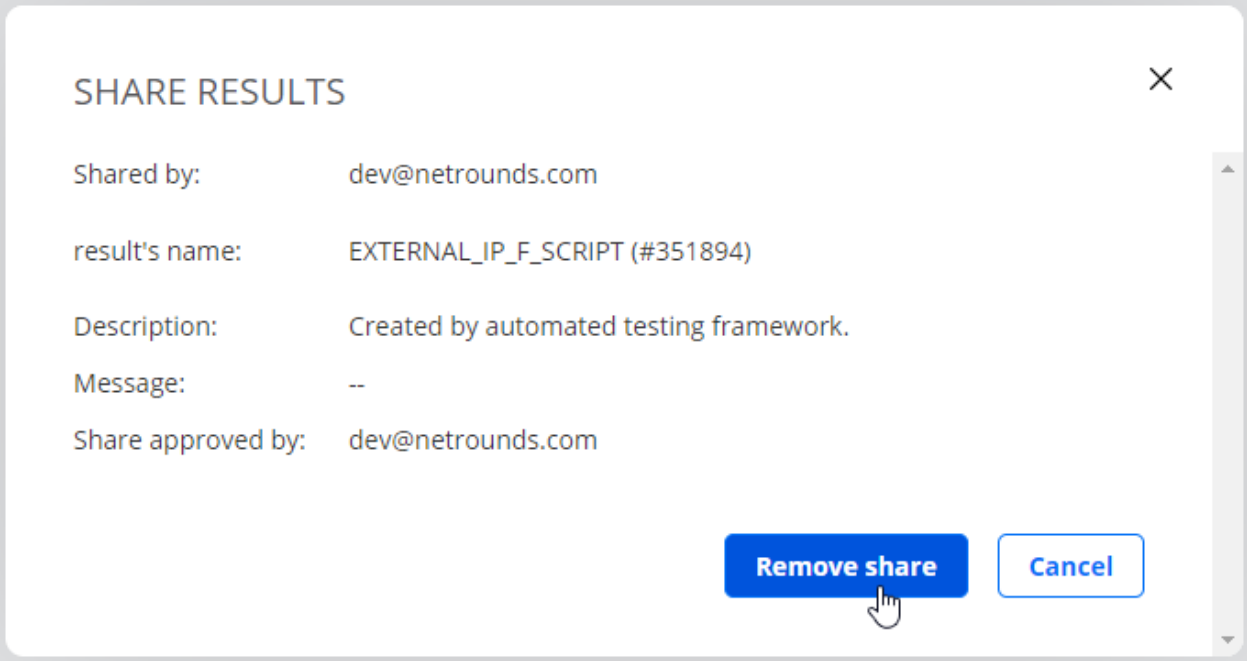
12.4.3 Removing a shared monitor

Both sharer and recipient can remove a share.

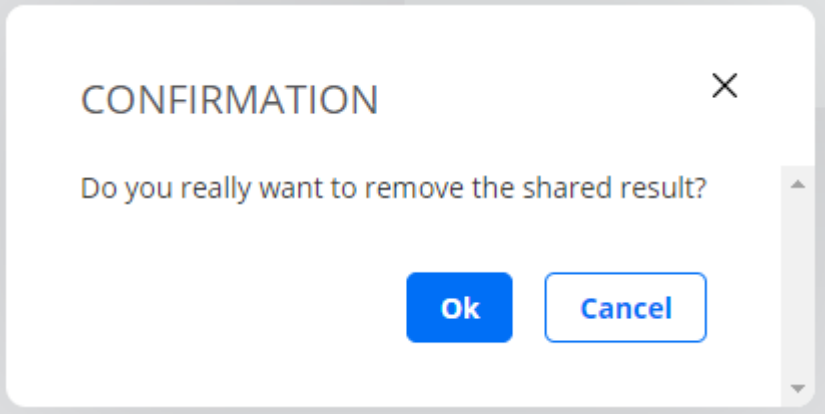
- The sharer clicks the  icon and then, in the dialog that appears, clicks the cross to the right of a monitor to unshare it, as shown in this screenshot:



- The recipient clicks the Remove share button:



In both cases, a confirmation dialog appears:



12.5 Sharing test and monitor results using URLs

You can share test and monitor results by providing URLs to a Control Center GUI showing the results. Optionally, you can have the URLs emailed to listed recipients.

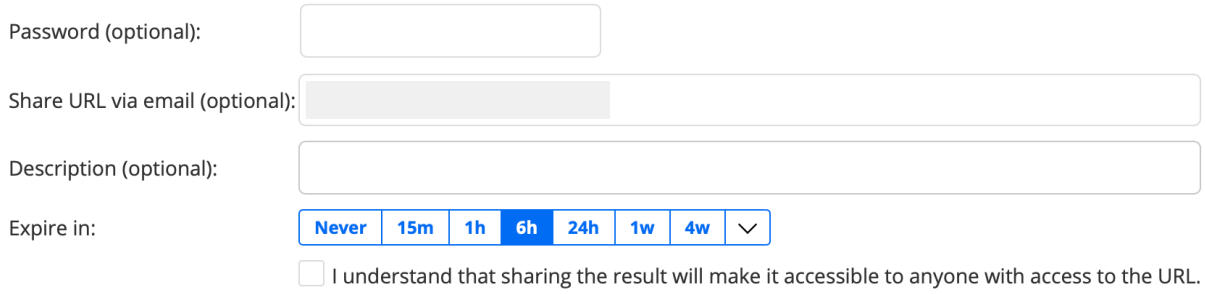
The share and unshare procedures are the same for tests and monitors, so only sharing of monitoring results is described here.

12.5.1 How to share a monitor

- Navigate to Monitoring in the main menu.


In the Share column, click the icon for the monitor you want to share.

- Click the Share using URL button.



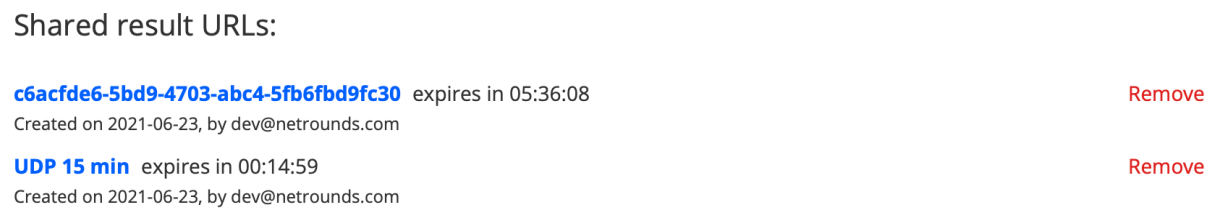
All settings except the expiry setting are optional.

- Password: You can protect the share with a password. The recipient will then need to enter the password in order to view the shared content.
- Share URL via email: Here you can enter email addresses to which the shared URL will be sent.
- Description: If you enter text here, it will appear in the URL link. If you do not enter a description, the link text will consist of a UUID string.

- Expire: Here you set the lifespan of the share. For example, 15m means that monitor results will be shared for 15 minutes from the moment you activate the share (see below). You can also click the down arrow to set a custom expiry date and time. If you want to keep sharing the test or monitor indefinitely, click Never. The default is Never.
- Check the I understand… checkbox. This is necessary to enable sharing. Please be aware that anyone with access to the shared URL will be able to inspect the test or monitor.
- Finally, click the Share result via URL button at the bottom.

You will be notified as to whether the sharing was successful or not.

Links to the shared results appear in a list in the Share results dialog:



In this example, no description has been entered for the first share, so that the link consists of a UUID. For the second share, a description has been provided.

12.5.2 How a shared monitor is presented to the recipient

To protect the sharer's integrity, the GUI shown to the recipient is stripped of all account-specific details. Only the test or monitor results themselves are presented, accompanied by the time interval selectors and the Report and Export buttons.

12.5.3 Removing a share

- To remove a share, click the Remove link for the share in the Shared result URLs section of the dialog.

13 Using a proxy

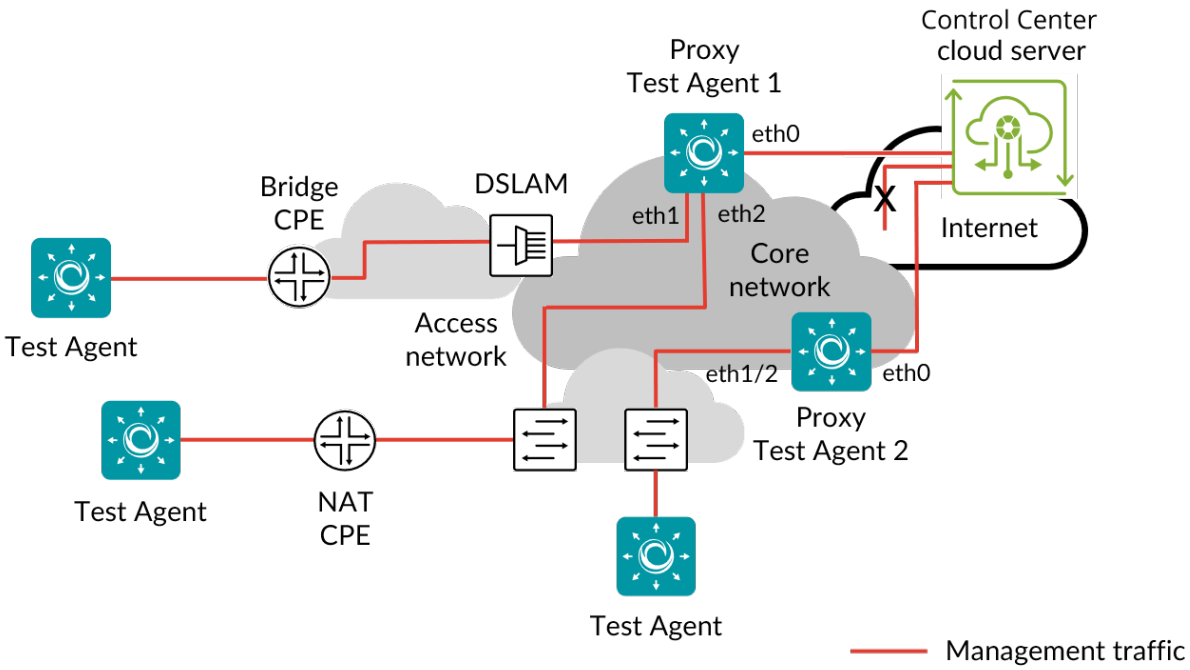
13.1 Using a proxy

The Routing Active Testing server needs a direct connection to all Test Agents to be able to conduct measurements and collect data. A *proxy* may be used to connect to Test Agents that cannot be reached by other means. The proxy can be either another Test Agent or a standard HTTP proxy. The two possibilities are discussed in turn below.

13.1.1 Using a Test Agent as proxy

Using one of your Test Agents as a proxy makes it possible to run tests with Test Agents that would otherwise not be reachable from the Internet. You might, for example, want to test and monitor an IP telephony network that does not allow any connection to the Internet. Other examples are IPTV networks and VPN connections.

The Routing Active Testing proxy is basically a forwarding proxy that sets up a forwarding table between the two interfaces of the Test Agent. The original "eth0" interface should be connected to the Routing Active Testing server, whilst the other interface is the new connection point for the Test Agents inside the network. The proxy will only forward needed management traffic between the Routing Active Testing server and the Test Agents to and from the closed network. There is no possibility of reaching anything other than the management interface on the Test Agents, i.e. "eth0".



For details on how to set up a Test Agent as proxy, see *this page* (page 486).

Routing Active Testing supports the use of multiple proxy Test Agents; one per access or metro network could be a natural allocation. Each of the proxy Test Agents can handle at least 15 regular Test Agents that are simultaneously running tests or monitors. A dedicated HW Small Test Agent in proxy mode is estimated to be capable of handling up to 100 Test Agents. For assistance in setting up such a maximum configuration, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

13.1.2 Using a standard HTTP proxy

It is also possible to connect a Test Agent to Control Center through a standard HTTP proxy. This is necessary if Test Agents are located behind a firewall in your network environment and if access to the Internet is required to pass through an HTTP proxy.

For HTTP proxy authentication, the modes “none” and “basic” are both supported.

You configure a Test Agent to use an HTTP proxy from the local console. See *this page* (page 209).

Please note that you cannot register a Test Agent with the Control Center via an HTTP proxy; this registration must be done in the usual manner, as explained *here* (page 205). Once registered, however, the Test Agent can use an HTTP proxy to connect.

14 Definitions and technical notes

14.1 Abbreviations

Term	Meaning
AMF	Access and Mobility Management Function
APN	Access Point Name
ARP	Address Resolution Protocol
AS	Autonomous System
ASN	Autonomous System Number
BB	Bottleneck Bandwidth
BDP	Bandwidth Delay Product
BGP	Border Gateway Protocol
BPDU	Bridge Protocol Data Unit
BSS	Business Support System
CC	Continuity Count
CDP	Cisco Discovery Protocol
CESoE	Circuit Emulation Services over Ethernet
CFM	Connectivity Fault Management
CGMP	Cisco Group Management Protocol
CIR	Committed Information Rate
CPE	Customer-Premises Equipment
CRC	Cyclic Redundancy Check
CSS	Cascading Style Sheets
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
DS	Differentiated Services
DSCP	Differentiated Services Code Point
DSLAM	Digital Subscriber Line Access Multiplexer
DTP	Dynamic Trunking Protocol
DUT	Device Under Test
DV	Delay Variation
EAPoL	Extensible Authorization Protocol over LAN
Ec/Io	Per-chip signal-to-noise ratio
EIGRP	Enhanced Interior Gateway Routing Protocol
EIR	Excess Information Rate
EPC	Evolved Packet Core
ES	Errored Second(s)
ESMC	Ethernet Synchronization Messaging Channel
ETH	Ethernet
EVC	Ethernet Virtual Connection
EVPL	Ethernet Virtual Private Line
EVP-LAN	Ethernet Virtual Private LAN
FEC	Forward Error Correction
FM	Fault Management
FTP	File Transfer Protocol
GSM	Global System for Mobile Communication
GSMP	General Switch Management Protocol
HLS	HTTP Live Streaming

continues on next page

Table 14.1 – continued from previous page

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IMAP	Internet Message Access Protocol
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPP	IP Precedence
IS-IS	Intermediate System – Intermediate System
ISP	Internet Service Provider
KPI	Key Performance Indicator
L2CP	Layer 2 Control Protocols
LACP	Link Aggregation Control Protocol
LAMP	Link Aggregation Marker Protocol
LAN	Local Area Network
LLC	Logical Link Control
LLDP	Link Layer Discovery Protocol
LTE	Long Term Evolution
MAC	Medium Access Control
MANO	Management and Orchestration
MBH	Mobile Backhaul
MC	Multicast
MCC	Mobile Country Code
ME	Maintenance Entity
MEF	Metro Ethernet Forum
MEG	Maintenance Entity Group
MEL	MEG (Maintenance Entity Group) Level
MEP	MEG (Maintenance Entity Group) End Point
MIB	Management Information Base
MIP	MEG (Maintenance Entity Group) Intermediate Point
MITM	Man In The Middle
MLD	Multicast Listener Discovery
MNC	Mobile Network Code
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MPLS	Multiprotocol Layer Switching
MPTS	Multi Program Transport Stream
MSTP	Multiple Spanning Tree Protocol
MVRP	Multiple VLAN Registration Protocol
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NDP	Neighbor Discovery Protocol
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NG-RAN	Next Generation Radio Access Network
NIC	Network Interface Controller (*or:* Card)
NMS	Network Management Station
NOC	Network Operations Center

continues on next page

Table 14.1 – continued from previous page

NTP	Network Time Protocol
OAM	Operations, Administration, and Management (*or:* Maintenance)
OCA	Open Connect Appliance
OSPF	Open Shortest Path First (protocol)
OSS	Operations Support System
OTT	Over-The-Top
OUI	Organizationally Unique Identifier
OWAMP	One-Way Active Measurement Protocol
PAT	Program Association Table
PCAP	Packet Capture
PCP	Priority Code Point
PCR	Program Clock Reference
PD	Packet Delay
PDU	Protocol Data Unit
PDV	Packet Delay Variation
PGW	Packet data network Gateway
PID	Packet Identifier
PLMN	Public Land Mobile Network
PLR	Packet Loss Ratio
PM	Performance Management
PMT	Program Map Table
PPP	Point-to-Point Protocol
PPPoE	Point-to-Point Protocol over Ethernet
PVST	Per-VLAN Spanning Tree
QAM	Quadrature Amplitude Modulation
Q-in-Q	Informal name for IEEE 802.1ad, amendment to IEEE 802.1Q-1998
QoS	Quality of Service
RAN	Radio Access Network
RARP	Reverse Address Resolution Protocol
RAT	Radio Access Technology
RGW	Residential Gateway
RIPv2	Routing Information Protocol version 2
RPCAP	Remote Packet Capture
RRH	Remote Radio Head
RSCP	Received Signal Code Power
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indication
RSTP	Rapid Spanning Tree Protocol
RTP	Real-time Transport Protocol
RTT	Round-Trip Time
SaaS	Software as a Service
SAP	Service Access Point
SAVI	Source Address Validation Improvement
SCTP	Stream Control Transmission Protocol
SD	Slice Differentiator
SDN	Software-Defined Networking
SES	Severely Errored Second(s)
SFD	Start of Frame Delimiter
SFP	Small Form-factor Pluggable (transceiver)

continues on next page

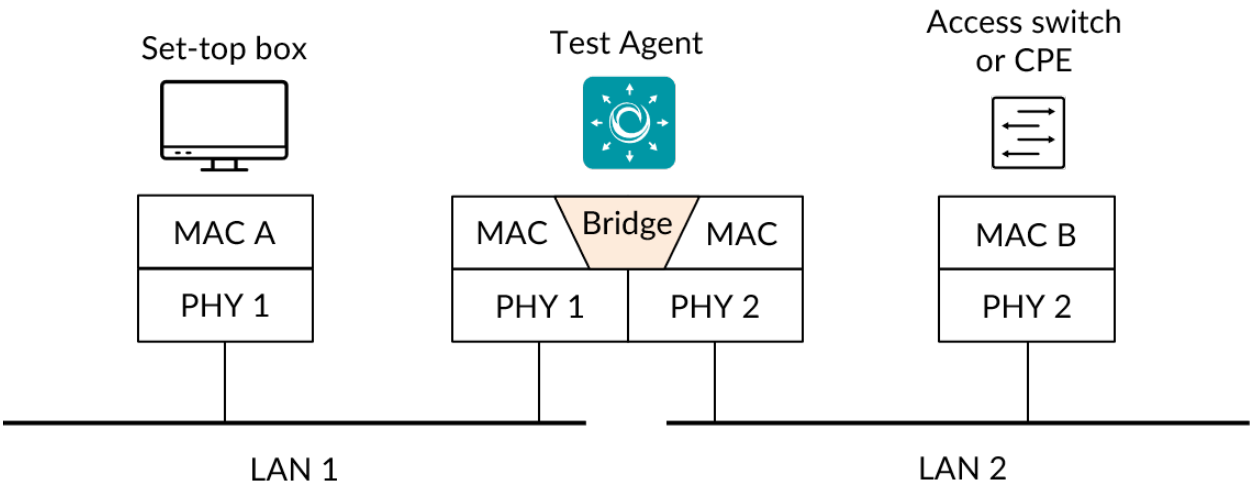
Table 14.1 – continued from previous page

SINR	Signal to interference-plus-noise ratio
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLAAC	Stateless Address Autoconfiguration
SLM	Synthetic Loss Measurement
SLS	Service Level Specification
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SP	Service Provider
SSH	Secure Shell
SSL	Secure Sockets Layer
SST	Slice/Service Type
STB	Set-Top Box
STP	Spanning Tree Protocol
TA	Test Agent
TAC	Tracking Area Code
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOS	Type of Service
TS	Transport Stream
TTL	Time To Live
TWAMP	Two-Way Active Measurement Protocol
UAS	Unavailable Seconds
UDP	User Datagram Protocol
UNI	User–Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTF	UCS (Universal Character Set) Transformation Format
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNF	Virtualized Network Function
VoIP	Voice over IP
vTA	Virtual Test Agent
VTP	VLAN Trunk Protocol
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
Y.1563	ITU-T Y.1563: Ethernet frame transfer and availability performance
Y.1564	ITU-T Y.1564: Ethernet service activation test methodology
Y.1731	ITU-T Y.1731: Performance monitoring in a service provider network

14.2 Bridge, bridging

Network bridging describes the action of network equipment that allows two or more Layer 2 networks, or two or more Layer 2 network segments, to create an aggregate network. Bridging is distinct from *routing*, which allows the networks to communicate independently as separate networks. A network bridge is a network device that connects multiple network segments.

In the picture below, a Test Agent in Routing Active Testing is configured and running in bridge mode. Test Agents support bridging of physical interfaces. This particular setup is used for *inline IPTV testing* (page 324) and allows all IPTV traffic to pass through the Test Agent, which measures quality on the same channels as the set-top box joins.



14.3 DSCP/DiffServ and IP Precedence

Differentiated Services Code Point (DSCP) is a means of classifying and managing network traffic and of providing quality of service (QoS) in modern Layer 3 IP networks. It uses the 6-bit Differentiated Services (DS) field in the IP header for the purpose of packet classification.

IP Precedence is another means to classify and differentiate traffic in a Quality of Service enabled network. The relationship between DSCP and IP Precedence is detailed in the table below.

DSCP name	DS (binary)	DS (decimal)	IP Precedence
CS0	000000	0	0
CS1	001000	8	1
AF11	001010	10	1
AF12	001100	12	1
AF13	001110	14	1
CS2	010000	16	2
AF21	010010	18	2
AF22	010100	20	2
AF23	010110	22	2
CS3	011000	24	3
AF31	011010	26	3
AF32	011100	28	3
AF33	011110	30	3
CS4	100000	32	4
AF41	100010	34	4
AF42	100100	36	4
AF43	100110	38	4
CS5	101000	40	5
EF	101110	46	5
CS6	110000	48	6
CS7	111000	56	7

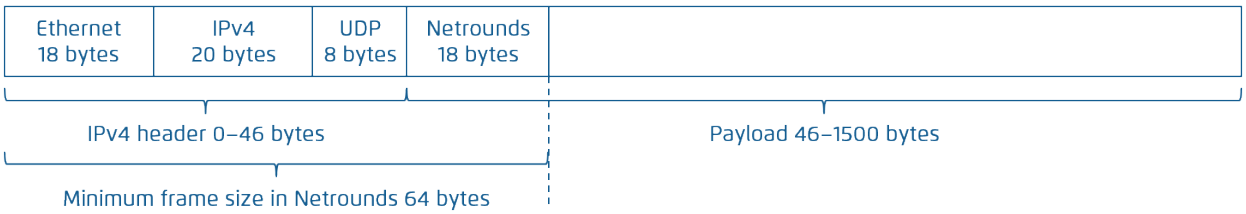
14.4 Layer 2 Ethernet frame sizes

The Layer 2 Ethernet frame as described here includes Ethernet headers, i.e. the CRC, but not the Inter Frame Gap, Preamble, or Start of Frame Delimiter (SFD).

The maximum frame size depends on the interface MTU (Maximum Transmission Unit); the default value is 1500 bytes.

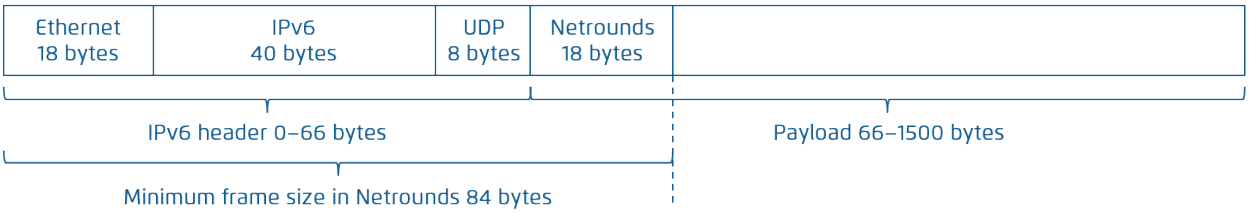
The minimum frame size for IPv4 is 64 bytes, where the Ethernet header takes up 18 bytes, the IPv4 header 20 bytes, and the UDP header 8 bytes. The remaining 18 bytes are payload, where Routing Active Testing places a sequence number, a timestamp, a checksum, and a flow ID.

- **Minimum IPv4 frame size** = 18 (Ethernet) + 20 (IPv4) + 8 (UDP) + 18 (payload) = **64 bytes**



For IPv6 the minimum frame size is 84 bytes, since the IPv6 header is 40 bytes long.

- **Minimum IPv6 frame size** = 18 (Ethernet) + 40 (IPv6) + 8 (UDP) + 18 (payload) = **84 bytes**



14.5 MPEG basics

An MPEG packet is 188 bytes in size, and usually seven MPEG packets are contained in one IP packet. Therefore, if one IP packet is lost, this usually entails the loss of seven MPEG packets. Since packet loss is detrimental to IPTV quality, it should be closely monitored.

Lost MPEG packets are equivalent to Continuity Count (CC) errors. Each MPEG transport stream packet contains a 4-bit counter which continuously increments from 0 to 15, wrapping around to zero on reaching the maximum value. The purpose of the counter is to enable recognition of missing or repeated transport stream packets, thus drawing attention to any multiplexer or IP network problems.

14.6 MPEG metrics

In the course of *IPTV testing* (page 321), several MPEG metrics are calculated and reported by Routing Active Testing. This is done by the Test Agents inspecting the headers of the MPEG stream, including RTP headers.

MPEG loss: This is the MPEG packet loss calculated from the Continuity Count field (4 bits) in the MPEG transport stream header.

Values in the Continuity Count field are required to be sent in order (starting at 0, going up to 7 in increments of one, then wrapping around to 0 and beginning a new cycle). A continuity count error indicates that one of three possible errors has occurred:

- A continuity count value was skipped in the sequence of packets.

- Continuity count values arrived out of order.
- The same continuity count value arrived twice in a row.

Program Clock Reference (PCR) jitter: To enable a decoder to present synchronized content, a Program Clock Reference (PCR) is transmitted in the adaptation layer of the MPEG transport stream. Routing Active Testing uses this timestamp field to calculate the PCR *jitter (delay variation)* (page 474) for the received MPEG stream.

Program Allocation Table (PAT) errors: The Program Allocation Table (PAT) lists all programs available in the MPEG transport stream, where each individual program is identified by its PID and points to a PMT (see below).

According to the standards, a PAT should be received every half second on a multicast group. If no PAT is received within that interval, a PAT error is triggered.

Program Map Table (PMT) errors: The Program Map Table (PMT) contains information about the programs in the MPEG stream. There is one PMT for each program.

According to the standards, a PMT should be received every half second on a program. If no PMT is received within that interval, a PMT error is triggered.

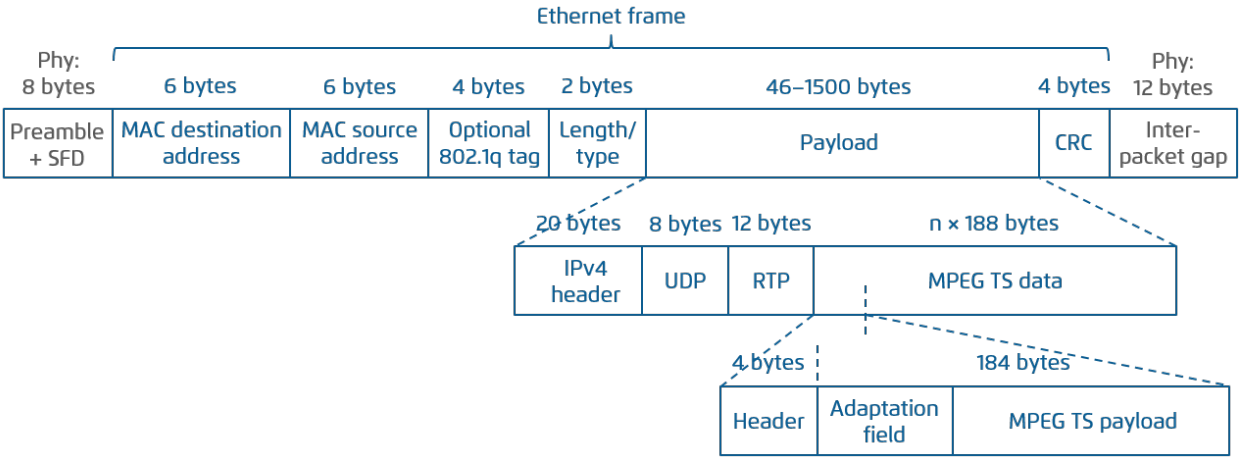
Packet identifier (PID) errors: On regular audio/video MPEG streams, the standards stipulate that a frame should be received once every 5 seconds. If no frame is received for 5 seconds, Routing Active Testing starts counting PID errors (one for every second during which no frame is received).

RTP jitter, loss and misorders: If the MPEG streams contain RTP headers, Routing Active Testing will also calculate RTP jitter, loss, and misorderings, all of which are analogous to the corresponding metrics for IP. Whether or not the MPEG stream contains RTP headers depends on the encoder at the TV head-end.

14.7 MPEG rate vs. MPEG transport rate

This article describes the difference between MPEG rate and MPEG transport rate, both of which are reported in Routing Active Testing.

The picture shows the frame structure for an MPEG transport stream (TS) over Ethernet. The MPEG TS frame is a fixed-length field of 188 bytes, and up to seven of these MPEG TS frames are multiplexed into the payload field of the IP frame.



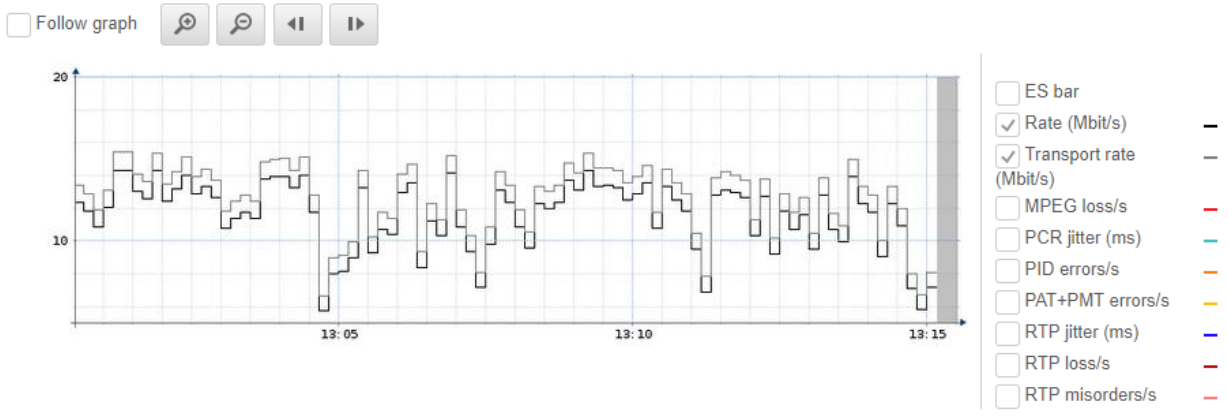
In the table, a theoretical overhead is calculated for MPEG rate and transport rate. This overhead is valid for a single channel; it will be different if several MPEG streams are multiplexed in the MPEG TS.

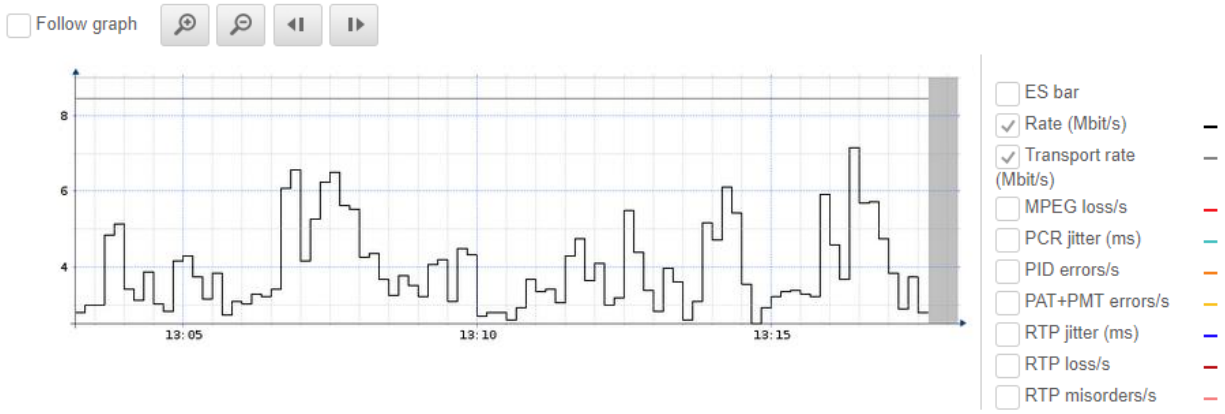
Protocol	Overhead (bytes)	Theoretical overhead (%)
MPEG over Ethernet with 802.1q, with RTP	8 preamble + 14 header + 4 VLAN + 4 CRC + 12 gap + 20 IPv4 + 8 UDP + 12 RTP = 82 bytes/packet MPEG TS header = = 4 × 7 = 28 bytes/packet	Overhead (bytes) / MPEG TS payload (bytes) = (82 + 28) / (184 × 7) = **8.5404%**
MPEG over Ethernet with 802.1q, without RTP	8 preamble + 14 header + 4 VLAN + 4 CRC + 12 gap + 20 IPv4 + 8 UDP = 70 bytes/packet MPEG TS header = = 4 × 7 = 28 bytes/packet	Overhead (bytes) / MPEG TS payload (bytes) = (70 + 28) / (184 × 7) = **7.6087%**

"Rate (Mbit/s)" displayed in the Routing Active Testing view below (showing output from an IPTV MPEG monitoring session) is the MPEG rate (= rate of MPEG stream from the coder) averaged over the chosen interval.

Client/Channel	ES history	Rate (Mbit/s)	Errored Seconds			
			Total	Invalid stream	MPEG loss	Jitter
Crown:br0		27.57	1s (0.11%)	0s (0%)	1s (0.11%)	0s (0%)
Viasat SVT1 HD		11.95	1s (0.11%)	0s (0%)	1s (0.11%)	0s (0%)
Viasat TV3 HD		10.57	0s (0%)	0s (0%)	0s (0%)	0s (0%)
Viasat TV3 SD		5.05	0s (0%)	0s (0%)	0s (0%)	0s (0%)

You can click one of the channels to display a more detailed graph, showing both the MPEG rate (black) and the transport rate (gray). The transport rate is the rate including all Ethernet, IP, and UDP headers. Note that in the second graph, the transport rate is constant.





The difference between the two rates consists of the overhead from the protocol layers. This has to be considered when multiplexing IPTV channels on a link: just adding up the MPEG rates, disregarding the overhead, may result in overloading the link. Any other (non-MPEG) traffic on the link must of course also be taken into account.

14.8 Routing Active Testing server

14.8.1 How sensitive is the management connection to the Routing Active Testing server?

The Test Agents are very robust to network disturbances affecting the management connection from the Test Agent to the Routing Active Testing server. However, like all applications that traverse an IP network, the underlying communication protocols (in this case TCP) do pose some requirements on the network connection.

Our internal tests show that Routing Active Testing starts to become affected at around 10% loss in combination with 100–200 ms one-way delay (200–400 ms round-trip delay). Even higher loss can usually be tolerated without noticeable performance degradation, provided that the delay stays low (one-way delay in the order of tens of milliseconds – not hundreds).

Conditions degrading the performance of the management connection are only seen in networks with significant problems or on low bit rate satellite connections. You will then notice things like slow and/or asynchronous updates of measurement graphs.

14.8.2 Is my measurement performance affected by the geographical distance to my Routing Active Testing server?

No, it is not.

No measurement traffic ever occurs between your Test Agents and the Routing Active Testing server. The Test Agents communicate with the server over an encrypted link, sending collected measurements and receiving control traffic.

Most of the post-processing of the measurement data is done on the Routing Active Testing server. All packet-level processing, however, is done in real time in each Test Agent to achieve best performance and accuracy. The Test Agents periodically upload their data to the Routing Active Testing server for further post-processing and storage.

14.8.3 Where are the Routing Active Testing cloud servers located geographically?

We use Amazon's globally available data centers to host our cloud servers. We are continuously adding more servers on different continents as our customer base grows.

14.9 Priority Code Point (PCP)

Priority Code Point (PCP) is a means of classifying and managing network traffic and of providing quality of service (QoS) in modern Layer 2 Ethernet networks. It uses the 3-bit PCP field in the VLAN header for the purpose of packet classification.

The PCP field was introduced by the IEEE P802.1p working group.

Note: When a Test Agent Application attempts to configure PCP settings in outgoing IP packets, it cannot be guaranteed that the settings are indeed carried through. This is because the Test Agent Application does not control the host it is running on and its interface configurations.

To ensure that PCP settings are applied as intended when using a Test Agent Application, you need to manually map (on the host OS) the `SO_PRIORITY` bits set on the socket to the PCP bits in the VLAN header. Below is an example of a command doing this mapping for VLAN 123 on interface eth0:

```
ip link set eth0.123 type vlan egress 0:0 1:1 2:2 3:3 4:4 5:5 6:6 7:7
```

Warning: Making this kind of changes may cause malfunctions on the host. You need to be fully aware of the consequences of the command you intend to run.

Read more about how to use `ip link set` here: https://manpages.ubuntu.com/manpages/artful/man8/ip-link.8.html

To learn more about `SO_PRIORITY`, see this article: https://man7.org/linux/man-pages/man7/socket.7.html

14.10 SNMP

Routing Active Testing supports both SNMPv2c and SNMPv3 trap messages for sending alarms or error conditions to external systems. How to set up such alarms is covered on *this page* (page 42).

The SNMP traps are sent from the host where Control Center is installed; no SNMP traps are sent directly from any Test Agent.

14.10.1 Version 2c

The SNMPv2 implementation in Routing Active Testing follows ► IETF RFC 1901.

- IP address: The IP address of the SNMP manager (trap sink).
- Port: The port of the SNMP manager (trap sink). Default: 162.
- Community: The community string used for authentication.

14.10.2 Version 3

The SNMPv3 implementation in Routing Active Testing follows ► IETF RFC 3410.

- IP address: The IP address of the SNMP manager (trap sink).
- Port: The port of the SNMP manager (trap sink). Default: 162.
- Community: The community string used for authentication.
- Engine ID: The engine id to use. This should be the same in |paa-product| and in the manager.
- User name: The user name to use for authentication.
- Security: The security level to use for sent traps. The permitted security levels are: *No encryption* (noAuthNoPriv); *Authentication only* (authNoPriv); *Authentication and privacy/encryption* (authPriv).
- Authentication password: Password to use for authentication protocol, minimum 8 characters.
- Authentication protocol: Protocol to use for authentication. The permitted protocols are: *MD5*; *SHA*; *SHA-224*; *SHA-256*; *SHA-384*; *SHA-512*
- Privacy password: Password to use for privacy protocol, minimum 8 characters.
- Privacy protocol: Protocol to use for privacy. The permitted protocols are: *DES*; *AES*; *AES-192*; *AES-256*

14.11 TCP implementation in Routing Active Testing

Test Agents use the CUBIC TCP implementation. CUBIC uses an optimized congestion control algorithm for high-bandwidth and high-latency networks.

For more information on CUBIC, see ► tools.ietf.org/html/draft-rhee-tcp-cubic-00.

Routing Active Testing does not tweak the Linux default settings in any major way, as the PC would then no longer behave as a normal user PC – which it should, since quite commonly a measurement is made precisely in order to detect or prevent possible user experience problems.

To select the TCP window size, Routing Active Testing uses CUBIC's window scaling algorithm. The window size therefore varies, but Routing Active Testing does have default and maximum window sizes predefined.

Packet sizes, too, are controlled by CUBIC. Normally, the packets will have full size (around 1500 bytes).

For Test Agent Applications, the TCP implementation and settings depend on the operating system of the platform used.

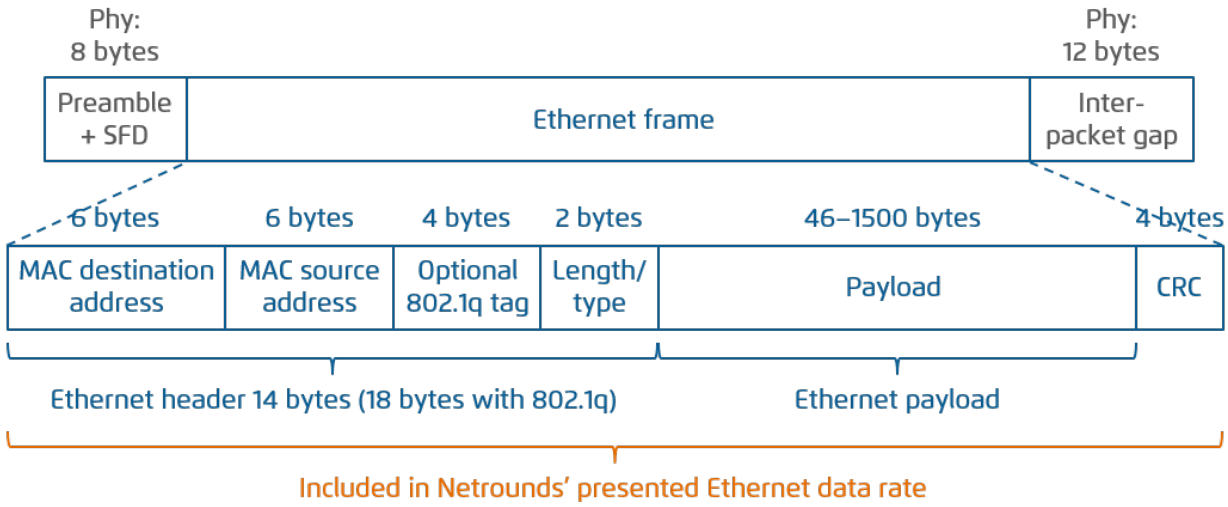
Note, finally, that when validating performance it is often useful to complement TCP testing with UDP measurements.

14.12 Theoretical maximum throughput with Routing Active Testing Test Agents

This article discusses theoretical maximum throughput in Routing Active Testing as compared to line rate, differences being due to protocol overhead. We frame our theoretical reasoning in the context of Test Agents and their expected maximum performance. Besides overhead, there are other limiting factors which are not considered here, such as CPU limitations and overbooking of network links.

Theoretical performance is calculated for Ethernet and for TCP and UDP over Ethernet. Read more about Ethernet frame sizes *here* (page 512).

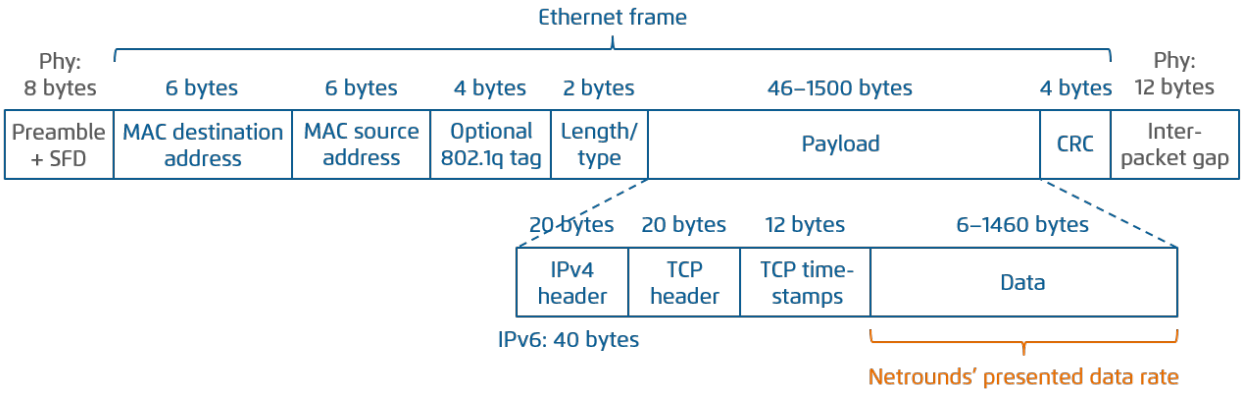
14.12.1 Ethernet data rates



The following table shows theoretical maximum Ethernet data rates in Routing Active Testing UDP performance tests as percentages of the line rate. In these calculations, the header and CRC are included in the data rate. This is also the basis for the absolute Ethernet data rates presented in Routing Active Testing.

Protocol	Ethernet overhead (bytes)	Theor. max. data rate (%)
Ethernet, no 802.1q, MTU = 1500 bytes	8 preamble + 14 header + 4 CRC + 12 gap = 38 bytes/packet	(1500 + 18) / (1500 + 38) = **98.6996%**
Ethernet, no 802.1q, MTU = 494 bytes	8 preamble + 14 header + 4 CRC + 12 gap = 38 bytes/packet	(494 + 18) / (494 + 38) = **96.2406%**
Ethernet, no 802.1q, MTU = 46 bytes	8 preamble + 14 header + 4 CRC + 12 gap = 38 bytes/packet	(46 + 18) / (46 + 38) = **76.1905%**
Ethernet with 802.1q, MTU = 1500 bytes	8 preamble + 18 header + 4 CRC + 12 gap = 42 bytes/packet	(1500 + 22) / (1500 + 42) = **98.7030%**
Ethernet with 802.1q, MTU = 494 bytes	8 preamble + 18 header + 4 CRC + 12 gap = 42 bytes/packet	(494 + 22) / (494 + 42) = **95.5224%**
Ethernet with 802.1q, MTU = 46 bytes	8 preamble + 18 header + 4 CRC + 12 gap = 42 bytes/packet	(46 + 22) / (46 + 42) = **72.7273%**

14.12.2 TCP over Ethernet data rates

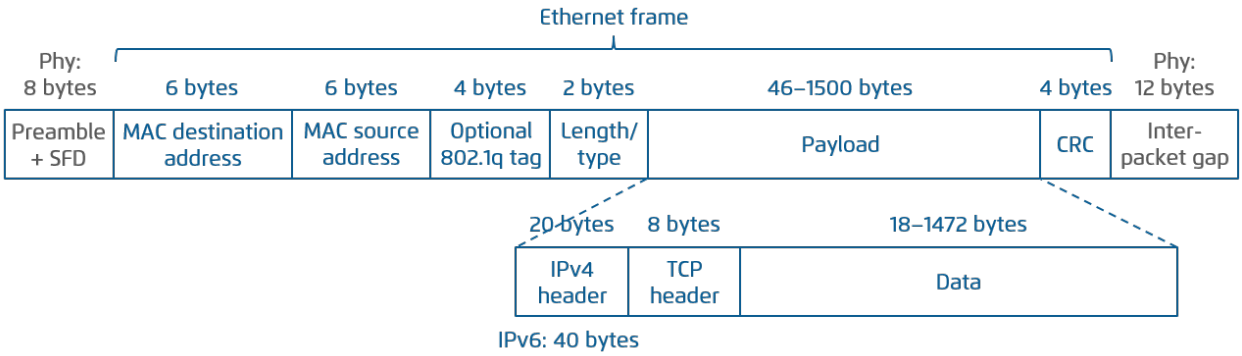


The following table shows theoretical maximum TCP data rates in Routing Active Testing TCP performance tests as percentages of the line rate. In these calculations, only TCP data ("Data" in the above figure) counts as payload. This is also how absolute TCP data rates are presented in Routing Active Testing.

It is assumed throughout that no header compression takes place.

Protocol	TCP/IP overhead (bytes)	Theor. max. data rate (%)
TCP over Ethernet, no 802.1q, IPv4	20 IPv4 + 20 TCP + 12 timestamps = 52 bytes/packet	(1500 – 52) / (1500 + 38) = **94.1482%**
TCP over Ethernet, 802.1q, IPv4	20 IPv4 + 20 TCP + 12 timestamps = 52 bytes/packet	(1500 – 52) / (1500 + 42) = **93.9040%**
TCP over Ethernet, no 802.1q, IPv6	40 IPv6 + 20 TCP + 12 timestamps = 72 bytes/packet	(1500 – 72) / (1500 + 38) = **92.8479%**
TCP over Ethernet, 802.1q, IPv6	40 IPv6 + 20 TCP + 12 timestamps = 72 bytes/packet	(1500 – 72) / (1500 + 42) = **92.6070%**

14.12.3 UDP over Ethernet data rates



The following table shows theoretical maximum data rates in UDP tests as percentages of the line rate. These percentages are provided as a reference; no such figures are presented in Routing Active Testing. Rather, the data rate presented for UDP flows is the Ethernet data rate (see *above* (page 518)).

Protocol	UDP/IP overhead (bytes)	Theor. max. data rate (%)
UDP over Ethernet, no 802.1q, IPv4	20 IPv4 + 8 UDP = 28 bytes/packet	(1500 – 28) / (1500 + 38) = **95.7087%**
UDP over Ethernet, 802.1q, IPv4	20 IPv4 + 8 UDP = 28 bytes/packet	(1500 – 28) / (1500 + 42) = **95.4604%**
UDP over Ethernet, no 802.1q, IPv6	40 IPv6 + 8 UDP = 48 bytes/packet	(1500 – 48) / (1500 + 38) = **94.4083%**
UDP over Ethernet, 802.1q, IPv6	40 IPv6 + 8 UDP = 48 bytes/packet	(1500 – 48) / (1500 + 42) = **94.1634%**

14.13 VLAN

Routing Active Testing supports both untagged and tagged interfaces on Test Agents (except on Test Agent Applications). VLAN tagging is defined in the ► IEEE 802.1Q standard. A Test Agent can have at most 125 simultaneous VLANs defined. For large numbers of VLANs, a HW Large Test Agent is required.

For Ethernet activation tests, Routing Active Testing supports VLAN stacking (Q-in-Q) according to the ► IEEE 802.1ad standard. It is not possible to generate synthetic traffic (UDP and TCP) using VLAN stacking.

15 Release notes

15.1 Release notes, Routing Active Testing software version 3.0.0

This is the first release of the product under the name Routing Active Testing following the acquisition of Netrounds by Juniper Networks.

15.1.1 New features

15.1.1.1 NFX150 added as officially supported Test Agent hardware

Test Agent Appliances can now be installed on Juniper's NFX150-C-S1 Network Services Platform.

The hardware previously used for preinstalled Test Agents is no longer available for purchase. Please note that Test Agents will not be sold preinstalled on the NFX150; rather, the installation needs to be done separately.

15.1.1.2 Juniper licensing alignment

The licensing for Routing Active Testing will be handled somewhat differently from this release onward. For you as a customer, the new procedure boils down to these steps:

- After you install Control Center, the system will be disabled until a license has been activated.
- Juniper will send you one or more pre-generated licenses.
- You install the license file(s) in the Control Center using the `ncc license activate` command (as before).

15.1.1.3 Flexible external IP support

Control Center used to offer a setting "Use public address" which allowed a Test Agent to act as a server in measurements, even if it is behind a NAT router. However, this setting has now been replaced with more flexible external IP support:

- Possible to set in each individual task if the private IP or external IP should be used (the old setting was configured globally for the Test Agent)
- Works for all interfaces (not just the management interface)
- Uses the last public address by default
- Possible to manually override the external IP used
- Works for both Test Agent Appliance and Test Agent Application
- Works for both IPv4 and IPv6

15.1.1.4 Restyling of user interface

The look and feel of the Control Center user interface has been modified with respect to UX design, fonts, and color schemes.

15.1.1.5 Restyling of technical documentation

The layout of the in-app help and standalone PDF documents has been redesigned, and names and wordings have been changed where necessary.

15.1.2 Improvements

None

15.1.3 Bug fixes

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.1.4 Removed features

- The DVB-C functionality has been dropped.

15.2 Release notes, Routing Active Testing software version 3.1.0

15.2.1 Introduction

15.2.1.1 General notes

- Please note that not all features of Routing Active Testing are available to every customer; rather, availability is dependent on what product or service has been purchased.

15.2.1.2 Upgrading to this version

- In this version, an additional plugin package needs to be installed. For details, see the Upgrade Guide.
- This version introduces a new configuration file `/etc/netrounds/plugin.yaml`. During installation, this file needs to be updated with the correct database connection details if the latter have been changed from the default.
- Backup procedures need to be extended to cover the new technologies introduced in this version. For full details, see the Operations Guide, chapter “Backing Up Product Data”.

15.2.2 New and changed features

15.2.2.1 Streaming API

A streaming API and client have been added to Routing Active Testing. This API allows export of all data in Routing Active Testing to an external system. Data streaming is managed by the well-known and highly scalable Kafka event streaming platform.

Using the streaming API is optional, and it is offered only for on-premise Control Center installations.

Please note that the streaming API has only been tested with up to 15,000 streams.

15.2.2.2 Test Agent on Raspberry Pi

Routing Active Testing supports installation of Test Agent Applications on Raspberry Pi platforms.

15.2.2.3 Test Agent for ARM architecture

Routing Active Testing supports installation of Test Agent Applications on ARM processor hardware (32-bit and 64-bit architectures). These Test Agents work identically to the 64-bit AMD versions which have been previously (and are still) offered.

15.2.2.4 DNS testing with Test Agent Application

The DNS test and monitor task has been implemented for Test Agent Applications. It works the same way as for Test Agent Appliances.

15.2.2.5 Two configurable TWAMP percentiles

It is now possible to specify and present two *percentiles* for TWAMP delay values to get a clearer view of their distribution. Commonly used percentiles include the 90th and 99th, which mean, respectively, "90% (99%) of the data points are below this value". The percentiles in Routing Active Testing are however freely configurable.

15.2.2.6 Configurable sender port for TWAMP

A sender (source) port can optionally be set for a TWAMP reflector in the Routing Active Testing inventory. If no sender port is set, the system will use a random port as before.

This is useful for reflectors that require test and source ports to be known in advance.

15.2.2.7 Sharing test and monitor results via URLs

Routing Active Testing already has a mechanism for sharing measurement results (and more) between accounts. This has now been supplemented by a function for sharing test and monitor results via URLs.

Such a function is essential in a scenario with multiple Control Centers. It is also handy for sharing data with an external party that does not have an account in Routing Active Testing. The shared view of a test or monitor is stripped of account-specific details.

15.2.2.8 Moving a Test Agent to a different Control Center

If you have multiple Routing Active Testing Control Centers deployed, you can now move a Test Agent permanently from one Control Center to another by a simple action in the Control Center GUI. You specify the new Control Center host, the port to connect to, and the new credentials.

15.2.2.9 Offline registration of Test Agents

In certain situations, especially when deploying a large number of Test Agents, you may find it convenient to first preconfigure the Test Agents with registration details without being dependent on network connectivity, and register them only later. This can now be done using an Offline registration utility in the Test Agent local console.

15.2.2.10 Test Agent Application: Register and start in one command

When installing a Test Agent Application as a native app in Linux, you can use a new shortcut (`register-run` command) which first registers the Test Agent with Control Center and then starts it.

15.2.2.11 Dynamic plugins

Plugins are used by Test Agent Applications to execute test and monitoring tasks. It is now possible to upload new plugins for use dynamically, that is, in between releases, and also to switch more easily between different versions of a particular plugin.

15.2.2.12 Improved dashboard rendering performance

To speed up the presentation on the Control Center dashboard for large sets of monitors, the monitor list is now by default sorted by a precomputed and cached 15-minute SLA indicator. SLA indicators for other time intervals can be displayed in parallel on the dashboard.

15.2.2.13 DNS: Request lifetime expiry reporting

A new DNS KPI, “ES lifetime”, keeps track of how often no DNS response was obtained before the “Request lifetime” period expired. Contrasting this with the “ES response” KPI, you can distinguish late responses from instances where no response was received at all.

15.2.2.14 Tech preview: TimescaleDB

Note: This feature is made available in version 3.1.0 as a tech preview. This means that it is offered for customers to inspect and explore, while Juniper Networks does not guarantee satisfactory performance and does not take responsibility for malfunctions or data loss. By default the feature is disabled.

A TimescaleDB time-series database is introduced in Routing Active Testing to provide enhanced ingestion performance and scalability. More specifically, it enables connecting an external dashboard and retrieving metrics.

Saving data in the TimescaleDB database is optional; please note that if you turn this on, data will be saved in two places in parallel. The long-term plan for data storage in Routing Active Testing is for TimescaleDB to replace the existing time-series database.

TimescaleDB is offered only for on-premise Control Center installations.

15.2.3 Deprecated features

15.2.3.1 Authentication using TACACS+

Authentication by means of TACACS+ is no longer supported. Going forward, Routing Active Testing supports authentication using LDAP.

15.2.4 Known issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.2.5 Resolved issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.3 Release notes, Routing Active Testing software version 3.2.0

15.3.1 Introduction

15.3.1.1 General notes

- Please note that not all features of Routing Active Testing are available to every customer; rather, availability is dependent on what product or service has been purchased.

15.3.2 New and changed features

15.3.2.1 Collection of Junos TWAMP measurements

Routing Active Testing now has the ability to collect TWAMP measurement results from Juniper routers and switches running Junos. The Test Agent connects to the Juniper device via NETCONF, evaluates errored second criteria, and reports all data back to Control Center.

15.3.2.2 Fully scalable streaming API

The streaming API now supports up to 100,000 concurrent streams.

15.3.2.3 Path trace BGP AS reporting

A path taken by a packet through a network to a specified destination can be defined in terms of a BGP AS path, which indicates the autonomous systems (ASes) that routing information passed through to get to the destination.

The Path trace tool has been augmented with a feature that has the Test Agent report autonomous systems (BGP AS numbers) discovered along the path. This allows you to determine, for example, whether the path is the intended standard one or deviates from the standard.

15.3.2.4 Lower packet rates configurable in Path trace tool

The minimum traceroute packet rate in the Path trace tool has been lowered to 0.1 pps. This lets you prevent artificial packet loss from being reported when traceroute packets encounter a router which limits its ICMP TTL exceeded response rates.

15.3.2.5 Path trace in Test Agent Application

The Path trace feature, previously available in the Test Agent Appliance, is now also supported in the Test Agent Application.

15.3.2.6 Periodically active TWAMP streams

Test Agents can now run TWAMP traffic towards reflectors in periodic bursts instead of a continuous stream. The Test Agents then repeats a cycle where it sends TWAMP packets for a given period, then stays silent for the rest of the cycle. The periodicity option allows you to test against a large number of reflectors in round-robin fashion.

15.3.2.7 Full TWAMP for IPv6

Full TWAMP (which uses the TWAMP control protocol) can now be run towards IPv6 reflectors. Previously only TWAMP Light was compatible with IPv6.

15.3.2.8 TWAMP hardware timestamping in Test Agent Application

Test Agent Application now supports the use of hardware timestamps in TWAMP measurements.

15.3.2.9 Enhanced Control Center Web security compliance

The default TLS configuration for Control Center now requires clients to support TLS 1.2 or higher and enables HSTS.

15.3.3 Deprecated features

None

15.3.4 Known issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.3.5 Resolved issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.4 Release notes, Routing Active Testing software version 3.3.0

15.4.1 Introduction

15.4.1.1 General notes

- Please note that not all features of Routing Active Testing are available to every customer; rather, availability is dependent on what product or service has been purchased.

15.4.2 New and changed features

15.4.2.1 Collection of Junos RPM measurements

Routing Active Testing is now able to collect Junos RPM (TCP, UDP, ICMP, and HTTP) measurement results from Juniper routers and switches running Junos. As in the case of Junos TWAMP introduced earlier, the Test Agent connects to the Juniper device via NETCONF, evaluates errored second criteria, and reports all data back to Control Center.

15.4.2.2 Netflix Speedtest

Test Agents in this release can download Netflix test segments over HTTPS from one or several OCAs (Open Connect Appliances). The bandwidth is measured in the course of download or upload to OCA URLs provided by api.fast.com servers and compared to errored second thresholds.

The Netflix Speedtest plugin is available on both Test Agent Application and Test Agent Appliance.

15.4.2.3 Production-ready TimescaleDB

TimescaleDB, introduced as a tech preview in Routing Active Testing 3.1.0, is now offered as a production-ready feature. Please note that TimescaleDB is still by default disabled.

15.4.2.4 Configurable criterion for Unavailable Seconds (UAS)

A period of Unavailable Seconds (UAS) is triggered by a number of consecutive severely errored seconds (SES). Until now, that number has been fixed at 10. In this release, the SES count threshold has been made configurable for the TWAMP task.

15.4.2.5 UDP loopback over IPv6

The UDP loopback task now supports IPv6 as well as IPv4.

15.4.2.6 Automated backup and restore

New `ncc` CLI commands have been implemented for taking a backup of a Routing Active Testing Control Center and for restoring such a system from backup. This simplifies backup and restore procedures.

15.4.3 Deprecated features

None

15.4.4 Known issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.4.5 Resolved issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.5 Release notes, Routing Active Testing software version 3.4.0

15.5.1 Introduction

15.5.1.1 General notes

- Please note that not all features of Routing Active Testing are available to every customer; rather, availability is dependent on what product or service has been purchased.

15.5.2 New and changed features

15.5.2.1 5G core network testing

Test Agent Applications can connect to a 5G core network and run tests and monitors to measure the performance of that network.

The tests and monitors can make use of any task types supported by Test Agent Applications, while connecting to the 5G core network is done by emulating a gNodeB and one or several UEs and setting up a tunnel interface for data transmission for each UE.

Standard 5G KPIs are reported, such as 5G standalone UE-to-gNB RRC setup time, SCTP connection time, and NG setup time.

15.5.2.2 Revised filtering of alarms

The filtering of alarms in the Control Center GUI has been improved. It is now done according to the following definitions:

- *Active alarms:* All alarms that are currently active (have not been cleared or suppressed).
- *Manually suppressed:* All alarms that have been active during the selected time period and have been suppressed.
- *Auto-cleared:* All alarms that have been cleared during the selected time period and have not been suppressed.

15.5.3 Deprecated features

None

15.5.4 Known issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.5.5 Resolved issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.6 Release notes, Routing Active Testing software version 4.0.0

15.6.1 Introduction

15.6.1.1 General notes

- Please note that not all features of Routing Active Testing are available to every customer; rather, availability is dependent on what product or service has been purchased.

15.6.2 New and changed features

15.6.2.1 Running Test Agents on Juniper routers

This release adds the ability to install and run Test Agent Applications on select Juniper ACX routers. Specifically, the ACX models supporting the Test Agent are:

- ACX7100-32C, ACX7100-48L
- ACX7509

This feature requires Junos OS 22.3R1-EVO or later.

The Test Agent runs in Junos OS Evolved as a third-party container. The following active assurance measurement tasks are supported:

- UDP
- TCP
- HTTP
- DNS
- Ping
- Path trace
- IPTV
- OTT video

15.6.3 Deprecated features

None

15.6.4 Known issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.6.5 Resolved issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.7 Release notes, Routing Active Testing software version 4.1.0

15.7.1 Introduction

15.7.1.1 General notes

- Please note that not all features of Routing Active Testing are available to every customer; rather, availability is dependent on what product or service has been purchased.

15.7.2 New and changed features

15.7.2.1 Running Test Agents on ACX7024 (Ultron)

This release adds the ability to install and run Test Agent Applications on the Juniper ACX7024 (Ultron) platform.

This feature requires Junos OS 22.4R1-EVO or later.

The Test Agent runs in Junos OS Evolved as a third-party container. The following active assurance measurement tasks are supported:

- UDP
- TCP
- HTTP
- DNS
- Ping
- Path trace
- IPTV
- OTT video
- System monitor. This task retrieves details on the server on which it is running.

15.7.2.2 Key-value tagging

Previously, Routing Active Testing has offered simple, "one-dimensional" tags for labeling Test Agents and certain other items. This release replaces tags with more sophisticated **key-value** tagging, where you can define multiple keys, each of which can take an arbitrary value. For example, you may want to define a key "country", with country names as values, and another key "operator".

Please note that this feature is not fully backward compatible with the Control Center REST API:

- Tags are now returned as a dictionary instead of a list of strings.
- Tags are now set directly on each taggable resource (such as a Test Agent). The special resource for tag assignment (`/tags/assign/`) is no longer used.

15.7.2.3 Support for UEFI boot

Test Agents now support UEFI boot in addition to legacy boot.

15.7.3 Deprecated features

None

15.7.4 Known issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

15.7.5 Resolved issues

Please refer to the version of these release notes found at https://www.juniper.net/documentation/product/us/en/routing-active-testing.

16 Sales and support

16.1 General sales and support information

In order to purchase or upgrade Routing Active Testing, please contact your Juniper Networks partner if applicable, or else your local Juniper Networks account manager or sales representative.

For support in using Routing Active Testing, please contact Juniper Networks technical support at https://support.juniper.net/support/requesting-support.

End of Engineering and End of Support dates for Routing Active Testing releases are found on this page: https://support.juniper.net/support/eol/software/paragon_aa/.