## Contents

1. **An overview of Netrounds** 5

2. **Getting started** 6
   - 2.1 Easy steps for sign-up to Netrounds 6
   - 2.2 Netrounds getting started wizards 7
   - 2.3 Guided tour of the main menu 7
   - 2.4 Getting started with HTTP and DNS measurements 12
   - 2.5 Getting started with IPTV measurements 13
   - 2.6 Getting started with remote packet capture and traffic analysis 15
   - 2.7 Getting started with network performance measurements 15
   - 2.8 Getting started with your own Speedtest server 17
   - 2.9 Getting started with IP telephony (SIP and VoIP) measurements 18

3. **Setting up your account** 19
   - 3.1 Managing your account 19
   - 3.2 Administering users and permissions 20
   - 3.3 Setting up IPTV channels 21
   - 3.4 Setting up DVB-C channels 22
   - 3.5 Setting up SIP accounts 24
   - 3.6 Setting up Ping hosts 26
   - 3.7 Setting up TWAMP reflectors 29
   - 3.8 Setting up Y.1731 MEPs 32
   - 3.9 Configuring Speedtest 33
   - 3.10 Configuring SLA (Service Level Agreement) thresholds 37
   - 3.11 Setting up alarms 37
   - 3.12 Changing the report logo 45
   - 3.13 Changing user settings 46
   - 3.14 Creating API tokens 48

4. **Test Agents** 49
   - 4.1 Introduction to Test Agents 49
   - 4.2 Installing Test Agents 64
   - 4.3 Configuring Test Agents from the Netrounds GUI 76
   - 4.4 Configuring Test Agents from the local console 108
   - 4.5 Further technical information on Test Agents 131

5. **Licensing** 138
   - 5.1 Licensing and streams in Netrounds 138

6. **Working with tests and monitors** 144
   - 6.1 Introduction to tests and monitors 144
   - 6.2 Building tests 145
   - 6.3 Options for running tests 149
   - 6.4 The Tests view 151
   - 6.5 View showing an individual test 152
   - 6.6 Building monitors 154
   - 6.7 The Monitoring view 156
   - 6.8 View showing an individual monitor 158
   - 6.9 Reports on tests and monitors 160
   - 6.10 Creating templates 163
   - 6.11 Applying tags 173
   - 6.12 Icons used for tests and monitors 175

© 2020 Netrounds. All rights reserved.
CONTENTS

7 Task types ................................................................. 176
  7.1 Overview of task types ............................................. 176
  7.2 Common test and monitor parameters ............................ 176
  7.3 Listing of task types supporting IPv6 ........................... 177
  7.4 Network performance testing ..................................... 179
  7.5 IPTV and OTT video testing ...................................... 201
  7.6 HTTP and DNS testing ........................................... 218
  7.7 SIP testing .......................................................... 224
  7.8 Wi-Fi network testing ............................................. 228
  7.9 Mobile network testing ........................................... 232
  7.10 Ethernet service activation testing ............................. 234
  7.11 Transparency testing ............................................ 249
  7.12 Reflector-based testing .......................................... 273
  7.13 Security testing .................................................. 312
  7.14 Utilities for testing .............................................. 325

8 Dashboard ................................................................. 326
  8.1 Insight dashboard ................................................... 326

9 Applications ............................................................. 335
  9.1 Remote packet capturing .......................................... 335
  9.2 Speedtest ............................................................ 338
  9.3 Setting up a Test Agent to act as proxy in tests ............... 343

10 Alarms ................................................................. 344
  10.1 Introduction to alarms ............................................ 344
  10.2 Activating alarms for a monitor ................................ 344
  10.3 Alarm dashboard .................................................. 346

11 Collaboration and sharing ............................................ 347
  11.1 Sharing Test Agents .............................................. 347
  11.2 Sharing templates ................................................ 351
  11.3 Sharing test and monitoring results ............................ 354

12 Definitions and technical notes ..................................... 358
  12.1 Abbreviations ..................................................... 358
  12.2 Bridge, bridging ................................................... 361
  12.3 Delay variation (DV), jitter ..................................... 362
  12.4 DSCP/DiffServ and IP Precedence ............................... 362
  12.5 Errored seconds (ES) metric: Method of calculation ........ 363
  12.6 Layer 2 Ethernet frame sizes .................................... 364
  12.7 Mean Opinion Score (MOS) ..................................... 365
  12.8 MPEG basics ....................................................... 365
  12.9 MPEG metrics ..................................................... 365
  12.10 MPEG rate vs. MPEG transport rate ......................... 366
  12.11 Netrounds server ................................................. 368
  12.12 Priority Code Point (PCP) ..................................... 369
  12.13 Using a proxy .................................................... 369
  12.14 Resolution of Netrounds measurement data .................... 370
  12.15 Secure End-user Connections (SEC) .......................... 371
  12.16 Security in Netrounds ........................................... 371
  12.17 Severely Errored Seconds (SES) ............................... 372
  12.18 SLA (Service Level Agreement) ............................... 372
  12.19 SNMP .............................................................. 373
  12.20 TCP implementation in Netrounds .............................. 373

© 2020 Netrounds. All rights reserved.
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.21</td>
<td>Theoretical maximum throughput with Netrounds Test Agents</td>
<td>374</td>
</tr>
<tr>
<td>12.22</td>
<td>Unavailable Seconds (UAS)</td>
<td>377</td>
</tr>
<tr>
<td>12.23</td>
<td>VLAN</td>
<td>377</td>
</tr>
<tr>
<td>13.1</td>
<td>Release notes, Netrounds software version 2.34.1</td>
<td>378</td>
</tr>
<tr>
<td>13.2</td>
<td>Release notes, Netrounds software version 2.34.0</td>
<td>378</td>
</tr>
<tr>
<td>13.3</td>
<td>Release notes, Netrounds software version 2.33.0</td>
<td>379</td>
</tr>
<tr>
<td>13.4</td>
<td>Release notes, Netrounds software version 2.32.0</td>
<td>380</td>
</tr>
<tr>
<td>13.5</td>
<td>Release notes, Netrounds software version 2.31.0</td>
<td>381</td>
</tr>
<tr>
<td>13.6</td>
<td>Release notes, Netrounds software version 2.30.0</td>
<td>382</td>
</tr>
<tr>
<td>13.7</td>
<td>Release notes, Netrounds software version 2.29.0</td>
<td>383</td>
</tr>
<tr>
<td>13.8</td>
<td>Release notes, Netrounds software version 2.28.0</td>
<td>384</td>
</tr>
<tr>
<td>13.9</td>
<td>Release notes, Netrounds software version 2.27.0</td>
<td>386</td>
</tr>
<tr>
<td>13.10</td>
<td>Release notes, Netrounds software version 2.26.0</td>
<td>387</td>
</tr>
<tr>
<td>13.11</td>
<td>Release notes, Netrounds software version 2.25.0</td>
<td>388</td>
</tr>
<tr>
<td>13.12</td>
<td>Release notes, Netrounds software version 2.24.0</td>
<td>392</td>
</tr>
<tr>
<td>13.13</td>
<td>Release notes, Netrounds software version 2.23.2</td>
<td>393</td>
</tr>
<tr>
<td>13.14</td>
<td>Release notes, Netrounds software version 2.23.1</td>
<td>394</td>
</tr>
<tr>
<td>13.15</td>
<td>Release notes, Netrounds software version 2.23.0</td>
<td>394</td>
</tr>
<tr>
<td>13.16</td>
<td>Release notes, Netrounds software version 2.22.3</td>
<td>395</td>
</tr>
<tr>
<td>13.17</td>
<td>Release notes, Netrounds software version 2.22.2</td>
<td>396</td>
</tr>
<tr>
<td>13.18</td>
<td>Release notes, Netrounds software version 2.22.1</td>
<td>396</td>
</tr>
<tr>
<td>13.19</td>
<td>Release notes, Netrounds software version 2.22.0</td>
<td>398</td>
</tr>
<tr>
<td>13.20</td>
<td>Release notes, Netrounds software version 2.21.0</td>
<td>398</td>
</tr>
<tr>
<td>13.21</td>
<td>Release notes, Netrounds software version 2.20.0</td>
<td>399</td>
</tr>
<tr>
<td>13.22</td>
<td>Release notes, Netrounds software version 2.19.0</td>
<td>401</td>
</tr>
<tr>
<td>13.23</td>
<td>Release notes, Netrounds software version 2.18.0</td>
<td>401</td>
</tr>
<tr>
<td>13.24</td>
<td>Release notes, Netrounds software version 2.17.0</td>
<td>403</td>
</tr>
</tbody>
</table>
1 An overview of Netrounds

Netrounds is a programmable, active test and service assurance platform for physical, hybrid and virtual networks. Netrounds’ automation capabilities 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 Netrounds 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. Netrounds Control Center is offered as a SaaS solution at https://app.netrounds.com, but it can also be deployed on-premise. Netrounds 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.

Netrounds Control Center remotely controls Netrounds’ 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, mobile radio, 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 Netrounds Control Center.

The present documentation is the main support documentation for Netrounds. Further product documentation is available at https://portal.netrounds.com. Here you can also file support tickets and requests for new features in Netrounds. If you have an on-premise installation of Netrounds Control Center, you can download Netrounds Control Center and various other Netrounds software. To gain access to the Netrounds Portal, click the “request login” link at the URL just mentioned.
2 Getting started

2.1 Easy steps for sign-up to Netrounds

To create a Netrounds free trial account (valid for 30 days), go to www.netrounds.com/see-try, click the button Request free trial, and fill in the required information.

On completing the sign-up, you should see the Netrounds Dashboard:

From here you can go right ahead and explore the measurement toolbox of Netrounds.

The top section of the Dashboard has the following quick-link buttons:

- Start first monitoring: Quick start and getting started wizards (page 7)
- Explore: Explore this documentation (page 5)
- Collaborate: Collaborate and invite colleagues (page 20)

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

To start monitoring and troubleshooting with all the available features in Netrounds, using your own Netrounds active traffic generating Test Agents, you can download Test Agent software and install them on your own x86-based hardware. Read more here (page 64).

For guidance on preparations for specific measurements supported by Netrounds, see the Getting started pages (page 6) in this section.
2.2 Netrounds getting started wizards

2.2.1 “Start now” section

Here you can get a feel for how Netrounds works without having to create regular monitors. To try out a feature under this heading, click its Start button, fill in the hosts/sites, and select a Netrounds Test Agent interface to use.

2.2.2 “Install and start” section

Below are links to getting started guides for each of the features in this section. The Get started buttons provided in the user interface likewise take you to these guides.

- **IPTV and Digital CATV** (page 13)
- **Network performance** (page 15)
- **Speedtest** (page 17)
- **IP telephony, VoIP, and SIP** (page 18)
- **Traffic analysis** (page 15)
- **HTTP and DNS** (page 12)

2.3 Guided tour of the main menu

2.3.1 Dashboard

The Dashboard gives an overview of Netrounds and is a starting point for setting up new tests and monitors. Under the headings Monitors and Tests, the most recent measurement sessions are displayed.

- For monitors, an **SLA** (page 372) indicator and an **Errored Seconds (ES)** (page 363) bar give direct visual feedback on measured quality. To view log messages for the monitor, hover over the “i” button. (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 159).
- For 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 defined in Netrounds.

How to set up alarms is explained on this page (page 37).

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 145) 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 154) for full instructions on how to build monitors.

2.3.5 Apps menu

From this tab you access the Speedtest (page 338) and Remote packet capture (page 335) features.

### Applications

**Speedtest**

Use browser-based speed tests to simplify customer support.

**Remote Packet Capture**

Troubleshoot app problems using remote packet capture and analysis.

2.3.6 Test Agents

2.3.6.1 Interface info tab

Under My Test Agents, you will find all your registered Test Agents.
Under Shared with me, any shared (page 347) 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>License</th>
<th>No. of streams</th>
<th>Used streams</th>
<th>Available streams</th>
<th>Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws-eu-north-1-1</td>
<td>SW.Test Agent Large</td>
<td>500</td>
<td>38</td>
<td>462</td>
<td></td>
</tr>
<tr>
<td>aws-eu-north-1-2</td>
<td>SW.Test Agent Large</td>
<td>500</td>
<td>8</td>
<td>492</td>
<td></td>
</tr>
</tbody>
</table>

At the bottom, under the heading Licenses, are displayed the types of license connected to your account. Please refer to [this page](page 138) (page 138) for more detailed information.
2.3.7 Account

Read more [here](#) (page 19) about the various items under Account & Settings.
2.3.8 Docs
Clicking this button takes you to the contents page of the present documentation.

2.3.9 Quick start
This button links to the Quick start wizards (page 7).

2.3.10 Top bar: “Bell” button

This button will light up if a notification is received; for example, if an alarm (page 346) is triggered or if someone shares a Test Agent (page 347) to your account.

2.3.11 Top bar: User name
Click your Netrounds user name on the top bar in order to:
• edit your Netrounds user profile
• switch to a different Netrounds account (if you have access to several accounts)
• log out from Netrounds.

demo@netrounds.com

Edit profile
Log out

2.4 Getting started with HTTP and DNS measurements
To get started with HTTP and DNS measurements in Netrounds, please follow these simple steps.

2.4.1 Sign up for Netrounds
Sign up for a trial account completely free of charge as follows:
• Navigate to www.netrounds.com/see-try.
• Click the button Request free trial.
• Fill in the required information.

The trial account gives you access to five software Test Agents (with Medium license) which among other things are capable of HTTP and DNS requests.
2.4.2 Download a Netrounds Test Agent

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

- Under Test Agents in the Netrounds user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for software Test Agents given [here](page 64).

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

- In the cloud server case, to login.netrounds.com using TCP port 443.
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.4.3 Start your first test

Follow the instructions below to start a simple HTTP test.

1. Log in to your account in the Netrounds 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 144).

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

- [HTTP](page 219)
- [DNS](page 222)

2.5 Getting started with IPTV measurements

To get started with IPTV measurements, please follow these simple steps.

2.5.1 Sign up for Netrounds

Sign up for a trial account completely free of charge as follows:

- Navigate to [www.netrounds.com/see-try](www.netrounds.com/see-try).
- Click the button Request free trial.
- Fill in the required information.

The trial account gives you access to five software Test Agents (with Medium license), which among other things are capable of IPTV MPEG measurements.
2.5.2 Download a Netrounds Test Agent

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

- Under Test Agents in the Netrounds user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for software Test Agents given here (page 64).

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

- In the cloud server case, to https://login.netrounds.com using TCP port 443. (For Test Agents Lite, the URL is agent.netrounds.com, while the port is the same.)
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

However, as an alternative setup, one Netrounds 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 Netrounds cloud servers.

Please read here (page 343) for more information about setting up Netrounds using a proxy.

2.5.3 Add IPTV channels

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

2.5.4 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 Netrounds 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 144).

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

- IPTV MPEG (page 202)
- IPTV MPEG inline (page 205)
- IPTV channel zapping time (page 207)
- OTT testing: HTTP Live Streaming (HLS) (page 212)
• **DVB-C MPEG** (page 210)

### 2.5.5 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.

### 2.6 Getting started with remote packet capture and traffic analysis

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

#### 2.6.1 Sign up for Netrounds

Sign up for a trial account completely free of charge as follows:

- Navigate to [www.netrounds.com/see-try](http://www.netrounds.com/see-try).
- Click the button Request free trial.
- Fill in the required information.

The trial account gives you access to five software Test Agents (with Medium license), which among other things can be activated as remote traffic capture devices.

#### 2.6.2 Download a Netrounds Test Agent

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

- Under Test Agents in the Netrounds user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for software Test Agents given [here](page 64).

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

- In the cloud server case, to [https://login.netrounds.com](https://login.netrounds.com) using TCP port 443.
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

#### 2.6.3 Perform your first remote packet capture

Once the Test Agent has connected to your Netrounds account, it will show up in the Test Agents view. Read more [here](page 335) on how to perform your first remote packet capture.

### 2.7 Getting started with network performance measurements

To get started with network performance measurements, please follow these simple steps.
2.7.1 Sign up for Netrounds

Sign up for a trial account completely free of charge as follows:

- Navigate to www.netrounds.com/see-try.
- Click the button Request free trial.
- Fill in the required information.

The trial account gives you access to five software Test Agents (with Medium license), which among other things are capable of TCP and UDP network performance measurements.

2.7.2 Download Netrounds Test Agents

In this step we will download, install, and configure Netrounds 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 Netrounds user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for software Test Agents given here (page 64).

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

- In the cloud server case, to https://login.netrounds.com using TCP port 443. (For Test Agents Lite, the URL is agent.netrounds.com, while the port is the same.)
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.7.3 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 Netrounds 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 Netrounds 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 144).

A full treatment of network performance testing is found on the following pages:
• TCP (page 180)
• Multisession TCP (page 182)
• UDP (page 184)
• VoIP UDP (page 187)
• Multicast UDP (page 190)
• TCP throughput test according to RFC 6349 (page 193)
• QoS policy profiling (page 197)

2.8 Getting started with your own Speedtest server

To get started with your own Speedtest server, please follow these simple steps.

2.8.1 Sign up for Netrounds

Sign up for a trial account completely free of charge as follows:

• Navigate to www.netrounds.com/see-try.
• Click the button Request free trial.
• Fill in the required information.

The trial account gives you access to five software Test Agents (with Medium license) which among other things can be activated as Speedtest responders for browser-based tests from your end-users.

2.8.2 Download a Netrounds Test Agent

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

• Under Test Agents in the Netrounds user interface, click the Download button.
• Download the desired type of Test Agent.
• Follow the installation instructions for software Test Agents given here (page 64).

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

• In the cloud server case, to https://login.netrounds.com using TCP port 443.
• In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).

2.8.3 Activate the Speedtest responder

Once the Test Agent has connected to your Netrounds 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 33).
2.8.4 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 Netrounds 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 Netrounds Test Agent.
6. The result is stored in your Netrounds account. To see the result, go to Apps > Speedtest.

Full details are found here (page 338).

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

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

2.9.1 Sign up for Netrounds

Sign up for a trial account completely free of charge as follows:

- Navigate to www.netrounds.com/see-try.
- Click the button Request free trial.
- Fill in the required information.

The trial account gives you access to five software Test Agents (with Medium license), which among other things are capable of SIP-based measurements.

2.9.2 Download Netrounds Test Agents

In this step we will download, install, and configure Netrounds 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 Netrounds user interface, click the Download button.
- Download the desired type of Test Agent.
- Follow the installation instructions for software Test Agents given here (page 64).

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

- In the cloud server case, to https://login.netrounds.com using TCP port 443.
- In the on-premise server case, to the host IP using TCP port 6000 (default, configurable).
2.9.3 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 24).

2.9.4 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 Netrounds 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 Netrounds 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 144).

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

• SIP (page 225)

3 Setting up your account

3.1 Managing your account

3.1.1 How to change or reset your Netrounds password

To change or reset your password, log in to Netrounds, do one of the following:

• Access your personal profile by clicking your Netrounds 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 a LDAP or TACACS+ server, its password too is managed on that server and cannot be changed here. See the Installation Guide for further details on configuring LDAP and TACACS+ user management in Netrounds Control Center.
3.1.2 How to update the contact information for your Netrounds account

Your contact information is part of your Netrounds profile.
To update your profile, click your Netrounds user name on the top bar, then choose Edit profile and make the desired changes.

3.2 Administering users and permissions

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

**Note:** To change permissions for a user, you first delete the user, then add the user again with the new set of permissions.

<table>
<thead>
<tr>
<th>Email</th>
<th>Name</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:demo@cisco.com">demo@cisco.com</a></td>
<td>admin</td>
<td></td>
</tr>
<tr>
<td>[redacted]@cisco.com</td>
<td>admin</td>
<td></td>
</tr>
</tbody>
</table>

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

The permission levels are *Admin*, *Write*, and *Read*. Over and above this, there is a *Register only* role which permits registration of Test Agents and nothing else.

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

3.2.1 User permissions required for Netrounds 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 Netrounds (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*

- **DVB-C channels**: *Write* for everything except DVB-C channel discovery, which requires *Admin*
• **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

### 3.3 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 in |nr-product| 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. |nr-product| 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.

### 3.3.1 Uploading your channel plan as a CSV file

Alternatively, instead of entering channels manually in the user interface, you can upload a channel plan as a comma-separated file (*.csv).

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

```
channel_name,multicast_ip,udp_port[,source_ip[,program_number]]
```

Channel names cannot contain commas (,).

Please make sure that the file is encoded in UTF-8 format if it contains international characters. 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.

### 3.4 Setting up DVB-C channels

Configuration of DVB-C channels is done under Account > DVB-C channels. Here you define the DVB-C channels available in your network. All channels in the resulting channel can then be selected in the user interface when you set up a DVB-C task.
Click the Add button to add a new DVB-C channel. The following parameters need to be defined for each channel:

- **Name**: DVB-C channel name. This is the name shown when listing channels.
- **Frequency**: The frequency of the channel in Hz.
- **Symbol rate**: The symbol rate of the channel, stated as symbols per second.
- **FEC**: The Forward Error Correction for the channel. Valid settings are: 1/2, 2/3, 3/4, 5/6, 7/8, and NONE.
- **Modulation**: The modulation used for the channel. Valid settings are: AUTO, 16-QAM, 32-QAM, 64-QAM, 128-QAM, and 256-QAM.
- **Program number**: The program number of the channel.

### 3.4.1 Discovering DVB-C channels

Netrounds has a built-in channel discovery function for DVB-C channels.

#### DVB-C Channels discovery

- **Test Agent**:adapter  
  ![DVBC:U130517154306](image)
- **Frequency**
- **Symbol rate**
- **FEC**:  
  ![AUTO](image)
- **Modulation**:  
  ![16-QAM](image)
- **Discover**

- Change “Test Agent:adapter” to the Test Agent that is connected to the DVB-C distribution network.
- Indicate the frequency, symbol rate, FEC, and modulation used by the TV distributor.
- Click Discover. The discovery scan will only cover the channels in the network information table in the DVB-C stream with the frequency given. To get all available channels, you may have to perform this procedure repeatedly.

**Note**: The discovered channel plan will populate the channel list automatically and remove any channels previously listed.

### 3.4.2 Uploading your DVB-C channel plan as a CSV file

Alternatively, instead of entering channels manually in the user interface, you can upload a channel plan as a comma-separated file (*.csv).

This file must have one channel per line, defined according to the following syntax:
Channel names cannot contain commas (,).
Please make sure that the file is encoded in UTF-8 format if it contains international characters. 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 DVB-C channels to be deleted.

### 3.5 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 224).

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.

Netrounds’ 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:

  ![SIP Accounts](image)

  - 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. 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.

### 3.5.1 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 (*.csv).
This file must have one SIP account per line, defined according to the following syntax:

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

Please make sure that the file is encoded in UTF-8 format if it contains international characters. 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 Setting up Ping hosts

Ping testing requires at least one Ping host, towards which the Netrounds Test Agent (acting as initiator) sends the test stream. Ping hosts usually reside in third-party equipment; however, Test Agents also have Ping host functionality and can be added to the Netrounds 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 293).

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

This dialog appears:

```
Add object

Name: 

Host: 

(IPV6 checkbox)

(GPS Latitude): 

(GPS Longitude): 

[Save] [Cancel]
```

- **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 check 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 103).
Finish by clicking the Save button.

### 3.6.1 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 173).

In addition, when importing hosts from a CSV file (see *below* (page 27)), you can apply tags to hosts directly in that file, without any user interface actions.

### 3.6.2 Ping host view

The view shows all Ping hosts known to Netrounds. 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.

![PING Hosts](image)

### 3.6.3 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 them as a comma-separated file (*.csv).

- Click the Import button to start the import procedure.

An import dialog appears which also summarizes the requirements on the CSV file syntax. See *below* (page 28).

- 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 recom- mended 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.
3.6.3.1 CSV file syntax requirements

CSV files are parsed according to IETF RFC 4180.

- The first line in the file is parsed as a header describing the fields **name**, **host**, **tags**, **gps_lat**, and **gps_long**.
- In the remainder of the file, one host is expected on each line.
- The fields **tags**, **gps_lat**, and **gps_long** are optional.
- String values must be enclosed within double quotes: "...".
- If a string contains a double quote, it must be escaped with another, like this: ".".
- If several tags are given in the **tags** string, they must be separated by commas.
- Latitude and longitude are given in decimal degrees:
  - Latitude is a value between −85.06 and +85.06, where a minus sign means “south”.
  - Longitude is a value between −180 and +180, where a minus sign means “west”.

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

```
"name","host","tags","gps_lat","gps_long"
"Host","123.122.121.111","tag,tag with spaces",65.58,22.15
```

3.6.4 Exporting Ping hosts to a CSV file

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

- Click the Export button.
- In the dialog that opens, click Export.

The exported file will adhere to the requirements stated above for imported CSV files, and it will be encoded in UTF-8.

3.6.5 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 26) a Ping host.

3.6.6 Deleting Ping hosts

To delete one or several Ping hosts, do as follows:

- Check 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:

![Delete all button]

**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.

### 3.7 Setting up TWAMP reflectors

TWAMP testing requires at least one TWAMP reflector, towards which the Netrounds Test Agent (acting as initiator) sends the test stream. TWAMP reflectors usually reside in third-party equipment; however, Test Agents also have TWAMP reflector functionality and can be added to the Netrounds 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](#).

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

This dialog appears:

![Add object dialog]

- **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 check the IPv6 checkbox.
- **Test port:** UDP port number for the TWAMP reflector.
- **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.
- **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”.

© 2020 Netrounds. All rights reserved. 29
• 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 103).

Finish by clicking the Save button.

### 3.7.1 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 173).

In addition, when importing reflectors from a CSV file (see *below* (page 30)), you can apply tags to reflectors directly in that file, without any user interface actions.

### 3.7.2 TWAMP reflectors view

The view shows all TWAMP reflectors known to Netrounds. 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.

<table>
<thead>
<tr>
<th>TWAMP Reflectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Reflector 1</td>
</tr>
<tr>
<td>Reflector 2</td>
</tr>
<tr>
<td>Reflector 3</td>
</tr>
<tr>
<td>test</td>
</tr>
<tr>
<td>test2</td>
</tr>
</tbody>
</table>

### 3.7.3 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 them as a comma-separated file (*.csv).

• Click the Import button to start the import procedure.

An import dialog appears which also summarizes the requirements on the CSV file syntax. See *below* (page 31).

• 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.

3.7.3.1 CSV file syntax requirements

CSV files are parsed according to IETF RFC 4180.

• The first line in the file is parsed as a header describing the fields name, host, port, ctrl_port, tags, gps_lat, and gps_long.
• In the remainder of the file, one reflector is expected on each line.
• The fields tags, ctrl_port, gps_lat, and gps_long are optional. If ctrl_port is left empty, TWAMP Light will be used (the control protocol is disabled).
• String values must be enclosed within double quotes: "...".
• If a string contains a double quote, it must be escaped with another, like this: ".".
• If several tags are given in the tags string, they must be separated by commas.
• Latitude and longitude are given in decimal degrees:
  – Latitude is a value between –85.06 and +85.06, where a minus sign means “south”.
  – Longitude is a value between –180 and +180, where a minus sign means “west”.

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

```
"name","host","port","ctrl_port","tags","gps_lat","gps_long"
"Reflector","123.122.121.111",10,7000,"tag,tag with spaces",65.58,22.15
```

3.7.4 Exporting TWAMP reflectors to a CSV file

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

• Click the Export button.
• In the dialog that opens, click Export.

The exported file will adhere to the requirements stated above for imported CSV files, and it will be encoded in UTF-8.

3.7.5 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 29) a TWAMP reflector.

3.7.6 Deleting TWAMP reflectors

To delete one or several TWAMP reflectors, do as follows:

• Check the box on the far left for each reflector you wish to delete.
• Click the trash can button that appears at the top of the view.
3.8 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 send streams towards from the Netrounds Test Agent.

The MEPs defined here are used as test points in Y.1731 testing (page 273).

- 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.

3.8.1 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 (*.csv).

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

```
name, mac, meg_level
```

MEP names cannot contain commas (,).

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

**Note:** Uploading a Y.1731 MEP list will overwrite all existing Y.1731 MEPs. 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.

3.8.2 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 32) a Y.1731 MEP.

3.8.3 Deleting Y.1731 MEPs

To delete one or several Y.1731 MEPs, do as follows:

• Check 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.

3.9 Configuring Speedtest

For general information about Netrounds' Speedtest function and its presentation, see this page (page 338). Configuration of Speedtest is done under Account > Speedtest. The following settings are available:
3.9.1 General tab

- Default technology: Type of Speedtest technology to use by default: Websocket or Flash. You can accept or override this setting when starting a test.

- 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 35) below).

Note: The ports for WebSocket and Flash cannot be the same. Furthermore, be aware that web browsers normally block connections to ports commonly used by other protocols, so 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.

- 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 left unchecked, 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.


Confirm your settings by clicking the Save config button.

3.9.2 WebSocket tab

- WebSocket port: TCP destination port used by the web browser towards the Speedtest Test Agent when the test is done using WebSocket. Default: 80.

3.9.3 Flash tab

- Disable Flash Speedtest: Check to disable Flash-based Speedtest.

- Flash port: TCP destination port used by the web browser towards the Speedtest Test Agent when the test is done using Flash. Default: 5050.

- Test duration (s): Duration of the downstream and upstream tests in seconds. Default: 10 s in each direction. Max: For Flash, 3600 s. For WebSocket, 10 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.
• TCP sessions: The numbers of parallel TCP sessions used to measure download and upload rates. Default: 8 sessions. Max: For WebSocket, 8 sessions is the upper limit.

3.9.4 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 Netrounds Speedtest app and on the public Speedtest page:

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

3.9.5 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.

3.9.6 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.

3.9.7 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 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:

3.9.8 Hiding Test Agent interfaces used in Speedtest

It may happen that you don’t want to display all Test Agent interfaces that are busy with Speedtest on the public Speedtest page. To hide an interface, go to the Test Agent settings, click the interface, and add the text “[bbq:hidden]” in the Description field for that interface. See the example below, where the interface “Birch:eth1” will not show up under Server on the public Speedtest page:
3.10 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 Netrounds to quickly display if 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.

| Good SLA (%) | 99.95 |
| Acceptable SLA (%) | 99.5 |

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 \((100 – \text{ES}) \geq 99.95\%\), 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 \((100 – \text{ES}) \geq 99.5\%\), 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 \((100 – \text{ES}) < 99.5\%\), i.e. \(\text{ES} > 0.5\%)\.

Read more about SLAs [here](page 372).

3.11 Setting up alarms

Alarm and SNMP (page 373) configuration is done under Account > Alarms. Specifically, the items configured are:

- SMNP managers
- Email lists
- Alarm templates
3 SETTING UP YOUR ACCOUNT Release 2.34.1.final.0

• Automatic alarm suppressions

3.11.1 SNMP Manager settings

Netrounds can send SNMP traps to an SNMP manager whenever the SLA is violated during a monitoring session. Netrounds 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 Netrounds Control Center is installed, 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 Netrounds alarm MIB can be downloaded from https://portal.netrounds.com.

When setting up a monitor (page 154), 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, choose 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:
3 SETTING UP YOUR ACCOUNT

Add/Update SNMP Manager

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td></td>
</tr>
<tr>
<td>Version:</td>
<td>SNMP v2c, SNMP v3</td>
</tr>
<tr>
<td>IP address:</td>
<td></td>
</tr>
<tr>
<td>Port:</td>
<td></td>
</tr>
<tr>
<td>Engine ID:</td>
<td>080005f8505000001</td>
</tr>
<tr>
<td>User name:</td>
<td></td>
</tr>
<tr>
<td>Security:</td>
<td>No authentication, no privacy</td>
</tr>
</tbody>
</table>

- 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 Netrounds 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 choose 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 choose 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.
3.11.2 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.
3.11.2.1 Sending alarms by text (SMS)

You can also have alarms sent to you by text (SMS messages). This functionality is purchased as a separate feature.

If this feature is activated, you can enter an SMS recipient in the Emails field in the following format:

```
<phone number>@sms.netrounds.com
```

Alarms will then be forwarded to the specified phone number. Your account permissions may impose a limit on the number of alarm SMS messages that can be sent each month.

3.11.3 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 38).

• 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 42).

Note: You can choose either an SNMP manager, or an email list, or one of each.
3.11.3.1 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. Check the Send only once box 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.

3.11.3.2 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; check the box next to the heading to enable it.

- Trigger alarm on “no data received”: Enables triggering an alarm if no data is received. This must be enabled for the below settings 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.
3.11.3.3 Alarm template name

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

3.11.3.4 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 344).

3.11.4 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.

![New Alarm Suppression](image)

- 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.

3.12 Changing the report logo

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

3.13 Changing user settings

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

- Click your user name in the top right corner of the screen, and select Edit profile.
3 Setting Up Your Account

User settings

API tokens

- 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.

- Time format: Select date and time formats as desired.

Click the Save button to save your settings.

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

Note: If your user is managed by a LDAP or TACACS+ server, its password too is managed on that server and cannot be changed here. See the Installation Guide for further details on configuring LDAP and
3.14 Creating API tokens

If you are going to perform orchestration of Netrounds 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 button Add API Token.

• A new token is generated and added to the My API tokens list (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 Account API tokens, 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 Netrounds, 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, Test Agent Lite, 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 can be packaged and delivered by Netrounds in a number of ways. The following terms are used:
  - **Preinstalled Test Agent**: Hardware delivered from Netrounds with preinstalled Test Agent Appliance software.
  - **Software Test Agent**: Test Agent Appliance software downloaded by the customer and installed on customer-provided x86 hardware.
  - **Virtual Test Agent**: Test Agent Appliance software downloaded by the customer and run as a virtual machine (VM) on a hypervisor.

- A **Test Agent Application** is a Test Agent with limited functionality, as detailed below (page 51). It consists of software downloaded by the customer and installed as an application on a Linux computer.

- A **Test Agent Lite** is another Test Agent with limited functionality; again, see below (page 51). It consists of software downloaded by the customer and installed in one of the following ways:
– **PC Test Agent**: As an application on a Linux computer.

– **Embedded Test Agent**: As an application on top of OpenWrt, installed on a SoC device.

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 or Test Agents Lite), “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 Agents Lite were previously referred to as “Agents”, while Test Agent Appliances were called “Probes”.

Over and above this, we use the term **browser Test Agent** to denote Test Agent software running in a web browser during the execution of certain network performance tests.

### 4.1.1.1 Basics of Test Agent Appliances

**Preinstalled Test Agents**

A preinstalled Test Agent is simply plugged physically into your network and managed from Netrounds Control Center.

How to get started with a preinstalled Test Agent is covered on [this page](#) (page 64).

**Software Test Agents**

There are two options for installing a software Test Agent: permanently on a PC hard disk, or on a USB memory for temporary use of the PC.

Read more about the two options [here](#) (page 66).

**Virtual Test Agents**

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 Netrounds Control Center.

### 4.1.1.2 Basics of Test Agent Applications

Test Agent Applications are designed for installation on top of a Linux system. It runs 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.

Test Agent Applications have a limited feature set compared to Test Agent Appliances, as laid out [here](#) (page 51). Test Agent Applications also have performance limitations which are due to their lack of control of the lower protocol layers.
4.1.1.3 Basics of Test Agents Lite

Test Agent Lite is deprecated and is no longer offered for sale. It is replaced by Test Agent Application. However, existing Test Agents Lite are still supported in Netrounds systems.

Test Agents Lite have a limited feature set similar to that of Test Agent Appliances.

PC Test Agents

A PC Test Agent is a Test Agent Lite application that runs on top of a Linux computer. PC Test Agents are compatible with and can exchange TCP/UDP traffic with other Test Agents. The performance of a PC Test Agent is constrained by the computing power and the operating system of its host PC.

Embedded Test Agents

An embedded Test Agent is a Test Agent Lite application run on top of OpenWrt and typically installed on a SoC (system-on-chip) device.

4.1.1.4 Basics of browser Test Agents

A browser Test Agent runs in a web browser whenever a network performance test such as Speedtest (page 338) is executed. You do not need to download this software, and it is compatible with any web browser.

4.1.1.5 Capabilities of Test Agents

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

In the case of Test Agents Lite, limited feature sets are supported, as explained in footnotes to the table. Some features require additional hardware; again, this is indicated in footnotes.

The measurement features themselves are covered here (page 335) (Speedtest, packet capture) and here (page 176) (all other features in the table).
<table>
<thead>
<tr>
<th>Feature</th>
<th>TA Appliance</th>
<th>TA Lite</th>
<th>TA Application</th>
<th>No. of TA interfaces needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP, UDP</td>
<td>Yes</td>
<td>Yes(^1)</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>IPTV MPEG</td>
<td>Yes</td>
<td>Yes(^2)</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>DVB-C MPEG</td>
<td>Yes(^3)</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>OTT video (HLS)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>HTTP, DNS, Ping</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>SIP</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Mobile network logging</td>
<td>Yes(^4)</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Wi-Fi network logging</td>
<td>Yes(^5)</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Security tests</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1 or 2 depending on test</td>
</tr>
<tr>
<td>Ethernet service activation</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>2</td>
</tr>
<tr>
<td>Transparency tests</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>2</td>
</tr>
<tr>
<td>Y.1731, TWAMP</td>
<td>Yes</td>
<td>No</td>
<td>TWAMP</td>
<td>1</td>
</tr>
<tr>
<td>Speedtest(^6)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Packet capture</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
</tbody>
</table>

### 4.1.1.6 Configuring Test Agents

There are two ways to configure Test Agent Appliances in Netrounds:

- From the Netrounds GUI. This is covered [here](#) (page 76).
- From the Test Agent local console, accessed from the command line. The local console is described [here](#) (page 108).

These interfaces are partly overlapping with respect to their functionality. *Registration* (page 109) of Test Agents, however, must be done from the local console.

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

### 4.1.2 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 347) to your account from other accounts will appear.

---

1. Stateful TCP throughput; UDP packet delay variation, UDP packet loss.
2. Multicast-based IPTV MPEG.
3. Requires DVB-C dongle; details [here](#) (page 210).
4. Requires mobile network modem; details [here](#) (page 232).
5. Requires Wi-Fi network card; details [here](#) (page 230)
6. TCP throughput testing initiated from a web browser.
The Download button at the top of the view is used to download Test Agent software. See these pages (page 64) 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 55).

**4.1.2.1 Columns in the Test Agents view**

**Name column**

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Icon used for Test Agent Appliance" /></td>
<td>Icon used for Test Agent Appliance</td>
</tr>
<tr>
<td><img src="image" alt="Icon used for Test Agent Application" /></td>
<td>Icon used for Test Agent Application</td>
</tr>
<tr>
<td><img src="image" alt="Icon used for generic Test Agent" /></td>
<td>Icon used for generic Test Agent (see below)</td>
</tr>
<tr>
<td><img src="image" alt="Icon used for Test Agent Lite" /></td>
<td>Icon used for Test Agent Lite</td>
</tr>
</tbody>
</table>

**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 57).

**Description column**

Plain-text description of the Test Agent entered in the heading of the Test Agent properties view: see this page (page 57).

**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 Netrounds 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 97).

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 347).

Test Agents Lite cannot be shared, so for them this column is always empty.
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.

### 4.1.2.2 Actions applicable to Test Agents

You can select one or several Test Agents by checking the boxes on the far left of 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 61) 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.

### 4.1.2.3 Filtering the Test Agents view

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

- The Clear Filter button clears all filter settings.
- Tags column allows filtering by tag, with options for All, Online, Offline, In use, and Free.

Filtering of Test Agents by status (online/offline) has already been covered here (page 54).

You can also filter Test Agents by name, by typing characters into the “Search” text box, or by tag, by checking the Tags box and then typing or selecting tag names as explained here (page 62).

### 4.1.2.4 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>License</th>
<th>No. of streams</th>
<th>Used streams</th>
<th>Available streams</th>
<th>Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws-eu-north-1-1</td>
<td>SW-Test Agent Large</td>
<td>500</td>
<td>38</td>
<td>462</td>
<td></td>
</tr>
<tr>
<td>aws-eu-north-1-2</td>
<td>SW-Test Agent Large</td>
<td>500</td>
<td>48</td>
<td>492</td>
<td></td>
</tr>
</tbody>
</table>

Read more about licensing and streams on this page (page 138).

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.

<table>
<thead>
<tr>
<th>Test Agent Licenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
</tr>
<tr>
<td>SW-Test Agent Mini</td>
</tr>
<tr>
<td>SW-Test Agent Small</td>
</tr>
<tr>
<td>SW-Test Agent Medium</td>
</tr>
<tr>
<td>SW-Test Agent Large</td>
</tr>
<tr>
<td>SW-Test Agent 8800</td>
</tr>
<tr>
<td>SW-Agent Mini</td>
</tr>
<tr>
<td>SW-Agent Small</td>
</tr>
<tr>
<td>SW-Agent Medium</td>
</tr>
<tr>
<td>Unlimited</td>
</tr>
<tr>
<td>IPTV</td>
</tr>
<tr>
<td>DVB-C</td>
</tr>
<tr>
<td>TWAMP</td>
</tr>
</tbody>
</table>

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.
Again, please refer to the page on *licensing and streams* (page 138) for more detailed information, especially on the counting of streams for various task types.

### 4.1.3 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.

<table>
<thead>
<tr>
<th>Tab</th>
<th>TA Appliance</th>
<th>TA Lite</th>
<th>TA Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Details</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Interfaces</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DVB-C</td>
<td>Yes*</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Applications</td>
<td>Yes*</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NTP</td>
<td>Yes*</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Streams</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>License</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Utils/Unregister</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GPS Location</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Platform Information</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

#### 4.1.3.1 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.

*aws-eu-north-1-1*

[Click here to add a description]

#### 4.1.3.2 Details tab

This tab shows some basic properties of a Test Agent Lite:

- Operating system of the device where the Test Agent is installed
- Test Agent status (Online/Offline)
• Test Agent Lite software version
• Public IP address of Test Agent

4.1.3.3 Interfaces tab

See this page (page 77), which covers everything about Test Agent interface configuration.

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Applications</th>
<th>NTP</th>
<th>Streams</th>
<th>License</th>
<th>Util</th>
<th>GPS Location</th>
<th>Platform Information</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>IPv4 address</th>
<th>IPv6 address</th>
<th>MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0 (Management)</td>
<td></td>
<td>172.31.44.61/20 (dhcp)</td>
<td>2a05:0d10:67a:9002:a863:9005:dd8b:410128 (dhcp)</td>
<td>0a:23:74:1e:33:56</td>
</tr>
</tbody>
</table>

4.1.3.4 DVB-C tab

This tab appears only for Test Agents equipped with a DVB-C USB device (and thus capable of DVB-C MPEG (page 210) measurements). The tab lists the DVB-C adapters of that USB device.

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>DVB-C</th>
<th>Streams</th>
<th>License</th>
<th>Util</th>
<th>GPS Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>U130517154306</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U160214000007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.3.5 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 Netrounds. For further information, see this page (page 97).
4.1.3.6 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 98).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Start/Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedtest</td>
<td>Speedtest</td>
<td>Enable</td>
</tr>
<tr>
<td>Proxy</td>
<td>Proxy for management traffic</td>
<td>Enable</td>
</tr>
<tr>
<td>Live remote packet capture</td>
<td>Live capture from Test Agent</td>
<td>Enable</td>
</tr>
</tbody>
</table>

| Time offset: -0.000 ms Sync NTP            |
| Interface: (Same as management)            |
| Server: 169.254.169.123                    |
| Enable IPv6:                               |

4.1.3.7 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 138) about the number of streams consumed by testing of various services and by applications.
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.

Utils tab

This tab holds some useful utilities for troubleshooting and management. See this page (page 101) for full coverage.
4.1.3.10 GPS Location Tab

This tab holds geographical coordinates for the Test Agent. See this page (page 103) for more information.

4.1.3.11 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 104) for more information.

4.1.4 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.

In the Test Agents view it is possible to:

- add a tag to selected Test Agents
- remove a tag from (“untag”) selected Test Agents
- delete all instances of a tag.

In the search box, in addition to the text search function, you can filter the view on a selected subset of tags.

4.1.4.1 Adding tags to Test Agents

- First check the Test Agents that you want to tag.
- Click the Tags button.
- Then type the name of the tag in the text box (see screenshot) and click the Tag button.
The tags will show up as small boxes in the description field.

If you hover over a tag with the mouse, it will expand to display the whole text string of the tag. You can also remove the tag by clicking the red cross that appears on top of it.

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

### 4.1.4.2 Using tags to filter Test Agents

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

Check one or several tags to filter the Test Agents view on these tags. The screenshot below shows the result of filtering on the "workforce" tag.

- To display the full list again, just uncheck 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 check the Tags option and select the name of the tag(s) to filter on. See the example below.

### 4.1.4.3 Untagging Test Agents

To remove a tag from Test Agents:

- First check the Test Agents that you want to untag.
- Click the Tags button.
- Then select the name of the tag to remove (“Lulea” in the example below), and click the Untag button.
- You are prompted to confirm this action. Click Untag.

The tag is now removed from the selected Test Agents.

You can also untag by hovering over a tag and clicking the red cross:
4.1.4.4 Deleting a tag

To delete a tag altogether, do as follows:

- Select at least one Test Agent carrying the tag you want to delete.
- Click the Tags button.
- Select or type a tag name in the input field.
- Click the Delete tag button.

**Warning:** All instances of the tag will be removed. You are prompted to confirm your action:

```
Confirmation

Do you really want to DELETE this tag? This will remove all instances of it.

Delete  Cancel
```

4.2 Installing Test Agents

4.2.1 Getting started with preinstalled Test Agents

Your delivery will contain the ordered combination of Netrounds 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 108).

- 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 Netrounds 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 Netrounds server once the Test Agents have been registered and have come online.
  – If you are using your own on-premise Netrounds 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 118) to set the login server to the Netrounds 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 Netrounds server, make sure that the path between the Test Agent’s management interface and Netrounds 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 112). 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 112).

• Registration (on-premise server only): If you are using an on-premise Netrounds server, register the Test Agent to the previously created Netrounds account on the server. To this end, use the Register option in the Test Agent local console as explained here (page 109). (With a cloud server, the registration will be automatic, as explained below (page 65).)

• After an IP address has been assigned, the Test Agent will connect to your specific Netrounds 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 Netrounds server:
  – in the cloud server case, to https://login.netrounds.com using TCP port 443;
  – 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 Netrounds server will communicate with the Test Agent using the reverse direction on the same TCP session.

• If you are making use of the Netrounds cloud server, the Test Agent will now automatically register with the server and appear in your Netrounds account.

• Log in to your account in the Netrounds 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 52) for full coverage of the Test Agents view.

• A valid license is assigned to the preinstalled Test Agent by Netrounds 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 Netrounds technical support at support@netrounds.com.

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 Netrounds user interface to the Test Agent.

4.2.2 Installing software Test Agents: Introduction

You can install a software Test Agent:

• 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 67) for further instructions.
• on x86 hardware of your own for permanently using that hardware as a Test Agent. See this page (page 69).

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 can be downloaded from here.

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 109) 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 112) 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 Netrounds server: in the cloud server case, to https://login.netrounds.com using TCP port 443; 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 Netrounds 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 Netrounds (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 369) for instructions.

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

This page explains how to install a software Test Agent 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 Netrounds 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 Netrounds 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 drive 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 drive 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:

![Test Agent Admin Menu](image)

- 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 110).
- 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 112). 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 Netrounds server. How to do this is explained here (page 109).

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 Netrounds 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 100).

4.2.4 Installing a software Test Agent on your own hardware

This page explains how to install Test Agent 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 software Test Agent as a bootable image on a USB memory stick (page 67), 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 drive 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 drive.

The hardware will now boot up as a Netrounds 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.

![Test Agent Admin Menu]

- 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 110).

- 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 112). 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 Netrounds server. How to do this is explained [here](#) (page 109).

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 Netrounds 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 100).

### 4.2.5 Installing a Test Agent on virtualization platforms

This page gives an overview of how you can deploy a Netrounds Test Agent on some popular hypervisor platforms. The various Test Agent images are provided either directly from Netrounds or from your NFV orchestration partner. Full information is found in separate technical guides, available at [https://portal.netrounds.com](https://portal.netrounds.com).

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 Netrounds technical support at support@netrounds.com. The Test Agent RAW image may be seen as a replacement for the ISO image which was provided in certain older Netrounds versions.

- **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 Netrounds Control Center (domain name/IP address and port).

- **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.

- **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 Netrounds Control Center using the console once the Test Agent has booted.

- **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”).

- **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.

- **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.

#### 4.2.5.1 Format of “cloud-config” metadata

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

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

```yaml
#cloud-config
netrounds_test_agent:
  name: MyvTA
  email: john.doe@example.com
  password: secret
  account: theaccount
  server: <login-server>:443

  admin_password: secret
  root_password: secret
  ...disable.

  management_interface: eth0
  management_mtu: 1500
  management_address_type: dhcp
  # Test Agent management interface
  # MTU on management interface
  # 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: ntp.netrounds.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. The system would revert to default behaviour, if these parameters are...
4.2.5.2 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.

4.2.5.3 Hardware requirements

The compute resources available to the virtual Test Agent will impact its performance. Hardware requirements for virtual Test Agents are found in the Test Agents data sheet; go to this page to download it.

4.2.5.4 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 Netrounds 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.

4.2.6 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.

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 Netrounds technical support at support@netrounds.com.

4.2.6.1 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 Netrounds server on TCP port 6800 (default).

You do not need to open any incoming connections, since the Netrounds server will communicate in the reverse direction on the same TCP session that the Test Agent Application initiated.

4.2.6.2 Installing a Test Agent Application under Linux

Downloading the application

- Download the Test Agent Application software from Netrounds 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.

Installing the application

• Unpack the Test Agent Application tarball.
• Move the release folder to a more permanent location, for example /opt/test-agent-application:

```bash
sudo mv test-agent-application_<version_number> /opt/test-agent-application
sudo chown -R root:root /opt/test-agent-application
```

Registering the application

To connect the Test Agent to Netrounds Control Center (NCC), run the following command:

```bash
sudo /opt/test-agent-application register \ 
  --config /etc/test-agent-application.conf \ 
  --ncc-host <NCC hostname> \ 
  --account <NCC account> \ 
  --email <NCC user email> \ 
  --password <NCC user password> \ 
  --name <Test Agent name>
```

This will register the Test Agent Application to Netrounds Control Center and write its credentials to /etc/test-agent-application.conf. The Test Agent Application will be tied to your Netrounds account through an encrypted and secure connection, and it will appear in the Test Agents view in Netrounds Control Center with an “offline” icon next to it.

If Netrounds Control Center does not have a valid SSL certificate, add the --ssl-no-check flag to the command. Note that this is not recommended for security reasons.

Starting the application

To start up the registered Test Agent, run this command:

```bash
test-agent-application --config /etc/test-agent-application.conf
```

If Netrounds Control Center does not have a valid SSL certificate, add the --ssl-no-check flag to the above command. Again, note that this is not recommended for security reasons.

Automatic application start

To have systemd start the Test Agent Application automatically on boot, add the following to /etc/systemd/system/test-agent-application.service:

```ini
[Unit]
Description=Netrounds Test Agent Application
After=network.target

[Service]
ExecStart=/opt/test-agent-application --config /etc/test-agent-application.conf
```
If Netrounds Control Center does not have a valid SSL certificate, add the `--ssl-no-check` flag to the ExecStart command.

Reload the systemd units and start the Test Agent Application:

```bash
systemctl daemon-reload
systemctl start test-agent-application.service
```

The application's status icon should now turn green in the Test Agents view, meaning that the application is ready to use.

**Using the Test Agent Application**

Log in to Netrounds Control Center using 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.

**Running a Test Agent Application in a container**

This instruction assumes that the commands are executed in a directory containing the following Dockerfile:

```Dockerfile
FROM debian:buster-slim
RUN apt-get update &&
    apt-get install -y ca-certificates &&
    rm -rf /var/lib/apt/lists/*
COPY test-agent-application /test-agent-application
ENTRYPOINT ["/test-agent-application/test-agent-application"]
```

- Extract the Test Agent Application release tarball, and rename the release directory to test-agent-application.
- Build the docker container:

  ```bash
docker build -t netrounds/test-agent-application .
  ```

- Now register the Test Agent Application to Netrounds Control Center if you have not already done so:

  ```bash
docker run --rm -v $(pwd):/config netrounds/test-agent-application \
    register --config /etc/test-agent-application.conf ... 
  ```

  In place of “...” enter the same additional arguments as above (page 74).

- Start the Test Agent as follows:

  ```bash
docker run -d -v $(pwd):/config netrounds/test-agent-application --config /config/agent.conf
  ```

See the help output for more information about the command line arguments:
Troubleshooting

- Check that no firewalls are blocking the connection to Netrounds Control Center on TCP port 6800 (default).
- Check that Netrounds Control Center has a valid SSL certificate or that you have added the --ssl-no-check flag 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`
- Enable debug logging to get more detailed logging by adding the --log-level DEBUG argument to the Test Agent Application start command.
- If you are still having problems, contact support@netrounds.com and we will help you.

4.2.7 Installing a Test Agent Lite

Test Agent Lite is no longer available for purchase. It is superseded by Test Agent Application, which is installed as described here (page 73).

4.2.8 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 Netrounds GUI

4.3.1 Viewing Test Agent status and properties

To inspect the current status, current activity, and various properties of a Netrounds 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 76).
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:

**Version: 2.22.0**

You can then click the arrow in order to upgrade the Test Agent software to the latest version.

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>IPv4 address (dhtcp)</th>
<th>IPv6 address</th>
<th>MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0 (Management)</td>
<td>172.31.44.61/20 (dhtcp)</td>
<td>2e05:dd1f:67e9:002e:895c:9995:dd58:a491:128 (dhtcp)</td>
<td>0a:23:74:1e:33:56</td>
<td></td>
</tr>
</tbody>
</table>

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 86).
- **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 86).
- **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 87).
- **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 91).
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 73).

4.3.2.2 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

4.3.2.3 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 87).

For an introduction to using mobile interfaces, see also the page on mobile network measurements (page 232).

**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 Netrounds-internal management interface for connection to the Netrounds 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 Netrounds 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. Empty MTU value means that it will not be set. It may be changed at any time to a value specified in a DHCP lease or a Router Advertisement (RA).
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: “192.168.1.24/24”.

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, Netrounds 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:

<table>
<thead>
<tr>
<th>Netmask</th>
<th>Prefix length</th>
<th>Netmask</th>
<th>Prefix length</th>
</tr>
</thead>
<tbody>
<tr>
<td>255.255.255.255</td>
<td>/32</td>
<td>255.254.0.0</td>
<td>/15</td>
</tr>
<tr>
<td>255.255.255.254</td>
<td>/31</td>
<td>255.252.0.0</td>
<td>/14</td>
</tr>
<tr>
<td>255.255.255.252</td>
<td>/30</td>
<td>255.248.0.0</td>
<td>/13</td>
</tr>
<tr>
<td>255.255.255.248</td>
<td>/29</td>
<td>255.240.0.0</td>
<td>/12</td>
</tr>
<tr>
<td>255.255.255.240</td>
<td>/28</td>
<td>255.224.0.0</td>
<td>/11</td>
</tr>
<tr>
<td>255.255.255.224</td>
<td>/27</td>
<td>255.192.0.0</td>
<td>/10</td>
</tr>
<tr>
<td>255.255.255.192</td>
<td>/26</td>
<td>255.128.0.0</td>
<td>/9</td>
</tr>
<tr>
<td>255.255.255.128</td>
<td>/25</td>
<td>255.0.0.0</td>
<td>/8 (Class A)</td>
</tr>
<tr>
<td>255.255.255.0</td>
<td>/24 (Class C)</td>
<td>255.0.0.0</td>
<td>/7</td>
</tr>
<tr>
<td>255.255.254.0</td>
<td>/23</td>
<td>252.0.0.0</td>
<td>/6</td>
</tr>
<tr>
<td>255.255.252.0</td>
<td>/22</td>
<td>248.0.0.0</td>
<td>/5</td>
</tr>
<tr>
<td>255.255.248.0</td>
<td>/21</td>
<td>240.0.0.0</td>
<td>/4</td>
</tr>
<tr>
<td>255.255.240.0</td>
<td>/20</td>
<td>224.0.0.0</td>
<td>/3</td>
</tr>
<tr>
<td>255.255.224.0</td>
<td>/19</td>
<td>192.0.0.0</td>
<td>/2</td>
</tr>
<tr>
<td>255.255.192.0</td>
<td>/18</td>
<td>128.0.0.0</td>
<td>/1</td>
</tr>
<tr>
<td>255.255.128.0</td>
<td>/17</td>
<td>0.0.0.0</td>
<td>/0</td>
</tr>
<tr>
<td>255.255.0.0</td>
<td>/16 (Class B)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 Netrounds server host name (for the public cloud server, https://login.netrounds.com) in order to set up the encrypted management connection to the Netrounds server. For an on-premise Netrounds 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.

“Use public address” option

This option is available for both dynamic and static IPv4 addresses.

Advanced: Use public address:

Enabling this option 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 Netrounds 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 Netrounds, and the details are not dealt with here.

For a Test Agent that is not behind NAT, this option is not applicable. Consequently, you need to use this option only if both client and server are behind NAT (otherwise, simply pick the Test Agent that is behind NAT to act as client).

This option is not applicable to Test Agents Lite.

An example of port forwarding and use of the Use public address option is shown in the diagram below. In this example, UDP is tested on port 5000. Other tasks may use different ports.

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 |nr-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. Netrounds 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 79) 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 Mobile interface configuration details

The Mobile support in Netrounds enables you to run tests and monitoring sessions over a mobile interface using a preinstalled mPCIe 4G modem in a Test Agent provided by Netrounds.

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 87).

4.3.3.1 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.

### Interface usb0

<table>
<thead>
<tr>
<th>Status</th>
<th>Interface</th>
<th>Mobile Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has link</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>Auto</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Hardware timestamping</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**RX:**
- 24 bits/s (3 bytes/s)
- 22326 packets (0 packets/s)

**TX:**
- 32 bits/s (4 bytes/s)
- 22681 packets (0 packets/s)

**DNS:**
- 195.67.199.18,
- 195.67.199.19

**Routes:**
- 0.0.0.0/0 -> 90.236.190:33

**MTU:**
- -

*This will only reset the counters temporarily, until you close this dialog.*

The Interface tab looks as follows:
<table>
<thead>
<tr>
<th>Status</th>
<th>Interface</th>
<th>Mobile Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Management</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAT (mode/band)</td>
<td>LTE - AUTO</td>
<td></td>
</tr>
<tr>
<td>APN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP version</td>
<td>IPv4 &amp; IPv6 mixed</td>
<td></td>
</tr>
</tbody>
</table>

- **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 77) 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.
### 4.3.3.2 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.

4.3.3.3 Related topics

• Use the Mobile logger (page 232) task to collect mobile network measurements.
• Use the Mobile switcher (page 234) task to change mobile interface parameters.

4.3.4 Wi-Fi interface configuration details

The Wi-Fi support in Netrounds enables you to run tests and monitors over a Wi-Fi interface. For details on supported hardware, see here (page 228).

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 231) 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 77).

4.3.4.1 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.

<table>
<thead>
<tr>
<th>Interface eth0</th>
<th>Reset packet counters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status</strong></td>
<td></td>
</tr>
<tr>
<td>Has link: Yes</td>
<td></td>
</tr>
<tr>
<td>Speed: undefined</td>
<td></td>
</tr>
<tr>
<td>Address: 172.31.44.61/20 (dhcp)</td>
<td></td>
</tr>
<tr>
<td>2a05:0016:07a:9002:e663:9995:dd0b:416/128 (dhcp)</td>
<td></td>
</tr>
<tr>
<td>MAC: 0a:23:74:1e:33:56</td>
<td></td>
</tr>
<tr>
<td>Hardware timestamping: No</td>
<td></td>
</tr>
<tr>
<td>RX: 4.1 Mbit/s (495.1 KiB/s)</td>
<td></td>
</tr>
<tr>
<td>35593587 packets (376 packets/s)</td>
<td></td>
</tr>
<tr>
<td>4.0 Mbit/s (493.7 KiB/s)</td>
<td></td>
</tr>
<tr>
<td>TX: 39145668 packets (388 packets/s)</td>
<td></td>
</tr>
<tr>
<td>DNS: 172.31.0.2</td>
<td></td>
</tr>
<tr>
<td>Routes: 0.0.0.0/0 -&gt; 172.31.32.1</td>
<td></td>
</tr>
<tr>
<td>::/0 -&gt; fe80::233:7179:32d.786e</td>
<td></td>
</tr>
<tr>
<td>MTU: 1500 (IPv4)</td>
<td></td>
</tr>
<tr>
<td>1500 (IPv6)</td>
<td></td>
</tr>
</tbody>
</table>

Note: This will only reset the counters temporarily, until you close this 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

<table>
<thead>
<tr>
<th>Status</th>
<th>Wi-Fi Status</th>
<th>Interface</th>
<th>IPv4 address</th>
<th>IPv4 DHCP server</th>
<th>IPv6 address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### General:
- **BSSID**: 00:0b:9c:0e: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
- **TX bitrate**: 866.7 MBit/s
- **TX MCS**: 9
- **TX SGI**: yes
- **TX MIMO**: 2
- **RX bitrate**: 866.7 MBit/s
- **RX MCS**: 9
- **RX SGI**: yes
- **RX MIMO**: 2
**Interface tab**

- **Management**: Check this box if the Wi-Fi interface is to be used for Test Agent management. Compare the page on *Test Agent interface configuration* (page 77) 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 229) 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 check 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 Select access point is unchecked, 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.

- **Security**: The type of authentication used in the Wi-Fi network. One of:

– **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 Security. The following parameters occur:

- **Cipher**:
  - AUTO: The cipher type is automatically selected by the Wi-Fi card.
  - 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**: Check this to allow 40 MHz channels. If unchecked, only 20 MHz channels are allowed.
  - **MCS index**: Check 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.
  - **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. Check 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 81).

IPv6 Address tab

This tab is the same as for a regular, wired interface; it is covered here (page 84).

4.3.4.2 Related topics

• Use the Wi-Fi logger (page 230) task to collect Wi-Fi network measurements.
• Use the Wi-Fi switcher (page 231) task to change Wi-Fi interface parameters.
• Use the Wi-Fi scanner (page 229) task to scan for Wi-Fi networks.

4.3.5 Configuration of applications

On the Applications tab of the Test Agent configuration dialog, you can enable and disable the following Netrounds applications: Proxy, Speedtest, and Live remote packet capture.

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Applications</th>
<th>NTP</th>
<th>Streams</th>
<th>License</th>
<th>Utilis</th>
<th>GPS Location</th>
<th>Platform Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
<td>Start/Stop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speedtest</td>
<td>Speedtest</td>
<td>Enable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proxy</td>
<td>Proxy for management traffic</td>
<td>Enable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Live remote packet capture</td>
<td>Live capture from Test Agent</td>
<td>Enable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 338).

Proxy: Proxy functionality for Netrounds management, used when some of your Test Agents do not have direct Internet access. Read more here (page 343).
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 until 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.

Netrounds also has another packet capture function, *Remote Packet Capture* (page 335) 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 Netrounds web interface without the aid of Wireshark.

### 4.3.6 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 362). These measurements are dependent on an accurate clock. By default, Test Agents will synchronize their internal clock to an NTP server provided by Netrounds (ntp.netrounds.com); 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.

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Applications</th>
<th>NTP</th>
<th>Streams</th>
<th>License</th>
<th>Utilis</th>
<th>GPS Location</th>
<th>Platform Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time offset:</td>
<td>0</td>
<td>-0.000 ms</td>
<td>Sync NTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface:</td>
<td>0</td>
<td>(Same as management)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server:</td>
<td>0</td>
<td>169.254.169.123</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable IPv6:</td>
<td>0</td>
<td></td>
<td>Restore defaults</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Time offset:** 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 a stricter requirement on synchronization: the maximum time offset allowed is 4 ms. If this requirement is not met, a UDP task will still start and will produce other measurements, but no one-way delay will be obtained.
- 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 Sync NTP button restarts and resets the Test Agent’s NTP client. Doing this will not usually improve the synchronization, however. Normally, given an adequate connection, the NTP sync will be good enough for all kinds of testing. If it is not, check the connection.

Interface: The Test Agent network interface used for time synchronization.

Server: The domain name or IP address of the NTP server.

If you are using one Netrounds Test Agent as a proxy (page 343) 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 ntp.netrounds.com.

If you are running the Test Agent in Amazon WS, 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.

4.3.6.1 Using multiple NTP servers

To make use of multiple NTP servers, enter a DNS name into the Server field (for example, pool.ntp.org) which resolves to multiple A records, each pointing to an NTP server. The Test Agent will then pick one of the NTP servers, and if that one is unavailable it will try another. However, only one NTP server will be used at any given time.

It is not possible to enter multiple NTP server addresses directly into the Server field.

4.3.7 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 of Netrounds products, as detailed here (page 138).
Running as well as completed jobs can also be viewed on the Tests and Monitoring screens accessed from the main menu.

### 4.3.8 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 software Test Agents. Netrounds preinstalled Test Agents, by contrast, have licenses assigned at delivery.

Read more about licensing on this page (page 138).

The following screenshot shows a Test Agent with an unassigned license:
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.

<table>
<thead>
<tr>
<th>Name</th>
<th>License</th>
<th>No. of streams</th>
<th>Used streams</th>
<th>Available streams</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws-eu-north-1-1</td>
<td>SW-Test Agent Large</td>
<td>500</td>
<td>35</td>
<td>462</td>
</tr>
<tr>
<td>aws-eu-north-1-2</td>
<td>SW-Test Agent Large</td>
<td>500</td>
<td>8</td>
<td>492</td>
</tr>
</tbody>
</table>

4.3.9 Test Agent utilities

The Utils tab of the Test Agent configuration dialog holds some useful utilities for troubleshooting and management.
4.3.9.1 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 Netrounds 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
```

4.3.9.2 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
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
```
4.3.9.3 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:f6:3 <REACHABLE>
```

4.3.9.4 Update tab

Here you can manually initiate a Test Agent software update. Note that the Test Agent will normally be restarted following the update.

4.3.9.5 Reboot tab

From this tab you can reboot the Test Agent.

4.3.9.6 Unregister tab

From here you can unregister the Test Agent from the Netrounds system.

Note that an unregistered Test Agent cannot be re-registered. If you have accidentally unregistered one of your Test Agents, please contact Netrounds technical support at support@netrounds.com.

4.3.10 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”.

### 4.3.11 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.

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Applications</th>
<th>NTP</th>
<th>Streams</th>
<th>License</th>
<th>Util</th>
<th>GPS Location</th>
<th>Platform Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform:</td>
<td>unknown-pc</td>
<td>PC Engines</td>
<td>AuthenticAMD</td>
<td>1.0</td>
<td>4.0.7</td>
<td>AMD GX-412TC SCC</td>
<td>Memory: 1.8 GiB</td>
</tr>
</tbody>
</table>

### 4.3.12 Managing Test Agent Applications and Test Agents Lite

Clicking a Test Agent Application or Test Agent Lite opens a dialog displaying some of the properties of the device. The dialogs are similar for both types of Test Agent and are therefore covered on the same page.

Test Agent Applications and Test Agents Lite 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.
4.3.12.1 Details tab *(Test Agent Lite only)*

<table>
<thead>
<tr>
<th>Details</th>
<th>Interfaces</th>
<th>Streams</th>
<th>Licenses</th>
<th>Utils</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Linux docker-vm 4.18.0-21-generic #22~18.04.1-Ubuntu SMP Thu May 16 15:07:19 UTC 2019 x86_64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Offline (Last seen: )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Agent version</td>
<td>unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last known public address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3.12.2 Interfaces tab

This tab shows all interfaces on the computer where the Test Agent is running.

<table>
<thead>
<tr>
<th>Details</th>
<th>Interfaces</th>
<th>Streams</th>
<th>Licenses</th>
<th>Utils</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
<td>IPv4 address</td>
<td>IPv6 address</td>
<td>MAC address</td>
</tr>
<tr>
<td>WNC 5</td>
<td></td>
<td>169.254.228.145/24</td>
<td></td>
<td>7e.7a:91:46:55:ba</td>
</tr>
<tr>
<td>WNC 4</td>
<td></td>
<td>169.254.54.179/24</td>
<td></td>
<td>7e.7a:91:46:55:b9</td>
</tr>
</tbody>
</table>

4.3.12.3 Streams tab

This tab lists all streams that are being used by monitors and tests currently running on the Test Agent.

<table>
<thead>
<tr>
<th>Details</th>
<th>Interfaces</th>
<th>Streams</th>
<th>Licenses</th>
<th>Utils</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitorings</td>
<td></td>
<td>Using 0 of 6800 streams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Streams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No running monitoring sessions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3.12.4 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.
4.3.12.5 Unregister tab

This tab holds a button for unregistering the Test Agent.

Note that to re-register a Test Agent Application or Test Agent Lite, 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 Netrounds account.

4.3.13 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 checking 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. 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 126), or via cloud-config (cloud-init). They cannot be deleted from Netrounds Control Center.

### 4.3.13.1 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:w6gKxgzBDOxFDbmKp9p+3Y3a437J0IuiM3/oNQac/a2s test-agent-key
The key's randomart image is:
+---[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

4.4 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 108).

4.4.1.3 Software Test Agent

Connect a standard USB or PS/2 keyboard and VGA display to the device where the software 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.

4.4.1.4 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 the document “How to Deploy Virtual Test Agents in OpenStack”, available at https://portal.netrounds.com.

4.4.2.1 Registration process

- Use the text-based menu and navigate to Register.

![Test Agent Admin Menu]

You will be asked whether you want to enable Test Agent management over IPv6. If enabled, this is configured as described here (page 112). The registration process itself is the same regardless of your choice here.

![Do you want enable IPv6 for management traffic?]

- Under Server, you should normally use the host name or IP address of the Netrounds server.
  - In the cloud server case, this is https://login.netrounds.com. For this server alone, the Test Agent will connect to port 443.
  - 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 343), you need to point to the static IP address of the proxy Test Agent, with port 443 specified.
- Under E-mail, enter the email address that serves as user name for your Netrounds account.
• Under Test Agent name, enter a name for the Test Agent. This name is what will be shown in the Test Agents view.

The message “Registration successful” should now appear. The Test Agent will then be visible in the Test Agents view for your Netrounds account: https://<Control Center host IP>/<your account>/genalyzer.

4.4.3 Configuring a Test Agent from the local console

4.4.3.1 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://<nr-product> 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 Netrounds account.
- **Connecting**: The Test Agent is trying to connect to the Netrounds server.
- **Logged in**: The Test Agent is connected and logged in to the Netrounds 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.

### 4.4.3.2 Configuring the management interface on a Test Agent

For configuration of Test Agent management over a mobile network interface, see [this page](#) (page 127).
- In the text-based menu, navigate to Configure management.

```plaintext
Test Agent Admin Menu

0 Register
1 Show system status
2 Show interface status
3 Configure management
4 Utilities
5 Exit
```

- Select which interface should handle management traffic to and from the server (preferably, “eth0”). Configuration of the other interfaces is done via the Netrounds server; see *Test Agent interface configuration* (page 77).
• Then select interface speed and duplex on the management port.

• You are also asked whether to use a VLAN tag for the management interface. Choose Yes or No. If you choose 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 109) 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 115).)
• 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 server setting follows. You can use either the default NTP server, ntp.netrounds.com, or a local one (specified as an IP address or host name). 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 98). To access this service, set NTP server to 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.
4.4.3.3 Using Test Agent utilities

• To access the Test Agent utilities, navigate to Utilities in the text-based menu.

Changing login server

• To use Netrounds’ proprietary proxy function or a standard HTTP (web) proxy, navigate to Change login server. The concept of Test Agent proxy is described here (page 343).
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.

Choose No if you are not using a standard HTTP proxy.

Check that the settings are correct, then choose 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**

- Fill in the requested credentials.
Is the following information correct?

Server: staging:6000
HTTP Proxy: Yes
Proxy Server: proxy.example.com:8080
Proxy Authentication type: basic
Proxy Username: test

Ping

- Select interface and enter the host to ping.
The Ping session is displayed as follows:

```
PING webc.sunet.se (192.36.171.231) from 10.0.2.15: 56(84) bytes of data.
64 bytes from 192.36.171.231: icmp_seq=1 ttl=68 time=12.7 ms
64 bytes from 192.36.171.231: icmp_seq=2 ttl=68 time=12.6 ms
64 bytes from 192.36.171.231: icmp_seq=3 ttl=68 time=12.3 ms
64 bytes from 192.36.171.231: icmp_seq=4 ttl=68 time=12.3 ms
--- webc.sunet.se ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 1505ms
rtt min/avg/max/mdev = 12.324/12.520/12.720/0.173 ms
```

**Traceroute**

- Select interface and enter the host to traceroute.
The Traceroute session is displayed as follows:

```
traceroute to www.sunet.se (192.36.171.231), 30 hops max, 60 byte packets
 1 10.0.2.2  0.260 ms  0.129 ms  0.137 ms
 2 192.168.1.1  0.674 ms  0.759 ms  0.612 ms
 3 195.22.87.1  1.194 ms  0.979 ms  1.082 ms
 4 213.50.154.25  1.234 ms  1.268 ms  1.391 ms
 5 62.95.54.122  13.558 ms  13.407 ms  13.159 ms
 6 195.245.240.24  12.109 ms  12.055 ms  12.479 ms
 7 103.105.102.18  12.325 ms  12.561 ms  12.401 ms
 8 192.36.171.231  12.201 ms  12.132 ms  12.057 ms
```

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.

Provided that the two entries match, the admin password will be changed and a confirmation will be given as shown below.

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.
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 107).
- Enter the SSH key.

Note: The SSH key must not have any options specified. 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 Netrounds Control Center; see this page (page 106). Note that keys added from the local console cannot be managed from Control Center but can only be displayed there.

4.4.4 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 110).

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"). See the Test Agent datasheet.
• In the text-based menu, navigate to Configure management.

![Test Agent Admin Menu]

- Select which interface should handle management traffic to and from the server (here, “usb0”). Configuration of the other interfaces is done via the Netrounds server; see Test Agent interface configuration (page 77).

![Select management interface]

- Enter the APN (Access Point Name) for the mobile subscription you are using.

![Enter APN]

- 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, ntp.netrounds.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.
4.4.5 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 110).

4.4.5.1 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, ntp.netrounds.com.
- Finally, confirm the settings for the management port.

All interfaces are now reset to the factory default, with management on eth0 using DHCP.

4.4.5.2 Resetting the NTP server

- To reset the NTP server to the Netrounds 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 98) of its configuration dialog. Click Restore defaults.

4.4.5.3 Restoring the login server and the HTTP proxy

(Here, the Netrounds server is assumed to reside in the public cloud.)

- Navigate to Change login server.
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

Hardware specifications for preinstalled Test Agent hardware currently available from Netrounds are found in the Netrounds Test Agent datasheet.

To obtain preinstalled Test Agents, please contact Netrounds sales at sales@netrounds.com.

Below are specifications for hardware which is no longer available for purchase but is still supported and included here for reference:

- **HW Small** (page 131)
- **HW Medium Plus** (page 132)

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 135) for a description of the front panel LEDs on the HW Small device.

**Note:** This device is no longer available for purchase.
<table>
<thead>
<tr>
<th><strong>Operating system</strong></th>
<th>Debian distribution with real-time Linux kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong></td>
<td>1 × AMD Geode LX 800, 500 MHz</td>
</tr>
<tr>
<td><strong>CPU clock speed</strong></td>
<td>500 MHz (single)</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>256 MB</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>2 GB CF (flash)</td>
</tr>
<tr>
<td><strong>Network interfaces</strong></td>
<td>3 × 10BASE-T/100BASE-TX (RJ45)</td>
</tr>
<tr>
<td><strong>Ports</strong></td>
<td>1 × serial RS-232 + 2 × USB</td>
</tr>
<tr>
<td><strong>Voltage</strong></td>
<td>18 V (6 W), 230 V network adapter included, 1.8 m (6 ft) cord</td>
</tr>
<tr>
<td><strong>Dimensions (H × D × W)</strong></td>
<td>30 × 157 × 168 mm (1.2 × 6.2 × 6.6 in)</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>800 g (1.8 lb)</td>
</tr>
</tbody>
</table>

**4.5.1.2 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.

**Note:** This device is no longer available for purchase.
### 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)

#### 4.5.2 Preinstalled Test Agent performance characteristics

Below is an overview of performance characteristics of the various hardware models on which Test Agents can be preinstalled.
<table>
<thead>
<tr>
<th>Performance category</th>
<th>HW Small</th>
<th>HW Medium</th>
<th>HW Medium Plus</th>
<th>HW Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bidirectional UDP, 1500 byte packets</td>
<td>Line rate</td>
<td>Line rate</td>
<td>Line rate</td>
<td>Line rate</td>
</tr>
<tr>
<td>100 Mbit/s</td>
<td>500 Mbit/s</td>
<td>1000 Mbit/s</td>
<td>4 Gbit/s</td>
<td></td>
</tr>
<tr>
<td>Unidirectional TCP, multiple sessions</td>
<td>Line rate</td>
<td>Line rate</td>
<td>Line rate</td>
<td>Line rate</td>
</tr>
<tr>
<td>100 Mbit/s</td>
<td>1000 Mbit/s</td>
<td>1000 Mbit/s</td>
<td>10 Gbit/s</td>
<td></td>
</tr>
<tr>
<td>IPTV MPEG measurement</td>
<td>90 Mbit/s</td>
<td>300 Mbit/s</td>
<td>900 Mbit/s</td>
<td>3 Gbit/s</td>
</tr>
<tr>
<td>Equivalent number of SD channels at 4 Mbit/s</td>
<td>22</td>
<td>75</td>
<td>225</td>
<td>750</td>
</tr>
<tr>
<td>Traffic generation performance (packets per second)</td>
<td>19,500 pps</td>
<td>78,000 pps</td>
<td>390,000 pps</td>
<td>390,000 pps</td>
</tr>
<tr>
<td>Concurrent TCP sessions</td>
<td>500</td>
<td>500</td>
<td>1000</td>
<td>2000</td>
</tr>
<tr>
<td>Accuracy</td>
<td>About 1 ms</td>
<td>About 1 ms</td>
<td>About 1 ms</td>
<td>About 1 ms</td>
</tr>
</tbody>
</table>

For the HW Medium hardware, please refer to the page *HW Medium hardware TCP/UDP performance* (page 134) for more detailed information about TCP/UDP performance.

### 4.5.3 HW Medium hardware TCP/UDP performance

The HW Medium Test Agent hardware has TCP/UDP performance characteristics as described below. The Test Agent was evaluated in a controlled lab environment with no other source than the system under test.
4.5.3.1 TCP performance

TCP throughput, unidirectional: 941 Mbit/s (line rate)
This is equal to the theoretical throughput.

4.5.3.2 UDP performance

UDP throughput, unidirectional, for different packet sizes:

<table>
<thead>
<tr>
<th>Packet size (bytes)</th>
<th>Throughput (Mbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1518</td>
<td>987 (line rate)</td>
</tr>
<tr>
<td>1280</td>
<td>900</td>
</tr>
<tr>
<td>1024</td>
<td>700</td>
</tr>
<tr>
<td>512</td>
<td>300</td>
</tr>
<tr>
<td>256</td>
<td>100</td>
</tr>
<tr>
<td>128</td>
<td>70</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
</tr>
</tbody>
</table>

UDP throughput, bidirectional, for different packet sizes:

<table>
<thead>
<tr>
<th>Packet size (bytes)</th>
<th>Throughput (Mbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1518</td>
<td>500</td>
</tr>
<tr>
<td>1280</td>
<td>400</td>
</tr>
<tr>
<td>1024</td>
<td>350</td>
</tr>
<tr>
<td>512</td>
<td>200</td>
</tr>
<tr>
<td>256</td>
<td>100</td>
</tr>
<tr>
<td>128</td>
<td>50</td>
</tr>
<tr>
<td>64</td>
<td>20</td>
</tr>
</tbody>
</table>

4.5.4 HW Small hardware LED indicators

This page describes how to interpret the LED indicators on the front panel of the HW Small hardware unit.
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.

4.5.5 Test Agents: Frequently asked questions

4.5.5.1 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 Netrounds implementation also ensures that network packet processing always gets top priority in the CPU.

4.5.5.2 What is the data rate of the control traffic from a Test Agent to the Netrounds 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 Netrounds server.

4.5.5.3 What happens if the management link between a Test Agent and the Netrounds server is lost?

All your Test Agents will continue their activities even if the connection to the Netrounds server is temporarily lost. The Test Agent will store all measurements locally for up to one hour and upload them to the Netrounds server.
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 executing a distributed and automated test, it will be aborted if one of the Test Agents becomes unavailable during the test.

4.5.5.4 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, Netrounds 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 Netrounds is set to 100 ms in this case.

4.5.5.5 Do Test Agents require calibration?

No calibration is needed for Netrounds Test Agents. Calibration is most relevant for hardware-based components with extremely high accuracy (in the order of nanoseconds).

4.5.5.6 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 Netrounds 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.

4.5.5.7 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.
4.5.5.8 Is it possible to install a downloadable Test Agent on Apple hardware?

No, this is currently not supported.

5 Licensing

5.1 Licensing and streams in Netrounds

This page describes how the licensing and its enforcement work for Netrounds accounts. It also describes the stream concept and how it is applied in the context of licensing.

5.1.1 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, do 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 60) for details.

A Test Agent is activated when it is first registered with your Netrounds account. Deactivating a Test Agent from your account must be done manually.

The total number of Test Agents registered to a Netrounds 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.

5.1.2 Streams

The concept of stream in Netrounds is an abstract one used for licensing purposes. It is basically something which a Test Agent consumes in executing tests and monitors.

Netrounds 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Streams</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Monitor</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Web sites monitor</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

- **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.

### 5.1.2.1 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 139) 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) \( \times \) (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.

**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.

**5.1.2.2 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.

DVB-C MPEG

On each Receiver, one stream is used for each DVB-C channel.

OTT - HLS

One stream used on each Client.

IPTV channel zapping

One stream used on each Client, regardless of the number of Channels selected.

5.1.2.3 Stream consumption: HTTP and DNS

HTTP

One stream used on each Client.

DNS

One stream used on each Client.

5.1.2.4 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.

5.1.2.5 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.

5.1.2.6 Stream consumption: Security testing

One stream used on each of Customer and ISP. (Some security tasks only have a subset of these roles.)

5.1.2.7 Stream consumption: Ethernet service activation testing

One stream each used on Sender and Receiver.

5.1.2.8 Stream consumption: Transparency testing

One stream each used on Sender and Receiver.

5.1.2.9 Stream consumption: Reflector-based testing

- Y.1731 (page 273) tasks: On each Client, one stream used for each MEP entered.
- TWAMP/TWAMP Light (page 286) task: On each Sender, one stream used for each Reflector and Test Agent Reflector entered.
- TWAMP Reflector (page 291) task: One stream used for each Reflector (Test Agent interface) entered.
- Ping (page 293) task (Ping to multiple defined hosts): On each Client, one stream used per Host entered.
- BWPing (page 295) task: One stream used. (Only one Sender and one Host can be selected in each task.)
- Path trace (page 299) task: The number of streams used is equal to the value of the Max TTL parameter.
- UDP loopback (page 306) task: One stream used. (Only one Client and one Host can be selected in each task.)
5.1.2.10 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”.

5.1.3 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).

5.1.3.1 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.

5.1.3.2 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.

5.1.4 Users in Netrounds

A user is defined as a named user within a Netrounds account with the 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 Netrounds 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 138) section of this page.
6 Working with tests and monitors

6.1 Introduction to tests and monitors

Measurements in Netrounds are conducted mainly in tests and monitors. (Besides these, Test Agents can also run a number of applications. They are covered here (page 335).)

6.1.1 Definitions

A test consists of one or several steps, which are executed sequentially. Each step has a specified, finite duration, and entails running one task 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.
6.1.2 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 145)
- Building monitors (page 154)

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.

6.1.3 Prerequisites

The prerequisites for creating both tests and monitors are an active Netrounds account and a number of registered Netrounds Test Agents (the precise number depends on the kind of testing to be performed). In some cases, Test Agents Lite or Test Agent Applications can be used; their functionality is however limited. See this page (page 49) for full details.

If you haven’t yet installed your own Netrounds Test Agents, please consult the installation guides found here (page 64).

6.2 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).
If test templates (page 163) 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.

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. A standard Cubic TCP implementation is used.

When a TCP task starts, the Test Agents will start sending a TCP stream between them. This TCP stream will compete for bandwidth with all other traffic on the network link, thus giving a view of the available performance on that link.
6.2.1 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.

6.2.2 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.

6.2.3 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.

6.2.4 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.

6.2.5 Removing an element from a test sequence

You can remove a task or a step from a test sequence by clicking its Remove button.

6.2.6 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 149).

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 163).

6.3 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 check the Schedule box 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 152).
6.4 The Tests view

The Tests view lists all tests currently defined in the Netrounds 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 175).
- **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 354).

6.4.1 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 152).

6.4.2 Multi-select function

You can select a test by checking the box on the far left. You can select all tests by clicking the checkmark box at the top of the list and choosing “All”. Choose “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 152).)

6.4.3 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.
6.4.4 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 145).

6.5 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server MTU</td>
<td>1500</td>
</tr>
<tr>
<td>TCP Implementation</td>
<td>Linux TCP stack as in RFC 753, 1122, 2001 with NewReno and SACK extensions</td>
</tr>
<tr>
<td>Number of TCP streams</td>
<td>2</td>
</tr>
<tr>
<td>Socket buffer size</td>
<td>4.0 KB</td>
</tr>
</tbody>
</table>

- For a service performance test such as TCP or UDP, a table like the one below is displayed.

<table>
<thead>
<tr>
<th>Stream</th>
<th>ES history</th>
<th>Rate (Mbps)</th>
<th>Loss (%)</th>
<th>Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws-eu-north-1-1.eth0 (IPv4) (server) -&gt; aws-eu-north-1-2.eth0 (IPv4) (client)</td>
<td></td>
<td>10</td>
<td>0.00</td>
<td>0.03</td>
</tr>
<tr>
<td>aws-eu-north-1-1.eth0 (IPv4) (server) -&gt; aws-eu-north-1-2.eth0 (IPv4) (client)</td>
<td></td>
<td>10</td>
<td>0.00</td>
<td>0.02</td>
</tr>
</tbody>
</table>

- The ES history bar shows the history of *errored seconds* (page 363) 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 160)) and are simply a subset of such a report.

• Where relevant, a timestamped event log appears (for the test in its entirety).
• For security tests, a listing of operations is provided, with a pass/fail outcome indicated for each operation.

6.5.1 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 160).

• 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 160).

6.6 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).

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 163) 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 344) for more information.

6.6.1 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.

6.7 The Monitoring view

The Monitoring view lists all monitors currently defined in the Netrounds 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 175).
- Tags: Tags applied to the monitor. See the page Applying tags to monitors and templates (page 173).
- 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 354).

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 158).
6.7.1 Multi-select function

You can apply a number of functions to monitors that you have selected by checking their boxes on the far left. You can select all monitors by clicking the checkmark button at the top and choosing “All”. Choose “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):

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag button</td>
<td>See the page <em>Applying tags to monitors and templates</em> (page 173).</td>
</tr>
<tr>
<td>Play button</td>
<td>Click this button to start a monitor that is currently stopped.</td>
</tr>
<tr>
<td>Stop button</td>
<td>Click this button to stop a monitor that is currently running.</td>
</tr>
<tr>
<td>Trash can button</td>
<td>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.)</td>
</tr>
</tbody>
</table>

6.7.2 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 check the Tags box, 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.
6.7.3 Applying tags to monitors

See the page Applying tags to monitors and templates (page 173).

6.7.4 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 154).

6.8 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:

<table>
<thead>
<tr>
<th>Client</th>
<th>ES history</th>
<th>Response time average (ms)</th>
<th>First byte received (ms)</th>
<th>Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws-eu-north-1-1: eth0 (IPv4)</td>
<td><img src="image" alt="ES history bar" /></td>
<td>17</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>aws-eu-north-1-2: eth0 (IPv4)</td>
<td><img src="image" alt="ES history bar" /></td>
<td>13</td>
<td>13</td>
<td>0</td>
</tr>
</tbody>
</table>

- The ES history bar shows the history of *errored seconds* (page 363) 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 160)) 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 372).

Click this button to show event log messages for the monitoring session. These messages are by default hidden.

<table>
<thead>
<tr>
<th>Level</th>
<th>Message</th>
<th>Start time</th>
<th>End time</th>
</tr>
</thead>
</table>

6.8.1 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 160).

• Export button: Export monitoring results in comma-separated format. Again, see the page Reports on tests and monitors (page 160) for details.

• Delete button: Clicking this button will delete the monitor.
Warning: This action cannot be undone, and you are prompted to confirm it.

6.9 Reports on tests and monitors

In Netrounds 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 Netrounds 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 (*.csv).
– For tests whose output consists of an event log, that log is reproduced in a plain-text file (*.txt).

Below are some extracts from a report:
HTTP

Test agents

Clients:

<table>
<thead>
<tr>
<th>Client Name</th>
<th>MAC Address</th>
<th>IPv4 Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws-eu-north-1-1.eth0 (IPv4)</td>
<td>0a:23:74:1e:33:56</td>
<td>172.31.44.61/20</td>
</tr>
<tr>
<td>aws-eu-north-1-2.eth0 (IPv4)</td>
<td>0a:46:21:08:c0:0e</td>
<td>172.31.46.181/20</td>
</tr>
</tbody>
</table>

General

URL: http://1.1.1.99:80/test.html

Time between requests (s): 5.0

Thresholds for errored seconds (ES)

Response Code: No response code validation

Timeout: 200

Advanced

Lifetime (ms): 4000
Here is a sample of what the graphs may look like:

![Graph Image]

### 6.10 Creating templates

The template feature in Netrounds 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.

6.10.1 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, check the Template input box.
- 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 Netrounds orchestration guides for further details.)

In the screenshot below, Clients and URL are treated as template inputs.
6.10.1.1 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:

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 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 145):
6.10.2 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 166)).
- 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 154).

6.10.3 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 Netrounds 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.

![URL Input 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

![Input Table]

Warning: changing variable names in existing templates might break existing orchestration scripts. Be sure you know what you are doing!

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.

6.10.4 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.

### 6.11 Applying tags

Tags defined in Netrounds can be applied to:

- monitors
- templates
- Test Agents
- TWAMP reflectors.

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.

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;
- delete all instances of a tag.

In the search box, in addition to the text search function, you can filter the view on a selected subset of tags.

### 6.11.1 Adding tags to monitors

- First check the monitors that you want to tag.
- Click the Tags button at the top of the page.
- 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 as a small box in the Tags column.

If you hover over a tag with the mouse, it will expand to display the whole text string of the tag.

It is possible to add several tags to the same monitor, as shown below:

### 6.11.2 Using tags to filter monitors

- Check the Tags checkbox in the search field, and a drop-down list appears holding all defined tags.
- Check one or several tags to filter the Monitoring view on these tags.
- To display the full list again, just uncheck the tag names in the search field.

### 6.11.3 Untagging monitors

To remove a tag from monitors:

- First check 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 (“North” in the example below), and click the Untag button.

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

The tag is now removed from the selected monitors.

You can also untag by hovering over a tag and clicking the red cross that appears on it:
6.11.4 Deleting a tag

To delete a tag altogether, do as follows:

- Select at least one monitor that carries the tag you want to delete.
- Select the tag name in the drop-down box (“AWS” in the example that follows).
- Click the Delete tag button.

**Warning:** All instances of the tag will be removed. You are prompted to confirm your action:

```
Confirmation

Do you really want to DELETE this tag? This will remove all instances of it.

Delete  Cancel
```

6.12 Icons used for tests and monitors

6.12.1 Icons for tests

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>The test is scheduled to be run at a future time.</td>
</tr>
<tr>
<td>⌛</td>
<td>The test is pending or waiting to be run.</td>
</tr>
<tr>
<td>🔵</td>
<td>The test is running.</td>
</tr>
<tr>
<td>🟢</td>
<td>The test has completed and passed the test criteria.</td>
</tr>
<tr>
<td>🟠</td>
<td>The test has completed but failed the test criteria.</td>
</tr>
<tr>
<td>⚠️</td>
<td>The test has aborted because of an error.</td>
</tr>
<tr>
<td>🕳️</td>
<td>The test has been canceled manually.</td>
</tr>
<tr>
<td>⏰</td>
<td>The test has been skipped.</td>
</tr>
</tbody>
</table>
6.12.2 Icons for monitors

- 🔄 A monitoring session is running.
- ⏳ A monitoring session has been stopped manually.

7 Task types

7.1 Overview of task types

The following table lists the task categories found in the Netrounds 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 149), 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.

<table>
<thead>
<tr>
<th>Task category</th>
<th>Test</th>
<th>Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network performance (page 179)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>IPTV and OTT video (page 201)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HTTP and DNS (page 218)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SIP (page 224)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Mobile network (page 232)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Wi-Fi network (page 228)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Ethernet service activation (page 234)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Transparency (page 249)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reflector-based (page 273)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Security (page 312)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Utilities (page 325)</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

7.2 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.

7.2.1 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 363) that may occur without triggering a fail for this test step. Default: 0.

---

1 All tasks can be run in monitors except TCP throughput test according to RFC 6349 (page 193) and QoS policy profiling (page 197).
2 The Mobile logger (page 232) task can be used in monitors, while the Mobile switcher (page 234) task is available in tests only.
3 The Wi-Fi logger (page 230) task can be used in monitors, while the Wi-Fi switcher (page 231) task is available in tests only.
4 All tasks can be run in monitors except BWping (page 295).
7 TASK TYPES

• 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.

7.2.2 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.

7.2.3 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%.

7.3 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.
<table>
<thead>
<tr>
<th>Task category</th>
<th>Tasks supporting IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network performance</td>
<td>UDP</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
</tr>
<tr>
<td></td>
<td>Multicast UDP</td>
</tr>
<tr>
<td></td>
<td>VoIP UDP</td>
</tr>
<tr>
<td>IPTV and OTT video</td>
<td>IPTV MPEG</td>
</tr>
<tr>
<td></td>
<td>IPTV channel zapping</td>
</tr>
<tr>
<td>HTTP and DNS</td>
<td>DNS</td>
</tr>
<tr>
<td></td>
<td>HTTP</td>
</tr>
<tr>
<td>SIP</td>
<td>–</td>
</tr>
<tr>
<td>Ethernet service activation</td>
<td>–</td>
</tr>
<tr>
<td>Transparency</td>
<td>L2 Ethernet control protocols</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – Custom Ethertype</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – Custom VLAN</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – Ethertypes</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – IPv6</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – MAC address limit</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – Multicast</td>
</tr>
<tr>
<td></td>
<td>L2 transparency – VLAN</td>
</tr>
<tr>
<td>Reflector-based</td>
<td>ETH-DM</td>
</tr>
<tr>
<td></td>
<td>ETH-LB</td>
</tr>
<tr>
<td></td>
<td>ETH-SLM</td>
</tr>
<tr>
<td></td>
<td>Ping</td>
</tr>
<tr>
<td></td>
<td>BWPing</td>
</tr>
<tr>
<td></td>
<td>TWAMP Light</td>
</tr>
<tr>
<td>Security</td>
<td>–</td>
</tr>
</tbody>
</table>
7.4 Network performance testing

7.4.1 Introduction to TCP/UDP performance testing

Netrounds 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 Netrounds 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, Netrounds supports both point-to-point and hub-and-spoke setups. In a hub-and-spoke setup, all selected Netrounds “clients” will exchange traffic with the Netrounds Test Agent selected as “server”. For UDP, TCP and VoIP UDP, Netrounds also supports the full-mesh setup, where the system automatically generates connections between all selected Netrounds 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 180)
- Multisession TCP (page 182)
- UDP (page 184)
- VoIP UDP (page 187)
- Multicast UDP (page 190)
- TCP throughput test according to RFC 6349 (page 193)
- QoS policy profiling (page 197)
7.4.2 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. Netrounds uses a standard Cubic TCP implementation; for more information, see the page on TCP implementation (page 373) in Netrounds.

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 flow between them. This TCP stream 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.

7.4.2.1 Prerequisites

To run TCP measurements you need to have at least two Netrounds Test Agents installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

Then add a TCP task to your test or monitor and fill in the mandatory parameters below:

7.4.2.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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 flows. 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): Threshold data rates (goodput) for TCP. If the bit rate during one second is lower than this rate, 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 expected rate (goodput; Mbit/s). If no value is given, the functionality will not be used. Min: 0 Mbit/s. Default: By default this field is left blank.

TCP retransmissions cannot cause errored seconds directly, since Netrounds 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 362). The available choices are listed in the drop-down box. Default: “0 / IPP 0”. Note: Test Agents Lite currently do not support DSCP settings in outgoing IP packets.

- Up/Down VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See this page (page 369). Range: 0 … 7. Default: 0. Note: Test Agents Lite and Test Agent Applications currently do not support PCP settings in outgoing IP packets.
• 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.

### 7.4.2.3 Result metrics

- **Rate (Mbit/s):** TCP data rate (goodput).
- **ES (%):** Aggregated errored second (ES) percentage, taking into account all types of error.
- **SLA:** *Service level agreement* (page 372) fulfillment: equal to \((100 - ES)\)%.

### 7.4.3 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.

Netrounds uses a standard Cubic TCP implementation. For more information, see the page on *TCP implementation* (page 373) in Netrounds.
When a Multisession TCP task starts, the Test Agent will start the selected number of TCP flows. 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.

### 7.4.3.1 Prerequisites

To run multisession TCP measurements you need to have two Netrounds Test Agents installed. If you haven’t already done the installation, consult the installation guides found [here](page 64).

Then add a Multisession TCP task to your test or monitor and fill in the mandatory parameters below:

### 7.4.3.2 Parameters

See the [common parameters page](page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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.

7.4.3.3 Result metrics

• **Rate (Mbit/s):** Total data rate (goodput) measured for all TCP sessions combined.
• **Connected:** Number of connected TCP flows.
• **Active:** Number of active TCP flows.
• **Disconnects:** Number of TCP flow 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 372) fulfillment: equal to (100 – **ES total**) %.

7.4.4 UDP

The pictures show hub-and spoke (left) or full-mesh (right) UDP traffic generation between two or more Netrounds 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 Netrounds 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.

### 7.4.4.1 Prerequisites

To run UDP measurements you need to have at least two Netrounds Test Agents installed. If you haven’t already done the installation, consult the installation guides found [here](#) (page 64).

Then add a UDP task to your test or monitor and fill in the mandatory parameters below:

### 7.4.4.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the **test step** (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for **monitors**: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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 362) 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 362) 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). Note: Test Agent Lite currently does not support verifying received DSCP values.

**Thresholds for severely errored seconds (SES)**

- **Up/Down loss (%)**: Packet loss threshold for triggering a severely errored second (page 372). 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**

- **Up/Down DSCP/IPP**: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 362). The available choices are listed in the drop-down box. Default: “0 / IPP 0”. Note: Test Agent Lite currently does not support DSCP settings in outgoing IP packets.
• **Up/Down VLAN priority (PCP):** The Priority Code Point to be used in the VLAN header. See this page (page 369). Range: 0 … 7. Default: 0. Note: Test Agents Lite and Test Agent Applications currently do not support PCP settings in outgoing IP packets.

• **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. Netrounds 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. Netrounds 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.

### 7.4.4.3 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 362).
- **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 377) percentage.
- **SLA:** *Service level agreement* (page 372) fulfillment: equal to (100 – ES) %.

### 7.4.5 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 rate and packet loss, 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.

### 7.4.5.1 Prerequisites

To run VoIP UDP measurements you need to have at least two Netrounds Test Agents installed. If you haven’t already done the installation, consult the installation guides found [here](page 64).

Then add a VoIP UDP task to your test or monitor and fill in the mandatory parameters below:

### 7.4.5.2 Parameters

See the [common parameters page](page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for monitors: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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 streams may lead to fewer misorderings (misordered packets are counted as lost). For example, suppose we are using a single stream and packets arrive in the order 0, 2, 1, 3, 4. This sequence contains one misordering. Now suppose that we instead use two streams with packets 0, 1, 4 arriving in one stream 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


**Thresholds for errored seconds (ES)**


• 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). Note: Test Agent Lite currently does not support verifying received DSCP values.

**Advanced**

• Up/Down DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See [this page](#) (page 362). The available choices are listed in the drop-down box. Default: “0 / IPP 0”. Note: Test Agent Lite currently does not support DSCP settings in outgoing IP packets.

• Up/Down VLAN priority (PCP): The Priority Code Point to be used in the VLAN header. See [this page](#) (page 369). Range: 0 … 7. Default: 0. Note: Test Agents Lite and Test Agent Applications currently do not support PCP settings in outgoing IP packets.

**7.4.5.3 Result metrics**

• Rate (Mbit/s): Ethernet rate of the VoIP UDP flow.

• Loss (%): Packet loss in percent.
7.4.6 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 Netrounds 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.

### 7.4.6.1 Prerequisites

To run Multicast UDP measurements you need to have at least two Netrounds Test Agents installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

Then add a Multicast UDP task to your test or monitor and fill in the mandatory parameters below:

### 7.4.6.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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.

#### 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 362) 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 362) 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). Note: Test Agent Lite currently does not support verifying received DSCP values.
Thresholds for severely errored seconds (SES)

- **Loss (%):** Packet loss threshold for triggering a *severely errored second* (page 372). 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

- **DSCP/IPP:** The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See *this page* (page 362). The available choices are listed in the drop-down box. Default: “0 / IPP 0”. Note: Test Agent Lite currently does not support DSCP settings in outgoing IP packets.
- **VLAN priority (PCP):** The Priority Code Point to be used in the VLAN header. See *this page* (page 369). Range: 0 … 7. Default: 0. Note: Test Agents Lite and Test Agent Applications currently do not support PCP settings in outgoing IP packets.

7.4.6.3 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 362).
- **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 377) percentage.
- **SLA:** *Service level agreement* (page 372) fulfillment: equal to (100 – ES) %.
7.4.7 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 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 flows; 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: \((\text{RTT} - \text{RTT}_\text{baseline}) / \text{RTT}_\text{baseline} \times 100\). 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.

---

### 7.4.7.1 Prerequisites

To run a TCP throughput test you need to have two Netrounds Test Agents installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

Then create a new TCP test and fill in the mandatory parameters below:

### 7.4.7.2 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 \( \ldots \) 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 362) 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 369) 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 flows 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.

### 7.4.7.3 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-to-client MTU</td>
<td>8950</td>
</tr>
<tr>
<td>TCP implementation</td>
<td>Linux TCP stack as in RFC 753, 1122, 2001 with CUBIC and SACK extensions</td>
</tr>
<tr>
<td>Number of TCP streams</td>
<td>1</td>
</tr>
<tr>
<td>Socket buffer size</td>
<td>4.0 KIB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baseline RTT (ms)</th>
<th>RTT under load (ms)</th>
<th>Buffer delay percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.31</td>
<td>0.22</td>
<td>-29.35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Direction</th>
<th>Throughput (Mbits)</th>
<th>TCP transfer time ratio</th>
<th>TCP efficiency percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTA1:eth0 (IPv4) -&gt; VTA2:eth0 (IPv4)</td>
<td>156.92</td>
<td>0.01</td>
<td>99.98</td>
</tr>
</tbody>
</table>

2019-06-11 10:13:14: Per-socket buffer limit on Test Agents: 9765.0 KIB
2019-06-11 10:13:15: Getting path MTU for direction VTA1:eth0 (IPv4) -> VTA2:eth0 (IPv4)
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 MB
7.4.8 QoS policy profiling

This task runs TCP and UDP sessions 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 flows 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 flows 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 flows 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.

7.4.8.1 Prerequisites

To run QoS policy profiling, you need to have two Netrounds Test Agents installed. If you haven’t already
done the installation, consult the installation guides found here (page 64).
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:

7.4.8.2 Parameters

All deviations are calculated as \( \frac{\text{measured} - \text{expected}}{\text{expected}} \times 100\% \).

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 flows 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 prob-
  lems 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 362) used for the TCP or UDP flow. Range: 0 ... 63. Default: 0.
- VLAN priority (PCP): Priority Code Point (page 369) 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 364) 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 362) allowed for the UDP flow. Range: 0 ... 1000 ms. No default.
7 TASK TYPES

7.4.8.3 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.

7.4.8.4 Result metrics – UDP delay

- **Loss (%):** The loss in the UDP flow before the TCP flows 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 flows are started.
- **Delay deviation (%)**: The delay for the UDP flow under TCP load.
- **Jitter (ms):** The jitter for the UDP flow before the TCP flows 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.

<table>
<thead>
<tr>
<th>Class</th>
<th>DSCP/PCP</th>
<th>Expected rate (Mbps)</th>
<th>Measured rate (Mbps)</th>
<th>Expected rate (%)</th>
<th>Measured rate (%)</th>
<th>Rate deviation (%)</th>
<th>Allowed rate deviation (%)</th>
<th>Allowed rate deviation (Mbps)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3</td>
<td>20</td>
<td>5.64</td>
<td>5.14</td>
<td>60.0</td>
<td>60.0</td>
<td>1.71</td>
<td>5.0</td>
<td>5.36 - 5.92</td>
<td>PASS</td>
</tr>
<tr>
<td>B2</td>
<td>18</td>
<td>2.92</td>
<td>2.67</td>
<td>30.0</td>
<td>30.0</td>
<td>1.71</td>
<td>5.0</td>
<td>2.98 - 3.35</td>
<td>PASS</td>
</tr>
<tr>
<td>B1</td>
<td>10</td>
<td>0.94</td>
<td>0.96</td>
<td>10.0</td>
<td>10.01</td>
<td>1.78</td>
<td>5.0</td>
<td>0.89 - 0.99</td>
<td>PASS</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.71</td>
<td></td>
<td>10.0</td>
<td></td>
</tr>
</tbody>
</table>

The second screenshot shows TCP bandwidth results for two QoS classes (“BC”, “BE”) and UDP delay results for a real-time QoS class (“RT”).

<table>
<thead>
<tr>
<th>Class</th>
<th>DSCP</th>
<th>Loss (%)</th>
<th>Loss under load (%)</th>
<th>Delay (ms)</th>
<th>Delay under load (ms)</th>
<th>Delay deviation (%)</th>
<th>Allowed delay deviation (%)</th>
<th>Allowed delay deviation (ms)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>460</td>
<td>0.0</td>
<td>0.0</td>
<td>0.20</td>
<td>22.26</td>
<td>7.2%</td>
<td>10.0</td>
<td>0.26</td>
<td>PASS</td>
</tr>
</tbody>
</table>

Both measurement types have pass/fail criteria defined according to the above description.

7.5 IPTV and OTT video testing

7.5.1 Introduction to IPTV/MPEG testing

Netrounds 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 Netrounds on the following pages:

- **IPTV MPEG** (page 202)
- **IPTV MPEG inline** (page 205)
- **IPTV channel zapping time** (page 207)
- **IGMP channel join/leave** (page 215)
- **DVB-C MPEG** (page 210)
- **Multicast group limit** (page 217)

---

### 7.5.2 IPTV MPEG

Netrounds 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. Netrounds will measure and highlight MPEG loss, PCR jitter, rate, packet loss, continuity count (CC) errors, and any general problems with the stream.
When an IPTV MPEG task starts, the Netrounds Test Agents will join the channels by sending IGMP join messages. Once they receive the streams, the Test Agents will continuously measure quality.

**Note:** Netrounds 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 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.

### 7.5.2.1 Prerequisites

To run IPTV/MPEG measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found [here](page 64).

Also make sure that you have prepared Netrounds 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:

### 7.5.2.2 Parameters

See the [common parameters page](page 176) for the following:

- Parameters that are set on the **test step** (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for **monitors**: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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).

**Thresholds for errored seconds (ES)**

- **MPEG loss (CC errors/s)**: Maximum tolerated MPEG packet loss (CC errors) per second. See [this page](page 365). Min: 0 packets per second. Default: 2 packets per second.
- **Jitter**: Maximum tolerated PCR and RTP jitter (delay variation) in the received streams. See [this page](page 362). 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.
7.5.2.3 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 366) 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 stream contains RTP headers, Netrounds will calculate RTP jitter, loss, and misorderings, which are basically the same as the corresponding metrics for IP. Whether or not the 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 372) fulfillment: equal to (100 – ES total) %. 
7.5.3 IPTV MPEG inline

The IPTV MPEG inline task lets you monitor the quality of TV channels that customers are watching. When this task starts, Netrounds 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; Netrounds will also alert you about any general problems with the stream.

**Note:** Netrounds 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 Netrounds’ channel list, no measurement data will be obtained. Note also that Netrounds 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 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).

### 7.5.3.1 Prerequisites

To do IPTV inline measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found [here](page 64).

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 Netrounds 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:
7.5.3.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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 86).

Thresholds for errored seconds (ES)

- MPEG loss (CC errors/s): Maximum tolerated MPEG packet loss (CC errors) per second. See this page (page 365). Min: 0 packets per second. Default: 2 packets per second.
- Jitter: Maximum tolerated PCR and RTP jitter (delay variation) in the received streams. See this page (page 362). 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.

7.5.3.3 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 366) for further details.
- MPEG loss: MPEG packet loss, calculated from the Continuity_count_error counter in the MPEG stream. See this page (page 365).
- 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 stream contains RTP headers, Netrounds will calculate RTP jitter, loss, and misorderings, which are basically the same as the corresponding metrics for IP. Whether or not the 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 372) fulfillment: equal to $(100 – \text{ES total}) \%$.

7.5.4 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 Netrounds 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).

### 7.5.4.1 Prerequisites

To run IPTV channel zapping measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found [here](#) (page 64).

Also make sure that you have prepared Netrounds 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 202) 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.

### 7.5.4.2 Parameters

See the [common parameters page](#) (page 176) for the following:

- Parameters that are set on the [test step](#) (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for monitors: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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.

### 7.5.4.3 Result metrics

- **Average join delay (ms):** Average delay from sending an IGMP join until the stream arrived.
- **Minimum join delay (ms):** Minimum delay from sending an IGMP join until the stream arrived.
- **Maximum join delay (ms):** Maximum delay from sending an IGMP join until the stream arrived.
- **Failed joins:** Number of failed IGMP joins. A join fails if the stream does not arrive before an IGMP leave is sent.
- **Average leave delay (ms):** Average delay from sending an IGMP leave until the stream was stopped.
- **Minimum leave delay (ms):** Minimum delay from sending an IGMP leave until the stream was stopped.
- **Maximum leave delay (ms):** Maximum delay from sending an IGMP leave until the stream was stopped.
- **Failed leaves:** Number of failed IGMP leaves. A leave fails if the stream 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 372) fulfillment: equal to \((100 – \text{ES total})\) \%.

---

© 2020 Netrounds. All rights reserved.
7.5.5 DVB-C MPEG

Netrounds Test Agents can receive one or several DVB-C channels, measuring MPEG TS parameters in compliance with ETSI TR 101 290 (Measurement guidelines for DVB systems). When a DVB-C task starts, the Test Agents will tune to the frequency of the selected channels, then start monitoring them continuously. By using the DVB-C testing feature, you will quickly get a view of DVB-C 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. Netrounds will measure and highlight MPEG loss, PCR jitter, rate, and any general problems with the stream.

Note: Netrounds does not decrypt any of the MPEG streams; its quality measurements are based on the unencrypted MPEG headers only.

7.5.5.1 Prerequisites

First decide upon your network topology, then follow the steps below.

To run DVB-C/MPEG measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

To be able to monitor DVB-C channels, a Test Agent must be equipped with a special DVB-C USB device (dongle). Install this dongle if you haven’t already.

Also make sure that you have prepared Netrounds with your DVB-C channel list (page 22).

Then add a DVB-C MPEG task to your test or monitor and fill in the mandatory parameters below:
7.5.5.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) common to all test tasks: Delayed start.

General

- Receivers: Specify the DVB-C USB dongles you want to use as receivers. The devices are identified as “<Test Agent name>:<dongle serial number>”.
- Channels: Select the DVB-C channels that should be monitored from the preset channel list. Only channels that have the same frequency, symbol rate, FEC, and modulation can be selected at the same time.

Thresholds for errored seconds (ES)

- MPEG loss (CC errors/s): Maximum tolerated MPEG packet loss (CC errors) per second. See this page (page 365). Min: 0 packets per second. Default: 2 packets per second.
- Jitter: Maximum tolerated PCR and RTP jitter (delay variation) in the received streams. See this page (page 362). Min: 0 ms. Default: 50 ms.

7.5.5.3 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 366) 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 stream contains RTP headers, Netrounds will calculate RTP jitter, loss, and misordering, which are basically the same as the corresponding metrics for IP. Whether or not the stream contains RTP headers depends on the encoder at the head-end.
- PAT errors: A Program Allocation Table (PAT) should be received every half-second on a multicast group. If a PAT is not received within that interval, a PAT error is triggered.
- PMT errors: A Program Map Table (PMT) should be received every half-second on a program. If a PMT is not received within that interval, a PMT error is triggered.
- PID errors: On regular audio/video streams, a frame should be received every 5 seconds. If no frame is received within 5 seconds, one PID error is generated for each further second that elapses from that point.
• **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 372) fulfillment: equal to \((100 – \text{ES total})\) %.

---

### 7.5.6 OTT testing: HTTP Live Streaming (HLS)

A Netrounds 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.

7.5.6.1 Prerequisites

To run HLS measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

In you test or monitor, add an OTT - HLS task and fill in the mandatory parameters below:

7.5.6.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) common to all test tasks: Delayed start.

General

- Clients: Test Agent interfaces to use as clients.
- URL: URL of the 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 streams which you expect never to end, and for which you want to trigger an alarm if that happens. Default: Yes.

7.5.6.3 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 372) fulfillment: equal to (100 – ES total) %.

7.5.6.4 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).

![Graph 1](image1.jpg)

The second graph shows the actual playback rate (green) compared to the playback rate currently selected by the client (blue).

![Graph 2](image2.jpg)
7.5.7 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.

7.5.7.1 Reference

The test performed conforms to SEC (page 371) Access Certification ID “SEC-CM-MCAST-2”.

© 2020 Netrounds. All rights reserved.
7.5.7.2 Impact

DoS

7.5.7.3 Test procedure

Customer joins the multicast channels, listens for packets for 10 seconds, then leaves.

7.5.7.4 Fail criteria

- No data is received on one of the allowed multicast channels.
- Data is received on one of the disallowed multicast channels.

7.5.7.5 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
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.

7.5.8.1 Reference

The test performed conforms to SEC (page 371) Access Certification ID “SEC-V4-MCAST-1”.

7.5.8.2 Impact

DoS

7.5.8.3 Test procedure

1. Customer joins the allowed number of multicast channels.
2. Customer tries to join one additional channel.
7.5.8.4 Fail criteria

• In step 1, some channel is not received.
• In step 2, the total number of channels received exceeds the maximum number.

7.5.8.5 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.

7.6 HTTP and DNS testing

7.6.1 Introduction to HTTP and DNS testing

Netrounds 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 Netrounds HTTP and DNS testing on the following pages:

• HTTP (page 219)
• DNS (page 222)
7.6.2 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 Netrounds 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.

7.6.2.1 Prerequisites

To run HTTP measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found here (page 64).
In your test or monitor, add an HTTP task and fill in the mandatory parameters below:

### 7.6.2.2 Parameters

See the **common parameters page** (page 176) for the following:

- Parameters that are set on the *test step* (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for *monitors*: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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**: Here you can enter a (case-insensitive) regular expression against which the HTTP response content will be matched. 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 and port will be used as HTTP proxy.
- **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 Netrounds to measure too low connect time.
- **Proxy username**: User name used for HTTP proxy authentication.
• Proxy password: Password used for HTTP proxy authentication.

7.6.2.3 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 (KB):** 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 372) fulfillment: equal to (100 – ES total) %.
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.

### 7.6.3.1 Prerequisites

To run DNS measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found [here](page 64).

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:
7.6.3.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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 Netrounds are as follows:
  - A (IPv4 address; default)
  - AAAA (IPv6 address)
  - CNAME (canonical name)
  - MX (email)
  - PTR (pointer)
  - NS (name server)
  - SOA (start of authority)
  - SRV (service)
  - 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 will be indicated. 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. 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.

© 2020 Netrounds. All rights reserved.
• 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.

### 7.6.3.3 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.
- **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 372) fulfillment: equal to (100 – ES total) %.

### 7.7 SIP testing

#### 7.7.1 Introduction to SIP testing

Netrounds Test Agents 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 Netrounds SIP testing on [this page](page 225).
This task measures response times in SIP signaling and the voice quality of RTP media streams.

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.
7 TASK TYPES

7.7.2.1 Prerequisites

To run SIP measurements, you need to have at least two Netrounds Test Agents installed. If you haven't already done the installation, consult this page: Getting started with IP telephony (SIP and VoIP) measurements (page 18).

Make sure that you have prepared the Netrounds account with SIP accounts. Read more on this topic on the page Setting up SIP accounts (page 24).

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.

7.7.2.2 Parameters

See the common parameters page (page 176) for the following:

• Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
• SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
• Advanced settings (page 177) 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 365). 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 media streams. See this page (page 362). 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 streams. One of: GSM Full Rate, G.711 A-law (PCMA), or G.711 µ-law (PCMU). Default: GSM Full Rate (“GSM”).

7.7.2.3 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 362).
• **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 372) fulfillment: equal to \((100 – \text{ES total})\)%.

### 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) stream being received for 4 seconds.

### 7.7.3 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.

#### 7.7.3.1 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).

#### 7.7.3.2 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.

### 7.8 Wi-Fi network testing

#### 7.8.1 Introduction to Wi-Fi network testing

Netrounds 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 Netrounds, a specific hardware model with an mPCle Wi-Fi card from Intel is required (“HW Medium Wi-Fi”; read more in the [Test Agent datasheet](#)). Such Wi-Fi-capable preinstalled Test Agents ship with the *iwlwifi* driver.

If you are using x86 hardware not provided by Netrounds, 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 Netrounds 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

For further information on how to purchase Test Agents for Wi-Fi testing, please contact Netrounds sales at sales@netrounds.com.

### 7.8.1.1 Disclaimer

Wi-Fi result metrics are affected by many factors beyond Netrounds' control, 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.

### 7.8.1.2 Related topics

- Use the **Wi-Fi interface configuration dialog** (page 91) to configure the Wi-Fi card.
- Use the **Wi-Fi logger** (page 230) task to collect Wi-Fi network measurements.
- Use the **Wi-Fi switcher** (page 231) task to change Wi-Fi interface parameters.

### 7.8.2 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 91).

#### 7.8.2.1 Parameters

- Interface: Test Agent Wi-Fi interface to use for scanning.

#### 7.8.2.2 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](#).

---

### 7.8.3 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 228).

You can change the Wi-Fi interface and its settings dynamically while running a test using the **Wi-Fi switcher** (page 231) task.

For normal configuration, use the **Wi-Fi interface configuration settings** (page 91).

#### 7.8.3.1 Parameters

See the [common parameters page](#) (page 176) for the following:

- Parameters that are set on the **test step** (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for **monitors**: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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, Netrounds 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.

#### 7.8.3.2 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.

---

### 7.8.4 Wi-Fi switcher

This task is a utility which lets you change Wi-Fi interface parameters while running a [Wi-Fi logger](page 230) task. When this task executes, it modifies the existing configuration of the Wi-Fi interface (described on this page (page 91)).

**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 Netrounds 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.

### 7.8.4.1 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 91).

7.8.4.2 Result metrics

- Outcome of configuration switch: Success or failure
- Log detailing the configuration switch process

7.9 Mobile network testing

7.9.1 Introduction to mobile network testing (including configuration)

Certain Netrounds Test Agents (currently, Test Agents preinstalled on HW Medium Mobile hardware; read more in the Test Agent datasheet) can measure network performance and user experience in mobile networks by means of a built-in mobile network interface.

This function is currently limited to European frequency bands and to the LTE (4G), WCDMA (3G), and GSM (2G) 3GPP standards.

For further information, please contact Netrounds sales at sales@netrounds.com.

7.9.1.1 Related topics

- Use the Mobile interface configuration dialog (page 87) to configure the 4G modem.
- Use the Mobile logger (page 232) task to collect mobile network measurements.
- Use the Mobile switcher (page 234) task to change mobile interface parameters.

7.9.2 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 Netrounds with this extra hardware support.

You can change the mobile interface and its settings (APN, RAT, band) using the Mobile switcher (page 234) utility.
7.9.2.1 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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, Netrounds 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.

7.9.2.2 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.

7.9.3 Mobile switcher

This task is a utility which lets you change mobile interface parameters while running a Mobile logger (page 232) task.

7.9.3.1 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).

7.10 Ethernet service activation testing

7.10.1 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.
7.10.1.1 Compliance with ITU-T Y.1564

Netrounds 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 Netrounds 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 streams. The picture below, taken from the Y.1564 specification, illustrates the relationship between CIR, EIR, and the color coding of traffic.

![Bandwidth vs. Time Graph](image)

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 179) tests. These send and receive multiple TCP and UDP data streams 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.

7.10.1.2 Compliance with MEF Carrier Ethernet 2.0

Netrounds is also compliant with Metro Ethernet Forum Carrier Ethernet (CE) 2.0 services certification (E-Line, E-LAN, E-Tree, and E-Access).
7.10.1.3 Physical test setup

For Ethernet service activation tests, you need to install and register at least two Netrounds 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 Netrounds 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.

7.10.1.4 Further reading

For details on individual tasks handling Ethernet service activation tests, see the following pages:

- Simple CIR validation test (page 237)
- Step load CIR test (page 239)
- EIR configuration test (color-aware) (page 241)
- EIR configuration test (non-color-aware) (page 243)
- Traffic policing test (color-aware) (page 245)
- Traffic policing test (non-color-aware) (page 247)
7.10.2 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 stream the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

7.10.2.1 Reference

The test performed conforms to ITU-T Y.1564 section 8.1.2, test A.1 (“CIR configuration test: Simple CIR validation”).

7.10.2.2 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 stream.

7.10.2.3 Fail criteria

- The test fails if any of the thresholds Max frame loss, Max frame delay, or Max frame jitter is exceeded.

7.10.2.4 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 362) for conforming packets. Min: 0 ms. Max: 1000 ms. No default.

Traffic profile


• 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.


7.10.2.5 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.
7.10.3 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 stream 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).

7.10.3.1 Reference

The test performed conforms to ITU-T Y.1564 section 8.1.2, test A.2 (“CIR configuration test: Step load CIR test”).

7.10.3.2 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 stream. 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.

7.10.3.3 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.
7.10.3.4 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 362) for conforming packets. Min: 0 ms. Max: 1000 ms. No default.

Traffic profile

- **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.

7.10.3.5 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.
7.10.4 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 stream (two QoS classes) are present. In accordance with ITU-T Y.1564, the green stream is checked against CIR and the yellow stream against EIR. For both streams the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

7.10.4.1 Reference

The test performed conforms to ITU-T Y.1564 section 8.1.2, test B.1 (“EIR configuration test, colour aware”).

7.10.4.2 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 streams (green and yellow).

7.10.4.3 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 stream.
7.10.4.4 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**

- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.

**Traffic profile for CIR (green) stream**


**Traffic profile for EIR (yellow) stream**

7.10.4.5 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.

7.10.5 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 stream. For the generated stream the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

7.10.5.1 Reference

The test performed conforms to ITU-T Y.1564 section 8.1.2, test B.2 (“EIR configuration test, non-colour aware”).

7.10.5.2 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 stream.

7.10.5.3 Fail criteria

• The test fails if the measured rate is less than $CIR \times (1 – \text{Max frame loss / 100})$.

7.10.5.4 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

• 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.

7.10.5.5 Result metrics

• Rate (Mbit/s): Ethernet data rate.
• Loss (%): Packet loss.
• Delay (ms): Average one-way delay.
• Jitter (ms): Jitter (delay variation) (page 362).
• Pass/fail outcome of test.
7.10.6 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 stream (two QoS classes) are present. In accordance with ITU-T Y.1564, the green stream is checked against CIR and the yellow stream against EIR. For both streams the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

7.10.6.1 Reference

The test performed conforms to ITU-T Y.1564 section 8.1.2, test C.1 (“Traffic policing test: Colour aware”).

7.10.6.2 Test procedure

- The sender Test Agent generates green frames at a rate equal to CIR, and yellow frames at a rate of $1.25 \times EIR$. (However, if EIR is less than 20% of CIR, the yellow frames are instead generated at a rate of $EIR + 0.25 \times CIR$.) 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 streams (green and yellow).
7.10.6.3 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 stream, or the total received rate is higher than $1.05 \times (\text{CIR} + \text{EIR})$.

7.10.6.4 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

- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0%. Max: 100%. No default.

Traffic profile for CIR (green) stream


Traffic profile for EIR (yellow) stream

- DSCP: Differentiated Services Code Point. Range: 0 … 63. Default: 0.
7. TASK TYPES


7.10.6.5 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.

7.10.7 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 stream and one QoS class. For the generated stream the following parameters need to be specified: frame size, source and destination UDP port, DSCP value, and VLAN priority (p-bits).

7.10.7.1 Reference

7.10.7.2 Test procedure

- The sender Test Agent generates frames at a rate of \( CIR + 1.25 \times EIR \). (However, if \( EIR \) is less than 20\% of \( CIR \), the frames are instead generated at a rate of \( 1.25 \times CIR + EIR \).)
- The receiver Test Agent measures received rate, loss, delay, and jitter on the stream.

7.10.7.3 Fail criteria

- The test fails if the measured rate is higher than \( CIR + 1.05 \times EIR \) or less than \( CIR \times (1 – \text{Max frame loss}) \).

7.10.7.4 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**

- Max frame loss (%): Maximum frame loss ratio for conforming packets. Min: 0\%. Max: 100\%. No default.

**Traffic profile**


7.10.7.5 Result metrics

- **Rate** (Mbit/s): Ethernet data rate.
- **Loss** (%): Packet loss.
• **Delay (ms):** Average one-way delay.
• **Jitter (ms):** *Jitter (delay variation)* (page 362).
• **Pass/fail** outcome of test.

### 7.11 Transparency testing

#### 7.11.1 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.

#### 7.11.1.1 Physical test setup

To perform transparency tests, two Netrounds 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 Netrounds 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.
7.11.2 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 0x8809 defines different protocols depending on subtype.

7.11.2.1 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.

7.11.2.2 Fail criteria

The test fails if some frame is treated differently from the expected outcome.

7.11.2.3 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 250).
7.11.2.4 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.
- Ethertype: Ethertype in decimal format (e.g. 65534) or in hexadecimal format (e.g. 0x1122). Ethertype 0x8100 is not supported, nor are Ethertypes below 0x0600.
- Subtype: (Optional) Subtype for Slow Protocols, a comma-separated list of subtype ranges in decimal format or hexadecimal format (e.g. 0x01-0x03). Subtype 0x01 = LACP, subtype 0x02 = LAMP, subtype 0x03 = Link OAM, subtype 0x04 = ESMC.
- 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.

7.11.2.5 Result metrics

- Pass/fail for the tested custom Ethertype per destination MAC address
7.11.3 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 369) are not modified by the network.

7.11.3.1 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.

7.11.3.2 Fail criteria

The test fails if not all frames are received, or if the VLAN ID or PCP has been modified.

7.11.3.3 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 250).

7.11.3.4 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.


• 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.

7.11.3.5 Result metrics

• **Pass/fail** for the tested custom VLAN
7.11.4 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

7.11.4.1 Test procedure

For each protocol to be tested, five frames are generated and sent.

7.11.4.2 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.

7.11.4.3 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 250).
7.11.4.4 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


7.11.4.5 Result metrics

- **Pass/fail** for each tested protocol
7.11.5 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.

7.11.5.1 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.

7.11.5.2 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.

7.11.5.3 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 250).

7.11.5.4 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.

7.11.5.5 Result metrics

• Pass/fail for selected Ethertypes and LLC/SNAP protocols

7.11.5.6 Reference section

Ethertypes

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Ethertype</th>
</tr>
</thead>
<tbody>
<tr>
<td>3COM Loop Detect</td>
<td>0x9003</td>
</tr>
<tr>
<td>3COM TCP-IP Sys</td>
<td>0x9002</td>
</tr>
<tr>
<td>3COM XNS Sys Mgmt</td>
<td>0x9001</td>
</tr>
<tr>
<td>AppleTalk</td>
<td>0x809b</td>
</tr>
<tr>
<td>AppleTalk ARP</td>
<td>0x80f3</td>
</tr>
<tr>
<td>CESoE</td>
<td>0x88d8</td>
</tr>
<tr>
<td>DEC LANBridge</td>
<td>0x8038</td>
</tr>
<tr>
<td>Frame Relay ARP</td>
<td>0x0808</td>
</tr>
<tr>
<td>GSMP</td>
<td>0x880c</td>
</tr>
<tr>
<td>IBM SNA Service</td>
<td>0x80d5</td>
</tr>
<tr>
<td>MPLS</td>
<td>0x8847</td>
</tr>
<tr>
<td>Novell 8137</td>
<td>0x8137</td>
</tr>
<tr>
<td>Novell 8138</td>
<td>0x8138</td>
</tr>
<tr>
<td>PPP</td>
<td>0x880b</td>
</tr>
<tr>
<td>PPPoE Discovery Stage</td>
<td>0x8863</td>
</tr>
<tr>
<td>PPPoE Session Stage</td>
<td>0x8864</td>
</tr>
<tr>
<td>Raw Frame Relay</td>
<td>0x6559</td>
</tr>
<tr>
<td>SNMP</td>
<td>0x814c</td>
</tr>
</tbody>
</table>

LLC/SNAP protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Destination MAC</th>
<th>SAP</th>
<th>CTRL</th>
<th>OUI</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDP</td>
<td>01:00:0c:cc:cc:cc</td>
<td>0xaa</td>
<td>0x03</td>
<td>0x0000c</td>
<td>0x2000</td>
</tr>
<tr>
<td>Cisco VTP</td>
<td>01:00:0c:cc:cc:cc</td>
<td>0xaa</td>
<td>0x03</td>
<td>0x0000c</td>
<td>0x2003</td>
</tr>
<tr>
<td>Cisco DTP</td>
<td>01:00:0c:dd:dd:dd</td>
<td>0xaa</td>
<td>0x03</td>
<td>0x0000c</td>
<td>0x2004</td>
</tr>
<tr>
<td>CGMP</td>
<td>01:00:0c:dd:dd:dd</td>
<td>0xaa</td>
<td>0x03</td>
<td>0x0000c</td>
<td>0x2001</td>
</tr>
</tbody>
</table>
7.11.6 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.

7.11.6.1 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.
<table>
<thead>
<tr>
<th>Multicast address range</th>
<th>No. of addresses in range</th>
</tr>
</thead>
<tbody>
<tr>
<td>224.0.0.1 ... 224.0.0.63</td>
<td>63</td>
</tr>
<tr>
<td>224.10.10.0 ... 224.10.10.63</td>
<td>64</td>
</tr>
<tr>
<td>228.0.128.0 ... 228.0.128.63</td>
<td>64</td>
</tr>
<tr>
<td>230.70.80.64 ... 230.70.80.127</td>
<td>64</td>
</tr>
<tr>
<td>239.240.250.0 ... 239.240.250.63</td>
<td>64</td>
</tr>
</tbody>
</table>

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.

### 7.11.6.2 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.

### 7.11.6.3 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 250).

### 7.11.6.4 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.

### 7.11.6.5 Result metrics

- **Pass/fail** on IP header integrity and IP multicast
7.11.7 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.

7.11.7.1 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:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast Listener Discovery Protocol, version 1 (MLD)</td>
<td>MLD query and report messages</td>
</tr>
<tr>
<td>Multicast Listener Discovery Protocol, version 2 (MLDv2)</td>
<td>MLD query and report messages</td>
</tr>
<tr>
<td>Neighbor Solicitation</td>
<td>Neighbor Solicitation messages</td>
</tr>
<tr>
<td>Neighbor Advertisement</td>
<td>Neighbor Advertisement messages</td>
</tr>
<tr>
<td>Router Solicitation</td>
<td>Router Solicitation messages</td>
</tr>
<tr>
<td>Router Advertisement</td>
<td>Router Advertisement messages</td>
</tr>
<tr>
<td>ICMPv6 Echo</td>
<td>ICMPv6 Echo messages</td>
</tr>
<tr>
<td>DHCPv6 Solicit</td>
<td>DHCPv6 Solicit messages</td>
</tr>
</tbody>
</table>

The test fails for a particular protocol if the message sent on that protocol does not pass transparently.

### 7.11.7.2 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 250).

### 7.11.7.3 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.

### 7.11.7.4 Result metrics

- **Pass/fail** on IPv6 header integrity, overall and for each IPv6 protocol.
7.11.8 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.

7.11.8.1 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.

7.11.8.2 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 250).
7.11.8.3 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.

7.11.8.4 Result metrics

- **Pass/fail** on minimum number of MAC addresses
- **Pass/fail** on maximum number of MAC addresses
7.11.9 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.

7.11.9.1 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.

7.11.9.2 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.
### 7.11.9.3 Protocols tested

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Destination MAC</th>
<th>Ethertype</th>
</tr>
</thead>
<tbody>
<tr>
<td>STP</td>
<td>01:80:c2:00:00:00</td>
<td>LLC/SNAP</td>
</tr>
<tr>
<td>ARP</td>
<td>ff:ff:ff:ff:ff:ff</td>
<td>0×0806</td>
</tr>
<tr>
<td>RARP</td>
<td>ff:ff:ff:ff:ff:ff</td>
<td>0×8035</td>
</tr>
<tr>
<td>LLDP</td>
<td>01:80:c2:00:00:0e</td>
<td>0×88cc</td>
</tr>
<tr>
<td>Ethernet Configuration Test Protocol</td>
<td>cf:00:00:00:00:00</td>
<td>0×9000</td>
</tr>
<tr>
<td>IP global broadcast</td>
<td>ff:ff:ff:ff:ff:ff</td>
<td>0×0800</td>
</tr>
<tr>
<td>IP local broadcast</td>
<td>ff:ff:ff:ff:ff:ff</td>
<td>0×0800</td>
</tr>
<tr>
<td>MPLS multicast addresses</td>
<td>01:00:5e:80:00:01</td>
<td>0×8848 (see Note 1)</td>
</tr>
<tr>
<td>MPLS multicast addresses</td>
<td>01:00:5e:8d:dd:dd</td>
<td>0×8848</td>
</tr>
<tr>
<td>MPLS multicast addresses</td>
<td>01:00:5e:8f:ff:01</td>
<td>0×8848</td>
</tr>
</tbody>
</table>

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.

### 7.11.9.4 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 250).

### 7.11.9.5 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.

### 7.11.9.6 Result metrics

- **Pass/fail** on STP
- **Pass/fail** on General multicast/broadcast protocols
• Pass/fail on MPLS multicast

7.11.10 L2 transparency – VLAN

This task verifies VLAN transparency, that is, that packets with user-specified VLAN tag, VLAN priority (PCP) (page 369), and DSCP (page 362) are not modified by the network.

7.11.10.1 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.
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.

<table>
<thead>
<tr>
<th>Outer VLAN ID</th>
<th>Inner VLAN ID</th>
<th>Inner p-bits</th>
<th>DSCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4095</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>4083</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>100</td>
<td>3995</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>151</td>
<td>3944</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>518</td>
<td>3577</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>1034</td>
<td>3061</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>1435</td>
<td>2660</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>1801</td>
<td>2294</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>2048</td>
<td>2047</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>4095</td>
<td>6</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

7.11.10.2 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 250).

7.11.10.3 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.

7.11.10.4 Result metrics

• Pass/fail on VLAN transparency
• Pass/fail on Q-in-Q transparency

---

7.11.11 DSCP remapping

This task verifies the expected remapping of Differentiated Services Code Point (page 362) values between two points in your network.

7.11.11.1 Test procedure

UDP packets are sent from the sender Test Agent to the receiver Test Agent with different DSCP values, and Netrounds checks if they are mapped correctly by the network.

7.11.11.2 Fail criteria

The test fails if any received DSCP differs from the expected DSCP value.
7.11.11.3 Limitations

IPv6 is not supported for this task.

7.11.11.4 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.

7.11.11.5 Result metrics

- Pass/fail for each tested DSCP
7.11.12 Layer 4 destination port DSCP remapping

This task verifies the expected DSCP (Differentiated Services Code Point) (page 362) remapping between two points in your network, with specific UDP or TCP destination ports indicated.

7.11.12.1 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.

7.11.12.2 Fail criteria

The test fails for a port range if the DSCP mapping for any port in that range differs from what is expected.

7.11.12.3 Limitations

IPv6 is not supported for this task.

7.11.12.4 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.

### 7.11.12.5 Result metrics

• **Pass/fail** for each tested port range

### 7.11.12.6 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).

---

### 7.11.13 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.
7.11.13.1 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 79)). 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.

7.11.13.2 Limitations

IPv6 is not supported for this task.

7.11.13.3 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.
- 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.

7.11.13.4 Result metrics

- Calculated path MTU with pass/fail indication

7.12 Reflector-based testing

7.12.1 Introduction to Y.1731 testing

Netrounds 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 Netrounds 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

### 7.12.1.1 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

7.12.1.2 RFC 2544 and Y.1564

In Netrounds you can create tests according to IETF RFC 2544 and ITU-T Recommendation Y.1564, where the Netrounds Test Agent will generate Y.1731 traffic using specified traffic patterns which will be reflected by the Y.1731-capable equipment.

7.12.1.3 Related topics

- Y.1731 Ethernet loopback (ETH-LB) (page 275)
- Y.1731 delay measurement (ETH-DM) (page 278)
- Y.1731 synthetic loss measurement (ETH-SLM) (page 281)

---

7.12.2 Y.1731 Ethernet loopback (ETH-LB)

This task lets you send traffic from a Netrounds 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.

### 7.12.2.1 Prerequisites

To run a Y.1731 ETH-LB measurement you need to have at least one Netrounds Test Agent installed as well as one or several Y.1731-enabled devices in your network. See the installation guides found [here](#) (page 64) 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 Netrounds, as explained on [this page](#) (page 32).

Then add an ETH-LB task to your test or monitor and fill in the mandatory parameters below:

### 7.12.2.2 Parameters

See the [common parameters page](#) (page 176) for the following:

- Parameters that are set on the [test step](#) (page 176) level: Duration, Fail threshold, and Wait for ready.
- [SLA thresholds](#) (page 177) for monitors: SLA Good and SLA Acceptable.
- [Advanced settings](#) (page 177) 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.

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 362) 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 372). 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 369). Default: 0.

7.12.2.3 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 377) percentage.
• **SLA**: *Service level agreement* (page 372) fulfillment: equal to \(100 - ES\) %.

---

**7.12.3 Y.1731 delay measurement (ETH-DM)**

This task lets you send traffic from a Netrounds 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.

### 7.12.3.1 Prerequisites

To run a Y.1731 ETH-DM measurement, you need to have at least one Netrounds Test Agent installed as well as one or several Y.1731-enabled devices in your network. See the installation guides found [here](#) (page 64) 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 Netrounds, as explained on [this page](#) (page 32).

Then add an ETH-DM task to your test or monitor and fill in the mandatory parameters below:

### 7.12.3.2 Parameters

See the [common parameters page](#) (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for monitors: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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.

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 362) 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 372). 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 369). Default: 0.

7.12.3.3 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.
7.12.4 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.

### 7.12.4.1 Prerequisites

To run a Y.1731 ETH-SLM measurement you need to have at least one Netrounds Test Agent installed as well as one or several Y.1731-enabled devices in your network. See the installation guides found [here](page 64) 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 Netrounds, as explained on [this page](page 32).

Then add an ETH-SLM task to your test or monitor and fill in the mandatory parameters below:

### 7.12.4.2 Parameters

See the [common parameters page](page 176) for the following:

- Parameters that are set on the [test step](page 176) level: Duration, Fail threshold, and Wait for ready.
- [SLA thresholds](page 177) for monitors: SLA Good and SLA Acceptable.
- [Advanced settings](page 177) 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.
- **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 372). 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 369). Default: 0.

7.12.4.3 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.
7 TASK TYPES

- **ES loss (%)**: Errored second percentage for packet loss.
- **SES (%)**: Aggregated severely errored second (SES) percentage.
- **Unavailable seconds (%)**: *Unavailable second (UAS)* (page 377) percentage.
- **SLA**: *Service level agreement* (page 372) fulfillment: equal to \((100 - \text{ES})\) %.

7.12.5 Introduction to TWAMP testing

Netrounds 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 Netrounds Test Agent is typically placed in the core part of the network, or in the data center, and initiates UDP streams towards TWAMP-capable routers or other devices. These reflect the streams back to the Netrounds Test Agent, which collects measurements and calculates various network KPIs.

The scenario just described is handled by the task *TWAMP/TWAMP Light* (page 286). 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 Netrounds 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.

### 7.12.5.1 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 286) task. Further, a distinct task type called TWAMP Reflector (page 291) 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 286) 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 286) below.

**External TWAMP senders**

To use third-party devices as TWAMP senders rather than Test Agents, use the *TWAMP Reflector* (page 291) 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 291) 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 286) task.

### 7.12.5.2 Related topics

- *Setting up TWAMP reflectors* (page 29)
- *Setting up TWAMP measurements* (page 286)

---

### 7.12.6 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.
TWAMP Light works with both IPv4 and IPv6. Full TWAMP is currently supported for IPv4 only. Test Agents acting as reflectors support TWAMP Light only.

### 7.12.6.1 Prerequisites

To perform TWAMP measurements, you need to install at least one Netrounds 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 64). The procedure is different depending on what kind of hardware you have available.

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 them in Netrounds. 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.

### 7.12.6.2 Parameters

See the [common parameters page](#) (page 176) for the following:
• Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.

• SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.

• Advanced settings (page 177) 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, list them here with their host names and port numbers.

• 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. 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 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.

• 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 288).

Timestamping methods

All timestamping for TWAMP in Netrounds, 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 Netrounds 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 Netrounds 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) 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 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 372). 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**

- **Frame size (bytes):** Size of Layer 2 Ethernet frame for the stream. See this page (page 364). 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 362). 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 369).
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.

7.12.6.3 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): Actual rate at which the senders sent 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.
• Far-end loss (%): Lost packets 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 (%): Lost packets 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.
• ES (%): Aggregated errored second (ES) percentage, taking into account all types of error.
• ES delay (%): Accumulated errored second percentage for two-way delay.
• ES delay variation (%): Accumulated 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 377) percentage.
• SLA: Service level agreement (page 372) fulfillment: equal to (100 – ES) %.
7.12.7 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.

7.12.7.1 Prerequisites

To run a TWAMP Reflector task, you need to install at least one Netrounds 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 64). 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.

7.12.7.2 Parameters

See the common parameters page (page 176) for the following:
• Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
• SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
• Advanced settings (page 177) 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, it is recommended that you point directly to these reflectors when configuring the task: see this page (page 286).

For special use cases (page 286), you might want to start the Test Agent TWAMP reflectors, add them manually to the TWAMP inventory as detailed on this page (page 29), 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.

**7.12.7.3 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.
7.12.8 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 Netrounds 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.

7.12.8.1 Prerequisites

To run Netrounds Ping measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

In your test or monitor, add a Ping task and fill in the mandatory parameters below:

7.12.8.2 Parameters

See the common parameters page (page 176) for the following:
Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.

SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.

Advanced settings (page 177) 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 Netrounds. How to set up such an inventory is covered here (page 26).

- Hosts: Here you enter hosts to ping as IP addresses or host names.
- Hosts inventory: Here you select 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): Jitter threshold for triggering an errored second. If the jitter (delay variation) (page 362) 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
- DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See this page (page 362). Default: “0 / IPP 0”.
- Protocol: Network protocol to use for sending Ping packets: ICMP or UDP. Default: ICMP.
7.12.8.3 Result metrics

- **Delay average (ms):** Average Ping delay during the selected time period.
- **Delay min (ms):** Minimum Ping delay.
- **Delay max (ms):** Maximum Ping delay.
- **DV (ms):** Ping delay variation.
- **Loss (%):** Percentage of Ping packets lost.
- **ES loss:** Number of errored seconds triggered by excessive Ping packet loss.
- **ES delay:** Number of errored seconds triggered because the Delay (Ping delay) threshold was exceeded.
- **ES response:** Number of errored seconds triggered because sent and received ICMP data did not match, or because the expected ICMP reply was not received — for example, if Time-To-Live (TTL) expired and the ICMP packet was dropped by a router.
- **ES DV:** Number of errored seconds triggered by excessive Ping delay variation.
- **ES total:** Aggregated errored seconds, taking into account all types of error.
- **SLA:** Service level agreement (page 372) fulfillment: equal to \((100 - \text{ES total})\) %.

---

7.12.9 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 284). 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 streams 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 293) 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.

**Note:** This test requires exclusive access to the Test Agent, meaning that no other tests or monitors can be assigned to the Test Agent.

### 7.12.9.1 Prerequisites

To perform BWPing measurements, you need to install at least one Netrounds Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found [here](page 64).

In your test or monitor, add a BWPing task and fill in the mandatory parameters below:
7.12.9.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- Advanced settings (page 177) 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 stream. See this page (page 364). Min: 64 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 362). 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.

7.12.9.3 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, Netrounds 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 Netrounds. 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 Netrounds 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.

7.12.9.4 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.
7.12.10 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 Netrounds Path trace tool 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.
7.12.10.1 Prerequisites

To run Netrounds Path trace measurements you need to have at least one Netrounds Test Agent installed. If you haven’t already done the installation, consult the installation guides found here (page 64).

In your test or monitor, add a Path trace task and fill in the mandatory parameters below:

7.12.10.2 Parameters

See the common parameters page (page 176) for the following:

- Parameters that are set on the test step (page 176) level: Duration, Fail threshold, and Wait for ready.
- SLA thresholds (page 177) for monitors: SLA Good and SLA Acceptable.
- Advanced settings (page 177) 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: 2. 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 362) 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 362) 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 372). 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**


• Max TTL: The maximum value of the Time-To-Live parameter. The same parameter also limits the number of streams 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: 64. Default: 30.

• DSCP/IPP: The Differentiated Services Code Point or IP Precedence to be used in IP packet headers. See [*this page* (page 362)]. 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.

**7.12.10.3 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 302). 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 303). 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 Netrounds format. See below (page 305).

**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 appear grayed out.
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 362), 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:
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 3 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 Netrounds.

- 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 304).

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 303) section.

<table>
<thead>
<tr>
<th>Router address</th>
<th>ES bar</th>
<th>Min round-trip delay (ms)</th>
<th>Avg round-trip delay (ms)</th>
<th>Max round-trip delay (ms)</th>
<th>Avg round-trip DV (ms)</th>
<th>Loss (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 52.93.2.131</td>
<td></td>
<td>3.16</td>
<td>4.21</td>
<td>17.38</td>
<td>1.97</td>
<td></td>
</tr>
<tr>
<td>2. 52.96.167.229</td>
<td></td>
<td>3.22</td>
<td>4.44</td>
<td>5.21</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>3. 195.2.25.193</td>
<td>ae3.xc1.skl.cfg.net</td>
<td>29.97</td>
<td>30.36</td>
<td>33.95</td>
<td>0.62</td>
<td></td>
</tr>
<tr>
<td>4. 195.2.2.206</td>
<td>ae32.xc1.atl.cfg.net</td>
<td>30.10</td>
<td>31.09</td>
<td>42.27</td>
<td>1.67</td>
<td></td>
</tr>
<tr>
<td>5. 195.2.28.165</td>
<td>ae14.xc1.hex.cfg.net</td>
<td>30.06</td>
<td>31.23</td>
<td>40.43</td>
<td>1.78</td>
<td></td>
</tr>
<tr>
<td>6. 190.2.24.157</td>
<td>ae19.xc1.lnd.cfg.net</td>
<td>30.14</td>
<td>30.25</td>
<td>30.74</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>7. 195.2.30.114</td>
<td>ae15.xc1.lns.cfg.net</td>
<td>29.91</td>
<td>29.98</td>
<td>30.52</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>8. 195.2.15.74</td>
<td></td>
<td>29.66</td>
<td>31.35</td>
<td>58.93</td>
<td>3.59</td>
<td></td>
</tr>
<tr>
<td>9. 194.70.97.50</td>
<td>ams2-ocx1-1-atl-0-s4.router.demon.net</td>
<td>32.93</td>
<td>33.29</td>
<td>39.56</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>10. 194.70.68.34</td>
<td>synchronoss-gw.sw1.cfg.net</td>
<td>55.78</td>
<td>57.56</td>
<td>58.85</td>
<td>0.66</td>
<td></td>
</tr>
<tr>
<td>11. 194.70.173.64</td>
<td></td>
<td>34.69</td>
<td>34.74</td>
<td>34.84</td>
<td>0.04</td>
<td></td>
</tr>
</tbody>
</table>

7.12.11 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 307).

**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 284) or UDP echo (page 293), where the reflector does create a new packet which it sends back to the Test Agent.

This task works only with IPv4.

### 7.12.11.1 Prerequisites

To perform UDP loopback measurements, you need to install at least one Netrounds Test Agent. For guidance on how to deploy a new Test Agent, see the installation guides found here (page 64).

You also need to have a device (router or switch) available in your network that can serve as reflector of UDP packets.

**Reflectors 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.

```plaintext
! 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 194.168.1.1 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.
loopback direction internal

! Specify the packet profile, defining the packets to be generated.
profile packet
! Specify the source IP address.
source-ip-addr 193.168.1.1
! 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
```
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 307)) 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:

```bash
! Create a "loopback Y" interface with an IP address and define it as "inside".
! In the above example, this IP address is 196.155.102.120 (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 232.157.100.122
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
! 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.

### 7.12.11.2 Parameters

See the [common parameters page](page 176) for the following:

- Parameters that are set on the **test step** (page 176) level: Duration, Fail threshold, and Wait for ready.
- **SLA thresholds** (page 177) for **monitors**: SLA Good and SLA Acceptable.
- **Advanced settings** (page 177) 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 307) 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 stream. See this page (page 364). Min: 64 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.

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 362) 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 362) 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 372). 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 362). 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 369). Min: 0. Max: 7. Default: 0.

7.12.11.3 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 misordering.

• **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 377) percentage.

• **SLA:** Service level agreement (page 372) fulfillment: equal to (100 – ES) %.

### 7.13 Security testing

#### 7.13.1 Introduction to security testing

The security tests in Netrounds 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 Netrounds cannot be mitigated by end-users; it is the network that must provide protection.

The Netrounds security tests (SEC specification) 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.

Netrounds supports security testing on the IPv4 protocol.

To perform security tests, one or two Netrounds 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 Netrounds 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 SEC and 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 83), 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 312) for an overview of all supported security features.
7.13.2 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.

7.13.2.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-V4-DHCPSTARV-1” and to SAVI section 3.1.2.

7.13.2.2 Impact

DoS

7.13.2.3 Test procedure

1. Customer verifies connectivity to ISP.
2. Customer takes the allowed number of IPv4 addresses.
3. Customer then sends another DHCP request.

7.13.2.4 Fail criteria

- Customer cannot obtain the allowed number of IPv4 addresses.
- Customer can obtain more than the allowed number of IPv4 addresses.

7.13.2.5 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.

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.

7.13.3.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-V4-CP-FRAG-1” and to SAVI section 3.1.2.

7.13.3.2 Impact

DoS

7.13.3.3 Test procedure

1. Customer sends a valid DHCP packet.
2. Customer sends DHCP packets fragmented into 40 byte and 104 byte fragments.

7.13.3.4 Fail criteria

• ISP does not receive the valid DHCP packet.
• ISP receives any fragment of a fragmented packet.

7.13.3.5 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.
7.13.4 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.

7.13.4.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-V4-FRAG-1” and to SAVI section 3.1.2.

7.13.4.2 Impact

DoS, Abuse, Illegal access to content

7.13.4.3 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.

7.13.4.4 Fail criteria

- ISP does not receive the non-fragmented packets.
- ISP receives all fragments of a fragmented packet.
7.13.4.5 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.

7.13.5 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.
7.13.5.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-CM-NEM-1” and to SAVI section 3.1.7.

7.13.5.2 Impact

MITM, DoS, Abuse

7.13.5.3 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.

7.13.5.4 Fail criteria

• One of the TCP ports is listening for traffic.
• Customer receives an answer to an SNMP Get, Ping Request, or NTP Get.

7.13.5.5 Parameters

General

• Customer: A Test Agent interface acting as a customer.
• Management IPs: IP addresses used to manage equipment, separated by commas.
7.13.6 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

7.13.6.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-V4-xxRP-1” and to SAVI section 3.1.7.

7.13.6.2 Impact

MITM, DoS
7.13.6.3 Test procedure

Customer listens during 60 seconds for traffic on each of the above protocols.

7.13.6.4 Fail criteria

- A packet from any router redundancy protocol is received by Customer.

7.13.6.5 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.

7.13.7 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

7.13.7.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-V4-ROUTE-1” and to SAVI section 3.1.7.

7.13.7.2 Impact

MITM, DoS

7.13.7.3 Test procedure

Customer sends multicast join messages used by the above routing protocols and then listens during 60 seconds for traffic on each protocol.

7.13.7.4 Fail criteria

- A packet from any routing protocol is received at Customer.

7.13.7.5 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.
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.

7.13.8.1 References

The test performed conforms to SEC (page 371) Access Certification ID “SEC-CM-SPT-1” and to SAVI section 3.2.3.

7.13.8.2 Impact

DoS, MITM

7.13.8.3 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.

7.13.8.4 Fail criteria

• An STP BPDU packet arrives at Customer.

7.13.8.5 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.

7.14 Utilities for testing

7.14.1 Delay

This task is a utility pausing 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.

© 2020 Netrounds. All rights reserved.
7.14.1.1 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 Dashboard

8.1 Insight dashboard

8.1.1 Introduction

Besides the basic dashboard covered here (page 7), Netrounds comes equipped with a more powerful vehicle of presentation called the Insight dashboard.

The Insight dashboard is fully user-configurable and is composed of building blocks (widgets), each of which holds a time series graph, a collection of ES bars, a table, or some other presentation. Data can be combined and filtered in a variety of ways.

The dashboard is chiefly geared towards the presentation of monitoring data. However, there are also facilities for listing tests with their outcomes.

To access the Insight dashboard, click Dashboard on the left-hand bar and select Insight dashboard.

8.1.2 Creating a new dashboard

You can create multiple dashboards and show one of these at a time.

At the outset the Dashboard view is empty; there are no preconfigured dashboards. To create one:

- Click the three-line button in the top left corner of the Dashboard view.
- Select New.
Dashboards that you create are collected in a list which is accessed by clicking the name of the currently selected dashboard. You switch between dashboards by selecting them in this list.

8.1.3 Editing a dashboard

To start editing a dashboard, do as follows:

- Select the dashboard from the list.
- Click the edit button (pen) next to the dashboard name.

8.1.3.1 Global dashboard functions

- **Time range:** Select the time interval to view. This works the same way as in the basic dashboard: see [this page](page 7).
  - **Previous/Next buttons:** Use these buttons to step between adjacent time intervals of the length currently selected. For example, with "15min" selected as interval, you can jump backwards from the interval \([-15 \ldots 0]\) min to the intervals \([-30 \ldots -15]\) min, \([-45 \ldots -30]\) min, etc.

- **Refresh interval:** Choose between 10 s and 30 s.

- **Filter:** Here you can apply a filter that applies to all widgets. For example, you can show only data with a specified tag attached. Note that further filters can be set in individual widgets (see [this section](page 330)).

- **Refresh now:** Click this button to refresh the dashboard immediately.

- **Pause refreshing/Restart refreshing:** Click once to pause refreshing of data on the dashboard. Click once more to resume data refresh.

8.1.3.2 Widget types

Dashboards in Netrounds present data in the following widget types:
Monitor-related

- **Time series graph**: This widget plots metrics as a function of time in the form of curves.
- **Top ES list**: This widget ranks rows in terms of errored seconds with the worst-performing row on top. Each row may represent, for example, a Test Agent or a single stream.
- **Table**: This widget presents metrics in a table.

Test-related

- **Test list**: This widget lists *one-time* tests defined in Control Center with indications of their status (e.g. Waiting, Running, Passed).
- **Recurring test list**: This widget lists *periodic* tests defined in Control Center with indications of their status.

Alarm-related

- **Alarm list**: This widget lists triggered alarms of the kinds defined in Control Center.

### 8.1.3.3 Adding a new widget to a dashboard

- Drag the box with the desired widget type to the dashboard grid, and drop the box where you want it.

Add widget

```
| Alarm list | Time series (lines) | Top ES list | Table | Test list | Recurring test list |
```

The default and initial widget size is $3 \times 4$; you can change that later by dragging the edges of the widget.

### 8.1.3.4 Configuring a widget

- Click the pen icon at top right in the widget to open a configuration dialog.

This dialog is common to all widget types, although for certain widgets parts of the dialog are omitted, while other widgets have special options.
Add widget *Time series (lines)*

**Set a title** *

| Title |

**Set grouping**

*Group by*

| ▼ stream_id |

*Sort by*  

| Highest | ▼ |

| Limit | ▼ |

**Set data filters**

| Add filter |

**Configure metrics** *

| Add metrics |

**Advanced options**  

---

**Setting a widget title**

- Set a title: Enter a title for the widget header here.

**Setting the time range (Alarm list only)**

Alarm lists have a special setting named Use dashboard time range. If this is set on, the widget will show alarm information only from the time range selected on the global time bar. You can turn the setting off to have all existing alarm information shown in the widget, regardless of whether it is within the time range set. The latter is useful in order to ensure that users do not miss any active alarms.

**Grouping data**

For simplicity, the text below deals with a time series widget; however, the configuration options are the same for top ES list and table widgets. For a top ES list, one curve corresponds to one ES bar, and for a table, one curve corresponds to one table row.
• **Group by**

This setting allows you to plot subsets of data separately.

At the outset, with nothing selected to group by, all data will be presented as a single group. That is, a single curve will be plotted in the graph.

If you select one parameter to group by, data will be plotted separately for each value of the parameter selected. This assumes that the Limit setting is not exceeded (see below).

*Example:* If you group by stream, data is presented separately for each stream.

If you select multiple parameters, a curve will be plotted for each combination of parameter values that occurs.

*Example:* If one parameter `ta_id` (Test Agent id) has three values \{1, 3, 5\} and another parameter `monitor_id` has two values \{26, 27\}, then one curve is drawn for each of \(3 \times 2 = 6\) pairs of “coordinates”: \{(1, 26), (1, 27), (3, 26), (3, 27), (5, 26), (5, 27)\}

If you have a large enough number of groups, the legend will be paginated. The Limit setting (see below) can be used to prevent this.

• **Sort by**

This determines how parameter values are sorted.

The sorting does not affect the presentation of data for a time series widget, while it does for a top ES list or table. However, for all widgets, the sort order affects what data is presented when the Limit setting comes into play; see **Limit** below.

• **Limit**

This is the maximum number of curves that can be drawn in the graph.

If this setting prevents the inclusion of all values of a parameter selected under Group by, a subset will be selected starting from the top or bottom, depending on the Sort by setting. The default limit is 10.

**Setting data filters**

You can filter your data on all parameters appearing in this box.

When you select a parameter here, a new box appears beneath it labeled with the parameter name. In the latter box, type or select the values that the filter should pass.

*Example:* If you pick `monitor_id` and enter the values “1, 2”, you will include data from the monitors with id = 1 and id = 2 only.

It is possible to set more than one filter.
Configuring metrics

Here you select what metric or metrics to plot.

The selected metrics appear in separate boxes beneath the Configure metrics box.

To only show metrics from a particular task type, set a “Task type” filter in the Set data filters box.

Advanced settings for metrics are as follows:

- **Aggregate by:** This determines how to process raw data into presented data. For example, suppose that we have data from a multitude of streams, and no grouping is applied to the data set. All data points for a given time instant then need to be compressed into a single presented value. The available aggregation options are Sum, Count, Max, Min, and Average; the default depends on the metric.
- **Line style:** One of solid, dashed, or dotted.
- **Color:** Line color.
- **On:** Select where to display the y-axis labels (on the left or right).

Additional columns for table

Select dimension to display as additional column for table.

You can add dimensions to be displayed as a column in the table widget. Select the dimension that will be added as another column.

Entering a widget description

- **Set description:** Here you can optionally enter a description of the widget. It will appear in the widget tooltip.

Advanced options

- **Show legend:** Show or hide the graph legend.

8.1.3.5 Moving widgets

You can move widgets around by dragging them. Note that if other widgets are in the way, they will be pushed along with the widget you are dragging; they will not be jumped over.

8.1.3.6 Cloning a widget

You can make a copy of a widget as follows: Click the three-dot button at top right, select Clone to, and then select what dashboard to put the new widget on. The new widget is identically configured to the old one.

8.1.3.7 Exporting data from a widget

To export data from a widget, click the three-dot button at top right and select Export as. Supported export formats are CSV and JSON.
8.1.3.8 Deleting widgets

- Click the pen icon on the dashboard to enter edit mode.
- Click the widget’s trash can button at top right.

8.1.4 Widget-specific features

8.1.4.1 Time series graph

- The metrics displayed in the graph are task type specific.
- Click an item in the graph legend to hide and show the corresponding metric. When a metric is hidden, it is grayed out in the legend.
- Click in the graph to show data at that point in time.

• Click and drag to zoom in on a selected interval. This operation is “local” and does not affect other widgets.
If you hold Ctrl while clicking and dragging, you will again zoom the time series graph, but the selected time interval is also applied to the rest of the dashboard, so that the data presented is confined to this interval.

- You can also hold Shift and scroll with the mouse wheel in order to zoom in and out in small increments.
- Click the reset button to reset the zoom setting (that is, revert to the default zoom).

### 8.1.4.2 Top ES list, Table

Just as for time series graphs, the metrics displayed in these widgets depend on the task type.

### 8.1.4.3 Test list, Recurring test list

In the Test list and Recurring test list widgets, the status of a test is indicated by one of these icons:
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 aborted because of an error.
The test has been canceled manually.
The test has been skipped.

In the Recurring test list widget, the status of test instances is indicated in the Status column: for example, the number of passed, failed, and scheduled tests.

8.1.4.4 Alarm list

For each alarm listed in an Alarm list, the following information is supplied:

- Summary: Description of the event that raised the alarm.
- Severity: Current severity of the alarm, if any: one of Critical, Major, Minor, or Warning.
- Max severity: The highest severity which the alarm has had over the course of its duration.
- Test Agent: Test Agent for which the alarm was raised.
- Raised: Time when the alarm was raised.
- Cleared: Time when the alarm was cleared.
- Task: Monitoring task for which the alarm was raised.
- Monitor: Monitor for which the alarm was raised.

8.1.5 Drill-down features

Drill-down features allow you to zero in on subsets of data by moving from one widget type to another, optionally in two steps. Details follow below.

8.1.5.1 Time series graph

When you click a data point (drawn as a small circle) in a time series graph, a new table widget opens. This table preserves the Group by setting from the original time series widget. For example, if the time series graph is grouped by Test Agent, then the table will have one row for each Test Agent. Moreover, the table will inherit any filter settings made both in the dashboard and in the time series graph.
Clicking a row in the table displays a new graph with metrics for that row alone. For example, if a table row represents a Test Agent, the graph will distinguish all streams for that Test Agent. The metrics shown are specific to the type of task performed. An ES metric in such a graph is presented as a background shading which extends to the full height of the graph.

8.1.5.2 Top ES list, Table

When you click a row in either of these widget types, a new widget opens displaying this row alone, with additional task-specific parameters and metrics shown. Clicking the row here, in turn, again produces a graph (as described in the previous section).

8.1.6 Deleting a dashboard

- Go to the dashboard you want to delete, then click the three-line button and select Delete.

9 Applications

9.1 Remote packet capturing

|nr-product| 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.

9.1.1 Non-live packet capture

Use this method to capture real user traffic on any of your Test Agent interfaces directly from your Netrounds 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:
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 Netrounds server, and is available only as long as you stay on the Remote Packet Capture page.

### 9.1.2 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.
- Check the Enable box on the right.

- You now get to select a capture interface and optionally which interface to capture from (the two can be different):
9.2 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. The main differences between Netrounds’ Speedtest and publicly available tools such as bredbandskollen.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 Netrounds’ Speedtest function is that it supports additional and more advanced testing and troubleshooting features, suitable for use if the Speedtest indicates a network performance issue.

Below, Netrounds’ built-in Speedtest function 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 Netrounds Control Center resides). For further information, consult the document “Creating a Custom Speedtest Web Page”, available at https://portal.netrounds.com.

### 9.2.1 Prerequisites

To start using the Netrounds 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. How to enable and configure Speedtest is described on the page Configuring Speedtest (page 33).

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.

### 9.2.2 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. You are now invited to choose whether you want to run the application using Adobe Flash or WebSocket. WebSocket makes it possible to run Speedtest from mobile devices or other browsers that do not have Flash installed.
On clicking one of the buttons, you are taken to the public Speedtest page. Its URL differs depending on the technology chosen:

- **Flash**: https://<Control Center host IP>/your account/speedtest
- **WebSocket**: http://<Control Center host IP>/your account/speedtest-websocket

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.
  - **Flash**: The number of sessions is configurable *(by default, 8 (page 33)).*
  - **WebSocket**: 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.
  - **Flash**: The number of sessions is configurable *(by default, 8 (page 33)).*
  - **WebSocket**: 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 Netrounds user interface. Click Apps in the main menu:

**Applications**

- **Speedtest**
  Use browser-based speed tests to simplify customer support.

- **Remote Packet Capture**
  Troubleshoot app problems using remote packet capture and analysis.

Then click the Speedtest box to go to the Speedtest result pages:
9.2.3.1 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.
9.2.3.2 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.

9.3 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 Netrounds 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 Netrounds proxy enables a forwarding mechanism for the management traffic between the Test Agents and the Netrounds cloud server in networks that do not have public IP addresses.

A proxy Test Agent always communicates on port 443 with other Test Agents.

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 Netrounds user interface, navigate to Test Agents, and click the relevant Test Agent in this view.
   - Go to the Applications tab, and enable Proxy. (See also this page (page 97).)
   - It is recommended 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 Netrounds GUI).
   - If the Test Agent is already registered with the Netrounds cloud server, go to Utilities > Change login server and change the login server from https://login.netrounds.com to the IP address of the proxy Test Agent interface ("eth1", etc.) configured in step 2 (the internal address to the proxy). Set the port to 443.
   - If the Test Agent is not yet registered, do the registration according to this page (page 109).

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 ntp.netrounds.com to your internal NTP server for all Test Agents, or:
   - Change the NTP server for all internal Test Agents from ntp.netrounds.com to the IP address of the proxy Test Agent’s “eth1” interface, and let the proxy Test Agent continue to synchronize to ntp.netrounds.com.

See the page Test Agent NTP configuration (page 98) for more information about NTP.

The proxy functionality is now configured and ready to use. This allows the Test Agents to communicate with the Netrounds cloud server via the proxy Test Agent, which forwards their traffic to the Netrounds cloud.

10 Alarms

10.1 Introduction to alarms

Monitors in Netrounds 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 363) within a specified time window exceeds a defined threshold. In other words, this kind of alarm is raised when the level of SLA (page 372) 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 Netrounds account, as described here (page 37).

How to activate an alarm for a monitor is explained on this page (page 344).

An overview of all alarms defined in your account is found on the alarm dashboard (page 346).

10.2 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 154).
Clicking Add new alarm opens the following dialog, which is mostly identical to the one for setting up alarm templates (page 42):

Add/Update Alarm

Provide alarm notification method for the triggered Alarms:

SNMP manager: Select existing... or Add New Manager...

Send trap per: Monitor Stream

Email list: Select existing... or Add New Email...

Trigger alarm on "error seconds"

Window size (s): 60

Send interval (s): 3,600

Send only once

Raise Clear

Critical threshold (s): 6

Major threshold (s): 1

Minor threshold (s):

Warning threshold (s):

Action (optional): Free text field

Trigger alarm on no data

Trigger alarm on "no data received"

Severity: Warning

Threshold (s): 1800

Load from template...

Add/Update alarm Cancel

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 42).
To finalize the activation of an alarm, click Add/Update alarm.

Below is an example of an active SNMP manager.

### 10.3 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 45).

#### Automatically cleared alarms

© 2020 Netrounds. All rights reserved.
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).

---

### 11 Collaboration and sharing

#### 11.1 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 335) function is disabled on shared Test Agents, and changing interface configurations is not possible.

Test Agents Lite cannot be shared.

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”.

#### 11.1.1 How to share a Test Agent

- Navigate to Test Agents in the main menu.

  ![Share Icon](image)

- 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.
11.1.2 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:

*Confirmation*

You have received a shared Test Agent **vta1**

from **demo** by D. E. Mo (**demo@**[redacted]**.com**).

Do you want to accept it?

- **Accept**
- **Reject**

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:

*Share Test Agent*

- **Shared by:** D. E. Mo (**demo@**[redacted]**.com**)
- **probe's name:** vta1
- **Description:** --
- **Message:** --
- **Share approved by:** D. E. Mo (**demo@**[redacted]**.com**)

- **Remove share**
- **Cancel**
11.1.3 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:

![Share Test Agent dialog]

• The **sharee** clicks the icon and then, in the dialog that appears, the Remove share button:

![Share Test Agent dialog]

In both cases, a confirmation dialog appears:
11.2 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”.

11.2.1 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.

In the dialog that appears, enter the account to share the template to. You can also optionally enter a message for the recipient.

- Check the I understand… checkbox.
- Click the Share template button.
11.2.2 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:
11.2.3 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:

```
Share template

Shared by: D. E. Mo (demo@domain.com)
template’s name: UDP template
Description: This is a template for UDP
Message: --
Share approved by: D. E. Mo (demo@domain.com)
Last Modified: 2019-06-14 11:33:31

Remove share  Cancel
```

- 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:

```
Share template

Share to: Enter account name This value is required.
Message (optional): Optional message to recipient
I understand that this template will be shared to the account specified above read more

Accepted template shares:

- cisco by D. E. Mo (demo@domain.com) Remove

Share template
```

• 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:

**Confirmation**

Do you really want to remove the shared template?

- **Ok**
- **Cancel**

### 11.3 Sharing test and monitoring results

You can share specific live test and monitoring results with partners and customers. The results are then shown on your partner’s or customer’s Netrounds 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 “sharee.”

#### 11.3.1 How to share a monitor

- Navigate to Monitoring in the main menu.

  ![Share icon]

  In the Share column, click the icon for the monitor to be shared.

- 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.

11.3.2 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:

**Share Results**

- Shared by: demo@netrounds.com
- result's name: AWS Monitor
- Description: --
- Message: --
- Share approved by: demo@netrounds.com

**11.3.3 Removing a shared monitor**

Both sharer and sharee 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 sharee clicks the Remove share button:

In both cases, a confirmation dialog appears:
# Definitions and technical notes

## 12.1 Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>APN</td>
<td>Access Point Name</td>
</tr>
<tr>
<td>ARP</td>
<td>Address Resolution Protocol</td>
</tr>
<tr>
<td>BB</td>
<td>Bottleneck Bandwidth</td>
</tr>
<tr>
<td>BDP</td>
<td>Bandwidth Delay Product</td>
</tr>
<tr>
<td>BPDU</td>
<td>Bridge Protocol Data Unit</td>
</tr>
<tr>
<td>BSS</td>
<td>Business Support System</td>
</tr>
<tr>
<td>CC</td>
<td>Continuity Count</td>
</tr>
<tr>
<td>CDP</td>
<td>Cisco Discovery Protocol</td>
</tr>
<tr>
<td>CESoE</td>
<td>Circuit Emulation Services over Ethernet</td>
</tr>
<tr>
<td>CFM</td>
<td>Connectivity Fault Management</td>
</tr>
<tr>
<td>CGMP</td>
<td>Cisco Group Management Protocol</td>
</tr>
<tr>
<td>CIR</td>
<td>Committed Information Rate</td>
</tr>
<tr>
<td>CPE</td>
<td>Customer-Premises Equipment</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>DS</td>
<td>Differentiated Services</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>DSLAM</td>
<td>Digital Subscriber Line Access Multiplexer</td>
</tr>
<tr>
<td>DTP</td>
<td>Dynamic Trunking Protocol</td>
</tr>
<tr>
<td>DUT</td>
<td>Device Under Test</td>
</tr>
<tr>
<td>DV</td>
<td>Delay Variation</td>
</tr>
<tr>
<td>DVB-C</td>
<td>Digital Video Broadcasting – Cable</td>
</tr>
<tr>
<td>EAPoL</td>
<td>Extensible Authorization Protocol over LAN</td>
</tr>
<tr>
<td>Ec/Io</td>
<td>Per-chip signal-to-noise ratio</td>
</tr>
<tr>
<td>EIGRP</td>
<td>Enhanced Interior Gateway Routing Protocol</td>
</tr>
<tr>
<td>EIR</td>
<td>Excess Information Rate</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>ES</td>
<td>Errored Second(s)</td>
</tr>
<tr>
<td>ESMC</td>
<td>Ethernet Synchronization Messaging Channel</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETH</td>
<td>Ethernet</td>
</tr>
<tr>
<td>EVC</td>
<td>Ethernet Virtual Connection</td>
</tr>
<tr>
<td>EVPL</td>
<td>Ethernet Virtual Private Line</td>
</tr>
<tr>
<td>EVP-LAN</td>
<td>Ethernet Virtual Private LAN</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FM</td>
<td>Fault Management</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communication</td>
</tr>
<tr>
<td>GSMP</td>
<td>General Switch Management Protocol</td>
</tr>
<tr>
<td>HLS</td>
<td>HTTP Live Streaming</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGMP</td>
<td>Internet Group Management Protocol</td>
</tr>
<tr>
<td>IMAP</td>
<td>Internet Message Access Protocol</td>
</tr>
<tr>
<td>IMEI</td>
<td>International Mobile Equipment Identity</td>
</tr>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber Identity</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPP</td>
<td>IP Precedence</td>
</tr>
<tr>
<td>IS-IS</td>
<td>Intermediate System – Intermediate System</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>L2CP</td>
<td>Layer 2 Control Protocols</td>
</tr>
<tr>
<td>LACP</td>
<td>Link Aggregation Control Protocol</td>
</tr>
<tr>
<td>LAMP</td>
<td>Link Aggregation Marker Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LLC</td>
<td>Logical Link Control</td>
</tr>
<tr>
<td>LLDP</td>
<td>Link Layer Discovery Protocol</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MANO</td>
<td>Management and Orchestration</td>
</tr>
<tr>
<td>MBH</td>
<td>Mobile Backhaul</td>
</tr>
<tr>
<td>MC</td>
<td>Multicast</td>
</tr>
<tr>
<td>ME</td>
<td>Maintenance Entity</td>
</tr>
<tr>
<td>MEF</td>
<td>Metro Ethernet Forum</td>
</tr>
<tr>
<td>MEG</td>
<td>Maintenance Entity Group</td>
</tr>
<tr>
<td>MEL</td>
<td>MEG (Maintenance Entity Group) Level</td>
</tr>
<tr>
<td>MEP</td>
<td>MEG (Maintenance Entity Group) End Point</td>
</tr>
<tr>
<td>MIB</td>
<td>Management Information Base</td>
</tr>
<tr>
<td>MIP</td>
<td>MEG (Maintenance Entity Group) Intermediate Point</td>
</tr>
<tr>
<td>MITM</td>
<td>Man In The Middle</td>
</tr>
<tr>
<td>MLD</td>
<td>Multicast Listener Discovery</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Layer Switching</td>
</tr>
<tr>
<td>MPTS</td>
<td>Multi Program Transport Stream</td>
</tr>
<tr>
<td>MSTP</td>
<td>Multiple Spanning Tree Protocol</td>
</tr>
<tr>
<td>MVRP</td>
<td>Multiple VLAN Registration Protocol</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmission Unit</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>NDP</td>
<td>Neighbor Discovery Protocol</td>
</tr>
<tr>
<td>NFV</td>
<td>Network Function Virtualization</td>
</tr>
<tr>
<td>NFVI</td>
<td>Network Function Virtualization Infrastructure</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Controller (or: Card)</td>
</tr>
<tr>
<td>NMS</td>
<td>Network Management Station</td>
</tr>
<tr>
<td>NOC</td>
<td>Network Operations Center</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>OAM</td>
<td>Operations, Administration, and Management (or: Maintenance)</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First (protocol)</td>
</tr>
<tr>
<td>OSS</td>
<td>Operations Support System</td>
</tr>
<tr>
<td>OTT</td>
<td>Over-The-Top</td>
</tr>
<tr>
<td>OUI</td>
<td>Organizationally Unique Identifier</td>
</tr>
<tr>
<td>OWAMP</td>
<td>One-Way Active Measurement Protocol</td>
</tr>
<tr>
<td>PAT</td>
<td>Program Association Table</td>
</tr>
<tr>
<td>PCAP</td>
<td>Packet Capture</td>
</tr>
<tr>
<td>PCP</td>
<td>Priority Code Point</td>
</tr>
<tr>
<td>PCR</td>
<td>Program Clock Reference</td>
</tr>
<tr>
<td>PD</td>
<td>Packet Delay</td>
</tr>
<tr>
<td>PDV</td>
<td>Packet Delay Variation</td>
</tr>
<tr>
<td>PGW</td>
<td>Packet data network Gateway</td>
</tr>
<tr>
<td>PID</td>
<td>Packet Identifier</td>
</tr>
<tr>
<td>PLR</td>
<td>Packet Loss Ratio</td>
</tr>
<tr>
<td>PM</td>
<td>Performance Management</td>
</tr>
<tr>
<td>PMT</td>
<td>Program Map Table</td>
</tr>
<tr>
<td>PPP</td>
<td>Point-to-Point Protocol</td>
</tr>
<tr>
<td>PPPoE</td>
<td>Point-to-Point Protocol over Ethernet</td>
</tr>
<tr>
<td>PVST</td>
<td>Per-VLAN Spanning Tree</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>Q-in-Q</td>
<td>Informal name for IEEE 802.1ad, amendment to IEEE 802.1Q-1998</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RARP</td>
<td>Reverse Address Resolution Protocol</td>
</tr>
<tr>
<td>RAT</td>
<td>Radio Access Technology</td>
</tr>
<tr>
<td>RGW</td>
<td>Residential Gateway</td>
</tr>
<tr>
<td>RIPv2</td>
<td>Routing Information Protocol version 2</td>
</tr>
<tr>
<td>RPCAP</td>
<td>Remote Packet Capture</td>
</tr>
<tr>
<td>RRH</td>
<td>Remote Radio Head</td>
</tr>
<tr>
<td>RSCP</td>
<td>Received Signal Code Power</td>
</tr>
<tr>
<td>RSRP</td>
<td>Reference Signal Received Power</td>
</tr>
<tr>
<td>RSRQ</td>
<td>Reference Signal Received Quality</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indication</td>
</tr>
<tr>
<td>RSTP</td>
<td>Rapid Spanning Tree Protocol</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>RTT</td>
<td>Round-Trip Time</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SAP</td>
<td>Service Access Point</td>
</tr>
<tr>
<td>SAVI</td>
<td>Source Address Validation Improvement</td>
</tr>
<tr>
<td>SDN</td>
<td>Software-Defined Networking</td>
</tr>
<tr>
<td>SEC</td>
<td>Secure End-user Connections</td>
</tr>
<tr>
<td>SES</td>
<td>Severely Errored Second(s)</td>
</tr>
</tbody>
</table>

Continued on next page
12.2 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 Netrounds Test Agent is configured and running in bridge mode. Test Agents support bridging of physical interfaces. This particular setup is used for inline IPTV testing (page 205) and allows all IPTV traffic to pass through the Test Agent, which measures quality on the same channels as the set-top box joins.
12.3 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.

Netrounds’ jitter calculation for synthetic traffic follows IETF RFC 3393. In short, Netrounds 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.

12.4 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.

Netrounds Test Agents Lite currently do not support DSCP settings in outgoing IP packets.

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.
A bit of history on Type of Service (ToS), IP Precedence, and DiffServ (DSCP) can be found here.

### 12.5 Errored seconds (ES) metric: Method of calculation

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 Netrounds 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 Netrounds 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 370). 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.
12.5.1 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.

12.6 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 Netrounds places a sequence number, a timestamp, a checksum, and a stream 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.
12 DEFINITIONS AND TECHNICAL NOTES

- **Minimum IPv6 frame size** = 18 (Ethernet) + 40 (IPv6) + 8 (UDP) + 18 (payload) = **84 bytes**

<table>
<thead>
<tr>
<th>Ethernet</th>
<th>IPv6</th>
<th>UDP</th>
<th>Netrounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 bytes</td>
<td>40 bytes</td>
<td>8 bytes</td>
<td>18 bytes</td>
</tr>
</tbody>
</table>

IPv6 header 0–64 bytes

Payload 66–1500 bytes

Minimum frame size in Netrounds 84 bytes

12.7 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:

<table>
<thead>
<tr>
<th>MOS</th>
<th>Quality</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Perceptible</td>
</tr>
<tr>
<td>3</td>
<td>Fair</td>
<td>Annoying</td>
</tr>
<tr>
<td>2</td>
<td>Poor</td>
<td>Very annoying</td>
</tr>
<tr>
<td>1</td>
<td>Bad</td>
<td>Impossible to communicate</td>
</tr>
</tbody>
</table>

12.8 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.

12.9 MPEG metrics

In the course of *IPTV testing* (page 201), several MPEG metrics are calculated and reported by Netrounds. 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. Netrounds uses this timestamp field to calculate the PCR jitter (delay variation) (page 362) for the received MPEG stream.

**Program Allocation Table (PAT) errors:** The Program Allocation Table (PAT) lists all programs available in the 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 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 streams, the standards stipulate that a frame should be received once every 5 seconds. If no frame is received for 5 seconds, Netrounds starts counting PID errors (one for every second during which no frame is received).

**RTP jitter, loss and misorders:** If the IPTV streams contain RTP headers, Netrounds will also calculate RTP jitter, loss, and misorderings, all of which are analogous to the corresponding metrics for IP. Whether or not the stream contains RTP headers depends on the encoder at the TV head-end.

### 12.10 MPEG rate vs. MPEG transport rate

This article describes the difference between MPEG rate and MPEG transport rate, both of which are reported in Netrounds.

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 (%)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Overhead (bytes)</th>
<th>Theoretical overhead (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG over Ethernet with 802.1q, with RTP</td>
<td>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</td>
<td>Overhead (bytes) / MPEG TS payload (bytes) = (82 + 28) / (184 × 7) = 8.5404%</td>
</tr>
<tr>
<td>MPEG over Ethernet with 802.1q, without RTP</td>
<td>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</td>
<td>Overhead (bytes) / MPEG TS payload (bytes) = (70 + 28) / (184 × 7) = 7.6087%</td>
</tr>
</tbody>
</table>

“Rate (Mbit/s)” displayed in the Netrounds 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.

<table>
<thead>
<tr>
<th>Client/Channel</th>
<th>ES history</th>
<th>Rate (Mbit/s)</th>
<th>Error Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Total</td>
</tr>
<tr>
<td>- Crown:brtl</td>
<td></td>
<td>27.57</td>
<td>0.0%</td>
</tr>
<tr>
<td>- Viasat SVT1 HD</td>
<td></td>
<td>11.95</td>
<td>0.0%</td>
</tr>
<tr>
<td>- Viasat TV3 HD</td>
<td></td>
<td>10.57</td>
<td>0.0%</td>
</tr>
<tr>
<td>- Viasat TV3 SD</td>
<td></td>
<td>5.85</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

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.

12.11 Netrounds server

12.11.1 How sensitive is the management connection to the Netrounds server?

The Netrounds Test Agents are very robust to network disturbances affecting the management connection from the Test Agent to the Netrounds 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 Netrounds 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.
12.11.2 Is my measurement performance affected by the geographical distance to my Netrounds server?

No, it is not.

No measurement traffic ever occurs between your Test Agents and the Netrounds server. The Test Agents communicate with the Netrounds 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 Netrounds 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 Netrounds server for further post-processing and storage.

12.11.3 Where are Netrounds’ 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.

12.12 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.

Netrounds Test Agents Lite and Test Agent Applications currently do not support PCP settings in outgoing IP packets.

The PCP field was introduced by the IEEE P802.1p working group.

12.13 Using a proxy

The Netrounds 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 Netrounds Test Agent or a standard HTTP proxy. The two possibilities are discussed in turn below.

12.13.1 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 Netrounds 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 Netrounds server, whilst the other interface is the new connection point for the Test Agents inside the network. The Netrounds proxy will only forward needed management traffic between the Netrounds 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.”
Netrounds 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 Netrounds technical support at support@netrounds.com.

### 12.13.2 Using a standard HTTP proxy

It is also possible to connect a Test Agent to Netrounds 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 110).

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 109). Once registered, however, the Test Agent can use an HTTP proxy to connect.

### 12.14 Resolution of Netrounds measurement data

Time series of Netrounds measurement data are stored in a round-robin database, where older data is progressively consolidated into lower resolutions.

#### 12.14.1 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 Netrounds 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:

<table>
<thead>
<tr>
<th>Timeframe</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 hours</td>
<td>10 seconds</td>
</tr>
<tr>
<td>2 days</td>
<td>1 minute</td>
</tr>
<tr>
<td>1 week</td>
<td>5 minutes</td>
</tr>
<tr>
<td>1 month</td>
<td>20 minutes</td>
</tr>
<tr>
<td>1 year</td>
<td>4 hours</td>
</tr>
</tbody>
</table>

12.14.2 Resolution for tests

Tests in Netrounds 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.

12.15 Secure End-user Connections (SEC)

SEC is the Swedish industry standard on how to build, certify, and secure broadband networks. A SEC-certified network ensures a safe and qualitative connection where all types of spoofing and poisoning are made impossible. The SEC recommendation can be found [here](#).

12.16 Security in Netrounds

12.16.1 Access to Netrounds Control Center

Access to Netrounds Control Center takes place over SSL (using HTTPS), ensuring secure communication between your web browser and the Netrounds server.

12.16.2 Accounts and user permissions

The measurement data you collect with Netrounds is visible in Netrounds Control Center only to yourself and to any users you invite to your account. How to invite users and specify what permissions they should be granted is described [here](#) (page 20). As always, you should ensure that your passwords are not easy to break.

12.16.3 Communication with Test Agents

Only Netrounds Control Center is authorized to interact remotely with Test Agents. Everything to do with Test Agent measurements – configuration and execution of tests and monitors, and retrieval of results – is handled from Netrounds Control Center. Each Test Agent communicates with Netrounds Control Center by initiating an outgoing OpenVPN connection encrypted with a 2048-bit RSA key. Requirements on open ports are given in the Test Agent [installation instructions](#) (page 64).

There is however one way to access a Test Agent locally, namely, through a text-based interface called the local console. The local console must be used for Test Agent registration, and it can also be used to configure and troubleshoot the Test Agent in various ways. See [these pages](#) (page 108).

Test Agents are generally closed to incoming traffic, which means that they do not accept incoming connections on any port. Exceptions are as follows:
• Test Agents will listen to traffic on TCP/UDP ports specified in tests and monitors.

• If the Speedtest (page 338) application is enabled on a Test Agent, it will listen to the TCP port specified in the Speedtest settings (page 33).

12.16.4 Stored data

The only information stored in Netrounds about the network where a Test Agent is placed and the traffic it receives are the parameter settings and measurement results that the user can see in the Netrounds Control Center web GUI. An example of parameter settings is the Test Agent interface configuration; examples of measurement results are statistics on delay, jitter and packet loss obtained from a test. The physical location of the Test Agent is stored only if you enter geographical coordinates yourself in the Test Agent’s properties (GPS Location tab). Actual customer data is not stored – only measurement reports and statistics.

Data from packet capture (whether live or non-live) is not stored on the Test Agents, nor on the Netrounds server, as explained here (page 335).

12.17 Severely Errored Seconds (SES)

The Severely Errored Seconds (SES) metric is very similar to the regular Errored Seconds (ES) (page 363) 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. Netrounds has its own user-configurable SES thresholds for packet loss, delay and jitter. SES statistics appear in the table view in the result report.

12.18 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 Netrounds, all metrics related to service degradation are transformed into errored seconds (ES) (page 363) by setting appropriate thresholds for the metrics.

The level of SLA fulfillment given in monitoring results is calculated as 100 – ES (%). With Netrounds’ default SLA thresholds, 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 figures 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 Netrounds 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 37).
**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.

### 12.19 SNMP

Netrounds 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 38).

The SNMP traps are sent from the host where Netrounds Control Center is installed; no SNMP traps are sent directly from any Test Agent.

#### 12.19.1 Version 2c

Netrounds’ SNMPv2 implementation 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.

#### 12.19.2 Version 3

Netrounds’ SNMPv3 implementation 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 |nr-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, minimum 8 characters.
- Privacy password: Password to use for authentication, minimum 8 characters.

### 12.20 TCP implementation in Netrounds

Netrounds 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.

Netrounds 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, Netrounds uses CUBIC’s window scaling algorithm. The window size therefore varies, but Netrounds 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 Agents Lite and 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.

### 12.21 Theoretical maximum throughput with Netrounds Test Agents

This article discusses theoretical maximum throughput in Netrounds as compared to line rate, differences being due to protocol overhead. We frame our theoretical reasoning in the context of Netrounds 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 364).

#### 12.21.1 Ethernet data rates

```latex
\begin{tabular}{|c|c|c|c|c|}
\hline
Phy: & Phy: & \textbf{Ethernet frame} & \textbf{Inter-packet gap} \\
8 bytes & 12 bytes & & \\
\hline
Preamble + SFD & & & \\
6 bytes & 6 bytes & 4 bytes & 2 bytes & 46–1500 bytes & 4 bytes \\
\hline
MAC destination address & MAC source address & Optional 802.1q tag & Length/ type & Payload & CRC \\
\hline
\end{tabular}
```

The following table shows theoretical maximum Ethernet data rates in Netrounds 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 Netrounds.
### Protocol

<table>
<thead>
<tr>
<th>Protocol, no 802.1q, MTU = 1500 bytes</th>
<th>Ethernet overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ethernet overhead</strong></td>
<td>8 preamble + 14 header + 4 CRC + 12 gap = 38 bytes/packet</td>
<td>(1500 + 18) / (1500 + 38) = 98.6996%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol, no 802.1q, MTU = 494 bytes</th>
<th>Ethernet overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ethernet overhead</strong></td>
<td>8 preamble + 14 header + 4 CRC + 12 gap = 38 bytes/packet</td>
<td>(494 + 18) / (494 + 38) = 96.2406%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol, no 802.1q, MTU = 46 bytes</th>
<th>Ethernet overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ethernet overhead</strong></td>
<td>8 preamble + 14 header + 4 CRC + 12 gap = 38 bytes/packet</td>
<td>(46 + 18) / (46 + 38) = 76.1905%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol, with 802.1q, MTU = 1500 bytes</th>
<th>Ethernet overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ethernet overhead</strong></td>
<td>8 preamble + 18 header + 4 CRC + 12 gap = 42 bytes/packet</td>
<td>(1500 + 22) / (1500 + 42) = 98.7030%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol, with 802.1q, MTU = 494 bytes</th>
<th>Ethernet overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ethernet overhead</strong></td>
<td>8 preamble + 18 header + 4 CRC + 12 gap = 42 bytes/packet</td>
<td>(494 + 22) / (494 + 42) = 95.5224%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protocol, with 802.1q, MTU = 46 bytes</th>
<th>Ethernet overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ethernet overhead</strong></td>
<td>8 preamble + 18 header + 4 CRC + 12 gap = 42 bytes/packet</td>
<td>(46 + 22) / (46 + 42) = 72.7273%</td>
</tr>
</tbody>
</table>

#### 12.21.2 TCP over Ethernet data rates

![Ethernet frame diagram](image)
The following table shows theoretical maximum TCP data rates in Netrounds 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 Netrounds.

It is assumed throughout that no header compression takes place.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>TCP/IP overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP over Ethernet, no 802.1q, IPv4</td>
<td>20 IPv4 + 20 TCP + 12 timestamps = 52 bytes/packet</td>
<td>((1500 - 52) / (1500 + 38) = 94.1482%)</td>
</tr>
<tr>
<td>TCP over Ethernet, 802.1q, IPv4</td>
<td>20 IPv4 + 20 TCP + 12 timestamps = 52 bytes/packet</td>
<td>((1500 - 52) / (1500 + 42) = 93.9040%)</td>
</tr>
<tr>
<td>TCP over Ethernet, no 802.1q, IPv6</td>
<td>40 IPv6 + 20 TCP + 12 timestamps = 72 bytes/packet</td>
<td>((1500 - 72) / (1500 + 38) = 92.8479%)</td>
</tr>
<tr>
<td>TCP over Ethernet, 802.1q, IPv6</td>
<td>40 IPv6 + 20 TCP + 12 timestamps = 72 bytes/packet</td>
<td>((1500 - 72) / (1500 + 42) = 92.6070%)</td>
</tr>
</tbody>
</table>

### 12.21.3 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 Netrounds. Rather, the data rate presented for UDP streams is the Ethernet data rate (see above (page 374)).
12 DEFINITIONS AND TECHNICAL NOTES

### UDP/IP overhead (bytes)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>UDP/IP overhead (bytes)</th>
<th>Theor. max. data rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP over Ethernet, no 802.1q, IPv4</td>
<td>20 IPv4 + 8 UDP = 28 bytes/packet</td>
<td>(1500 – 28) / (1500 + 38) = 95.7087%</td>
</tr>
<tr>
<td>UDP over Ethernet, 802.1q, IPv4</td>
<td>20 IPv4 + 8 UDP = 28 bytes/packet</td>
<td>(1500 – 28) / (1500 + 42) = 95.4604%</td>
</tr>
<tr>
<td>UDP over Ethernet, no 802.1q, IPv6</td>
<td>40 IPv6 + 8 UDP = 48 bytes/packet</td>
<td>(1500 – 48) / (1500 + 38) = 94.4083%</td>
</tr>
<tr>
<td>UDP over Ethernet, 802.1q, IPv6</td>
<td>40 IPv6 + 8 UDP = 48 bytes/packet</td>
<td>(1500 – 48) / (1500 + 42) = 94.1634%</td>
</tr>
</tbody>
</table>

#### 12.22 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 372). A period of unavailability starts with 10 consecutive SES and ends with 10 consecutive non-ES (page 363). Those first 10 consecutive SES are part of the period of unavailability. See the diagram below, taken from Y.1563, for an illustration:

Netrounds 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.

#### 12.23 VLAN

Netrounds supports both untagged and tagged interfaces on Test Agents (except on Test Agent Applications and Test Agents Lite). 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, Netrounds 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.
13 Release notes

13.1 Release notes, Netrounds software version 2.34.1

This release is a bug fix release and therefore does not contain any major new features or improvements. Regarding the bug fixes, please refer to the pdf version of these release notes, available at https://portal.netrounds.com.

13.2 Release notes, Netrounds software version 2.34.0

13.2.1 New features

13.2.1.1 Access to Test Agents via SSH

All Test Agents can now be accessed via SSH (this was previously possible only for Test Agents deployed in Amazon Web Services or Microsoft Azure). Public SSH keys can be uploaded to a Test Agent either from Control Center or from the Test Agent local console. Users log in by supplying their corresponding private SSH key.

By enabling remote login to the Test Agent, this feature facilitates troubleshooting, especially in cases where the Test Agent has lost its connection to Control Center or has never come online on being shipped to site.

13.2.1.2 Test Agent Application support in ConfD

This release introduces full support for the containerized Test Agent Application in ConfD. In other words, such Test Agents can be configured and controlled via the NETCONF/YANG interface.

13.2.2 Improvements

13.2.2.1 Support for more Test Agents and streams in Control Center

Netrounds Control Center now supports up to 8000 Test Agents and 100,000 concurrent streams.

13.2.2.2 Higher-performance UDP and TCP: Multiple streams per task

The achievable total throughput in UDP and TCP has been increased by allowing multiple streams per task. The maximum number of streams is 64.

13.2.2.3 Increased maximum send rate for BWPing

The “maximum send rate” user interface limitation in BWPing tasks has been increased from 50 Mbit/s to 4000 Mbit/s.

13.2.2.4 PCAP buffer expanded

The PCAP buffer used in (non-live) remote packet capture has been enlarged from 1 MB to 15 MB.
13.2.2.5 Port 6000 default in Test Agent registration to on-premise Control Center

In the Test Agent registration procedure, when you specify a server different from the SaaS server, the Test Agent will connect to port 6000 if you do not enter a port. You can use a different port by specifying it explicitly in the registration dialog.

13.2.2.6 Test Agent Appliance OS update

The Test Agent Appliance OS has been upgraded to Debian Buster.

13.2.3 Bug fixes

Please refer to the pdf version of these release notes, available at https://portal.netrounds.com.

13.2.4 Removed features

None

13.3 Release notes, Netrounds software version 2.33.0

13.3.1 New features

13.3.1.1 Enhanced user and account handling in ncc CLI command

The ncc command has been expanded with the following:

- ncc user-list for listing Network Control Center users
- ncc user-update for updating users’ passwords
- ncc user-delete: This by default deactivates a user and prevents it from logging in. Actually deleting the user requires a force option.
- ncc user-activate for re-activating users that have been deactivated
- ncc account-update for changing the owner of an account.

13.3.1.2 Enhanced NETCONF user management via CLI

The following commands have been added for managing NETCONF users:

- ./list-netconf-users for listing NETCONF users
- ./update-netconf-user for updating the passwords of NETCONF users
- ./delete-netconf-user for deleting NETCONF users.
13.3.2 Improvements

13.3.2.1 Additional details in SNMP traps

SNMP traps now include the following additional information:

- Timestamp indicating when a stream failed
- Remote endpoint of the stream (TWAMP reflector, other Test Agent, etc.)

13.3.3 Bug fixes

Please refer to the pdf version of these release notes, available at https://portal.netrounds.com.

13.3.4 Removed features

None

13.4 Release notes, Netrounds software version 2.32.0

Release 2.32.0 mainly contains security enhancements and therefore does not include any major new features.

**Note:** If you intend to use the REST API, you need to change the value of the parameter `REST_TOKEN_LIFETIME` in the file `/etc/netrounds/netrounds.conf` to whatever is required in your case. As of version 2.32.0, the lifetime of REST API tokens has been limited and is 10 years by default.

13.4.1 Improvements

13.4.1.1 Security enhancements

The security in Netrounds has been enhanced in a multitude of ways, including the following:

- The product uses only NIST-approved cryptographic algorithms (this also relates to hashing of sensitive data in the database).
- The product allows definition of authentication throttling for all authentication channels.
- The product allows enabling of a single-session mechanism (only one session is allowed for a user).
- The product is protected against clickjacking.
- The product’s REST API authentication mechanisms allow limited-time access as well as management of tokens.
- The product follows OWASP rules related to setting all security HTTP headers and allows managing them.
- Separate admin and root passwords can be set in Test Agent Appliance. This can optionally be done using cloud-init.
- Inactive SSH sessions on a Test Agent Appliance are automatically closed.
- Login banners are configurable in both Control Center and Test Agent, allowing display of warning notices.

13.4.2 Bug fixes

Please refer to the pdf version of these release notes, available at https://portal.netrounds.com.

13.4.3 Removed features

None

13.5 Release notes, Netrounds software version 2.31.0

13.5.1 New features

13.5.1.1 SNMP traps per stream

Alarms in the form of SNMP traps can now optionally be sent per stream rather than per monitor task, which was previously the only option. This is especially valuable for TWAMP and other reflector-based monitoring. On the Alarms dashboard, SNMP traps are now differentiated by stream.

13.5.1.2 Support for /31 and /32 subnets on Test Agents

Netrounds now allows assigning /31 and /32 subnets to Test Agent interfaces, through DHCP as well as static address assignment.

For IPv6, the corresponding /127 and /128 subnets are likewise supported.

13.5.1.3 Ping host inventory

Ping hosts can be organized in an inventory (similar to those for IPTV channels, etc.). When setting up test and monitor tasks, users can then pick Ping hosts from the inventory instead of specifying them manually.

13.5.1.4 Support for Control Center and Test Agents in IPv6-only network

Running Netrounds in an all-IPv6 network is now fully supported. This encompasses Test Agent management as well as configuration and execution of tests and monitors, optionally via ConfD or REST.

13.5.1.5 Speedtest via REST

The Speedtest application can be controlled via the REST API.
13.5.1.6 Test Agent Application

A new kind of Test Agent called Test Agent Application has been developed as a replacement for Test Agent Lite, which is now end of sale (see announcement here). Test Agent Application is currently capable of TCP, UDP, and TWAMP; further functionality will be added in future releases.

Test Agent Application runs 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.

13.5.2 Improvements

13.5.2.1 Customizable Speedtest page

A library of the Speedtest client-side Javascript code has been created, making it possible for a customer to create their own Speedtest page for their users.

13.5.2.2 Port configurable for SNMP managers

The port on which SNMP managers receive SNMP traps from Netrounds can now be configured arbitrarily. Previously, SNMP traps were always sent to UDP port 162.

13.5.3 Bug fixes

None

13.5.4 Removed features

None

13.6 Release notes, Netrounds software version 2.30.0

13.6.1 New features

13.6.1.1 TWAMP reflector functionality in Test Agent

Test Agents are now capable of acting not only as TWAMP senders but also as TWAMP reflectors. The senders may be other Test Agents, or they may be third-party devices. In the latter case, a new task type called "TWAMP Reflector" is used on the reflector Test Agent.

13.6.2 Improvements

None

13.6.3 Bug fixes

None
13.6.4 Removed features

None

13.7 Release notes, Netrounds software version 2.29.0

13.7.1 New features

13.7.1.1 Enhanced scalability

The number of streams that Netrounds Control Center can monitor concurrently has been increased.

13.7.1.2 Virtual Test Agent deployment in Google Cloud

If you are running applications in Google Cloud, you can now quality-assure these applications by deploying a Netrounds Test Agent in GCP (Google Cloud Platform). Typical examples of such applications are SAP and SharePoint.

The feature is added alongside the existing support for deploying Test Agents on other virtualization/cloud computing platforms such as Amazon WS and Microsoft Azure.

13.7.1.3 Test Agent kernel update

The kernel has been upgraded to the latest Long Term Support version, 4.19.

13.7.1.4 LDAP and TACACS+ authentication

As an alternative to local authentication, Netrounds Control Center can now authenticate users by contacting an LDAP or TACACS+ server where Netrounds users and their privileges are managed.

13.7.1.5 New look-and-feel of user interface

The Netrounds graphical user interface has had a facelift, with many improvements being introduced in the process.

Some highlights:

- Breadcrumb string introduced at top of window to enhance clarity of navigation.
- Plus-sign buttons for creating tests and monitors replaced with pop-up menu selection.
- The general visual appearance has been tweaked in various ways (wider errored second bars, new color scheme, new fonts and font sizes, and more).

13.7.2 Improvements

13.7.2.1 Export and import of templates

Test and monitor templates can be exported and reimported via the Netrounds REST and NETCONF & YANG APIs. This makes it easy to reuse a template in a different installation of Netrounds Control Center.
13.7.2.2 Tagging of SNMP traps

If you assign tags to a monitor, SNMP traps triggered for that monitor are now automatically assigned the same tags.

13.7.2.3 Tagging via REST and NETCONF

The full tagging functionality in Netrounds is now available also in the REST and NETCONF & YANG APIs. Tags can be applied to Test Agents, TWAMP reflectors, monitors, and monitor templates.

13.7.2.4 Tags included in NETCONF notifications

Similarly to the preceding item, NETCONF notifications regarding SLA violations now include the same tags as the monitor for which the violation occurred.

13.7.2.5 Monitor ID included in NETCONF monitor status report

When retrieving a status report for a monitor via NETCONF, the ID of the monitor is now included in the response.

13.7.2.6 Predictable interface renaming

The procedure of mapping MAC addresses to Test Agent interface names (eth0, etc.) has been improved so that it gives a predictable result.

13.7.3 Bug fixes

None

13.7.4 Removed features

None

13.8 Release notes, Netrounds software version 2.28.0

13.8.1 New features

13.8.1.1 New Path trace feature

Rerouting data in a network 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 most useful. Netrounds introduces a new such feature in version 2.28.0.

The Path trace tool continuously sends 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.
In the user interface, routes are visualized in a graph. You can select the time of any route change to view the resulting route, labeled with metrics such as delay and jitter for each hop.

A key property of the new Path trace tool is the continuous detection of network paths, which keeps the results up-to-date and relevant.

The Path trace tool supports both IPv4 and IPv6.

13.8.1.2 Up to 125 VLANs on a Test Agent

The number of VLANs that can be configured on a Test Agent has been greatly increased in this release, up to a maximum of 125. This capacity will be further extended in future releases.

13.8.1.3 TWAMP Light IPv6 support

TWAMP reflectors can be configured with IPv6 addresses, and TWAMP Light tests and monitors can be run over IPv6. Full TWAMP IPv6 support is planned for a future release.

13.8.1.4 Password strength

User passwords in Netrounds Control Center are checked against a set of requirements. In an on-premise installation it is now possible to switch from the default requirements to alternative, stricter ones which include a longer password string.

13.8.1.5 Stronger OpenVPN encryption for on-premise Control Center

When generating OpenVPN certificates for on-premise Netrounds Control Center installations, an RSA key length of 2048 bits is now used.

13.8.2 Improvements

13.8.2.1 Delay variation for Ping

Delay variation (jitter) has been added to the set of KPIs obtained with the Ping tool.

13.8.2.2 MTU configuration on Test Agent management interface

You can now configure the MTU (Maximum Transmission Unit) also on the management interface of a Test Agent. This can be done through the Control Center GUI or, for a virtual Test Agent, in the cloud-init configuration.

13.8.2.3 Maximum MTU size increased to 9216 bytes

The maximum MTU value allowed in Netrounds has been increased from 9000 to 9216 bytes.
13.8.3 Bug fixes

13.8.3.1 DHCP MTU handling

In version 2.27.0, a policy was introduced in the Netrounds Test Agents not to change the MTU of an interface based on information in DHCP or Router Advertisements. This caused an issue in some OpenStack environments, where the MTU is advertised as lower than the default 1500 bytes. The change has therefore been reverted.

13.8.4 Removed features

The old path trace function, available as an option in the Ping task type, has been removed.

13.9 Release notes, Netrounds software version 2.27.0

13.9.1 New features

13.9.1.1 IPv6 support for IPTV

IPTV can now be tested and monitored in an IPv6 network. Specifically, IPTV MPEG and channel zapping have been added to the range of testing and monitoring tasks that can be performed over an IPv6 connection. The IGMP channel zapping task (which was IPv4-specific) has been extended with support for the IPv6 MLD (Multicast Listener Discovery) protocol and renamed to the more generic “IPTV channel zapping”.

13.9.1.2 IPv6 supported over Test Agent management connection

Test Agents can now be managed from Netrounds Control Center over an IPv6 connection as well as an IPv4 one. This enables management of Test Agents in an all-IPv6 network.

13.9.1.3 REST: Full results for script-based tests

For script-based tests, such as transparency tests, only a pass/fail result has hitherto been retrievable via the REST API. In this version of Netrounds, you can get the full results of such tests with a REST GET command.

13.9.1.4 One-way delay estimation for TWAMP

The sender Test Agent can use timestamps from TWAMP packets to estimate one-way delay. This provides an alternative to synchronization with NTP when the latter is not available. The algorithm continually judges the reliability of its estimates.

13.9.2 Improvements

13.9.2.1 IPv6 support for NTP

Connecting to an NTP server over IPv6 is now supported.
13.9.2.2 Speedtest TCP info included in export

The detailed TCP metrics obtained when running Speedtest using WebSockets (those found on the TCP info tab) are now also included when exporting Speedtest results.

13.9.3 Bug fixes

13.9.3.1 Not possible to send alarms as SNMP traps

In version 2.26.0 it was not possible to configure alarms to be sent as SNMP traps to an SNMP manager without also having an alarm email list defined. The reason was that the alarm config validation erroneously required both an SNMP manager and an email list to be present.

13.9.3.2 Threshold settings in monitor alarm config lost on editing

When editing a monitor where an alarm had previously been configured, alarm threshold settings were lost. For example, Trigger alarm on “no data received” was unchecked even if previously checked.

13.9.4 Removed features

None

13.10 Release notes, Netrounds software version 2.26.0

13.10.1 New features

13.10.1.1 Virtual Test Agent deployment in Amazon Web Services and Microsoft Azure

Customers who run business-critical applications in AWS (Amazon Web Services) can now quality-assure these applications by deploying a Netrounds AMI (Amazon Machine Image) Test Agent in Amazon EC2 (Elastic Compute Cloud). Typical examples of such applications are SAP and SharePoint.

Similarly, in this release, Netrounds Test Agents can be deployed as a Windows VHD (Virtual Hard Disk) image in the Microsoft Azure cloud computing service, enabling the same range of testing and monitoring in that environment as well.

The features are added alongside the existing support for deploying Test Agents on other virtualization/cloud computing platforms such as VMware and OpenStack.

13.10.1.2 Connecting to vTAs over SSH

To ensure that it is possible to troubleshoot a vTA deployed on Amazon EC2 or Azure, it is necessary to set up a communication channel independent of the connection to the Netrounds Control Center. For this purpose, Netrounds supports adding an SSH key pair in the course of creating the vTA. Later on you can then connect to the vTA management IP address over SSH.
13.10.1.3 Alarms in REST and NETCONF & YANG APIs

Alarms and associated resources can now be managed through the REST and NETCONF & YANG APIs:

- When you create a monitor, you can associate an alarm with the monitor by including it in the monitor configuration.
- Alarm templates can be defined, and you can then point to such a template when setting up an alarm for a monitor. (Alternatively, you can supply the alarm configuration directly, without referring to a template.)
- It is also possible to configure SNMP managers and lists of email addresses to which alarms should be sent.

13.10.2 Improvements

13.10.2.1 Layer 2 transparency tests over Wi-Fi

It is now possible to run all transparency tests over Wi-Fi interfaces. Previously, Layer 2 transparency tests required wired interfaces and could not be run over Wi-Fi.

13.10.2.2 Speedtest over WebSocket

The Speedtest application now uses WebSocket by default rather than Adobe Flash. This release also introduces a new WebSocket-based implementation of Speedtest which has considerably improved performance compared to the old WebSocket-based Speedtest.

13.10.3 Bug fixes

In UDP tests, warnings caused by the sender (such as “Sender is overloaded”) and send failures due to network problems (such as no route to host) were shown for the wrong direction. For example, the message

```
ta2:eth1 (IPv4) (server) <-> ta3:eth1 (IPv4) (client): Sender is overloaded.
```

was shown when in fact ta2 was overloaded.

13.10.4 Removed features

None

13.11 Release notes, Netrounds software version 2.25.0

13.11.1 New features

13.11.1.1 Wi-Fi testing

Netrounds Test Agents have acquired the capability to connect to Wi-Fi networks and to run tests and monitors over such networks. This requires a Test Agent with a Wi-Fi Network Interface Card (NIC). A preinstalled Test Agent can be equipped with such a card; alternatively, if you are using your own x86 hardware, you need to purchase such a NIC separately.
Testing functionality is as follows:

- Scan for Wi-Fi networks.
- Wi-Fi logger: Set up a Wi-Fi connection and log Wi-Fi network parameters.
- Wi-Fi switcher: Change the Wi-Fi interface configuration while running a Wi-Fi logger task.

The following Wi-Fi standards are supported: IEEE 802.11g, 802.11n, and 802.11ac.

Please note that Wi-Fi result metrics are affected by many factors beyond Netrounds’ control, 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.

### 13.11.1.2 Test Agent kernel update

The Test Agent software has gone through a major revamping in this release:

- The processor architecture has been upgraded from 32-bit (i386) to 64-bit (amd64).
- The kernel has been upgraded to the latest Long Term Support version, 4.14.
- Debian has been upgraded to DebianStretch.

This helps support Test Agent installation on modern laptops with modern NICs and their drivers, and it also paves the way for a lot of future Test Agent development.

**Note:** This Test Agent upgrade is important and sensitive. It is vital not to unplug the Test Agent during the upgrade. Test Agents installed for example on a USB memory or in a virtual environment may have to be reinstalled (depends on the amount of memory allocated).

### 13.11.1.3 Re-launch of security tests

A subset of the security tests offered in certain previous Netrounds versions are re-launched in version 2.25. The following tests are kept in the “Security” category (renamed from “Access security”):

- DHCP starvation
- Fragmented DHCP packets
- Fragmented TCP/UDP headers
- Management protocol scanning
- Router redundancy protocol listening
- Routing protocols
- STP - Spanning Tree Protocol

The following tests, which relate to IPTV, have been moved to the “IPTV & OTT video” category:

- IGMP join/leave
- Multicast group limit

The remaining security tests previously offered have been removed (access security tests).
13.11.1.4 REST API: Detailed metrics now exportable for tests

In the REST API it is now possible to export detailed result metrics not only for monitors but also for tests. By default, averaged metrics taken over the full duration of the test are reported. Optionally, you can also export detailed (second-by-second) metrics for each task performed in the test.

(Previously, the outcome of tests was reported as “passed” or “failed” for each test step and for the test as a whole.)

13.11.1.5 More powerful TWAMP measurement: Up to 8800 streams @ 10 pps

Netrounds TWAMP testing has been enhanced up to a maximum of 8800 TWAMP streams at a rate of 10 packets per second.

This throughput rate requires HW Large or equivalent hardware. It also requires the use of four Test Agent interfaces as each interface can cope with a maximum of 2200 streams (the limiting factor is the number of hardware timestamps that each NIC can produce).

13.11.1.6 “Max rate” ES criterion added to TCP test/monitor

A “max TCP rate” erred second criterion has been added for the TCP test and monitor. Such a criterion is relevant to use for example if shaping is applied to the bandwidth used, so that the data rate should not exceed a given value. The feature can also be used to validate that a given total bandwidth is shared (sufficiently) equally among a number of TCP flows.

13.11.2 Improvements

13.11.2.1 Minimum rate lowered for TWAMP

The TWAMP tool is now capable of going down to a rate of 2 pps also when using 87-byte frames. (Previously this was not possible as a minimum rate threshold of 0.01 Mbit/s was applied.)

13.11.2.2 Test Agent images downloadable in QCOW2 and OVA formats

Test Agent images in QCOW2 and OVA formats can now be downloaded directly from the Netrounds webapp. The download dialog with its instructions has been improved generally.

13.11.3 Bug fixes

13.11.3.1 Problem using rate variables in templates

Rate variables (e.g. TWAMP rate, Y.1731 rate, UDP loopback rate) could not be used in test and monitor templates. – Such variables can now be used normally in templates.

13.11.3.2 Non-unique input variables in templates

Multiple identically named input variables were allowed in test and monitor templates. – Each input variable must now have a unique name.
13.11.3.3 Title bar and tabs did not display in Test Agent interface config dialog

In Chrome version 65.0.3325.162 (Official Build, 64-bit, Linux), for Test Agent management interfaces, the title bar and tab names in the interface configuration dialog did not display. – This glitch has been eliminated.

13.11.3.4 VoIP UDP task: “Expected DSCP” thresholds placed under wrong heading

In the configuration dialog for the VoIP UDP task, the thresholds for Up/Down expected DSCP wrongly appeared under the Advanced heading instead of under “Thresholds for errored seconds (ES)”. – The thresholds have been moved to their proper category.

13.11.3.5 Speedtest Start button not working

Start button on (Flash-based) public Speedtest page did not work in SaaS version 2.24.0. – This has been fixed.

13.11.3.6 DSCP remapping bug

DSCP remapping test passed (although it should fail) in the following circumstances: Expected DSCP set to 0; nonzero DSCP sent in a network that does not do any DSCP remapping.

13.11.3.7 REST API: Metrics could not be fetched for periodic tests

Attempts to fetch metrics for a monitor containing periodic tests failed with HTTP error 500.

13.11.3.8 Report generation failed for monitors with non-ASCII characters in their names

Clicking the Report button for a monitor having at least one non-ASCII character in its name resulted in a “Whoopsie daisy” error message.

13.11.3.9 Live remote packet capture failed

Live remote packet capture failed for one customer. – This was caused by a hidden feature for setting user name and password in the packet capture tool. To eliminate the problem, this hidden feature was removed.

13.11.3.10 TWAMP and UDP loopback received rate missing in reports

The received rate indicated in reports on UDP tests was missing in reports on TWAMP and UDP loopback. – This rate has now been added.

13.11.4 Removed features

• The access security tests have been removed from the webapp. See the section on security tests above (page 389).
• The Darkstat application has been removed from the webapp.
• Support for the “HW Small” preinstalled Test Agent has been terminated (it can no longer be updated after the release of version 2.24.0). The reason is that HW Small does not support the new 64-bit architecture. Customers should contact Netrounds sales at sales@netrounds.com for replacement of HW Small units.

13.12 Release notes, Netrounds software version 2.24.0

13.12.1 New features

13.12.1.1 BWPing: High-speed ICMP Ping for activation testing and troubleshooting

Alongside the existing Ping (page 293) measurement, this release introduces a new tool for high-speed ICMP Ping. Its primary purpose is to achieve high throughput in service activation testing and troubleshooting. BWPing is based on the open source BWPing tool and supports bandwidths up to about 30 Mbit/s per tool instance. Since Ping is more or less universally available in customer-premise equipment, the new feature is especially useful for service activation testing towards devices that lack TWAMP support. Multiple streams can be set up in order to test up to six different QoS classes concurrently.

13.12.1.2 Cisco compliant UDP Echo

To enable UDP Echo measurement towards Cisco routers, the UDP Echo frame format has been modified to comply with Cisco specifications. UDP Echo is available as an option in the Ping (page 293) task.

13.12.1.3 Enhanced TWAMP scalability

Each Netrounds Test Agent is now capable of generating up to 8800 concurrent streams towards TWAMP reflectors, and to collect measurements for these streams.

13.12.1.4 DSCP validation for TWAMP

Differentiated Services Code Point (DSCP) is a means of classifying and managing network traffic and of providing quality of service in modern Layer 3 IP networks. If the received DSCP differs from what is expected, this may imply that the service traffic is not prioritized according to expectations, which may impair service quality.

In this release the parameter Expected DSCP, previously available for all UDP tasks, has been added for TWAMP as well. Setting this parameter enables validation of DSCP values, both at instantiation of a service and throughout its lifetime. If the received and expected DSCP values do not agree, an Errored Second is triggered.

13.12.1.5 REST API refinements

• Retrieval of tests and monitors now exhibits the full configuration of each measurement task with all its parameter settings. To give just one example, for an HTTP test, the URL of the HTTP server is indicated. Previously, only the task name and type were obtained through the REST API.

• IPTV and SIP support: IPTV and SIP channels are now managed via REST as independent inventory items, just like TWAMP reflectors and Y.1731 MEPs. This means that IPTV and SIP channels no longer need to be hard-coded into test and monitor templates but can be defined as resources through the REST API and used as input to tests and monitors.
13.12.1.6 NETCONF & YANG refinements

- IPTV and SIP support: The YANG model has been extended to handle IPTV and SIP channels, so that these can be managed directly in NETCONF orchestration as well (compare the section on the REST API above (page 392)).

13.12.1.7 Test Agent IP and MAC addresses in reports

When generating a report on a test or monitor in the Netrounds web GUI, the IP and MAC addresses are now indicated for each Test Agent participating in the measurement.

13.12.1.8 Extra logging for troubleshooting of on-premise installations

In on-premise installations of Netrounds, additional debug logging can optionally be turned on in order to ease troubleshooting.

13.12.2 Bug fixes

13.12.2.1 DSCP remapping bug

DSCP remapping test passed (although it should fail) in the following circumstances: Expected DSCP set to 0; nonzero DSCP sent in a network that does not do any DSCP remapping.

13.12.3 Removed features

- The Test Agent Lite installation package for Windows has been removed from the Netrounds Control Center web GUI.
- The XML-RPC/JSON API is no longer supported. It is replaced by the REST and NETCONF & YANG APIs.

13.13 Release notes, Netrounds software version 2.23.2

13.13.1 New features

This release is a bug fix release and therefore does not contain any major new features or improvements.

13.13.2 Bug fixes

Fix for a memory leak in the protocol used for communicating with the Test Agents.

13.13.3 Removed features

None
13.14 Release notes, Netrounds software version 2.23.1

13.14.1 New features

This release is a bug fix release and therefore does not contain any major new features or improvements.

13.14.2 Bug fixes

Various minor bugs have been fixed.

13.14.3 Removed features

None

13.15 Release notes, Netrounds software version 2.23.0

This release targets the Netrounds SaaS solution as well as on-premise installations of Netrounds. The items that follow apply to both unless otherwise stated below.

13.15.1 New features

13.15.1.1 New measurement type: UDP loopback

In activation testing, it is often desirable to achieve very high data throughput. TWAMP would seem to be a natural choice for this kind of test, but in practice it is often incapable of attaining the required data rates because many routers handle the loopback of packets in the CPU, that is, in software.

In order to achieve very high data rates, it is necessary to have packets reflected in hardware. With such a procedure, however, the reflected packets will not be TWAMP compliant. Netrounds has therefore designed a new UDP loopback (page 306) measurement task, where UDP packets are pushed from a Test Agent (installed for example in a head-end) towards a router or other network device acting as reflector. The reflector device loops each UDP packet back to the Test Agent in hardware, thereby enabling data speeds on the order of 1 Gbit/s. Special configuration of the reflector device is required for this hardware loopback.

This feature is primarily intended for customers using Cisco equipment.

13.15.1.2 Retrieving performance metrics through Netrounds REST API

The data retrieval parts of the Netrounds REST API have been refined to allow retrieval of comprehensive monitor performance metrics, computed by Netrounds for successive time intervals.

The REST API functions used to extract performance metrics consist of GET operations applied to monitors. The full set of metrics for a monitor is obtained by getting it individually.

The time resolution of monitor performance metrics is user-configurable, the highest resolution being 10 seconds.
13.15.2 Improvements

13.15.2.1 GPS coordinates for Test Agents

It is now possible to indicate a Test Agent’s geographical coordinates as configurable properties. These are entered on a new tab GPS Location (page 103) in the Test Agent properties dialog. Coordinates can also be set and retrieved through the REST API. Supplying the GPS coordinates of Test Agents enables plotting of the devices in Google Maps and similar services.

13.15.2.2 Alarms by Text (SMS)

Alarms can be sent in SMS messages. This feature is purchased separately. SMS alarm delivery is configured by forwarding alarms in plain-text format to a special email address “<phone number>@sms.netrounds.com”, as explained here (page 42). This feature is available for SaaS customers only.

13.15.2.3 New colors in Speedtest graphs

The colors in Speedtest (page 338) graphs (both on the public page and in the Netrounds Control Center view) have been changed to shades of blue. This is to make the coloring more neutral and to prevent the interpretation of the colors as indicators of quality.

13.15.3 Bug fixes

None

13.15.4 Removed features

None

13.16 Release notes, Netrounds software version 2.22.3

13.16.1 New features

This release is a bug fix release and therefore does not contain any major new features or improvements.

13.16.2 Bug fixes

13.16.2.1 Migration failed in connection with 2.22.2 upgrade

A migration issue arose when upgrading to version 2.22.2, affecting customers who use the SIP feature in Netrounds. – This issue has been fixed.

13.16.3 Removed features

None
13.17 Release notes, Netrounds software version 2.22.2

13.17.1 New features

This release is a bug fix release and therefore does not contain any major new features or improvements.

13.17.2 Bug fixes

13.17.2.1 Issues with SIP testing when Test Agent did not register to a SIP server

SIP tasks where the “No” option was selected for Registration during test cycles had the following issues:

• DSCP was not set correctly. – This has been fixed.
• Clients could not set up calls to the hub, since the hub was not listening on the standard SIP port 5060.
  – Here the old behavior has been restored, so that the hub binds to port 5060 and clients make SIP calls directly to the IP address of the hub Test Agent on that port. This has the consequence that in “no registration” mode, only one instance of the SIP task can now be running on a given hub interface.

13.17.3 Removed features

None

13.18 Release notes, Netrounds software version 2.22.1

This release targets on-premise installations of Netrounds.

13.18.1 New features

13.18.1.1 Carrier-grade TWAMP

The TWAMP implementation has been entirely reworked and now offers:

• greatly improved scalability in terms of number of concurrent TWAMP streams supported
• vastly higher measurement accuracy thanks to the use of hardware timestamping. This requires a Test Agent NIC with hardware timestamping support. Even without this, however, there is a marked improvement in accuracy since a new and more sophisticated method of software timestamping has also been introduced for TWAMP. Specifically, user-space timestamps (taken in user applications) have been replaced by timestamps taken inside the Linux kernel. Full details are found here (page 288).

13.18.1.2 Orchestration through Netrounds REST API

In recent years, REST (REpresentational State Transfer) has emerged as the standard architectural design for web services and web APIs. In this release, a REST API is provided to simplify orchestration tasks. The REST API is provided as a Software Development Kit (SDK) together with extensive documentation and Python code examples. Automating your testing tasks in order to ensure a good end-user experience and solve problems faster has never been easier.

The REST API is a RESTful API that adheres to the standardized OpenAPI definition format, which makes it very simple to use.
Key operations that can be handled through the REST API include:

- Creating and deploying virtual Test Agents
- Updating Test Agent software
- Defining TWAMP reflectors and Y.1731 MEPs
- Creating tests and monitors from predefined templates
- Running tests and retrieving test results
- Starting and stopping monitors, and retrieving their SLA status

A guide to orchestration using the REST API is provided as a separate document. Rich documentation is also provided as an API browser.

### 13.18.1.3 DSCP validation

Differentiated Services Code Point (DSCP) is a means of classifying and managing network traffic and of providing quality of service in modern Layer 3 IP networks. It uses the 6-bit Differentiated Services (DS) field in the IP header to classify packets.

A common problem in today’s networks is DSCP bleaching, which means that the received DSCP differs from what is expected. This in turn may imply that the service traffic is not prioritized according to expectations, which may impair service quality.

A new parameter “Expected DSCP” has been added for all UDP tasks (UDP, Multicast UDP, VoIP UDP) to enable validation of DSCP values, both at instantiation of a service and throughout its lifetime. If the received and expected DSCP values do not agree, an Errored Second is triggered.

### 13.18.1.4 Import and export of test and monitor templates

To facilitate reuse of test and monitor templates across on-premise Netrounds Control Center installations, functions have been added for exporting templates from one Netrounds Control Center and import them into an account on another. The export and import are done from the command line using the `ncc` command:

```
ncc template export
```

```
ncc template import
```

To be exportable, templates must have general applicability. It is not possible to export templates with hard-coded resources such as specific Test Agents, Y.1731 MEPs, or IPTV channels.

### 13.18.1.5 Alarm for monitors that stop producing data

A new type of alarm has been introduced which is triggered when a monitor stops delivering data, for example because the Test Agent executing it has gone offline. New criteria for this have been added to the alarm template for monitors.

### 13.18.1.6 Test Agent deployment as Amazon AMI (beta)

Note: The AMI offered in this release is a beta version which is available for trials and proofs of concept.

Customers who run business-critical applications in AWS (Amazon Web Services) can now quality-assure these applications by deploying a Netrounds AMI (Amazon Machine Image) Test Agent in Amazon EC2 (Elastic Compute Cloud). SAP and SharePoint may be mentioned as representative examples of such applications.
An AMI is a special type of virtual appliance used to create a virtual machine within Amazon EC2, which is part of Amazon Web Services. The AMI serves as the basic unit of deployment for services delivered using EC2.

The Test Agent AMI is added alongside the existing support for deploying Test Agents on other virtualization/cloud computing platforms such as VMware, VirtualBox, and OpenStack.

13.18.2 Bug fixes

13.18.2.1 No validation of time window size when creating alarm templates

When creating an alarm template, it was possible to set the time window size to a value incompatible with errored second thresholds (i.e. to have a window shorter than the number of errored seconds required to raise an alarm). The validation was done only later, when applying the alarm template.

13.18.3 Removed features

None

13.19 Release notes, Netrounds software version 2.22.0

This was an internal release.

13.20 Release notes, Netrounds software version 2.21.0

13.20.1 New features

This release is primarily a bug fix release and therefore does not contain any major new features.

13.20.2 Improvements

13.20.2.1 NETCONF & YANG: Possible to start test on Test Agent that has never been online

Using the NETCONF API, it is now possible to create a Test Agent and a test at the same time, and to run the test on the Test Agent. This is an important use case for orchestration.

13.20.2.2 NETCONF & YANG: Improved error messages for start test RPC

Error messages for the RPC used to start a test in Netrounds have been made more informative.

13.20.2.3 Support documentation search function

A search function has been added in the in-app support documentation interface.
13.20.3 Bug fixes

13.20.3.1 Interoperability issues towards Huawei TWAMP reflectors

Running TWAMP Full towards a Huawei TWAMP reflector produced the Netrounds error message “Some aspect of the request is not supported”. – The reason was faulty packet padding in the TWAMP handshake.

13.20.4 Removed features

None

13.21 Release notes, Netrounds software version 2.20.0

13.21.1 New features

13.21.1.1 NETCONF & YANG API

Netrounds version 2.20.0 makes the NETCONF & YANG API generally available, and it has also been enriched with some new features. The NETCONF & YANG API allows you to integrate Netrounds Control Center with a network service orchestrator or NFV orchestrator, enabling automated network service activation, continuous monitoring, and remote troubleshooting in closed feedback loops. Driving these operational lifecycle tasks programmatically through an NFV or service orchestrator is what we refer to as the concept of Orchestrated Assurance.

Through the NETCONF & YANG API you can comprehensively orchestrate Netrounds operations – crucially, the following tasks (many others are available in the API):

- Test Agent actions. Virtual Test Agents can be created, configured, and deployed for measurement. There are also commands for rebooting Test Agents and for upgrading Test Agent software. In short, the API affords you full control of your system’s Test Agents.
- Importing TWAMP and Y.1731 reflectors. The TWAMP and Y.1731 protocols can be used to reflect test traffic off embedded reflectors, extending test coverage and visibility without additional investments.
- Setting up and running tests and monitors based on predefined templates, using Test Agents and reflectors. The use of predefined templates allows test and monitor templates to be created once and shared throughout your organization, promoting consistent processes and streamlined operations.
- Retrieving test results and monitoring session SLA status. These commands are used to retrieve the outcome of Netrounds measurements. Automated follow-up actions can be configured in the orchestrator in case tests fail or SLA criteria are not met.
- Getting Netrounds system status reports (indicating uptime, memory usage, load average, and software version). Such reports allow you to keep continuous track of Netrounds system health and performance.

Additional commands available with the NETCONF & YANG API include ones for:

- Making use of pre-existing Test Agents (i.e. ones not created through the API)
- Listing defined test and monitor templates
- Listing tests and monitors that are currently executing
- Getting notifications of status changes (to Test Agents, tests, monitoring sessions, etc.)

Recent additions to the NETCONF & YANG API are these commands:
• Start/stop already created monitor (new Netrounds feature)
• “Alarm state changed” notification
• “Use public address” option for Test Agents (found here (page 82) in the Netrounds GUI)
• DHCP client vendor id (Option 60) for Test Agents (found here (page 81) in the Netrounds GUI)
• Reboot Test Agent
• Update Test Agent software
• Get Netrounds system status (uptime, memory usage, load average, software version)

13.21.1.2 Licensing solution

As Netrounds Control Center is now shipped for on-premise installation in customer environments, a licensing solution has been introduced protecting the Control Center software. The licensing solution has the following properties:

• Controls which features customer can use
• Controls the time period for which the customer can use the server
• Prevents use of illegally copied and distributed Control Centers (piracy)

13.21.1.3 Test Agent repository

A Test Agent repository has been added to on-premise server to allow Test Agents to be updated/upgraded.

13.21.1.4 In-app support

Full support documentation is now provided within the Netrounds application. This applies to Netrounds managed in our public cloud as well as to on-premise Netrounds installations.

13.21.2 Improvements

13.21.2.1 Improved bulk import and management of TWAMP reflectors

• TWAMP reflectors can be tagged in Control Center for grouping or association with specific templates/monitors/Test Agents
• Tag field also included in CSV file format for TWAMP reflector import
• Lat/long coordinates included in CSV format (for future use)
• Existing reflectors no longer overwritten when importing new ones
• Export of TWAMP reflectors to CSV

See this page (page 29).
13.21.3 Bug fixes

13.21.3.1 Wrong default value for “Recursive requests” in DNS monitoring

During DNS monitoring the message “Response did not contain an answer” was obtained, although the DNS server was in fact responding to requests. – The reason was that “Recursive requests” was by default turned off although it should be turned on.

13.21.3.2 Erroneous “software too old” messages

When a Test Agent went offline during IPTV monitoring, it was erroneously reported that the Test Agent’s software version was too old.

13.21.3.3 Path MTU test hung

The Path MTU test occasionally failed to complete for no valid reason.

13.21.3.4 Errors in JSON-RPC API documentation

Many examples in the JSON-RPC API documentation contained an error in how Test Agent interfaces were specified.

13.21.4 Removed features

None

13.22 Release notes, Netrounds software version 2.19.0

13.22.1 New features

This release is the first on-premise installation of Netrounds.

It does not introduce new Netrounds features, and the Netrounds SaaS (public cloud) solution is not affected by this release.

13.23 Release notes, Netrounds software version 2.18.0

13.23.1 New features

13.23.1.1 Speedtest with WebSockets

The Speedtest (page 338) application can now alternatively be run using WebSockets instead of Flash. Speedtest with WebSockets has the following properties:

- Flash is replaced with standard javascript web APIs such as WebSockets and WebWorkers.
- The test traffic is sent over WebSockets, which allows for IPv6 and HTTP proxy support.
- Additional TCP metrics are collected by sampling the Linux kernel “tcp_info” struct on the Test Agent. These metrics are presented on a new tab TCP info on the test page.
• The test page is hosted at a different address than the Flash-based Speedtest: http://app.
netrounds.com/<account name>/speedtest-websocket.
• The test page is hosted over HTTP instead of HTTPS.

Speedtest with WebSockets currently has the following limitations:
• The test duration for upload and download tests is limited to 10 s. The number of TCP sessions during
these tests is limited to 8.
• There might be a big difference in performance between different browsers on the same system.

13.23.1.2 User interface enhancements

• The new Test Agent nomenclature has been adopted throughout the user interface:
  – Probe/Agent has been replaced by Test Agent, which covers both and corresponds to the older
  term Genalyzer.
  – Probe and Agent are distinguished as Test Agent Appliance and Test Agent Lite respectively
  where needed.
• Drop-down boxes with up to seven items have been replaced by toggle buttons, making selection
  quicker.
• The ability to filter by tag in the Monitoring section of the Dashboard has been restored.

13.23.1.3 Transparency test for new Ethernet protocols

A newly introduced test (page 255) checks transparency for the following Ethernet control protocols:
• LACP, Link Aggregation Control Protocol
• EAPoL, Extensible Authorization Protocol over LAN
• MVRP, Expected outcome for Multiple VLAN Registration Protocol
For each protocol to be tested, five frames are generated and sent, and a pass/fail result is obtained.

13.23.1.4 New Path MTU test

A new task type Path MTU discovery (page 272) has been added in the Transparency category. Further-
more, the RFC 6349 TCP throughput test (page 193) has been updated to use the new Path MTU discovery
algorithm in its first step.

13.23.2 Improvements

13.23.2.1 Sign-up page

The Netrounds sign-up page has been modified to achieve a smoother transition from the netrounds.com
website.
13.23.3  Bug fixes

13.23.3.1  IPTV crashes

The IPTV tool crashed if the same multicast group was used for several IPTV channels.

13.23.3.2  DVB-C

Unhelpful error message (“Unknown error”) when trying to run multiple concurrent DVB-C monitoring ses-
sions with channels on different frequencies (muxes). This is not supported since the hardware can only
tune to one frequency at a time. – An adequate error message is now shown when attempting the above.

13.23.4  Removed features

13.23.4.1  HW Small

On November 30, 2016 (End-of-sale date), the HW Small platform was withdrawn from the Netrounds of-
fering. Support for HW Small will end on November 30, 2017. If you are using this platform, please plan a
hardware swap before this date. More information on this will follow.

13.23.4.2  Support for old Web browser versions

Support for the following Web browser versions has been dropped:

• Internet Explorer 9

13.24  Release notes, Netrounds software version 2.17.0

13.24.1  New features

13.24.1.1  Revamped user interface

The Netrounds graphical user interface has been given a major overhaul, making it even more powerful and
easy to use. Some highlights:

• The menu bar on top has been replaced by a sidebar on the left, which is collapsible.
• You can create a new test or monitor by clicking a plus-sign button on the sidebar.
• The user interface adapts to the screen or window width to make the best use of the available space.
• The general visual appearance has been tweaked in various ways (color scheme, font sizes, etc.).

13.24.1.2  Favorites in test/monitor builder

In the views where you build and start your tests and monitors, a Favorites menu will be shown if you have
marked any of your templates as a favorite (done by clicking the template’s star icon under My Templates).
The Favorites menu gives you quicker access to the test building blocks that you use most frequently.
13.24.1.3 TWAMP control protocol support

Netrounds previously supported the TWAMP Light version of the TWAMP protocol (IETF RFC 5357), designed for measurement of two-way packet loss and delay. TWAMP Light does not require the TWAMP control protocol, which performs a handshake between initiator and reflector. Netrounds now adds support for “full-fledged” TWAMP including the control protocol. In configuring the reflectors to be used in your tests, you specify the TWAMP control port to engage on each reflector. You can still run TWAMP Light if you like; this is done by not specifying a control port.

13.24.1.4 Config-drive support in virtual Test Agents

The virtual Test Agent for virtual NFV environments has been updated to support mounting a configuration drive to retrieve user-data. This is an alternative to requesting user-data using HTTP Get (the AWS EC2 method).

It is now also possible to set a static IP address using cloud-init as a complement to DHCP. Further, support has been included for setting an NTP server and for connecting via an HTTP proxy.

13.24.2 Improvements

13.24.2.1 Improved security tests

Access security tests (page 312) have been improved so that they can be run also when the ISP is on a different IP network, that is, behind a Layer 3 hop. This will be the case if each customer is on a separate VLAN (i.e. has a separate Layer 2 connection).

13.24.2.2 L2 transparency – IPv6: Expected outcome per protocol

In the L2 transparency – IPv6 (page 261) test, the expected outcome (Pass/Drop) can now be specified for each protocol tested, just as in the “L2 transparency – Multicast” test.

13.24.2.3 L2 transparency – IPv6: Support for MLDv2

Multicast Listener Discovery (MLD) version 2 is now supported in the L2 transparency – IPv6 (page 261) test.

13.24.2.4 Configurable VLAN priority for TWAMP

VLAN priority, that is, the Priority Code Point (PCP) to be used in the VLAN header, has been made user-configurable for TWAMP (page 286) tests (just as for stream-based tests). There is no PCP/DSCP marking for control traffic, however.

13.24.2.5 IPv4/IPv6 filters for mobile interfaces

When selecting a mobile interface in the Mobile switcher (page 234) utility, new filtering checkboxes appear for IPv4 and IPv6. By default both of these are checked.
13.24.2.6 Improved API documentation

The Netrounds API documentation has been reworked and considerably extended, with many examples added to facilitate practical use of the API.

13.24.3 Bug fixes

13.24.3.1 TCP sender overload warning

For TCP, a sender overload warning was given when the send buffer filled up, although this can happen not only because of CPU overload but also because of TCP throttling (rate control). In the latter case, the warning was misleading. – This warning has been removed, since Netrounds cannot distinguish the two scenarios described and identify cases of actual overload.

13.24.3.2 L2 transparency – Multicast: Frame padding caused test to fail

The frame integrity check in the L2 transparency – Multicast (page 265) test failed due to extra padding (zeros) being inserted in the frames on the receiver side.

13.24.4 Removed features

13.24.4.1 Recurly support removed

The support for credit card payment through Recurly has been removed from the Netrounds web application. Under Account, the screens Subscription, Shipping address, Billing information, and Purchase history have therefore been removed.

Regarding payment options, please contact Netrounds sales at sales@netrounds.com.