

XenaManager User Manual

Release 102

Teledyne LeCroy Xena

Jun 03, 2025

TABLE OF CONTENTS

1 Overview **3**

2 Installation and Upgrade **5**
2.1 Installation . . . 5
2.2 Upgrade Chassis . . . 6

3 Getting Started **7**
3.1 Getting Started with Traffic Generation & Analysis . . . 7
3.1.1 Add Chassis . . . 7
3.1.2 Configure Module . . . 9
3.1.3 Add Port . . . 9
3.1.4 Configure Port . . . 9
3.1.5 Add Stream . . . 14
3.1.6 Configure Stream . . . 16
3.1.7 Set Up Simple Bidirectional Traffic . . . 18
3.1.8 Stream Scheduler . . . 18
3.1.9 Filters . . . 20
3.1.10 Capture . . . 21
3.1.11 Histogram . . . 22
3.1.12 Global Statistics . . . 23
3.1.13 Logging and Reporting . . . 25
3.1.14 Statistics Charting . . . 26
3.2 Getting Started with Network Impairment . . . 26
3.2.1 Configure Chimera Module . . . 27
3.2.2 Add Chimera Port . . . 27
3.2.3 Enable Impairment . . . 28
3.2.4 Configure Flow Filter . . . 28
3.2.5 Configure Flow Impairments . . . 29
3.2.6 Chimera Statistics . . . 29

4 General **33**
4.1 Chassis Management . . . 33
4.1.1 General . . . 33
4.1.2 Troubleshooting . . . 37
4.1.3 Software Maintenance Activation . . . 37
4.2 Chassis and Module Firmware Compatibility . . . 39

4.3 Chassis Time Synchronization . . . 39
4.3.1 Overview . . . 39
4.3.2 Service and License Control . . . 40
4.3.3 Test Module Configuration . . . 44
4.4 Testbed Management . . . 45
4.4.1 Configuration Hierarchy . . . 45
4.4.2 Setting Current Testbed . . . 46
4.4.3 Creating Testbed . . . 46
4.4.4 Adding and Removing Testbed Ports . . . 46
4.4.5 Editing Testbed . . . 47
4.4.6 Removing Testbed . . . 47
4.5 Resource Management . . . 47
4.5.1 Chassis Resources . . . 47
4.5.2 Reservation Mechanism . . . 48
4.5.3 Reservation Hierarchy . . . 48
4.5.4 Reserving Resource . . . 48
4.5.5 Releasing Resource . . . 48
4.5.6 Relinquish Resource . . . 49
4.5.7 Handling Multiple Resources . . . 49
4.6 Ribbon Menu . . . 49
4.6.1 Edit Menu . . . 49
4.6.2 Operations Menu . . . 50
4.6.3 View Menu . . . 52
4.6.4 Options Menu . . . 53
4.6.5 Tools Menu . . . 54
4.6.6 Minimizing Ribbon . . . 55
4.6.7 Additional Features . . . 55
4.7 Saving and Loading Configuration . . . 56
4.7.1 Working With Test Port . . . 56
4.7.2 Working With Test Module . . . 58
4.7.3 Working With Testbed . . . 58
4.8 UI Customization . . . 60
4.8.1 Docking Panels . . . 60
4.8.2 Docking Positions . . . 60
4.8.3 Moving Docking Panels . . . 61
4.8.4 Positioning Panels . . . 61
4.8.5 Floating Panels . . . 61
4.8.6 Hiding Panels . . . 61
4.8.7 Restoring Default Layout . . . 61
4.9 CLI Script Client . . . 62
4.10 Different Types Of Xena Configuration Files . . . 63

5 TG and L1 **65**
5.1 Overview . . . 65
5.1.1 General Purpose Tab Panels . . . 65
5.1.2 Selected Resource Panels . . . 65
5.1.3 Testbed Centric Panels . . . 66
5.2 Available Resources Panel . . . 67

5.2.1 Testbed Selector . . . 67
5.2.2 Resource Tree View . . . 67
5.2.3 Toolbar Options . . . 67
5.2.4 Show Only Used Ports . . . 67
5.2.5 Reserve Used Ports . . . 68
5.2.6 Release All My Resources . . . 68
5.2.7 Tree View Columns . . . 68
5.2.8 Multiple Selections . . . 68
5.2.9 Right-click Options . . . 68
5.3 Resource Properties . . . 68
5.3.1 Overview . . . 68
5.3.2 Chassis Properties . . . 69
5.3.3 Module Properties . . . 72
5.3.4 Port Properties . . . 77
5.3.5 Stream Properties . . . 154
5.4 Port Statistics . . . 179
5.4.1 Port Transmit Statistics . . . 179
5.4.2 Port Receive Statistics . . . 181
5.5 Port Configuration Grid . . . 184
5.5.1 Grid Operations . . . 184
5.6 Stream Configuration Grid . . . 185
5.6.1 Accessing the Packet Header Editor . . . 185
5.6.2 Grid Operations . . . 186
5.7 Global Statistics . . . 186
5.7.1 Overview . . . 186
5.7.2 Statistics Tabs . . . 187
5.7.3 Latency and Jitter . . . 192
5.8 Filters . . . 193
5.8.1 Overview . . . 193
5.8.2 Filter Details . . . 194
5.8.3 Using Filters . . . 194
5.9 Capture . . . 194
5.9.1 Overview . . . 195
5.9.2 Configuration . . . 195
5.10 Histograms . . . 197
5.10.1 Overview . . . 197
5.10.2 Configuration . . . 197
5.10.3 Histogram Results . . . 198
5.11 Gauge/Meter . . . 199
5.11.1 Overview . . . 199
5.11.2 Configuration . . . 199
5.11.3 Gauge Menu . . . 200
5.12 Event Log . . . 200
5.12.1 Event Types . . . 201
5.12.2 Event States . . . 201
5.12.3 Event Monitoring . . . 201
5.12.4 Event Columns . . . 202
5.12.5 Event Log Management . . . 202

5.13 Communication Trace . . . 202
5.13.1 Trace Format . . . 202
5.13.2 Available Columns . . . 202
5.13.3 Filtering . . . 203
5.13.4 Performance Impact . . . 203
5.14 Logging and Reporting . . . 203
5.14.1 Overview . . . 204
5.14.2 Controlling Logging State . . . 209
5.14.3 Output Formats . . . 209
5.15 Statistics Charting . . . 210
5.15.1 Overview . . . 211
5.15.2 Charting Details . . . 212
5.15.3 Chart Control . . . 212
5.15.4 Visual Aspects . . . 213
5.16 Stream Scheduler . . . 213
5.16.1 Overview . . . 214
5.17 Stream Wizard . . . 219
5.17.1 Overview . . . 219
5.17.2 Getting Started . . . 219
5.17.3 Detailed Information . . . 224
5.18 Replay PCAP File . . . 228
5.18.1 Comparing Replayed with Original PCAP . . . 229

6 Impairment 231
6.1 Overview . . . 231
6.1.1 CLI Script Interface . . . 231
6.1.2 Ethernet Packet Forwarding . . . 232
6.1.3 Chimera Impairment Pipeline . . . 232
6.2 Latency / Jitter Explained . . . 234
6.2.1 Extended Timing Mode . . . 234
6.2.2 Minimal Latency . . . 235
6.2.3 Maximum Latency (without packet loss) . . . 236
6.2.4 Maximum latency (Reduced Bandwidth Latency) . . . 236
6.2.5 Latency and Multiple Flows . . . 237
6.3 Module Properties . . . 237
6.3.1 SyncE . . . 237
6.3.2 Emulator Bypass . . . 239
6.4 Port Properties . . . 240
6.4.1 Reed-Solomon Forward Error Correction (RS-FEC) . . . 240
6.4.2 Test Payload (TPLD) Size . . . 241
6.4.3 FCS Error Mode . . . 242
6.4.4 Link Flap . . . 244
6.4.5 PMA Error Pulse Injection . . . 247
6.5 Packet Flow . . . 248
6.6 Flow Filter . . . 249
6.6.1 Overview . . . 249
6.6.2 Basic Mode . . . 252
6.6.3 Extended Mode . . . 264

6.7 Flow Impairment . . . 269
6.7.1 Overview . . . 269
6.7.2 Impairments Types . . . 275
6.7.3 Impairments Distributions . . . 283
6.7.4 Scheduler Functions . . . 300
6.8 Logging and Reporting . . . 302
6.9 Statistics Charting . . . 303

7 FAQ **305**
7.1 Broken Hardware . . . 305
7.1.1 Q: How to tell whether there is a problem with your hardware or if your module is broken? . . . 305
7.2 Manually Calibrate The Negative Delay Between Test Modules . . . 305
7.3 Traffic Capture . . . 305
7.3.1 Q: What is traffic capture? . . . 306
7.3.2 Q: Is Xena compatible with WireShark? . . . 306
7.3.3 Q: Are the 4-byte Ethernet FCS checksum field values included in pcap files? . . . 306
7.3.4 Q: How large is the packet Capture buffer? . . . 306
7.3.5 Q: Which packet protocols can the capture engine decode? . . . 307
7.3.6 Q: How do I start/stop the traffic capture function? . . . 307
7.3.7 Q: How can I define my packet capture criteria? . . . 307
7.4 Chassis and Software . . . 307
7.4.1 Q: Which OS does XenaManager run on? . . . 307
7.4.2 Q: Are Xena testers field upgradeable to future product releases? . . . 307
7.4.3 Q: How do I change the chassis IP address? . . . 307
7.4.4 Q: I have forgotten the chassis IP address. What do I do? . . . 307
7.4.5 Q: I have forgotten the password to my tester, how do I log on? . . . 308
7.4.6 Q: Why do the test port LEDs start blinking after power-on and reboot? 308
7.4.7 Q: Is the test module inside a XenaCompact identical to the XenaBay test modules? . . . 308
7.4.8 Q: Are transceiver plug-in modules hot-swappable? . . . 308
7.4.9 Q: Can I daisy-chain multiple XenaCompact testers into a single logical test unit? . . . 308
7.4.10 Q: Which TCP ports are used by the Xena testers? . . . 309
7.5 Test Source NAT Using XenaManager . . . 309
7.5.1 Testbed Topology . . . 309
7.5.2 Stream Configuration Using XenaManager For Source NAT Testing . 309
7.6 Port Operation . . . 312
7.6.1 Q: How can I start/stop traffic on all ports simultaneously? . . . 312
7.6.2 Q: Can multiple users reserve test ports on the same test module? . . . 314
7.6.3 Q: Can multiple users view the same port on a test module? . . . 314
7.6.4 Q: How can I move a port configuration between ports? . . . 314
7.6.5 Q: Each time I restart traffic on a port, is the generated packet pattern 100% identical? . . . 314
7.6.6 Q: Does a test port send any training packets? . . . 314
7.6.7 Q: Can I view the port configuration of ports which are reserved by another user? . . . 314

7.6.8 Q: Can I reserve a test port even when it is reserved by another user? . 314
7.6.9 Q: What does it mean that the port property ppm speed-reduction is emulated? . . . 315
7.6.10 Q: Which packets are looped when a port is in Rx-to-Tx loop mode? . 315
7.6.11 Q: Which packets are looped when a port is in Port-to-Port (inline) loop mode? . . . 315
7.7 Scripting & Automated Testing . . . 315
7.7.1 Q: Which features are available via scripting? . . . 315
7.7.2 Q: Where can I find the documentation of the Xena API scripting commands? . . . 316
7.7.3 Q: Is there a Tcl, Perl, Java or API available? . . . 316
7.7.4 Q: Can I run Xena2544 against a device which performs Network Address Translation (NAT)? . . . 316
7.7.5 Q: Are there any wizards for setting up test patterns, e.g. a full mesh traffic load on an n-port DUT? . . . 316

8 Glossary of Terms **317**

Index **323**

Release
102

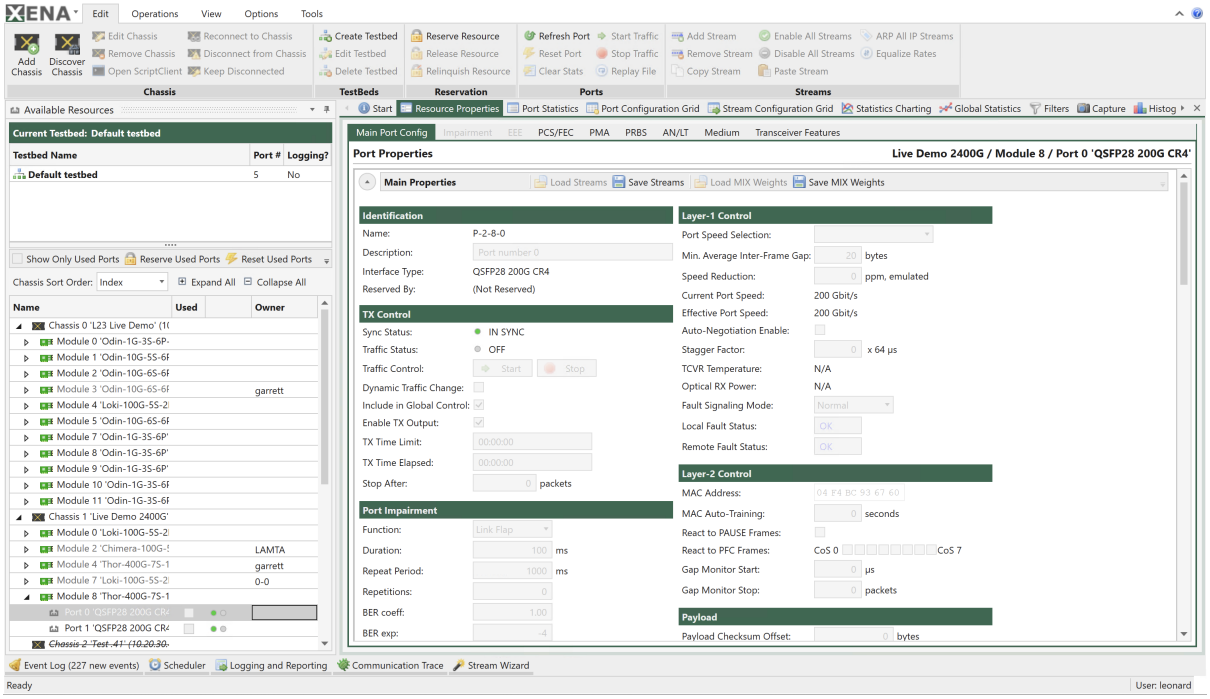
Version
102

Date
Jun 03, 2025

CHAPTER

ONE

OVERVIEW



XenaManager serves as the primary software application utilized for the management and configuration of Xena Networks' Ethernet testing products. This software facilitates the establishment of connections to one or more testers by utilizing their respective IP addresses, providing an extensive point-and-click user interface for configuring and operating these testers.

XenaManager is instrumental in configuring and generating streams of Ethernet traffic, supporting speeds of up to 800 Gbps, between Xena test equipment and Devices Under Test (DUTs), while also facilitating result analysis. Its user-friendly interface makes it a valuable tool for conducting a wide range of core test scenarios, catering to the needs of semiconductor and Network Equipment Manufacturers (NEMs), network service providers, and hyperscalers.

Key features of XenaManager include:

- Traffic Generation (*TG*) for Ethernet speeds of up to 800 Gbps, suitable for both functional and performance testing.
- Comprehensive Layer-1 testing capabilities, encompassing *PCS*/*FEC*, *PMA*, *PRBS*, *AN* & *LT*, Signal Integrity, Error Injection, and more.

- Network impairment functionalities, allowing users to introduce conditions such as drop, latency/jitter, corruption, duplication, misordering, shaper, and policer.
- Simple and efficient management of ports and traffic streams, customizable to replicate real-world scenarios.
- Packet editor that supports both protocol and byte-level packet definitions.
- Automatic protocol decoding of incoming packets.
- Support for multiple Xena chassis, which can be shared by multiple users across different subnets.

In summary, XenaManager is a versatile and user-friendly software application that plays a crucial role in configuring, testing, and analyzing Ethernet networks, making it an indispensable tool for a variety of industries and use cases.

CHAPTER

TWO

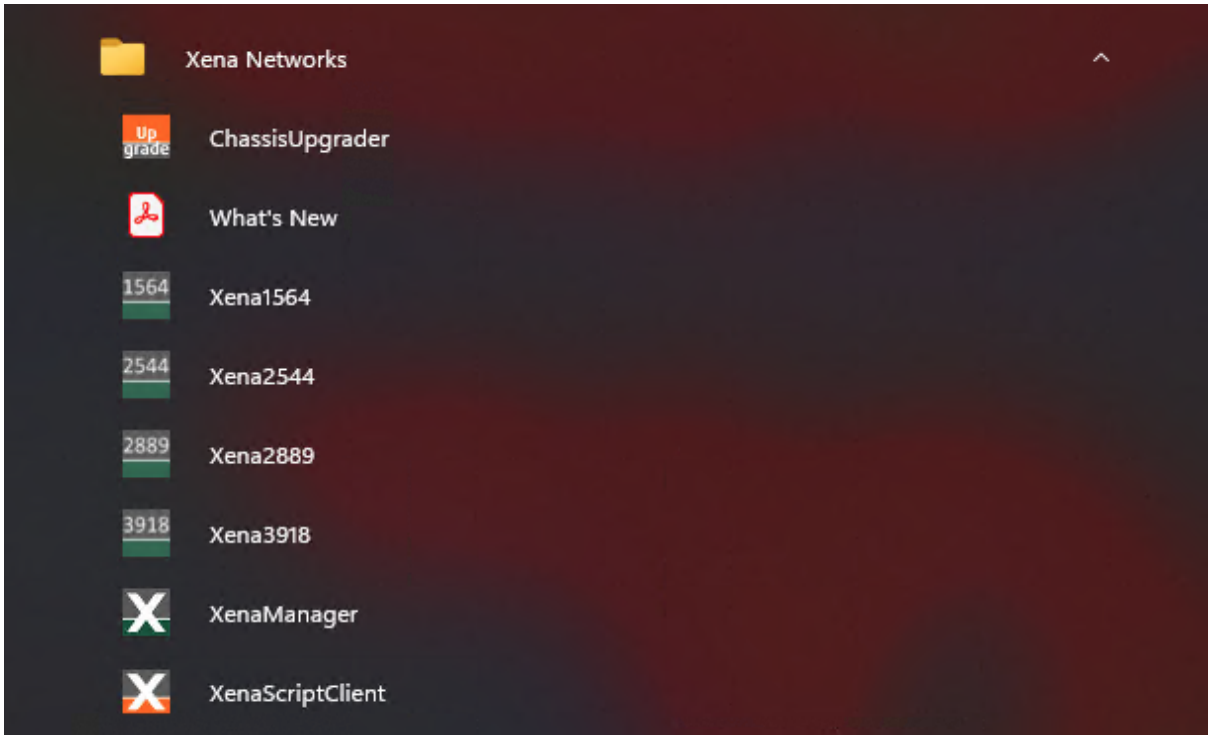
INSTALLATION AND UPGRADE

2.1 Installation

XenaManager is a Windows desktop application compatible with Windows 10 and later versions.

It comes pre-installed as an integral component of the standard Xena software release package, which is available for download from Software Downloads.

Following the installation, you can locate a shortcut to the application in the *Start* → *Programs* → *Xena Network* menu, and additionally (if you opted for this during the setup process), on your desktop.



2.2 Upgrade Chassis

Xena ChassisUpgrader Manual

Xena ChassisUpgrader is installed as part of a Xena Software release and is linked to that release. It is thus not possible for the ChassisUpgrader to install a different release than the one it has been installed as part of.

CHAPTER

THREE

GETTING STARTED

3.1 Getting Started with Traffic Generation & Analysis

This section is designed to help you begin using XenaManager to set up a simple bi-directional layer-2 Ethernet switching test scenario.

3.1.1 Add Chassis

A testbed is essentially a set of ports that you are actively utilizing. Certain panels within XenaManager will exclusively display data for ports that are part of your ongoing testbed. This encompasses both the *Port Configuration Grid*, *Stream Configuration Grid*, as well as *Global Statistics*.

1. Press the *Add Chassis* button located to the left in the ribbon bar at the top of the application.

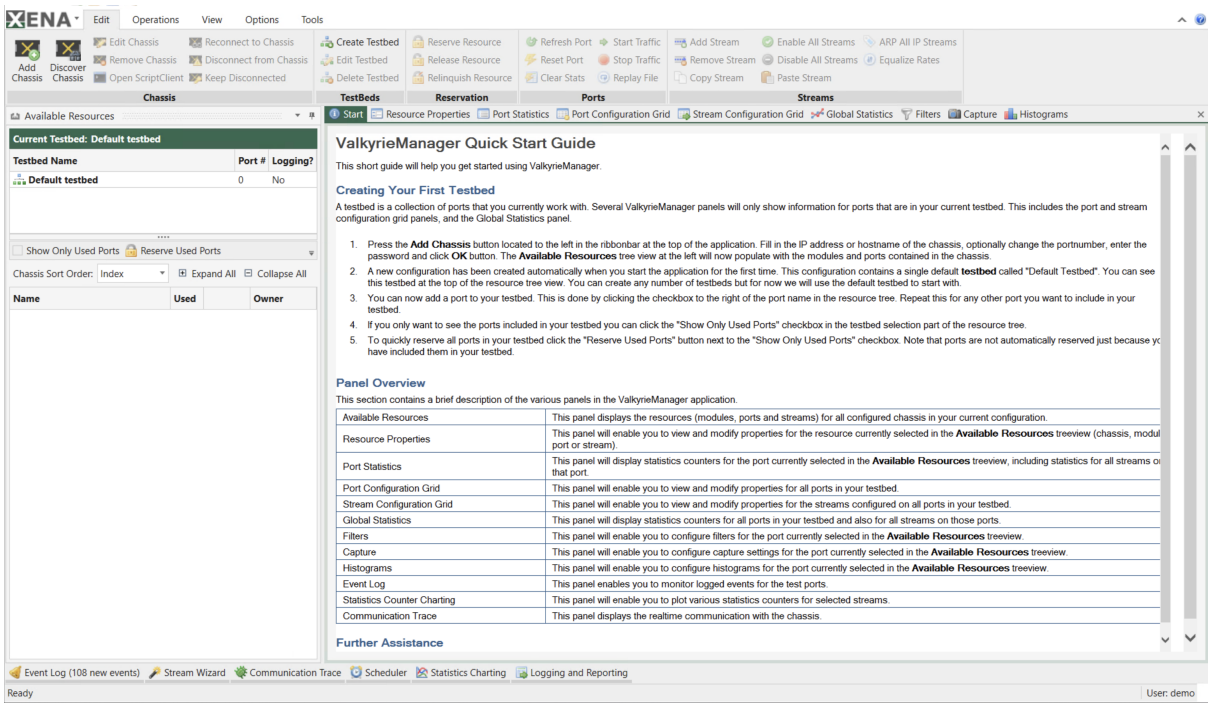


Fig. 3.1: Add chassis

2. Fill in the IP address or hostname of the chassis, optionally change the Chassis Port Number, enter the password (default is xena) and click *OK* button. The *Available Resources* tree view at the left will now populate with the modules and ports contained in the chassis.

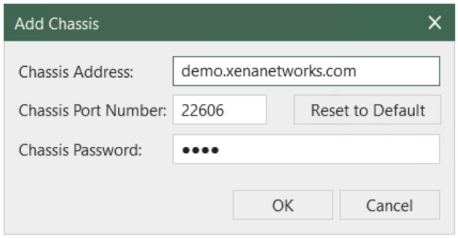


Fig. 3.2: Enter IP address, port number and password to add chassis

3. A new configuration has been created automatically when you start the application for the first time. This configuration contains a single default testbed called *Default Testbed*. You can see this testbed at the top of the resource tree view. You can create any number of testbeds but for now we will use the default testbed to start with.

4. Use *Options* → *Set Username* to indicate who owns the port reservation.

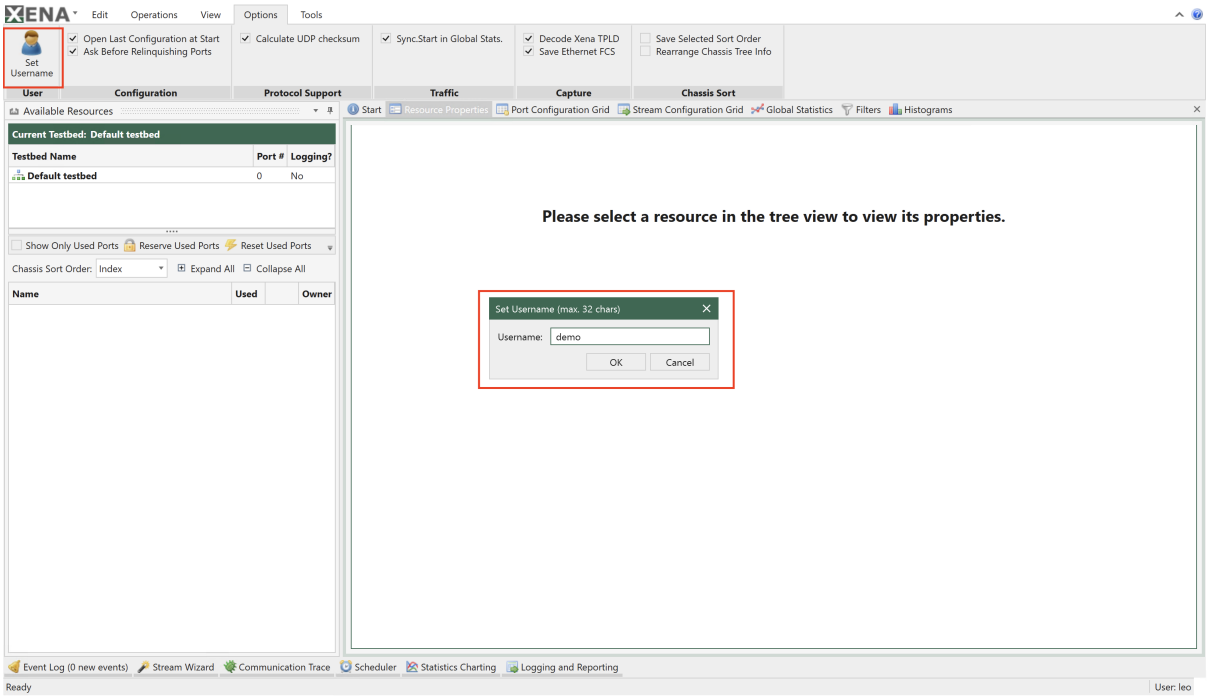


Fig. 3.3: Set username

5. You can now add ports to your testbed. This is done by clicking the checkbox in the *Used* column to the right of the port name in the resource tree. Add two ports to your testbed in this way. Please ensure that the two ports are connected through a standard layer-2 switch. 5. If you only want to see the ports included in your testbed you can click the *Show Only Used Ports* checkbox in the testbed selection part of the resource tree. 6. To quickly reserve all ports in your testbed click the *Reserve Used Ports* button next to the *Show Only Used Ports* checkbox.

3.1.2 Configure Module

1. Reserve a module, and go to *Resource Properties* → *Main Module Config* to configure module media configuration and port configuration.

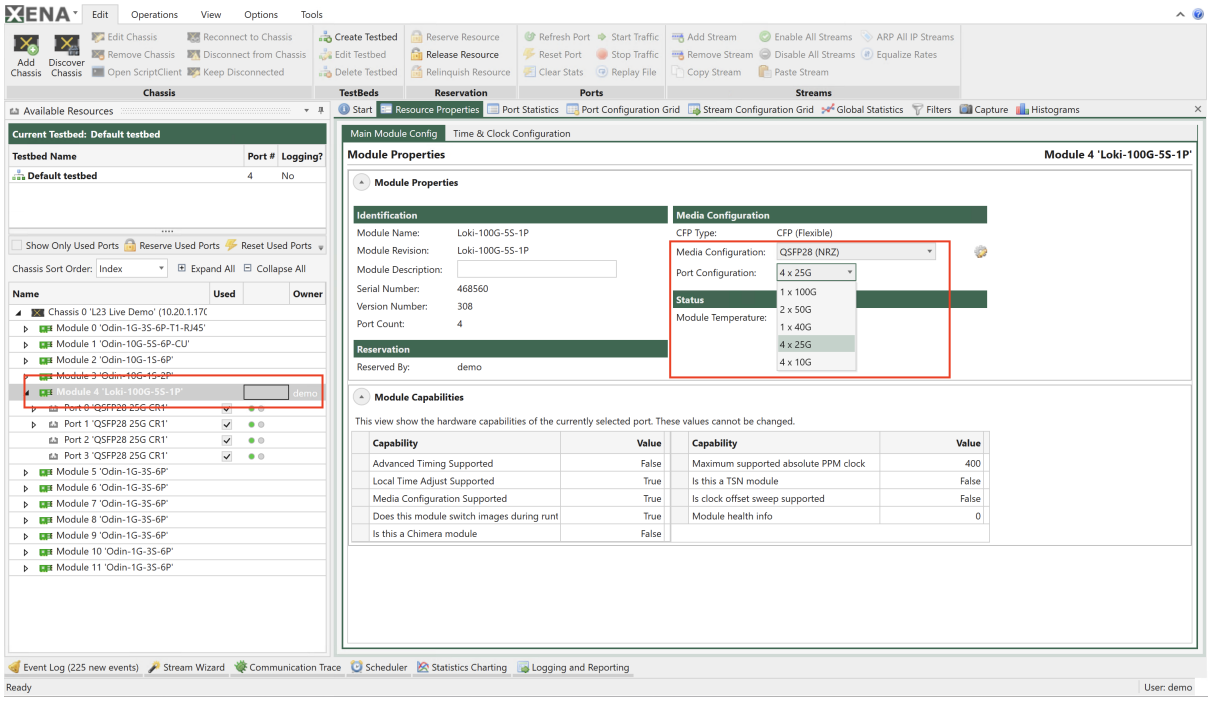


Fig. 3.4: Module media configuration and port configuration

2. Configure the required clock configuration from *Resource Properties* → *Time & Clock Configuration*

> **Note**
>
> Note that ports are not automatically reserved just because you have included them in your testbed.

3.1.3 Add Port

1. Select the port(s) you want to use.
2. Click *Reserve Used Ports*, check *Show Only Used Ports*

3.1.4 Configure Port

Select the port(s) to configure and click *Resource Properties* tab. Configure the following port parameters to accommodate your test.

1. Min. Average Inter-Frame Gap

- Set to 20 –> 12B (Minimum allowed by Ethernet at 100% load) +8B Preamble
- Can be set to 16B to achieve >100% load for port pressure testing

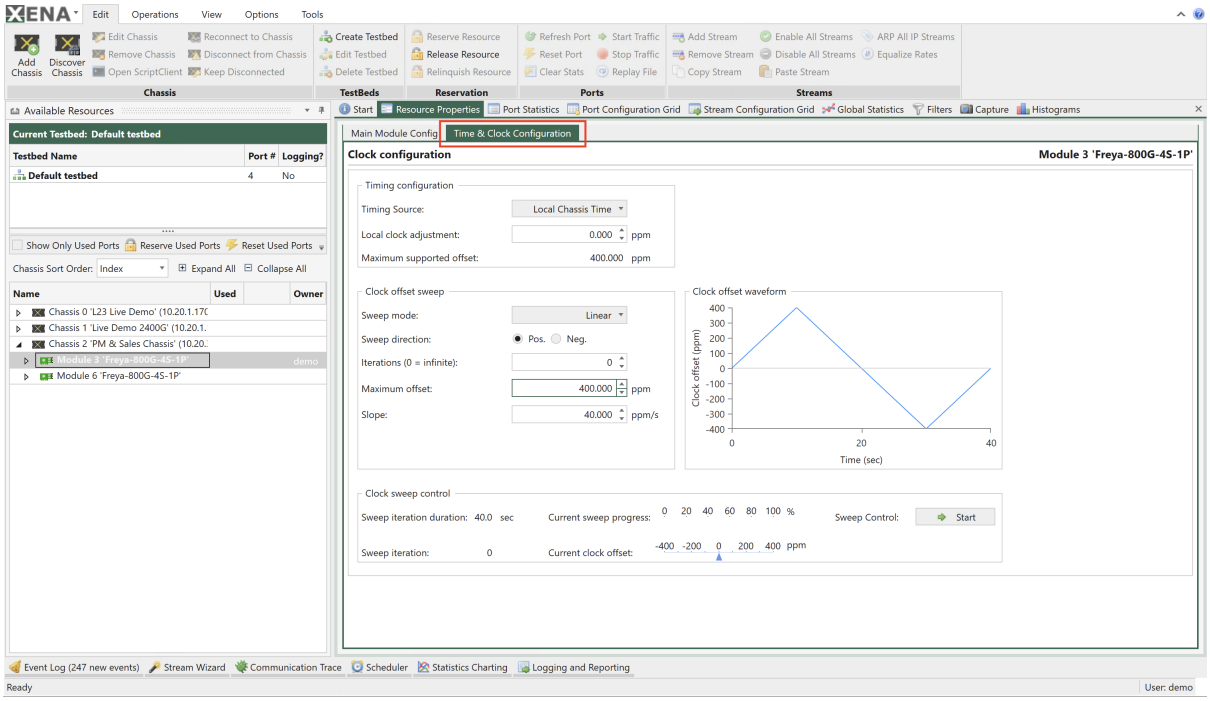


Fig. 3.5: Module media configuration and port configuration

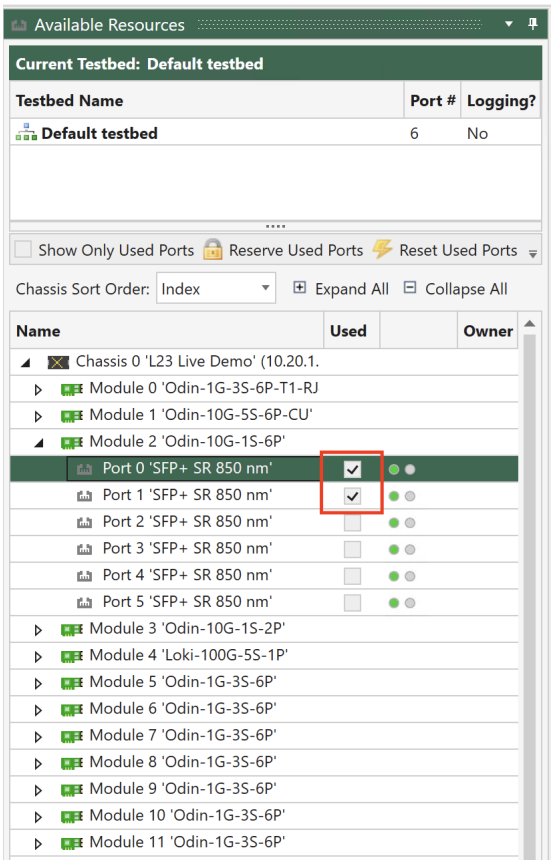


Fig. 3.6: Select ports

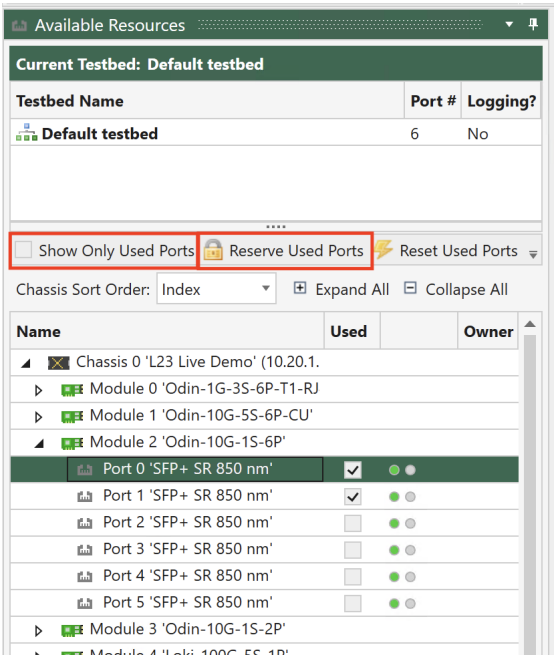


Fig. 3.7: Reserve used ports and show only used ports

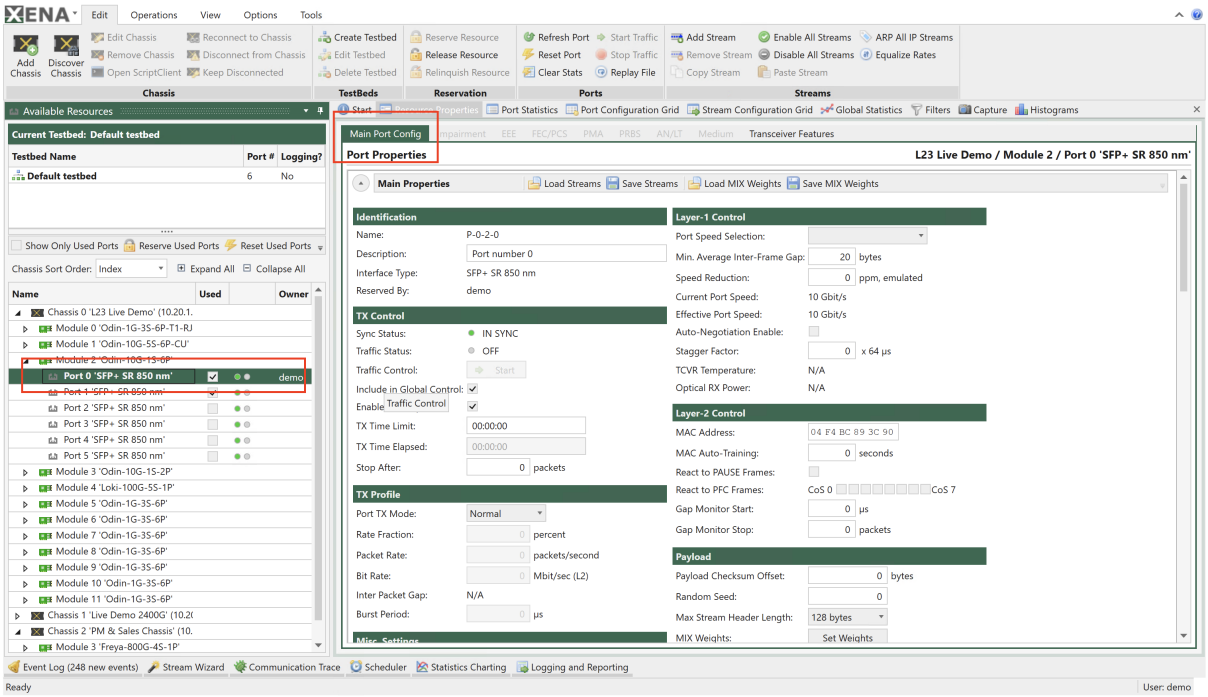


Fig. 3.8: Configure ports

- Values range between 16B to 20B depending on module.

2. MAC Address
   - Used as default SRC.MAC for each stream
   - Used when sending Ping or replying to ARP
3. MAC Auto-Training
   - Used to train DUT with Xena MAC so stream won't be flooded
4. React to PAUSE Frames
   - This means enable Flow Control on this port
5. Gap Monitor
   - Used to monitor(time) the disruptions of service to traffic
   - Gap Monitor Start: After how many μs would the Monitor start
   - Gap Monitor Stop: After how many packets would Monitor stop
   - Results can be seen in port statistics (type of GAP can only be set)
6. Payload Checksum Offset
   - Used to enter a Headers + Payload Data Integrity Checksum
   - Should start from offset 14 for pure L2 packets
   - Should start after IP offset for L3 and beyond packets (because of TTL)
7. Max. Stream Header Length
   - When user wants to set headers larger than 128
   - Number of streams will be downsized to 1/2
8. Loopback Mode
   - Off: Traffic flows naturally out of the port
   - L1 RX to TX: Any received packet is bounced back through TX
   - L2 RX to TX: Same as 8.2 yet it also swaps MAC SRC<>DST
   - L3 RX to TX: Same as 8.3 yet it also swaps IP SRC<>DST
   - TX(on) to RX: Packet goes out of TX but also internally direct to RX
   - TX(off) to RX: Packet goes directly to RX (No link sync needed)
   - Port to port: Any received packet goes out through the neighbor port
9. Latency offset
   - Used to automatically eliminate transceiver + cable latency
   - Set either manually or via Port Statistics *calibrate* button
10. IPv4

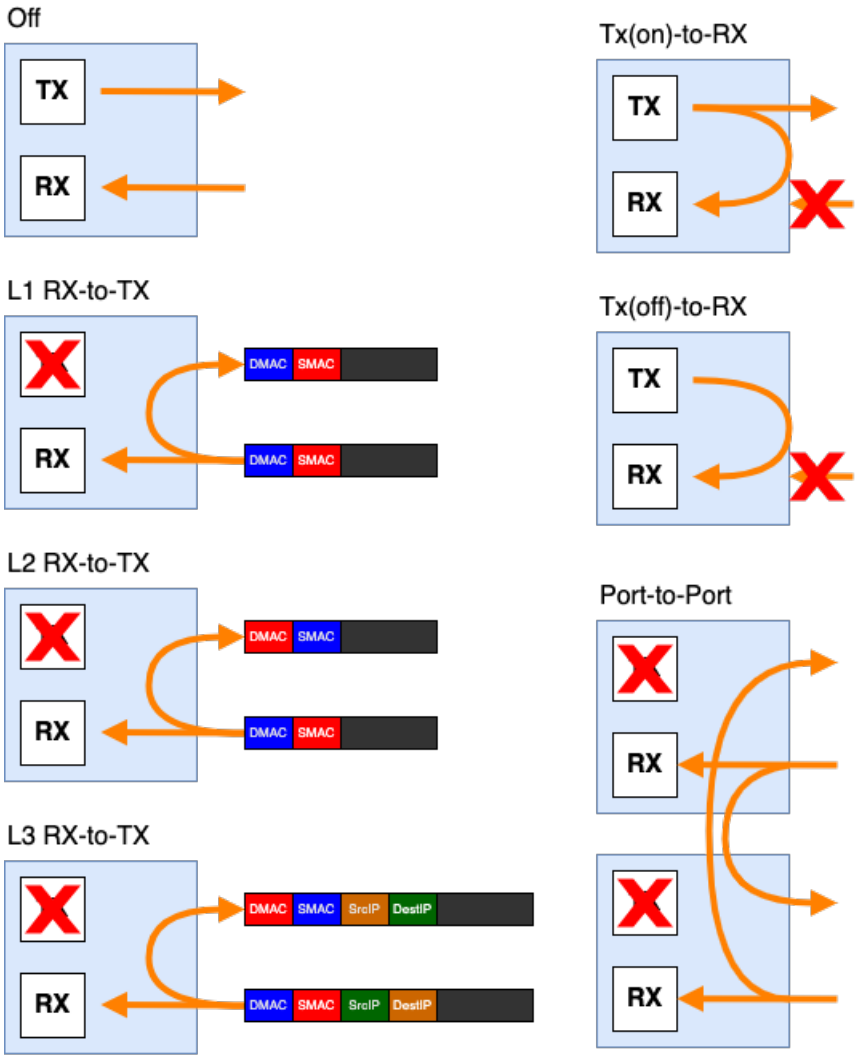


Fig. 3.9: Loopback mode

- Address/Subnet/Gateway used for PING and ARP functionality

11. Reply to ARP/PING Requests
    - Enable port's ability to reply to incoming ARP/PINGrequests
12. ARP and PINGv4 address Wildcard:
    - Used to enable multi unique ARP/PING requests

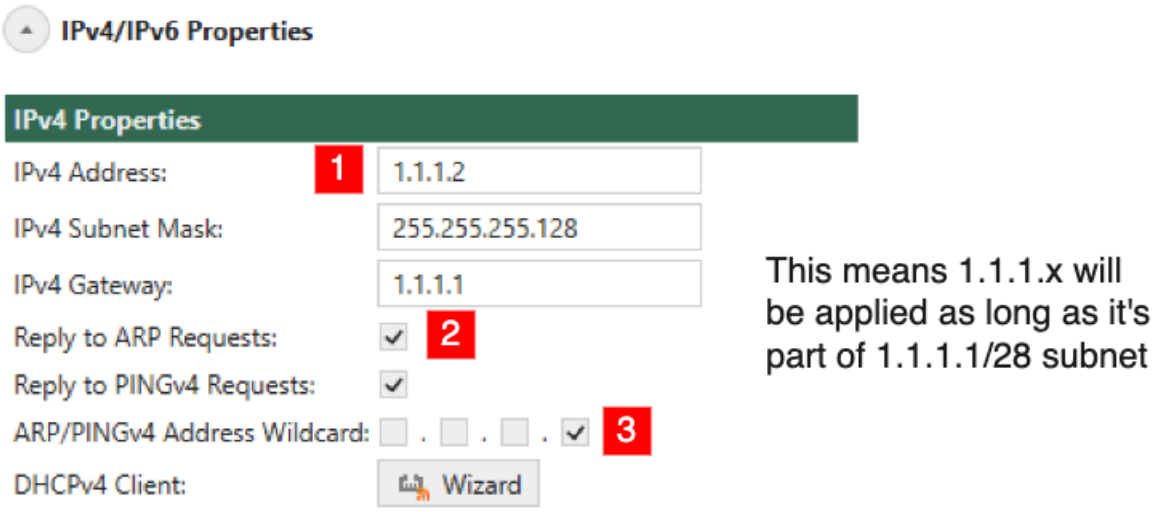


Fig. 3.10: IP settings

Some module supports the following.

13. Port Impairment
    - Link Flap: Set Duration, Repeat Period and Repetitions (0=continuous)
    - PMA Errors: Set BER coeff and BER exp for the error insertion
14. Payload Mode
    - Extended Payload
    - Custom Data Field

See also

See application note Freely Programmable Test Packets (Custom Data Fields) for details.

3.1.5 Add Stream

1. To add a stream, click *Edit Menu* → *Add Stream* or right-click port and choose *Add Stream*, to add multiple streams select the *Add Multiple Streams* option.
2. Copy Stream feature can also be used when user right clicks on any stream and then select copy stream and paste it anywhere.

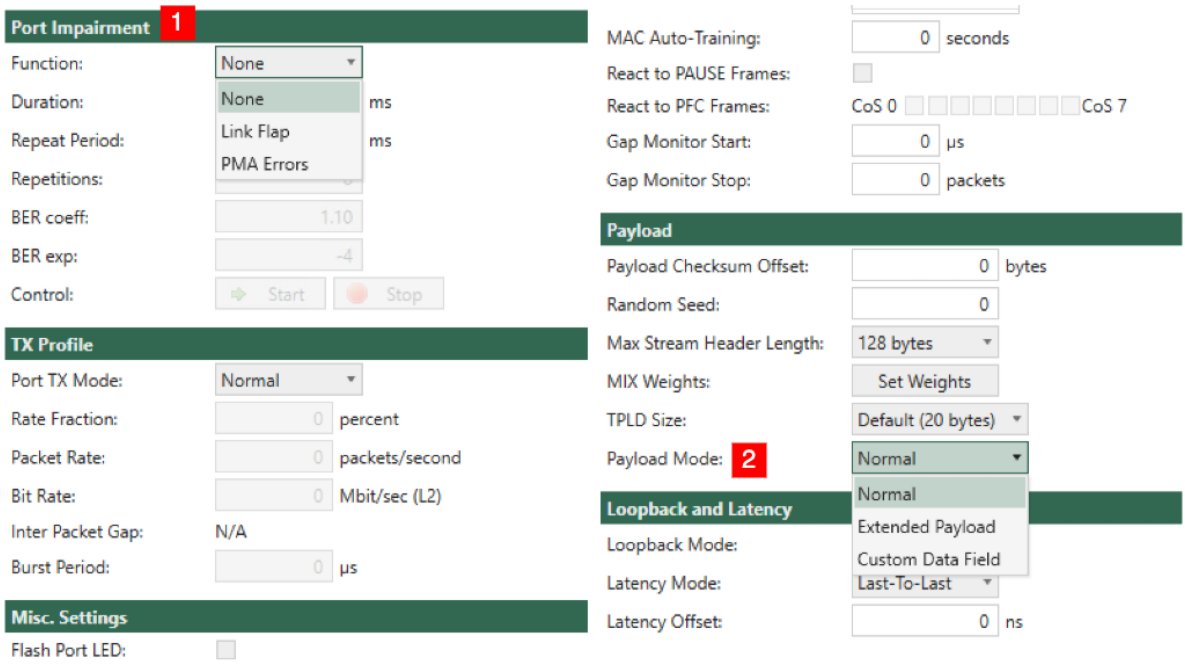


Fig. 3.11: Configure port impairment

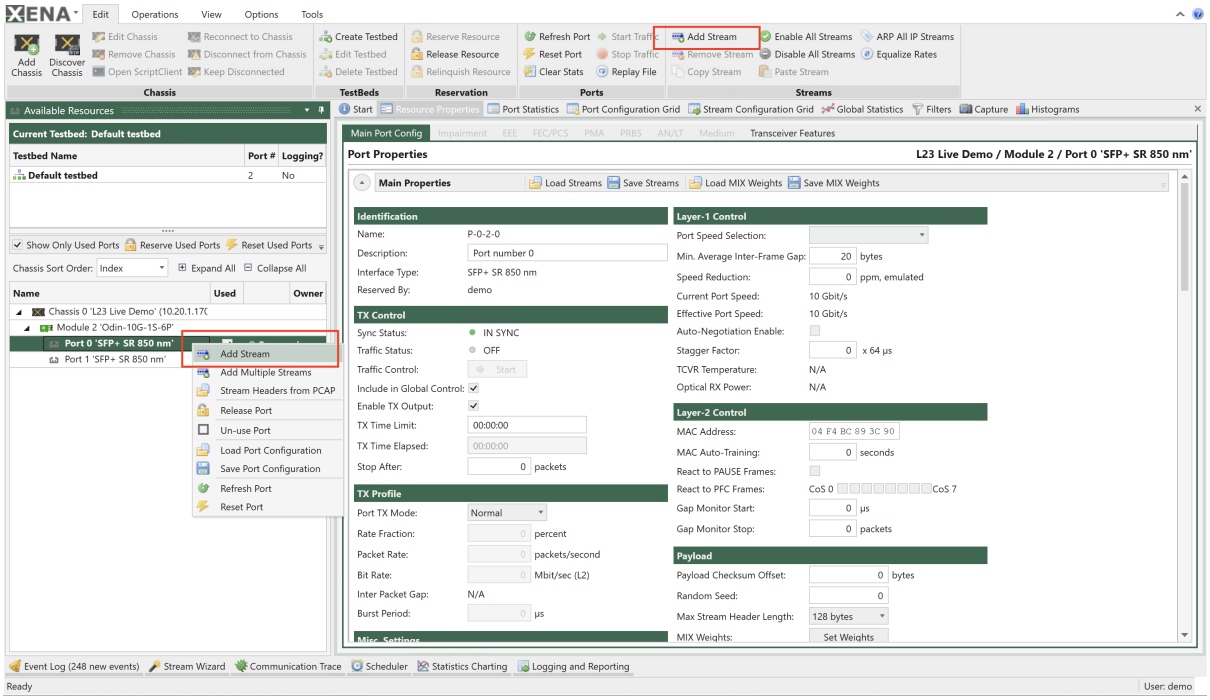


Fig. 3.12: Add stream

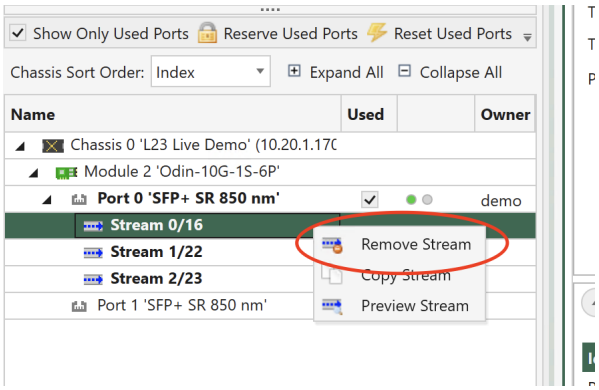


Fig. 3.13: Copy stream

3.1.6 Configure Stream

Select the new stream on the first port and ensure that you have selected the *Resource Properties* panel. The panel will now display the properties for the stream.

1. Insert test payload, *TID*: This is the stream ID used to identify Latency/Jitter/Packet Loss.
2. Description: Stream Description text (e.g. "Upstream connected to DUT Port 11")
3. Stream State
   - Disabled: Stream is not started when traffic is ON nor is it included in port rate usage.
   - Suppressed: Stream is not started when traffic is ON, but it is included in port rate usage. ("Paused State") can be switched to enabled on the fly.
   - Enabled: Stream is started when traffic is ON.
4. Stop After: Send specific number of packets and stop traffic. Also used in sequential mode as stream packet quantity.
5. Stream Transmission Profile:
   - Percent is L1 rate including IFG + Preamble.
   - Configuring on field actually changes all the others accordingly.
   - Grayed text can be edited. To have it set you need one more click.
6. Burst used to configure bursty traffic. Density sets the inner IFG inside the burst. There is a trade-off between the stream rate and the Burst rate.

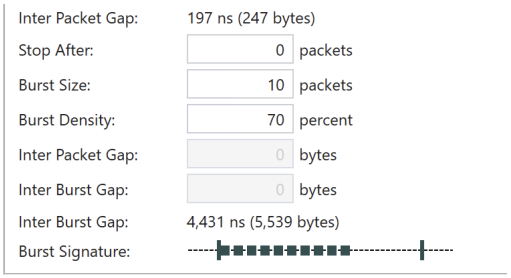


Fig. 3.14: Burst profile

7. Error Injection: Can send specific errors on the fly - but only when traffic is ON.
8. Insert Frame Checksum, *FCS*: Unchecking this checkbox will cause error frames.
9. Packet Length:

- Fixed - for min value =x all packets will be x
- Incrementing - for min value =100 and max value=200 e.g. *100, 101, 102, 103, . . . ,197, 198, 199, 200, 100, 101, 102*
- Butterfly - for min=100 max=200 e.g. **100**, 200, **101**, 199, **102**, 198, **103**, 197, **104**, 196, **105** . . .
- Random - random values between min. and max.
- Mix - sends internet mixture of packet sizes. For some modules 4 packet sizes are programmable. If not supported the programming boxes are dimmed.

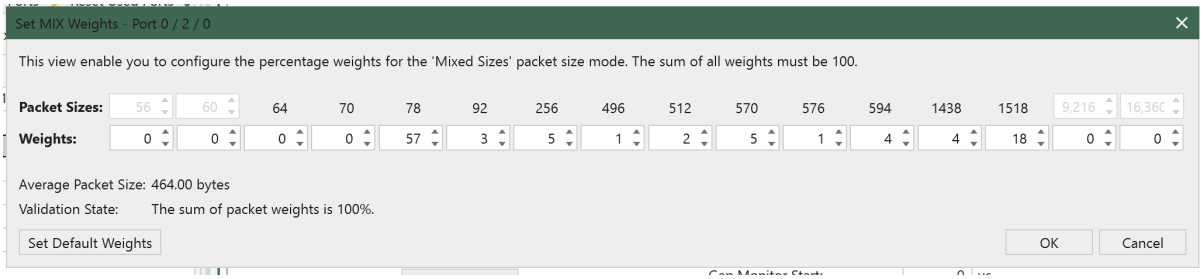


Fig. 3.15: MIX Weights sets can be loaded saved via the port resource properties

10. Payload Type:

- Incrementing 8-bits - means *00 01 02 03 04 05 . . .* Provides built in payload integrity check for payload.
- PRBS-31 - provides Pseudo Random Bit Sequence of `2^31-1` pattern. Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports.
- Random - provides Random bit Sequence pattern. Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports.
- Pattern - you can set your own custom pattern. Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports.
- Decrementing 8-bit: means *FF FE FD FC FB FA . . .* Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports.
- Incrementing 16-bit: means *00 00 00 01 00 02 00 03 00 04 00 05 . . .* Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports.
- Decrementing 16-bit: means *FF FF FF FE FF FD FF FC FF FB FF FA. . .* Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports.

Important

When using Incrementing 16-bit or Decrementing 16-bit, you need to check the option *Payload Start From 0* to have the payload *00 00 00 01 00 02 00 03 00 04 00 05 ...* or *FF FF FF FE FF FD FF FC FF FB FF FA....*

11. Scroll down to the *Packet Header Definitions* section in the stream properties view. Here you will find a Wireshark-like protocol header editor which allows you to define the protocol headers for the stream. Expand the Ethernet segment to view the fields in the segment.

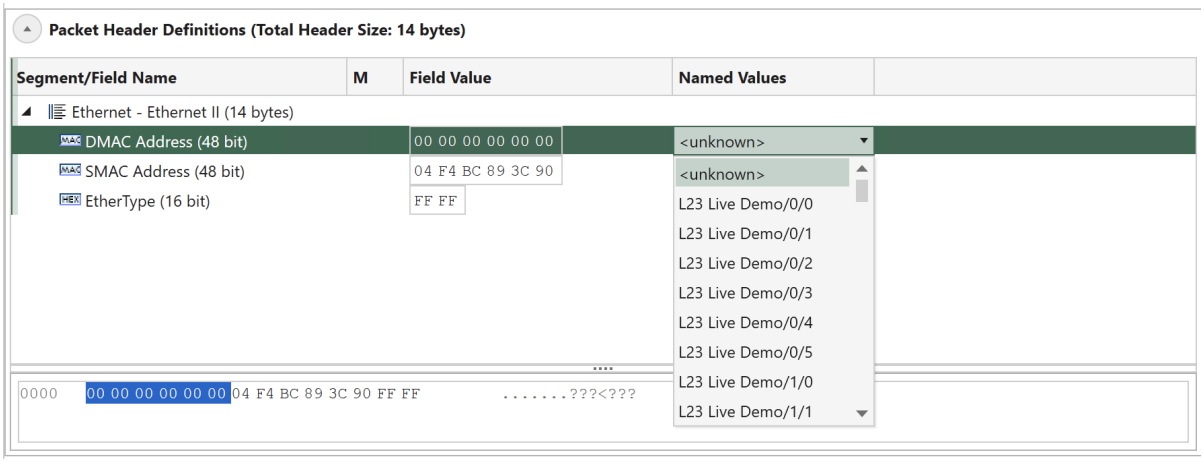


Fig. 3.16: Packet header definition

Note

Note that the Src MAC Address field has been automatically set to the MAC address of the containing port. Expand the dropdown-box in the *Named Values* column for the Dst MAC Address and locate the other port in your testbed. Note that the Raw Value column is also automatically updated with the MAC address of the peer port.

3.1.7 Set Up Simple Bidirectional Traffic

1. Add 1 stream for each traffic port (right-click and select *Add Stream*)
2. Select both streams using the *Available Resources* panel and CTRL.
3. Clicking *Pair Streams* results in:

3.1.8 Stream Scheduler

The Stream Scheduler can be used to build a series of actions (operations) based on existing streams in the current testbed.

Before starting the Stream Scheduler, you must reserve ports and configure ports and streams. Example: 120 times the traffic is running for 5 seconds and then stopped for 5 seconds:

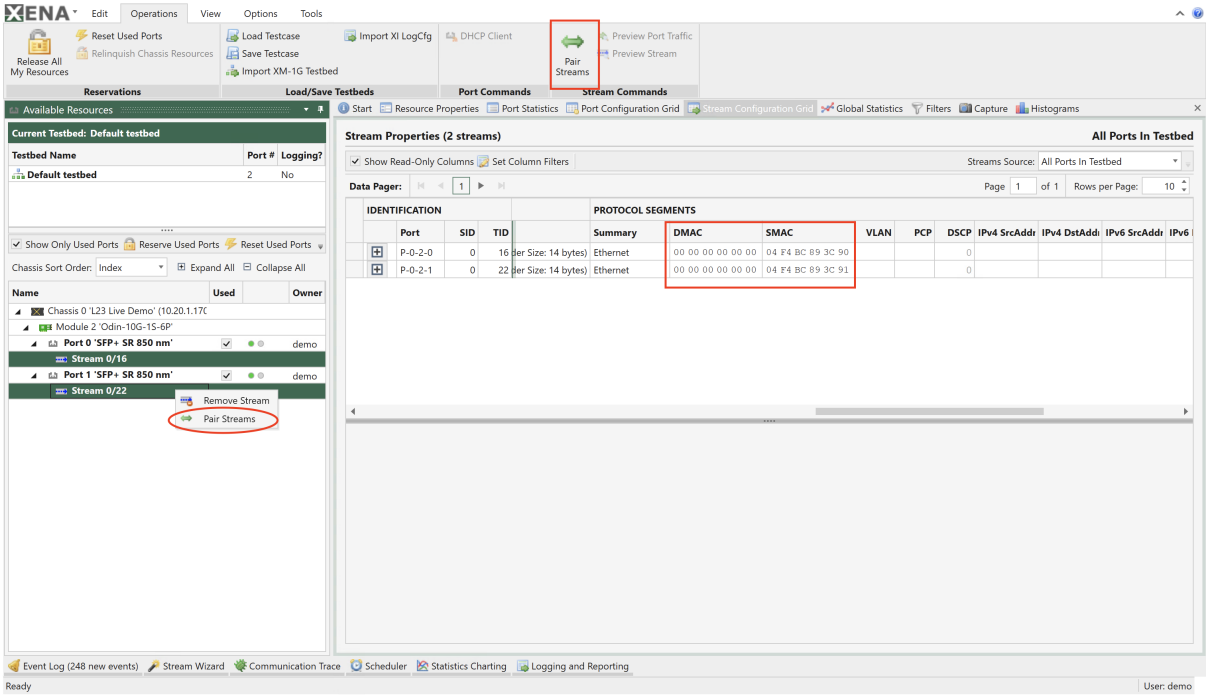


Fig. 3.17: Pairing streams

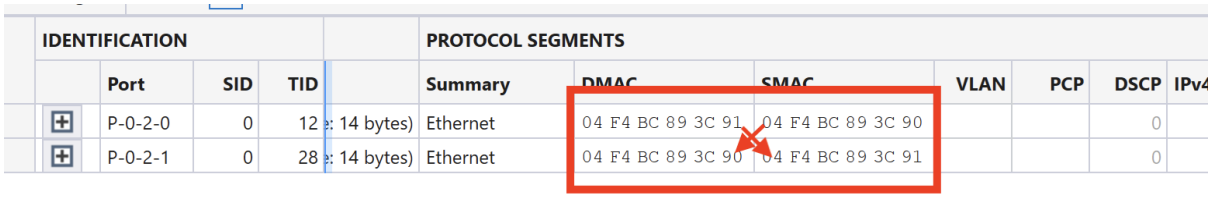


Fig. 3.18: Streams paired

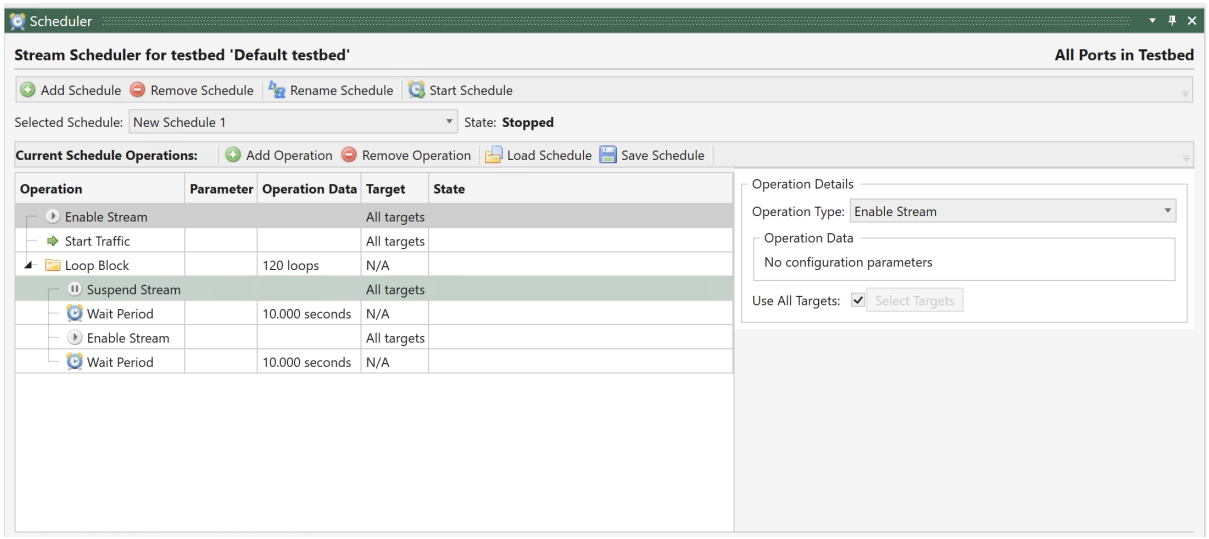


Fig. 3.19: Stream scheduler

3.1.9 Filters

Filters are used in order to get statistics on specific types of packets either specific content or specific packet size.

> **Note**
>
> These filters can also trigger the Capture mechanism or focus Histogram results.

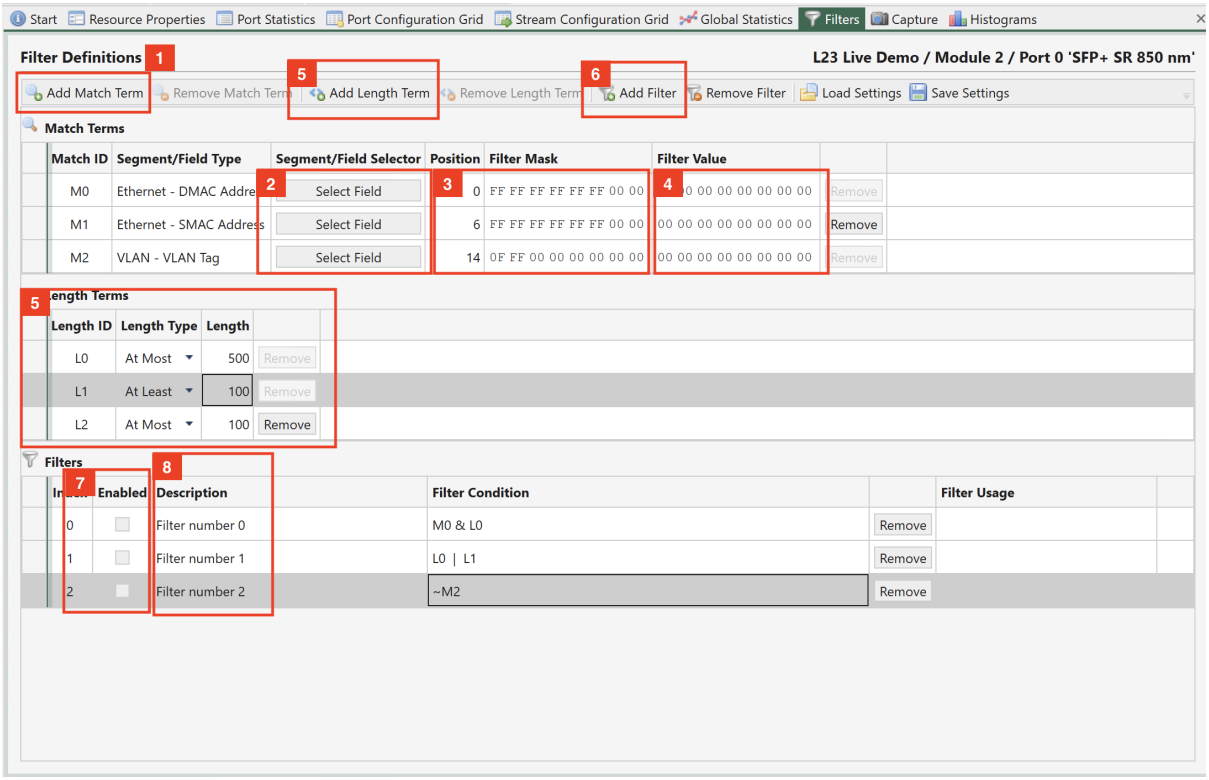


Fig. 3.20: Port filters

1. **Add match term** - click to add new match term. Added to provided Statistics for a matched packet (e.g. Packets of VLAN 100)
2. Click to access the field you want to match (headers must be added manually per match term).
3. **Position** is set to beginning of field e.g. if you want to match last octet of IP, the offset should be incremented manually. **Mask** - to focus on a specific Byte the other should be set to `00`
4. **Value** - The value you would like to match (the value is in Hex so 50Dec = 32Hex)
5. **Length term** - used to find specific packet sizes
6. **Add filter** - click to add/build a new filter based on match terms
7. **Enable** - checkbox to enable a filter to be present in the results and capture trigger
8. **Describe** - Name of the filter

Note

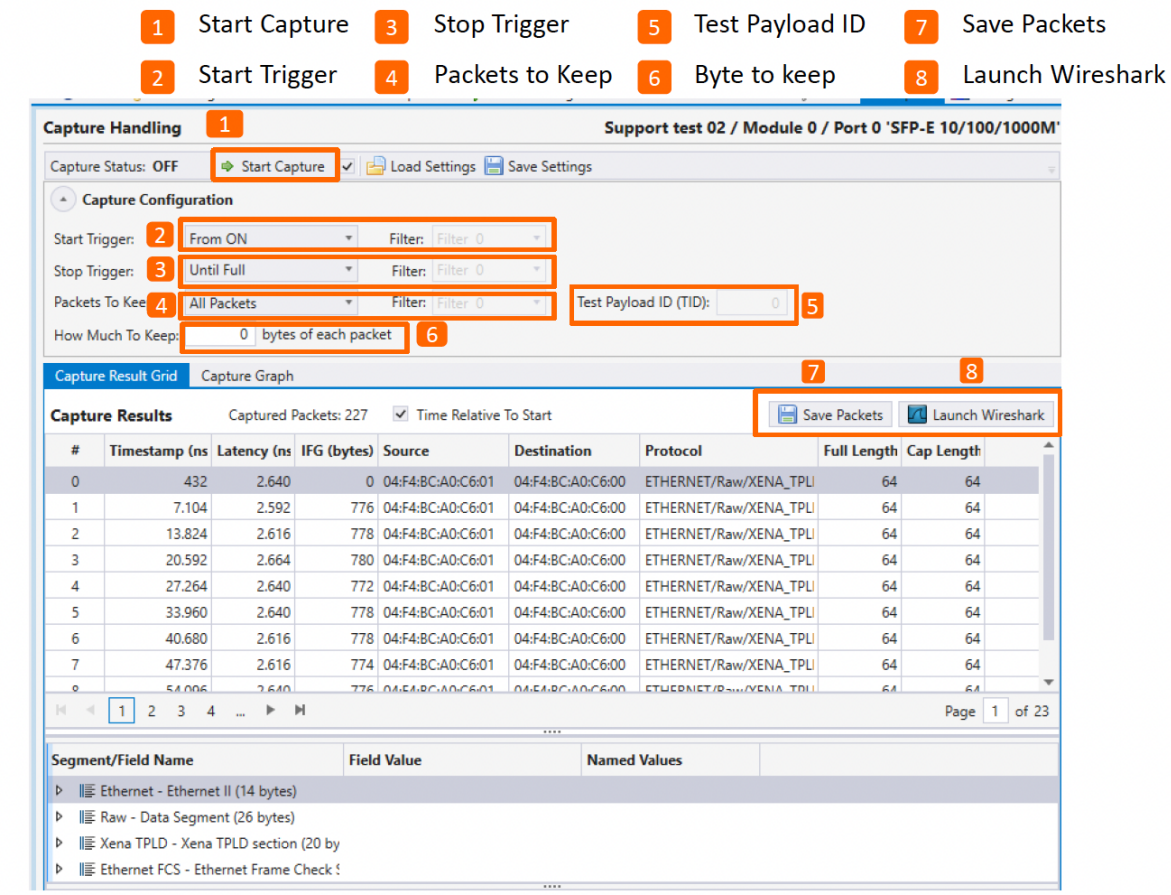
Filter Condition - Build a filter based on pre built terms using the &, |, and ~ operators.

Filter results under *Global Statistics* → *Port Statistics*

Filter Traffic

Name	Description	RX (%)	RX (bit/s)	RX (pps)	RX (bytes)	RX (packets)
P-0-10-0	(Aggregated filter counters)	29.585	290,855,140	31,231	8,139,506,404	6,939,422
Filter 0	SRC IP = 1.1.1.10	1.005	9,857,620	1,192	116,137,606	113,177
Filter 1	VLAN ID= 100	0.008	79,160	10	1,078,644	1,041
Filter 2	Runts	0.005	35,730	72	900,469	14,533
Filter 3	Jumbo	8.673	85,749,370	6,115	2,444,688,239	1,397,063
Filter 4	Legal Packet Size	11.214	109,319,420	17,653	3,131,110,945	4,002,011
Filter 5	Illegal Packet Size	8.680	85,813,840	6,189	2,445,590,501	1,411,597

3.1.10 Capture



1. Checkbox enabled means when click Start/Stop in global view, capture mechanism will Start/Stop on this port.
2. Start Triggers

- From ON: Means Automatically Start
- From FCS error: First FCS error seen triggers Capture Start
- From payload error: First Payload error seen triggers Capture Start
- Filter: First packet answering Filter condition triggers Capture Start

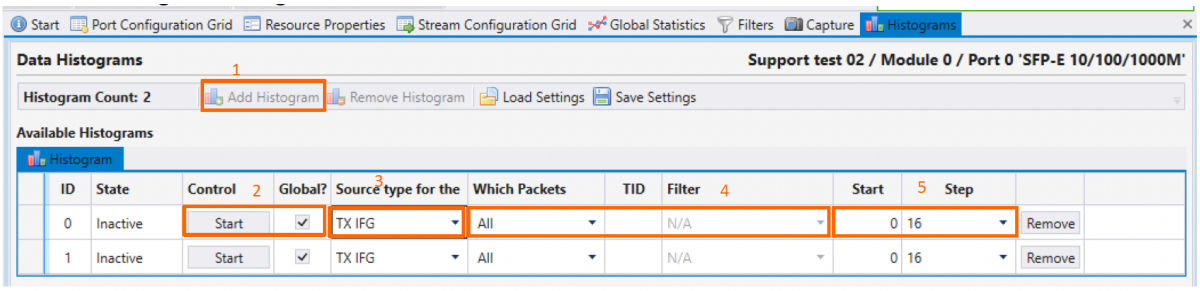
3. Stop Triggers
   - Until full: Means Automatically stop when buffer full
   - Until FCS error: First(/2nd) FCS error seen triggers Stop
   - Until payload error: First(/2nd)Payload error seen triggers Stop
   - Filter: First (/2nd) packet: answering Filter condition triggers Stop
   - Until User Stop: Capture will keep capturing FILO till manually stopped.
4. Which packets to keep (which will be left in capture buffer)
   - All: All packets are captured
   - With FCS error: Only FCS error frames
   - With payload error: Only payload error frames
   - Without test payload: Only non stream packets remain
   - With test payload: keeps only packets that are part of stream x(5) fill in the payload ID
   - Filter: keeps only packets answering Filter X conditions
5. Test payload ID to capture
6. Bytes to keep in capture buffer.
7. Save capture buffer as PCAP file.
8. Open Capture buffer with Wireshark

3.1.11 Histogram

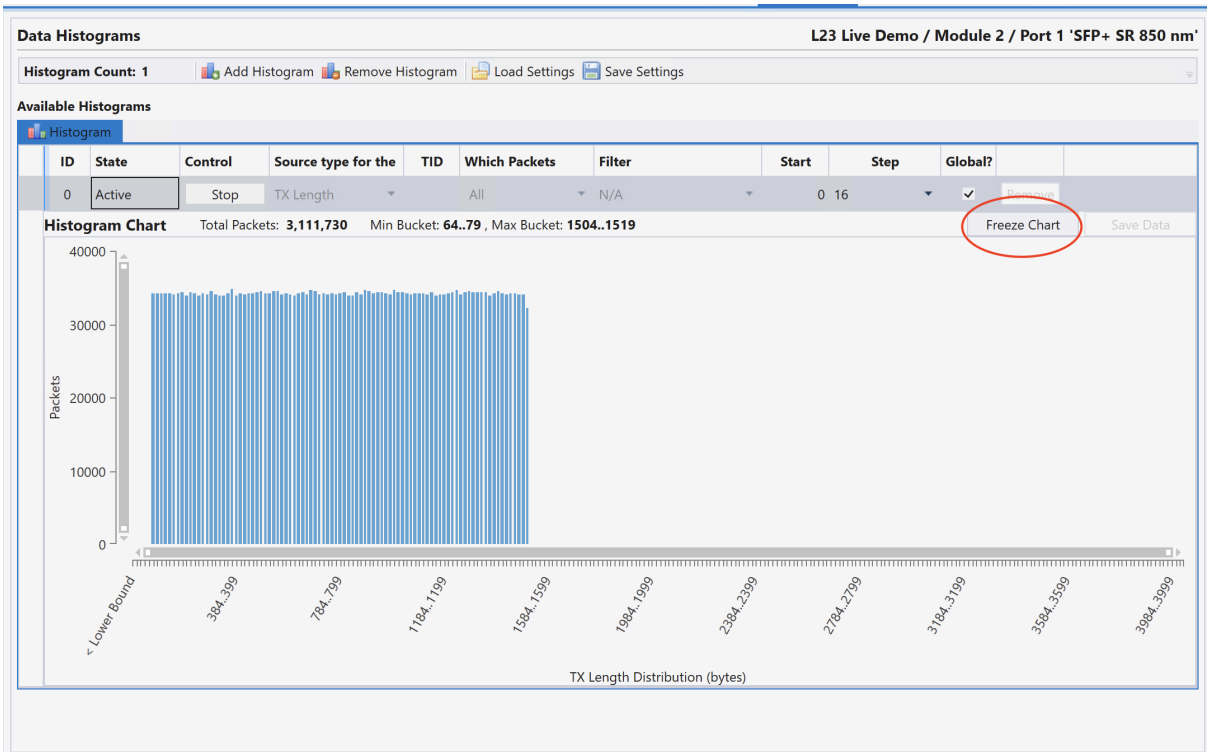
Histograms are used to plot different distributions of values gathered over time e.g.

- Tx/Rx Length (Packet size)
- Rx Latency (Latency and Jitter may drift over time)
- Rx Jitter
- Rx IFG (an additional way of observing Jitter behavior)

1. Add histogram - Multiple histograms can be run simultaneously.
2. Start Histogram - Start Manually or use checkbox to start from Global
3. Select the type of measurement you would like to track using the histogram.
4. Select which packets will be monitored by this histogram, either specific TID or packets answering a specific filter.



5. X-axis range choose the minimum offset and the resolution (step)
6. Use the *Freeze* button to freeze the view and enable the Save option.



3.1.12 Global Statistics

1. Go to the *Global Statistics* panel. You should now see your two test ports in the testbed in a grid view.
2. Press the *Clear Counters* button in the toolbar at the top of the panel to ensure that you start with a clean view.
3. Press the *Start Traffic* button to start traffic on all test ports in your testbed. You should now see the TX and RX traffic counters start to increment for both ports.
4. Press the *Stop Traffic* button to stop the traffic on both ports.

Note

The Global Statistics view will exclusively display ports and streams utilized by your current

testbed. If you need to briefly examine the statistics counters for a different port, you can utilize the single-resource *Port Statistics* panel. This panel will provide statistics for the currently selected port, regardless of whether it is part of your testbed or not.

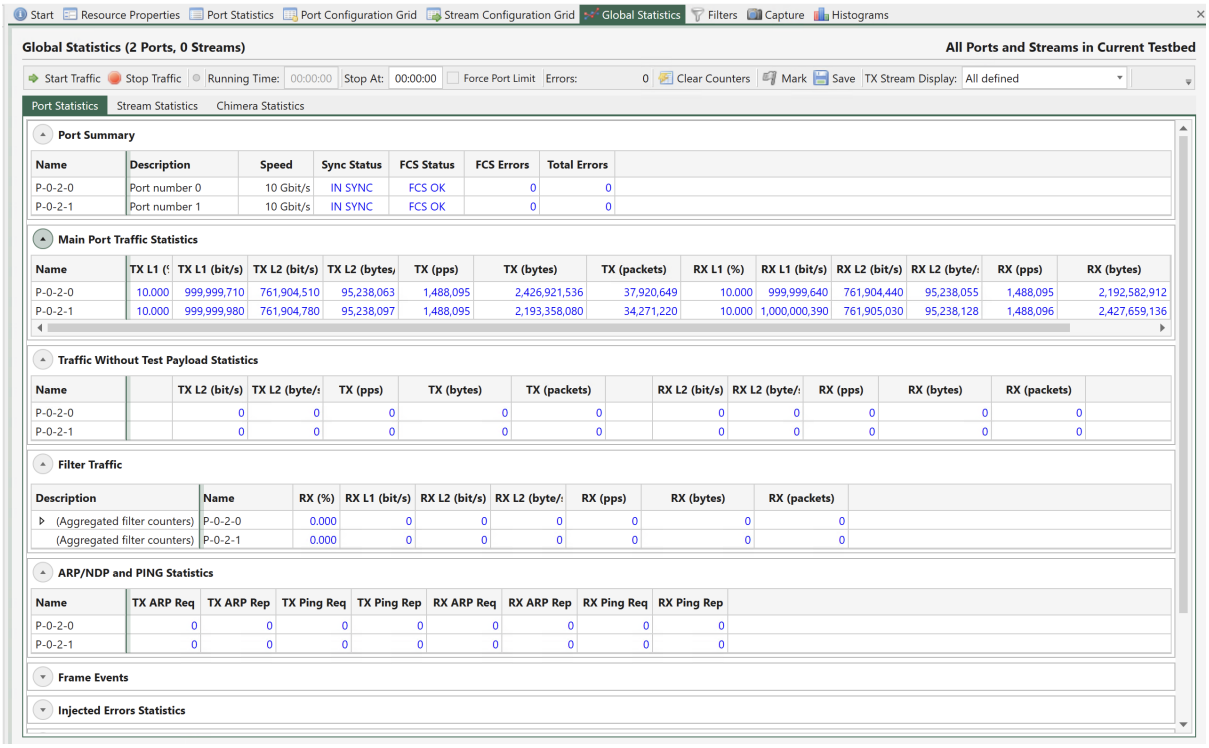


Fig. 3.21: Global statistics - port statistics

Note

(TX-RX) packets gives packet loss results based on TX packets and RX packets.

RX packets can consist of two parts: `RX_1` originated from the TX port, `RX_2` duplicated by the DUT

1. **`(TX-RX) > 0`:**
    a. `TX > RX_1+RX_2`. Transmitted packets more than received even if there is duplication from the DUT.
2. **`(TX-RX) = 0`:**
    a. `TX-RX_1 = 0`, and `RX_2 = 0`. All transmitted packets are correctly received, and there is no duplication introduced by the DUT.
    b. `TX-RX_1 = d`, but `RX_2 = d`. Some transmitted packets are not received but the duplications by DUT cancels out the difference, where d is positive.
3. **`(TX-RX) < 0`:**
    a. `TX-RX_1 = 0, RX_2 > 0`. There is duplication from the DUT.

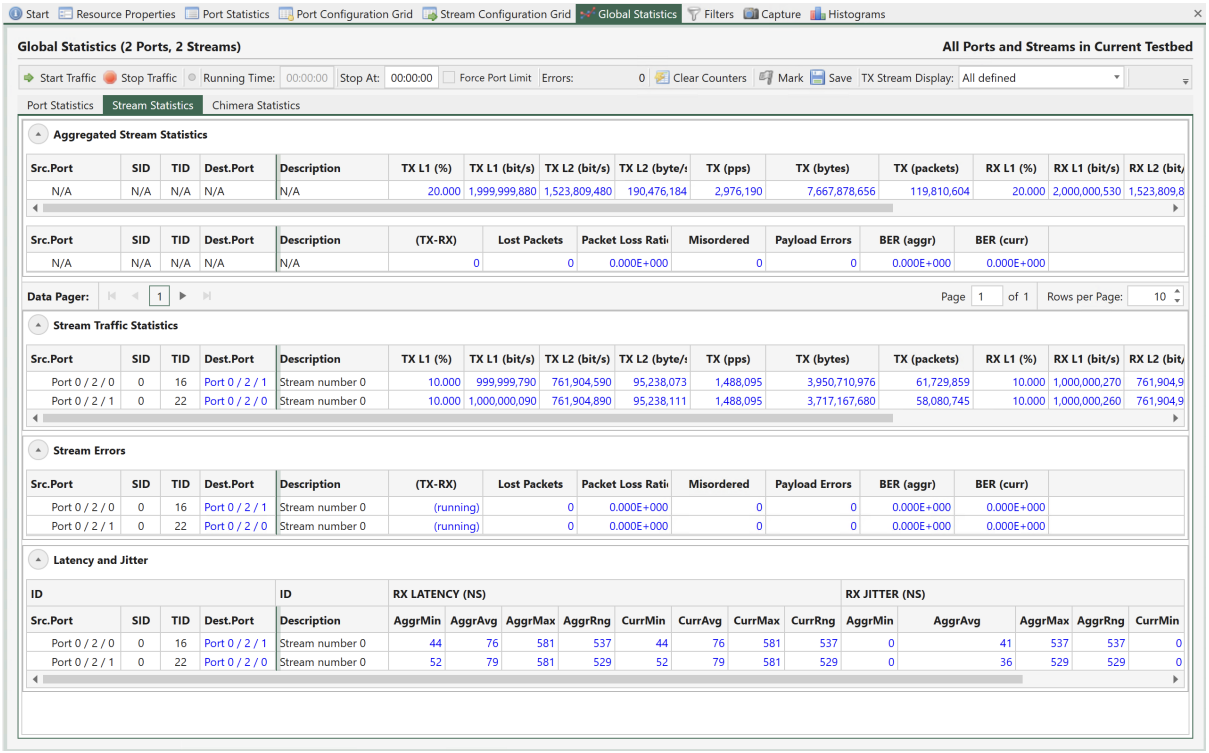


Fig. 3.22: Global statistics - stream statistics

b. `TX-RX_1` = d, `RX_2` > d. Duplication from the DUT is large than the lost, where d is positive

Lost Packets

Given that RX can consist of two parts as mentioned above, we cannot be certain about the `TX-RX_1`. Consequently, we need use the sequence numbers in the packets to calculate "the holes in the sequence". For example, when the TX sends #0,#1,#2,#3,#4,#5,#6, but #3 is not received by the RX, then `TX-RX_1` = 1. But if #3 is not truly lost but is delayed by the DUT and later received by the RX, the `TX-RX_1` will be back to 0.

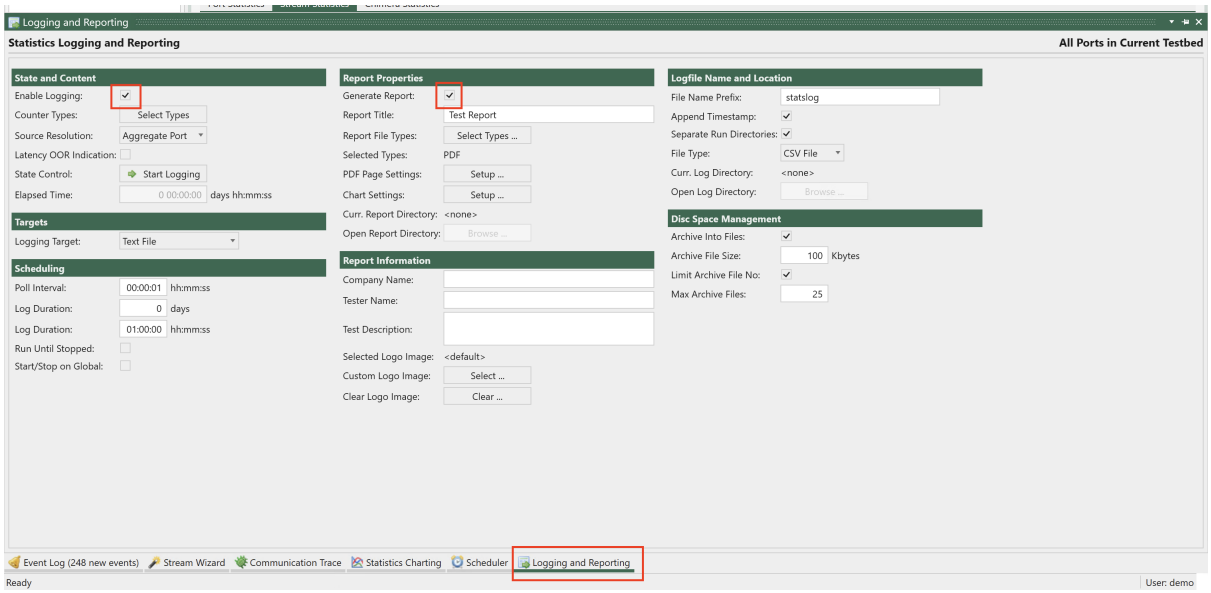
TX: #0,#1,#2,#3,#4,#5,#6 RX: #0,#1,#2,#4,#5,#6 (Lost Packets = 1) (after a while...) RX: #3 (Lost Packets = 0)

Unfortunately, there is no one mechanism that can handle both scenarios at the moment.

3.1.13 Logging and Reporting

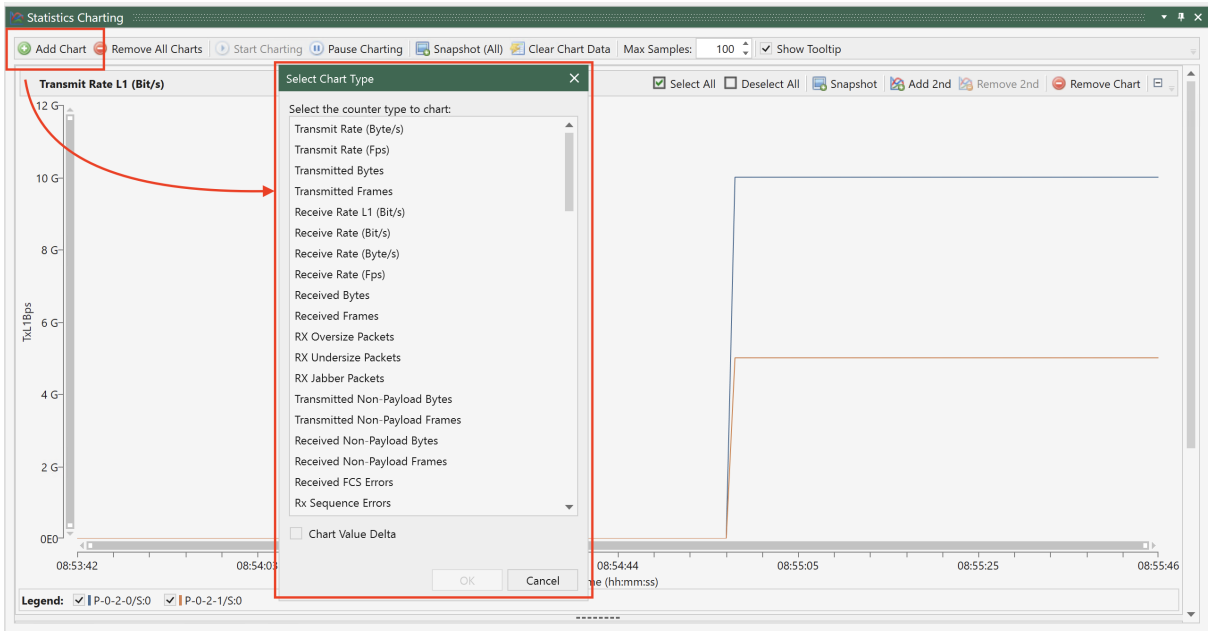
Enable Counter Logging - in order to save results over time and record all results for each second that passes. Enable Generate Report - in order to generate a report of accumulated results.

Click *Select Types*, and select the counters you want to record and/or include in the report



3.1.14 Statistics Charting

Click *Add Chart* and select the chart types you want. Then click *Start Charting*.



3.2 Getting Started with Network Impairment

This section is intended to guide you in using XenaManager to initiate a basic traffic impairment scenario for network impairment.

3.2.1 Configure Chimera Module

1. Description: In this field add a Module Description text
2. Port Configuration: Choose the port speed

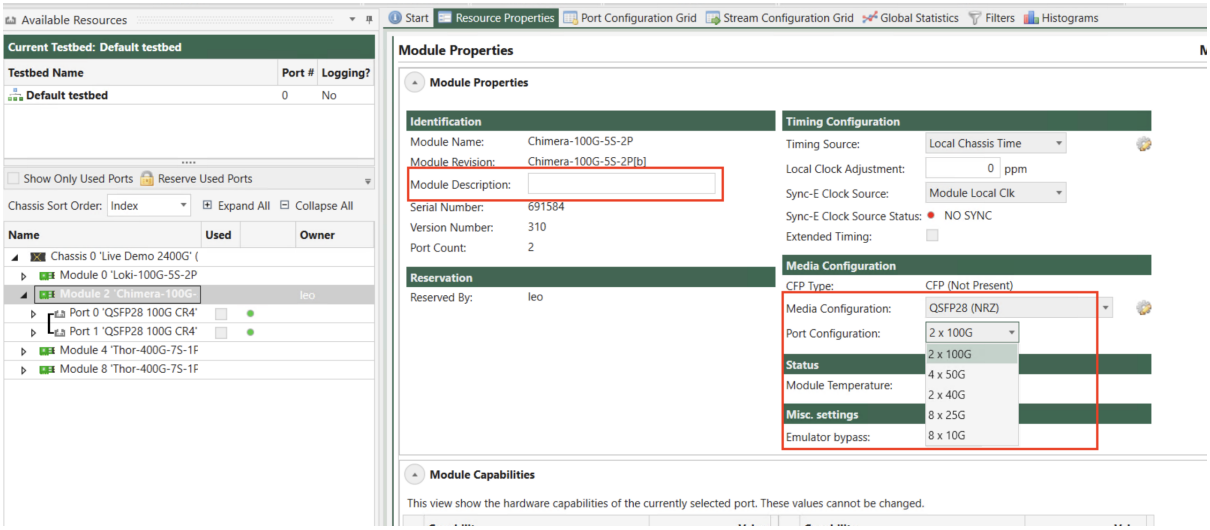


Fig. 3.23: Configure chimera module

3.2.2 Add Chimera Port

1. Select the port(s) you want to use.
2. Click *Reserve Used Ports*, check *Show Only Used Ports*

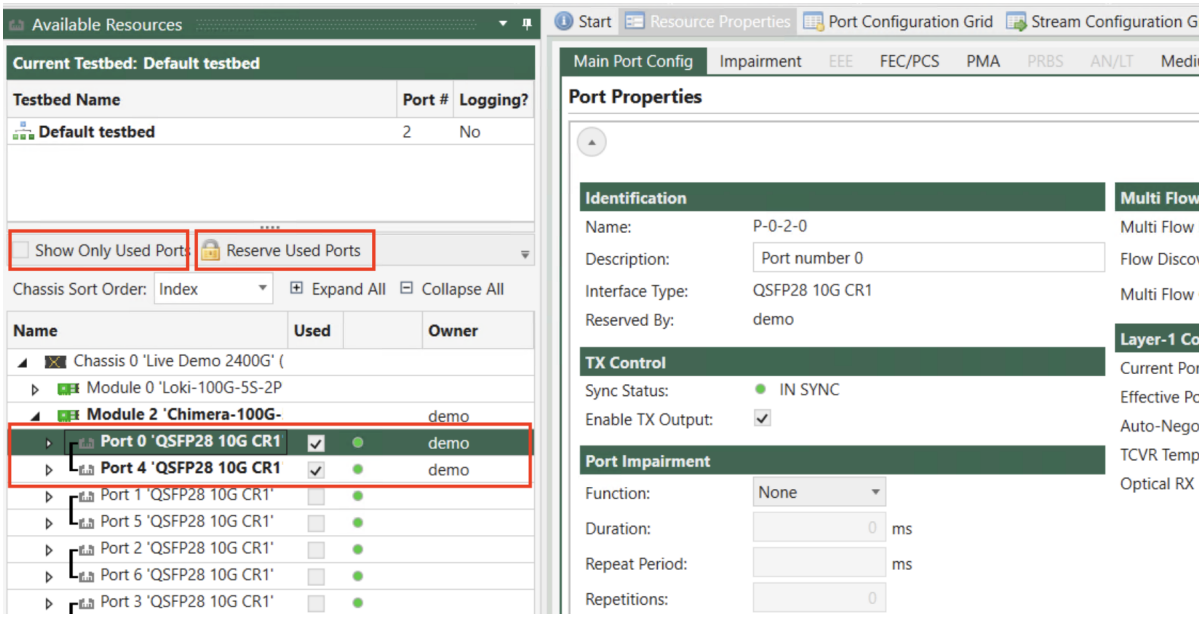


Fig. 3.24: Reserve ports

3.2.3 Enable Impairment

You need to enable impairment on the ingress port, where the traffic enters.

1. Click port
2. Go to *Impairment Config* tab.
3. Enable *Enable Impairment*. When you observe a green wave icon alongside the in-sync indicator, it indicates that the port is capable of performing impairments.

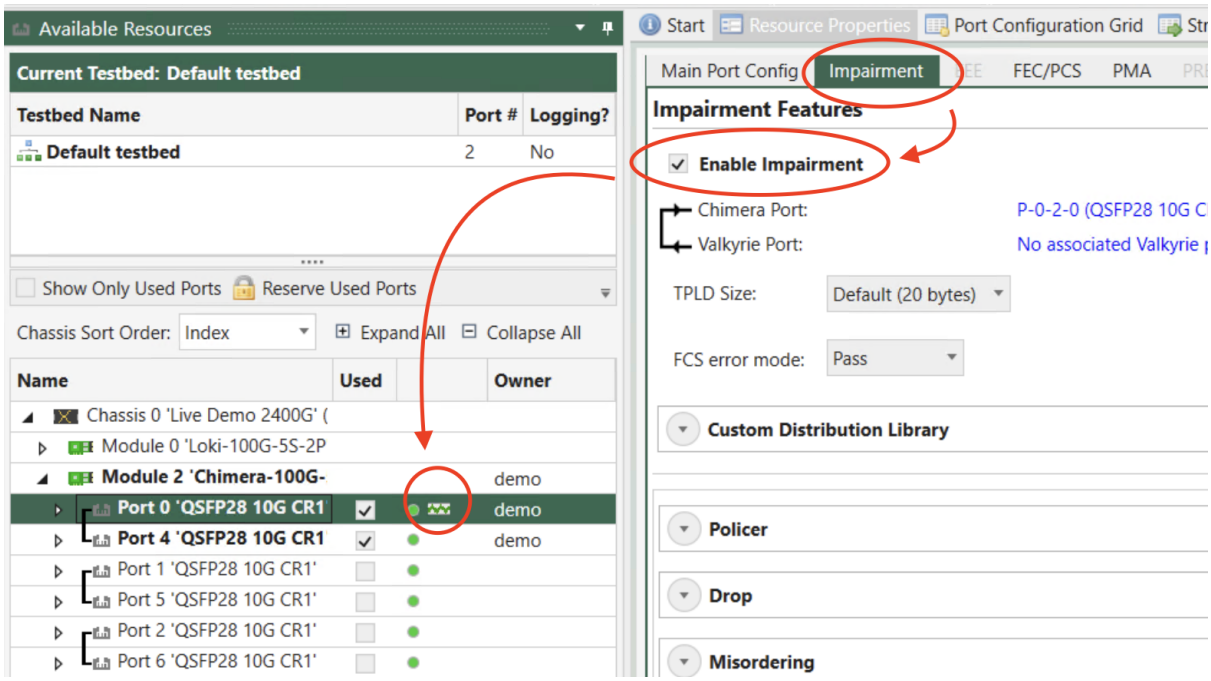


Fig. 3.25: Enable impairment on port

3.2.4 Configure Flow Filter

Let's set up a flow to capture all frames containing a single VLAN tag with a value of 10. Here's how you can configure it.

1. Expand a port and select *Flow (1)*.
2. Change flow description to `VLAN 10`. This helps you to identify the flow later.
3. Enable the filter. This sets the filter in action.
4. Choose `1 VLAN Tag` for *Layer 2+*.
5. Check *Tag* and enter `10` in *Value*.
6. Click *Apply* to apply the changes.

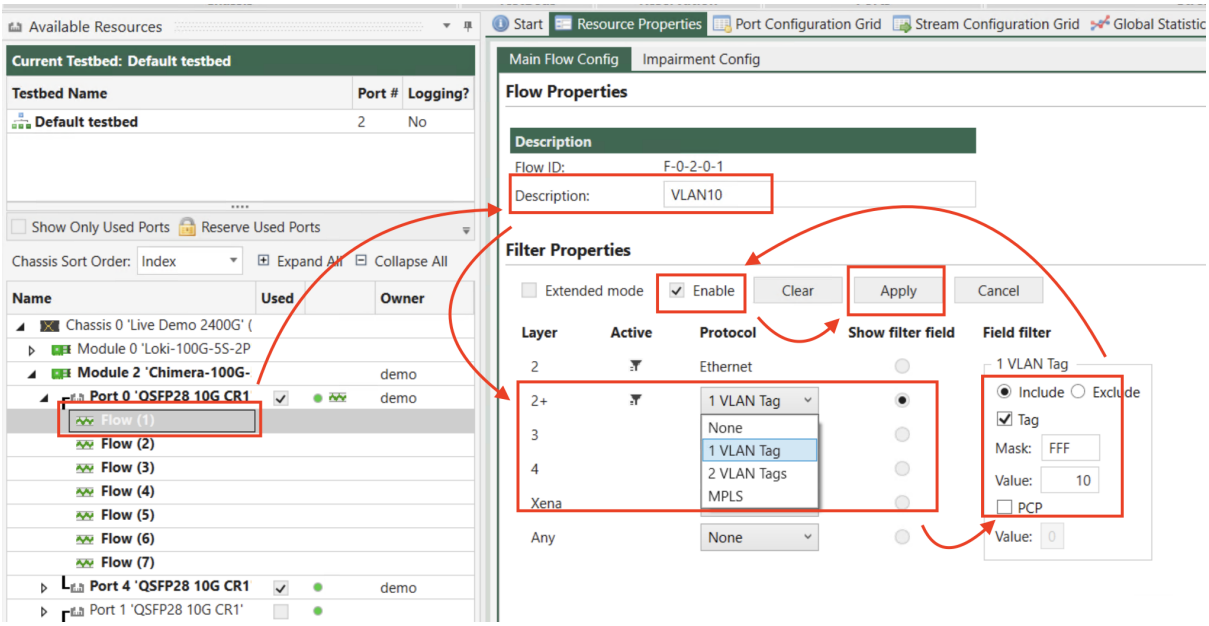


Fig. 3.26: Configure flow filter

3.2.5 Configure Flow Impairments

Now that we have a filter in place to capture all frames with a single VLAN tag carrying the value 10, we can proceed to discard 20 packets every 0.1 second from the filtered frames using the Drop function..

1. Select *Impairment Config* tab.
2. Expand *Drop* and choose `Fixed Burst` in *Distribution*.
3. Choose `Repeat` in *Scheduling* and enter `0.1` in *Repeat Period*.
4. Enter `20` in *Burst Size*.
5. Click *Apply* to apply the changes.
6. Click *Start* to start the impairment.

Now you can see the drop function is active. To stop the drop function, simply click *Stop*.

3.2.6 Chimera Statistics

Now let's go to check the drop statistics.

1. Go to Global Statistics tab.
2. Click Chimera Statistics tab.
3. Under *Total Counters*, the RX and TX statistics on *Flow (1) VLAN 10* show you the traffic statistics captured by the flow filter.
4. Under *Packet Drop*, you can see the drop statistics on *Flow (1) VLAN 10*.

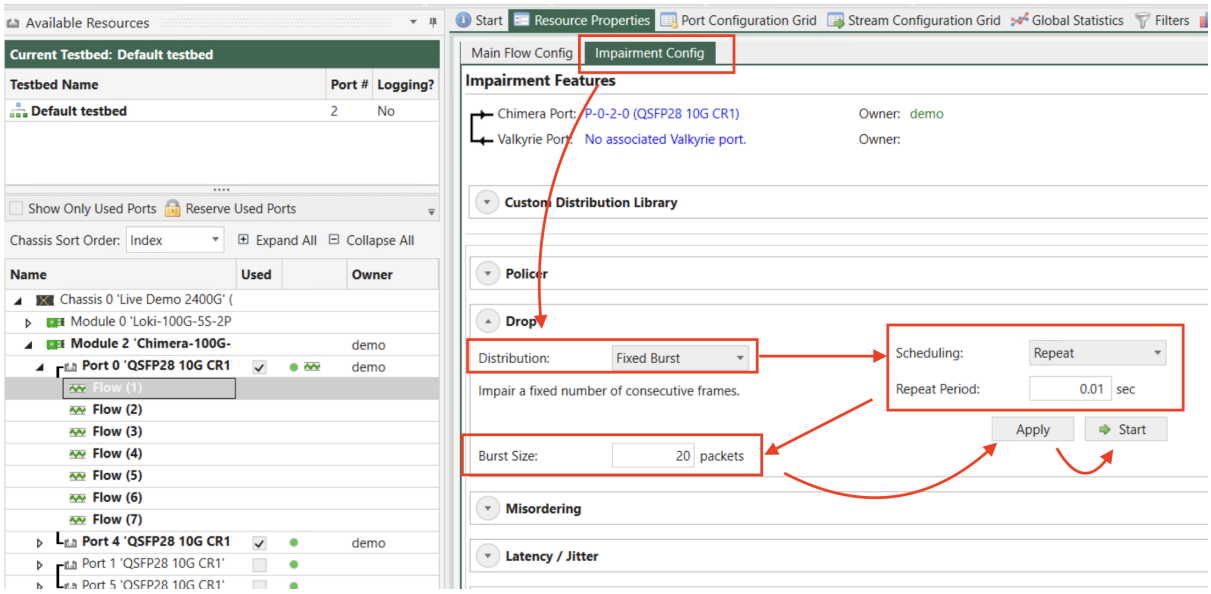


Fig. 3.27: Configure flow impairment

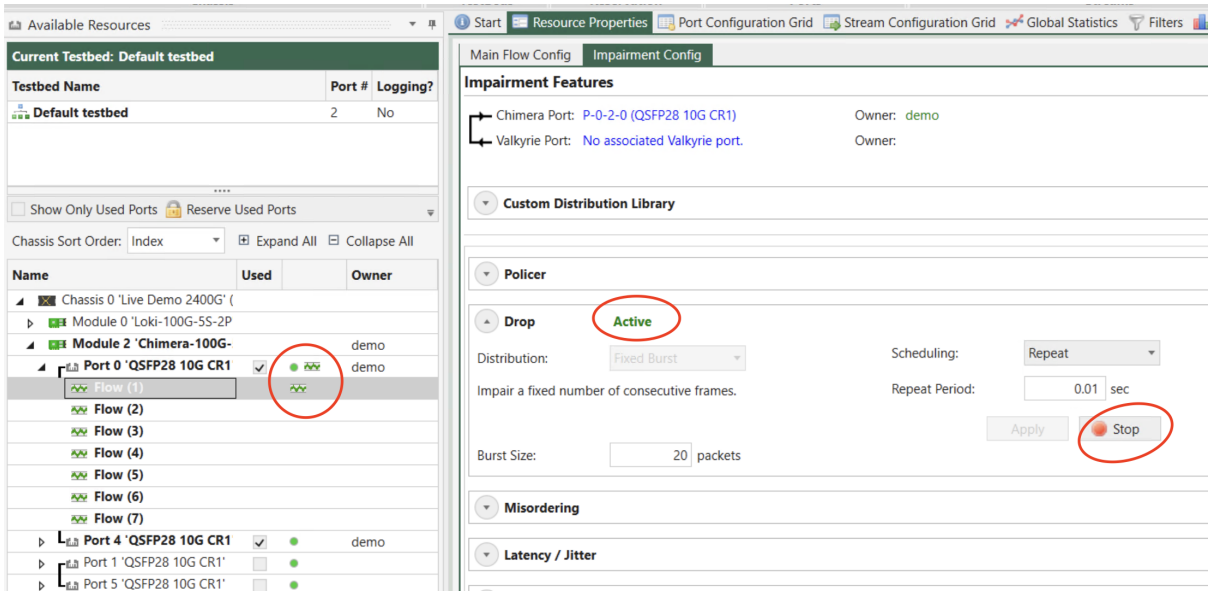


Fig. 3.28: Configure flow impairment

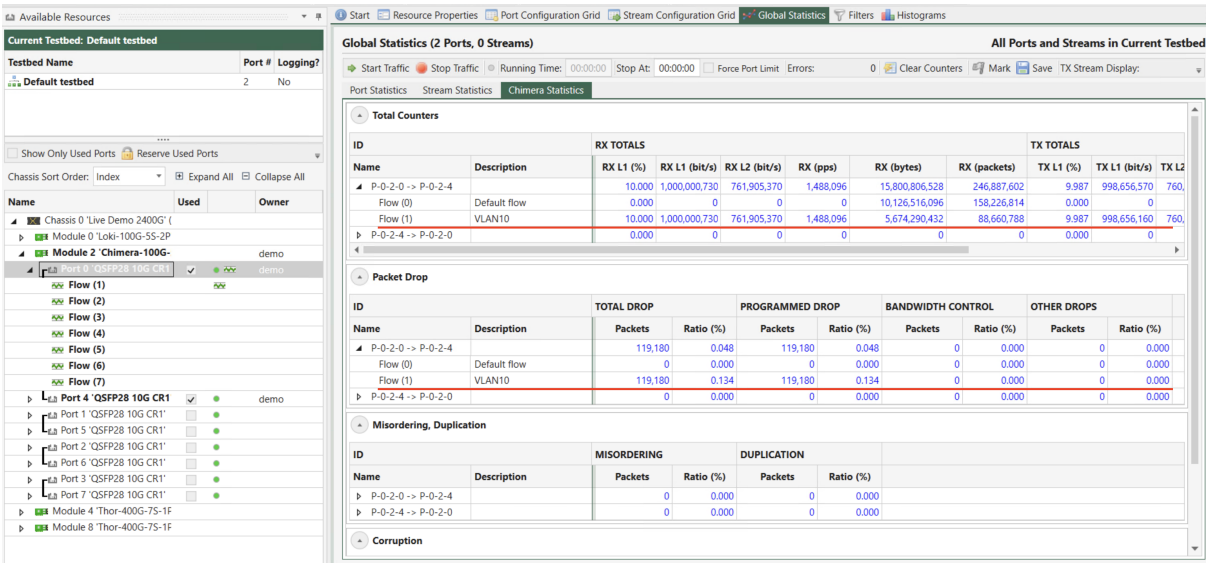


Fig. 3.29: View impairment statistics in Global Statistics

CHAPTER

FOUR

GENERAL

4.1 Chassis Management

4.1.1 General

Discover Chassis

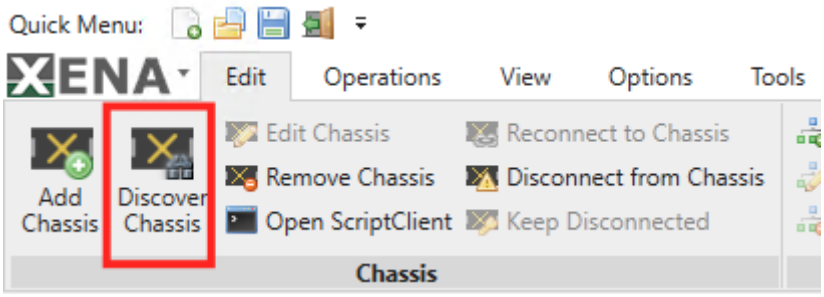


Fig. 4.1: Discover chassis button

The chassis discovery function allows you to see a list of active Xena test chassis connectable in your network. This is useful when you don't know the IP address of the chassis you want to connect to.

Important

You may need permission from your network/firewall settings in order for your test chassis to be discovered by XenaMananger.

- Allow IP address `224.2.1.3`
- Allow `UDP` port `22607`

Connecting to Chassis

Chassis definitions are contained in the overall test configuration. You can add a chassis by pressing the *Add Chassis* button in the main *Edit* ribbon menu. You will then see the following dialog window:

1. Fill in the IP address or hostname for the chassis in the *Chassis Address* field.

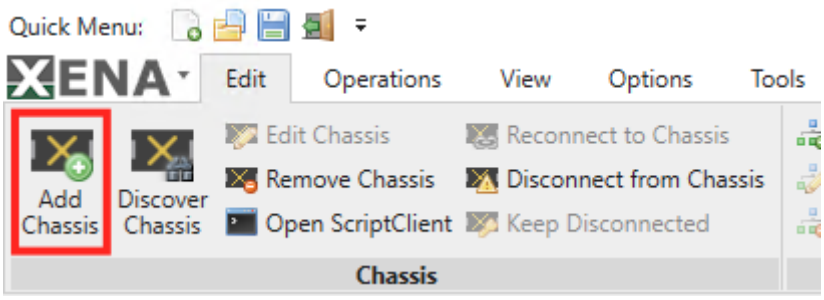


Fig. 4.2: Add chassis button

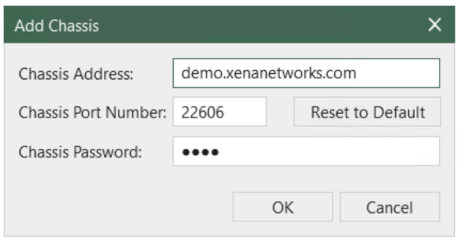


Fig. 4.3: Add chassis dialog window

2. Optionally change the *Chassis Port Number* value if you connect to the chassis through a NAT router that changes the port number. The default port value is **22606**. If you have changed the port value and want to revert to the default value you can press the *Reset to Default* button.
3. Enter the assigned password for the chassis in the *Chassis Password* field.
4. Press the *OK* button.

The next time you open the *Add Chassis* window, it will remember the last values you entered. If you have changed the port number and need to revert to the default Xena port number just press the *Reset to Default* button.

Note

Please be aware that if you attempt to add or reconnect to an offline chassis, or if the network settings are incorrectly configured on the client PC and/or chassis, you will receive the message: "Connection to chassis failed."

Editing Chassis Address

If you need to modify the address or password details for a chassis you can select the chassis in the resource tree view and press the *Edit Chassis* button in the ribbon menu. You will then see a window similar to the *Add Chassis* window where you can change one or more of the values.

Note

Note that the *Edit Chassis* button will only be enabled if you are **not currently connected** to the chassis (we assume that if you are connected to the chassis you have no need for changing

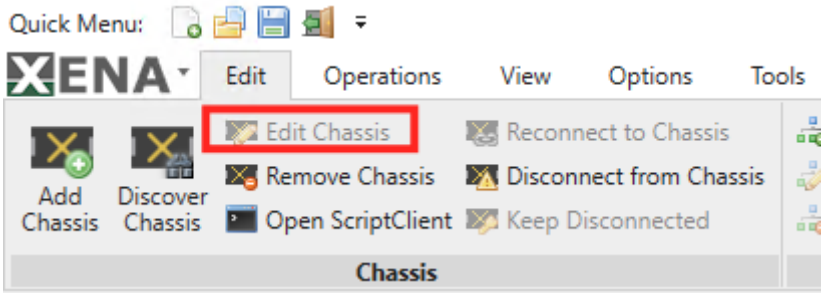


Fig. 4.4: Edit chassis button

the defined address)

This action is also available in the right-click context menu for the chassis item in the tree view.

Reconnecting to Chassis

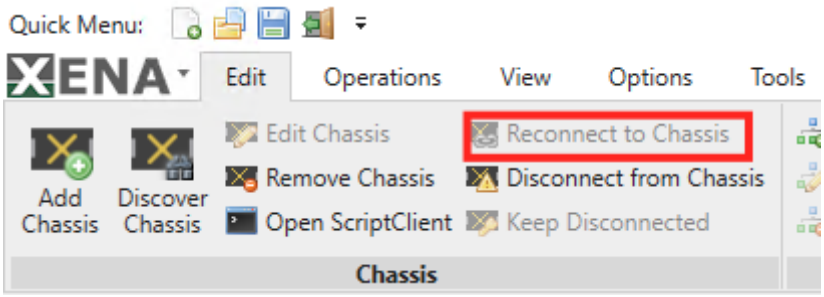


Fig. 4.5: Reconnect chassis button

If you have lost the connection to a chassis, for instance due to a local network connectivity outage, you can manually reconnect by selecting the chassis in the resource tree view and press the *Reconnect to Chassis* button in the ribbon menu.

This action is also available in the right-click context menu for the chassis item in the tree view.

Note

Please be aware that if you attempt to add or reconnect to an offline chassis, or if the network settings are incorrectly configured on the client PC and/or chassis, you will receive the message: "Connection to chassis failed."

Disconnect from Chassis

You can forcibly disconnect from a defined chassis without removing the definition from the configuration. This will also prevent XenaManager from making any attempt to reconnect to the chassis, until you specifically choose to reconnect to that chassis. You can use this option if you have a chassis defined in your configuration that you know will be offline for a longer period of time.

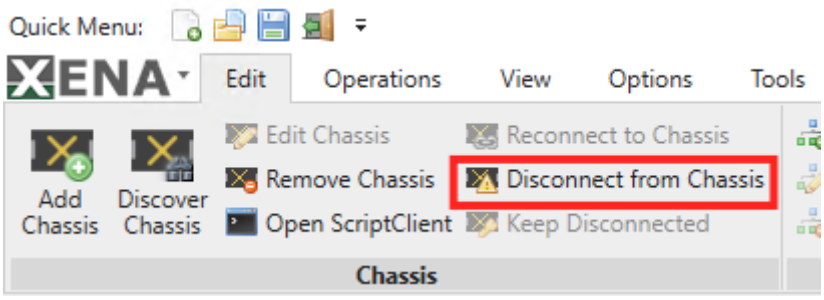


Fig. 4.6: Disconnect chassis button

Remove Chassis

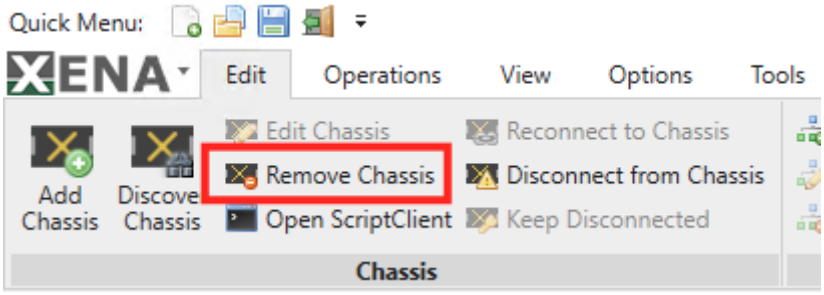


Fig. 4.7: Remove chassis button

If you no longer need a certain chassis in your test configuration simply select the chassis and press the *Remove Chassis* button in the ribbon menu.

This action is also available in the right-click context menu for the chassis item in the tree view.

Refresh Chassis

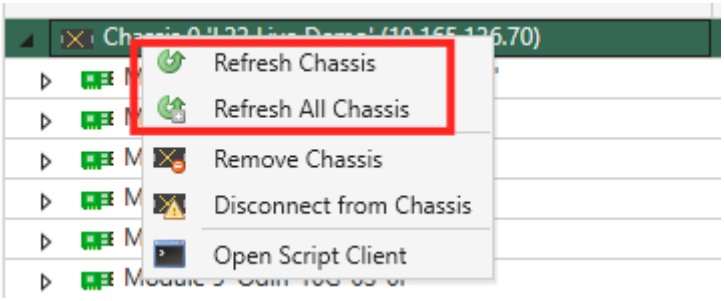


Fig. 4.8: Refresh chassis button

You can also refresh the chassis configuration by selecting an appropriate option in the right-click context menu:

- *Refresh Chassis*: This will refresh the chassis instance configuration.
- *Refresh All Chassis*: This will refresh the chassis and associated resources, i.e. all modules and ports contained in it.

4.1.2 Troubleshooting

- If the **password is lost**

The default value of the password is **xena**, which can be changed from the *Chassis Properties* panel of XenaManager.

If the password is forgotten, the following method can be used to gain access to the chassis: after power-on when the test port LEDs start flashing, for the next two minutes the chassis will accept its own serial number (which is printed on the label at the back of the chassis) as a backup password.

- If the **IP address is lost**

The extension port (EXT port) is not used in normal operation. It serves as a backup with a known IP address.

The EXT port is pre-configured with the following IP setup:

- Address = `172.16.255.200`
- Subnet = `255.255.255.0`
- Gateway = none

You must configure your PC port statically to an IP address in the `172.16.255.x` range, and then you will be able to ping the chassis again. Now start XenaManager, and connect to the chassis using the IP address `172.16.255.200`. Under the *Chassis Resource Properties* you can then see which IP address is configured for the MGMT port, and you can reserve the chassis and change it if necessary. Changes to the IP address of the MGMT port take effect after rebooting the chassis.

The location of EXT ports on XenaCompact and XenaBay chassis is shown in Fig. 4.9 and Fig. 4.10.

> **Note**
>
> Note that the IP configuration of the EXT port cannot be changed, and that you should not configure the MGMT port to use this subnet.

> **Note**
>
> XenaBay supports 10M/100M/1G on MGMT and EXT port.
>
> XenaCompact supports 1G/10G on MGMT and EXT port.

4.1.3 Software Maintenance Activation

Please refer to Xena ChassisUpgrader User Manual.



Fig. 4.9: Extension port on XenaCompact chassis

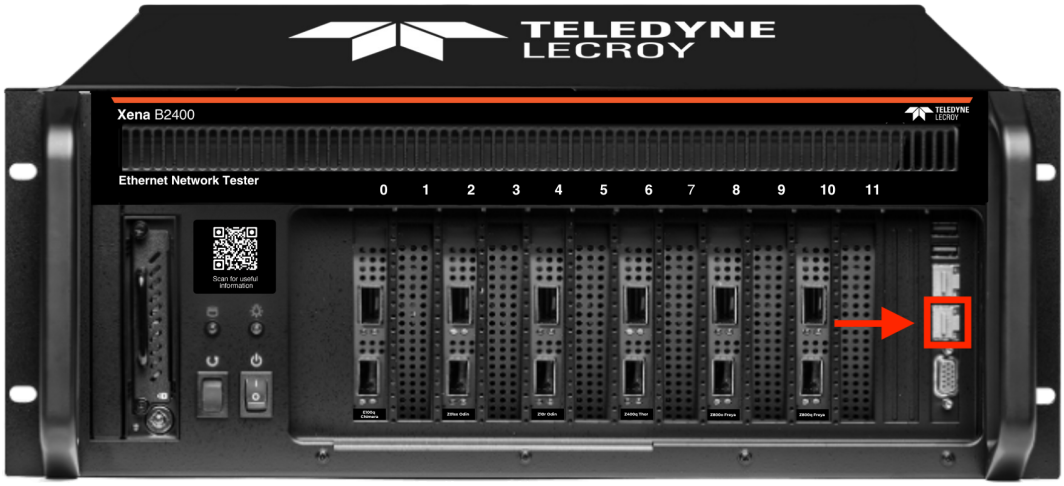


Fig. 4.10: Extension port on XenaBay chassis

4.2 Chassis and Module Firmware Compatibility

Every time XenaManager connects to a chassis, it checks if the installed chassis and module firmware are compatible. If the installed software is incompatible, a dialog window will pop up, as shown in the screenshot below, displaying the results for each test module and system software in the chassis:

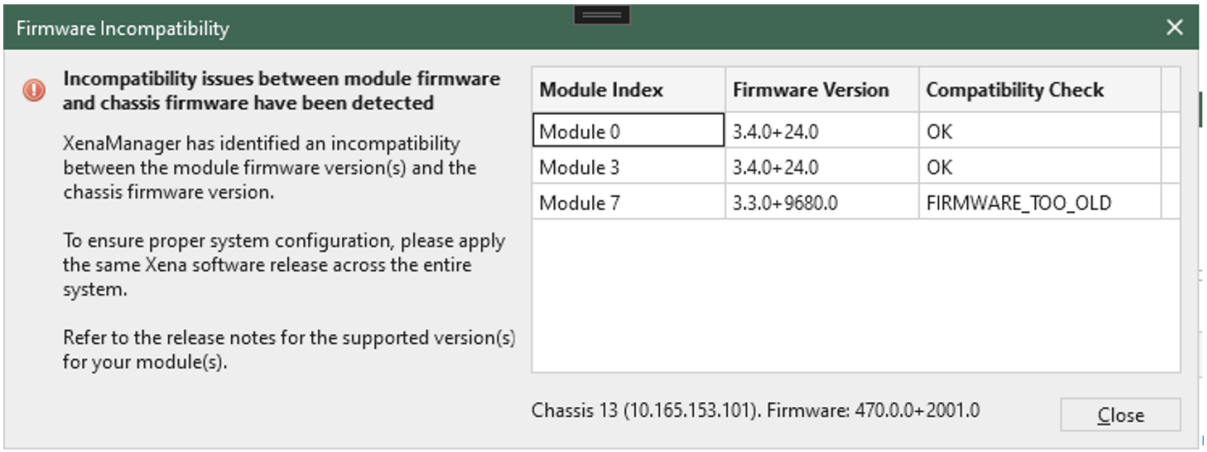


Fig. 4.11: Chassis and module firmware compatibility check

The user should perform a downgrade or upgrade of the system to ensure it is running on a compatible version. Please note: Upgrade options might depend on the End-of-Life (EoL) status of modules/chassis and/or a valid maintenance agreement for the products.

4.3 Chassis Time Synchronization

This section describes how to setup and monitor time synchronization between multiple Xena test chassis.

Important

This function requires additional software installed on your Xena test chassis. Read Time Synchronization for more details.

4.3.1 Overview

Capabilities

The chassis time synchronization feature enables multiple Xena testers to synchronize their local time to each other. This can be used for various purposes:

- One-Way Latency (OWL) measurements between two test chassis.
- Synchronized traffic start between multiple chassis.
- Accurate timestamping of captured packets in exported PCAP files.

The timing network consisting of your Xena testers may be configured in a very flexible way supporting multiple scenarios:

- One tester may serve time to the other testers (and any other host on your network) using any combination of NTP, PTP or RFC 868 TIME *.
- Each tester may obtain its own time from an external NTP, PTP or GPS source.

Note

(*) Note that the RFC 868 TIME protocol can only set the time with a precision of 1 second.

About TimeKeeper

TimeKeeper is an advanced time synchronization solution from the company FSMLabs. Xena Networks uses the TimeKeeper solution to keep the local time on each Xena test chassis in sync. The TimeKeeper solution must be installed on each Xena chassis that will participate in the timing setup.

The TimeKeeper solution runs as a separate service on the Xena chassis but is configured and monitored through the XenaManager.

Licensing

Each Xena chassis running the TimeKeeper solution will require an additional software license. The license is time-limited and must be periodically renewed for the TimeKeeper solution to continue to work.

The TimeKeeper license comes in two types: A client-only license that only allows the Xena chassis to obtain its time from an external source and a full license (aka. a server license) that also allows the Xena chassis to serve time to other hosts.

Important

Contact Xena sales for details on the availability and pricing of the TimeKeeper licenses.

TimeKeeper Configuration

The TimeKeeper configuration is accessed as part of the chassis resource property page. If the TimeKeeper solution is installed on the chassis an additional sub-tab named Time Service Configuration will be visible when you select the chassis resource property page, as shown in Fig. 4.12.

4.3.2 Service and License Control

The TimeKeeper service state can be controlled by the buttons in the top toolbar:

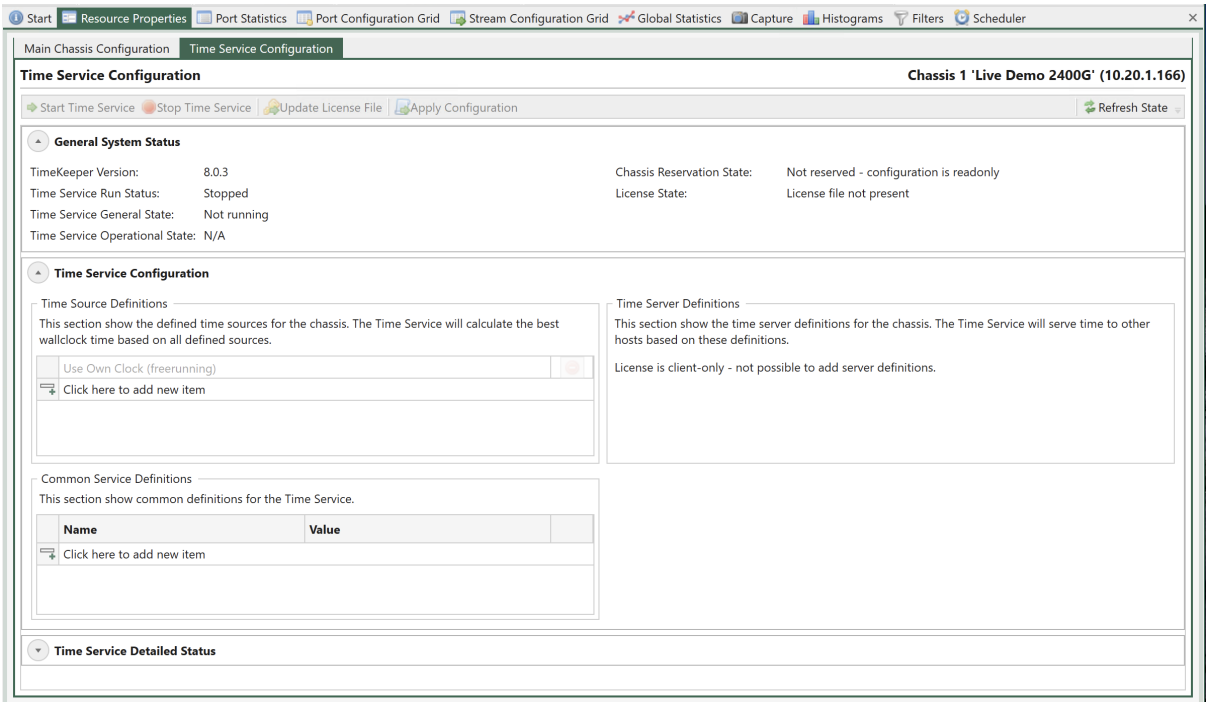


Fig. 4.12: TimeKeeper configuration

Table 4.1: TimeKeeper Service and License Control

Name	Explanation
Start Time Service	Start the TimeKeeper service if it is not already running.
Stop Time Service	Stop the TimeKeeper service if it is running.
Update License File	Upload a new TimeKeeper license file to the Xena chassis.
Apply Configuration	Apply a changed configuration for the TimeKeeper service. Invoking this option will also restart the TimeKeeper service.
Refresh State	The TimeKeeper status will be automatically refreshed every 5 seconds. You can however manually refresh the status by clicking this button.

General State

The general state of the TimeKeeper solution can be monitored in the General System State section at the top. The following values are provided:

Table 4.2: TimeKeeper General State

Value	Explanation
Version	The currently installed version of TimeKeeper.
Run Status	The current state of the TimeKeeper service (started or stopped)
General State	The general state of TimeKeeper
Operational State	A more detailed state of TimeKeeper
Reservation State	The current chassis reservation state (must be reserved in order to change configuration)
License State	The current license state (valid or invalid) and scope (client-only or full server).

Time Source Configuration

A time source represents the source from which this chassis will synchronize its OS kernel time. The following source types are supported:

- An external NTP server
- An external PTP server
- An internal SpectraTime GPS module (an optional hardware add-on for the chassis). PPS device is not used.

You must configure at least one time source for a chassis. It is possible to configure multiple time sources for a single chassis. The TimeKeeper solution will extract the optimal time based on the contributions from all configured sources.

Time Source Parameters

Table 4.3: Time Source Parameters

Parameter	Explanation	Applies To
Server Address	The address of the server to source time from	NTP Server, PTP Server
Interface	The network interface to listen on. If left empty all available network interfaces will be used.	PTP Server

Time Server Configuration

If a Xena chassis has been provided with a full TimeKeeper server license it may also serve time to the network, including other Xena chassis in the network. You can configure several different time server definitions for a chassis.

Time Server Parameters

Table 4.4: Time Server Parameters

Parameter	Explanation	Applies To
Interface	The network interface to send messages on. If left empty all available network interfaces will be used.	PTP Server

License Scenarios

This section describes various configuration scenarios and the required hardware and licenses.

Local Datacenter Scenario

If you have a number of co-located Xena testers in the same physical location and connected to the same local network you can use one of the testers as a time server. This tester will then serve time to the rest of the network, including but not limited to the other Xena testers. The best results will be obtained by using PTP between the Xena testers.

If you have N testers you will need one full TimeKeeper server license and N-1 client-only TimeKeeper licenses. You can configure the time server to synchronize to a public NTP server but if you need a very accurate local time you can optionally equip the time server tester with a SpectraTime GPS module.

Important

Note that the SpectraTime GPS module must be purchased and installed through Xena Networks.

Remote Networked Scenario

If you have two or more Xena testers in remote locations which are connected to the Internet you can then use a public NTP service to synchronize each of the testers. Please note that using a public NTP server will most likely be less accurate than the other solutions.

If you have N testers you will then need N client-only TimeKeeper licenses.

Remote Scenario (no Internet)

If you have two or more Xena testers in remote locations which are not connected to the Internet you can equip each tester with a GPS module.

If you have N testers you will then need N client-only TimeKeeper licenses and N GPS modules.

Important

Note that the SpectraTime GPS module must be purchased and installed through Xena Networks.

4.3.3 Test Module Configuration

By default, each test module will use its own internal clock for latency timestamps and scheduling traffic start. The internal module clock is synchronized with the general PCI clock on the chassis but two or more chassis will of course not share the same clock.

Enabling External Clock Sync

Perform the following steps to enable each test module to synchronize to the TimeKeeper-controlled Operating System (OS) kernel clock:

1. Open XenaManager and reserve the test module you want to configure.
2. Set the Latency Reference option to *Lock to External Time*.

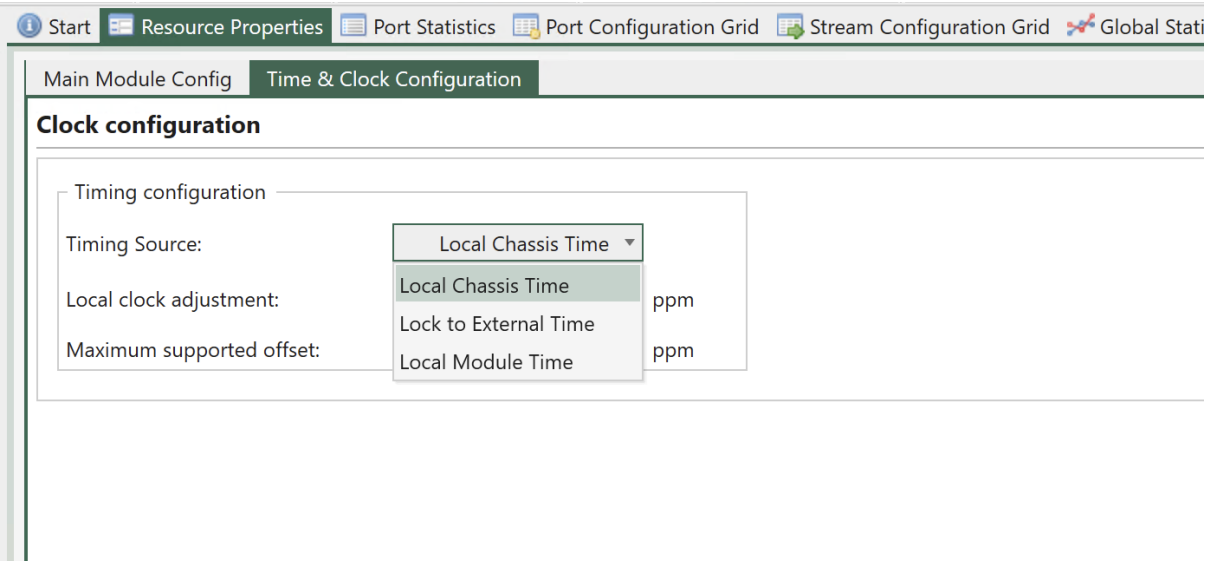


Fig. 4.13: Enabling external clock sync

The module will now attempt to synchronize its internal clock to the OS kernel clock.

Monitoring Clock Sync Accuracy

You can monitor the accuracy using the following state properties:

Table 4.5: Clock Sync State Properties

Name	Explanation
External Clock Diff	The currently measured difference between the OS kernel time and the local module time.
External Clock Stats	Statistics counters: • AP: Number of polls when in "Adjusting" state. • SP: Number of polls when in "Steady" state. • SE: Number of state transitions to "Steady" state. • SS: Number of "spikes" seen when in "Steady" state.
External Clock State	One of the following: • Adjusting: The module clock is adjusting to the OS kernel time. • Steady: The module clock has been within +/- 500 ns from the OS kernel time for the last 5 seconds.

The module clock will usually synchronize to the OS kernel time with approx. 15-20 seconds. But if the OS kernel time is also being adjusted by TimeKeeper you will experience a larger adjustment period.

4.4 Testbed Management

This page provides instructions on how to handle different testbeds within a test configuration.

A testbed is essentially a set of ports that you are actively utilizing. Certain panels within XenaManager will exclusively display data for ports that are part of your ongoing testbed. This encompasses both the **Port Configuration Grid**, **Stream Configuration Grid**, as well as **Global Statistics**.

4.4.1 Configuration Hierarchy

The top-most configuration entity you work with is the test configuration file. This file contains all information about these items:

- Connected chassis
- Testbeds
- Currently selected testbed

All testbeds thus share the same pool of chassis (and by extension their ports).

4.4.2 Setting Current Testbed

You can only have one active testbed at a time. The active testbed is selected with the *Current Testbed* control at the top of the *Available Resources* panel.

To select a testbed as the current you can either select the radio button in the *Select* column or you can simply double-click on the testbed entry.

When you change the selected testbed the content of all the dependent panels (see below) will also change.

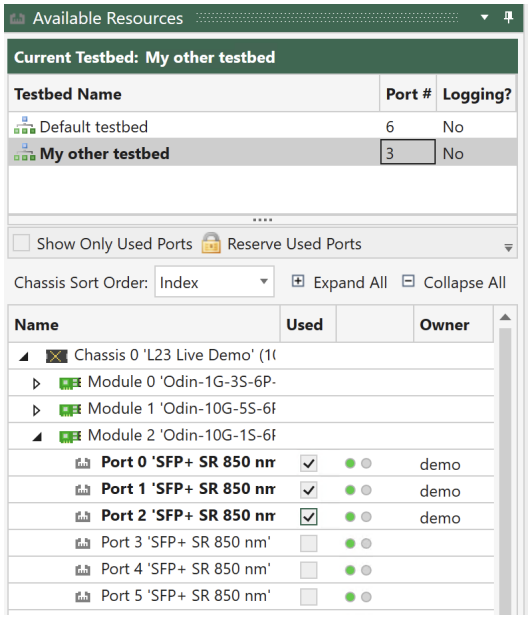


Fig. 4.14: Setting current testbed

4.4.3 Creating Testbed

You can create a new testbed definition by clicking the *Create Testbed* button in the ribbon menu at the top of the application. You will then be presented with a window where you can provide a unique name for the new testbed and optionally also provide a longer description.

The description will be shown as a tooltip when you hover with the mouse over the testbed selector.

When you create a new testbed this will automatically be set as the currently selected testbed.

4.4.4 Adding and Removing Testbed Ports

You can add a port to your currently active testbed by selecting the checkbox in the *Used* column next to the port name in the *Available Resources* tree view. You remove a port from your testbed by deselecting the checkbox.

You can also select multiple ports in the tree view, right-click and select the *Use Ports* menu item. This also works when you want to deselect multiple ports.

You can quickly reserve all ports in your current testbed by clicking the Reserve Used Ports button just below the testbed selector.

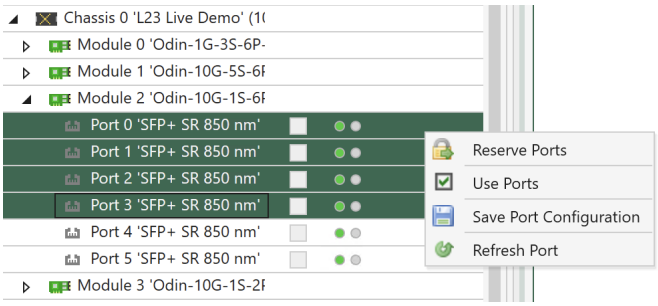


Fig. 4.15: Add and remove testbed ports

Note

Note that your reserved testbed ports will not be automatically released when you change your current testbed.

4.4.5 Editing Testbed

You can edit both the testbed name and the description by clicking the *Edit Testbed* button in the ribbon menu.

4.4.6 Removing Testbed

To remove a testbed you need to select the testbed with the testbed selector and then click the *Remove Testbed* button in the ribbon menu.

You can select multiple testbeds for removal at the same time using either the *Remove Testbed* button in the ribbon menu or right-click and select *Delete Testbed* in the pop-up menu.

Note

Note that your reserved testbed ports will not be automatically released when you change your current testbed.

4.5 Resource Management

4.5.1 Chassis Resources

A *chassis resource* can encompass the chassis itself, a test module residing within the chassis, or a test port situated on a module.

Xena testers have the capability to facilitate multiple concurrent connections from various Xena clients, including XenaManager, scripting clients, Xena2544, and so on. Once a client has successfully established a connection with the chassis, they can inspect any chassis resource. However, in order to modify the resource configuration, the client must initially reserve the resource.

4.5.2 Reservation Mechanism

At any given time, a specific resource can only be reserved by a single client. This reservation persists even in the event of a client disconnection. If the client later reconnects and authenticates itself with the same username, any previously pending reservations will seamlessly transfer to the new connection.

The reservation is linked to a combination of the connection ID within the chassis and the provided username. The username functions as a tag for the reserved resource, with the chassis having no concept of actual user accounts. Multiple connections can use the same username, but each resource can only be reserved by a single connection simultaneously.

By default, the username for XenaManager is derived from the Windows username of the current user. You have the option to modify the username for XenaManager in the Options menu, with a maximum of 8 characters allowed for the username.

4.5.3 Reservation Hierarchy

Reservations operate on a hierarchical exclusive basis, which means that if a user like Albert has reserved a specific test module, then another user like Bertha will be unable to reserve any port on that module. The same principle applies to chassis-level reservations. It's important to note, however, that user Albert does not reserve individual ports on the test module by reserving the test module itself.

In typical traffic generation operations, there is usually no need to reserve entire modules or chassis. Port reservations are typically sufficient for regular operations. Reserving modules and chassis becomes necessary primarily when conducting tasks like system maintenance, software upgrades, or changing port types on specific modules.

4.5.4 Reserving Resource

To reserve a chassis resource, you can choose the resource in the tree view and then either click the corresponding button in the ribbon menu or right-click the resource and select the appropriate menu item.

Once the resource is reserved, all configuration options for that resource will become accessible.

For a swift reservation of all ports within your current testbed, you can simply click the *Reserve Used Ports* button located just below the testbed selector.

4.5.5 Releasing Resource

To release any resource that you have previously reserved, follow these steps:

1. Select the resource in the tree view.
2. Click the *Release Resource* button in the ribbon menu.

Alternatively, you can right-click the resource and choose the corresponding menu item to release it.

4.5.6 Relinquish Resource

To forcibly take a resource away from another user, you can choose the *Relinquish Resource* option. However, please note that you will be prompted to confirm this action before it is executed to ensure you intend to proceed with this action.

Before deciding to relinquish resources reserved by another user, it is advisable to confirm if that user has an active connection on the chassis. This precaution can help maintain good relationships with your co-workers.

To check for active connections on a chassis, follow these steps:

1. Select the chassis in the *Available Resources* tree view.
2. Activate the *Resource Properties* tab.
3. Look at the bottom of the chassis properties panel, where active connections are listed. This will allow you to verify if another user is actively connected to the chassis before proceeding.

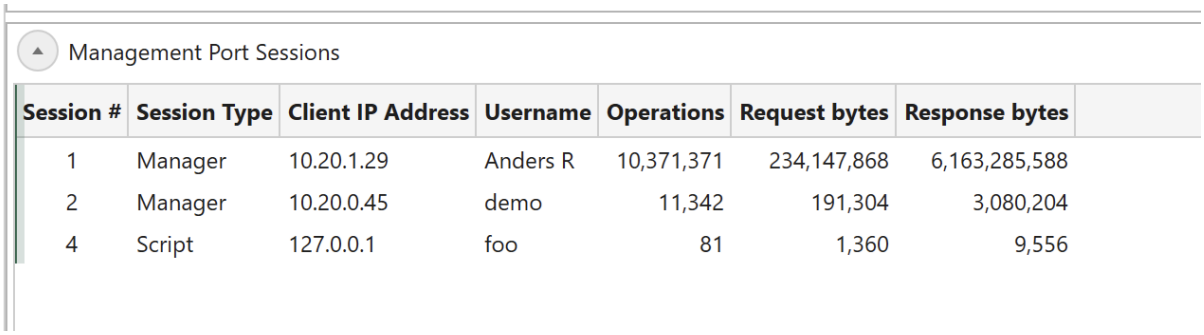


Fig. 4.16: Chassis session panel

4.5.7 Handling Multiple Resources

It is possible to operate on multiple resources in the tree view using the standard Windows `Shift+Click` or `Ctrl+Click` mouse operations.

4.6 Ribbon Menu

XenaManager incorporates a modern ribbon menu, akin to applications like Microsoft Word. Here, we provide explanations for each of the submenu items.

4.6.1 Edit Menu

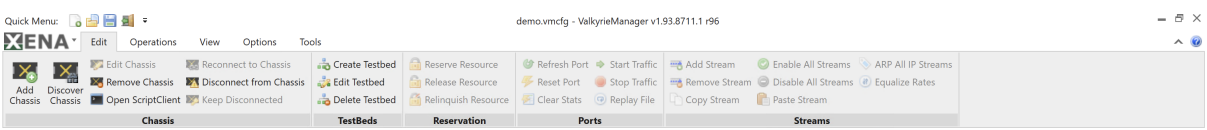


Fig. 4.17: Edit menu

This menu houses essential editing functions, and their availability depends on the context of your selection in the *Available Resources* tree view. Each function is explained in detail in other sections of this manual, and further explanations are not provided on this page.

4.6.2 Operations Menu

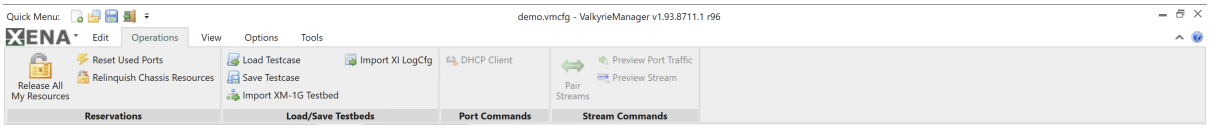


Fig. 4.18: Operations menu

This menu hosts functions capable of executing more intricate operations on one or multiple resources.

Release All My Resources

This button will release all the resources, including chassis, modules, and ports, that have been reserved by you. It provides a convenient way to clean up your reservations, especially when dealing with a large number of chassis, modules, and/or ports, without having to manually release each one individually through the *Available Resources* tree view.

Reset Used Port

This button will restore all test bed ports that you have reserved to their default settings.

Relinquish Chassis Resources

This button will relinquish all resources on the selected chassis that have been reserved by any user.

Import XM-1G Testbed

By clicking this button, you can import a legacy XenaManager testbed definition. All you need to do is click the button and choose the testbed file you've previously exported from the legacy XenaManager. This action will generate a new XenaManager testbed, incorporating the definitions from the legacy definition.

See also

XenaManager is the earlier software that has now been succeeded by XenaManager.

Load Test Case

This button allows you to load settings from a XenaManager test case file.

Save Test Case

This button allows you to save the complete port configurations for all ports in your current testbed.

Import XI LogCfg

This button allows you to import XenaIntegrator logging configuration.

DHCP Client

This button allows you to activate the DHCP client for a single selected port.

Pair Streams

This operation works on two streams that are defined on different ports. To perform this operation, you need to select the two streams in the *Available Resources* tree view.

When invoked, this operation will ensure that certain fields in the defined packet headers for each of the two streams point to the other stream as follows:

- Ethernet segment: The DMAC Address field will be set to the MAC address of the peer port.
- IPv4/IPv6 segment: The Destination IP Address field will be set to the defined IP address for the peer port.

This configuration ensures that when traffic is initiated, the traffic from one port will effectively reach the other port. For IP traffic, it's important to note that if the two ports are located on different IP subnetworks, you may need to resolve the IP gateway MAC address using ARP for proper communication.

Preview Port Traffic

This operation allows you to preview the actual packets that will be sent on a port before initiating a test. To utilize this function, you need to select a port stream in the *Available Resources* tree view.

When invoked, this operation will perform the following actions:

1. Stop traffic on the port if it is currently active.
2. Set the port in Tx(off)-to-Rx loopback mode.
3. Setup and start capture on the port itself.
4. Start traffic on the port.
5. Let the traffic run until the capture buffer runs full. The traffic will also be stopped after 10 seconds if the buffer is still not full.
6. Collect the captured packets and save them to a temporary file.
7. Restore the saved port and stream settings.

8. If Wireshark is installed on your system, it will be launched to view the captured packets when you use this feature. However, if Wireshark is not installed, you will need to utilize the *Capture* panel within the application to inspect the packets.

Preview Stream

This operation allows you to preview the actual packets that will be sent on a stream before initiating a test. This feature is particularly valuable if you have applied one or more modifiers to the stream and want to verify that the resulting packets appear as expected.

The function require that you select a single stream in the *Available Resources* tree view.

When invoked, this operation will perform the following actions:

1. Stop traffic on the port if it is currently active.
2. Disable all other streams on the port after saving their initial state.
3. Set the port in Tx(off)-to-Rx loopback mode.
4. Setup and start capture on the port itself.
5. Start traffic on the port.
6. Let the traffic run until the capture buffer runs full. The traffic will also be stopped after 10 seconds if the buffer is still not full.
7. Collect the captured packets and save them to a temporary file.
8. Restore the saved port and stream settings.
9. If Wireshark is installed on your system, it will be launched to view the captured packets when you use this feature. However, if Wireshark is not installed, you will need to utilize the *Capture* panel within the application to inspect the packets.

4.6.3 View Menu

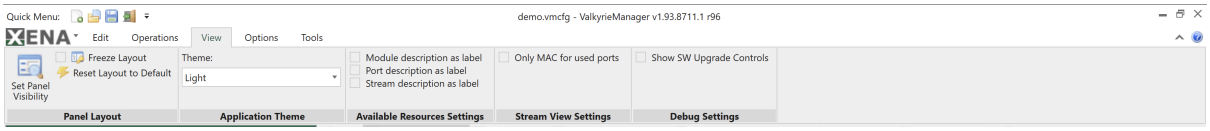


Fig. 4.19: View menu

This menu contains functions that affect the visual appearance of the application.

Panel Layout

Checking the *Freeze Layout* checkbox disables the ability to show or hide panels and to drag panels to other docking positions or to make them float-able. You can use this to protect yourself against unintended changes.

Clicking the *Set Panel Visibility* button will open a dialog that allows you to control the visibility for each of the function panel tabs available in the application. You can also hide any of the panels by selecting it and then clicking the little "X" to the right of the tab panel header as shown

in the adjacent example. To bring the panel back you can use the above mentioned dialog and click the checkbox next to the name of the hidden panel.

When you make changes to the layout the new layout will be restored when you startup the application again. The Reset Layout to Default button will delete the saved layout. The next time you start the application the original layout will thus be restored.

Application Theme

Select the layout theme for XenaManager from the menu.

Available Resource Settings

If you check the *Module description as label* option, the module description label will be used to name the module entries in the resource tree view instead of using the default module number identification.

If you check the *Port description as label* option, the port description label will be used to name the port entries in the resource tree view instead of using the default port number identification.

If you check the *Stream description as label* option, the stream description label will be used to name the stream entries in the resource tree view instead of using the default module/port number identification.

Stream View Settings

When defining MAC addresses in the *Stream Packet Header Definitions* panel, you can limit the list of selectable MAC addresses to those belonging to ports marked as used by checking *Only MAC for used ports*.

Debug Settings

This section contains settings intended for advanced users. The *Show SW Upgrade Controls* will unlock the manual software upgrade control in the chassis and module properties. This is as indicated only recommended for advanced users who fully understand what they are doing.

4.6.4 Options Menu

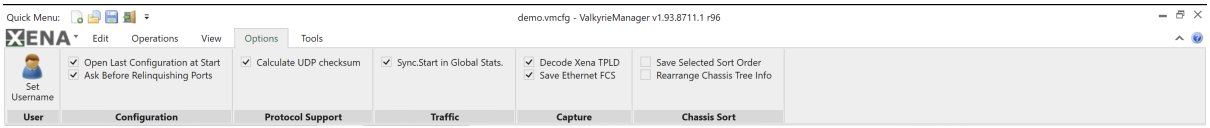


Fig. 4.20: Options menu

This menu contains various functions that affect the behavior of the application.

Set Username

The default username for the application is your Windows username. You can change this with this function.

Open Last Configuration as Start

If this option is checked the configuration file that was active when you closed down the application the last time will automatically be re-opened on the next application start.

Ask Before Relinquishing Ports

If this option is checked you will be asked to confirm if you really want to relinquish ports reserved by other users. This is also the recommended setting.

Calculate UDP Checksum

For modules not supporting automatic UDP checksum calculation: Attempt to calculate UDP checksums based on a static header with all-zero payload and no TPLD.

Sync Start in Global Stats

If this option is checked the Start button in the Global Statistics panel will use a synchronized port start mechanism for the ports if the chassis firmware version supports this feature.

Decode Xena TPLD

When this box is checked the XenaManager capture function will attempt to decode and show the Xena TPLD in captured packets.

Save Ethernet FCS

When this box is checked the XenaManager capture function will save Ethernet *FCS* when generating PCAP files.

Save Selected Sort Order

Saves the Chassis Sort Order defined for the Available Resources tree. The saved Selected Sort Order will be used when the XenaManager is re-launched.

Rearrange Chassis Tree Info

When this box is checked the chassis information in the Available Resources tree will be shown, starting with the selected Chassis Sort Order criteria.

4.6.5 Tools Menu

This menu contains various shortcuts to other tools.

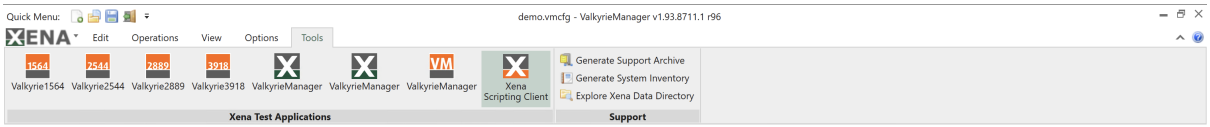


Fig. 4.21: Tools menu

Xena Test Applications

This section will show an icon for each of the other Xena test applications installed together with the XenaManager, **Xena2544**, **Xena2889**, **Xena3918**, and **Xena1564**. You can launch each of these applications by pressing the icon button.

Support

If you click the Generate Support Archive button the application will create a compressed ZIP archive containing both the currently loaded configuration file and the content of the Logs and Settings directories. This file can then be emailed to your support representative.

Clicking the Explore Xena Data Directory will open a Windows Explorer in the data directory for the XenaManager. Here you can find configuration and settings files, log files and any support archive files you may have created.

4.6.6 Minimizing Ribbon

The ribbon menu will by default be shown fully expanded. In order to free up screen space you can minimize it by clicking the arrow next to the *Help* icon in the menu title line as shown below.



Fig. 4.22: Minimize ribbon using arrow icon

You can also use the little arrow in the *Quick Menu* strip as shown below.

4.6.7 Additional Features

Application Menu

The application menu can be accessed by clicking the Xena icon in the top left corner.

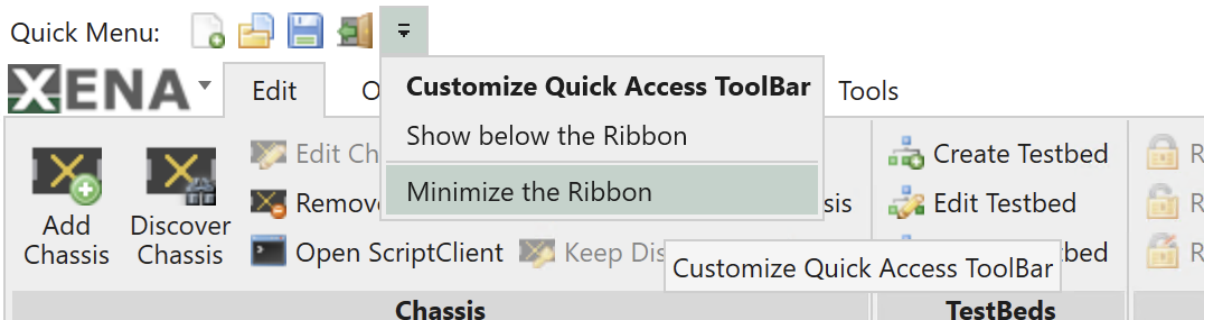


Fig. 4.23: Minimize ribbon using quick menua

Using the functions in the *Configurations* section you can either create a new test configuration, load an existing configuration from file, or save the current configuration.

The *Recent Configurations* section in the middle allow you to load any recently loaded or saved configurations.

Quick Menu Toolbar

The *Quick Menu* toolbar at the top of the application provides easy shortcuts to the most used application-level commands.

4.7 Saving and Loading Configuration

The actual configuration of the test modules, test ports, streams, and other similar settings are not saved as part of the testbed configuration. This configuration is typically stored on the test chassis themselves. This approach offers the advantage of making the configuration accessible to all connected users, ensuring consistency and shared access to the configuration settings.

While the configuration settings for ports and modules on the test chassis are not inherently persistent, you have the option to manually save these configurations to one or more local files on your PC if you wish to preserve them.

4.7.1 Working With Test Port

Saving Port Configurations

You can save all configuration parameters for a port to a single file thus enabling you to restore them at a later stage. This includes all port-level parameters such as filters, histograms and capture setup and also all stream and modifier configuration for that port.

To save the configuration for a port you simply right-click on the port and select *Save Port Configuration* as shown in the Fig. 4.24. You will then be asked for a filename and location for the configuration file.

You can also select multiple ports and save their configurations in a single operation.

The port configuration will be saved to a file with extension `*.xpc` (Xena Port Configuration). Each `*.xpc` file will only contain the configuration for a single port. So if you select multiple ports you will get one configuration file for each port.

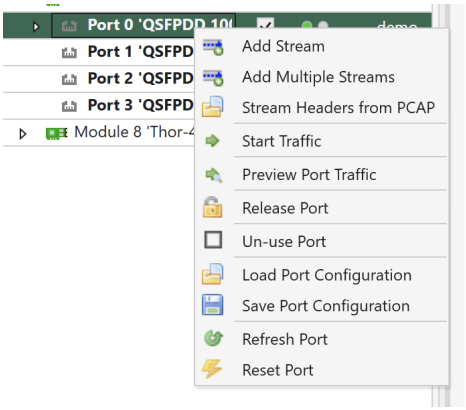


Fig. 4.24: Save port configuration

If you want to save multiple port configurations to a single file please refer to the following sections regarding testbed configurations.

Note

Note that you do not have to reserve the port in order to save its configuration.

Restoring Port Configurations

You can subsequently restore a port configuration from a saved configuration file. This means that all existing configuration on that port will be replaced with configuration from the loaded file.

To restore a port configuration for a port you simply right-click on the port and select *Load Port Configuration*, as shown in as shown in the Fig. 4.24. You will then be asked for a filename and location for the configuration file.

You can also select multiple ports and select to load their configuration form a single file. Please see the next section for information about issues when loading a configuration to another port than it was saved from.

Note

Note that you will have to reserve the port in order to load its configuration.

Moving Port Configuration

It is possible to load a port configuration on a different port than the one it was saved from.

If the port type of the new port is the same as the original port the operation is generally trivial. If the two ports are different certain port parameters may fail to load on the new port but this will not prevent the remaining parameters to load. The XenaManager will inform you about any failing parameters.

- MAC and IP Address Issues

The port MAC address and IPv4/IPv6 addresses are all saved as part of the port configuration. So if you load a port configuration from a different port you will thus also assign the MAC and IP addresses of the old port to the new port. Usually this is not what you want so the XenaManager will warn you about this and ask you what you want to do. You will then be given the option to preserve the original addresses of the new port.

- TID Issues

The various streams created on a port is also saved in the port configuration. This also includes the Test ID (*TID*) integer value for each stream. In most test scenarios it is important to have a unique TID value for each stream, at least inside a single testbed. Otherwise you will not be able to determine the source stream of a packet when it is received on a port.

If you load a port configuration from a different port then all streams from the original port will thus be recreated on the new port including the TID value assigned to the original streams. This may not be what you want so the XenaManager will ask you how you want to handle this. You will be given the option to either use the original value or to assign a new unique value to the new streams.

- IP Address Issues

If the streams defined in the port configuration contain an IP protocol segment the Source IP Address field in the protocol header will usually be set to the assigned port IP address. The XenaManager will ask you if you want to modify the protocol header fields to indicate the IP address of the new port or if you want to retain the original protocol header value.

4.7.2 Working With Test Module

You can save and load test module configurations. This can be useful if the test module configuration affects the test port type and number (as it does for e.g. Z400 Thor and Z100 Loki test modules) or if you are using the External Clock Sync function to synchronize the date and time across several test chassis.

The module configuration will be saved to a file with extension `*.xmc` (Xena Module Configuration). The operations are similar to the saving and loading of port configurations as described above.

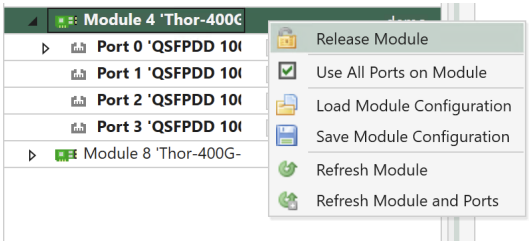


Fig. 4.25: Save module configuration

4.7.3 Working With Testbed

In the previous section we described how to save and restore individual port and module configurations. In this section we will describe how you can save a number of configurations to a

single file. This file is called a **test case** file and will have either extension `*.xtc` (old legacy format) or extension `*.xtc2` (new format).

This function works in the context of a testbed, i.e. it works for the ports that are included in the testbed plus the parent test modules for those ports.

Saving Testbed Configuration

You can save the configuration for all the ports in your testbed by using the *Save Testcase* menu item in the *Operations* menu.

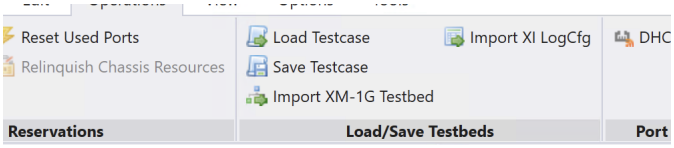


Fig. 4.26: Save testbed configuration

You will then be asked to enter a filename for the saved configuration. The default name will be the testbed name.

You can then select the format version for the saved configuration file. You can choose between these options:

- New format version (`*.xtc2`) which also support saving of the parent module configurations.
- Old (v1) format (`*.xtc`) which only support saving the port configurations.

If you choose the new format version you will be asked if you also want to save the parent module configurations in the testcase. You should choose this if your port configuration requires a certain module configuration.

Restoring Testbed Configurations

You can restore a full testbed configuration by using the *Load Testcase* menu item in the *Operations* menu.

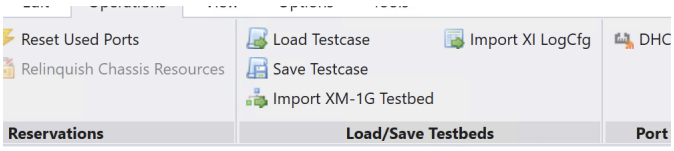


Fig. 4.27: Restore testbed configuration

Note

Note that the restore operation expects that all chassis, modules and ports which was involved in the original configuration save operation are still present. You cannot restore a testbed to a different set of chassis, module and/or ports.

4.8 UI Customization

This section describes the general layout and intended usage of the user interface elements in the XenaManager application.

4.8.1 Docking Panels

The XenaManager uses a so-called docking panel framework where each panel can be docked in various positions. The user can thus customize the layout of the application to some extent.

4.8.2 Docking Positions

Any panel can be docked in several positions. Fig. 4.28 below shows the three standard positions.

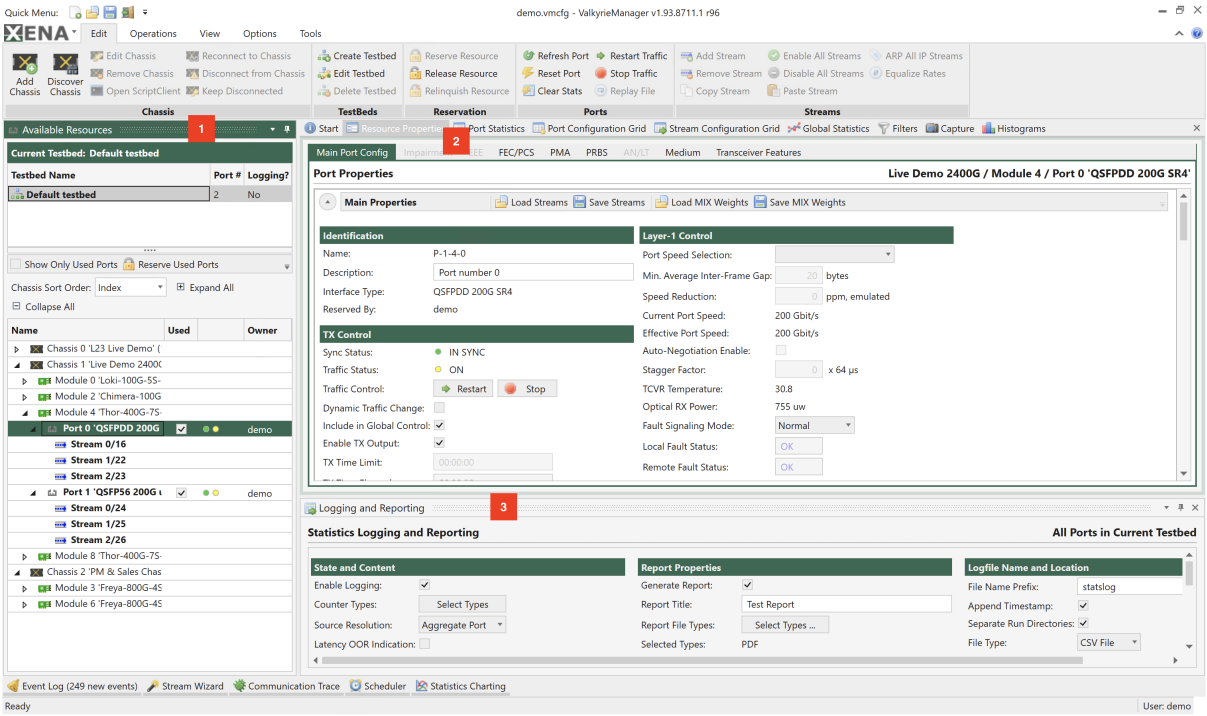


Fig. 4.28: Docking positions

1. Left
2. Center
3. Bottom

Note

It is also possible to dock a panel in these positions:

- Right (creating a sidebar similar to the Left position)
- Top (above the Document Center tab)

4.8.3 Moving Docking Panels

To move a panel to a new docking position perform the following actions:

1. Grab the tab header with the mouse and drag it to release it from the present location.
2. You will now see a *compass rose* with arrows in all four directions, as shown below.
3. Hover the mouse over the arrow that represents the position where you want the panel to go and release the mouse.
4. You can also hover over the center in the *compass rose* in which case you will target the center position.

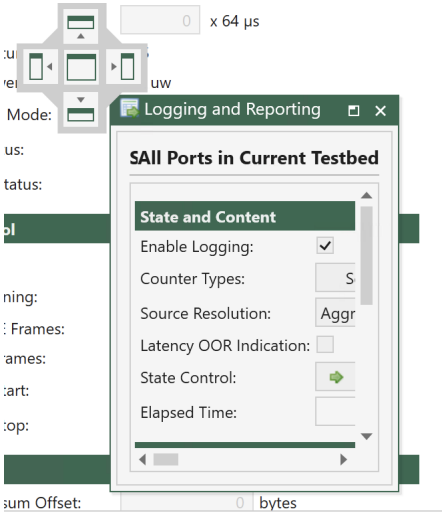


Fig. 4.29: Moving docking panel

4.8.4 Positioning Panels

You can change the relative position of a panel by grabbing the panel header with the mouse and drag it left or right within the position tab it is currently located in.

4.8.5 Floating Panels

You can also choose to let a panel float outside the docking framework. Just drag it loose from the current position and release it where you want it to be located.

4.8.6 Hiding Panels

You can hide a panel by right-clicking on the corresponding tap and select *Hide*.

4.8.7 Restoring Default Layout

If your layout gets messed up, you can easily revert to the default layout via the *View* menu and clicking the *Reset Layout to Default* button.

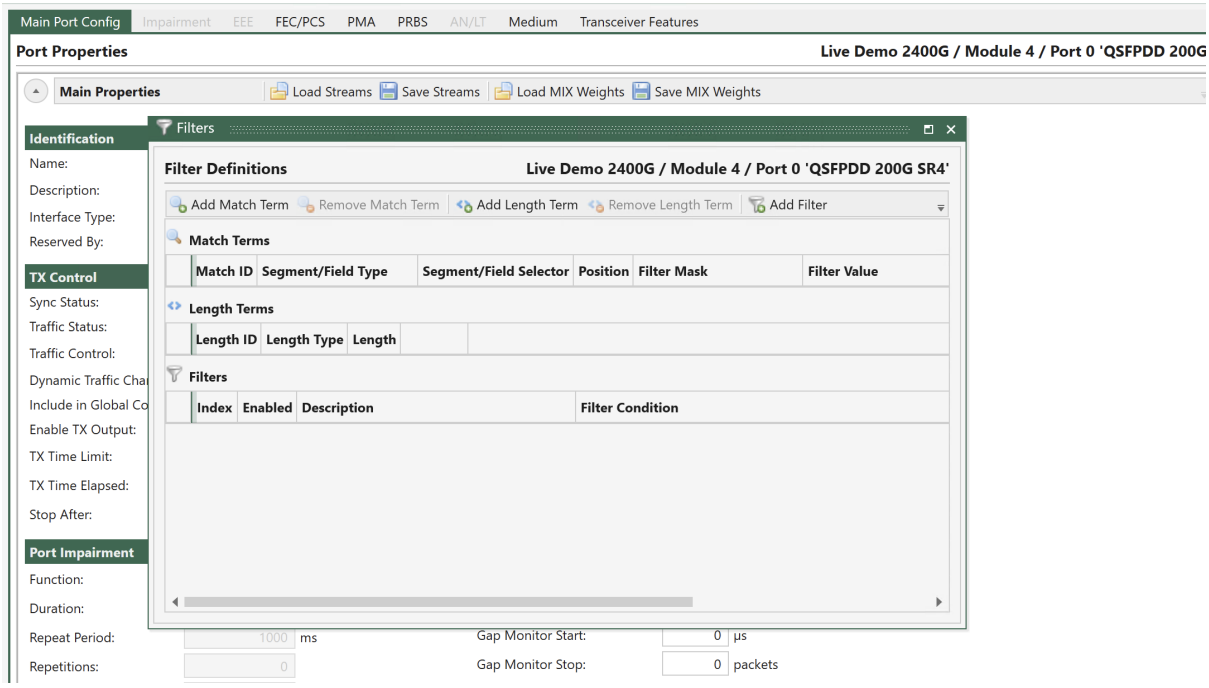


Fig. 4.30: Floating panels

4.9 CLI Script Client

Xena modules offer complete control through scripting commands, enabling access to all configuration settings and statistical data via scripting.

One approach to configuring Xena modules using script commands is by utilizing the built-in *Script Client* within XenaManager.

To open the script client, right click on the chassis and select *Open Script Client*. This is illustrated in Fig. 4.31.

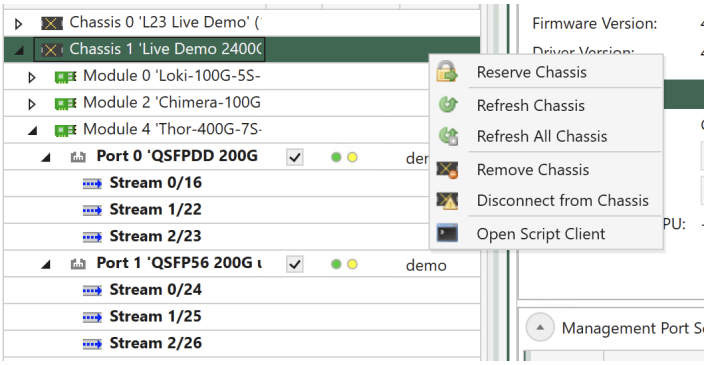


Fig. 4.31: How to open Script Client.

Once the script client is open, you can directly input script commands. The use of the script client is illustrated in Fig. 4.32.

When using the script client to enter script commands, there are two ways to specify the module and port on which to apply the command:

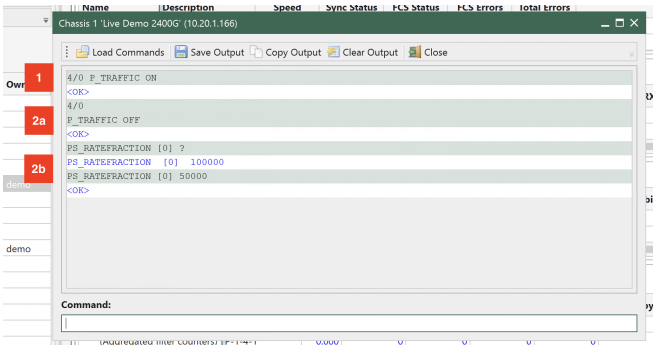


Fig. 4.32: Use the Script Client

1. Writing the <MODULE>/<PORT> in front of the command. (See Fig. 4.32 - step 1, which illustrates how to apply the command to module 7 and port 0).
2. First, enter the <MODULE>/<PORT> (Fig. 4.32 - step 2a). Subsequently, enter the commands, which will then be executed on the module and port previously configured (Fig. 4.32 - step 2b).

Because the module and port will depend on the chassis configuration, all script commands in the following will be listed without the module and port values, i.e. to use the script command examples directly in the script client, the user is required to configure the <MODULE> / <PORT> in advance as illustrated in Fig. 4.32 - step 2a.

Important

All CLI commands for Xena are documented on the following link: XOA CLI Documentation

4.10 Different Types Of Xena Configuration Files

Xena test applications keep their configurations in special file formats with an application-specific extension. The extensions used are as follows:

Table 4.6: Different Types Of Xena Configuration Files

File	Extension	Description
.vm-cfg	XenaManager configuration file in JSON format.	It contains information of the added chassis, UI settings, selected testbed, and used ports. But it doesn't recreate the streams on the port.
.v25	Xena2544 configuration in JSON format.	It contains information of the added chassis, UI settings, selected testbed, used ports, and test configuration.
.v39	Xena3918 configuration in JSON format.	It contains information of the added chassis, UI settings, selected testbed, used ports, and test configuration.
.v28	Xena2889 configuration in JSON format.	It contains information of the added chassis, UI settings, selected testbed, used ports, and test configuration.
.v15	Xena1564 configuration in JSON format.	It contains information of the added chassis, UI settings, selected testbed, used ports, and test configuration.
.xtc2	XenaManager Testcase (testbed) configuration file in Xena scripting API format.	It contains the configuration of the ports, streams on the ports. If you are to recreate your testbed (ports, streams), you should use Operations > Save Testcase in XenaManager.
.xtc1	Legacy configuration file of XenaManager Testcase (testbed).	Replaced by .xtc2 already.
.xpc	XenaManager Port configuration in Xena scripting API format.	It saves all information of a port including the streams on it.

CHAPTER

FIVE

TG AND L1

5.1 Overview

One of the prominent functionalities offered by XenaManager is its *TG* capability, which supports Ethernet speeds of up to 800 Gbps. This makes it well-suited for conducting both functional and performance testing. This section also provides an overview of the tab panels available within XenaManager.

5.1.1 General Purpose Tab Panels

Start

This tab panel is the default visible panel when you start the XenaManager for the first time. It contains a short Getting Started guide for the application. You can close this panel once you don't feel you need it anymore.

Available Resources

Available Resources Panel displays the resources (modules, ports and streams) for all configured chassis in your current testbed configuration.

Communication Trace

Communication Trace displays the raw detailed realtime communication with the chassis. It is mainly used for debugging the communication in case of problems but it can also be used as a help for users who wants to write automation scripts.

5.1.2 Selected Resource Panels

Resource Properties

Resource Properties will enable you to view and modify properties for the resource currently selected in the Available Resources tree view (chassis, module, port or stream).

Port Statistics

Port Statistics will display statistics counters for the port currently selected in the Available Resources tree view, including statistics for all streams on that port.

Filters

Filters will enable you to configure filters for the port currently selected in the Available Resources tree view.

Capture

Capture will enable you to configure capture settings for the port currently selected in the Available Resources tree view.

Histograms

Histograms will enable you to configure histograms for the port currently selected in the Available Resources tree view.

5.1.3 Testbed Centric Panels

Port Configuration Grid

Port Configuration Grid will enable you to view and modify properties for all ports in your testbed.

Stream Configuration Grid

Stream Configuration Grid will enable you to view and modify properties for the streams configured on all ports in your testbed.

Global Statistics

Stream Configuration Grid will display statistics counters for all ports in your testbed and also for all streams on those ports.

Statistics Charting

Statistics Charting will enable you to plot various statistics counters for selected streams.

Logging and Reporting

Logging and Reporting enables you to enable periodic logging of counters from your testbed ports. It also let you configure reports in PDF and HTML format of the counters.

Event Log

Event Log enables you to monitor logged events for the test ports.

5.2 Available Resources Panel

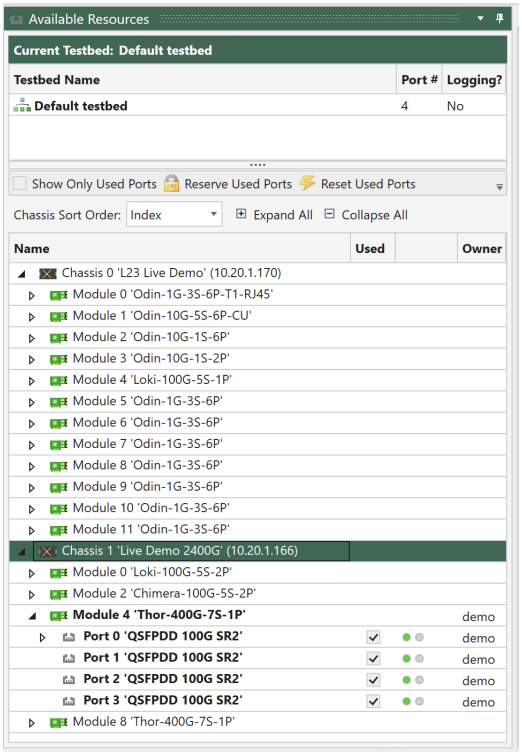


Fig. 5.1: Available Resources Panel

5.2.1 Testbed Selector

The testbed selector is located at the top of the panel. This functionality is explained in more detail on this page.

5.2.2 Resource Tree View

All available chassis resources are shown in the hierarchical tree view below the testbed selector. Each connected chassis is shown at the top-level with their contained resources below.

The content of certain of the other panels depend on the selection state of the Available Resources panel.

5.2.3 Toolbar Options

The toolbar at the top of the tree view provides quick options for viewing and reserving the resources.

5.2.4 Show Only Used Ports

Toggles between showing all available resources on all connected chassis or only the ports that you have chosen to include in your testbed.

5.2.5 Reserve Used Ports

Reserve all ports that you have included in your current testbed.

5.2.6 Release All My Resources

Releases all resources (chassis, modules and ports) that you may have reserved. This option may only be visible when you click the small down-arrow at the right of the toolbar.

5.2.7 Tree View Columns

The tree view contains the following columns:

Table 5.1: Tree view columns

Column	Explanation
Name	The unique name of the resource
Used	Indicates whether the resource is used by the current testbed. This column is only valid for testports.
(unnamed)	Contains icons representing the current sync (green: SYNC, red: NO SYNC) and traffic state (grey: traffic OFF, yellow: traffic ON) for a testport.
Owner	Username of the current owner of the resource, i.e. the user who has currently reserved the resource.

5.2.8 Multiple Selections

It is possible to operate on multiple resources in the tree view using the standard Windows `Shift+Click` or `Ctrl+Click` mouse operations.

5.2.9 Right-click Options

Each resource in the tree view supports a right-click menu, which contains various actions which are valid for the current resource state and type.

5.3 Resource Properties

5.3.1 Overview

This section describes the common XenaManager Resource Properties.

Viewing Resource Properties

The Resource Properties page provides a detailed view of all properties for a specific resource (chassis, module, port or stream). To view the properties for a given resource you must select the resource in the *Available Resources* treeview.

The properties are grouped together according to their functional area relation.

The page can display properties for a single resource at a time. If you want to view multiple ports or streams at the same time please refer to *Port Configuration Grid* or *Stream Configuration Grid*.

Editing Properties

In order to change properties for a resource you need to reserve the resource first.

> **Note**
>
> Note that certain properties may be disabled depending on the state of the resource. Most port and stream properties will for instance be disabled when traffic is active on the port.

Property Tooltip

Each property edit control is prefixed with a descriptive label. If you hover the mouse over the label an even more descriptive tooltip will be displayed.

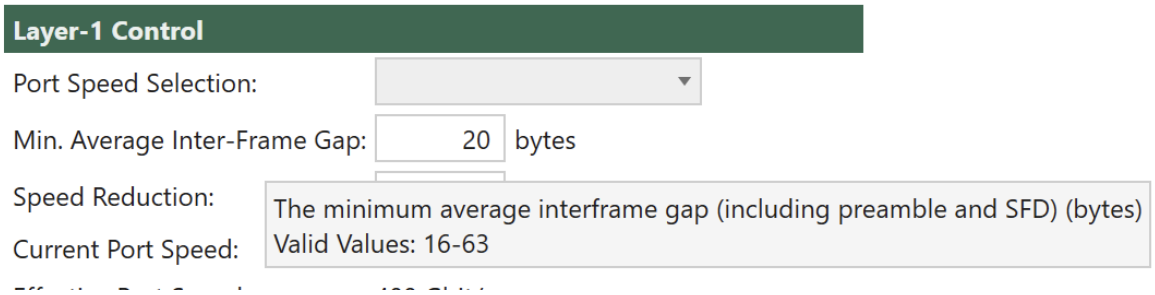


Fig. 5.2: Property tooltip

If the property only accepts values from a specific value range the tooltip will also show this information.

5.3.2 Chassis Properties

This section describes the available chassis properties for XenaManager.

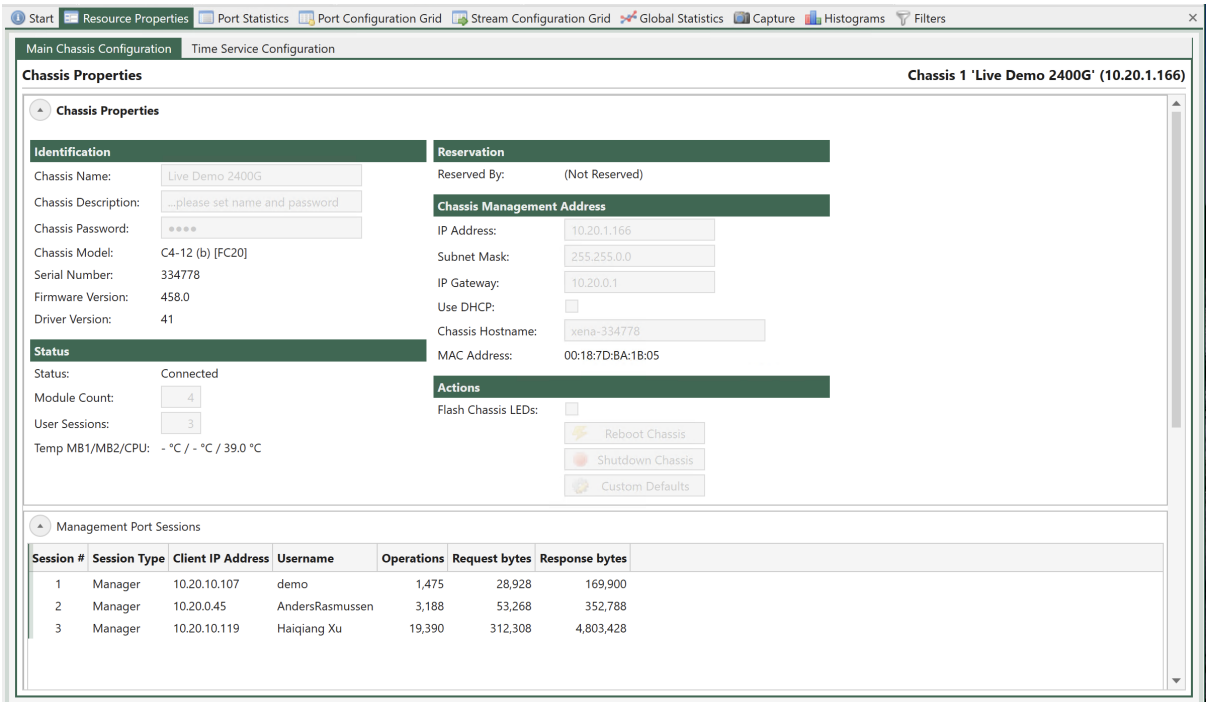


Fig. 5.3: Chassis Properties

Configuration and Status

Identification

Table 5.2: Identification

Property	Explanation
Chassis Name	The name you have administratively assigned to the chassis during installation.
Chassis Description	The description you have administratively assigned to the chassis during installation.
Chassis Password	The password you want users to provide when logging on to the chassis.
Chassis Model	The Xena chassis model identification. This value cannot be changed.
Serial Number	The Xena chassis serial number. This value cannot be changed. You should provide this number if you have to request technical support for your system.
Firmware Version	The version of the currently running chassis firmware. You should provide this number if you have to request technical support for your system.
Version Number	The currently loaded firmware version number. You should provide this number if you have to request technical support for your system.
Driver Version	The version of the currently used PCI bus driver firmware.
Firmware	You can use this button to upload single firmware image files to the chassis. It is however recommended that you use the ChassisUpgrader program to upgrade your chassis as this provides a more automatic and user-friendly upgrade mechanism. This option is only visible when you have enabled the Show SW Upgrade Controls option in the View menu.
Temp MB1/MB2/CPU	Temperature readings for motherboard and CPU.

Status

Table 5.3: Status

Property	Explanation
Status	The connection status of the chassis.
Module Count	The number of detected modules in the chassis
User Sessions	The number of current user sessions (connections)

Reservation

Table 5.4: Reservation

Property	Explanation
Reserved By	If the chassis is currently reserved by someone this field contains the username of the reserver.

Chassis Management Address

Table 5.5: Chassis Management Address

Property	Explanation
IP Address	The static management IP address of the chassis. See this page for details on how to modify the address.
Subnet Mask	The subnet mask for the management port of the chassis.
IP Gateway	The default gateway for the management port of the chassis.
Use DHCP	Checking this option will enable the chassis to obtain an IP address using DHCP. The static IP address control mentioned above will then not be used.
Chassis Hostname	The chassis hostname used when sending DHCP requests to a DHCP server. The default value is "xena-".
MAC Address	The MAC address of the main management port. You can use this when setting up a static address assignment in your DHCP server. Right-click on the value to copy the value to the clipboard.

Actions

Table 5.6: Actions

Property	Explanation
Flash Chassis LEDs	When enabled this property will cause the chassis LEDs to flash, making it easier to identify it if you have several Xena test chassis installed.
Reboot Chassis	This button will reboot the Xena chassis. You can use this to recover from an error situation or if you have changed the chassis IP address.
Shutdown Chassis	This button will shutdown the chassis. Note that you will have to manually power-cycle the chassis to bring it up again!

Management Port Sessions

This table shows the currently active user sessions on the chassis. You can, for instance, use this to check if a user that has reserved a resource you want to use is currently active.

5.3.3 Module Properties

Clicking on a module will reveal the module properties within the *Resource Properties* tab.

Main Module Config

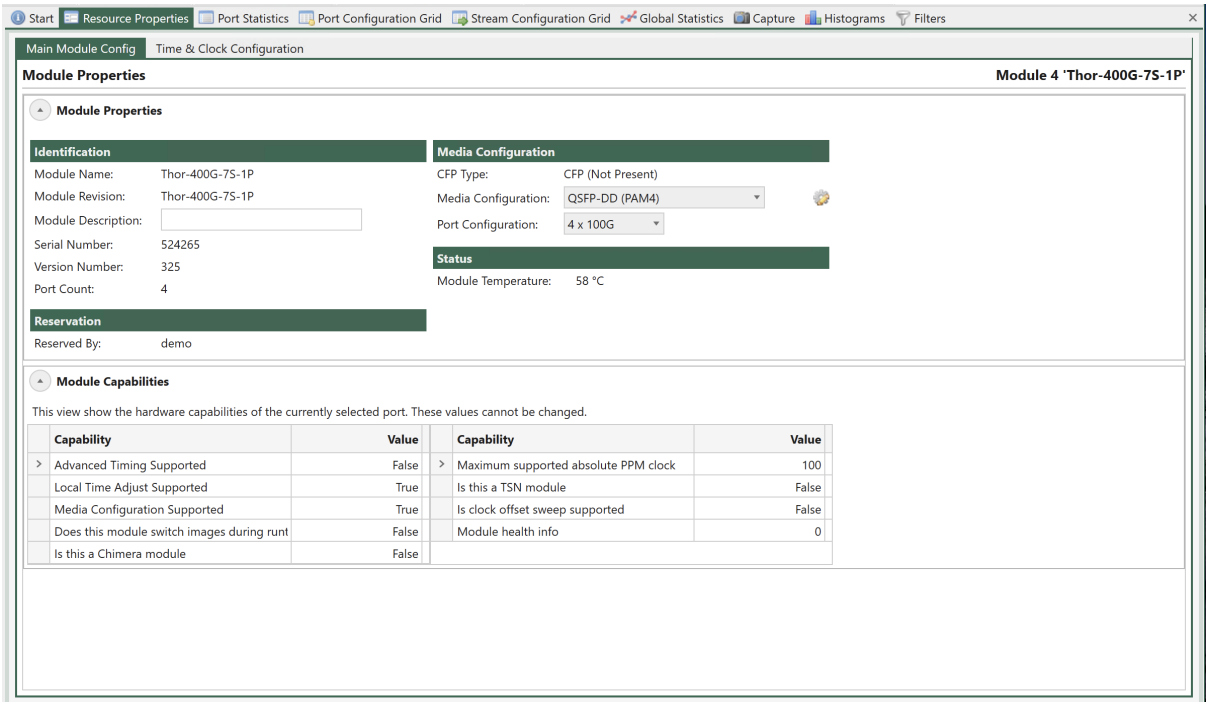


Fig. 5.4: Module Properties

Identification

Table 5.7: Identification

Property	Explanation
Module Name	The Xena test module model type.
Module Revision	The Xena test module model type plus revision information if relevent.
Module Description	A user defined description of the Xena test module.
Serial Number	The Xena test module serial number.
Version Number	The currently loaded firmware version number.
Firmware	This button allows you to manually upgrade this module with a firmware image that has been uploaded using the controls in the chassis panel. Usually, it is recommended that you use the firmware upgrade functions in the ChassisUpgrader program as this provides a more automated and user-friendly approach.
Port Count	The number of detected ports on the module.

Status

Table 5.8: Status

Property	Explanation
Module Temperature	The current module temperature in degrees (Celsius).

Reservation

Table 5.9: Reservation

Property	Explanation
Reserved By	If the module is currently reserved by someone this field contains the username of the reserver.

Media Configuration

Table 5.10: Media Configuration

Property	Explanation
CFP Type	Describes the type of CFP. The following values are supported • Not a CFP: This is not a CFP-based test module. • CFP (Not Present): No transceiver, the CFP cage is empty. • CFP (Not Flexible): Transceiver present, supporting a fixed speed and port-count. • CFP (Flexible): Transceiver present, supporting flexible speed and port-count. If CFP type is Not a CFP configuration and speed settings cannot be changed at module level. It may be possible to change speed at port level. For the other values it is possible to change Media Configuration i.e. which cage(s) are used on the module and if relevant in what configuration. It may also be possible to change Port Configuration i.e. number of ports and port speed.
Media Configuration	This property specifies which cage(s) are used on the module and if relevant in what configuration
Port Configuration	If enabled this property specifies number of ports and port speed on the module **Note** This property is not supported for non-CFP modules.

Time and Clock Configuration

Timing Configuration

Table 5.11: Timing configuration

Property	Explanation
Timing Source	Control how the test module time-stamp clock is running, either freely in the chassis, on the module itself or locked to external system time (requires the TimeSynch option). Running with free chassis ore module time allows nano-second precision measurements of latencies, but only when the transmitting and receiving ports are in the same chassis or on the same module. Running with locked external time enables inter-chassis latency measurements, but can introduce small time-discontinuities as the test module time is adjusted.
Local Clock Adjustment	Makes a small adjustment to the local clock of the test module, which drives the TX rate of the test ports. The property value is the desired adjustment from the nominal value, in parts-per-billion, positive or negative.
SMA Output Function	For test modules with SMA connectors, this property selects the function of the SMA output.
SMA Input Function	For test modules with SMA connectors, this property selects the function of the SMA input.
TX Clock Source	For test modules with advanced timing features, this property selects what drives the port TX rates.
TX and SMA Clock Filter	For test modules with advanced timing features, this property determines the loop bandwidth on the TX clock filter.

Clock Sweep

Note

Clock sweep function is not available across all modules.

Clock offset sweep provides the ability to let the system make controlled clock variations on all ports on a module. The function works together with the Timing configuration function.

Clock Offset Sweep

Sweep will be done as a symmetrical increase and decrease in ppm starting from the configured *Local clock Adjustment*

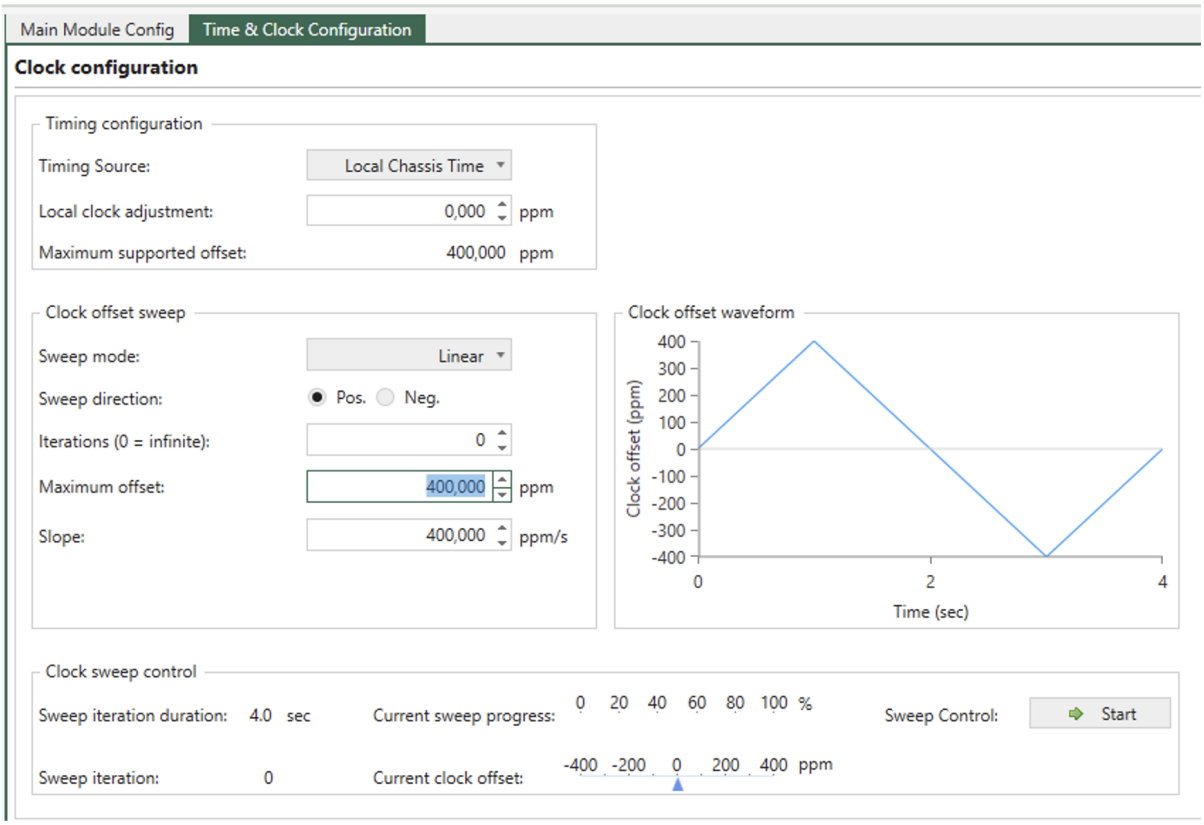


Fig. 5.5: Clock sweep

Table 5.12: Clock offset sweep

Property	Explanation
Sweep mode	Determines if the sweep is done as a *Linear* or *Step* sweep. Step size and duration is calculated based on maximum offset and sweep iteration duration.
Sweep direction	Determines if the sweep should start with an increase or decrease in ppm
Iterations	Determines if the system performs a fixed number of iterations before stopping, or infinite iterations. Setting the parameter to *0*, will make infinite sweeps until stopped by the user
Maximum Offset	Determines the maximum ppm offset value possible. The value can be configured until ppm offset reaches the limit of *Maximum supported offset* deducted the numerical value of the *Local clock adjustment*.
Slope	Configures ppm/s. Adjusting slope will affect the sweep iteration duration.

Note

Max Sweep duration is 64 secs. It is not possible to start sweep unless Max Sweep duration is 64 sec or less

Clock Offset Waveform

Clock offset waveform shown in Fig. 5.5 illustrates the configured sweep configuration with the parameters listed above.

Clock Sweep Control

Table 5.13: Clock sweep control

Property	Explanation
Sweep iteration duration	Shows the duration (in seconds) of an iteration for the configured ppm sweep
Sweep iteration	Shows the number of iterations done of the configured ppm sweep
Current sweep progress	Shows the progress (in %) of the current iteration for the configured ppm sweep
Current clock offset	Indicates the value of the current clock offset (in ppm).
Sweep control	Start and stop ppm sweep **Note** PPM sweep can be done both with and without active traffic on the ports on the module.

5.3.4 Port Properties

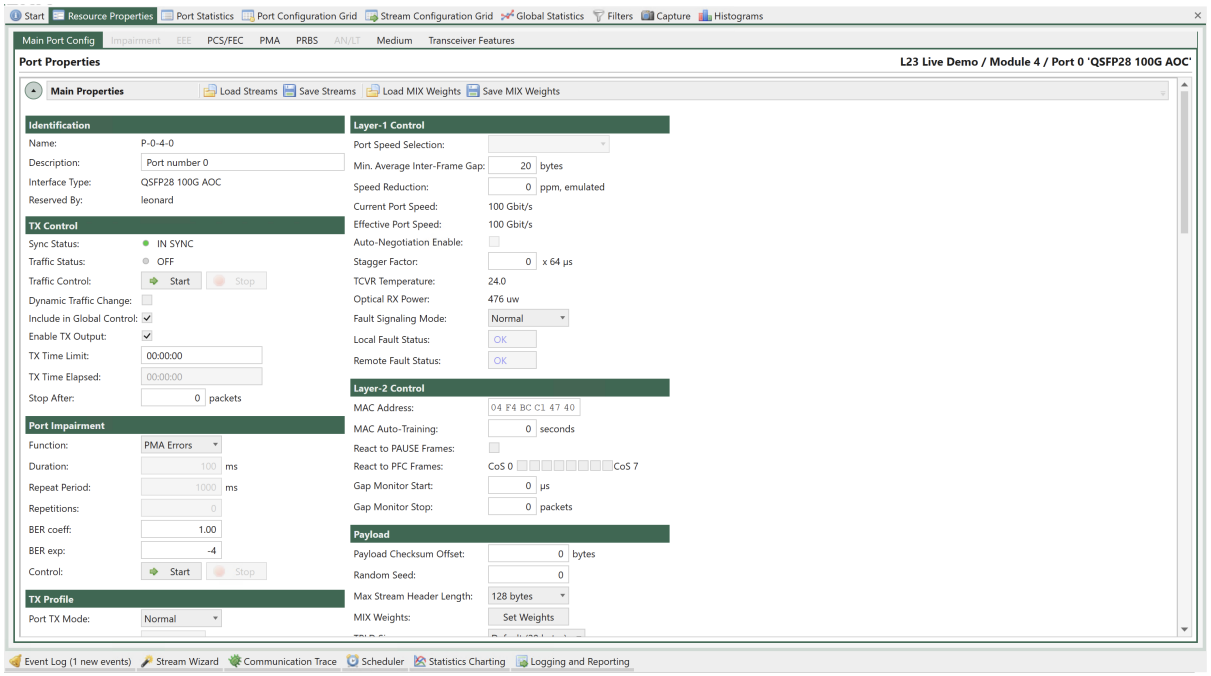
Clicking on a port will reveal the port properties within the *Resource Properties* tab. Within this tab, you will find sub-tabs associated with *TG*, as well as the Physical Layer.

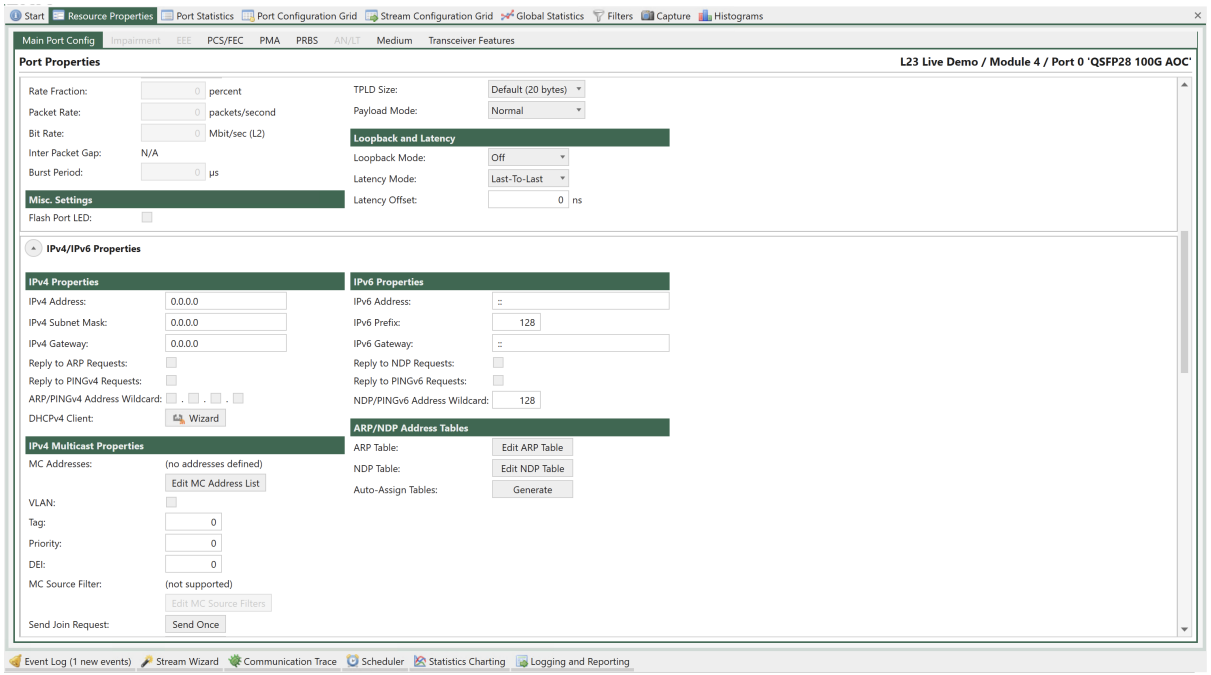
Main Port Config

Identification

Table 5.14: Identification

Property	Explanation
Name	The unique short-form name for the port
Description	A user-definable string label for the port
Interface Type	The Xena port interface type
Reserved By	If the port has been reserved by a user, this field will show the username of the reserver.


Start
Resource Properties
Port Statistics
Port Configuration Grid
Stream Configuration Grid
Global Statistics
Filters
Capture
Histograms
Main Port Config
Impairment
EEE
PCS/FEC
PMA
PRBS
AN/LT
Medium
Transceiver Features
Port Properties
L23 Live Demo / Module 4 / Port 0 'QSFP28 100G AOC'
Main Properties
Load Streams
Save Streams
Load MIX Weights
Save MIX Weights
Identification
Name: P-0-4-0
Description: Port number 0
Interface Type: QSFP28 100G AOC
Reserved By: leonard
TX Control
Sync Status: IN SYNC
Traffic Status: OFF
Traffic Control: Start Stop
Dynamic Traffic Change:
Include in Global Control:
Enable TX Output:
TX Time Limit: 00:00:00
TX Time Elapsed: 00:00:00
Stop After: 0 packets
Port Impairment
Function: PMA Errors
Duration: 100 ms
Repeat Period: 1000 ms
Repetitions: 0
BER coeff: 1.00
BER exp: -4
Control: Start Stop
TX Profile
Port TX Mode: Normal
Layer-1 Control
Port Speed Selection:
Min. Average Inter-Frame Gap: 20 bytes
Speed Reduction: 0 ppm, emulated
Current Port Speed: 100 Gbit/s
Effective Port Speed: 100 Gbit/s
Auto-Negotiation Enable:
Stagger Factor: 0 x 64 µs
TCVR Temperature: 24.0
Optical RX Power: 476 uw
Fault Signaling Mode: Normal
Local Fault Status: OK
Remote Fault Status: OK
Layer-2 Control
MAC Address: 04 F4 BC C1 47 40
MAC Auto-Training: 0 seconds
React to PAUSE Frames:
React to PFC Frames: CoS 0 CoS 7
Gap Monitor Start: 0 µs
Gap Monitor Stop: 0 packets
Payload
Payload Checksum Offset: 0 bytes
Random Seed: 0
Max Stream Header Length: 128 bytes
MIX Weights: Set Weights
Event Log (1 new events)
Stream Wizard
Communication Trace
Scheduler
Statistics Charting
Logging and Reporting


Start
Resource Properties
Port Statistics
Port Configuration Grid
Stream Configuration Grid
Global Statistics
Filters
Capture
Histograms
Main Port Config
Impairment
EEE
PCS/FEC
PMA
PRBS
AN/LT
Medium
Transceiver Features
Port Properties
L23 Live Demo / Module 4 / Port 0 'QSFP28 100G AOC'
Rate Fraction: 0 percent
Packet Rate: 0 packets/second
Bit Rate: 0 Mbit/sec (L2)
Inter Packet Gap: N/A
Burst Period: 0 µs
Misc. Settings
Flash Port LED:
TPLD Size: Default (20 bytes)
Payload Mode: Normal
Loopback and Latency
Loopback Mode: Off
Latency Mode: Last-To-Last
Latency Offset: 0 ns
IPv4/IPv6 Properties
IPv4 Properties
IPv4 Address: 0.0.0.0
IPv4 Subnet Mask: 0.0.0.0
IPv4 Gateway: 0.0.0.0
Reply to ARP Requests:
Reply to PINGv4 Requests:
ARP/PINGv4 Address Wildcard:
DHCPv4 Client: Wizard
IPv4 Multicast Properties
MC Addresses: (no addresses defined)
Edit MC Address List
VLAN:
Tag: 0
Priority: 0
DEI: 0
MC Source Filter: (not supported)
Edit MC Source Filters
Send Join Request: Send Once
IPv6 Properties
IPv6 Address: ::
IPv6 Prefix: 128
IPv6 Gateway: ::
Reply to NDP Requests:
Reply to PINGv6 Requests:
NDP/PINGv6 Address Wildcard: 128
ARP/NDP Address Tables
ARP Table: Edit ARP Table
NDP Table: Edit NDP Table
Auto-Assign Tables: Generate
Event Log (1 new events)
Stream Wizard
Communication Trace
Scheduler
Statistics Charting
Logging and Reporting

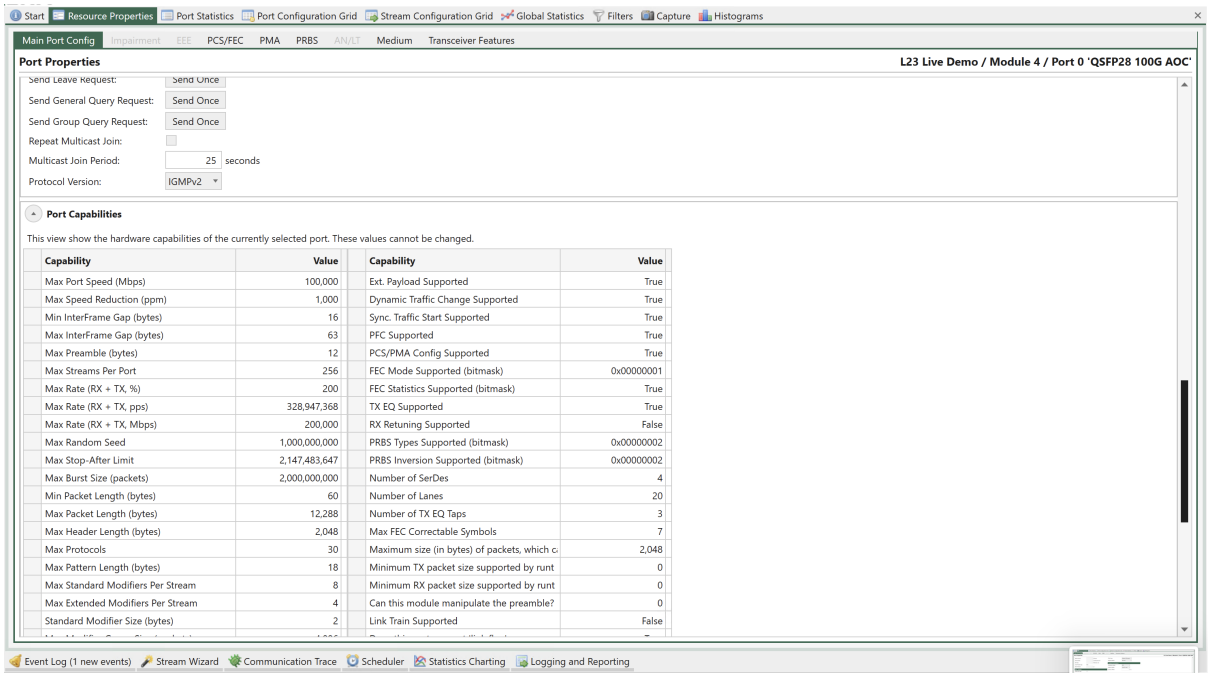


Fig. 5.6: Port Properties - Main

TX Control

Table 5.15: TX Control

Property	Explanation
Sync Status	The current sync state for the port. The port can either be `IN SYNC` (sync detected) or `NO SYNC` (no sync detected).
Traffic Status	The current traffic status for the port.
Traffic Control	This button enables you to either start or stop traffic on the port. Or restart traffic with dynamic changes seamlessly.
Dynamic Traffic Change	If this option is checked, the port will allow dynamic changes to the traffic while the traffic is running on the port. As soon as the Restart button is pressed, traffic is changed dynamically seamlessly.
Include in Global Control	If this option is checked and the port is part of the current testbed the port traffic state will be controlled by the Start/Stop buttons in the *Global Statistics* panel.
Enable TX Output	Determines if the port should enable its transmitter, or keep the outgoing link down
TX Time Limit	The maximum amount of time the port should transmit when enabled. If set to zero the port will transmit until stopped manually.
TX Time Elapsed	The amount of time the port has currently been transmitting
Stop After	Stop port transmission after the specified number of packets are sent

Note

(*) Feature is only supported by legacy 40G/100G ports. (**) Feature requires software release 76 or higher.

TX Profile

Table 5.16: TX Profile

Property	Explanation
Port TX Mode	This property determines the scheduling mode for outgoing traffic from the port, i.e. how multiple streams are merged onto one physical port.
Rate Fraction	The port-level rate of the traffic transmitted for a port in sequential TX mode, expressed as a percentage of the effective rate for the port. **Only available** when the *Port TX Mode* is set to *Sequential*.
Packet Rate	The port-level rate of the traffic transmitted for a port in sequential TX mode, expressed in packet per second. **Only available** when the *Port TX Mode* is set to *Sequential*.
Bit Rate	The port-level rate of the traffic transmitted for a port in sequential TX mode, expressed in bits per second. **Only available** when the *Port TX Mode* is set to *Sequential*.
Inter Packet Gap	The calculated mean inter-packet gap with the current TX profile settings. **Only available** when the *Port TX Mode* is set to *Sequential*.
Burst Period	Time in micro seconds from start of sending a group of bursts till start of sending next group of bursts. **Only available** when the *Port TX Mode* is set to *Burst*.

Important

The scheduling mode for outgoing traffic from the port, specifying how multiple logical streams are merged onto one physical port. There are four primary modes:

- **Normal Interleaved**: The streams are treated independently, and are merged into a combined traffic pattern for the port, which honors each stream's ideal packet placements as well as possible. **This is the default mode**.
- **Strict Uniform**: This is a slight variation of normal interleaved scheduling, which emphasizes strict uniformity of the inter-packet-gaps as more important than hitting the stream rates absolutely precisely.
- **Sequential**: Each stream in turn contribute one or more packets, before continuing to the next stream, in a cyclical pattern. The count of packets for each stream is obtained from the *Stop After* of the stream. The individual rates for each stream are ignored, and instead the overall rate is determined at the port-level. This in turn determines the rates for each stream, taking into account their packet lengths and counts. The maximum number of packets in a cycle (i.e. the sum of *Stop After* for all enabled streams) is 500.

If the packet number is larger than 500, will be returned when attempting to start the traffic.

- **Burst**: When this mode is selected, frames from the streams on a port are sent as bursts as depicted below: The Burst Period is defined in the *Burst Period*. For the individual streams, the number of packets in a burst is defined by the *Burst Size* in the stream transmission profile. Refer to the Fig. 5.7 below for an illustration of the burst mode.

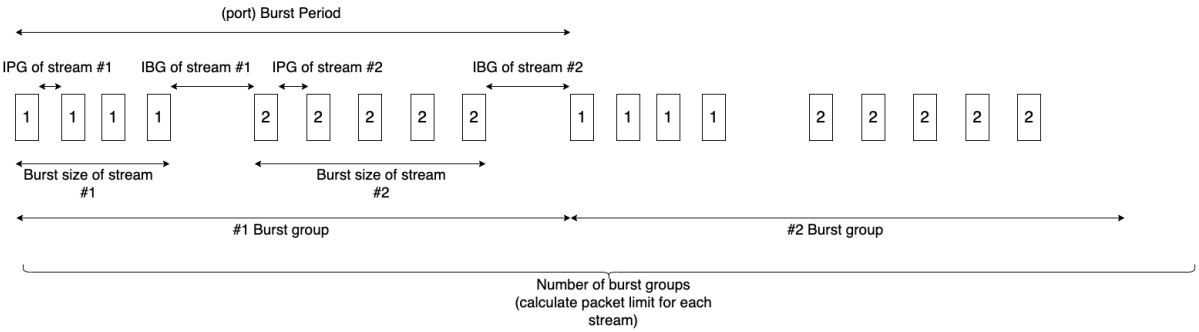


Fig. 5.7: Port TX Burst Mode

Port Impairments

This section describes impairments which will affect the entire port. I.e it will affect all streams defined for the selected port.

Link Flap

TG port can be configured to emulate that the physical link is down or unstable. This feature is called **Link Flap**. Link flap is implemented in 2 ways: **Logical Link Flap** and **Optical Link Flap**.

Notice that link flap is configured at a port level and will affect all flows configured for the selected port.

> **Note**
>
> Note that logical link flap and *PMA* error pulse inject (see Section *PMA Error Injection*) are mutually exclusive.

Logical Link Flap

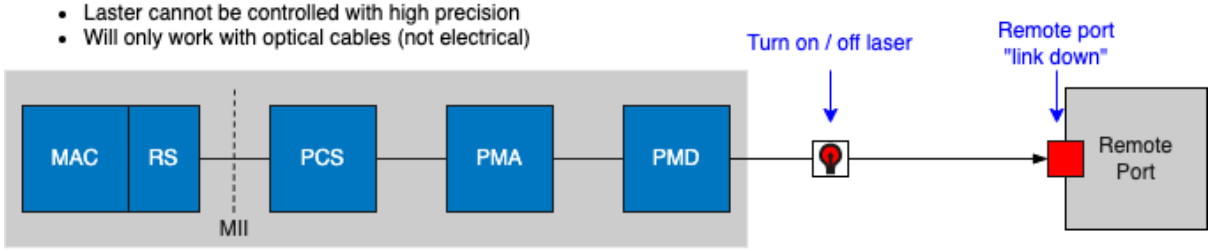
Link Flap under *Port Impairments* is **Logical Link Flap**. Logical link flap is implemented by scrambling the Tx *PCS* encoding to prevent the peer port from getting a link. It is not implemented by turning the physical transmitter on or off.

Logical Link Flap works for both electrical cables (*DAC* cables) and optical cables.

Logical link flap is configured under the Main Port Config tab as illustrated in Fig. 5.9.

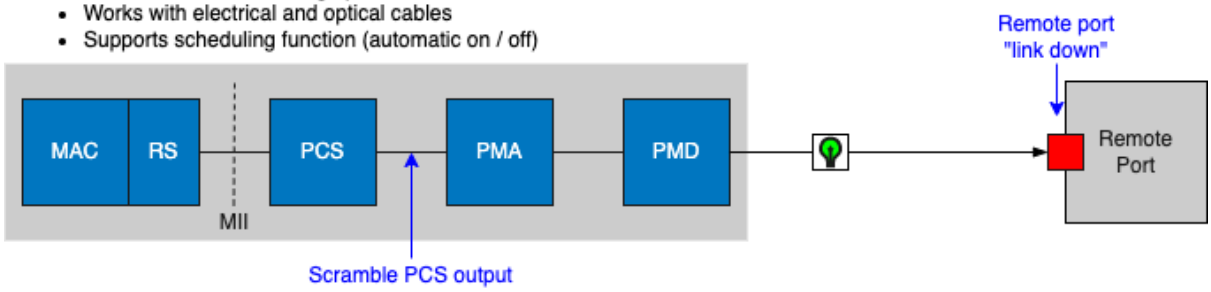
Optical Link Flap

- Manual on / off
- Laster cannot be controlled with high precision
- Will only work with optical cables (not electrical)



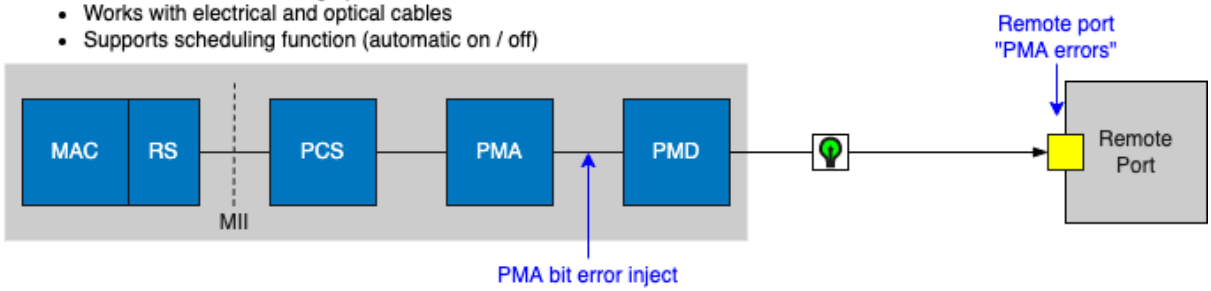
Logical Link Flap

- Scramble PCS output signal
- Can be controlled with high precision
- Works with electrical and optical cables
- Supports scheduling function (automatic on / off)



PMA Error Inject

- Insert bit errors at PMA level
- Can be controlled with high precision
- Works with electrical and optical cables
- Supports scheduling function (automatic on / off)



Scheduling Function

- 10 ms precision

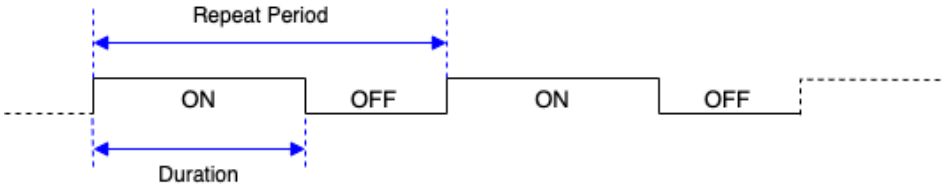


Fig. 5.8: Port impairments

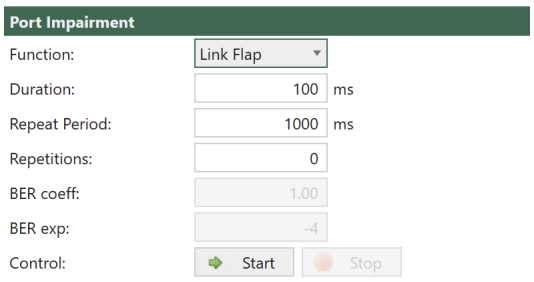


Fig. 5.9: Configuration of Logical Link Flap.

Logical Link Flap supports a repetitious pattern, where the link is taken down for a period (Duration) and then brought up again. This is repeated after a configurable time (Repeat Period). The flapping is repeated a configurable number of times or continuously (Repetitions).

Please observe that Link Flap is configured at a port level and will affect all streams configured for the selected port.

Pressing *Start* will start the configured link flap, pressing *Stop* will stop any ongoing link flapping.

Logical link flap is configured as follows:

Table 5.17: TG port link flap

Parameter	Description
Duration	Duration of the link flap.
Repeat Period	Period after which to restart link flap.
Repetitions	How many times to restart the link flap.

Optical Link Flap

To simulate the event of the optical link going down, it is possible to manually turn the optical transmitter off and on.

Optical link flap only works for optical cables, i.e., it will not work with *DAC* cables for instance. Optical link flap does not support repetitious patterns as described above for logical link flap.

Optical link flap is configured on the *Main Port Config* tab as illustrated in Fig. 5.10.

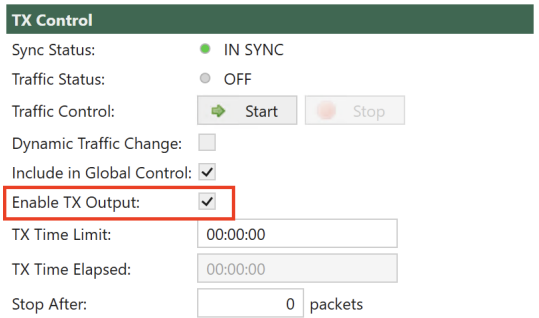


Fig. 5.10: Configuration of Optical Link Flap.

Use *Enable Tx Output* to turn the optical transmitter off / on.

PMA Error Injection

PMA Error Injection allows the user to insert bit errors onto the link.

Please observe that PMA Error Injection is configured at a port level and will affect all streams configured for the selected port.

Table 5.18: Port Impairments

Property	Explanation
Function	Enables Link Flap or PMA error injection.
Duration Link Flap	Time in ms the link is taken down. Range: 10 ms to 1000 ms; step size 1 ms.
Repeat Period	Time between link down. Range:10 ms to 50000 ms; step size 10 ms. "Repeat Period" must be larger than "Duration".
Repetitions Link Flap	Number of Link Flaps. Range: 0, 1 to 64K; step size 1. 0 = continuous until stopped.
BER Coeff	PMA Error injection: Bit Error Rate coefficient. Range 0.01 to 9.99; step size 0.01.
BER Exp	PMA Error injection: Bit Error Rate exponent. Range -17 to -3; step size 1
Control	Pressing *Start* will start the configured Link Flap/PMA Error injection, pressing *Stop* will stop any ongoing Link Flap/PMA Error injection.

Misc. Settings

Table 5.19: Misc. Settings

Property	Explanation
Flash Port LEDs	If checked, this property will make the test port LED for a particular port flash on and off with a 1-second interval. This is helpful when you need to identify a specific port within a chassis.

Layer-1 Control

Table 5.20: Layer-1 Control

Property	Explanation
Port Speed Selection	Controls the port speed selection. This property is only available for ports that support a configurable port speed.
Min. Average Inter-Frame Gap	The minimum average total interframe gap (including preamble and SFD)
Speed Reduction	Allows you to specify a speed reduction value for the port. The speed reduction is specified as a PPM value between 0 and 100 in steps of 10. The speed reduction is applied to the transmit side of a port, resulting in an effective traffic rate that is slightly lower than the rate of the physical interface. Speed reduction is effectuated by inserting short idle periods in the generated traffic pattern to consume part of the port's physical bandwidth. The port's clock speed is not altered.
Current Port Speed	The currently detected port speed
Effective Port Speed	The effective speed of the port taking any configured speed reduction into account.
Auto-Negotiation Enable	Controls whether the port will support auto-negotiation
BroadR-Reach Mode	Controls whether a BroadR-Reach transceiver will be in `Master` or `Slave` mode. This property is only shown when a BroadR-Reach transceiver is installed in the port or the port itself supports Automotive Ethernet. **Note** We support Technica Engineering BroadR-Reach transceivers. • TE-1440 Technica Engineering 100/1000Base-T1 Transceiver
Stagger Factor	This property delays the start of traffic generation on one port relative to the activation of global start. The delay is programmed in steps of 64 µs. The Stagger Factor will work between ports on test modules installed in the same chassis. NB: This requires that *Sync.Start in Global Stats*. under the *Options* tap has been checked.
TCVR Temperature	The currently detected transceiver temperature if supported by the transceiver.
Optical RX Power	The currently detected received optional power. This property value is only available for optical ports if supported by the transceiver.
Fault Signaling Mode	Sets the remote/local fault signaling behavior of the port (performed by the Reconciliation Sub-layer). The following modes can be configured: 1. Normal: Acts according to 802.3 standard: i.e. when receiving a bad signal, it transmits **Remote Fault indications** on the output and when receiving a **Remote Fault indication** from the far-side it will transmit IDLE sequences. By default, this mode is enabled, 2. Force Local: Port will continuously transmit **Local Fault indication** on the TX output (which is usually not allowed by the standard). 3. Force Remote: Port will continuously transmit **Remote Fault indication** on the TX output. 4. Disabled: Port will relay the traffic from the TX core regardless of what it receives on the input.

Layer-2 Control

Table 5.21: Layer-2 Control

Property	Explanation
MAC Address	The port MAC address
MAC Auto-Training	The interval in seconds with which the port should broadcast a MAC learning frame. Set to 0 to disable.
React to PAUSE Frames	Control whether the port should react to received PAUSE frames
React to PFC Frames	Control whether the port should react to received PFC (Priority Flow Control) frames. Use check boxes to select which priority levels the port reacts to.
Gap Monitor Start	Specifies the time period that will trigger the gap monitor start. The maximum allowed gap between packets, in microseconds, 0 to 134.000 microseconds. (0 = disable gap monitor)
Gap Monitor Stop	Specifies the number of packets to receive to stop the gap monitor. The minimum number of good packets required, 0 to 1024 packets. (0 = disable gap monitor)

Note

The gap-start and gap-stop criteria for the port's gap monitor. The gap monitor expects a steady stream of incoming packets, and detects larger-than-allowed gaps between them. Once a gap event is encountered it requires a certain number of consecutive packets below the threshold to end the event.

Refer to XOA CLI Documentation for more details.

Payload

Table 5.22: Payload

Property	Explanation
Payload Checksum Offset	The offset where payload checksum calculation starts. Valid values: 0; 8-127.
Random Seed	Used when generating traffic that requires random variation in packet length, payload, or modified fields
Max Stream Header Length	The maximum length of the defined stream headers. If you increase this you will at the same time reduce number of streams supported by the port. For a port that by default supports 256 streams
MIX Weights	Specify the weights for the MIX size packet distribution if supported by the port. **See also** Read details in *MIX Weights*.
TPLD Size	Specify the size of the *TPLD* for the port streams if supported by the port. • Default is **Normal**, which is a 20 byte TPLD. • **Micro** is a condensed version, which is useful when generating very small packets with relatively long headers (like IPv6). It has the following characteristics compared to the normal TPLD. – Only 6 byte long. – Less accurate mechanism to separate Xena-generated packets from other packets is the network - it is recommended not to have too much other traffic going into the receive Xena port, when micro TPLD is used. – No sequence checking (packet loss or packet misordering). The number of received packets for each stream can still be compared to the number of transmitted packets to detect packet loss once traffic has been stopped. – No payload error checking. When the *TPLD Size* is changed, it will affect **ALL** streams on the port. **See also** Read more at *Test Payload*.
Payload Mode	Specify the payload mode used for the port streams if supported by the port. The following options are available: • Normal: The packet payload type is determined by the Payload Type property on the streams. This is the default behavior. • Extended Payload: Enable support for the extended payload feature for streams on this port (not supported by all modules). Read more in *Stream Properties > Extended Payload*. • Custom Data Field: Enable support for the custom data field feature for streams on this port (not supported by all modules). Read more in *Stream Properties > Custom Data Field*

MIX Weights

Internet Mix or IMIX refers to typical internet traffic traversing network equipment such as routers, switches or firewalls. When measuring equipment performance using an IMIX of packets, the performance is assumed to resemble what can be seen in "real-world" conditions.

IMIX configuration is per port. It means each test port can be configured a certain IMIX pattern.

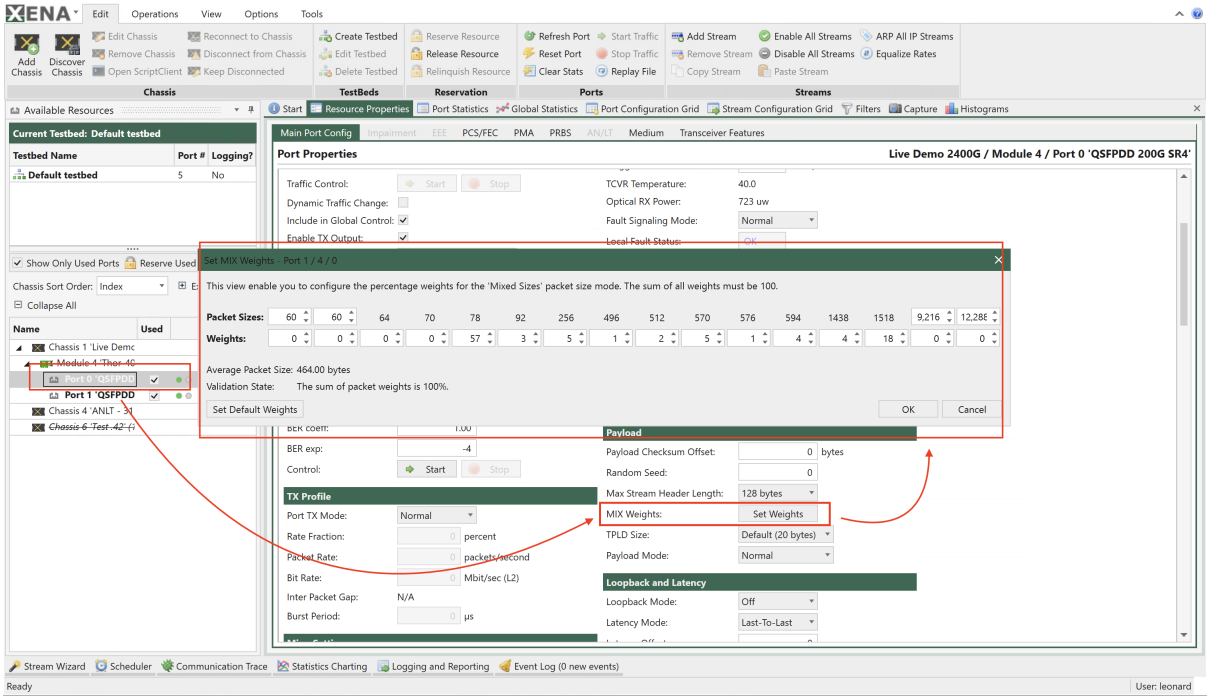


Fig. 5.11: Port MIX weights configuration

Each port has the same default IMIX configuration. If you want to customize the configuration, you should follow the steps:

1. Reserve a test port.
2. Click the port and find *Resource Properties* → *Main Port Config* → *Main Properties* → *Payload* → *MIX Weights*, click *Set Weights*.
3. Configure the desired IMIX.
4. You can save the port's IMIX weight configuration by clicking *Save MIX Weights* on the top bar.
5. You can also load a IMIX weight configuration to the port by clicking *Load MIX Weights*.

To use the configured IMIX to generate packets, you should follow the step:

1. Create a stream on the port you have configured IMIX weights.
2. Click the stream and find *Resource Properties* → *Stream Properties* → *Packet Content* → *Packet Size Type*, and select *Mixed Sizes*.

The stream will generate packet sizes based on the port's IMIX weight configuration.

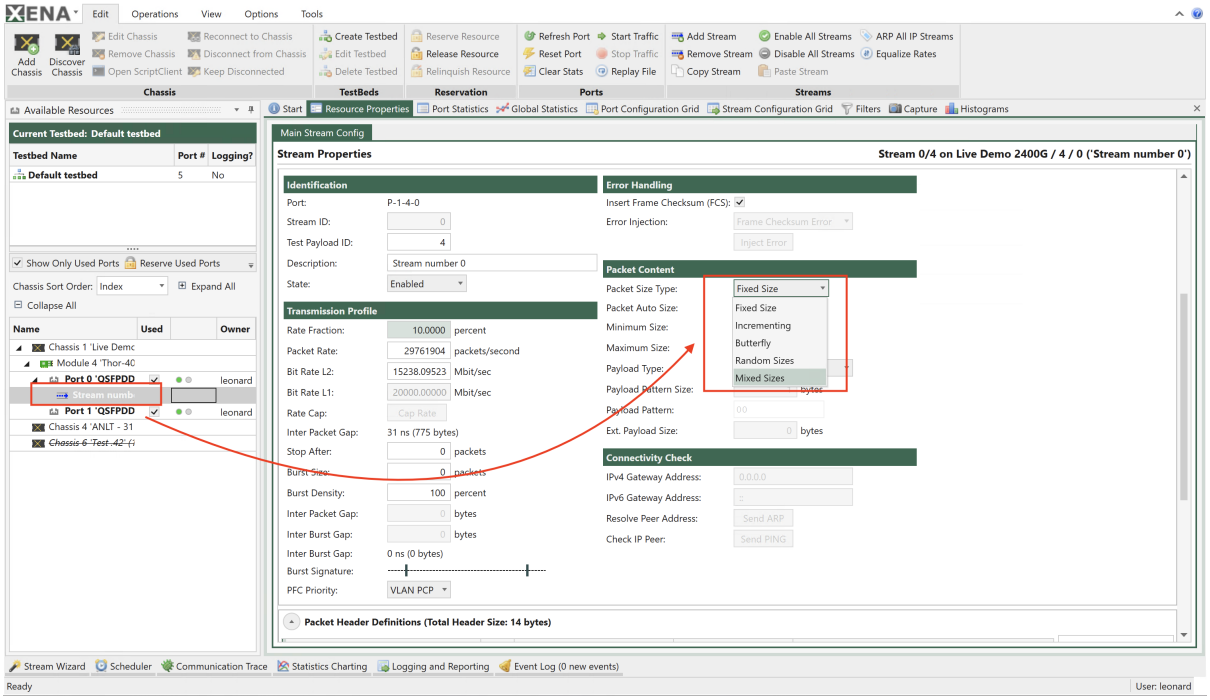


Fig. 5.12: Stream MIX weights selection

Loopback and Latency

Table 5.23: Loopback and Latency

Property	Explanation
Loopback Mode	The port loopback mode. (illustrated in Fig. 5.13) • Off: Traffic flows naturally out of the port • L1 RX-to-TX: Any received packet is bounced back through TX • L2 RX-to-TX: Same as L1 RX-to-TX yet it also swaps SRC MAC address with DST MAC address • L3 RX-to-TX: Same as L1 RX-to-TX yet it also swaps SRC IP address with DST IP address • TX(on)-to-RX: Packet goes out of TX but also internally direct to RX • TX(off)-to-TX: Packet goes directly to RX (No link sync needed) • Port-to-port: Any received packet goes out through the neighbor port Refer to P_LOOPBACK in XOA CLI Documentation for more details.
Latency Mode	The port latency calculation mode. Latency is measured by inserting a time-stamp in each packet when it is transmitted, and relating it to the time when the packet is received. There are four separate modes for calculating the latency: • Last-bit-out to last-bit-in, which measures basic bit-transit time, independent of packet length. • First-bit-out to last-bit-in, which adds the time taken to transmit the packet itself. • Last-bit-out to first-bit-in, which subtracts the time taken to transmit the packet itself. The same latency mode must be configured for the transmitting port and the receiving port; otherwise invalid measurements will occur. • First-bit-out to first-bit-in, which adds the time taken to transmit the packet itself, and subtracts the time taken to transmit the packet itself. The same latency mode must be configured for the transmitting port and the receiving port; otherwise invalid measurements will occur. Refer to P_LATENCYMODE in XOA CLI Documentation for more details.
Latency Offset	The calibrated latency offset value.

Important

As shown in the figure above, the port won't provide TX counters, when the port loopback mode is set to the following values:

- L1 RX-to-TX
- L2 RX-to-TX

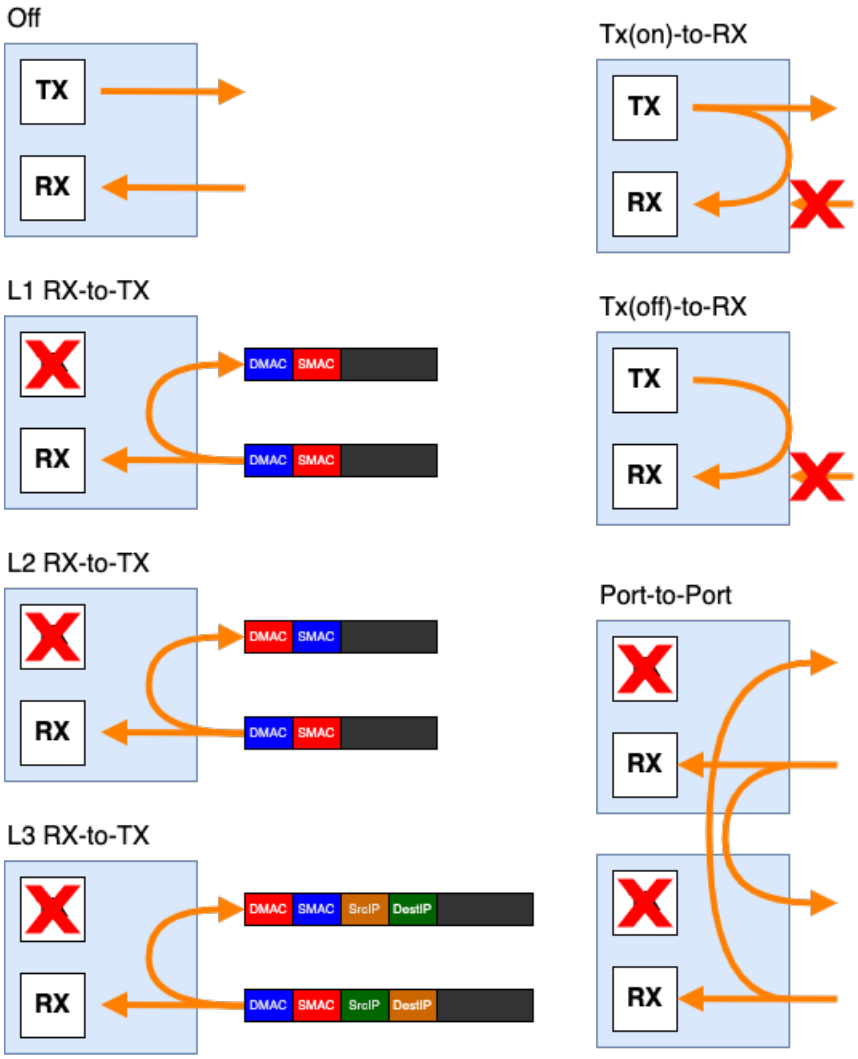


Fig. 5.13: Loopback Mode

- L3 RX-to-TX
- Port-to-Port

IPv4 Properties

Table 5.24: IPV4 Properties

Property	Explanation
IPv4 Address	The IPv4 network address for the port. The address is used as the default source address field in the IP header of generated stream traffic, and the address is also used for support of the ARP and PING protocols.
IPv4 Subnet Mask	The IPv4 subnetwork mask for the port.
IPv4 Gateway	The default IPv4 gateway address for the port.
Reply to ARP Requests	Control whether the port will reply to incoming ARP requests
Reply to PINGv4 Requests	Control whether the port will reply to incoming PING requests
ARP/PINGv4 Address Wildcard	Specifies a mask that makes the port reply to ARP/PING for the masked addresses
DHCPv4 Client	This button will launch the DHCP Wizard for the port. Read more in *DHCPv4 Client Wizard*.

DHCPv4 Client Wizard

If your DUT contains a *DHCP* server (IPv4) you can use this to quickly assign IP addresses to your test port and/or the streams configured on the port. The addresses must be acquired from the DHCP server prior to starting the traffic. and will then be stored as part of the port and stream configuration.

Important

At present only IPv4 is supported. Support for IPv6 may be added in the future.

Wizard Operation

Selecting Targets

When you open the wizard you will be presented with the start page shown below. You can now select to acquire addresses for the port itself, the existing streams on the port, or both.

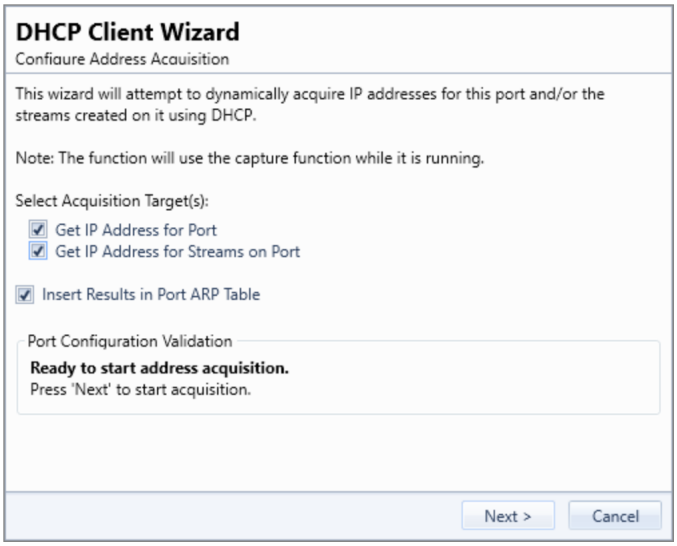


Fig. 5.14: DHCP Client Wizard - Address Acquisition

Note

Please note that if you select to acquire addresses for your streams then they must all contain an IPv4 protocol segment. If the wizard detects that this is not the case you must exit the wizard and correct this manually.

Setting SMAC for Streams

To acquire addresses for streams each stream must be configured with a unique SMAC address in the initial Ethernet protocol segment.

On initial launch the wizard will determine if the SMAC addresses for the streams are unique within the scope of the port. If not it will offer to assist you in assigning unique addresses as indicated in the Fig. 5.15 below.

Note

Please note that the wizard will not ensure that the stream SMAC addresses are globally unique. It will only check if the SMAC addresses are unique within the scope of the port on which they reside.

Acquiring Addresses

Once all requirements are satisfied the wizard will start to acquire addresses from the DHCP server. You can follow the progress in the wizard as shown in Fig. 5.16 below.

The *Counters* field at the top show the number of DHCP packet sent and received. The grid below that show a summary of the communication with the DHCP server.

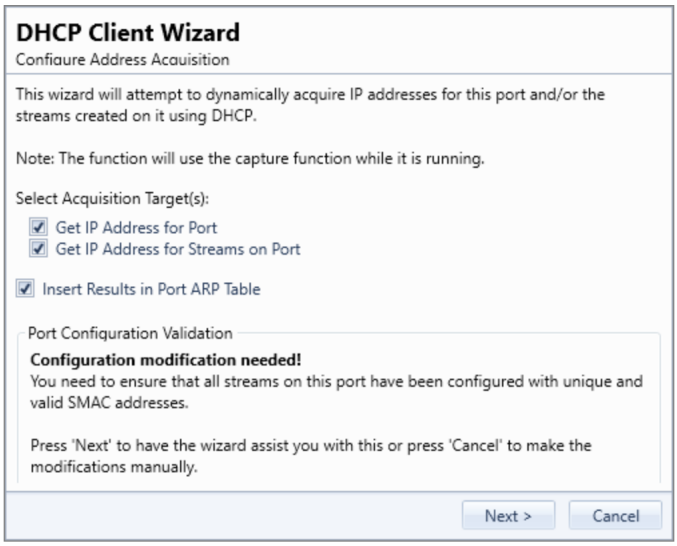


Fig. 5.15: DHCP Client Wizard - Setting SMAC

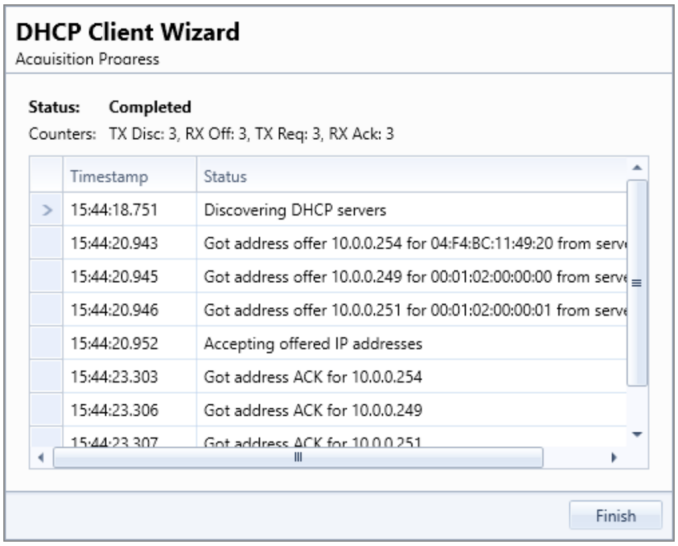


Fig. 5.16: DHCP Client Wizard - Acquiring Addresses

IPv4 Multicast Properties

Table 5.25: IPv4 Multicast Properties

Property	Explanation
MC Addresses	Specifies a list of multicast addresses to send IGMPv2IGMPv3 requests to.
Send Join Request	When this button is clicked a single IGMPv2 Join request is sent to the specified multicast address.
VLAN	Check this box to add a VLAN tag header to sent IGMPv2IGMPv3 requests
Tag	Value of VLAN tag
Priority	Priority bits in VLAN tag header
DEI	Drop Eligible Indicator in VLAN tag header
Send Requests	For Protocol Version IGMPv2 a single request is sent when the related button is clicked.
Send Join Request	Send a Join Request
Send Leave to AllRouters	Send Leave to IP Address (224.0.0.2)
Send Leave Request	Send a Leave Request
Send General Query Request	Send a General Query Request
Send Group Query Request	For Protocol Version IGMPv3 a single request is sent when the related button is clicked: • Send Change to Exclude Request • Send Change to Include Request • Send Include Request • Send Exclude Request
Repeat Multicast Join	Control whether the Join command should periodically be re-transmitted
Multicast Join Period	The Join retransmit period in seconds
Protocol Version	Set Multicast Protocol version to IGMPv2 or IGMPv3
Group Record Bundling	Configure if a single membership report bundles multiple multicast group records to decrease the number of packets sent when using IGMPv3. (Only available when Protocol Version is `IGMPv3`)

IPv6 Properties

Table 5.26: IPv6 Properties

Property	Explanation
IPv6 Address	The IPv6 network address for the port. The address is used as the default source address field in the IP header of generated stream traffic, and the address is also used for support of the NDP and PING protocols.
IPv6 Prefix	The IPv6 subnetwork prefix for the port,
IPv6 Gateway	The default IPv6 gateway address for the port.
Reply to NDP Requests	Control whether the port will reply to incoming NDP requests. **Note** When you enable this *Reply to NDP Requests* option, the port will reply to Neighbor Solicitation **sent to both unicast and multicast addresses**. **See also** **RFC 4861 Section 7.2**
Reply to PINGv6 Requests	Control whether the port will reply to incoming PING requests
NDP/PINGv6 Address Wildcard	Specifies a prefix that makes the port reply to NDP/PING for the masked addresses

ARP/NDP Address Tables

Each Xena test port contains two address tables, one for IPv4 (ARP) and one for IPv6 (NDP). Each address table can contain a number of entries. Each entry defines a set of criteria for handling incoming ARP/NDP requests. Each table can be accessed by the *Edit ARP Table* and *Edit NDP Table* buttons in this section. Pressing each button will launch a dialog as shown below

New entries can be added by pressing the *Click here to add new item* bar. Existing entries can be edited by selecting the various fields or deleted by pressing the button with the red stop sign to the right.

Changes to the table are not sent to the test chassis until the *OK* button is pressed.

As an alternative you can press the *Auto-Assign Tables Generate* button. If you do that the content of the ARP and NDP address tables are automatically generated based on the defined streams for the port. All existing entries in the tables will be removed.

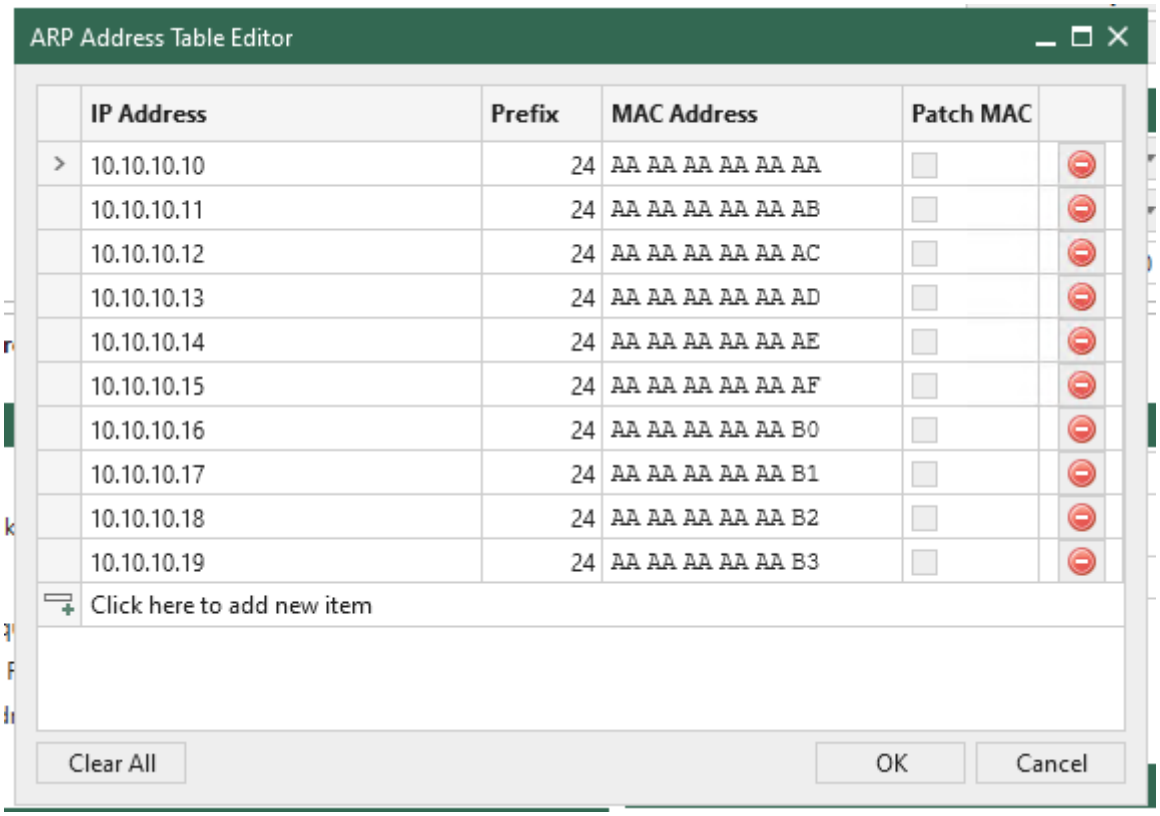


Fig. 5.17: ARP Table

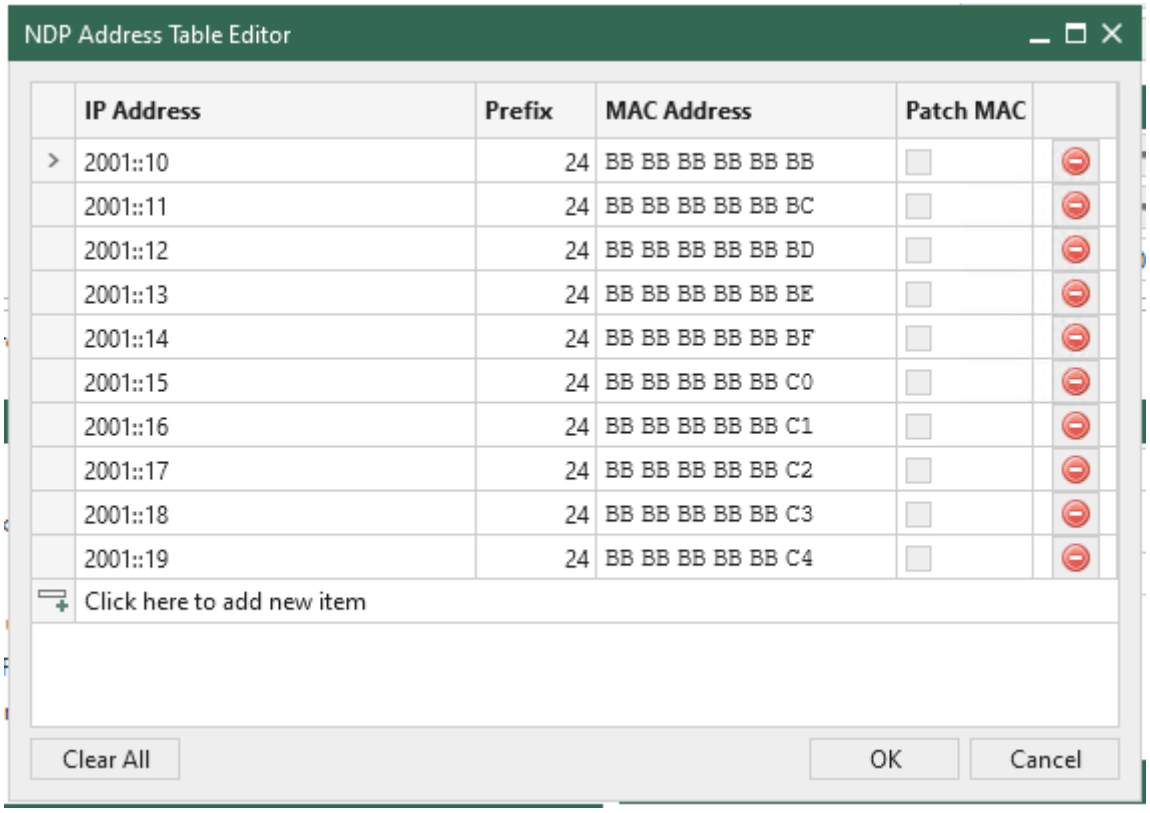


Fig. 5.18: NDP Table

General Handling of Incoming Requests

Any incoming ARP or NDP request is handled in the following prioritized order:

1. If the address table for the IP version used by the request contains 1 or more entries the table is searched for a match for the Target IP Address in the request. If a match is found a reply is formatted according to the matched entry definition and sent. If no matches are found in the table the request is ignored.
2. If the address table for the IP version used by the request is empty the request is handled by the legacy method, i.e. matched to the defined port IP address optionally masked by the wildcard value.

Address Table Matching

Incoming ARP/NDP requests are matched to the address table by comparing the *Target IP Address* in the request with the IP Address value for each entry in the table, masked by the *Prefix* value. If a match is found the search is stopped and the matched entry used for the reply.

If two or more entries would match the Target IP Address then only the first matching entry is used.

The *Prefix* value can be used to have each entry match multiple IP addresses. Example: An IPv4 entry with IP Address = 10.0.0.1 and Prefix = 28 will match any address in the range 10.0.0.0 to 10.0.0.15. The default value for the Prefix is a full host mask, which means that only the specified IP address will match.

Formatting the Reply

If a match is found the ARP/NDP reply will be formatted according to the following rules:

1. If the MAC Address field in the match entry is all-zeros (which is the default value) the SMAC address in the reply is set to the port MAC address. Otherwise the defined MAC Address value is used.
2. If the Patch MAC option is checked the least significant bytes in the SMAC address is patched with the least significant bytes in the Target IP Address. Which bytes are patched is controlled by the Prefix value. This feature is relevant when using a Prefix value to reply to a range of IP addresses to ensure that each "emulated" port is returning a unique MAC address.

Port Resource Commands

This section describes the resource-specific commands available for ports in *Edit* → *Ports*.

Table 5.27: Port Resource Commands

Property	Explanation	Must Reserve?
Refresh Port	Reload the configuration for the port and all child resources (streams, modifiers, etc) from the test chassis.	No
Reset Port	Reset the port configuration to default settings. Note that this removes all dynamic resources such as streams, modifiers, etc.	Yes
Clear Stats	Clear all TX and RX statistics counters on the port.	Yes
Start Traffic	Start traffic on the port. The port must contain at least one enabled stream.	Yes
Stop Traffic	Stop traffic on port.	Yes
Replay File	Load a PCAP file and replay it on the port. This function is described in details on *Replay PCAP File*.	Yes

EEE

Energy Efficient Ethernet (EEE) is a set of technologies and standards designed to reduce the power consumption of Ethernet networks and network devices, such as switches and network interface cards (NICs). EEE aims to minimize energy usage during periods of low network activity by transitioning network devices into low-power or sleep modes.

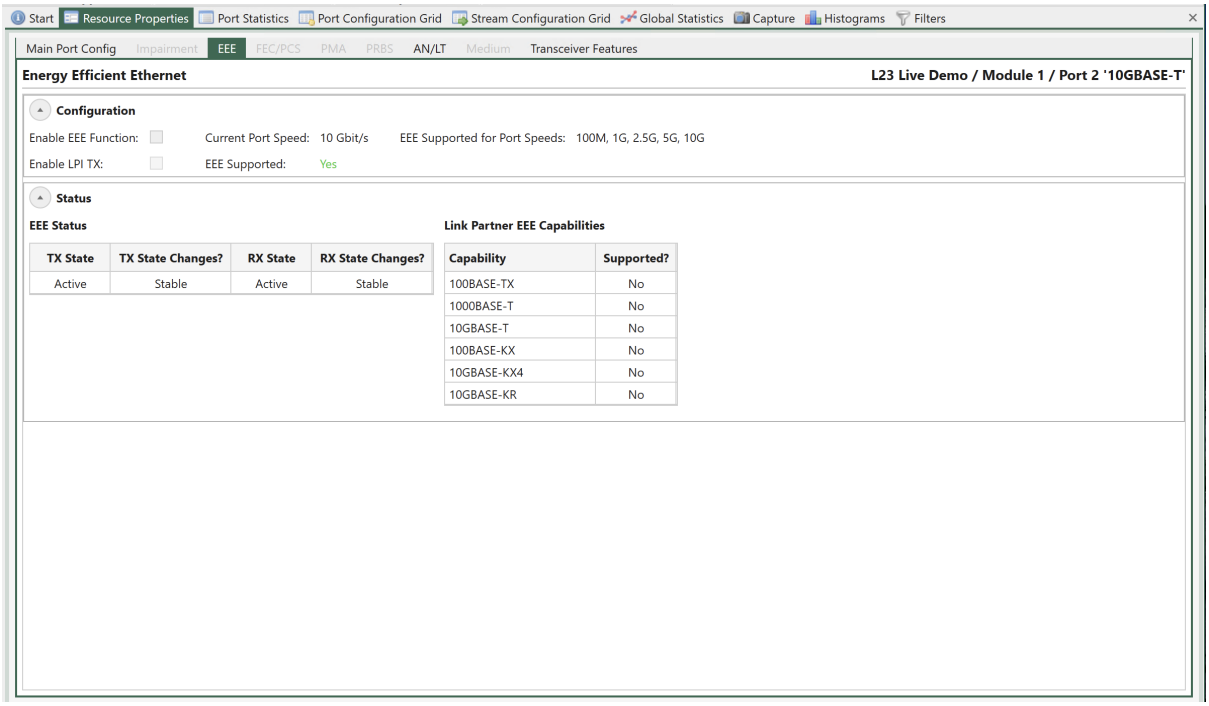


Fig. 5.19: Port Properties - EEE

Table 5.28: EEE Configuration

Property	Explanation
Enable EEE function	Enable/disable the EEE function on the port
Enable LPI TX	Enables/disables the transmission of Low Power Idles (LPIs) on the port. When enabled, the transmit side of the port will automatically enter low-power mode (and leave) low-power mode in periods of low or no traffic. LPIs will only be transmitted if the Link Partner (receiving port) has advertised EEE capability for the selected port speed during EEE auto-negotiation.

PCS/FEC

PCS (Physical Coding Sublayer) and FEC (Forward Error Correction) are two essential components of the Ethernet standard that work together to ensure reliable and error-free data transmission over Ethernet networks. These components are primarily associated with the Ethernet physical layer, which is Layer 1 of the OSI model.

FEC Config

PCS Variant

Important

PCS Variant configuration is only available on Z800 Freya module.

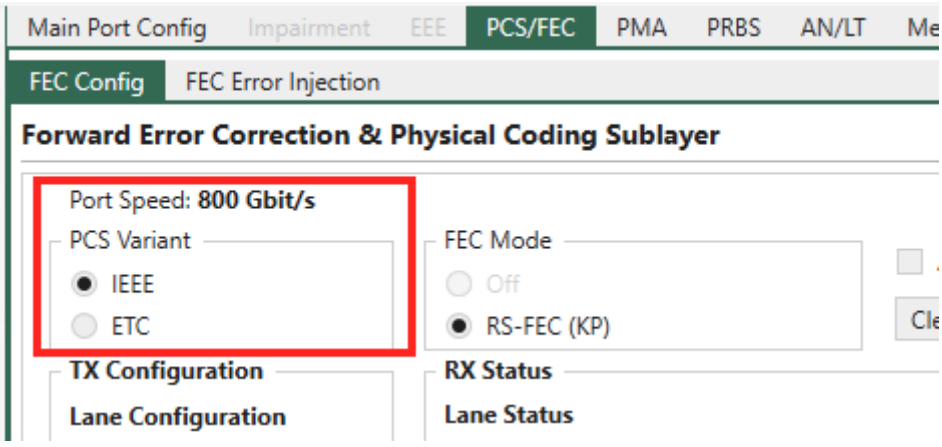


Fig. 5.20: PCS Variant selection

Both IEEE and ETC (Ethernet Technology Consortium) have specifications for 100 Gb/s serdes lane rate based 800GbE:

- IEEE: 800GBASE-CR8/KR8
- ETC: 800G-ETC-CR8/KR8

When testing with 800G on Z800 Freya using XenaManager, you can either manually configure the 800G variant by selecting IEEE or ETC in *Resource Properties* → *PSC/FEC* → *PCS Variant*, or let auto-negotiation decide.

FEC Mode

Important

For PAM4 ports, FEC is mandatory.

Depending on the serdes lane speed and port speed, you will see different FEC modes.

- For 112G serdes PAM4, 800G/400G/200G port speed, RS-FEC KP is mandatory.

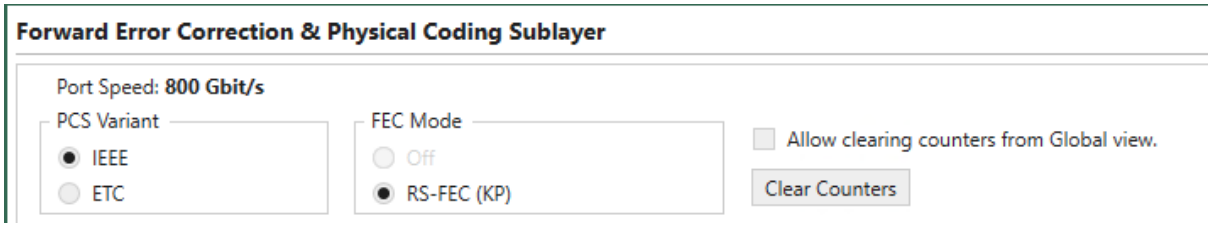


Fig. 5.21: 800G-200G with 112G serdes PAM4. RS-FEC KP is mandatory.

- For 112G serdes PAM4, 100G port speed, RS-FEC KP and RS-FEC-Int.

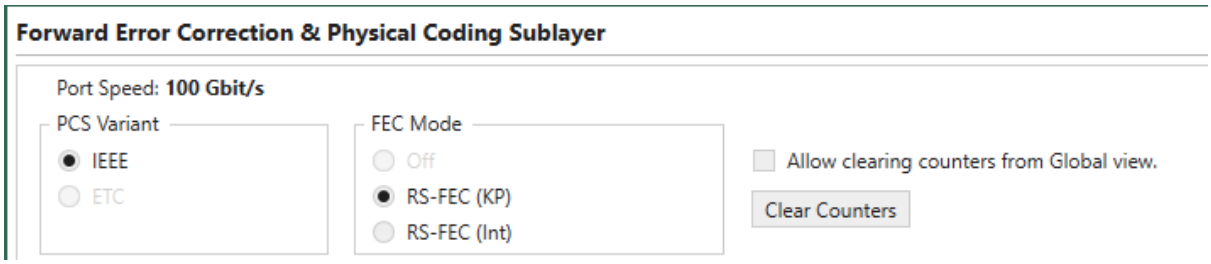


Fig. 5.22: 100G with 112G serdes PAM4. RS-FEC KP and RS-FEC-Int.

- For 56G serdes PAM4, RS-FEC KP is mandatory.

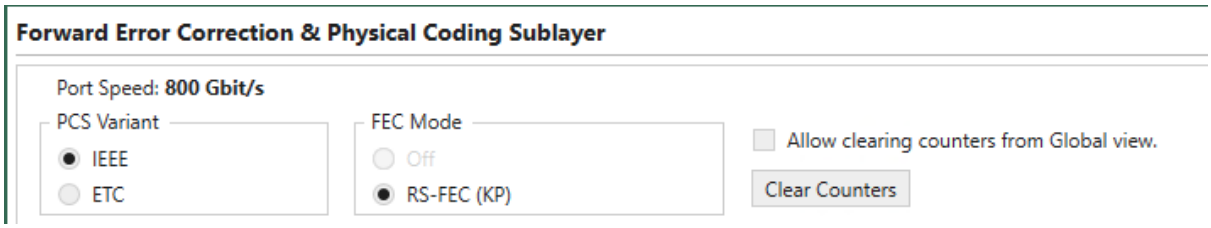


Fig. 5.23: 56G serdes PAM4. RS-FEC KP is mandatory.

- For 25G serdes NRZ, RS-FEC KR and Off.

Clear Counters will clear the counters on the page and by checking *Allow clearing counters from Global view* you can use the clear counters facilities in the *Global Statistics* pages to clear the counters in this page.

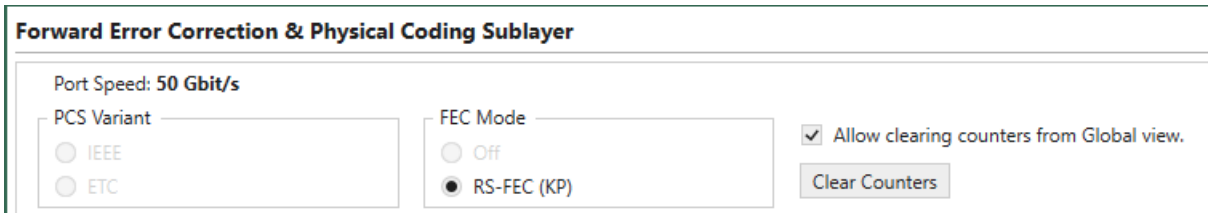


Fig. 5.24: 25G serdes NRZ. RS-FEC KR and Off.

What is shown on the rest of this page depends on the configuration of the selected port.

Lane Configuration & Lane Status

PAM4 Ports (56G and 112G serdes Lane)

The images below show what you can see on a PAM4 port.

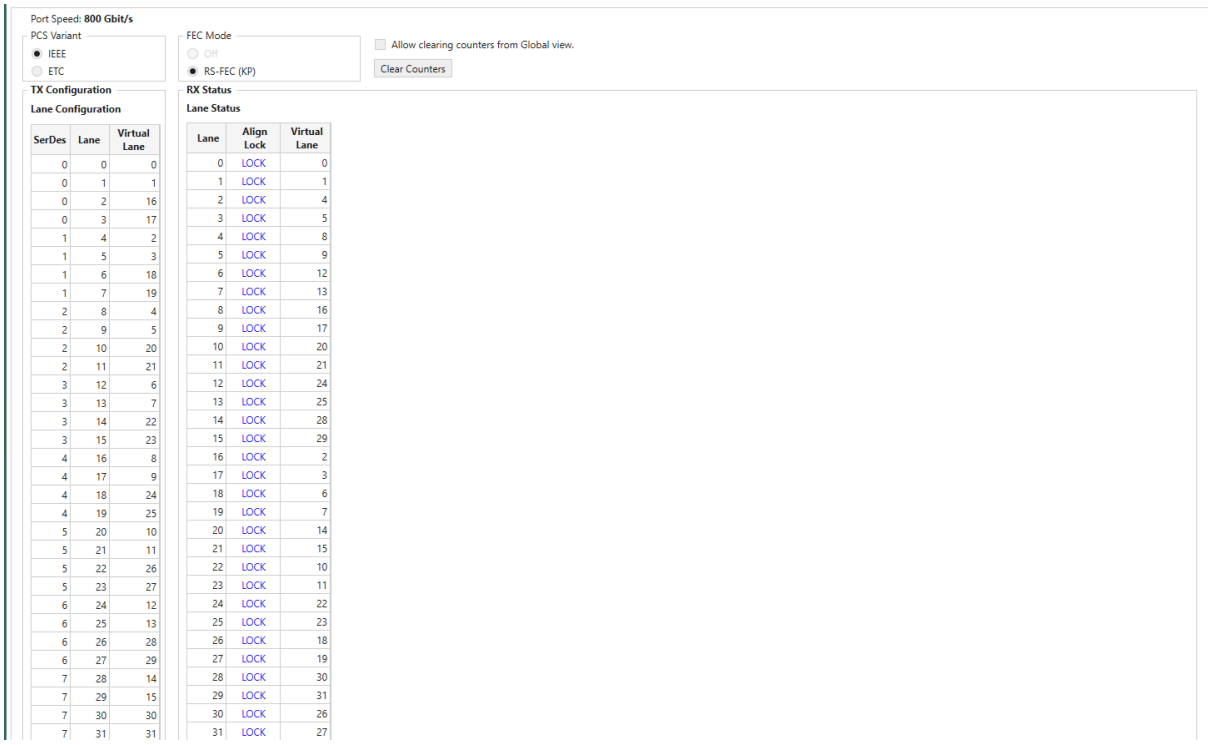


Fig. 5.25: 112G serdes PAM4, 800G port speed on Z800 Freya module. (Only partial screen)

In the *TX Configuration* section you can for each physical Lane on the port see the SerDes, and Lane. You can change the *Virtual Lane* used for the physical lane and set Skew Bits for each lane.

In the *RX Status* section you can for each physical Lane see if it is in *Align Lock* (Alignment lock), the *Virtual Lane* used for the physical lane and set Skew Bits for each lane. In addition, you for each lane can see number of *Corrected Bit Errors* and the *estimated Pre-FEC BER*.

The lower part of the page contains FEC statistics for the port:

- Total corrected FEC Symbols

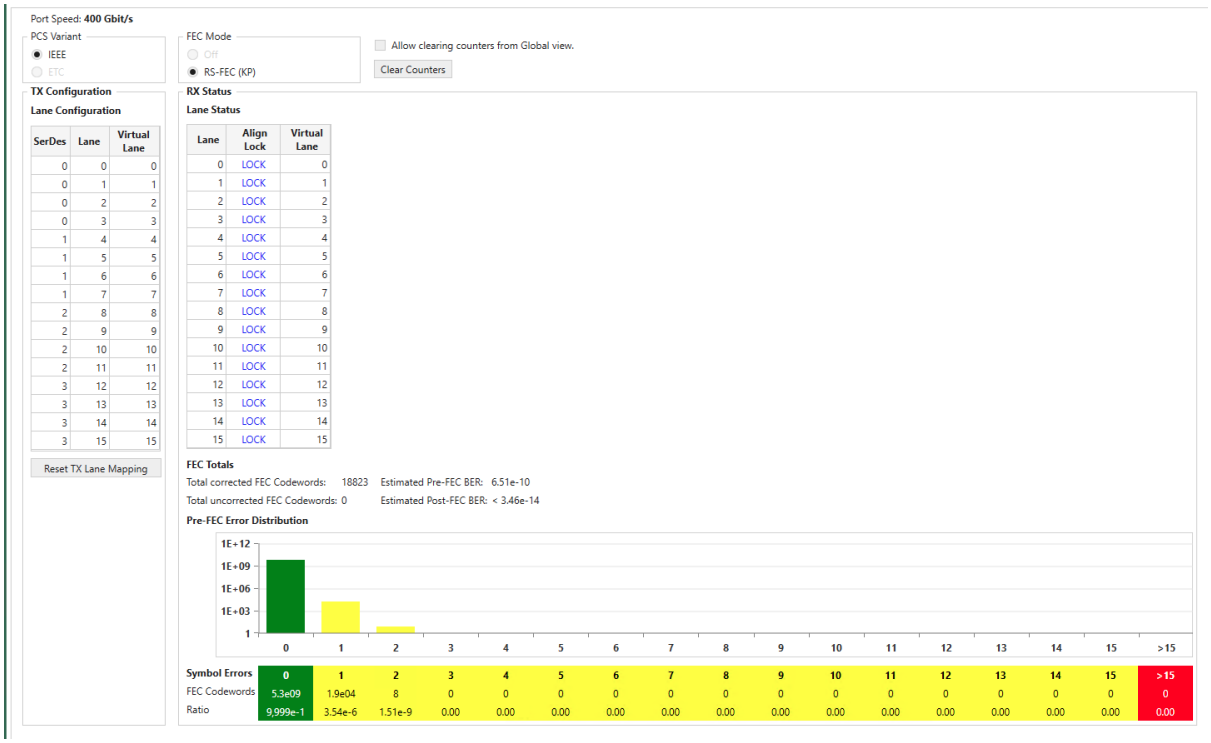


Fig. 5.26: 112G serdes PAM4, 400G port speed on Z800 Freya module.

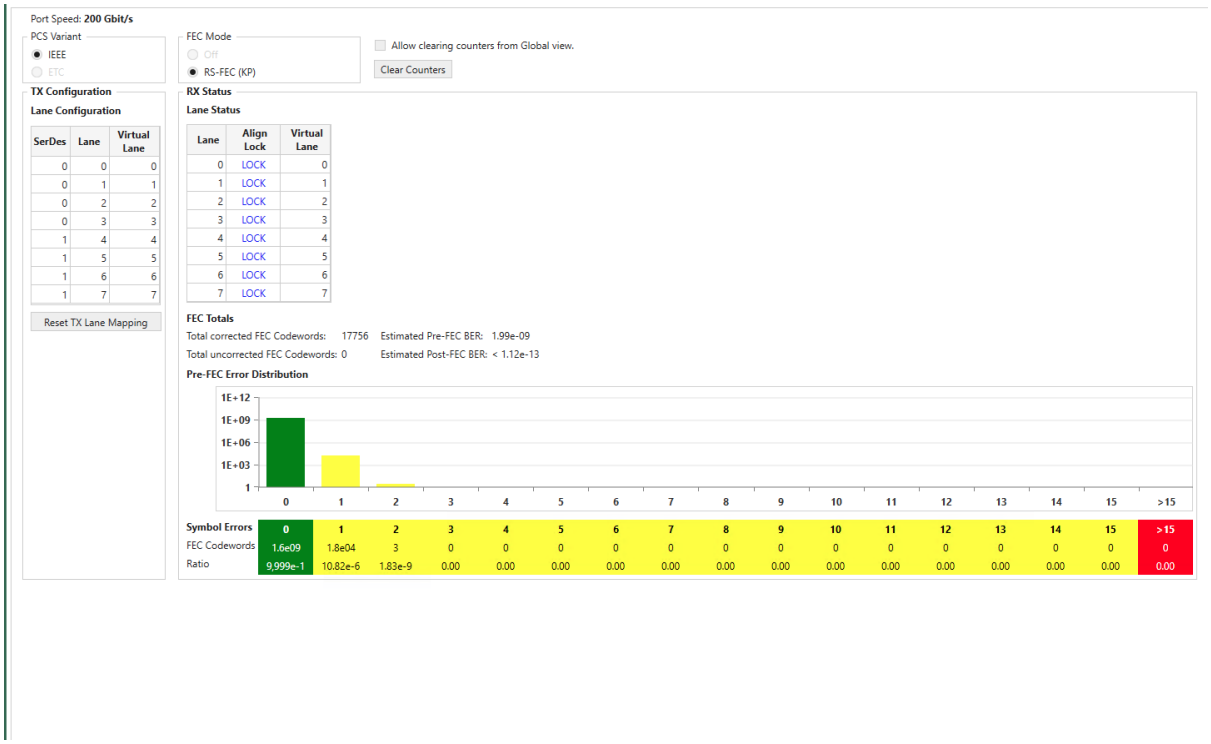


Fig. 5.27: 112G serdes PAM4, 200G port speed on Z800 Freya module.

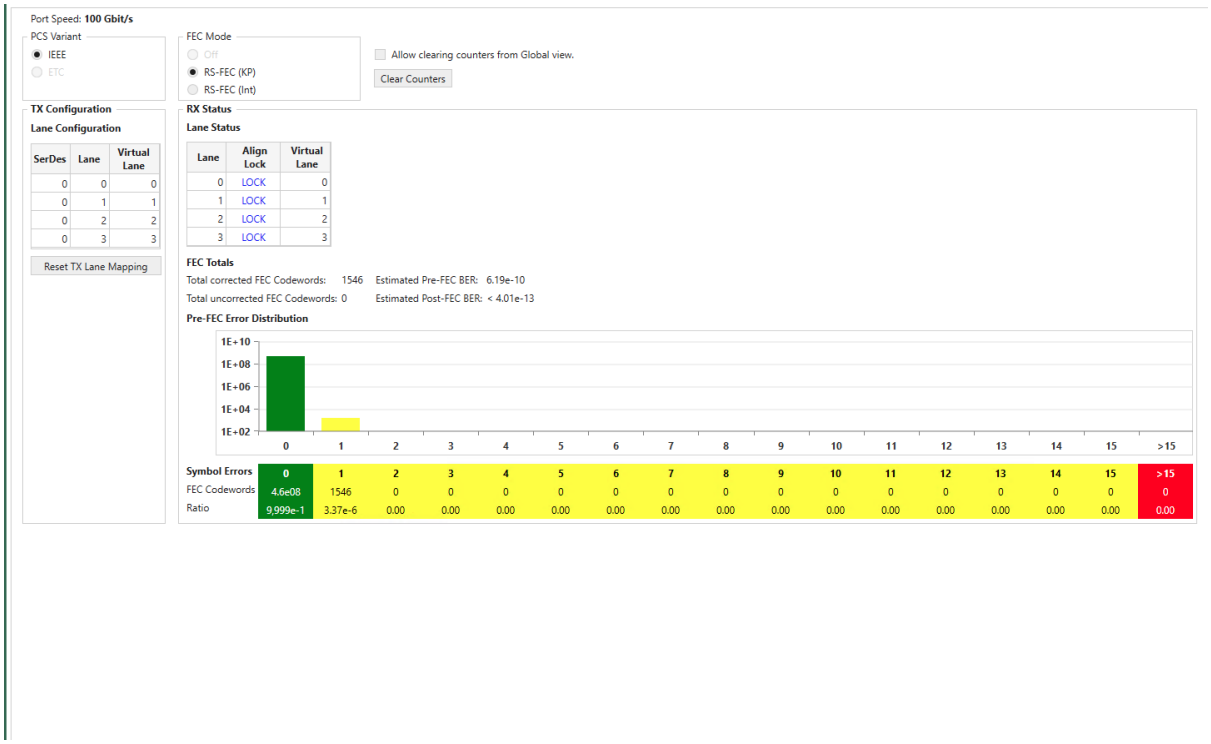


Fig. 5.28: 112G serdes PAM4, 100G port speed on Z800 Freya module.

- Total uncorrected FEC blocks
- Estimated Pre-FEC BER
- Estimated Post-FEC BER

You will also find the Pre-FEC Error Distribution graph. Here you can see number of received FEC blocks with 0, 1, 2.. up to 15 symbol errors. This is what the RS-FEC used for PAM4 signals will correct. You can also see number of received FEC blocks with more than 15 symbol errors. This is more than the RS-FEC used for PAM4 signals can correct so this will cause uncorrected FEC blocks to be counted and most likely also cause errors at higher layers in the received signal.

> **Note**
>
> Z800 Freya module supports **RS-FEC Int**, which is specified in IEEE 802.3ck CL 161. When Z800 Freya port is configured to **100GBASE (with 112 serdes speed)**, the option of **RS-FEC Int** will be enabled for selection in *Resource Properties* → *PCS/FEC*

NRZ Ports (25G serdes Lane) With FEC

The image below shows what you can see on a NRZ port with FEC, in this case a 100G port.

Most is similar to what you can control and see for PAM4 ports. Please also observe that the RS-FEC used for NRZ signals will correct up to 7 symbol errors. The Pre-FEC Error Distribution graph is adjusted accordingly.

Some ports also support `Firecode FEC` when running at 10G and 25G NRZ.

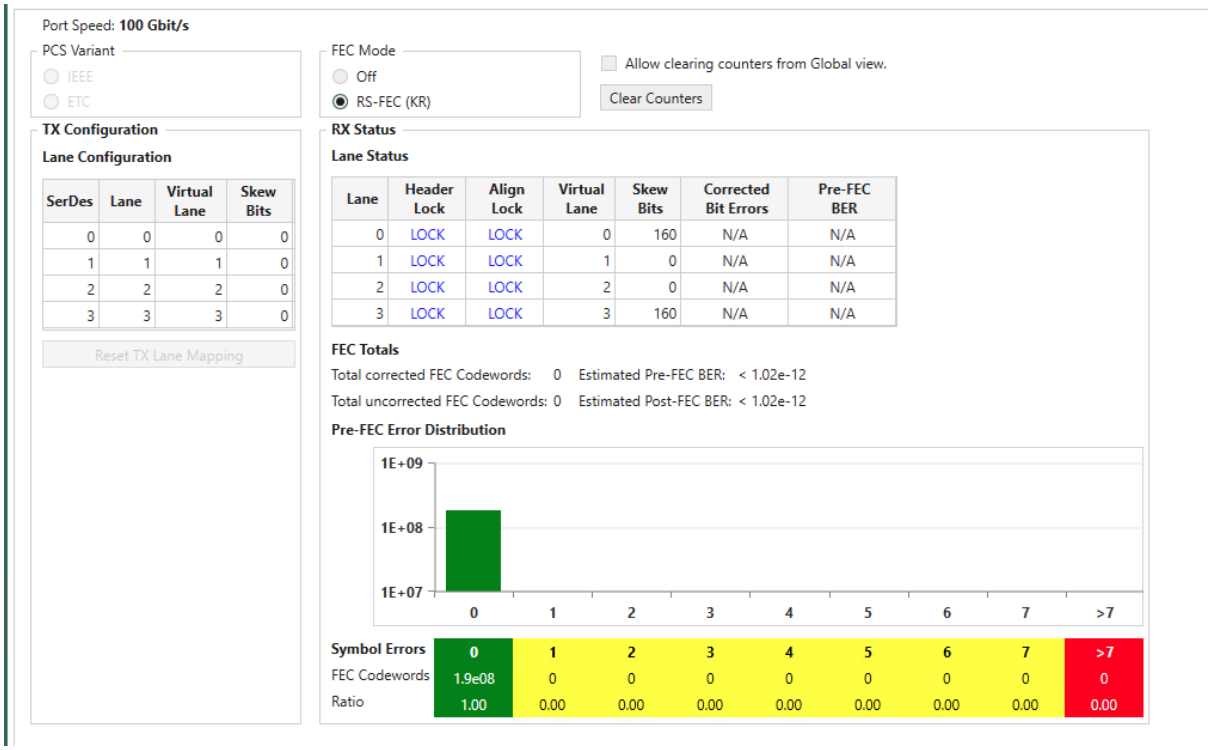


Fig. 5.29: Port Properties - PCS/FEC - FEC Config (NRZ Ports w/ FEC)

> **Note**
>
> Port speeds of 10G/25G on Z100 Loki and Z400 Thor support **Firecode FEC**.

NRZ Ports (25G serdes Lane) No FEC

The image below shows what you can see on a NRZ port without FEC, in this case a 100G port.

Most is similar to what you can control and see for PAM4 ports in the *Transmit Configuration* and *Receive Status* tables. Please observe however that in the *Receive Status* table you can see additional counters for *Header Errors*, *Alignment Errors*. The *Corrected Bit Errors* and the *Pre-FEC BER* counters are not updated for ports without RS-FEC.

> **Note**
>
> Port speeds of 10G/25G on Z100 Loki and Z400 Thor support **Firecode FEC**.

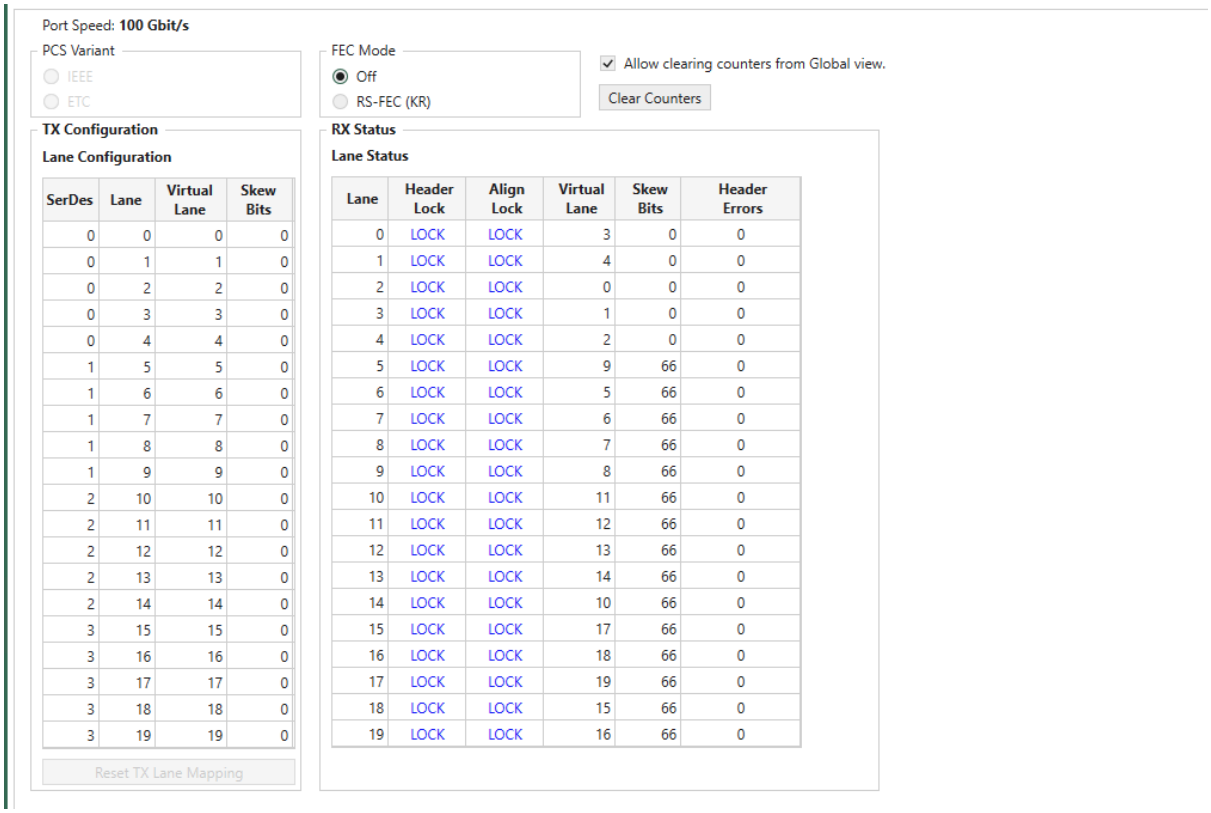


Fig. 5.30: Port Properties - PCS/FEC - FEC Config (NRZ Ports w/o FEC)

FEC Error Injection

FEC Error Injection feature to Z800 Freya for RS-FEC KP (544,514) and RS-FEC-Int (544,514).

Codeword, Symbol and Bit Errors

- FEC is a technique used to detect and correct a certain number of errors in a bit stream by appending redundant bits and error-checking code to the message block before transmission.
- FEC uses `n` symbol codewords (CW) consisting of a data block that is `k` symbols long and a parity block (the code and redundant bits) that is`` n-k`` symbols long. We denote a particular FEC by the ordered pair `(n,k)`.

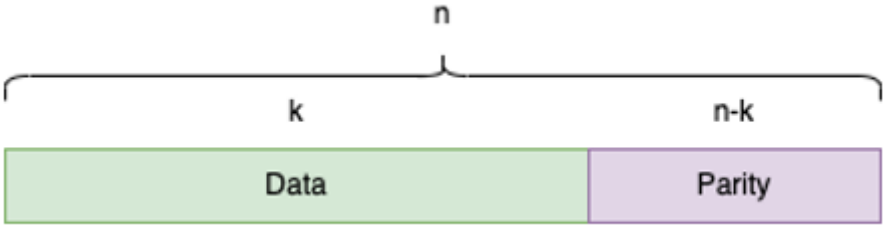


Fig. 5.31: RS-FEC notation

For RS-FEC KP (544,514) and RS-FEC-Int (544,514), 10 bits form a FEC symbol and 544

symbols a FEC codeword, where 514 of them are data symbols and 30 are parity symbols, as depicted below.

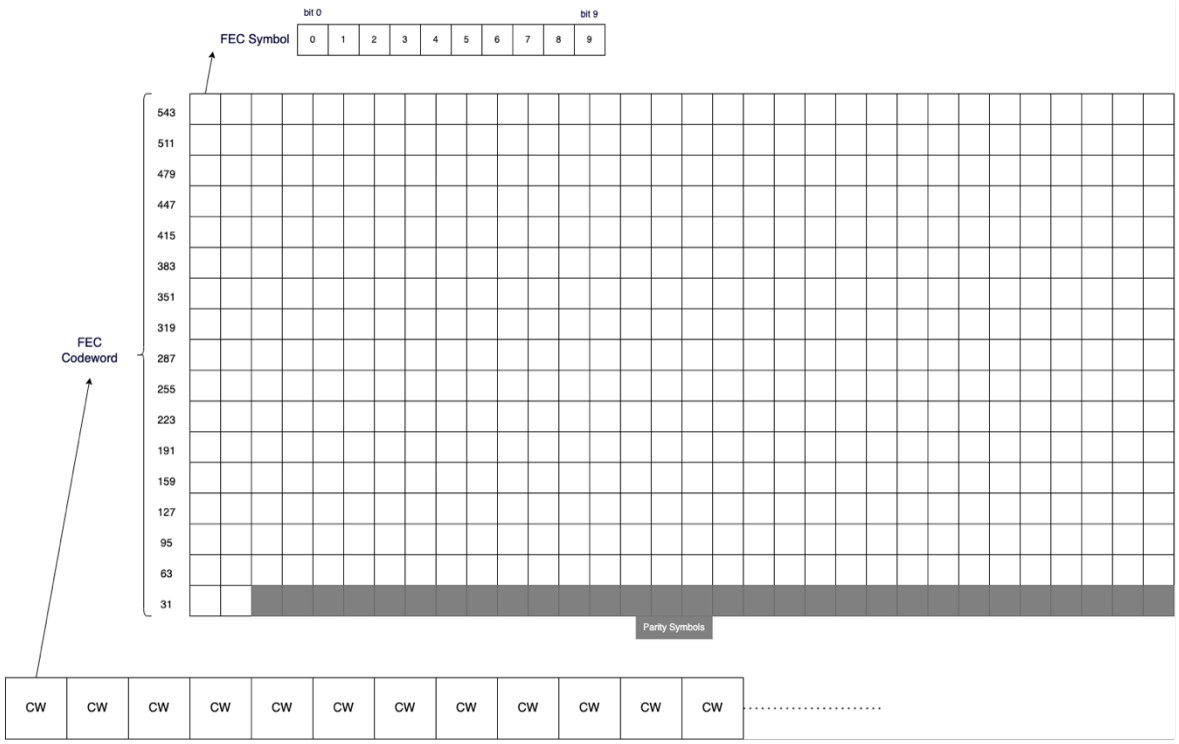


Fig. 5.32: FEC Codeword, Symbol, and Bit

Codeword & Symbol Error Configuration

On the codeword level, you configure the number of codewords of an injection cycle. The *Codewords Per Cycle* must be an even integer for 112G SerDes, e.g. 2, 4, 6, 8, etc. Inside a cycle, you can configure the number of Consecutive Errored Codewords, in which at least one errored symbol is injected. As depicted above, the errored codewords are placed before the error-free.

There are two pre-defined profiles you can load:

- Max Consecutive Uncorrectable w/o Link Loss
- Min Consecutive Uncorrectable with Link Loss

After loading the profile, you can still customize it to meet your test requirements.

Important

You need to click the *Apply* button to commit the errored symbol index list after loading the profile.

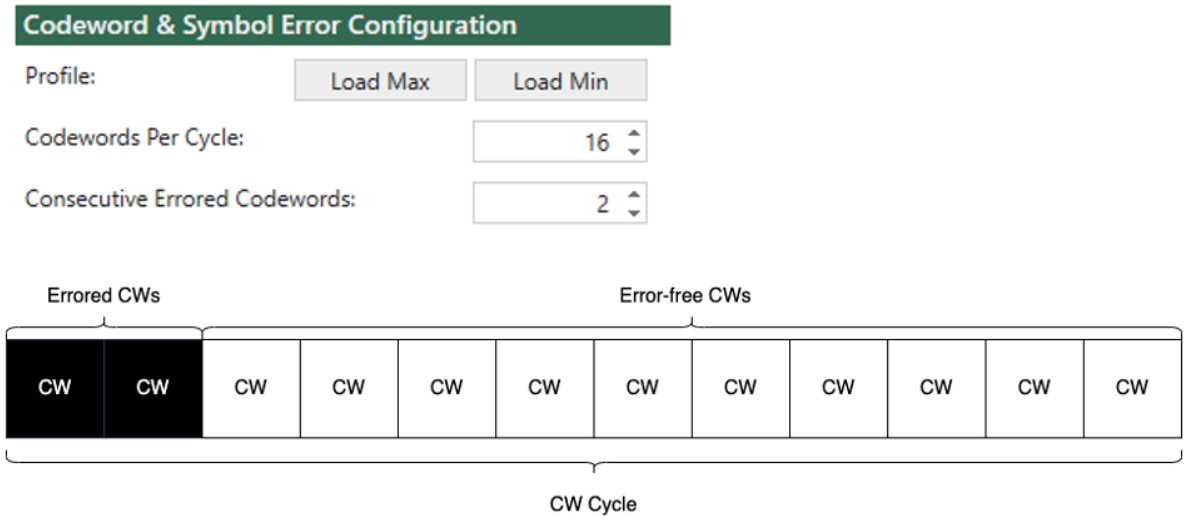


Fig. 5.33: Injection cycle illustration

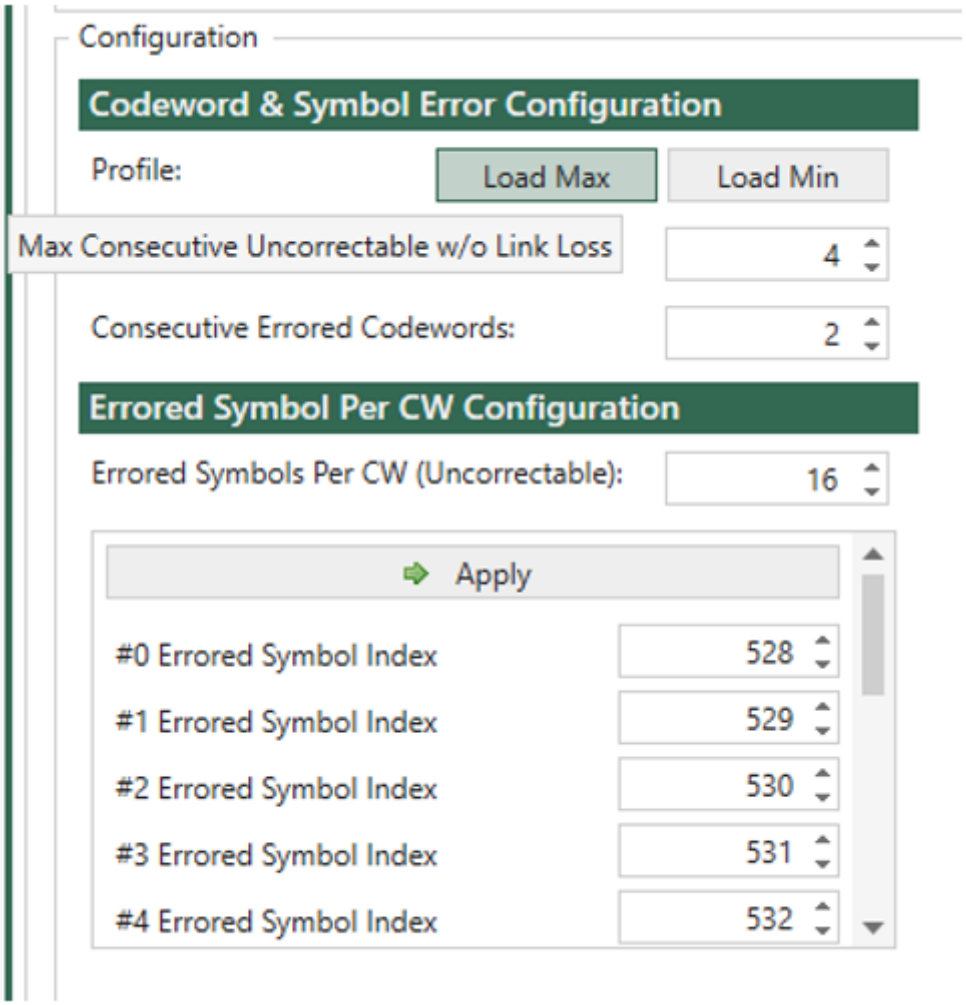


Fig. 5.34: Load Max Consecutive Uncorrectable w/o Link Loss

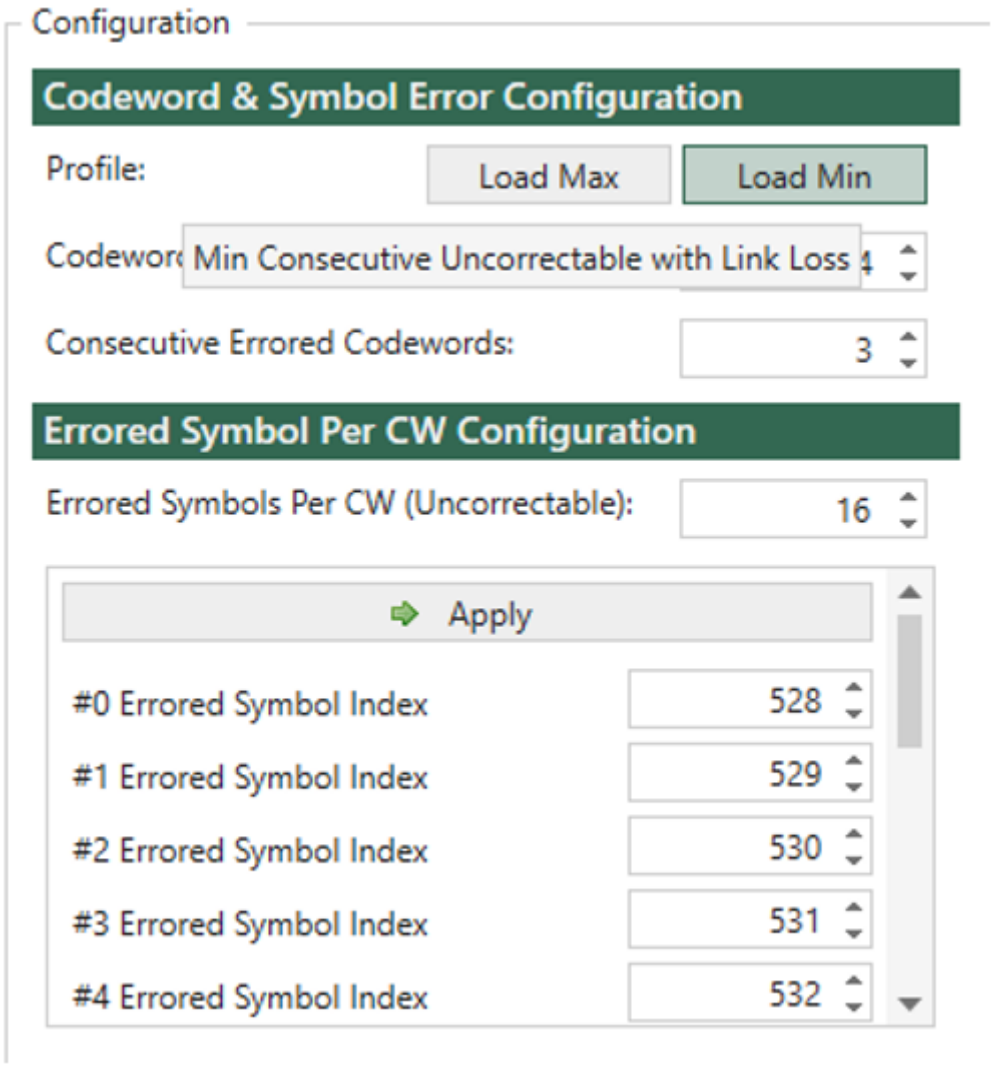


Fig. 5.35: Load Min Consecutive Uncorrectable with Link Loss

Errored Symbol Per CW Configuration

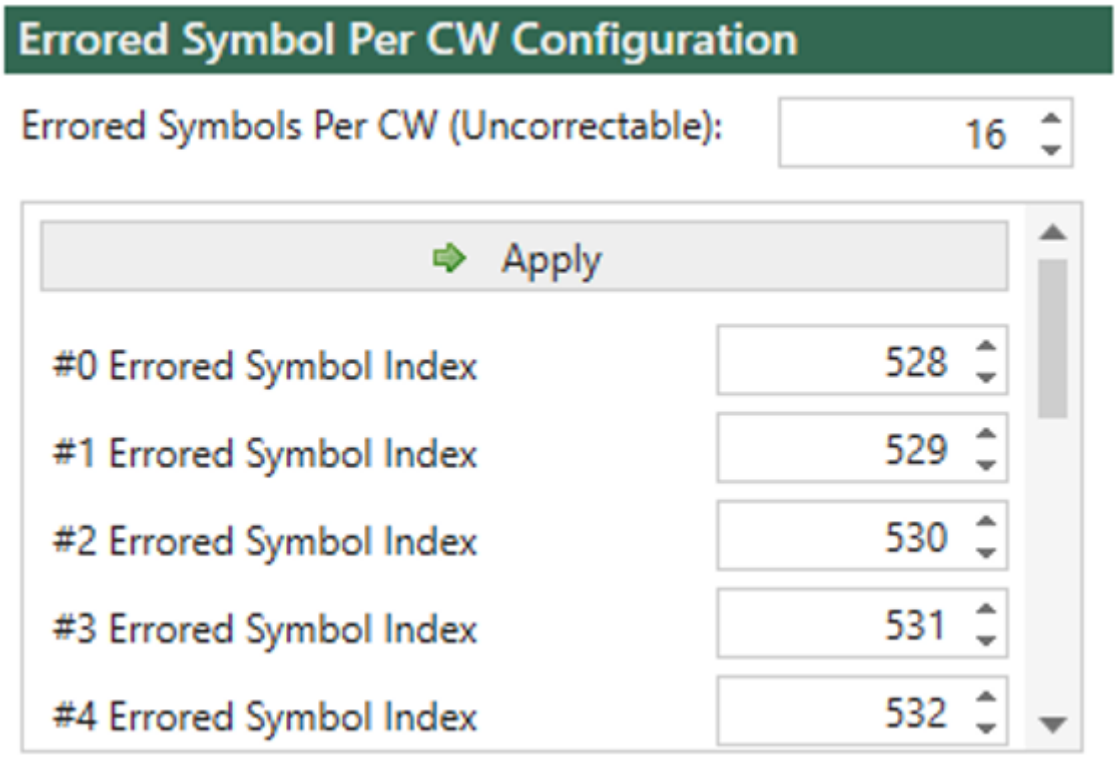


Fig. 5.36: Errored Symbols Per CW and Errored Symbol List

On the symbol level, you configure the quantity of errored symbols inside an errored CW in *Errored Symbols Per CW*, and the **position of each errored symbol** in the list below it.

For RS-FEC KP and RS-FEC-Int, a CW with more than 15 errored symbols is uncorrectable, which you can see as you configure.

Every time the errored symbol index list is changed, you need to click *Apply* to commit the change to the chassis. If you have many indices to change, you can use the tab key on the keyboard to go from one index to the next, or shift + tab to go back.

Bit Error Mask Configuration

On the bit level, you configure the quantity of errored bits inside an errored symbol, and their position. Use the bit mask pattern checkbox to select which bits are errored inside an errored symbol. The most significant bit (bit 9) is placed on the left, and the least significant bit (bit 0) on the right.

There are three modes you can choose:

- Static: The bit error pattern stays the same for all errored symbols.
- Rotate: The bit error pattern shifts one bit to the most significant bit per errored symbol.
- Increment: When this mode is selected, the bit error pattern is ignored. Instead, the bit error pattern initiates from 000000001, 000000010, 000000011, continuing up to 111111111, and repeating the sequence as 000000001…

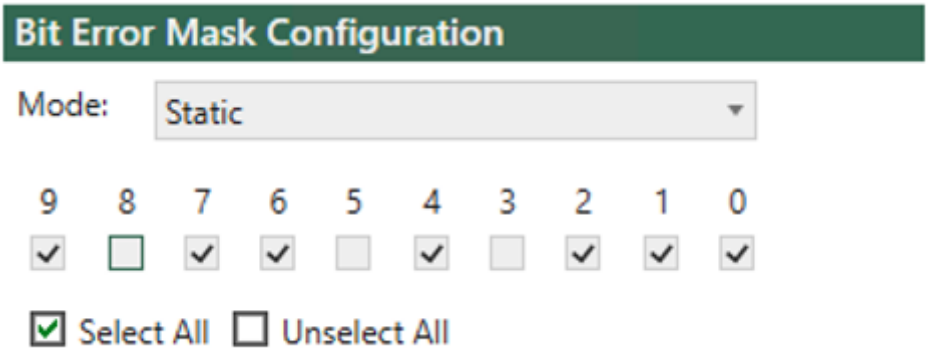


Fig. 5.37: Bit Error Mask Configuration - Mask Configuration

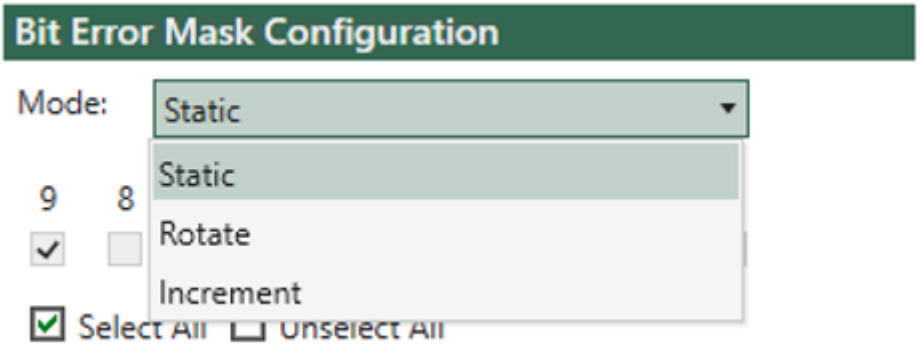


Fig. 5.38: Bit Error Mask Configuration - Mode

Loop Mode



Fig. 5.39: Loop Mode

There are three loop modes for you to control how many cycles to inject:

- Single: Inject a single cycle.
- Continuous: continuously inject cycles until explicitly stop.
- Repeat: Inject a specific number of cycles (max 65,535)

FEC Engines

Z800 Freya ports in different speeds have different numbers of FEC engines. all FEC engines of a port are enabled. FEC engine selection is not supported yet.

serdes Speed	Port Speed	FEC Engines
112G	800G	4
	400G	2
	200G	2
	100G (RS-FEC KP)	1
	100G (RS-FEC-Int)	1
56G	400G	4
	200G	2
	100G	1

TX Error Injection Statistics

In the TX statistics, you can find the following counters:

- Total transmitted CWs
  - Total correctable CWs (errored but with <=15 errored symbols)
  - Total uncorrectable CWs (errored and with >15 errored symbols)
  - Total error-free CWs

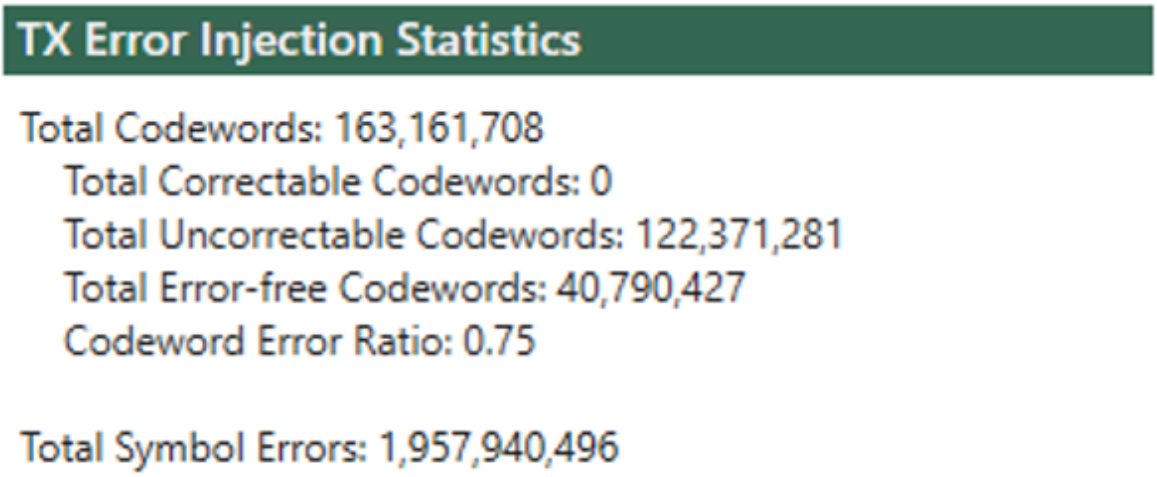


Fig. 5.40: TX Error Injection Statistics

  - CW error ratio (1-total-error-free/total-cws)
- Total transmitted errored symbols

> **Attention**
>
> Each FEC engine injects errors independently according to the configurations. This means the total counters is what you configure on the UI **multiplied by** the number of FEC engines.

PMA

Physical Medium Attachment (PMA) refers to a sublayer within the physical layer (Layer 1) of the OSI model. Its primary role is to manage the interface between the physical medium (the actual transmission medium, whether it's copper, fiber, or wireless) and the higher layers of the OSI model. PMA is a crucial part of the physical layer in the OSI model. It handles the specifics of transmitting and receiving data over various physical media and interfaces with the higher layers to ensure the successful transmission and reception of data. Different networking technologies and standards have their own PMA implementations tailored to the characteristics of their respective physical media.

Precoding

PAM4 (Pulse Amplitude Modulation with 4 levels) precoding is a signal processing technique used in high-speed data communication systems to improve the quality and reliability of the transmitted data. Precoding is applied before transmitting PAM4 signals to mitigate impairments and reduce the likelihood of errors during data transmission. It is commonly used in optical and electrical communication systems.

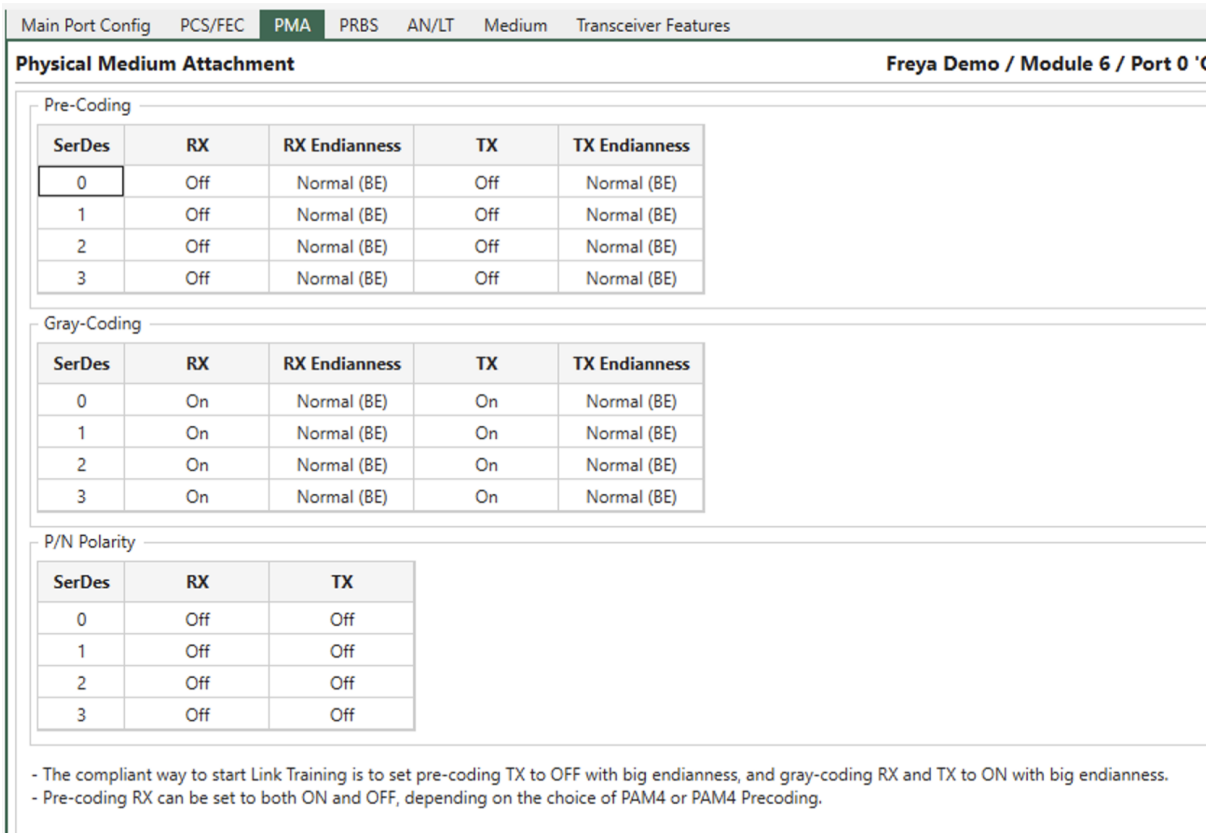


Fig. 5.41: Port Properties - PMA

Table 5.29: Precoding

Serdes	Pre-coding can be configured per Serdes
RX Config	You can set the RX to `OFF/ON`. (if Link Training is ON, `AUTO` is also available.)
RX Status	The actual RX status.
RX Endianness	Endianness in the RX direction, either `Normal` (big-endian) or `Reverted` (little-endian)
TX Config	You can set the RX to `OFF/ON`.
TX Status	The actual TX status.
TX Endianness	Endianness in the TX direction, either `Normal` (big-endian) or `Reverted` (little-endian)

Gray Coding

Gray coding in PAM4 (Pulse Amplitude Modulation with 4 levels) refers to a specific coding scheme used in optical and electrical communication systems that utilize PAM4 signaling. PAM4 is a modulation scheme that encodes information in the amplitude levels of the transmitted pulses. It uses four different amplitude levels to represent multiple bits of data per symbol.

In PAM4, Gray coding is often employed to ensure that transitions between symbol values are more controlled and have reduced susceptibility to errors during signal transmission. Gray coding for PAM4 is designed to minimize the bit error rate and make the system more robust.

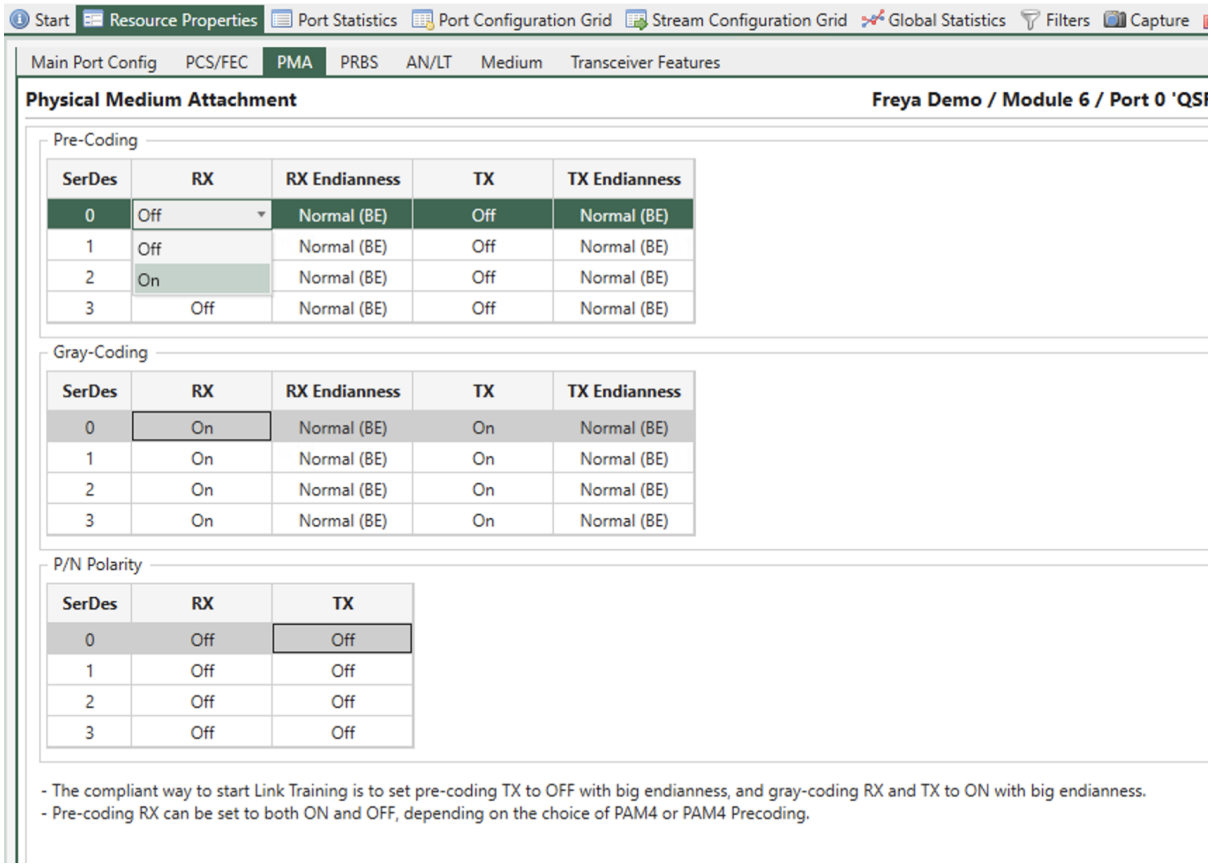


Fig. 5.42: Port Properties - Precoding Configuration

Here's a simplified example of how Gray coding can be applied in PAM4:

- In standard binary coding, you might represent four levels (level 1: 00, level 2: 01, level 3: 10, level 4: 11) using two bits each. However, the transition between these levels can be error-prone due to noise and other factors.
- With Gray coding, the mapping is adjusted to minimize the chance of errors during transitions. For PAM4, the Gray code might look something like this:
  - Level 0 (00) is represented as 00
  - Level 1 (01) is represented as 01
  - Level 2 (10) is represented as 11
  - Level 3 (11) is represented as 10

In this Gray-coded representation, you can see that adjacent symbol values differ in only one bit, reducing the likelihood of errors during transitions between symbols. This improved transition behavior helps in achieving better signal integrity and lower bit error rates in PAM4 communication systems, which is particularly important in high-speed data transmission over optical and electrical channels.

Table 5.30: Gray Coding

Serdes	Gray coding can be configured per Serdes
RX Config	You can set the RX to `OFF/ON`.
RX Status	The actual RX status.
RX Endianness	Endianness in the RX direction, either `Normal` (big-endian) or `Reverted` (little-endian)
TX Config	You can set the RX to `OFF/ON`.
TX Status	The actual TX status
TX Endianness	Endianness in the TX direction, either `Normal` (big-endian) or `Reverted` (little-endian)

Important

The compliant way to start Link Training is:

- Set **pre-coding TX** to `OFF`, `Normal (Big Endian)`.
- Set **Gray-coding RX and TX** to `ON`, `Normal (Big Endian)`.
- Pre-coding RX set `ON` will let the port request **PAM4 Precoding** from the link partner during Link Training.
- Pre-coding RX set `OFF` will let the port request **PAM4** from the link partner during Link Training.

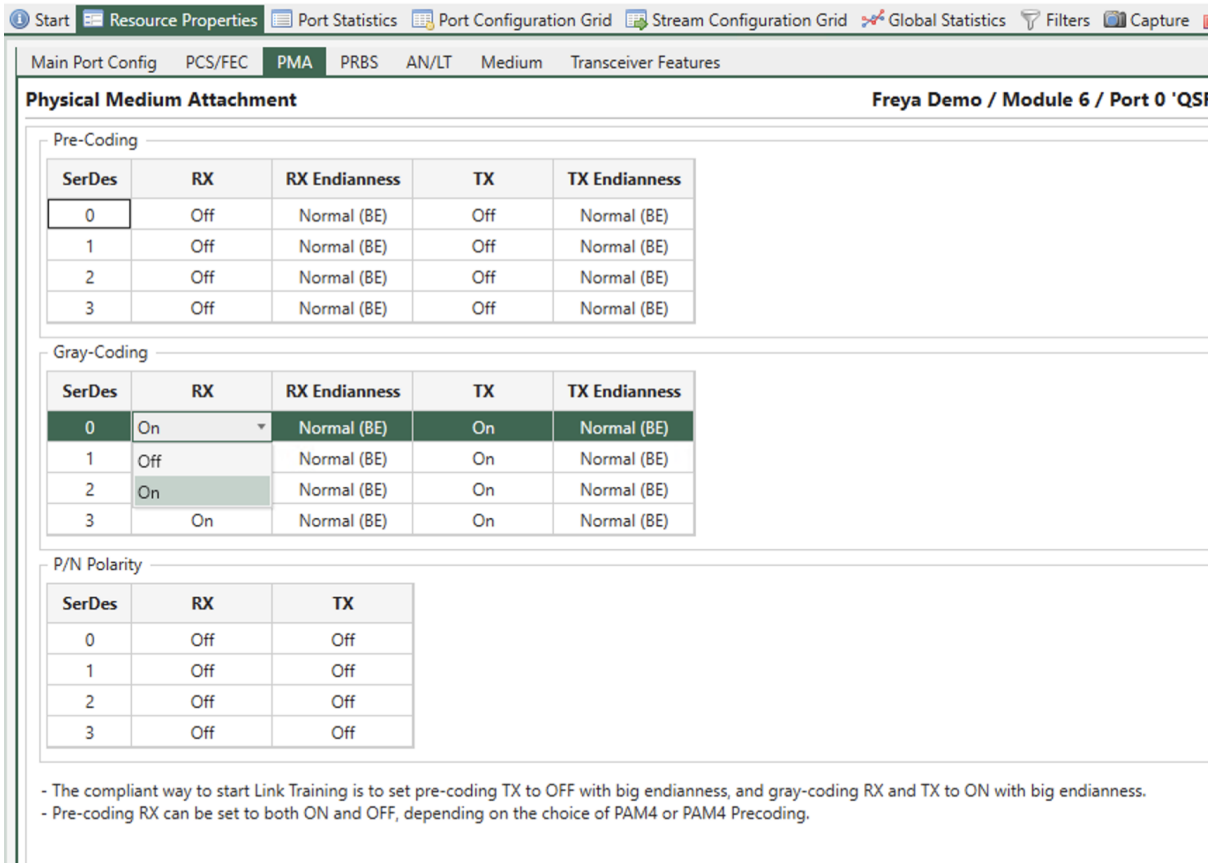


Fig. 5.43: Port Properties - Gray Coding Configuration

P/N Polarity

In a SERDES link, you transmit data at high speeds over differential pairs. Each pair has two signals:

- P (positive)
- N (negative)

Normally, the transmitter sends the differential signal correctly: when the P is high, N is low, and vice versa. However, in real hardware, especially during PCB design or cable assembly, sometimes the P and N wires get accidentally swapped.

You can enable the P/N polarity swap of a SERDES in PMA tab in XenaManager, as shown in the screenshot below. By default, the RX and TX P/N polarity swap is off.

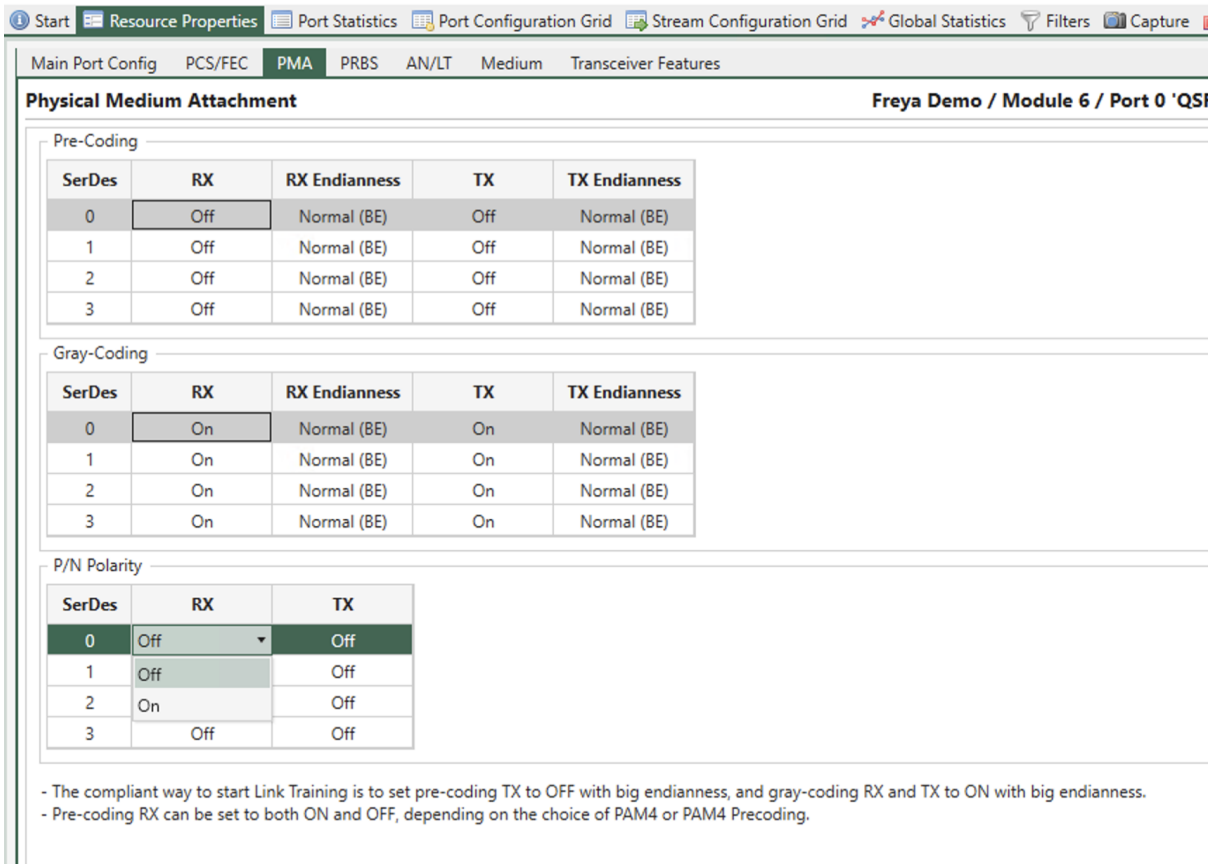


Fig. 5.44: Port Properties - P/N Polarity Configuration

PRBS

Pseudo Random Binary Sequence (*PRBS*) is a technique used for testing and evaluating the performance of Ethernet interfaces, particularly those operating at high speeds. PRBS is a type of test pattern used to generate a known, repetitive, and pseudorandom sequence of binary (0s and 1s) data.

All ports supporting Layer 1 PRBS testing will support the Polynomial `PRBS31`. If more are supported they can be selected from the drop down menu you get when you click on the selection box. Some ports supporting Layer 1 PRBS testing will support `Invert Pattern`. Check the

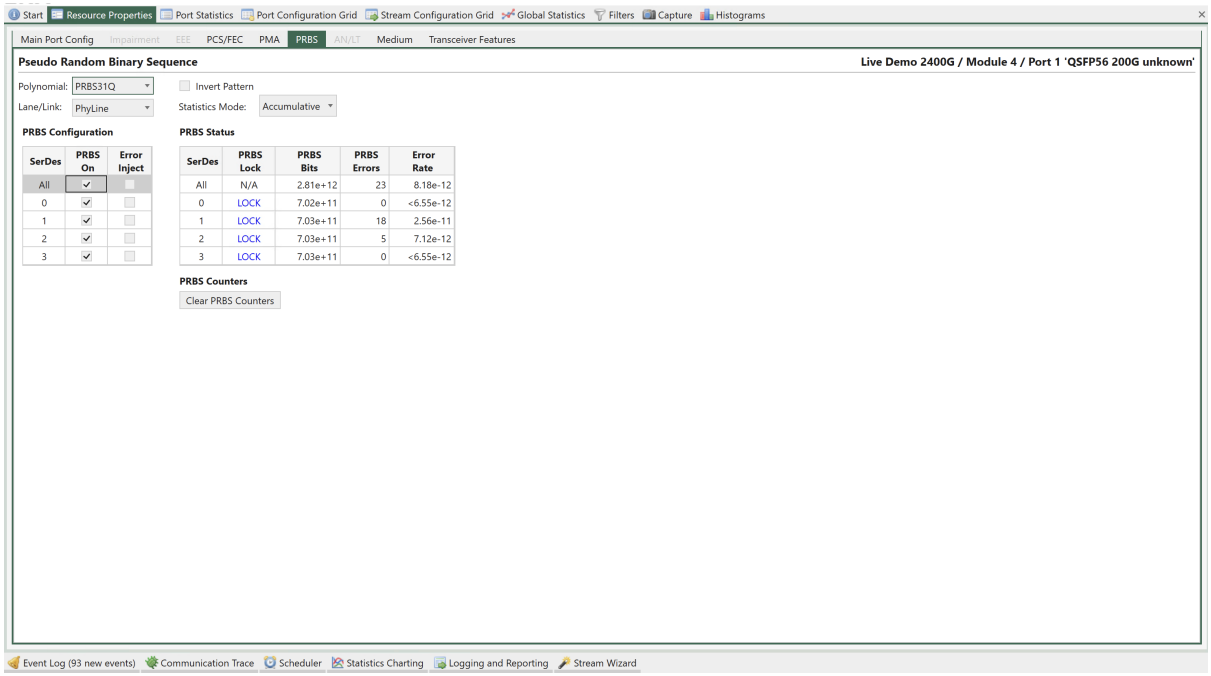


Fig. 5.45: Port Properties - PRBS

box to invert the test pattern if supported by the port. With *Statistics mode* `Accumulative`, PRBS Status Counters are accumulated since last time *Clear PRBS Counters* was pressed. With Last Second PRBS Status Counters are shown for the last second. Lane/Link is for future use.

PRBS Configuration

The physical lanes of ports supporting Layer 1 PRBS testing can be set to PRBS mode, where they transmit a *PRBS* bit pattern, which can be useful for testing physical cabling and connectors.

On the transmit side, you select whether each lane should be in PRBS mode, and also whether it should be subject to error injection:

Test Patterns

On **Z800 Freya 112G** serdes, available patterns are:

- PRBS13Q (IEEE 802.3 Claus 120.5.11.2.1)
- PRBS31Q (IEEE 802.3 Claus 120.5.11.2.2)
- SSPRQ (IEEE 802.3 Claus 120.5.11.2.3)
- Square Wave (IEEE 802.3 Claus 120.5.11.2.4)

Note

Why SSPRQ?

The PRBS13Q pattern (IEEEE 802.3 Clause 120.5.11.2.1) is conveniently short (8191 symbols), but is much less stressful than long periods of random data. In contrast, the PRBS31Q

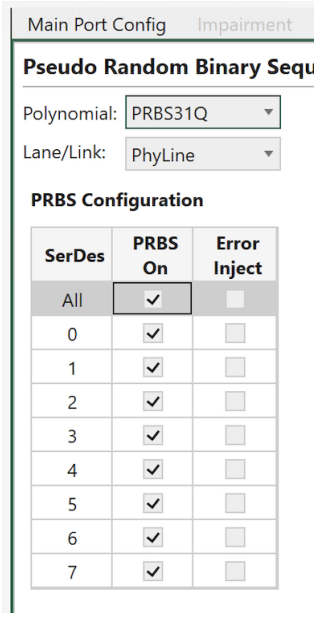


Fig. 5.46: Port Properties - PRBS TX configuration

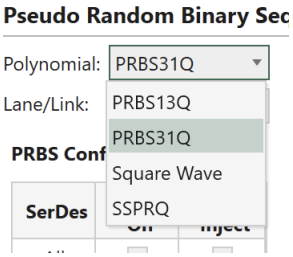


Fig. 5.47: Test Patters on Z800 Freya

pattern (IEEEE 802.3 Clause 120.5.11.2.2) is more stressful than long periods of random data. However, its length of 2,147,483,647 symbols makes it challenging for easy capture and analysis.

The SSPRQ pattern, short for Short Stress Pattern Random Quaternary, (IEEEE 802.3 Clause 120.5.11.2.3) strikes a balance, being short enough for capture while being more stressful than extended random data periods. SSPRQ consists of a repeating PAM4 symbol sequence of 2^16-1, derived from 4 sequences incorporating key stressors from PRBS31. Despite its stress-inducing nature, SSPRQ remains manageable for analysis with advanced tools such as Equalization and Jitter/Noise analysis, making it suitable for Optical TX testing purposes.

On **Z400 Thor 56G** serdes, available patterns are:

- PRBS7Q
- PRBS9Q
- PRBS10Q
- PRBS11Q
- PRBS13Q
- PRBS15Q
- PRBS20Q
- PRBS23Q
- PRBS31Q
- PRBS49Q
- PRBS58Q

On **Z100 Loki 25G** serdes, available patterns are:

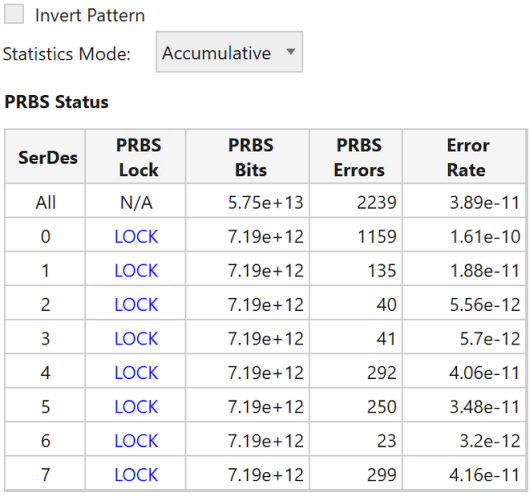
- PRBS31

Error Injection

Errors can be injected individually by clicking a button, or continuously by specifying a rate. Error injection also works for lanes that are not in PRBS mode, and can thus be used to simulate bit-level errors into the CAUI level.

PRBS Status

On the receive side, you can see whether each physical lane has locked onto the PRBS pattern, and the number of bit errors while in PRBS lock:



SerDes	PRBS Lock	PRBS Bits	PRBS Errors	Error Rate
All	N/A	5.75e+13	2239	3.89e-11
0	LOCK	7.19e+12	1159	1.61e-10
1	LOCK	7.19e+12	135	1.88e-11
2	LOCK	7.19e+12	40	5.56e-12
3	LOCK	7.19e+12	41	5.7e-12
4	LOCK	7.19e+12	292	4.06e-11
5	LOCK	7.19e+12	250	3.48e-11
6	LOCK	7.19e+12	23	3.2e-12
7	LOCK	7.19e+12	299	4.16e-11

Fig. 5.48: Port Properties - PRBS RX status

AN/LT

Auto-Negotiation and Link Training (AN/LT) provides functions to help you fine-tune the protocol to its optimal state, test interoperability between different vendors, and protocol compliance for different implementations.

Auto-negotiation (AN) was originally designed for Ethernet over twisted pair up to 1G. Beyond exchanging speed capabilities for the link participants, AN has evolved for today's Ethernet to include additional configuration information for establishing reliable and consistent connections. AN allows the devices at the end points of a link to negotiate common transmission parameters capabilities like speed and duplex mode, exchange extended page information and media signaling support. At higher speeds and signaling the choice of FEC may be relevant. It is during auto negotiation the end points of a link share their capabilities and choose the highest performance transmission mode they both support.

Once the ports in the link have completed the requisite AN information exchange and reached agreement, the link partners move to the next step, link training (LT), the exchange of Training Sequences. This is essential to tune the channels for optimal transmission. During link training the two end points of the link will exchange signals.

Auto-Restart

- The *ANLT Auto-Restart After Link Toggling* option has been introduced in tab *Resource Properties –> AN/LT*. Enable AN+LT auto-restart when a link down condition is detected. A "link down" state signifies the loss of a valid input signal, which can occur due to events such as cable unplugging and re-plugging, TX disable, or link flap on the link partner's end. The auto-restart process will continue until the link is re-established.
- The *ANLT Auto-Restart After Link Training Failure* option has been introduced in tab *Resource Properties → AN/LT*. If LT is enabled and experiences a failure on either side, the port will initiate the AN+LT restart process repeatedly until LT succeeds. This functionality is only applicable when LT is enabled.

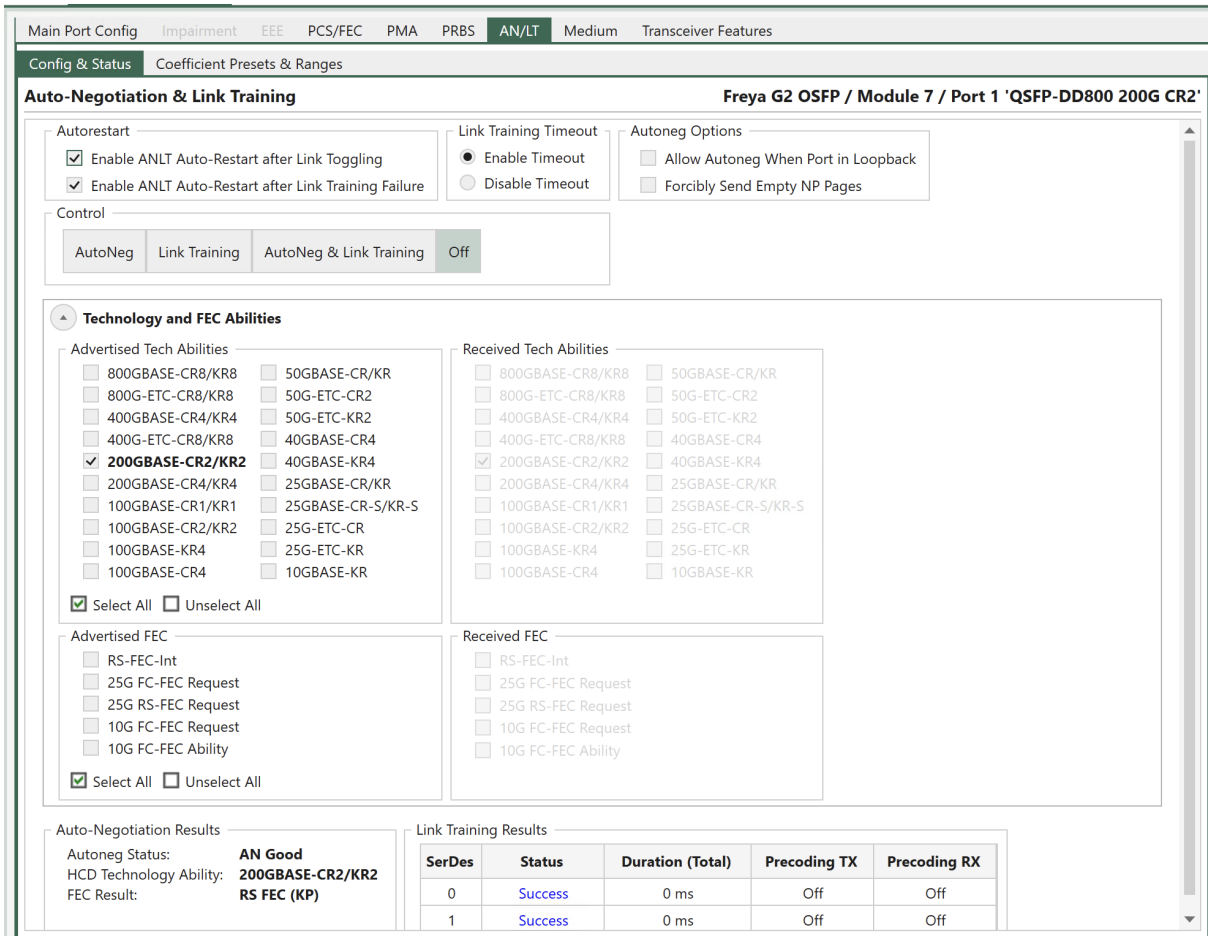


Fig. 5.49: Port Properties - AN/LT on Z800 Freya module with media L1/ANLT

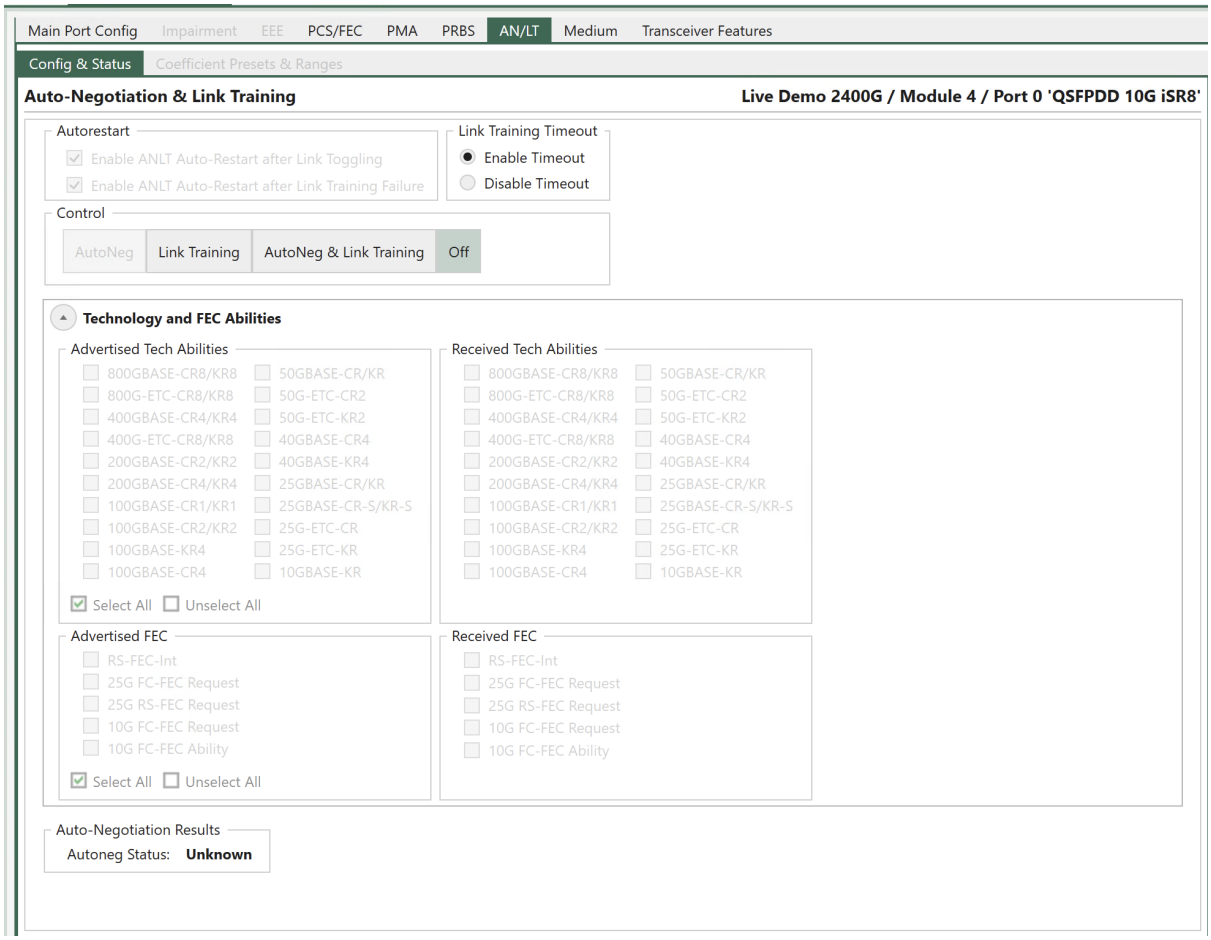


Fig. 5.50: Port Properties - AN/LT on Z400 Thor module

Link Training Timeout

The *Link Training Timeout* allows you to enable or disable link training timeout.

Autoneg Options

Autoneg Options allow you to change Autoneg's behavior that is different from the standard way.

- *Allow Autoneg When Port in Loopback* option makes it possible for autoneg to work even if the port is in loopback mode, i.e. the port TX side is transmitting to the port RX side. Standard autoneg is not design for port loopback scenario, but with this option enabled, the Z800 Freya port is able to carry out autoneg when in loopback.
- *Forcibly Send Empty NP Page* option controls whether the port sends empty Next Pages even if the pages are empty. This option helps you to be able to carry out successful autoneg interoperability test with various vendors.



Fig. 5.51: Autoneg Options

ANLT Control

For AN/LT control buttons:

- **Start AN**: the port starts to run auto-negotiation with the auto-negotiation configuration. Link training will not be run.
- **Start LT**: the port starts to run link training with the link training configuration. Auto-negotiation will not be run. While Link Training can be essential to make some electrical interfaces work, Auto Negotiation may not be required, if the link speed is fixed or if it can be manually set at both end points of a link.
- **Start AN+LT**: the port starts auto-negotiation and link training. Auto Negotiation and Link Training are in principle two independent processes. However, when both are to be done, Auto Negotiation must start first to determine the overall mode for a link and then the Link Training. Hereby you get the sequence shown in the figure below.
- **Stop**: the port stops AN/LT. In some instances, Auto Negotiation and Link Training are not required to establish a communication path: High speed optical transceivers and interfaces typically only run at one speed, so there is no need the negotiate this. Link Training is only required for electrical interfaces - in some cases (e.g. when short cables are used) an electrical interface may become operational just using default settings of the terminal equipment in the communication path. The IEEE 802.3 by specification allows for force connect over electrical interfaces in these instances.

To manual restart AN/LT, you will need to click Stop and then click one of the three start buttons depending on your testing requirements.

Note

Start AN is only applicable on Z800 Freya module.



Fig. 5.52: Auto-Negotiation and Link Training Sequence

Autoneg Configuration

Note

Only applicable on Z800 Freya modules.

For auto-negotiation configuration:

- *Advertised Tech Abilities* ` allows you to select the technology abilities the port should advertise. Technology abilities that are supported by the port are in bold. To guarantee that port succeeds in finding the highest common denominator of technology abilities with the remote port, it is recommended to select only the supported technology abilities.
- *Advertised FEC* allows you to select the FEC abilities the port should advertise. FEC abilities that are supported by the port are in bold. To guarantee that port succeeds in FEC abilities negotiation, it is recommended to select only the supported FEC abilities.

  What you get and set are the 5 F bits in the ANEG frame, i.e. F0 - F4.

  F0 - F3 are only applicable for 10G and 25G SERDES.

  F4 is "RS-FEC-Int", which is only supported for 100G on 100G-SERDES.
- *Received Tech Abilities* displays the received technology abilities from the remote port.
- *Received FEC Abilities* displays the received FEC abilities from the remote port.
- ***Auto-Negotiation Results*** **includes**
  - Auto-Negotiation Status: the FSM state of the auto-negotiation protocol.
  - HCD Technology Ability: the highest common denominator of the technology ability negotiation. In case of unsuccess, "failed" will be displayed.
  - FEC Result: the FEC result. This result can also be found in PCS/FEC tab panel.
- ***Link Training Results*** **includes:**
  - Link training status
  - Link training duration, from start to both ends declares TRAINED
  - TX precoding config

- RX precoding status

On Z800 Freya module, the technology abilities supported by the current port setting are displayed in bold. You are allowed to select the advertised technology abilities and FEC capabilites as needed for auto-negotiation protocol testing, even though the current port setting doesn't support all the selected technology abilities. In case the HCD technology ability is not supported by the port, the port will choose to continue AN/LT with the default technology ability it is capable of.

Important

The PCS variant, i.e., IEEE or ETC, and FEC mode derived from AN/LT will update your manual configuration in *PCS/FEC*

Coefficient Presets

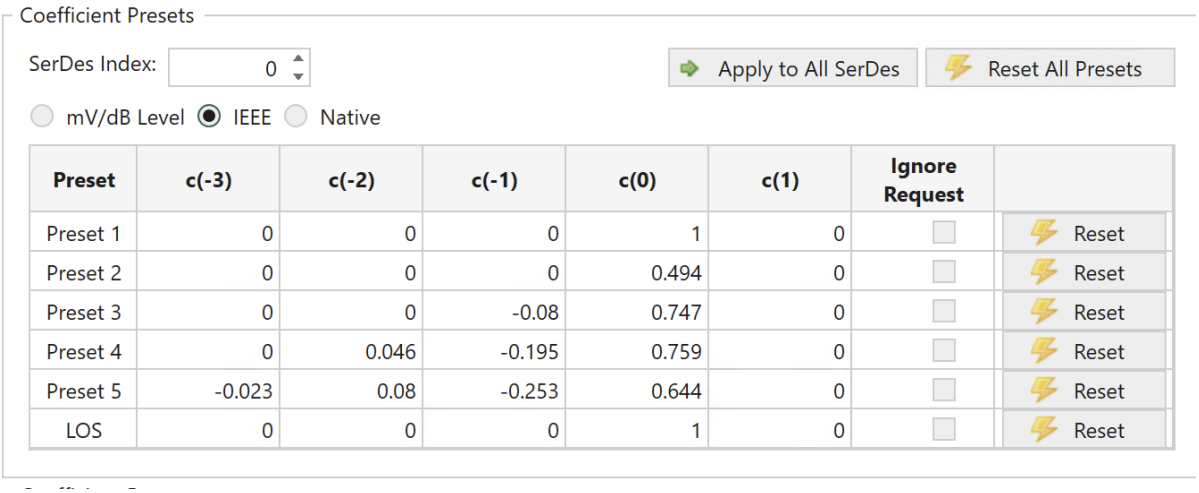


Fig. 5.53: AN/LT - Configurable Coefficient Presets

You can configure the port's Link Training Presets 1-5 and the loss-of-sync preset (LOS Preset) to your preferred values, provided they fall within the specified limits.

You can also configure the port to either ignore or accept the preset request.

- Ignore: the port acknowledges the preset request without acting upon it (acknowledge but no changes).
- Accept: the port acknowledges the preset request and apply the corresponding preset value

Coefficient Ranges and Responses

You can set the lower and upper bounds of the transmit equalizer for Link Training for the SerDes, as well as define the SerDes's response to increment or decrement requests when these bounds are reached.

The responses include:

- Auto. The response is automatically determined by the port.

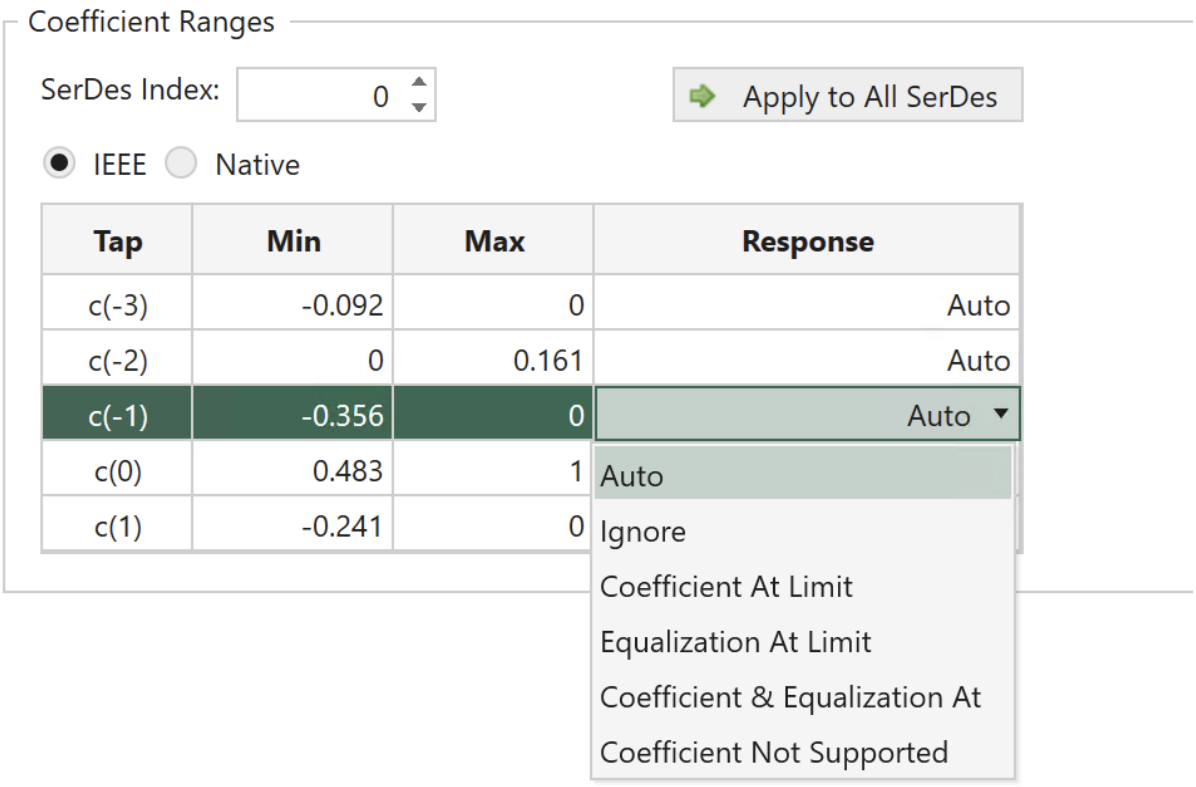


Fig. 5.54: AN/LT - Configurable Coefficient Ranges and Responses

- Ignore. The increment/decrement request will be "acknowledged" without acting upon it.
- Coefficient At Limit. Respond Coefficient At Limit (coefficient status = b010) to the increment/decrement request.
- Equalization AT Limit. Respond Equalization Limit (coefficient status = b100) to the increment/decrement request.
- Coefficient and Equalization AT Limit. Respond Coefficient At Limit and Equalization Limit (coefficient status = b110) to the increment/decrement request.
- Coefficient Not Supported. Respond Coefficient Not Supported (coefficient status = b011) to the increment/decrement request.

Medium

TX Taps

This sub-tab allows users to manually control and monitor the equalizer settings of the individual SerDes in the transmission direction (towards the transceiver cage). The affected SerDes is selected using the *SerDes Index*. This feature can, for example, be used to improve the signal quality over a directly attached copper cable (e.g. CR4) in the absence of automatic TX tuning auto-negotiation or to test a transceiver using various TX equalization settings.

Note

The words “cursors” and “taps” are used interchangeably.

The IEEE 802.3 standard specifies the cursors by means of FIR filter coefficients. The TX ports of the Z800 Freya module support 3 cursors: Main (`c(0)`), Post (`c(1)`), Pre1 (`c(-1)`), Pre2 (`c(-2)`), and Pre3 (`c(-3)`). The cursors are functionally defined by the Fig. 5.55 from IEEE 802.3ck.

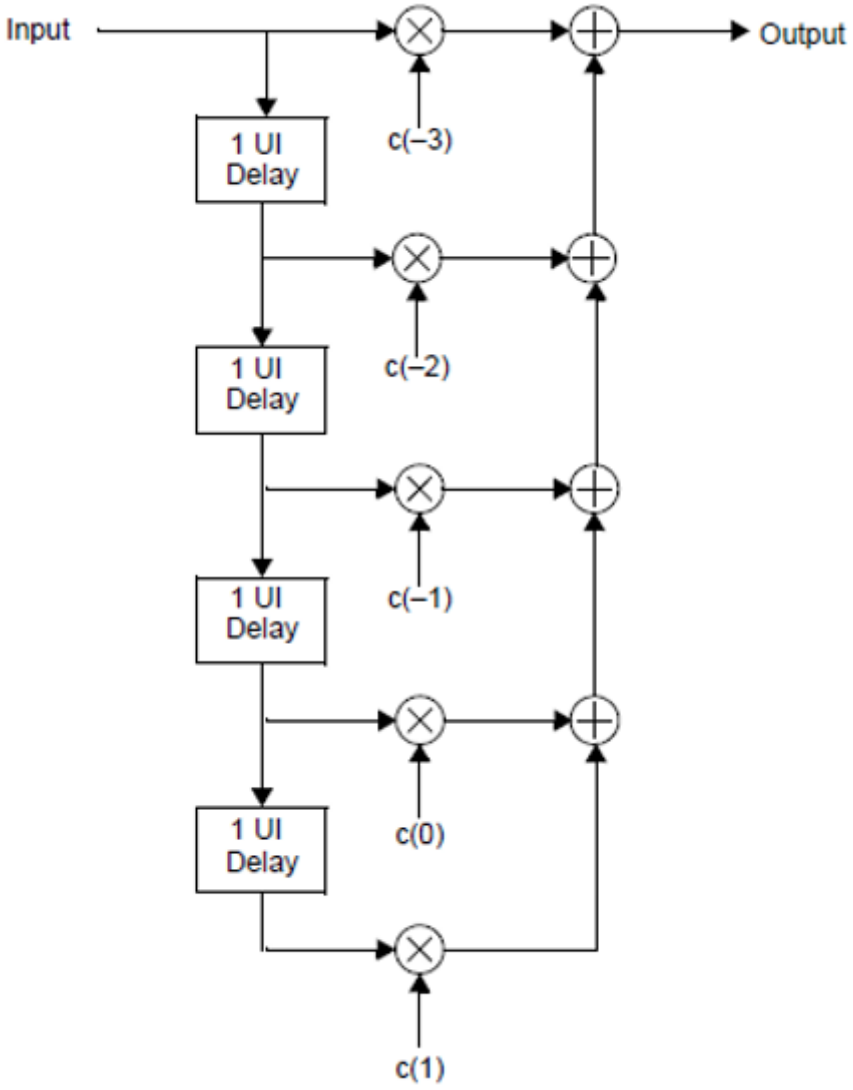


Fig. 5.55: Transmit equalizer functional model

When selecting IEEE format on the *Port* → *Medium* → *TX Taps* page, the c-coefficients can be inspected and changed directly. The following rules apply:

- 0.5 approx. <= c(0) <= 1
- -0.4 approx <= c(1) <= 0
- -0.4 approx <= c(-1) <= 0

- 0 <= c(-2) <= 0.25 approx.
- -0.25 approx <= c(-3) <= 0
- The sum of the absolute value of each coefficients must be <= 1.
- A sum of 1 corresponds to a TX output voltage swing of 1000 mVpp approximately.

In XenaManager, the tap settings of Z800 Freya can be entered in 3 different formats:

- mV/dB Level format (default format)
- IEEE format
- Native format (for internal development use)

mV/dB Level Format (default format)

The Main cursor (`c(0)`) is specified as the voltage swing in `mV` assuming all other cursors are 0.

The Pre- and Post cursors are entered as the equalization level in dB. The equalization level for any Pre/Post cursor, `c(n)`, is defined as:

$$E = 20 * Log((c(0) + |c(n)|)/(c(0)\check{}|c(n)|))$$

where `Log` is the base-10 logarithm, and `E` is the dB value of the cursor.

Since the equalization level for each cursor depends on both the cursor itself and the Main cursor, and since there are limits on cursor range as well as sum of (absolute) cursors, values will sometimes be changed automatically. In general, it is attempted to keep the dB levels as specified by the user, by lowering the output swing. When you enter values in this format, always verify that the resulting settings reflect your intention.

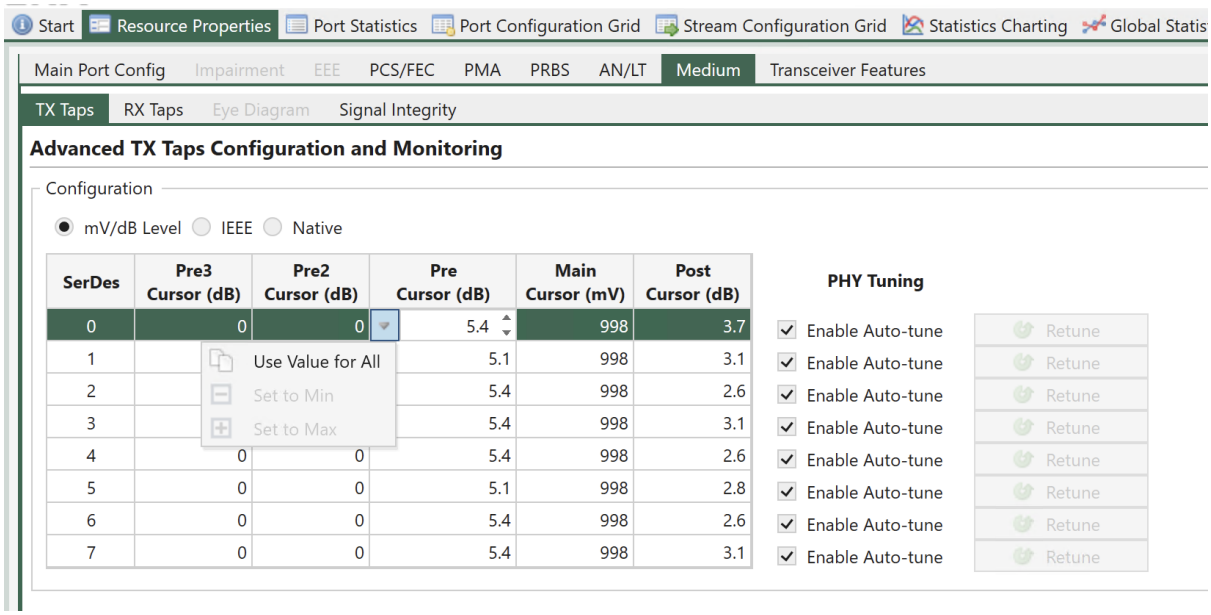


Fig. 5.56: Port Properties - TX Taps (mV/dB Level, available on Z800 Freya module)

IEEE Format

With this format, the c values (coefficients) are entered directly.

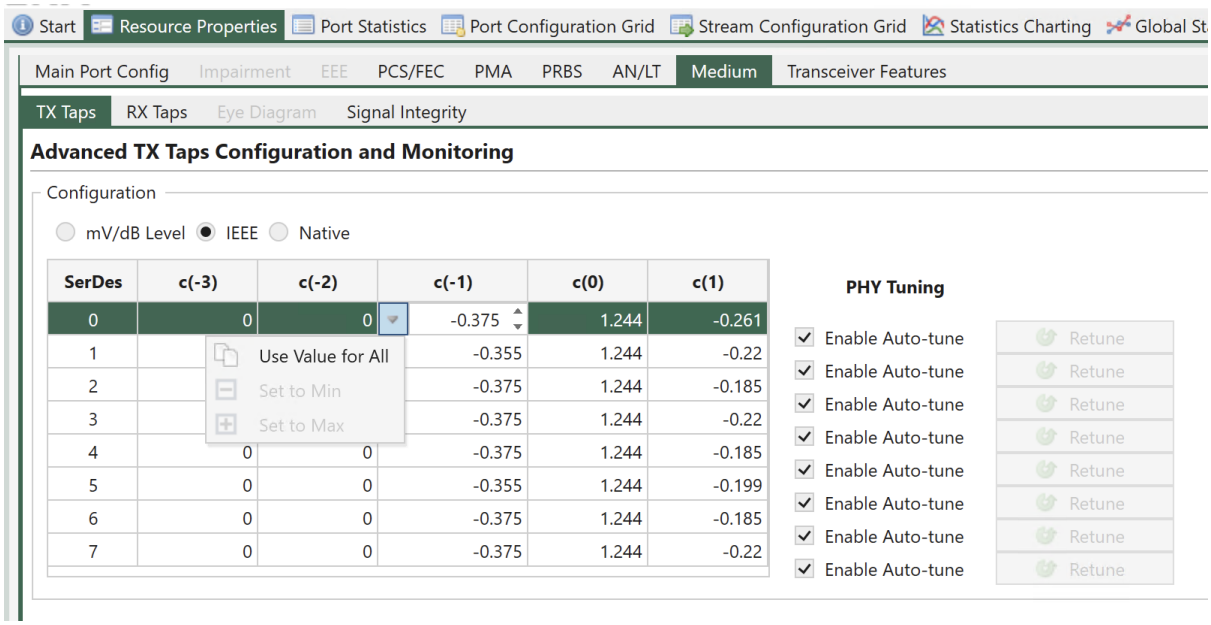


Fig. 5.57: Port Properties - TX Taps (IEEE Coefficient, available on Z800 Freya module)

Native Format

This is an internal, hardware near format. Values entered in the other formats are converted to Native format, choosing the values that makes the best fit to what is entered.

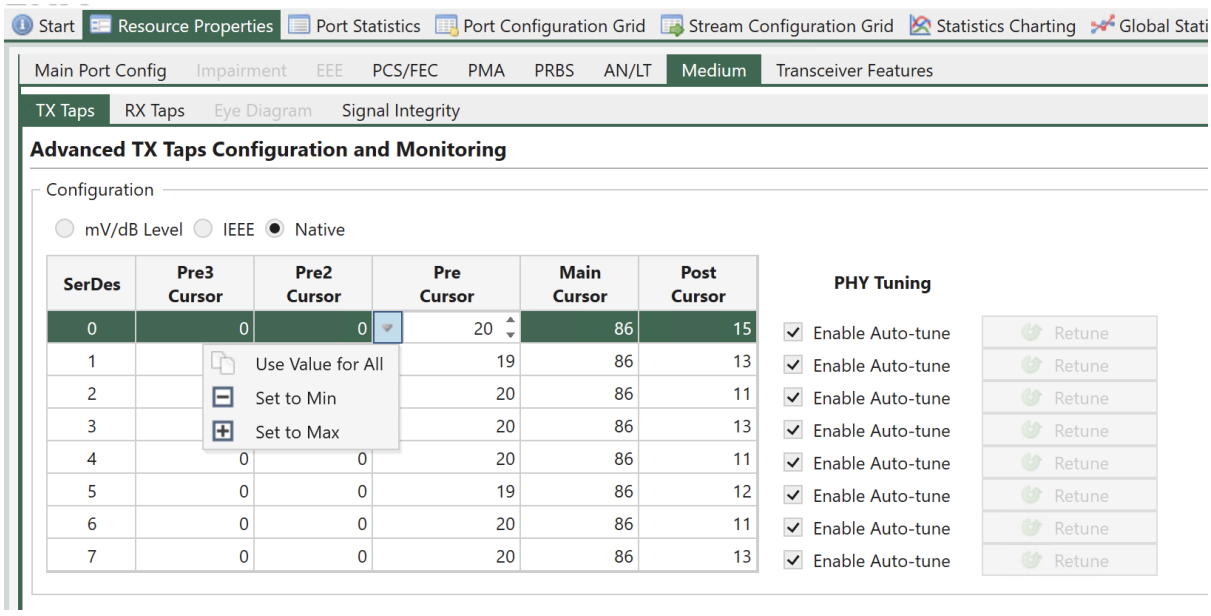


Fig. 5.58: Port Properties - TX Taps (Native Value)

> **Note**
>
> The values of the equalizer parameters to be used depends on the port type.

> **Important**
>
> mV/dB Level and IEEE views are only available on Z800 Freya modules

Enable Auto-adjust and *Retune* allows the user to control the tuning in the TX and RX direction: The user can enable or disable the automatic retuning, which is performed on the interfaces as soon as a signal is detected by the transceiver. This is useful if a bad signal causes the PHY to continuously retune or if for some other reason it is preferable to use manual retuning. Regardless of whether the automatic tuning is enabled, the user may also trigger a manual retuning.

RX Taps

Some modules only have one *RX CTLE* parameter, where-as some instead have a high band and low band frequency equalizer. Depending on module the available options will be made configurable.

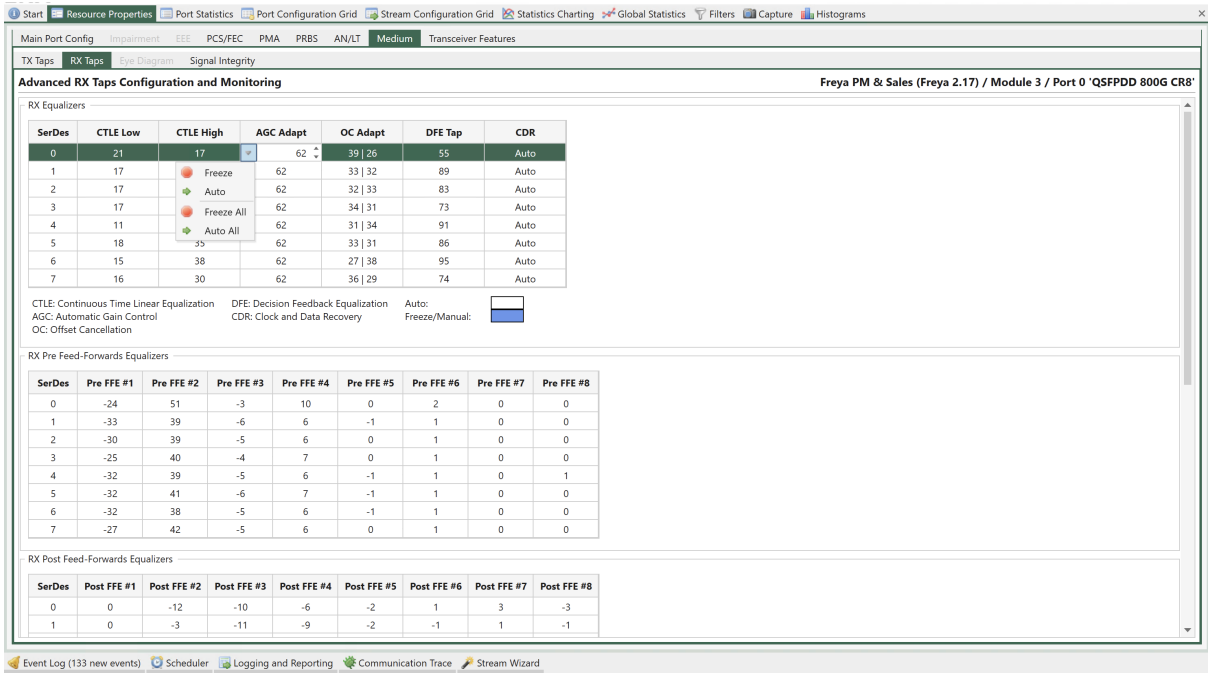


Fig. 5.59: Port Properties - RX Taps (available on Z800 Freya module)

Eye Diagram

Some ports support a bit-error-rate (BER) eye diagram. If the eye diagram is supported, a third section *Eye Diagram* is added to the *Medium* ` configuration section as shown in Fig. 5.61.

The bit-error-rate (BER) eye diagram allows the user to get a direct visual representation of the signal quality. The eye diagram is formed by changing the sampling point of the PHY step by

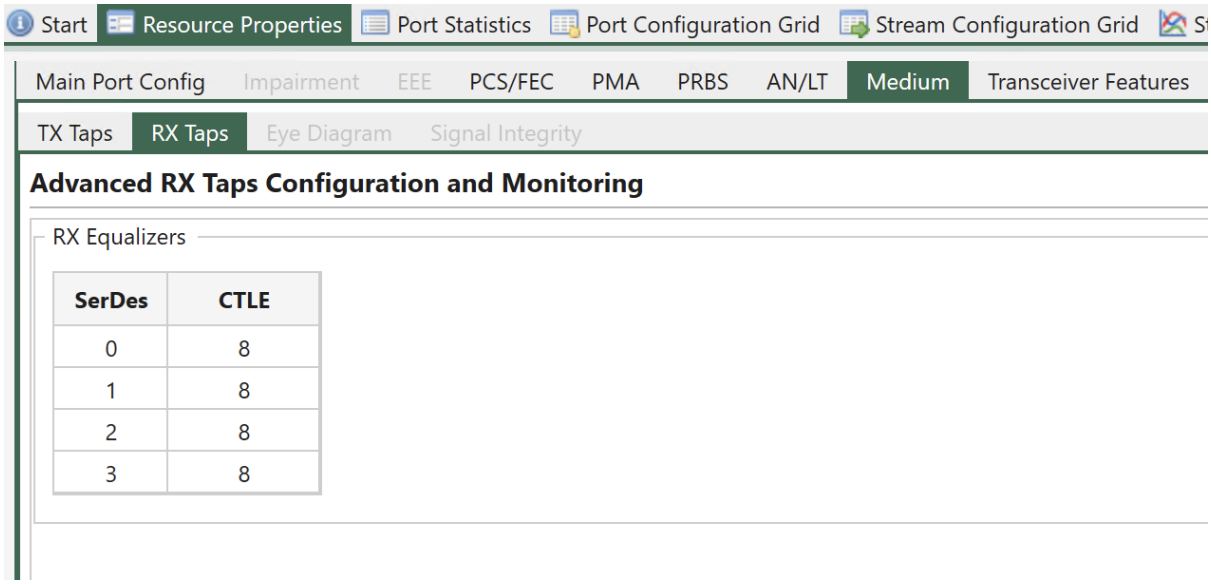


Fig. 5.60: Port Properties - RX Taps (available on Z400 Thor module)

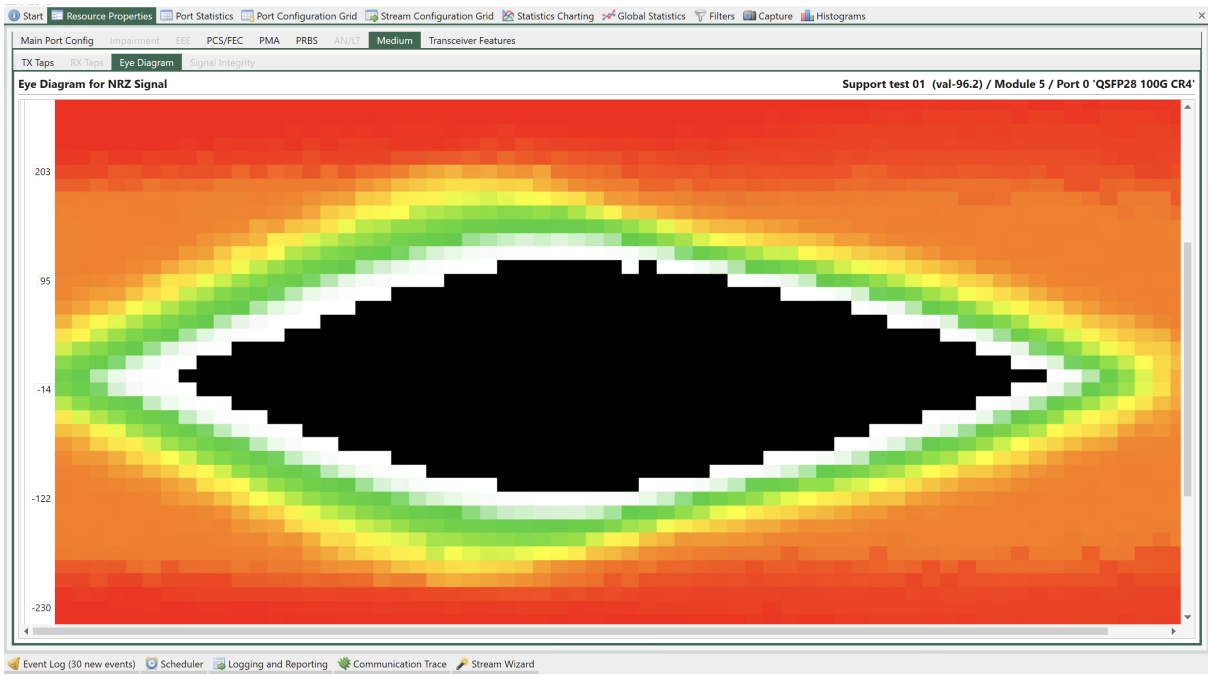


Fig. 5.61: Port Properties - Eye diagram (available on Z100 Loki module)

step in the time dimension (sampling delay) and the amplitude dimension (0/1 threshold). For each sampling point (x,y), 1 million bits are measured and the number of bit-errors is counted. A simple division gives the BER. The result is the BER eye diagram shown above.

The color map shows the measured bit-error-rate for each point going from 1 million (maximum red) to zero (black). The color scale is logarithmic. Higher resolutions give a more clear diagram and higher values of X and Y will also give a higher precision in the vertical and horizontal bathtub curve estimations, respectively (see Eye Data below). However, the time it takes to measure the eye is directly proportional to the number of sampling points (X*Y).

Diagram Control

This section controls the collection of the BER eye diagram and eye data as well as the parameters associated with this measurement.

Table 5.31: Diagram Control

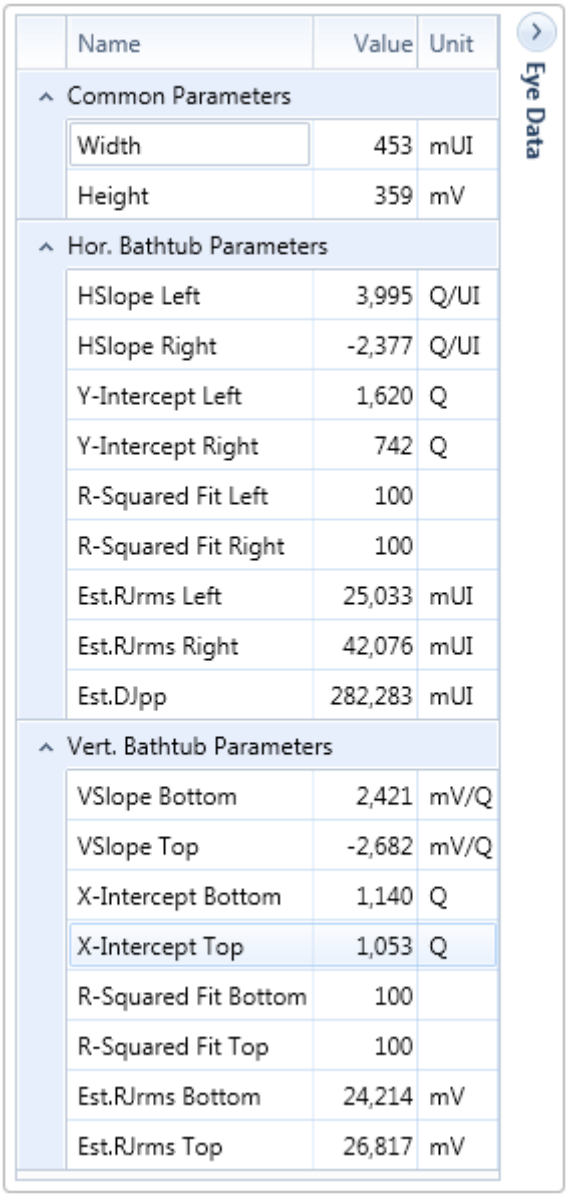
Property	Explanation
Collection state	Shows the current state of the eye diagram measurement.
Start/Stop Collection	Start/Stop eye diagram measurement.
Refresh Data	Re-read current eye diagram and eye data measurement from the chassis. Press this to view the current eye measurement for the SerDes if one already exists.
Generate Report	Generates a pdf report of the eye diagram. The report is stored in the following folder: `\Documents\Xena\XenaManager\Reports`
X Resolution	Resolution of the time-axis (horizontal). Calculated from X Exponent as `2^(exp) + 1`. Min/Max = 9 / 65.
Y Resolution	Resolution of the voltage axis (vertical). Calculated from Y Exponent as `2^(exp) - 1`. Min/Max = 7 / 255.
X Exponent	User-settable X resolution exponent. Min/Max = 3 / 6.
Y Exponent	User-settable Y resolution exponent. Min/Max = 3 / 8.
SerDes Index	The SerDes for which the settings and controls listed above, as well as the Per SerDes TX PHY Tuning listed below applies. Valid values = 0..3.

Note

Note that higher values of X and Y will give you a higher precision in the vertical and horizontal bathtub curve estimations, respectively. However, the time it takes to measure the eye is directly proportional to the number of sampling points (X*Y).

Eye Data

The eye data table provides an estimate of several parameters of the eye, including width, height, and jitter. Future releases will also include link BER estimates based on the horizontal and vertical bathtub curves.



Name	Value	Unit
Common Parameters		
Width	453	mUI
Height	359	mV
Hor. Bathtub Parameters		
HSlope Left	3,995	Q/UI
HSlope Right	-2,377	Q/UI
Y-Intercept Left	1,620	Q
Y-Intercept Right	742	Q
R-Squared Fit Left	100	
R-Squared Fit Right	100	
Est.RJrms Left	25,033	mUI
Est.RJrms Right	42,076	mUI
Est.DJpp	282,283	mUI
Vert. Bathtub Parameters		
VSlope Bottom	2,421	mV/Q
VSlope Top	-2,682	mV/Q
X-Intercept Bottom	1,140	Q
X-Intercept Top	1,053	Q
R-Squared Fit Bottom	100	
R-Squared Fit Top	100	
Est.RJrms Bottom	24,214	mV
Est.RJrms Top	26,817	mV

Fig. 5.62: Eye data

Table 5.32: Eye Data

Name	Description
Width	Estimated eye-width in mUI
Height	Estimated eye-height in mV
HSlope left	Left slope of the horizontal bathtub curve
HSlope right	Right slope of the horizontal bathtub curve
Y-intercept left	Intersection with the Y-axis on the left side
Y-intercept right	Intersection with the Y-axis on the right side
R-squared fit left	Quality assessment of the estimation. Max = 100.
R-squared fit right	Quality assessment of the estimation. Max = 100.
Est RJrms left	Estimated random jitter (rms) - left side
Est RJrms right	Estimated random jitter (rms) - right side
Est DJpp	Estimated deterministic jitter
VSlope bottom	Bottom slope of the vertical bathtub curve
VSlope top	Top slope of the vertical bathtub curve
X-intercept bottom	Intersection with the bottom X-axis
X-intercept top	Intersection with the top X-axis
R-squared fit bottom	Quality assessment of the estimation. Max = 100.
R-squared fit top	Quality assessment of the estimation. Max = 100.
Est RJrms bottom	Estimated random jitter (rms) - bottom
Est RJrms top	Estimated random jitter (rms) - top

Signal Integrity

Signal Integrity View (SIV) can be found under *Resource Properties* → *Medium* → *Signal Integrity*. With SIV, you can easily analyze PAM4/NRZ signal loss and noise impairments.

When a signal is transmitted from one switch port to another, the signal is degraded by several factors. The signal level will be reduced due to the inherent resistance of the wires causing gradual loss of signal as the signal moves from the transmitter to the receiver. Limitations in the bandwidth will lead to Inter-Symbol Interference (ISI) and inductive coupling between electrical lanes as well as connectors lead to cross talk. Impedance mismatches cause reflections (Return Loss). At both the transmitter and receiver jitter can occur and finally the signal will be degraded by thermal noise.

Signal integrity is a measure of the quality of the transmitted signal. It is commonly analyzed using a so-called eye-diagram which is constructed by overlapping the pulses in a bit sequence as shown for an NRZ-modulated signal. The signals are sampled at the center of the pulses and if the voltage level is higher than Vref it is interpreted as a “1” and if it is below Vref a “0”.

With no distortion on the transmission the eye is nice and open. However, in a real system the eye will look more “closed”, where some of the “1”s will be mistaken as “0”s and thus lead to

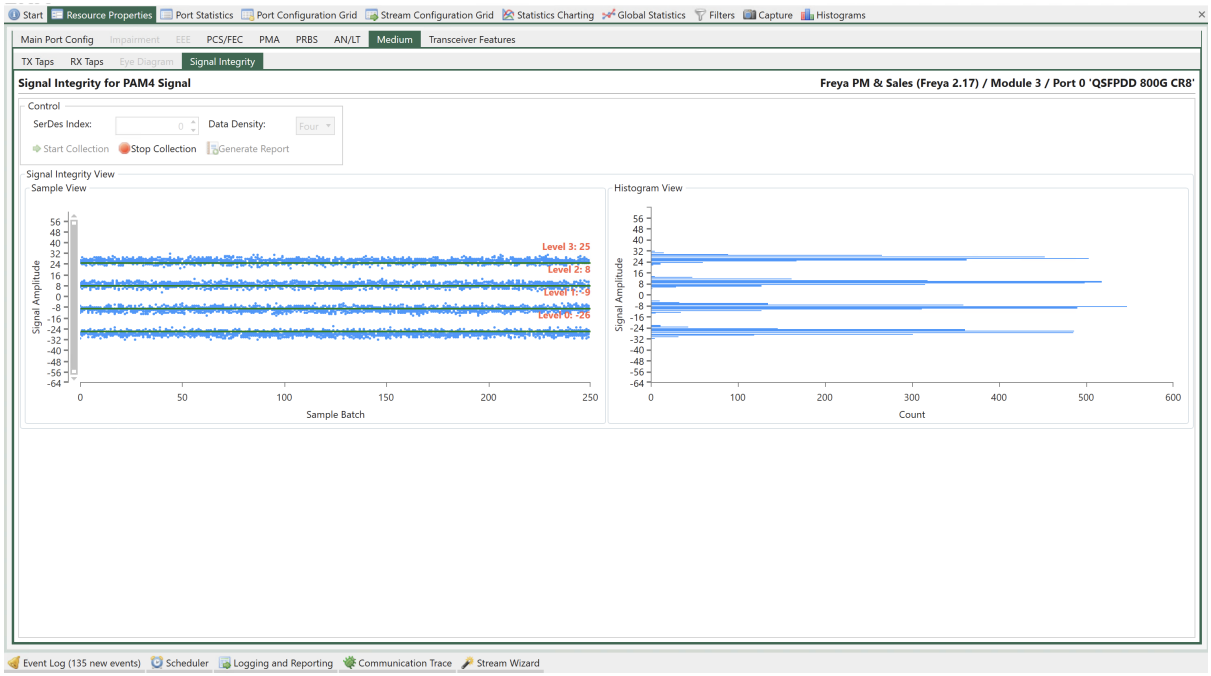


Fig. 5.63: Port Properties - Signal Integrity View (available on Z800 Freya module L1/ANLT media mode as well as NRZ mode)

bit errors.

In general, impairments to the signal level such as loss and noise will cause the eye to “close” in the vertical direction whereas delay and jitter impairments will cause the eye to “close” in the horizontal direction

For PAM4 modulated signals however, the situation is a bit more complicated. Since PAM4 has four levels, the eye diagram consists of three eyes rather than one. Obviously the PAM4 eyes are much narrower than the NRZ eye which means that PAM4 is less tolerant to distortions than NRZ.

For very high speeds, the pre-FEC eye diagram is likely going to be almost closed and difficult to use for analyzing signal integrity. Instead, a vertical slicer eye diagram is used. The vertical slicer eye is constructed by plotting the sampled, recovered signal level for a large number of samples.

The sample view clearly shows the effects of signal loss and noise. The histogram view next to it shows the distribution shape of each slice, making it easier for you to tell the characteristics of each eye slice.

Table 5.33: SIV Control

Property	Explanation
SerDes Index	The index of the serdes lane you want to scan signal integrity.
Data Density	Controls the number of data samples to show on the UI. A higher density value gives you more data points on the view at the cost of higher computational resource used by the GUI application.
Start Collection	Start signal integrity data collection.
Stop Collection	Stop signal integrity data collection.
Generate Report	Generate a PDF report including the the sample view and histogram view.

Sample View

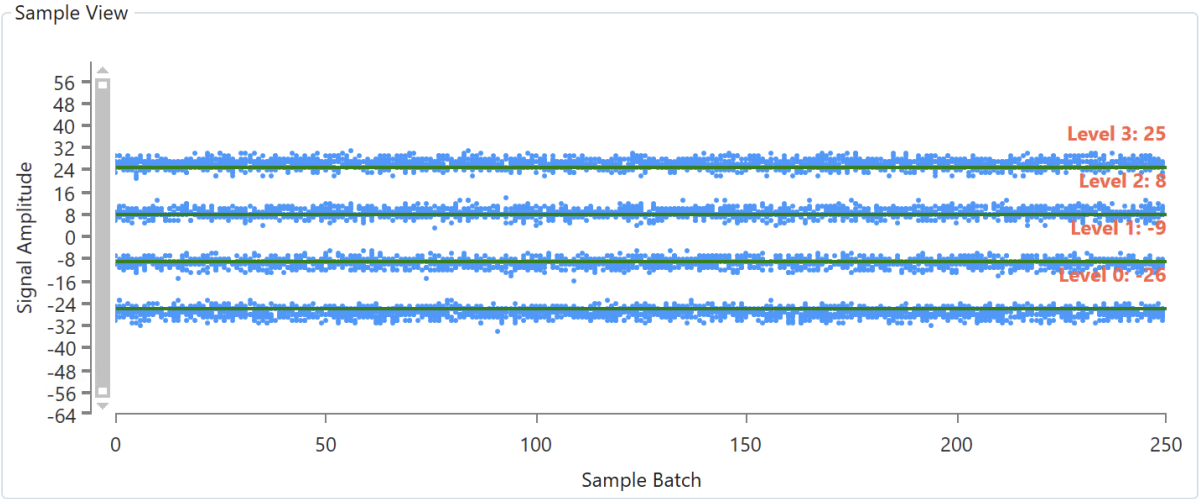


Fig. 5.64: Signal Integrity View - Sample View (available on Z800 Freya module - L1/ANLT media mode as well as NRZ mode)

For PAM4 signal, there are 4 levels, each of which indicates an amplitude level in PAM4. Ideally, if the PAM4 signal has no distortion, you should see 4 straight lines. As the PAM4 signal integrity degrades, you will see more scattered data points. The red text, i.e., Level 3, Level 2, Level 1, and Level 0 on the left show the average amplitude of each level.

For NRZ signal, two of the levels are set to the boundaries of the signal, and the other two are used for the signal.

The y-axis is a relative scale from -64 to 63. The x-axis is sample batch.

Histogram View

The histogram view shows the distribution of signal integrity data points. Ideally, if the signal has no distortion, you should see 4 straight lines for PAM4 and 2 for NRZ. As the signal integrity degrades, the distribution diagrams become “wider and shorter”

The y-axis scale is the same as the sample view, from -64 to 63. The x-axis is the data point count.

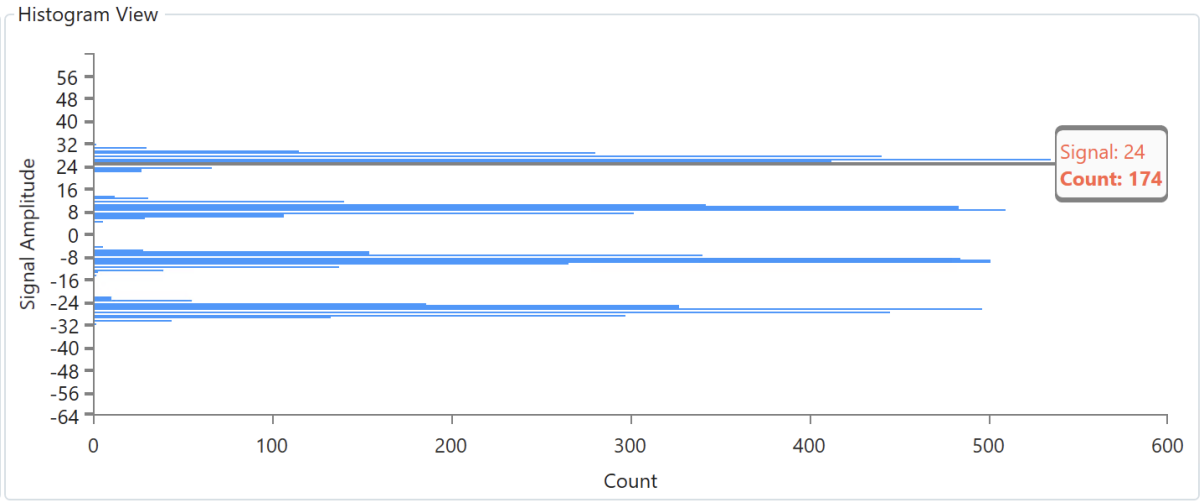


Fig. 5.65: Signal Integrity View - Histogram View (available on Z800 Freya module - L1/ANLT media mode)

Transceiver Features

This section only applies to ports that support direct access to their transceiver through a well-defined register interface such as the MII register interface.

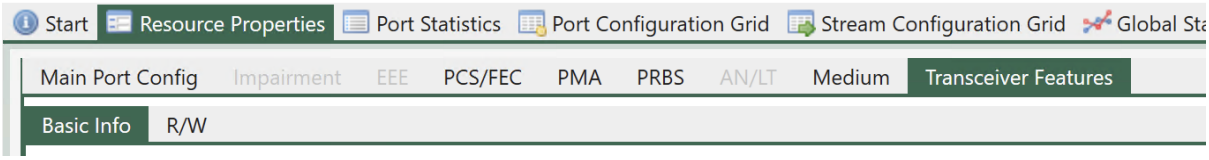


Fig. 5.66: Port Properties - Transceiver features

This function is mainly provided for debugging purposes and will normally not be required for ordinary test usage.

Basic Info

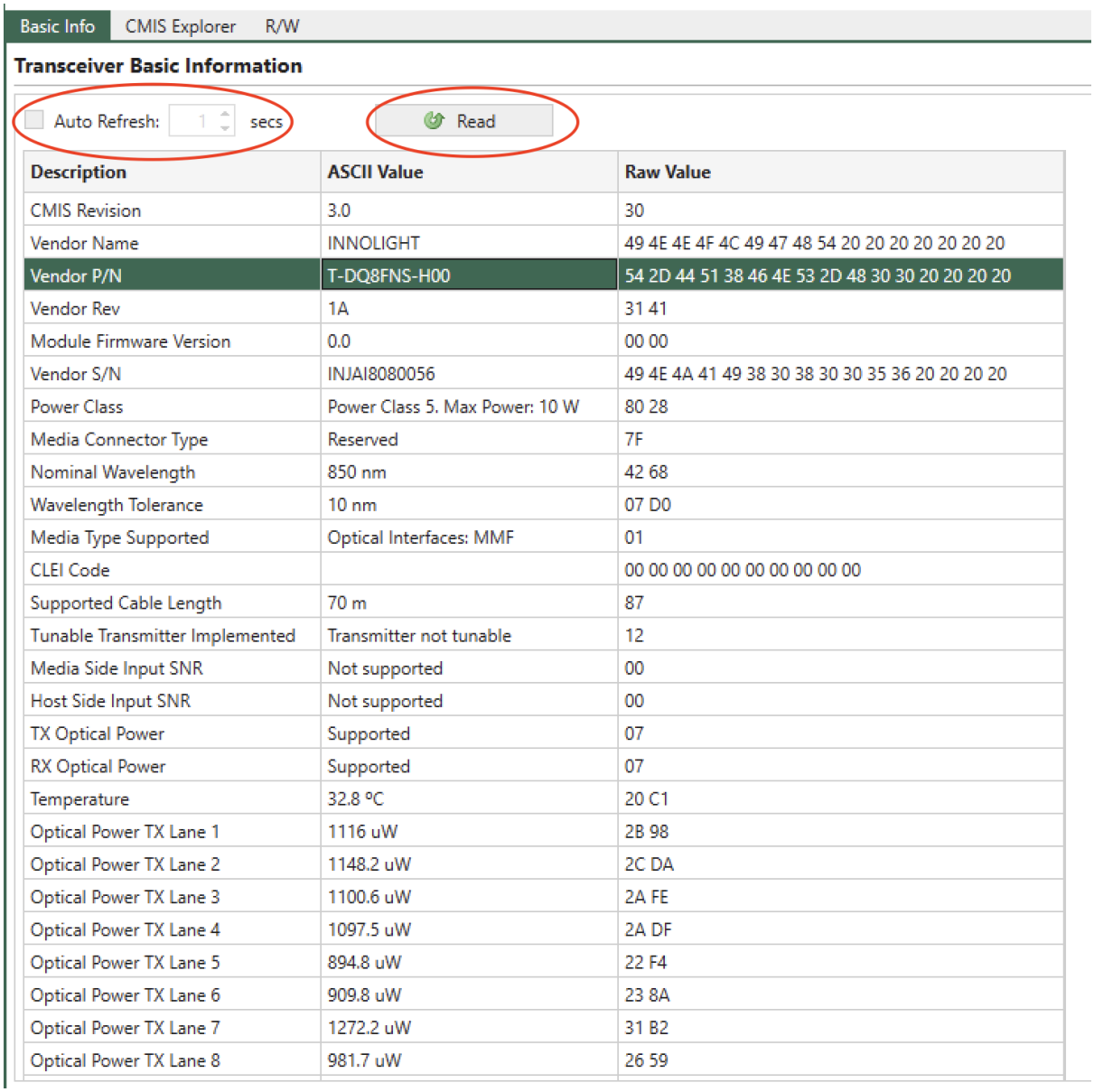
Basic Info tab panel provides **basic transceiver information summary**. The information on this page is obtained from reading management interface of the transceiver and may not necessarily be the same as what is indicated in its data sheet.

Read Table Info

On-demand read and auto-refresh.

- Clicking *Read* button trigger an on-demand read the transceiver memory.
- If you need to periodically refresh, you can enable auto-refresh by enabling *Auto Refresh* with a polling interval in seconds.

Important



Description	ASCII Value	Raw Value
CMIS Revision	3.0	30
Vendor Name	INNOLIGHT	49 4E 4E 4F 4C 49 47 48 54 20 20 20 20 20 20 20
Vendor P/N	T-DQ8FNS-H00	54 2D 44 51 38 46 4E 53 2D 48 30 30 20 20 20 20
Vendor Rev	1A	31 41
Module Firmware Version	0.0	00 00
Vendor S/N	INJAI8080056	49 4E 4A 41 49 38 30 38 30 30 35 36 20 20 20 20
Power Class	Power Class 5. Max Power: 10 W	80 28
Media Connector Type	Reserved	7F
Nominal Wavelength	850 nm	42 68
Wavelength Tolerance	10 nm	07 D0
Media Type Supported	Optical Interfaces: MMF	01
CLEI Code		00 00 00 00 00 00 00 00 00 00
Supported Cable Length	70 m	87
Tunable Transmitter Implemented	Transmitter not tunable	12
Media Side Input SNR	Not supported	00
Host Side Input SNR	Not supported	00
TX Optical Power	Supported	07
RX Optical Power	Supported	07
Temperature	32.8 °C	20 C1
Optical Power TX Lane 1	1116 uW	2B 98
Optical Power TX Lane 2	1148.2 uW	2C DA
Optical Power TX Lane 3	1100.6 uW	2A FE
Optical Power TX Lane 4	1097.5 uW	2A DF
Optical Power TX Lane 5	894.8 uW	22 F4
Optical Power TX Lane 6	909.8 uW	23 8A
Optical Power TX Lane 7	1272.2 uW	31 B2
Optical Power TX Lane 8	981.7 uW	26 59

Fig. 5.67: Basic Info Table

The GUI only reads the transceiver memory when you click the *Basic Info* tab. Clicking on a port doesn't trigger a read operation on the transceiver. This is sometimes useful when you don't want the GUI to interrupt your operations on the transceiver.

Search

The Basic Info table allows you to search the description by `Ctrl-F`, as shown below.

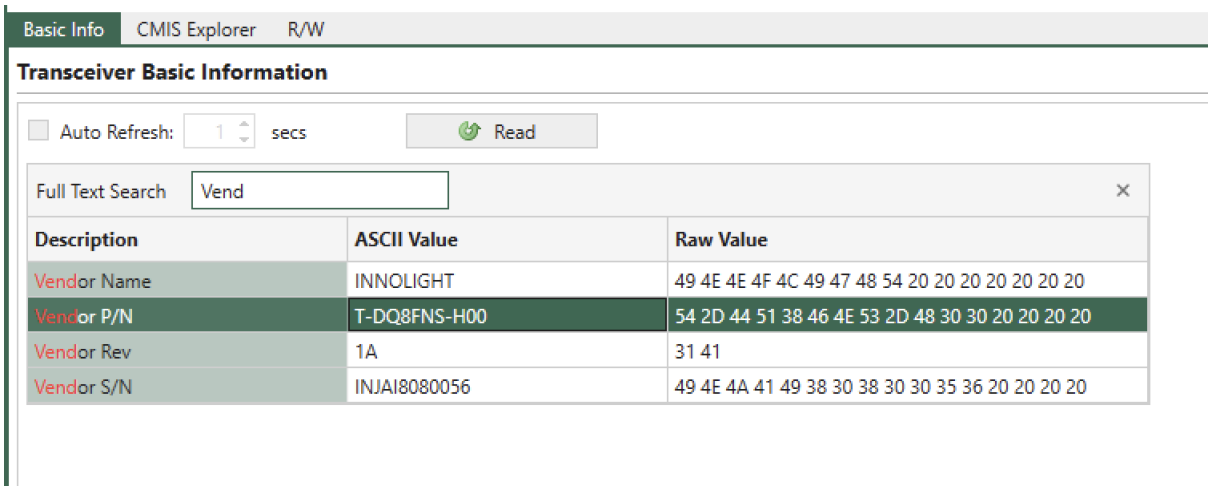


Fig. 5.68: Basic Info Table Search Bar

CMIS Explorer

CMIS Explorer provides a **fully customizable memory byte view** of the transceiver memory that is plugged in the socket cage. CMIS Explorer can be found under *Port* → *Resource Properties* → *Transceiver Features*.

Customization

The CMIS Explorer allows you to customize the transceiver memory page definitions. You can:

- **Decide which page to display**, allowing you to make your own byte viewer.
- **Specify which bytes on a page to show**, helping you concentrate on the bytes you work with.
- **Change descriptions for both pages and bytes**, which is very useful when you want to remember the the use of some particular bytes on a page.

When you launch XenaManager for the first time, the application creates a folder at `C:\Users\username\Documents\Xena\XenaManager\TcvrCmisDefs` and generates a CMIS page definition JSON file for each page according to the CMIS specification. Consequently, there are 257 definition files for each CMIS specification. These definition files are used by XenaManager to populate the CMIS page selection drop-down list.

When you want to customize a page, simple go to the corresponding folder and open the corresponding JSON file for that page, as shown in Fig. 5.70.

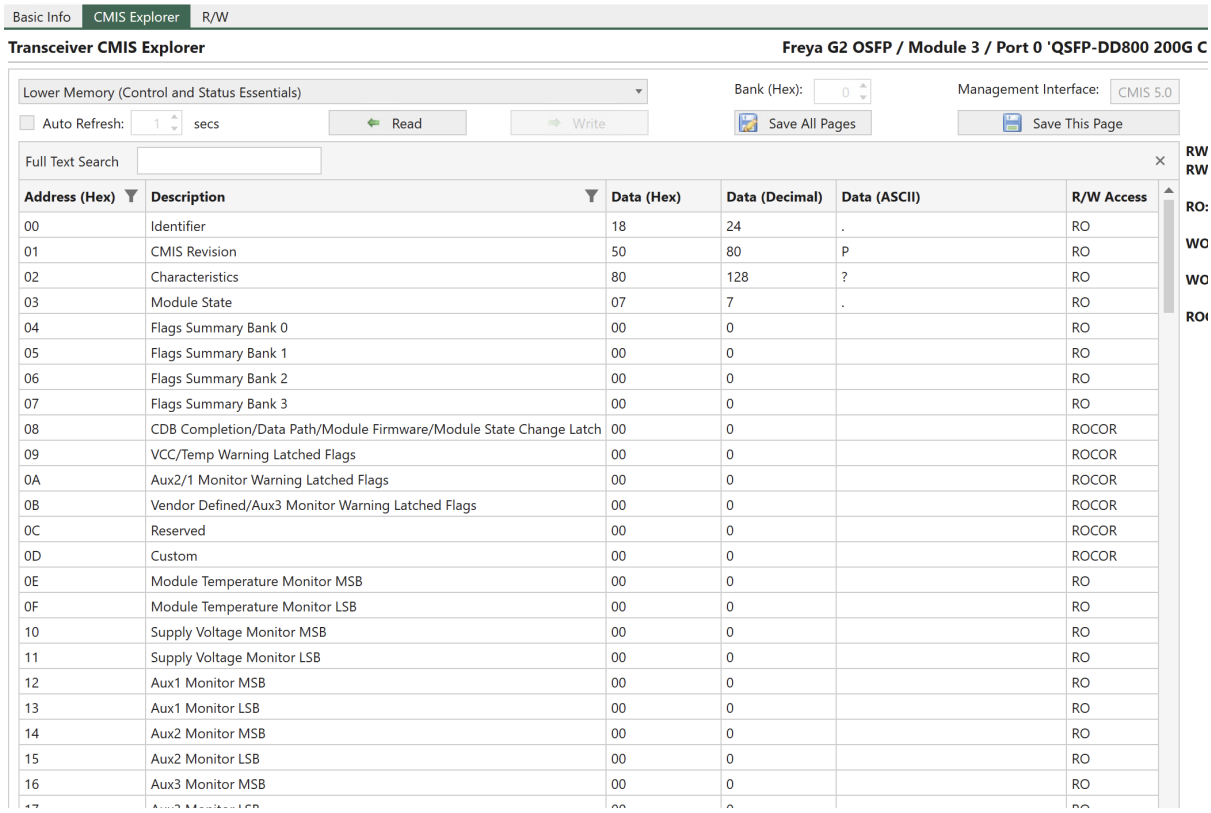


Fig. 5.69: CMIS Explorer Overview

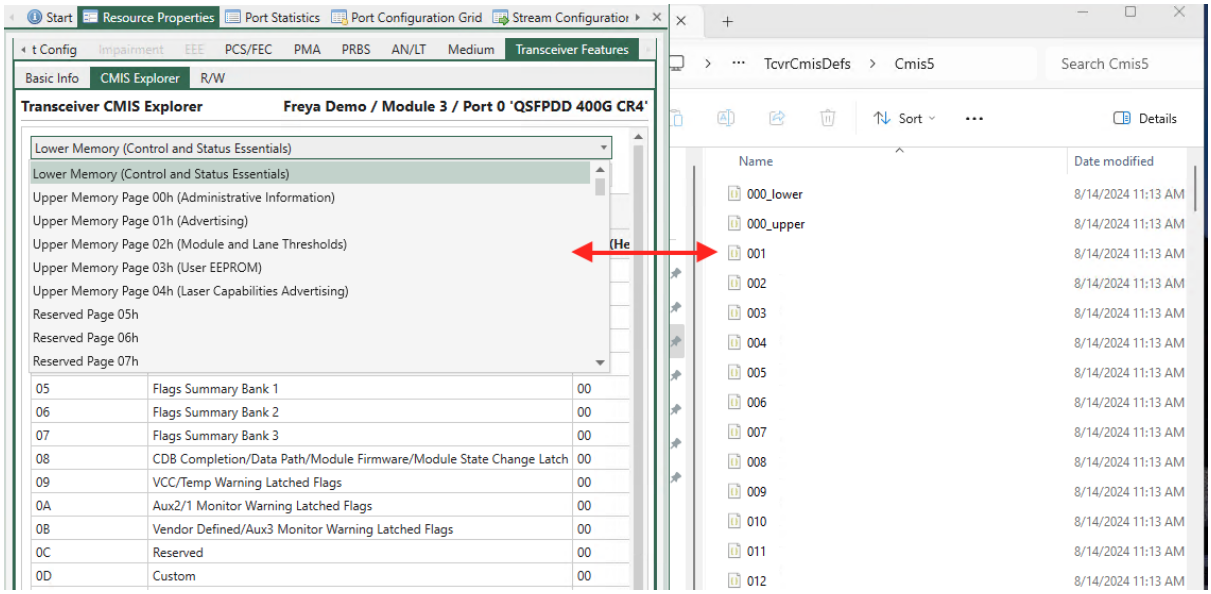


Fig. 5.70: Customize a page in CMIS Explorer

You can change the following values in the definition JSON files as shown in Fig. 5.71.:

- `page_description`: this is the page name shown in the drop-down selection list
- `reg_list`: this list contains the bytes you want the CMIS Explorer to show on this page
  - `description`: this is the description of the byte
  - `address`: address of the byte

Important

It is highly recommended that you save a copy of the entire `C:\Users\username\Documents\Xena\XenaManager\TcvrCmisDefs` folder before you customizing the CMIS Explorer. This is because XenaManager doesn't provide the undo feature.

Note

If a subfolder in `C:\Users\username\Documents\Xena\XenaManager\TcvrCmisDefs` cannot be found when XenaManager is launched, XenaManager will regenerate the default definition files for that subfolder.

For example, if `C:\Users\username\Documents\Xena\XenaManager\TcvrCmisDefs` is missing, XenaManager will generate `C:\Users\username\Documents\Xena\XenaManager\TcvrCmisDefs` and its default definition files. If the subfolder is not empty, XenaManager will not change any file in it.

Page Selection

Use the page selection drop-down list to select the page you want to view.

Bank Selection

Bank selection (for banked pages). Use the Bank (Hex) stepper field to select the bank.

Management Interface

View the CMIS specification version of the transceiver in *Management Interface*.

Description Column

Description of each byte.

Value Columns

Each table row corresponds to a byte on the page. The value of a byte is presented in three columns:

- Data (Hex): raw binary value of the byte in hex format.

```
{
  "version": "1.0",
  "cmis_rev": "Cmis5",
  "page": 5,
  "page_description": "Reserved Page 05h",
  "is_banked": false,
  "reg_list": [
    {
      "description": "Reserved",
      "address": 128,
      "access_type": "RW"
    },
    {
      "description": "Reserved",
      "address": 129,
      "access_type": "RW"
    },
    {
      "description": "Reserved",
      "address": 130,
      "access_type": "RW"
    },
    {
      "description": "Reserved",
      "address": 131,
      "access_type": "RW"
    },
    {
      "description": "Reserved",
      "address": 132,
      "access_type": "RW"
    },
    {
      "description": "Reserved",
      "address": 133,
      "access_type": "RW"
    },
```

Fig. 5.71: Definition file in JSON format

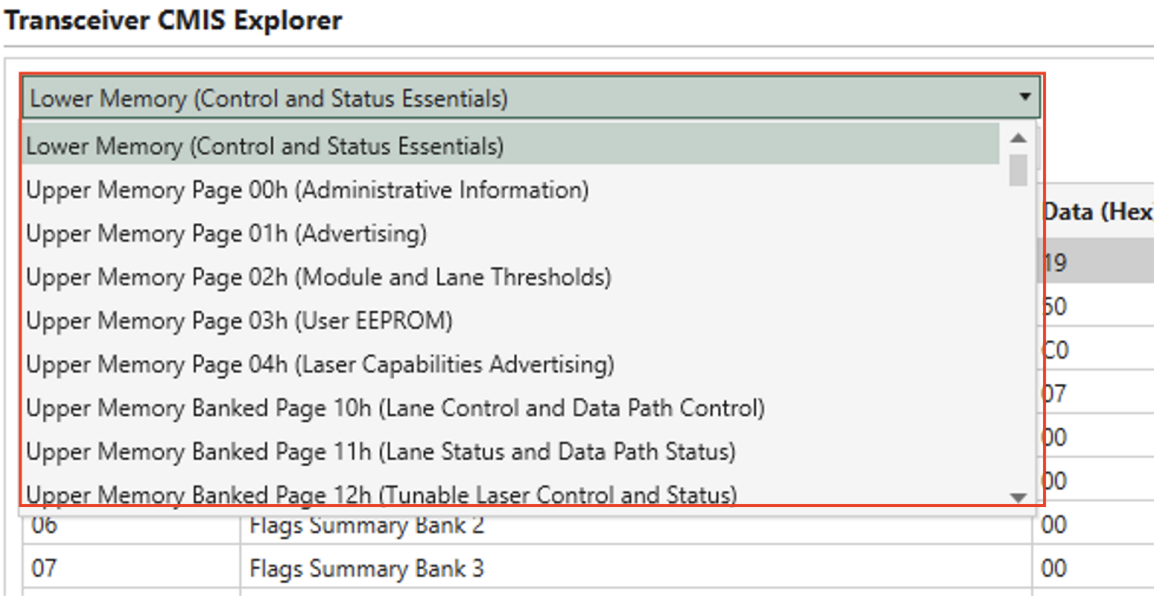


Fig. 5.72: Page Selection

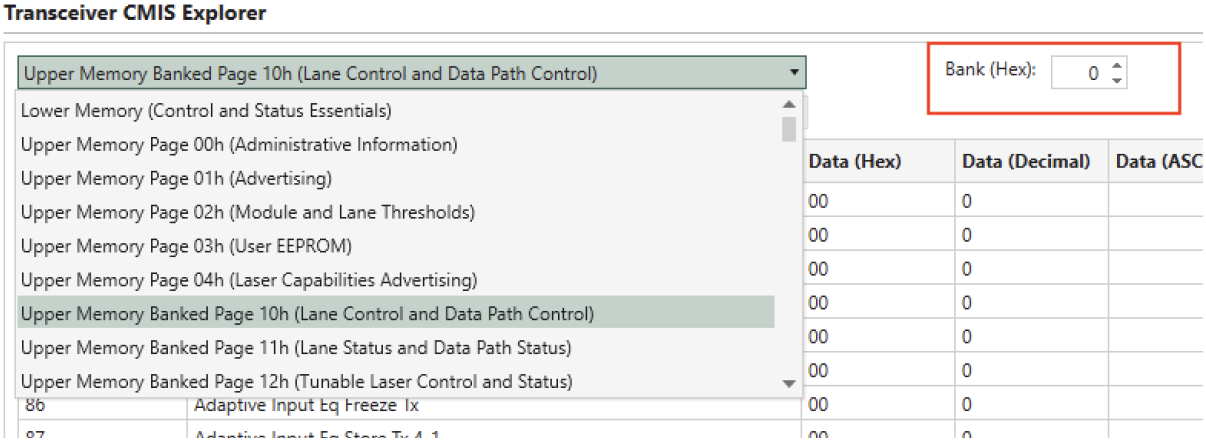


Fig. 5.73: Bank Selection

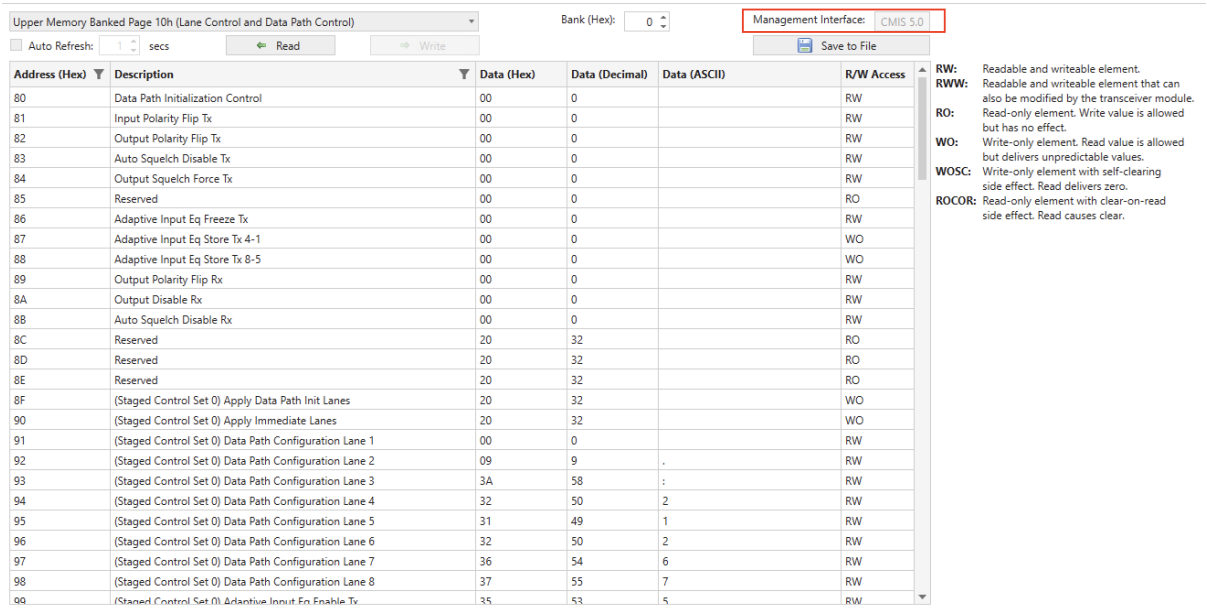


Fig. 5.74: Management Interface

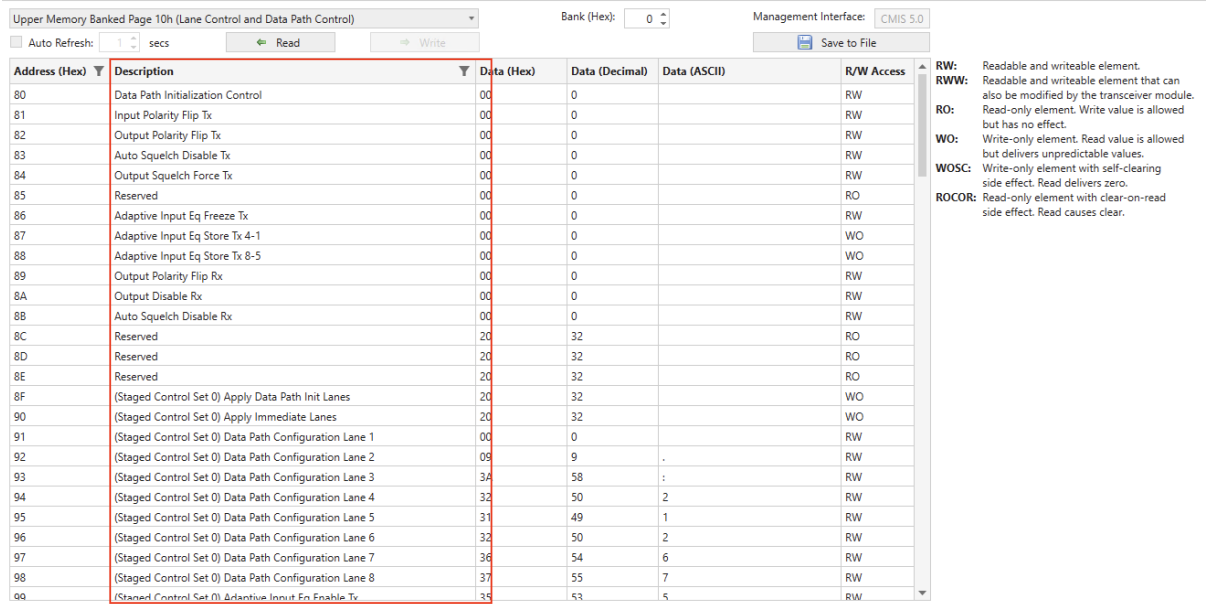
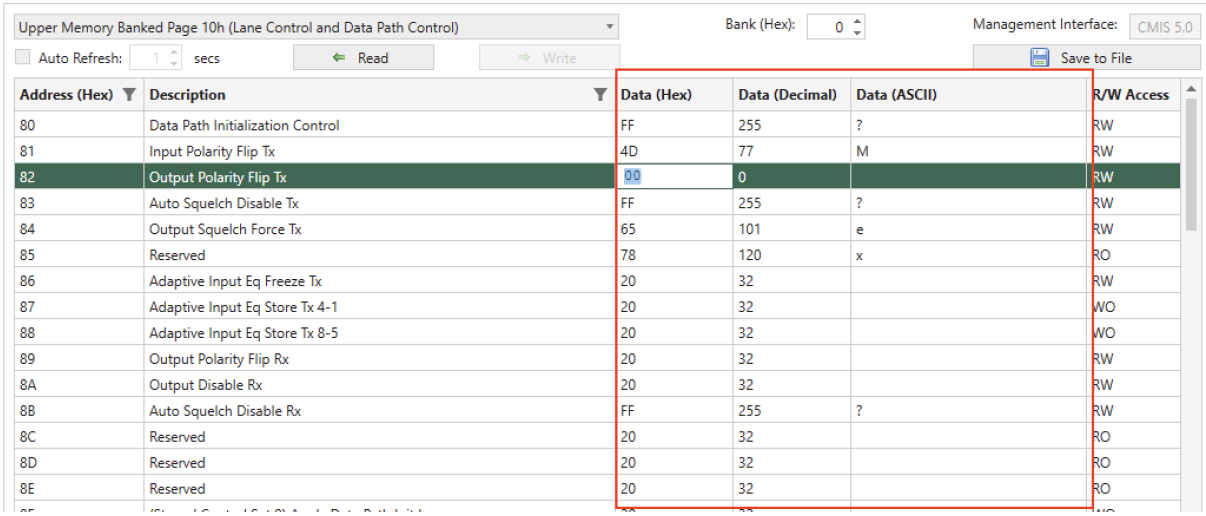


Fig. 5.75: Byte Description

- Data (Decimal): decimal format of the byte value.
- Data (ASCII): ASCII format of the byte value.



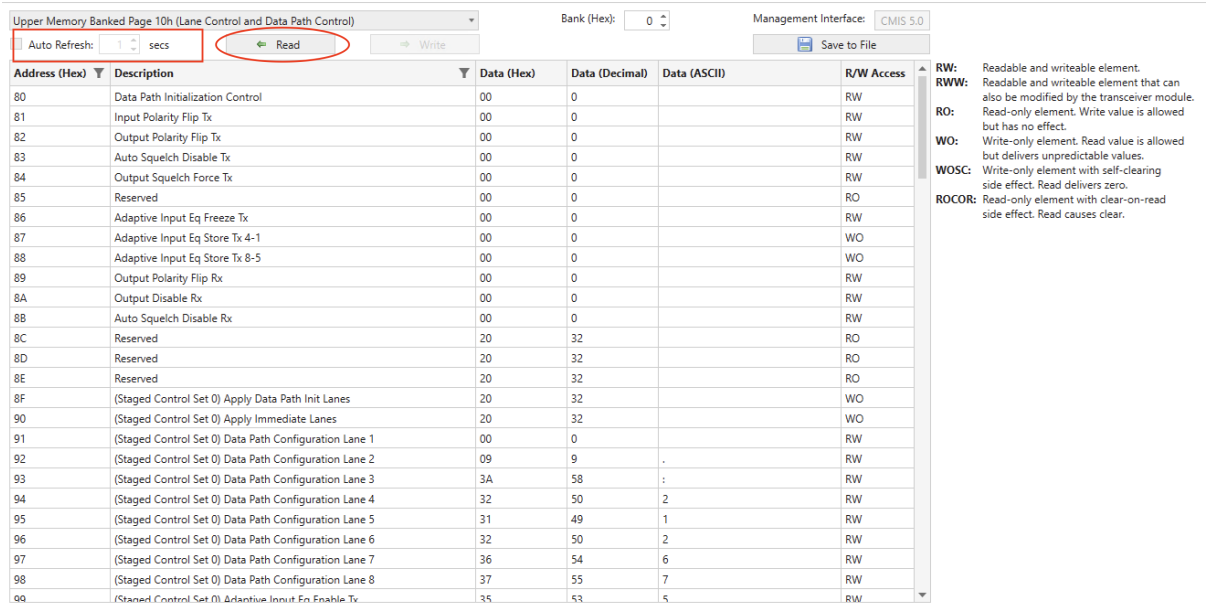
Address (Hex)	Description	Data (Hex)	Data (Decimal)	Data (ASCII)	R/W Access
80	Data Path Initialization Control	FF	255	?	RW
81	Input Polarity Flip Tx	4D	77	M	RW
82	Output Polarity Flip Tx	00	0		RW
83	Auto Squelch Disable Tx	FF	255	?	RW
84	Output Squelch Force Tx	65	101	e	RW
85	Reserved	78	120	x	RO
86	Adaptive Input Eq Freeze Tx	20	32		RW
87	Adaptive Input Eq Store Tx 4-1	20	32		WO
88	Adaptive Input Eq Store Tx 8-5	20	32		WO
89	Output Polarity Flip Rx	20	32		RW
8A	Output Disable Rx	20	32		RW
8B	Auto Squelch Disable Rx	FF	255	?	RW
8C	Reserved	20	32		RO
8D	Reserved	20	32		RO
8E	Reserved	20	32		RO

Fig. 5.76: Byte Value

Read Values

On-demand read and auto-refresh.

- Clicking *Read* button trigger an on-demand read of all the bytes of the selected page/bank.
- If you need to periodically monitor the page/bank, you can enable auto-refresh by enabling *Auto Refresh* with a polling interval in seconds.



Address (Hex)	Description	Data (Hex)	Data (Decimal)	Data (ASCII)	R/W Access
80	Data Path Initialization Control	00	0		RW
81	Input Polarity Flip Tx	00	0		RW
82	Output Polarity Flip Tx	00	0		RW
83	Auto Squelch Disable Tx	00	0		RW
84	Output Squelch Force Tx	00	0		RW
85	Reserved	00	0		RO
86	Adaptive Input Eq Freeze Tx	00	0		RW
87	Adaptive Input Eq Store Tx 4-1	00	0		WO
88	Adaptive Input Eq Store Tx 8-5	00	0		WO
89	Output Polarity Flip Rx	00	0		RW
8A	Output Disable Rx	00	0		RW
8B	Auto Squelch Disable Rx	00	0		RW
8C	Reserved	20	32		RO
8D	Reserved	20	32		RO
8E	Reserved	20	32		RO
8F	(Staged Control Set 0) Apply Data Path Init Lanes	20	32		WO
90	(Staged Control Set 0) Apply Immediate Lanes	20	32		WO
91	(Staged Control Set 0) Data Path Configuration Lane 1	00	0		RW
92	(Staged Control Set 0) Data Path Configuration Lane 2	09	9	.	RW
93	(Staged Control Set 0) Data Path Configuration Lane 3	3A	58	:	RW
94	(Staged Control Set 0) Data Path Configuration Lane 4	32	50	2	RW
95	(Staged Control Set 0) Data Path Configuration Lane 5	31	49	1	RW
96	(Staged Control Set 0) Data Path Configuration Lane 6	32	50	2	RW
97	(Staged Control Set 0) Data Path Configuration Lane 7	36	54	6	RW
98	(Staged Control Set 0) Data Path Configuration Lane 8	37	55	7	RW
99	(Staged Control Set 0) Adaptive Input Eq Enable Tx	35	53	5	RW

Fig. 5.77: Read Values

Write Values

Write values into register bytes. Click on the cells in Data (Hex) or Data (Decimal) to enter the corresponding value you want to write.

- Click the Write button to commit the write changes. This allows you to batch write into the page/bank of the transceiver.

Attention

Writing into RO bytes is allowed but the value will be ignored by the transceiver.

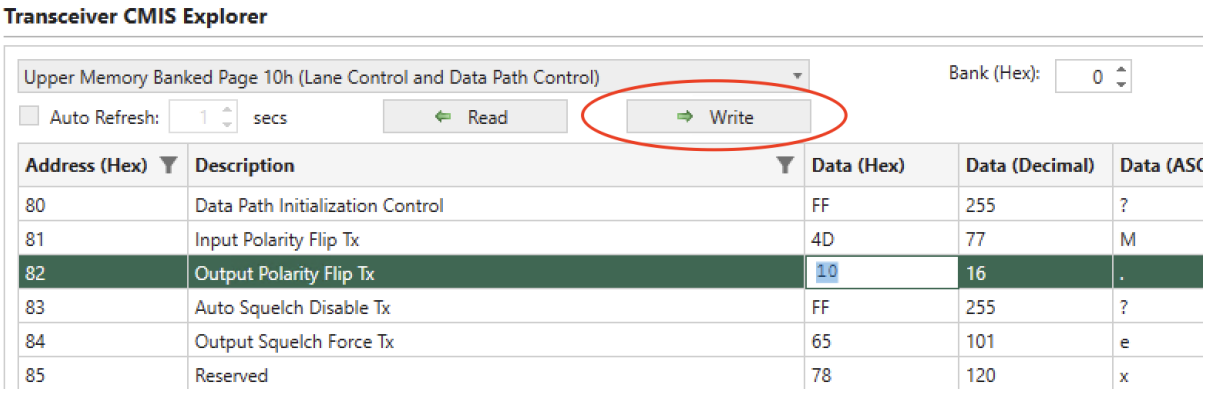


Fig. 5.78: Write Values

R/W Access

The CMIS Explorer shows the read/write access level of each byte.

R/W Access	Description
RW	A readable and writeable element.
RWW	A readable and writeable element that can also be modified by the module.
RO	A read-only element. A WRITE of a value to a read-only element is allowed but has no effect.
WO	A write-only element. A READ from a write-only element is allowed but delivers unpredictable values.
WOSC	A write-only element with self-clearing side effect. A READ from a WO/SC element is allowed and delivers a zero value, except transiently when reading before the module has evaluated and cleared the non-zero bits written.
RO-COR	A read-only element with clear-on-read (clear-after-read) side effect. All bits in a RO/COR Byte are cleared by the module after the Byte value has been read.

Save This Page

Save selected the page/bank into a csv file. Click *Save This Page* button to save the entire page into a csv file, including all information shown in the displayed table.

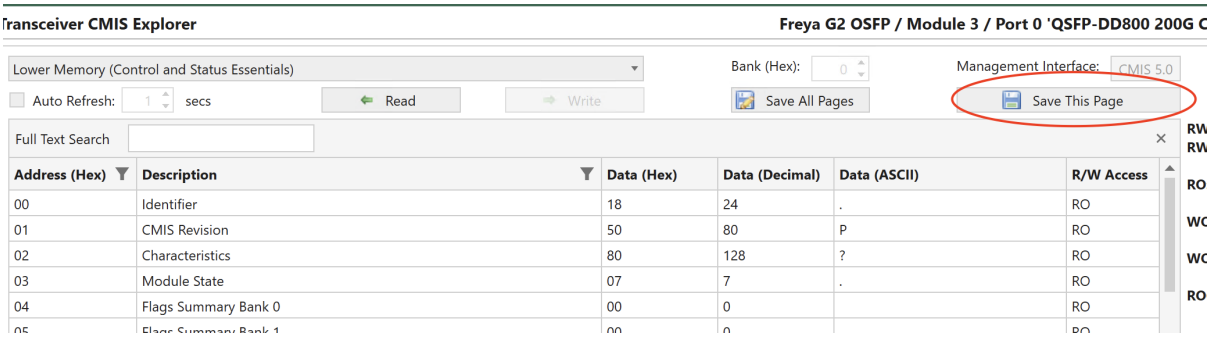


Fig. 5.79: Save This Page

Save All Pages

When you click *Save All Pages* in *Transceiver Features* → *CMIS Explorer*, XenaManager will save the register values of all pages from the drop-down list. Each page's data is saved into a separate CSV file.

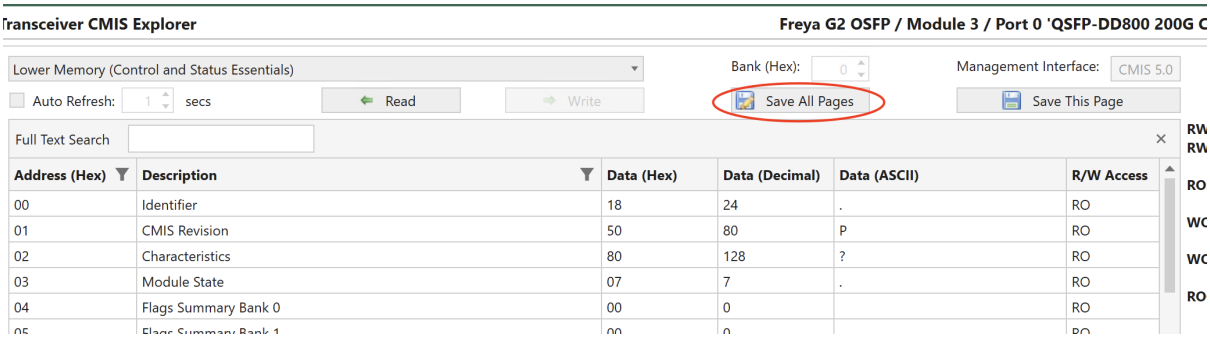


Fig. 5.80: Save All Pages

The saving process takes 2-3 minutes depending on the number of pages in the drop-down list.

Search Bar

The CMIS Explorer table allows you to search the byte description using the search bar. By default the search bar is enabled. To bring back a closed search bar, use `Ctrl-F`.

Supported CMIS Specification

CMIS Explorer supports the following CMIS specifications:

- CMIS 3.0
- CMIS 4.0
- CMIS 5.0

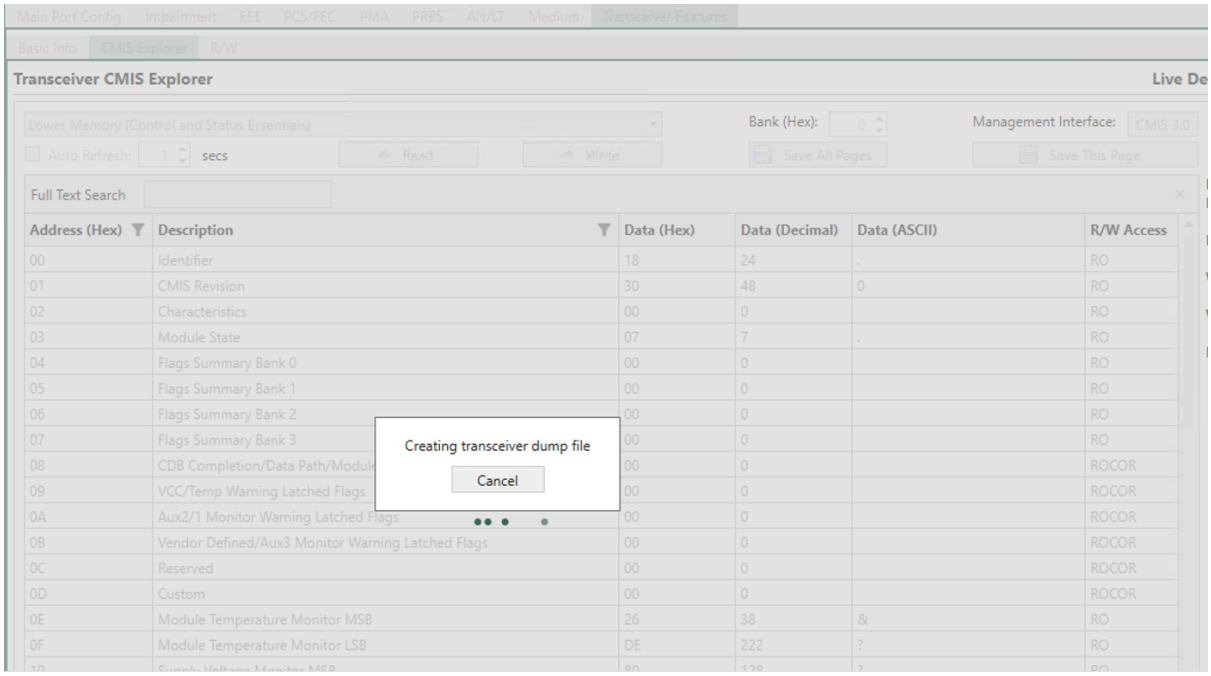


Fig. 5.81: The process of saving all pages takes 2-3 minutes depending on the how many pages are in the drop-down list.

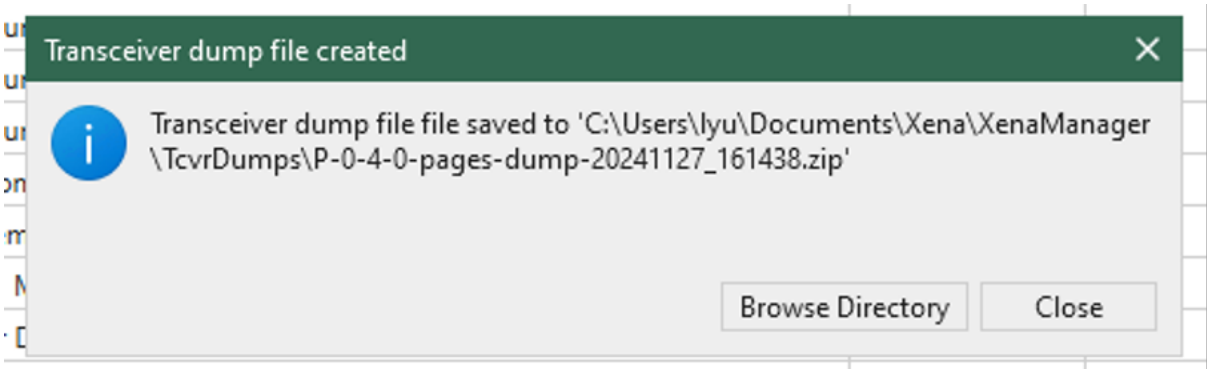


Fig. 5.82: When saving is done, the path to the saved zip file is provided.

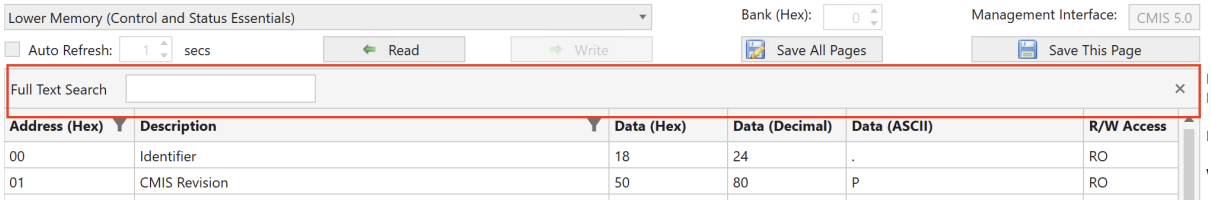


Fig. 5.83: CMIS Explorer Search Bar

- CMIS 5.1
- CMIS 5.2

Important

The GUI only reads the transceiver memory when you click the *CMIS Explorer* tab, or changing page/bank. Clicking on a port doesn't trigger a read operation on the transceiver. This is sometimes useful when you don't want the GUI to interrupt your operations on the transceiver.

R/W

R/W tab panel allows you to directly read and write to the transceiver register.

I2C Access Speed Configuration

The *I2C* RX/TX transceiver access speed on the Z400 Thor 112G PAM4 can now be configured, enabling an increase in the I2C access speed to a maximum of 800KHz. This enhancement facilitates quicker diagnostics and firmware updates among other tasks involving transceivers.

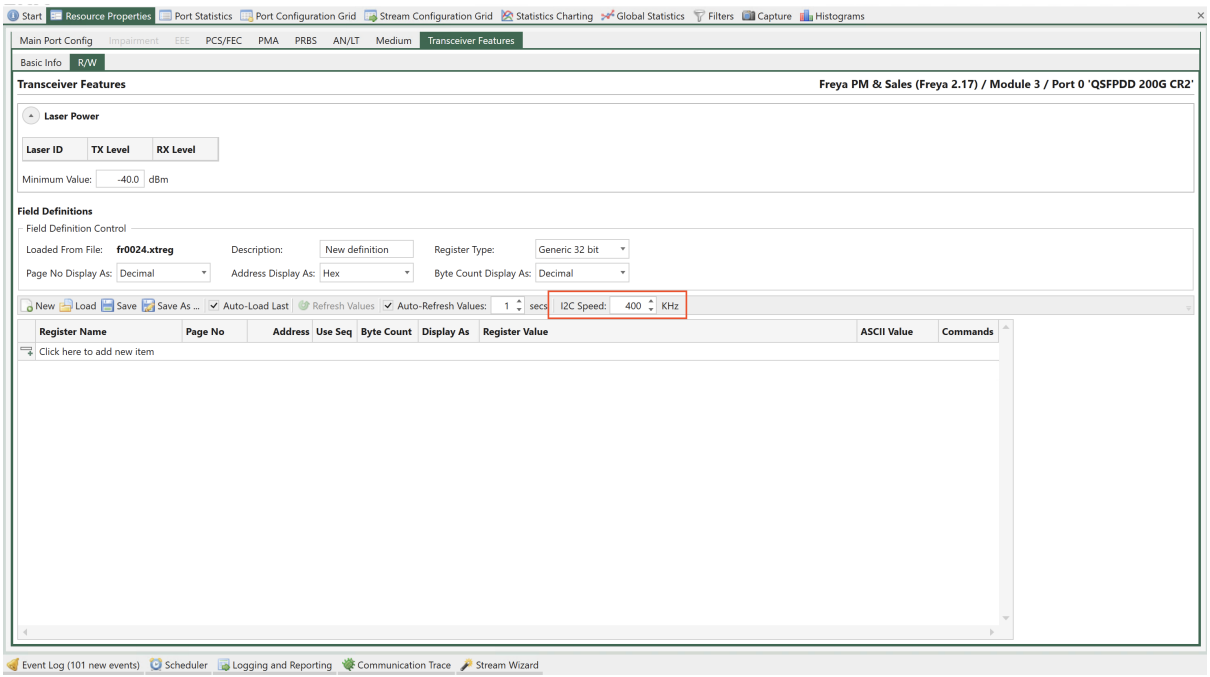


Fig. 5.84: Port Properties - I2C access speed configuration

Laser Power

In addition to register access the The *Transceiver Features* panel can show laser power information if that is supported by the module and the port transceiver.

TX and RX power level will be shown for each laser in the transceiver together with an overall reading, which shows the lowest values for the displayed power levels. You can set a Minimum

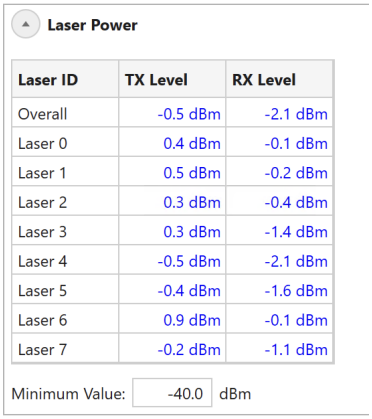


Fig. 5.85: Port Properties - Laser power

value: If any you the level readings are below the Minimum value they will be highlighted with a red color.

Register Access

The *Transceiver Features* panel provides access to the register interface supported by the port transceiver. It is possible to both read and write register values.

All supported registers for a transceiver are shown in Fig. 5.86.

All registers for a given transceiver type is typically organized in sets called pages. Each register within a given page is then identified by an address.

It is possible to read consecutive addresses from a page using sequential access by ticking *Use Seq* and configure *byte count*.

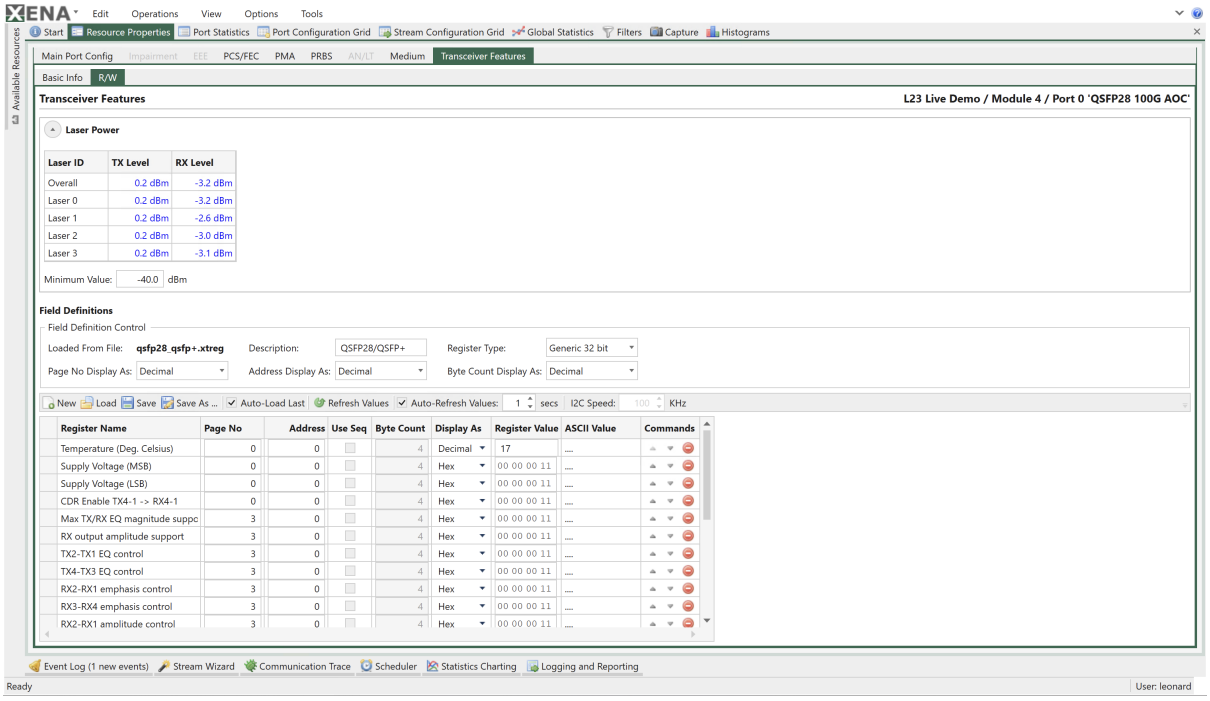


Fig. 5.86: Port Properties - Register access

Reading Register Values

The register values can be read manually by pressing the *Refresh Values* in the panel toolbar. The panel can also refresh the values periodically if the *Auto-Refresh* option is enabled.

The field values will primarily be displayed using the selected field display type (hex, decimal or binary) but it will also be displayed as ASCII characters for convenience.

Writing Register Values

The register values may also be changed by the user by changing the value in the *Register Value* column. The new value is applied when the `Enter` key is pressed.

It is not possible to change the ASCII character value directly.

Register Definitions

Each set of supported register fields for a given transceiver type is defined in a separate file with extension `.xtreg`. The data definition is formatted using JSON notation.

You can load a register definition file by pressing the Load button ion the toolbar. If the *Auto-Load Last* option is selected then the last loaded definition will automatically be loaded the next time XenaManager is started.

Built-in Register Definitions

The XenaManager is shipped with a set of commonly used register definitions, such as the MII register set mentioned above. These files will be kept in the folder `Documents\Xena\XenaManager\TcvrDefs`.

Creating or Modifying Definitions

It is also possible to modify the built-in register definitions or create your own from scratch.

To create a new definition you should press the *New* button in the toolbar. You can also change an existing definition by loading it and saving it under a new name.

Changing Display Options

The top subpanel called *Field Definition Control* defines the overall handling of all register fields in the definition. You can change the display type (hex or decimal) of both address and page number fields. You can also change the bit width (16 or 32 bit) of the register addresses.

Adding or Removing Fields

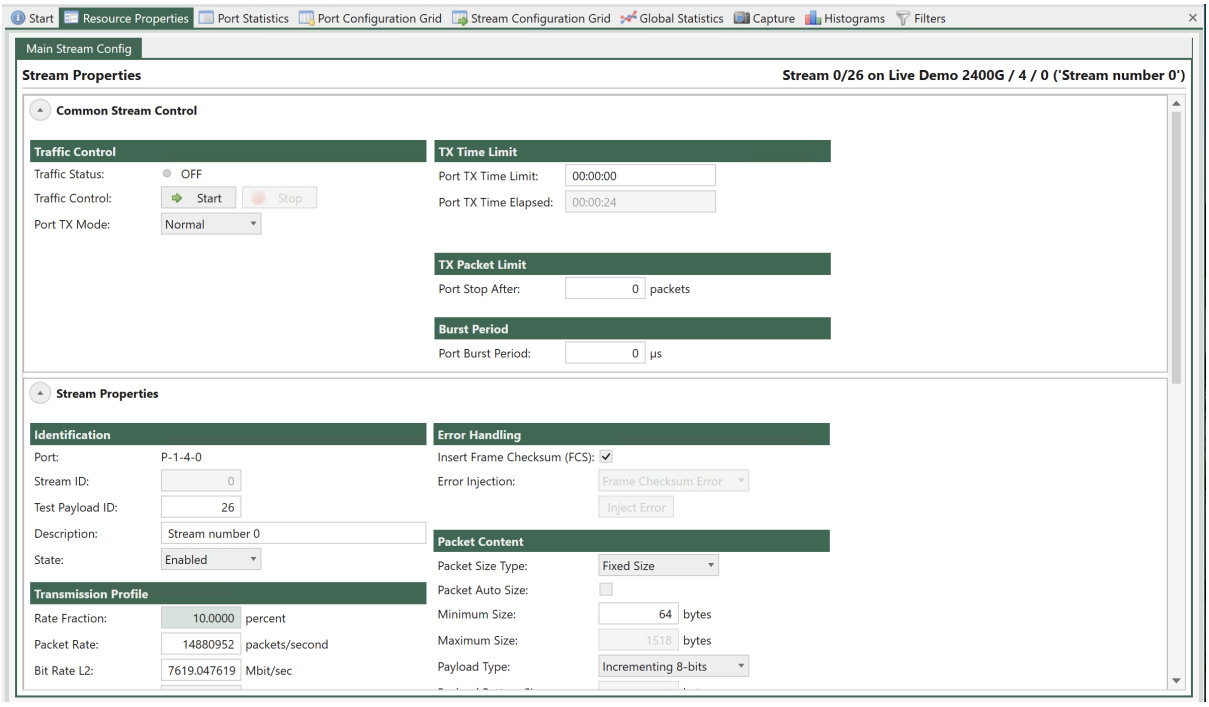
You can add a new register field by pressing the area at the bottom of the field definition table labeled *Click here to add a new item*. The new item will be added to the bottom of the table.

You can reorder the field by using the up- and down-arrows in the *Commands* column.

To remove a field press the *Delete* icon in the *Commands* column.

5.3.5 Stream Properties

This section describes the available stream properties for XenaManager.



Common Stream Control

This area contains port-level controls that affect all streams on that port. They are shown on the streams property page for the sake of convenience.

Traffic Control

Table 5.34: Traffic Control

Property	Explanation
Traffic Status	The current traffic status for the port (OFF: traffic is off, ON: traffic is on)
Traffic Control	This button enables you to either start or stop traffic on the port
Port TX Mode	This property determines the scheduling mode for outgoing traffic from the port, i.e. how multiple logical streams are merged onto one physical port. Refer to CLI command - P_TXMODE for further information.
Port Stop After	Stop port transmission after the specified number of packets are sent
Port Burst Period	Time in micro seconds from start of sending a group of bursts till start of sending next group of bursts

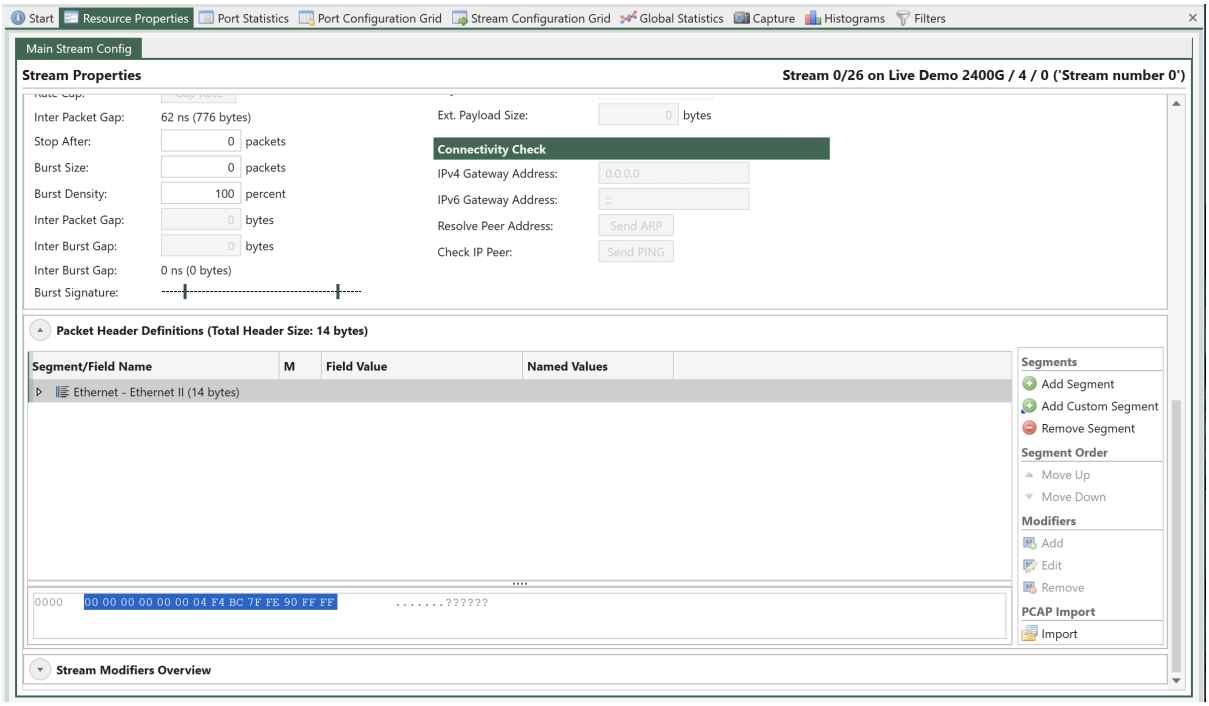


Fig. 5.87: Stream Properties

TX Time Limit

Table 5.35: TX Time Limit

Property	Explanation
Port TX Time Limit	The maximum time the port should transmit
Port TX Time Elapsed	The amount of time the port has been transmitting.

Main Properties

This area contain all stream-level configuration properties, except those related to protocol header and modifier definitions.

Identification

Table 5.36: Identification

Property	Explanation
Port	The parent port name
Stream ID	The unique stream ID
Test Payload ID	The test payload ID (*TID*) carried in the Xena test payload area. This field can be empty if no TID value is needed.
Description	A user-modifiable description label for the stream
State	The stream enable state.

Transmission Profile

Table 5.37: Transmission Profile

Property	Explanation
Rate Fraction	The stream traffic rate expressed as a percentage of the effective rate for the port.
Packet Rate	The stream traffic rate expressed as packets per second.
Bit Rate L2	The stream traffic rate expressed as bits per second seen on Layer 2.
Bit Rate L1	The stream traffic rate expressed as bits per second seen on Layer 1.
Rate Cap	This command can be used to cap the rate for disabled streams. The button will only be enabled if the sum of the defined stream bandwidth actually exceeds the available port bandwidth.
Inter Packet Gap	The calculated mean inter-packet gap with the current TX profile settings. This denotes the space between the end of the preceding packet and the start of the following packet.
Seq.Packets	The number of sequential packets sent before switching to the next stream (packets). This property is only configurable when the Port TX Mode is set to Sequential.
Stop After	Stop stream transmission after the specified number of packets are sent. This value can be empty or zero, which means that the stream will continue to transmit until traffic is stopped at the port level. The Stop after function is not an option when Port TX Mode is set to *Sequential*.
Burst Size	The number of packets in each burst (packets). Valid range 0-500 • When *Port TX Mode* is *Burst*, range is 0-10000.
Burst Density	The density of the burst expressed as a percentage value between 0 and 100. • A value of 100 means that the packets are packed tightly together, only spaced by the minimum inter-frame gap. • A value of 0 means even, non-bursty, spacing. • The exact spacing achieved depends on the other enabled streams of the port. • Not used when *Port TX Mode* is *Burst*.
Inter Packet Gap	Gap between packets in a burst. • Only used when *Port TX Mode* is *Burst*
Inter Burst Gap	Gap between the burst from this stream and the burst from the next stream. • Only used when *Port TX Mode* is *Burst*
Inter Burst Gap	The calculated inter-burst gap with the current burst settings. This denotes the space between the end of the last packet in the preceding burst and the start of the first packet in the following burst.
Burst Signature	A graphical depiction of the current burst settings

Refer to the Fig. 5.88 below for an illustration of the burst mode.



Fig. 5.88: Port TX Burst Mode

Error Handling

Table 5.38: Transmission Profile

Property	Explanation
Insert Frame Checksum (FCS)	Control if a valid frame checksum is added to the stream packets. Default is enabled.
Error Injection	Specifies the type of error that is injected into the traffic stream. The following types of errors can be specified: • Frame Checksum Error: Injects an Ethernet FCS error. • Sequence Error: Injects a sequence error in the Xena Test Payload. This will result in a lost packet being counted. Only applicable if the stream has a TID. • Misordering Error: Injects a misordering error in the Xena Test Payload. Only applicable if the stream has a TID. • Payload Integrity Error: Injects an error by changing a byte in the payload. • Test Payload Error: Injects an error in the Xena Test Payload sequence forcing the packet to not being recognized at the receiving port as a Xena test packet. It will then be counted as a no-test-payload packet. It is only possible to inject errors on a stream if traffic is active on the parent port.
Inject Error	Inject a single error of the specified type into the traffic stream. This option is only enabled when traffic is active on the parent port.

Packet Content

Table 5.39: Packet Content

Property	Explanation
Packet Size Type	The size distribution of the packets transmitted for the stream **Important** Read how to generate mixed packet size in *MIX Weights*.
Packet Auto Size	If selected XenaManager minimum packet size is auto adjusted so it equals the configured packet size in the stream.
Minimum Size	The lower limit of the packet size (if required by the size type)
Maximum Size	The upper limit of the packet size (if required by the size type)
Payload Type	The type of payload data used in the Xena payload section. • Incrementing 8-bits - means *00 01 02 03 04 05 ...* Provides built in payload integrity check for payload. • PRBS-31 - provides Pseudo Random Bit Sequence of `2^31-1` pattern. Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports. • Random - provides Random bit Sequence pattern. Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports. • Pattern - you can set your own custom pattern. Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports. • Decrementing 8-bit: means *FF FE FD FC FB FA ...* Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports. • Incrementing 16-bit: means *00 00 00 01 00 02 00 03 00 04 00 05 ...* Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports. • Decrementing 16-bit: means *FF FF FF FE FF FD FF FC FF FB FF FA...* Payload integrity error detection requires non-zero *Payload Checksum Offset* in port properties of both TX and RX ports. **Important** When using Incrementing 16-bit or Decrementing 16-bit, you need to check the option *Payload Start From 0* to have the payload *00 00 00 01 00 02 00 03 00 04 00 05 ...* or *FF FF FF FE FF FD FF FC FF FB FF FA....*
Payload Pattern	The pattern of bytes to be repeated when the type is set to *Pattern*.
Payload Pattern Size	When choosing *Pattern* as Payload Type, it is possible to define the size of the repeated payload pattern part.
Ext. Payload Size	The size of the extended payload if this option has been enabled on the parent port. Refer to Freely Programmable Test Packets (Custom Data Fields) for details.

Connectivity Check

Table 5.40: Connectivity Check

Property	Explanation
IPv4 Gateway Address	The IPv4 gateway address used to resolve the DMAC address for the stream. Only valid if the stream contains an IPv4 protocol segment.
IPv6 Gateway Address	The IPv6 gateway address used to resolve the DMAC address for the stream. Only valid if the stream contains an IPv6 protocol segment.
Resolve Peer Address	Send an *ARP* or *NDP* request to the peer in order to resolve the MAC address. Only valid of an IPv4 or IPv6 segment has been defined with a valid Dest. IP address is defined.
Check IP Peer	Send a PING request to the peer in order to check the connectivity. Only valid of an IPv4 or IPv6 segment has been defined with a valid Dest. IP address is defined.

The Xena tester will set the Target IP Address in any ARP/NDP request sent from a Xena test port to a value in the following prioritized order:

1. Stream gateway IP address for the IP version used by the stream if defined.
2. Port gateway IP address for the IP version used by the stream if defined and stream Dest IP Address is not in same subnet as the port gateway (the legacy method).
3. Stream Dest. IP Address

Important

You can create multiple IPv4/IPv6 streams on a port to simulate traffic from multiple IPv4/IPv6 client endpoints behind a physical port. Each endpoint can have different source MAC addresses and source IP addresses. At this point, the MAC address of the physical test port become unimportant for testing because it is the "virtual clients" behind that matter. However, to make each "virtual client" successfully resolve the destination's MAC address (either the gateway or the destinations), you must configure (or auto-generate) the port's *ARP/NDP Address Tables* accordingly by clicking the *Generate* in *Main Port Config*

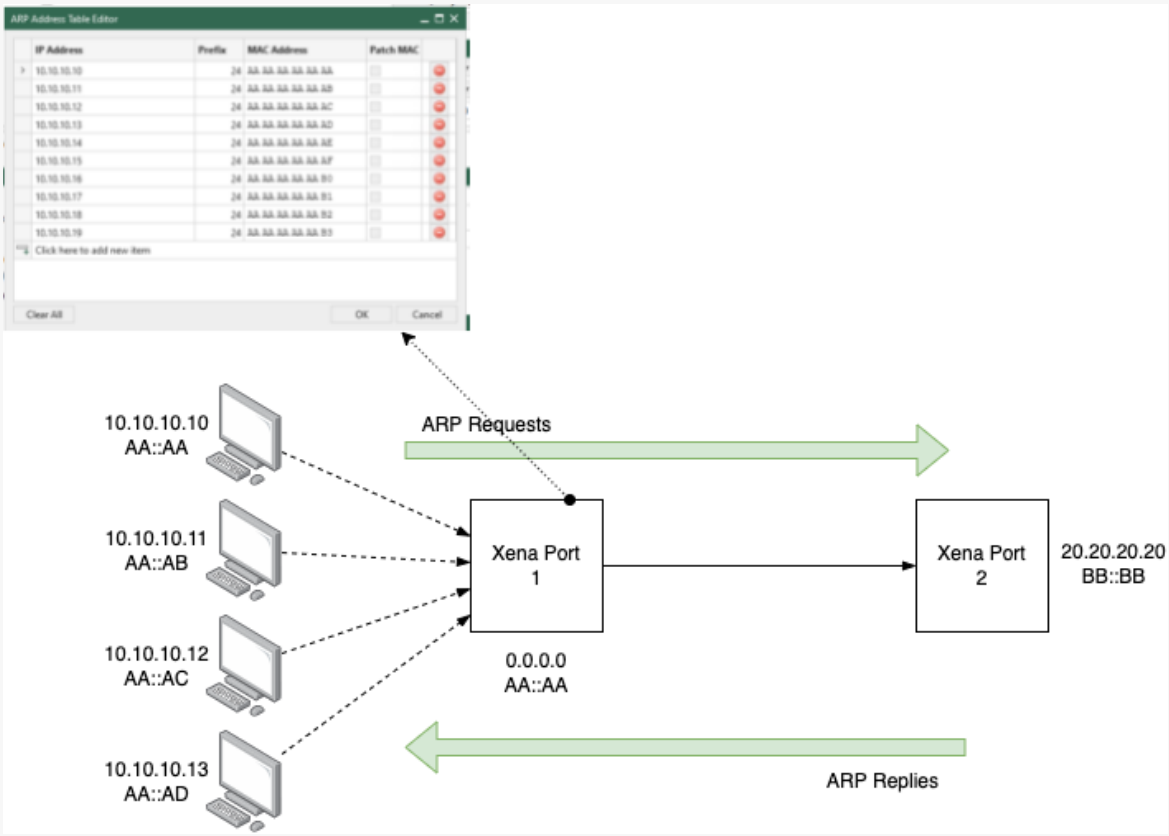


Fig. 5.89: Generate ARP/NDP table for the port in order to successfully do ARP/NDP resolution for streams

Packet Header Definition

This section describes the Packet Header editor used by the stream configuration pages. The editor controls the definition the protocol header segments and the associated field modifiers.

Overview

The defined protocol segments are shown in a Wireshark-like tree structure. All fields for a given segment header are shown as child rows under the segment row. Any modifiers defined on fields are shown as child rows under the field row.

Tree Columns

The treeview contains these columns:

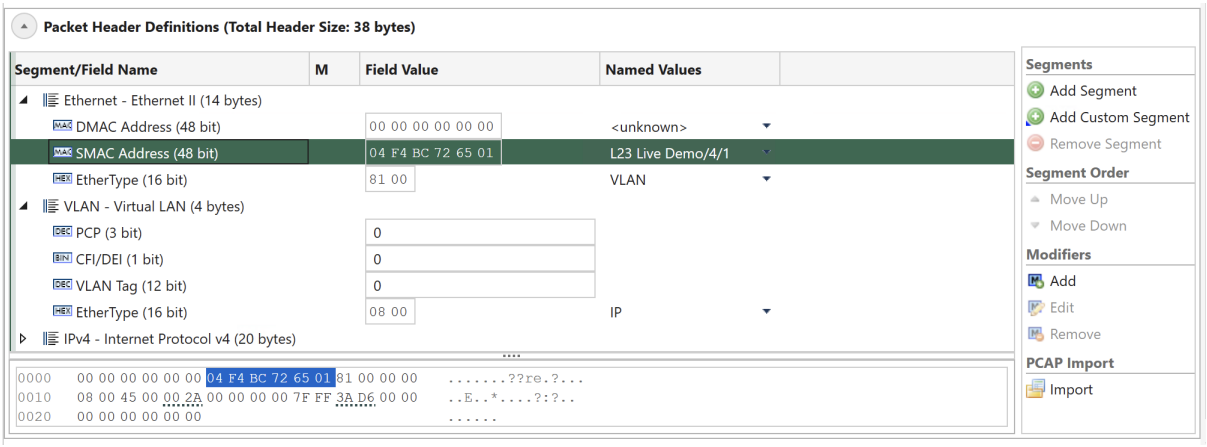


Fig. 5.90: Protocol segment profile editor

Table 5.41: Packet Header Treeview

Column	Explanation
Segment/Field Name	The name of the segment or field
M	Contains an icon that indicate if a collapsed segment or field row contains one or more modifiers.
Field Value	The actual field value in the common value representation for that field.
Named Values	Certain fields may get their value from a list of standardized or well-known named values. Instead of entering the value directly you can select the value from the dropdown list in this column.

Field Type Icons

Each field row is prefixed with an icon indicating the value representation for the field. The following representations are used:

- DEC: Decimal representation
- BIN: Binary representation
- HEX:Hexadecimal representation
- MAC: A MAC address
- IP4: An IPv4 address
- IP6: An IPv6 address

Raw Hex Editor

At the bottom of the treeview you will find a raw hex editor which allow you to inspect and optionally modify the raw hex data for the segment definitions. Any changes you make in the raw editor will be written to the chassis and the associated field value controls will be updated accordingly.

When you select a segment or a field the relevant parts in the hex editor will be highlighted. The hex editor will also underline the areas affected by any defined modifier.

The left part of the hex editor contains an address list and the right part show the current raw data decoded as printable ASCII.

Segment Headers

Adding a Segment Header

To add a new segment header to the existing definition press the *Add Segment* button in the in the command panel to the right. You will now be presented with a list of known protocol types in alphabetical order. You can select one or more types using the standard Windows `Ctrl+Click` or `Shift+Click` operations. When you are done press the OK button.

XenaManager will check if the total size of the configured segments exceeds *maximum header size* and *minimum packet length*, and if that is the case user will be presented with a message, which will also provide the option to increase "*maximum header size* and *minimum packet length*.

> **Note**
>
> If a segment is removed, the *maximum header size* and *minimum packet length* are not adjusted.

Moving a Segment Header

With the exception of the first Ethernet segment you can move segment headers up or down in the list after you have added them. Select the segment you want to move and use either the Move Up or the *Move Down* button in the command panel to the right.

Any modifiers you have defined in segments affected by the move will be moved automatically.

Removing a Segment Header

Select the segment you want to remove and use the *Remove Segment* button in the command panel to the right.

Any modifiers you have defined in the removed segment will also be removed automatically.

Import From PCAP File

Instead of manually building the segment headers you can instead import the structure from a PCAP file. Note that this operation will replace any segments you may have added manually.

To import the segment structure from a PCAP file simply press the Import button in the command panel to the right and select a PCAP file on disk which contain the packet you want to import. The packets in the PCAP file will now be decoded and a list of the found packets will be shown. You should then select the packet you want to import and press the OK button.

The import function will use any trailing data in the packet as one or more custom data segments.

> **Note**
>
> You can download and try these `PCAP sample files`

Custom Segment Headers

Custom Segments resemblances to Raw Segments, but they offer the ability to customize the header structure and assign custom names to fields. This feature, known as the `Custom Segment` feature, lets you craft your own protocol segment, typically represented as a JSON file. This customization significantly enhances usability and tailors the segment to your specific needs.

You can define your own custom segment as the steps described below:

1. The new Custom segment files should be saved with `.xdef` extension.
2. The content inside these `.xdef` files must use the JSON formatting.
3. The `.xdef` files must be placed in `C:\Users\<username>\Documents\Xena\XenaManager\SegmentDefs`
4. Click *Add Custom Segment*, and you will see the added segment.

> **Note**
>
> You can download and try `these custom segment definitions`.

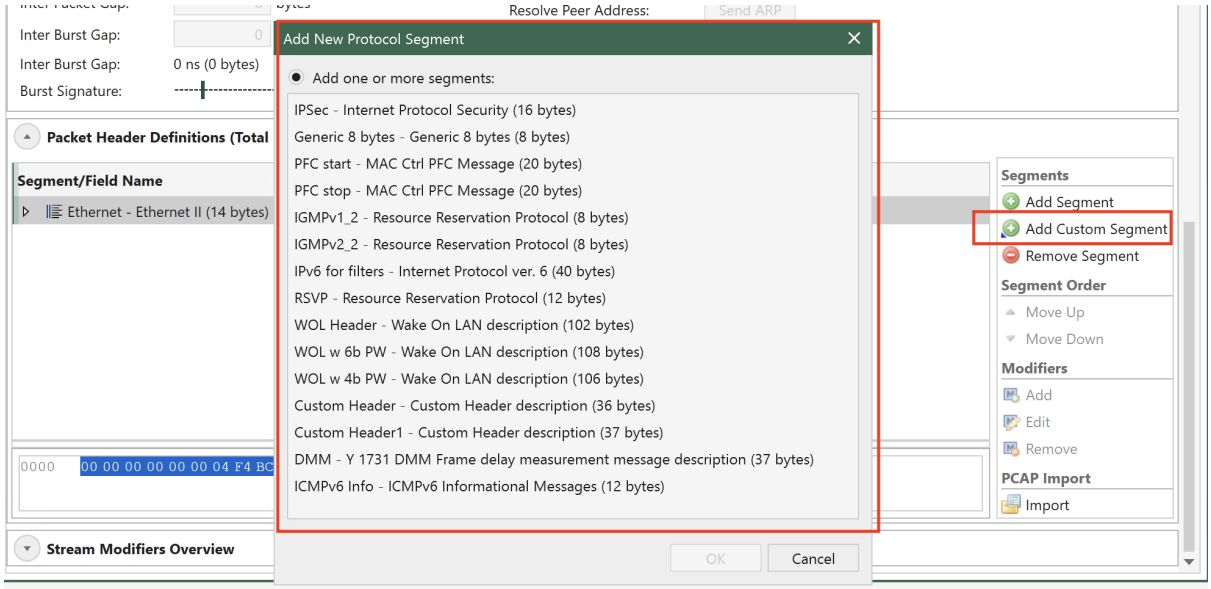


Fig. 5.91: Add Custom Segments

The JSON format of `.xdef` file:

```
{
  "Name": "Custom Header",
  "Description": "Custom Header description",
  "SegmentType": 140,                //=> accepted values range: [128␣
↪to 191]. It's a unique key, so if more than one file with same␣
↪SegmentType are detected, it'll load only the first detected one
  "ProtocolFields": [                //=> array of fields for the␣
↪Custom Segment, you can define as many as you need but always␣
↪following JSON formatting rules
    {
      "Name": "Custom field 1",
      "BitLength": 8,
      "DisplayType": "Decimal",      //=> accepted values: [Decimal,␣
↪Binary, Hex, IpV4Address, IpV6Address, MacAddress, DisplayString]
      "DefaultValue": "0"   //=> accepted values can be defined in␣
↪Decimal or Hex format. Can be separated by commas with no space (,)␣
↪and each value corresponds to one byte.
    },
    {
      "Name": "Custom field 2",
      "BitLength": 4,
      "DisplayType": "Binary",
      "DefaultValue": "11"  //=> decimal “11” which will be displayed␣
↪binary in UI as “1011”. Maximum value in this field would be “1111”␣
↪(4 bits length / “15” decimal)
    },
    {
      "Name": "Custom field 3",
      "BitLength": 2,
      "DisplayType": "Hex",
      "DefaultValue": "0xF"          //=> XenaManager will skip this␣
↪value and default it to 3 as 0xF is greater than max. possible value␣
↪for a 2-bit field definition (range 0-3 decimal)
    },
    {
      "Name": "Custom field 4",
      "BitLength": 16,
      "DisplayType": "Hex",
      "DefaultValue": "0xFF, 0xFF"  //=> field defined as 16 bits, so␣
↪we define 2-byte default values separated by comma and in Hex format,
↪ but we can also use Decimal format
    },
    {
      "Name": "Custom field 5",
      "BitLength": 48,
      "DisplayType": "DisplayString",
      "DefaultValue": "0x48,0x65,0x6c,0x6c,0x6f,0x00" //=> field␣
```

(continues on next page)

(continued from previous page)

```
↪defined as 6 bytes/characters, so to define default value Decimal or␣
↪Hex format of ASCII values can be used and all 6 bytes needs to be␣
↪defined
    } ,
    {
      "Name": "MAC Address",
      "BitLength": 48,              //=> MAC address field's length is␣
↪always 48 / 6 bytes
      "DisplayType": "MacAddress",
      "DefaultValue": "0,0,0,0,0,0"
    },
    {
      "Name": "IPv4 Addr",
      "BitLength": 32,      //=> IPv4 address field's length is always␣
↪32 / 4 bytes
      "DisplayType": "IpV4Address",
      "DefaultValue": "0,0,0,0"
    },
    {
      "Name": "IPv6 Addr",
      "BitLength": 128,     //=> IPv6 address field's length is always␣
↪128 / 16 bytes
      "DisplayType": "IpV6Address",
      "DefaultValue": "0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"
    }
  ]
}
```

Setting Field Values

You can change any field value by using the associated edit control in the *Field Value* column. For those fields that have a set of well-known values associated you can also choose one of these values from the dropdown list in the *Named Values* column.

Finally you may edit the content of the fields directly in the hex editor panel if you are so inclined.

Next-Protocol Type Fields

Certain protocol segment types (such as Ethernet, *VLAN* and IP) contain fields that indicate the type of the next segment. The segment editor will attempt to set such fields to a correct value when you add, remove or move segments. You can however override the value afterwards if necessary.

Modifiers

Modifiers are specified directly on the field hey are supposed to modify.

Adding Modifier

To add a modifier select the field you want to modify and click the *Add* button in the *Modifiers* section in the command panel to the right. You will now be presented with a window allowing you to specify the properties for the modifier. Press the OK button when you are done.

The new modifier will be shown as a child row under the field row. The value in the *Field* Value column is a read-only string representation of the modifier settings.

Editing Modifier

To edit the properties of an existing modifier select the modifier and click the *Edit* button in the *Modifiers* section in the command panel to the right.

Removing Modifier

To remove a modifier select the modifier and click the *Remove* button in the *Modifiers* section in the command panel to the right.

Modifier Endianness

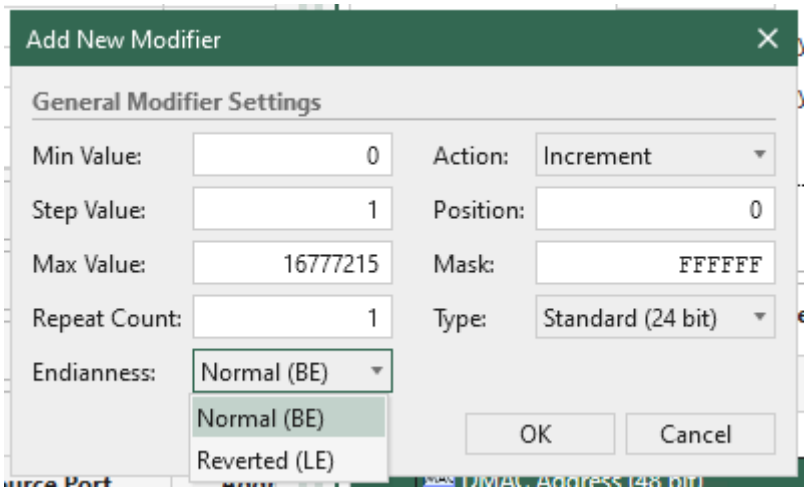


Fig. 5.92: Configure modifier endianness

The endianness of a modifier is specified in the modifier editor. The endianness is used to determine how the modifier value is applied to the field value. The endianness can be set to either *Normal (BE)* or *Reverted (LE)*.

- *Normal (BE)* means that the modifier inc/dec the field value from the LSB (Least Significant Bit) to the MSB (Most Significant Bit).
- *Reverted (LE)* means that the modifier inc/dec the field value from the MSB to the LSB, as shown in the illustration below.

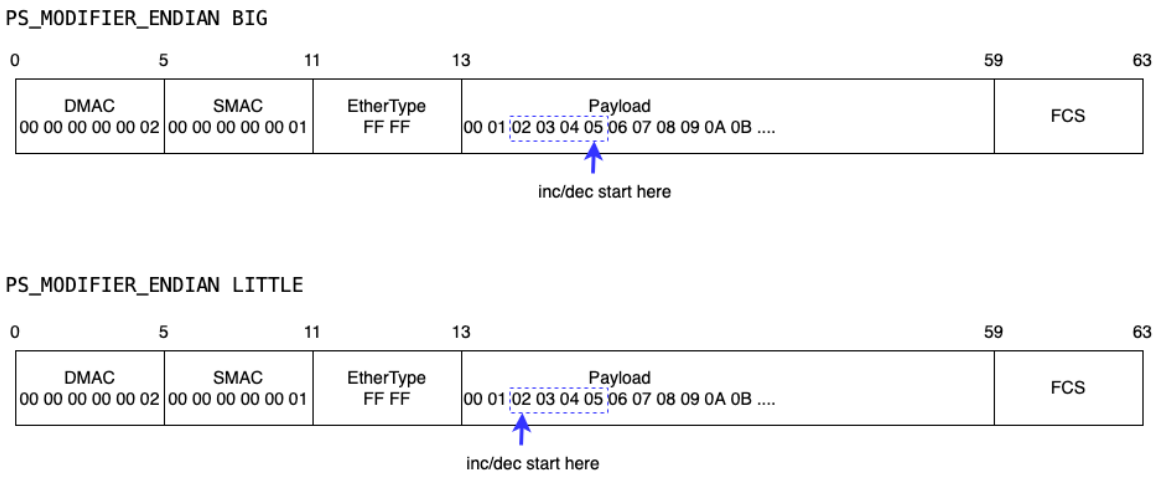


Fig. 5.93: Modifier endianness illustration

Custom Data Field (CDF)

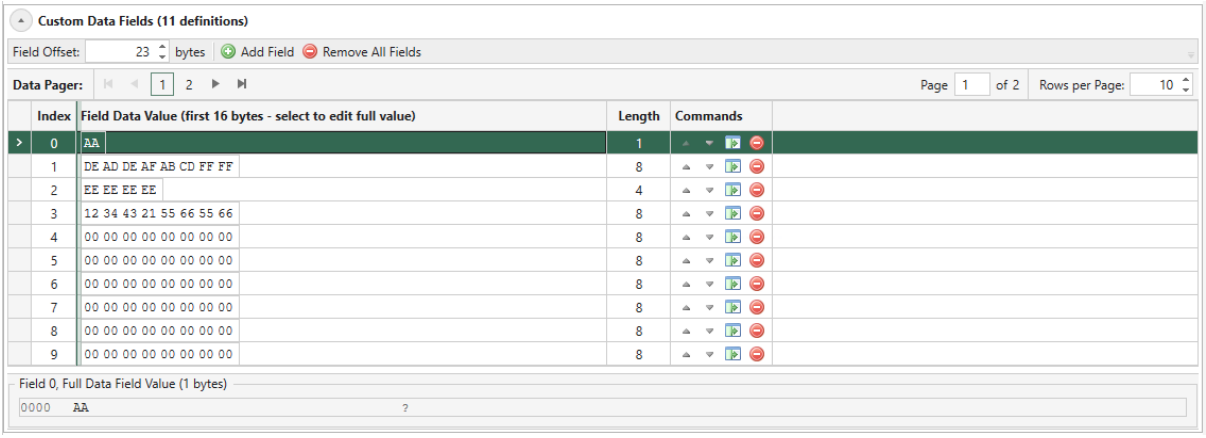


Fig. 5.94: Custom Data Field Editor Overview

Custom Data Field allows you to modify any consecutive bytes in a packet, ranging from one byte to the entire packet. The feature requires that the *Payload Mode* on the parent port has been set to *Custom Data Field*. This enables the feature for all streams on this port.

See also

Read more in Freely Programmable Test Packets (Custom Data Fields) for details.

CDF Field Offset

The CDF offset for the stream is the location in the stream data packets where the various CDF data will be placed. All fields for a given stream uses the same offset value. The default value is 0, which means that the CDF data will be placed at the very start of the packet, thus overwriting the packet protocol headers. If you want the CDF data to start immediately after the end of the packet protocol headers you will have to set the CDF field offset manually.

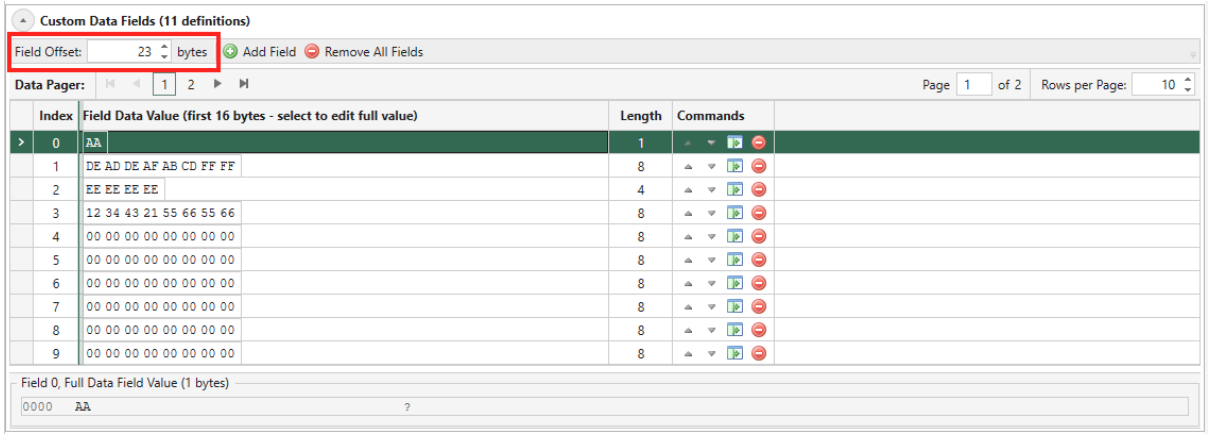


Fig. 5.95: CDF Field Offset

Add/Remove CDF Field

Click the *Add Field* to start adding a CDF field. Then enter the size of the CDF (by default 8 bytes). This will append a row in the CDF editor table with default values of *00*.

You can add more than one CDF fields on a stream. If you have more than one CDF fields on a stream, the stream will generate packet using the CDF fields in a round-robin fashion, which means the first packet uses CDF #0, the second packet uses CDF #1, and so on.

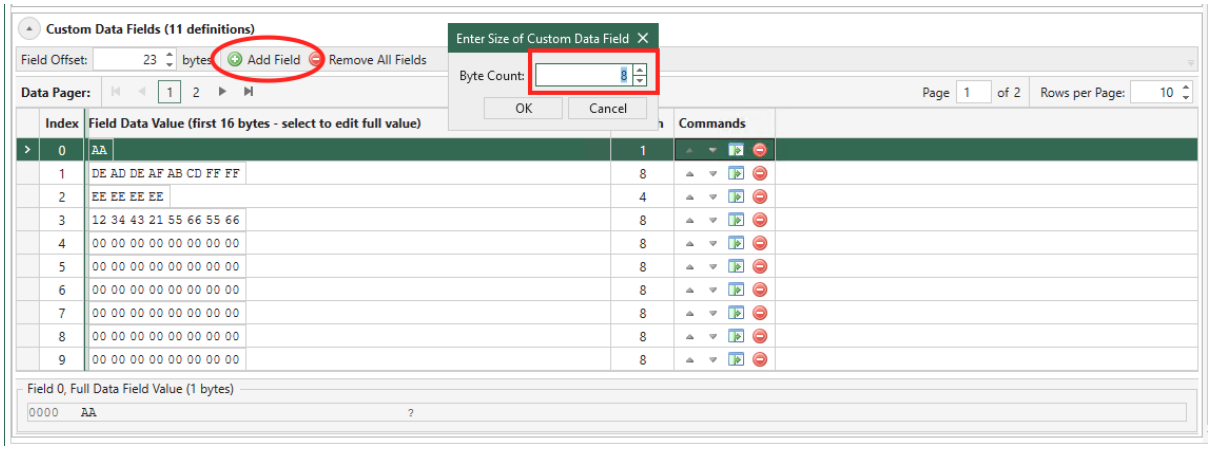


Fig. 5.96: Add CDF Field

Click the *Remove* button on a CDF to remove it. To remove all CDFs, click the button *Remove All Fields*.

Edit CDF Field

You can change the CDF index by clicking arrow buttons in *Commands* as shown above.

You can change the CDF size by clicking third button from the left in *Commands* as shown above. It is possible to define fields with different data lengths for each stream. If the length of a data field exceeds (packet length - CDF offset) defined for the stream the field data will be truncated when transmitted.

To edit the CDF data value, you should use the hex string edit cell.

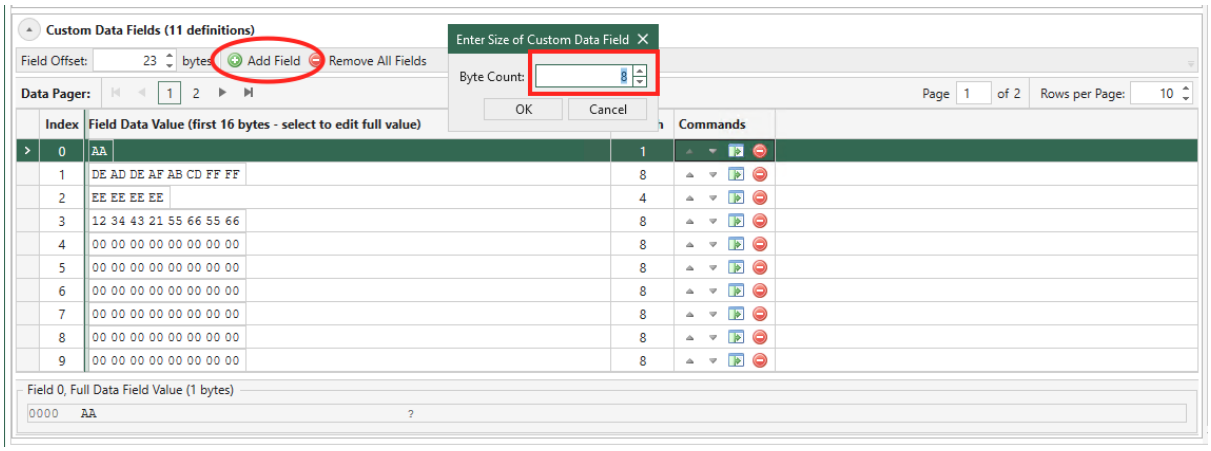


Fig. 5.97: Remove CDF Field

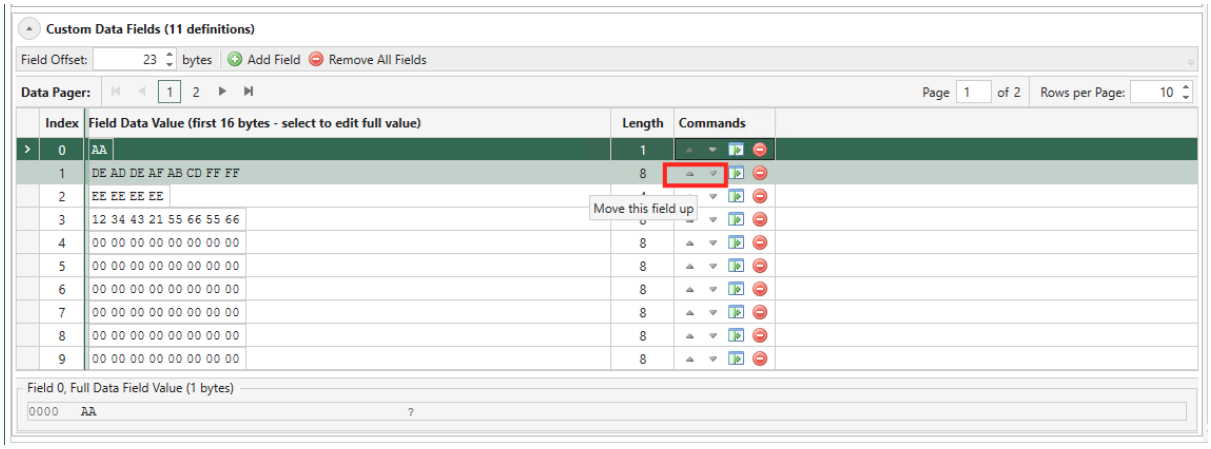


Fig. 5.98: Move CDF Field

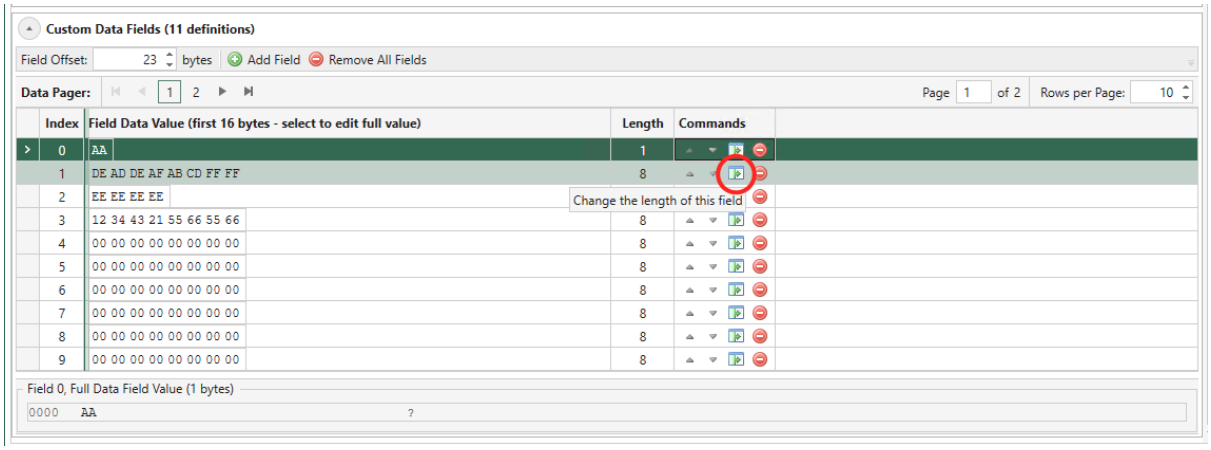


Fig. 5.99: Change Size

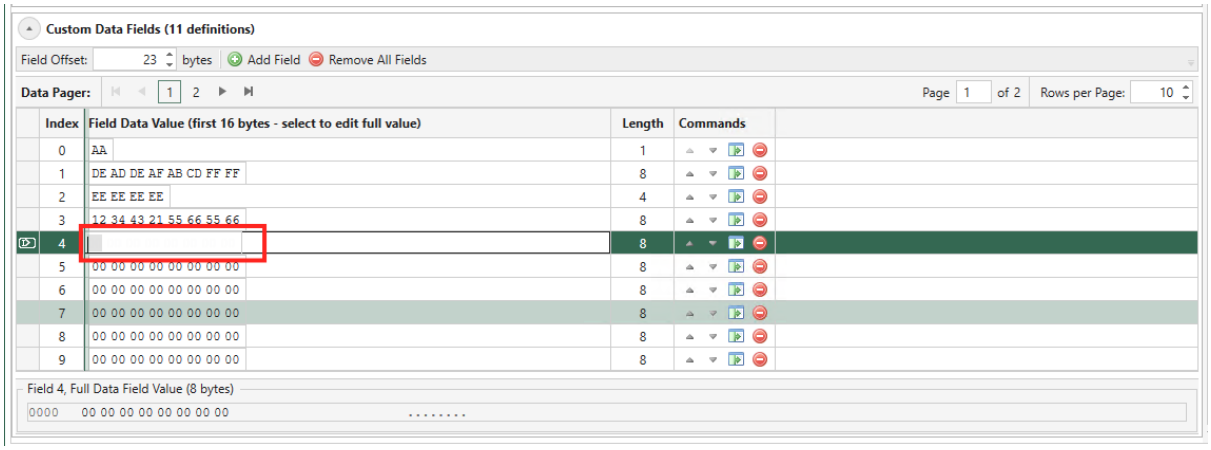


Fig. 5.100: Enter Values

Does My Test Module Support CDF?

In the *Port Capabilities* part of the *Port Properties* panel you can see if the test module supports CDFs: If Custom Data Fields Supported is True, the test module supports CDFs (see Fig. 5.101).

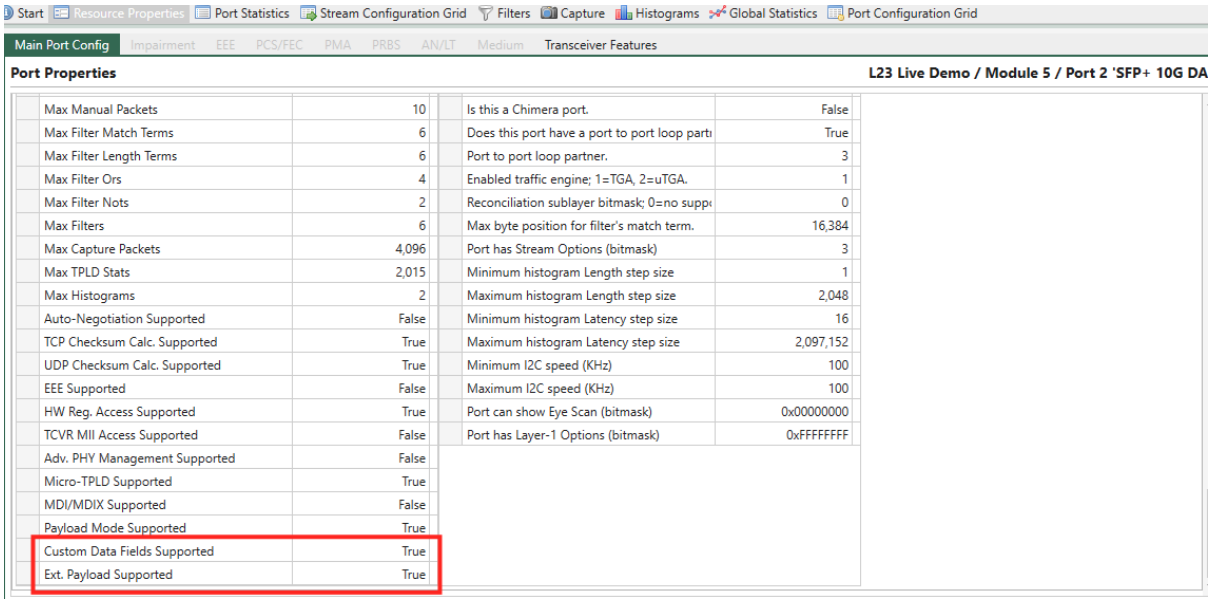


Fig. 5.101: Check if CDF or Ext. Payload is supported

CDF Specifications

The total area available for the CDF function depends on the test module type and port configuration.

Basic CDF specifications per 100G-and-beyond port:

- Max number of CDFs: 4096
- Max CDF size: Maximum Packet Size for the port.
- Max CDF offset = 2032

- Max CDF Memory: 262144 bytes (256K bytes)

> **Note**
>
> The Maximum Packet Size for the port is found in the Port Capabilities part of the port properties panel as the Max Packet Length (please observe that different data rates of a port may have different Maximum Packet Size).

When operating the module at lower port speeds, the ports share the above CDF memory and max. number of CDFs per port as follows:

- CDF Memory per port = Max 100G-and-beyond CDF Memory / number of ports
- Max number of CDFs per port = Max 100G-and-beyond number of CDFs / number of ports

CDF Memory Allocation

The amount of CDF memory used by a single stream is calculated in three steps:

1. Calculate the CDF size The size of a CDF is a combination of the number of data bytes it contains and its offset into the packet. It is calculated by the following formula:

```
CDF size = datasize (in bytes) + (offsetbytes_value modulo 8).
```

For example, a CDF with an offset of 3 and a data size of 234 bytes will have a CDF size of 237 bytes; if the offset is 11, the CDF size is also 237 bytes (11 modulo 8 is 3).

2. Calculate the CDF block memory size

   The CDF memory for each stream is split into blocks, each containing a single CDF. The CDF block sizes are identical for all CDFs on the stream and `2^n` bytes long, dimensioned to fit the largest CDF (see step 1). The block sizes can range from 64 bytes to the MTU for the port.

   As an example: If the largest CDF entry defined for a stream is 237 bytes, the block size will be 256 bytes. Hence, all CDFs for that stream will use 256 bytes of memory, regardless of their individual sizes.

3. Calculate the CDF memory usage for a single stream

   Based on the two steps above, the formula for calculating how much CDF memory a single stream uses is:

```
Stream CDF memory usage = CDF_memory_block_size * number_of_CDFs
```

Hence, if the stream described in the example above has 500 CDFs, the total used CDF memory for the stream is: 256 bytes * 500 = 128000 bytes.

Modifiers on CDF

Modifiers cannot be set on CDFs as the feature itself can be viewed as having some of the same characteristics as the modifier option but on a much larger scale.

> **Tip**
>
> You can see CDF as a stream modifier. When multiple CDFs are applied to a stream, the resulting packets appear as if they have an intricate modifier. This modifier varies in both size and values.

Extended Payload

The Extended Payload feature is enabled by selecting *Extended Payload* in the *Payload Mode* field of the parent port panel. This enables the feature for all streams on this port. Once the extended payload is enabled, it is possible to set the desired size of the *Packet Content* area for the stream using the stream's *Extended Payload Editor*.

Extended Payload Editor is a powerful UI feature that allows you to customize/configure the entire payload area of a stream in addition to the predefined payload pattern e.g. Incrementing 8-bits, Pattern, Random, and PRBS-31.

Please note that the *Extended Payload Editor* is only enabled when you set the port's *Payload Mode* to *Extended Payload* as shown in Fig. 5.102. This will enable all the streams under the port to accept the extended payload configuration.

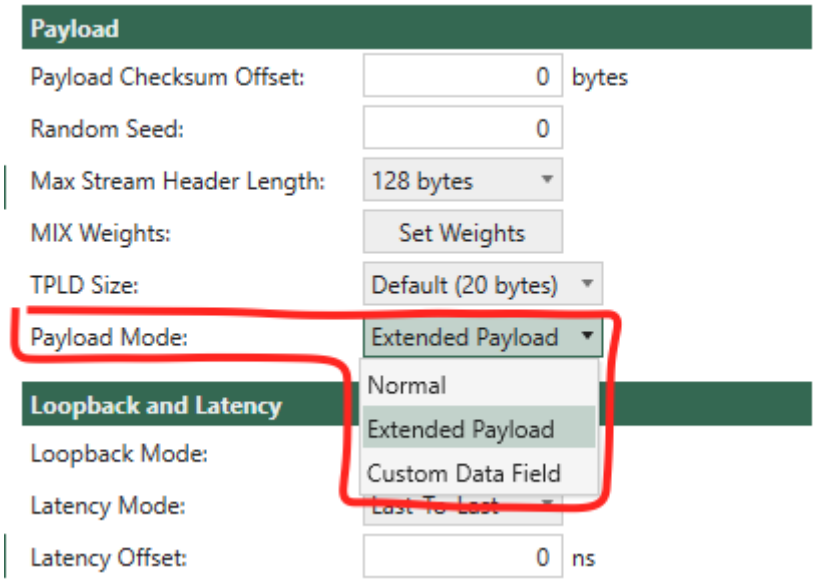


Fig. 5.102: Set port's payload mode to Extended Payload to enable the feature.

You can find the *Extended Payload Editor* as shown in Fig. 5.103 on the stream once the parent port has *Payload Mode* to *Extended Payload*. Click *Open Editor* and paste the payload data in hex strings and click *OK*. Then you can see your payload data in a byte-ascii grid view where each row contains 16 bytes.

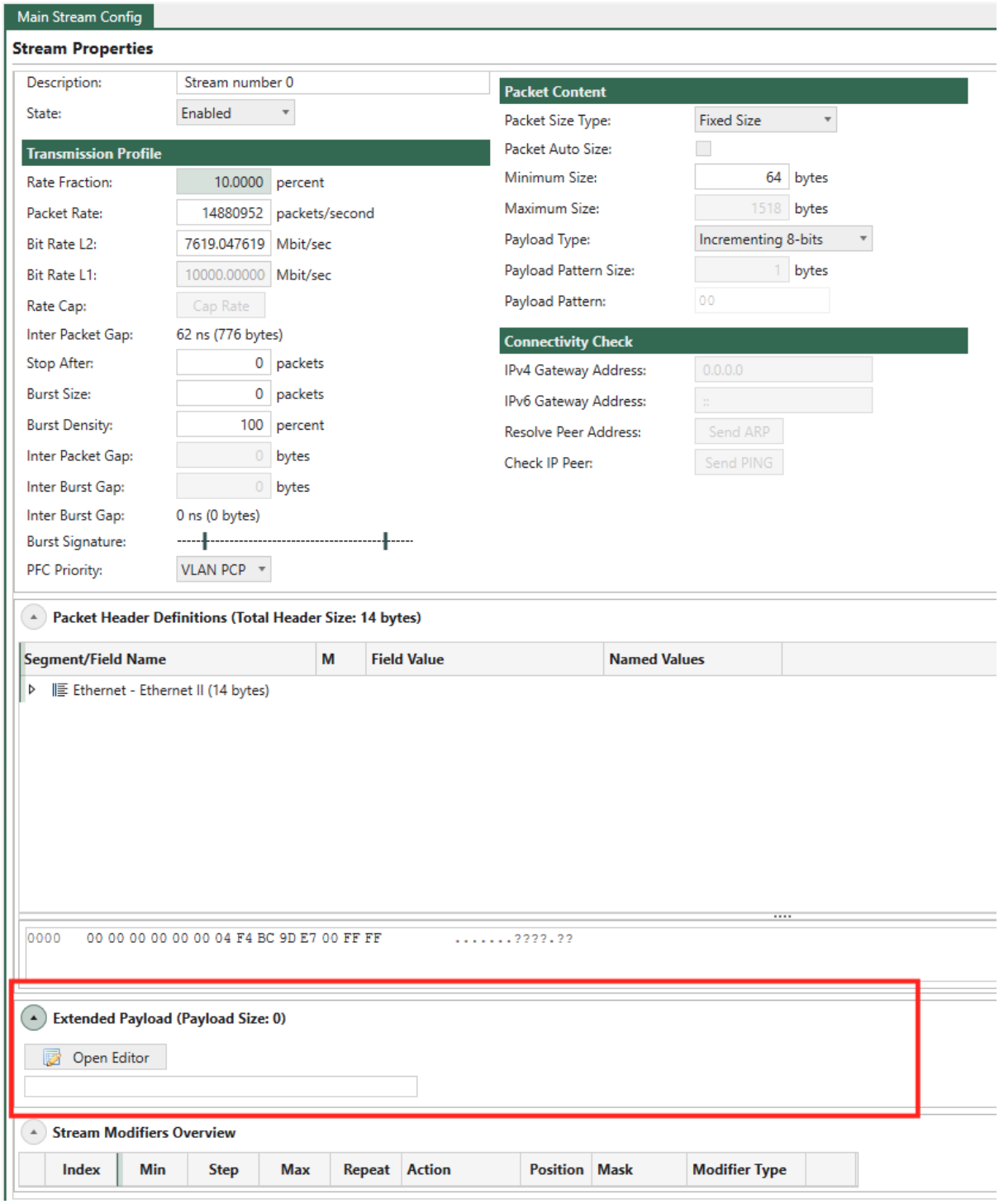


Fig. 5.103: Open Extended Payload editor

If you want to do adjustment on some bytes, you can click on the byte and enter the value as shown in Fig. 5.104.

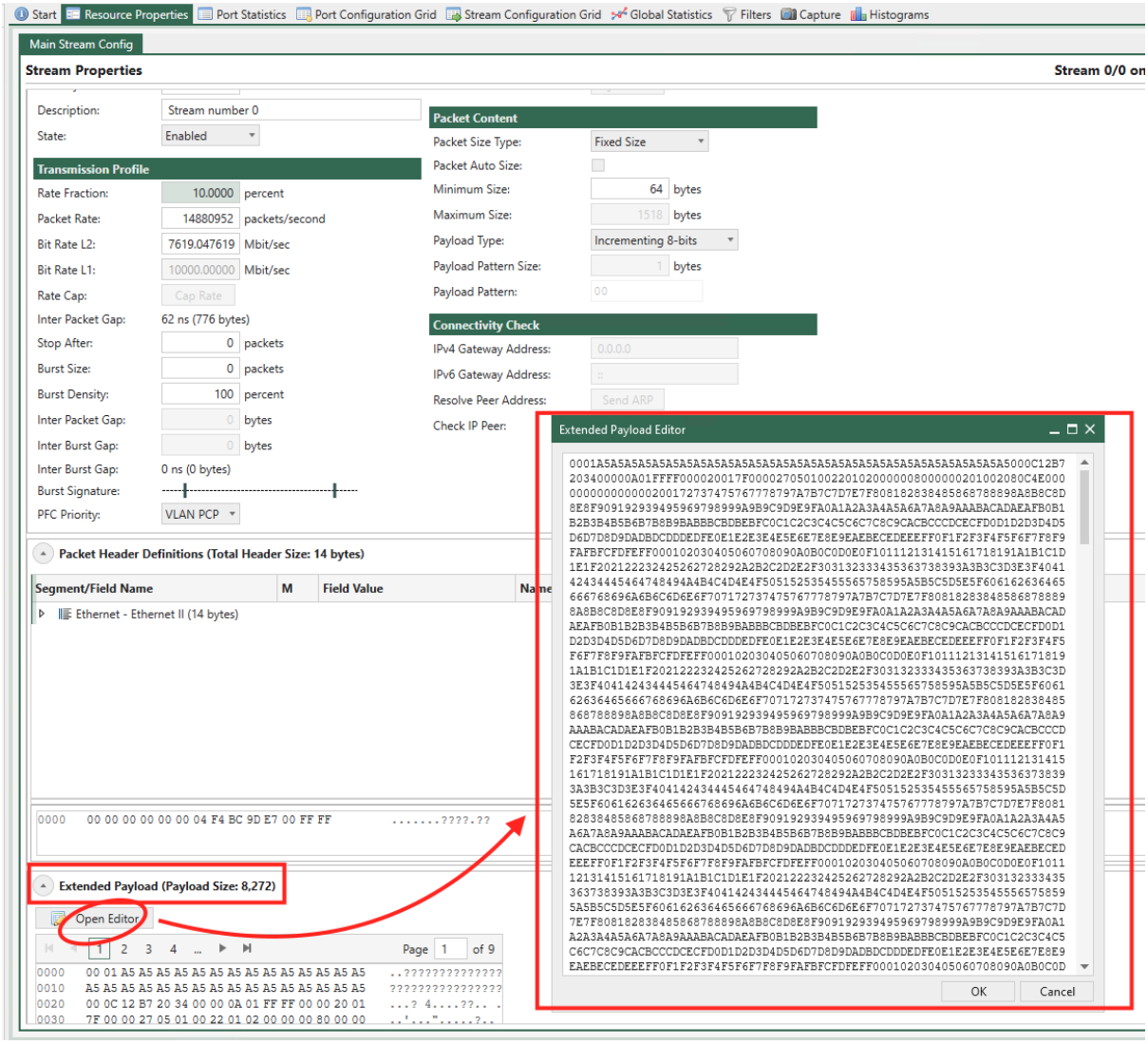


Fig. 5.104: Adjust some byte values

If the *Extended Payload* is not empty, the stream will use the provided payload data. If empty, the stream will use the configuration from *Packet Content* → *Payload Type*.

To quickly import the payload data from a pcap file, you can use Wireshark to copy the payload data , as shown in Fig. 5.105, and then paste it into the *Extended Payload Editor*.

Note

Please note that the total test packet size must be minimum the size of the **stream protocol headers + extended payload + TPLD + FCS**.

- If the total packet size is shorter than that, the Extended Payload is cut to fit the test packet.
- If it is longer, padding will be inserted between the Extended Payload and the TPLD.

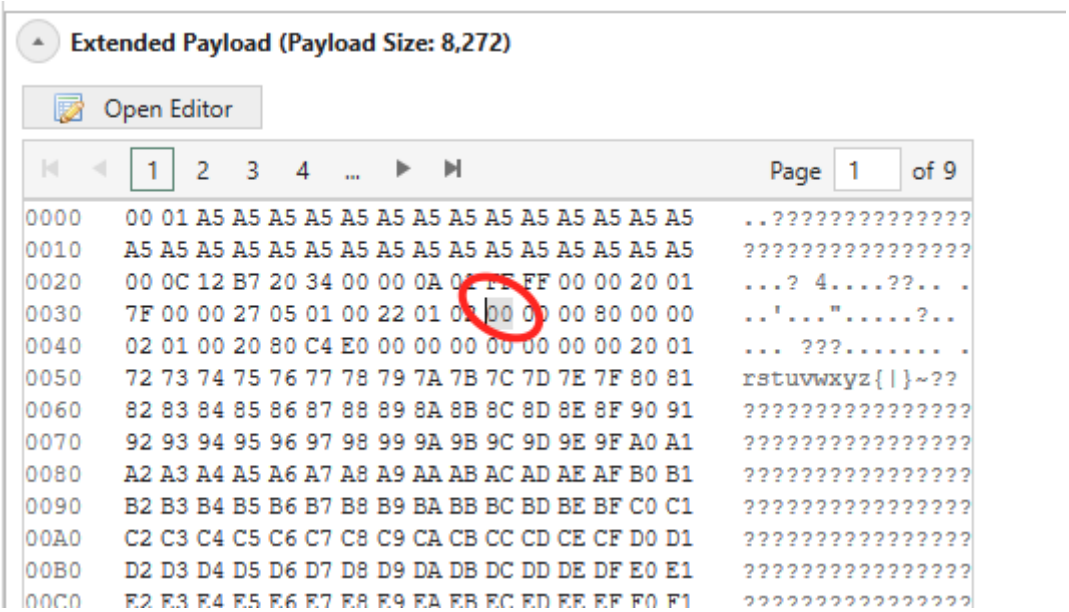


Fig. 5.105: Copy packet payload from pcap

Like a CDF, the Extended Payload is fit into a block, which is `2^n` bytes long. The block sizes can range from 64 bytes to MTU for the port; the smallest size where the Extended Payload fits in is used. Bytes in the block that is not used by the Extended Payload is filled with padding like CDF blocks.

Fig. 5.106 illustrates a test packet with 120 bytes Extended Payload in a 166 byte test packet. For the test packet in Fig. 5.106 this means that there is room for the Extended Payload plus the remaining 8 bytes in the memory block that the Extended Payload occupies.

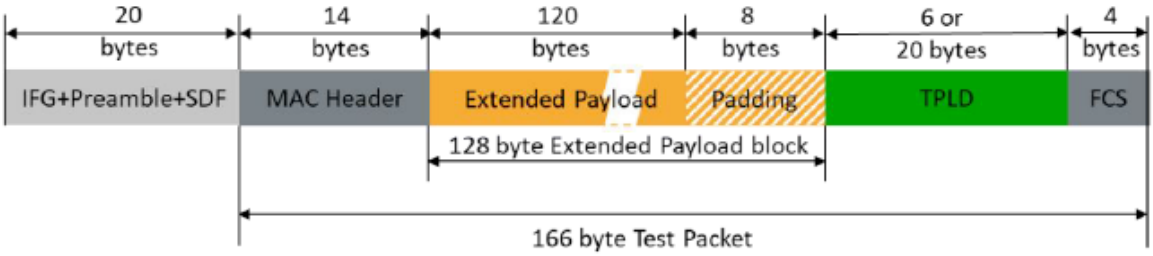


Fig. 5.106: Test packet with 120 bytes Extended Payload

See also

Read more in Freely Programmable Test Packets (Custom Data Fields) for details.

Does My Test Module Support Extended Payload?

In the *Port Capabilities* part of the *Port Properties* panel you can see if the test module supports CDFs: If Ext. Payload Supported is True, the test module supports Extended Payload (see Fig. 5.101).

Extended Payload Memory Allocation

The Extended Payload feature is a special application of the CDF feature with an offset matching the headers in the test package. Therefore, the sections on CDF specifications and CDF Memory Allocation also apply for the Extended Payload feature

Modifiers on Extended Payload

It is possible to set a modifier in the Extended Payload area just as it is for a normal protocol field.

> **Tip**
>
> It is different from CDF because Extended Payload is fundamentally a more advanced payload compared to the basic.

Test Payload

Test Payload (Normal)

Each Xena test packet contains a special proprietary data area called the *Test Payload Data (TPLD)*, which contains various information about the packet and is identified by a *Test Payload ID (TID)*. The *TPLD* is located just before the Ethernet FCS and consists of the following sections:

Table 5.42: Default TPLD (20 or 22 bytes)

Field	Length	Explanation
Payload Integrity Checksum (optional)	2 bytes	See the *note*.
Sequence Number	3 bytes	Packet sequence number used for loss and misordering detection.
Timestamp	4 bytes	Timestamp value used for latency measurements.
Test Payload ID (TID)	2 bytes	Test payload identifier used to identify the sending stream.
Payload Integrity Offset	1 byte	Offset in packet from where to calculate payload integrity.
First Packet Flag	1 bit	Set if this is the first packet after traffic is started.
Checksum Enabled	1 bit	Set if payload integrity checksum is used.
<reserved>	7 bits	
Payload Integrity Offset (MSB)	3 bits	Offset in packet from where to calculate payload integrity, MSB (bits 10:9:8)
Timestamp Decimals	4 bits	Additional decimals for the timestamp.
Checksum	8 bytes	TPLD integrity checksum (CRC-32)
Total TPLD Size	**20 or 22 bytes**	

Note

If the Payload Checksum Offset is enabled on the parent port (value > 0) , then an additional 2-byte checksum field is inserted in the beginning of the TPLD, just before the Sequence Number. This increases the total size of the TPLD to 22 bytes.

Test Payload (Micro)

Table 5.43: Micro-TPLD (6 bytes)

Field	Length	Explanation
First Packet Flag	1 bit	Packet sequence number used for loss and misordering detection.
<reserved>	1 bit	
Test Payload ID (TID)	10 bits	Test payload identifier used to identify the sending stream.
Timestamp	28 bits	Timestamp value used for latency measurements.
Checksum	8 bits	TPLD integrity checksum (CRC-32)
Total Micro-TPLD Size	**6 bytes**	

The selection between the default TPLD and the micro-TPLD is done on the parent port. It is thus not possible to use different TPLD types for streams on the same port.

Timestamp Calculation

Timestamp consists of two parts: <Timestamp> and <Timestamp Decimals> (<Timestamp Decimals> not available in micro TPLD).

Timestamp is calculated into **nanoseconds** by the equation below:

Timestamp = <Timestamp>*8 + <Timestamp Decimals>/2

```
For instance:

if <Timestamp> = 0x7a2ce291, and <Timestamp Decimals> = 0x0d,
then Timestamp = 2049761937*8 + 13/2 = 16,398,095,502 nsec
```

Disabling Test Payload

The Test Payload can also be completely disabled for any given stream by setting the *Test Payload ID (TID)* value for the stream to the value **-1**.

Xena Wireshark Plugin for TPLD

You can download the Xena TPLD Wireshark pluggin from xenanetworks/tpld-parser-wireshark

Xena offers a Wireshark integration via a dedicated LUA plugin. It allows users to easily read the information in the Xena test signature field. The plugin supports Wireshark 3.x and above.

5.4 Port Statistics

This section describes the XenaManager port statistics page. The page displays statistics information for the currently selected port and all streams defined on that port.

The page will only display data for a single port at a time. If you want to monitor statistics data for multiple ports at a time please refer to *Stream Configuration Grid*.

5.4.1 Port Transmit Statistics

This area contains statistics for all data transmitted by the port.

Commands in TX Toolbar

The area contains a toolbar with the following commands:

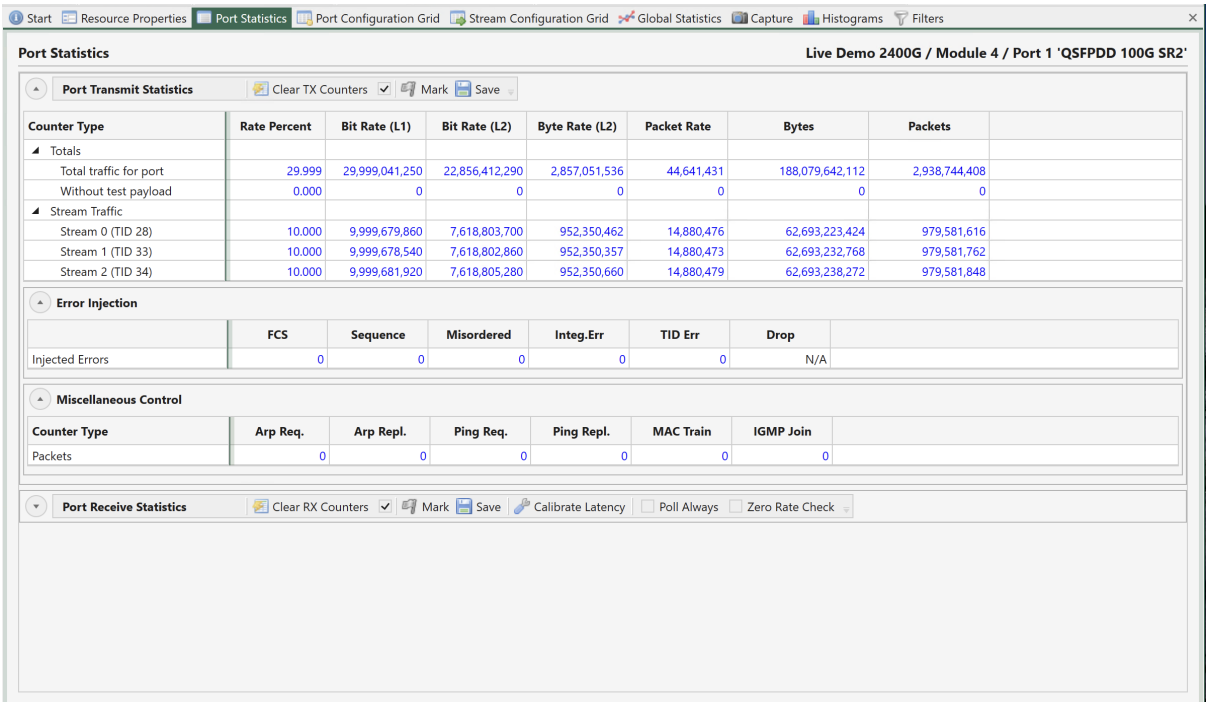


Fig. 5.107: Port TX statistics

Table 5.44: Commands in TX Toolbar

Command	Explanation
Clear TX Counters	Clear the current TX statistics counters for this port.
(checkbox)	If checked the TX statistics counters for this port will be affected when the Clear Statistics command is executed in the Global Statistics panel.
Mark	Set the font color of the current counter values to gray. Any counter value that changes afterwards will revert to blue. This makes it easy to check if a value changes over time.
Save	Allow you to save the current counters to a CSV text file.

Main Transmit Statistics

Table 5.45: Main Transmit Statistics

Statistics Type	Explanation
Total Traffic for Port	This row show statistics counters for all traffic transmitted on the port regardless of type. This is the sum of the traffic sent without test payload and the traffic sent for each active stream.
Without Test Payload	This row show statistics counters for the part of the transmitted traffic that is sent without test payload.
Stream Traffic	This branch contains a row for each stream currently active on the port.

Common Column Header Functions

The two statistics tabs also share a common functionality with regard to the grid column headers. You can reorder the columns in the grid by dragging a column header to a new location. The new order will be remembered the next time you start XenaManager. The following options are available when right-clicking on the grid column headers:

Table 5.46: Common Column Header Functions

Functions	Explanation
Hide Column	Hide the selected column from view. This selection will be remembered the next time you start XenaManager.
Reset Column Order	Resets any custom column order you may have configured to the default order.
View All Columns	Show all columns you may have hidden previously.

Error Injection

This section show the number of errors manually injected by the user for each possible error type.

Miscellaneous Control

This section show the number of transmitted ARP/NDP and PING requests and replies, MAC training packets and IGMP Joins.

5.4.2 Port Receive Statistics

This area contains statistics for all data received by the port.

Commands in RX Toolbar

The area contains a toolbar with the following commands:

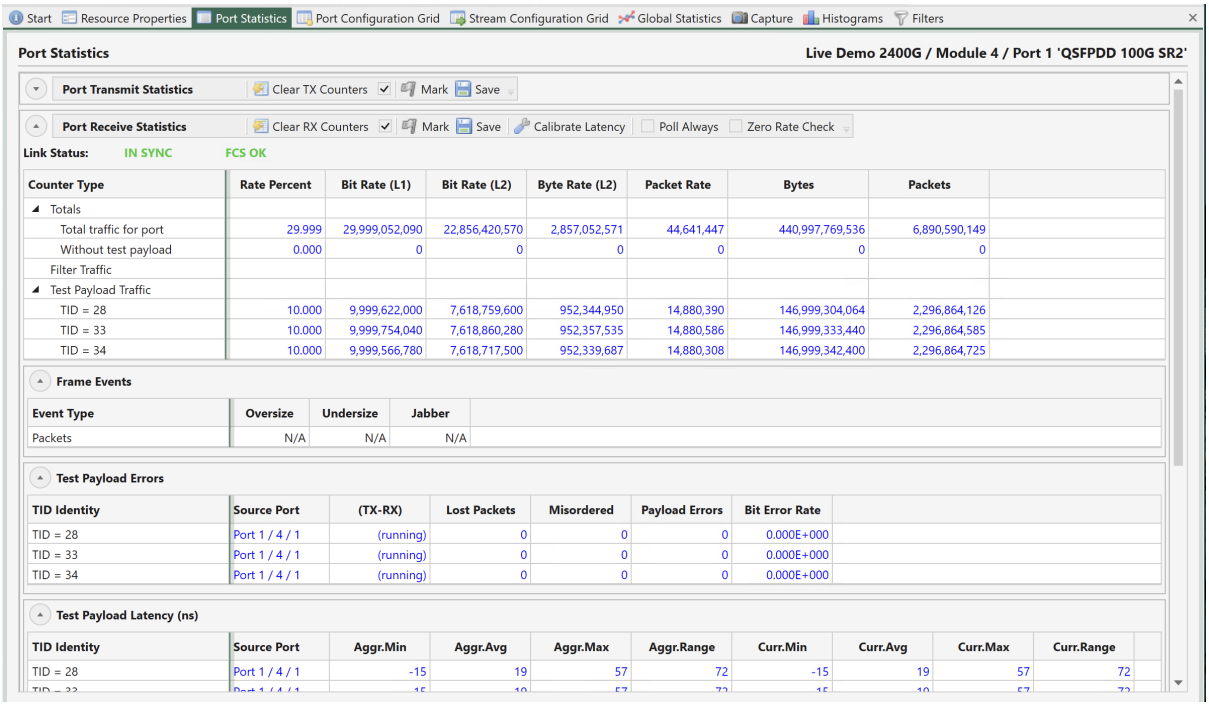


Fig. 5.108: Port RX statistics

Table 5.47: Commands in RX Toolbar

Command	Explanation
Clear RX Counters	Clear the current RX statistics counters for this port.
(checkbox)	If checked the RX statistics counters for this port will be affected when the Clear Statistics command is executed in the Global Statistics panel.
Mark	Set the font color of the current counter values to gray. Any counter value that changes afterwards will revert to blue. This makes it easy to check if a value changes over time.
Save	Allow you to save the current counters to a CSV text file.
Calibrate Latency	When pressed the current average latency value will be saved as the calibrated latency offset.
Poll Always	Normally a port is only polled for statistics counters when it is visible in a statistics panel, such as this panel or the Global Statistics panel. But if you want the port to be polled always you can check this box. This function can be useful if you want to monitor the state of the port over a long period of time.

Main Receive Statistics

Table 5.48: Main Receive Statistics

Statistics Type	Explanation
Total Traffic for Port	This row show statistics counters for all traffic received on the port regardless of type. This is the sum of the traffic received without test payload and the traffic received for each TID.
Without Test Payload	This row show statistics counters for the part of the received traffic that is sent without test payload.
Filter Traffic	This branch contains a row for each active filter on the port.
Test Payload Traffic	This branch contains a row for each TID received on the port.
Frame Events	Shows number of Oversize, Undersize and jabber packets.

Test Payload Specific Counters

A number of sections show counters received for each TID:

Table 5.49: Test Payload Specific Counters

Counter Type	Explanation
Test Payload Errors	This section contain a row for the payload errors received for each TID on the port.
Test Payload Latency	This section contain a row for the payload latency measured for each TID on the port.
Test Payload Jitter	This section contain a row for the payload jitter measured for each TID on the port.

Misc. Counters

This section show statistics for various other counter types.

Table 5.50: Misc. Counters

Counter Type	Explanation
Arp Req.	Received ARP/NDP requests
Arp Repl.	Received ARP/NDP replies
Ping Req.	Received PING requests
Ping Repl.	Received PING replies
FCS Errors	Received packets with *FCS* errors
PAUSE	Received PAUSE frames
Gap Count	Number of gap counts detected
Gap Dur.	Current detected gap duration

5.5 Port Configuration Grid

This section explains how to use the port configuration grid.

The Port Configuration Grid show all ports currently in your testbed. It does not show any other ports, not even if they have been reserved by you.

For a more detailed description of each port property please refer to the *Main Port Config*.

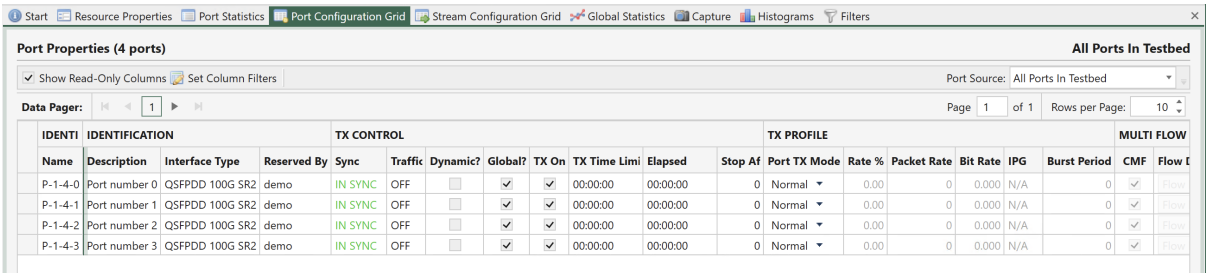


Fig. 5.109: Port configuration grid

5.5.1 Grid Operations

Column Filtering

Each grid contains a lot of columns which may slow down the loading of the grid and/or may overload your visibility. This section explains how you can limit the number of displayed columns and select which columns you want to be shown.

- Show Read-Only Columns

The toolbar contains a checkbox which toggles the visibility of the read-only columns. This is useful if you only want to show columns that actually allow you to configure something.

- Set Column Filters

If you want more control over the displayed columns you can press the *Set Column Filters* button in the toolbar. This will allow you to filter the columns either based on their group or individually. The filter selections will be remembered the next time you start up the XenaManager.

Frozen Columns

The first set of columns that uniquely identifies the entity in each row will be frozen which means that they will not scroll out of view even if you scroll the columns all the way to the right.

Right-Click Operations

Each cell in the grid may support one or more of the right-click actions described below.

Use Value for All

If this action is selected the value for this cell will be used for all the other rows.

Use Value for All w.Increment

If this action is selected the value for this cell will be used as a template value for all the other rows, but it will be incremented for each row. The increment will be performed based on the value type. A rate value of e.g. `11.4` will be incremented to `12.4`. An IP address of `10.0.0.4` will be incremented to `10.0.0.5`.

5.6 Stream Configuration Grid

The Stream Configuration Grid show all streams for ports that meet the Stream Source selection criteria. It does not show streams for any other ports, not even if they have been reserved by you.

The default criteria is *All Ports In Testbed* which is the same criteria as is used for the *Port Configuration Grid*.

You can however change the criteria using the dropdown box in the local toolbar to *Currently Selected Port(s)*. This will only show streams for the ports you have selected in the *Available Resources* tree view.

For a more detailed description of each stream property please refer to *Stream Properties*.

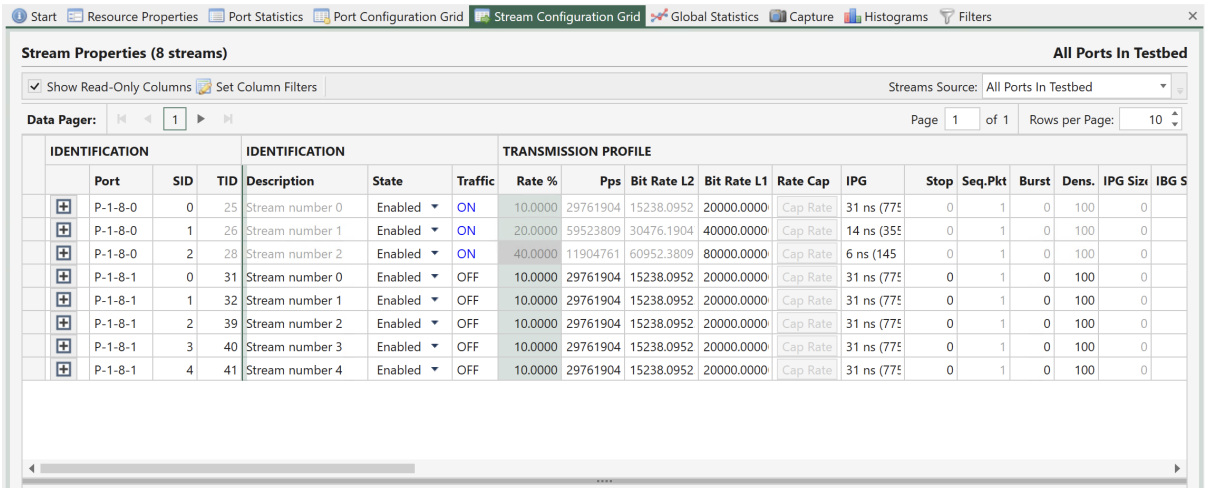


Fig. 5.110: Stream configuration grid

5.6.1 Accessing the Packet Header Editor

It would be unrealistic to display all possible protocol segment fields in the grid. We have chosen to display a few commonly used fields. For the rest you can access the Packet Header Editor described on this page by clicking on the plus sign at the start of the row. The Packet Header Editor will then expand below the grid rows as shown in Fig. 5.110.

5.6.2 Grid Operations

Column Filtering

Each grid contains a lot of columns which may slow down the loading of the grid and/or may overload your visibility. This section explains how you can limit the number of displayed columns and select which columns you want to be shown.

- Show Read-Only Columns

The toolbar contains a checkbox which toggles the visibility of the read-only columns. This is useful if you only want to show columns that actually allow you to configure something.

- Set Column Filters

If you want more control over the displayed columns you can press the *Set Column Filters* button in the toolbar. This will allow you to filter the columns either based on their group or individually. The filter selections will be remembered the next time you start up the XenaManager.

Frozen Columns

The first set of columns that uniquely identifies the entity in each row will be frozen which means that they will not scroll out of view even if you scroll the columns all the way to the right.

Right-Click Operations

Each cell in the grid may support one or more of the right-click actions described below.

Use Value for All

If this action is selected the value for this cell will be used for all the other rows.

Use Value for All w.Increment

If this action is selected the value for this cell will be used as a template value for all the other rows, but it will be incremented for each row. The increment will be performed based on the value type. A rate value of e.g. `11.4` will be incremented to `12.4`. An IP address of `10.0.0.4` will be incremented to `10.0.0.5`.

5.7 Global Statistics

This section describes the XenaManager Global Statistics panel.

5.7.1 Overview

The Global Statistics panel show all ports and streams currently in your testbed. It does not show any other ports (or streams on these ports), not even if they have been reserved by you.

5.7.2 Statistics Tabs

The panel is divided into two tabs, *Port Statistics* and *Stream Statistics*. The first tab show testbed-global *Port Statistics* and the other show testbed-global Stream Statistics.

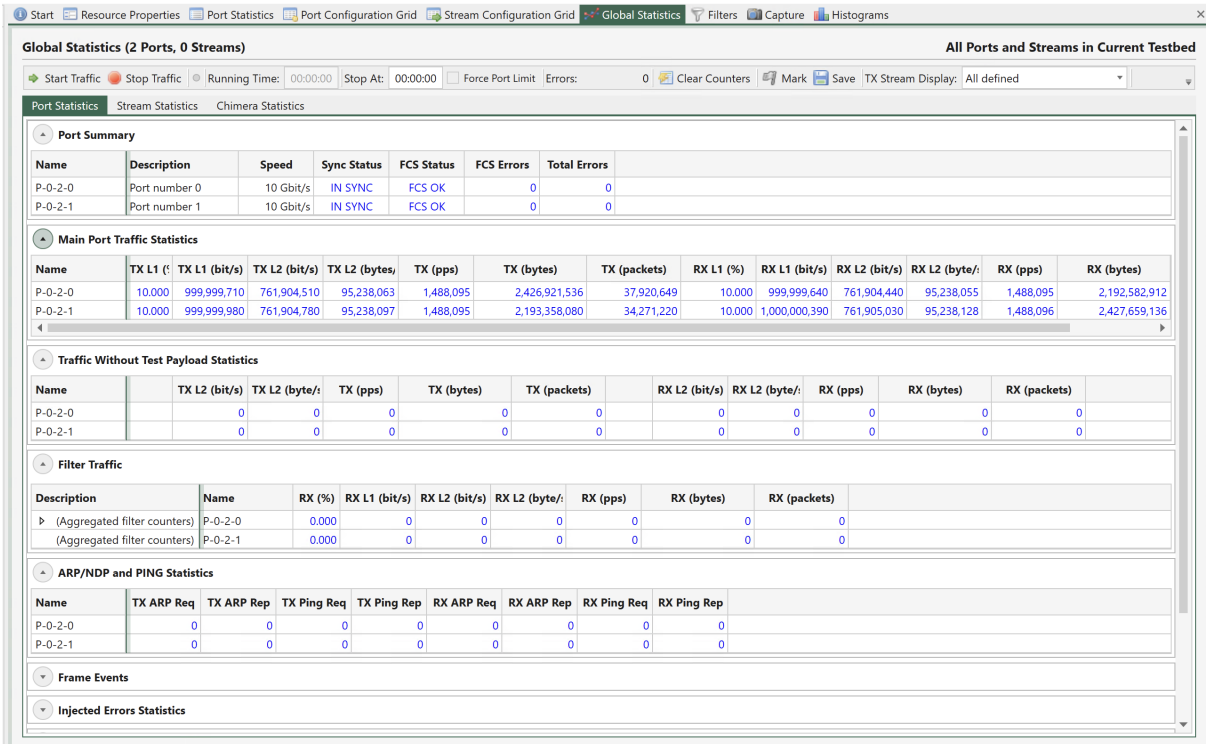


Fig. 5.111: Global Statistics - Port Statistics

Common Toolbar Functions

The two statistics tabs share a common toolbar with the following functions:

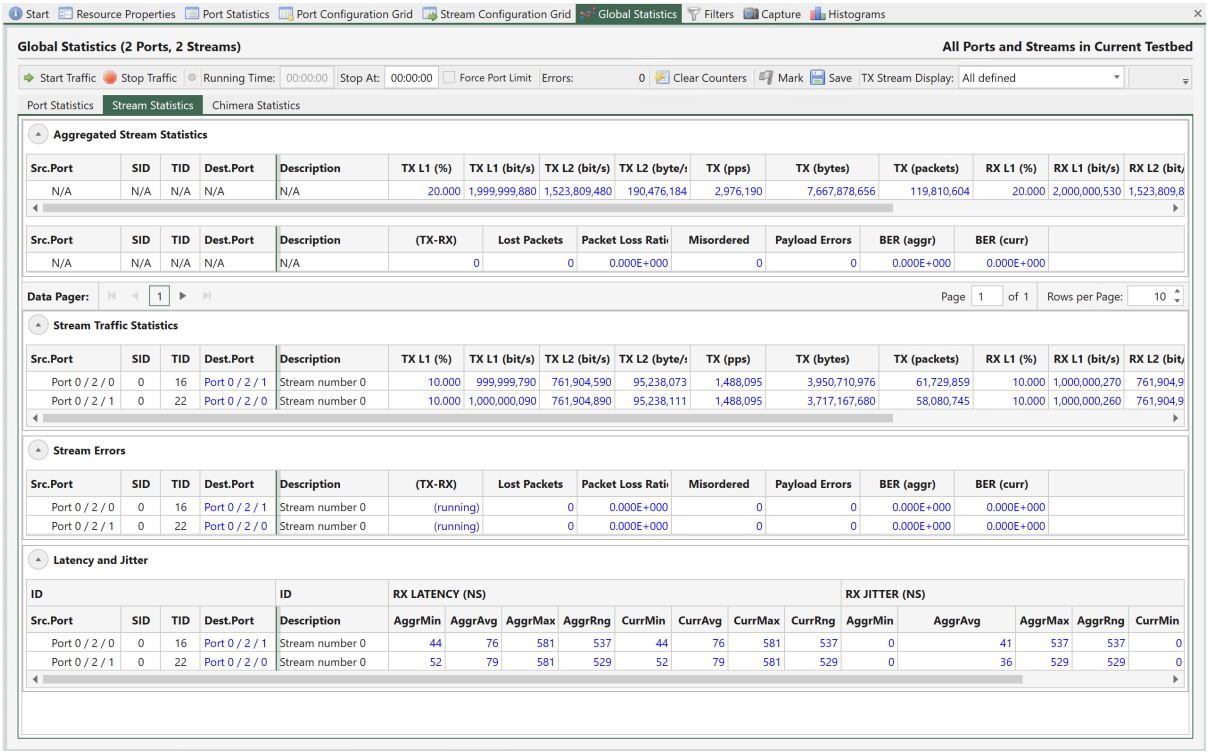


Fig. 5.112: Global Statistics - Stream Statistics

Table 5.51: Common Toolbar Functions

Function	Explanation
Start Traffic	Start traffic on all ports in your testbed. Ports that are already transmitting are not affected. This command may also affect any capture and histograms defined for the ports if you have enabled it.
Stop Traffic	Stop traffic on all ports in your testbed. Ports that are not transmitting are not affected. This command may also affect any capture and histograms defined for the ports if you have enabled it.
Running Time	Show the amount of time that has elapsed since you performed a Start Traffic command in this panel. If an individual port had been started before this point in time this is not reflected in the Running Time value.
Stop At	Allow you to specify a time limit for the port transmission. When the Running Time exceeds this value the port traffic will be automatically stopped.
Errors	Display the total number of errors on all ports in your testbed.
Clear Counters	Clear the current statistics counters for all streams on all ports.
Mark	Set the font color of the current counter values to gray. Any counter value that changes afterwards will revert to blue. This makes it easy to check if a value changes over time.
Save	Allow you to save the current counters to a CSV text file.
Force Port Limit	Force Port limit will apply *Stop after* to all the used ports. **Note** Force Port Limit will overwrite any TX Time Limit value configured on Port property page on the used ports, and use the Stop after value.

Common Column Header Functions

The two statistics tabs also share a common functionality with regard to the grid column headers. You can reorder the columns in the grid by dragging a column header to a new location. The new order will be remembered the next time you start XenaManager. The following options are available when right-clicking on the grid column headers:

Table 5.52: Common Toolbar Functions

Function	Explanation
Hide Column	Hide the selected column from view. This selection will be remembered the next time you start XenaManager.
Reset Column Order	Resets any custom column order you may have configured to the default order.
View All Columns	Show all columns you may have hidden previously.

Port Statistics

The Port Statistics tab show statistics counters for all ports in your testbed. In general each port is represented by a single row which contain both Tx and Rx counters for that port.

Port Summary

The Port Summary section provides a brief overview of the main port state properties for the testbed ports.

Traffic Statistics Counters

The available statistics counters for each port are the same as for the individual port statistics page described in *Port Statistics*.

Stream Statistics

The Stream Statistics tab show statistics counters for all streams on all ports in your testbed. Each counter type is explained in the individual port statistics page described in *Port Statistics*.

TID Matching

The counters are shown in a grid view where each row represent both ends of a stream. The stream "ends" are matched together using the Test Payload ID (*TID*) which is configured on the stream at the transmit end and transferred to the received end within the Xena test payload inside each packet.

To enable an accurate matching of Tx and Rx stream ends it is imperative that the used TID values are unique within the testbed. Otherwise it will be impossible to determine which stream on which port was the sender of a given packet.

Aggregated Stream Statistics

This section show the aggregated counter values for all streams in the view.

Stream Traffic Statistics

This section show the main stream traffic counters for each stream.

Each row in the grid represents a test-stream end-to-end. The stream entity is identified by the TID value. The Tx counter values are read from the transmitting port and the Rx counter values are read form the receiving port.

TID Conflicts

If two or more streams in your testbed use the same *TID* value the Stream Statistics grid will not be able to accurately determine where the various TID contributions originate from on the receiving side. The grid will show this situation as a single parent row representing all receive-side contributions from the given TID value. The transmit-side contributions will be shown as N/A.The source port will be shown as Multiple.

Each transmit-side contribution will be listed as a child row. You can expand the child rows by clicking the expander icon to the left of the row.

Stream Errors

The Stream Errors section show the errors detected for each end-to-end stream entity. The mechanism for showing TID conflicts explained above is also used here.

The following error counters are shown:

Table 5.53: Stream Errors

Name	Explanation
(TX-RX)	The difference between the sent packets for the stream on the transmitting port and the received packets on the matching TID entry on the receiving port. This value is not accurate while the traffic is running as it is not possible to accurately read TX and RX counters on different ports at the exact same time. So this value is only shown when traffic is stopped.
Lost Packets	The calculated loss based on the embedded sequence number in the test payload section in the received packets. This value is somewhat accurate while the traffic is running. It is especially good at detecting on-going loss. But it cannot detect lost packets at the very start or at the very end of the packet stream since the receiver cannot know how many packets was sent before the first packet it receives or how many packets are actually lost after the last packet it receives.
Misordered	The number of misordered packets detected, i.e. packets arriving out of sequence compared to the embedded sequence number in the test payload section. The same uncertainly regarding packets at the very start or at the very end explained above applies here as well.
Payload Errors	The number of packets received that failed the test payload integrity check. These packets are not counted as lost or misordered as they strictly speaking are valid Ethernet packets. But their presence indicates that the DUT/SUT changed something in the payload section which caused the payload integrity check to fail. Payload Integrity Error detection is supported regardless of the *Payload Type* <*stream-packet-content-label*>. For Payload Types that are not *Incrementing 8-bit*, you need to set the same non-zero *Payload Checksum Offset* on both the TX and the RX ports
Bit Error Rate	This value is an **estimated bit error rate (BER) measured at Layer-2** over the time span since the traffic counters was last cleared. The BER value provided is estimated based on the assumption that 1 errored packet equals 1 bit error. If more than one bit error occurred in one errored packet, this will not be detected by the Xena tester. **Based on this assumption** the estimated BER is calculated as follows by the XenaManager: `BER = DroppedFrames * RxFrames / (8 * RxBytes) * (RxFrames + DroppedFrames)`

Note

(TX-RX) packets gives packet loss results based on TX packets and RX packets.

RX packets can consist of two parts: `RX_1` originated from the TX port, `RX_2` duplicated by the DUT

1. **`(TX-RX) > 0`:**
    a. `TX > RX_1+RX_2`. Transmitted packets more than received even if there is duplication from the DUT.

2. **(TX-RX) = 0:**
    a. TX-RX_1 = 0, and RX_2 = 0. All transmitted packets are correctly received, and there is no duplication introduced by the DUT.
    b. TX-RX_1 = d, but RX_2 = d. Some transmitted packets are not received but the duplications by DUT cancels out the difference, where d is positive.
3. **(TX-RX) < 0:**
    a. TX-RX_1 = 0, RX_2 > 0. There is duplication from the DUT.
    b. TX-RX_1 = d, RX_2 > d. Duplication from the DUT is large than the lost, where d is positive

Lost Packets

Given that RX can consist of two parts as mentioned above, we cannot be certain about the TX-RX_1. Consequently, we need use the sequence numbers in the packets to calculate "the holes in the sequence". For example, when the TX sends #0,#1,#2,#3,#4,#5,#6, but #3 is not received by the RX, then TX-RX_1 = 1. But if #3 is not truly lost but is delayed by the DUT and later received by the RX, the TX-RX_1 will be back to 0.

TX: #0,#1,#2,#3,#4,#5,#6 RX: #0,#1,#2,#4,#5,#6 (Lost Packets = 1) (after a while...) RX: #3 (Lost Packets = 0)

Unfortunately, there is no one mechanism that can handle both scenarios at the moment.

Note

BER = ErroredBitsReceived / TotalBitsTransferred

= DroppedFrames * ErroredBitsPerDroppedFrame / TotalFramesTransferred * BitsPerTransferredFrame

= DroppedFrames * ErroredBitsPerDroppedFrame / (RxFrames + DroppedFrames) * BitsPerTransferredFrame

Let ErroredBitsPerDroppedFrame = e, and BitsPerTransferredFrame = b, and assume:

- there is only one errored bit in a dropped frame = e = 1
- received frames and dropped frames have the same bits per frame, $b_{rx} = b_{dropped} = b$ =(8 * RxBytes)/RxFrames

Then, BER = DroppedFrames * RxFrames /(8 * RxBytes) * (RxFrames + DroppedFrames)

5.7.3 Latency and Jitter

The Latency and Jitter section show the latency and jitter values calculated for each end-to-end stream entity. The mechanism for showing TID conflicts explained above is also used here.

Jitter measurement is only available for TPLD ID < 32, as shown in the screenshot below.

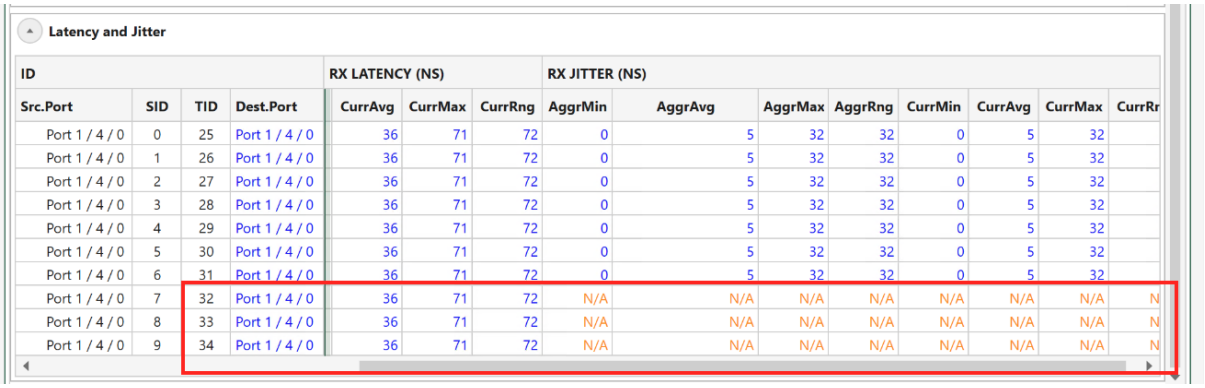


Fig. 5.113: Jitter measurement is only available for TPLD ID < 32.

5.8 Filters

This section describes the XenaManager Filters panel. The panel allows you to configure the filters for the currently selected port.

5.8.1 Overview

Every port has a filter mechanism for inspecting all the received packets and recognizing particular patterns within the packets. Filters are defined under their own *Filters* panel in the content area of the XenaManager. Filters are independent of the test payloads and provide an alternative method for analyzing the train of received packets.

Filters are logical conditional expressions on a number of basic true-or-false terms, which can be of two types: match terms and length terms.

- Match terms look for a particular pattern of bits at a particular position within each packet.
- Length terms look for packets that are longer or shorter than a particular size.
- A number of these two terms can then be combined into a single filter condition.

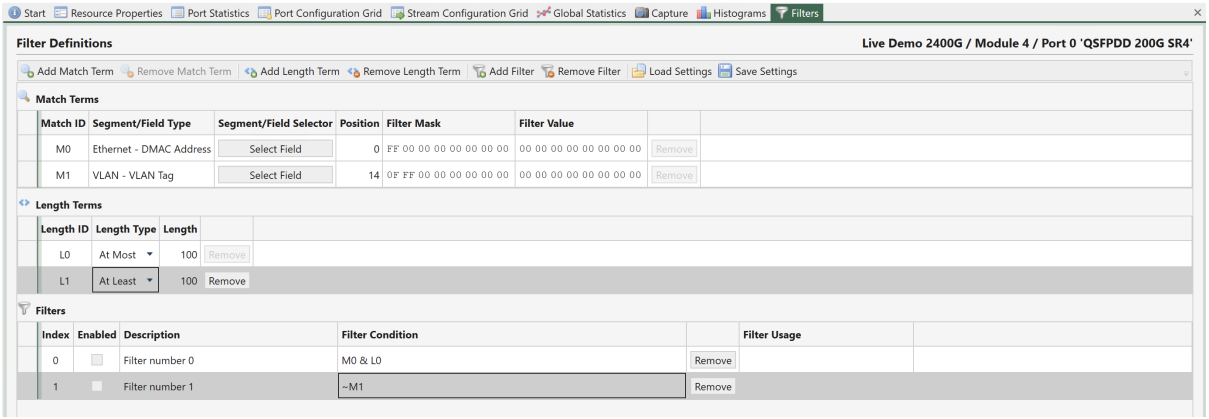


Fig. 5.114: Port filters

5.8.2 Filter Details

Match Terms

As stated above match terms look for a particular pattern of bits at a particular position within each packet. Like a modifier, a match term will typically correspond to a particular protocol field.

And like a modifier you can select the protocol field where you want to position the filter. However, since a filter is not related to any stream definition you need to manually click the *Add* button to build the needed protocol segments.

Match terms also consist of a filter mask and a filter value. The mask indicates which part of the value you want to match on. The filter value is the actual value you want to match on. A match term is identified with the code `M<index>`, where `<index>` is a non-negative integer identifying the match term.

Length Terms

As stated above length terms look for packets that are longer or shorter than a particular size. If you want to look for packets that fit within a certain range you will need to define two length terms; one that looks for packets larger than or equal to the minimum size (At Least) and one that looks for packets smaller than or equal to the maximum size (At Most).

A length term is identified with the code `L<index>`, where `<index>` is a non-negative integer identifying the length term.

Filter Condition

Each filter consist mainly of a filter condition that combines one or more match terms and one or more length terms. The filter condition is built using a boolean expression using the match and length term identifiers names. The condition can use the usual Boolean operators `&`, `|`, and `~`. The `|` operator has the lowest precedence.

Example: `M0 & L0 & ~M1` means **match M0 but not M1 and also fulfill L0**

5.8.3 Using Filters

Filters can be used in different ways: the port will accumulate separate statistics for packets satisfying the filter condition, the capture mechanism can use the filters as start/stop/keep criteria, and likewise for the histogram mechanism.

5.9 Capture

This section describes the XenaManager Capture panel. The panel allow you to configure the capture settings for the currently selected port and to inspect the result of the capture.

5.9.1 Overview

All packets arriving at a port are counted and analyzed if they contain test payloads. In addition, selected packets can be retained (captured) for closer inspection using the capture mechanism.

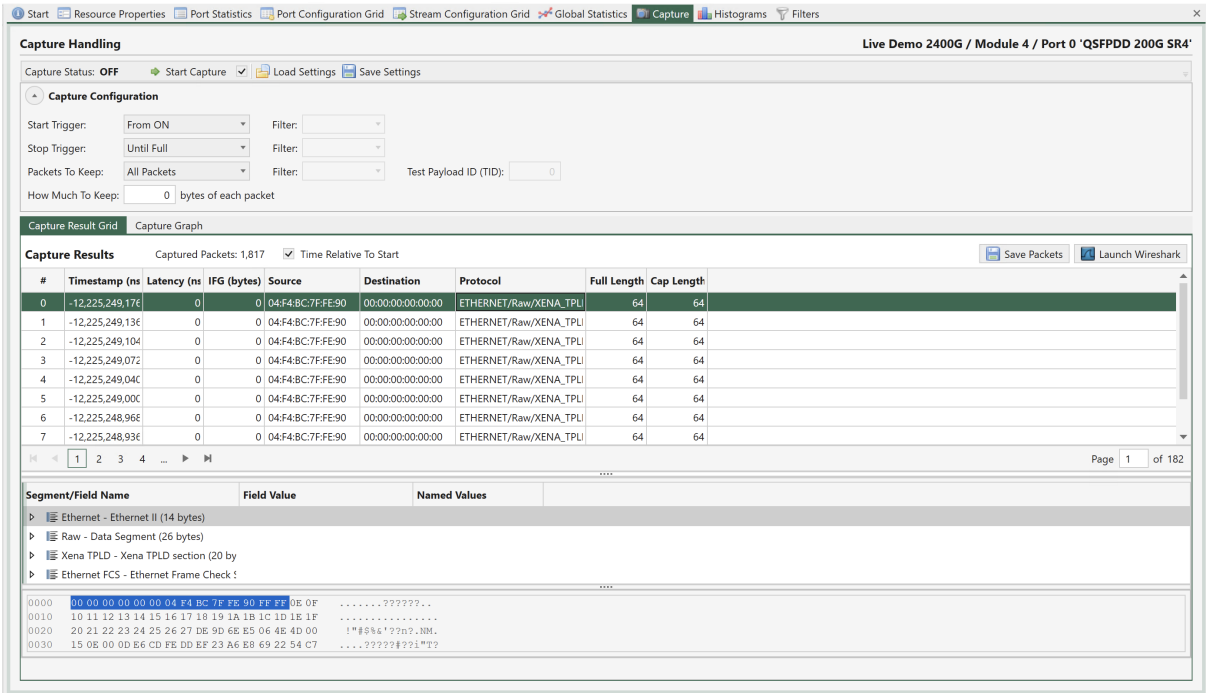


Fig. 5.115: Port capture

5.9.2 Configuration

Start Trigger

The *Start Trigger* control when the capture function actually begins to collect packets in the capture buffer. The following options are supported:

- From ON: The default behavior is to begin collection when capture is started.
- From FCS Error: Begin collection when the first FCS error in a received packet is detected.
- From Filter: Begin collection when the first packet that matches the specified filter is received.
- From Payload Error: Begin collection when the first payload error in a received packet is detected.

> **Note**
>
> Different modules and ports have different options available.

Stop Trigger

The *Stop Trigger* can be used to control when the capture function stops collecting packets. When using a start trigger, capturing is automatically stopped if the internal capture buffer runs full. When using only a stop trigger, the hardware capture buffer retains as many packets as possible up until the stop trigger event. The following options are supported:

- Until Full: The default behavior is to stop collection when the capture buffer is full.
- Until FCS Error: Stop collection when the first FCS error in a received packet is detected.
- Until Filter: Stop collection when the first packet that matches the specified filter is received.
- Until Payload Error: Stop collection when the first payload error in a received packet is detected.
- Until User Stop: Stop when the user stops.

Packets To Keep

This option control what type of packets to keep. This may help you make the most of the limited capture buffer.

How Much To Keep

This option how much of the captured packets to keep. Using this option will increase the number of packets you can keep in the internal capture buffer. The XenaManager will always report the total length of the packet even if it has been truncated due to this option.

Starting Capture

You can manually start capture on a port by pressing the *Start Capture* button at the top of the panel. If the checkbox next to the button is checked the Global Statistics Start Traffic button will also start capture of the ports in your testbed.

Capture Results

The captured packets will be uploaded from the chassis while capture is ongoing. You can thus inspect them both when the capture is still in progress and when the capture operation has completed.

Results Grid

Each captured packet will be displayed as a row in the *Result Grid*. The following values are reported for each packet:

- Timestamp: The timestamp for when the packet was received relative to capture start.
- Latency: The latency value calculated from the Xena test payload data (not valid for other types of packets)
- IFG: The Inter-Frame Gap compared to the previous packet.
- Source: The SMAC address from the packet.

- Destination: The DMAC address from the packet.
- Protocol: A summary of the decoded packet headers in the packet.
- Full Length: The original length of the captured packet before any optional truncation due to the How Much To Keep option has been performed.
- Captured Length: The actual length of the captured packet after optional truncation.

If you select a packet in the grid a Wireshark-like packet header view will be displayed below the grid where the packet content can be inspected.

Capture Graph

The XenaManager also provides a graphical histogram view of the length or spacing of the captured packets, as well as the latencies.

Saving or Exporting Capture Data

By using the Save Packets button you can save the captured packets to a PCAP or a PCAP-NG (PCAP Next Generation) file.

You can also launch Wireshark directly with the captured packets as an argument by pressing the Launch Wireshark button. This obviously require that Wireshark has been installed on your PC.

5.10 Histograms

This section describes the XenaManager Histogram panel. The panel allow you to configure the histogram settings for the currently selected port and to inspect the result of the histograms data collection.

5.10.1 Overview

Histograms analyze a stream of packets, either at the transmit side or the receive side of a port, and classify them into a number of buckets, counting how many packets go into each bucket.

A histogram is configured with a fixed number of buckets and a value range. The first and last bucket handles all the packets that don't fit within the specified range. All the other buckets each handle a sub-span of the range, determined by the histogram configuration.

Histograms complement the statistics counters function, which just provide aggregate counts, and the capturing function, which provides per-packet information.

5.10.2 Configuration

You can create up to two histograms per port. Usually a single histogram will be sufficient for most uses. You can add a new histogram by pressing the *Add Histogram* button in the toolbar. To delete a histogram you should select the histogram and press the *Remove Histogram* button in the toolbar.

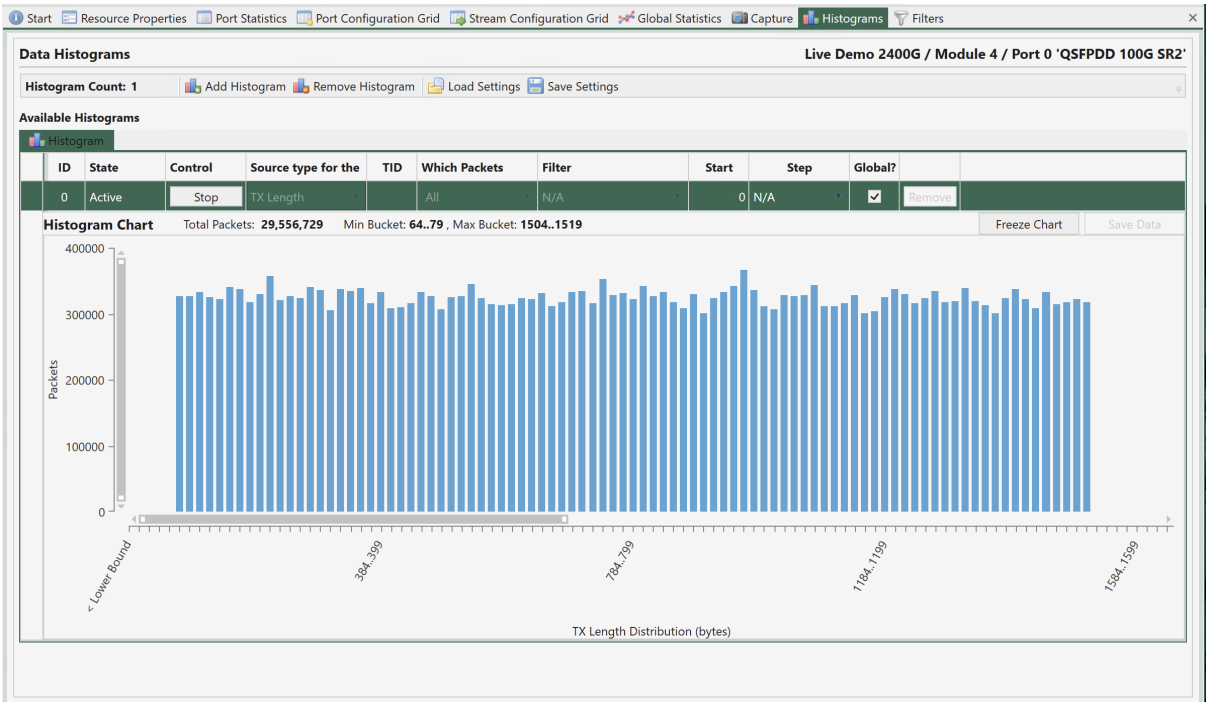


Fig. 5.116: Port histograms

Each histogram is listed as a row in the configuration grid view and has the following configuration properties:

- State: Indicates whether the histogram is currently activated or not.
- Control: Allow you to control the activation state of the histogram.
- Global?: Allow histogram control from *Global Statistics* view.
- Source Type: Determines what type of metric is used as the source for the histogram.
- Which Packets: Control which type of packets are used for the histogram.
- TID: If the Which Packets property is set to Test Payload this field should contain the Test Payload ID (*TID*).
- Filter: If the Which Packets property is set to Filter you should select the filter you want to use here.
- Start: The lowest value in the valid range. Any value lower than this will be placed in the first bucket.
- Step: The span of each bucket.

5.10.3 Histogram Results

Viewing Charts

Once a histogram is active you can view the realtime chart of the collected data by selecting the histogram row. When a histogram is activated all the buckets are empty. As packets are encountered (according to the source) their data is registered and placed in the correct bucket according to the range specification. You can see this progress in the chart.

Hovering the mouse over a particular vertical bar pops up a little window with the information about that bucket as shown in the image below. The Accumulated value indicates the sum of the values up to and including the value in focus.

You can temporarily freeze the chart update by pressing the *Freeze Chart* button. No data will be lost and when you unfreeze the chart it will be updated with all the samples that was collected in the background.

You can use the chart scrollbars to zoom and pan the results as described here.

Saving Results

The bucket counts can also be saved to a CSV text file file for more detailed analysis by pressing the *Save Data* button.

5.11 Gauge/Meter

This section describes the XenaManager Gauge/Meter window. The window allows you to display the layer-2 traffic rate of a port or a stream in a Gauge (or Meter).

5.11.1 Overview

Gauges display the current layer-2 traffic in bit/s for a port or a stream for a quick visual overview of one or more traffic results. The gauge transforms the layer-2 traffic into the visual representation of the gauge and will display the numerical value in the same window. The gauge auto scales to the bit rate of the port carrying the monitored traffic.

5.11.2 Configuration

You can activate gauges when traffic is running. To activate a gauge for a port or a stream, right-click on the port or stream in the *Available Resources* tree in the left side of the XenaManager. You now get a menu with options for the port/stream, including Add Gauge. When you click *Add Gauge*, the gauge window will appear. You can continue to add more gauges to show information for the ports and streams that are relevant to you.

At the top of the gauge window you can see if the gauge shows traffic for a port (Port Mode) or a stream (Stream Mode). You will also find identification of chassis, module, port and stream(s). If you left-click and hold on the top line in the gauge window, you can move it around on your PC screen.

You can resize the gauge window by dragging in the low right corner of the window.

Keyboard Resizing Shortcuts:

- +: Doubles the gauge window size.
- -: Restores the gauge window size.

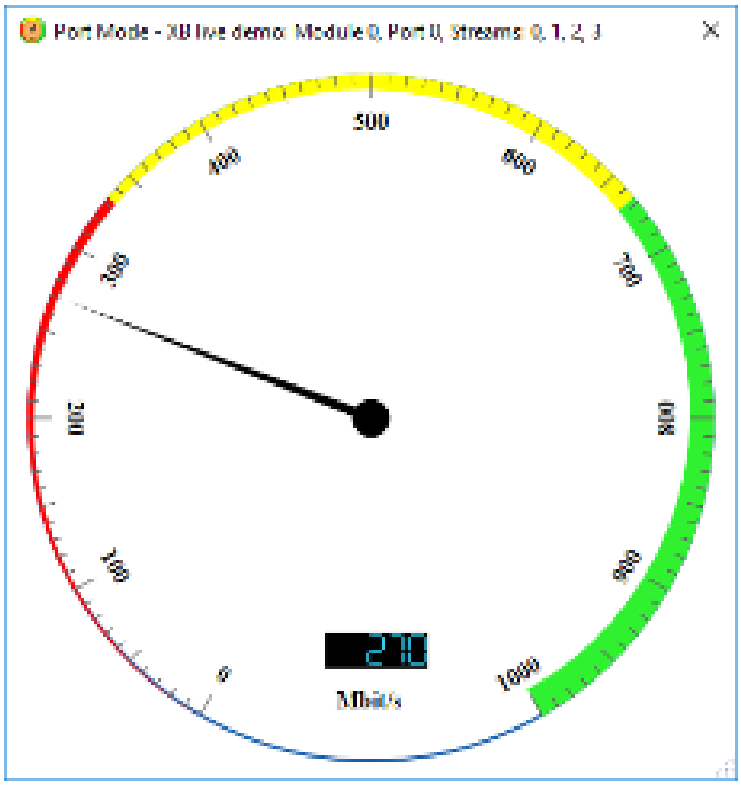


Fig. 5.117: Port gauge/meter

5.11.3 Gauge Menu

If you right-click on the gauge window you get the *Gauge Menu*:

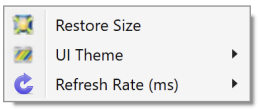


Fig. 5.118: Gauge menu

- Restore Size: Restores the size of the gauge window to its default size
- UI Theme: Allow you to change the appearance of the gauge.
- Refresh Rate (ms): Allow you to select how often the gauge is updated.

To close the gauge window, click on the X in the top left of the gauge window or press `Alt+F4`. The gauge window will automatically close when the traffic it monitors is stopped or the XenaManager is closed.

5.12 Event Log

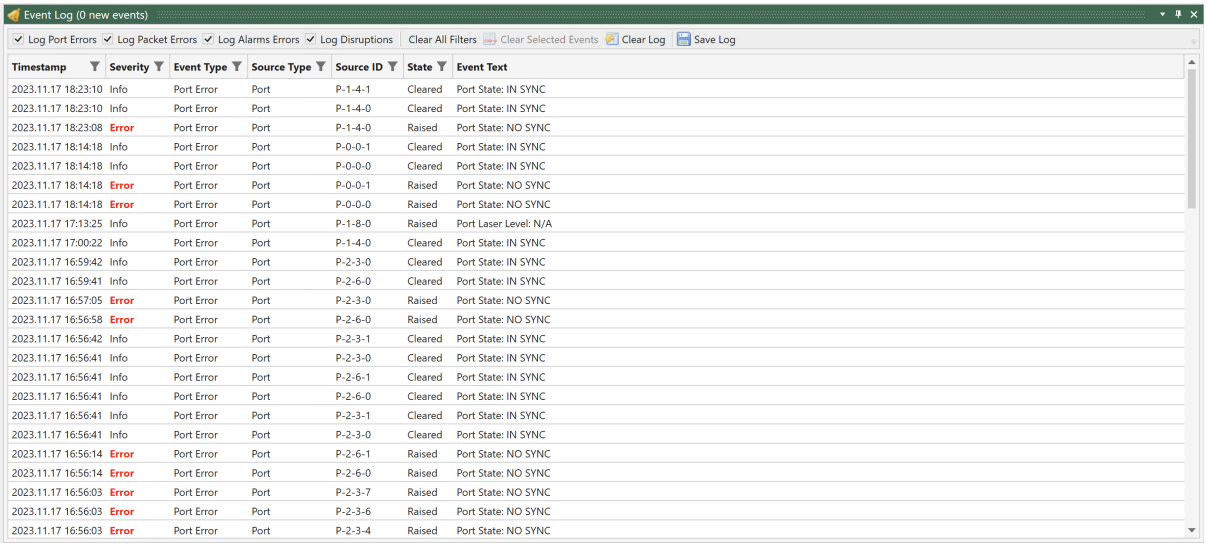
This section describes the XenaManager Event Log panel. This panel can be used to view events from all connected ports.

5.12.1 Event Types

The *Event Log* displays events of the following types:

- Port Errors: Indicate an error that affects the operation of the whole port.
- Packet Errors: Indicate an error in a received packet.
- Disruption: Indicate that the port-level gap monitor has detected a gap in the received data stream.

You can control the logging of each of these types in the Event Log panel toolbar, as indicated in Fig. 5.119.



Timestamp	Severity	Event Type	Source Type	Source ID	State	Event Text
2023.11.17 18:23:10	Info	Port Error	Port	P-1-4-1	Cleared	Port State: IN SYNC
2023.11.17 18:23:10	Info	Port Error	Port	P-1-4-0	Cleared	Port State: IN SYNC
2023.11.17 18:23:08	Error	Port Error	Port	P-1-4-0	Raised	Port State: NO SYNC
2023.11.17 18:14:18	Info	Port Error	Port	P-0-0-1	Cleared	Port State: IN SYNC
2023.11.17 18:14:18	Info	Port Error	Port	P-0-0-0	Cleared	Port State: IN SYNC
2023.11.17 18:14:18	Error	Port Error	Port	P-0-0-1	Raised	Port State: NO SYNC
2023.11.17 18:14:18	Error	Port Error	Port	P-0-0-0	Raised	Port State: NO SYNC
2023.11.17 17:13:25	Info	Port Error	Port	P-1-8-0	Raised	Port Laser Level: N/A
2023.11.17 17:00:22	Info	Port Error	Port	P-1-4-0	Cleared	Port State: IN SYNC
2023.11.17 16:59:42	Info	Port Error	Port	P-2-3-0	Cleared	Port State: IN SYNC
2023.11.17 16:59:41	Info	Port Error	Port	P-2-6-0	Cleared	Port State: IN SYNC
2023.11.17 16:57:05	Error	Port Error	Port	P-2-3-0	Raised	Port State: NO SYNC
2023.11.17 16:56:58	Error	Port Error	Port	P-2-6-0	Raised	Port State: NO SYNC
2023.11.17 16:56:42	Info	Port Error	Port	P-2-3-1	Cleared	Port State: IN SYNC
2023.11.17 16:56:41	Info	Port Error	Port	P-2-3-0	Cleared	Port State: IN SYNC
2023.11.17 16:56:41	Info	Port Error	Port	P-2-6-1	Cleared	Port State: IN SYNC
2023.11.17 16:56:41	Info	Port Error	Port	P-2-6-0	Cleared	Port State: IN SYNC
2023.11.17 16:56:41	Info	Port Error	Port	P-2-3-1	Cleared	Port State: IN SYNC
2023.11.17 16:56:41	Info	Port Error	Port	P-2-3-0	Cleared	Port State: IN SYNC
2023.11.17 16:56:14	Error	Port Error	Port	P-2-6-1	Raised	Port State: NO SYNC
2023.11.17 16:56:14	Error	Port Error	Port	P-2-6-0	Raised	Port State: NO SYNC
2023.11.17 16:56:03	Error	Port Error	Port	P-2-3-7	Raised	Port State: NO SYNC
2023.11.17 16:56:03	Error	Port Error	Port	P-2-3-6	Raised	Port State: NO SYNC
2023.11.17 16:56:03	Error	Port Error	Port	P-2-3-4	Raised	Port State: NO SYNC

Fig. 5.119: Event log

5.12.2 Event States

Some events are raised when a monitored value crosses a certain threshold and cleared when the monitored value falls below the threshold again. This includes for instance the port sync state and the laser Rx level events and the disruption events.

Other events are merely raised when a certain criterion is met. This includes for instance the packet error events. These events are not cleared.

5.12.3 Event Monitoring

For detection of most event types the port needs to be polled continuously. It is only the port sync event that can be detected without polling.

However to decrease the performance impact of too much polling the XenaManager will by default only poll ports that are visible in a panel that requires the polled information. This primary includes the various statistics panels. So if you are currently not viewing e.g. the statistics panel for a given port the port may not be polled.

If you require a given port to be polled regardless of its current visibility you can enable the *Poll Always* property in the *Port Receive Statistics* toolbar in the *Port Statistics* panel.

5.12.4 Event Columns

The Event Log panel contains the following columns:

- Timestamp: The timestamp when the event was detected by the XenaManager. Note: This does not represent the time when the event occurred in the chassis but the time when the event was detected on the PC. The accuracy is thus in the seconds range.
- Source Type: The type of the event source.
- Source ID: A unique identification of the event source.
- State: The event state (see above for details)
- Event Type: The event type (see above for details)
- Event Text: A textual description for the event which may provide more details.

5.12.5 Event Log Management

The event log is not persistent and the content will be cleared when you close down the XenaManager application. You can save the current content of the event log to a CSV text file by clicking the Save Log button in the toolbar. You can also manually clear the event log by clicking the Clear Log button in the toolbar.

5.13 Communication Trace

This section describes the XenaManager Communication Trace panel. This panel show the decoded communication with connected chassis using Xena Management Protocol (XMP), which is Xena's proprietary protocol for administrating chassis.

The panel is mainly used for debugging the communication in case of problems but it can also be used as a help for users writing automation scripts who wants to see how a certain request is formatted.

For details on the XMP requests please refer to the XOA CLI Documentation.

5.13.1 Trace Format

The trace entries are displayed in a standard grid view as shown below. Each request or reply is listed in a separate row in the grid. Requests sent from the XenaManager are shown in green whereas successfully replies from the chassis are shown in blue. Error replies from the chassis are shown in red.

5.13.2 Available Columns

The trace grid view offers the following columns:

- Time: A timestamp for the trace event with millisecond accuracy.
- Dir: The direction (Tx or Rx).
- ChassisName: Name of the chassis

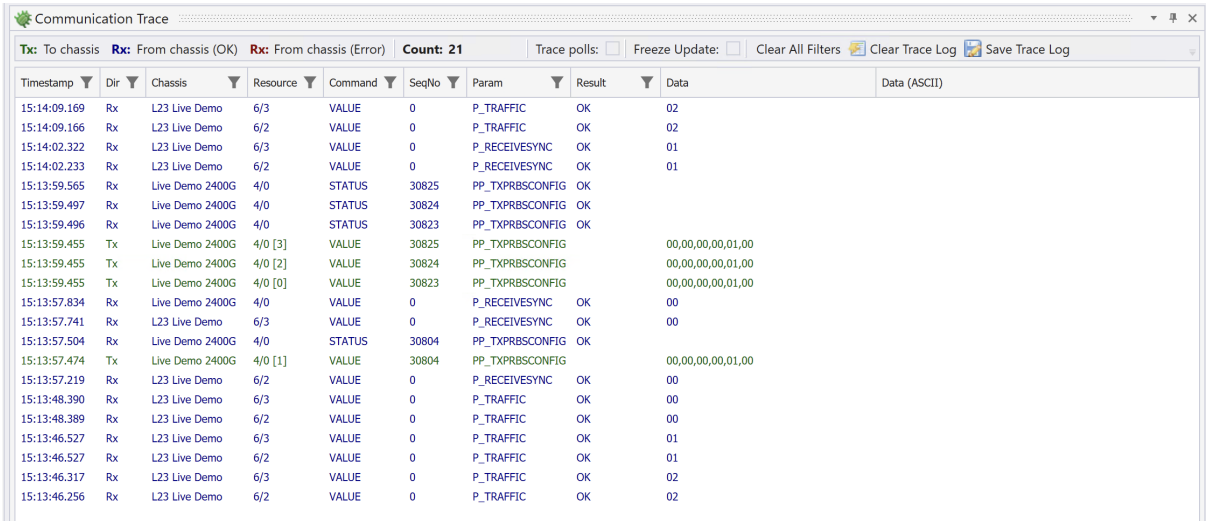


Fig. 5.120: Communication trace

- Target: The request target on the form `<module index>/<port index> [optional stream index]`.
- Command: The XMP command type
- SeqNo: The XMP sequence number.
- Param: The XMP request parameter.
- Arguments: Any arguments as a list of hexadecimal numbers
- Result: The result code for a reply.

5.13.3 Filtering

Several column provide support for filtering the displayed trace messages. This is indicated by the funnel icon in the column header. If you click this icon you can select how you want to filter the displayed trace messages.

5.13.4 Performance Impact

Having the panel open at all times is usually not recommended. If a lot of traffic is going to and from the chassis this may impact the performance of the PC, especially during polling.

5.14 Logging and Reporting

Note

Chimera module and its ports for network impairment measurement also offer support for logging and reporting functionalities.

5.14.1 Overview

The Logging and Reporting function allows you to periodically poll statistics counters for all ports in a testbed and log those counters to a CSV or XML file.

Port Scope

The logging function works on all ports in a given testbed. You can enable logging on multiple testbeds at the same time.

Configuration Panel

This function is handled by the *Logging and Reporting* panel as shown in Fig. 5.121. This panel is by default shown in the bottom part as one of the auto-hide panels.

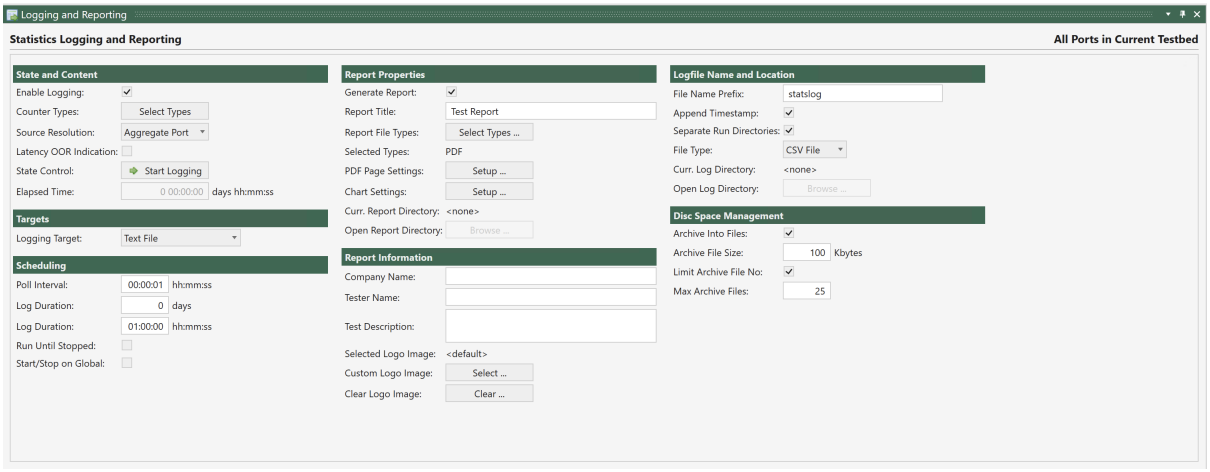


Fig. 5.121: Logging and Reporting and Report tab

Logging Configuration

The *Logging and Reporting* panel provides the following configuration options:

Table 5.54: Logging Configuration Options

Option	Explanation
Enable Logging	Selects whether this Logging and Reporting definition is enabled or not.
Coun Types	Pressing this button will enable you to select which counters to include in the log. See *Counter Types* for the detailed description of the available counter types.
State Control	This button will either start or stop a logging session.
Elaps Time	Shows the elapsed time for an active logging session.
Poll Interval	Specifies the interval between polls. The default value is 1 second. The minimum value is 1 second.
Log Duration	Specifies the total desired duration of a collection period. You can specify this duration as a number of days + a `hour::minute::second` option. The total duration could thus for instance be 2 days, 4 hours and 30 minutes. This option is only valid if the Run Until Stopped option is not selected.
Run Until Stopp	If this option is selected the collection will run until it is manually stopped.
File Name Prefix	This string will be used as the prefix for the logging filenames.
Append Times tamp	If selected a timestamp on the form `YYYYMMDD_HHMMSS` will be appended to the filename.
Separate Run Directories	All log files will be located under the <ProgramData>XenaXenaIntegratorPortLog directory. If this option is selected the log files for different logging runs will be placed in separate subdirectories under this master directory. The subdirectory name will be a timestamp on the form `YYYYMMDD_HHMMSS`. If the option is not selected all logging files will be placed directly in the PortLog directory described above.
File Type	This determines the format of the log ile. You can select between CSV (Comma Separated Value) or XML format.
Archi Large Files	Selecting this option will make the logging function save the current log file to an archive file and start a new log file when the log file reaches a certain size. The archive files will be named <`prefix`>.<`archive no`>.<`extension`>. The archive numbering will be sequential so that the file with the highest number is the most recent archive file. The currently active log file will still be called <`prefix`>.<`extension`>. Archive File Size: The file size where archiving should take place.
Limit	If this option is selected the application will limit the number of archive files for a single logging run. This can be used for long running logging tasks to prevent the

Counter Types

The following counter types are available:

- Transmit Rate L1 (Bit/s)
- Transmit Rate (Bit/s)
- Transmit Rate (Byte/s)
- Transmit Rate (Fps)
- Transmitted Bytes
- Transmitted Frames
- Receive Rate L1 (Bit/s)
- Receive Rate (Bit/s)
- Receive Rate (Byte/s)
- Receive Rate (Fps)
- Received Bytes
- Received Frames
- RX Oversize Packets
- RX Undersize Packets
- RX Jabber Packets
- Transmitted Non-Payload Bytes
- Transmitted Non-Payload Frames
- Received Non-Payload Bytes
- Received Non-Payload Frames
- Received FCS Errors
- Rx Sequence Errors
- Rx Packet Loss Ratio
- Rx Sequence Misorders
- Rx Payload Errors
- Rx Bit Error Rate (aggregated)
- Rx Bit Error Rate (current)
- Latency - 1 sec. avg (ns)
- Latency - 1 sec. min (ns)
- Latency - 1 sec. max (ns)
- Latency - aggr.avg (ns)

- Latency - aggr.min (ns)
- Latency - aggr.max (ns)
- Jitter - 1 sec. avg (ns)
- Jitter - 1 sec. min (ns)
- Jitter - 1 sec. max (ns)
- Jitter - aggr.avg (ns)
- Jitter - aggr.min (ns)
- Jitter - aggr.max (ns)
- Transmitted ARP Requests
- Transmitted ARP Replies
- Transmitted PING Requests
- Transmitted PING Replies
- Injected FCS Errors
- Injected SEQ Errors
- Injected MIS Errors
- Injected Integ. Errors
- Injected TID Errors
- Transmitted MAC Training Frames
- Transmitted IGMP Join
- Received ARP Requests
- Received ARP Replies
- Received PING Requests
- Received PING Replies
- Calculated Gap Count
- Calculated Gap Duration
- Received PAUSE Frames
- Received PFC Frames
- Received PFC Quanta CoS 0
- Received PFC Quanta CoS 1
- Received PFC Quanta CoS 2
- Received PFC Quanta CoS 3
- Received PFC Quanta CoS 4

- Received PFC Quanta CoS 5
- Received PFC Quanta CoS 6
- Received PFC Quanta CoS 7
- Chimera Total Packet Drop
- Chimera Total Packet Drop Ratio (ppm)
- Chimera Programmed Drop
- Chimera Programmed Drop Ratio (ppm)
- Chimera Bandwidth Drop
- Chimera Bandwidth Drop Ratio (ppm)
- Chimera Other Drops
- Chimera Other Drops Ratio (ppm)
- Chimera Misordering
- Chimera Misordering Ratio (ppm)
- Chimera Duplication
- Chimera Duplication Ratio (ppm)
- Chimera Corruption
- Chimera Corruption Ratio (ppm)
- Chimera FCS Checksum
- Chimera FCS Checksum Ratio (ppm)
- Chimera IP Checksum
- Chimera IP Checksum Ratio (ppm)
- Chimera UDP Checksum
- Chimera UDP Checksum Ratio (ppm)
- Chimera TCP Checksum
- Chimera TCP Checksum Ratio (ppm)
- TSN offset pre-servo average last second.
- TSN offset post-servo average last second.
- TSN Rx interarrival average last second.
- TSN P delay average last second.
- TSN Neighbor Rate Ratio average last second.
- Received Lane Pre-FEC BER
- Total uncorrected FEC blocks count

- Total corrected FEC symbols count
- Estimated received Pre-FEC BER
- Estimated received Post-FEC BER
- Received FEC blocks stats
- Received FEC blocks ratio stats

Importing Legacy XenaIntegrator Configurations

It is possible to import a legacy XenaIntegrator Port Logging Definition as a new XenaManager testbed. Since the XenaIntegrator Port Logging Definition contains a definition of the ports which will provide the logging counters the import process will automatically perform the following steps:

- Check if the Xena chassis defined in the legacy configuration are already defined in the XenaManager configuration. If not, the necessary chassis definition will be created.
- Create a new testbed with the name `Testbed XI: <definition label>` where `<definition label>` is the name originally used for the Port Logging Definition in XenaIntegrator.
- Add the defined logging ports to the new testbed.
- Migrate the other logging configuration to the new testbed.

To import a legacy XenaIntegrator Port Logging Definition simply click the Import XI LogCfg button in the Operations menu and select the XenaIntegrator configuration file you want to import.

5.14.2 Controlling Logging State

Starting and Stopping Logging

As stated above the State Control button allows you to start or stop the logging process. While the logging is in progress it will not be possible to change any configuration parameters.

Monitoring Progress

While the logging is in progress the *Elapsed Time* counter will increment showing the total duration of the logging process. The Current Log Directory field will show the full path to the current logging directory. Clicking the *Open Log Directory* button will launch a Windows Explorer in this directory.

5.14.3 Output Formats

CSV File Format

The CSV file will contain a number of lines. Each line will represent all enabled logging data for one port for a single poll. A line will have the following format:

```
<Timestamp>, <Port ID>, { <CounterValue>, }*
```

Table 5.55: L2 Counter Types

Field	Explanation
Timestamp	The data and time for the logged data line on the form YYYYMMDD-HHMMSS.
Port ID	The port identification on the form P-<chassis>-<module>-<port>.
CounterValue	The counter value. All values are expressed as a decimal number.

The file will also contain a header row describing the selected counter types.

XML File Format

The XML file format will be similar to the following example:

```
<?xml version="1.0" encoding="utf-8"?>
<!--XenaIntegrator Statistics Counters-->
<PollSamples>
    <SelectedCounterTypes Values="TxBps,TxFps,TxBytes,TxFrames,RxBps,
↪RxFps,RxBytes,RxFrames" />
    <Element Timestamp="20130331-174155" Type="Notification" Text="Log␣
↪initialized" />
    <Element Timestamp="20130331-174156" Type="Sample" Port="P-0-10-2"␣
↪Values="0,0,0,0,0,0,2.91E+06,4.3E+04" />
    <Element Timestamp="20130331-174156" Type="Sample" Port="P-0-10-3"␣
↪Values="0,0,0,0,0,0,1.51E+05,581" />

    <etc>
</PollSamples>
```

All data is kept under a root tag called <PollSamples>.

The first node is called <SelectedCounterTypes>. The Value attribute describes the selected counter types in comma-separated format.

Each poll sample is represented using the <Element> node tag and has the Type attribute set to Sample. The Values attribute contains the sample values in the same order as is given by the <SelectedCounterTypes> tag.

<Element> nodes with Type = Notification represents notification messages.

5.15 Statistics Charting

This section describes the Statistics Charting panel which can be used to view a real-time chart of various statistics counter values from selected streams.

The section describes the new version of the charting panel introduced in XenaManager version 1.43. The original simpler charting panel is no longer supported.

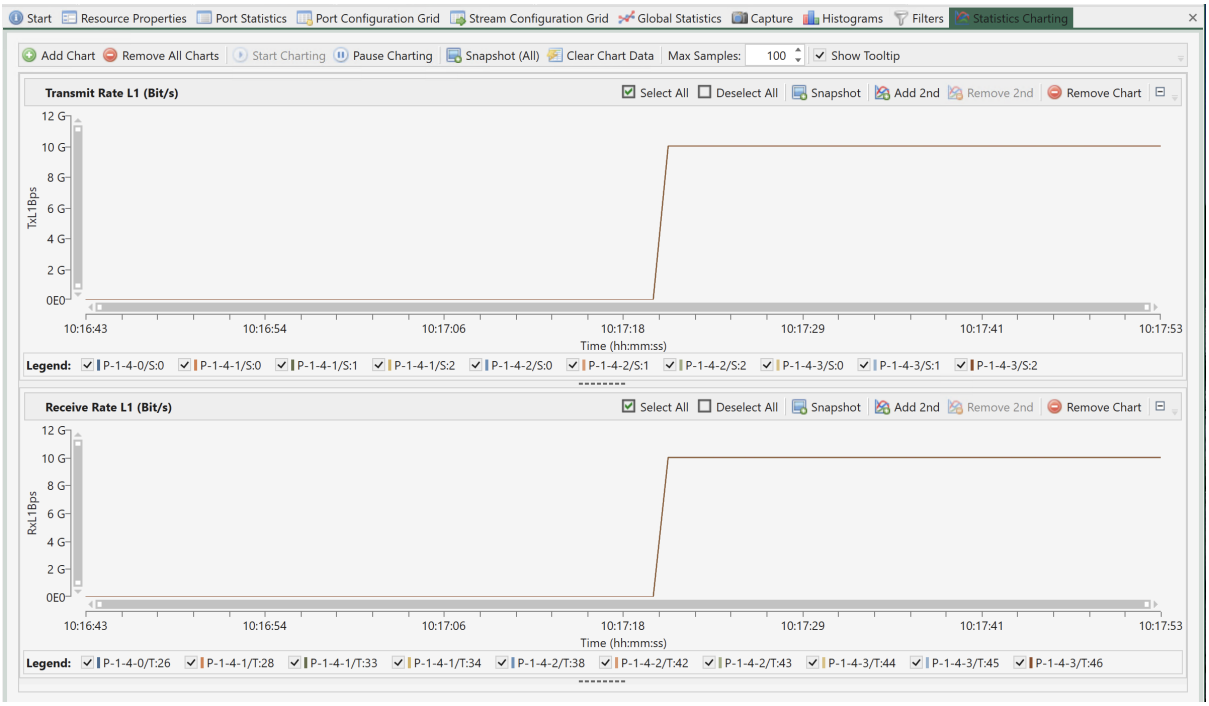


Fig. 5.122: Statistics Charting

5.15.1 Overview

Charted Parameters

The charting panel allow you to view real-time charts of a number of monitored parameters. You can define multiple charts within the chart panel which can each display separate parameters. Each panel can optionally display two different parameters where each parameter then is associated with its own Y-axis (left or right).

Selecting Data Sources

The charting panel will always be associated with the ports and streams in the current testbed. It is possible to select exactly which streams are used by each individual panel.

Counter Types

It is possible to chart all the counter types available in the statistics panels.

Port Polling Aspects

When charting receive-side counters it is important to ensure that the port(s) you expect the packets to arrive on are polled for counters.

To decrease the performance impact of too much polling the XenaManager will by default only poll ports that are visible in a panel that requires the polled information. This primary includes the various statistics panels. So if you are currently not viewing e.g. the statistics panel for a given port the port may not be polled.

When you add a stream to the charting function the XenaManager knows which port this stream is defined on and will ensure that any such port is polled. This ensures that any transmit-side

counters will always be polled. But the XenaManager cannot know which port(s) the packets sent by the streams actually arrive on. You need to help the XenaManager by either ensuring that you are viewing the statistics panel for the receiving port or by enabling the the Poll Always property in the Port Receive Statistics toolbar in the Port Statistics panel.

5.15.2 Charting Details

> **Note**
>
> Chimera module and its ports for network impairment measurement also offer support for statistics charting functionalities.

This section explains how to configure and use the charting function.

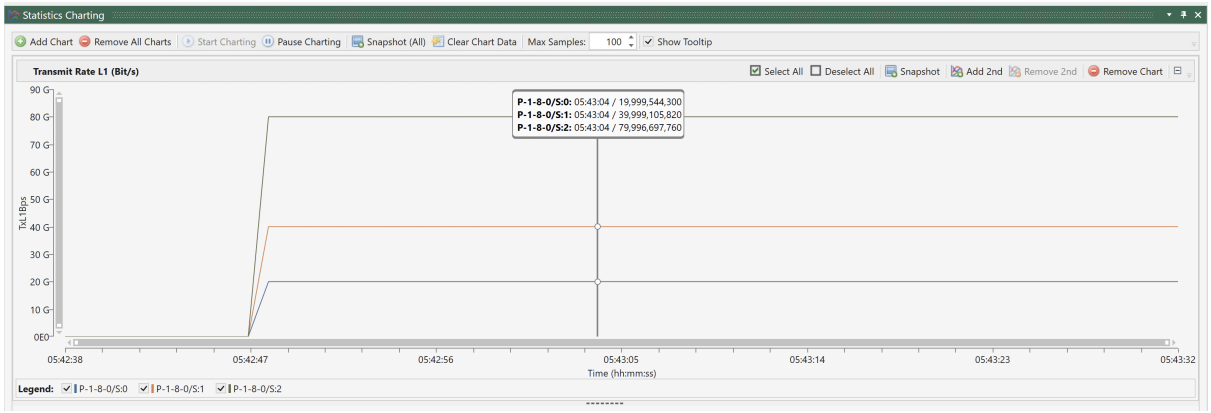


Fig. 5.123: Charting details

5.15.3 Chart Control

Add and Remove Charts

You can add any number of charts to the chart panel. The defined charts will be stacked vertically on top of each other. To add a new chart simply click the Add Chart button in the top toolbar. You can now select the parameter type which you want to be charted from a dialog.

To remove a chart simply click the Remove Chart button in the chart toolbar. If you want to remove all charts you can also click the Remove All Charts button in the top toolbar.

Start and Pause Charting

When you have added the chart(s) you want to use you need to start the charting function by clicking the Start Charting button in the toolbar. To pause the charting function you can click the Pause Charting button. You can re-start the chart by clicking the Start Charting button again.

The data will continue to be collected in the background so the chart will be fully updated with the collected data once you resume charting.

Add a Second Parameter

When you add a chart the selected chart parameter will by default be associated with the left Y-axis. It is possible to add a second parameter to a chart by clicking the *Add 2nd* button in the chart toolbar. The second parameter will be associated with the right Y-axis.

Selecting Stream Sources

By default all streams in your current testbed will be part of the charts. The streams are shown in the legend below each chart.

You can select which streams are part of a chart by checking or unchecking the checkbox in front of each stream in the legend. You can also control the state for all streams by using the *Select All* and *Deselect All* buttons in the panel toolbar.

5.15.4 Visual Aspects

Controlling Chart Size and Visibility

The size of each chart can be controlled by holding and dragging the dotted handle at the bottom of each sub-chart. It is also possible to control the visibility of a chart completely by clicking the little *plus/minus* icon in the right side of the chart header.

Chart Sample Span

The number of samples in the chart is determined by the Max Samples property in the panel toolbar. Once the total number of samples in the chart has reached this number older samples will be dropped from the chart when new samples are added.

Controlling Tooltip

By default a rather large tooltip with information about the plot points under the mouse will be shown when you hover the mouse over the chart. You can disable this function in the panel toolbar.

Zoom and Pan

You can use the chart scrollbars to zoom and pan the results as described on this page.

Taking Snapshots

You can grab a snapshot of the charts by using one of the Snapshot buttons. This action will generate an image and copy that to the Windows clipboard. You can then paste it into your favorite reporting tool, such as Word or Excel.

5.16 Stream Scheduler

This section describes the Stream Scheduler panel which can be used to build a series of actions based on existing streams in the current testbed.

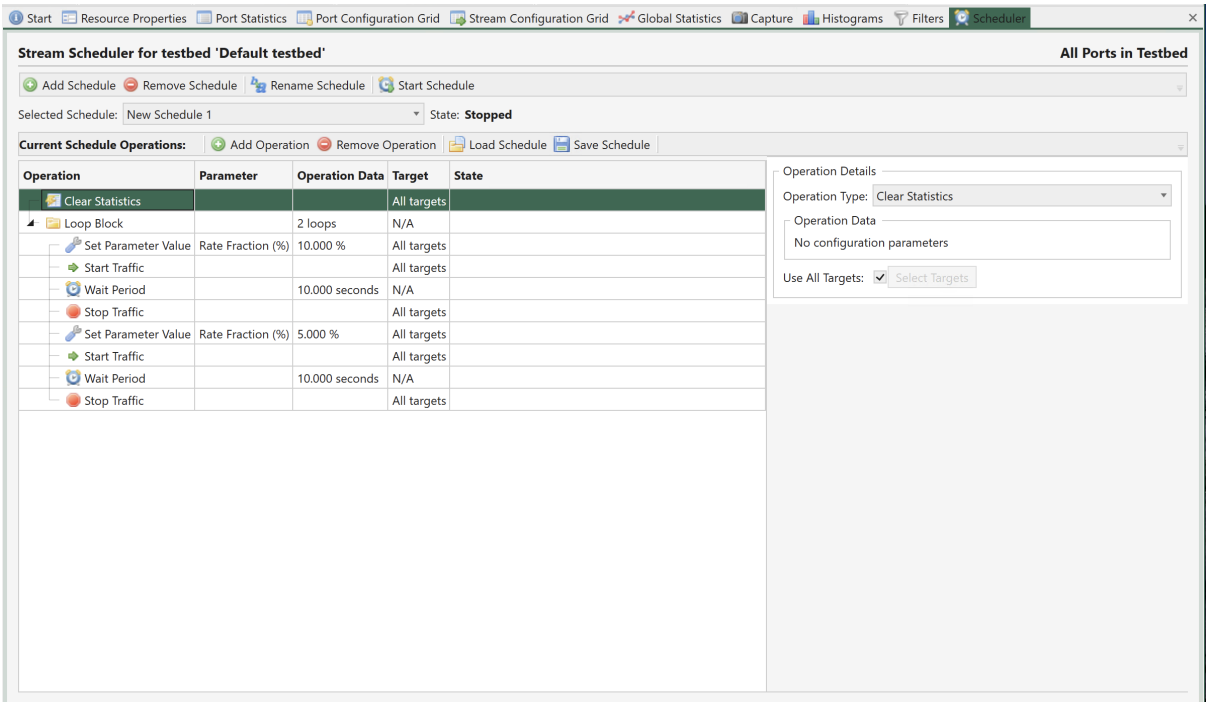


Fig. 5.124: Stream Scheduler

5.16.1 Overview

The Stream Scheduler works closely together with the currently selected testbed and works exclusively with the streams defined on the used ports.

Schedules

Each testbed can contain several schedules. A schedule is simply a collection of operations that will be executed sequentially (although with some looping support as described below).

Operations

As stated above a schedule is basically a collection of operations that will affect the traffic generation. The following operations are supported:

Table 5.56: Scheduler Operations

Operation Name	Target Type Explanation	Explanation
Set Parameter Value	Port or stream	Set the value of a supported stream or port parameter, such as a stream rate.
Wait Period	None	Pause the scheduling for a specified number of seconds, typically to let the traffic run for a specified period of time.
Start Traffic	Port	Start the traffic on selected port(s).
Stop Traffic	Port	Stop the traffic on selected port(s).
Clear Statistics	Port	Clear all statistics counters on all ports used in the current testbed.
Loop Block	None	Enable specifying a block of operations that can be repeated for a specified number of times.
Enable Stream	Stream	Enable selected stream(s).
Disable Stream	Stream	Disable selected stream(s).
Suspend Stream	Stream	Suspend the selected stream(s).
Custom Port Command	Port	Send a custom command to the port(s) selected as targets for the command. This command can be any port-level script command. See this link for details.
Custom Stream Command	Stream	Send a custom command to the stream(s) selected as targets for the command. This command can be any stream-level script command. See this link for details.

Note

Note on custom commands: The scheduler will perform a certain level of consistency checking on the normal commands but it will not be able to perform any consistency check on any custom commands.

Targets

Some operations can be performed on selected targets, which are either streams or ports. These operations can either apply to all valid targets or you can select exactly which targets you want the operation to operate on.

Valid targets are ports included in the current testbed or streams defined on those ports. Certain operations, such as the Wait Period operation, are not associated with any specific targets as they apply to the schedule as a whole.

Common Scenarios

This section explains how to perform common schedule operations.

Creating Simple Schedule

First you should setup a simple configuration consisting of two ports, each with a single stream paired to each other (you can use the Stream Wizard for this). Then you can perform the following actions in order to define a simple schedule for your streams:

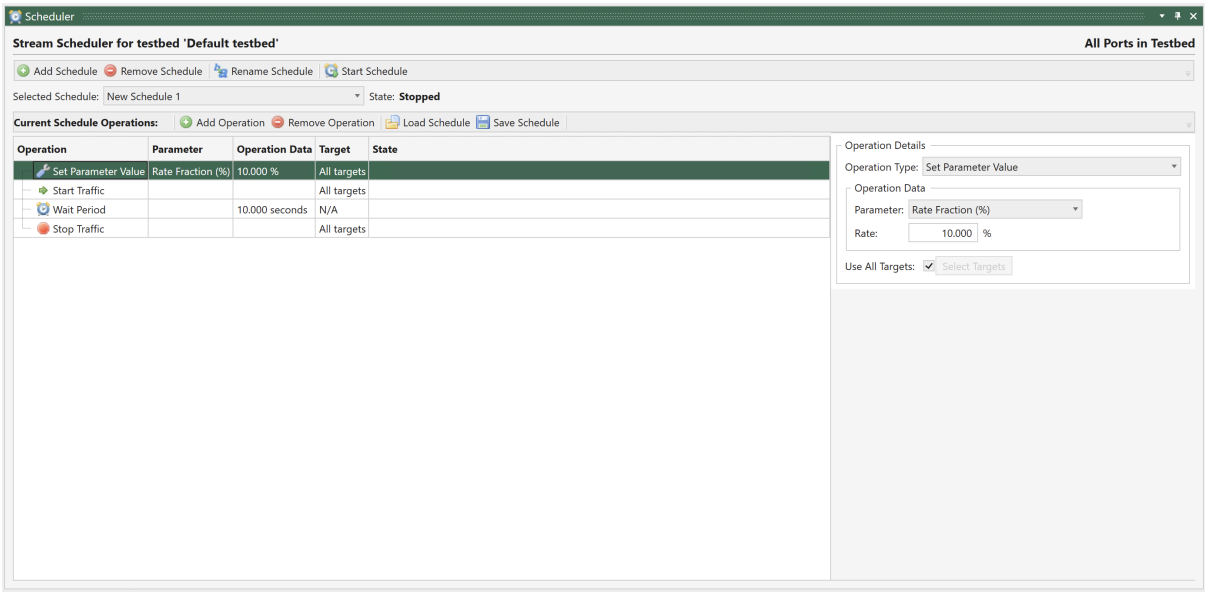


Fig. 5.125: Create simple schedule

1. Bring up the Scheduler panel by selecting it in the lower panel strip. Push the little "paper-pin" in the upper right corner to prevent it from auto-hiding.
2. Add a new schedule by clicking the Add Schedule in the upper panel toolbar.
3. Click the Add Operation button in the schedule operations toolbar and select the Clear Statistics operation in the list.
4. Also add the following operations in the specified order:
   - Set Parameter Value
   - Start Traffic
   - Wait Period
   - Stop Traffic
5. You should just use the default value for each operation for now.
6. Press the Start Schedule button in the upper panel toolbar. The schedule will now perform the specified operations and stop after that.

Changing Traffic Rate

This section explains how to change the traffic rate of the streams after a while. Your streams will start with a 10% rate but after 10 seconds their rate will drop to 5%. The section will extend the schedule defined in the last section.

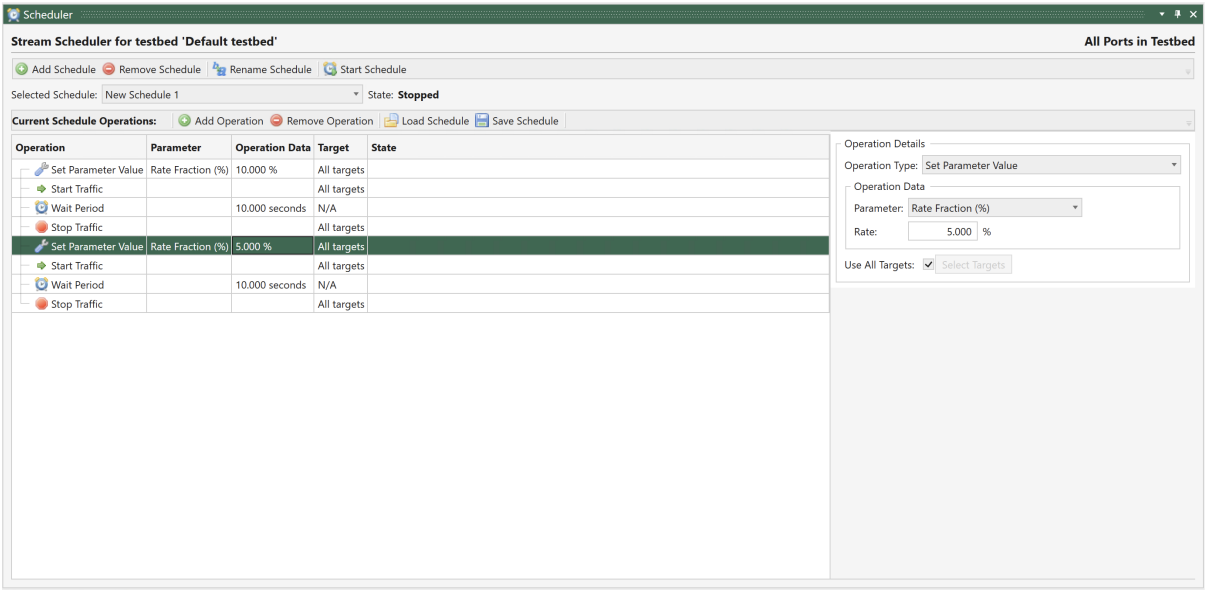


Fig. 5.126: Change traffic rate

Perform the following actions:

1. Select the last *Stop Traffic* operation and insert these additional operations after it:
   - Set Parameter Value
   - Start Traffic
   - Wait Period
   - Stop Traffic
2. We need to stop the traffic while changing the rate value as the Xena tester does not support rate changes while the traffic is running.
3. Select the second *Set Parameter Value* operation and change the *Rate* value to 5% as shown in the image to the right.
4. Press the Start Schedule button in the upper panel toolbar. The schedule will now again perform the specified operations and stop after that. If you want to follow the progress of the rate you can use the Statistics Charting panel for that.

Changing Operations Order

This section explains how you can insert a new operation and move it to the desired location. Perform the following actions:

1. Add a single Clear Statistics operation to the end of the list.

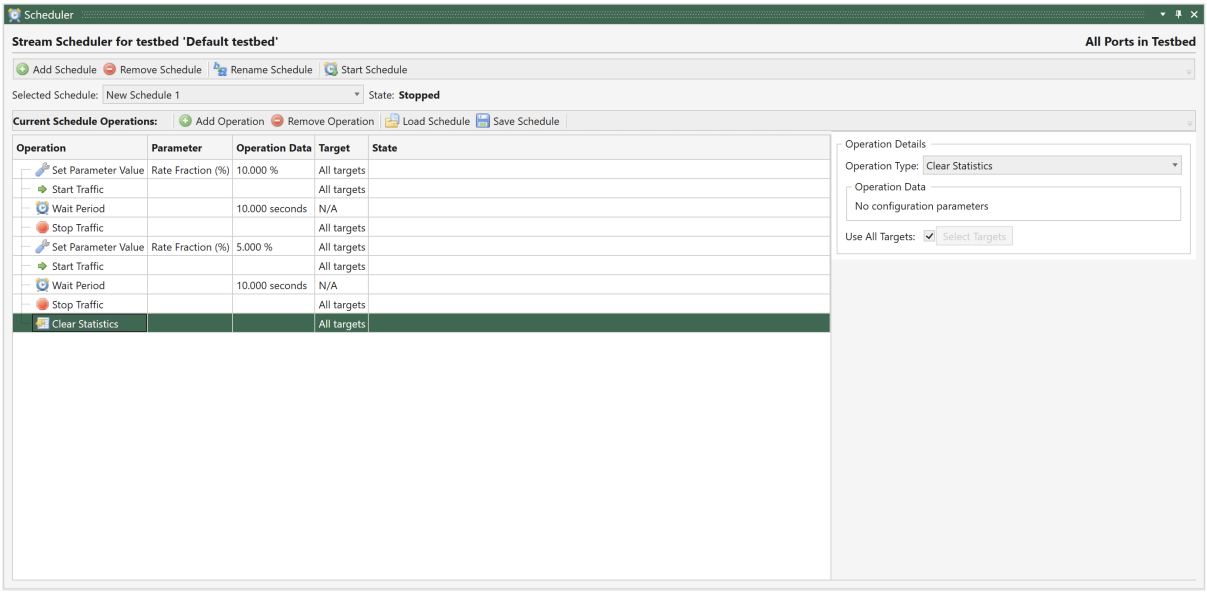


Fig. 5.127: Changing operation order

2. Drag the new operation to the top of the list until you see a guideline on top of the uppermost operation (see screenshot).
3. Drop the operation at the new location.
4. Now all statistics counters will be cleared before traffic is started for the first time.

Adding Loop Section

It is possible to repeat a group of operations for a specified number of times by adding a Loop Block operation. This operation can contain a number of other operations which will be executed sequentially the specified number of times.

Perform the following actions to add a loop block with a repeat count of 2 and to move most of your existing operations into it:

1. Add a *Loop Block* operation to the end of the list.
2. Using the mouse drag and move the loop operation just below the top-most clear operation.
3. Drag the operation just below the loop block on top of the loop block so that it is shown indented compared to the loop operation.
4. Drag each of the other operations to the bottom of the previous operation as shown in the screenshot until they are all indented under the loop block.
5. Start the schedule and observe that all the operations in the loop block are executed twice.

Note

Loop blocks can be nested so that one loop block can contain another loop block.

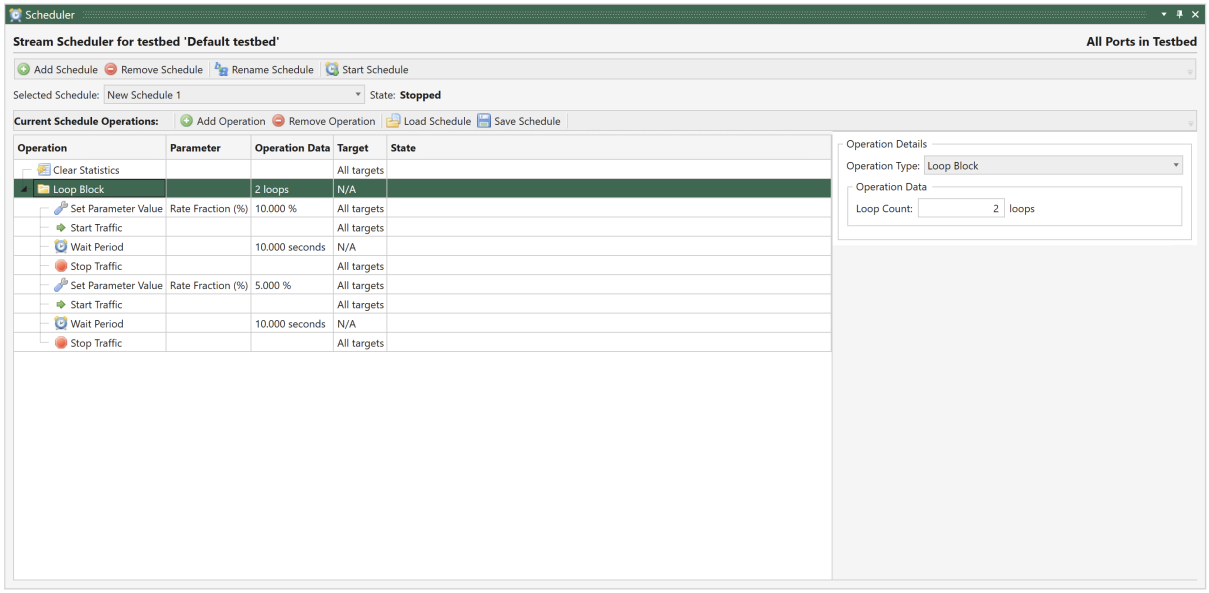


Fig. 5.128: Add loop section

5.17 Stream Wizard

This section describes the XenaManager Stream Wizard panel.

5.17.1 Overview

The stream wizard will help you generate a potentially large number of connected streams based on a set of defined stream templates for a given topology.

With the Stream Wizard you can:

- Define persisted port properties so that ports are setup in a predictable way every time you run the wizard.
- Define stream templates to ensure common setup of actual stream instances.
- Define multiple streams per port to allow for different protocol header and rate setup.
- Ensure that source and destination addresses in the protocol headers are set correctly.
- Validate the whole configuration before stream creation.

The Stream Wizard is closely integrated with the Testbed concept and will operate on the ports you have included for use in your current testbed. You can thus only have one wizard configuration per testbed.

The Stream Wizard is available in XenaManager version 1.10 and newer.

5.17.2 Getting Started

Wizard Panel

The Stream Wizard is controlled through the Stream Wizard panel which is initially located in the lower hidden tab panel as shown in Fig. 5.129 below.

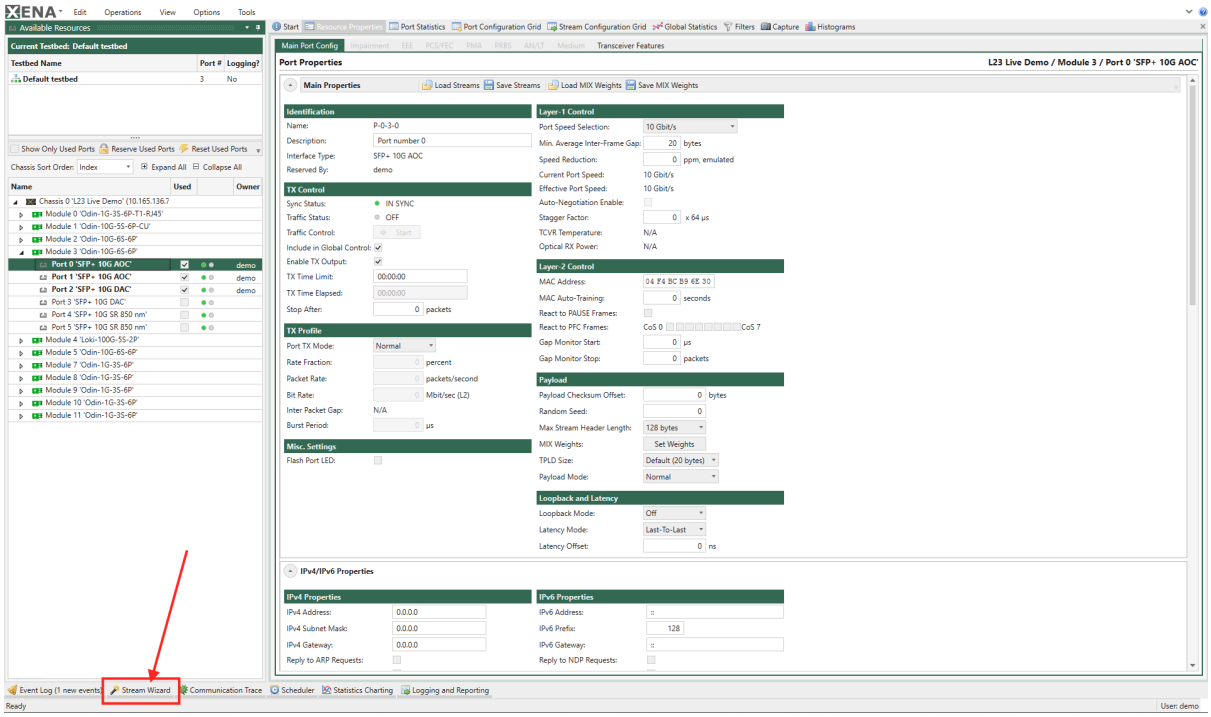


Fig. 5.129: Location of Stream Wizard tab

When you regularly work with the wizard you may want to pin the panel and move it to the main tabbed part of the work area for easier access.

Step-by-Step Configuration

To define and create a set of streams on your ports please follow these steps:

1. Optionally create a new testbed.
2. Include a set of ports in the testbed. You don't have to reserve the ports as this is handled by the wizard.
3. Set the desired topology in the wizard *Topology* panel.
4. Set group for the ports and their peers.

- Set the value for the desired port properties in the wizard *Ports* panel. Enable the properties you want to configure using *Select Port Property Types*.
- Specify the number of desired streams per port in the wizard *Streams* panel.
- Also set the value for the desired stream properties. Enable the properties you want to configure using *Select Stream Property Types*.
- Once you are ready you can press the *Create Streams* button to generate the streams.
- You can repeat this cycle as many times as you want if you need to change parts of the wizard configuration.

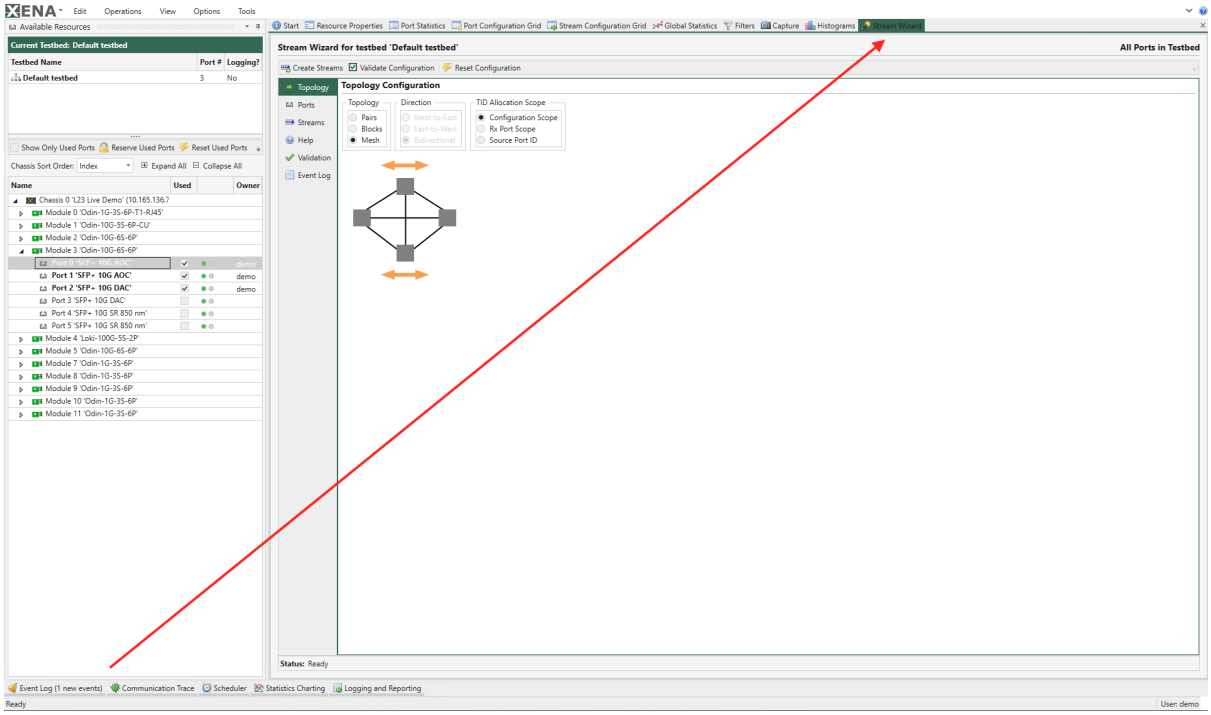


Fig. 5.130: Pinned Stream Wizard

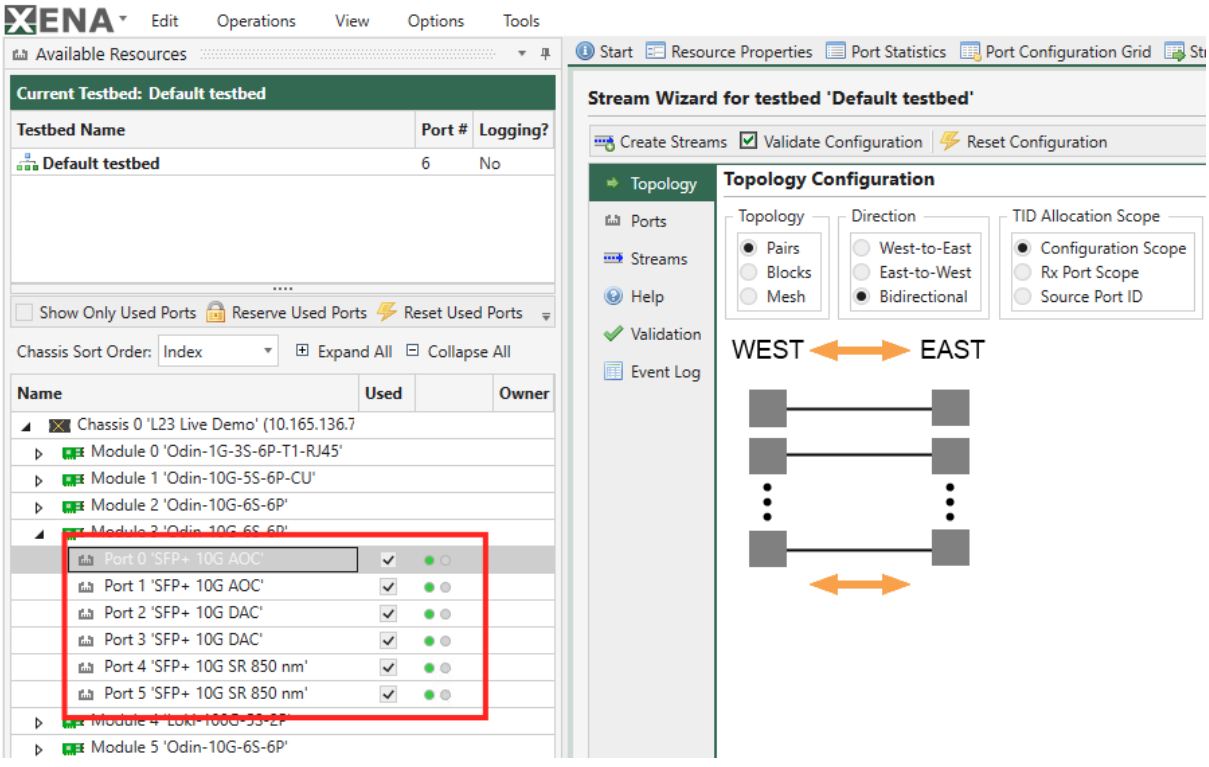


Fig. 5.131: Step 2: Select ports

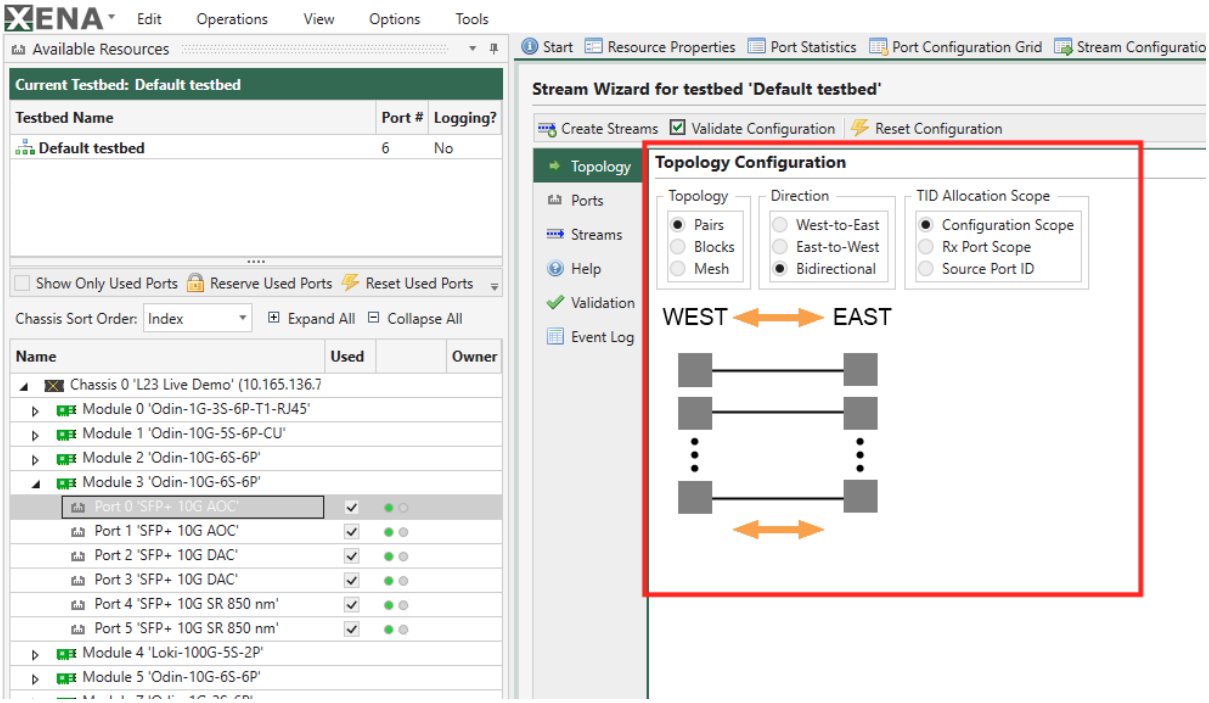


Fig. 5.132: Step 3: Select topology

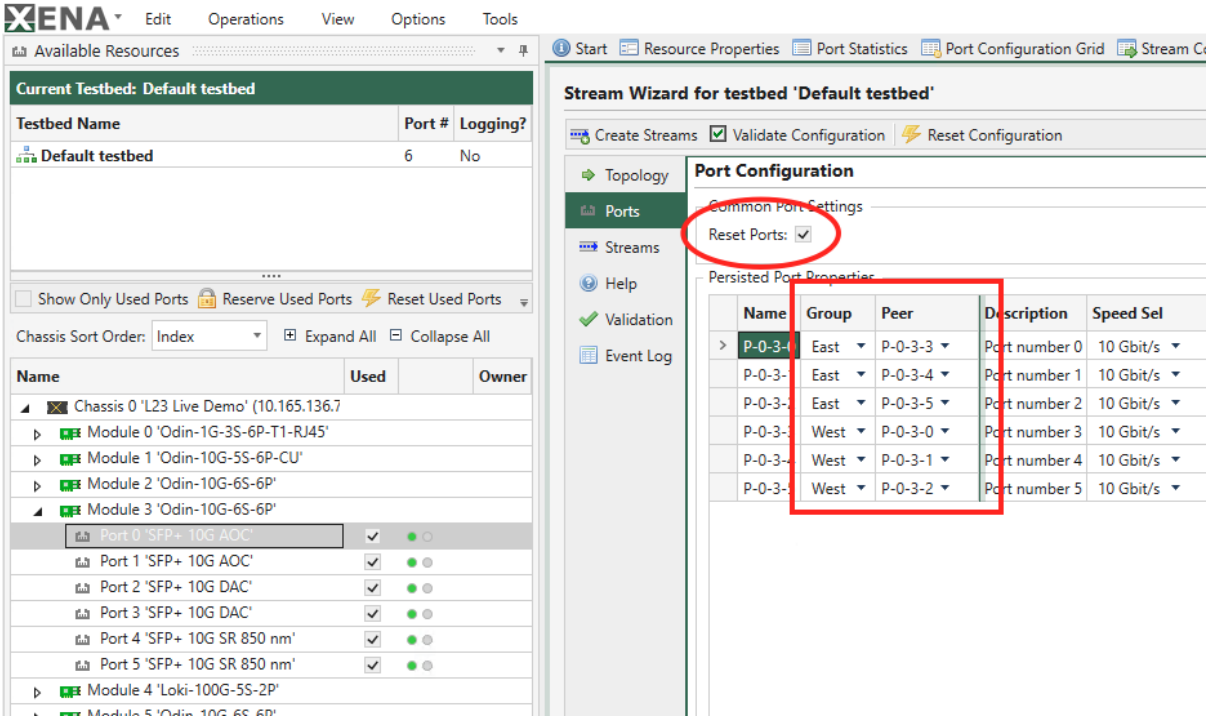


Fig. 5.133: Step 4: Set group and peer

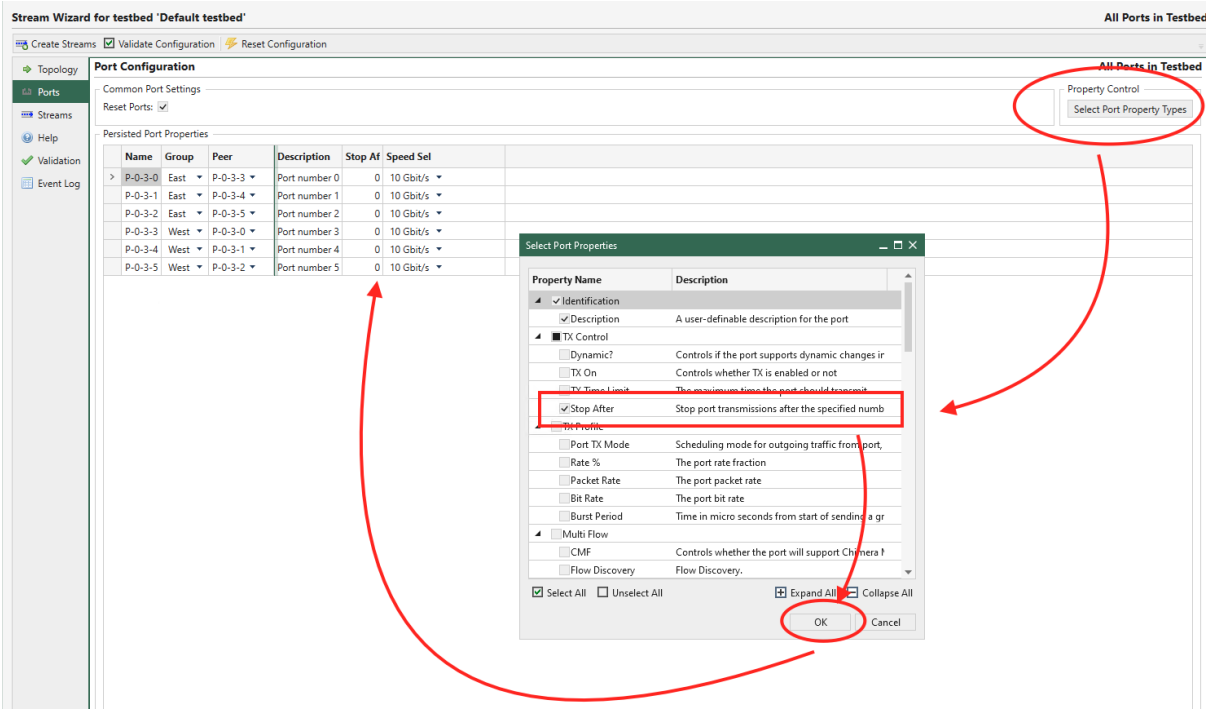


Fig. 5.134: Step 5: Set port properties

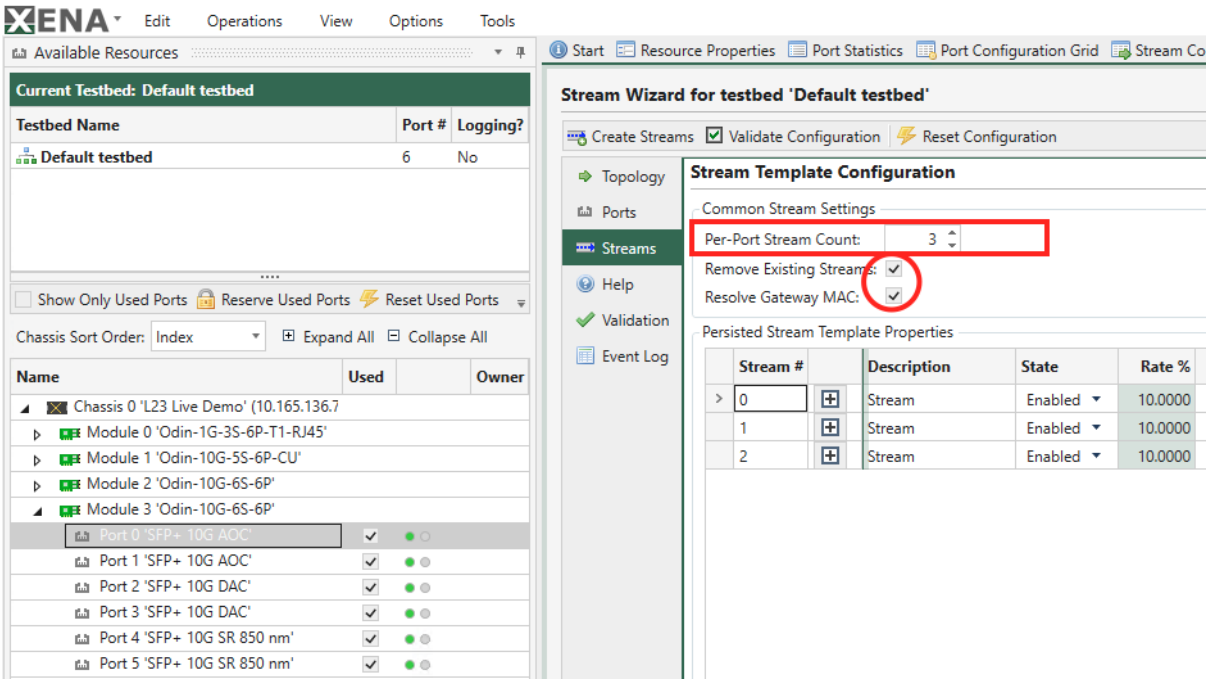


Fig. 5.135: Step 6: Set stream count per port

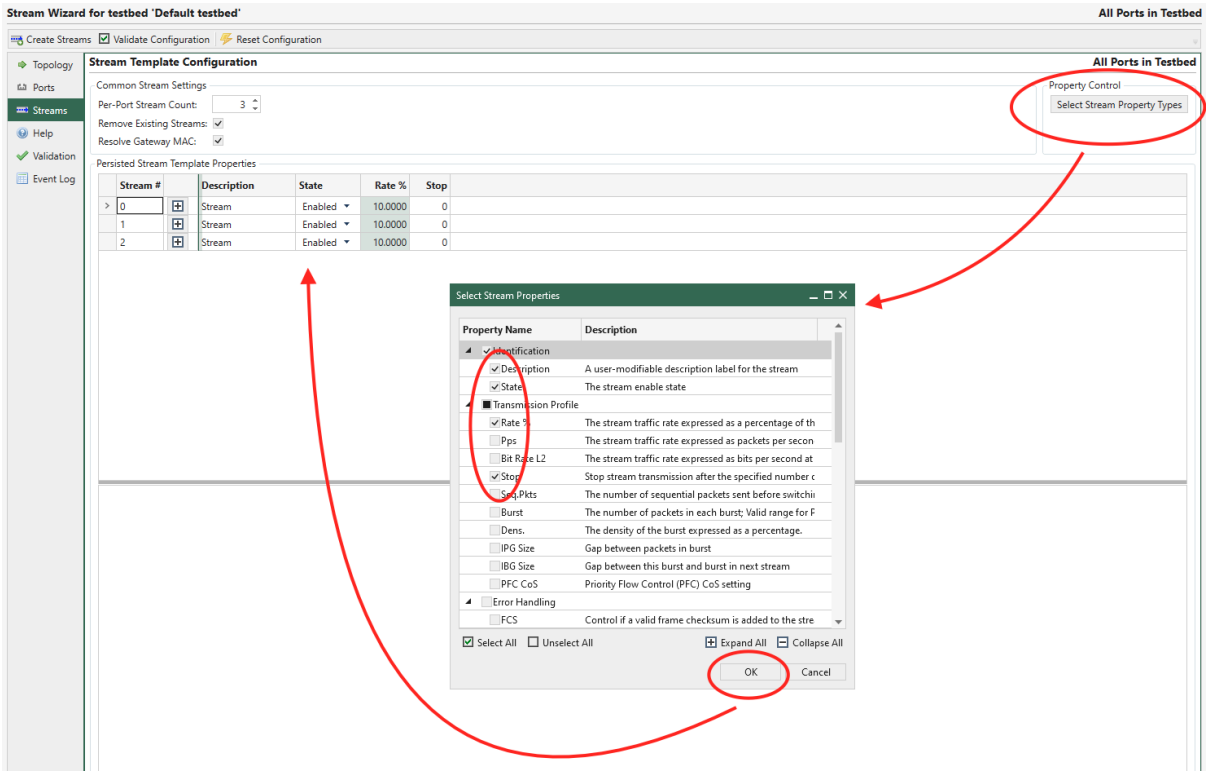


Fig. 5.136: Step 7: Set stream properties

5.17.3 Detailed Information

Toolbar Buttons

The top wizard toolbar contains the following buttons:

- Create Streams: Pressing this button will make the wizard create the streams defined by the configuration. Before creating the streams the wizard will validate if the configuration is valid. If this check fails the stream creation will be aborted
- Validate Configuration: Pressing this button will just execute the configuration validation step described above. You can use this to quickly check your configuration while building it.
- Reset Configuration: Pressing this button will reset the wizard configuration to the default value.

Topology Settings

The wizard will generate and pair streams according to the selected port topology. The following topology choices are available:

- Pairs Topology: Each port is paired together with another port. These two ports only communicate with each other. There must thus be an even number of ports in your testbed.
- Blocks Topology: Each port is placed in either the **West** or the **East** group. Each port in one group communicates with all ports in the other group but not with any port in its own group.

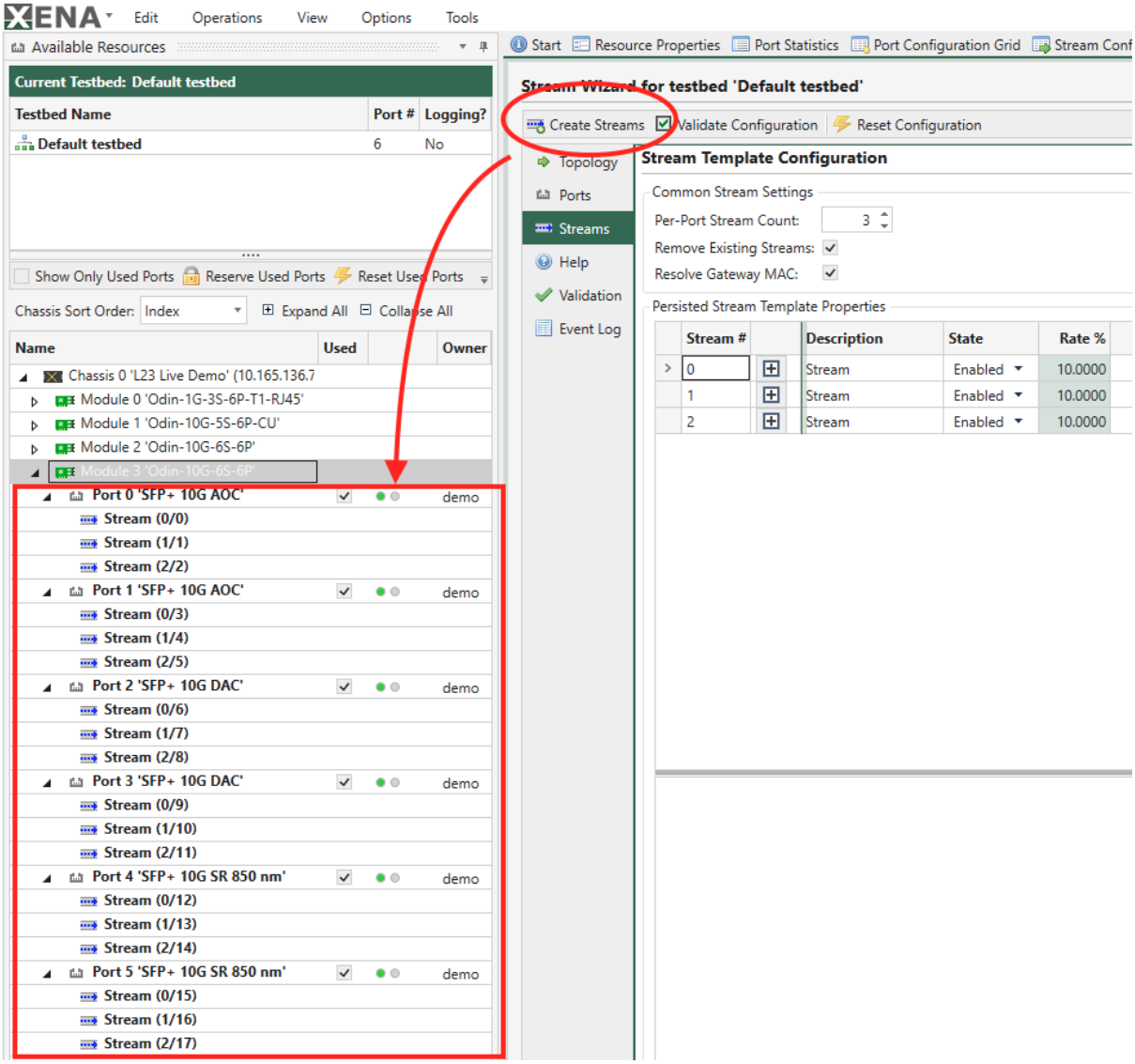


Fig. 5.137: Step 8: Create streams

- Mesh Topology: Each port communicate with all other ports.

Port Configuration

The port configuration is handled by the Ports sub-tab in the main Stream Wizard panel.

It is possible to define a set of properties that will be applied to the ports in your testbed. The default configuration will show a few properties as shown in the image below. But you can choose which properties to apply to your ports by pressing the Select Port Property Types button located in the upper right corner.

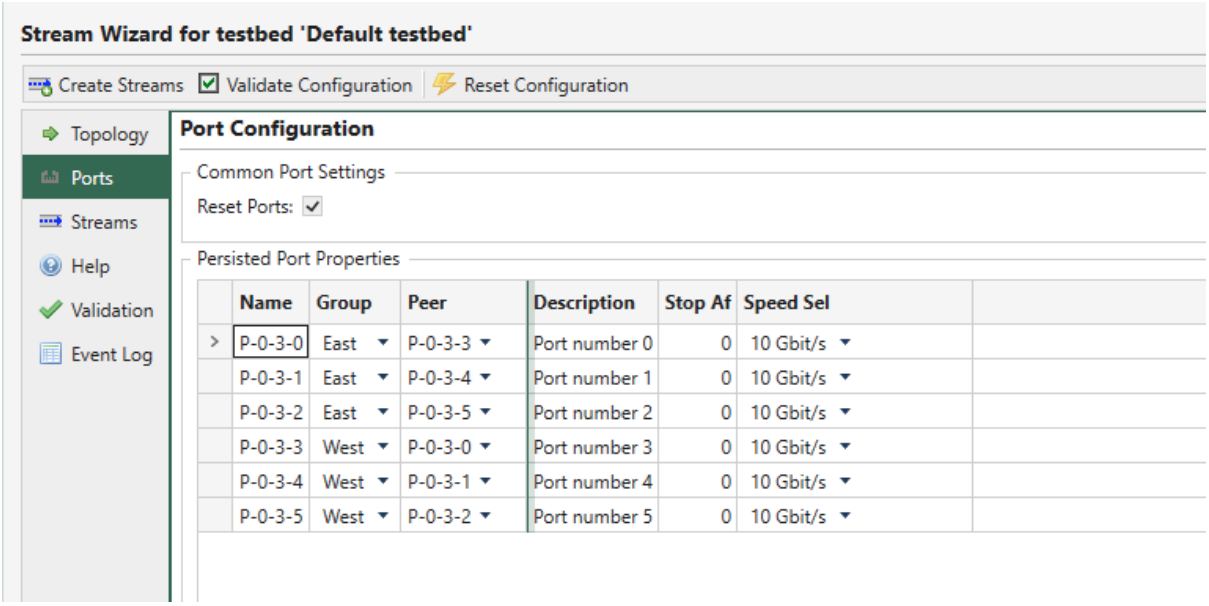


Fig. 5.138: Stream Wizard - Port configuration

Any port property that is not specifically set in your wizard configuration will be set to the default value of the port (or left at the current value if you have chosen not to reset the ports).

The *Reset Ports* checkbox will control whether the ports are reset to their default state before applying the specified properties. It is normally recommended to enable this option as it ensures that the resulting configuration is reproducible every time you run the wizard. But you may have special reasons for not wanting to reset the ports, such as wanting to retain a specific custom setup.

Stream Template Configuration

The stream template configuration is handled by the Streams sub-tab in the main Stream Wizard panel. You can define a number of stream templates for the configuration using the *Per-Port Stream Count* selector.

Each stream template will be used to create a single stream on each source port for each of that port's peer ports.So if you have 3 ports in your testbed and you specify a Mesh topology each port will have two peer ports. If you define for instance 3 stream templates then each port will end up containing a total of 6 streams where the first 3 streams goes to the first peer port and the other 3 streams goes to the other peer port.

For each stream template it is possible to define a set of stream properties that will be applied to the actual streams in your testbed. You can choose which properties to include by pressing the *Select Stream Property Types* button located in the upper right corner. The type of the selected properties are common for all stream templates but the value for each property can be different for each template.

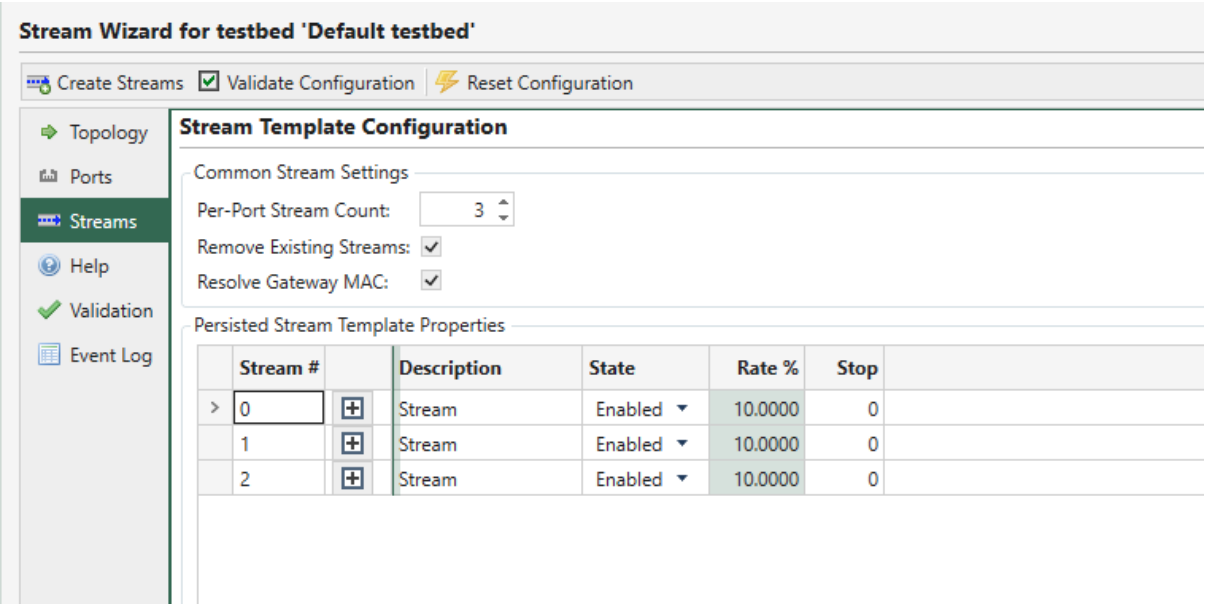


Fig. 5.139: Stream Wizard - Stream Template configuration

Any stream property that is not specifically set in your wizard configuration will be set to the default value of the stream.

The *Remove Existing Streams* checkbox will control whether the existing streams are removed before creating the new streams. It is normally recommended to enable this option as it ensures that the resulting configuration is reproducible every time you run the wizard.

> **Note**
>
> Note that if you have selected the *Reset Ports* option in the Ports sub-panel then all existing streams will be removed regardless of the value of the *Remove Existing Streams* option.

The Stream Wizard will automatically ensure that the Source and Destination MAC and IP fields in the Ethernet and (optional) IP headers will match the port pairing. If the Resolve Gateway MAC option is selected then the Stream Wizard will try to resolve the MAC address of any defined gateway addresses and use this address as the *DMAC* address instead.

Validation Errors Panel

If the configuration validation will reveal any errors in the configuration you can view the detailed list in the Validation sub-panel. Each error is shown on a different line. Each line shows both a description of the error, together with the resource type and identifier that caused the error.

Wizard Execution Event Log

The Event Log shows a log of all actions performed by the wizard when creating the streams. This will also show any errors optionally encountered by the wizard when creating the streams.

Persistence

The stream wizard configuration will automatically be saved as part of the current testbed configuration. You can thus adjust the wizard configuration and re-generate your streams over and over.

> **Note**
>
> Please note that if you make manual changes to the actual port and/or stream configuration after the wizard has created the initial configuration then these changes **will not be retained** in the wizard configuration!

5.18 Replay PCAP File

The XenaManager is capable of replaying the packets in a PCAP file on a single test port. You access this function through by selecting the *Edit* → *Replay File* menu item when the port you want to use is selected. You will have to reserve the port before the function is available.

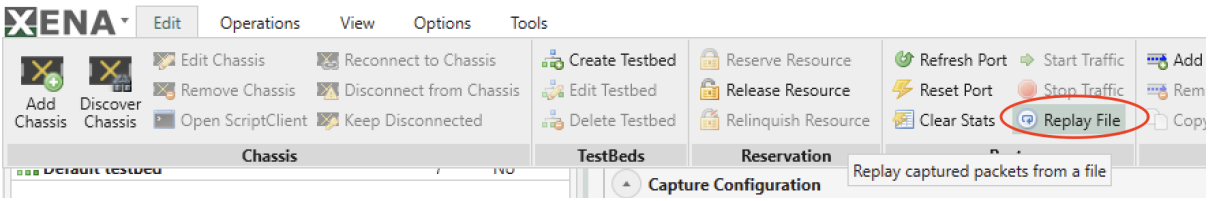


Fig. 5.140: Replay File function

When you select the *Replay File* menu item you will be asked to select the PCAP file you want to use. XenaManager supports both traditional PCAP files and the newer PCAP-NG format. After you have selected the PCAP file then content of the file is imported and a replay control windows will be shown (see screenshot below).

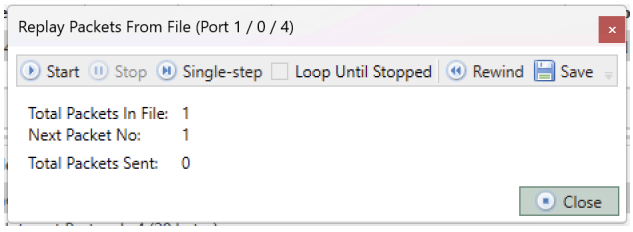


Fig. 5.141: Replay packets from a pcap file

Important

XenaManager assumes that the PCAP file you provide doesn't include FCS checksum field. When doing the replay, XenaManager will append the 4-byte FCS checksum field to the replayed packets. Read *Comparing Replayed with Original PCAP* for more.

The following commands are available in the control window:

Table 5.57: PCAP Replay Commands

Command	Explanation
Start	Start transmitting packets form the PCAP file as fast as possible. When all packets are sent the replay is automatically stopped (unless the *Loop Until Stopped* option is selected).
Stop	Stop the packet transmission.
Single-Step	Send the next packet in the packet sequence and stop.
Loop Until Stopped	If selected, the transmission will start over when the end is reached. If not selected the transmission is stopped when the end is reached.
Rewind	Reset the current packet position to the first packet in the sequence.

Important

The following limitations apply to the PCAP Replay function:

Packets will not be re-sent with the original inter-packet timing from the PCAP file. The transmission control is handled by the PC running XenaManager. Packets are sent one at a time and the next packet is not sent until the last packet was successfully transmitted. The transmission timing is thus influenced by both network and Windows OS latency.

The maximum packet size that can be transmitted is 2 Kbyte.

See also

Xena offers better Wireshark integration via a dedicated LUA plugin. You can download the `Wireshark plugin`.

5.18.1 Comparing Replayed with Original PCAP

When you capture packets in Wireshark on your computer, which is behind a NIC, the captured packets don't include the FCS checksum. But when you capture packets on Xena test equipment, the capture is directly on the wire. This means, Xena test equipment captures everything on the wire, including the 4-byte FCS checksum field.

If you want to exclude the FCS checksum in the captured PCAP file, you should disable *Option → Save Ethernet FCS*.

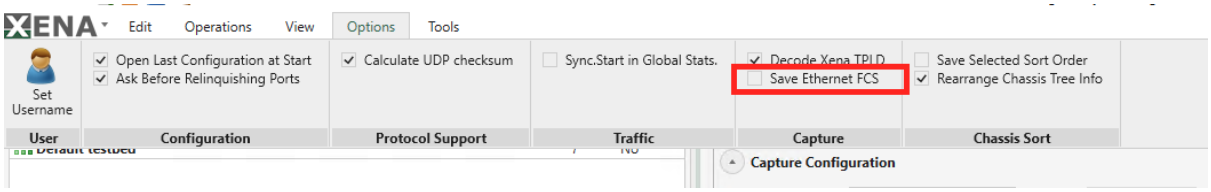


Fig. 5.142: Exclude Ethernet FCS checksum field in PCAP

After disabling *Option → Save Ethernet FCS*, the captured PCAP will not contain the FCS checksum. In Wireshark, if you want to see the capture length instead of the length on the wire, you can add a field to the column as shown in Fig. 5.143.

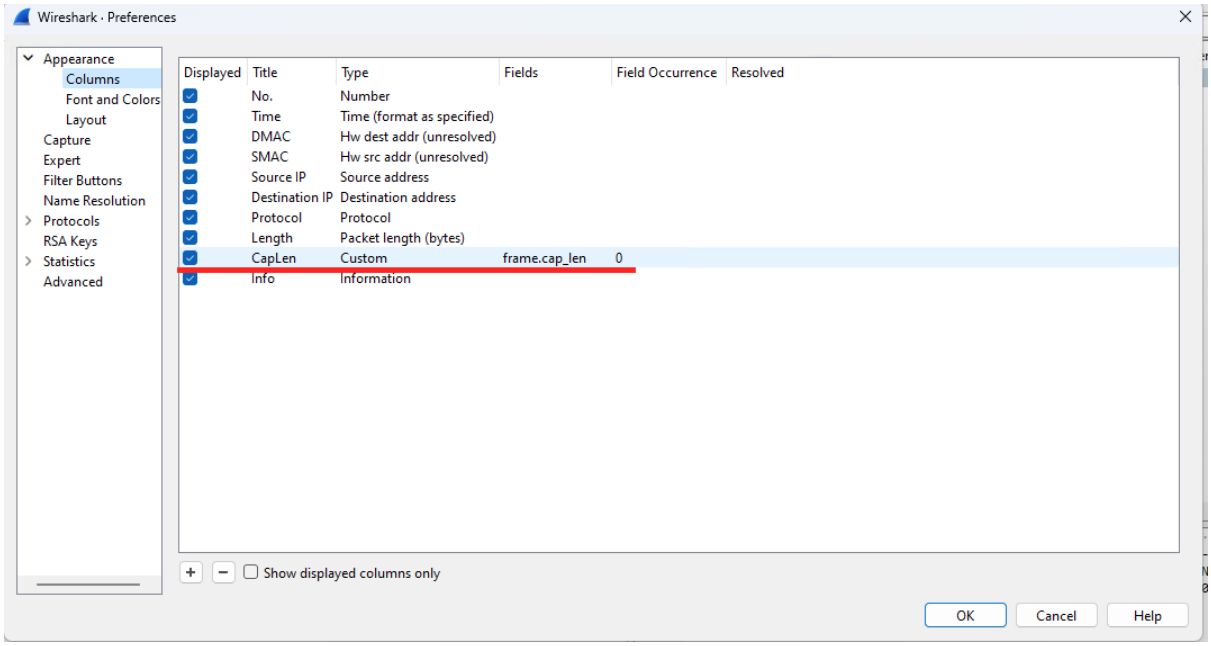


Fig. 5.143: Add capture length field in Wireshark view

CHAPTER

SIX

IMPAIRMENT

6.1 Overview

Chimera is an Ethernet network emulator, which is inserted between two Ethernet ports to emulate a network, by applying impairments to the packets being forwarded between the ports.

Chimera modules can be installed in a Compact chassis or in a XenaBay 2400G. If installed in XenaBay 2400G, Chimera modules can co-exist with TG modules, i.e. Z10 Odin, Z100 Loki, Z400 Thor, and Z800 Freya.

XenaManager provides a single interface for configuring and monitoring test modules.

This document explains the functionality of Chimera and how to configure it. The configuration is illustrated using both the User Interface (UI) named XenaManager and the script commands. Chimera supports the general Xena script commands for ports and modules and a number of Chimera specific script commands for configuring impairments.

This document assumes that the user is logged into the Chimera impairment emulator with XenaManager.

6.1.1 CLI Script Interface

Chimera can be 100% controlled using scripting commands, i.e., all configuration and all statistics can be accessed via scripting.

This document includes examples on how to configure Chimera using script commands, by providing simple script examples for each of the described functions. For further details on the script commands supported by Chimera, please refer to Impairment CLI for XOA CLI commands

In the script command examples in this document, SYMBOLIC constants are used where possible. The constants are followed by the numeric number in parenthesis.

```
PEF_IPV4SETTINGS [fid,0] ON (1) INCLUDE (1)
```

This includes the following SYMBOLIC constants:

- ON = 1
- INCLUDE = 1

The *fid* is the Flow ID, which is explained in the following section.

See also

Read more in *CLI Script Client*

6.1.2 Ethernet Packet Forwarding

Chimera is an impairment module, which works as a "bump-in-a-wire" forwarding Ethernet packets of sizes from 56 bytes to 12288 bytes. The ports are divided into fixed port pairs, and packets are forwarded between the ports in the port pair. The port pairs for different speeds are illustrated in Table 6.1.

Table 6.1: Chimera port pairs

Speed	Port Pairs
100G / 40G	Port 0 <–> Port 1
50G	Port 0 <–> Port 2 Port 1 <–> Port 3
25G / 10G	Port 0 <–> Port 4 Port 1 <–> Port 5 Port 2 <–> Port 6 Port 3 <–> Port 7

The port pairs as seen in XenaManager are illustrated in Fig. 6.1.

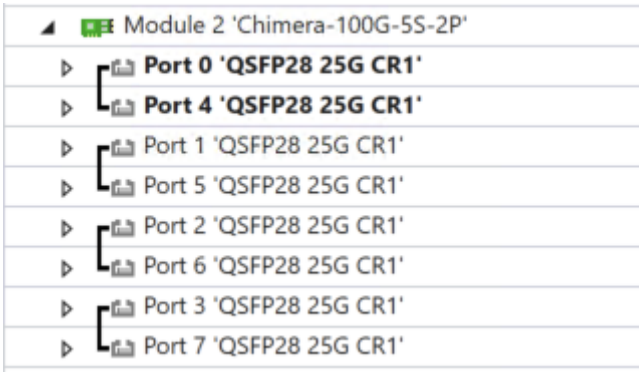


Fig. 6.1: Chimera port pairs in XenaManager.

6.1.3 Chimera Impairment Pipeline

Chimera implements up to 8 flows per port. Each flow implements a separate impairment pipeline, where the packets of that flow can be impaired independently of other flows.

Packets received on a port in Chimera will pass through the active flow filters. If the packet matches a filter, the packet is mapped to the corresponding flow and passes through the corresponding impairment pipeline.

The impairment pipeline includes the following configurable impairments:

- Policing
- Drop

- Misorder
- Latency / Jitter
- Corruption
- Duplication
- Shaping

See also

Read more about *Impairments Types*.

Notice that the impairment pipeline includes a delay block. This delay block is responsible for generating both a fixed latency and a variable latency in terms of jitter. Throughout this document, the delay block is referred to as the "latency / jitter" impairment block.

The flow filters and corresponding impairment pipelines are illustrated in Fig. 6.2.

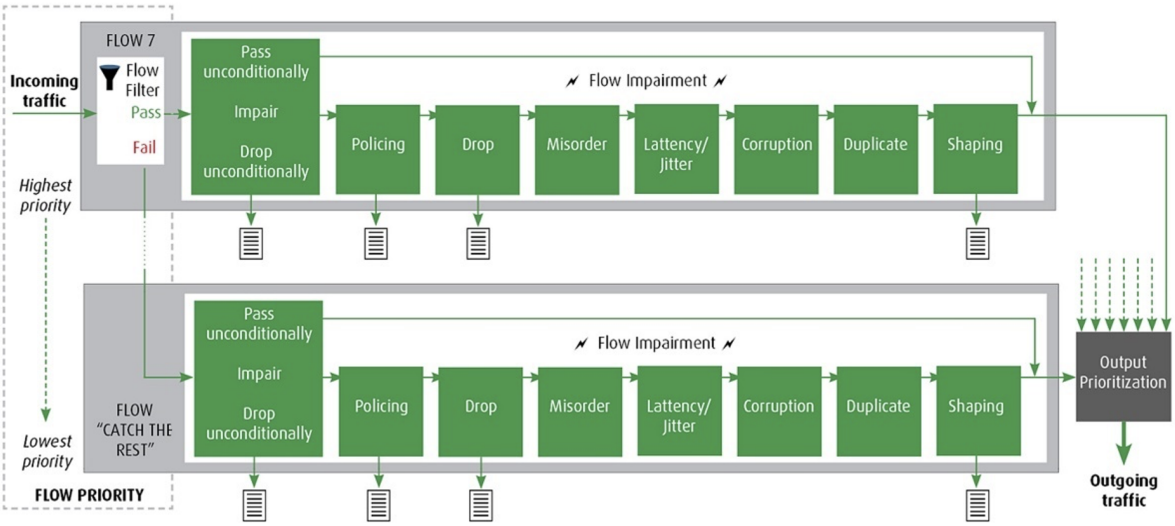


Fig. 6.2: Chimera impairment pipeline

Each impairment in each flow impairment pipeline can be individually configured. Notice that Fig. 6.2 illustrates that packets may be dropped because of the following impairments:

- Policing
- Drop
- Shaping

After the flow specific impairment pipeline, all packets destined for a given output port are merged into a common packet stream and forwarded to wire.

6.2 Latency / Jitter Explained

The architecture of the Chimera delay block puts certain limits on the minimum and maximum latency that can be configured in the latency / jitter impairment described in section 10.6.

This section describes these limits, along with timing configuration parameters and the timing accuracy that can be expected from the latency / jitter impairment, depending on the configuration. Besides the latency of the active emulator core described in this section, notice the emulator bypass mode described in section 3.2.

The minimum latency that can be configured for any latency distribution is described in Section *Minimal Latency*. Regarding the maximum latency that can be configured for any latency distribution, there are two limits to be aware of:

- Lossless Latency:

  The maximum latency that can be guaranteed without the risk of packet loss at wire speed.

- Lossy Latency (Reduced Bandwidth Latency):

  The maximum latency supported by Chimera. At this latency, there will be loss at wire speed. It is possible to calculate a reduced BW which can be supported without loss. Sending packets at a higher bandwidth than the guaranteed BW will eventually result in packet loss. See Section *Maximum latency (Reduced Bandwidth Latency)* for details on reduced BW.

When configuring a latency distribution, it is possible to configure the maximum delay to the lossy latency limit. However, when configuring a maximum above the lossless latency limit, sending packets at a higher BW than the guaranteed reduced rate may result in packets being lost.

Maximum Lossless Latency is described in Section *Maximum Latency (without packet loss)*, while Reduced Bandwidth Latency is described in Section *Maximum latency (Reduced Bandwidth Latency)*. Finally, Section *Latency and Multiple Flows* describes the latency / jitter accuracy that can be expected depending on configuration.

6.2.1 Extended Timing Mode

The latency / jitter impairment can operate in either “Normal Timing Mode” or “Extended Timing Mode”. Normal timing mode will allow high precision latency and jitter, with a maximum configurable latency of 1.9 sec.

Extended timing mode allows configuring latencies up to 19.5 sec. at the expense of the latency and jitter precision. The minimum configurable latency is unaffected by the setting of extended timing mode.

The timing mode is configured at the module level as illustrated in Fig. 6.3, and will apply to the entire Chimera module.

Note

Changing the timing mode will reset all configured latency / jitter parameters in the entire

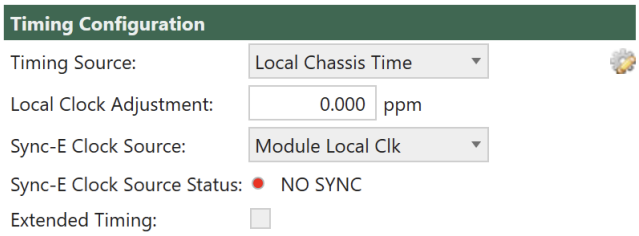


Fig. 6.3: Configuring Extended Timing Mode

module to default values, including those configured for custom distributions. Hence, modifying the timing mode will require all timing values in the module to be reconfigured.

To configure extended timing mode:

Go to the *Module* → *Resource Properties* tab and select the required timing mode (e.g. off).

This will set the extended timing mode to off and cause all timing parameters in the module to be reset to default values.

Due to decreased latency / jitter precision, it is recommended that users enable extended timing mode only when the increased maximum delay is required for your testing.

Note

Corresponding CLI command: `M_LATENCYMODE`

6.2.2 Minimal Latency

Chimera supports the following minimum latency (intrinsic latency) depending on port speed:

Table 6.2: Chimera minimum latency (intrinsic latency)

Port Speed	Minimum Latency
8 x 10G (no RS-FEC)	13 µs (13,000 ns)
8 x 25G	7.3 µs (7,300 ns)
2 x 40G (no RS-FEC)	7.1 µs (7,100 ns)
4 x 50G (no RS-FEC)	7.1 µs (7,100 ns)
2 x 100G	7.1 µs (7,100 ns)

The minimum latency is significantly increased for 10G due to the store and forward delay of a 10K Ethernet packet at 10G. Notice that the minimum latency is unaffected by the setting of the timing mode described in Section *Extended Timing Mode*.

6.2.3 Maximum Latency (without packet loss)

Due to the amount of memory needed to support latency, there is an upper limit to the latency which can be supported without loss. The maximum latency without loss which can be configured for a flow depends on the number of ports and flows currently active on the port.

Table 6.3: Chimera-100G-5S-2P maximum latency

Port Speed	1 active flow	2 active flows	3-4 active flows	5-8 active flows
8 x 10G (no RS-FEC)	1,990 ms	995 ms	497 ms	248 ms
8 x 25G	796 ms	398 ms	199 ms	99 ms
2 x 40G (no RS-FEC)	497 ms	248 ms	124 ms	62 ms
4 x 50G (no RS-FEC)	398 ms	199 ms	99 ms	49 ms
2 x 100G	199 ms	99 ms	49 ms	24 ms

Table 6.4: Chimera-100G-5S-2P[b] maximum latency

Port Speed	1 active flow	2 active flows	3-4 active flows	5-8 active flows
8 x 10G (no RS-FEC)	3,990 ms	1,995 ms	997 ms	498 ms
8 x 25G	1,596 ms	798 ms	399 ms	199 ms
2 x 40G (no RS-FEC)	997 ms	498 ms	249 ms	124 ms
4 x 50G (no RS-FEC)	798 ms	399 ms	199 ms	99 ms
2 x 100G	399 ms	199 ms	99 ms	49 ms

The total amount of latency / jitter memory inside Chimera is constant. This memory is divided equally between the active flows, as is reflected in Table 6.3 and Table 6.4, which illustrates how the maximum lossless delay depends on the number of active flows.

The distribution of memory among active flows mentioned above, requires re-allocating the memory, when the number of active flows is modified. If traffic is running through the filters, when memory is re-allocated, packets will be lost on all active flows.

To avoid packet loss due to memory re-allocation, enable all required filters, before starting the traffic. Subsequently modifying the filters will not result in any packet loss.

6.2.4 Maximum latency (Reduced Bandwidth Latency)

Chimera supports latencies above what is listed in Table 6.3 and Table 6.4, but in such cases, it can only be guaranteed to be lossless at a reduced bandwidth given by:

```
ReducedBW(Gb/s)=(LossLessLatency*Speed(Gb/s))/ConfiguredLatency
```

where the `LosslessLatency` is taken from Table 6.3 and Table 6.4, and `ConfiguredLatency` is the latency currently configured for the flow (> `LosslessLatency`).

In case the average data rate on the flow exceeds the reduced bandwidth, packets will be dropped.

The maximum latency which can be configured for `reduced bandwidth = 1.9 sec (normal timing mode) / 19.5 sec`. (extended timing mode). See Section *Extended Timing Mode* for details.

6.2.5 Latency and Multiple Flows

When only the default flow is configured on a port, the uncertainty on the configured latency is +/- 50 ns.

When multiple flows are configured on a port, there will be an added latency due to the fact that packets from multiple flows need to be merged onto the same physical link at the Chimera output port. This is a basic property of Ethernet. In this case, the added latency will depend on the number of flows configured on the port and the maximum packet size on the active flows.

When adding a 2nd flow (flow #1) to the port, the packets of the default flow risk waiting to be merged into the common output packet stream due to transmission of a packet on flow #1. Worst case, this is a maximum size packet (= 10K bytes), in which case 802 ns (for 100 G) will be added to the latency of the packet of the default flow (see Table 6.5).

For every flow added to the port, the packets of a given flow risk waiting another maximum packet size to be merged at the output. The influence of the multiple flows is very random, but worst case, a packet scheduled to be sent on a port with 8 flows configured will experience an increased delay of 7 x maximum packet delay. The worst case added latencies in cases with 8 flows on a port are listed in Table 6.5.

Table 6.5: Chimera maximum added latency

Speed	Delay of 10K pkt (ns)	Max added delay (ns)
100G	802	5,600
50G	1,603	11,200
40G	2,004	14,000
25G	3,206	22,400
10G	8,016	56,100

6.3 Module Properties

This section describes the settings that will affect the entire module, as opposed to port level or flow level settings.

6.3.1 SyncE

Chimera implements a single clock domain for clocking all Tx ports. The source clock can be configured to be an internally generated clock or a recovered clock from one of the active Rx ports.

To successfully synchronize the Tx ports to a recovered Rx clock, the internal circuitry must be able to lock on the configured Rx clock. Chimera implements a locking signal which indicates whether Chimera is currently locked to a Rx port. This “Rx lock” signal MUST be ON before you can trust that the Tx ports are running of the configured Rx port.

If the Rx lock signal returns `NOVALIDTXCLK`, it implies that Chimera could not lock to the configured Rx port clock, in which case it will fall back to running off the internally generated clock.

Notice that when configured to run off the internal clock (Module Local Clock), the Rx lock signal will always return `NOVALIDTXCLK`.

Table 6.6: Chimera minimum latency :widths: auto :header-rows: 1

Parameter	Legal values	Comments	Step size
Clk sourc	0 to 9	0 (=MODULELOCALCLOCK) 1 Not supported 2 (=P0RXCLK) - All speeds 3 (=P1RXCLK) - All speeds 4 (=P2RXCLK) - 50G / 25G / 10G 5 (=P3RXCLK) - 50G / 25G / 10G 6 (=P4RXCLK) - 25G / 10G 7 (=P5RXCLK) - 25G / 10G 8 (=P6RXCLK) - 25G / 10G 9 (=P7RXCLK) - 25G / 10G	1

Value = 1 is not supported and if so configured, the status will return: `NOVALIDTXCLK`.

You can only configure one of the available ports as clock source. The number of available ports depends on the selected speed mode. See Table 1 for valid ports depending on port speed.

Selecting a port which does not exist for the selected speed mode will cause the internally generated clock to be selected as Tx clock source.

Fig. 6.4 illustrates how to configure SyncE in the UI (Select *Module → Resource Properties* ` tab).

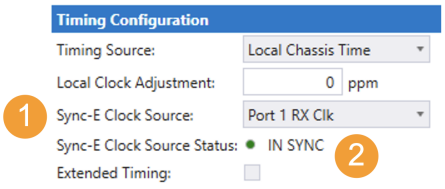


Fig. 6.4: Configuring SyncE

To configure SyncE:

1. Select the required clock source from the dropdown menu.
2. IN SYNC indicates if Chimera was able to lock to the selected input Rx port. (IN SYNC is not valid when selecting module local clock.)

The example above illustrates how to lock Tx output to the recovered clock from Rx port 1. Further the green light, indicates that the module was able to recover the configured clock.

Note

The example below illustrates how to configure the same example using script commands.

```
M_TXCLOCKSOURCE P1RXCLK
```

The example below illustrates how to query if Chimera successfully locked to the configured Rx clock.

```
M_TXCLOCKSTATUS ?
```

Note

Notice, that the SyncE implementation described above, implies that Chimera is not Ethernet Synchronization Message Channel (*ESMC*) message aware and that all ESMC messages will pass transparently through Chimera if not explicitly configured for impairment using a flow filter.

6.3.2 Emulator Bypass

It is possible to completely bypass the emulator core by directly connecting the input ports to the output ports for minimum latency. Setting the bypass mode, will affect all the ports of the module.

The emulator bypass mode is a convenient way of inactivating Chimera in the test setup, without physically removing the cables.

While in bypass mode, Chimera can be configured and statistics are updated, but this will have no effect on the output traffic. I.e. Chimera settings / statistics must be completely disregarded for the duration of the bypass.

The constant latency introduced by Chimera when in bypass mode, is listed in Table 6.7, for different port speeds and FEC settings. In addition to the constant latency listed in Table 6.7, Chimera will introduce a jitter of ± 50 ns.

Table 6.7: Chimera latency delay in emulator bypass mode

	10G	25G	25G (FEC)	40G	50G	100G	100G (FEC)
Latency (ns)	1150	600	1250	1000	550	600	1400

Fig. 6.5 illustrates how to configure emulator bypass mode in the UI (Select *Module* → *Resource Properties* tab).

Note

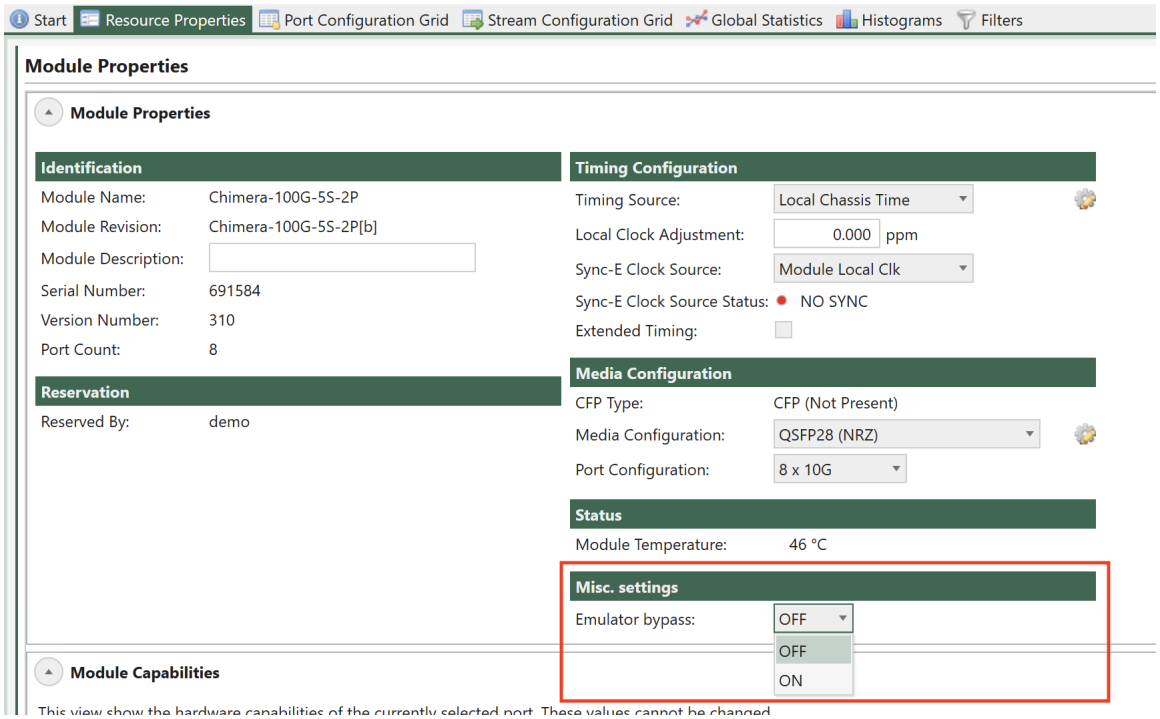


Fig. 6.5: Configuring emulator bypass mode

Corresponding CLI command: `M_EMULBYPASS`

6.4 Port Properties

This section describes settings which will affect the entire port, i.e. it will affect all the flows defined for the selected port.

6.4.1 Reed-Solomon Forward Error Correction (RS-FEC)

Chimera ports support RS-FEC for 25G and 100G speeds.

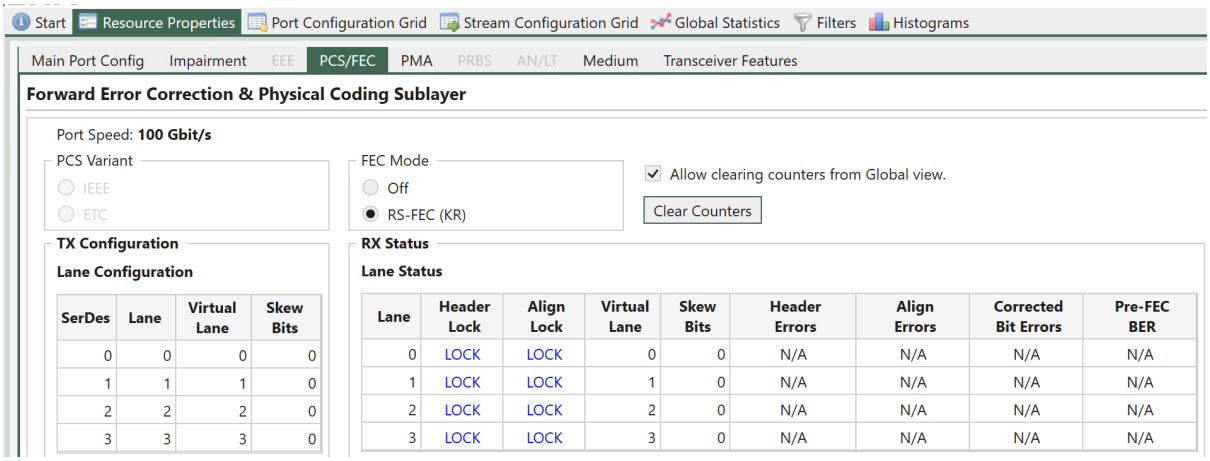
To configure RS-FEC on a port, select the port in the UI and go to the *Resource Properties* → *PCS/PMA Config & Status* tab as illustrated in Figure 8 and click the *Enable RS-FEC* option.

Note

Note that Link Training and Autoneg are currently not supported for Chimera.

Note

Corresponding CLI command: `PP_FECMODE`



SerDes	Lane	Virtual Lane	Skew Bits
0	0	0	0
1	1	1	0
2	2	2	0
3	3	3	0

Lane	Header Lock	Align Lock	Virtual Lane	Skew Bits	Header Errors	Align Errors	Corrected Bit Errors	Pre-FEC BER
0	LOCK	LOCK	0	0	N/A	N/A	N/A	N/A
1	LOCK	LOCK	1	0	N/A	N/A	N/A	N/A
2	LOCK	LOCK	2	0	N/A	N/A	N/A	N/A
3	LOCK	LOCK	3	0	N/A	N/A	N/A	N/A

Fig. 6.6: Enable RS-FEC on Chimera port

6.4.2 Test Payload (TPLD) Size

The Xena traffic generators support inserting a Test Payload (TPLD) into the transmitted packets (see Xena Test Payload). The TPLD contains meta data, which can be used by the Xena receiving device to provide miscellaneous statistics. When Chimera is connected to a Xena traffic generator, Chimera can use the TPLD in the incoming packets for flow filtering (see section 7.5).

The TPLD supports 2 sizes:

- Default (20 bytes)
- Micro (6 bytes)

To use the TPLD for filtering in Chimera, it must be configured for the same TPLD format, as the transmitting Xena traffic generator.

Fig. 6.7 illustrates how to configure the TPLD size for a selected port.

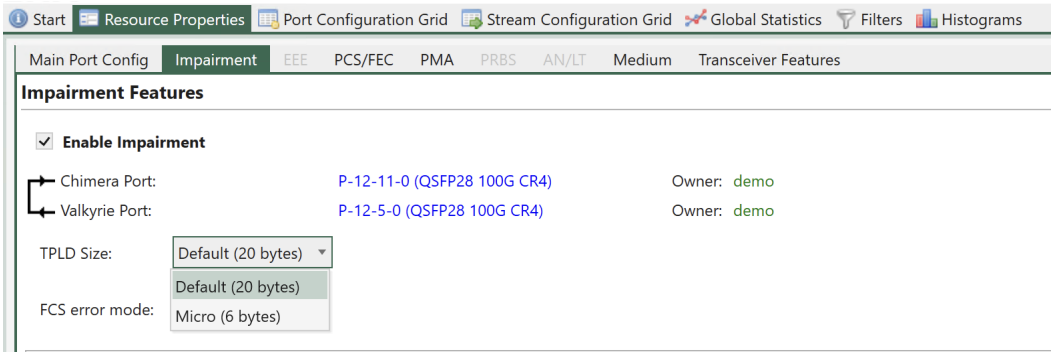


Fig. 6.7: TPLD format configuration.

> **Note**
>
> Notice that this setting is common to all flow filters on this port.

Note

Corresponding CLI command: `PE_TPLDMODE`

6.4.3 FCS Error Mode

When packets with an *FCS* error is received on a Chimera port, they are counted by the port statistics as illustrated in Fig. 6.8.

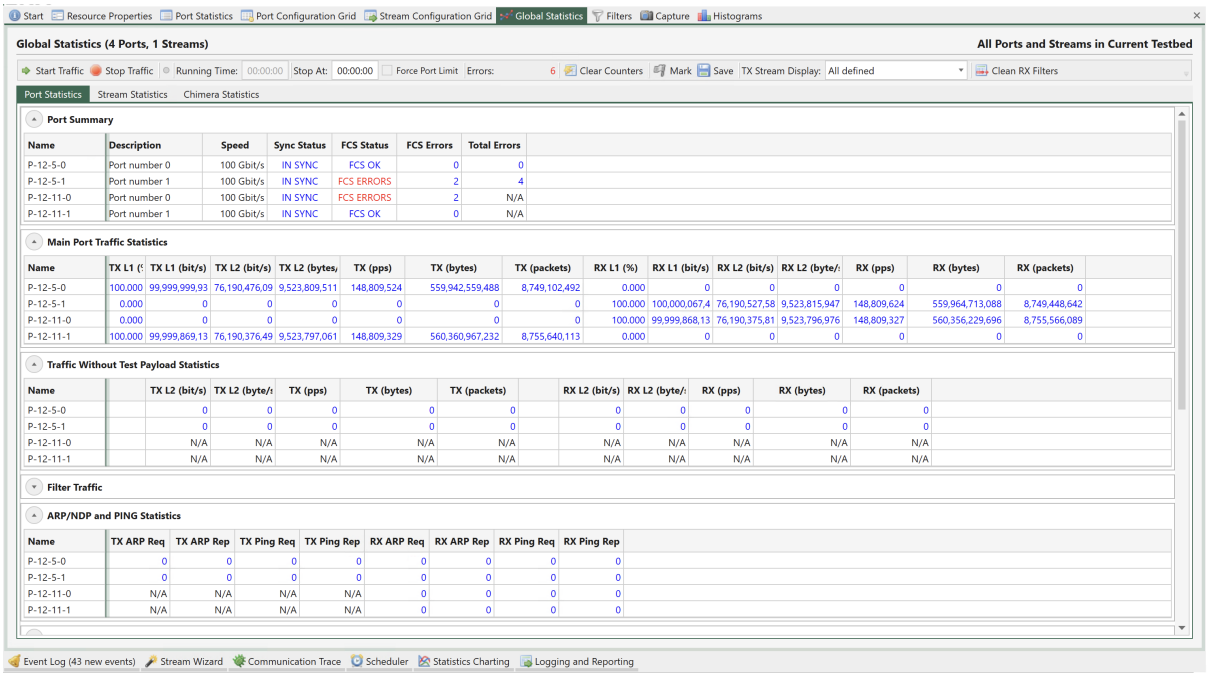


Fig. 6.8: Chimera FCS errors port statistics.

Chimera supports two FCS error modes:

- Pass mode

In this mode FCS errored packets are processed by Chimera as any other packet, i.e., the flow filter is a applied and the packet is subject to flow impairment and forwarded onto the output port.

- Discard mode

In this mode FCS errored packets are filtered by the flow filters and mapped to the corresponding impairment flow, where they are discarded and counted as OTHER DROPS, as shown in Fig. 6.9.

Fig. 6.10 illustrates how to configure the FCS error mode for a selected port.

Note

Corresponding CLI command: `PE_FCSDROP`

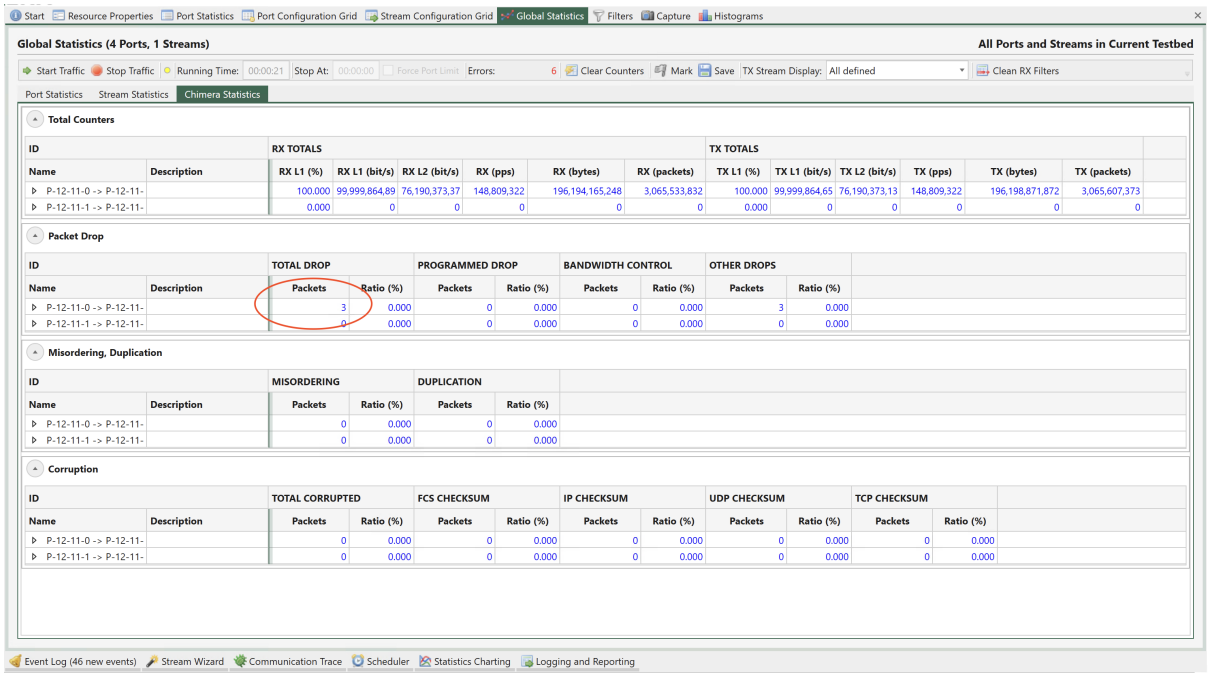


Fig. 6.9: FCS errored packets are discarded as OTHER DROPS

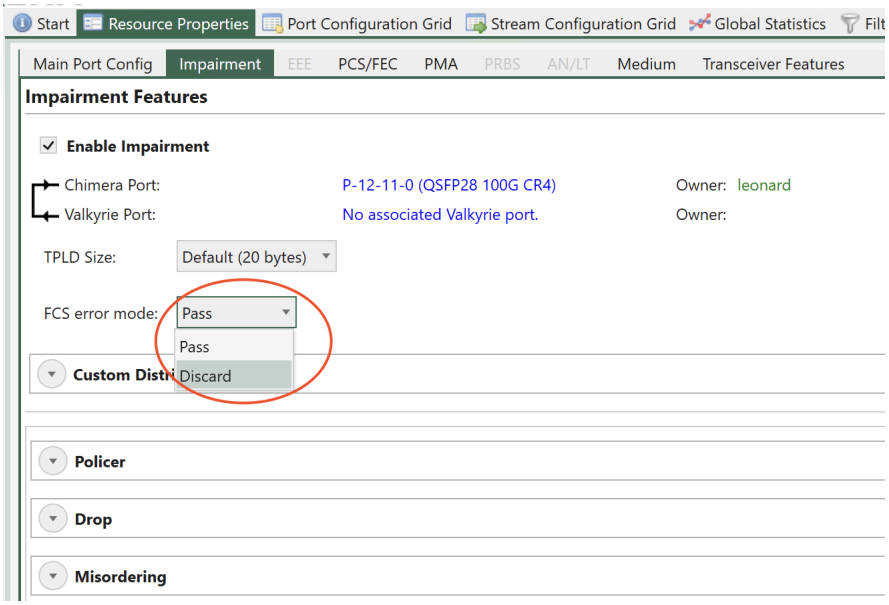


Fig. 6.10: Chimera FCS error mode

6.4.4 Link Flap

Chimera can be configured to emulate that the physical link is down or unstable. This feature is called Link Flap. Link flap is implemented in 2 ways: **Logical Link Flap** and **Optical Link Flap**.

Notice that link flap is configured at a port level and will affect all flows configured for the selected port.

> **Note**
>
> Note that logical link flap and PMA error pulse inject (see Section *PMA Error Pulse Injection*) are mutually exclusive.

Logical Link Flap

Logical link flap is implemented by scrambling the Tx PCS encoding to prevent the peer port from getting a link. It is not implemented by turning the physical transmitter on or off.

Logical link flap works for both electrical cables (*DAC* cables) and optical cables.

Logical link flap is configured under the Main Port Config tab as illustrated in Fig. 6.12.

Logical link flap supports a repetitious pattern, where the link is taken down for a period (Duration) and then brought up again. This is repeated after a configurable amount of time (Repeat Period). The flapping is repeated a configurable number of times or continuously (Repetitions).

Pressing *Start* will start the configured link flap, pressing *Stop* will stop any ongoing link flapping.

Logical link flap is configured as follows:

Table 6.8: Chimera port link flap

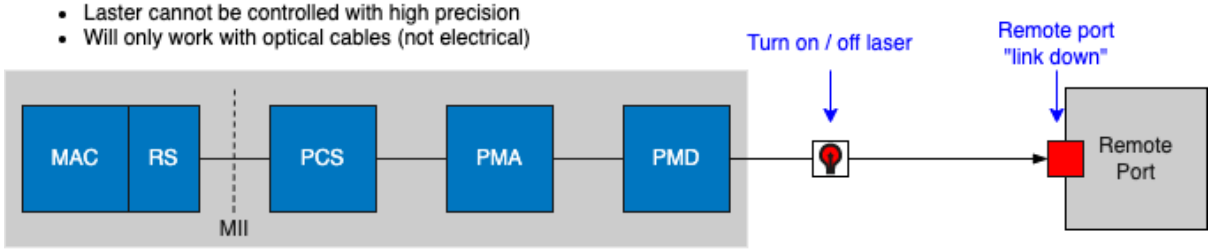
Parameter	Description
Duration	Duration of the link flap.
Repeat Period	Period after which to restart link flap.
Repetitions	How many times to restart the link flap.

> **Note**
>
> For valid parameter ranges please refer to XOA CLI Documentation.

> **Note**
>
> The example below illustrates how to configure a link flap pattern, which will bring down the link for 120 ms and repeat this every 1.2 sec. This will be repeated 2346 times.

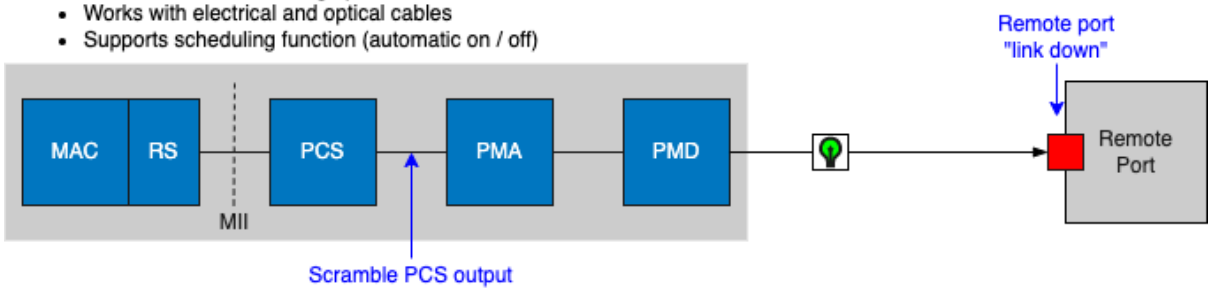
Optical Link Flap

- Manual on / off
- Laster cannot be controlled with high precision
- Will only work with optical cables (not electrical)



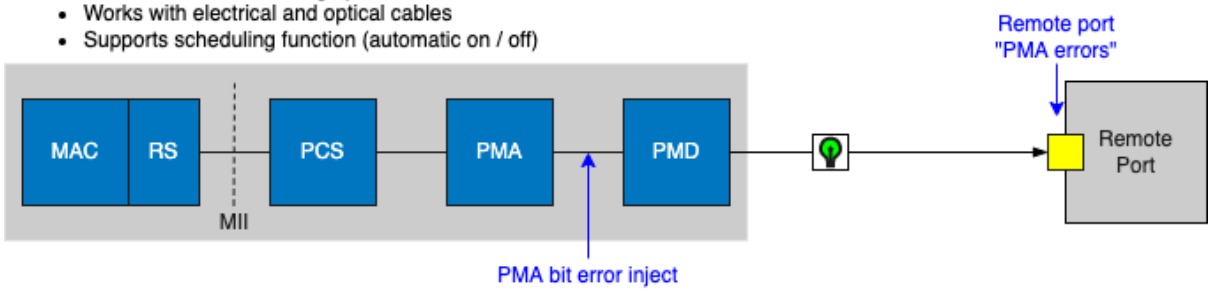
Logical Link Flap

- Scramble PCS output signal
- Can be controlled with high precision
- Works with electrical and optical cables
- Supports scheduling function (automatic on / off)



PMA Error Inject

- Insert bit errors at PMA level
- Can be controlled with high precision
- Works with electrical and optical cables
- Supports scheduling function (automatic on / off)



Scheduling Function

- 10 ms precision

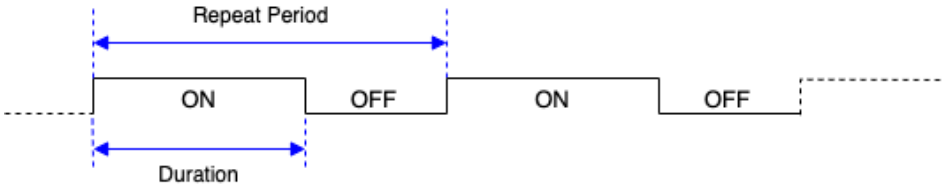


Fig. 6.11: Port impairments

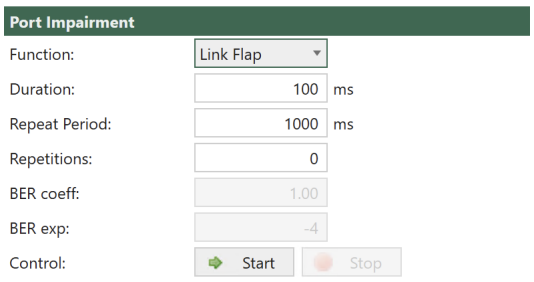


Fig. 6.12: Configuration of Logical Link Flap.

```
PP_LINKFLAP_PARAMS 120 1200 2346
PP_LINKFLAP_ENABLE ON
```

Optical Link Flap

To simulate the event of the optical link going down, it is possible to manually turn the optical transmitter off and on.

Optical link flap only works for optical cables, i.e., it will not work with *DAC* cables for instance. Optical link flap does not support repetitious patterns as described above for logical link flap.

Optical link flap is configured on the *Main Port Config* tab as illustrated in Fig. 6.13.

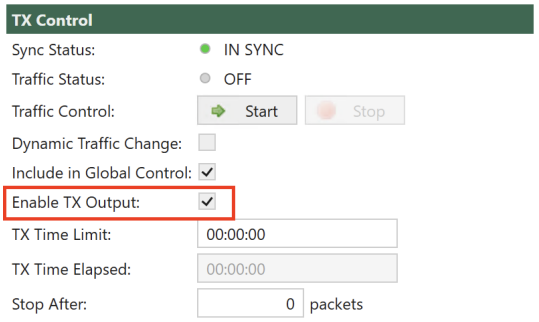


Fig. 6.13: Configuration of Optical Link Flap.

Use *Enable Tx Output* to turn the optical transmitter off / on.

Note

The example below illustrates how to turn the optical transmitter on and off.

```
P_TXENABLE OFF
P_TXENABLE ON
```

6.4.5 PMA Error Pulse Injection

PMA error pulse allows the user to insert pulses of bit errors onto the link. If FEC is enabled, PMA errors are injected after the addition of the FEC bits, so that at the receiving end, FEC will correct as many of the PMA errors as possible.

Notice that PMA error pulse is configured at a port level and will affect all flows configured for that port. For BER insertion on a specific flow, see section 11.1.6.

Logical link flap (see Section *Logical Link Flap*) and PMA error pulse inject are mutually exclusive.

PMA errors can be inserted with a fixed distance dependent on the selected port speed. The supported distances between two adjacent PMA errors and the corresponding BER for all speeds are listed in Table 6.9, where n is an integer number.

Table 6.9: Minimum distance between PMA errors

Speed	Supported PMA error distance	Supported PMA bit error rate
25G / 10G	n * 256 bits	0.39 % / n
50G	n * 512 bits	0.20 % / n
40G / 100G	n * 1024 bits	0.10 % / n

When PMA pulse error injection is configured, the actual BER applied to the link is rounded to the value of n, which is closest to the configured value.

PMA error pulse injection is configured under the *Main Port Config* tab as illustrated in Fig. 6.14.

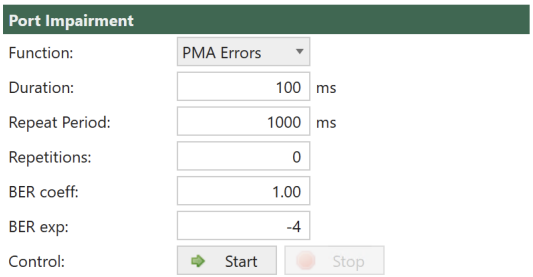


Fig. 6.14: Chimera PMA error pulse injection.

It is possible to configure the length of the error pulse (Duration) and the BER during the pulse (BER coeff and BER exp). The burst is repeated after a programmable period (Repeat Period). The bursts will be repeated a configurable number of times (Repetitions).

Pressing *Start* will start the configured PMA error pulse, pressing *Stop* will stop any ongoing PMA error injection.

PMA error pulse inject is configured as follows:

Table 6.10: PMA error pulse inject configuration

Parameter	Explanation
Duration	Duration of the PMA error pulse.
Repeat Period	Period after which to restart the PMA error pulse.
BER Coeff	BER coefficient.
BER Exp	BER exponent
Repetitions	How many times to restart the PMA error pulse.

Note

For valid parameter ranges please refer to XOA CLI Documentation.

The BER during error pulses is calculated as follows:

```
BER = coeff / 100 * 10^exp
```

Note

Notice that the actual BER is rounded to the values listed in Table 6.9.

The example below illustrates how to configure a PMA error pulse inject pattern using CLI PP_PMAERRPUL_PARAMS and PP_PMAERRPUL_ENABLE , which apply PMA errors for 430 ms and repeat this every 2.430 sec. `BER = 2.34 * 10e-12`. This will be repeated 2346 times.

- Duration = 430 (ms)
- Repeat Period = 2430 (ms)
- BER coefficient = 2.34
- BER exponent = -12
- Repetitions = 2346

```
PP_PMAERRPUL_PARAMS 430 2430 234 -12 2346
PP_PMAERRPUL_ENABLE 1
```

6.5 Packet Flow

When an Ethernet packet enters Chimera, it is assigned to a flow. Impairments are configured independently for every flow. Every Rx port has a default flow. Additional flows can be added to the Rx port by configuring a corresponding flow filter. A maximum of 7 flow filters can be configured on every Rx port. Packets which match the flow filter, will be mapped to the corresponding flow. The flow filters are prioritized so packets are first matched against the filter of flow #7, subsequently flow filter #6 and finally filter #1.

Packets which do not match any flow filter, will be assigned to the port default flow (= flow #0).

This results in a total of maximum 8 flows for every port independent of port speed.

The Chimera flows as seen in XenaManager are illustrated in Fig. 6.15.

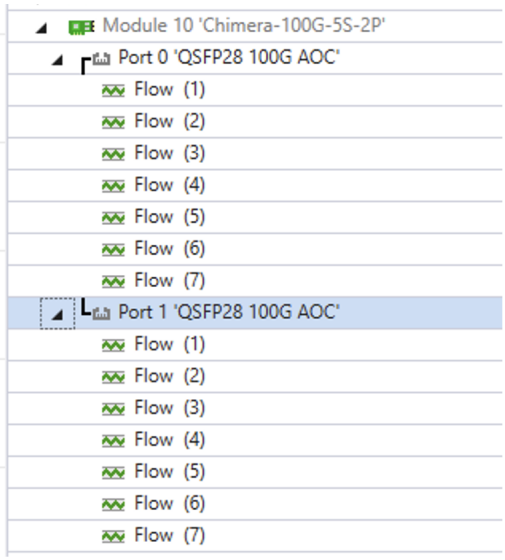


Fig. 6.15: Chimera flow filters in XenaManager

Once the packet is mapped to a flow, it will pass through the associated impairment pipeline (see *Overview*).

At the output of the impairment pipeline, packets from different flows will be merged into a common packet flow, which is transmitted to the output port. This is illustrated in Fig. 6.16.

6.6 Flow Filter

6.6.1 Overview

As described in Section *Overview*, flows in Chimera are defined by the flow filters.

If a packet matches a given filter, the packet is mapped to the corresponding flow.

Note

Notice, that modifying the number of active flow filters while traffic is running through Chimera, will result in packet drops, as described in Section *Maximum Latency (without packet loss)*.

Important

Notice that **Extended mode** and **Basic mode** are **mutually exclusive**.

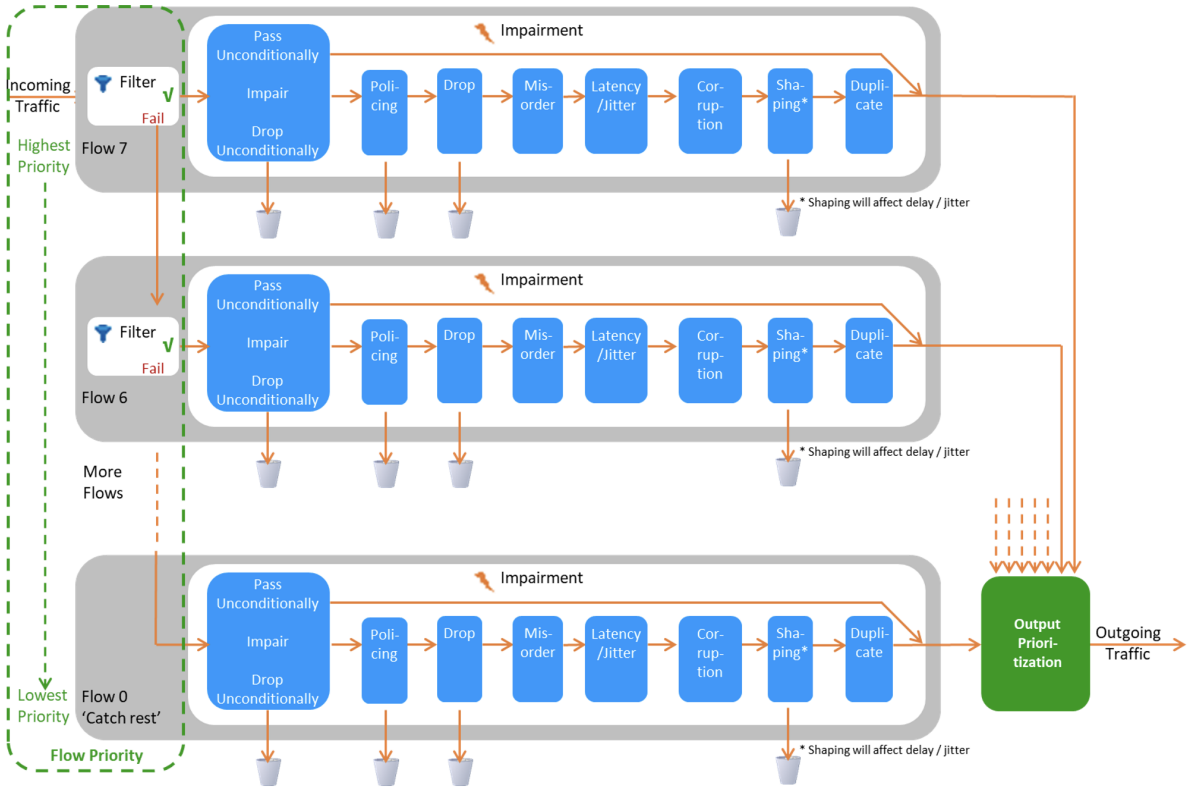


Fig. 6.16: Chimera packet flow and impairments.

Updating Flow Filter Registers

Flow filters can be updated during runtime with traffic applied to the input ports. To guarantee that filtering is always coherent, Chimera implements two sets of registers in the flow filters:

- Working registers: used for flow filtering.
- Shadow registers: used for updating flow filters.

All registers in the flow filters have both a **working register** and a **shadow register**.

Shadow registers can be written and read, while working registers can only be read.

Applying the script command `PEF_APPLY` will transfer all the shadow register values to the working registers instantaneously for all flow filter settings, including all sub-filters in basic mode (see Section *Basic Mode*), so flow filters are always coherent. This allows updating the shadow registers, without the risk of using intermediate filtering values.

Note

The following example illustrates how to specify filter and register type when accessing the flow filter registers using script commands.

```
PEF_ENABLE[fid,filter_type] ON
```

- `fid`: indicates the filter to configure (filter ID: 1 - 7)
- `filter_type`: indicates shadow (0) or working (1)

As described above, it is never legal to write to a register with filter_type = working (1).

Flow Filtering Modes

The flow filters can be configured in two different modes:

- Basic mode - a simple way of configuring a limited number of networking protocols.
- Extended mode - allows configuring of any networking protocol within the first 128 bytes.

Note

Notice that Extended mode and Basic mode are mutually exclusive, since they are both using the same FPGA filters.

Basic mode is selected as default. To select Extended mode in the UI, click Extended mode as illustrated in Fig. 6.17.

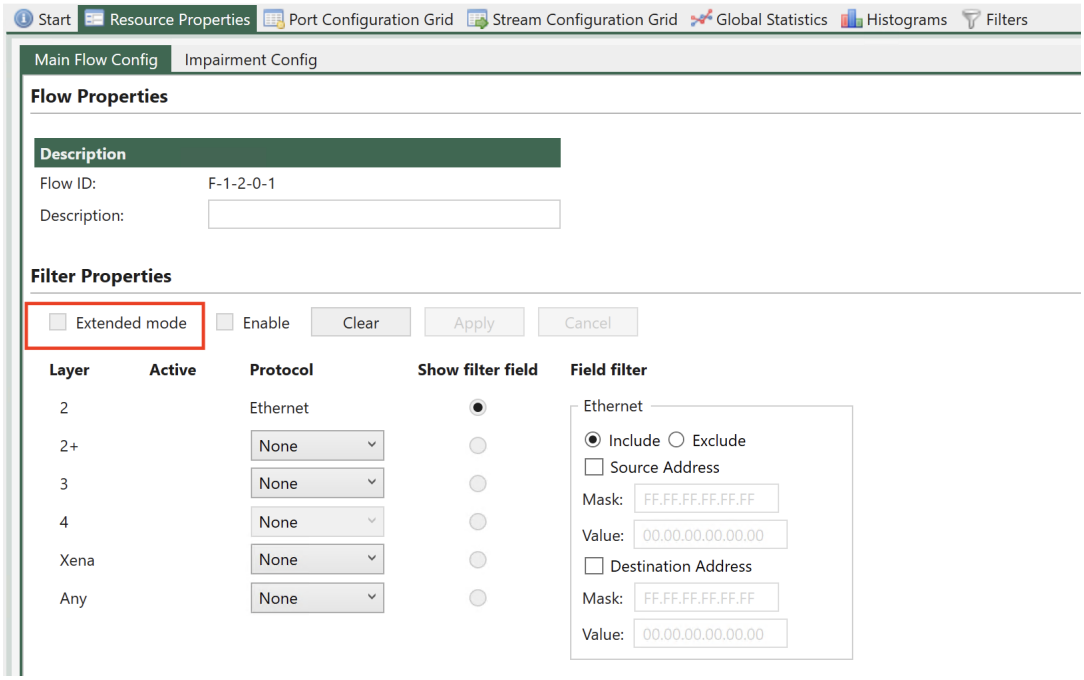


Fig. 6.17: Configure flow filtering mode in UI.

Note

The example below illustrates how to configure Extended mode flow filtering.

```
PEF_MODE [fid,0] EXTENDED
PEF_APPLY[fid,0]
```

Basic mode is described in Section *Basic Mode*, while Extended mode is described Section *Extended Mode*.

If you are new to networking protocols and encapsulation, you can start with *Basic Mode*, however if you have some experience with networking, it is recommended to always use *Extended Mode*.

6.6.2 Basic Mode

In *Basic mode*, the flow filters are composed of multiple sub-filters, which match against different protocol layers. Sub-filters are named after the protocol layer at which they are applied.

The configuration options available in basic mode are illustrated in Fig. 6.18.

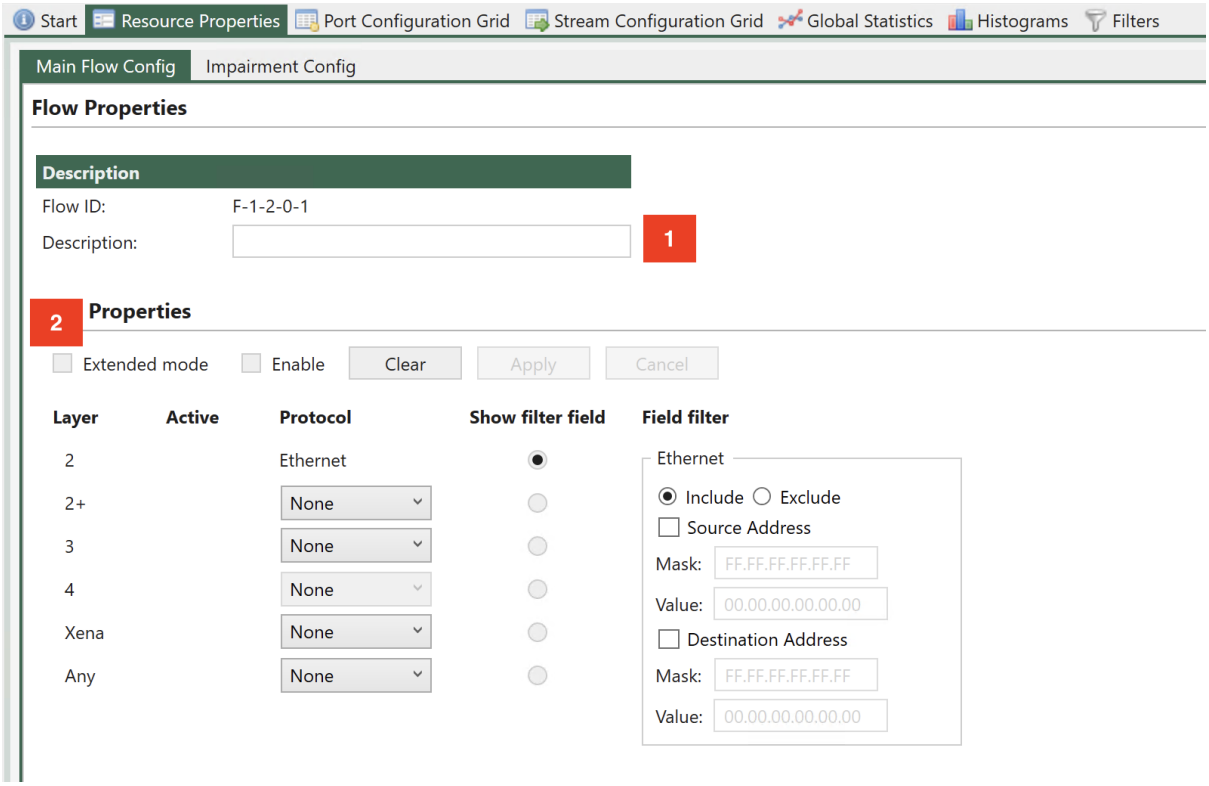


Fig. 6.18: Flow sub-filters.

Fig. 6.18 illustrates that it is possible to enter a flow description, which will be used in the UI to identify statistics

To enter basic mode you must de-select *Extended mode* (See Fig. 6.18).

> **Note**
>
> Corresponding CLI command: `PEF_MODE [fid,0] BASIC` and `PEF_APPLY[fid,0]`

In Basic mode, the sub-filters are used to define the encapsulation of packets, even when they are not used for filtering. E.g. defining 1 VLAN at sub-filter 2+ (as illustrated in :numref:`` 18) specifies that packets must have 1 VLAN, even when no fields in the VLAN are used for matching.

Flow filters can be defined on a port without enabling impairments. Simply configure the desired

flow sub-filters, click *Enable* and then *Apply*. If this is done, flow statistics will be updated, but no impairments are active.

To activate impairments at the flow level, the impairments must first be enabled at the port level (see Figure 38). If impairments are not enabled at the port level, all flow impairments are inactive.

All sub-filters allow specifying whether packets that match the corresponding sub-filter will be mapped to the flow (*Include* option)or whether packets that do not match the sub-filter are mapped to the flow (*Exclude* option).

Basic mode implements the shadow and working registers.

Ethernet Sub-Filter

A sub-filter can be applied at the Ethernet layer. The filter includes the following fields:

- Ethernet Destination MAC address (DMAC).
- Ethernet Source MAC address (SMAC).

The UI Ethernet sub-filter options are illustrated in Fig. 6.19.

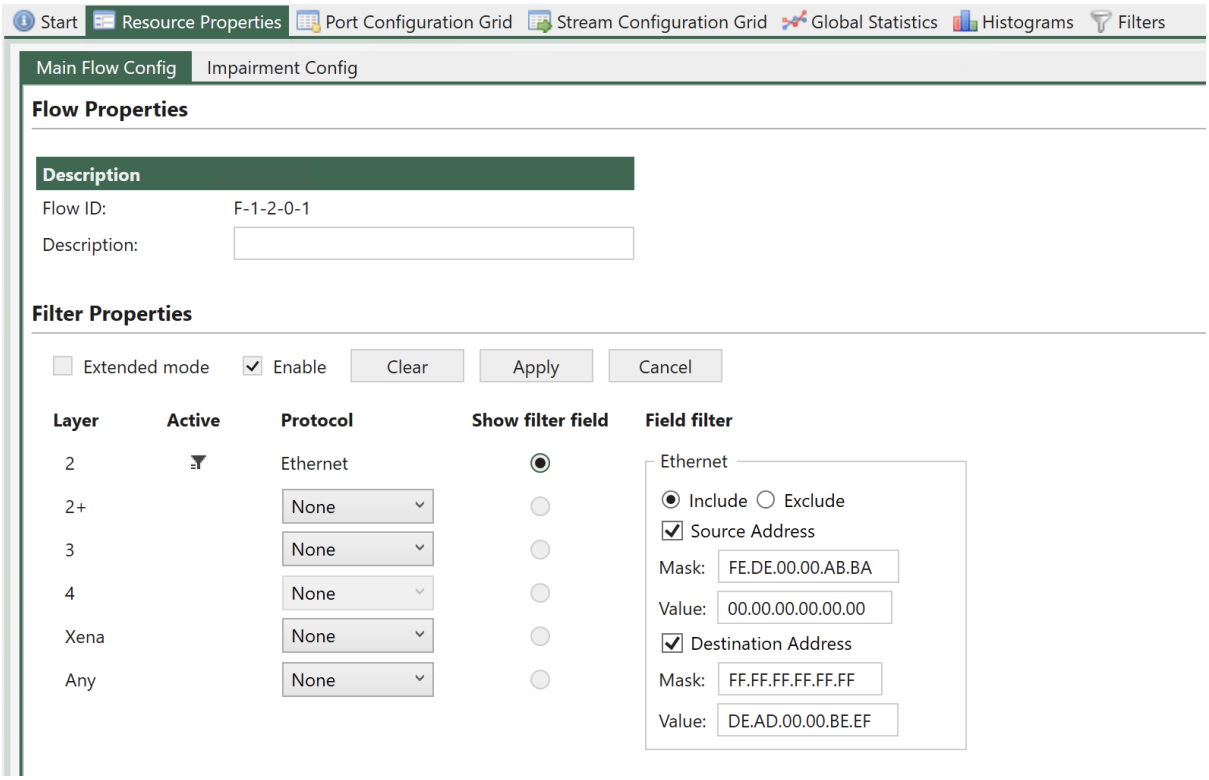


Fig. 6.19: Ethernet sub-filter.

To match against SMAC or DMAC, check the boxes *Source Address* or *Destination Address* respectively.

If none of these checkboxes are checked, the sub-filter will not be used for matching. However, the remaining sub-filters will always assume that packets are Ethernet packets, including DMAC, SMAC and Ethertype.

When matching against DMAC / SMAC, a 48 bit mask is configured to identify which bits in the MAC address are used for matching. Filtering bits configured = 1 will be used for matching.

Note

The example below illustrates how to filter for packets with `DMAC = 0xFEDE0000ABBA` and `SMAC = 0xDEAD0000BEEF` (all bits used for matching).

```
PEF_ETHSETTINGS[fid,0] AND INCLUDE
PEF_ETHSRCADDR [fid,0] ON 0xFEDE0000ABBA 0xffffffffffff
PEF_ETHDESTADDR[fid,0] ON 0xDEAD0000BEEF 0xffffffffffff
PEF_ENABLE     [fid,0] ON
PEF_APPLY      [fid]
```

Layer 2+ Sub-Filter

The Layer 2+ sub-filter allows the user to specify 1 VLAN, 2 VLANs or an *MPLS* label after the Ethernet header.

Single VLAN Tag

This sub-filter allows filtering based on a single VLAN tag. The filter includes the following fields:

- VLAN ID (*VID*)
- VLAN PCP bits

Selecting *1 VLAN Tag* causes the flow filter to verify that the *TPID* is `0x8100`, in addition to any VID / *PCP* matching configured.

The available UI configuration options for *1 VLAN Tag* are illustrated in Fig. 6.20.

To match against the VID, check the Tag checkbox and to match against the PCP bits, check the PCP checkbox.

The VID matching includes a mask to indicate that only selected bits (mask bit = 1) are used for matching. The PCP value is always matched against all 3 PCP bits in the UI.

If none of the checkboxes Tag or PCP are checked, no filtering is done on the VID / PCP values, but selecting the *1 VLAN Tag* option will indicate that packets mapped to this flow by higher layer sub-filters must have a single VLAN present. This includes checking `TPID = 0x8100`.

Note

The example below illustrates how to filter packets with `VID = 1234` and `PCP = 3` (all bits used for matching).

```
PEF_L2PUSE       [fid,0]              VLAN1 (1)
PEF_VLANSETTINGS [fid, 0]             AND (1) INCLUDE (1)
PEF_VLANTAG      [fid, 0, VLAN1 (0)] ON (1)  1234 0xFFF
PEF_VLANPCP      [fid, 0, VLAN1 (0)] ON (1)  3    0x7
```

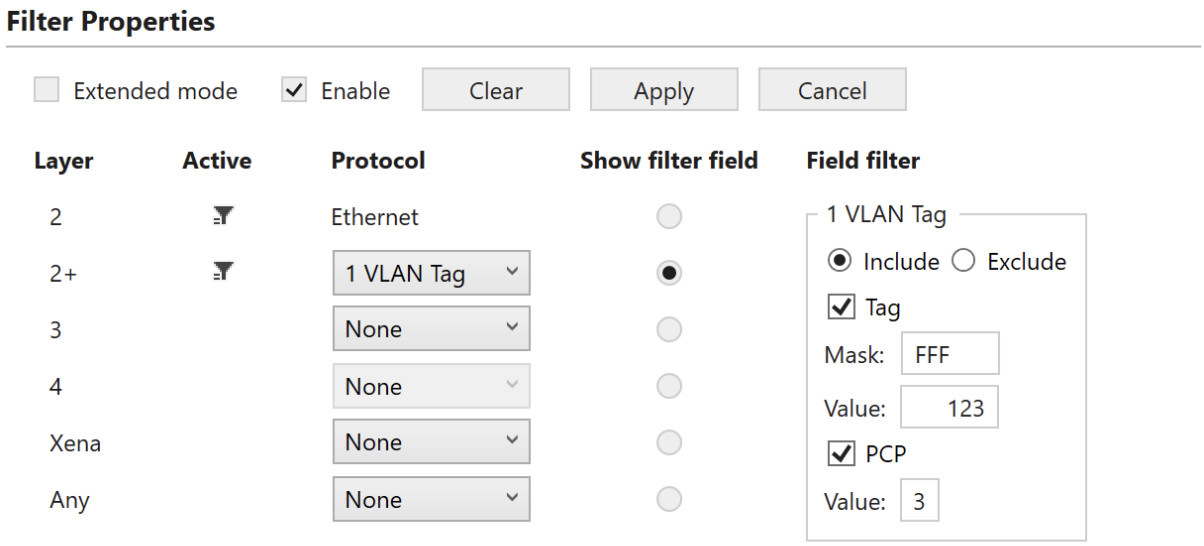


Fig. 6.20: Layer 2+ sub-filter (1 VLAN)

```
PEF_ENABLE        [fid,0]              ON
PEF_APPLY         [fid]
```

Double VLAN Tags

This sub-filter allows filtering based on a *2 VLAN Tags*. The filter includes the following fields:

- Inner VLAN VID.
- Inner VLAN PCP bits.
- Outer VLAN VID.
- Outer VLAN PCP bits.

Selecting *2 VLAN Tags* causes the flow filter to verify that the TPID of the inner VLAN is 0x8100 and the TPID of the outer VLAN is 0x88A8, in addition to any VID / PCP checking configured.

The available UI configuration options for *2 VLAN Tags* are illustrated in Fig. 6.21.

To match against the inner and / or outer VID, check the corresponding Tag checkbox and to match against the inner and / or outer PCP bits, check the corresponding PCP checkbox.

The VID matching includes a mask to indicate that only selected bits (mask bit = 1) are used for matching. The PCP value is always matched against all 3 PCP bits in the UI.

If none of the checkboxes Tag or PCP are checked, no filtering is done on the VID / PCP values, but selecting the *2 VLAN Tags* option will indicate that packets mapped to this flow by higher layer sub-filters must have two VLANs tags present. This includes the TPID matching described above.

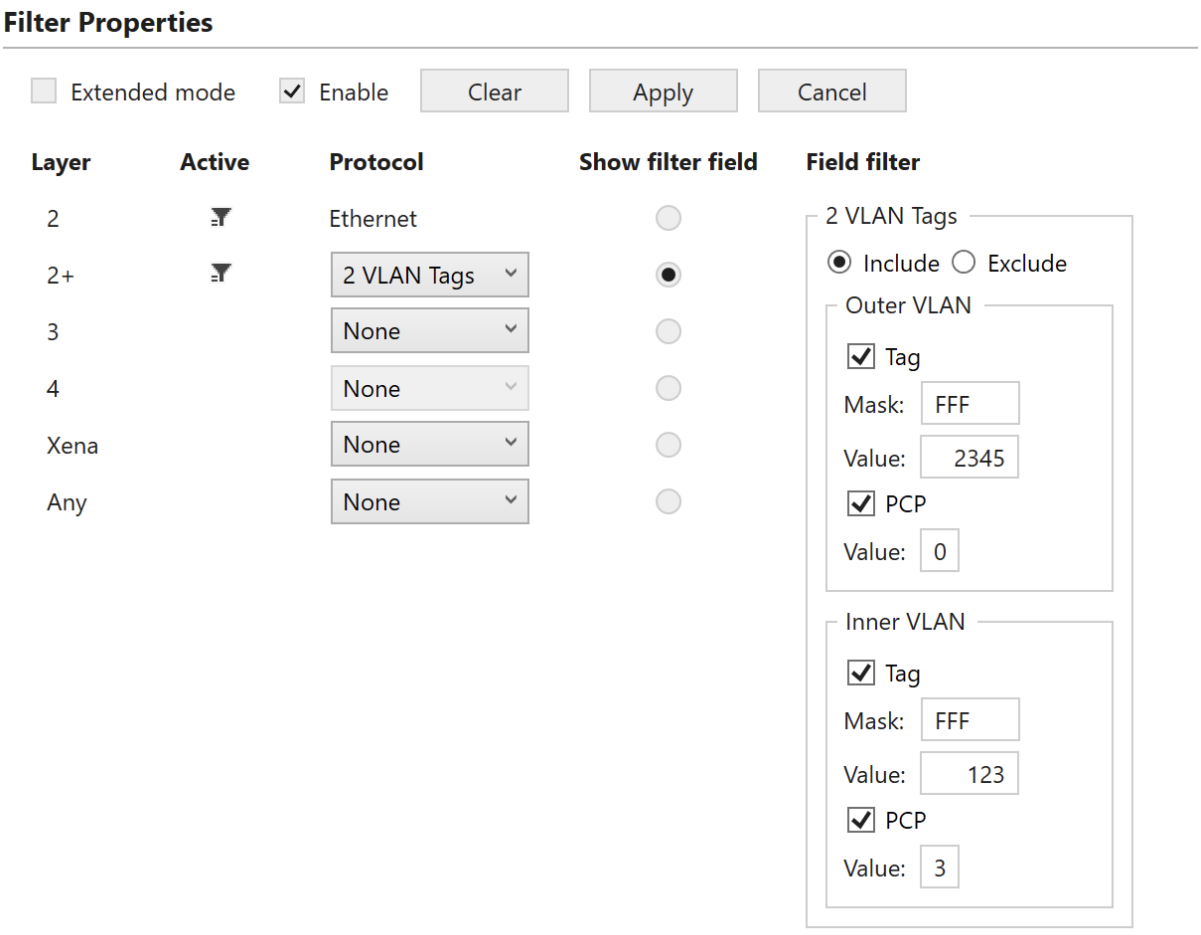


Fig. 6.21: Layer 2+ sub-filter (2 VLANs)

Note

The example below illustrates how to filter for packets with outer `VID = 2345`, `outer PCP = 0`, `inner VID = 1234` and `inner PCP = 3` (all bits used for matching).

```
PEF_L2PUSE        [fid,0]              VLAN2 (2)
PEF_VLANSETTINGS  [fid, 0]             AND (1) INCLUDE (1)
PEF_VLANTAG       [fid, 0, VLAN2 (1)] ON (1)  2345 0xFFF
PEF_VLANPCP       [fid, 0, VLAN2 (1)] ON (1)  0    0x7
PEF_VLANTAG       [fid, 0, VLAN1 (0)] ON (1)  1234 0xFFF
PEF_VLANPCP       [fid, 0, VLAN1 (0)] ON (1)  3    0x7
PEF_ENABLE        [fid,0]              ON
PEF_APPLY         [fid]
```

MPLS

This sub-filter allows filtering based on a *MPLS* label. The filter includes the following fields:

- MPLS label.
- MPLS traffic class (*TC*) bits.

The available UI configuration options for MPLS are illustrated in Fig. 6.22.

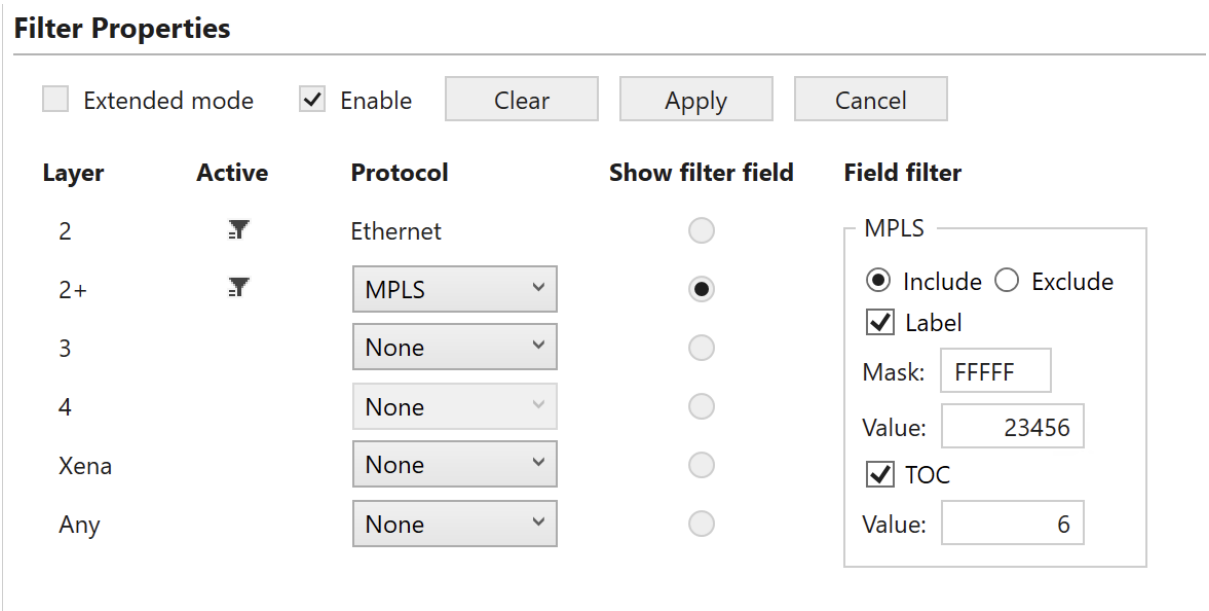


Fig. 6.22: Layer 2+ sub-filter (MPLS)

To match against the MPLS label, check the corresponding Label checkbox and to match against the Traffic Class (TC) bits, check the TOC checkbox. The label matching includes a mask to indicate that only selected bits (mask bit = 1) are used for matching. The TC value is always matched against all 3 TC bits in the UI.

If none of the checkboxes Label or Exp/ToC are checked, no filtering is done on the label / TC values, but selecting the MPLS option will indicate that packets mapped to this flow by higher layer sub-filters must have a MPLS label (32 bits) present.

Note

The example below illustrates how to filter for packets with label = 23456 and TC = 6 (all bits used for matching).

```
PEF_L2PUSE        [fid,0] MPLS (3)
PEF_MPLSSETTINGS [fid,0] AND (1) INCLUDE (1)
PEF_MPLSLABEL     [fid,0] ON (1) 23456 0xFFFFF
PEF_MPLSTOC       [fid,0] ON (1) 6 0x07
PEF_ENABLE        [fid,0] ON
PEF_APPLY         [fid]
```

Layer 3 Sub-Filter

The Layer 3 sub-filter allows the user to specify an IPv4 or an IPv6 header after the Ethernet header described in Section *Ethernet Sub-Filter* and the layer 2+ encapsulation described in Section *Layer 2+ Sub-Filter*.

There is no explicit selection between the IPv4 and the IPv6 filtering in the script commands, but if both are configured, only the later will be active.

IPv4

This sub-filter allows filtering based on the following IPv4 fields:

- Destination IP address (*DIP*)
- Source IP address (*SIP*)
- IPv4 *DSCP*

The available UI configuration options for IPv4 are illustrated in Fig. 6.23.

To match against IPv4 SIP, DIP or DSCP/ToC, check the corresponding checkboxes.

The DIP and SIP matching includes a mask to indicate that only selected bits (mask bit = 1) are used for matching. The DSCP value is always matched against all 6 bits in the UI.

If none of the checkboxes SIP, DIP or DSCP/ToC are checked, no filtering is done at the IPv4 layer, but selecting the IPv4 option will indicate that packets mapped to this flow by higher layer sub-filters must have a IPv4 header present.

Note

The example below illustrates how to filter for packets with `SIP = 11.22.33.44`, `DIP = 33.44.55.66` and `DSCP = 5` (all bits used for matching).

```
PEF_L3USE         [fid,0] IP4 (1)
PEF_IPV4SETTINGS [fid,0] ON (1) INCLUDE (1)
PEF_IPV4SRCADDR  [fid,0] ON (1) 11.22.33.44 0xFFFFFFFF
PEF_IPV4DESTADDR [fid,0] ON (1) 33.44.55.66 0xFFFFFFFF
PEF_IPV4DSCP      [fid,0] ON (1) 0x14 0xFC
```

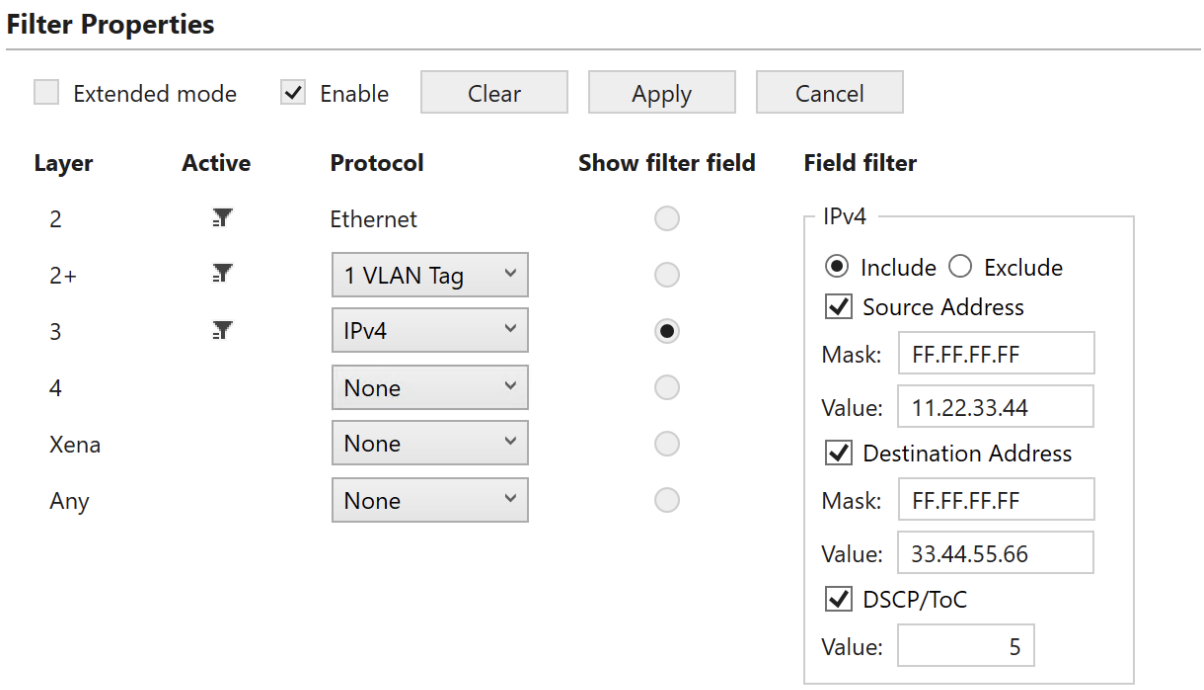


Fig. 6.23: Layer 3 sub-filter (IPv4)

```
PEF_ENABLE        [fid,0] ON
PEF_APPLY         [fid]
```

IPv6

This sub-filter allows filtering based on the following IPv6 fields:

- Destination IP address (DIP)
- Source IP address (SIP)
- IPv6 DSCP

The available UI configuration options for IPv6 are illustrated in Fig. 6.24.

To match against IPv6 SIP, DIP or DSCP/ToC, check the corresponding checkboxes.

The DIP and SIP matching includes a mask to indicate that only selected bits (mask bit = 1) are used for matching. The DSCP value is always matched against all 6 bits in the UI.

If none of the checkboxes SIP, DIP or DSCP are checked, no filtering is done at the IPv6 layer, but selecting the IPv6 option will indicate that packets mapped to this flow by higher layer sub-filters must have a IPv6 header present.

Note

The example below illustrates how to filter for packets with SIP =

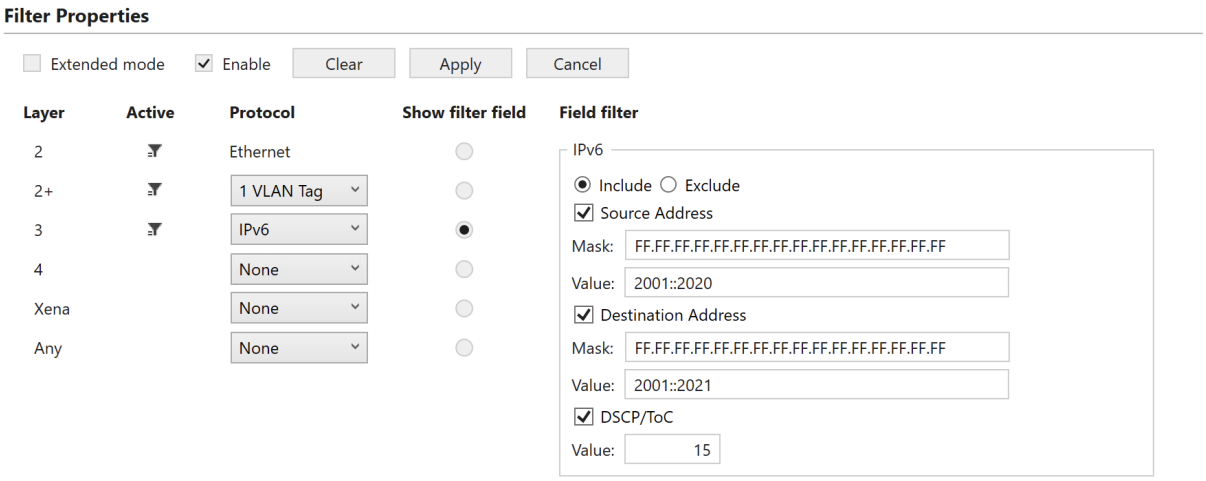


Fig. 6.24: Layer 3 sub-filter (IPv6)

```
2001:123:4455::6677:89A:BBCC, DIP = 2020:a98:8877::6655:432:1100 and
DSCP = 15 (all bits used for matching).
PEF_L3USE         [fid,0] IP6 (2)
PEF_IPV6SETTINGS [fid,0] ON (1) INCLUDE (1)
PEF_IPV6SRCADDR  [fid,0] ON (1)  0x20010123445500000000006677089ABBCC␣
↪0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
PEF_IPV6DESTADDR [fid,0] ON (1) 0x20200a9888770000000006655043211 00␣
↪0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
PEF_IPV6TC        [fid,0] ON (1) 0x3C 0xFC
PEF_ENABLE        [fid,0] ON
PEF_APPLY         [fid]
```

Layer 4 Sub-Filter

The Layer 4 sub-filter allows the user to specify a UDP or a TCP header after the Ethernet header described in Section *Ethernet Sub-Filter*, the layer 2+ encapsulation described in Section *Layer 2+ Sub-Filter* and layer 3 sub-filtering described in Section *Layer 3 Sub-Filter*.

> **Note**
>
> Notice that the UDP / TCP port configuration is only available if a IPv4/IPv6 header was configured at layer 3.

There is no explicit selection between the UDP port and the TCP port filtering in the script commands, but if both are configured, only the later will be active.

TCP

This sub-filter allows filtering based on the following TCP fields:

- TCP source port.
- TCP destination port.

The available UI configuration options for TCP filtering are illustrated in Fig. 6.25.

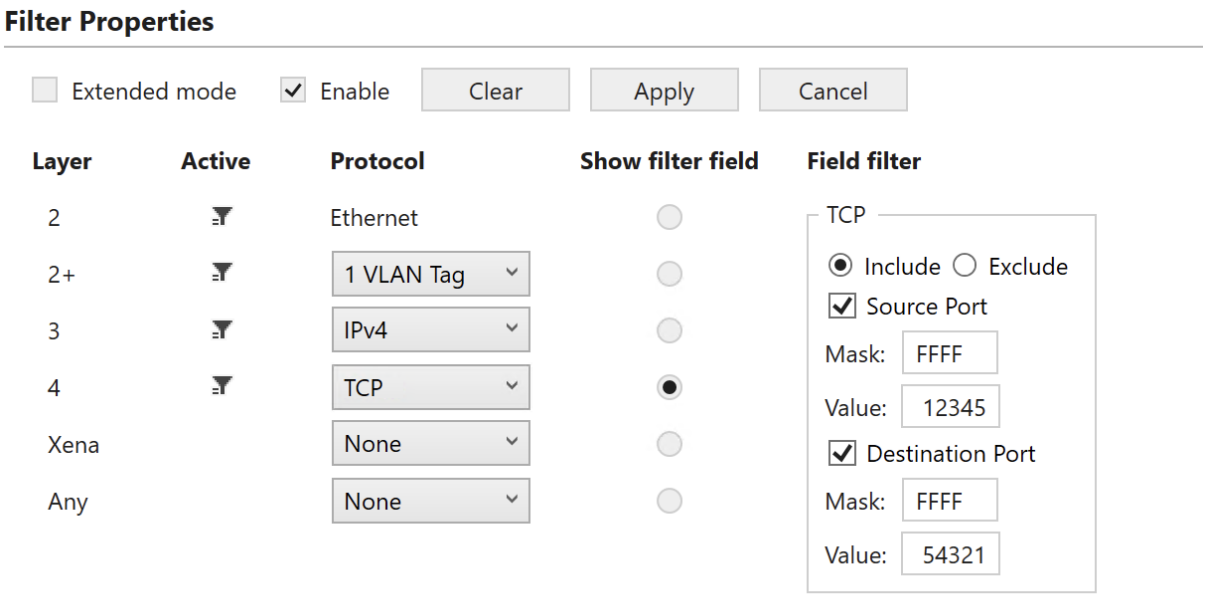


Fig. 6.25: Layer 4 sub-filter (TCP).

To match against the TCP source port or the destination port, check the corresponding checkboxes.

Both the TCP source port and destination port matching includes a mask to indicate that only selected bits (mask bit = 1) are used for matching. If none of the checkboxes Source Port or Destination Port are checked, no filtering is done at the TCP layer, but selecting the TCP option will indicate that packets mapped to this flow by other sub-filters must have a TCP header present.

Note

The example below illustrates how to filter for packets with TCP `source port = 12345` and `destination port = 54321` (all bits used for matching).

```
PEF_TCPSETTINGS [fid,0] ON (1) INCLUDE (1)
PEF_TCPSRCPORT  [fid,0] ON (1) 12345 0xFFFF
PEF_TCPDESTPORT [fid,0] ON (1) 54321 0xFFFF
PEF_ENABLE      [fid,0] ON (1)
PEF_APPLY       [fid]
```

UDP

This sub-filter allows filtering based on the following UDP fields:

- UDP source port.
- UDP destination port.

The available UI configuration options for UDP filtering are illustrated in Fig. 6.26.

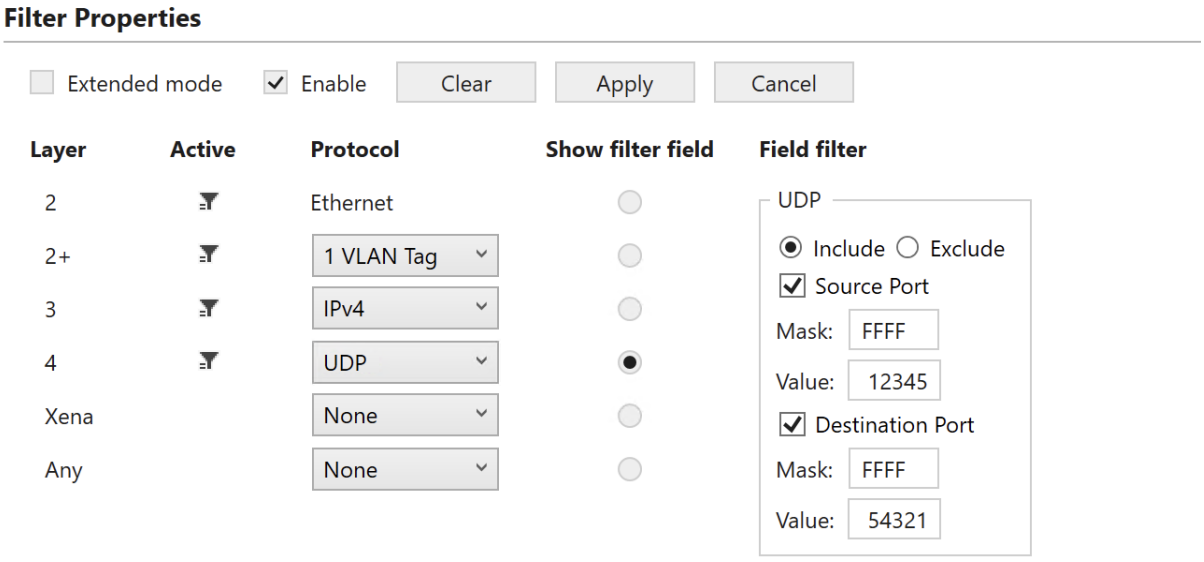


Fig. 6.26: Layer 4 sub-filter (UDP)

To match against the UDP source port or the destination port, check the corresponding checkboxes.

Both the UDP source port and destination port matching include a mask to indicate that only selected bits (mask bit = 1) are used for matching. If none of the checkboxes Source Port or Destination Port are checked, no filtering is done at the UDP layer, but selecting the UDP option will indicate that packets mapped to this flow by higher layer sub-filters must have a UDP header present.

Note

The example below illustrates how to filter for packets with UCP `source port = 12345` and `destination port = 54321` (all bits used for matching).

```
PEF_UDPSETTINGS [fid,0] ON (1) INCLUDE (1)
PEF_UDPSRCPORT  [fid,0] ON (1) 12345 0xFFFF
PEF_UDPDESTPORT [fid,0] ON (1) 54321 0xFFFF
PEF_ENABLE      [fid,0] ON (1)
PEF_APPLY       [fid]
```

TPLD Sub-Filter

When using a Xena traffic generator, it is possible to insert a Test Payload (TPLD) into the Tx packets. The TPLD includes a Test Payload Identifier (*TID*), which can be used for flow filtering in Chimera. Notice that the configured Chimera TPLD size must be the same as the TPLD size configured on the traffic generator.

The flow filters support matching against 16 TPLD TID values.

The available UI configuration options for TPLD TID filtering are illustrated in Fig. 6.27.

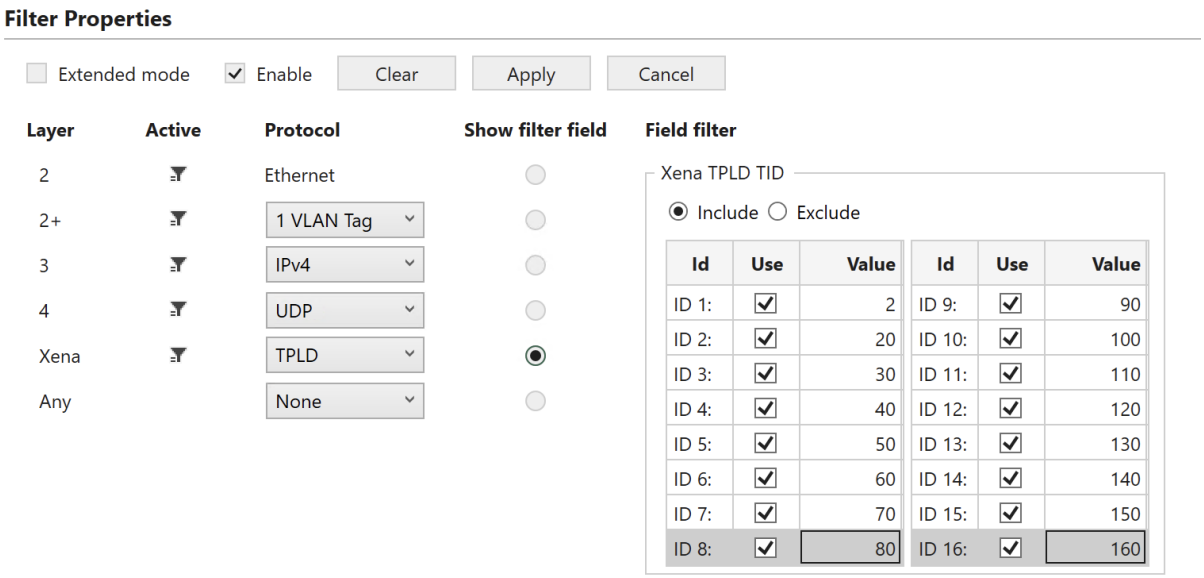


Fig. 6.27: TLPD sub-filter.

To filter based on the TPLD TID, check the Use checkbox and fill in the required TID value. Valid values for the TPLD TID are 0 to 2015 for default size (1023 for Micro) and must match what is configured in the Xena traffic generator.

Note

Configuring multiple TPLD TID values is done by configuring a TID value for multiple indices (Index 0 to 15). The example below illustrates how to filter for packets with TPLD TPID = 987 (index = 5).

```
PEF_TPLDSETTINGS [fid,0]   AND (1) INCLUDE (1)
PEF_TPLDCONFIG   [fid,0,5] ON (1) 987
PEF_ENABLE       [fid,0]   ON (1)
PEF_APPLY        [fid]
```

Any Field Sub-Filter

This filter can match 6 consecutive bytes at a configurable offset within the incoming packets. Furthermore, there is a mask to indicate which bits are to be used for matching.

Notice that in addition to the byte match configured here, the packet must match any encapsulation defined in sections 7.1 to 7.4. The available UI configuration options for Any Field filtering are illustrated in Fig. 6.28.

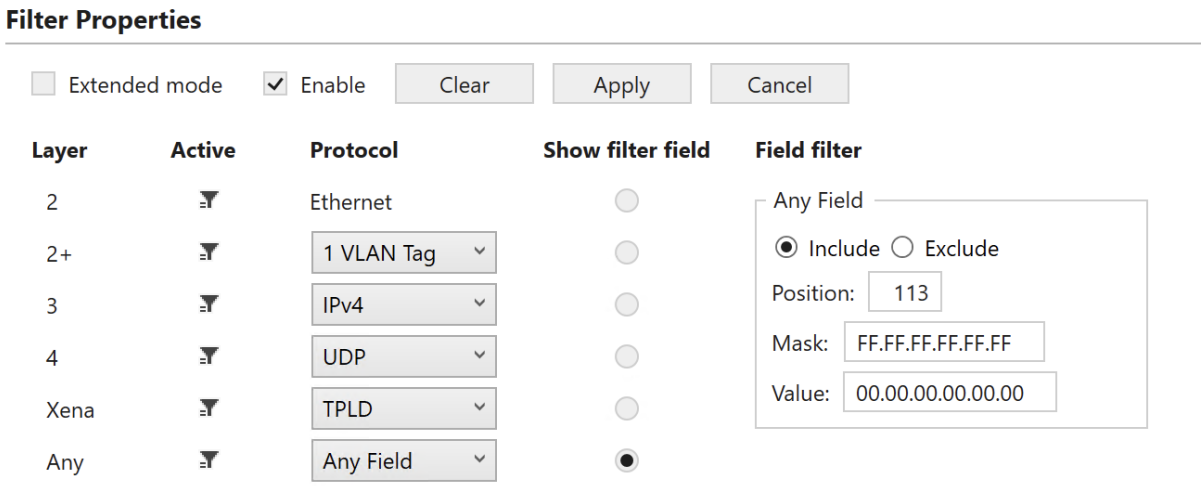


Fig. 6.28: Any Field filtering.

Parameter `Position` means byte offset in packet. Its range is from 0 to 122, step size is 1.

Note

The example below illustrates how to filter for packets including a 6 bytes value of 0x112233445566 starting at a byte offset of 113 (All bits used for matching).

```
PEF_ANYSETTINGS [fid,0] AND (1) INCLUDE (1)
PEF_ANYCONFIG   [fid,0] 113 0x112233445566 0xFFFFFFFFFFFF
PEF_ENABLE      [fid,0] ON (1)
PEF_APPLY       [fid]
```

6.6.3 Extended Mode

Extended filtering mode allows the user to filter on any pattern within the first 128 bytes of the packet.

The filtering is done by specifying a filter value of 128 bytes and filter mask of 128 bytes.

The extended filter is illustrated in Fig. 6.29.

Fig. 6.29 illustrates that if the filter mask of a given byte is non-zero (bytes 2, 3, 125 and 126), the corresponding filter value byte is matched against the corresponding byte in the incoming packet at the bit positions indicated by a 1 in the mask bit.

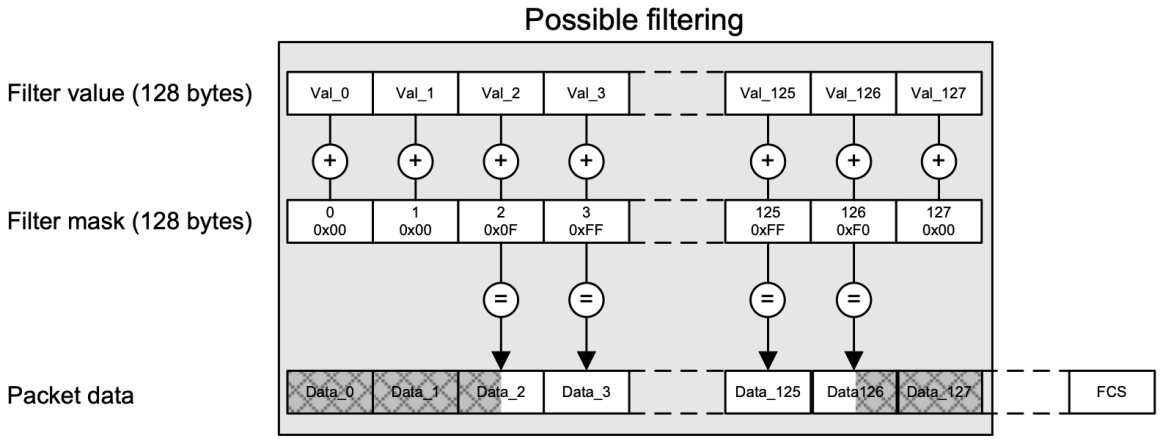


Fig. 6.29: Extended Filtering mode.

If the mask byte is zero (bytes 0, 1, 127), the corresponding byte in the incoming packet is ignored in the flow filter.

If an incoming packet is shorter that the specified filter, it will not match.

UI Configuration

To configure a flow filter using extended filtering, select the relevant flow and go to the tab *Resource Properties* → *Main Flow Config* and select *Extended mode*, which will bring up the UI interface illustrated in Fig. 6.30.

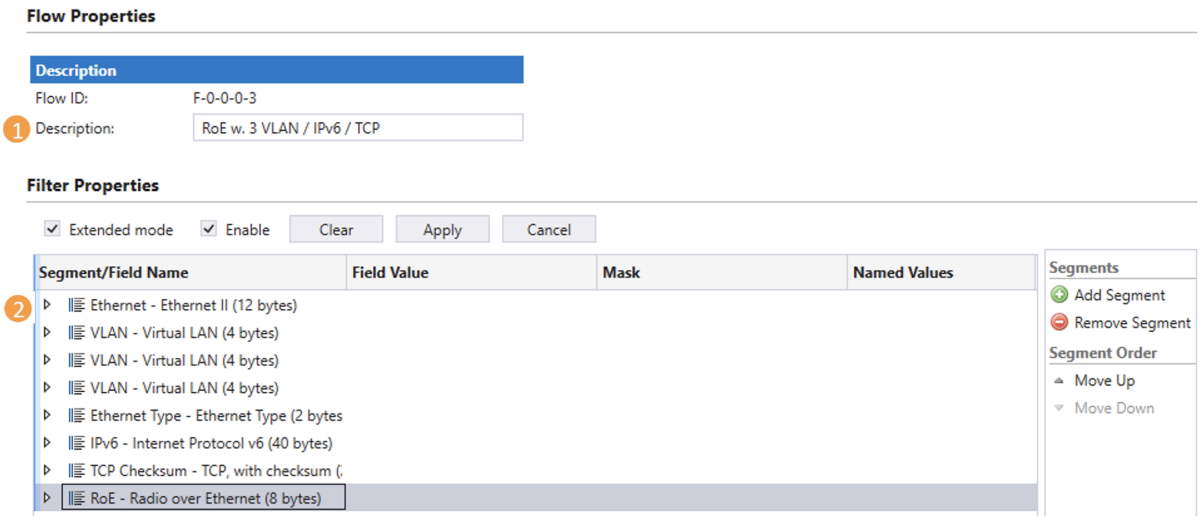


Fig. 6.30: Extended filtering.

At the top it is possible to assign a descriptive text to the flow (#1), which is used to identify the flow in the statistics tabs.

The protocol segment list illustrated in Fig. 6.30 (#2) can be manipulated using the *Add Segment* and *Remove Segment* buttons.

To add a protocol layer use the Add Segment button and select relevant protocols from the predefined protocol list illustrated in Fig. 6.31.

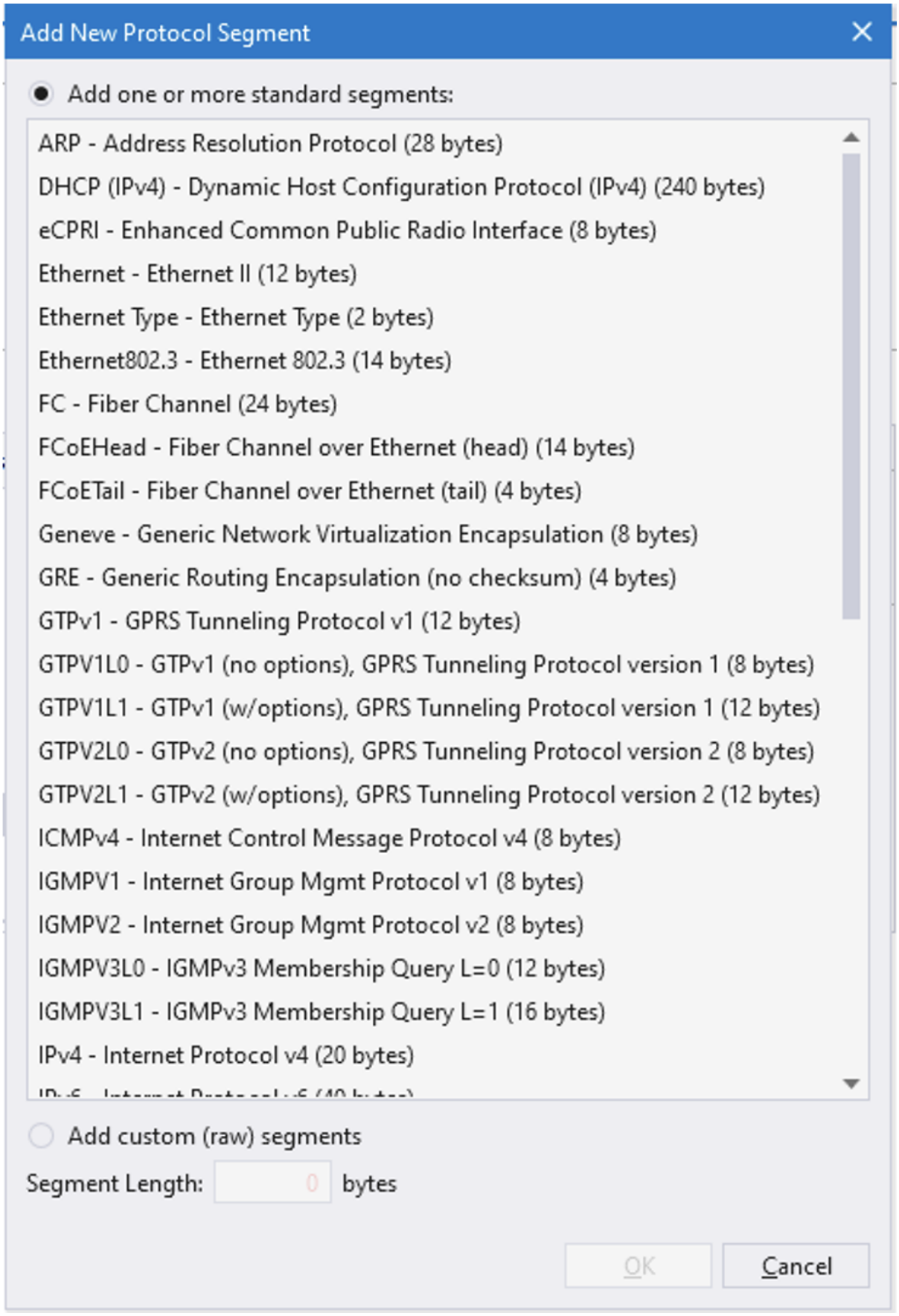


Fig. 6.31: Extended filtering protocols.

To add custom protocol segments use *Add custom (raw) segments* at the bottom of Fig. 6.31, by simply supplying the length of the segment in bytes.

> **Note**
>
> Notice that the total sum of the protocols can not exceed 128 bytes.

For each of the selected protocols, it is configurable which bits in the packet to match and which values to match against. An example of how to configure the IPv6 encapsulation is illustrated in Fig. 6.32.

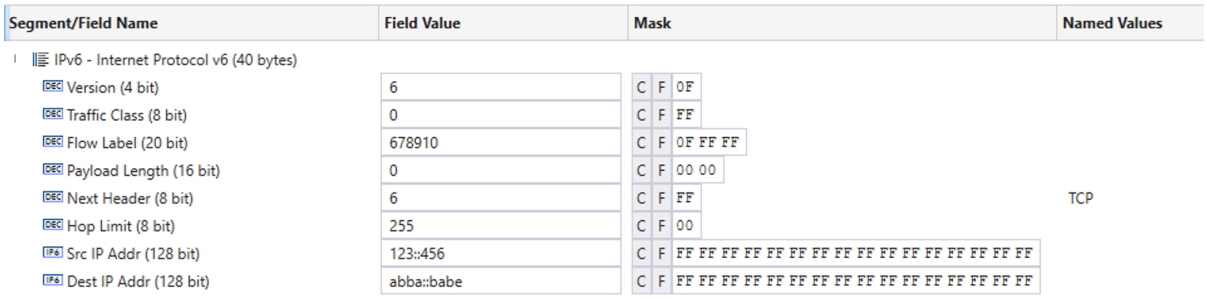


Fig. 6.32: Pv6 filter configuration.

Once the filter is complete, be sure to click *Enable* and press the *Apply* button, illustrated in Fig. 6.30.

For info on how to configure extended filtering using scripting, see Section *Scripting*.

Scripting

The bytes in the filter value and filter mask are structured as a list of protocols. Each protocol requires a certain number of bytes to be specified in the filter value and filter mask. Each protocol in the protocol list is identified using a protocol index, reflecting the number of the protocol in the list. The first protocol will be assigned protocol index = 1.

Fig. 6.33 illustrates an example of a protocol list including the protocols names, protocol indices (in parenthesis) and the length of each protocol in bytes.

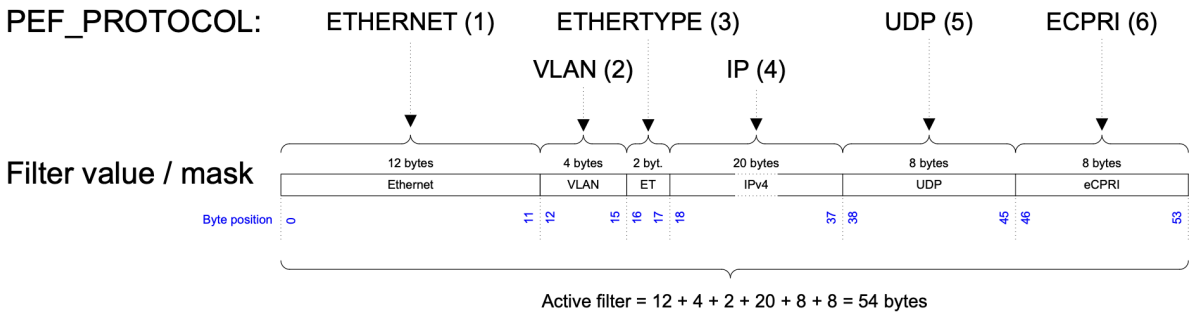


Fig. 6.33: Flow filter protocol specification.

The protocol list shown here contains the protocols listed in Table 6.11.

Table 6.11: Example of a protocol list

Protocol	Protocol index	Length (bytes)
ETHERNET	1	12
VLAN	2	4
ETHERTYPE	3	2
IPv4	4	20
UDP	5	8
eCPRI	6	8

The filter protocol list is specified using the script command `PEF_PROTOCOL` and takes as argument all the protocols to be included in the current filter. Notice, that the protocol list must include ETHERNET at protocol index = 1, to indicate that the incoming packets are Ethernet packets. In addition to the predefined protocols, it is possible to define custom protocols.

Note

The example below illustrates how to select extended filtering mode and configure the protocol list illustrated in Fig. 6.33.

```
PEF_MODE    [fid,0] EXTENDED
PEF_PROTOCOL[fid,0] ETHERNET VLAN ETHERTYPE IP UDP ECPRI
PEF_APPLY   [fid]
```

The combined length of the protocols configured using `PEF_PROTOCOL` defines the length of the active filter in bytes. Referring to the example from Fig. 6.33, this amounts to 54 active filter bytes. This implies that the first 54 bytes of the filter value and the filter mask must be configured. The remaining filter mask bytes will automatically be set to zero.

Note

Notice that the active filter can never be configured to be more than 128 bytes.

Once the protocol list has been defined, the protocol index implicitly assigned to every protocol, can be used to set the protocol filter value and protocol filter mask, using `PEF_VALUE` and `PEF_MASK` respectively.

Using the protocol index as input to `PEF_VALUE` and `PEF_MASK` only the value of that protocol is modified. The protocol value must reflect the byte values in the protocol being referenced. Figure 34 illustrates the configuration of the 8 bytes in the UDP protocol.



Fig. 6.34: Configuring UDP protocol segment.

Note

The example below illustrates how to configure the UDP (protocol index = 5) protocol filtering from the example in Fig. 6.33 with UDP `source Port = 0x1122` and UDP `length = 0x3344`, while UDP destination port and UDP checksum are ignored.

```
PEF_VALUE[fid,0,5] 0x1122000033440000
PEF_MASK [fid,0,5] 0xFFFF0000FFFF0000
PEF_APPLY[fid]
```

Protocol index 0 has a special significance. It is used to work on the entire active filter value and filter mask as a single array of bytes.

Once the protocol list has been defined, you can use index 0 to define the entire filter value / mask or use the individual protocol indices to configure the protocols individually. I.e. using protocol index 0 for the example in Fig. 6.33 will modify all 54 bytes of the active filter value / mask.

6.7 Flow Impairment

6.7.1 Overview

The impairments currently supported in Chimera are:

- Ingress policing
- Packet drop
- Misordering
- Packet corruption (FCS/IP/TCP/UDP)
- Packet duplication
- Latency / jitter
- Egress shaper

Chimera supports a variety of distributions that can be used to apply the supported impairments. The distribution for a given impairment determines how often the impairment is applied to the flow in terms of time or number of packets between the impairments.

The complete set of distributions and which impairments they support is described in detail in Section *Impairments Distributions*.

Furthermore, the impairments can be turned on and off automatically using a scheduler. The scheduler allows the impairment to be applied in 2 modes:

- Continuous
- Repeated Pattern

Note

Notice that *Accumulate and Burst* and *Random Burst* are not continuous by nature and implement a modified scheduler function. See Section *Scheduler Functions* for details.

When configured for *Continuous Mode*, the impairment will be applied continuously to the flow until turned off by the user.

When configured for *Repeated Pattern Mode*, the scheduler allows specifying a *Duration* and a *Repeat* period. The scheduler will (re-)start the impairment at intervals equal to the configured repeat period. When started, the impairment will be on for a period equal to the configured duration after which it is turned off. This pattern is repeated until the impairment is turned off by the user.

The repeat pattern configuration is illustrated in Fig. 6.35.

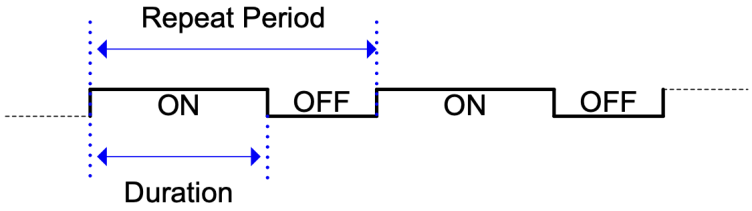


Fig. 6.35: Scheduler ON / OFF function.

The details of the scheduling function are described in *Scheduler Functions*.

Integration with Xena TG

When using Chimera together with a Xena traffic generator, the UI supports visually associating the connected TG and network impairment ports for better overview and ease of configuration.

Fig. 6.36 illustrates how to associate two TG and network impairment ports.

To associate TG and network impairment ports:

1. Select the two ports to be associated using the `Ctrl` button.
2. Right click with the mouse on one of the ports.
3. Select Associate TG and network impairment Ports.

Selecting the TG port in the UI, you can now see the associated Chimera port and access the Chimera port impairment configuration. This is illustrated in Figure 37.

Note

Notice that the port association is an UI property only. It is not available with script commands.

The following sub-sections describe the configuration of impairments using script commands. Users that solely use the UI can go to Section *Impairments Types*.

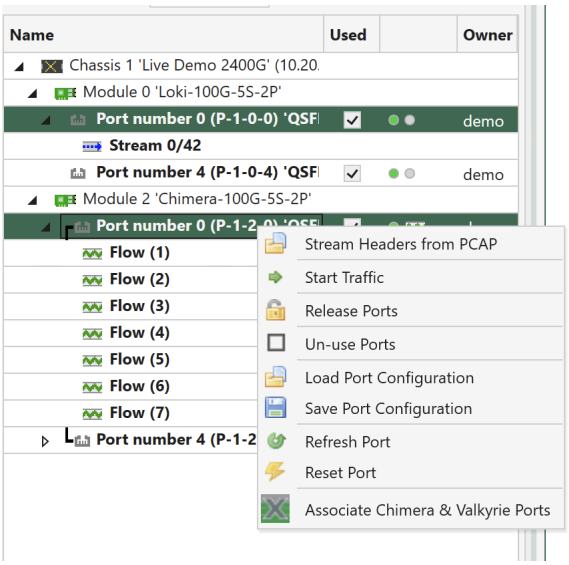


Fig. 6.36: Associating the TG and network impairment ports in the UI.

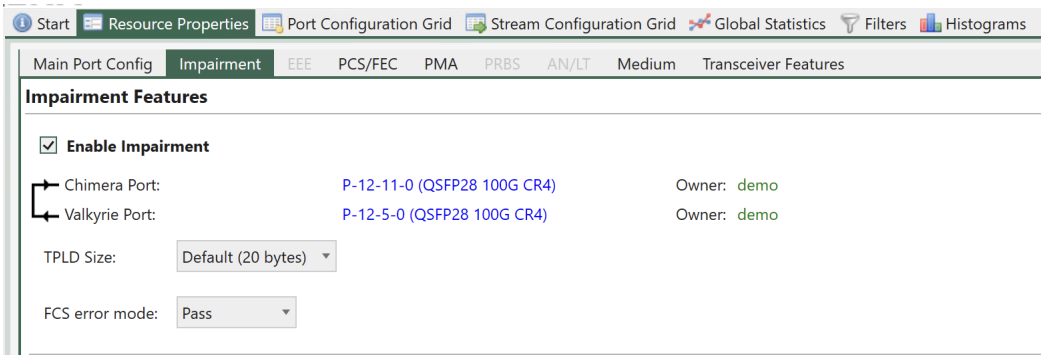


Fig. 6.37: Configuring the Chimera port impairment from the TG port.

Script Configuration of Impairments

When using script commands to configure impairments, the script commands are grouped into 3 groups:

- Impairment configuration
- Scheduler configuration
- Distribution configuration

In addition to the configuration, the impairment can be turned on and off with the given configuration.

To configure the different impairments, each impairment is assigned an **impairment ID** (*iid*) as illustrated in Table 6.12.

Table 6.12: Impairment IDs

Impairment ID	Impairment Name
0	DROP
1	MISORDER
2	DELAY / JITTER
3	DUPLICATION
4	CORRUPTION
5	POLICER
6	SHAPER

The impairment ID is used in combination with the filter ID (fid) in the scheduler command and the distribution commands to configure the impairments in each flow. The fid / iid addressing is illustrated in Table 6.13.

Table 6.13: How to configure impairments using [fid,iid]

Flow to configure	Impairment to configure	How to address: [fid, iid]
0	Drop	[fid=0, iid=0]
3	Corruption	[fid=3, iid=4]
7	Misordering	[fid=7, iid=1]

The impairment configuration is described in the following sub-sections, followed by two configuration examples in Section *Configuration Example*.

Impairment Configuration

Some impairments need configuration of the impairment event itself. E.g. for packet corruption, it is required to configure at which level (FCS/IP/TCP/UDP) the corruption takes place, while for drop, there is nothing to configure, because when a drop event occurs, the packet is simply dropped.

Table 10 lists the impairments which have an associated impairment configuration script command.

Table 6.14: Impairment configuration commands.

Impairment Name	Configuration command
Drop	N.A.
Misordering	`PE_MISORDER[fid]`
Delay / Jitter	N.A.
Duplication	N.A.
Corruption	`PE_CORRUPT[fid]`
Policer	`PE_BANDPOLICER[fid]`
Shaper	`PE_BANDSHAPER[fid]`

The commands all take a fid as input. The details of the impairment script commands are described in Section *Impairments Types*.

Scheduler Configuration

A single scheduler command is defined for all impairments.

`PED_SCHEDULE[fid,iid]`

The schedule command takes fid and iid to identify the impairment to configure. The scheduling command is described in detail in Section *Scheduler Functions*.

Distribution Configuration

The distribution commands are used to configure when to apply the selected impairment to the packets in the flow. The supported distribution commands are listed in Table 11.

Table 6.15: Distribution configuration commands

Distribution Name	Configuration command
Off	PED_OFF[fid, iid]
Constant Latency	PED_CONST[fid, iid]
Accumulate & Burst	PED_ACCBURST[fid, iid]
Step	PED_STEP[fid, iid]
Fixed probability	PED_FIXED[fid, iid]
Random probability	PED_RANDOM[fid, iid]
Fixed burst	PED_FIXEDBURST[fid, iid]
Random Burst	PED_RANDOMBURST[fid, iid]
Gilbert-Elliot	PED_GE[fid, iid]
Bit Error Rate	PED_BER[fid, iid]
Uniform	PED_UNI[fid, iid]
Gaussian (Normal)	PED_GAUSS[fid, iid]
Poisson	PED_POISSON[fid, iid]
Gamma	PED_GAMMA[fid, iid]
Custom	PED_CUSTOM[fid, iid]

Note

Notice the PED_OFF distribution. This is the default distribution at power up. It contains no impairment configuration and the impairment is turned off. Assigning the PED_OFF distribution to an impairment will clear all impairment configuration and turn off the impairment.

The distribution commands take fid and iid to identify the impairment to configure. Depending on the type of impairment, the distribution commands will either specify the number of packets between applying the impairment (inter-packet) or the delay applied to each packet (latency / jitter).

The details of the distribution commands are described in Section *Impairments Distributions*.

Configuration Example

This sub-section contains two examples of how to configure impairments using script commands.

FCS corruption with fixed drop probability:

This example illustrates how to configure a fixed drop probability of 5.4321 % at the Ethernet FCS layer with a duration of 1 sec. and a repeat period of 1.5 sec.

This will be configured for fid = 0 (Port default flow).

The corruption iid = 4 is found in Table 6.12.

- Impairment configuration

Configure impairment corruption at the Ethernet FCS level `PE_CORRUPT[0] ETH`

- Scheduler configuration

Configure duration = 1 sec and repeat period = 1.5 sec `PED_SCHEDULE[0,4] 1000 1500`

- Distribution configuration

Configure fixed distance drop probability of 5.4321 % `PED_FIXED[0,4] 54321`

- Turn on the impairment.

```
PED_ENABLE[0,4] ON
```

- Turn off impairment.

```
PED_ENABLE[0,4] OFF
```

Gaussian Jitter:

This example illustrates how to configure a Gaussian jitter distribution with an average = 50 us and standard deviation = 2 us. This will be configured for fid = 7 (Highest priority flow filter).

The latency / jitter iid = 2 is found in Table 6.12.

- Impairment configuration

There is no impairment configuration for latency / jitter. (See Table 6.14).

- Scheduler configuration

The scheduling command is generally not supported for latency / jitter impairments.

- Distribution configuration

Configure Gaussian distribution with average delay = 50 us and std dev = 2 us

```
PED_GAUSS[7,2] 50000 2000
```

- Turn on the impairment

```
PED_ENABLE[7,2] ON
```

- Turn off impairment

```
PED_ENABLE[7,2] OFF
```

6.7.2 Impairments Types

This section describes the impairments which are configured on a per flow basis. For a definition of flows, see Section *Packet Flow*.

As described in Section *Overview*, each impairment can be assigned a set of distributions (see Section *Impairments Distributions*) and a scheduler (see Section *Scheduler Functions*). This section will focus on the Chimera impairments, including examples of how to configure selected distributions and the scheduler. However, for an elaborate description of the distributions available for each impairment, see section 11. For an elaborate description of the scheduler.

To enable flow impairments, the impairments must be enabled on the port level. This is illustrated Fig. 6.38.

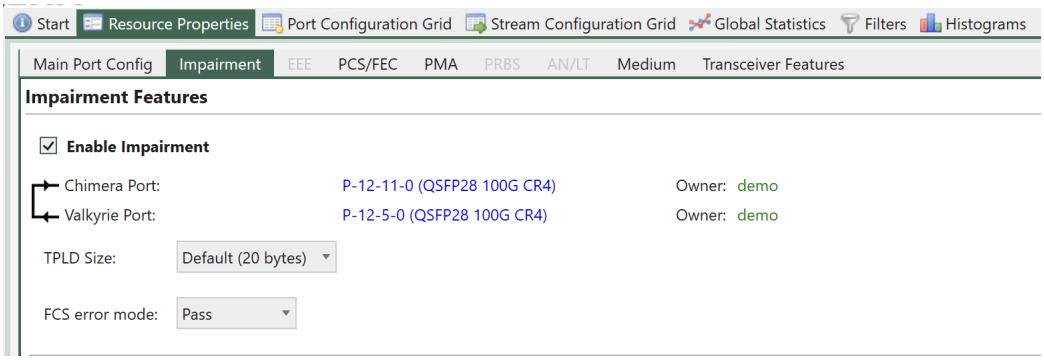


Fig. 6.38: Enable impairments.

If impairments are configured at the flow level but not enabled at the port level, they will not have any effect on the flow.

Note

The following example illustrates how to enable flow impairments on a port.

```
P_EMULATE ON
```

To configure the impairments for a given flow, first select the flow to configure using UI. Fig. 6.39 illustrates how to select flow 0 (Port default flow).

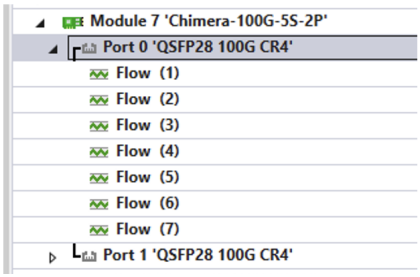


Fig. 6.39: Select flow to configure in the UI.

Then, expand the impairment window for the impairment to configure. The available impairment windows are illustrated Fig. 6.40.

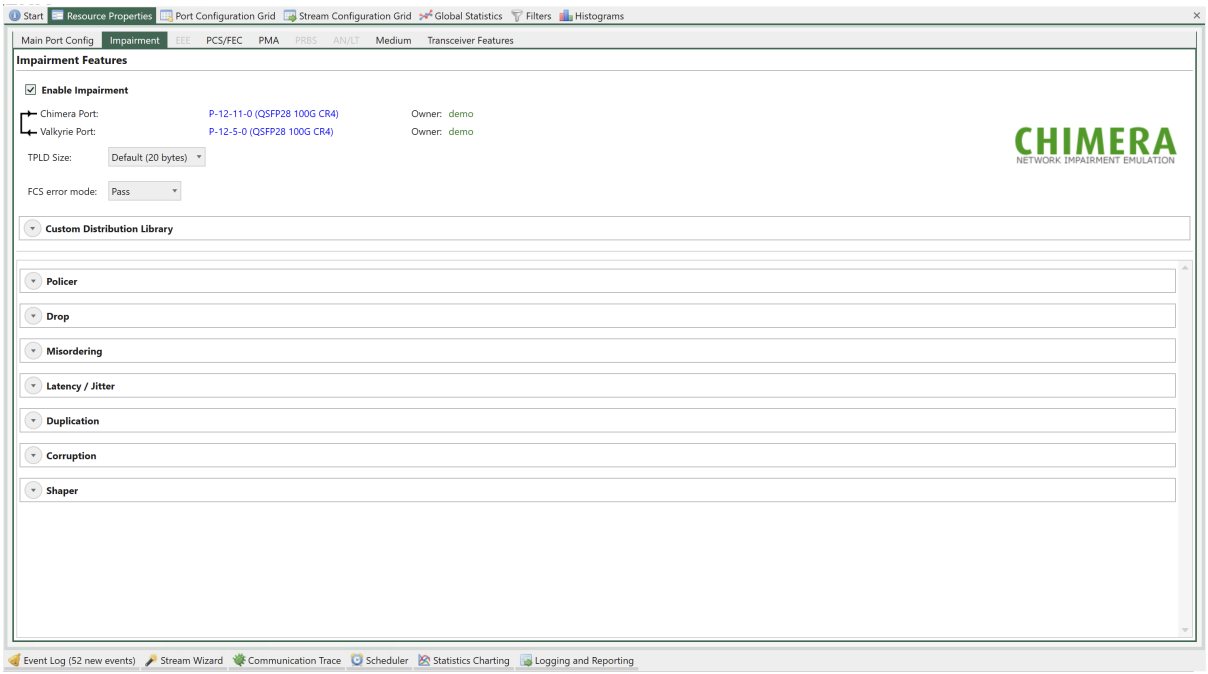


Fig. 6.40: Select impairment to configure in the UI.

The configuration of each impairment is explained in the following sub-sections.

See also

Read more about *Impairments Distributions* and *Scheduler Functions*.

Duplication (iid = 3)

Packet duplication will duplicate a packet, so the packet is transmitted twice in the Ethernet packet flow. The duplicate packet is inserted right after the original packet.

Notice that packet duplication is located after the shapers, just before the Tx port in the impairment pipeline. I.e., enabling packet duplication will add packets to the packet flow after the

shapers, hereby increasing the BW compared to what was configured in the shaper. For further details on shapers, see section 10.5.

Configuration of duplication using the UI is illustrated Fig. 6.41.

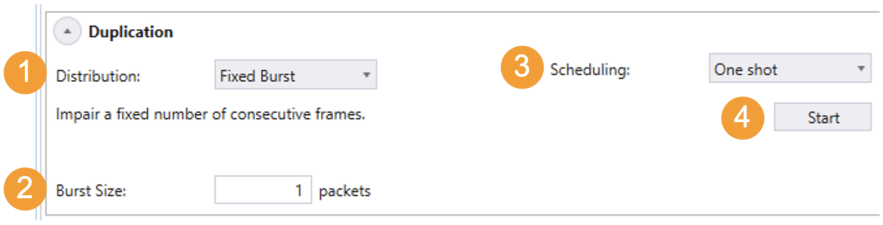


Fig. 6.41: Duplication configuration in the UI.

(There is no configuration of the impairment for duplication.)

To configure duplication:

1. Select the relevant distribution from the distribution dropdown menu (e.g. Fixed Burst).
2. Supply the parameters to configure the selected distribution (e.g. Burst Size = 1).
3. Configure the scheduler (e.g. One shot).
4. Press *Start* to activate the impairment.

The configuration shown Fig. 6.41 41 will cause the next packet to be duplicated, implementing a burst of 1 duplicated packet.

Note

For every impairment, distribution *Fixed Burst* can be used to manually insert a number of consecutively impairments. Simply configure the burst size and press *Start*.

Note

The example below illustrates how to duplicate a single packet (Schedule = One shot).

```
PED_SCHEDULE[fid,3]      1    0
PED_FIXEDBURST[fid,3]    1
PED_ENABLE[fid,3]        ON
```

Packet Drop (iid = 0)

Packet drop will cause packets to be removed from the Ethernet packet flow. Configuration of drop using the UI is illustrated Fig. 6.42.

(There is no configuration of the impairment for Drop).

To configure Drop:

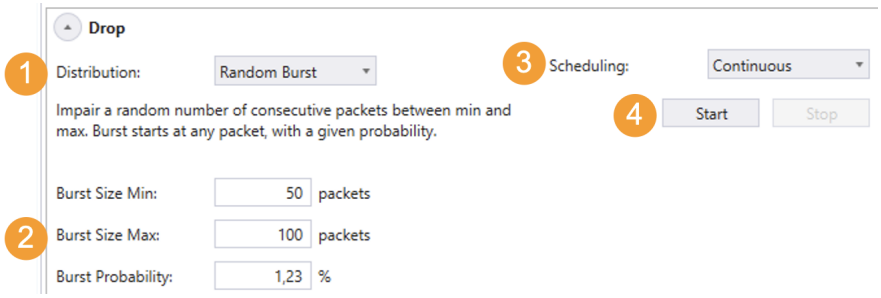


Fig. 6.42: Drop configuration in the UI.

1. Select the relevant distribution from the distribution dropdown menu (e.g. Random Burst).
2. Supply the parameters to configure the selected distribution. (e.g. Burst Size Min = 50, Burst Size Max = 100 and Burst Probability = 1.23 %.)
3. Configure the scheduler (e.g. Continuous)
4. Press *Start* to activate the impairment.
5. To stop dropping packets press *stop*.

The configuration shown Fig. 6.42 will cause a drop burst to start at any packet with a probability of 1.23 %. The size of each burst will be selected randomly between 50 packets and 100 packets.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,0]       1   0
PED_RANDOMBURST[fid,0]    50 100 1230
PED_ENABLE[fid,0]         ON
```

Misordering (iid = 1)

Misordering causes packets to be taken out of the Ethernet packet flow and delayed for a configurable number of packets, after which they are re-inserted into the packet flow. The number of packets that the packet is delayed is referred to as the *Misorder Depth*.

At any point in time, only a single packet can be in queue to be re-inserted. As a result, the following limitation applies to the values of probability and depth.

The number of distributions which support misordering are limited to:

- Fixed Burst with burst size = 1
- Fixed Rate

Configuration of misordering using the UI is illustrated Fig. 6.43.

To configure misordering:

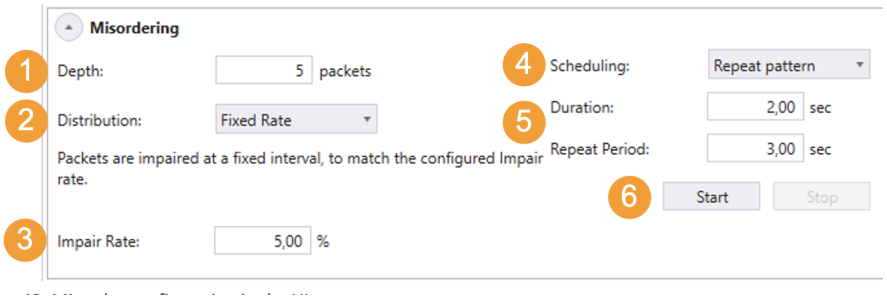


Fig. 6.43: Misorder configuration in the UI.

1. Configure the misordering depth.
2. Select the relevant distribution from the distribution dropdown menu (e.g. Fixed Rate).
3. Supply the parameters to configure the selected distribution (e.g. Impair Rate = 5%)
4. Select the relevant scheduler function from the dropdown menu (e.g. Repeat Pattern)
5. Configure the selected scheduler function (e.g. Duration = 2 sec, Repeat Period = 3 sec)
6. Press *Start* to activate the impairment.

The configuration above will cause 5% of the packets to be extracted from the packet flow and re-inserted after 5 packets.

Note

The example below illustrates how to configure misordering as illustrated Fig. 6.43.

```
PE_MISORDER  [fid]   5
PED_SCHEDULE [fid,1] 2000 3000
PED_FIXED    [fid,1] 50000
PED_ENABLE   [fid,1] ON
```

Corruption (iid = 4)

Chimera supports packet corruption at the following protocol layers:

- Ethernet FCS
- IP
- TCP
- UDP

Corruption is done by altering a bit in the checksum for the configured protocol. Furthermore, when corruption is done at IP / TCP / UDP level, the Ethernet FCS is corrected, so the checksum error only appears at the configured level.

Note

Note: when configuring corruption at IP / TCP / UDP level, the flow filter must include the selected layer in the flow filter.

If corruption is configured at the UDP level, the flow filter must include all relevant protocols:

- Ethernet
- (optionally) VLAN(s) / MPLS
- IPv4 / IPv6
- UDP

This implies that IP / TCP / UDP corruption is not supported for the port default flow (flow = 0), because this flow has no filter. Configuration of UDP checksum corruption using the UI is illustrated Fig. 6.44.

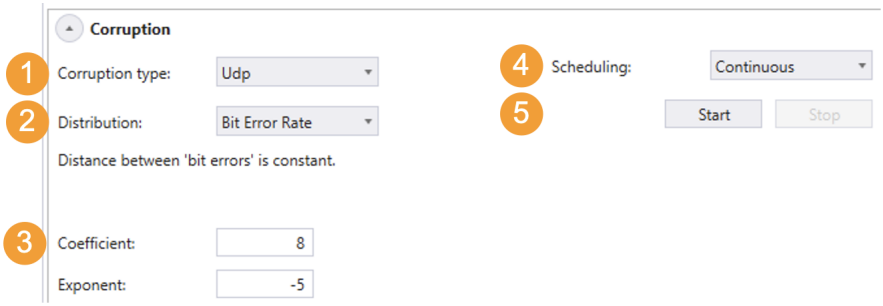


Fig. 6.44: UDP checksum configuration in the UI.

To configure UDP checksum corruption:

1. Configure the protocol layer to corrupt (e.g. UDP).
2. Select the relevant distribution from the distribution dropdown menu (e.g. Bit Error Rate).
3. Supply the parameters to configure the selected distribution (e.g. Coefficient = 8, Exponent = -5)
4. Select the relevant scheduler function from the dropdown menu (e.g. Continuous).
5. Press *Start* to activate the impairment.

The configuration above will cause an UDP checksum error to be inserted into the packet flow for every 8 * 10-5 bits of packet data.

Note

The example below illustrates how to configure corruption as illustrated Fig. 6.44.

```
PE_CORRUPT  [fid ]   UDP (3)
PED_SCHEDULE[fid7,4] 1   0
PED_BER     [fid7,4] 8   -5
PED_ENABLE  [fid,0]  ON
```

Flow BW Control

Chimera implements policers at every flow input, which will drop all packets that exceed the configured bandwidth (*CIR*) and burst size (*CBS*).

Likewise, Chimera implements shapers at the output, which will shape the outgoing traffic to a configurable bandwidth (CIR) and burst size (CBS). If excess packets are available, there is a buffer of configurable size (*Buffer size*) for storing excess packets. If the buffer overflows due to shaper BW limiting, packets will be dropped.

Policers and shapers implement a leaky bucket and update the current fill level. The fill level is reduced with the rate specified by CIR and increased with the packet size when a packet is forwarded. If the bucket fill level is above the CBS when a packet arrives at the policer / shaper, the packet will be dropped (policer) or held back (shaper). If the fill level is below the CBS, the packet will be forwarded, and the size of the packet is added to the fill level.

It is configurable whether the policer and shaper will be applied at layer 1 or layer 2. When configured for layer 2, only the Ethernet packets starting with the DMAC and ending with the FCS are counted as part of the traffic. When shaping at layer 1 the minimum IPG (= 12 bytes) and the preamble (= 8 bytes) are considered part of the traffic.

When configuring the parameters of the policer and shaper, the following limits apply:

Table 6.16: Policer and shaper limits

Parameter	Legal values	Comments	Step size
CIR	0 to 1,000,000	Value is multiplied by 100 kbps	1 (= 100 kbps)
CBS	0 to 2,097,152	Bytes	1 byte
Buffer size	0 to 2,097,152	Bytes	128 bytes

Furthermore, the CBS must be configured >= maximum packet size in flow + 64 bytes to function correctly.

Important

Policers and shapers do not support any impairment distributions or scheduler functions.

Ingress policers (iid = 5)

Configuration of the ingress policer using the UI is illustrated Fig. 6.45.

To configure ingress policer:

1. Select at which layer to implement the policer from dropdown menu (e.g. Layer 1).
2. Supply the parameters to configure policer (e.g. CIR = 9.9 Gbps, CBS = 2 Mbyte)
3. Press *Start* to activate the policer.

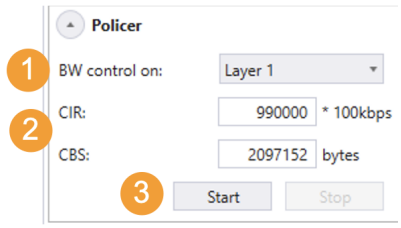


Fig. 6.45: Configuring flow policer in the UI.

Note

The example below illustrates how to configure the policer as illustrated Fig. 6.45.

```
PE_BANDPOLICER [fid] ON L1 990000 2097152
```

Egress Shapers (iid = 6)

Notice that the shapers are located before the packet duplication in the impairment pipeline. I.e., if packet duplication is configured, duplicate packets are added to the flow after the shaper, and the resulting output bandwidth will be higher than the one configured in the shaper.

Furthermore, the amount of memory allocated for the shaper buffer will be taken from the buffer used for generating latency, so when memory is allocated for shaper buffering, the guaranteed lossless latency listed in Table 3 will decrease accordingly.

Configuration of the shaper using the UI is illustrated Fig. 6.46.

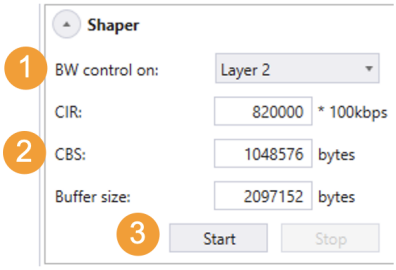


Fig. 6.46: Configuring flow shapers.

To configure egress shaper:

1. Select at which layer to implement the shaper from dropdown menu (e.g. Layer 2).
2. Supply the parameters to configure the shaper (e.g. CIR = 8.2 Gbps, CBS = 1 Mbyte, Buffer Size = 2 Mbyte)
3. Press *Start* to activate the impairment.

Note

The example below illustrates how to configure the shaper as illustrated Fig. 6.46.

```
PE_BANDSHAPER [fid] ON L2 820000 1048576 2097152
```

Latency / Jitter (iid = 2)

The latency / jitter impairment differs significantly from the other impairments described above, because it affects the delay of each packet. As a consequence, the distributions which can be assigned to the latency / jitter impairment define latencies rather than the distance in packets between two impaired packets. Figure 47 illustrates how to configure a flow for a Gaussian jitter distribution with an average delay of 50 us and a standard deviation of 2.5 us.

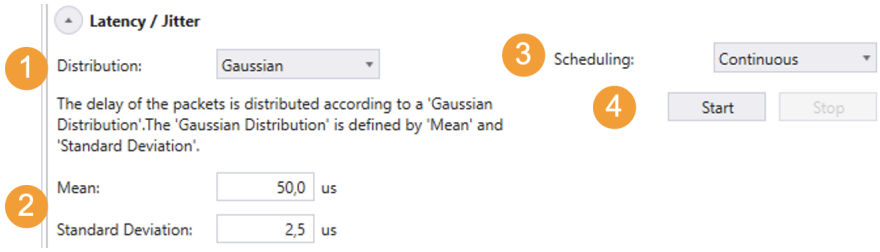


Fig. 6.47: Latency jitter configuration in the UI.

(There is no configuration of the impairment for latency / jitter).

To configure latency / jitter:

1. Select the relevant distribution from the distribution dropdown menu (e.g. Gaussian).
2. Supply the parameters to configure the selected distribution (e.g. Mean = 50.0 us, Standard Deviation = 2.5 us.).
3. Configure the scheduler (e.g. Continuous).
4. Press *Start* to activate the impairment.
5. To stop dropping packets, press *stop*.

The configuration shown Fig. 6.47 will cause a Gaussian jitter distribution to be applied to the packets on the flow.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_GAUSS [fid,2] 50000 2500
PED_ENABLE[fid,0] ON
```

6.7.3 Impairments Distributions

As stated above, Chimera supports a very flexible distribution scheme with a variety of distributions which can be applied depending on the impairment type.

Overall, Chimera supports two different ways of implementing the distribution:

- Inter-packet: The distribution determines the distance between the impaired packets in terms of packets. For example, the number of packets between two dropped packet could be 10. This kind of distribution applies to the following impairments:
  - Drop
  - Corruption
  - Duplication
  - Misordering
- Latency: The distribution determines the time the packet is delayed in Chimera. If it is not a constant latency, this type of distribution specifies a combination of fixed latency and jitter. E.g., the jitter distribution for the packets in flow could follow a Gaussian distribution.

This kind of distribution applies to the following impairments:

- Latency / jitter

The policers and shapers do not support assignment of a distribution function.

Furthermore, some distribution functions are implemented as logic functions, while others are implemented using table look up to approximate mathematical functions. When implementing an inter-packet distribution, the table will contain 512 entries, while the table will contain 1024 entries for latency distributions.

The distributions supported in Chimera for each impairment are illustrated in Fig. 6.48.

"✓" : supported
"✖" : not supported

Impairments / Distributions	Drop	Corruption – (FCS / IP/TCP/UDP)	Duplication	Misordering	Latency / Jitter	Scheduler
Off	✓	✓	✓	✓	✓	N.A.
Logic based						
Manual	✓	✓	✓	✓	✖	N.A.
Fixed burst	✓	✓	✓	✖	✖	One shot / Repeat
Accumulate & Burst	✖	✖	✖	✖	✓	
Fixed Rate	✓	✓	✓	✓	✖	Continuous / Repeat Pattern
BER	✓	✓	✓	✖	✖	
Random Rate	✓	✓	✓	✖	✖	
Gilbert-Elliot	✓	✓	✓	✖	✖	
Random burst	✓	✓	✓	✖	✖	
Constant	✖	✖	✖	✖	✓	Continuous
Table lookup (inter packet: 512 / latency: 1024 samples)						
Uniform	✓	✓	✓	✖	✓	Continuous / Repeat Pattern (latency not supported)
Gamma	✓	✓	✓	✖	✓	
Gaussian	✓	✓	✓	✖	✓	
Poisson	✓	✓	✓	✖	✓	
Step	✖	✖	✖	✖	✓	
Custom	✓	✓	✓	✖	✓	

Fig. 6.48: Impairment distributions supported in Chimera

The following sub-sections will describe each distribution in detail and provide script examples of how to configure.

Logic-Based Distributions

This sub-section contains a description of the logical distributions.

Off Distribution

This distribution is default at power-up and contains no configuration of any impairment parameters. If assigned to an impairment it will turn off the impairment and clear all impairment configuration.

Manual Injection

It is often convenient during testing to manually introduce a limited number of impairments. To do this, the user can configure the fixed burst described in section 11.1.3.

Fixed Burst

Fixed Burst, when triggered, will impair a number of consecutive packets specified by the *Burst Size*. An example of how to configure fixed burst for drop is illustrated Fig. 6.49.

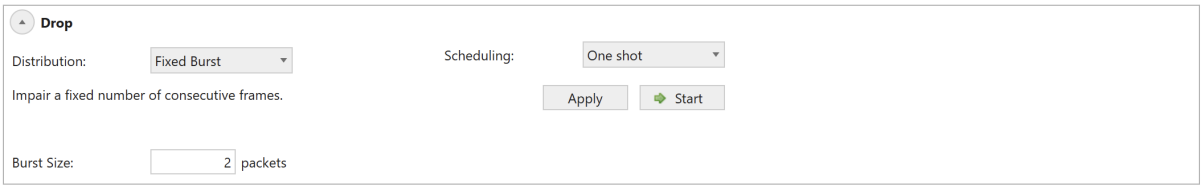


Fig. 6.49: Configuring *Fixed Burst* distribution in the UI.

Distribution parameters:

- Burst Size: Specifies the number of consecutive packets to impair

(For valid parameter ranges, please refer to the script command description.)

This configuration will cause 2 consecutive packets to be dropped when pressing *Start*.

Note

The example below illustrates how to configure the configuration above using script commands.

```
PED_SCHEDULE   [fid,0] 1 0
PED_FIXEDBURST [fid,0] 2
```

For fixed burst configured for one shot, there is a special command to determine whether there is a burst pending or not.

```
PED_ONESHOTSTATUS[fid,2] ?
```

This command will return the value of the register which is used to trigger a fixed burst. In case a 1 is returned, there is a burst pending. The next packet on the flow will trigger the burst.

Random Burst

Random Burst implements bursts of random size. The burst is triggered randomly based on a configurable per packet probability and subsequently impair a random number of consecutive packets chosen between minimum burst size (*Burst Min*) and maximum burst size (*Burst Max*).

An example of how to configure fixed burst for corruption is illustrated Fig. 6.50.

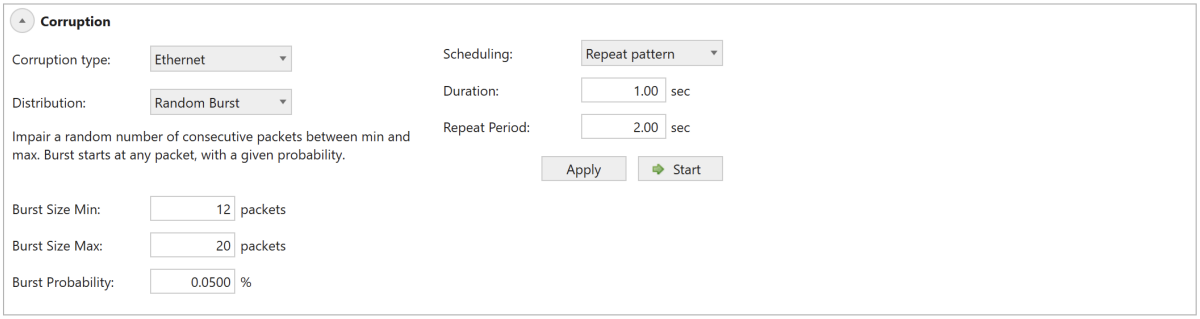


Fig. 6.50: Configuring *Random Burst* distribution in the UI.

Distribution parameters:

- Burst Size Min: Specifies the minimum burst size.
- Burst Size Max: Specifies the maximum burst size.
- Burst Probability: Specifies the probability that a burst will start at any given packet.

(For valid parameter ranges, please refer to the script command description.)

In the example above, every packet has a 0.05 % probability of starting a burst. The burst size will be randomly chosen between 15 and 20 packets. The impairment will be restarted every 2.0 sec and turned off after 1.0 sec.

Note

The example below illustrates how to configure the same using script commands.

```
PED_SCHEDULE    [fid,4] 100 200
PED_RANDOMBURST[fid,4] 15 20 500
```

Fixed rate

Fixed Rate will impair a configurable fraction of the packets in a predictable way, with nearly equal distance between impairments . An example of how to configure fixed rate for duplication is illustrated Fig. 6.51.

Distribution parameters:

- Impair rate: Fraction of packets to impair.

(For valid parameter ranges, please refer to the script command description.)

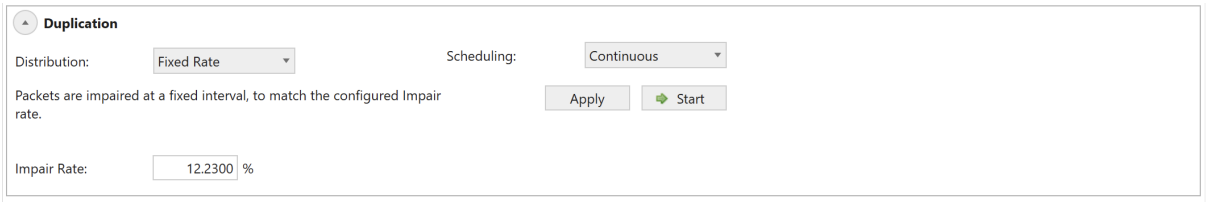


Fig. 6.51: Configuring *Fixed Rate* distribution in the UI.

In the example above, 12.23% of the packets will be duplicated. Duplication is done with (nearly) equal distance between duplicated packets in a predictable manner to match the configured impair rate.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,3] 1 0
PED_FIXED   [fid,3] 122300
```

Bit Error Rate (BER)

Bit Error Rate will impair the packets of a flow equivalent to a configured BER. E.g., if configured for a BER of 5*10-8, an impairment will be applied for every 0.2*108 bits on the flow. The impairments are applied in a predictable way, with nearly equal distance between impairments.

An example of how to configure bit error rate for drop is illustrated Fig. 6.52.

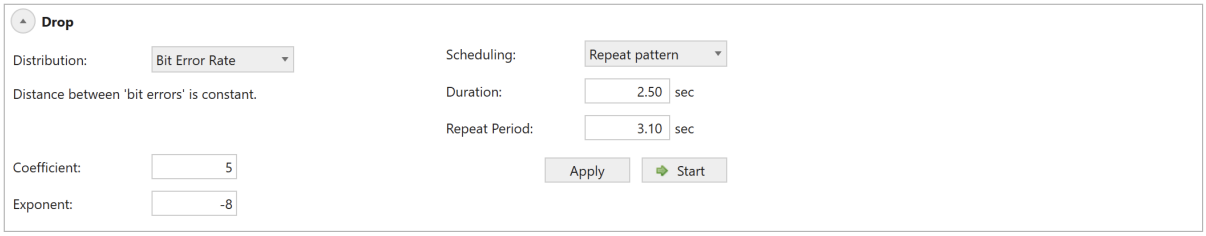


Fig. 6.52: Configuring *Bit Error Rate* distribution.

Distribution parameters:

- Coefficient: The mantissa of the configured BER.
- Exponent: The exponent of the configured BER.

`BER=Coefficient* 10^Exponent`

(For valid parameter ranges, please refer to the script command description.)

In the example above, one packet will be dropped for every 0.2*108 bits on the flow, equivalent to a BER of 5*10-8. The impairment will be restarted every 3.1 sec. and turned off after a duration of 2.5 sec.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,0] 250 310
PED_BER     [fid,0] 5 -8
```

Random Rate

Random Rate will impair a configurable fraction of the packets based on a per packet drop probability, i.e. unlike fixed rate, the impairment pattern is stochastic with an average equal to the configured *Impair Probability*.

An example of how to configure random rate for duplication is illustrated Fig. 6.53.

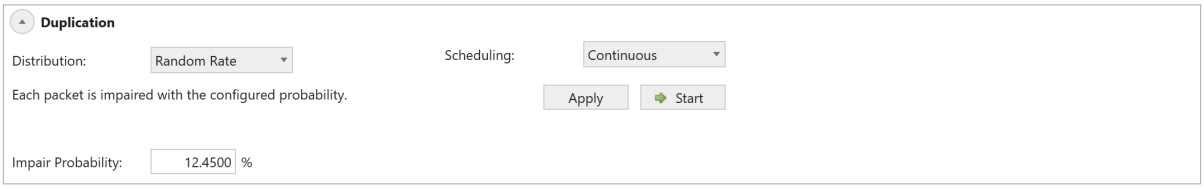


Fig. 6.53: Configuring *Random Rate* distribution in the UI.

Distribution parameters:

- Impair rate: Fraction of packets to impair.

(For valid parameter ranges, please refer to the script command description.)

In the example above, 12.45% of the packets will be duplicated. Duplication is done based on per packet probability.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,3] 1 0
PED_RANDOM  [fid,3] 124500
```

Gilbert-Elliot

The Gilbert-Elliot distribution defines two states, each with a separate packet impairment probability:

- Good state
- Bad state

In any of the two states, there is a certain probability that the system will transition to the other state. The Gilbert-Elliot algorithm is illustrated Fig. 6.54.

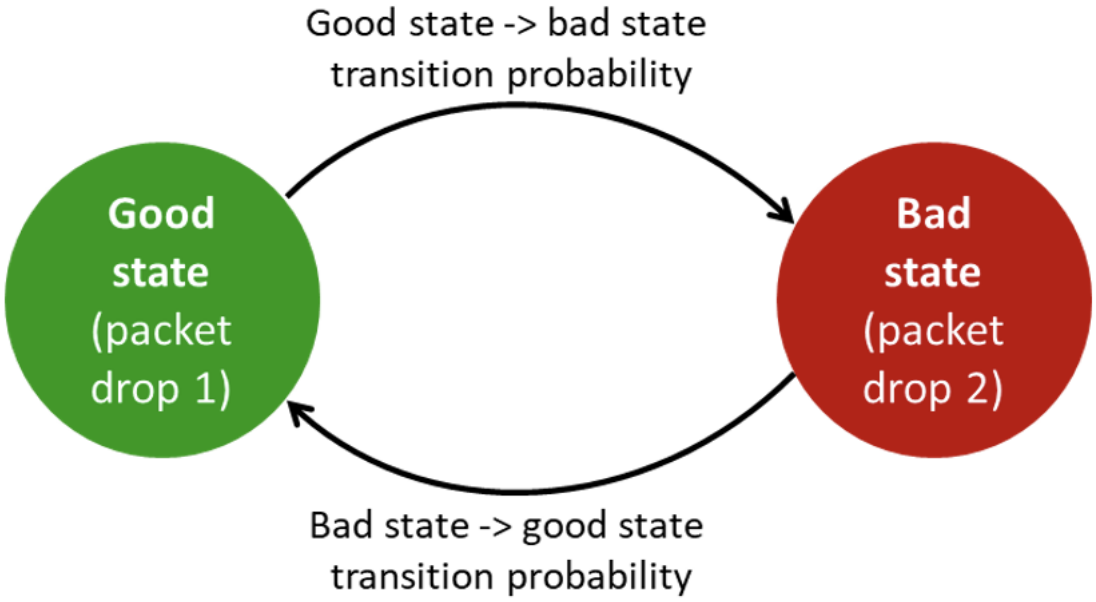


Fig. 6.54: Gilbert-Elliot two states.

When the system is in the *Good State* (*State 1*) ,there is a configurable impairment probability (*Drop 1*) and there is a configurable probability (*Transfer prob 1*) to transition to the *Bad State* (*State 2*). Likewise, when in the *Bad State*, there is a configurable impairment probability (*Drop 2*) and a configurable probability to transition to the *Good State* (*Transfer prob 2*).

An example of how to configure *Gilbert-Elliot* for TCP corruption is illustrated Fig. 6.55.

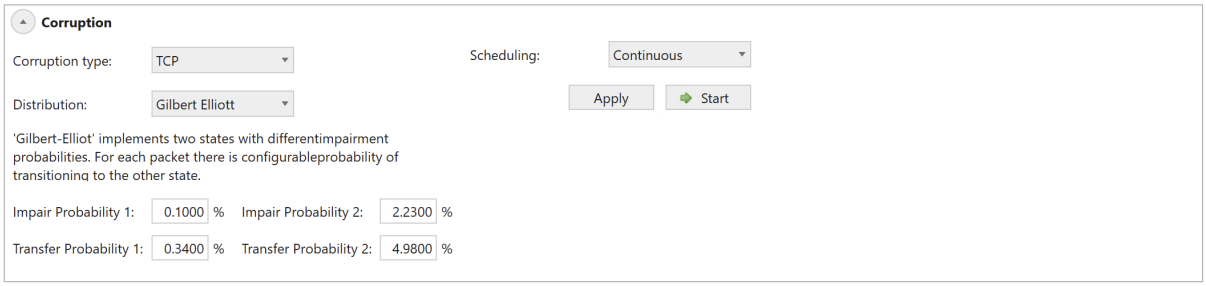


Fig. 6.55: Configuring Gilbert-Elliot distribution in the UI.

Distribution parameters:

- Impair Probability 1: The per packet impairment probability in state 1.
- Transfer Probability 1: The per packet probability of moving to state 2 from state 1.
- Impair Probability 2: The per packet impairment probability in state 2.
- Transfer Probability 2: The per packet probability of moving to state 1 from state 2.

(For valid parameter ranges, please refer to the script command description.)

In the example above, the probability of TCP corruption when in state 1 is 0.01 %, while the probability of transferring to state 2 is 0.34 %. When in state 2, the probability of TCP corruption when is 0.01 %, while the probability of transferring to state 1 is 0.34 %

Note

The example below illustrates how to configure the same configuration using script commands.

```
PE_CORRUPT  [fid]   TCP
PED_GE      [fid,4] 100 3400 22300 49800
PED_SCHEDULE[fid,4] 1 0
```

Accumulate & Burst

Chimera allows simulating temporary congestion in a network using the *accumulate and burst* distribution. For a configurable period (*Burst Delay*), packets are collected in a buffer, rather than forwarded to the output port. After this period of time, all the buffered packets are forwarded to the output as fast as possible, thus creating a burst. Once buffered packets have been transmitted from the buffer, packets will be forwarded with minimum latency.

The packet accumulation is triggered by the first packet received on the flow after the distribution was enabled. An example of how to configure *Accumulate & Burst* for latency / jitter corruption is illustrated Fig. 6.56 55.

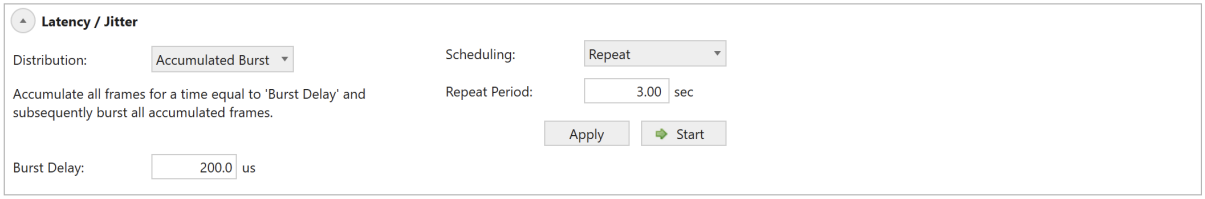


Fig. 6.56: Configuring “Accumulate and Burst” in the UI.

Distribution parameters:

- Burst Delay: Specifies the duration of the packet accumulation after receiving the first packet.

(For valid parameter ranges, please refer to the script command description.)

In the example above, packets are accumulated for 200 µs, and subsequently they are sent then to the output as fast as possible. The accumulation is re-triggered every 3.0 sec.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,2] 1 300
PED_ACCBURST[fid,2] 200000
```

For accumulate & burst configured for repeat, there is a special command to determine whether an accumulate & burst event is pending, or whether it has been triggered.

```
PED_ONESHOTSTATUS[fid,2] ?
```

This command will return the value of the register which is used to trigger an accumulate & burst event. In case a 1 is returned, there is an event pending. The next packet on the flow will trigger the accumulate & burst.

Constant Delay

Constant Latency will apply a constant latency to all packets in the flow.

Configuring constant latency in the UI is illustrated Fig. 6.57.

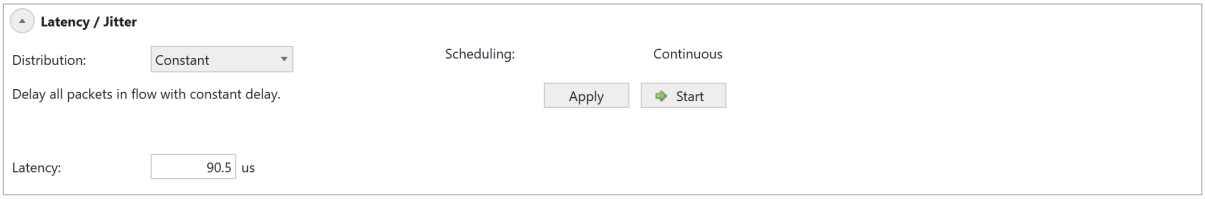


Fig. 6.57: Configuring Constant Latency in the UI.

Distribution parameters:

- Latency: Specifies the constant latency to be applied to all packets.

(For valid parameter ranges, please refer to the script command description.)

In the example above, all packets are delayed for 90.5 µs.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_CONST[fid,2] 90500
```

Table-Based Distributions

This sub-section describes the distributions which are implemented using a table lookup to approximate a mathematical function. Each table-based distribution exists in 2 flavors:

- Inter-packet distribution: The distribution describes the distance in packets between impairments. This is implemented with 512 table values.
- Latency distribution: This distribution describes the delay applied to the packets. This is implemented with 1024 table values.

This implies that when a table-based distribution is applied to the latency / jitter impairment, the table will contain 1024 values, while for all other impairments, it will contain 512 values.

The maximum data values that can be programmed into the table based distributions are listed in Fig. 6.58.

Entry type	Maximum value	
Inter-packet	262,143 packets	
Latency / jitter	30 ms	(Normal timing mode)
	302 ms	(Extended timing mode)

Fig. 6.58: Custom distribution maximum data values.

Note

Note: When applying a table-based distribution to latency / jitter, Chimera can only adjust the jitter within the existing IPG of the incoming packets. This implies that packet misordering cannot happen due to jitter. If sending packets with a smaller IPG than the latency specified in a distribution, the distribution function will not be applied as intended.

Uniform

The Uniform Distribution will randomly select the distance between impairments from a configured interval defined by a minimum (*min*) and a maximum value (*max*).

Fig. 6.59 illustrates how to configure a uniform distribution for drop.

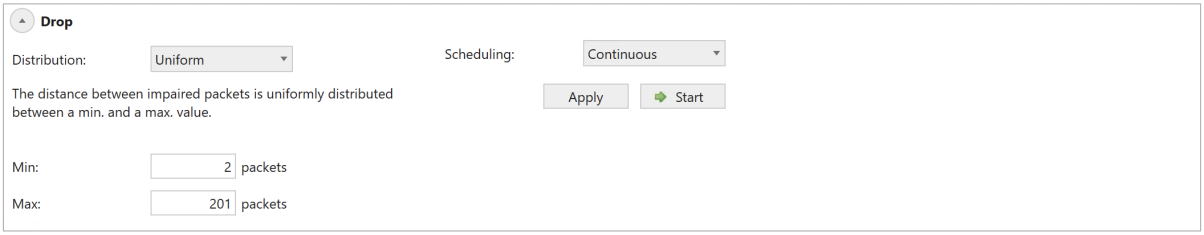


Fig. 6.59: Configuring Uniform distribution.

Distribution parameters:

- Min: Specifies the minimum number of packets/latency for the uniform distribution.
- Max: Specifies the maximum number of packets/latency for the uniform distribution.

(For valid parameter ranges, please refer to the script command description.)

In the example above, the distance between drops will be chosen randomly in the interval between 15 packets and 201 packets.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,0] 1 0
PED_UNI     [fid,0] 15 201
```

Gaussian

The Gaussian (Normal) distribution implements an approximation of the mathematical function, which is defined by a mean value `mean` and a standard deviation `sd`. The Gaussian distribution is illustrated Fig. 6.60. When a flow is configured for Gaussian Jitter, the mean latency of packets is equal to the configured mean latency, and the deviations of single packets from the mean will be according to the Gaussian distribution.

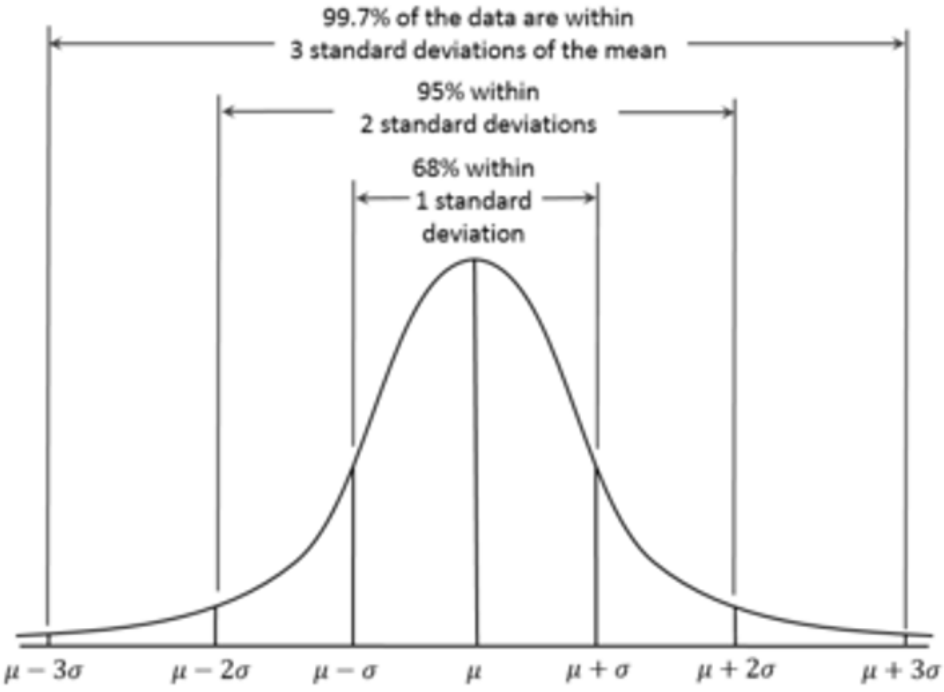


Fig. 6.60: Gaussian distribution.

Chimera limits the Gaussian function to the following latency interval:

`mean - 3 x sd <= simulated values <= mean + 3 x sd`

Figure 59 illustrates how to configure Gaussian distribution for latency / jitter.

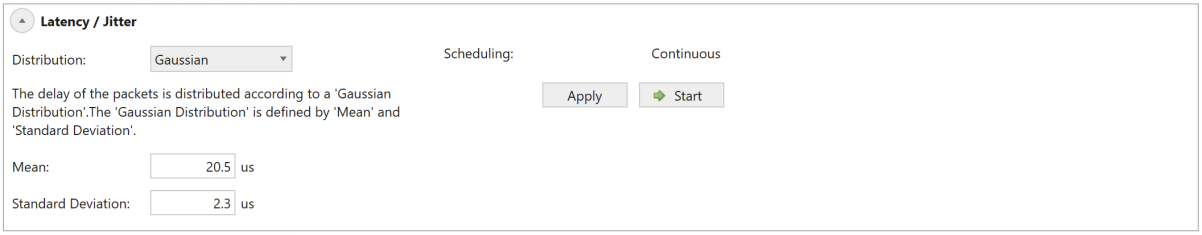


Fig. 6.61: Configuring Gaussian jitter

Distribution parameters:

- Mean: Specifies the mean value for the Gaussian distribution.
- Standard Deviation: Specifies the standard deviation for the Gaussian distribution.

(For valid parameter ranges, please refer to the script command description.)

In the example above, a jitter with a Gaussian distribution with Mean = 20.5 µs and Standard Deviation = 2.3 µs is applied continuously to the flow.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_GAUSS[fid,2] 20500 2300
```

Gamma

The Gamma distribution approximates the mathematical function which is defined by a Shape parameter (κ) and the Scale parameter (θ). The Gamma distribution is illustrated Fig. 6.62 for different values of the Shape and Scale parameters. When a flow is configured for Gamma Latency / Jitter, the mean latency of packets is equal to the configured mean latency (see below), and the deviations of single packets from the mean will be according to the Gamma distribution.

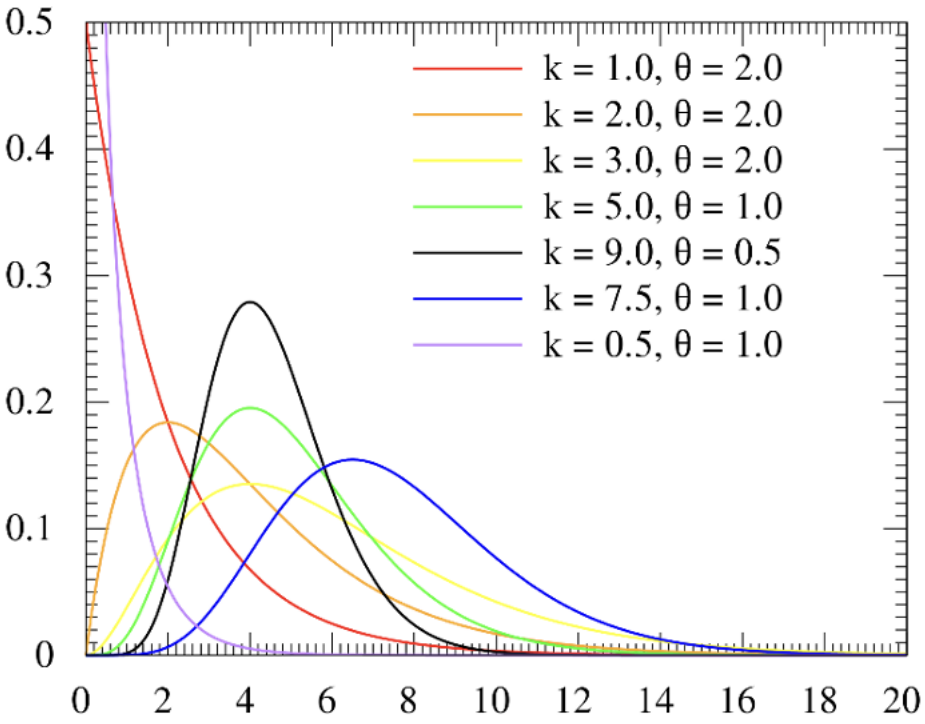


Fig. 6.62: Gamma distribution.

Chimera limits the Gamma function to the following latency interval:

Fig. 6.63 illustrates how to configure Gamma distribution for Latency / Jitter.

Distribution parameters:

- Shape (κ): Gamma distribution Shape parameter.
- Scale (θ): Gamma distribution Scale parameter

$$\mu - 4 \times \sigma <= \text{simulated values} <= \mu + 4 \times \sigma$$

Where:

$\sigma = \sqrt{\kappa * \theta^2}$ (standard deviation)

$\mu = \kappa * \theta$ (mean value)

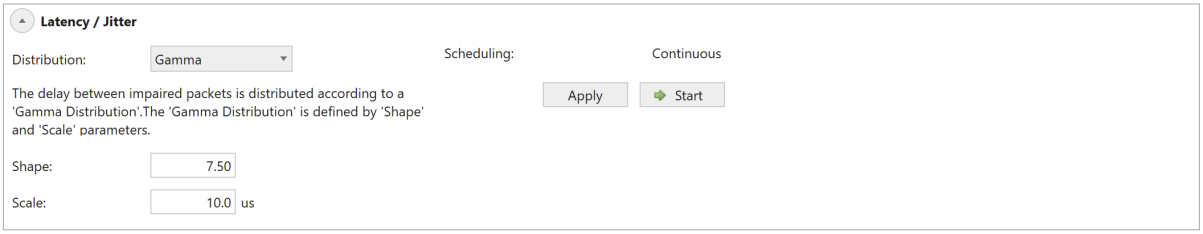


Fig. 6.63: Configuring Gamma distribution in the UI.

(For valid parameter ranges, please refer to the script command description.)

In the example above, Latency / Jitter is configured with a Shape parameter of 7.5 and a Scale parameter of 10.0 µs. For Latency / Jitter, it is not possible to configure a scheduler function.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_GAMMA [fid,2] 750 10000
```

Poisson

The Poisson distribution approximates the mathematical function which is defined by a mean value (λ). The Poisson distribution is illustrated Fig. 6.64. When a flow is configured for poisson jitter, the mean latency of packets is equal to the configured mean latency, and the deviations of single packets from the mean will be according to the Poisson distribution.

Chimera limits the Poisson function to the following latency interval:

Fig. 6.65 illustrates how to configure Poisson distribution for duplication.

Distribution parameters:

- Lambda (λ): Specifies the mean value for the Poisson distribution.

(For valid parameter ranges, please refer to the script command description.)

In the example above, duplication with a packet spacing defined by a Poisson distribution with Mean = 10 is applied to the flow for 2.0 sec and then stopped. This will be repeated every 4.0

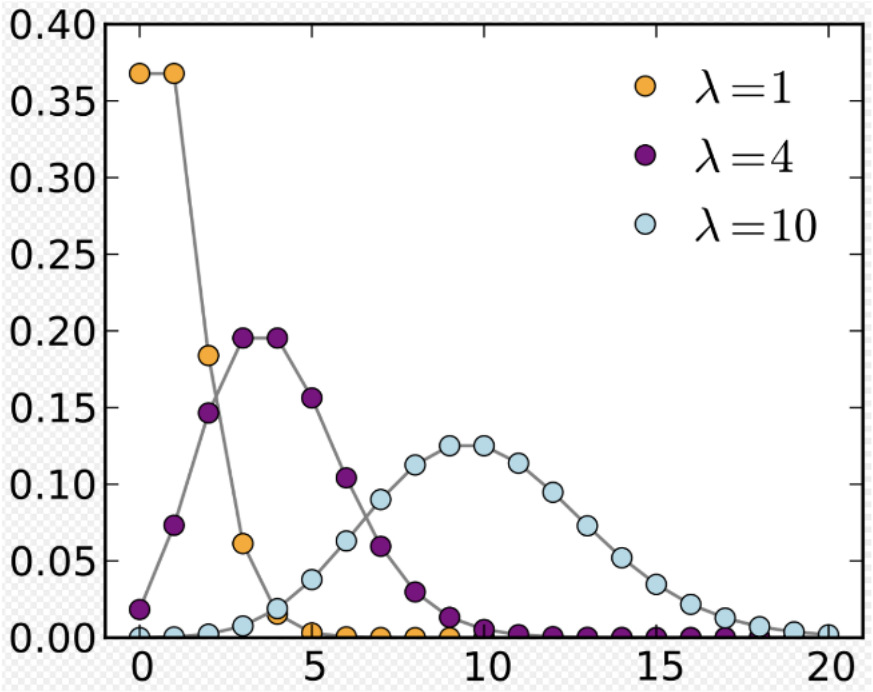


Fig. 6.64: Poisson distribution.

μ - 3 x σ <= simulated values <= μ + 3 x σ

Where:

$\sigma = \sqrt{\lambda}$ (standard deviation)

μ = λ (Mean value).

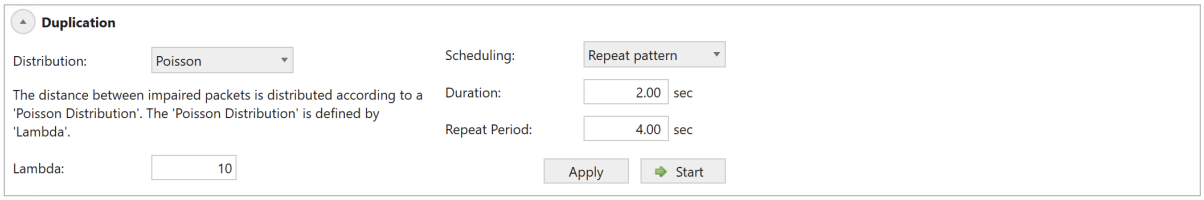


Fig. 6.65: Configuring Poisson distribution in the UI.

sec.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,3] 2000 4000
PED_POISSON [fid,3] 10
```

Step

The Step Distribution will apply an impairment to a flow, randomly altering between two configurable values. The step distribution is only applicable to latency / jitter.

Fig. 6.66 illustrates how to configure step distribution for Latency / Jitter.

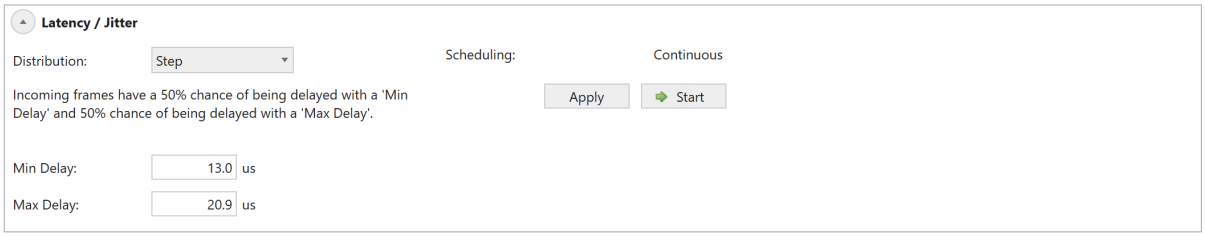


Fig. 6.66: Configuring Step Distribution for latency / jitter in the UI.

Distribution parameters:

- Min Delay: Specifies the minimum delay.
- Max Delay: Specifies the maximum delay.

(For valid parameter ranges, please refer to the script command description.)

In the example above, packets will randomly be delayed by either 7.5 µs or 20.9 µs.

Note

The example below illustrates how to configure the same configuration using script commands.

```
PED_STEP[fid,2] 7500 20900
```

Custom Distribution

In addition to the pre-defined distributions described above, Chimera supports the definition of *Custom Distributions*. Custom distributions are table-based distributions which are defined per port. They are identified by a Custom ID (cust_id), which identifies each custom distribution on that port. Chimera supports up to 40 custom distributions per port (cust_id: 1-40). Once the

custom distribution is defined, it can be applied to any of the impairments in the impairment pipeline.

A custom distribution is a table-based distribution, where the user can supply the values in the table. Furthermore, the user can configure whether the values in the table should be applied in a predictable order, reading out table index 0, 1, 2, … 511/1023 to 0, 1, 2, …, or whether the values are applied in a random order.

Finally, the user can supply a *Custom Name* for every custom distribution to make it easier to navigate within the distributions defined.

The custom distributions will support 512 table entries for inter-packet distributions and 1024 values for latency / jitter distributions. As a result, only custom distributions with 1024 entries may be assigned to latency / jitter, while custom distributions with 512 entries can be assigned to all other impairments except for misordering, which does not support custom distributions.

Custom distributions are defined using the script command: `PEC_VAL`.

The `PEC_VAL` has the following parameters:

- Linear: Determines whether table values are chosen randomly or in predictable order 0 -> highest row -> 0 etc.
- Symmetric: Reserved for future use - must be set to OFF (0).
- Num_entries: This indicates whether the distribution is an *inter-packet* distribution (=512 entries) or a latency / jitter distribution (=1024 entries).
- 512 or 1024 data values according to `num_entries`.

The UI supports managing the custom distributions using the *Custom Distributions Library* found on the port impairment tab. The custom distributions library provides an overview of the custom distributions defined for each port and can be used to create, delete and export custom distributions to a file for later use.

The custom distributions library is illustrated Fig. 6.67.

The custom distribution library lists which custom distributions are on the selected port and hence which distributions will appear in the custom distribution list, when assigning a custom distribution to an impairment.

The *Type* column indicates whether the distribution can be assigned to latency/jitter or one of the other impairments (inter-packet).

Files used to define custom distributions have the `*.xpc` extension and must contain the following commands:

```
PEC_VAL[cust_id]
PEC_COMMENT[cust_id]
```

It is optional whether the `*.xpc` file contains the `[MODULE]/[PORT]` in front of the commands. If these values are present, the custom distribution will be defined for the port, indicated by the `[PORT]` parameter. If `[MODULE] / [PORT]` is not defined, the custom distribution will be defined for the port currently selected in the UI.

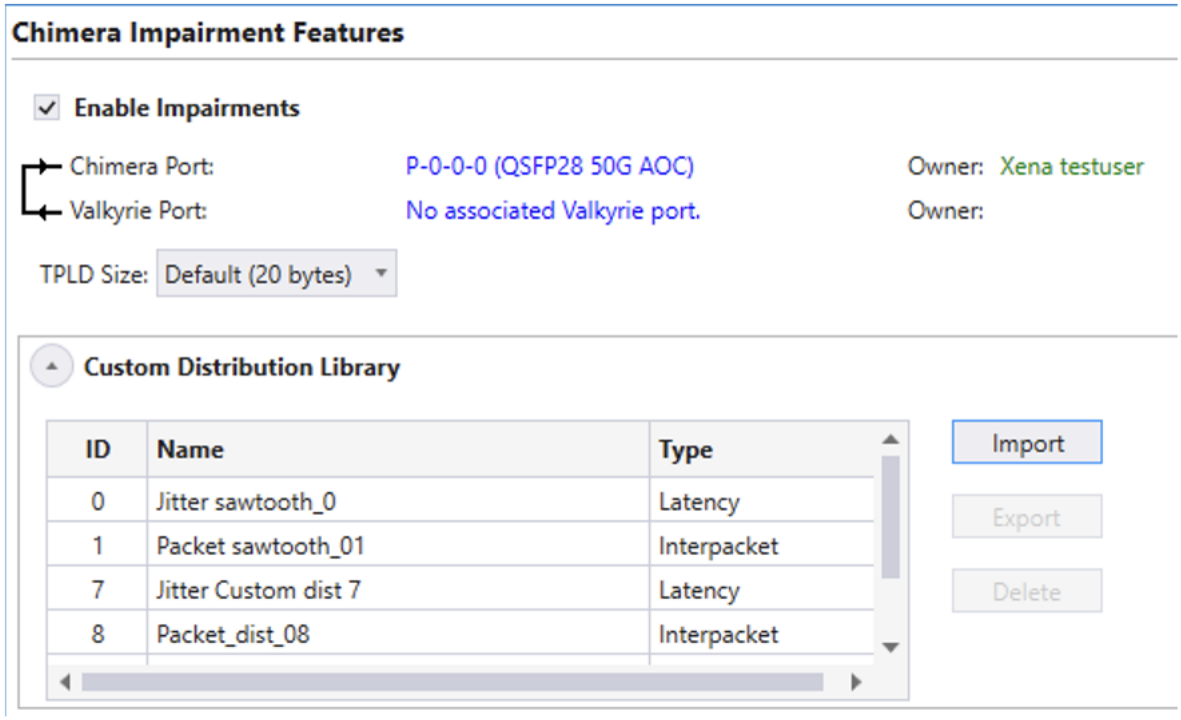


Fig. 6.67: Custom Distributions Library.

Custom distribution configuration example (inter-packet):

The example below illustrates how to configure a custom distribution for *inter-packet* with a `cust_id = 5` and with linear property = ON. Furthermore, it will assign a name to the distribution: *Sample inter-packet distribution*.

```
PEC_VAL[5] ON OFF 512 <DATA_0> <DATA_1> ... <DATA_511>
PEC_COMMENT[5] "Sample inter-packet distribution"
```

Once the inter-packet Custom Distribution has been defined, it is possible to assign it to an impairment. Fig. 6.68 illustrates how to use the distribution created above with drop.

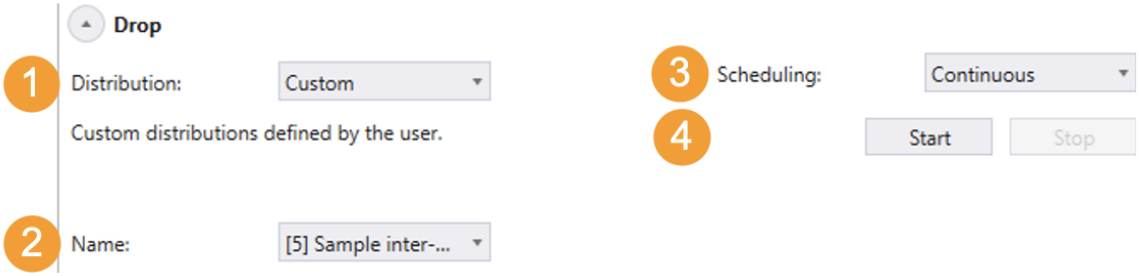


Fig. 6.68: Assigning Custom Distribution to drop.

To configure custom distribution for drop:

- Select *Custom Distribution* from the distribution dropdown menu.
- Select the required custom distribution from the custom distribution list (e.g. #5).

- Configure the scheduler (e.g. Continuous)
- Press *Start* to activate the Custom Distribution.

The example below illustrates how to configure the same configuration using script commands.

```
PED_SCHEDULE[fid,0] 1 0
PED_CUST    [fid,0] 5
```

Custom distribution configuration example (latency / jitter):

The example below illustrates how to configure a custom distribution for latency / jitter with a cust_id = 13 and with linear property = OFF. Furthermore, it will assign a name to the distribution: *Sample latency distribution*.

```
PEC_VAL[13] OFF OFF 1024 <DATA_0> <DATA_1> ... <DATA_1023>
PEC_COMMENT[13] "Sample latency distribution"
```

Once the latency / jitter custom distribution has been defined, it is possible to assign it to an impairment. Fig. 6.69 illustrates how to use the distribution created above with latency / jitter.

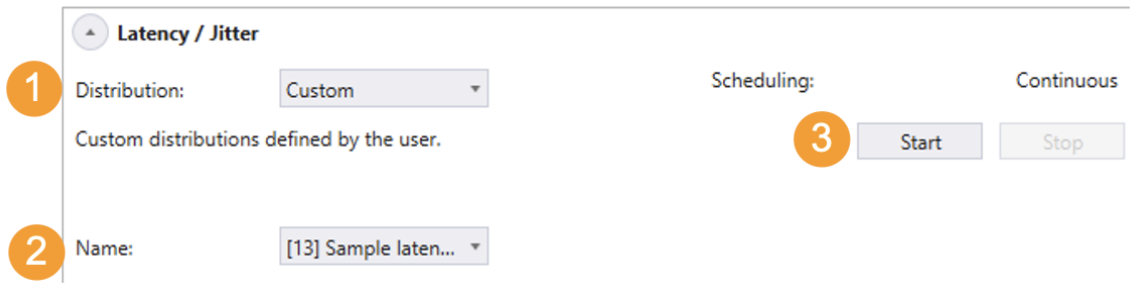


Fig. 6.69: Assigning Custom Distribution to latency / jitter in the UI.

To configure custom distribution for latency / jitter:

- Select *Custom Distribution* from the distribution dropdown menu.
- Select the required custom distribution from the custom distribution list (e.g. #13).
- Press *Start* to activate the custom distribution.

(Latency / jitter does not support a scheduler for custom distributions).

The example below illustrates how to configure the same configuration using script commands.

```
PED_CUST[fid,2] 13
```

6.7.4 Scheduler Functions

As described in Section *Overview*, Chimera supports automatically turning the impairments on and off with the configured distribution using a per impairment scheduler.

The scheduler is configured depending on the distribution type which is applied to the impairment. There are 2 types of distributions:

- Continuous distributions
- Burst distributions

Table 6.17 illustrates Bursty and Continuous distributions.

Table 6.17: Distributions overview

Continuous	Bursty
Random Burst	Fixed burst
Fixed Rate	Accumulate & Burst
Random Rate	
Bit Error Rate	
Gilbert-Elliot	
Uniform	
Gamma	
Gaussian	
Poisson	
Step	
Custom distribution	

Notice that for latency / jitter, only the Accumulate & Burst supports a scheduler. If other distributions are applied to latency / jitter, only continuous mode is supported.

Continuous Distributions

For continuous distributions, the scheduler can work in 2 modes:

- Continuous: When started, the impairment is applied continuously with the configured distribution until it is manually stopped.

The example below illustrates how to configure the scheduler in continuous mode for drop with fixed rate distribution of 1.23%.

```
PED_SCHEDULE[fid,0] 1 0
PED_FIXED   [fid,0] 12300
```

- Repeat Pattern: When started, the impairment is applied with the configured distribution in a repeated pattern. First it will be applied for a configurable duration and then turned off. It will be restarted for every repeat period.

This is illustrated in Fig. 6.70.

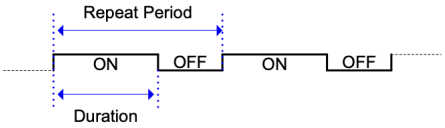


Fig. 6.70: Scheduler function Repeat Pattern

The example below illustrates how to configure the scheduler for duration = 1.20 sec and a repeat period = 5.2 sec, applying drop with random rate distribution of 3.3421 %.

```
PED_SCHEDULE[fid,0] 1200 5200
PED_RANDOM  [fid,0] 33421
```

Bursty Distributions

The bursty distributions are characterized by being bursty by nature, i.e. they will automatically terminate if not restarted. E.g., a fixed burst of 8 packets will automatically stop after dropping 8 packets.

For bursty distributions, the scheduler can work in 2 modes.

- One-Shot: When started, the impairment will be applied for the duration of the burst. When the burst terminates, the impairment is turned off.

The example below illustrates how to configure the scheduler for in shot mode for drop with a fixed burst distribution of 10 packets.

```
PED_SCHEDULE  [fid,0] 1 0
PED_FIXEDBURST[fid,0] 10
```

- Repeat Burst: When started, the impairment will be applied for the duration of the burst. The burst will be restarted every Repeat period.

This is illustrated in Fig. 6.71.

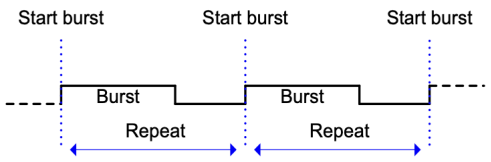


Fig. 6.71: Scheduler function Repeat Burst.

The example below illustrates how to configure the scheduler to restart the Accumulate & Burst every 2.36 sec with a Burst Delay of 0.654 sec.

```
PED_SCHEDULE[fid,2] 1 2360
PED_ACCBURST[fid,2] 654000
```

6.8 Logging and Reporting

Chimera module and its ports for network impairment measurement also offer support for logging and reporting functionalities.

Please refer to *Logging and Reporting*.

6.9 Statistics Charting

Chimera module and its ports for network impairment measurement also offer support for statistics charting functionalities.

Please refer to *Statistics Charting*.

CHAPTER

SEVEN

FAQ

7.1 Broken Hardware

7.1.1 Q: How to tell whether there is a problem with your hardware or if your module is broken?

A: Each port has 2 LEDs: a yellow one for running streams and a green one for linking up. Before using the software, when you connect the cassis the two leds will flash for one or two seconds.

To check if the module is broken simply reset the chassis. If none of the module's led flash it means it is broken and you should do an "RMA". If the leds are working you should contact support to find out what is the problem.

7.2 Manually Calibrate The Negative Delay Between Test Modules

1. Directly connect the two ports.(without any DUT)
2. Add one stream in each port.
3. Select the port which you want.
4. Select the *Port Statistics* menu.
5. Start all the traffic.
6. Click the *calibrate latency* button.
7. Do the same thing in another port.

7.3 Traffic Capture

This FAQ page deals with questions related to the traffic capture.

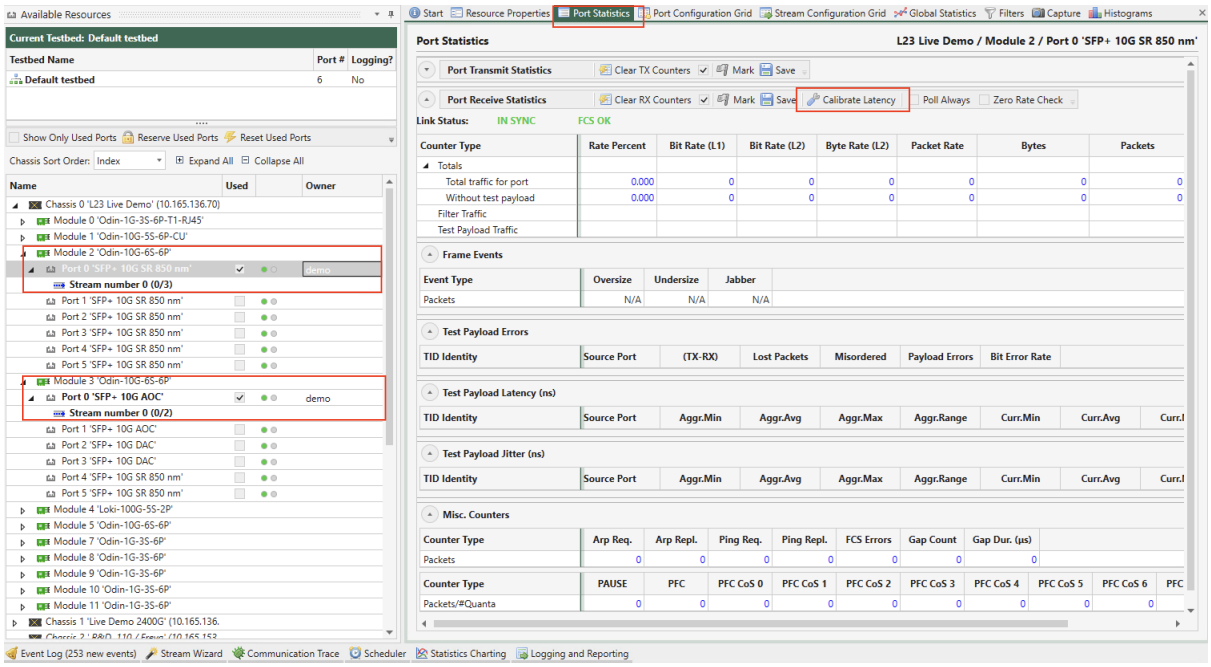


Fig. 7.1: Manually Calibrate The Negative Delay Between Test Modules

7.3.1 Q: What is traffic capture?

A: Capture allows selected packets of the incoming traffic on a port to be retained and inspected.

Triggers specify when to start capturing, when to stop, and which packets to keep while capturing is active. These events can be specified as user-defined filters or as built-in events such as packet checksum errors. Results can be viewed directly within the Xena environment, and packets can be saved for further analysis by external tools such as WireShark. There is also a packet-by-packet graphical display of key capture parameters, such as packet length and inter-frame gap.

7.3.2 Q: Is Xena compatible with WireShark?

A: Yes, the captured packets can be saved in either WireShark ".pcap" or ".pcapng" format, for further analysis in the WireShark environment. This is useful when analyzing packet protocols which are not (yet) supported in XenaManager.

7.3.3 Q: Are the 4-byte Ethernet FCS checksum field values included in pcap files?

A: The default setting is not to save FCS but this can be changed in the Capture panel.

7.3.4 Q: How large is the packet Capture buffer?

A: The wire-speed capture buffer is 64KB for the 10 Gbps test port, and 16KB per 10/100/1000M test port. The number of captured bytes per packets can be limited, e.g. to 64 bytes, so that only the packet header is captured. Then approximately 1000 or 250 packets can be captured per 10 Gbps and 10/100/1000M test port respectively. If the captured criteria

is specified to match an incoming lower speed stream with a 2-5 Mbps rate (such as a VoIP or IPTV stream), the capture buffer can store 4096 packets at any packet length.

7.3.5 Q: Which packet protocols can the capture engine decode?

A: XenaManager can decode several protocols, such as Ethernet, Ethernet II, VLAN, ARP, IPv4, IPv6, UDP, TCP, LLC, SNAP, GTP, ICMP, RTP, RTCP, and STP. Please refer to the XenaManager documentation for full details.

7.3.6 Q: How do I start/stop the traffic capture function?

A: The capture start/stop trigger events can be the arrival of a packet which matches a specific user-defined packet header filter (such as a specific Ethernet/IP address, VLAN ID, IP TOS etc) or a packet FCS checksum error.

7.3.7 Q: How can I define my packet capture criteria?

A: The capture function can be configured to retain packets which match a user-defined packet header filter (such as a specific Ethernet/IP address, VLAN ID, IP TOS etc), all traffic, or all traffic which does not include a test payload.

7.4 Chassis and Software

This FAQ page deals with questions related to test chassis and software.

7.4.1 Q: Which OS does XenaManager run on?

A: The XenaManager software requires at least Windows 10 or later.

7.4.2 Q: Are Xena testers field upgradeable to future product releases?

A: Yes, the Xena testers are fully upgradeable to future product releases. Both the embedded software and the integrated test modules can be upgraded from the XenaManager. If the upgrade includes a test module upgrade, the upgrade process requires a power cycle of the test unit.

7.4.3 Q: How do I change the chassis IP address?

A: The chassis management port is configured with a static IPv4 address (factory preset to 192.168.1.200). The management port's IP address can be changed from the XenaManager. For detailed instructions, refer to Xena Quick Start guide .

7.4.4 Q: I have forgotten the chassis IP address. What do I do?

A: If you forget the IP address of the chassis you need another way to get in touch with the chassis. This is done by making a point-to-point connection from your PC to the `Ext` Ethernet port that is positioned to the right of the `Mgmt` port on the chassis front panel.

The Ext port is pre-configured with the following IP setup:

Address = 172.16.255.200 Subnet = 255.255.255.0 Gateway = none

You must configure your PC port statically to an IP address in the 172.16.255.x range, and then you will be able to PING the chassis again. Now start the Manager, and connect to the chassis using address 172.16.255.200.

Under **CHASSIS PROPERTIES** you can now see which IP address is configured for the Mgmt port, and you can reserve the chassis and change it if necessary.

Note that the IP configuration of the Ext port cannot be changed, and that you should not configure the Mgmt port to use this subnet.

7.4.5 Q: I have forgotten the password to my tester, how do I log on?

A: In the first two minutes after the tester has booted (when the test-port LEDs start flashing) you can use the chassis serial number as the password. You can see the serial number on the label on the back of the chassis. Once logged on, set the password.

7.4.6 Q: Why do the test port LEDs start blinking after power-on and reboot?

A: Following reboot or power-on, the test port LEDs will start blinking after 2-3 minutes. This indicates that the tester is ready and now accepts XenaManager or scripting TCP/IP connections. When the first connection is made the LEDs revert to indicating link-sync status for each test port.

7.4.7 Q: Is the test module inside a XenaCompact identical to the XenaBay test modules?

A: Yes, a XenaCompact tester is functionally identical to a XenaBay tester - just with fewer test modules.

7.4.8 Q: Are transceiver plug-in modules hot-swappable?

A: The tester should be powered off when changing transceiver plug-in modules. Inserting or removing them with power on will not cause damage, but the tester will not detect any changes until it is re-booted or power-cycled.

7.4.9 Q: Can I daisy-chain multiple XenaCompact testers into a single logical test unit?

A: No, XenaCompact testers must be connected to individually by the XenaManager and script environments.

7.4.10 Q: Which TCP ports are used by the Xena testers?

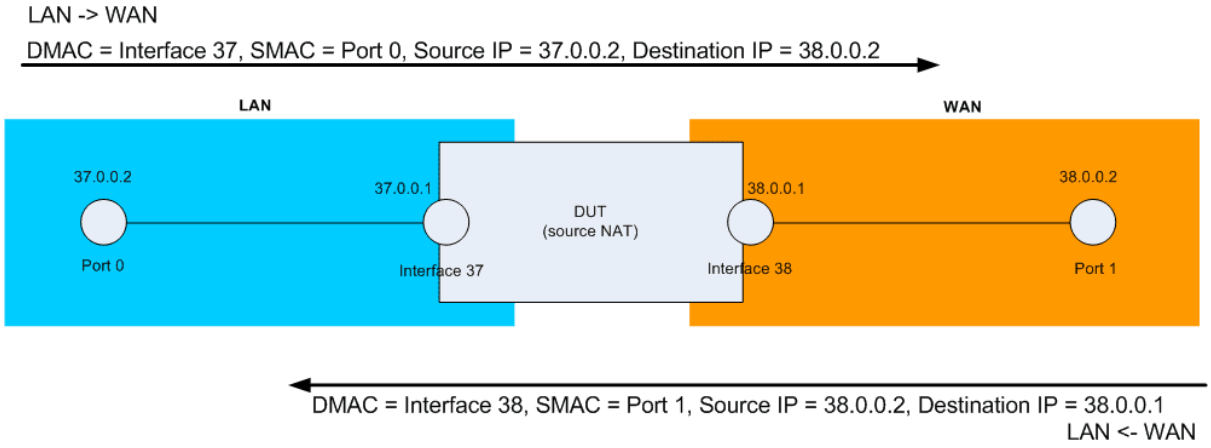
A: The test-based scripting API uses TCP port 22611. Xena tools such as XenaManager and Xena2544 uses TCP port 22606.

7.5 Test Source NAT Using XenaManager

Source NAT is typically used by internal users to access the internet; the source address is translated and thereby kept private.

7.5.1 Testbed Topology

We are using the testbed topology shown below to demonstrate how you can test a source NAT functions with bidirectional traffic, i.e. LAN-to-WAN and WAN-to-LAN.



The DUT has two interfaces, interface 37 and interface 38. The IP address of interface 37 is `37.0.0.1`, and the IP address of interface 38 is `38.0.0.1`.

Interface 37 is on the LAN side (incoming), and interface 38 is on the WAN side (outgoing). The source NAT policy is configured to replace the source IP address of the incoming traffic (LAN-to-WAN direction) with the outgoing interface address. This is exactly what a source NAT functions do.

We are going to show you how to create two streams, one for each direction, using XenaManager.

7.5.2 Stream Configuration Using XenaManager For Source NAT Testing

1. Reserve port 0 and port 1 on XenaManager.
2. Configure IP address for port 0 to be `37.0.0.2` *Resource Properties → Main Port Config → IPv4/IPv6 Properties*. Make sure the gateway is set to `37.0.0.1`, and check the options Reply to ARP Request and Reply to PINGv4 Request so the DUT is able to detect the port.
3. Configure IP address for port 1 to be `38.0.0.2` *Resource Properties → Main Port Config → IPv4/IPv6 Properties*. Make sure the gateway is set to `38.0.0.1`, and check the options *Reply to ARP Request* and *Reply to PINGv4 Request* so the DUT is able to detect the port.

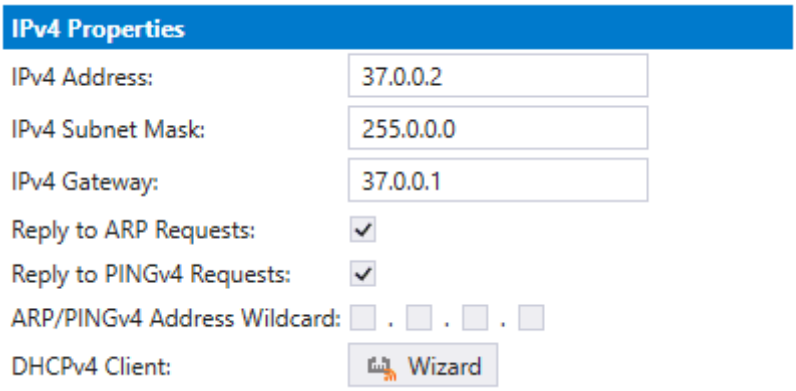


Fig. 7.2: Step 2

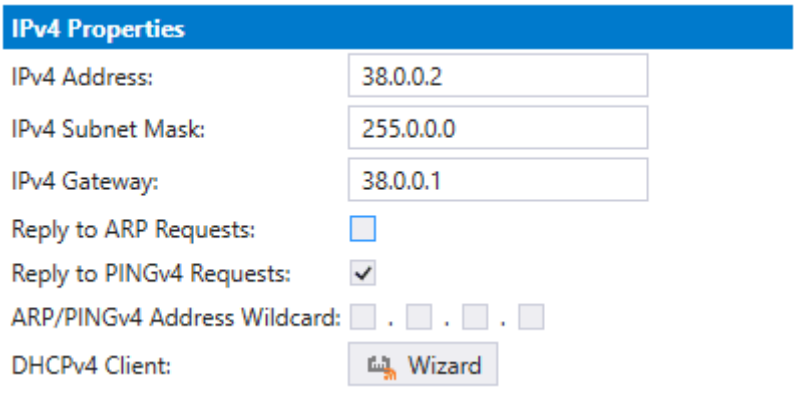


Fig. 7.3: Step 3

4. Create one IP stream on each port.

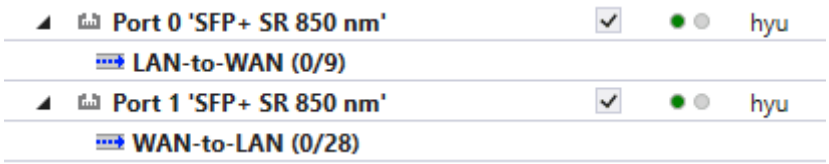


Fig. 7.4: Step 4

5. For stream LAN-to-WAN, source IP = `37.0.0.2`, destination IP = `38.0.0.2`, SMAC = port 0's MAC, DMAC = MAC address of DUT's interface 37. If you do not know the MAC address of DUT's interface, you can enter the IP address of the interface in *Connectivity Check* → *IPv4 Gateway Address* and click *Send ARP*. The test port will automatically resolve the DUT's interface MAC address using ARP and put the address value in the stream's DMAC field, as shown below.

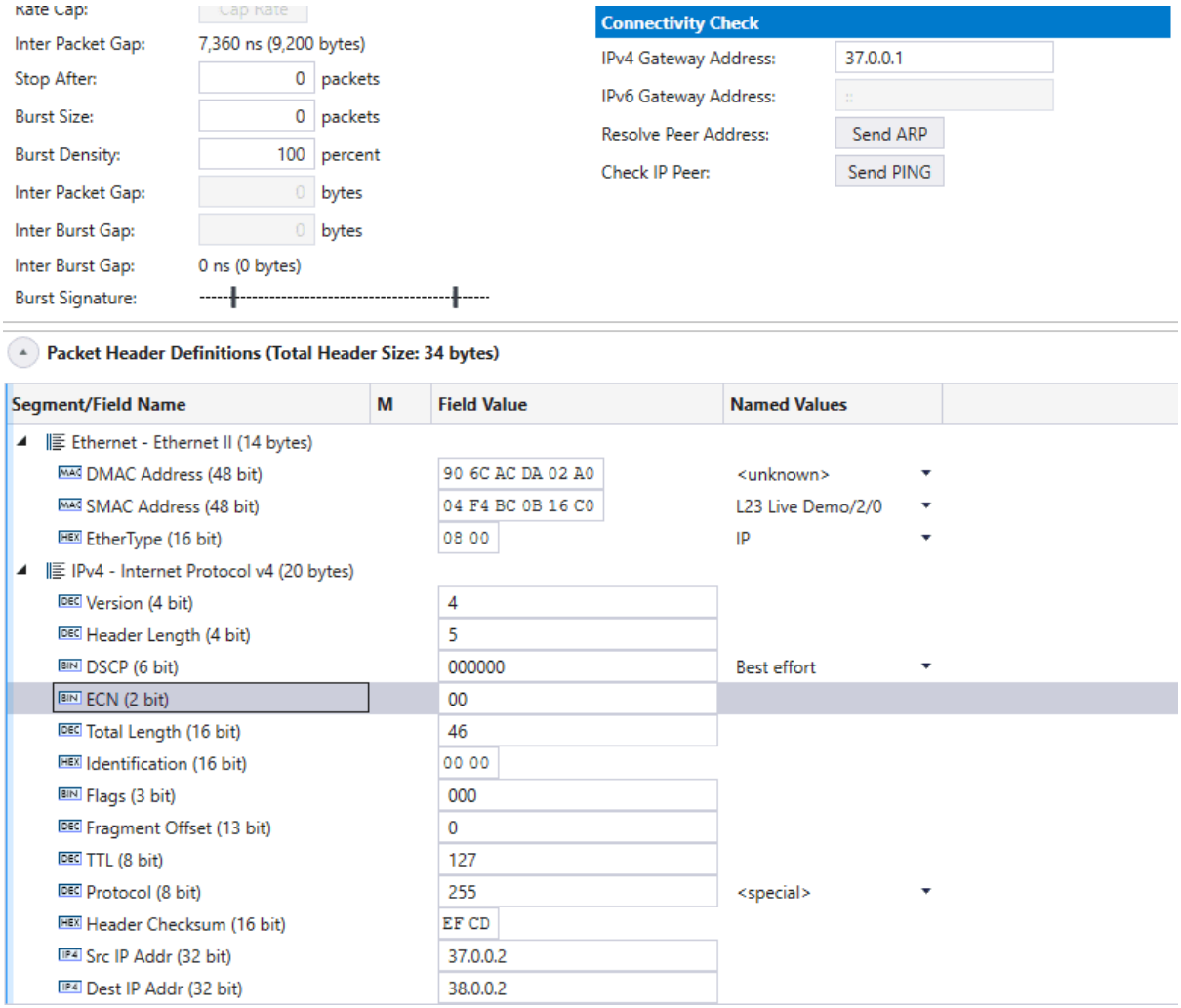


Fig. 7.5: Step 5

6. For stream WAN-to-LAN, source IP = `38.0.0.2`, destination IP = `38.0.0.1`, SMAC = port 1's MAC, DMAC = MAC address of DUT's interface 38. If you do not know the MAC address of DUT's interface, you can enter the IP address of the interface in

Connectivity Check → *IPv4 Gateway Address* and click *Send ARP*. The test port will automatically resolve the DUT's interface MAC address using ARP and put the address value in the stream's DMAC field, as shown below.

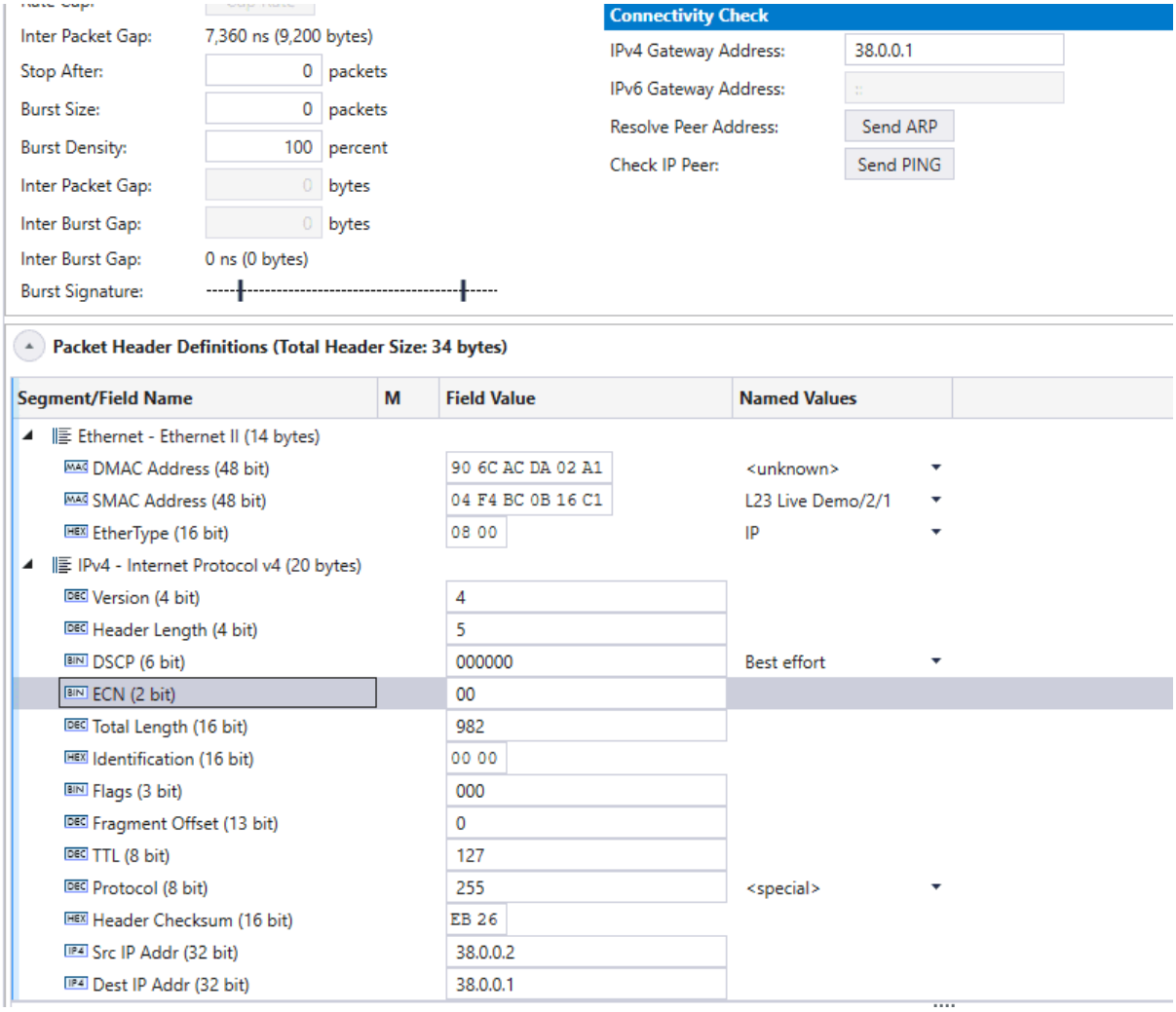


Fig. 7.6: Step 6a

7. Start traffic on both ports and you will check the bidirectional traffic in Global Statistics. You can also go to each port and capture the received traffic to a pcap file and examine it using Wireshark.

7.6 Port Operation

This FAQ page deals with questions related to the test ports.

7.6.1 Q: How can I start/stop traffic on all ports simultaneously?

A: Select the "Global" tab page in the XenaManager and click on the Start button. This will start traffic on all ports in your testbed. Pressing Start will also clear all TX/RX statistics on the ports. It will also re-start capture and any histograms that are marked as under global control.

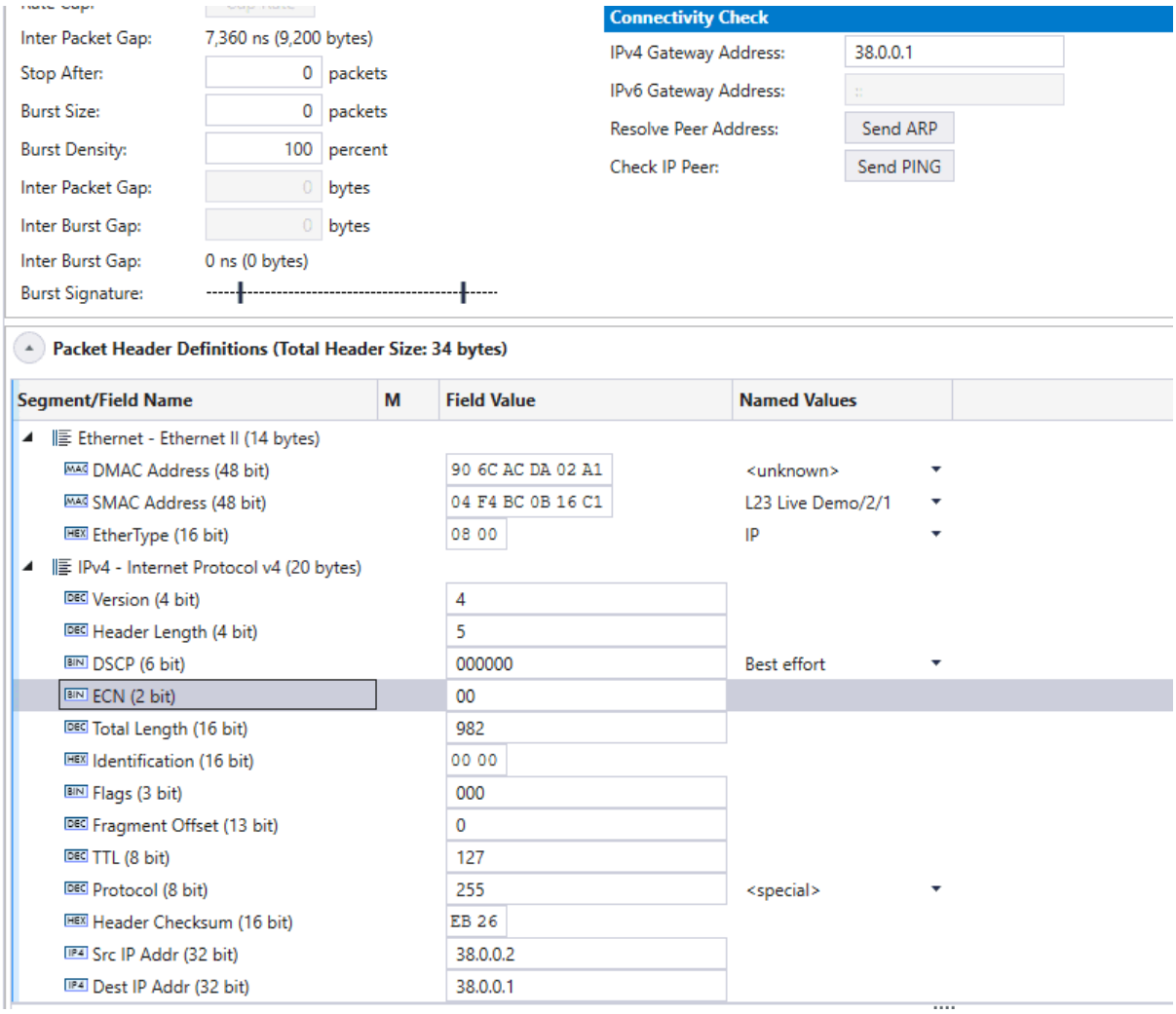


Fig. 7.7: Step 6b

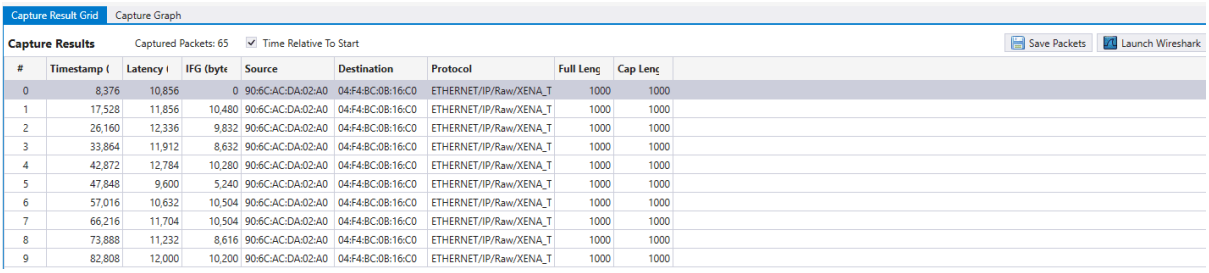


Fig. 7.8: Step 7

7.6.2 Q: Can multiple users reserve test ports on the same test module?

A: Yes, users can reserve test ports one-at-a time, and multiple users can therefore share the same test module. Any given port can however only be reserved by a single user at a time.

7.6.3 Q: Can multiple users view the same port on a test module?

A: Any number of users can view the configuration of any port, as long as they are logged onto the chassis. In order to modify the configuration the port must however be reserved as explained in the previous FAQ item.

7.6.4 Q: How can I move a port configuration between ports?

A: By using the port configuration save/load function. In the XenaManager testbed view, right click the port and select Save Port Configuration. Then, right-click the new port and select Load Port Configuration.

7.6.5 Q: Each time I restart traffic on a port, is the generated packet pattern 100% identical?

A: Yes, the generated traffic pattern is 100% identical each time the traffic is re-started, including all pseudo random generated numbers. To make each run produce a different pattern, you should clear the "Random seed" field in the "PORT PROPERTIES" panel on the "Properties" tab page.

7.6.6 Q: Does a test port send any training packets?

A: The test ports only generate MAC address training packets when this function is enable on the port Resource Properties panel. A test port which is not sending any traffic into the DUT/SUT, but is receiving traffic from the DUT/SUT, should therefore be enabled to generate MAC training packets to avoid MAC table timeout in the DUT/SUT.

7.6.7 Q: Can I view the port configuration of ports which are reserved by another user?

A: Yes, any user can view all configuration and traffic statistics for any test port, also when the test port is reserved by another user. To view the latest port stream configuration of a port reserved by another user, the user must manually select the port, right-click the mouse, and select Refresh Port.

7.6.8 Q: Can I reserve a test port even when it is reserved by another user?

A: Yes, you can reserve a port even when it is reserved by another user, by first performing a port Relinquish command on the Resource Properties panel. The relinquish command will take away the port reservation from the current owner, and the port can then be reserved by you.

7.6.9 Q: What does it mean that the port property ppm speed-reduction is emulated?

A: When the test port ppm speed-reduction function is enabled by specifying a ppm speed-reduction greater than zero, the system will automatically insert an idle traffic period every second, where the length of the idle period corresponds to the specified speed-reduction specified in units of ppm. The resulting effective test port speed is shown below the speed-reduction value. Enabling the ppm speed-reduction function does not change the clock speed of the test port link.

7.6.10 Q: Which packets are looped when a port is in Rx-to-Tx loop mode?

A: When a port is in `L2 Rx-to-Tx` loop mode, it loops traffic back while swapping MAC addresses. Similar, when a port is in L3 Rx-to-Tx loop mode, it loops traffic back while swapping both MAC and IP v4 addresses. Only traffic addressed to the device is forwarded (looped) with MAC and IP addresses swapped, and this is done at wire-speed (100% throughput) regardless of VLAN settings and without any loss.

See also

Read more in *Loopback and Latency*.

7.6.11 Q: Which packets are looped when a port is in Port-to-Port (inline) loop mode?

A: When looping traffic between a port pair (port <0, 1>, port <2, 3>, port <4, 5>, etc), all traffic is looped 100% transparent at L1, regardless of packet FCS errors and any other traffic characteristics.

See also

Read more in *Loopback and Latency*.

7.7 Scripting & Automated Testing

This FAQ page deals with questions related to scripting and automated testing.

7.7.1 Q: Which features are available via scripting?

A: The full feature set is available via the TCP/IP based command line interface. Anything that can be done from XenaManager can also be done via text-based commands.

See also

Read XOA CLI Documentation (online)

7.7.2 Q: Where can I find the documentation of the Xena API scripting commands?

A: Refer to the XOA CLI Documentation (online) for the scripting capability. For PDF download: XOA CLI Documentation (PDF)

7.7.3 Q: Is there a Tcl, Perl, Java or API available?

A: No, we do not provide a full API wrapper for any language. Xena customers have implemented their own proprietary Tcl, Perl, and Java scripting wrappers, where the wrapper converts from Xena's generic TCP/IP command line interface, to the customer's own and proprietary test equipment Tcl/Perl/Java/ API

See also

Examples of the wrappers are hosted on tdl-xoa-python-script-library

7.7.4 Q: Can I run Xena2544 against a device which performs Network Address Translation (NAT)?

A: Yes. The RFC 2544 identifies packets via the Test Payload inserted in each packet, and does not depend on seeing the original source address to validate the throughput. The Xena2544 therefore operates correctly also when the device under test performs NAT.

7.7.5 Q: Are there any wizards for setting up test patterns, e.g. a full mesh traffic load on an n-port DUT?

A: Yes, XenaManager features a stream Wizard that will help you create various configurations for many ports and streams.

CHAPTER

EIGHT

GLOSSARY OF TERMS

AN
Auto-Negotiation

ARP
Address Resolution Protocol

BER
Bit Error Rate

CBS
Committed Burst Size

CIR
Committed Information Rate

CLI
Command Line Interface

DAC
Direct Attach Cable

DHCP
Dynamic Host Configuration Protocol

DIP
Destination IP address

DMAC
Destination MAC address

DSCP
Differentiated Services Bode Point

eCPRI
Enhanced Common Public Radio Interface

ESMC
Ethernet Synchronization Message Channel

FC
Fiber Channel

FCoEHead
Fiber Channel over Ethernet Header

FCoETail
Fiber Channel over Ethernet Tail

FCS
The frame check sequence (FCS) is a four-octet cyclic redundancy check (CRC) that allows detection of corrupted data within the entire frame as received on the receiver side. According to the standard, the FCS value is computed as a function of the protected MAC frame fields: source and destination address, length/type field, MAC client data and padding (that is, all fields except the FCS).

FEC
Forward Error Correction

fid
Flow ID

Fps
Frames per second

Geneve
Generic Network Virtualization Encapsulation

GRE
Generic Routing Encapsulation

GTPv1
GPRS Tunneling Protocol v1

GTPV2L1 - GTPv2 (w/options)
GPRS Tunneling Protocol v2

I2C
I^2C (Inter-Integrated Circuit, eye-squared-C), alternatively known as I2C or IIC, is a synchronous, multi-controller/multi-target (controller/target), packet switched, single-ended, serial communication bus.

IBG
Inter Burst Gap

ICMPv4
Internet Control Message Protocol v4

IFG
Inter Frame Gap

IGMPV1
Internet Group Management Protocol v1

IGMPV2
Internet Group Management Protocol v2

IGMPV3
Internet Group Management Protocol v3

iid
Impairment ID

IPG
Inter Packet Gap

IPv4
Internet Protocol v4

IPv6
Internet Protocol v6

LAN
Local Area Network

LLC
Logical Link Control

LT
Link Training

MAC
Media Access Control

MAC-Ctrl
MAC Control

MPLS
Multiprotocol Label Switching (MPLS) is a routing technique in telecommunications networks that directs data from one node to the next based on labels rather than network addresses.

MPLS-TP OAM
MPLS-TP, OAM Header

N.A.
Not Available / Not Applicable

NDP
Neighbor Discovery Protocol

NVGRE
Generic Routing Encapsulation

PBB
Provider Backbone Bridging Tag

PCP
Priority Code Point

PCS
Physical Coding Sublayer

PFC
Priority Flow Control

PMA
Physical Medium Attachment

Pps
Packets per second

PRBS
Pseudorandom Binary Sequence is a binary sequence that, while generated with a deterministic algorithm, is difficult to predict and exhibits statistical behavior similar to a truly random sequence.

PWE
PW Ethernet Control Word

RoE
Radio over Ethernet

RTCP
Real-time Transport Control Protocol

RTP
Real-time Transport Protocol

Rx
Receive

SCTP
Stream Control Transmission Protocol

SIP
Source IP address

SMAC
Source MAC address

SNAP
Subnetwork Access Protocol

STP
Spanning Tree Protocol

TC
Traffic Class

TCP
Transmission Control Protocol

TG
Traffic Generation

TID
Test Payload Identifier. It is used to identify a sending stream.

TPID
Test Payload ID

TPLD
: Test Payload Data. Each Xena test packet contains a special proprietary data area called the Test Payload Data, which contains various information about the packet. The TPLD is located just before the Ethernet FCS.

Tx
: Transmit

UDP
: User Datagram Protocol

UI
: User Interface

VID
: VLAN ID

VLAN
: Virtual local area network (VLAN) is any broadcast domain that is partitioned and isolated in a computer network at the data link layer (OSI layer 2).

VXLAN
: Virtual eXtensible LAN

search

INDEX

A

AN, 317
ARP, 317

B

BER, 317

C

CBS, 317
CIR, 317
CLI, 317

D

DAC, 317
DHCP, 317
DIP, 317
DMAC, 317
DSCP, 317

E

eCPRI, 317
ESMC, 317

F

FC, 317
FCoEHead, 318
FCoETail, 318
FCS, 318
FEC, 318
fid, 318
Fps, 318

G

Geneve, 318
GRE, 318
GTPv1, 318
GTPV2L1 - GTPv2 (*w/options*), 318

I

I2C, 318
IBG, 318
ICMPv4, 318
IFG, 318
IGMPV1, 318
IGMPV2, 318
IGMPV3, 318
iid, 319
IPG, 319
IPv4, 319
IPv6, 319

L

LAN, 319
LLC, 319
LT, 319

M

MAC, 319
MAC-Ctrl, 319
MPLS, 319
MPLS-TP OAM, 319

N

N.A., 319
NDP, 319
NVGRE, 319

P

PBB, 319
PCP, 319
PCS, 319
PFC, 319
PMA, 320
Pps, 320
PRBS, 320
PWE, 320

R

RFC
    RFC 4861 Section 7.2, 96
RoE, 320
RTCP, 320
RTP, 320
Rx, 320

S

SCTP, 320

SIP, 320
SMAC, 320
SNAP, 320
STP, 320

T

TC, 320
TCP, 320
TG, 320
TID, 320
TPID, 320
TPLD, 321
Tx, 321

U

UDP, 321
UI, 321

V

VID, 321
VLAN, 321
VXLAN, 321