TELEDYNE LECROY
Everywhereyoulook™

TCP throughput testing

APPLICATION NOTE

How to make RFC 6349 based TCP throughput tests with the Teledyne LeCroy Xena layer 4-7 test solution

Contents

When enterprises want to ensure the quality and performance of their communication through a service provider's network they sign a Service Level Agreement (SLA) with the service provider. The SLA will contain worst case values for parameters like bandwidth, latency, packet jitter, frame loss ratio and availability. The service provider can verify that the requirements in the SLA are met using test methodologies like RFC 2544 and Y.1564. However even if these tests show that all criteria fulfilled, the enterprises may complain that they get less bandwidth than expected or that they experience long response times from the applications they use.

The reason for the complaints will in many cases be non-optimal configuration of the Transmission Control Protocol (TCP). TCP improves the "best effort" nature of IP networks by adding mechanisms guaranteeing that data sent to a recipient are actually delivered and in the right order. To provide this functionality TCP needs to buffer the data at both sending and receiving end of a connection. If these buffers are not dimensioned correctly, the enterprises may experience performance degradation.

The Internet Engineering Task Force (IETF) has defined RFC 6349 "Framework for TCP Throughput Testing", which provides a methodology for measuring end to-end TCP Throughput in a managed IP network. In addition to finding the TCP throughput at the optimal buffer size, RFC 6349 presents metrics that can be used to better understand the results. This Application Note describes how the Teledyne LeCroy Xena layer 4-7 test solution XenaBay controlled by XenaManager supports powerful TCP testing based on RCF 6349. In addition XenaBay supports establishing millions of concurrent TCP connections. This gives network operators valuable information on the number of users the network can handle, highlighting capacity bottlenecks.

TCP Testing

RFC 6349 based Throughput Testing

The RFC 6349 “Framework for TCP Throughput Testing” provides a methodology for measuring end to-end TCP Throughput in a managed IP network.

RFC 6349 testing is done in 3 steps:

- Identify the Path Maximum Transmission Unit (MTU)
- Identify Baseline Round-Trip Time (RTT) and Bottleneck Bandwidth (BB)
- TCP throughput tests

Before starting the TCP throughput tests RFC 6349 recommends that layer 2-3 tests are conducted to verify the integrity of the network. This may be manual measurements of throughput, loss and delay. This can also be done with RFC 2544 tests (although RFC 2544 was not intended to be used outside a lab) or Y.1564 tests.

Identify the Path MTU

To identify the Path MTU, Packetization Layer Path MTU Discovery (PLPMTUD) in accordance with RFC4821 should be conducted. It is important to identify the path MTU so the TCP tests can be configured to avoid that test frames are fragmented. RFC 4821 describes a method for Path MTU Discovery (PMTUD) where an Internet path is tested with increasing packet sizes.

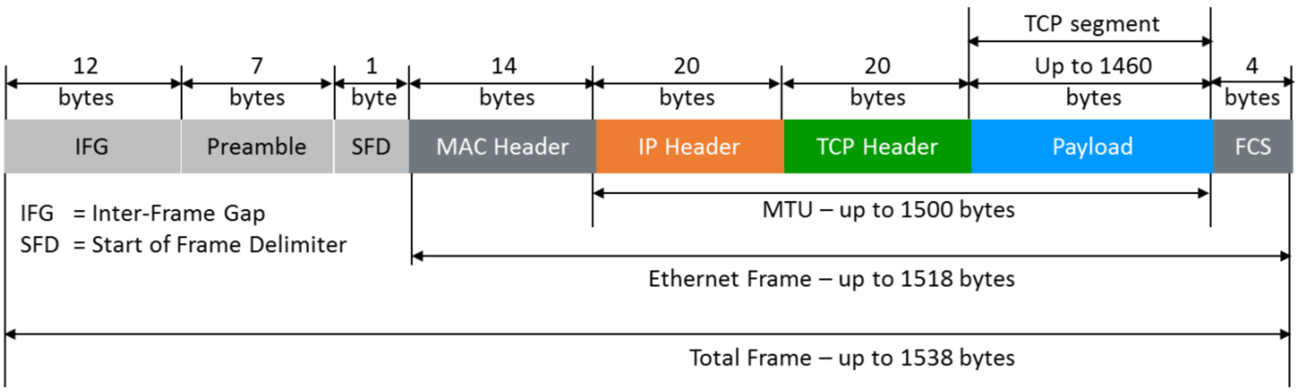


Figure 1: How the MTU and the TCP segment fits into an Ethernet/IP/TCP frame

The maximum supported Ethernet frame size is 1518 bytes from start of the MAC header to the end of the FCS, which equals a 1500 byte MTU (see figure 1). If it is known for sure that the network supports 1518 byte Ethernet frames without needing to fragment them, this may be sufficient. If there is a need to check the MTU this can be done with a Teledyne LeCroy Xena Layer 2-3 tester running a RFC 2544 Throughput test as described in the appendices.

Identify Baseline Round-Trip Time (RTT) and Bottleneck Bandwidth (BB)

After the MTU has been identified, the inherent RTT and the BB of the end-to-end network path must be measured. The BB can be identified with a Teledyne LeCroy Xena Layer 2-3 tester running using a RFC 2544 Throughput test as described in the appendices. However in many cases it is relevant to use the Committed Information Rate (CIR) defined for the line to be tested in a Service Level Agreement (SLA) between service provider and customer as BB.

The RTT can be identified using two Teledyne LeCroy Xena Layer 4-7 testers in the setup illustrated in figure 2.

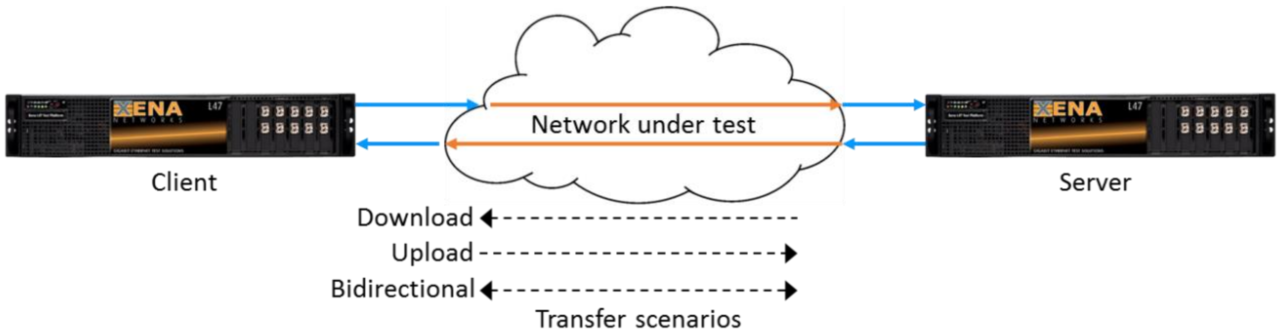


Figure 2: RTT and TCP throughput test configuration with two Teledyne LeCroy Xena layer 4-7 testers

- Activate the XenaManager
- Click on **Client <-> Server Topology** Quick start with a default Client and Server setup.
- Connect to the two Teledyne LecCroyXena layer 4-7 testers (Add in the Chassis Explorer)
- Reserve a port on both testers
- Click Test to open the Test Explorer
- Unfold ▸ Subnets and set Client IPv4 and Server IPv4 to relevant values for the network under test
- Under Test cases Click on Test case 0
- Change the name to “RTT Test”
- Click Add Scenario
- Under Stateful Loading sele Raw and TCP IPv4 . Click OK

- Set Client Subnet to Client IPv4 and Server Subnet to Server IPv4
- Select the two reserved ports as Client Port and Server Port
- Set Users to 1 (see figure 3)

Figure 3: Set Client and Server Subnet , Client and Server Port and number of Users

RFC 6349 defines the baseline RTT as the Round-Trip Time inherent to the network path under non-congested conditions. Therefore to minimize the load during the RTT test the configuration will be set to have only 1 user (connection).

- In the Test Explorer: Unfold Scenario 0 for t RTT Test Test case

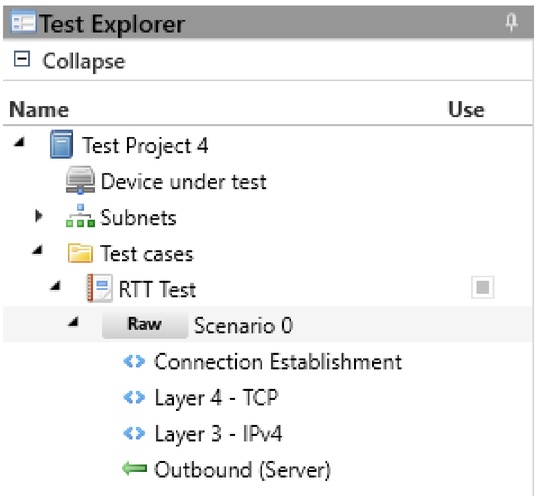


Figure 4: Test Explorer window with Scenario 0 unfolded

- Click on Layer 4 - TCP
- If MTU is less than 1500 bytes adjust Maximum TCP Segment Size (Client and Server) to MTU – 40 bytes (40 bytes is the size of the IP and TCP headers (see figure 1))
- For other parameters: Use the XenaManager default values

- Click on the Run Test tab and cli Run

Reporting

- When the test is stopped click on the tab
- In the Server section of the Statistics Explorer unfold the Scenario 0, Unfold TCP RTT and select RTT (figure 5)

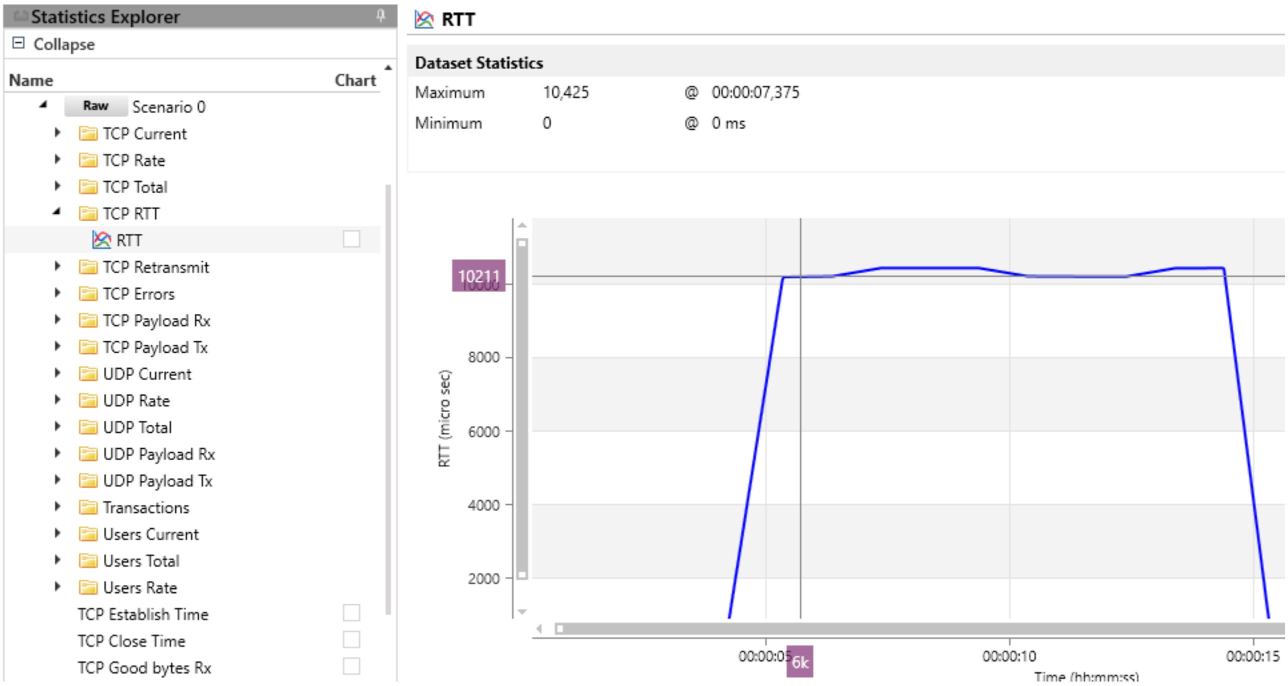


Figure 5: Statistics Explorer with RTT results including a graph of measured RTTs during the test. Use cursor to find minimum value during the period where data transmission is going on (“steady” period in figure 7).

The XenaManager will default to download (from server to client) so it will be the server that actually sends data. Therefore the RTT information is generated by the server end of the connection and can be found in the Server section of the Statistics Explorer.

- Read the RTT and use that as baseline RT
- Click on the 

tab and cli 

to prepare XenaManager for a new test

TCP Throughput Tests

Last step in a RFC 6349 test is the TCP Throughput Tests. Based the RTT and BB information, single and multiple connection TCP throughput tests should be performed to identify the network performance. In figure 5 the RTT was measured to 10.211 msec and. The BB/CIR of the connection is in this example known to be 100 Mbps. With this information the Bandwidth-Delay Product (BDP) can be calculated:

BDP = BB x RTT = 100 Mbps x 10.211 msec = 1.0211 Mbits, which equals 127,638 bytes.

The BDP is the size that should be used for the send and receive buffers (or windows) for the TCP connection to achieve the maximum possible throughput. However normal maximum size for these windows is 65535 bytes, so to fill up the pipe several connections must be activated concurrently. In this example we will use 4 connections each with a window size of 127,638 bytes / 4 = 31,909.5 bytes (rounded up to 31,910 bytes).

To generate this we can add a new Test case to the Test Project already made for the RTT identification.

- Click on the Edit tab and click Add Testcase
- Under Test cases Click on Test case 1
- Change the name to "Download" (the test case behavior scenario defaults to download from server to client)
- Click Add Scenario
- Under Stateful loading select Raw and TCP IPv4. Click OK
- Set Client Subnet to Client IPv4 and Server Subnet to Server IPv4
- Select the two reserved ports as Client Port and Server Port

Identity			*Subnets - Ports*			
Active	**Type**	**Name**	**Client Subnet**	**Server Subnet**	**Client Port**	**Server Port**
☐	Raw	Scenario 0	Client IPv4	Server IPv4	P-0-1-1	P-0-1-3

Figure 6: Set Client and Server Subnet and Port

- In the Test Explorer Window Unfold the Scenario 0 for Test Case 1
- Click on Connection Establishment
- Set Users to 4 to get 4 simultaneous connections (figure 7) (this could have been done earlier - in the previous step where Client and Server Subnets were defined)

Users	Start Offset	Rampup	Steady	Rampdown	Time Scale
4	0	5	10	5	Seconds

Figure 7: 4 Users selected in Connection Establishment, where also the actual Transfer duration (the “Steady” period) can be adjusted

- Click on Layer 4 - TCP for the Downlink Test Case
- Set TCP (Client) Windows Size to 31910 bytes
- Set TCP (Server) Windows Size to 31910 bytes
- If MTU is less than 1500 bytes click on Layer 4 – TCP and adjust Maximum TCP Segment Size (Client and Server) to MTU – 40 bytes (40 bytes is the size of the IP and TCP headers (see figure 1))

The Download test case is now ready to be executed. However RFC 6349 recommends to run the tests in each direction independently first and then to run them in both directions simultaneously. So before executing the Download test we can create Upload and Bidirectional test cases (see figure 2). These test cases will basically have same configuration as the Download test case – except of course with a different name and traffic direction.

- In the Test Explorer right-click on t Downlink test case and choose copy from the menu that pops up
- In the Test Explorer right-click (Test cases and choose paste from the menu that pops up
- Repeat last step

You should now have two new test cas Download - Copy Download - Copy (2)

- Click or Download - Copy and change name to Upload
- Select the two reserved ports as Client Port and Server Port (same as for the Download test case)
- Unfold the (new) Upload test case and clicl Raw Scenario 0 for the Upload test case.
- Change Behavior / Scenario to Upload

- Click or Download - Copy (2) and change name to Bidirectional
- Select the two reserved ports as Client Port and Server Port (same as for the Download test case)
- Unfold the (new) Bidirectional test case and click Raw Scenario 0 for the Bidirectional test case
- Change Behavior / Scenario to Bidirectional

There should now be 4 test cases in the Test Explorer as shown in figure 8.

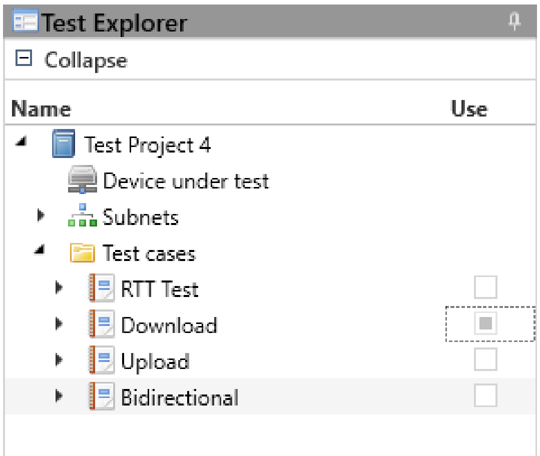


Figure 8: Test Explorer with 4 test cases (unfolded) – the Download test case is selected

- Select the Download test case
- Click on the Run Test tab and cli Run
- When the test is stopped click on t Reporting tab

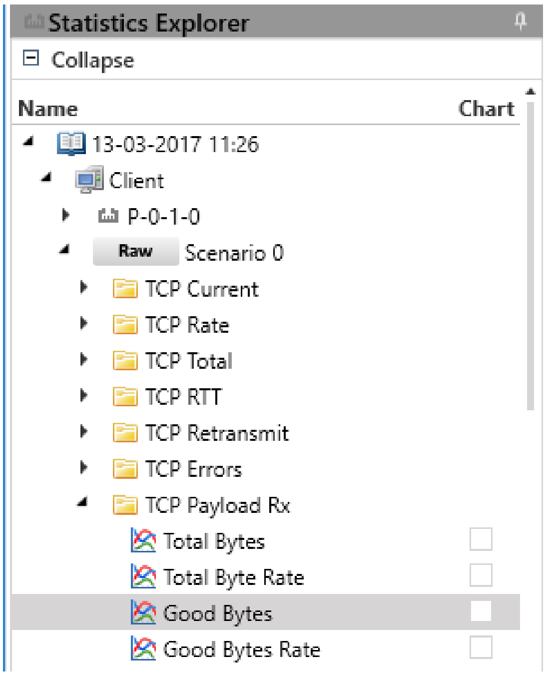


Figure 9: Statistics Explorer with TCP Payload Rx results unfolded for the Client

- In the Client section of the Statistics Explorer unfolc Raw Scenario 0 TCP Payload Rx

In the TCP Payload Rx folder there are 4 results:

- Total Bytes
- Total Bytes Rate
- Good Bytes
- Good Bytes Rate

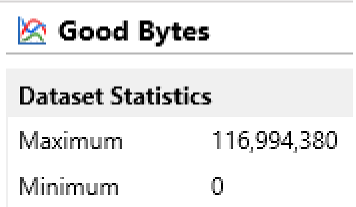


Figure 10: Good Bytes count as result of the Download test case

Total bytes are all bytes received; Good bytes are those acknowledged by the receiving end – excluding retransmitted bytes. Therefore the actual TCP throughput is the same as the Good Bytes count divided by the time that the transmission actually lasted i.e. the Transfer duration (the “steady” period in figure 7 i.e. 10 seconds).

$$\text{Actual TCP Throughput} = \frac{\text{Received Good Bytes}}{\text{Transfer duration}}$$

In this case the Good Bytes count is 116,994,380 meaning that the TCP throughput is 11,699,438 bytes/second. The actual TCP throughput can be expressed in bits/second (bps) by multiplying this number by 8:

Actual TCP Throughput = 11,699,438 bytes/second X 8 bits/byte = 93,595,504 bps

- In the Server section of the Statistics Explorer unfold the Scenario 0, Unfold TCP RTT and select RTT
- Unfold TCP Payload Tx and select Total Bytes
- Read Total Bytes (in this case 116,994,380 bytes) and use that for the TCP Efficiency % calculation (see later)

- Click on the  tab and cli  to prepare XenaManager for a new test

Maximum Achievable TCP Throughput Calculation

If you want to compare with what can be achieved on the line with BB = 100 Mbps (i.e. the Maximum Achievable TCP Throughput) you first calculate maximum Frame Per Second (FPS) for the connection. Bytes sent per frame are MTU plus Inter-Frame Gap (IFG; 12 bytes), Preamble (7 bytes), Start of Frame Delimiter (SFD; 1 byte) MAC header (14 bytes) and FCS (4 bytes) i.e. 1538 bytes in this case:

FPS = 100 Mbps / (1538 Bytes X 8 bits) = 8127.44 frames/second

This is then multiplied with the maximum segment size (MTU – 40 (Total length of IP and TCP address fields)) i.e. 1460:

Maximum Achievable TCP Throughput = 8127.44 X 1460 = 11,866,059.8 bytes/second = 94,928,478.4 bps

RCF 6349 Metrics

Together with the TCP throughput measurement, RFC 6349 presents metrics that can be used to better understand the results including:

- The TCP Transfer Time Ratio
- The TCP Efficiency Percentage

These metrics must be measured in each direction.

The **TCP Transfer Time Ratio** is the ratio between the time it actually takes to transfer a block of data compared with the ideal TCP transfer time i.e. what should be possible considering the BB of the Network Under Test (NUT) and the RTT/2:

$$\text{TCP Transfer Time Ratio} = \frac{\text{Actual TCP Transfer Time}}{\text{Ideal TCP Transfer Time}}$$

Where

$$\text{Ideal TCP Transfer Time} = \frac{\text{Size of the Block of Data}}{\text{Maximum Achievable TCP Throughput}}$$

For the TCP Transfer Time Ratio calculation all bytes that were received excl. retransmitted bytes (i.e. the received Good Bytes) can be considered to be the “Block of Data” and then the Actual TCP Transfer Time is the same as the ”Transfer duration”.

This means that:

$$\text{Ideal TCP Transfer Time} = \frac{\text{Received Good Bytes}}{\text{Maximum Achievable TCP Throughput}}$$

Hereby

$$\text{TCP Transfer Time Ratio} = \frac{\text{Transfer duration}}{\text{Received Good Bytes}} \text{ X Maximum Achievable TCP Throughput}$$

As shown earlier:

$$\text{Actual TCP Throughput} = \frac{\text{Received Good Bytes}}{\text{Transfer duration}}$$

So, in this case:

$$\text{TCP Transfer Time Ratio} = \frac{\text{Maximum Achievable TCP Throughput}}{\text{Actual TCP Throughput}}$$

In the example Maximum Achievable TCP Throughput is 94,928,478.4 bps, while Actual TCP Throughput was measured to be 93,595,504 bps. So TCP Transfer Time Ratio = 94,928,478.4 bps/93,595,504 bps = 1.01424.

The **TCP Efficiency Percentage** gives the percentage of Bytes that were not retransmitted:

$$\text{TCP Efficiency \%} = \frac{\text{Transmitted Bytes - Retransmitted Bytes}}{\text{Transmitted Bytes}} \text{ X 100}$$

As the Transmitted Bytes - Retransmitted Bytes is the same as the received Good Bytes (Client TCP Payload Rx Good Bytes) in XenaManager you just calculate the ratio between Good Bytes received by the client and Total Bytes transmitted by the server (Server TCP Payload Tx Total Bytes) and multiply by 100:

$$\text{TCP Efficiency \%} = \frac{\text{Client TCP Payload Rx Good Bytes}}{\text{Server TCP Payload Tx Total Bytes}} \text{ X 100}$$

In the example both Client TCP Payload Rx Good Bytes and Server TCP Payload Tx Total Bytes were measured to be 116,994,380 bytes, so the TCP Efficiency % = 100 %.

Once the results from Download test case have been calculated, the Upload and the Bidirectional test cases can be selected and run one by one and the related results can be calculated. Table 1 shows the parameters to use for the TCP Throughput and the TCP Efficiency % calculations in the 3 test cases.

Test case	Actual TCP Throughput	TCP Efficiency %
Download	$\frac{\text{Client TCP Payload Rx Good Bytes}}{\text{Transfer duration}}$	$\frac{\text{Client TCP Payload Rx Good Bytes}}{\text{Server TCP Payload Tx Total Bytes}}$ X 100
Upload	$\frac{\text{Server TCP Payload Rx Good Bytes}}{\text{Transfer duration}}$	$\frac{\text{Server TCP Payload Rx Good Bytes}}{\text{Client TCP Payload Tx Total Bytes}}$ X 100

Bidirectional (download)	Client TCP Payload Rx Good Bytes / Transfer duration	Client TCP Payload Rx Good Bytes / Server TCP Payload Tx Total Bytes	X 100
Bidirectional (upload)	Server TCP Payload Rx Good Bytes / Transfer duration	Server TCP Payload Rx Good Bytes / Client TCP Payload Tx Total Bytes	X 100

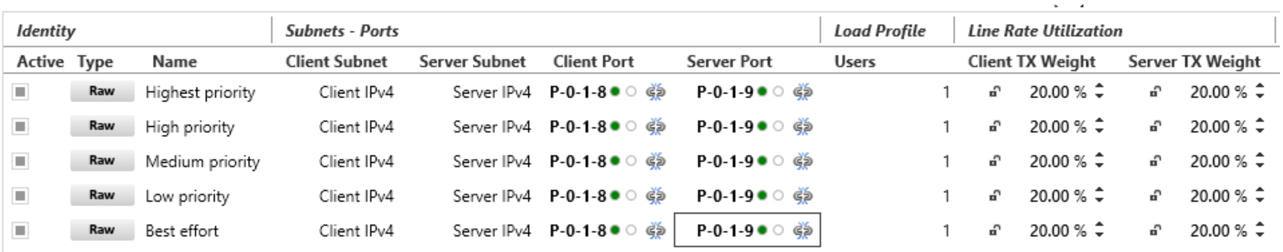
Table 1: Actual TCP Throughput and the TCP Efficiency % calculations.

TCP Throughput Tests with Reduced Buffer Sizes

RFC 6349 suggests that TCP throughput can be measured at different buffer sizes up to the BDP. This can be done by repeating one or more of the test cases on the previous pages with total window sizes at 25%, 50% and 75% of the BDP. In this case this can be done simply by reducing numbers Connection Establishment 1, 2 and 3 in to achieve the three window sizes. Run the 3 tests with the different windows sizes and calculate TCP throughput (based on Good Bytes) and the maximum theoretical TCP throughput. Measured and calculated values can now be compared.

TCP Throughput Tests using Connections with Different Priority

IP networks may support Differentiated Services or DiffServ, which is used for classifying, prioritizing and managing traffic through the network based on information in the Differentiated Services Code Point (DSCP) field in the IP header. This can be tested using concurrent connections with different DiffServ/DSCP values.



Identity			Subnets - Ports				Load Profile	Line Rate Utilization	
Active	Type	Name	Client Subnet	Server Subnet	Client Port	Server Port	Users	Client TX Weight	Server TX Weight
	Raw	Highest priority	Client IPv4	Server IPv4	P-0-1-8	P-0-1-9	1	20.00 %	20.00 %
	Raw	High priority	Client IPv4	Server IPv4	P-0-1-8	P-0-1-9	1	20.00 %	20.00 %
	Raw	Medium priority	Client IPv4	Server IPv4	P-0-1-8	P-0-1-9	1	20.00 %	20.00 %
	Raw	Low priority	Client IPv4	Server IPv4	P-0-1-8	P-0-1-9	1	20.00 %	20.00 %
	Raw	Best effort	Client IPv4	Server IPv4	P-0-1-8	P-0-1-9	1	20.00 %	20.00 %

Figure 11: Segment configuration for the Different priorities test case

- Create a new Test case, which can be named Different priorities
- Create a segment 5 times for the new test case
- In the setup page for the for the Different priorities test case change names for the 5 scenarios e.g. to Highest priority, High priority, Medium priority, Low priority and Best effort

- For the 5 scenarios: Set Client Subnet to Client IPv4 and Server Subnet to Cl Server IPv4
- For the 5 scenarios: Select the two reserved ports as Client Port and Server Port
- For the 5 scenarios: Set users to 1

	Source Port Minimum	DiffServ (Client)	DiffServ (Server)
Highest priority	49152	46	46
High priority	49153	10	10
Medium priority	49154	12	12
Low priority	49155	14	14
Best effort	49156	0	0

Table 2: Suggested values for Source Port Minimum, DiffServ (Client) and DiffServ (Server). With different Source Port values different applications will be emulated.

- In the Test Explorer: Unfold the Highest priority segment
- Click on Connection Establishment
- Untick Use Ephemeral Source Port Range: and set Source Port Minimum to 49152
- Click or Layer 3 - IPv4 and set DiffServ (Client) and DiffServ (Server) to the required values – some DSCP values are listed in table 3 in the appendices.
- Unfold the other segments and update them like the Highest priority segment. Table 2 shows suggested values for Source Port Minimum, DiffServ (Client) and DiffServ (Server) for the 5 segments In the appendices you will find more DSCP value examples

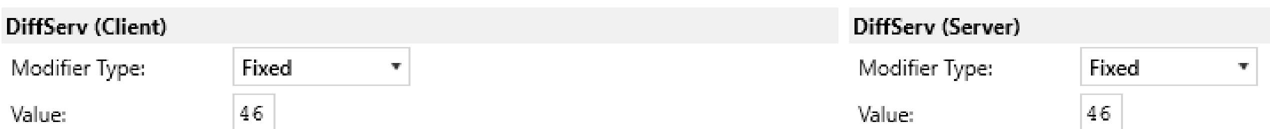


Figure 12: Defining DiffServ (Client) and DiffServ (Server) values for the Highest priority segment

- Run the test
- When the test is stopped click on tl Reporting tab
- In the Client section of the Statistics Explorer read the Good Bytes counters for the different segments and see how they got through the network. If the capacity is exceeded, there should be fewer (or no) bytes received by the lowest priority segments

- In the Server section of the Reporting/Statistics Explorer read the RTT counters for the different segments and see if there are differences

Extreme RFC 6349 Stress Load Testing

In addition to the tests defined in RFC 6349 it is also important to examine a networks behavior when many users run TCP based applications simultaneously and see if the network capacity is exceeded. To simulate this extreme RFC 6349 stress load testing is required, generating many hundred thousand TCP connections at one time. This also applies to the new networks based on Software-Defined Networking (SDN) and Network Function Virtualization (NFV), where software based appliances run as Virtual Network Functions (VNF) on Commercial Off-The-Shelf (COTS) hardware to provide the required functionality. Stress load testing can verify if the SDN based networks have the expected TCP connection capacity.

To configure an extreme RFC 6349 stress load test you can base it on one of the already conducted test cases e.g. the bidirectional test case. In Edit/T Connection Establishment and change number of users (connections) to the number of connections the network should be able to handle. A test can also be done with a number of connections higher than the network capacity. XenaBay supports up to 24 million TCP connections. In figure 13 1,000,000 users are selected.

Users	Start Offset	Rampup	Steady	Rampdown	Time Scale
1000000	0	5	10	5	Seconds

Figure 13: XenaManager configured to emulate 1,000,000 Users/connections

As for the previous test cases click on Run Test tab and c Run on

During and after the test XenaManager will present number of active users and the throughput, see figure 14. The Active Users graph shows behavior of a network capable of handling the high number of users.

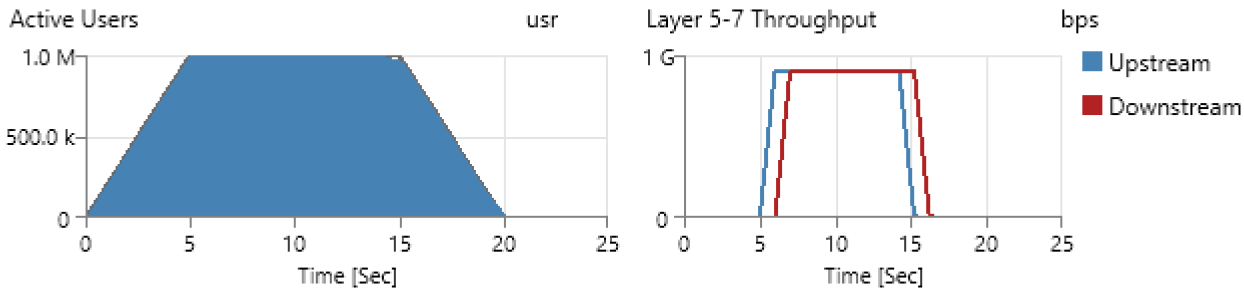


Figure 14: Graphs showing the networks ability to handle a high number of users

In the Reporting/Statistics Explorer the received Good Bytes for client and server can be read and TCP throughput can be calculated. Hereby it can be checked if the maximum throughput is also achieved when a high number of concurrent users/connections is activated on the network.

Appendices

Use RFC 2544 Throughput Test to Identify the Path MTU

A RFC 2544 throughput test can be used to find the actual MTU by sending traffic with increased frame size and having the Don't Fragment (DF) bit set in the IP header. If the traffic passes a network element that need to fragment the data the traffic will be discarded. The test should be done with Teledyne LeCroy Xena layer 2-3 testers using the test configuration shown in figure 15.

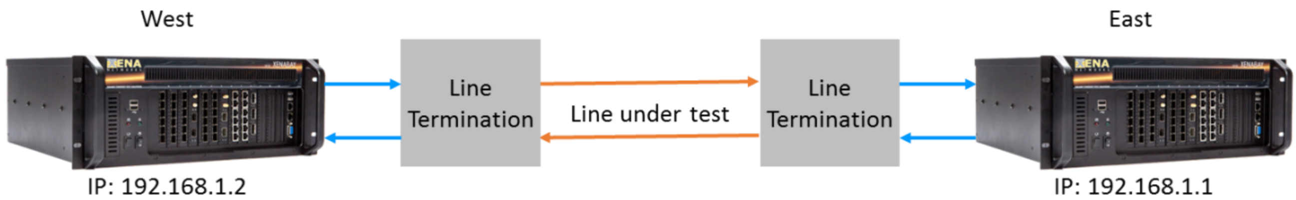
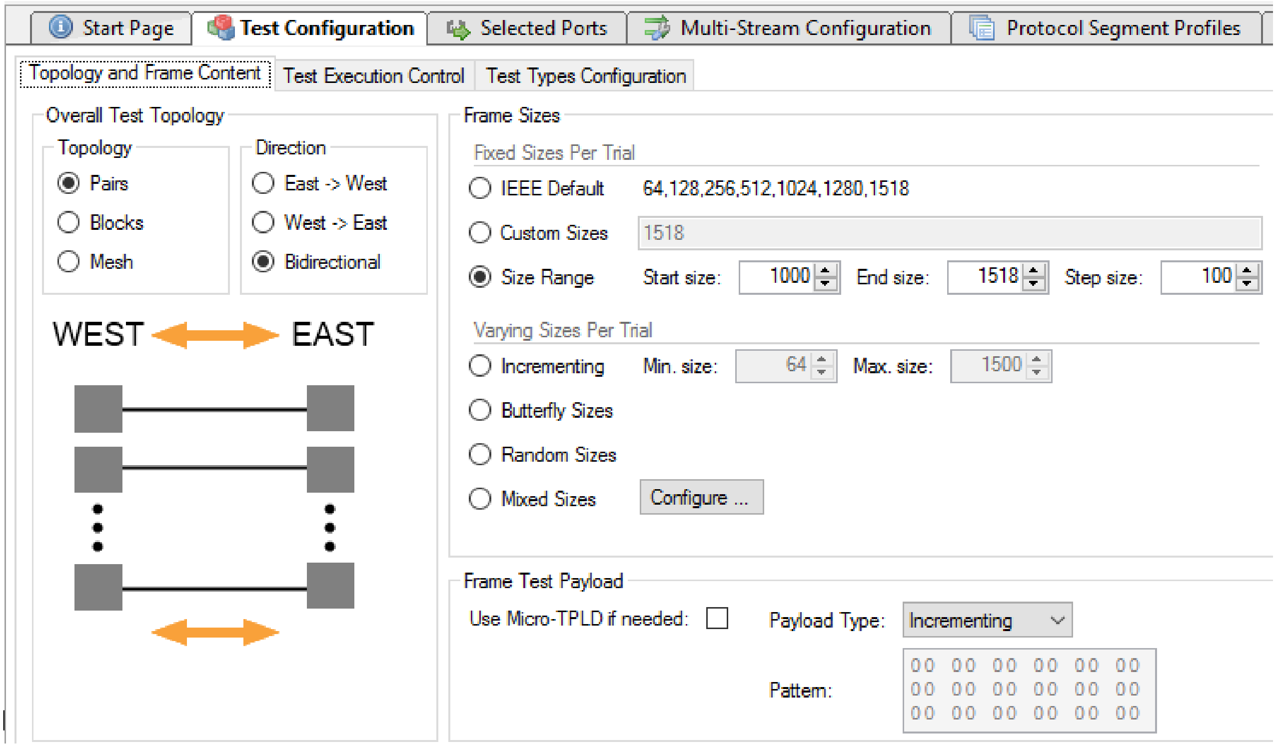


Figure 15: Path MTU and BB test configuration with two Teledyne LeCroy Xena layer 2-3 testers



- Activate the RFC 2555 test tool Valkyrie2544
- Connect to the two Teledyne LeCroy Xena layer 2-3 testers (Add Chassis)
- Reserve a port on both testers
- Set Test Configuration as shown in figure 16. The configuration will support bidirectional testing between two testers and tests with frame sizes from 1000 bytes to 1518 bytes. Step size is set to 100. If finer resolution is required the step size can be reduced.

- Set Test Types Configuration as shown in figure 17. The Throughput test is activated with 10% line load which should not overload the line. Test duration is set to minimum to speed up the test.

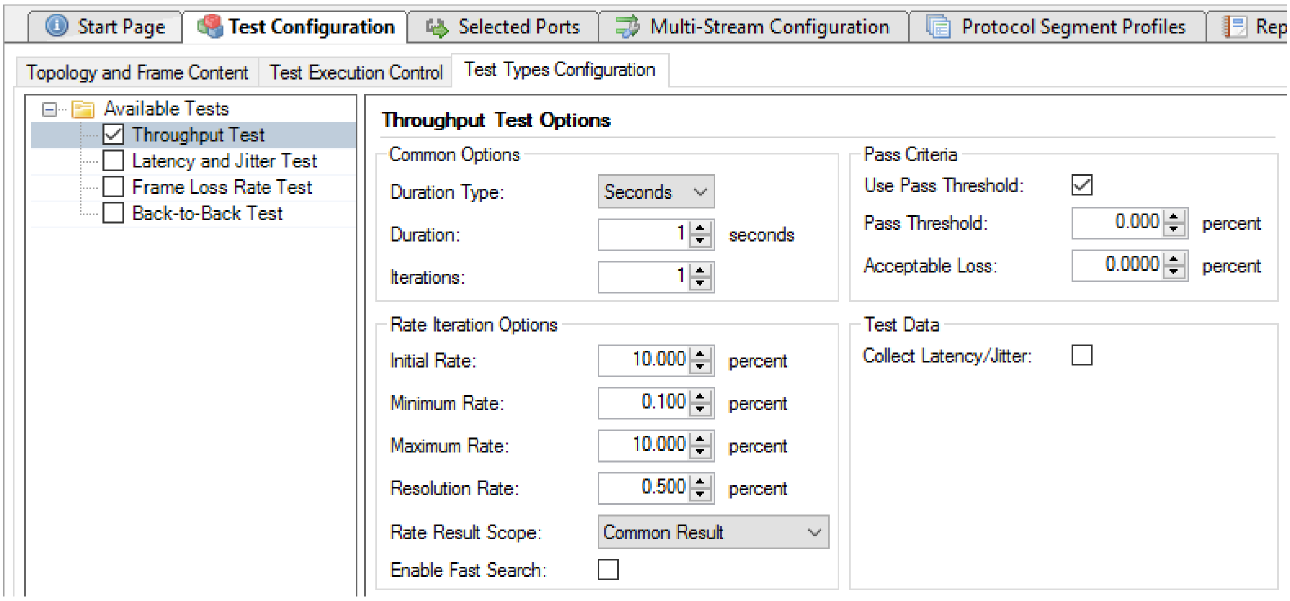
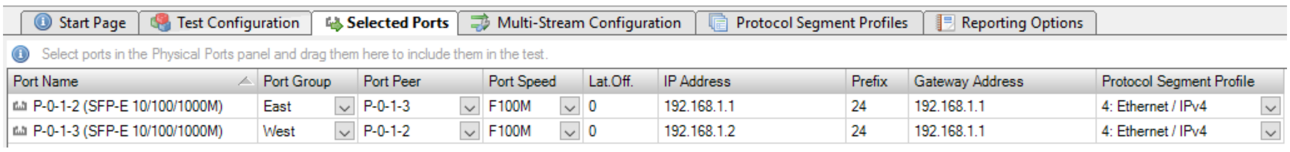


Figure 17: Test types configuration for Path MTU test

- Configure the Selected Ports as shown in figure 18 with the IP source addresses of the testers. Protocol Segment Profile must be Ethernet/IPv4.



Port Name	Port Group	Port Peer	Port Speed	Lat.Off.	IP Address	Prefix	Gateway Address	Protocol Segment Profile
P-0-1-2 (SFP-E 10/100/1000M)	East	P-0-1-3	F100M	0	192.168.1.1	24	192.168.1.1	4: Ethernet / IPv4
P-0-1-3 (SFP-E 10/100/1000M)	West	P-0-1-2	F100M	0	192.168.1.2	24	192.168.1.1	4: Ethernet / IPv4

Figure 18: Selected Ports configuration for Path MTU test

- In the Protocol Segment Profiles (figure 19): Select the Ethernet/IPv4 profile
- Unfold the IPv4 header as shown in figure 19 and set the Flags bits to “”010”
- Please observe that the IP address values shown for the IPv4 header will be overwritten when the test is running by the values entered in the Selected Ports page (figure 18).

The 3 bit Flags field in the IP header is used to control or identify fragments. The bits are (from most significant to least significant):

- bit 0: Reserved; must be zero
- bit 1: Don't Fragment (DF)
- bit 2: More Fragments (MF)

By entering the value 010 into the Flags bits field the DF bit is set.

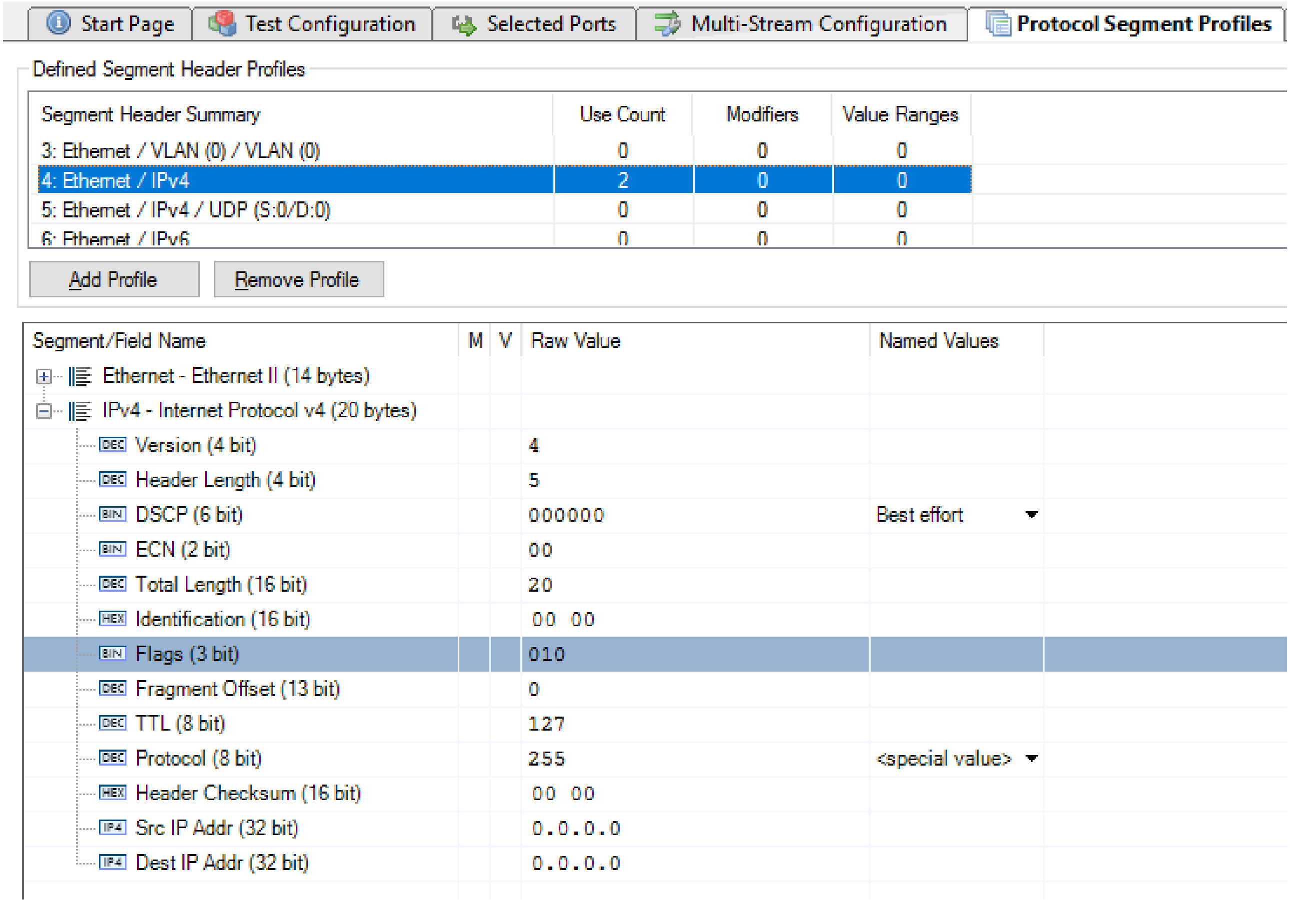


Figure 19: Protocol Segment Profiles for Path MTU test

- Press Start to activate the RFC 2544 test. The test will now run and at the end all test results will be available. The results are shown as totals for the 2 ports together (figure 20) and as individual results for the 2 ports. The highest frame size without frame loss (in this case 1518 bytes) must be used for the calculation of the MTU: The total length of MAC header and FCS (i.e. 18 bytes as shown in figure 1) must be subtracted from the 1518 bytes, giving an MTU of 1500 bytes.

Throughput Test

Totals

Frame Size	Result State	Iter. #	Tx Rate (Percent)	Tx (Frames)	Tx Rate (L1) (Bit/s)	Tx Rate (L2) (Bit/s)	Tx Rate (Fps)	Rx (Frames)	Loss (Frames)	Loss Rate (Percent)	BER (est)	FCS Errors (Frames)
1000	PASS	1	100 %	24,508	199.99 M	196.06 M	24,508	24,508	0	0 %	0	0
1100	PASS	1	100 %	22,320	199.99 M	196.42 M	22,320	22,320	0	0 %	0	0
1200	PASS	1	100 %	20,490	199.98 M	196.7 M	20,490	20,490	0	0 %	0	0
1300	PASS	1	100 %	18,938	199.99 M	196.96 M	18,938	18,938	0	0 %	0	0
1400	PASS	1	100 %	17,604	199.98 M	197.16 M	17,604	17,604	0	0 %	0	0
1500	PASS	1	100 %	16,446	199.98 M	197.35 M	16,446	16,446	0	0 %	0	0
1518	PASS	1	100 %	16,254	199.99 M	197.39 M	16,254	16,254	0	0 %	0	0

Figure 20: Test results (totals) for Path MTU test

Use RFC 2544 Throughput Test to Identify BB

The BB can be identified using a reconfigured RFC 2544 Throughput test with the test configuration shown in figure 15. In the Test Configuration page Frame Sizes must be changed to the MTU value just identified plus 18 bytes (total length of MAC header and FCS) i.e. 1518 bytes in this case (figure 21).

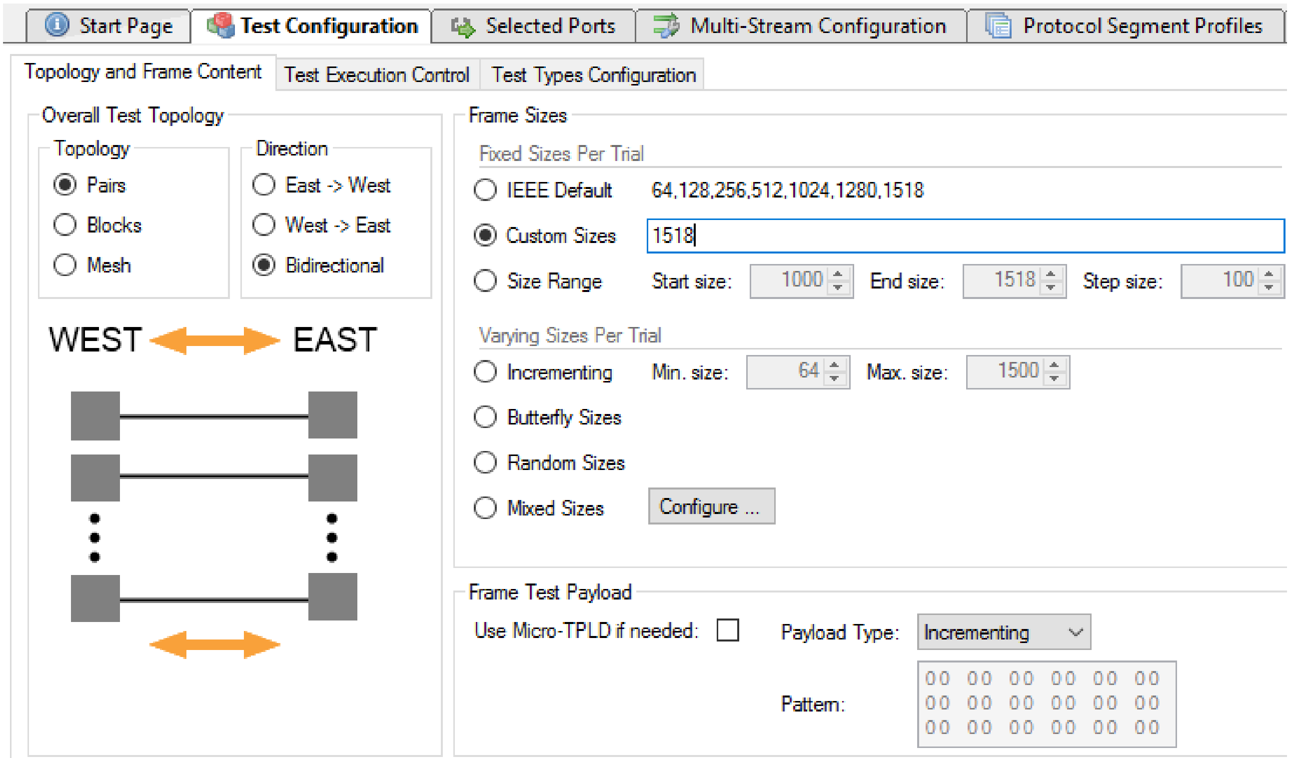


Figure 21: Test Configuration page for BB test

- Set Test Types Configuration as shown in figure 22. The Throughput test is will test with a traffic rate from 10% to 100% of the line rate. Test duration is set to the minimum value (1 sec) to speed up the test.

Enable Fast Search in figure 22: The default iteration algorithm used for the throughput test is a standard binary search, where the next attempted rate is found as the mean value of the sum of last passed and the last failed rate. If the fast search property is enabled the algorithm will take the measured loss rate into account when iterating down. This may in many cases result in a substantial reduction in the number of trials needed to reach the throughput rate result.

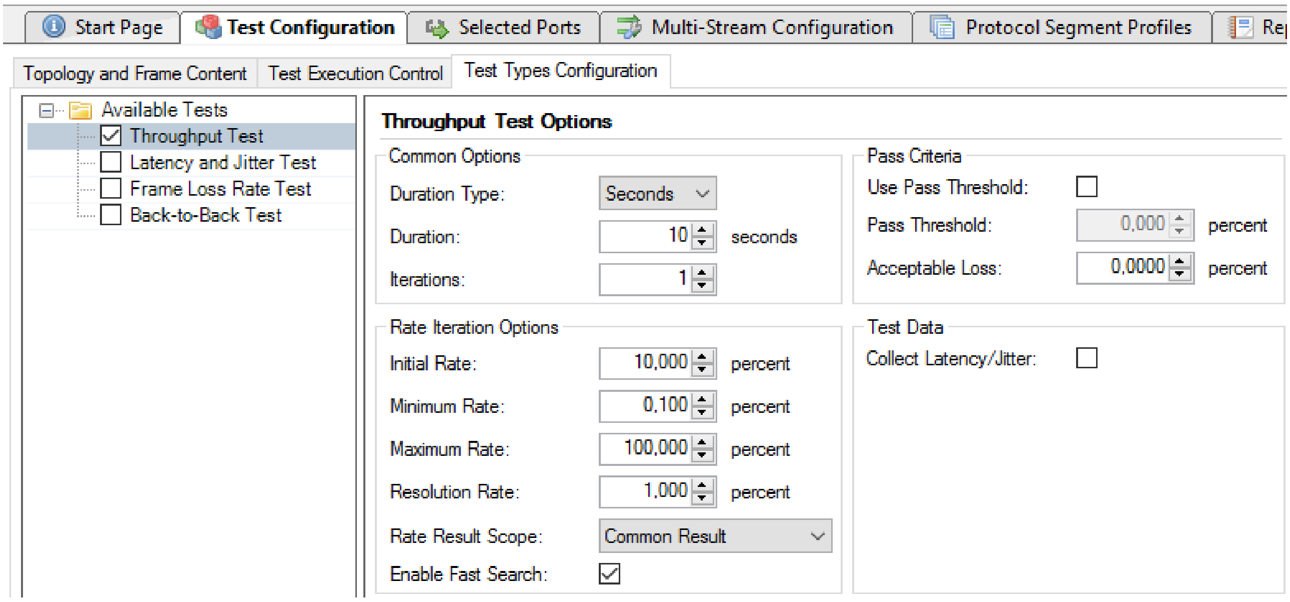


Figure 22: Test types configuration for the BB test

The test is started again and results are available after a while (see figure 23. Please inspect the Rx Rate (L1) value for both ports; the lowest value must be selected, which in this case is 99.99 Mbps.

Throughput Test

Frame Size	Result State	Iter. #	Tx Rate (Percent)	Tx (Frames)	Tx Rate (L1) (Bit/s)	Tx Rate (L2) (Bit/s)	Tx Rate (Fps)	Rx (Frames)	Loss (Frames)	Loss Rate (Percent)	BER (est)	FCS Errors (Frames)
Totals												
1518	**PASS**	1	100 %	162,540	199.99 M	197.39 M	16,254	162,540	0	0 %	0	0

Tx (Frames)	Tx Rate (L1) (Bit/s)	Tx Rate (L2) (Bit/s)	Tx Rate (Fps)	Rx (Frames)	Rx Rate (L1) (Bit/s)	Rx Rate (L2) (Bit/s)	Rx Rate (Fps)
Port : P-0-1-2							
81,270	99.99 M	98.69 M	8,127	81,270	99.99 M	98.69 M	8,127

Port : P-0-1-3							
Tx (Frames)	Tx Rate (L1) (Bit/s)	Tx Rate (L2) (Bit/s)	Tx Rate (Fps)	Rx (Frames)	Rx Rate (L1) (Bit/s)	Rx Rate (L2) (Bit/s)	Rx Rate (Fps)
81,270	99.99 M	98.69 M	8,127	81,270	99.99 M	98.69 M	8,127

Figure 23: Test results (totals) for the BB test

DSCP Value Examples

DSCP value	Decimal	Meaning	Drop probability
101 110	46	Expedited forwarding (EF)	N/A
000 000	0	Best effort	N/A
001 010	10	AF11	Low
001 100	12	AF12	Medium
001 110	14	AF13	High
010 010	18	AF21	Low
010 100	20	AF22	Medium
010 110	22	AF23	High
011 010	26	AF31	Low
011 100	28	AF32	Medium
011 110	30	AF33	High
100 010	34	AF41	Low
100 100	36	AF42	Medium
100 110	38	AF43	High

Table 3: DSCP value examples. AF (Assured Forwarding) can be used to split traffic into 4 groups