

APPLICATION NOTE

Concurrent Connection and Connection Per Second Testing

How to do basic CC and CPS testing on a firewall.

Rev 3

Contents

Application Note.....	3
Basic Concepts.....	4
TCP Connection	4
Load Profile.....	4
Connection Update.....	5
TCP CC and CPS Testing	5
1. Create Test Scenario.....	6
2. Configure Load Profile	7
3. Configure Connection Update	7
4. Set TCP Segment Size	8
HTTP CC and CPS Testing.....	9
1. Create Test Scenario.....	10
2. Configure Load Profile	10
3. Configure Connection Update	11
4. Set TCP Segment Size	12
5. Set HTTP Request and Response	13

APPLICATION NOTE

Firewalls are stateful inline network devices. Unlike a switch/router, a firewall caches TCP session state information in a session table and tracks the session until the session ends.

Different firewalls have different capacities for concurrent TCP connections (TCP CC) due to the finite space in the memory, so this is an important test criterion. Establishing a TCP connection is usually costlier than tearing it down because the firewall must register a new entry into the session table. Thus, how fast a firewall can establish TCP connections (TCP CPS) is another important performance index to test. Firewalls are inline devices - they inspect and forward incoming packets to their intended destinations. If a firewall can't process all the traffic it receives, it will be a performance bottleneck or point of failure on the network. Verifying throughput is therefore also critical when it comes to firewall performance testing.

This Application Note describes how to test these three critical performance parameters for a firewall: TCP CC, TCP CPS, and throughput against different packet sizes.

BASIC CONCEPTS

This section describes the basic concepts that are used in a test configuration.

TCP CONNECTION

A TCP connection is defined by 4 parameters: {source IP address, source port number, destination IP address and destination port number}, e.g. 192.168.1.111:49152 \leftrightarrow 21.2.3.40:8080.

Source IP address typically describe the IP address of a client, e.g. 192.168.1.111. Source port number describes the application running on the client, e.g. 49152. Destination IP address is the IP address of the server 21.2.3.40, which the client is establishes TCP connections to. Destination port number, 8080, describes the service on the server that listen from incoming TCP connections. Thus, the number of TCP connections is determined by:

$$N_{TCP} = N_{src_ip} \times N_{src_port} \times N_{dst_ip} \times N_{dst_port}$$

LOAD PROFILE

A *load profile* describes the “shape” of your traffic load. As shown in the figure below, a basic load profile is defined by 4 parameters:

- **Number of concurrent TCP connections**

The number of concurrent TCP connections defines how many TCP connections you want your DUT to maintain at any time during the steady phase. Connections may close and open during the test but the total number of concurrent TCP connections is maintained. This parameter is typically related to the TCP CC testing.

- **Ramp-up duration**

Ramp-up duration defines the time duration for all TCP connections to be established. Increasing or decreasing the ramp-up duration results in a lower or higher connection establish rate. This parameter is typically related to the TCP CPS testing.

- **Steady duration**

Steady duration defines how long you want the converged traffic status to run on your DUT. Throughput is supposed to reach to the highest since all the TCP connections are established and the concurrent number of connections is maintained. This parameter is usually related to soak testing, where stability of the DUT is tested.

- **Ramp-down duration**

Ramp-up duration defines the time duration for all TCP connections to be closed. Increasing or decreasing the ramp-down duration results in a lower or higher connection close rate.

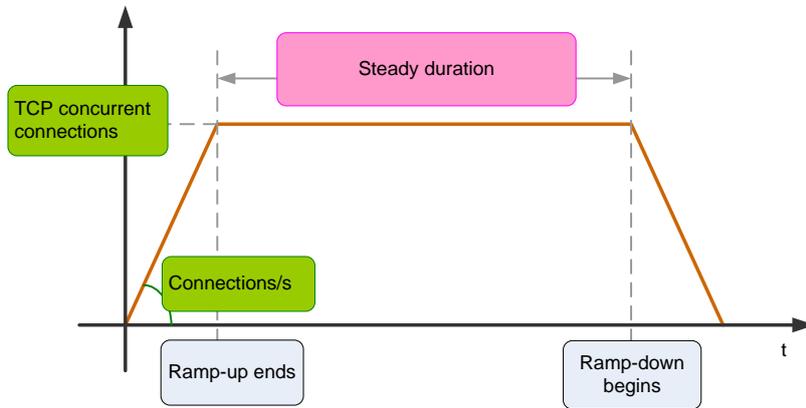


Figure 1. Concept of Load Profile

CONNECTION UPDATE

During the steady phase, connections can open and close at a fixed rate. In order to stabilize the number of concurrent TCP connections, the connection establish rate must equal to connection closure rate, as shown in the figure below.

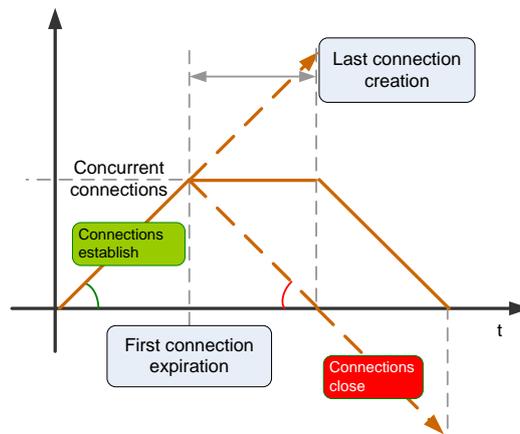


Figure 2. Concept of Connection Update

TCP CC AND CPS TESTING

The scenario to build is shown as in Figure 3. Maximum concurrent TCP connections is 1M. Connections establishment rate is 200K connections/s. Ramp-up duration is 5 seconds. After the ramp-up phase, connections will be created and closed at 200K connections/s, so that the number of concurrent connections does not change during the steady phase. Traffic direction is from servers to clients. In the

ramp-down phase, no more connections are created, and all are closed at 200K connections/s. Packet size is 800 bytes.

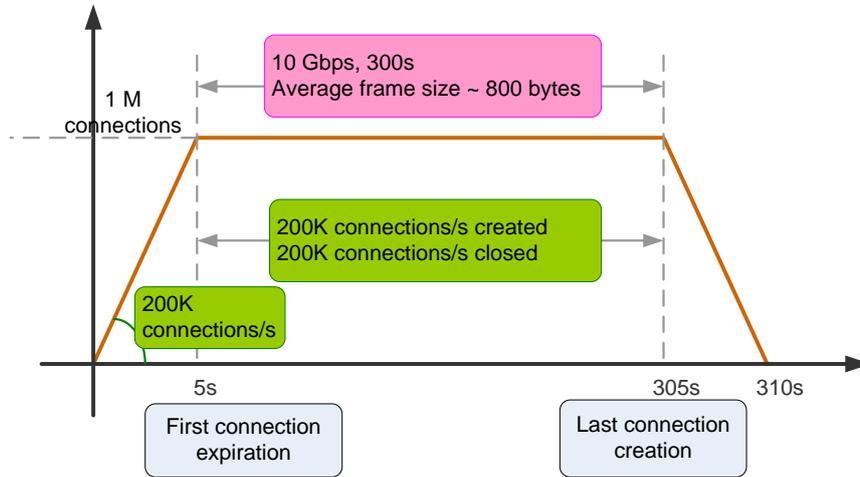


Figure 3. Target Test Scenario

1. CREATE TEST SCENARIO

Create a *RAW* test scenario. Select *No TLS*, *TCP*, and *IPv4* as shown below.

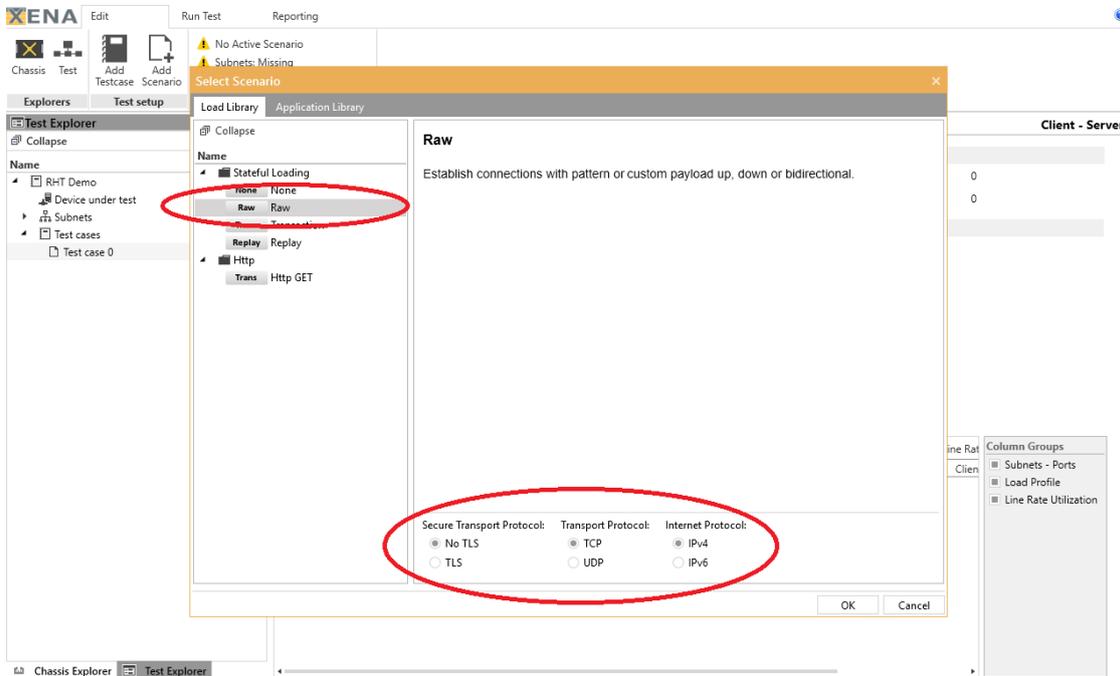


Figure 4. Create RAW Test Scenario

2. CONFIGURE LOAD PROFILE

Expand the test tree in *Test Explorer*, and click *Connection Establishment*.

As shown in Figure 4, change *Number of Source Ports* to 10, and you will see *Total Users: 100,000*, and *Total Connections: 1,000,000*.

Change *Steady* duration to 300 seconds.

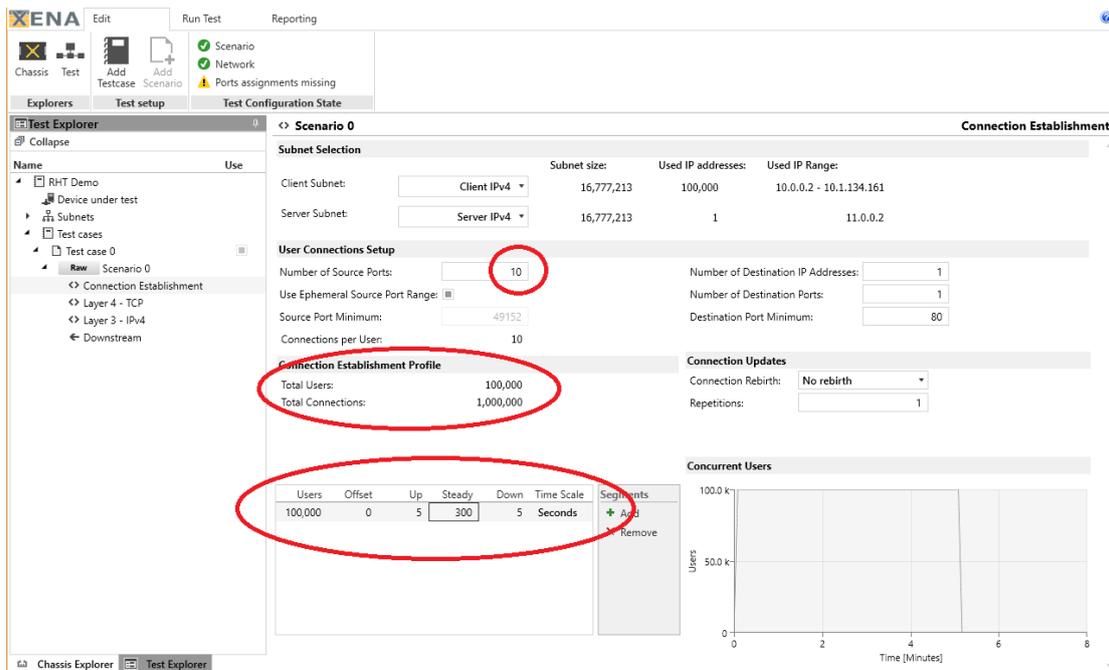


Figure 5. Configure Load Profile

3. CONFIGURE CONNECTION UPDATE

In *Connection Updates* section, change *Connection Rebirth* from *No rebirth* to *With same Src IP*. Delete the value in *Repetitions* or set it to 0 so that the connection opening and closing will continue until the ramp-down phase.

The reason for choosing *With same Src IP* is to avoid IP address depletion during the test. You can also choose *With new Src IP*. If you do so, as soon as a TCP connection is closed, a new connection with a new source IP address will be created. The new source IP address is from the client subnet.

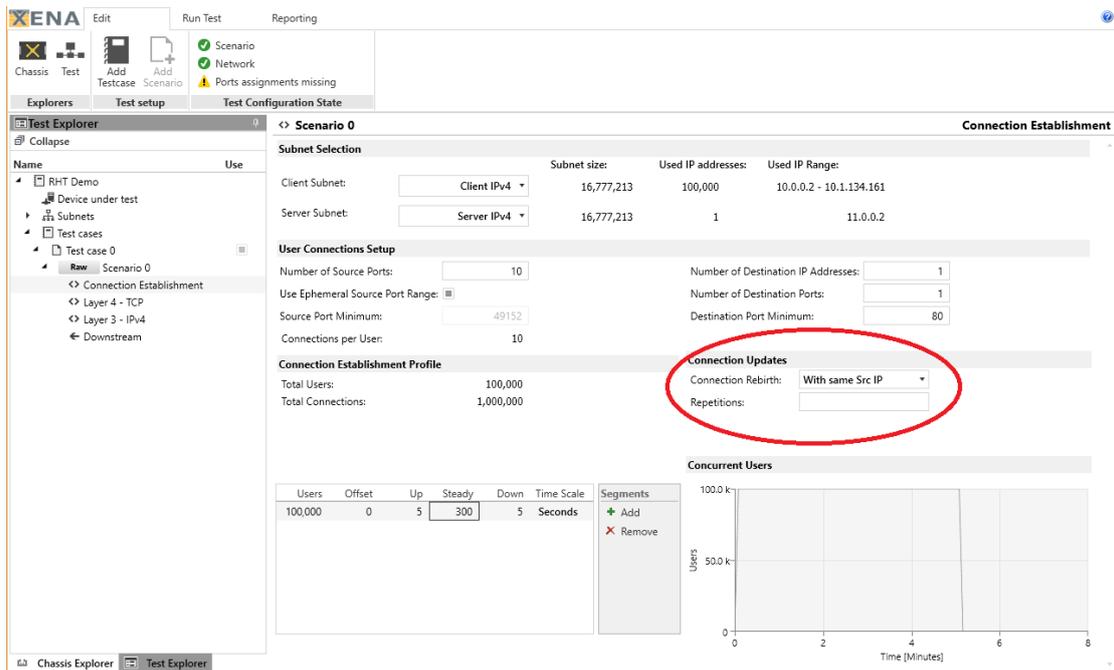


Figure 6. Configure Connection Updates

4. SET TCP SEGMENT SIZE

In *Test Explorer*, click *Layer 4 – TCP*. In *Maximum TCP Segment Size (Client)*, change value to 746 bytes. By doing this, the clients will advertise this value in their TCP SYN packets and the server will adjust the outgoing TCP segment size accordingly.

To generate traffic of 800-byte frame size, the TCP segment size should be 746 bytes, i.e. 800 bytes (frame size) - 14 bytes (Ethernet header) - 20 bytes (IP header) - 20 bytes of (TCP header).

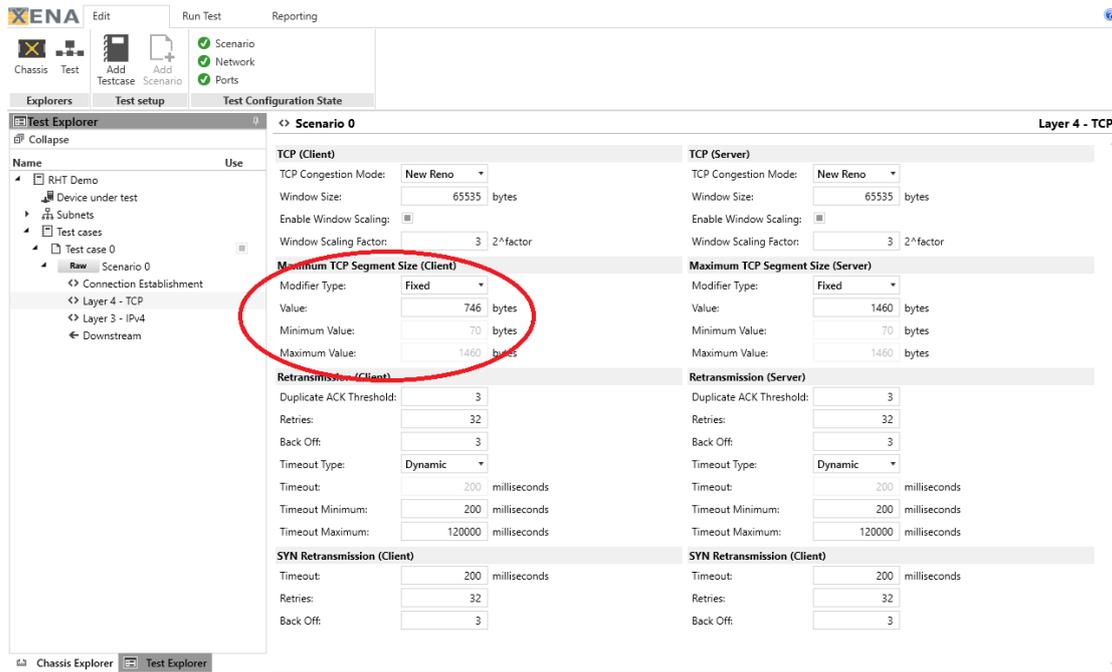


Figure 7. Set TCP Segment Size

HTTP CC AND CPS TESTING

The scenario to build is shown as in Figure 8. Maximum concurrent HTTP connections is 1M. Connections establishment rate is 200K connections/s. Ramp-up duration is 5 seconds. After the ramp-up phase, connections will be created and closed at 200K connections/s, so that the number of concurrent connections does not change during the steady phase. In the ramp-down phase, no more connections are created, and all are closed at 200K connections/s. Packet size is 800 bytes.

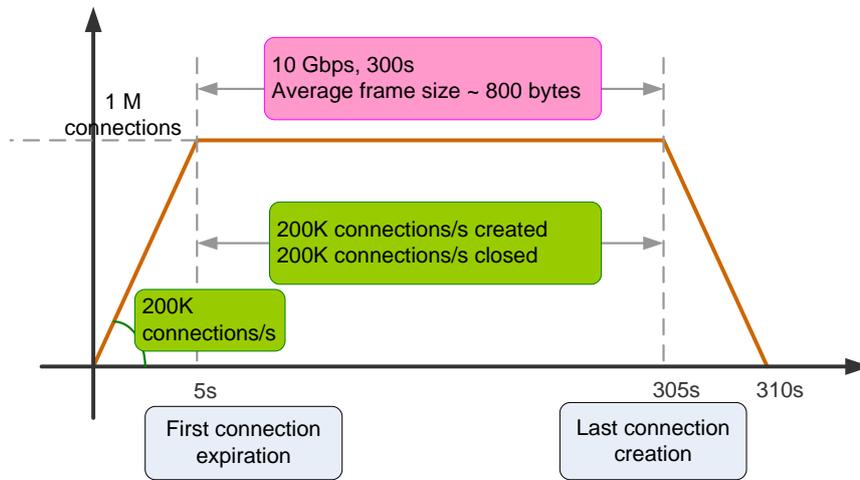


Figure 8. Target Test Scenario

1. CREATE TEST SCENARIO

Create a *HTTP GET* test scenario. Select *No TLS*, *TCP*, and *IPv4* as shown below.

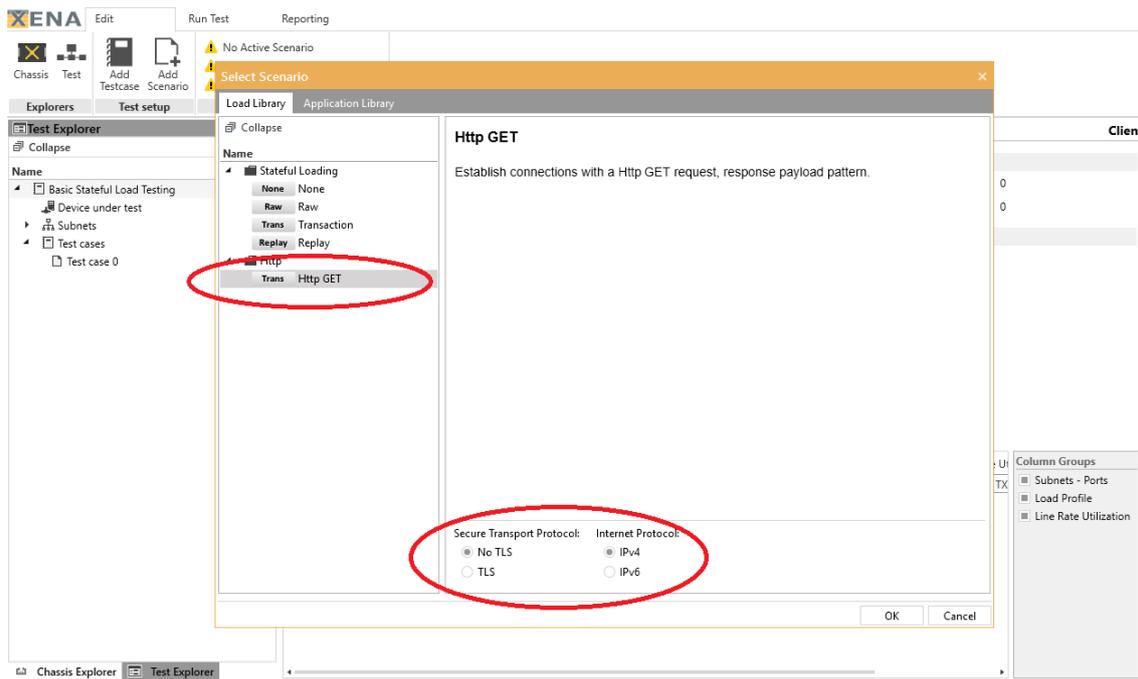


Figure 9. Create HTTP GET Test Scenario

2. CONFIGURE LOAD PROFILE

Expand the test tree in *Test Explorer*, and click *Connection Establishment*.

As shown in Figure 10, change *Number of Source Ports* to 10, and you will see *Total Users: 100,000*, and *Total Connections: 1,000,000*.

Change *Steady* duration to 300 seconds.

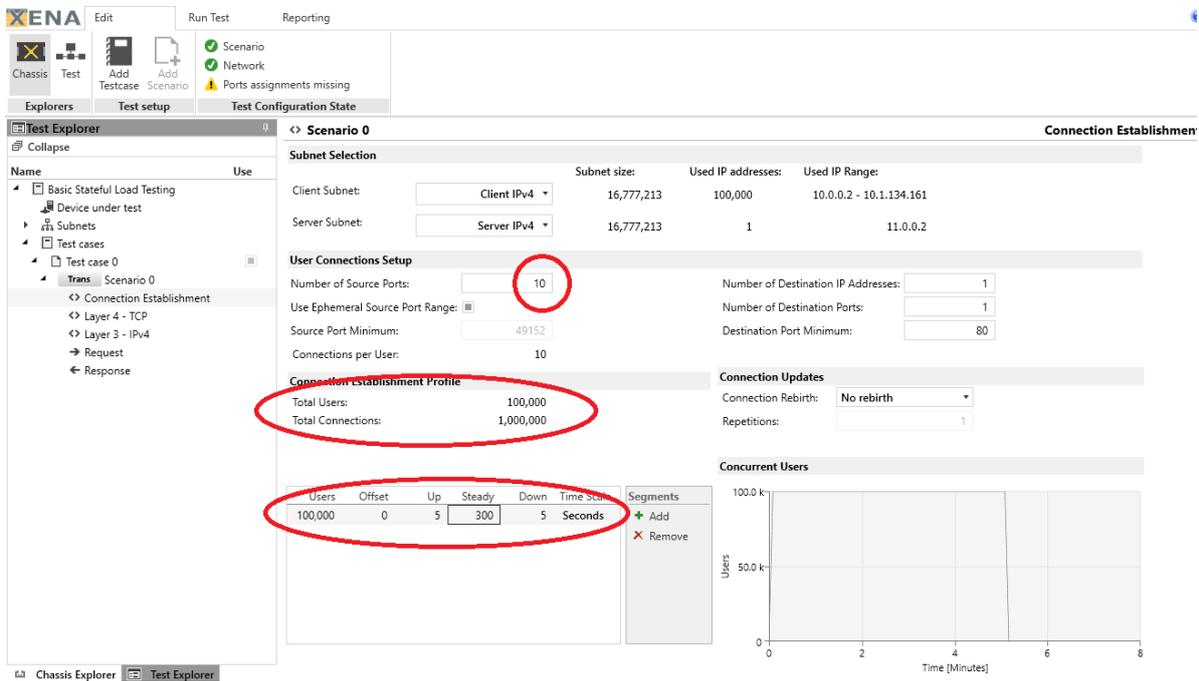


Figure 10. Configure Load Profile

3. CONFIGURE CONNECTION UPDATE

In *Connection Updates* section, change *Connection Rebirth* from *No rebirth* to *With same Src IP*. Delete the value in *Repetitions* or set it to 0 so that the connection opening and closing will continue until the ramp-down phase.

The reason for choosing *With same Src IP* is to avoid IP address depletion during the test. You can also choose *With new Src IP*. If you do so, as soon as a TCP connection is closed, a new connection with a new source IP address will be created. The new source IP address is from the client subnet.

The screenshot shows the XENA Test Explorer interface for Scenario 0. The 'Connection Updates' section is highlighted with a red circle. The 'Connection Rebirth' dropdown is set to 'With same Src IP' and 'Repetitions' is set to 1. Other sections include Subnet Selection, User Connections Setup, and Connection Establishment Profile.

Subnet Selection	Subnet size:	Used IP addresses:	Used IP Range:
Client Subnet: Client IPv4	16,777,213	100,000	10.0.0.2 - 10.1.134.161
Server Subnet: Server IPv4	16,777,213	1	11.0.0.2

User Connections Setup	Number of Source Ports:	Number of Destination IP Addresses:
Use Ephemeral Source Port Range:	10	1
Source Port Minimum: 49152		Number of Destination Ports: 1
Connections per User: 10		Destination Port Minimum: 80

Connection Establishment Profile	Total Users:	Total Connections:
	100,000	1,000,000

Users	Offset	Up	Steady	Down	Time Scale
100,000	0	5	300	5	Seconds

Concurrent Users graph showing Users (0 to 100.0 k) over Time (Minutes) (0 to 8).

Figure 11. Configure Connection Updates

4. SET TCP SEGMENT SIZE

In *Test Explorer*, click *Layer 4 – TCP*. In *Maximum TCP Segment Size (Client)*, change value to 746 bytes. By doing this, the clients will advertise this value in their TCP SYN packets and the server will adjust the outgoing TCP segment size accordingly.

To generate traffic of 800-byte frame size, the TCP segment size should be 746 bytes, i.e. 800 bytes (frame size) - 14 bytes (Ethernet header) - 20 bytes (IP header) - 20 bytes of (TCP header).

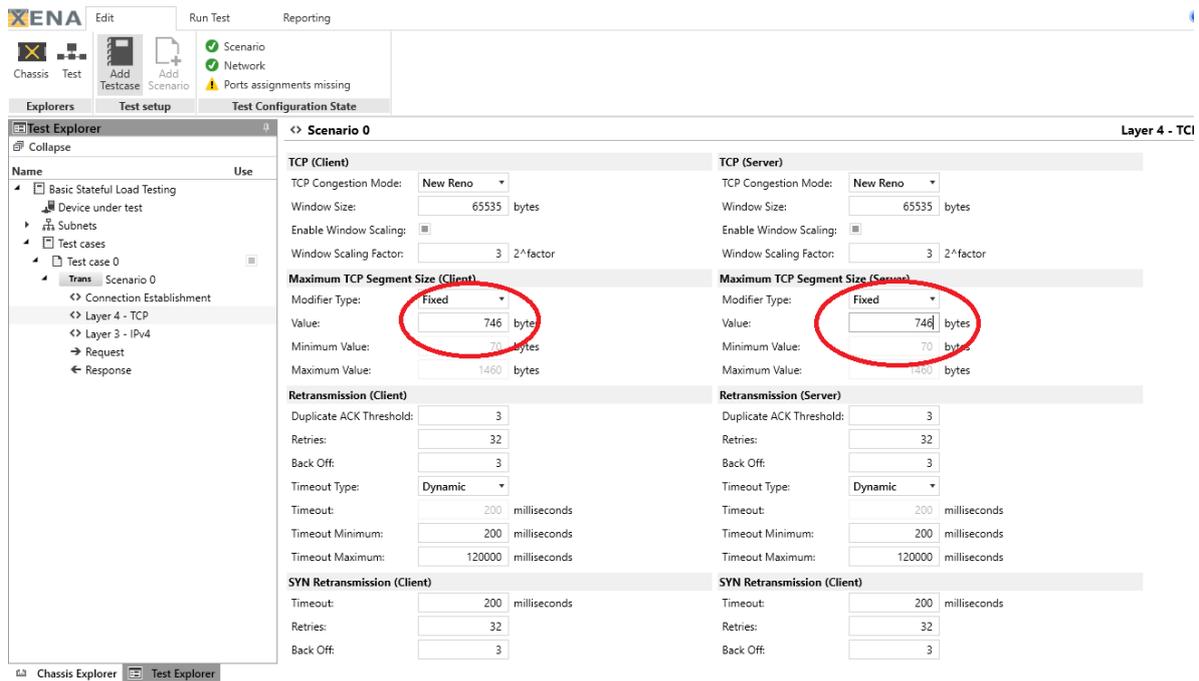


Figure 12. Set TCP Segment Size

5. SET HTTP REQUEST AND RESPONSE

In *Test Explorer*, click *Request* or *Response* to view/modify the HTTP Request or Response. To modify the payload, click *Import* on the right side to import any file. The total length of the payload should be less than 256 KB.

XENA Edit Run Test Reporting

Chassis Test Add Testcase Add Scenario

Scenario Network Ports assignments missing

Explorers Test setup Test Configuration State

Test Explorer → **Request** Request

Basic Stateful Load Testing

- Device under test
- Subnets
- Test cases
 - Test case 0
 - Trans Scenario 0
 - Connection Establishment
 - Layer 4 - TCP
 - Layer 3 - IPv4
 - Request
 - Response

Payload Configuration Pattern total length: 42 bytes

Transmit Setup

- Transmit During Ramp Up:
- Transmit Start Offset: 10 ms
- Transmit During Ramp Down:
- Transmit Stop Offset: 10 ms

Traffic Burst

- Enable Bursty Traffic:
- Traffic Busy Period: 10 ms
- Traffic Inactive Period: 10 ms

Payload Editor

Segment/Field Name	Value	Named Values	Segments
Request			<input type="checkbox"/> Import <input checked="" type="checkbox"/> Remove
.....			
00000000	47 45 54 20 2F 20 48 54 54 50 2F 31 2E 31 0D 0A	GET / HTTP/1.1..	
00000010	48 6F 73 74 3A 20 77 77 77 2E 79 6F 75 72 68 6F	Host: www.yourho	
00000020	73 74 2E 63 6F 6D 0D 0A 0D 0A	st.com...	

Segments

- Import
- Remove

Navigate

- Previous
- Next

XENA Edit Run Test Reporting

Chassis Test Add Testcase Add Scenario

Scenario Network Ports assignments missing

Explorers Test setup Test Configuration State

Test Explorer ← **Response** Response

Basic Stateful Load Testing

- Device under test
- Subnets
- Test cases
 - Test case 0
 - Trans Scenario 0
 - Connection Establishment
 - Layer 4 - TCP
 - Layer 3 - IPv4
 - Request
 - Response

Payload Configuration Pattern total length: 67 bytes

Transmit Setup

- Transmit During Ramp Up:
- Transmit Start Offset: 10 ms
- Transmit During Ramp Down:
- Transmit Stop Offset: 10 ms

Traffic Burst

- Enable Bursty Traffic:
- Traffic Busy Period: 10 ms
- Traffic Inactive Period: 10 ms

Payload Editor

Segment/Field Name	Value	Named Values	Segments
Response			<input type="checkbox"/> Import <input checked="" type="checkbox"/> Remove
.....			
00000000	48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D	HTTP/1.1 200 OK.	
00000010	0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A	.Content-Length:	
00000020	20 31 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65	1..Content-Type	
00000030	3A 20 74 65 78 74 2F 70 6C 61 69 6E 0D 0A 0D 0A	: text/plain...	
00000040	41 0D 0A	A..	

Segments

- Import
- Remove

Navigate

- Previous
- Next

Figure 13. Set HTTP Request and Response