

Benchmarking Methodology Working Group  
Internet-Draft  
Intended status: Informational  
Expires: March 6, 2020

B. Balarajah  
C. Rossenhoevel  
EANTC AG  
B. Monkman  
NetSecOPEN  
September 3, 2019

## **Benchmarking Methodology for Network Security Device Performance draft-ietf-bmwg-ngfw-performance-01**

### Abstract

This document provides benchmarking terminology and methodology for next-generation network security devices including next-generation firewalls (NGFW), intrusion detection and prevention solutions (IDS/IPS) and unified threat management (UTM) implementations. This document aims to strongly improve the applicability, reproducibility, and transparency of benchmarks and to align the test methodology with today's increasingly complex layer 7 application use cases. The main areas covered in this document are test terminology, traffic profiles and benchmarking methodology for NGFWs to start with.

### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 6, 2020.

### Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                        |  |                    |
|------------------------|--|--------------------|
| <a href="#">1.</a>     | <a href="#">Introduction</a>                                       | <a href="#">3</a>  |
| <a href="#">2.</a>     | <a href="#">Requirements</a>                                       | <a href="#">4</a>  |
| <a href="#">3.</a>     | <a href="#">Scope</a>  | <a href="#">4</a>  |
| <a href="#">4.</a>     | <a href="#">Test Setup</a>   | <a href="#">4</a>  |
| <a href="#">4.1.</a>   | <a href="#">Testbed Configuration</a>                              | <a href="#">4</a>  |
| <a href="#">4.2.</a>   | <a href="#">DUT/SUT Configuration</a>                              | <a href="#">5</a>  |
| <a href="#">4.3.</a>   | <a href="#">Test Equipment Configuration</a>                       | <a href="#">9</a>  |
| <a href="#">4.3.1.</a> | <a href="#">Client Configuration</a>                               | <a href="#">9</a>  |
| <a href="#">4.3.2.</a> | <a href="#">Backend Server Configuration</a>                       | <a href="#">11</a> |
| <a href="#">4.3.3.</a> | <a href="#">Traffic Flow Definition</a>                            | <a href="#">11</a> |
| <a href="#">4.3.4.</a> | <a href="#">Traffic Load Profile</a>                               | <a href="#">12</a> |
| <a href="#">5.</a>     | <a href="#">Test Bed Considerations</a>                            | <a href="#">13</a> |
| <a href="#">6.</a>     | <a href="#">Reporting</a>  | <a href="#">14</a> |
| <a href="#">6.1.</a>   | <a href="#">Key Performance Indicators</a>                         | <a href="#">15</a> |
| <a href="#">7.</a>     | <a href="#">Benchmarking Tests</a>                                 | <a href="#">16</a> |
| <a href="#">7.1.</a>   | <a href="#">Throughput Performance With NetSecOPEN Traffic Mix</a> | <a href="#">17</a> |
| <a href="#">7.1.1.</a> | <a href="#">Objective</a>  | <a href="#">17</a> |
| <a href="#">7.1.2.</a> | <a href="#">Test Setup</a>   | <a href="#">17</a> |
| <a href="#">7.1.3.</a> | <a href="#">Test Parameters</a>                                    | <a href="#">17</a> |
| <a href="#">7.1.4.</a> | <a href="#">Test Procedures and expected Results</a>               | <a href="#">19</a> |
| <a href="#">7.2.</a>   | <a href="#">TCP/HTTP Connections Per Second</a>                    | <a href="#">20</a> |
| <a href="#">7.2.1.</a> | <a href="#">Objective</a>  | <a href="#">20</a> |
| <a href="#">7.2.2.</a> | <a href="#">Test Setup</a>   | <a href="#">20</a> |
| <a href="#">7.2.3.</a> | <a href="#">Test Parameters</a>                                    | <a href="#">20</a> |
| <a href="#">7.2.4.</a> | <a href="#">Test Procedures and Expected Results</a>               | <a href="#">22</a> |
| <a href="#">7.3.</a>   | <a href="#">HTTP Throughput</a>                                    | <a href="#">23</a> |
| <a href="#">7.3.1.</a> | <a href="#">Objective</a>  | <a href="#">23</a> |
| <a href="#">7.3.2.</a> | <a href="#">Test Setup</a>   | <a href="#">23</a> |
| <a href="#">7.3.3.</a> | <a href="#">Test Parameters</a>                                    | <a href="#">23</a> |
| <a href="#">7.3.4.</a> | <a href="#">Test Procedures and Expected Results</a>               | <a href="#">25</a> |
| <a href="#">7.4.</a>   | <a href="#">TCP/HTTP Transaction Latency</a>                       | <a href="#">26</a> |
| <a href="#">7.4.1.</a> | <a href="#">Objective</a>  | <a href="#">26</a> |
| <a href="#">7.4.2.</a> | <a href="#">Test Setup</a>   | <a href="#">26</a> |
| <a href="#">7.4.3.</a> | <a href="#">Test Parameters</a>                                    | <a href="#">26</a> |
| <a href="#">7.4.4.</a> | <a href="#">Test Procedures and Expected Results</a>               | <a href="#">28</a> |
| <a href="#">7.5.</a>   | <a href="#">Concurrent TCP/HTTP Connection Capacity</a>            | <a href="#">29</a> |
| <a href="#">7.5.1.</a> | <a href="#">Objective</a>  | <a href="#">29</a> |
| <a href="#">7.5.2.</a> | <a href="#">Test Setup</a>   | <a href="#">30</a> |



|                    |  |    |
|--------------------|--|----|
| 7.5.3.             | Test Parameters . . . . .                          | 30 |
| 7.5.4.             | Test Procedures and expected Results . . . . .     | 31 |
| 7.6.               | TCP/HTTPS Connections per second . . . . .         | 33 |
| 7.6.1.             | Objective . . . . .                                | 33 |
| 7.6.2.             | Test Setup . . . . .                               | 33 |
| 7.6.3.             | Test Parameters . . . . .                          | 33 |
| 7.6.4.             | Test Procedures and expected Results . . . . .     | 35 |
| 7.7.               | HTTPS Throughput . . . . .                         | 36 |
| 7.7.1.             | Objective . . . . .                                | 36 |
| 7.7.2.             | Test Setup . . . . .                               | 36 |
| 7.7.3.             | Test Parameters . . . . .                          | 36 |
| 7.7.4.             | Test Procedures and Expected Results . . . . .     | 39 |
| 7.8.               | HTTPS Transaction Latency . . . . .                | 40 |
| 7.8.1.             | Objective . . . . .                                | 40 |
| 7.8.2.             | Test Setup . . . . .                               | 40 |
| 7.8.3.             | Test Parameters . . . . .                          | 40 |
| 7.8.4.             | Test Procedures and Expected Results . . . . .     | 42 |
| 7.9.               | Concurrent TCP/HTTPS Connection Capacity . . . . . | 43 |
| 7.9.1.             | Objective . . . . .                                | 43 |
| 7.9.2.             | Test Setup . . . . .                               | 43 |
| 7.9.3.             | Test Parameters . . . . .                          | 43 |
| 7.9.4.             | Test Procedures and expected Results . . . . .     | 45 |
| 8.                 | Formal Syntax . . . . .                            | 46 |
| 9.                 | IANA Considerations . . . . .                      | 46 |
| 10.                | Security Considerations . . . . .                  | 46 |
| 11.                | Acknowledgements . . . . .                         | 47 |
| 12.                | Contributors . . . . .                             | 47 |
| 13.                | References . . . . .                               | 47 |
| 13.1.              | Normative References . . . . .                     | 47 |
| 13.2.              | Informative References . . . . .                   | 47 |
| Appendix A.        | NetSecOPEN Basic Traffic Mix . . . . .             | 48 |
| Authors' Addresses | . . . . .  | 56 |

## **1. Introduction**

15 years have passed since IETF recommended test methodology and terminology for firewalls initially ([RFC2647], [RFC3511]). The requirements for network security element performance and effectiveness have increased tremendously since then. Security function implementations have evolved to more advanced areas and have diversified into intrusion detection and prevention, threat management, analysis of encrypted traffic, etc. In an industry of growing importance, well-defined and reproducible key performance indicators (KPIs) are increasingly needed: They enable fair and reasonable comparison of network security functions. All these reasons have led to the creation of a new next-generation firewall benchmarking document.



## **2. Requirements**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14 \[RFC2119\]](#), [\[RFC8174\]](#) when, and only when, they appear in all capitals, as shown here.

## **3. Scope**

This document provides testing terminology and testing methodology for next-generation firewalls and related security functions. It covers two main areas: security effectiveness configurations, followed by performance benchmark testing. This document focuses on advanced, realistic, and reproducible testing methods. Additionally, it describes test bed environments, test tool requirements and test result formats.

## **4. Test Setup**

Test setup defined in this document is applicable to all benchmarking test scenarios described in [Section 7](#).

### **4.1. Testbed Configuration**

Testbed configuration MUST ensure that any performance implications that are discovered during the benchmark testing aren't due to the inherent physical network limitations such as number of physical links and forwarding performance capabilities (throughput and latency) of the network device in the testbed. For this reason, this document recommends avoiding external devices such as switches and routers in the testbed wherever possible.

However, in the typical deployment, the security devices ( Device Under Test/System Under Test) are connected to routers and switches which will reduce the number of entries in MAC or ARP tables of the Device Under Test/System Under Test (DUT/SUT). If MAC or ARP tables have many entries, this may impact the actual DUT/SUT performance due to MAC and ARP/ND table lookup processes. Therefore, it is RECOMMENDED to connect aggregation switches or routers between test equipment and DUT/SUT as shown in Figure 1. The aggregation switches or routers can be also used to aggregate the test equipment or DUT/SUT ports, if the numbers of used ports are mismatched between test equipment and DUT/SUT.

If the test equipment is capable of emulating layer 3 routing functionality and there is no need for test equipment port



aggregation, it is RECOMMENDED to configure the test setup as shown in Figure 2.

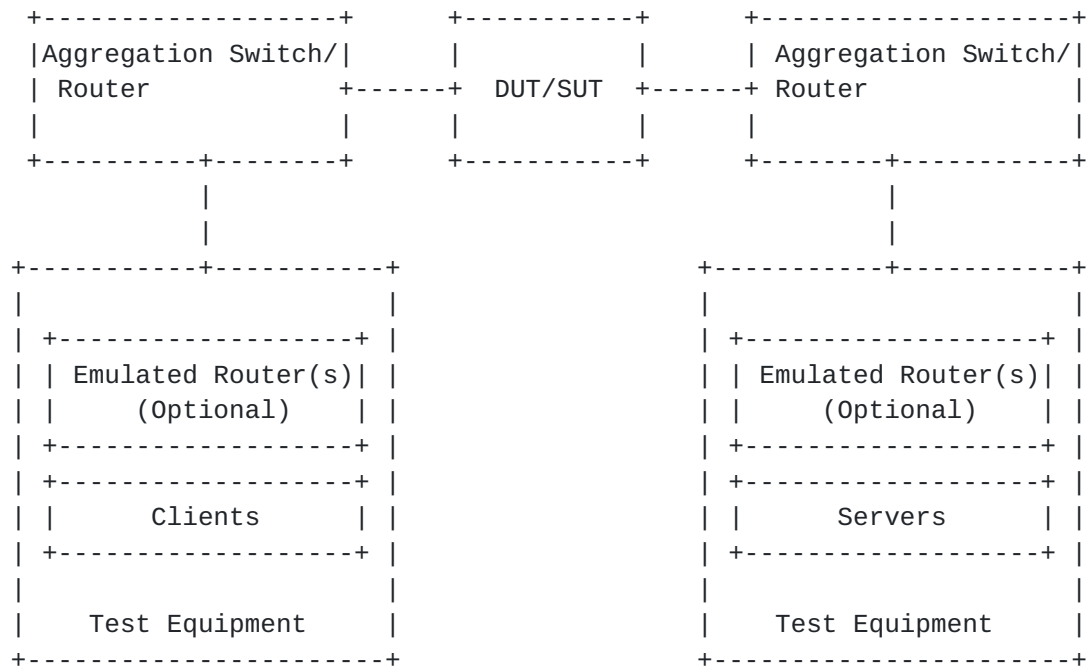


Figure 1: Testbed Setup - Option 1

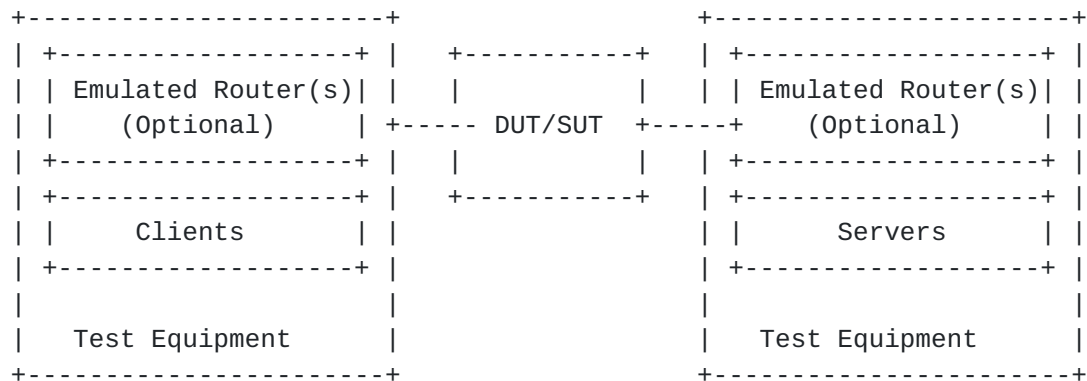


Figure 2: Testbed Setup - Option 2

#### 4.2. DUT/SUT Configuration

A unique DUT/SUT configuration MUST be used for all benchmarking tests described in [Section 7](#). Since each DUT/SUT will have their own unique configuration, users SHOULD configure their device with the same parameters and security features that would be used in the actual deployment of the device or a typical deployment in order to achieve maximum security coverage.



This document attempts to define the recommended security features which SHOULD be consistently enabled for all the benchmarking tests described in [Section 7](#). Table 1 below describes the RECOMMENDED sets of feature list which SHOULD be configured on the DUT/SUT.

Based on customer use case, users MAY enable or disable SSL inspection feature for "Throughput Performance with NetSecOPEN Traffic Mix" test scenario described in [Section 7.1](#)

To improve repeatability, a summary of the DUT configuration including description of all enabled DUT/SUT features MUST be published with the benchmarking results.

| +-----+<br>  NGFW  <br>+-----+  |             |          |
|---------------------------------|-------------|----------|
| +-----+                         | +-----+     | +-----+  |
| DUT Features                    | RECOMMENDED | OPTIONAL |
| +-----+                         | +-----+     | +-----+  |
| SSL Inspection                  | x           |          |
| +-----+                         | +-----+     | +-----+  |
| IDS/IPS                         | x           |          |
| +-----+                         | +-----+     | +-----+  |
| Web Filtering                   |             | x        |
| +-----+                         | +-----+     | +-----+  |
| Antivirus                       | x           |          |
| +-----+                         | +-----+     | +-----+  |
| Anti Spyware                    | x           |          |
| +-----+                         | +-----+     | +-----+  |
| Anti Botnet                     | x           |          |
| +-----+                         | +-----+     | +-----+  |
| DLP                             |             | x        |
| +-----+                         | +-----+     | +-----+  |
| DDoS                            |             | x        |
| +-----+                         | +-----+     | +-----+  |
| Certificate<br>  Validation     |             | x        |
| +-----+                         | +-----+     | +-----+  |
| Logging and<br>  Reporting      | x           |          |
| +-----+                         | +-----+     | +-----+  |
| Application<br>  Identification | x           |          |
| +-----+                         | +-----+     | +-----+  |

Table 1: DUT/SUT Feature List



In summary, DUT/SUT SHOULD be configured as follows:

- o All security inspection enabled
- o Disposition of all traffic is logged - Logging to an external device is permissible
- o Detection of Common Vulnerabilities and Exposures (CVE) matching the following characteristics when searching the National Vulnerability Database (NVD)
  - \* Common Vulnerability Scoring System (CVSS) Version: 2
  - \* CVSS V2 Metrics: AV:N/Au:N/I:C/A:C
  - \* AV=Attack Vector, Au=Authentication, I=Integrity and A=Availability
  - \* CVSS V2 Severity: High (7-10)
  - \* If doing a group test the published start date and published end date SHOULD be the same
- o Geographical location filtering and Application Identification and Control configured to be triggered based on a site or application from the defined traffic mix

In addition, it is also RECOMMENDED to configure a realistic number of access policy rules on the DUT/SUT. This document determines the number of access policy rules for three different classes of DUT/SUT. The classification of the DUT/SUT MAY be based on its maximum supported firewall throughput performance number defined in the vendor data sheet. This document classifies the DUT/SUT in four different categories; namely Extra Small, Small, Medium, and Large.

The RECOMMENDED throughput values for the following classes are:

Extra Small (XS) - supported throughput less than 1Gbit/s

Small (S) - supported throughput less than 5Gbit/s

Medium (M) - supported throughput greater than 5Gbit/s and less than 10Gbit/s

Large (L) - supported throughput greater than 10Gbit/s



The Access Control Rules (ACL) defined in Table 2 SHOULD be configured from top to bottom in the correct order as shown in the table.

(Note: There will be differences between how security vendors implement ACL decision making.) The configured ACL MUST NOT block the test traffic used for the benchmarking test scenarios.

|                   |                              |   |        | DUT/SUT        |    |     |     |  |
|-------------------|------------------------------|---|--------|----------------|----|-----|-----|--|
|                   |                              |   |        | Classification |    |     |     |  |
|                   |                              |   |        | #rules         |    |     |     |  |
| Rules Type        | Match Criteria               | Description   | Action | XS             | S  | M   | L   |  |
| Application layer | Application                  | Any application traffic NOT included in the test traffic  | block  | 5              | 10 | 20  | 50  |  |
| Transport layer   | Src IP and TCP/UDP Dst ports | Any src IP subnet used in the test AND any dst ports NOT used in the test traffic                   | block  | 25             | 50 | 100 | 250 |  |
| IP layer          | Src/Dst IP                   | Any src/dst IP subnet NOT used in the test  | block  | 25             | 50 | 100 | 250 |  |
| Application layer | Application                  | Applications included in the test traffic   | allow  | 10             | 10 | 10  | 10  |  |
| Transport layer   | Src IP and TCP/UDP Dst ports | Half of the src IP used in the test AND any dst ports used in the test traffic. One rule per subnet | allow  | 1              | 1  | 1   | 1   |  |
| IP layer          | Src IP                       | The rest of the src IP subnet range used in the test. One rule per subnet                           | allow  | 1              | 1  | 1   | 1   |  |

Table 2: DUT/SUT Access List



### **4.3. Test Equipment Configuration**

In general, test equipment allows configuring parameters in different protocol layers. These parameters thereby influence the traffic flows which will be offered and impact performance measurements.

This document specifies common test equipment configuration parameters applicable for all test scenarios defined in [Section 7](#). Any test scenario specific parameters are described under the test setup section of each test scenario individually.

#### **4.3.1. Client Configuration**

This section specifies which parameters SHOULD be considered while configuring clients using test equipment. Also, this section specifies the recommended values for certain parameters.

##### **4.3.1.1. TCP Stack Attributes**

The TCP stack SHOULD use a TCP Reno [[RFC5681](#)] variant, which include congestion avoidance, back off and windowing, fast retransmission, and fast recovery on every TCP connection between client and server endpoints. The default IPv4 and IPv6 MSS segments size MUST be set to 1460 bytes and 1440 bytes respectively and a TX and RX receive windows of 64 KByte. Client initial congestion window MUST NOT exceed 10 times the MSS. Delayed ACKs are permitted and the maximum client delayed Ack MUST NOT exceed 10 times the MSS before a forced ACK. Up to 3 retries SHOULD be allowed before a timeout event is declared. All traffic MUST set the TCP PSH flag to high. The source port range SHOULD be in the range of 1024 - 65535. Internal timeout SHOULD be dynamically scalable per [RFC 793](#). Client SHOULD initiate and close TCP connections. TCP connections MUST be closed via FIN.

##### **4.3.1.2. Client IP Address Space**

The sum of the client IP space SHOULD contain the following attributes. The traffic blocks SHOULD consist of multiple unique, discontinuous static address blocks. A default gateway is permitted. The IPv4 ToS byte or IPv6 traffic class should be set to '00' or '000000' respectively.

The following equation can be used to determine the required total number of client IP addresses.

Desired total number of client IP = Target throughput [Mbit/s] /  
Throughput per IP address [Mbit/s]



Based on deployment and use case scenario, the value for "Throughput per IP address" can be varied.

(Option 1) DUT/SUT deployment scenario 1 : 6-7 Mbit/s per IP (e.g. 1,400-1,700 IPs per 10Gbit/s throughput)

(Option 2) DUT/SUT deployment scenario 2 : 0.1-0.2 Mbit/s per IP (e.g. 50,000-100,000 IPs per 10Gbit/s throughput)

Based on deployment and use case scenario, client IP addresses SHOULD be distributed between IPv4 and IPv6 type. The Following options can be considered for a selection of traffic mix ratio.

(Option 1) 100 % IPv4, no IPv6

(Option 2) 80 % IPv4, 20% IPv6

(Option 3) 50 % IPv4, 50% IPv6

(Option 4) 20 % IPv4, 80% IPv6

(Option 5) no IPv4, 100% IPv6

#### **4.3.1.3. Emulated Web Browser Attributes**

The emulated web browser contains attributes that will materially affect how traffic is loaded. The objective is to emulate modern, typical browser attributes to improve realism of the result set.

For HTTP traffic emulation, the emulated browser MUST negotiate HTTP 1.1. HTTP persistency MAY be enabled depending on test scenario. The browser MAY open multiple TCP connections per Server endpoint IP at any time depending on how many sequential transactions are needed to be processed. Within the TCP connection multiple transactions MAY be processed if the emulated browser has available connections. The browser SHOULD advertise a User-Agent header. Headers MUST be sent uncompressed. The browser SHOULD enforce content length validation.

For encrypted traffic, the following attributes SHALL define the negotiated encryption parameters. The test clients MUST use TLSv1.2 or higher. TLS record size MAY be optimized for the HTTPS response object size up to a record size of 16 KByte. The client endpoint MUST send TLS Extension Server Name Indication (SNI) information when opening a security tunnel. Each client connection MUST perform a full handshake with server certificate and MUST NOT use session reuse or resumption. Cipher suite and key size should be defined in the parameter session of each test scenario.



#### **4.3.2. Backend Server Configuration**

This document specifies which parameters should be considered while configuring emulated backend servers using test equipment.

##### **4.3.2.1. TCP Stack Attributes**

The TCP stack on the server side SHOULD be configured similar to the client side configuration described in [Section 4.3.1.1](#). In addition, server initial congestion window MUST NOT exceed 10 times the MSS. Delayed ACKs are permitted and the maximum server delayed ACK MUST NOT exceed 10 times the MSS before a forced ACK.

##### **4.3.2.2. Server Endpoint IP Addressing**

The server IP blocks SHOULD consist of unique, discontinuous static address blocks with one IP per Server Fully Qualified Domain Name (FQDN) endpoint per test port. The IPv4 ToS byte and IPv6 traffic class bytes should be set to '00' and '000000' respectively.

##### **4.3.2.3. HTTP / HTTPS Server Pool Endpoint Attributes**

The server pool for HTTP SHOULD listen on TCP port 80 and emulate HTTP version 1.1 with persistence. The Server MUST advertise server type in the Server response header [[RFC2616](#)]. For HTTPS server, TLS 1.2 or higher MUST be used with a maximum record size of 16 KByte and MUST NOT use ticket resumption or Session ID reuse. The server MUST listen on port TCP 443. The server SHALL serve a certificate to the client. It is REQUIRED that the HTTPS server also check Host SNI information with the FQDN. Cipher suite and key size should be defined in the parameter section of each test scenario.

##### **4.3.3. Traffic Flow Definition**

This section describes the traffic pattern between client and server endpoints. At the beginning of the test, the server endpoint initializes and will be ready to accept connection states including initialization of the TCP stack as well as bound HTTP and HTTPS servers. When a client endpoint is needed, it will initialize and be given attributes such as a MAC and IP address. The behavior of the client is to sweep through the given server IP space, sequentially generating a recognizable service by the DUT. Thus, a balanced, mesh between client endpoints and server endpoints will be generated in a client port server port combination. Each client endpoint performs the same actions as other endpoints, with the difference being the source IP of the client endpoint and the target server IP pool. The client SHALL use Fully Qualified Domain Names (FQDN) in Host Headers and for TLS Server Name Indication (SNI).



#### **4.3.3.1. Description of Intra-Client Behavior**

Client endpoints are independent of other clients that are concurrently executing. When a client endpoint initiates traffic, this section describes how the client steps through different services. Once the test is initialized, the client endpoints SHOULD randomly hold (perform no operation) for a few milliseconds to allow for better randomization of start of client traffic. Each client will either open a new TCP connection or connect to a TCP persistence stack still open to that specific server. At any point that the service profile may require encryption, a TLS encryption tunnel will form presenting the URL request to the server. The server will then perform an SNI name check with the proposed FQDN compared to the domain embedded in the certificate. Only when correct, will the server process the HTTPS response object. The initial response object to the server MUST NOT have a fixed size; its size is based on benchmarking tests described in [Section 7](#). Multiple additional sub-URLs (response objects on the service page) MAY be requested simultaneously. This MAY be to the same server IP as the initial URL. Each sub-object will also use a conical FQDN and URL path, as observed in the traffic mix used.

#### **4.3.4. Traffic Load Profile**

The loading of traffic is described in this section. The loading of a traffic load profile has five distinct phases: Init, ramp up, sustain, ramp down, and collection.

1. During the Init phase, test bed devices including the client and server endpoints should negotiate layer 2-3 connectivity such as MAC learning and ARP. Only after successful MAC learning or ARP/ND resolution SHALL the test iteration move to the next phase. No measurements are made in this phase. The minimum RECOMMEND time for Init phase is 5 seconds. During this phase, the emulated clients SHOULD NOT initiate any sessions with the DUT/SUT, in contrast, the emulated servers should be ready to accept requests from DUT/SUT or from emulated clients.
2. In the ramp up phase, the test equipment SHOULD start to generate the test traffic. It SHOULD use a set approximate number of unique client IP addresses actively to generate traffic. The traffic SHOULD ramp from zero to desired target objective. The target objective will be defined for each benchmarking test. The duration for the ramp up phase MUST be configured long enough, so that the test equipment does not overwhelm DUT/SUT's supported performance metrics namely; connections per second, concurrent TCP connections, and application transactions per second. The



RECOMMENDED time duration for the ramp up phase is 180-300 seconds. No measurements are made in this phase.

3. In the sustain phase, the test equipment SHOULD continue generating traffic to constant target value for a constant number of active client IPs. The RECOMMENDED time duration for sustain phase is 600 seconds. This is the phase where measurements occur.
4. In the ramp down/close phase, no new connections are established, and no measurements are made. The time duration for ramp up and ramp down phase SHOULD be same. The RECOMMENDED duration of this phase is between 180 to 300 seconds.
5. The last phase is administrative and will occur when the test equipment merges and collates the report data.

## **5. Test Bed Considerations**

This section recommends steps to control the test environment and test equipment, specifically focusing on virtualized environments and virtualized test equipment.

1. Ensure that any ancillary switching or routing functions between the system under test and the test equipment do not limit the performance of the traffic generator. This is specifically important for virtualized components (vSwitches, vRouters).
2. Verify that the performance of the test equipment matches and reasonably exceeds the expected maximum performance of the system under test.
3. Assert that the test bed characteristics are stable during the entire test session. Several factors might influence stability specifically for virtualized test beds, for example additional workloads in a virtualized system, load balancing and movement of virtual machines during the test, or simple issues such as additional heat created by high workloads leading to an emergency CPU performance reduction.

Test bed reference pre-tests help to ensure that the maximum desired traffic generator aspects such as throughput, transaction per second, connection per second, concurrent connection and latency.

Once the desired maximum performance goals for the system under test have been identified, a safety margin of 10% SHOULD be added for throughput and subtracted for maximum latency and maximum packet loss.



Test bed preparation may be performed either by configuring the DUT in the most trivial setup (fast forwarding) or without presence of DUT.

## 6. Reporting

This section describes how the final report should be formatted and presented. The final test report MAY have two major sections; Introduction and result sections. The following attributes SHOULD be present in the introduction section of the test report.

1. The name of the NetSecOPEN traffic mix (see [Appendix A](#)) MUST be prominent.
2. The time and date of the execution of the test MUST be prominent.
3. Summary of testbed software and Hardware details
  - A. DUT Hardware/Virtual Configuration
    - + This section SHOULD clearly identify the make and model of the DUT
    - + The port interfaces, including speed and link information MUST be documented.
    - + If the DUT is a virtual VNF, interface acceleration such as DPDK and SR-IOV MUST be documented as well as cores used, RAM used, and the pinning / resource sharing configuration. The Hypervisor and version MUST be documented.
    - + Any additional hardware relevant to the DUT such as controllers MUST be documented
  - B. DUT Software
    - + The operating system name MUST be documented
    - + The version MUST be documented
    - + The specific configuration MUST be documented
  - C. DUT Enabled Features
    - + Configured DUT/SUT features (see Table 1) MUST be documented



- + Attributes of those featured MUST be documented
  - + Any additional relevant information about features MUST be documented
- D. Test equipment hardware and software
- + Test equipment vendor name
  - + Hardware details including model number, interface type
  - + Test equipment firmware and test application software version
4. Results Summary / Executive Summary
1. Results SHOULD resemble a pyramid in how it is reported, with the introduction section documenting the summary of results in a prominent, easy to read block.
  2. In the result section of the test report, the following attributes should be present for each test scenario.
    - a. KPIs MUST be documented separately for each test scenario. The format of the KPI metrics should be presented as described in [Section 6.1](#).
    - b. The next level of details SHOULD be graphs showing each of these metrics over the duration (sustain phase) of the test. This allows the user to see the measured performance stability changes over time.

### **[6.1](#). Key Performance Indicators**

This section lists KPIs for overall benchmarking tests scenarios. All KPIs MUST be measured during the sustain phase of the traffic load profile described in [Section 4.3.4](#). All KPIs MUST be measured from the result output of test equipment.

- o Concurrent TCP Connections  
This key performance indicator measures the average concurrent open TCP connections in the sustaining period.
- o TCP Connections Per Second  
This key performance indicator measures the average established TCP connections per second in the sustaining period. For "TCP/HTTP(S) Connection Per Second" benchmarking test scenario, the KPI



is measured average established and terminated TCP connections per second simultaneously.

- o Application Transactions Per Second  
This key performance indicator measures the average successfully completed application transactions per second in the sustaining period.
- o TLS Handshake Rate  
This key performance indicator measures the average TLS 1.2 or higher session formation rate within the sustaining period.
- o Throughput  
This key performance indicator measures the average Layer 2 throughput within the sustaining period as well as average packets per seconds within the same period. The value of throughput SHOULD be presented in Gbit/s rounded to two places of precision with a more specific Kbit/s in parenthesis. Optionally, goodput MAY also be logged as an average goodput rate measured over the same period. Goodput result SHALL also be presented in the same format as throughput.
- o URL Response time / Time to Last Byte (TTLB)  
This key performance indicator measures the minimum, average and maximum per URL response time in the sustaining period. The latency is measured at Client and in this case would be the time duration between sending a GET request from Client and the receipt of the complete response from the server.
- o Application Transaction Latency  
This key performance indicator measures the minimum, average and maximum the amount of time to receive all objects from the server. The value of application transaction latency SHOULD be presented in millisecond rounded to zero decimal.
- o Time to First Byte (TTFB)  
This key performance indicator will measure minimum, average and maximum the time to first byte. TTFB is the elapsed time between sending the SYN packet from the client and receiving the first byte of application data from the DUT/SUT. TTFB SHOULD be expressed in millisecond.

## [7.](#) Benchmarking Tests



## **7.1. Throughput Performance With NetSecOPEN Traffic Mix**

### **7.1.1. Objective**

Using NetSecOPEN traffic mix, determine the maximum sustainable throughput performance supported by the DUT/SUT. (see [Appendix A](#) for details about traffic mix)

This test scenario is RECOMMENDED to perform twice; one with SSL inspection feature enabled and the second scenario with SSL inspection feature disabled on the DUT/SUT.

### **7.1.2. Test Setup**

Test bed setup MUST be configured as defined in [Section 4](#). Any test scenario specific test bed configuration changes MUST be documented.

### **7.1.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

#### **7.1.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.

#### **7.1.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be noted for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target throughput: It can be defined based on requirements. Otherwise it represents aggregated line rate of interface(s) used in the DUT/SUT

Initial throughput: 10% of the "Target throughput"

One of the following ciphers and keys are RECOMMENDED to use for this test scenarios.



1. ECDHE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash Algorithm: ecdsa\_secp256r1\_sha256 and Supported group: secp256r1)
2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash Algorithm: rsa\_pkcs1\_sha256 and Supported group: secp256)
3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash Algorithm: ecdsa\_secp384r1\_sha384 and Supported group: secp521r1)
4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash Algorithm: rsa\_pkcs1\_sha384 and Supported group: secp256)

#### **7.1.3.3. Traffic Profile**

Traffic profile: Test scenario MUST be run with a single application traffic mix profile (see [Appendix A](#) for details about traffic mix). The name of the NetSecOPEN traffic mix MUST be documented.

#### **7.1.3.4. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of total attempt transactions
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. Maximum deviation (max. dev) of application transaction time or TTLB (Time To Last Byte) MUST be less than X (The value for "X" will be finalized and updated after completion of PoC test)  
The following equation MUST be used to calculate the deviation of application transaction latency or TTLB  
$$\text{max. dev} = \max((\text{avg\_latency} - \text{min\_latency}), (\text{max\_latency} - \text{avg\_latency})) / (\text{Initial latency})$$
  
Where, the initial latency is calculated using the following equation. For this calculation, the latency values (min', avg' and max') MUST be measured during test procedure step 1 as defined in [Section 7.1.4.1](#).  
The variable latency represents application transaction latency or TTLB.



Initial latency:= min((avg' latency - min' latency) | (max' latency - avg' latency))

- d. Maximum value of Time to First Byte (TTFB) MUST be less than X

#### **7.1.3.5. Measurement**

Following KPI metrics MUST be reported for this test scenario.

Mandatory KPIs: average Throughput, average Concurrent TCP connections, TTLB/application transaction latency (minimum, average and maximum) and average application transactions per second

Optional KPIs: average TCP connections per second, average TLS handshake rate and TTFB

#### **7.1.4. Test Procedures and expected Results**

The test procedures are designed to measure the throughput performance of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps.

##### **7.1.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of the all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to generate test traffic at the "Initial throughput" rate as described in the parameters [Section 7.1.3.2](#). The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#). The DUT/SUT SHOULD reach the "Initial throughput" during the sustain phase. Measure all KPI as defined in [Section 7.1.3.5](#). The measured KPIs during the sustain phase MUST meet validation criteria "a" and "b" defined in [Section 7.1.3.4](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to step 2.

##### **7.1.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to generate traffic at the "Target throughput" rate defined in the parameter table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#). The test equipment SHOULD start to measure and record all specified KPIs. The frequency of KPI metric measurements SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.



The DUT/SUT is expected to reach the desired target throughput during the sustain phase. In addition, the measured KPIs MUST meet all validation criteria. Follow step 3, if the KPI metrics do not meet the validation criteria.

#### **7.1.4.3. Step 3: Test Iteration**

Determine the maximum and average achievable throughput within the validation criteria. Final test iteration MUST be performed for the test duration defined in [Section 4.3.4](#).

### **7.2. TCP/HTTP Connections Per Second**

#### **7.2.1. Objective**

Using HTTP traffic, determine the maximum sustainable TCP connection establishment rate supported by the DUT/SUT under different throughput load conditions.

To measure connections per second, test iterations MUST use different fixed HTTP response object sizes defined in [Section 7.2.3.2](#).

#### **7.2.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

#### **7.2.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

##### **7.2.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.

##### **7.2.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)



Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target connections per second: Initial value from product data sheet (if known)

Initial connections per second: 10% of "Target connections per second"

The client SHOULD negotiate HTTP 1.1 and close the connection with FIN immediately after completion of one transaction. In each test iteration, client MUST send GET command requesting a fixed HTTP response object size.

The RECOMMENDED response object sizes are 1, 2, 4, 16, 64 KByte

#### **[7.2.3.3.](#) Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of total attempt transactions
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic should be forwarded at a constant rate
- d. Concurrent TCP connections SHOULD be constant during steady state. Any deviation of concurrent TCP connections MUST be less than 10%. This confirms the DUT opens and closes TCP connections almost at the same rate

#### **[7.2.3.4.](#) Measurement**

Following KPI metrics MUST be reported for each test iteration.

Mandatory KPIs: average TCP connections per second, average Throughput and Average Time to First Byte (TTFB).



#### **7.2.4. Test Procedures and Expected Results**

The test procedure is designed to measure the TCP connections per second rate of the DUT/SUT at the sustaining period of the traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IP types; IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution.

##### **7.2.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure the traffic load profile of the test equipment to establish "initial connections per second" as defined in the parameters [Section 7.2.3.2](#). The traffic load profile SHOULD be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial connections per second" before the sustain phase. The measured KPIs during the sustain phase MUST meet validation criteria a, b, c, and d defined in [Section 7.2.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.2.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target connections per second" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

During the ramp up and sustain phase of each test iteration, other KPIs such as throughput, concurrent TCP connections and application transactions per second MUST NOT reach to the maximum value the DUT/SUT can support. The test results for specific test iterations SHOULD NOT be reported, if the above mentioned KPI (especially throughput) reaches the maximum value. (Example: If the test iteration with 64 KByte of HTTP response object size reached the maximum throughput limitation of the DUT, the test iteration MAY be interrupted and the result for 64 KByte SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.



The DUT/SUT is expected to reach the desired target connections per second rate at the sustain phase. In addition, the measured KPIs MUST meet all validation criteria.

Follow step 3, if the KPI metrics do not meet the validation criteria.

#### **7.2.4.3. Step 3: Test Iteration**

Determine the maximum and average achievable connections per second within the validation criteria.

### **7.3. HTTP Throughput**

#### **7.3.1. Objective**

Determine the throughput for HTTP transactions varying the HTTP response object size.

#### **7.3.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

#### **7.3.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

##### **7.3.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.

##### **7.3.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)



Target Throughput: Initial value from product data sheet (if known)

Initial Throughput: 10% of "Target Throughput"

Number of HTTP response object requests (transactions) per connection: 10

RECOMMENDED HTTP response object size: 1 KByte, 16 KByte, 64 KByte, 256 KByte and mixed objects defined in the table

| Object size (KByte) | Number of requests/<br>Weight |
|---------------------|-------------------------------|
| 0.2                 | 1                             |
| 6                   | 1                             |
| 8                   | 1                             |
| 9                   | 1                             |
| 10                  | 1                             |
| 25                  | 1                             |
| 26                  | 1                             |
| 35                  | 1                             |
| 59                  | 1                             |
| 347                 | 1                             |

Table 3: Mixed Objects

#### **7.3.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Traffic should be forwarded constantly.



- c. Concurrent connections MUST be constant. The deviation of concurrent TCP connection MUST NOT increase more than 10%

#### **7.3.3.4. Measurement**

The KPI metrics MUST be reported for this test scenario:

Average Throughput, average HTTP transactions per second, concurrent connections, and average TCP connections per second.

#### **7.3.4. Test Procedures and Expected Results**

The test procedure is designed to measure HTTP throughput of the DUT/SUT. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution and HTTP response object sizes.

##### **7.3.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of the all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial Throughput" as defined in the parameters [Section 7.3.3.2](#).

The traffic load profile SHOULD be defined as described in [Section 4.3.4](#). The DUT/SUT SHOULD reach the "Initial Throughput" during the sustain phase. Measure all KPI as defined in [Section 7.3.3.4](#).

The measured KPIs during the sustain phase MUST meet the validation criteria "a" defined in [Section 7.3.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.3.4.2. Step 2: Test Run with Target Objective**

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired "Target Throughput" at the sustain phase. In addition, the measured KPIs must meet all validation criteria.

Perform the test separately for each HTTP response object size.



Follow step 3, if the KPI metrics do not meet the validation criteria.

#### **7.3.4.3. Step 3: Test Iteration**

Determine the maximum and average achievable throughput within the validation criteria. Final test iteration MUST be performed for the test duration defined in [Section 4.3.4](#).

### **7.4. TCP/HTTP Transaction Latency**

#### **7.4.1. Objective**

Using HTTP traffic, determine the average HTTP transaction latency when DUT is running with sustainable HTTP transactions per second supported by the DUT/SUT under different HTTP response object sizes.

Test iterations MUST be performed with different HTTP response object sizes in two different scenarios. one with a single transaction and the other with multiple transactions within a single TCP connection. For consistency both the single and multiple transaction test MUST be configured with HTTP 1.1.

Scenario 1: The client MUST negotiate HTTP 1.1 and close the connection with FIN immediately after completion of a single transaction (GET and RESPONSE).

Scenario 2: The client MUST negotiate HTTP 1.1 and close the connection FIN immediately after completion of 10 transactions (GET and RESPONSE) within a single TCP connection.

#### **7.4.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

#### **7.4.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

##### **7.4.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.



#### **7.4.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#) . Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target objective for scenario 1: 50% of the maximum connection per second measured in test scenario TCP/HTTP Connections Per Second ([Section 7.2](#))

Target objective for scenario 2: 50% of the maximum throughput measured in test scenario HTTP Throughput ([Section 7.3](#))

Initial objective for scenario 1: 10% of Target objective for scenario 1"

Initial objective for scenario 2: 10% of "Target objective for scenario 2"

HTTP transaction per TCP connection: test scenario 1 with single transaction and the second scenario with 10 transactions

HTTP 1.1 with GET command requesting a single object. The RECOMMENDED object sizes are 1, 16 or 64 KByte. For each test iteration, client MUST request a single HTTP response object size.

#### **7.4.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile. Ramp up and ramp down phase SHOULD NOT be considered.

Generic criteria:

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.



- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic should be forwarded at a constant rate.
- d. Concurrent TCP connections should be constant during steady state. This confirms the DUT opens and closes TCP connections at the same rate.
- e. After ramp up the DUT MUST achieve the "Target objective" defined in the parameter [Section 7.4.3.2](#) and remain in that state for the entire test duration (sustain phase).

#### **[7.4.3.4. Measurement](#)**

Following KPI metrics MUST be reported for each test scenario and HTTP response object sizes separately:

average TCP connections per second and average application transaction latency

All KPI's are measured once the target throughput achieves the steady state.

#### **[7.4.4. Test Procedures and Expected Results](#)**

The test procedure is designed to measure the average application transaction latencies or TTLB when the DUT is operating close to 50% of its maximum achievable throughput or connections per second. This test procedure CAN be repeated multiple times with different IP types (IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution), HTTP response object sizes and single and multiple transactions per connection scenarios.

##### **[7.4.4.1. Step 1: Test Initialization and Qualification](#)**

Verify the link status of the all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial objective" as defined in the parameters [Section 7.4.3.2](#). The traffic load profile can be defined as described in [Section 4.3.4](#).



The DUT/SUT SHOULD reach the "Initial objective" before the sustain phase. The measured KPIs during the sustain phase MUST meet the validation criteria a, b, c, d, e and f defined in [Section 7.4.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

#### [7.4.4.2](#). Step 2: Test Run with Target Objective

Configure test equipment to establish "Target objective" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

During the ramp up and sustain phase, other KPIs such as throughput, concurrent TCP connections and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support. The test results for specific test iterations SHOULD NOT be reported, if the above mentioned KPI (especially throughput) reaches to the maximum value. (Example: If the test iteration with 64 KByte of HTTP response object size reached the maximum throughput limitation of the DUT, the test iteration MAY be interrupted and the result for 64 KByte SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed. DUT/SUT is expected to reach the desired "Target objective" at the sustain phase. In addition, the measured KPIs MUST meet all validation criteria.

Follow step 3, if the KPI metrics do not meet the validation criteria.

#### [7.4.4.3](#). Step 3: Test Iteration

Determine the maximum achievable connections per second within the validation criteria and measure the latency values.

### [7.5](#). Concurrent TCP/HTTP Connection Capacity

#### [7.5.1](#). Objective

Determine the maximum number of concurrent TCP connections that the DUT/ SUT sustains when using HTTP traffic.



### **7.5.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

### **7.5.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

#### **7.5.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.

#### **7.5.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be noted for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target concurrent connection: Initial value from product data sheet (if known)

Initial concurrent connection: 10% of "Target concurrent connection"

Maximum connections per second during ramp up phase: 50% of maximum connections per second measured in test scenario TCP/HTTP Connections per second ([Section 7.2](#))

Ramp up time (in traffic load profile for "Target concurrent connection"): "Target concurrent connection" / "Maximum connections per second during ramp up phase"

Ramp up time (in traffic load profile for "Initial concurrent connection"): "Initial concurrent connection" / "Maximum connections per second during ramp up phase"



The client MUST negotiate HTTP 1.1 with persistence and each client MAY open multiple concurrent TCP connections per server endpoint IP.

Each client sends 10 GET commands requesting 1 KByte HTTP response object in the same TCP connection (10 transactions/TCP connection) and the delay (think time) between the transaction MUST be X seconds.

$X = (\text{"Ramp up time"} + \text{"steady state time"}) / 10$

The established connections SHOULD remain open until the ramp down phase of the test. During the ramp down phase, all connections SHOULD be successfully closed with FIN.

#### **7.5.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transaction) of total attempted transactions
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic should be forwarded constantly

#### **7.5.3.4. Measurement**

Following KPI metrics MUST be reported for this test scenario:

average Throughput, Concurrent TCP connections (minimum, average and maximum), TTLB/ application transaction latency (minimum, average and maximum) and average application transactions per second.

#### **7.5.4. Test Procedures and expected Results**

The test procedure is designed to measure the concurrent TCP connection capacity of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.



#### **7.5.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of the all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure test equipment to establish "Initial concurrent TCP connections" defined in [Section 7.5.3.2](#). Except ramp up time, the traffic load profile SHOULD be defined as described in [Section 4.3.4](#).

During the sustain phase, the DUT/SUT SHOULD reach the "Initial concurrent TCP connections". The measured KPIs during the sustain phase MUST meet the validation criteria "a" and "b" defined in [Section 7.5.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

#### **7.5.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target concurrent TCP connections". The test equipment SHOULD follow the traffic load profile definition (except ramp up time) as described in [Section 4.3.4](#).

During the ramp up and sustain phase, the other KPIs such as throughput, TCP connections per second and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support.

The test equipment SHOULD start to measure and record KPIs defined in [Section 7.5.3.4](#). The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target concurrent connection at the sustain phase. In addition, the measured KPIs must meet all validation criteria.

Follow step 3, if the KPI metrics do not meet the validation criteria.

#### **7.5.4.3. Step 3: Test Iteration**

Determine the maximum and average achievable concurrent TCP connections capacity within the validation criteria.



## **[7.6.](#) TCP/HTTPS Connections per second**

### **[7.6.1.](#) Objective**

Using HTTPS traffic, determine the maximum sustainable SSL/TLS session establishment rate supported by the DUT/SUT under different throughput load conditions.

Test iterations MUST include common cipher suites and key strengths as well as forward looking stronger keys. Specific test iterations MUST include ciphers and keys defined in [Section 7.6.3.2](#).

For each cipher suite and key strengths, test iterations MUST use a single HTTPS response object size defined in the test equipment configuration parameters [Section 7.6.3.2](#) to measure connections per second performance under a variety of DUT Security inspection load conditions.

### **[7.6.2.](#) Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

### **[7.6.3.](#) Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

#### **[7.6.3.1.](#) DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.

#### **[7.6.3.2.](#) Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)



Target connections per second: Initial value from product data sheet (if known)

Initial connections per second: 10% of "Target connections per second"

RECOMMENDED ciphers and keys:

1. ECHDE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash Algorithm: ecdsa\_secp256r1\_sha256 and Supported group: secp256r1)
2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash Algorithm: rsa\_pkcs1\_sha256 and Supported group: secp256)
3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash Algorithm: ecdsa\_secp384r1\_sha384 and Supported group: secp521r1)
4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash Algorithm: rsa\_pkcs1\_sha384 and Supported group: secp256)

The client MUST negotiate HTTPS 1.1 and close the connection with FIN immediately after completion of one transaction. In each test iteration, client MUST send GET command requesting a fixed HTTPS response object size. The RECOMMENDED object sizes are 1, 2, 4, 16, 64 KByte.

#### **7.6.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria:

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic should be forwarded at a constant rate
- d. Concurrent TCP connections SHOULD be constant during steady state. This confirms that the DUT open and close the TCP connections at the same rate



#### **7.6.3.4. Measurement**

Following KPI metrics MUST be reported for this test scenario:

average TCP connections per second, average Throughput and Average Time to TCP First Byte.

#### **7.6.4. Test Procedures and expected Results**

The test procedure is designed to measure the TCP connections per second rate of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

##### **7.6.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial connections per second" as defined in [Section 7.6.3.2](#). The traffic load profile CAN be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial connections per second" before the sustain phase. The measured KPIs during the sustain phase MUST meet the validation criteria a, b, c, and d defined in [Section 7.6.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.6.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target connections per second" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

During the ramp up and sustain phase, other KPIs such as throughput, concurrent TCP connections and application transactions per second MUST NOT reach the maximum value that the DUT/SUT can support. The test results for specific test iteration SHOULD NOT be reported, if the above mentioned KPI (especially throughput) reaches the maximum value. (Example: If the test iteration with 64 KByte of HTTPS response object size reached the maximum throughput limitation of the DUT, the test iteration can be interrupted and the result for 64 KByte SHOULD NOT be reported).



The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target connections per second rate at the sustain phase. In addition, the measured KPIs must meet all validation criteria.

Follow the step 3, if the KPI metrics do not meet the validation criteria.

#### **7.6.4.3. Step 3: Test Iteration**

Determine the maximum and average achievable connections per second within the validation criteria.

### **7.7. HTTPS Throughput**

#### **7.7.1. Objective**

Determine the throughput for HTTPS transactions varying the HTTPS response object size.

Test iterations MUST include common cipher suites and key strengths as well as forward looking stronger keys. Specific test iterations MUST include the ciphers and keys defined in the parameter [Section 7.7.3.2](#).

#### **7.7.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. must be documented.

#### **7.7.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

##### **7.7.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.



#### **7.7.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

Target Throughput: Initial value from product data sheet (if known)

Initial Throughput: 10% of "Target Throughput"

Number of HTTPS response object requests (transactions) per connection: 10

RECOMMENDED ciphers and keys:

1. ECHDE-ECDSA-AES128-GCM-SHA256 with Prime256v1 (Signature Hash Algorithm: ecdsa\_secp256r1\_sha256 and Supported group: secp256r1)
2. ECDHE-RSA-AES128-GCM-SHA256 with RSA 2048 (Signature Hash Algorithm: rsa\_pkcs1\_sha256 and Supported group: secp256)
3. ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 (Signature Hash Algorithm: ecdsa\_secp384r1\_sha384 and Supported group: secp521r1)
4. ECDHE-RSA-AES256-GCM-SHA384 with RSA 4096 (Signature Hash Algorithm: rsa\_pkcs1\_sha384 and Supported group: secp256)

RECOMMENDED HTTPS response object size: 1 KByte, 2 KByte, 4 KByte, 16 KByte, 64 KByte, 256 KByte and mixed object defined in the table below.



| Object size (KByte) | Number of requests/<br>Weight |
|---------------------|-------------------------------|
| 0.2                 | 1                             |
| 6                   | 1                             |
| 8                   | 1                             |
| 9                   | 1                             |
| 10                  | 1                             |
| 25                  | 1                             |
| 26                  | 1                             |
| 35                  | 1                             |
| 59                  | 1                             |
| 347                 | 1                             |

Table 4: Mixed Objects

#### **7.7.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Traffic should be forwarded constantly.
- c. The deviation of concurrent TCP connections MUST be less than 10%

#### **7.7.3.4. Measurement**

The KPI metrics MUST be reported for this test scenario:

Average Throughput, Average transactions per second, concurrent connections, and average TCP connections per second.



#### **7.7.4. Test Procedures and Expected Results**

The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution and HTTPS response object sizes.

##### **7.7.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of the all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "initial throughput" as defined in the parameters [Section 7.7.3.2](#).

The traffic load profile should be defined as described in [Section 4.3.4](#). The DUT/SUT SHOULD reach the "Initial Throughput" during the sustain phase. Measure all KPI as defined in [Section 7.7.3.4](#).

The measured KPIs during the sustain phase MUST meet the validation criteria "a" defined in [Section 7.7.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.7.4.2. Step 2: Test Run with Target Objective**

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired "Target Throughput" at the sustain phase. In addition, the measured KPIs MUST meet all validation criteria.

Perform the test separately for each HTTPS response object size.

Follow step 3, if the KPI metrics do not meet the validation criteria.

##### **7.7.4.3. Step 3: Test Iteration**

Determine the maximum and average achievable throughput within the validation criteria. Final test iteration MUST be performed for the test duration defined in [Section 4.3.4](#).



## **7.8. HTTPS Transaction Latency**

### **7.8.1. Objective**

Using HTTPS traffic, determine the average HTTPS transaction latency when DUT is running with sustainable HTTPS transactions per second supported by the DUT/SUT under different HTTPS response object size.

Scenario 1: The client MUST negotiate HTTPS and close the connection with FIN immediately after completion of a single transaction (GET and RESPONSE).

Scenario 2: The client MUST negotiate HTTPS and close the connection with FIN immediately after completion of 10 transactions (GET and RESPONSE) within a single TCP connection.

### **7.8.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

### **7.8.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

#### **7.8.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.

#### **7.8.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

RECOMMENDED cipher suites and key size: ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 bits key size (Signature Hash Algorithm: ecdsa\_secp384r1\_sha384 and Supported group: secp521r1)



Target objective for scenario 1: 50% of the maximum connections per second measured in test scenario TCP/HTTPS Connections per second ([Section 7.6](#))

Target objective for scenario 2: 50% of the maximum throughput measured in test scenario HTTPS Throughput ([Section 7.7](#))

Initial objective for scenario 1: 10% of Target objective for scenario 1"

Initial objective for scenario 2: 10% of "Target objective for scenario 2"

HTTPS transaction per TCP connection: test scenario 1 with single transaction and the second scenario with 10 transactions

HTTPS 1.1 with GET command requesting a single 1, 16 or 64 KByte object. For each test iteration, client MUST request a single HTTPS response object size.

#### **[7.8.3.3](#). Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile. Ramp up and ramp down phase SHOULD NOT be considered.

Generic criteria:

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of attempt transactions.
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic should be forwarded at a constant rate.
- d. Concurrent TCP connections should be constant during steady state. This confirms the DUT opens and closes TCP connections at the same rate.
- e. After ramp up the DUT MUST achieve the "Target objective" defined in the parameter [Section 7.8.3.2](#) and remain in that state for the entire test duration (sustain phase).



#### **7.8.3.4. Measurement**

Following KPI metrics MUST be reported for each test scenario and HTTPS response object sizes separately:

average TCP connections per second and average application transaction latency or TTLB

All KPI's are measured once the target connections per second achieves the steady state.

#### **7.8.4. Test Procedures and Expected Results**

The test procedure is designed to measure average application transaction latency or TTLB when the DUT is operating close to 50% of its maximum achievable connections per second. This test procedure can be repeated multiple times with different IP types (IPv4 only, IPv6 only and IPv4 and IPv6 mixed traffic distribution), HTTPS response object sizes and single and multiple transactions per connection scenarios.

##### **7.8.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of the all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure traffic load profile of the test equipment to establish "Initial objective" as defined in the parameters [Section 7.8.3.2](#). The traffic load profile can be defined as described in [Section 4.3.4](#).

The DUT/SUT SHOULD reach the "Initial objective" before the sustain phase. The measured KPIs during the sustain phase MUST meet the validation criteria a, b, c, d, e and f defined in [Section 7.8.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

##### **7.8.4.2. Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target objective" defined in the parameters table. The test equipment SHOULD follow the traffic load profile definition as described in [Section 4.3.4](#).

During the ramp up and sustain phase, other KPIs such as throughput, concurrent TCP connections and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support. The test results for specific test iterations SHOULD NOT be reported,



if the above mentioned KPI (especially throughput) reaches to the maximum value. (Example: If the test iteration with 64 KByte of HTTP response object size reached the maximum throughput limitation of the DUT, the test iteration MAY be interrupted and the result for 64 KByte SHOULD NOT be reported).

The test equipment SHOULD start to measure and record all specified KPIs. The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed. DUT/SUT is expected to reach the desired "Target objective" at the sustain phase. In addition, the measured KPIs MUST meet all validation criteria.

Follow step 3, if the KPI metrics do not meet the validation criteria.

#### **7.8.4.3. Step 3: Test Iteration**

Determine the maximum achievable connections per second within the validation criteria and measure the latency values.

### **7.9. Concurrent TCP/HTTPS Connection Capacity**

#### **7.9.1. Objective**

Determine the maximum number of concurrent TCP connections that the DUT/SUT sustains when using HTTPS traffic.

#### **7.9.2. Test Setup**

Test bed setup SHOULD be configured as defined in [Section 4](#). Any specific test bed configuration changes such as number of interfaces and interface type, etc. MUST be documented.

#### **7.9.3. Test Parameters**

In this section, test scenario specific parameters SHOULD be defined.

##### **7.9.3.1. DUT/SUT Configuration Parameters**

DUT/SUT parameters MUST conform to the requirements defined in [Section 4.2](#). Any configuration changes for this specific test scenario MUST be documented.



#### **7.9.3.2. Test Equipment Configuration Parameters**

Test equipment configuration parameters MUST conform to the requirements defined in [Section 4.3](#). Following parameters MUST be documented for this test scenario:

Client IP address range defined in [Section 4.3.1.2](#)

Server IP address range defined in [Section 4.3.2.2](#)

Traffic distribution ratio between IPv4 and IPv6 defined in [Section 4.3.1.2](#)

RECOMMENDED cipher suites and key size: ECDHE-ECDSA-AES256-GCM-SHA384 with Secp521 bits key size (Signature Hash Algorithm: ecdsa\_secp384r1\_sha384 and Supported group: secp521r1)

Target concurrent connections: Initial value from product data sheet (if known)

Initial concurrent connections: 10% of "Target concurrent connections"

Connections per second during ramp up phase: 50% of maximum connections per second measured in test scenario TCP/HTTPS Connections per second ([Section 7.6](#))

Ramp up time (in traffic load profile for "Target concurrent connections"): "Target concurrent connections" / "Maximum connections per second during ramp up phase"

Ramp up time (in traffic load profile for "Initial concurrent connections"): "Initial concurrent connections" / "Maximum connections per second during ramp up phase"

The client MUST perform HTTPS transaction with persistence and each client can open multiple concurrent TCP connections per server endpoint IP.

Each client sends 10 GET commands requesting 1 KByte HTTPS response objects in the same TCP connections (10 transactions/TCP connection) and the delay (think time) between each transactions MUST be X seconds.

$X = ("Ramp\ up\ time" + "steady\ state\ time") / 10$



The established connections SHOULD remain open until the ramp down phase of the test. During the ramp down phase, all connections SHOULD be successfully closed with FIN.

#### **7.9.3.3. Test Results Validation Criteria**

The following test Criteria is defined as test results validation criteria. Test results validation criteria MUST be monitored during the whole sustain phase of the traffic load profile.

- a. Number of failed Application transactions (receiving any HTTP response code other than 200 OK) MUST be less than 0.001% (1 out of 100,000 transactions) of total attempted transactions
- b. Number of Terminated TCP connections due to unexpected TCP RST sent by DUT/SUT MUST be less than 0.001% (1 out of 100,000 connections) of total initiated TCP connections
- c. During the sustain phase, traffic SHOULD be forwarded constantly

#### **7.9.3.4. Measurement**

Following KPI metrics MUST be reported for this test scenario:

Average Throughput, max. Min. Avg. Concurrent TCP connections, TTLB/application transaction latency and average application transactions per second

#### **7.9.4. Test Procedures and expected Results**

The test procedure is designed to measure the concurrent TCP connection capacity of the DUT/SUT at the sustaining period of traffic load profile. The test procedure consists of three major steps. This test procedure MAY be repeated multiple times with different IPv4 and IPv6 traffic distribution.

##### **7.9.4.1. Step 1: Test Initialization and Qualification**

Verify the link status of all connected physical interfaces. All interfaces are expected to be in "UP" status.

Configure test equipment to establish "initial concurrent TCP connections" defined in [Section 7.9.3.2](#). Except ramp up time, the traffic load profile SHOULD be defined as described in [Section 4.3.4](#).

During the sustain phase, the DUT/SUT SHOULD reach the "Initial concurrent TCP connections". The measured KPIs during the sustain



phase MUST meet the validation criteria "a" and "b" defined in [Section 7.9.3.3](#).

If the KPI metrics do not meet the validation criteria, the test procedure MUST NOT be continued to "Step 2".

#### **[7.9.4.2](#). Step 2: Test Run with Target Objective**

Configure test equipment to establish "Target concurrent TCP connections". The test equipment SHOULD follow the traffic load profile definition (except ramp up time) as described in [Section 4.3.4](#).

During the ramp up and sustain phase, the other KPIs such as throughput, TCP connections per second and application transactions per second MUST NOT reach to the maximum value that the DUT/SUT can support.

The test equipment SHOULD start to measure and record KPIs defined in [Section 7.9.3.4](#). The frequency of measurement SHOULD be 2 seconds. Continue the test until all traffic profile phases are completed.

The DUT/SUT is expected to reach the desired target concurrent connections at the sustain phase. In addition, the measured KPIs MUST meet all validation criteria.

Follow step 3, if the KPI metrics do not meet the validation criteria.

#### **[7.9.4.3](#). Step 3: Test Iteration**

Determine the maximum and average achievable concurrent TCP connections within the validation criteria.

### **[8](#). Formal Syntax**

### **[9](#). IANA Considerations**

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

### **[10](#). Security Considerations**

The primary goal of this document is to provide benchmarking terminology and methodology for next-generation network security devices. However, readers should be aware that there is some overlap



between performance and security issues. Specifically, the optimal configuration for network security device performance may not be the most secure, and vice-versa. The Cipher suites are recommended in this document are just for test purpose only. The Cipher suite recommendation for a real deployment is outside the scope of this document.

## **11. Acknowledgements**

Acknowledgements will be added in the future release.

## **12. Contributors**

The authors would like to thank the many people that contributed their time and knowledge to this effort.

Specifically, to the co-chairs of the NetSecOPEN Test Methodology working group and the NetSecOPEN Security Effectiveness working group - Alex Samonte, Aria Eslambolchizadeh, Carsten Rossenhoevel and David DeSanto.

Additionally, the following people provided input, comments and spent time reviewing the myriad of drafts. If we have missed anyone the fault is entirely our own. Thanks to - Amritam Putatunda, Chao Guo, Chris Chapman, Chris Pearson, Chuck McAuley, David White, Jurrie Van Den Breekel, Michelle Rhines, Rob Andrews, Samaresh Nair, and Tim Winters.

## **13. References**

### **13.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### **13.2. Informative References**

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), DOI 10.17487/RFC2616, June 1999, <<https://www.rfc-editor.org/info/rfc2616>>.



- [RFC2647] Newman, D., "Benchmarking Terminology for Firewall Performance", [RFC 2647](#), DOI 10.17487/RFC2647, August 1999, <<https://www.rfc-editor.org/info/rfc2647>>.
- [RFC3511] Hickman, B., Newman, D., Tadjudin, S., and T. Martin, "Benchmarking Methodology for Firewall Performance", [RFC 3511](#), DOI 10.17487/RFC3511, April 2003, <<https://www.rfc-editor.org/info/rfc3511>>.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", [RFC 5681](#), DOI 10.17487/RFC5681, September 2009, <<https://www.rfc-editor.org/info/rfc5681>>.

#### **Appendix A. NetSecOPEN Basic Traffic Mix**

A traffic mix for testing performance of next generation firewalls MUST scale to stress the DUT based on real-world conditions. In order to achieve this the following MUST be included:

- o Clients connecting to multiple different server FQDNs per application
- o Clients loading apps and pages with connections and objects in specific orders
- o Multiple unique certificates for HTTPS/TLS
- o A wide variety of different object sizes
- o Different URL paths
- o Mix of HTTP and HTTPS

A traffic mix for testing performance of next generation firewalls MUST also facility application identification using different detection methods with and without decryption of the traffic. Such as:

- o HTTP HOST based application detection
- o HTTPS/TLS Server Name Indication (SNI)
- o Certificate Subject Common Name (CN)

The mix MUST be of sufficient complexity and volume to render differences in individual apps as statistically insignificant. For example, changes in like to like apps - such as one type of video service vs. another both consist of larger objects whereas one news



site vs. another both typically have more connections than other apps because of trackers and embedded advertising content. To achieve sufficient complexity, a mix MUST have:

- o Thousands of URLs each client walks thru
- o Hundreds of FQDNs each client connects to
- o Hundreds of unique certificates for HTTPS/TLS
- o Thousands of different object sizes per client in orders matching applications

The following is a description of what a popular application in an enterprise traffic mix contains.

Table 5 lists the FQDNs, number of transactions and bytes transferred as an example client interacts with Office 365 Outlook, Word, Excel, PowerPoint, SharePoint and Skype.

| Office365 FQDN                  | Bytes      | Transaction |
|---------------------------------|------------|-------------|
| r1.res.office365.com            | 14,056,960 | 192         |
| s1-word-edit-15.cdn.office.net  | 6,731,019  | 22          |
| company1-my.sharepoint.com      | 6,269,492  | 42          |
| swx.cdn.skype.com               | 6,100,027  | 12          |
| static.sharepointonline.com     | 6,036,947  | 41          |
| spoprod-a.akamaihd.net          | 3,904,250  | 25          |
| s1-excel-15.cdn.office.net      | 2,767,941  | 16          |
| outlook.office365.com           | 2,047,301  | 86          |
| shellprod.msocdn.com            | 1,008,370  | 11          |
| word-edit.officeapps.live.com   | 932,080    | 25          |
| res.delve.office.com            | 760,146    | 2           |
| s1-powerpoint-15.cdn.office.net | 557,604    | 3           |
| appsforoffice.microsoft.com     | 511,171    | 5           |



|                                 |         |    |
|---------------------------------|---------|----|
| powerpoint.officeapps.live.com  | 471,625 | 14 |
| excel.officeapps.live.com       | 342,040 | 14 |
| s1-officeapps-15.cdn.office.net | 331,343 | 5  |
| webdir0a.online.lync.com        | 66,930  | 15 |
| portal.office.com               | 13,956  | 1  |
| config.edge.skype.com           | 6,911   | 2  |
| clientlog.portal.office.com     | 6,608   | 8  |
| webdir.online.lync.com          | 4,343   | 5  |
| graph.microsoft.com             | 2,289   | 2  |
| nam.loki.delve.office.com       | 1,812   | 5  |
| login.microsoftonline.com       | 464     | 2  |
| login.windows.net               | 232     | 1  |

Table 5: Office365

Clients MUST connect to multiple server FQDNs in the same order as real applications. Connections MUST be made when the client is interacting with the application and MUST NOT first setup up all connections. Connections SHOULD stay open per client for subsequent transactions to the same FQDN similar to how a web browser behaves. Clients MUST use different URL Paths and Object sizes in orders as they are observed in real Applications. Clients MAY also setup multiple connections per FQDN to process multiple transactions in a sequence at the same time. Table 6 has a partial example sequence of the Office 365 Word application transactions.

| FQDN                          | URL Path           | Object size |
|-------------------------------|--------------------|-------------|
| company1-my.sharepoint.com    | /personal...       | 23,132      |
| word-edit.officeapps.live.com | /we/WsaUpload.ashx | 2           |
| static.sharepointonline.com   | /bld/.../blank.js  | 454         |



|                                 |                      |           |  |
|---------------------------------|----------------------|-----------|--|
| +-----+-----+-----+             |                      |           |  |
| static.sharepointonline.com     | /bld/.../            | 23,254    |  |
|                                 | initstrings.js       |           |  |
| +-----+-----+-----+             |                      |           |  |
| static.sharepointonline.com     | /bld/.../init.js     | 292,740   |  |
| +-----+-----+-----+             |                      |           |  |
| company1-my.sharepoint.com      | /ScriptResource...   | 102,774   |  |
| +-----+-----+-----+             |                      |           |  |
| company1-my.sharepoint.com      | /ScriptResource...   | 40,329    |  |
| +-----+-----+-----+             |                      |           |  |
| company1-my.sharepoint.com      | /WebResource...      | 23,063    |  |
| +-----+-----+-----+             |                      |           |  |
| word-edit.officeapps.live.com   | /we/wordeditorframe. | 60,657    |  |
|                                 | aspx...              |           |  |
| +-----+-----+-----+             |                      |           |  |
| static.sharepointonline.com     | /bld/_layouts/.../   | 454       |  |
|                                 | blank.js             |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 19,201    |  |
|                                 | EditSurface.css      |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 221,397   |  |
|                                 | WordEditor.css       |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-officeapps-15.cdn.office.net | /we/s/.../           | 107,571   |  |
|                                 | Microsoft            |           |  |
|                                 | Ajax.js              |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 39,981    |  |
|                                 | wacbootwe.js         |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-officeapps-15.cdn.office.net | /we/s/.../           | 51,749    |  |
|                                 | CommonIntl.js        |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 6,050     |  |
|                                 | Compat.js            |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 54,158    |  |
|                                 | Box4Intl.js          |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 24,946    |  |
|                                 | WoncaIntl.js         |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 53,515    |  |
|                                 | WordEditorIntl.js    |           |  |
| +-----+-----+-----+             |                      |           |  |
| s1-word-edit-15.cdn.office.net  | /we/s/.../           | 1,978,712 |  |
|                                 | WordEditorExp.js     |           |  |



|                                |                      |         |  |
|--------------------------------|----------------------|---------|--|
| s1-word-edit-15.cdn.office.net | /we/s/.../jSanity.js | 10,912  |  |
| word-edit.officeapps.live.com  | /we/OneNote.ashx     | 145,708 |  |

Table 6: Office365 Word Transactions

For application identification the HTTPS/TLS traffic MUST include realistic Certificate Subject Common Name (CN) data as well as Server Name Indications (SNI). For example, a DUT MAY detect Facebook Chat traffic by inspecting the certificate and detecting \*.facebook.com in the certificate subject CN and subsequently detect the word chat in the FQDN 5-edge-chat.facebook.com and identify traffic on the connection to be Facebook Chat.

Table 7 includes further examples in SNI and CN pairs for several FQDNs of Office 365.

| Server Name Indication (SNI) | Certificate Subject<br>  Common Name (CN) |  |
|------------------------------|---|--|
| r1.res.office365.com         | *.res.outlook.com                         |  |
| login.windows.net            | graph.windows.net                         |  |
| webdir0a.online.lync.com     | *.online.lync.com                         |  |
| login.microsoftonline.com    | stamp2.login.microsoftonline.com          |  |
| webdir.online.lync.com       | *.online.lync.com                         |  |
| graph.microsoft.com          | graph.microsoft.com                       |  |
| outlook.office365.com        | outlook.com                               |  |
| appsforoffice.microsoft.com  | appsforoffice.microsoft.com               |  |

Table 7: Office365 SNI and CN Pairs Examples

NetSecOPEN has provided a reference enterprise perimeter traffic mix with dozens of applications, hundreds of connections, and thousands of transactions.

The enterprise perimeter traffic mix consists of 70% HTTPS and 30% HTTP by Bytes, 58% HTTPS and 42% HTTP by Transactions. By



connections with a single connection per FQDN the mix consists of 43% HTTPS and 57% HTTP. With multiple connections per FQDN the HTTPS percentage is higher.

Table 8 is a summary of the NetSecOPEN enterprise perimeter traffic mix sorted by bytes with unique FQDNs and transactions per applications.

| Application     | FQDNs | Transactions | Bytes      |
|-----------------|-------|--------------|------------|
| Office365       | 26    | 558          | 52,931,947 |
| Box             | 4     | 90           | 23,276,089 |
| Salesforce      | 6     | 365          | 23,137,548 |
| Gmail           | 13    | 139          | 16,399,289 |
| Linkedin        | 10    | 206          | 15,040,918 |
| DailyMotion     | 8     | 77           | 14,751,514 |
| GoogleDocs      | 2     | 71           | 14,205,476 |
| Wikia           | 15    | 159          | 13,909,777 |
| Foxnews         | 82    | 499          | 13,758,899 |
| Yahoo Finance   | 33    | 254          | 13,134,011 |
| Youtube         | 8     | 97           | 13,056,216 |
| Facebook        | 4     | 207          | 12,726,231 |
| CNBC            | 77    | 275          | 11,939,566 |
| Lightreading    | 27    | 304          | 11,200,864 |
| BusinessInsider | 16    | 142          | 11,001,575 |
| Alexa           | 5     | 153          | 10,475,151 |
| CNN             | 41    | 206          | 10,423,740 |
| Twitter Video   | 2     | 72           | 10,112,820 |
| Cisco Webex     | 1     | 213          | 9,988,417  |



|               |    |     |           |  |
|---------------|----|-----|-----------|--|
| Slack         | 3  | 40  | 9,938,686 |  |
| Google Maps   | 5  | 191 | 8,771,873 |  |
| SpectrumIEEE  | 7  | 145 | 8,682,629 |  |
| Yelp          | 9  | 146 | 8,607,645 |  |
| Vimeo         | 12 | 74  | 8,555,960 |  |
| Wikihow       | 11 | 140 | 8,042,314 |  |
| Netflix       | 3  | 31  | 7,839,256 |  |
| Instagram     | 3  | 114 | 7,230,883 |  |
| Morningstar   | 30 | 150 | 7,220,121 |  |
| Docusign      | 5  | 68  | 6,972,738 |  |
| Twitter       | 1  | 100 | 6,939,150 |  |
| Tumblr        | 11 | 70  | 6,877,200 |  |
| Whatsapp      | 3  | 46  | 6,829,848 |  |
| Imdb          | 16 | 251 | 6,505,227 |  |
| NOAAgov       | 1  | 44  | 6,316,283 |  |
| IndustryWeek  | 23 | 192 | 6,242,403 |  |
| Spotify       | 18 | 119 | 6,231,013 |  |
| AutoNews      | 16 | 165 | 6,115,354 |  |
| Evernote      | 3  | 47  | 6,063,168 |  |
| NatGeo        | 34 | 104 | 6,026,344 |  |
| BBC News      | 18 | 156 | 5,898,572 |  |
| Investopedia  | 38 | 241 | 5,792,038 |  |
| Pinterest     | 8  | 102 | 5,658,994 |  |
| Succesfactors | 2  | 112 | 5,049,001 |  |



|               |    |     |           |  |
|---------------|----|-----|-----------|--|
| AbaJournal    | 6  | 93  | 4,985,626 |  |
| Pbworks       | 4  | 78  | 4,670,980 |  |
| NetworkWorld  | 42 | 153 | 4,651,354 |  |
| WebMD         | 24 | 280 | 4,416,736 |  |
| OilGasJournal | 14 | 105 | 4,095,255 |  |
| Trello        | 5  | 39  | 4,080,182 |  |
| BusinessWire  | 5  | 109 | 4,055,331 |  |
| Dropbox       | 5  | 17  | 4,023,469 |  |
| Nejm          | 20 | 190 | 4,003,657 |  |
| OilGasDaily   | 7  | 199 | 3,970,498 |  |
| Chase         | 6  | 52  | 3,719,232 |  |
| MedicalNews   | 6  | 117 | 3,634,187 |  |
| Marketwatch   | 25 | 142 | 3,291,226 |  |
| Imgur         | 5  | 48  | 3,189,919 |  |
| NPR           | 9  | 83  | 3,184,303 |  |
| Onelogin      | 2  | 31  | 3,132,707 |  |
| Concur        | 2  | 50  | 3,066,326 |  |
| Service-now   | 1  | 37  | 2,985,329 |  |
| Apple itunes  | 14 | 80  | 2,843,744 |  |
| BerkeleyEdu   | 3  | 69  | 2,622,009 |  |
| MSN           | 39 | 203 | 2,532,972 |  |
| Indeed        | 3  | 47  | 2,325,197 |  |
| MayoClinic    | 6  | 56  | 2,269,085 |  |
| Ebay          | 9  | 164 | 2,219,223 |  |



|                           |     |       |             |
|---------------------------|-----|-------|-------------|
| +-----+-----+-----+-----+ |     |       |             |
| UCLAedu                   | 3   | 42    | 1,991,311   |
| +-----+-----+-----+-----+ |     |       |             |
| ConstructionDive          | 5   | 125   | 1,828,428   |
| +-----+-----+-----+-----+ |     |       |             |
| EducationNews             | 4   | 78    | 1,605,427   |
| +-----+-----+-----+-----+ |     |       |             |
| BofA                      | 12  | 68    | 1,584,851   |
| +-----+-----+-----+-----+ |     |       |             |
| ScienceDirect             | 7   | 26    | 1,463,951   |
| +-----+-----+-----+-----+ |     |       |             |
| Reddit                    | 8   | 55    | 1,441,909   |
| +-----+-----+-----+-----+ |     |       |             |
| FoodBusinessNews          | 5   | 49    | 1,378,298   |
| +-----+-----+-----+-----+ |     |       |             |
| Amex                      | 8   | 42    | 1,270,696   |
| +-----+-----+-----+-----+ |     |       |             |
| Weather                   | 4   | 50    | 1,243,826   |
| +-----+-----+-----+-----+ |     |       |             |
| Wikipedia                 | 3   | 27    | 958,935     |
| +-----+-----+-----+-----+ |     |       |             |
| Bing                      | 1   | 52    | 697,514     |
| +-----+-----+-----+-----+ |     |       |             |
| ADP                       | 1   | 30    | 508,654     |
| +-----+-----+-----+-----+ |     |       |             |
|                           |     |       |             |
| +-----+-----+-----+-----+ |     |       |             |
| Grand Total               | 983 | 10021 | 569,819,095 |
| +-----+-----+-----+-----+ |     |       |             |

Table 8: Summary of NetSecOPEN Enterprise Perimeter Traffic Mix

## Authors' Addresses

Balamuhunthan Balarajah

Email: [bm.balarajah@gmail.com](mailto:bm.balarajah@gmail.com)

Carsten Rossenhoevel

EANTC AG

Salzufer 14

Berlin 10587

Germany

Email: [cross@eantc.de](mailto:cross@eantc.de)



Brian Monkman  
NetSecOPEN

Email: [bmonkman@netsecopen.org](mailto:bmonkman@netsecopen.org)