

Moonv6 Test Suite
*Transition Traffic Forwarding
Test Suite*

Technical Document

Revision 0.01



*IPv6 Consortium
InterOperability Laboratory
Research Computing Center
University of New Hampshire*

*121 Technology Drive, Suite 2
Durham, NH 03824-3525
Phone: (603) 862-2804
Fax: (603) 862-4181
<http://www.iol.unh.edu>*

ACKNOWLEDGEMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals in the development of this test suite. This test suite belongs to the University of New Hampshire InterOperability Lab and is a collaborative effort of those listed below and the participants of Moonv6.

Paul Collman	Agilent Technologies
Chris Gillis	Agilent Technologies
Phillip Kazakov	Agilent Technologies
Thomas Peterson	University of New Hampshire
Benjamin Schultz	University of New Hampshire

INTRODUCTION

Overview

The University of New Hampshire's InterOperability Laboratory (IOL) is an institution designed to improve the interoperability of standards-based products by providing an environment where a product can be tested against other implementations of a standard. This suite of tests has been developed to help implementers evaluate the functioning of their Internet Protocol, version 6 firewall products. The tests do not determine if a product conforms to the IPv6 specifications, nor are they purely interoperability tests. Rather, they provide one method to isolate problems within a device. Successful completion of all tests contained in this suite does not guarantee that the tested device will interoperate with other IPv6 devices. However, combined with satisfactory operation in the IOL's semi-production environment, these tests provide a reasonable level of confidence that the Device Under Test will function well in most multi-vendor IPv6 environments.

In this test suite, when using interface oriented terms such as "accept...on the interface" or "configure ... on its interface", it is up to the firewall vendor to supply the desired functionality according to the implementation of the DUT. The term "interface" only describes the externally observable behavior, not the specifics of an internal configuration.

Acronyms

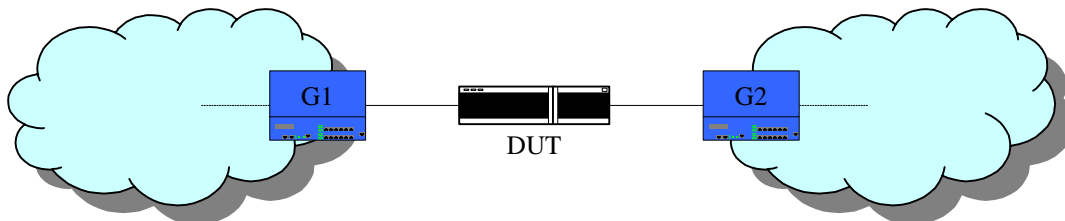
DUT: Device Under Test

TR: Testing Router

G: Traffic Generator

When several entities of the same type are present in a test configuration, a number is appended to the acronym to yield a label for each entity. For example, if there were three traffic generators in the test configuration, they would be labeled G1, G2 and G3.

Test Configuration



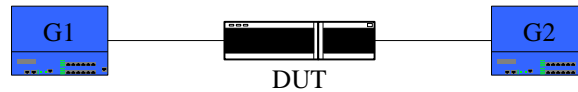
Basic Test Configuration

Test 1.1: IPv4 Baseline Testing

Purpose: To provide a baseline of IPv4 results to compare with results recorded in transitional IPv4 and IPv6 scenarios

Resource Requirements: Monitor to capture packets, packet generators.

Test Setup: Connect devices as shown below:



Procedure:

Part A: Traffic Forwarding in an IPv4 Network

1. Configure the DUT to allow all traffic
2. Configure the traffic generators to generate IPv4 traffic streams consisting of various protocol data including, HTTP, FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP. The percentage of bytes per second for each application is distributed according the following Realistic Protocol Distribution Simulation:
HTTP: 35%
FTP: 10%
POP3: 15%
SMTP: 15%
Telnet: 5%
NNTP: 3%
SNMP: 2%
SIP: 5%
H.323: 5%
RTP 5%
3. Observe and record the results.

Part B: Denying Application Traffic Based on Application Data (IPv4).

1. Configure the DUT to allow HTTP application traffic.
2. Configure the traffic generators to generate IPv4 traffic streams consisting of HTTP traffic.
3. Observe and record the results.
4. Configure the DUT to disallow HTTP application traffic based on application layer data.
5. Configure the traffic generators to generate IPv4 HTTP application traffic streams such that 50% matches the application layer data the DUT was configured to disallow in step 4.
6. Observe and record the results
7. Continue steps 1 through 6 for each of the following applications: FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP.

Part D: Allowing Peer-to-Peer Traffic (IPv4).

1. Configure the DUT to allow all traffic
2. Configure the traffic generators to generate IPv4 Peer-to-Peer application traffic streams.
3. Observe and record the results.

Part E: Denying Peer-to-Peer Traffic (IPv4).

1. Configure the DUT to disallow Peer-to-Peer application traffic.
2. Configure the traffic generators to generate IPv4 Peer-to-Peer application traffic streams.
3. Observe and record the results.

Observable Results:

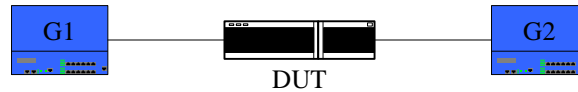
- Compare the results recorded in these tests with the results in the following tests to the results that will be recorded in the following tests. Examine the difference between the tests conducted in an IPv4 network and the following tests conducted in an IPv6 network and the tests conducted in an IPv4 and IPv6 transitional network.

Test 1.2: Forwarding IPv4 and IPv6 Traffic in a Transitional Environment

Purpose: To determine the effects of introducing IPv6 into an IPv4 network.

Resource Requirements: Monitor to capture packets, packet generators.

Test Setup: Connect devices as shown below:



Procedure:

Part A: Traffic Forwarding in an IPv6 Network

1. Configure the DUT to allow all traffic
2. Configure the traffic generators to generate IPv6 traffic streams consisting of various protocol data including, HTTP, FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP. The percentage of bytes per second for each application is distributed according the following Realistic Protocol Distribution Simulation:
HTTP: 35%
FTP: 10%
POP3: 15%
SMTP: 15%
Telnet: 5%
NNTP: 3%
SNMP: 2%
SIP: 5%
H.323: 5%
RTP 5%
3. Observe and record the results.

Part B: Traffic Forwarding in a Transitional Environment of IPv4 and IPv6

1. Configure the traffic generators to generate a mix of 10% IPv6 traffic streams and 90% IPv4 traffic streams consisting of various protocol data including, HTTP, FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP. The percentage of bytes per second for each application is distributed according the following Realistic Protocol Distribution Simulation:
HTTP: 35%
FTP: 10%
POP3: 15%
SMTP: 15%
Telnet: 5%
NNTP: 3%
SNMP: 2%
SIP: 5%
H.323: 5%
RTP 5%

2. Observe and record the results.
3. Increase the amount of IPv6 traffic generated by the traffic generators by 10% while observing and recording the results until 100% of the traffic generated is IPv6 traffic.
4. Observe and record the results.
5. Compare the results recorded in all parts.

Observable Results:

- Examine the difference in the forwarding capabilities of the DUT when forwarding traffic in an IPv4 network and an IPv6 network using the results recorded in Test 1.1.
- Determine how the introduction of IPv6 traffic in an IPv4 network affects the forwarding capabilities of the DUT.
- Determine how increasing the amount of IPv6 traffic in an IPv4 network affects the forwarding capabilities of the DUT.

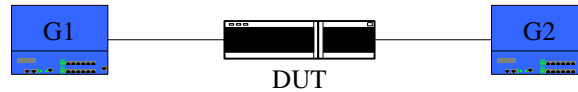
Possible Problems: None

Test 1.3: Disallowing IPv6 Traffic in a Transitional Environment

Purpose: To determine if the DUT can disallow forwarding traffic based on application data in an IPv6 environment and a transitional IPv4 and IPv6 environment.

Resource Requirements: Monitor to capture packets, packet generators.

Test Setup: Connect devices as shown below:



Procedure:

Part A: Allowing All Traffic (IPv6).

1. Configure the DUT to allow all traffic
2. Configure the traffic generators to generate IPv6 traffic streams consisting of HTTP traffic.
3. Observe and record the results.
4. Continue steps 1 through 3 for each of the following applications: FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP.

Part B: Denying Application Traffic Based on Application Data (IPv6).

1. Configure the DUT to allow HTTP application traffic.
2. Configure the traffic generators to generate IPv6 traffic streams consisting of HTTP traffic.
3. Observe and record the results.
4. Configure the DUT to disallow HTTP application traffic based on application layer data.
5. Configure the traffic generators to generate IPv6 HTTP application traffic streams such that 50% matches the application layer data the DUT was configured to disallow in step 4.
6. Observe and record the results.
7. Continue steps 1 through 6 for each of the following applications: FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP.

Part C: Allowing All Traffic (IPv4 and IPv6).

1. Configure the DUT to allow all traffic
2. Configure the traffic generators to generate IPv4 and IPv6 traffic streams consisting of HTTP traffic.
3. Observe and record the results.
4. Continue steps 1 through 3 for each of the following applications: FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP.

Part D: Denying Application Traffic Based on Application Data (IPv4 and IPv6).

1. Configure the DUT to allow HTTP application traffic.
2. Configure the traffic generators to generate IPv4 and IPv6 traffic streams consisting of HTTP traffic.
3. Observe and record the results.
4. Configure the DUT to disallow HTTP application traffic based on application layer data.

5. Configure the traffic generators to generate IPv4 and IPv6 HTTP application traffic streams such that 50% matches the application layer data the DUT was configured to disallow in step 4.
6. Observe and record the results.
7. Continue steps 1 through 6 for each of the following applications: FTP, POP3, SMTP, NNTP, Telnet, SNMP, SIP, H.323, and RTP.

Observable Results:

Part A:

- The DUT should forward all of the application traffic.

Part B:

- In steps 3 and 6 the DUT should not forward traffic containing the application data it is configured to disallow. The DUT should forward traffic that does not contain the application data it is configured to disallow.
- Compare the results recorded in Step 6 of Part B with the results recorded in Part A to determine the difference in the traffic forwarding performance when the DUT is under a blocking load.

Part C:

- The DUT should forward all of the application traffic.

Part D:

- In steps 3 and 6 the DUT should not forward traffic containing the application data it is configured to disallow. The DUT should forward traffic that does not contain the application data it is configured to disallow.
- Compare the results recorded in Step 6 of Part D with the results recorded in Part C to determine the difference in the traffic forwarding performance when the DUT is under a blocking load.

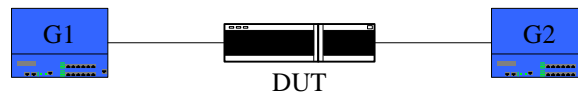
Possible Problems: None

Test 1.4: Disallowing Peer-to-Peer IPv6 Traffic in a Transitional Environment

Purpose: To determine if the DUT can disallow forwarding Peer-to-Peer application traffic in an IPv6 environment and a transitional IPv4 and IPv6 environment.

Resource Requirements: Monitor to capture packets, packet generators.

Test Setup: Connect devices as shown below:



Procedure:

Part A: Allowing Peer-to-Peer Traffic (IPv6).

1. Configure the DUT to allow all traffic
2. Configure the traffic generators to generate IPv6 Peer-to-Peer application traffic streams.
3. Observe and record the results.

Part B: Denying Peer-to-Peer Traffic (IPv6).

1. Configure the DUT to disallow Peer-to-Peer application traffic.
2. Configure the traffic generators to generate IPv6 Peer-to-Peer application traffic streams.
3. Observe and record the results.

Part C: Allowing Peer-to-Peer Traffic (IPv4 and IPv6).

1. Configure the DUT to allow all traffic.
2. Configure the traffic generators to generate IPv4 and IPv6 Peer-to-Peer application traffic streams.
3. Observe and record the results.

Part D: Denying Peer-to-Peer Traffic (IPv4 and IPv6).

1. Configure the DUT to disallow Peer-to-Peer application traffic.
2. Configure the traffic generators to generate IPv4 and IPv6 Peer-to-Peer application traffic streams.
3. Observe and record the results.

Observable Results:

Part A:

- The DUT should forward all of the Peer-to-Peer application traffic.

Part B:

- The DUT should not forward traffic containing Peer-to-Peer application data.

Part C:

- The DUT should forward all of the Peer-to-Peer application traffic.

Part D:

- The DUT should not forward traffic containing Peer-to-Peer application data.

Possible Problems: None