

## **Agilent Tests L2-7 at MoonV6**

University of New Hampshire, Interoperability Lab  
March 2004

The Moonv6 interoperability and test event is a collaborative effort to enhance awareness of IPv6 in North America. Moonv6 is a collaboration between the University of New Hampshire – Interoperability Labs (UNH-IOL), the US Department of Defense (DoD), the North American IPv6 Task Force (NAv6TF) and Internet2 (I2). This Phase II of Moonv6 also launched a network as a permanent IPv6 backbone available for global peering and data communications.

Agilent's RouterTester900 and NetworkTester test products played a major role in verifying key aspects of IPv6 across various network equipment vendors. This paper illustrates these key test scenarios.



**Agilent Technologies**

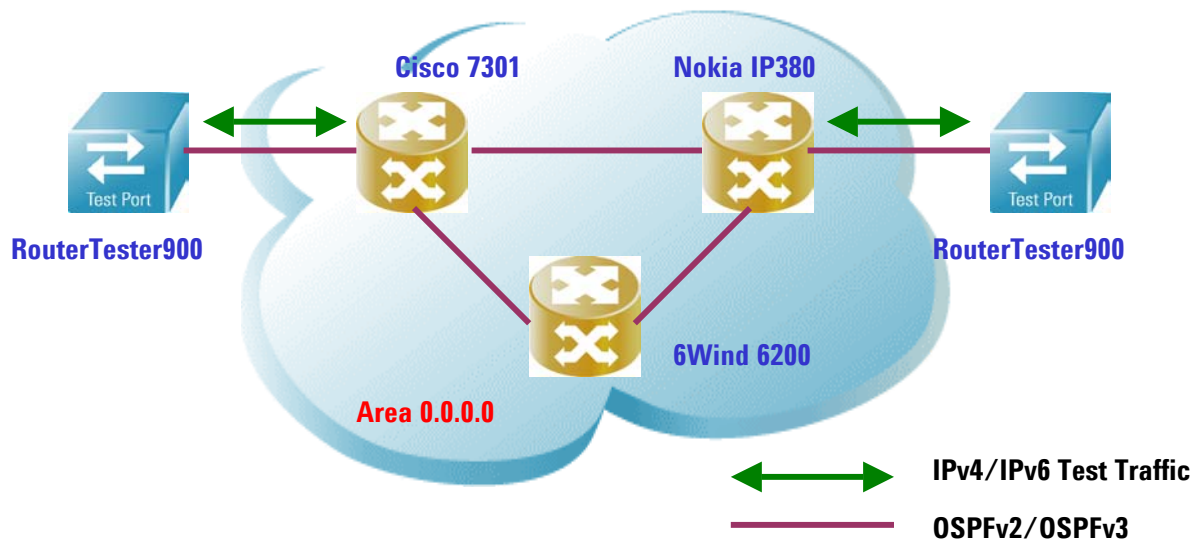
The Moonv6 Phase II event covered the demonstration and verification of a variety of IPv6 functionality, including:

- OSPFv3 testing
- BGP4+ testing
- IPv6 QoS Testing
- Firewall testing

Much of this functionality was initially tested in small, isolated, network topologies, with different vendors interoperating. The results of these initial tests assisted in the design and implementation of the final topology, which included wide area connections to other IPv6 sites. Further testing and end-to-end network evaluation was performed once the UNH edge network was connected to other Moonv6 sites.

## OSPFv3 Testing

The purpose of these tests was to verify basic OSPFv3 functional operation, OSPFv3 convergence, and mixed OSPFv2/OSPFv3 routing protocol operation.

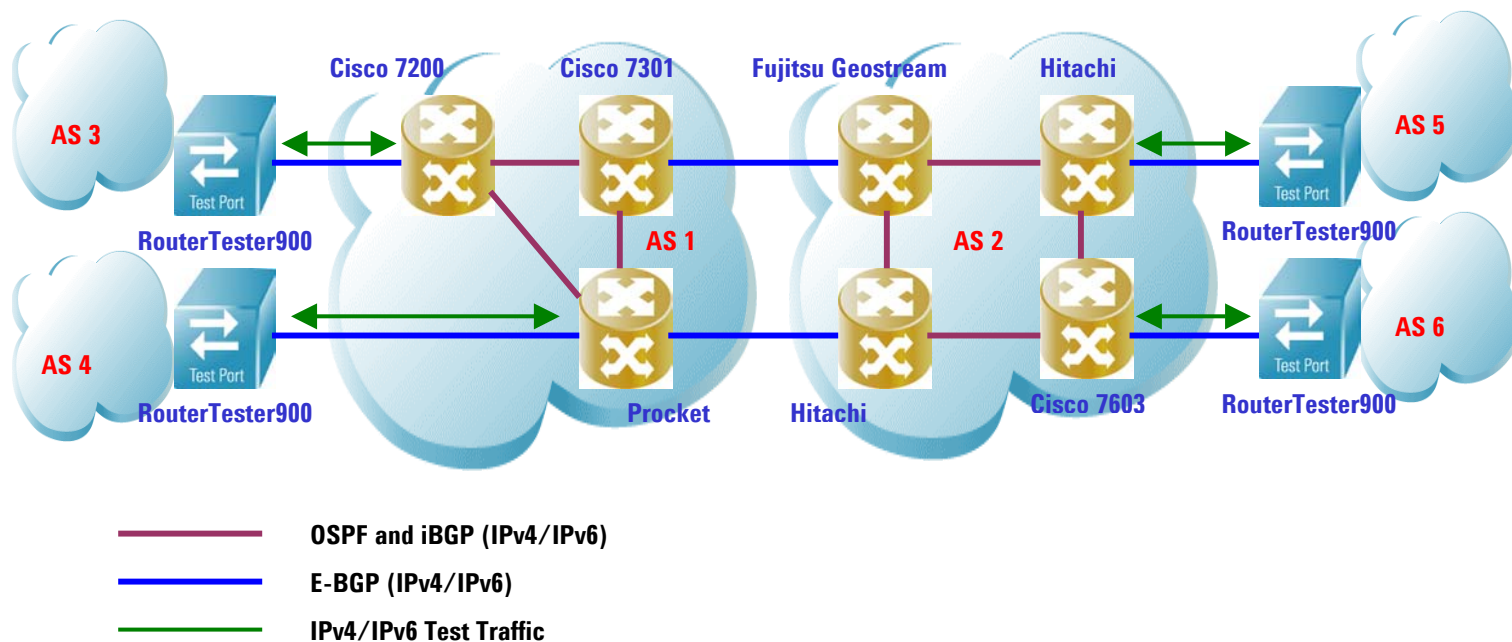


## Tests Executed

1. Basic OSPF functional testing
  - a. Simulate OSPFv3 topologies behind each RouterTester900 test port.
  - b. Advertise OSPFv3 routing topologies to network under test.
  - c. Verify routes updated accordingly in route tables and OSPF databases.
  - d. Send bidirectional IPv6 traffic between RouterTester900 test ports, and collate performance measurements (throughput, latency, loss).
2. OSPFv3 convergence testing
  - a. Configure equal cost links between all routers in network under test.
  - b. Transmit IPv6 test traffic in one direction between RouterTester900 test ports.
  - c. Disconnect or increase the cost on the preferred link.
  - d. Allow network to reconverge, observe traffic reroute on routers and collate measurements on receiving RouterTester900 test port.
  - e. Reconnect or reset the cost of the initially preferred link.
  - f. Allow network to reconverge, observe traffic reroute on routers and collate measurements on receiving RouterTester900 test port.
3. OSPFv2/OSPFv3 functional testing
  - a. Simulate both OSPFv2 and OSPFv3 topologies behind each RouterTester900 test port.
  - b. Advertise OSPFv2 and OSPFv3 routing topologies to network under test.
  - c. Verify routes updated accordingly in route tables and OSPF databases.
  - d. Send bidirectional IPv4 and IPv6 traffic between RouterTester900 test ports, and collate performance measurements (throughput, latency, loss).

## BGP4+ Testing

The purpose of these tests was to verify basic BGP4+ functional operation, BGP4+ convergence, and mixed BGP4 IPv4/IPv6 routing and traffic.



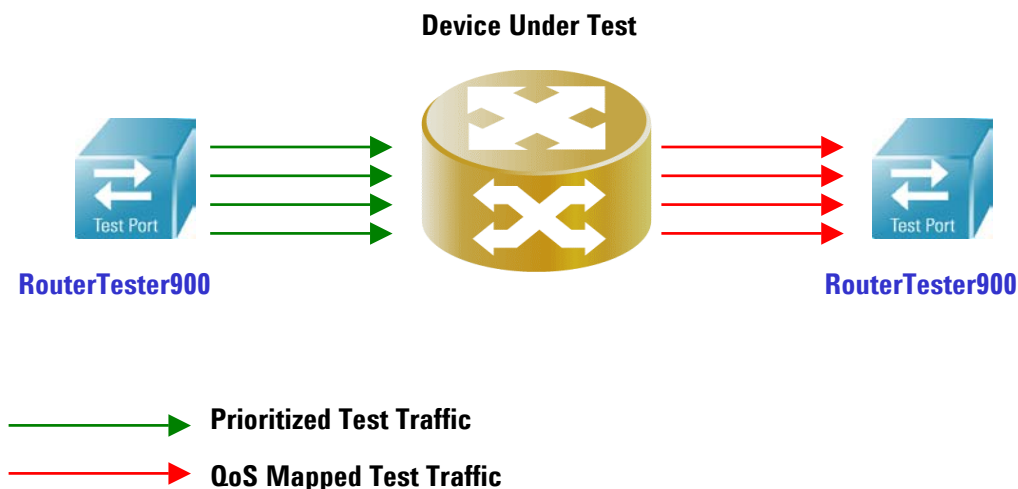
## Tests Executed

1. Basic BGP4+ functional testing
  - a. Simulate IPv6 networks behind each RouterTester900 test port.
  - b. Advertise BGP4 IPv6 prefixes to network under test.
  - c. Verify routes updated accordingly in route tables.
  - d. Send bidirectional IPv6 traffic between RouterTester900 test ports, and collate performance measurements (throughput, latency, loss).
2. BGP4+ convergence testing
  - a. Ensure topology is stable and bidirectional IPv6 traffic is flowing between RouterTester900 ports.
  - b. Disconnect a preferred link for some of the traffic.
  - c. Allow network to reconverge, observe traffic reroute on routers and collate measurements on receiving RouterTester900 test port.
  - d. Reconnect the preferred link.
  - e. Allow network to reconverge, observe traffic reroute on routers and collate measurements on receiving RouterTester900 test port.

3. BGP4+ performance testing
  - a. Simulate both IPv4 and IPv6 networks behind each RouterTester900 test port.
  - b. Advertise BGP4 IPv4 and IPv6 prefixes to network under test.
  - c. Verify routes updated accordingly in route tables.
  - d. Send bidirectional IPv4 and IPv6 traffic between RouterTester900 test ports, and collate performance measurements (throughput, latency, loss) for a long-term duration (i.e. overnight test).

## IPv6 QoS Testing

This section of testing was designed to verify conformance with DiffServ code mapping on routers in a dual IPv4/IPv6 stack network.



## Tests Executed

1. L3/L4 Header Mapping
  - a. From one RouterTester900 test port, advertise N IPv4 and IPv6 network prefixes to the device under test using any routing protocol.
  - b. On the other RouterTester900 test port, setup 2N traffic streams destined for each previously advertised network prefix.
  - c. Setup the device under test to map each IPv4/IPv6 traffic stream pair to a different DSCP value.
  - d. Start traffic generation and observe the QoS mapping at the receiving RouterTester900 test port.
  - e. Repeat test with each IPv4/IPv6 traffic stream pair configured with a different TCP destination port value (i.e. simulating an application)

## 2. DSCP Mapping

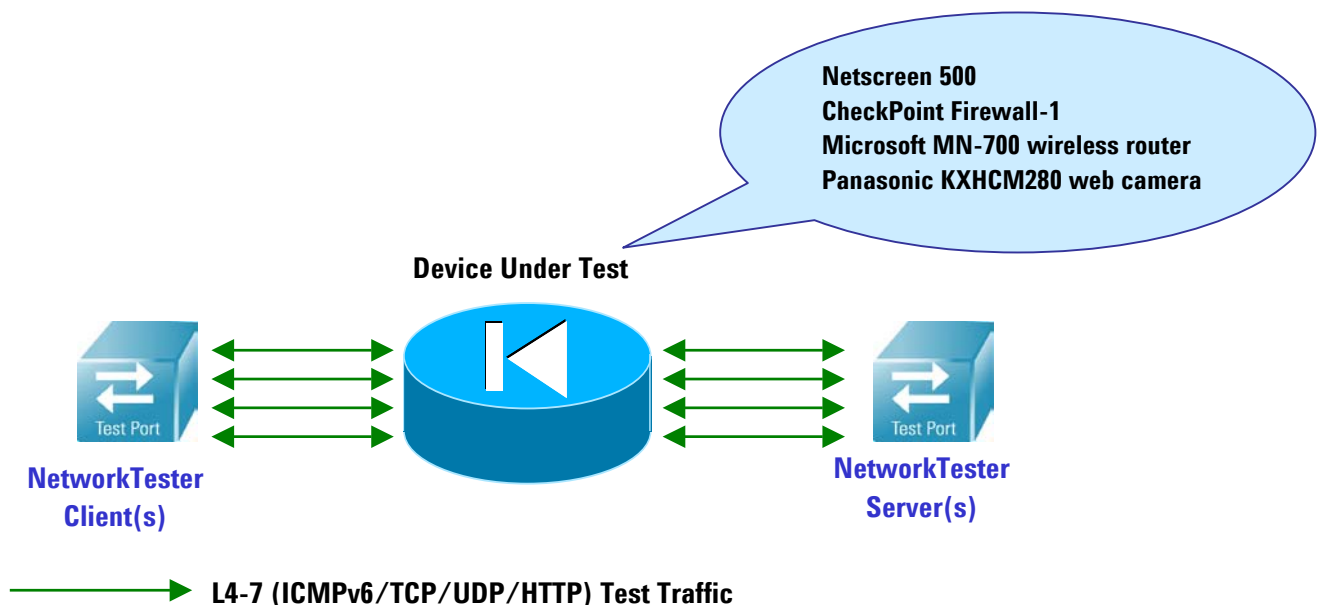
- a. From one RouterTester900 test port, advertise N IPv4 and IPv6 network prefixes to the device under test using any routing protocol.
- b. On the other RouterTester900 test port, setup 2N traffic streams destined for each previously advertised network prefix.
- c. Configure each IPv4/IPv6 traffic stream pair with a different DSCP value.
- d. Setup the device under test to map each IPv4/IPv6 traffic stream pair to a new DSCP value.
- e. Start traffic generation and observe the QoS mapping at the receiving RouterTester900 test port.

## 3. User Priority Bit (VLAN) Mapping

- a. From one RouterTester900 test port, simulate 2N(IPv4/IPv6) hosts on a single VLAN.
- b. Resolve link layer addresses using ARP and NDP.
- c. On the other RouterTester900 test port, setup 2N traffic streams destined for each simulated VLAN host.
- d. Configure each IPv4/IPv6 traffic stream pair with a different User Priority value in the VLAN tag.
- e. Setup the device under test to map each IPv4/IPv6 traffic stream pair to a new DSCP value.
- f. Start traffic generation and observe the QoS mapping at the receiving RouterTester900 test port.

## IPv6 Firewall Testing

This section of testing was designed to verify the basic functionality and operation of IPv6-based access authentication and firewall screening.



## Tests Executed

### Base IPv6 specifications

1. Address Autoconfiguration and Duplicate Address Detection
2. ICMPv6 Echo Requests and Replies
3. TCP/UDP Interoperability

### Basic security policy

1. Source/Destination Address Acceptance
  - a. Configure the DUT to accept traffic from a global IPv6 source address.
  - b. Configure and initiate TCP connections from the NetworkTester client (including the source IPv6 address configured in the previous step) to the NetworkTester server.
  - c. Verify successful establishment of TCP connections.
  - d. Repeat above test with the DUT configured to filter on the global IPv6 source *network* address.
  - e. Repeat above tests, now configuring a range of IPv6 source addresses for the TCP traffic.
  - f. Repeat above tests, now based on the IPv6 destination address.
2. Source/Destination Address Denial
  - a. Same as 1. above, except the DUT is configured to deny specified traffic.
3. UDP Port Numbers
  - a. Configure the DUT to accept all traffic destined for a UDP source port.
  - b. Configure and initiate UDP traffic from the NetworkTester client (including the UDP source port configured in the previous step) to the NetworkTester server.
  - c. Verify UDP traffic successfully received on NetworkTester server.
  - d. Repeat above test with the DUT configured to deny specified UDP traffic.
  - e. Repeat above tests, now based on the UDP destination port.
4. TCP Port Numbers
  - a. Configure the DUT to accept all traffic destined for a TCP source port.
  - b. Configure and initiate TCP connections from the NetworkTester client (including the TCP source port configured in the previous step) to the NetworkTester server.
  - c. Verify successful establishment of TCP connections.
  - d. Repeat above test with the DUT configured to deny specified TCP connections.
  - e. Repeat above tests, now based on the TCP destination port.
5. ICMPv6 Traffic
  - a. Verify DUT policy configuration to accept all ICMPv6 traffic.
  - b. Verify DUT policy configuration to deny all ICMPv6 traffic.

- c. Verify DUT policy configuration to deny ICMPv6 Echo Requests, accept all other ICMPv6 messages.
  - d. Verify DUT policy configuration to deny ICMPv6 Echo Replies, accept all other ICMPv6 messages.
6. Time Based Authorization
- a. Verify DUT policy configuration to deny TCP connections with a global source IPv6 address for a specific time (i.e. 2 minutes).
  - b. Repeat above test with DUT policy based on global source IPv6 *network* address, global destination IPv6 address and global destination IPv6 *network* address.
7. Combination Authorization
- a. Verify DUT policy configuration to deny TCP connections with a specific source/destination IPv6 address pair, for a specific time (i.e. 2 minutes)
  - b. Repeat above test with DUT policy based on (not limited to):
    - i. UDP source port, source IPv6 address, destination IPv6 address
    - ii. TCP source port, source IPv6 address, destination IPv6 address
    - iii. ICMPv6 Echo Request, source IPv6 address, destination IPv6 address
8. Denial of Service Attacks
- a. Verify DUT correctly identifies and blocks a TCP SYN flood attack.
  - b. Verify DUT correctly identifies and blocks a UDP flood attack.
  - c. Verify DUT correctly identifies and blocks an ICMPv6 flood attack.
  - d. Repeat above tests, and verify DUT can continue to allow authorized TCP connections once a flood attack is detected.
9. Ordered List Policy
- a. Configure DUT with two rules, first rule to deny traffic on a specific TCP destination port, and second rule to accept all TCP connections.
  - b. Configure and initiate TCP connections from the NetworkTester client (including the TCP destination port configured in the previous step) to the NetworkTester server.
  - c. Verify TCP connections are denied.
  - d. Repeat the above test, with the order of the two rules swapped in the DUT policy.
  - e. Verify TCP connections are now successfully established.
10. Stateful Inspection
- a. Verify DUT blocks IPv6 TCP traffic received out of state:
    - i. Verify DUT blocks the reception of a SYN-ACK before a SYN
    - ii. Verify DUT blocks the reception of an ACK before a SYN or SYN-ACK
    - iii. Verify DUT blocks the reception of all out of state TCP control messages (SYN-ACK, ACK, URG, PSH, FIN, FIN-ACK, RST)

## Deep Packet Inspection

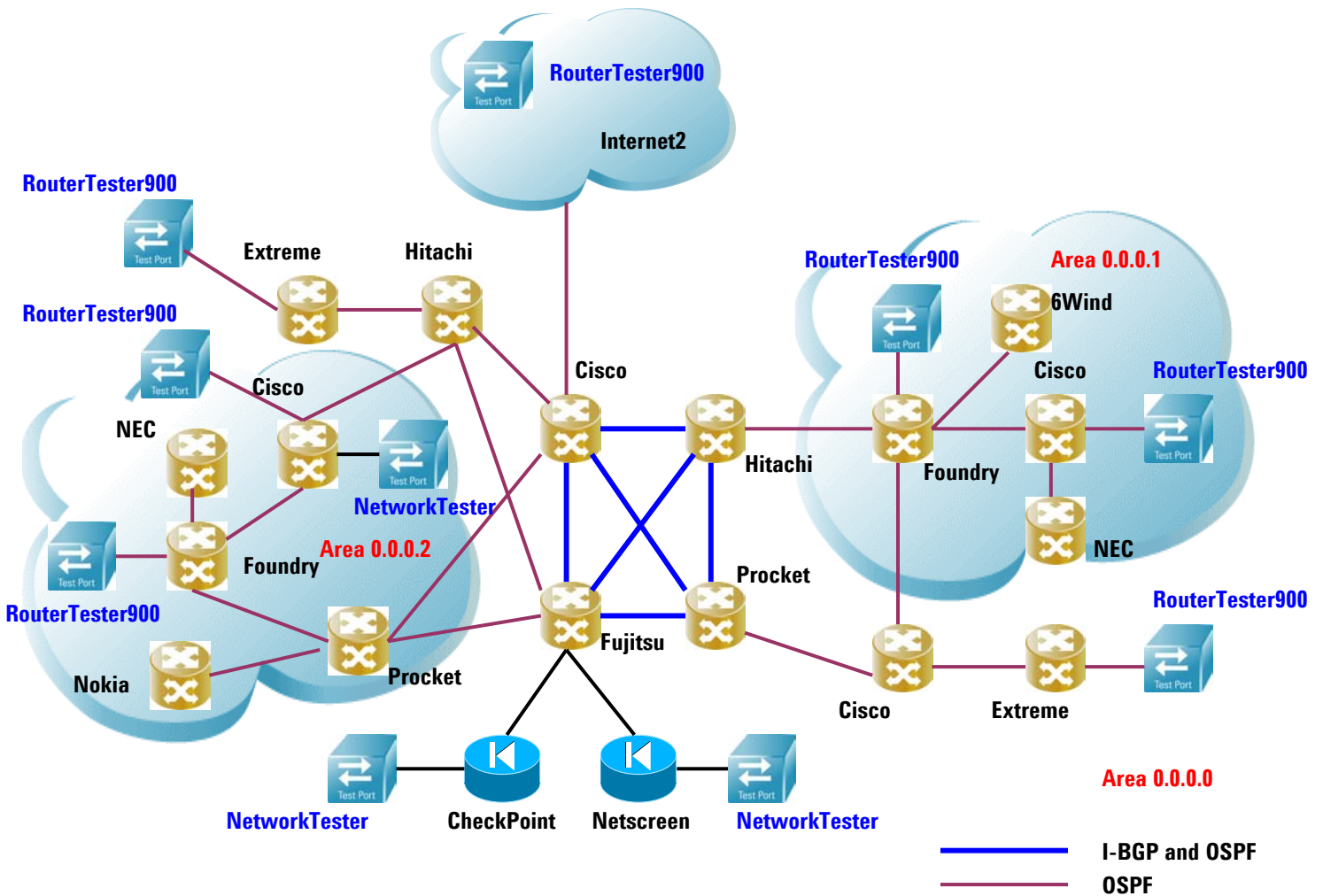
1. URL filtering
  - a. Verify DUT can block HTTP GET requests to long URLs.

- b. Verify DUT can block extended-ASCII characters received inside HTTP GET requests.
- c. Verify DUT can block HTTP GET requests to specific URLs.

### Firewall Performance

- 1. Verify maximum number of concurrent TCP connections supported by the DUT.
- 2. Verify maximum number of TCP connections/second supported by the DUT.
- 3. Verify a configured maximum number of concurrent TCP connections are operational on the DUT.
  - a. Configure DUT to allow TCP connections a specified time after configured threshold is breached.
  - b. Verify TCP connections can be re-established after timer expires.

### Final IPv6 Topology at UNH



The above diagram illustrates the final IPv6 topology commissioned at the UNH-IOL Moonv6 site. RouterTester900 and NetworkTester test ports were deployed at various endpoints through the local network, as well as also being located at a remote Moonv6 in Ft. Huachuca, AZ - accessed through Internet2.

RouterTester900 was used extensively for debugging during the commissioning of this local topology. With connectivity to remote Moonv6 sites, end-to-end network packet throughput and packet loss tests were performed using RouterTester900. Simulated OSPF topologies were advertised into the final topology and OSPF route flap convergence tests were also executed.

With a stable local topology, NetworkTester generated and measured HTTP traffic across the network infrastructure, between client and server test ports.

## **References**

Moonv6 Phase II Test Plans: [http://moonv6.sr.unh.edu/project/test\\_plans\\_phaseII.html](http://moonv6.sr.unh.edu/project/test_plans_phaseII.html)