ixia

Testing L7 Traffic Shaping Policies with IxChariot

IxChariot

Contents

Ixia
26601 West Agoura Road, Calabasas, CA 91302
(877) FOR-IXIA

This Test Plan Primer contains a general outline for testing a particular technology. Not all the capabilities of Ixia technology have been exposed in this document. Please feel free to contact us if additional capabilities are required.

1. Introduction

Prioritized access to bandwidth on congested WAN and Internet access links is increasingly managed by a set of dedicated traffic shaping devices or firmware features in higher-end routing products. The critical success criterion for these devices is to enforce policies at the application level, and, increasingly, at Layer 7. This allows network performance to be ensured for mission-critical applications while pacing network access for recreational applications and sessions (e.g., Yahoo Messenger and private web surfing). Mapping each type of traffic to a specific bandwidth allocation policy ensures that each traffic type receives appropriate bandwidthand that malicious activity is more effectively blocked.

Many of these traffic shaping systems offer features to specify bandwidth minimums and/or maximums for a seemingly infinite number of applications, sessions, users, IP addresses, port numbers, and other traffic subsets. This level of discrete control is essential. Using web traffic as an example, simply managing it by HTTP protocol and port number is no longer sufficient. Instead, intelligent L7 payload inspection functionality allows the administrator to devise traffic prioritization and access control policies based on the destination of the HTTP GET request.

The new script-embedded payload (SEPL) functionality released in IxChariot 6.0 was designed specifically to test the scalability of traffic shaping devices under load. With SEPL, you can easily define tens of thousands of L7 payloads that these devices will inspect and then implement the appropriate bandwidth allocation policies. In addition, the Denial of Service tests in IxChariot also provide concrete evidence on the ability of the System Under Test (SUT) to handle common attack traffic while maintaining required network performance for critical applications.

2. Overview of Layer 7 Payload Definition Options in IxChariot

With the introduction of script-embedded payloads in release 6.0, there are now two key options used to define the L7 payload in IxChariot.

It is important to note that in the IxChariot testing approach, L7 payloads are defined as all data that is encapsulated by TCP, UDP and RTP, using either IPv4 or IPv6 in the network layer.

2.1 Script-Embedded Payloads (SEPL)

The new Script Embedded Payloads (SEPL) feature in IxChariot offers a new way to define a specific L7 payload for every “SEND” command in an IxChariot script. A number of scripts in the “Internet” library of IxChariot have been converted to ship with an embedded L7 payload by default. As shown in the screen below, the ”SEND type” command specifies the “Embedded payload.”

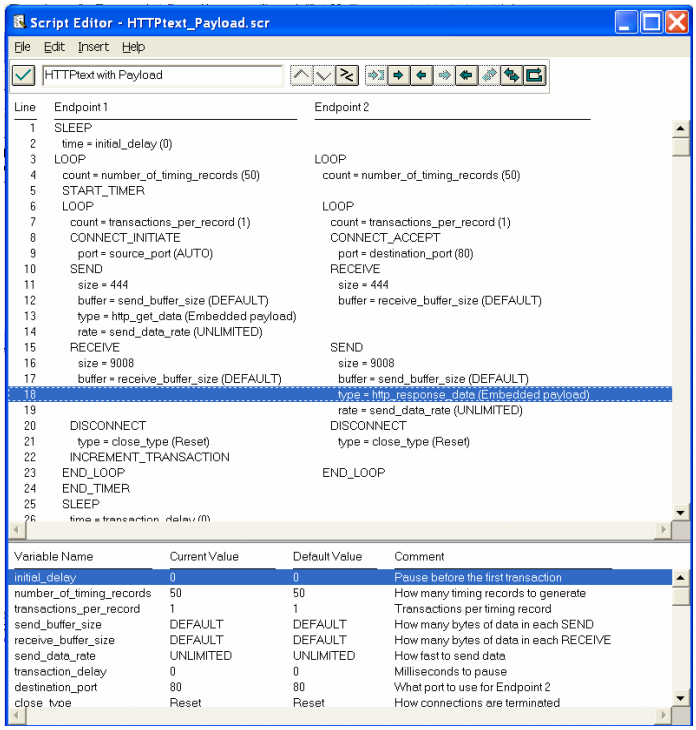

Figure 1: Script Editor dialog of HTTPtext_Payload script

Clicking on the “type” line in the “SEND” command opens the payload definition dialog box. You can use this option to define the L7 payload by either typing it in the provided field or by importing a binary or text file.

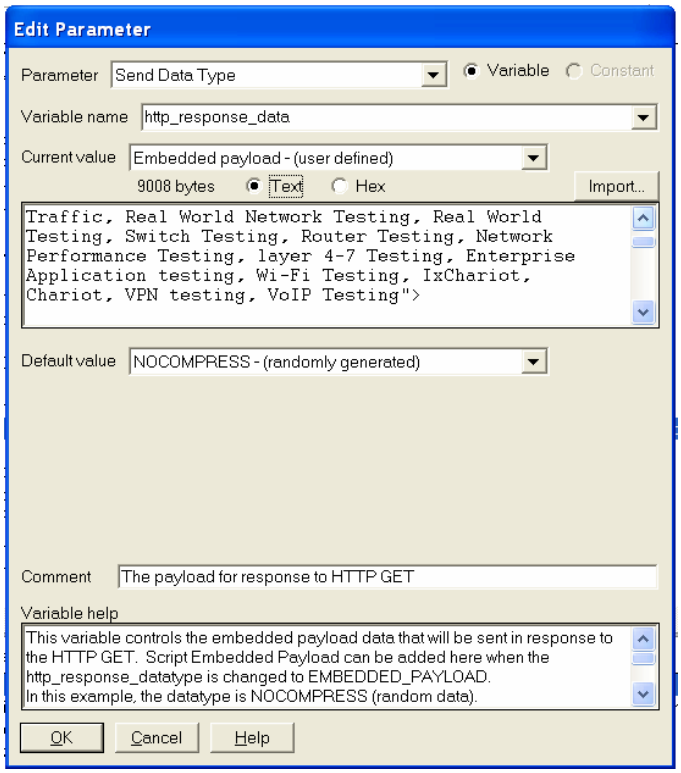

Figure 2: Definition dialog for custom L7 payload

Pressing the import button will open a dialog box where you can select a file (e.g., text file) and then create a slice of the file by changing the offset and/byte count to be included in the SEND command.

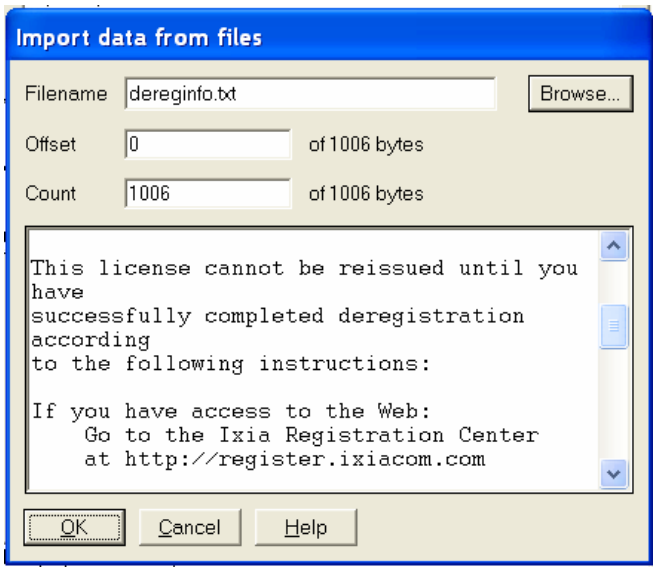

Figure 3: Import file dialog with sample text file

Note: SEPL payloads are limited to 27000 bytes.

2.2 Attaching Custom Compression Files (.cmp)

In addition to the "Embedded Payload" option in the SEND command of every IxChariot, there are also a number of pre-configured compression files (e.g., news.cmp, lena.cmp) that you can use to define a L7 payload. IxChariot lets you define up to ten different files (e.g., userXX.cmp) to be associated with every SEND command. Though these custom files need to be pre-loaded into the same host running the Performance Endpoint, their advantage lies in the fact that payload sizes for these files can extend to many megabytes.

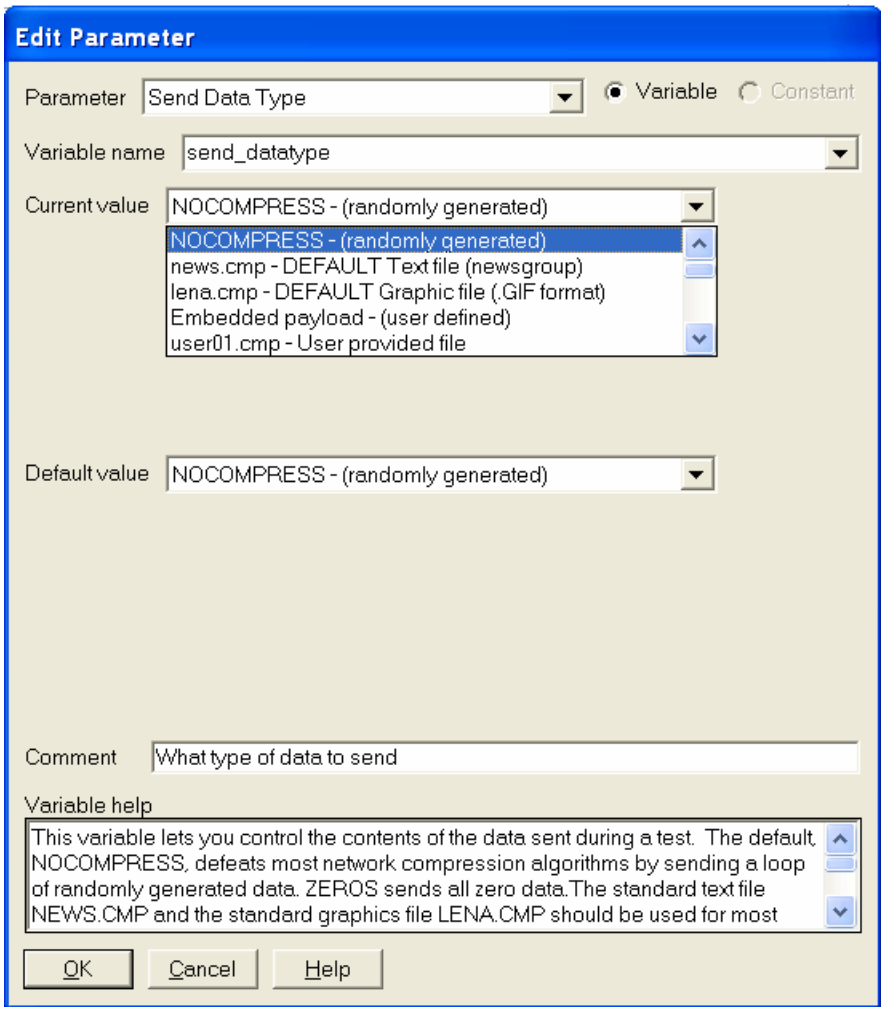

Figure 4: Dialog option to define pre-loaded or custom L7 payloads using .cmp files

3. Principal Test Cases for L7 Traffic Prioritization

3.1. Setup

A minimum of two Ixia ports are required with L2-7 traffic generation capabilities and/or server-based ports running the Performance Endpoint appropriate for the server operating system. These ports are physically connected to the "Inside" and "Outside" ports of the SUT, respectively. The IxChariot console can be run either directly on the Ixia chassis or separately on a workstation connected to the chassis' management port.

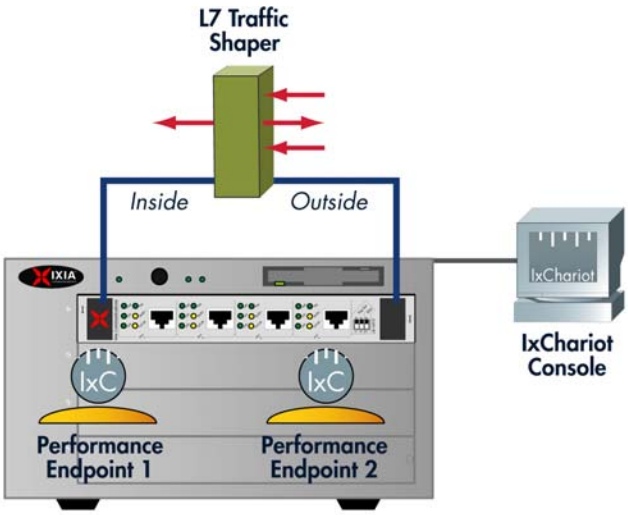

Figure 5: Setup for bracketing L7 traffic-shaping device under test by two Ixia ports running Performance Endpoint for IxOS

The following tests are based on using Performance Endpoints for IxOS.

3.2. Test Methodology

The test should involve as many ports running IxChariot Performance Endpoints as there are test ports accessing the network and/or System Under Test. In this case, there is a single ingress/egress port pair, so multiple interfaces are aliased on the Ixia ports to allow the creation of multiple pairs with different source/destination IP addresses. Tests are then executed with run options set to a fixed time.

Stateless traffic is generated using dedicated Ixia hardware FPGAs; whereas, stateful IxChariot traffic is generated using the port CPUs resident on the same ports of the Ixia Load Modules.

3.3. Test Case 1

Objective

The objective of the following test scenario is to establish a baseline throughput measurement of stateful traffic through the SUT without enabling any traffic prioritization policies.

Input Parameters

The primary input parameters for this test include:

- IPv4 addresses configured on the Ixia ports running the Performance Endpoints for IxOS to create one pair (e.g., 172.30.30.1 and 172.30.31.1)
- IxChariot scripts – "Throughput" and "HTTPtext_Payload"
- Network protocol – TCP
- Endpoint 1 (E1) Setup addresses (i.e., management port IPv4 addresses of Ixia port)
- Run Option set to a fixed duration of one minute

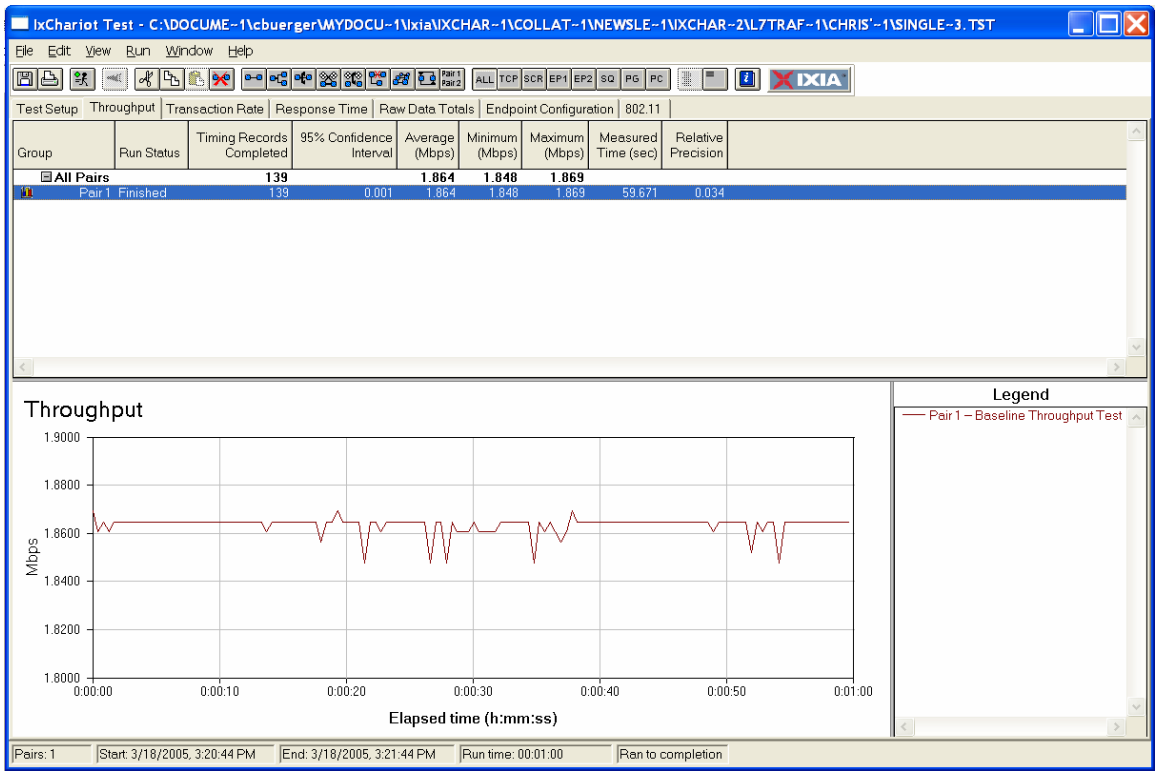

Figure 6: Baseline test result through the SUT with no L7 traffic-shaping policies using "Throughput" script

The test above relied on the standard "Throughput" script – a benchmark script in the IxChariot script library optimized to measure the L7 goodput (i.e, not including any header information) of a device and network. In this case, the average throughput by IxChariot was a little less than 1.9 Mbit/s, which is within range of the SUT's stated 2Mbit/s limit.

Since most of the following tests that exercise the L7 payload identification and shaping schemes of the SUT are based on the HTTPtext_Payload script in IxChariot, it is recommended that you re-run the same test again with the HTTPtext_Payload script to determine the baseline throughput for a pair running this particular script.

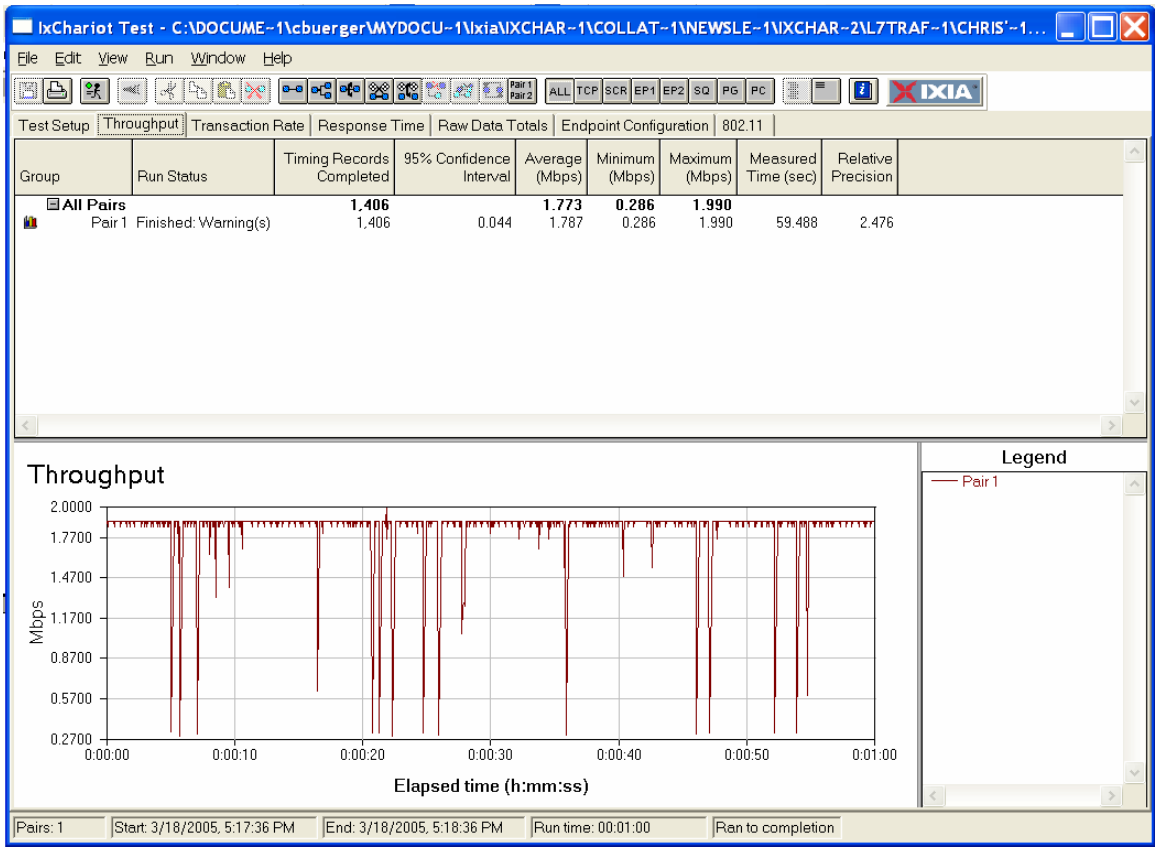

Figure 7: Baseline test result through the SUT with no L7 traffic-shaping policiesusing default “HTTPtext_Payload” script

In some test scenarios, it may make sense to overload the SUT by generating a high line rate of stateless traffic to determine whether prioritization policies are still accurately enforced. If this is the case, it is recommended that you determine a baseline throughput result for this traffic type as well.

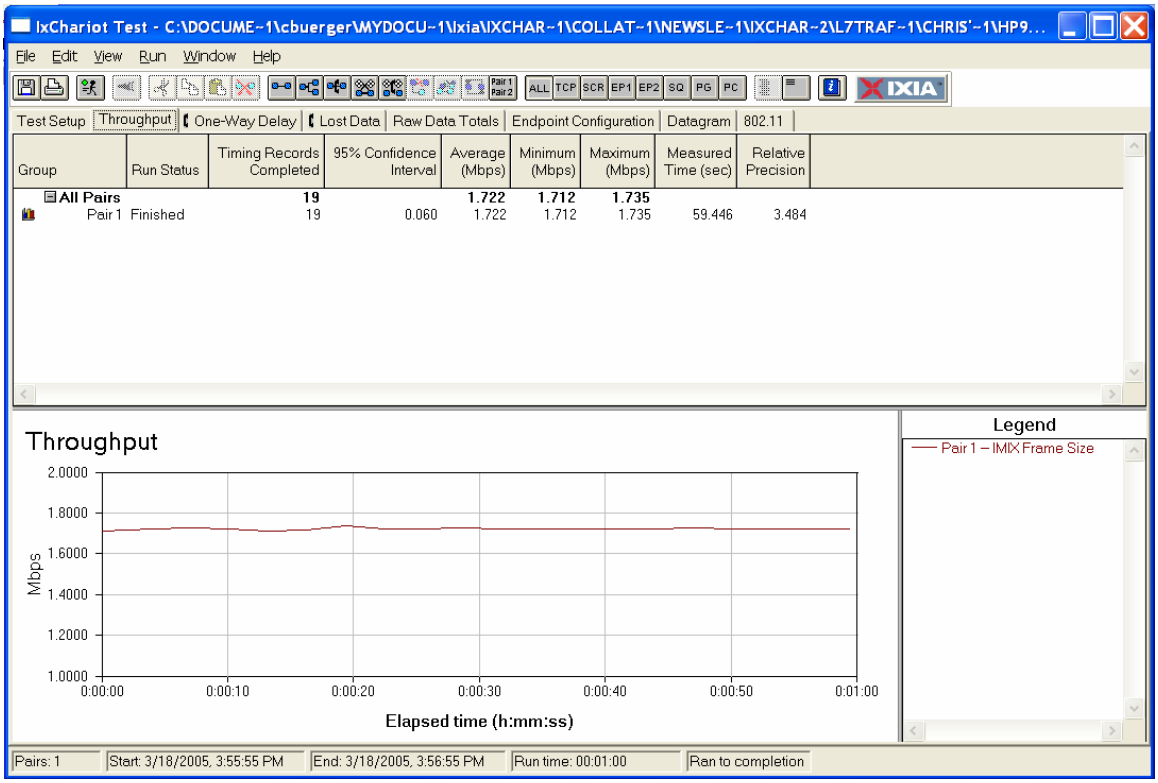

Figure 8: Baseline test result through the SUT with no L7 traffic-shaping policies using default "HTTPtext_Payload" script

3.4. Test Case 2

Objective

The objective of the following test scenario is to test the traffic-shaping functionality and performance of the SUT based on three different values defined in the HTTP payload. Each script and payload value has a corresponding policy defining the maximum bandwidth in kbit/s set in the SUT.

Input Parameters

The primary input parameters for this test include:

- IPv4 addresses configured on the Ixia ports running the Performance Endpoints for IxOS to create three and nine pairs using the same IP addresses (e.g., 172.30.30.1 and 172.30.31.1) and sequential IP addresses (e.g., 172.30.30.1-3 and 172.30.31.1-3), respectively
- A modified IxChariot "HTTPtext_Payload" script to include payload values that have an associated policy in the SUT. The policies used for this example include:
  - "Priority_0" = max. rate of 100 kbit/s
  - "Priority_1" = max. rate of 200 kbit/s
  - "Priority_2" = max. rate of 300 kbit/s

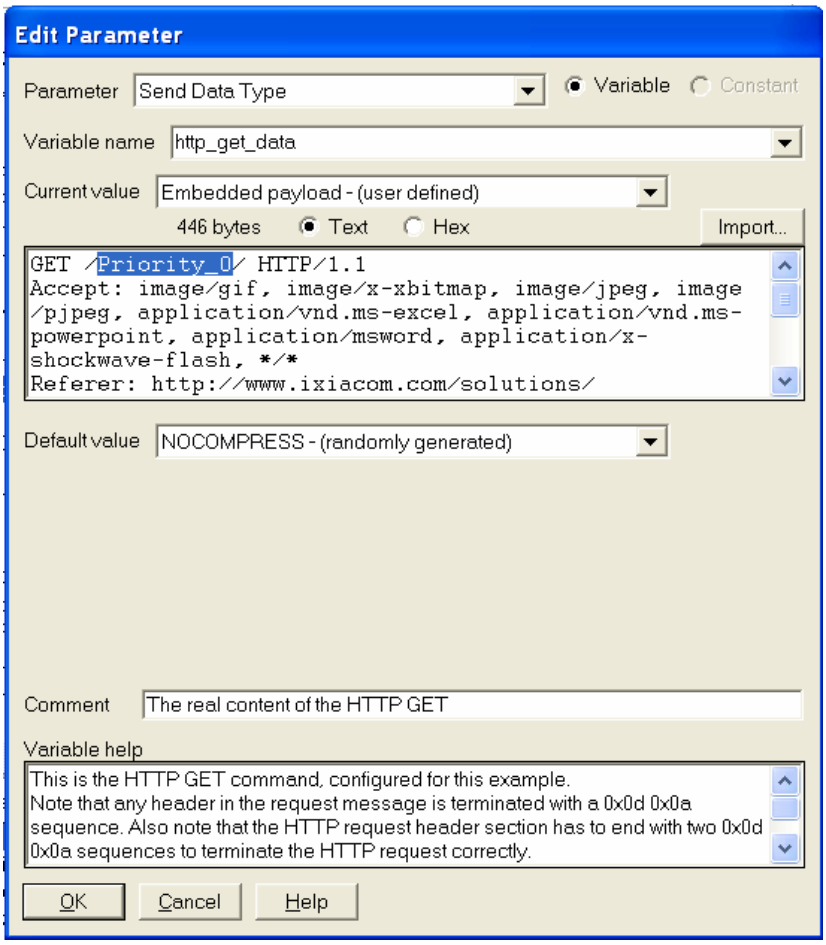

Figure 9: L7 payload definition for traffic policies

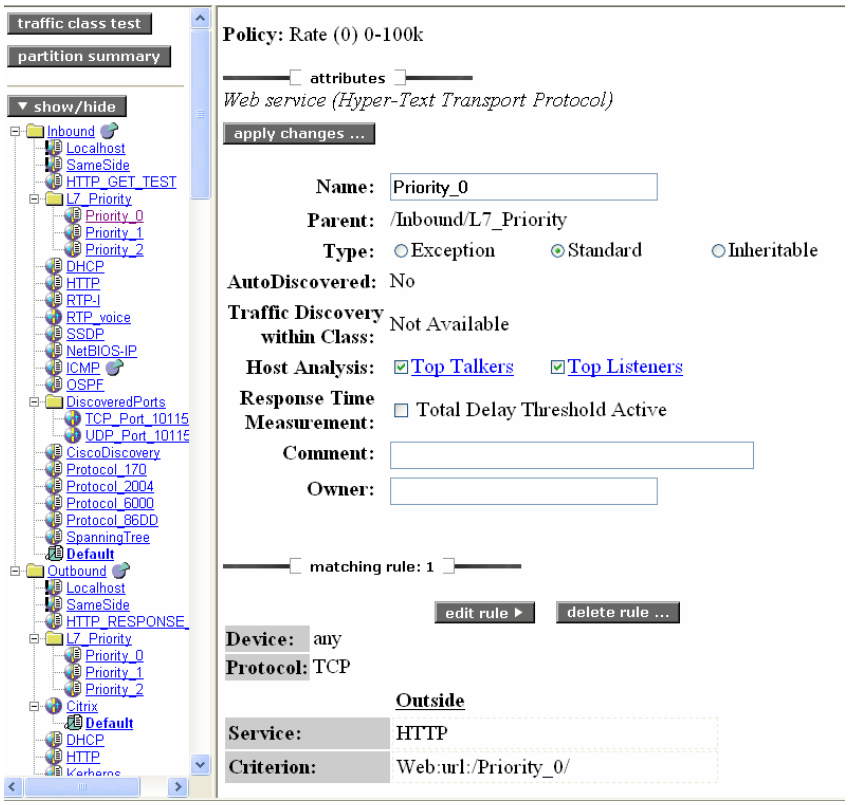

Figure 10: Matching rule ("Rate Limit") definition in the SUT

- Network protocol – TCP
- Endpoint 1 (E1) Setup addresses (i.e., management port IPv4 addresses of Ixia port)
- Run Option set to a fixed duration of one minute

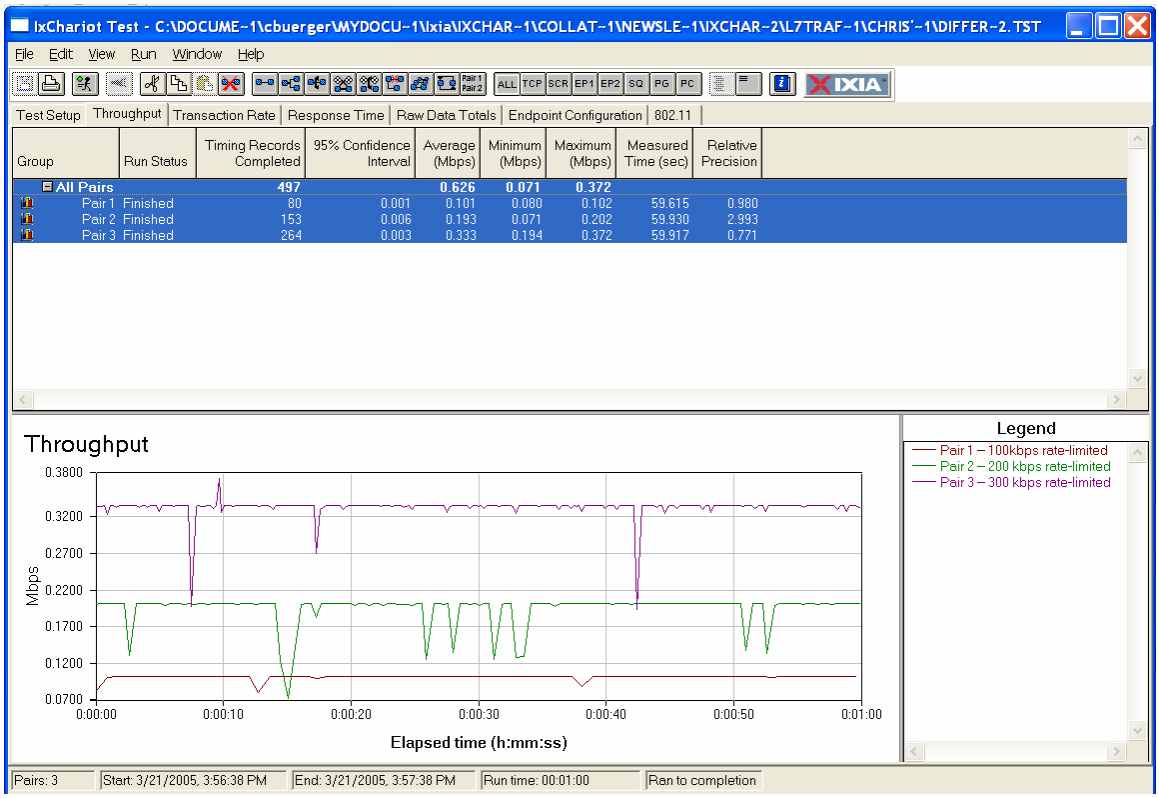

Figure 11: Results for three pairs with three matching maximum bandwidth rules

As can be seen in the above results, the SUT had no problems recognizing the payload values and enforcing the appropriate policy on a per-pair basis. However, the throughput deviation from the defined maximum value for pair 3 (i.e., 300 kbit/s) is significantly higher than for pairs 1 and 2. See average and maximum throughput values for pair 3.

Scaling the test to running a total of nine pairs representing three different sets of source and destination IP addresses reinforces this observation.

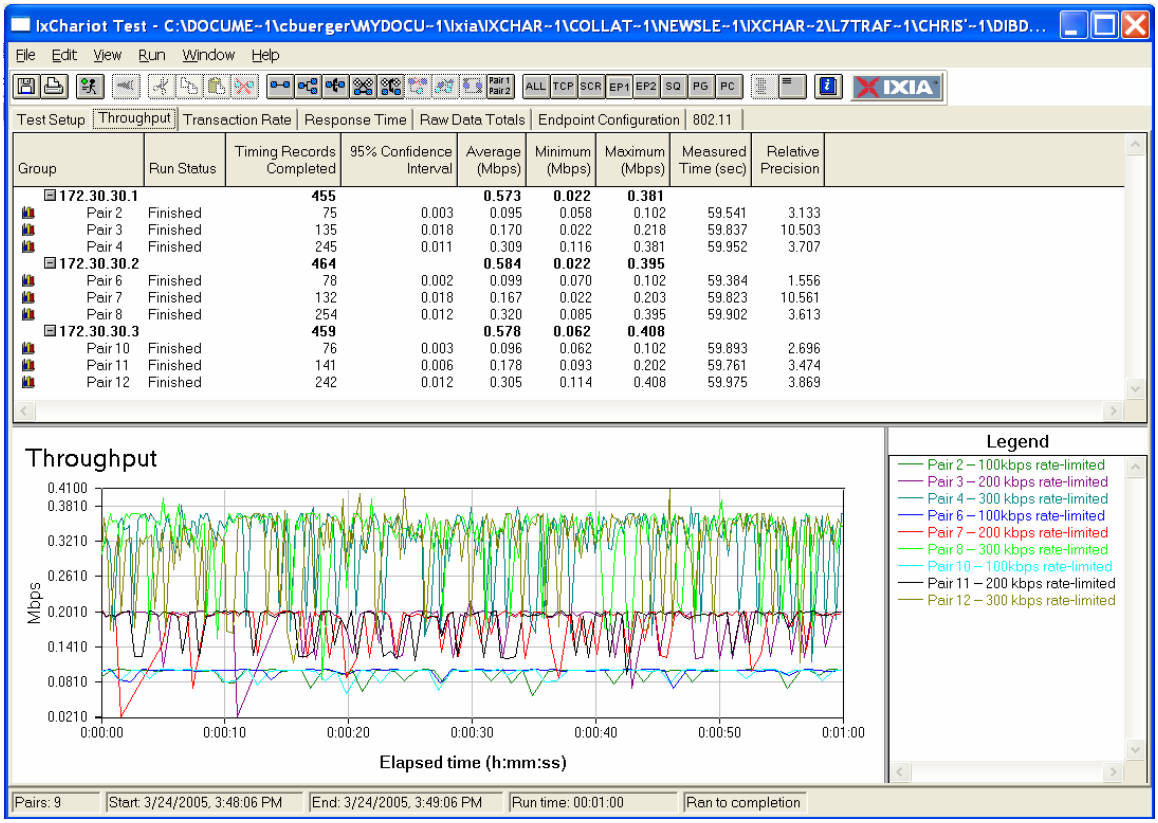

Figure 12: Results for nine pairs with three matching maximum bandwidth rules

IxChariot offers a number of ways to represent data to crystallize further points of investigation. Displaying the data on a per pair-group basis can help you measure whether the aggregate maximum throughput for three pairs using the same matching rule but different IP addresses is within the expected range (e.g., a maximum of three times the maximum bandwidth for the individual pair).

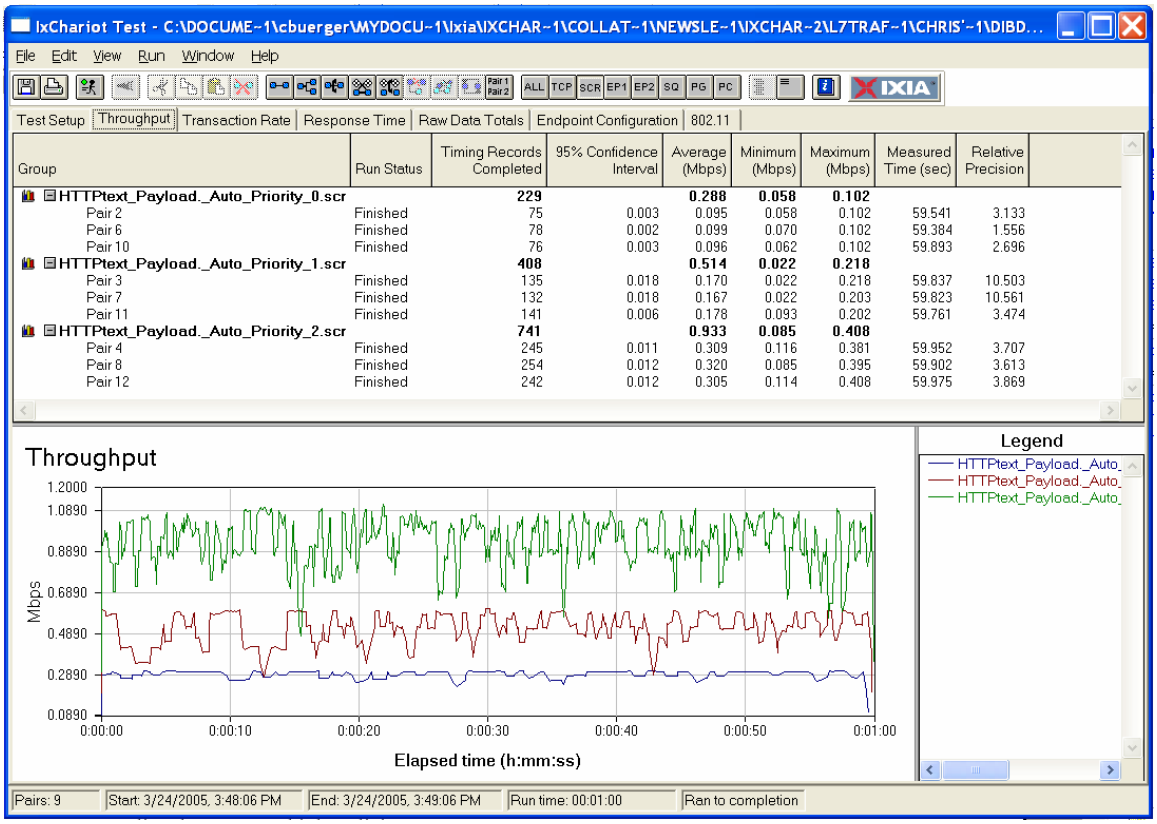

Figure 13: Results for nine pairs grouped by script name, with each script name representative of a different payload value

3.5. Test Case 3

Objective

The objective of the following test scenario is to test the traffic-shaping functionality and performance of the SUT based on three different values defined in the HTTP payload. Each test script and payload value has a corresponding policy defining the traffic's priority versus other traffic running through the SUT.

Input Parameters

The primary input parameters for this test include:

- IPv4 addresses configured on the Ixia ports running the Performance Endpoints for IxOS to create three pairs using the same IP addresses (e.g., 172.30.30.1 and 172.30.31.1)
- IxChariot "HTTPtext_Payload" script modified to include payload values that have an associated policy in the SUT. The policies used for this example include:
  - "Priority_0" = Priority 0 (lowest priority)
  - "Priority_1" = Priority 1
  - "Priority_2" = Priority 2

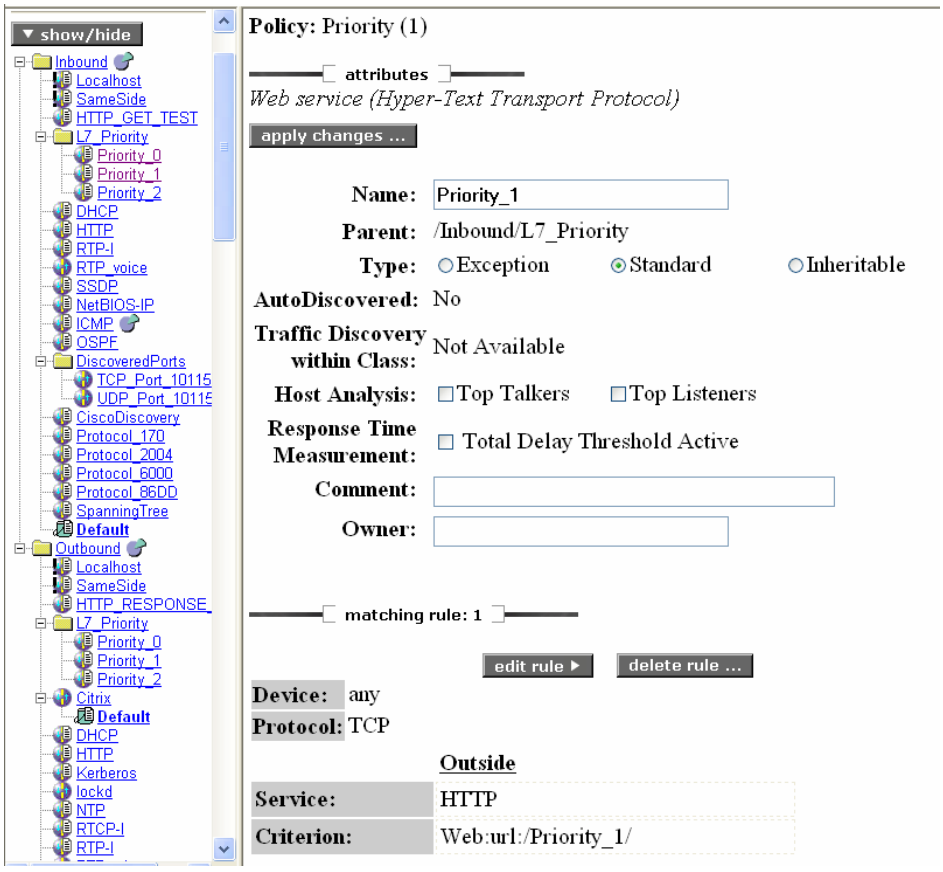

Figure 14: Matching rule ("Priority") definition in the SUT

- Network protocol – TCP
- Endpoint 1 (E1) Setup addresses (i.e. management port IPv4 addresses of Ixia port)
- Run Option set to a fixed duration of one minute

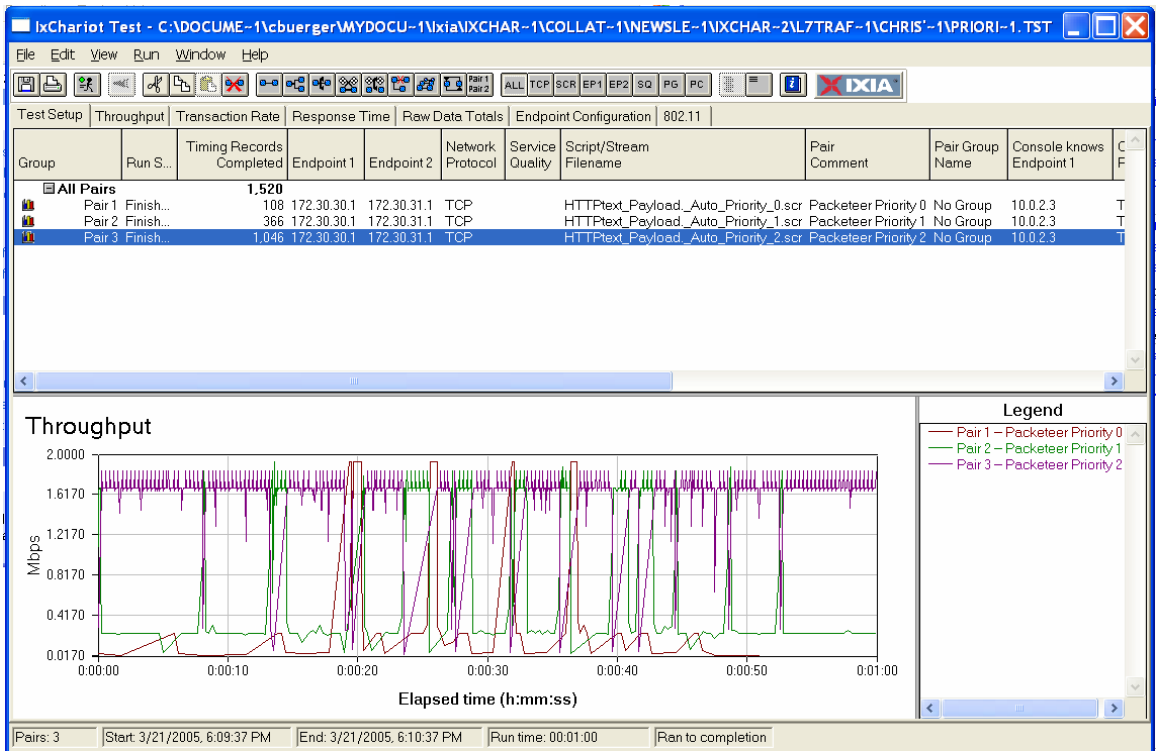

Figure 15: Results for three pairs with three matching priority rules

As shown in the above results, the SUT has no difficulty recognizing the payload values and enforcing the appropriate policy determining the relative priority of the data on a per-pair basis. Interestingly, the aggregate throughput for all three pairs is significantly lower than the baseline throughput measured in Test Case 1. This indicates that the "cost" of adding traffic shaping rules to overall throughput can be substantial.

3.6. Test Case 4

Objective

The objective of the following test scenario is to test the traffic-shaping performance of the SUT when a mixture of stateful and stateless IxChariot traffic is sent through the device. Stateful traffic can take the form of generic background traffic (e.g., IMIX) or Denial of Service (DOS) attack traffic.

Input Parameters

The primary input parameters for this test include:

- IPv4 addresses configured on the Ixia ports running the Performance Endpoints for IxOS to create four pairs using the same IP addresses (e.g., 172.30.30.1 and 172.30.31.1)
- IxChariot "HTTPtext_Payload" script modified to include payload values that have an associated policy in the SUT. The policies used for this example include:
  - "Priority_0" = max. rate of 100 kbit/s
  - "Priority_1" = max. rate of 200 kbit/s
  - "Priority_2" = max. rate of 300 kbit/s
- IxChariot hardware performance pair streams to generate stateless FPGA-driven traffic. Stream types used in the following test cases include:
  - IPv4_IMAX running at 2% line rate of the Ixia Load Module
  - IPv4_PingFlood running at 100% line rate of the Ixia Load Module
  - IPv4_SynFlood_port80 running at 2% line rate of the Ixia Load Module
- Network protocol for script pairs – TCP
- Endpoint 1 (E1) setup addresses (i.e., management port IPv4 addresses of Ixia port)
- Run option set to a fixed duration of one minute

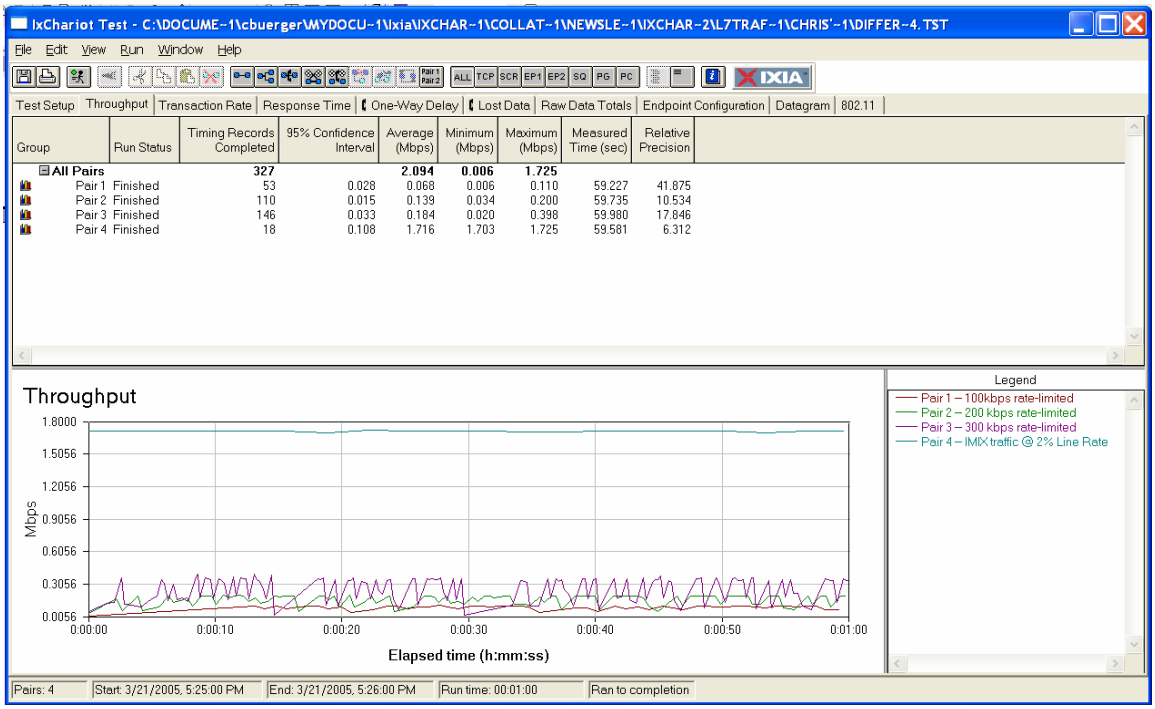

Figure 16: Results for three script pairs with matching maximum bandwidth rules and IMIX background traffic

As illustrated in the above results screen, adding stateless traffic to the mix of data sent across the SUT causes the stateful traffic to back off and show lower throughput than in Figure 11. The decrease in the average throughput (approximately 30%) is spread evenly across all stateful traffic pairs.

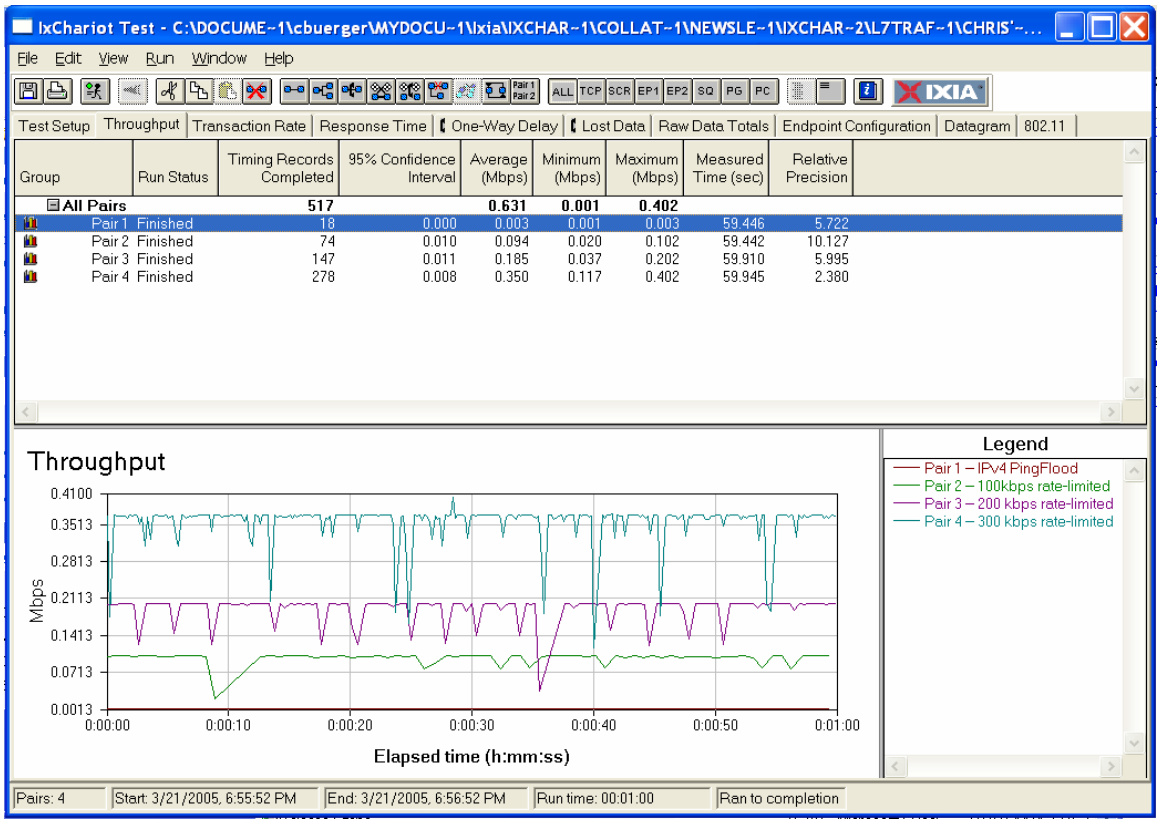

Figure 17: Results for three pairs with three matching bandwidth rules and Ping Flood traffic

This test shows that after identifying the Ping Flood traffic, the SUT throttles the ICMP traffic to have no impact on the throughput of any of the other pairs (2,3 and 4) running stateful traffic with matching traffic rules.

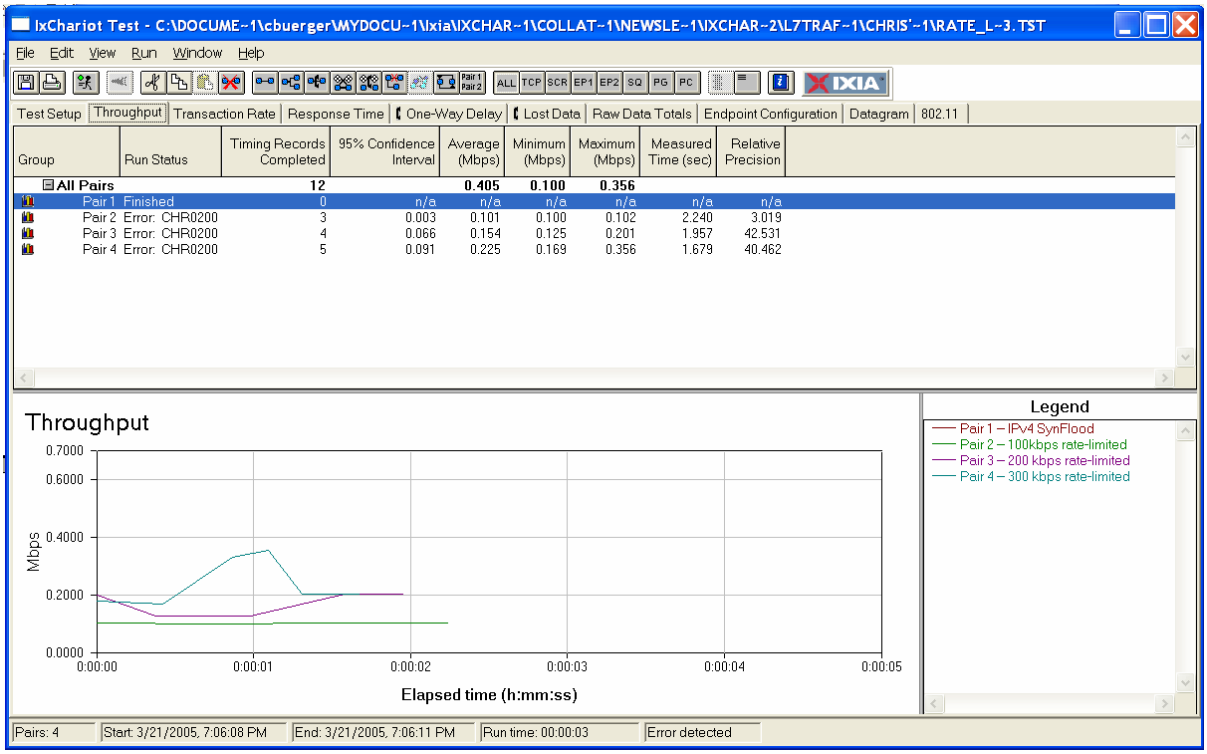

Figure 18: Results for three pairs with three matching bandwidth rules and TCP Syn Flood traffic

Care needs to be takes to not exceed the capabilities of the SUT, however. As shown in the above results screen, the SUT is not able to handle a low line rate (2%) stream of TCP Syn attacks for more than two seconds, consequently shutting down the link and causing IxChariot TCP timeout errors on the stateful traffic pairs.