NetLoad Inc.

11001...1101000..1000.10011...11100...1001

User Guide

Revision 8.9

User Guide for NetLoad Product Series

www.NetLoadInc.com

Table of Contents

Table of Figures

Copyright Notice

System Setup (Appliance Only)

This section provides the overview of the steps to physically install, configure, and use the product.

Step	Description
Step 1	Unpacking the system
Step 2	Installation
Step 3	Access and Configuration

Step 1: Unpacking the system

1. Remove and unpack the contents from the physical container
2. Examine the system and all components for physical damage
3. If no damage is visible proceed to Step 2

Step 2: Installation

1. Place the device on a sturdy surface or install into a 19” rack.
2. Do not block any airflow vents for proper operation and cooling. The system uses side and back vents for cooling.
3. Insert the Ethernet CAT5 cable into port MGMT0, and the other end into your Laptop or PC, or your network. “63N” systems have MGMT1 port, while “68N” do not.
4. Insert the power cable into device and then into the power outlet. The switch in the back of the system turns on the system. For “68N” Systems, dual redundant power supplies are provided.
5. “63N” system has copper Nx10/100/1G (dual copper/Fiber combo is available upon special request). “68N” has Nx10G Fiber ports.
6. On a single-system setup 4-port setup, the first 2 ports (non-Management) from left to right (Eth0 and Eth1) are Server ports, and the last 2 ports (Eth2 and Eth3) are Client ports. The ports are paired “Eth0-Eth2” and “Eth1-Eth3”. For multi-box configurations all ports can be configured as Servers or Clients. The ports are used as Server—Client pairs for testing the DUT. Appropriate copper cables (CAT5) or fiber (LC multi-mode 62.5/125uM – usually in orange color) should be used. If the user changes the pluggable transceivers on the fiber ports to single mode, appropriate matching fiber cables (LC single mode – usually in yellow color) should be used to connect to the DUT.

Step 3: Access the Configuration

1. Please make sure your PC has an IP address configured on the same subnet as the NetLoad device.
2. Access the device using your browser by entering http://192.168.1.1

Figure 1 -- Login

3. Please enter “root” for both Username and Password entries.
4. Press “Login” button and you are now in the main menu.

Figure 2 – Main Menu

5. Appliance Only -- Under “Administration” menu please select “Configure System” and then “System Setup”.

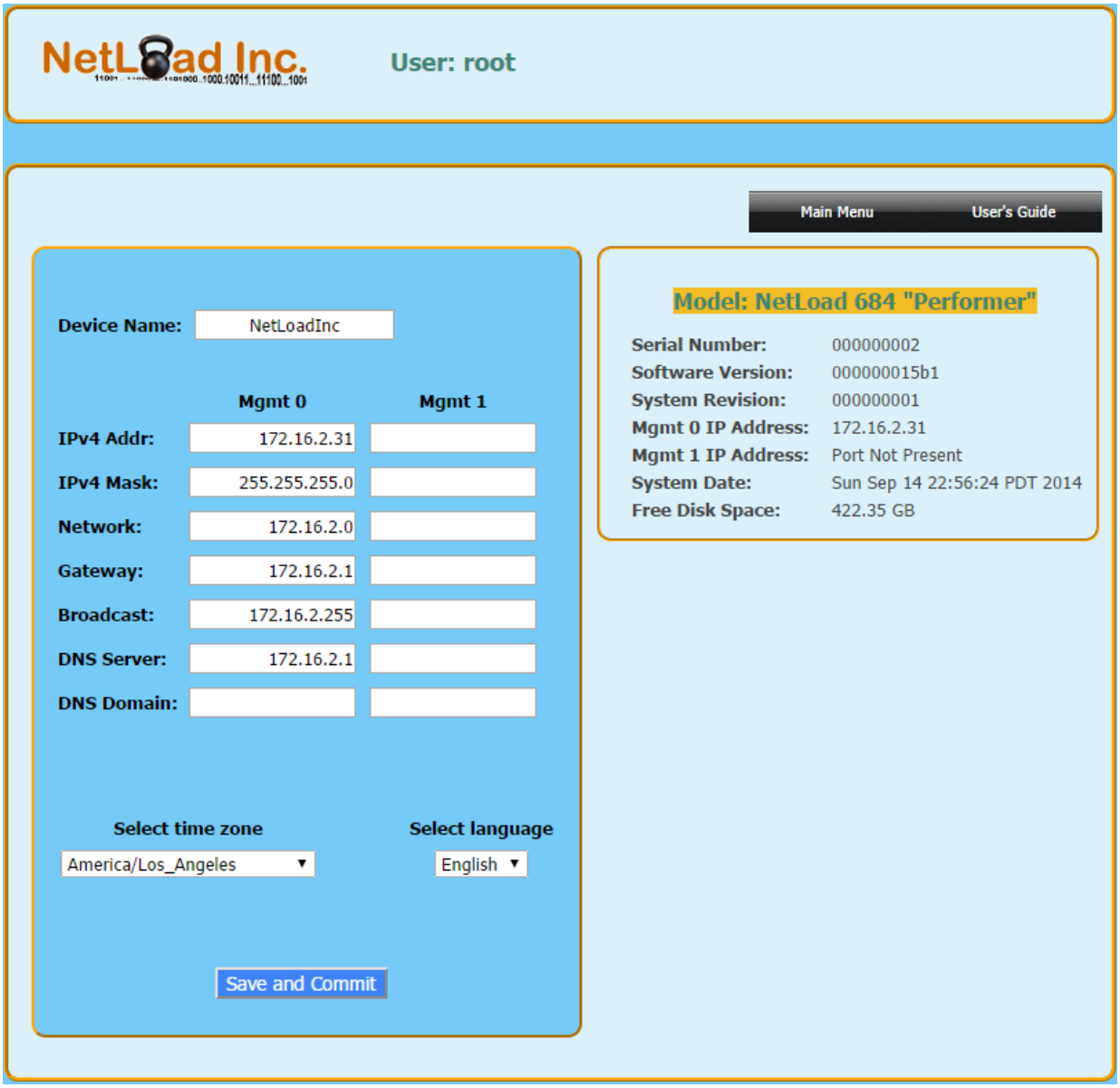


Figure 3 – System Configuration

6. Enter the values that you will use to access the system from this point forward. Please record these values for future reference if needed.
   Note: MGMT1 is not available on NetLoad "68N" systems.
7. Press "Save and Commit". The NetLoad will save your changes and restart the networking configuration.
8. The NetLoad will now be found at the new address you entered. Go to http://address_you_entered to access the device (https is also available).

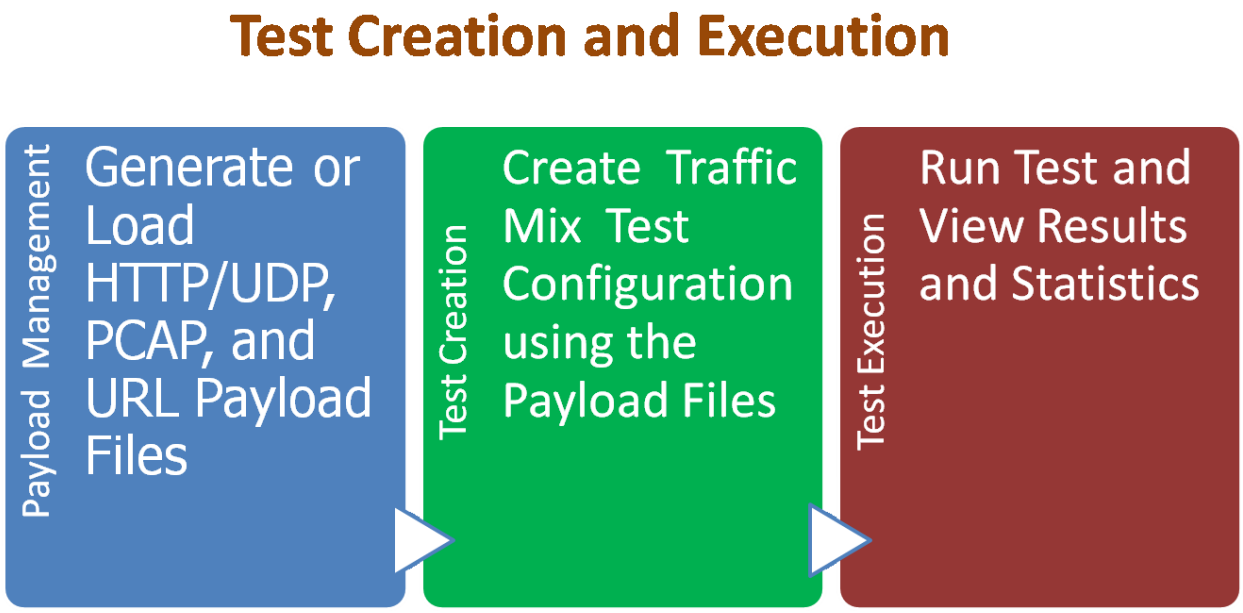
9. Under “Administration” please select “Configure System” and then “Change Login”.
10. Enter the new Username and Password you would like to use and select “Update Login”. Record these for future reference.

You have now completed base configuration of the system.

Overview

NetLoad products are simple to use traffic generators and testers geared for performance testing. They offer and mix multiple traffic types including TCP/HTTP traffic generation (stateful), PCAP replay of user-captured traffic, UDP traffic generation with variable packet sizes and fragmentation, and URL List traffic testing.

NetLoad products use a simple process to create and execute tests as illustrated in the following diagram. The following sections cover the details of the process.



This following section provides the overview of the steps to create new test configurations.

Step	**Description**
Step 1a	HTTP Transaction and UDP Payload File Management
Step 1b	URL/FW List File Management
Step 1c	PCAP Replay File Management
Step 2	Configure New Test
Step 2a	Configure Basic VLANs
Step 2b	Configure Routed Interfaces (Virtual Router)

Step 1a: HTTP Transaction Files and UDP Payload Files

For TCP/HTTP Transactions and UDP Datagrams, user-generated or system-generated payloads (files or web pages) are used for server-client interactions (GETs and POSTs) or UDP packet generation. The system has been pre-loaded with files of different transaction payload size.
This step will add additional files to the system as chosen by the user. Note that file names cannot exceed 24 characters.

1. From main menu, under “Test Setup” select “Manage Payloads”->”HTTP Transaction Files”. This shows the files currently on the system.
2. To add new transaction/payload files under “Test Setup” -> “HTTP Transaction Files” click on “Add Files”. Select up to 10 new files at a time and add them to the system (see Fig. 4).
3. To generate a file of specific byte size with incremental pattern, files under “Test Setup” -> “HTTP Transaction Files” click on “Add Files”. Enter value in bytes in “Generate File with Incremental Pattern” field and a file will be created.
4. Select “Upload/Generate Transaction Files”.

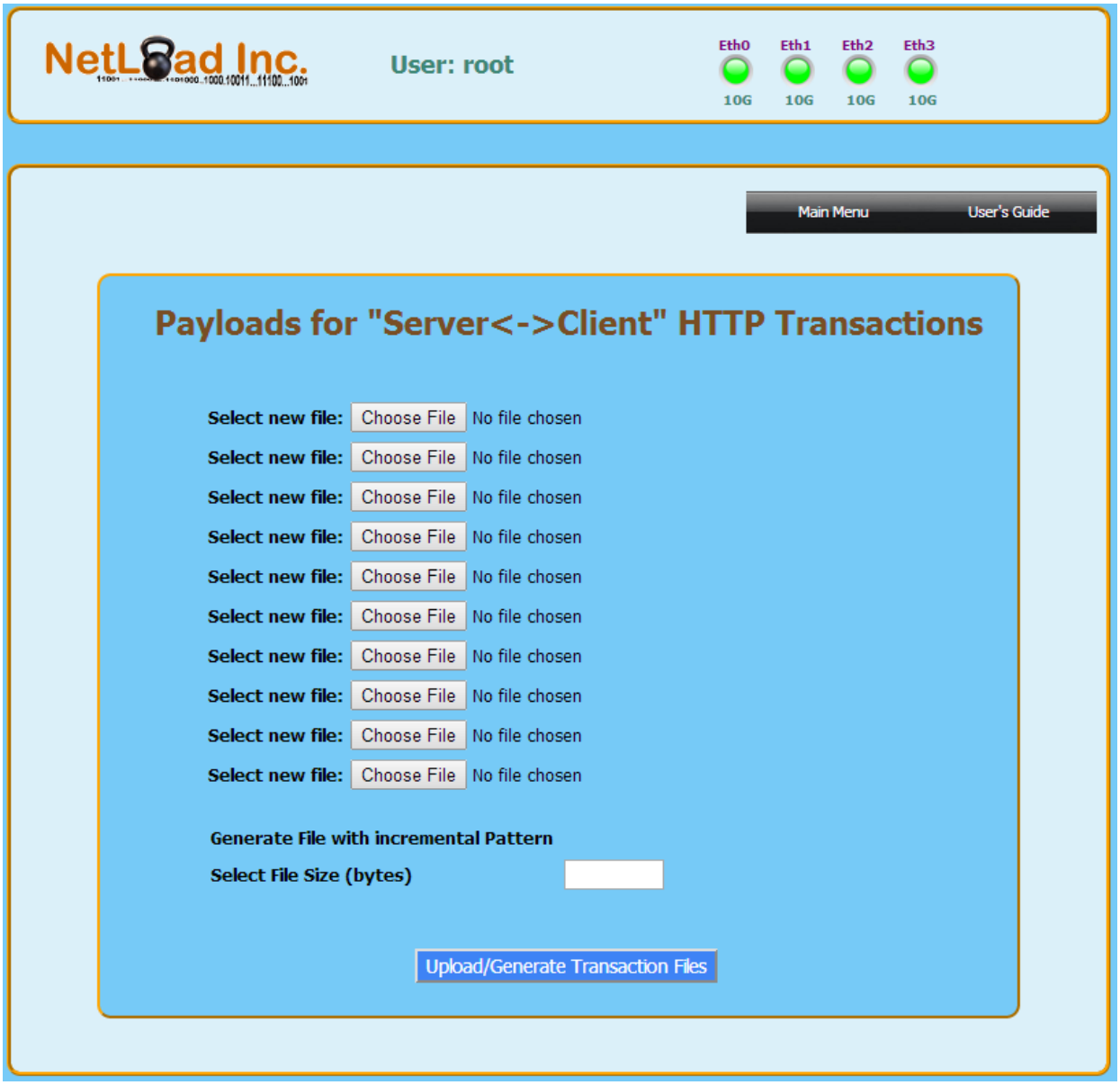


Figure 4 – Adding HTTP Transaction Payload Files

5. To remove files from the payload file area, under “Test Setup” select “Manage URL/Payload Files” and then select “Remove Files”. Checkmark files that are no longer needed and select “Remove Selected Files” button.

Step 1b: URL List File Management

This system allows user to verify that the DUT responds correctly to appropriate TCP or HTTP requests. To test, a user creates a URL List file that the system uses to generate TCP and HTTP transactions. The list file can be of any size, but must follow the sample format.

```
0,0,172.16.2.45,0,302,www.google.com,index
0,0,172.16.2.45,0,303,www.twitter.com,index
0,0,172.16.2.46,0,304,www.twitter.com
1,0,172.16.2.48,0,306,www.cnn.com
1,0,172.16.2.49,0,200,www.youtube.com
1,0,172.16.2.49,0,0,www.facebook.com
1,0,172.16.2.50,0,2000,www.bbc.com
```

The above format defines for each line:

- Client Port Index used to establish (start) TCP/HTTP connection (Valid index for a single client-server setup are 0 or 1)
- VLAN ID (0 if no VLANs are used)
  Source IP address of the Client (if 0, the system will use an address from the test port range defined by HTTP configuration client range)
- Destination IP address of the Server (if 0, the system will use an address from the test port range defined by HTTP configuration server range)
- HTTP response expected for this request from the HTTP allowed responses (100-500) based on behavior the user expects or '0' for no response expected. For FW testing, '2000' is used. The expected result is that SYN-ACK should never be received for this TCP connection request. For more information on proper HTTP Response codes see http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html or http://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- URL in the format specified as above example
- URI in the format specified as above example

NOTE: The list should contain no blank lines, and no comments to pass validation.

1. From main menu, under “Test Setup” select “Manage Payloads”->”URL/FW List Files”. This shows the files currently loaded on the system (no files are loaded on new systems).
2. To add new files, under “Test Setup” select Manage Payloads”->”URL/FW List Files” and then “Add Files” (see Fig. 5). Select new file and add to the system. The file is validated before upload is complete. If validation fails, error is indicated, and must be corrected before a file is uploaded.

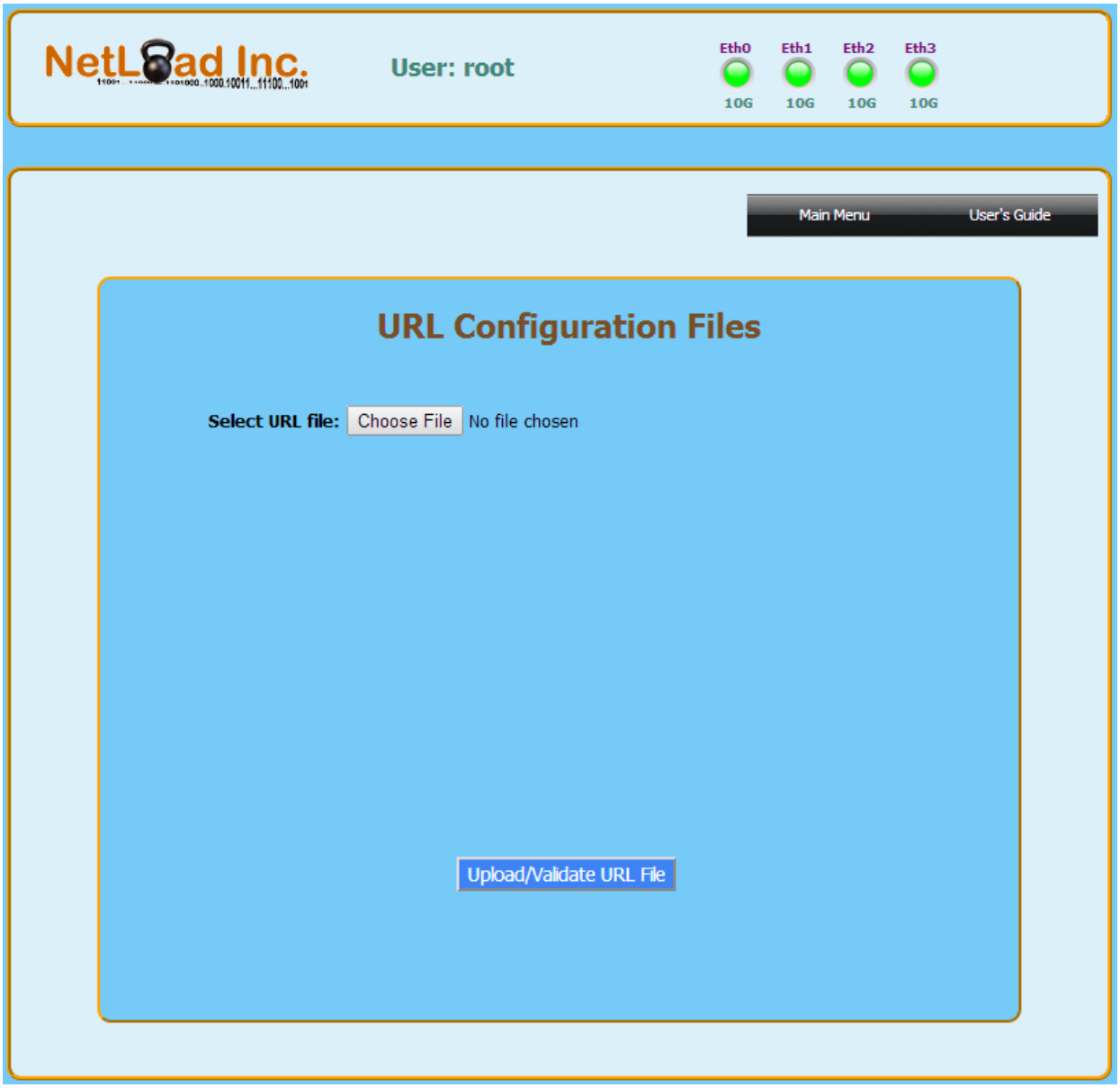


Figure 5 – Adding URL/FW List Files

3. To remove files from the URL file area, under Manage Payloads”->”URL/FW List Files” and select “Remove Files”. Checkmark files that are no longer needed and press “Remove Selected Files” button.

Step 1c: PCAP Replay File Management

This system allows user to replay PCAP files. It is also possible to mix PCAP traffic with HTTP Transaction traffic to create various test scenarios that combine many traffic types, including viruses, malware, etc.

1. From main menu, under “Test Setup” select “Manage Payloads”->”PCAP Replay Files”. This shows the files currently loaded on the system (no files are loaded on new systems).
2. To add new files, under “Test Setup” select Manage Payloads”->” PCAP Replay Files” and then “Add Files” (see Fig. 6). Select new file and add to the system. The file is validated before upload is complete. If validation fails, error is indicated, and must be corrected before a file is uploaded. Max file size permitted is 200MB. For files above 200MB, login using SSH and use SCP to load onto the system. The files must be placed into **/home/netload/configs/pcap** directory, and must have **.pcap** extension.

NOTE: The files must be valid PCAP files. PCAPNG files are not supported. Validation is not done on files loaded via SSH.

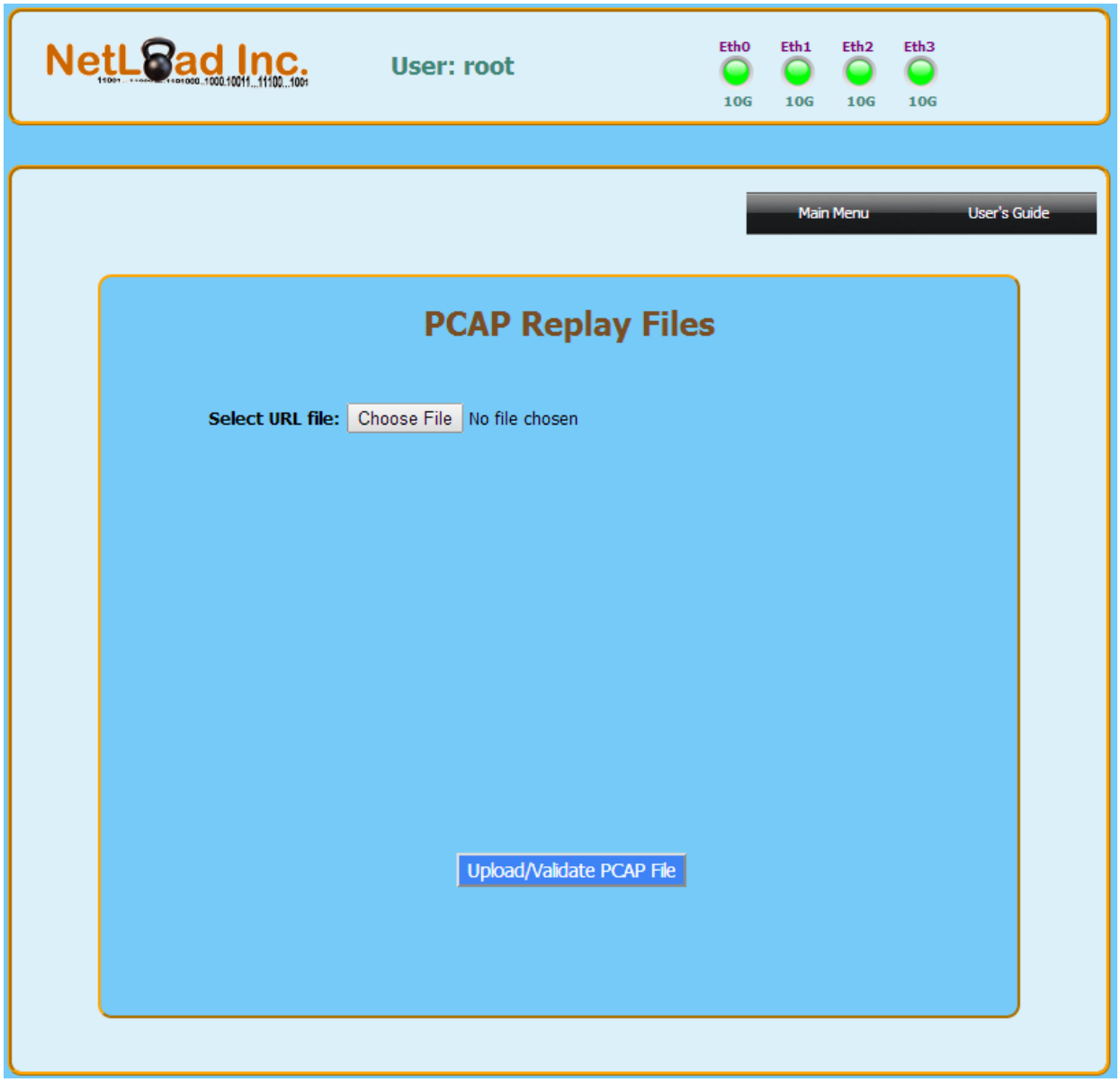


Figure 6 – Adding PCAP Files

3. To remove files from the PCAP file area, under Manage Payloads"->" PCAP Replay Files" and select "Remove Files". Checkmark files that are no longer needed and press "Remove Selected Files" button.

Step 2: Creating a test

Tests can be created either by using a completely new configuration, or by modifying and saving an existing configuration with the same or new name. Some basic configurations have been added to the system.
To create a configuration, select “Create Test Configuration” from Main Menu. For “Developer” systems, only “Static” configurations are available. For “Performer” systems, “Static” and “Dynamic” configurations are available.

Figure 7 – Static Traffic Mix and Test Type Selection

1. Select Traffic Mix using the triple slider (see Fig.7). For example, to select HTTP-only traffic, select 100% of HTTP by moving left-most slider to the left and middle and right-most sliders to the right.
2. For HTTP traffic type greater than 0%:
   a. Select the Balance between HTTP GETs and HTTP POSTS. Traffic distribution will be most symmetric with 50% - 50% distribution.
   b. Select HTTP Ramp-up time to reach the full HTTP Transactions/sec rate.
   c. Select HTTP SYN Burst Rate (how many SYNs are sent out in one burst to create connections) – some systems have load-balancing requirements. If your system does not have any requirements, we recommend leaving this at 20 SYNs /burst.
   d. HTTP Selected Payload File Count – select the HTTP Transaction Payload Files used as payload for the HTTP GETs and POSTs. Multiple (up to 10) files can be selected. Rate allocation (percentage of the overall HTTP traffic) for each Transaction Payload File is selected later in the configuration.
   e. TCP Optimization – different TCP clients and servers have slightly different session configurations. By selecting "Shortest Session" the most optimal TCP termination is selected (least number of protocol packets needed to complete the transaction and complete the HTTP/TCP session). "Standard Session" uses a longer termination exchange, producing a larger amount of control packets for each HTTP/TCP session.
   f. TCP Termination – terminate the TCP session with RST or FIN.
   g. Enable "Zero Data Transaction" allows for testing of pure TCP session setups and teardowns. No HTTP GETs or POSTs are made and no Payload Data is transferred when using this option and all Payload Files are NA.
3. For PCAP Replay traffic type greater than 0%:
   a. Select PCAP Ramp-up time to reach the full PCAP Packets/sec rate.
   b. Select PCAP Packet Burst Rate -- recommended rate is 20 Packets/burst for higher rates, and small Burst Rates for lower rates. For very slow rates, use Packet Burst Rate of 1.

c. PCAP Transmission Timing – Select “Timestamp” to send PCAP packets based on PCAP file packet time stamp contained in PCAP files. This option mimics the timing of the original PCAP file.
d. PCAP Timestamp Scale Factor – when “Timestamp” mode is used, the overall timing of the traffic can be scaled by a factor to speed up the packet transmission, or slow it down. Factor of “1” is default timing. To send packets at twice the original PCAP file timestamp rate, set PCAP Timestamp Scale Factor to 2. To send packets at half the rate, set PCAP Timestamp Scale Factor to 0.5.
e. Select PCAP files for each Ethernet port. Up to 25 files per port may be selected.

4. For URL/Firewall List traffic type greater than 0%:
   a. Select URL Ramp-up time to reach the full URL Connections/sec rate.
   b. Select URL SYN Burst Rate -- recommended rate is 20 SYNs/burst.
   c. URL List File – select the URL configuration file. The file format is covered in Step 1B.
5. For UDP traffic type greater than 0%:
   a. Select UDP Ramp-up time to reach the full UDP Packets/sec rate.
   b. Select UDP Packet Burst Rate -- recommended rate is 20 Packets/burst for higher rates, and lower or 1 for low Packet/sec rates.
   c. UDP Traffic: select “Unidirectional” for one direction traffic blasting, or select “Server Echo” for the server side to send back traffic that is received.
      NOTE: “Server Echo” must be selected to measure latency across the link.
   d. If “UDP Traffic: Server Echo” is selected, a ratio of traffic sent vs. received can be selected. For example, selection of 20K would mean that for every 20K packets sent by client side, the server will send back one packet.
   e. Select up to 3 UDP Payload files for each physical port. File content is used to create UDP IPv4 datagrams. The UDP datagrams are formatted into packets of user-configured size. The file can be used to create

multiple UDP datagrams (packets) of a single size or a mix, or can be used as a UDP Fragmented datagram up to 64K in size.

6. Traffic Priority – select which traffic type is the main traffic for the test. Other traffic will become "background" traffic. Once Primary traffic is finished sending its data, all other traffic types will stop transmission. For all traffic to run independently, select "None".
7. Advanced Port Control – enables individual physical port bandwidth control of each traffic type in the final step of the configuration. When disabled, all traffic on all ports is controlled by a single global bandwidth control.
8. Select "Continue" and move to the next screen to select the traffic distribution for each traffic type including HTTP traffic, each PCAP file for each port, and each UDP file for each port.
9. For HTTP Traffic greater than 0% (see Fig. 8):
   a. Select Weight Number for each HTTP Payload file previously selected to assign it a percentage of bandwidth. Weights can be "1 - 512", with larger ratios providing finer resolution rate between the files on each link. Total Weight must be a Power of 2.
   b. HTTP MSS – Select MSS (Maximum Segment Size) used for HTTP Traffic. MSS size will limit all packets to the selected size. Example: 1460 Byte Payload File would be transferred as a single Ethernet Packet when MSS is set to 1460 (Default). The same payload will be broken into 8 packets if MSS was set to 200 Bytes. With MSS set to 1, it would take 1460 Ethernet Packets to transfer this payload. The MSS for standard Ethernet Packet is 1460 Bytes (Default), and up to 9200 for Jumbo Frames.
   c. HTTP Round Trip Delay adds additional delay to the overall traffic. This delay is traffic-load dependent and is not a constant value at all rates. Default value is 0.

Figure 8a – HTTP Traffic Distribution

10. For PCAP Traffic greater than 0% (see Fig. 8b):
    a. Select Weight Number for each PCAP file previously selected for selected port to assign it a percentage of bandwidth. Weights can be "1 - 512", with larger ratios providing finer resolution rate between the files on each link. Total Weight must be a Power of 2.
    b. "Split" option can be used for PCAP files that contain captures of stateful bi-directional traffic. The "Split" function will examine the file and identify a direction for each packet. It will attempt to group them as Client-side and Server-side. The file will then be replayed on the associated Client-Server Pair similar to the original traffic exchange between clients and servers or 2 end points. Client packets will be the first ones to be sent. If the traffic is not TCP-based, the traffic will be split along the physical ports based on MAC address allocation.
    c. PCAP Split Mode Client->Server Delay is used with the "Split" function. It adds an additional delay between client-to-server packets, thus creating a more realistic timing. For example, if 6 packets are scheduled to be sent with a pattern of C-C-C-C-C-S (C-client, S-server), this Delay parameter will add a user-defined delay in µsec between the C-S packets.

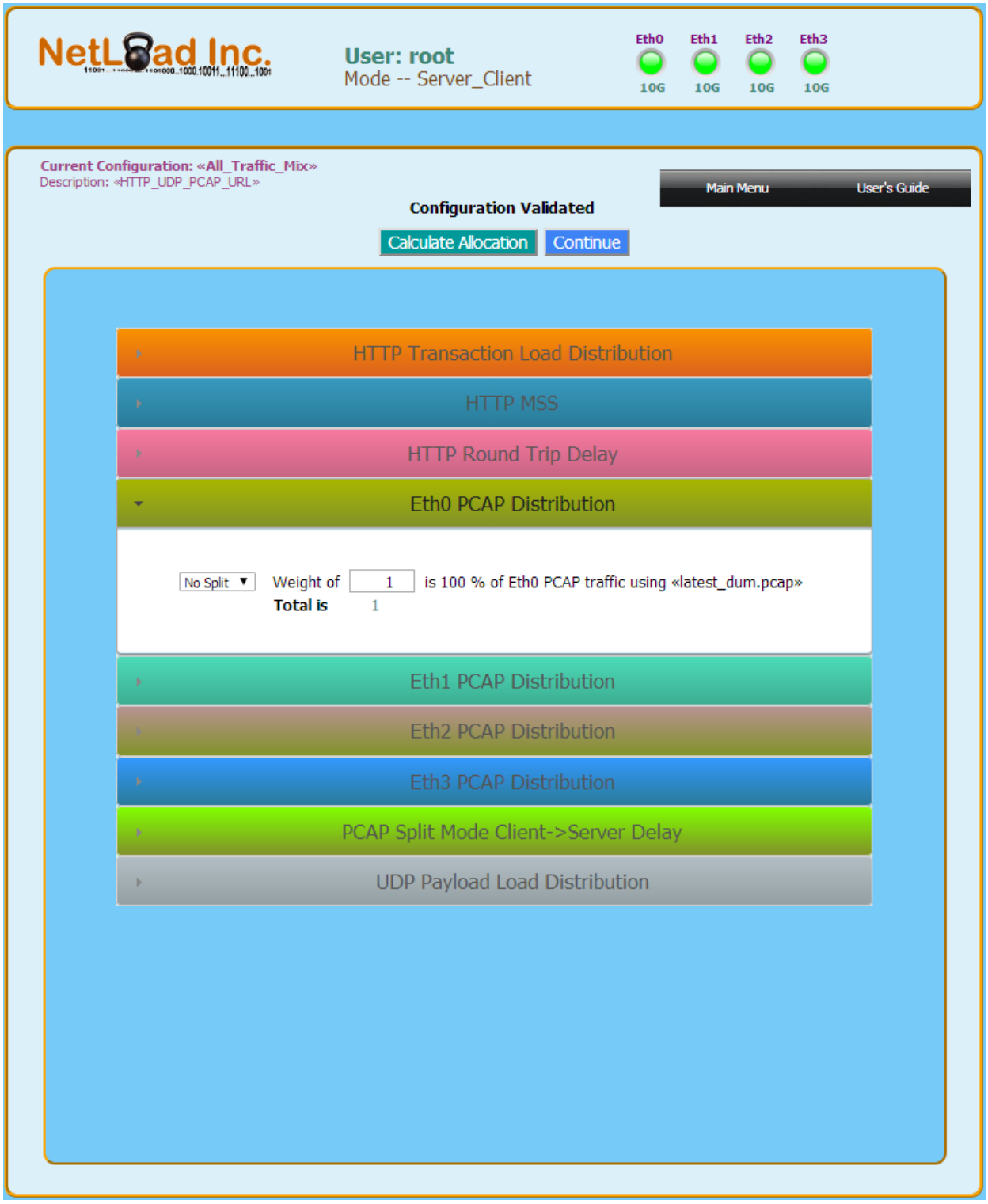


Figure 8b – PCAP Traffic Distribution

11. For UDP Traffic greater than 0% (see Fig. 8c):
    a. Select the number of Bytes for each UDP packet to be created from the selected UDP Payload file. To create 60-byte packets, select 60. This will create a stream of 60-byte packets (Ethernet) using the Payload file as UDP payload. The pattern will repeat for the full size of the Payload file. For a mix of packet sizes, add multiple packet sizes separated by ";" to create a UDP packet mix. The packet size pattern will repeat for the duration of the Payload File. To create a long-running UDP stream for a set packet size, select multiple RUN times option when running the test (covered further in the document).
    b. "Fragment" option is used to fragment a UDP datagram into multiple UDP IP fragments of user-configurable size. For example, if a 4Kbyte Payload file is selected with the "Fragment" option enabled and packet sizes of 60,128, and 256, the 4Kbyte file will be sent as a single UDP datagram fragmented into a repeating pattern of UDP IP fragments packaged into the selected packet sizes. Note that each fragment will be adjusted to be divisible by 8, so actual Packet sizes with fragmentation may be slightly different from the user-selected Packet sizes.
    c. "Drop 1 out of every N Fragments" selected together with "Fragment" option simulates lost fragment packets for negative testing. It will drop Fragments in the stream with probability of 1/N.

Figure 8c – UDP File Payload Allocation

12. Once Traffic Distribution is configured, select “Calculate Allocation” to validate the configuration. Once validated, select “Continue” to move to DUT Mode selection.
    a. Select NetLoad Tester operation mode. For single box operation, select “Single NetLoad Clients-Servers”. For multi-box configurations, select the mode most resembling your desired test setup (See Fig. 9a).

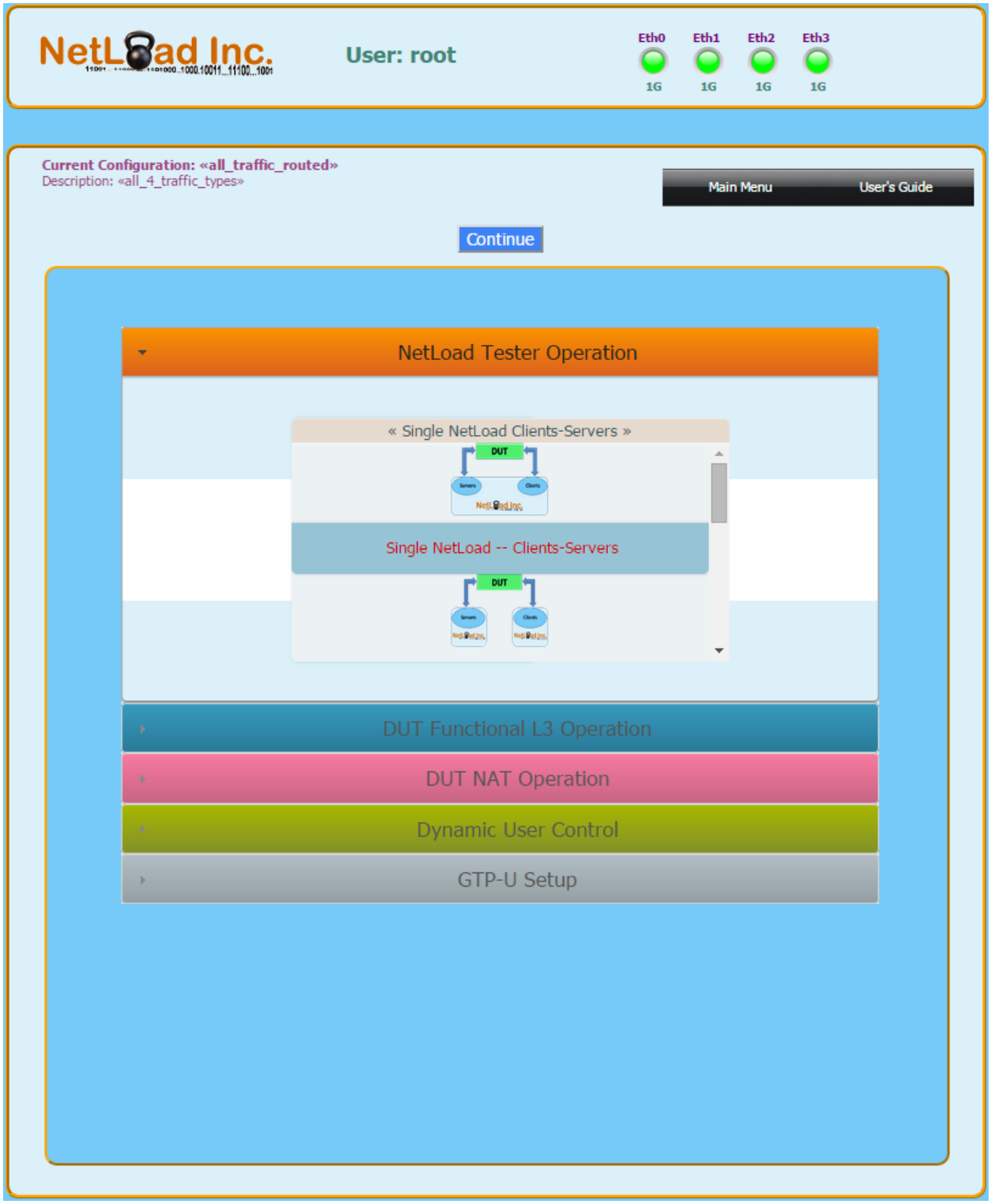


Figure 9a – Select NetLoad Tester Configuration

b. DUT Functional L3 Operation – for DUT operating transparently (example: L2 devices, IPS), select “Transparent” (or “Transparent with VLANs”) mode of operation. To enable Virtual Router setup to work with devices such as routers and L3 switches, selected “Routed Mode”. If L3 VLANs are enabled, select “Routed Mode with VLANs”. Use provided pictorial representation that most resembles your desired test setup (See Fig. 9b).

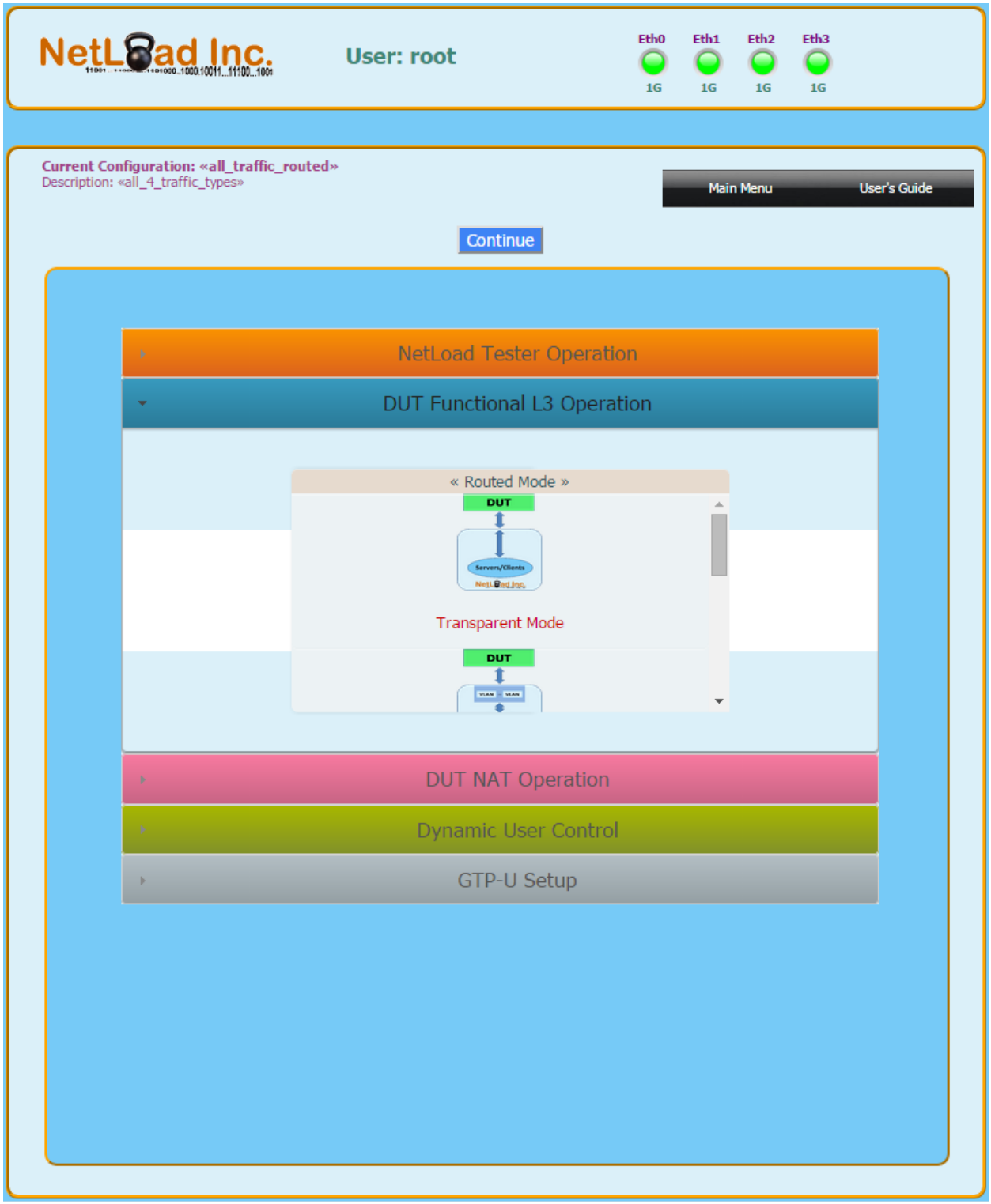


Figure 9b – Select DUT Mode

c. DUT NAT (Network Address Translation) – select “none” if DUT does not use NAT. Select a desired mode if DUT uses NAT.

Figure 9c – Select Dynamic User Control Options

13. (Performer Series Only) -- Dynamic User Control Options provide the system with basic knowledge of what to expect on the Dynamic User interface. The system utilizes the same Client/Server Ethernet port allocation as the standard TCP/HTTP mode.
    a. Max Users per Client Port – configure for max possible number of Dynamic Users an external application will have active. If "0", Dynamic User Mode is disabled.
    b. API Client TCP Port Number -- this is the TCP port number that NetLoad will use to communicate with an external system for specific purpose of adding/deleting users in Dynamic User Mode. If the external system uses two sources to add/delete user-related information (client-side and server-side), this port is used as client-side TCP port. This server is NOT used for general REST API interface. This parameter is ignored if Dynamic User Mode is disabled.
    c. API Server TCP Port Number -- this is the TCP port number that NetLoad will use to communicate with an external system in a dual-source configuration for specific purpose of adding/deleting users in Dynamic User Mode. If the external system uses a single source for all user communication, this port is not used and should be set to "0". This server is NOT used for general REST API interface. This parameter is ignored if Dynamic User Mode is disabled.
    d. User Creation Delay – this delay (in µs) is used to delay the user activity after the user information is sent to NetLoad from an external system. The delay is useful in dual-source configurations where part of the user profile is coming from a client-side source and part is coming from a server-side source. The delay can help with synchronization of the data prior to actual traffic being sent out on the wire.
    e. User Deletion Delay – this delay (in seconds) is used to delay the user removal process upon a REST API User "STOP" command. This will allow the system to complete any outstanding transactions before "Delete" of the user data from the system.

f. Optional User File – use User File to add users in combination with Dynamic Mode. The basic format is as following (see Appendix D for more details):

```
{
    "user_info":
      [
        {
            "Eth_Index":   0,
            "User_Id":     5,
            "User_Server_IPv4_Addr":      "2.3.4.5",
            "User_Client_IPv4_Addr":      "6.7.8.9",
            "Weight":         1,
            "Client_Tunnel_Info": {
            },
            "Server_Tunnel_Info": {
            }
        },
        {
            "Eth_Index":   1,
            "User_Id":     7,
            "Type":        "Client",
            "User_Server_IPv4_Addr":      "9.3.4.5",
            "User_Client_IPv4_Addr":      "2.7.8.9",
            "Weight":         2
            "Client_Tunnel_Info": {
            },
            "Server_Tunnel_Info": {
            }
        }
    ]
}
```

Figure 9d – Select GTP-U Encapsulation Setup

14. (Performer Series Only) -- GTP-U Setup is used in combination with User Control to emulate GTP user addition and deletion based on external system providing user-specific information. Please refer to Appendix D for additional information on per-tunnel rate control.
    The format for GTP user information is as following:

```
{
    "user_info":
      [
        {
            "Eth_Index":   0,
            "User_Id":     5,
            "User_Server_IPv4_Addr":       "2.3.4.5",
            "User_Client_IPv4_Addr":       "6.7.8.9",
            "Client_Tunnel_Info": {
              "Dst_IPv4_Addr":    "11.12.13.14",
               "Src_IPv4_Addr":   "21.22.23.24",
               "Dst_TEID":        456,
               "Src_TEID":        6789
            },
            "Server_Tunnel_Info": {
               "Dst_TEID":      6789,
               "Src_TEID":      456
            },
            "Weight":         1
        },
        {
            "Eth_Index":   0,
            "User_Id":     6,
            "Replay_File_Name":     "pcap_file.pcap",
            "User_Server_IPv4_Addr":       "3.4.5.6",
            "User_Client_IPv4_Addr":       "7.8.9.6",
            "Client_Tunnel_Info": {
              "Dst_IPv4_Addr":    "11.12.13.15",
               "Src_IPv4_Addr":   "21.22.23.25",
               "Dst_TEID":        856,
               "Src_TEID":        5789
            },
            "Server_Tunnel_Info": {
               "Dst_TEID":      9789,
               "Src_TEID":      156
            },
            "Weight":         1
```

```
        },
        {
            "Eth_Index":    1,
            "User_Id":      7,
            "Type":         "Client",
            "User_Server_IPv4_Addr":        "2.3.4.5",
            "User_Client_IPv4_Addr":        "6.7.8.9",
            "Client_Tunnel_Info": {
              "Dst_IPv4_Addr":     "11.12.13.14",
               "Src_IPv4_Addr":    "21.22.23.24",
               "Dst_TEID":         556,
               "Src_TEID":         9789
            },
            "Server_Tunnel_Info": {
                "Dst_TEID":       9789,
                "Src_TEID":       556
            },
            "Weight":           2
        }
    ]
}
```

a. No GTP Encapsulation – no GTP encapsulation will be done on either client or server side of the NetLoad system
b. Client and Server GTP encapsulation – both client and server side will send/expect GTP encapsulation
c. Client Only GTP Encapsulation – client side of the NetLoad system will encapsulation all traffic in GTP but the server side will expect un-encapsulated traffic
d. Server Only GTP Encapsulation – server side of the NetLoad system will encapsulation all traffic in GTP but the client side will expect un-encapsulated traffic

15. Once NetLoad Tester, DUT, Dynamic Mode, and GTP Encapsulation operation modes are configured, select “Continue”.
16. Configure the test parameters, Client-Server addresses, and Routing interfaces if “Routed Mode” was selected for the DUT (see Fig. 10):

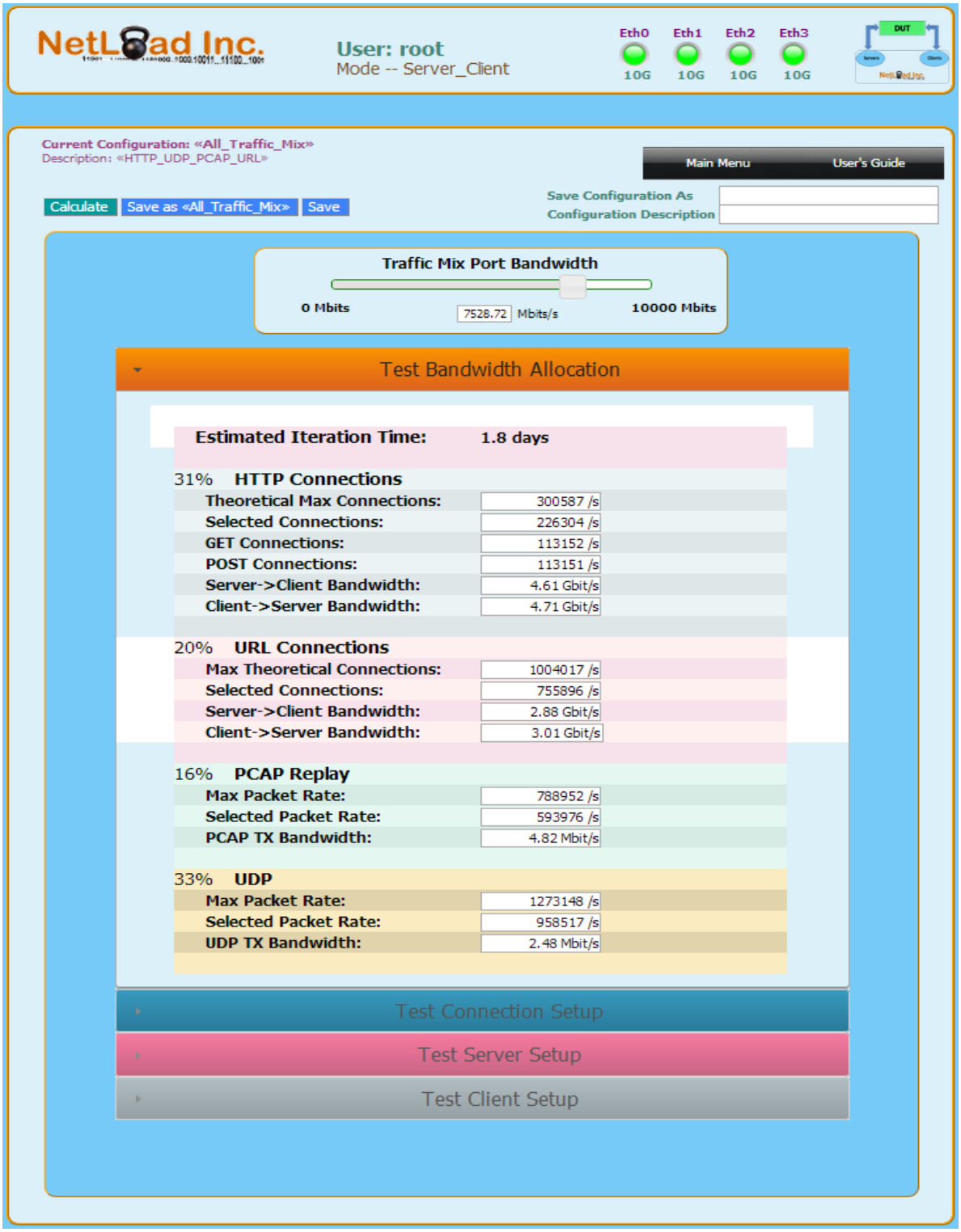


Figure 10a – Calculate and Select Test Parameters

17. For Non-Advanced Port Control Mode:

a. Using the slider, select desired bandwidth rate (applicable to globally to each port). For example, a rate of 500Mbits/s on a system that supports 1Gbit/s will adjust all parameters in the Test Bandwidth Allocation area to generate traffic at 500Mbits/s on each port. Slider range is within .01Mbits/s, but for high speed system (10G) the rate is within .1Mbits/sec.
b. Test Bandwidth Allocation will calculate the HTTP/TCP and URL Connections/sec and PCAP/UDP Packets/sec rates based on selected % rate of each traffic type, each HTTP Payload size and URL List size, and average Packet Size in the PCAP files and UDP files. Each traffic type is displayed to the user with the maximum throughput for the specific traffic type and port bandwidth, and selected values for the actual test. Note that for stateful traffic such as HTTP, the Server-Client and Client-Server traffic is not symmetric.
c. Test Connection Setup -- selects the total number of Servers and Clients on each port (these numbers together with the Client TCP Port Number create the total number of attempted connections to try for the test). The total number of connections attempted will be #Clients * #Servers * # Client TCP Ports * Number of port pairs enabled. The test duration is dependent on the above formula and the size of the selected transactions – the bigger the transaction, the longer each one takes to complete.
d. Active Connections – number of open connections that are maintained open throughout the test. As system closes TCP/HTTP connections after completion of HTTP GETs and POSTs, new connections are opened to maintain the total Active number throughout the test.
e. IP Address Sequence – select how the Server and Client IP addresses are selected during the test. Addresses are either incremented sequentially or selected randomly within the configured ranges.
f. Test Server Setup – selects Server Address pool for each active Server port (for Single Box configuration, Eth0 and Eth1 are Server ports, and Eth2 and Eth3 are client ports).

g. Test Client Setup – select Client Address Pools for each client port, disable port pairs (first port pair is always enabled).
h. Select TCP Port Range for the test to use. If not needed, use 32268 and 60998.
i. Select number of TCP ports used by each Client-Server Address pair during the test. The number of ports must be smaller than earlier defined TCP range.

18. For Advanced Port Control:
    a. Test Bandwidth Allocation is calculated independently for each port pair for the HTTP/TCP and URL Connections/sec and independently for each port for PCAP/UDP Packets/sec rates based on selected % of rate of individual traffic type, each HTTP Payload size and URL List size, and average Packet Size in the PCAP files and UDP files. Each traffic type is displayed to the user with the maximum throughput for the specific traffic type and specific port bandwidth, and selected values for the actual test. Note that for stateful traffic such as HTTP, the Server-Client and Client-Server traffic is not symmetric (See Fig. 10b).
    b. Port Bandwidth Allocation is selected by individual sliders (see Fig. 10c).

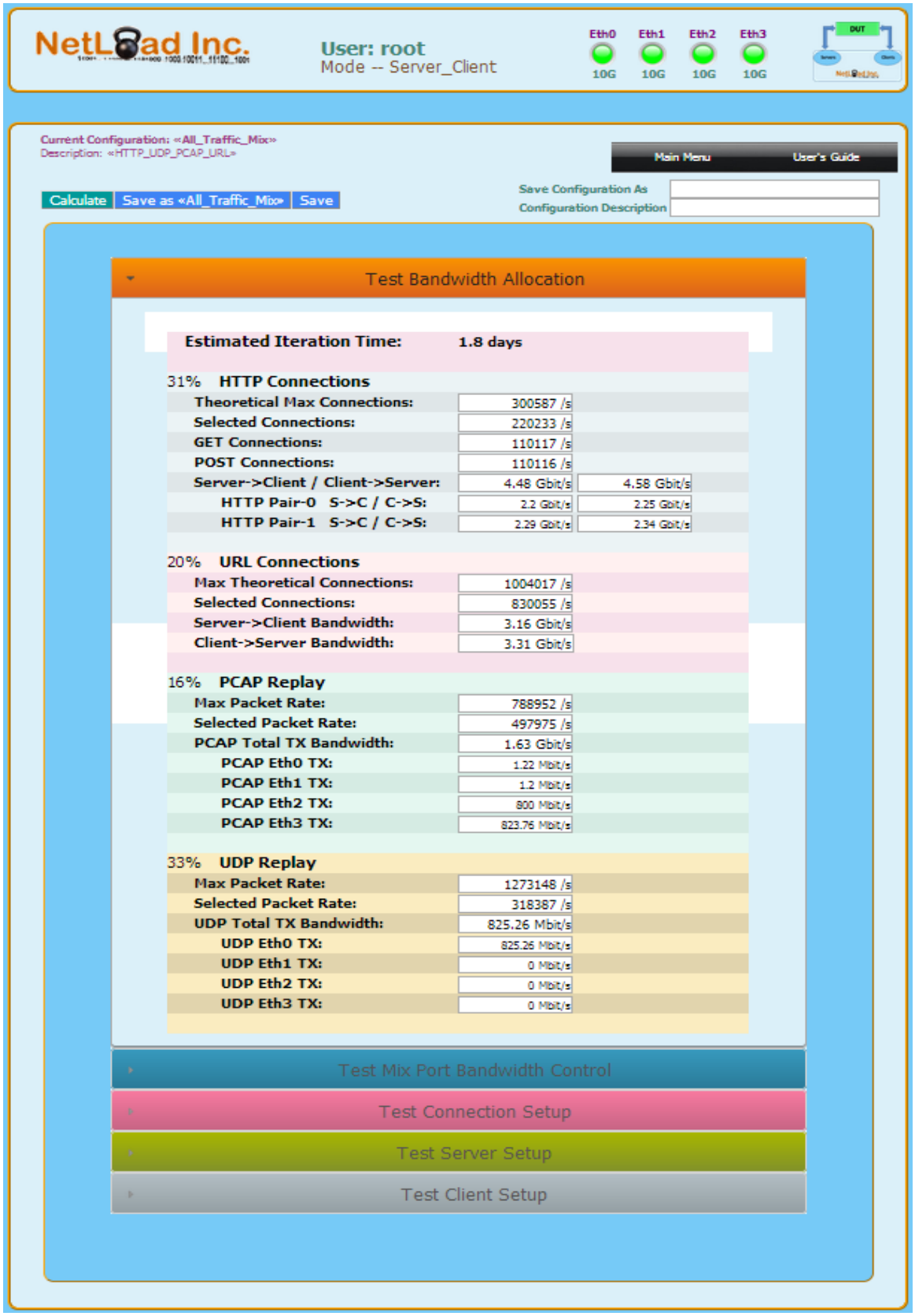


Figure 10b – Calculate and Select Test Parameters per Port

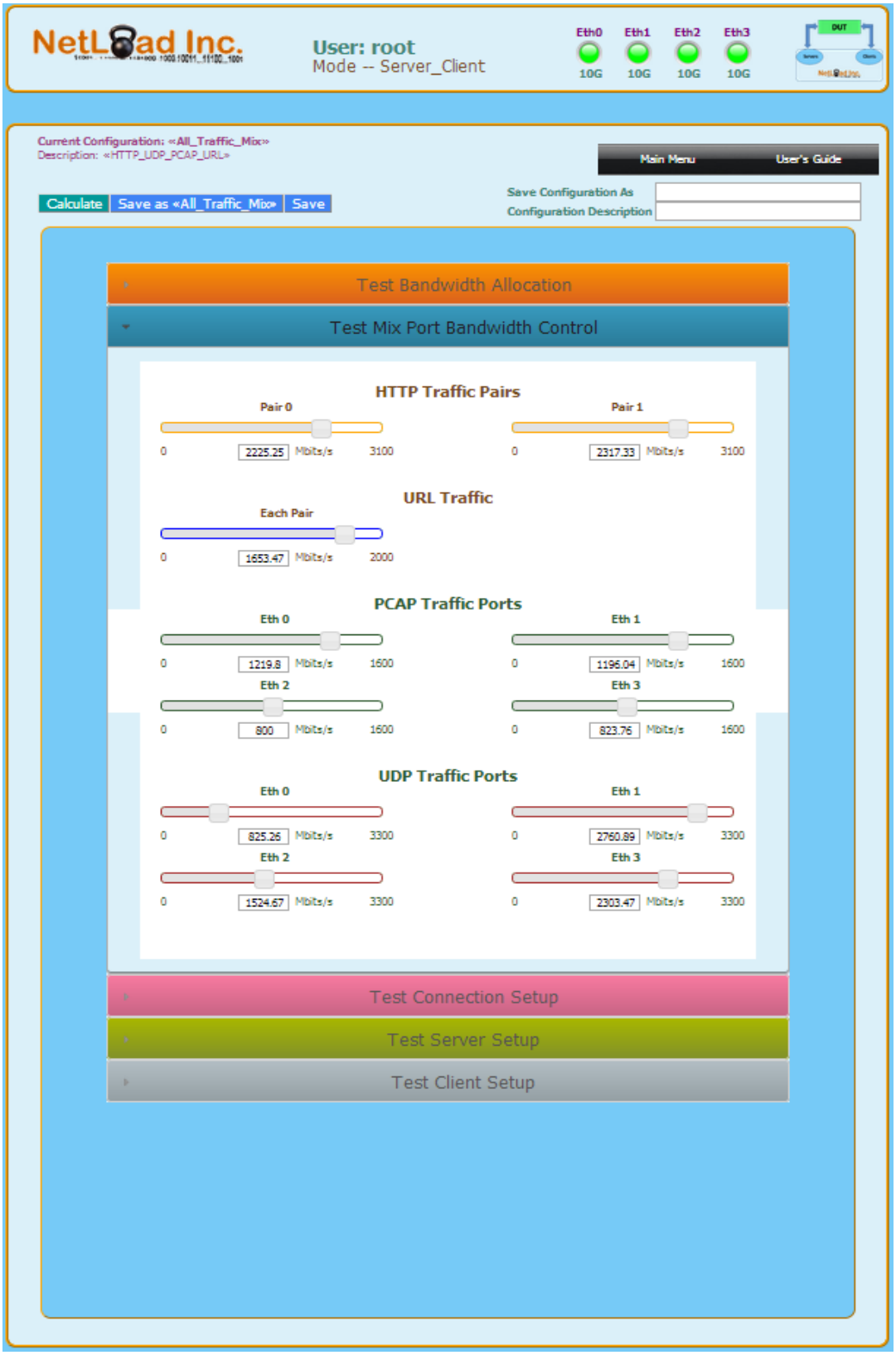


Figure 10c – Advanced Individual Port Bandwidth Control

Step 2a: Transparent Setup with VLANs

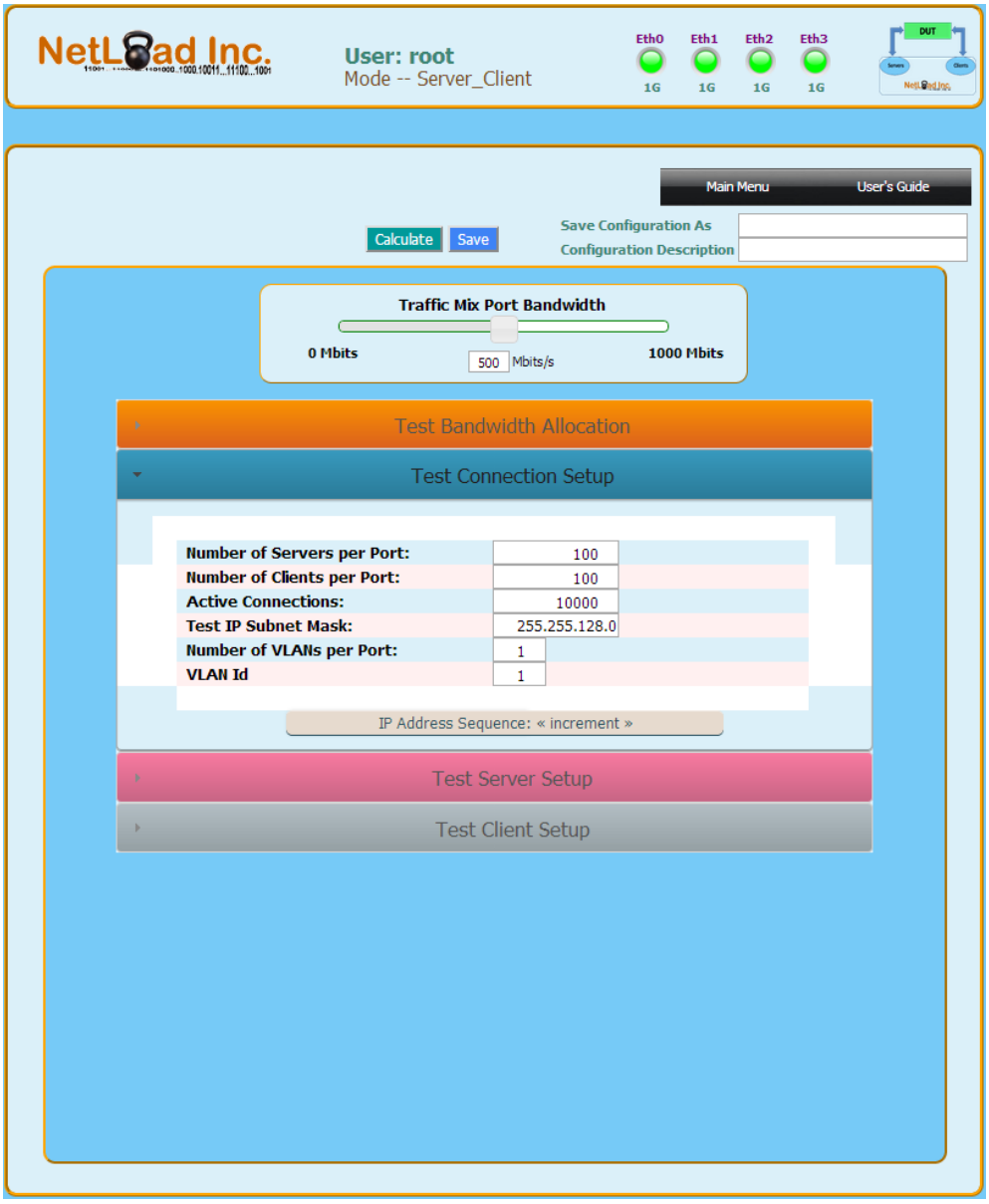


Figure 11 – Configure Basic Test with VLANs

1. Select “Number of VLANs per port” (see Fig. 11) – this parameter defines the number of VLANs in each individual physical Ethernet port. Valid number is between 1 and 4096.
2. VLAN Id – select the starting VLAN number.

Step 2b: Virtual Router Setup

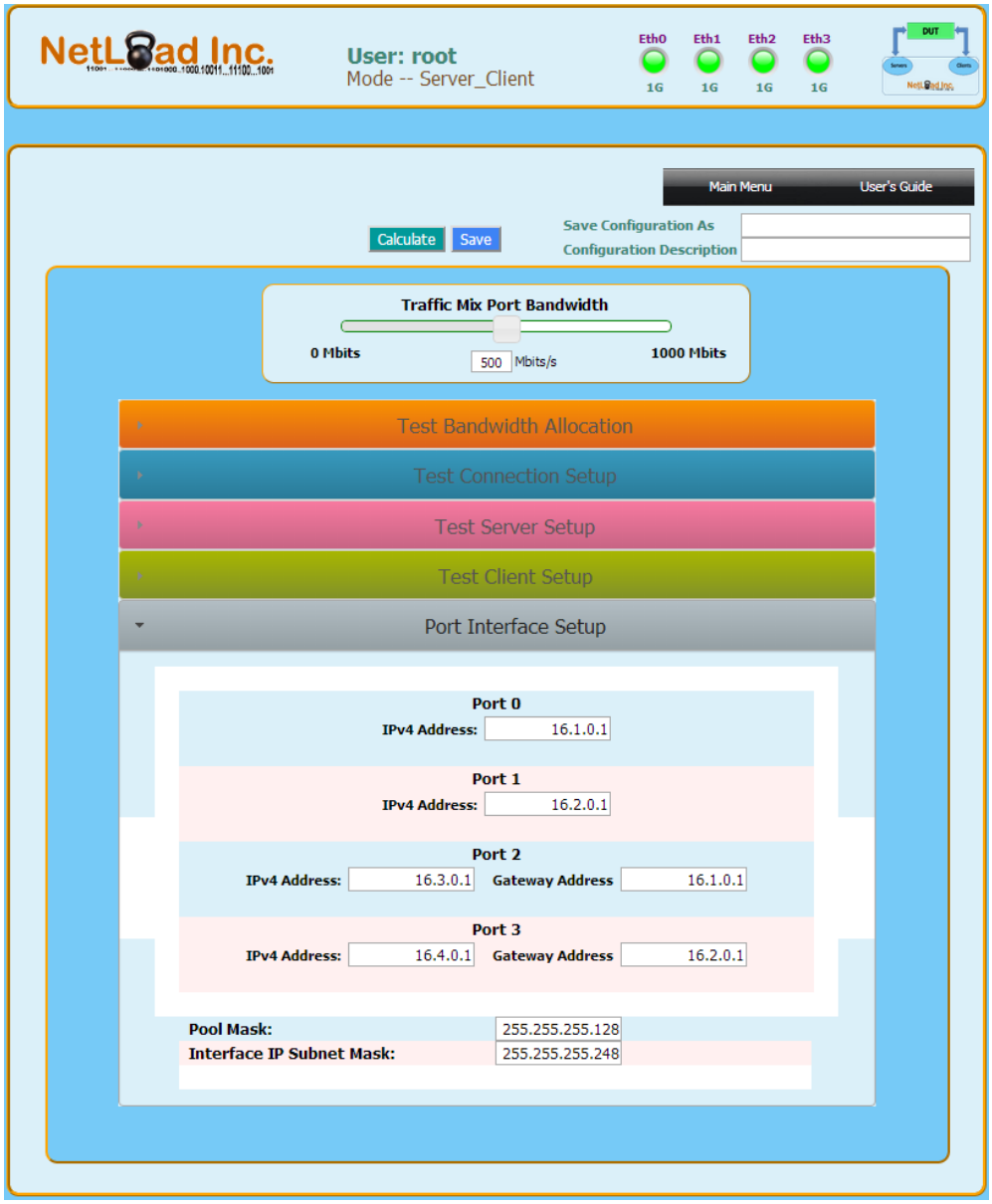


Figure 12 – Configure Interfaces for Routed Setup

1. Under “Port Interface Setup” (see Fig. 12):
   a. “IPv4 Address” – selects the IP address for the Virtual Router Physical port
   b. “IP Subnet Mask” – selects the IP address subnet for the Virtual Routers on each physical Client and Server port.

c. "IPv4 Gateway Address" – selects the Gateway address for the path between Clients and Servers. This address is usually set to the DUT IP address for the physical port that is connected to the client ports

2. Configure Static Routes on DUT with proper addresses.

Note: An example setup of "Virtual Router" and static routes is shown in Appendix A and Appendix B.

Batch Test Setup

This section provides the overview of the steps to setup a batch test configuration. The batch configuration can run up to 10 tests in sequence, report results for the overall batch, and compare the results with a “Gold” test result for each individual test.

Step	**Description**
Step 1	Configure New Batch Test
Step 2	Modify Existing Test

Step 1: Create New Batch Test

Note that file names cannot exceed 24 characters.

1. From Main Menu, under “Create Test Configurations” select “Create New Batch Test” (see Fig. 13):

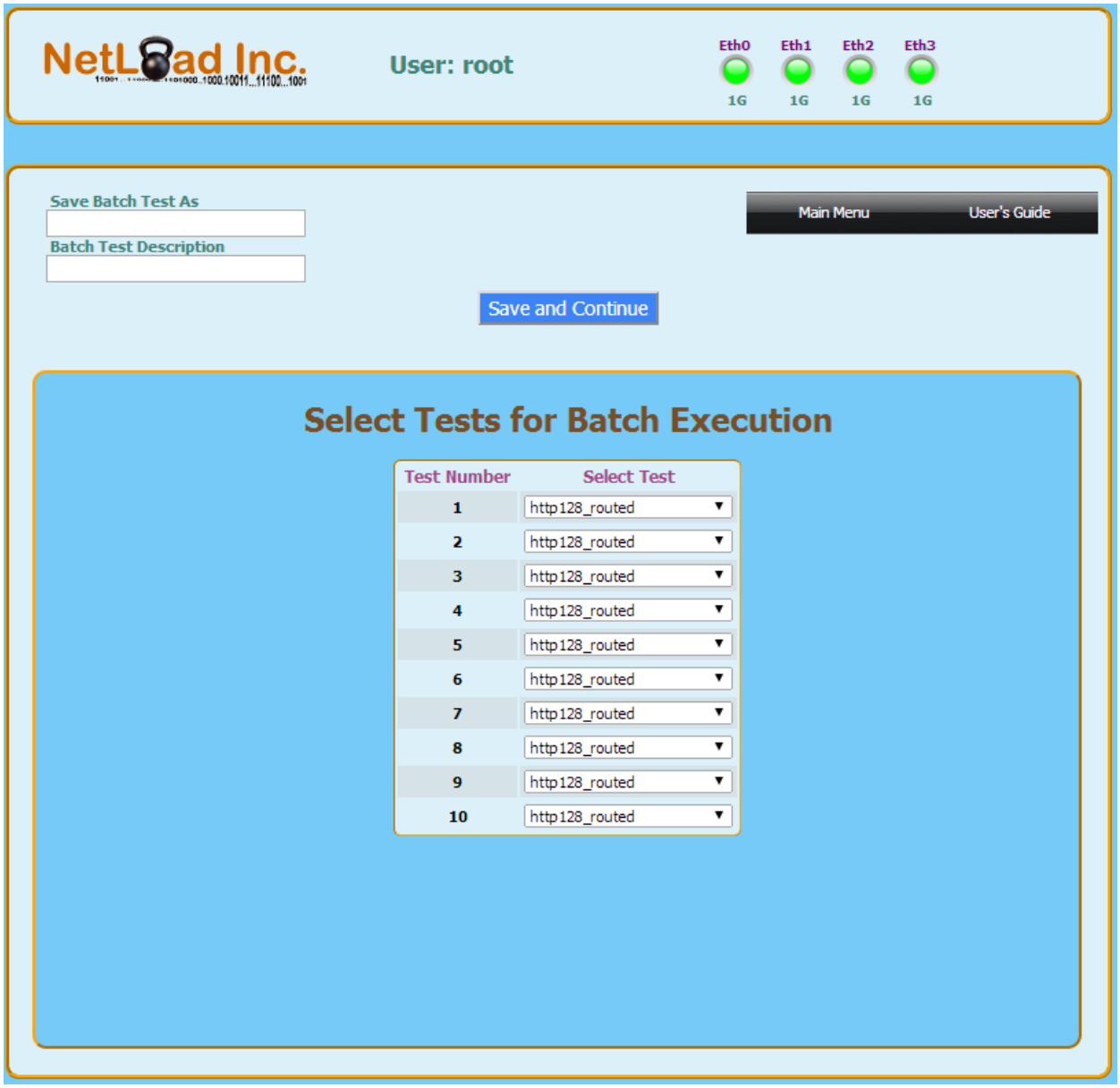


Figure 13 – Create New Batch Test

2. Select Name and Description for the new test.
3. For each Test Number, select the test you want to run. You can execute the same test multiple times.

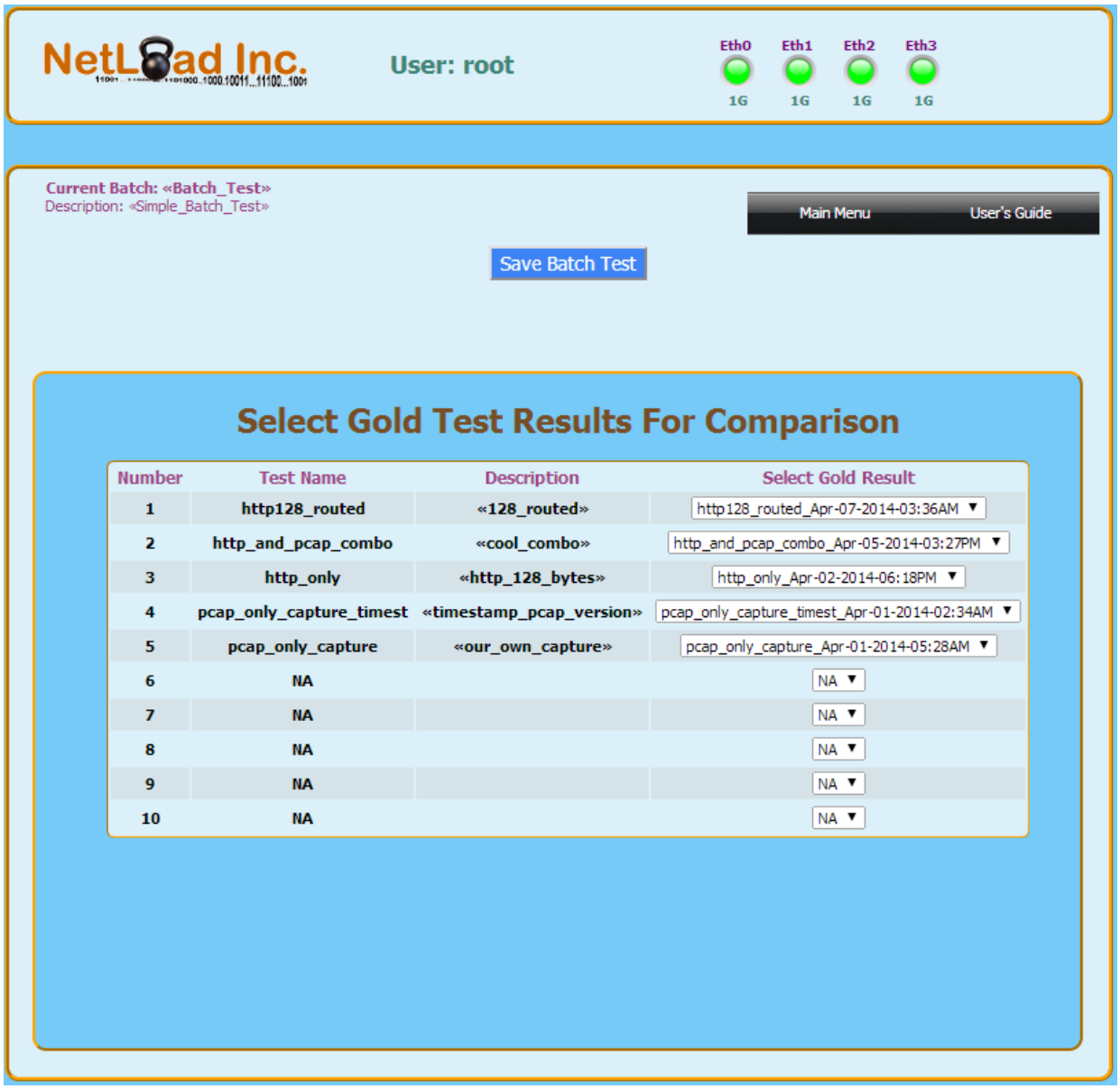


Figure 14 – Select "Gold" Results per Test

4. Select "Gold" Results from all available results for each specific test. If no results are chosen, the test will use the test result as "Gold" selection (see Fig. 14).
5. Select "Save Batch Test".

NOTE: Each test in the Batch must be at least 120 sec in duration.

Test Configuration Management and Backup

This section provides the overview of the steps to backup and upload test configurations from system to system and for backup purposes.

Step	Description
Step 1	Remove Tests
Step 2	Backup and Install Tests

Step 1: Backup Tests

The backup test utility will backup and restore an individual test, or all tests. It does not individually backup Batch tests.

1. From Main Menu, under “Manage Test Configurations” select “Backup Tests”.

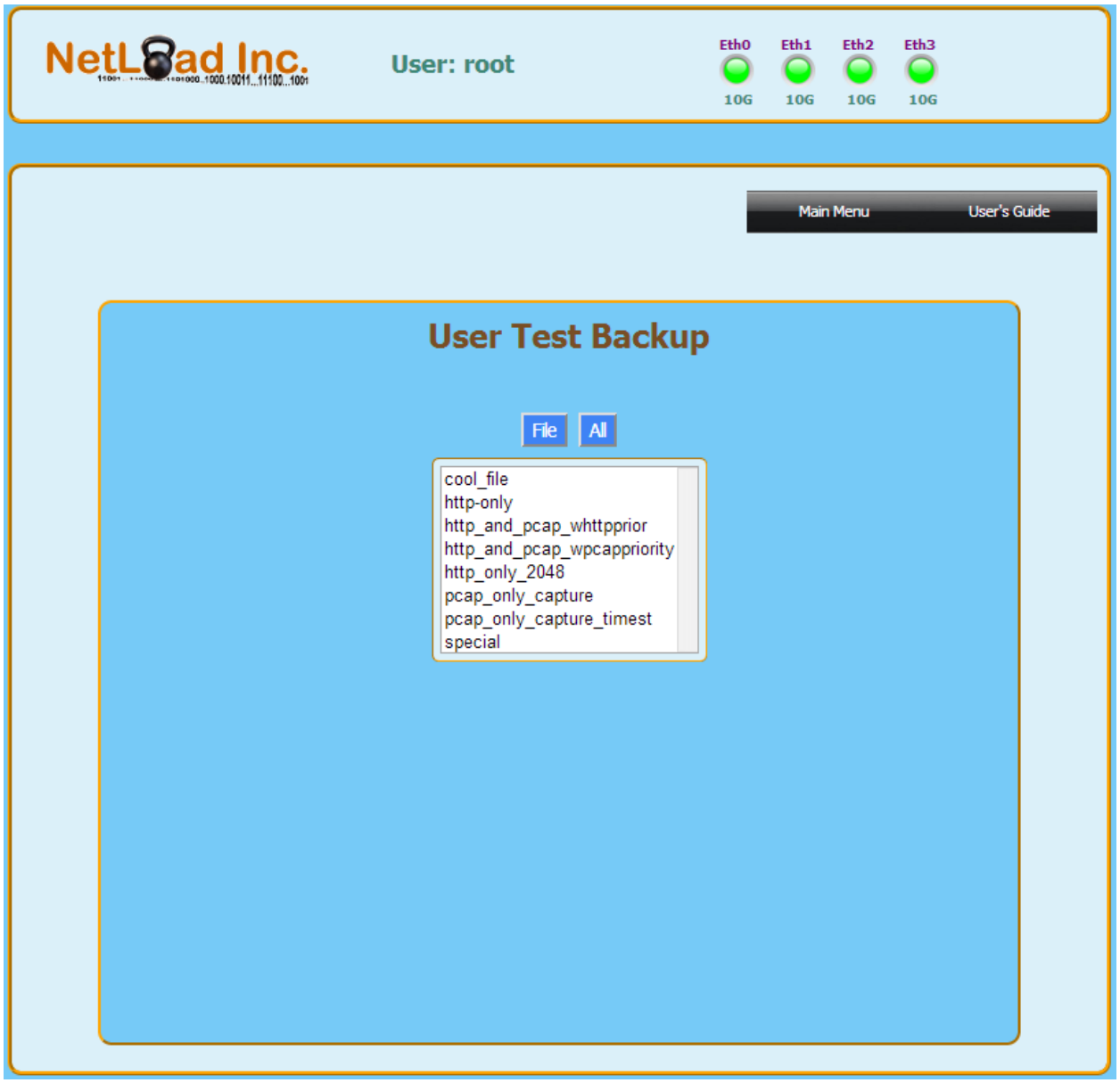


Figure 15 – Backup Tests

2. Select test to back up, or back up all regular and batch tests. Special care must be taken with PCAP Replay Files, as the large PCAP files may create a

very large backup image that won't be installable on the system via Web GUI (see Fig. 15).

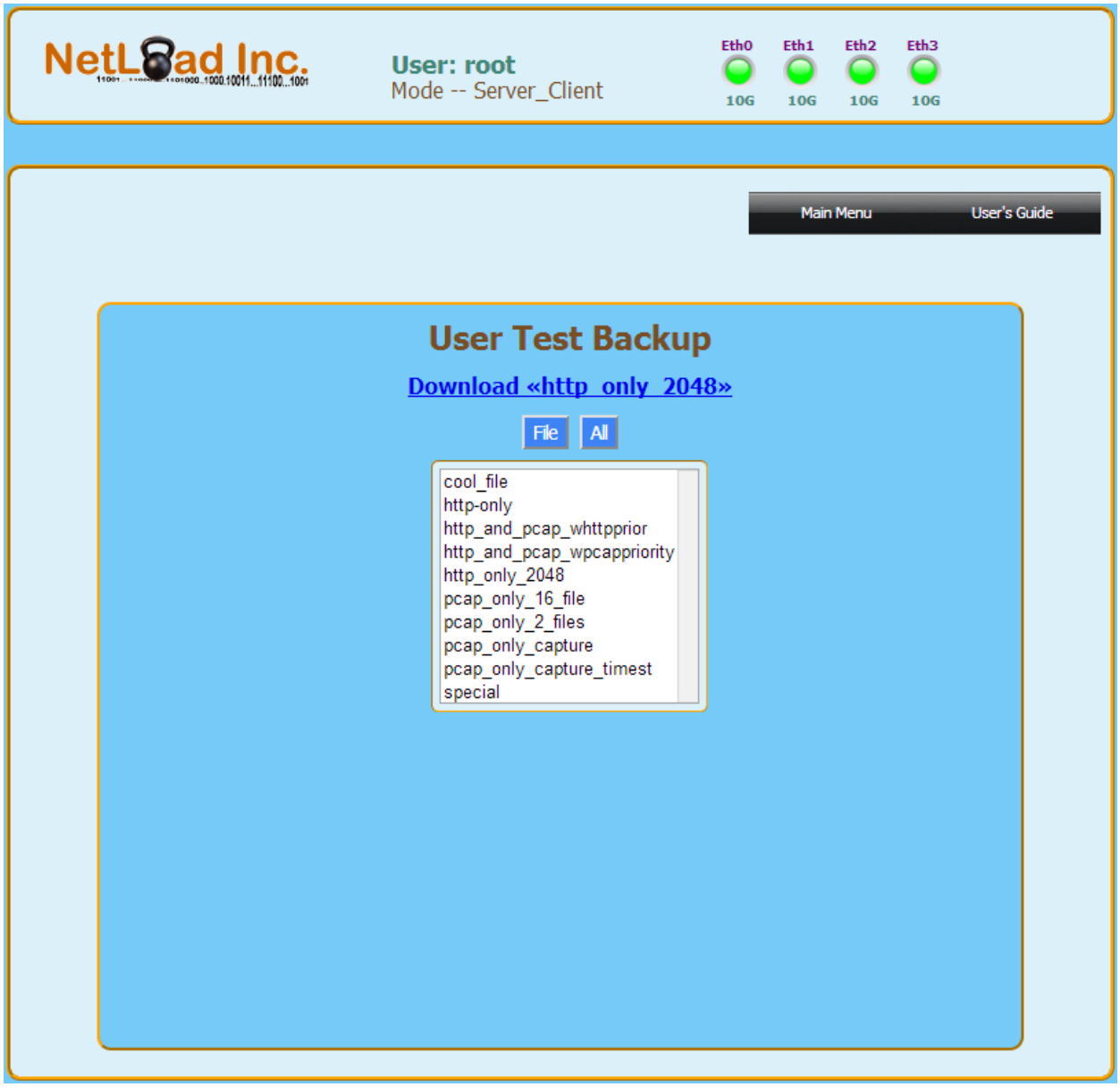


Figure 16 – Download Backup Tests

3. Select and download the backup file (see Fig. 16). This is an encrypted backup of specific test or all tests and associated payload, PCAP, and URL-list files. Results are not included accept for "Gold" results associated with Batch tests.

Step 2: Upload and Install Tests

The backup test utility will backup and restore an individual test, or all tests. It does not individually backup Batch tests.

1. From Main Menu, under “Manage Test Configurations” select “Upload Tests” (see Fig. 17).

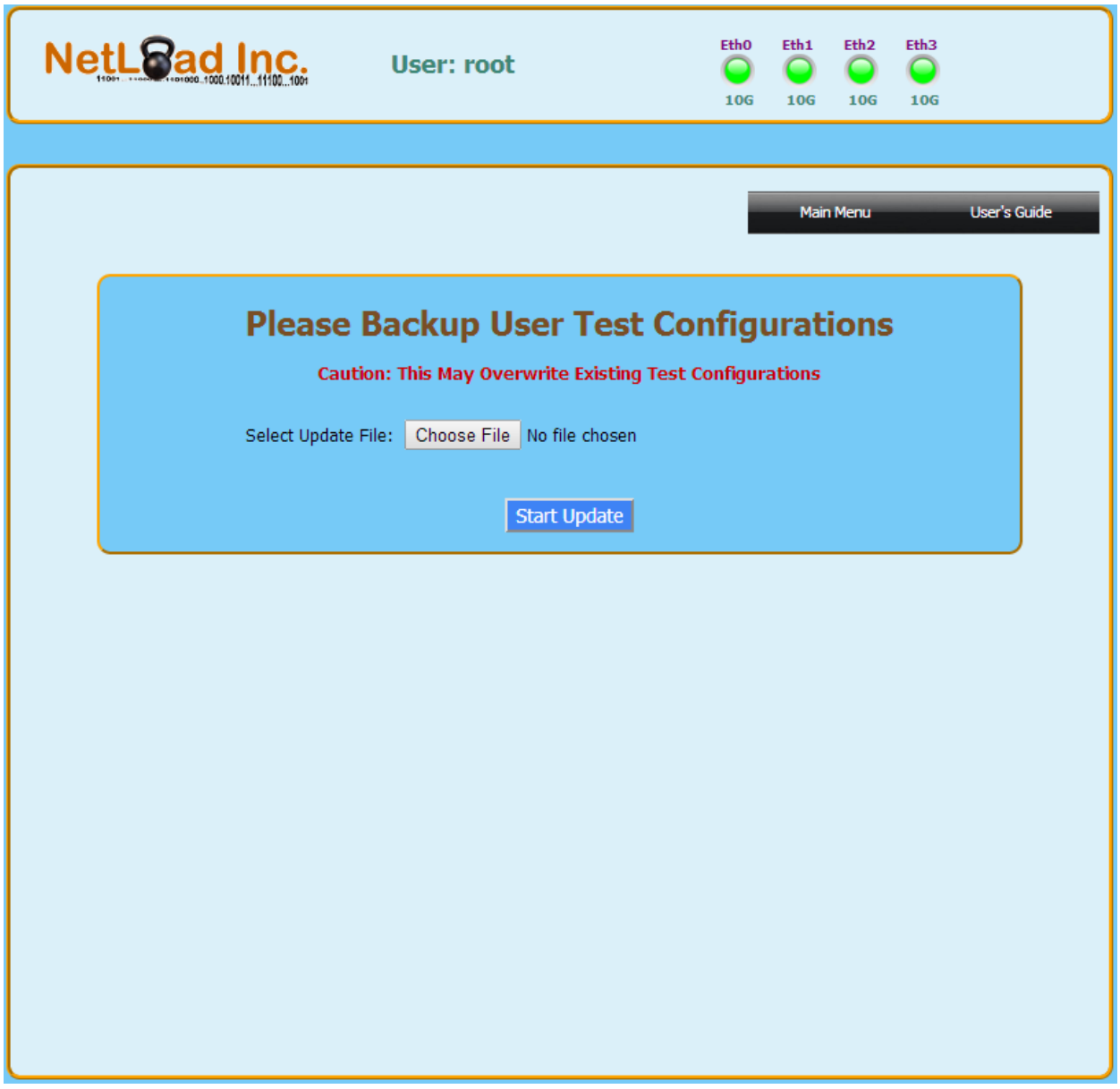


Figure 17 – Upload and Install Tests

2. Select archive to upload and select “Start Update”.

Note: Care must be taken with test files, payload files, PCAP files, and URL list files with same names, as the new files will overwrite the existing ones.

Running Tests

This section provides the overview of the steps to load and run test configurations.

Step	Description
Step 1a	Load and Run Single Test
Step 1b	Load and Run Batch Test
Step 2	View Running Test Status
Step 3	View All Results
Step 4	Results Management
Step 5	Download PCAP Capture File

Step 1a: Load and Run Single Test

This step will load user configuration and start the test.

1. From main menu, select “Manage Test Execution”.
2. Select “Set System Mode”. This will open a small window that allows the selection of DUT configuration. If “System Mode” is changed, the system will reboot for new setting to take effect.
    a. “In-Line” – DUT is an in-line device that forwards traffic from one port to another
    b. “Mirror” – DUT is a passive device that wants to monitor the traffic. In this mode Server and Client ports need to be attached to the DUT and DUT must receive traffic only without TX. NetLoad tester will create the HTTP transactions internally and mirror these onto the Server and Client ports as TX only.
    c. “PCAP/UDP Optimization” – in this mode a larger share of system resources is allocated to PCAP/UDP traffic and provides improved performance.
3. Select “Run Test”.
4. Select configuration you want to test (see Fig. 18a).

Note: Changes to “System Mode” will require a reboot of the system.

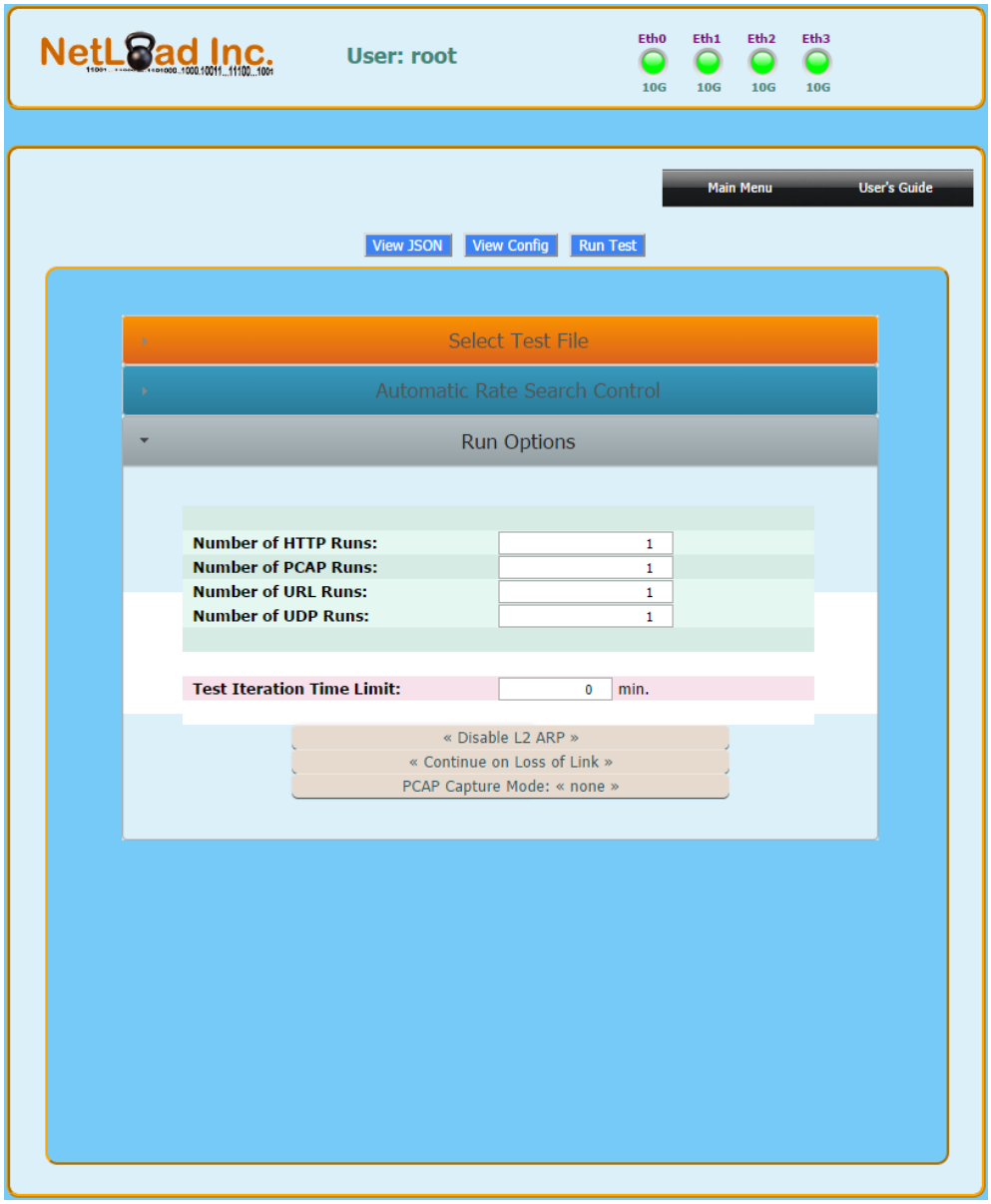


Figure 18a – Run Single Test

5. Under “Automatic Rate Search Control” select “Enable Auto Search” for the system to find the steady-state connection rate through the DUT. Example: selecting Acceptable Loss of ‘0’, the system will find the Connection Rate with zero packet loss (see Fig. 18b).

   Auto Rate Search combines Packet Loss/Connection Loss criteria to find the HTTP/TCP Connection Rate that provides for DUT steady-state operation.

To operate properly, the system will adjust the Ramp Rate and Number of Clients parameters to provide for proper feature operation.

When “Decremental Rate Adjustment” is selected, Auto Rate starts at the top of the performance curve as calculated by the system and reduces the rate by certain percentage upon loss detection. The test is restarted every time loss criteria are met.
When “Incremental Rate Adjustment” is selected, Auto Rate starts at the first “Auto Step Size” selected and increases the rate by this percentage until loss detection. When loss is detected, the test will back-off to the previous percentage rate and hold the rate for duration of the test. The test is NOT restarted every time.

Figure 18b – Automated HTTP Rate Search

6. Select the number of runs of each traffic type to run. For small PCAP, URL, or UDP files, a larger number of runs may be needed to generate higher traffic rates. For UDP, a large number of RUNs may be needed to generate continuous traffic.
7. Use "Test Iteration Time Limit" to limit your test to a certain time. Enter "0" to run test to estimated time. Enter a value in seconds to limit the test to this maximum run time.
8. Enable L2 ARPs if your system requires gratuitous L2 ARPs to learn the address of the interfaces (these may not be needed when using "Virtual Router" mode).
9. "Continue on Loss of Link" – based on selection the test will continue or stop based on LOL behavior
10. PCAP Capture Mode – for diagnostic or verification purposes, enable PCAP Capture Mode to capture traffic generated/seen by NetLoad System. See "Download PCAP Capture File" section for more instructions on this feature.
11. "REST API Log" – enables logging of REST API commands for Dynamic User Configuration only.

Step 1b: Load and Run Batch Test

This step will load user configuration and start the test.

1. From main menu, select “Manage Current Test”.
2. Select “Set System Mode”. This will open a small window that allows the selection of DUT configuration. If “System Mode” is changed, the system will reboot for new setting to take effect.
    a. “In-Line” – DUT is an in-line device that forwards traffic from one port to another
    b. “Mirror” – DUT is a passive device that wants to monitor the traffic. In this mode Server and Client ports need to be attached to the DUT and DUT must receive traffic only without TX. NetLoad tester will create the HTTP transactions internally and mirror these onto the Server and Client ports as TX only.
    c. “PCAP/UDP Optimization” – in this mode a larger share of system resources is allocated to PCAP/UDP traffic and provides improved performance.
3. Select “Run Batch Test”.
4. Select Batch Test you want to run (see Fig. 19).

Note: Changes to “System Mode” will require a reboot of the system.

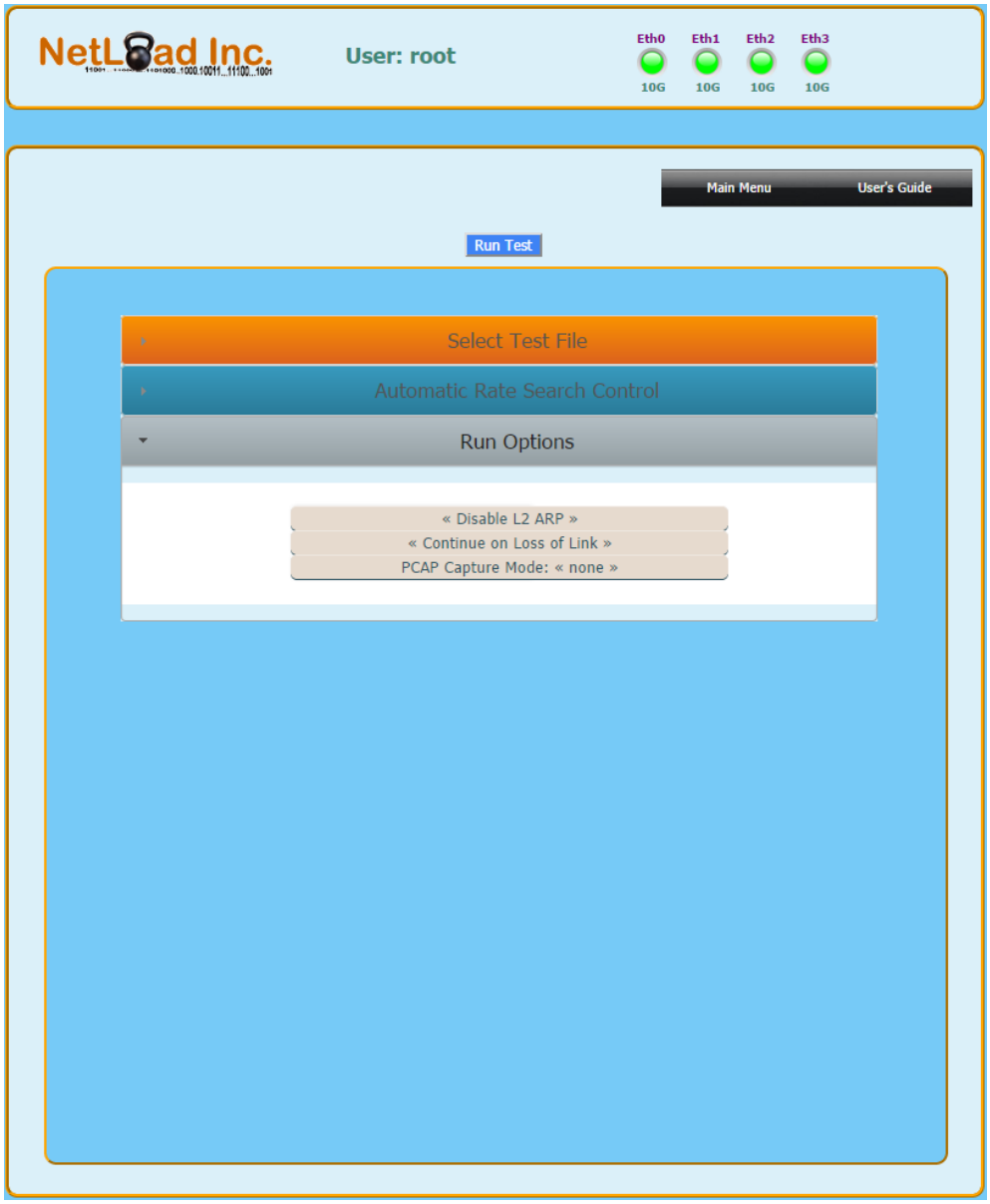


Figure 19 – Run Batch Test

5. “Enable Gratuitous ARP” for Layer 2 and Layer 3 devices. This will advertise the system MAC address on each port.
6. “Stop Test on Link Loss” parameter if checked will stop running test if loss of link is encountered on the test ports. If NOT checked, the test will continue running even if the ports are physically unplugged and link is lost.
7. Select “Run Test”.

Step 2: View Running Test

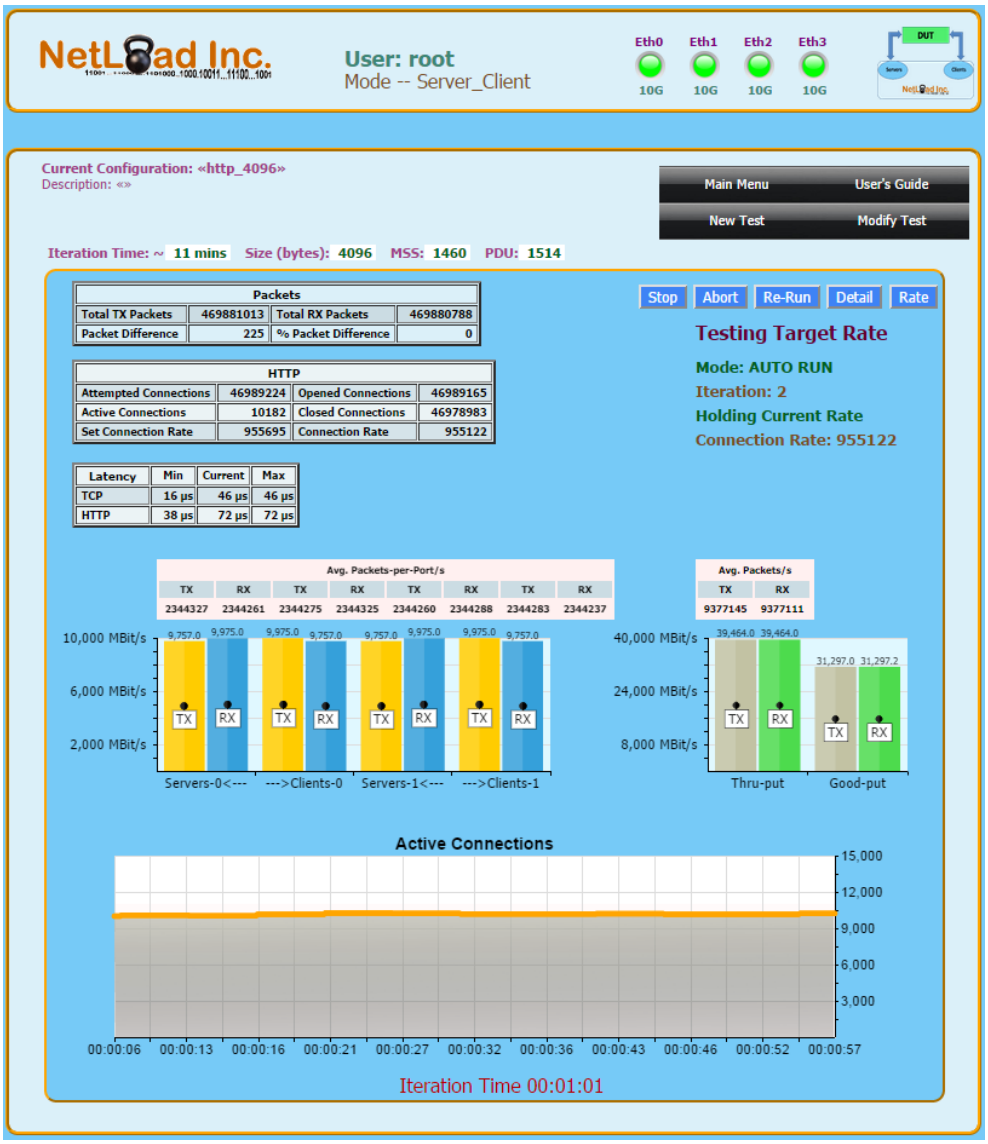


Figure 20a – View Running Individual HTTP Test

Close Window | Freeze Stats | Resume Stats

File Name: < all_traffic_transparent > Wed-Sep-17-22:08:03-2014

	Client Stats			Server Stats		
	Total	Rate	Max	Total	Rate	Max
PACKET_SENT	3209400	86328	86449	3214626	86328	86498
BYTES_SENT	894887700	24071365	24125839	836976086	22431118	22485824
BYTES_AND_IPG_SENT	971913300	0.209146Gb	0.209605Gb	914127110	0.196024Gb	0.196485Gb
SYN_SENT	1069800	28776	28816	0	0	0
ACK_SENT	1069800	28776	28816	534900	14388	14428
WGET_SENT	534900	14388	14428	0	0	0
HTTP_POST_REQUEST_SENT	534900	14388	14428	0	0	0
HTTP_DATA_SENT_BYTES	547737600	14733459	14774312	547737600	14733459	14774459
PACKET_RCV	3214626	86328	86498	3209400	86328	86449
BYTES_RCV	836976086	22431118	22485824	894887700	24071365	24125839
BYTES_AND_IPG_RCV	914127110	0.196024Gb	0.196485Gb	971913300	0.209146Gb	0.209605Gb
SYN_ACK_RCV	1069800	28776	28816	0	0	0
RST_RCV	1059800	28775	28817	0	0	0
HTTP_POST_RESPONSE_RCV	534900	14388	14428	0	0	0
HTTP_DATA_FINISH_RCV	1069800	28776	28816	534900	14388	14428
HTTP_RESPONSE_RCV	534900	14388	14428	0	0	0
HTTP_DATA_RCV_BYTES	547737600	14733459	14774459	547737600	14733459	14774312
PACKET_RCV_BCAST_OR_MCAST	1	0	0	0	0	0
PACKET_RCV_WRONG_MAC_ADDR	77	0	49	0	0	0
INCORRECT_PIP_PORT	88	1	1	0	0	0
PACKET_REPLAY_SENT	0	0	0	78	0	49
SYN_ACK_SENT	0	0	0	1069800	28776	28816
RST_SENT	0	0	0	1059800	28775	28817
HTTP_RESPONSE_SENT	0	0	0	534900	14388	14428
HTTP_POST_RESPONSE_SENT	0	0	0	534900	14388	14428
SYN_RCV	0	0	0	1069800	28776	28816
ACK_RCV	0	0	0	1069800	28776	28817
WGET_RCV	0	0	0	534900	14388	14428
HTTP_POST_REQUEST_RCV	0	0	0	534900	14388	14428
PACKET_RCV_WRONG_ETH_PORT	0	0	0	8	0	7

ACTIVE CONNECTIONS	10000
TX-RX PACKET DIFF	0

	UDP Client Stats			UDP Server Stats		
	Total	Rate	Max	Total	Rate	Max
PACKET_RCV_UDP	15148	0	8453	0	0	0
PACKET_UDP_SENT	0	0	0	15148	0	8453

	URL Client Stats			URL Server Stats		
	Total	Rate	Max	Total	Rate	Max
PACKET_SENT	40	0	39	16	0	15
BYTES_SENT	4528	0	4525	960	0	959
BYTES_AND_IPG_SENT	5488	0.000000Gb	0.000044Gb	1344	0.000000Gb	0.000011Gb
SYN_SENT	24	0	23	0	0	0
PACKET_RCV	16	0	15	32	0	31
BYTES_RCV	960	0	959	4048	0	4045
BYTES_AND_IPG_RCV	1344	0.000000Gb	0.000011Gb	4816	0.000000Gb	0.000039Gb
SYN_ACK_RCV	16	0	15	0	0	0
SYN_SENT_DROP_CONNECTION	4	0	3	0	0	0
SYN_ACK_SENT	0	0	0	16	0	15
SYN_RCV	0	0	0	16	0	15
URL_DROPPED	0	0	0	16	0	15

LATENCY (us)	CURRENT	MAX	MIN
SYN_2_SYN_ACK	14.62	14.99	14.55
GET_POST_2_RESPONSE	19.52	20.35	17.82

Figure 20b – View Detail

1. From Main Menu, select “View Running Test”. The display shows all the pertinent stats including attempted, open connections, and closed connections, bandwidth on each port and direction, packet differences, overall thru-put and good-put, and state and time of the current test iteration. For URL List and FW testing, the stats include attempted URLs and pertinent responses as selected by the user-uploaded URL List file.
2. Select “Detail” to view “raw” detailed run-time information.
3. Select “Rate” to adjust rate on a running test. Rate Scalar can be any number 0-100,000 with steps of .001.

4. Select “Abort” to hard stop the existing test. All open sessions will remain open and no results will be saved.
5. Select “Stop Test” to gracefully shut down a running test. The test will wind down all the open connections and save the results based on existing data.

Note: This option is not available for Batch Test execution.

6. Select “Re-Run” to re-run the selected test again.
7. To modify the setup and re-run the test, select “Modify Test”.

Note: You can move throughout other menus while a test is running. You can also logout out of the system, and login at a different time to view the test and final results.

Note: This option is not available for Batch Test execution.

Note: Tests with less than 10 seconds of execution time will not have their results saved as in general these are for debug purpose only.

Step 3a: View Individual Results

This allows you to view previous test runs.

1. From “Main Menu” select “Manage Test Results” and “View All Results”.
2. Select the base test results you want to view (see Fig. 21).

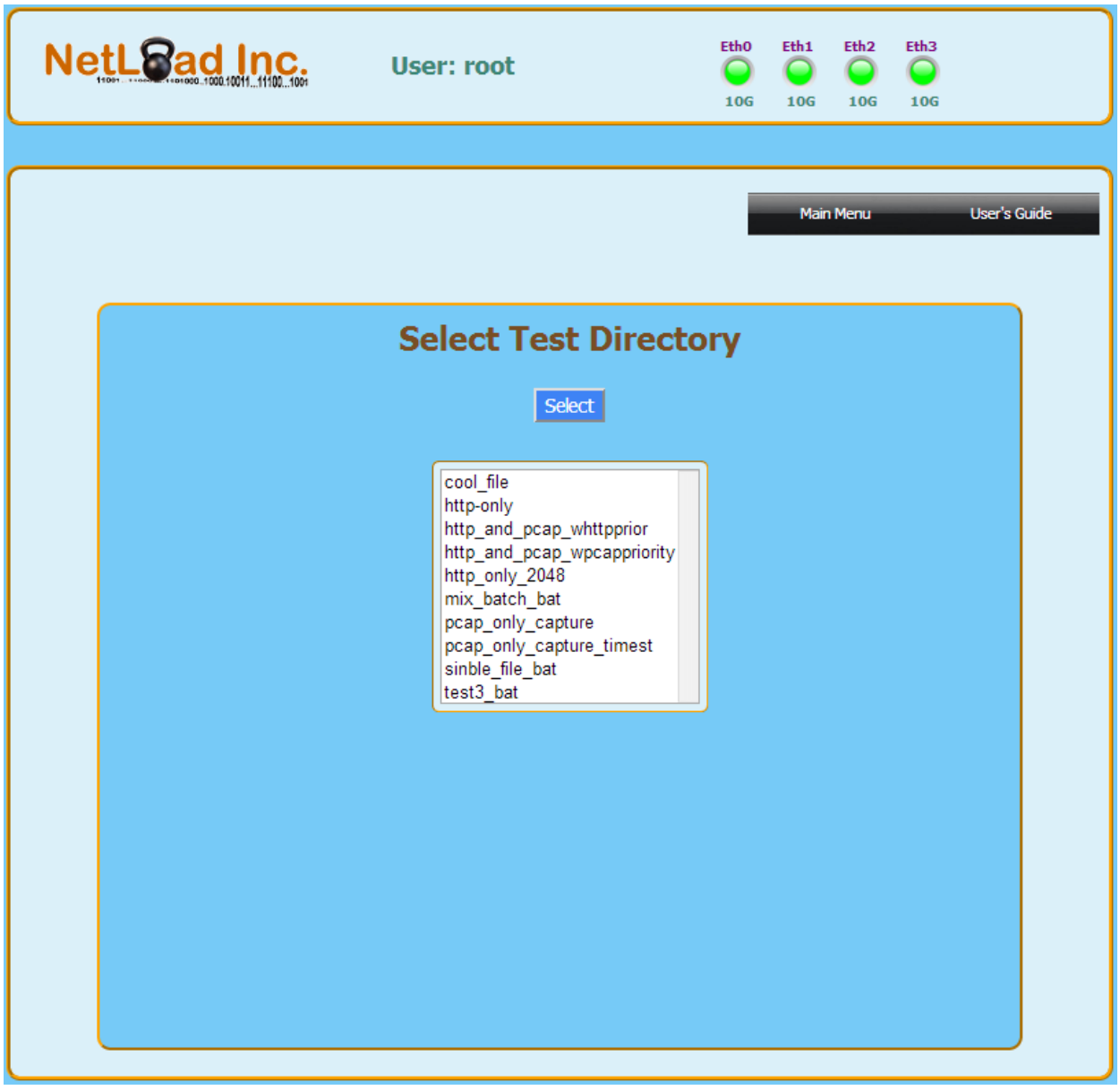


Figure 21 – Select Base Test to View

3. Select the specific result based on name and date (see Fig. 22).

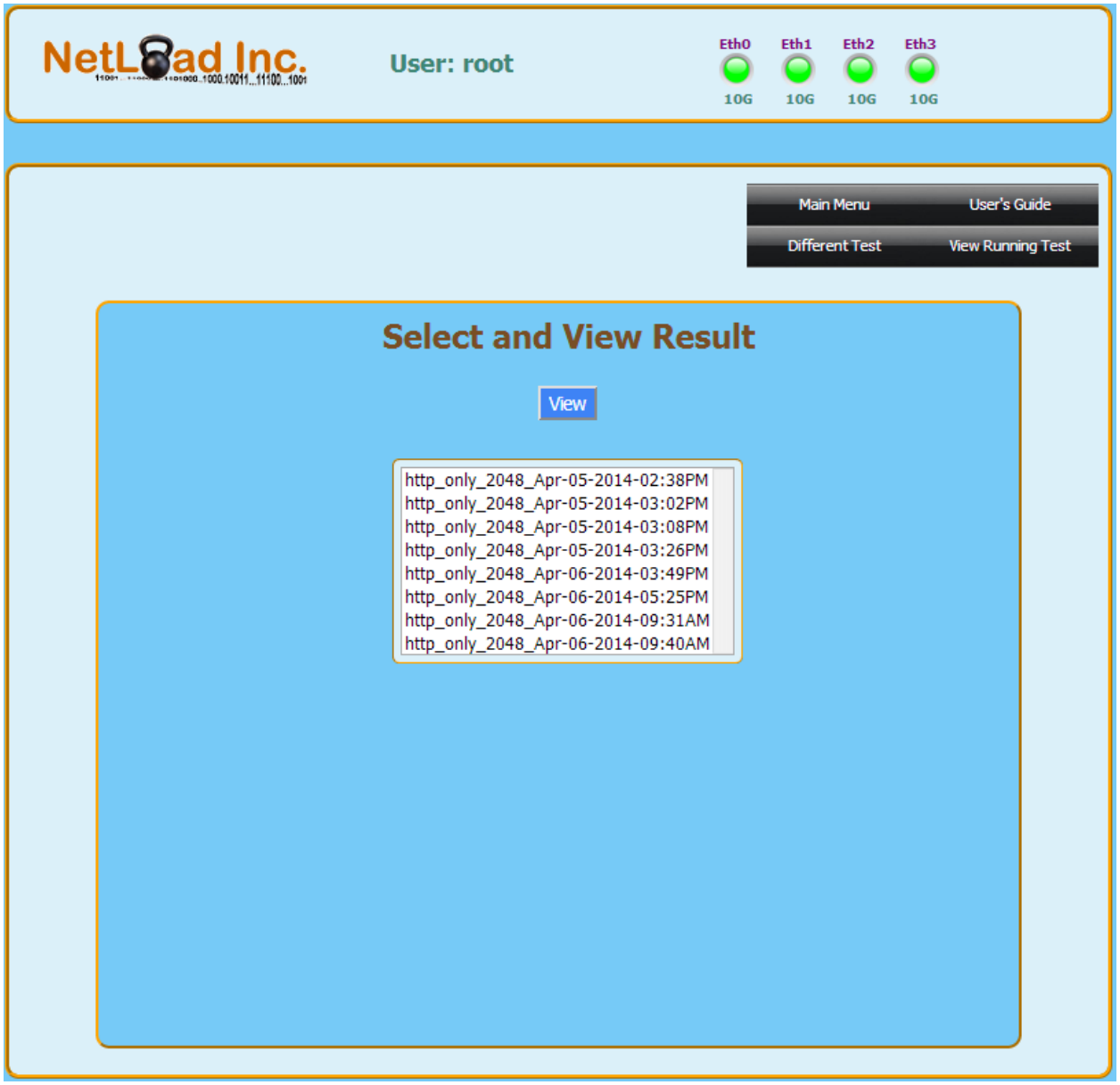


Figure 22 – Select Results to View

4. View Test Results for the selected Test Run (see Fig. 23).

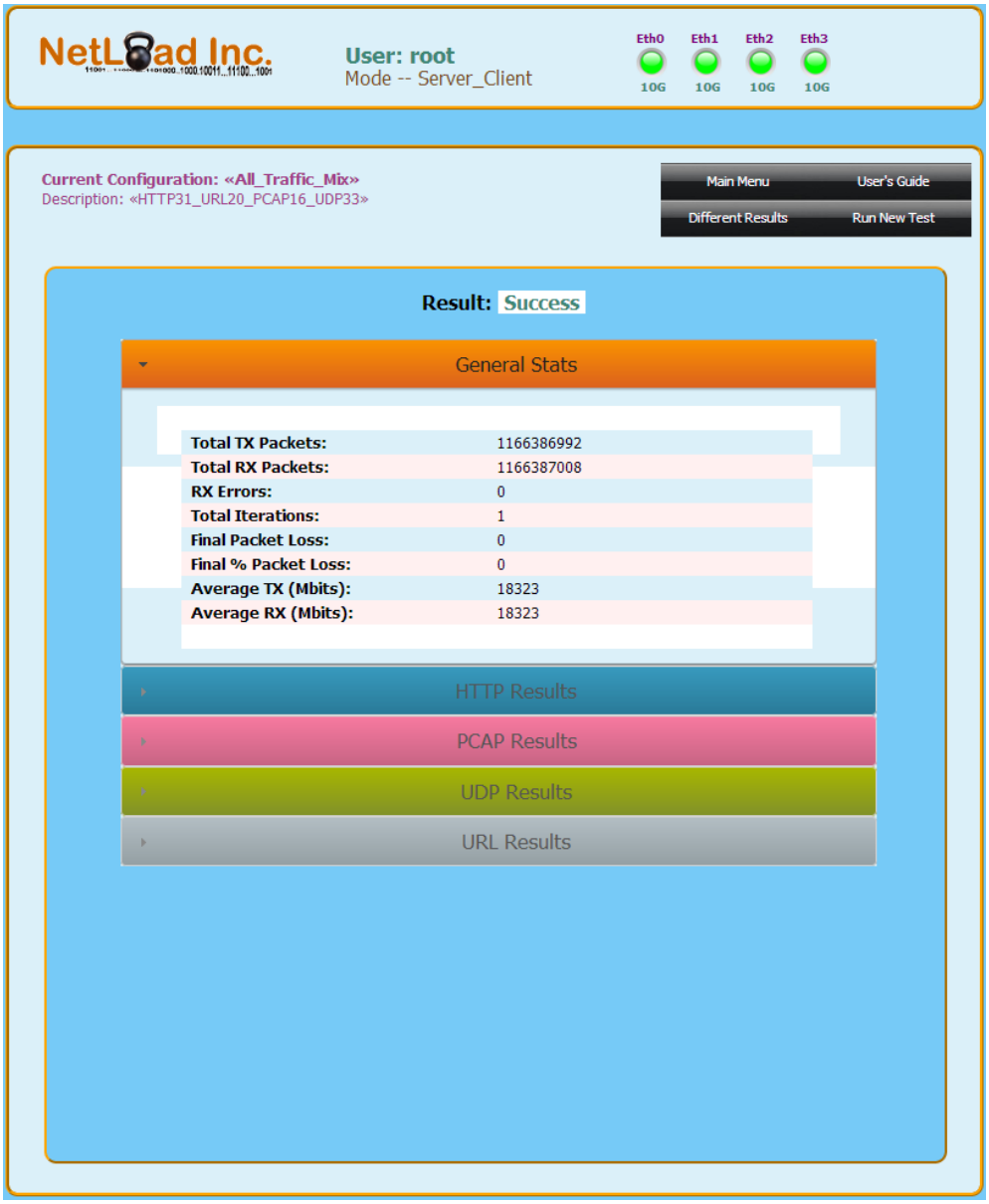


Figure 23 – View Test Results

5. Major information about the selected test is displayed in the each Test Results Table for each traffic type.

Step 3b: View Batch Test Results

This allows you to view previous test runs.

1. From “Main Menu” select “Manage Test Results” and “View All Results”.
2. Select the base Batch test results you want to view. Batch tests are designated with base name and “_bat” at end of the name.
3. Select the specific Batch Test based on the name and date.
4. View Test Results for the selected Batch Test Run (see Fig. 24).

Figure 24 – View Batch Test Results

5. For further details about each individual test run and comparison against "Gold" results, select "Details" and select "View Details". The detailed results of a particular test will be displayed (see Fig. 25).

Figure 25 – View Comparison Details of each test in Batch

Step 4: Results File Management

This allows you to manage previous test runs.

1. From “Main Menu” select “Manage Test Results” and “Download Results” or “Remove Previous Results”.
2. Select the base test for which you want to download (or remove) results. The files are stored in a .JSON format.

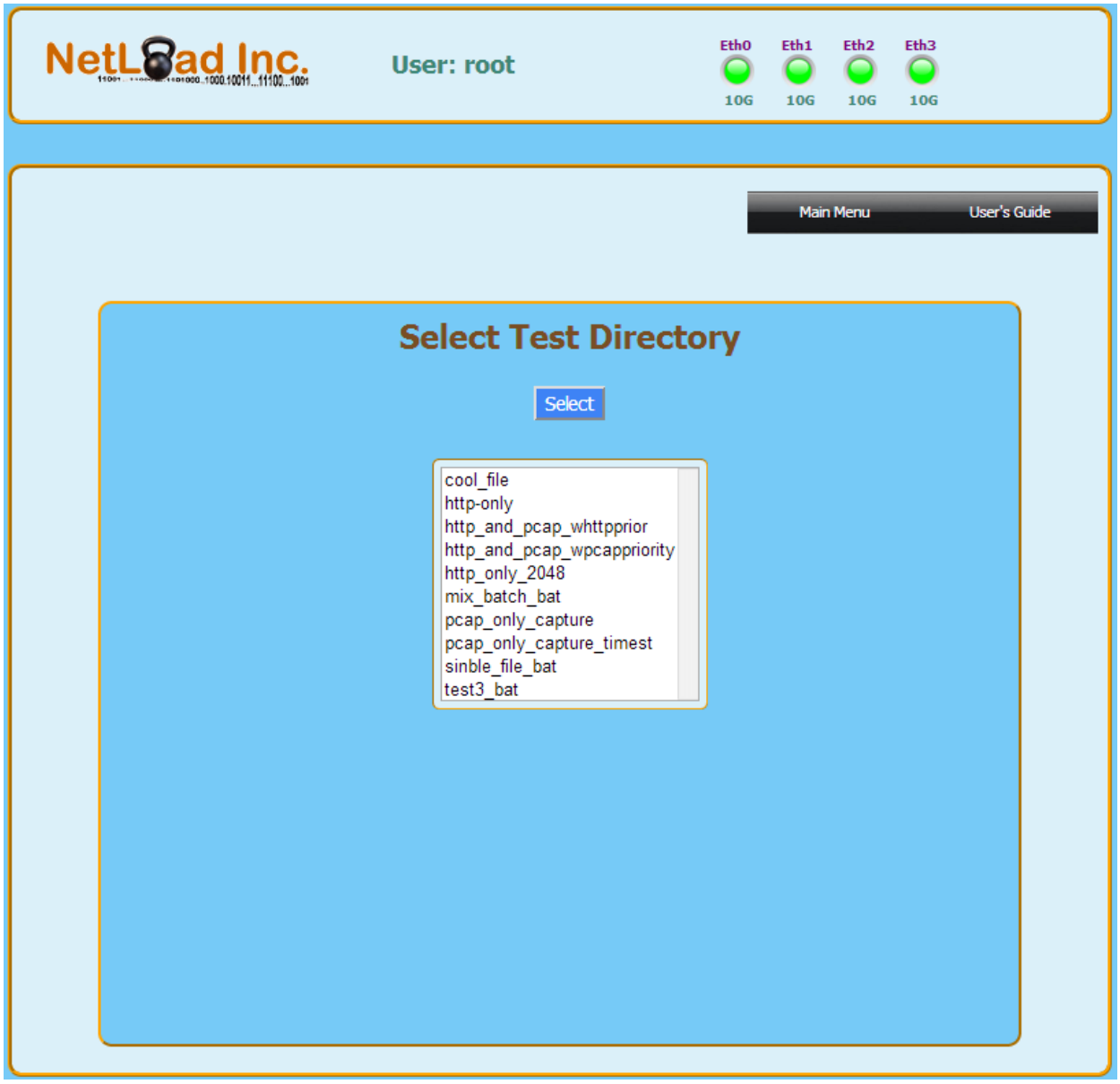


Figure 26 – Select Base Results Directory for Download/Removal

Figure 27 – Download Results Files

3. Download Results. The results will be displayed by the browser in .JSON format which is easily selectable and transportable into other formats.

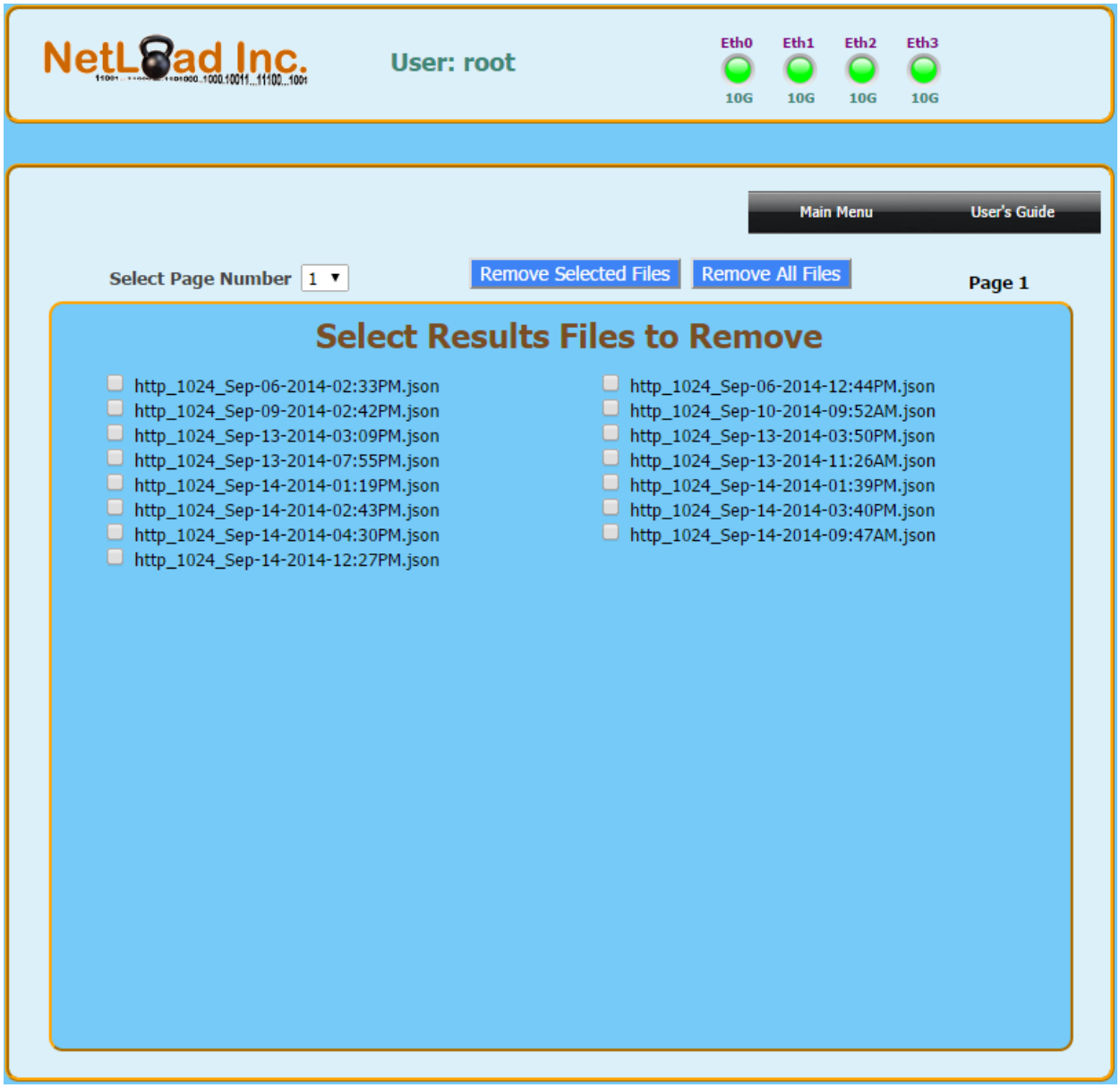


Figure 28 – Remove Old Results Files

4. Select the results to remove. This allows for removal of old results that are no longer needed.

Step 5: Capture PCAP File Download

1. From “Main Menu” select “Manage Test Results” and “Download PCAP File”.
2. Select “NetLoadInc_date.pcap”. This file is only valid for the current test run and is overwritten on subsequent runs due to its size of up to 1Gbyte.

Figure 29 – Download PCAP File

Step 6: Support File Download

1. From “Main Menu” select “Manage Test Results” and “Download Support File”.
2. Select “NetLoad_Support_date.tar.gz”. This file is only valid for the current test run and is overwritten if the command is executed multiple times.

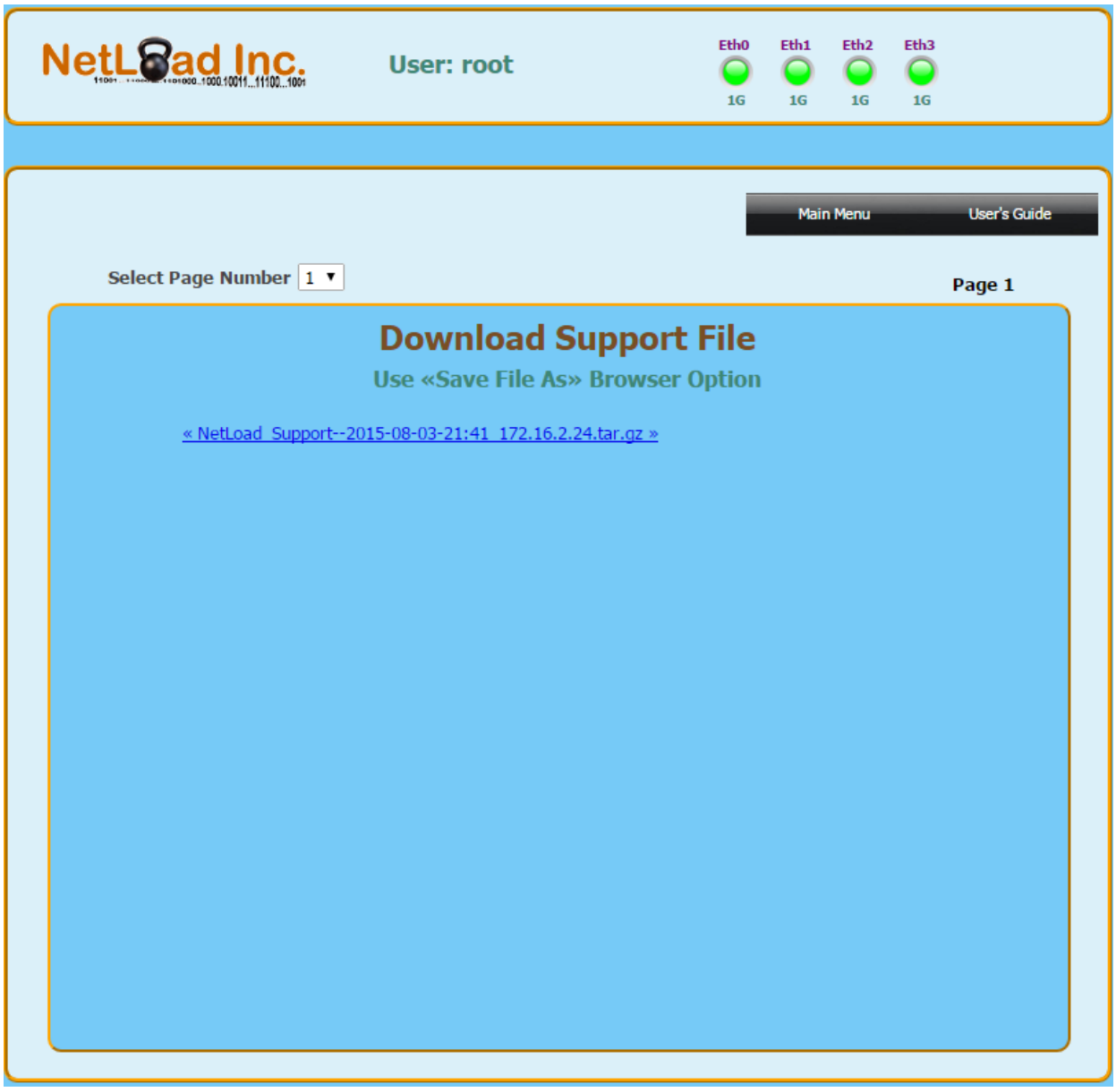


Figure 30 – Download Support File

System Management

This section provides the overview of the system management.

Task	**Description**
Task 1	Restore Defaults
Task 2	Update Firmware
Task 3	Change Login
Task 4	System Shutdown
Task 5	System Reboot

Task 1: Restore Defaults

This step will restore system defaults to factory setup.

Figure 31 – Restore Configuration

Task 2: Firmware Update

This step will update the firmware and run-code on your system. Your test configurations and system setup are preserved.

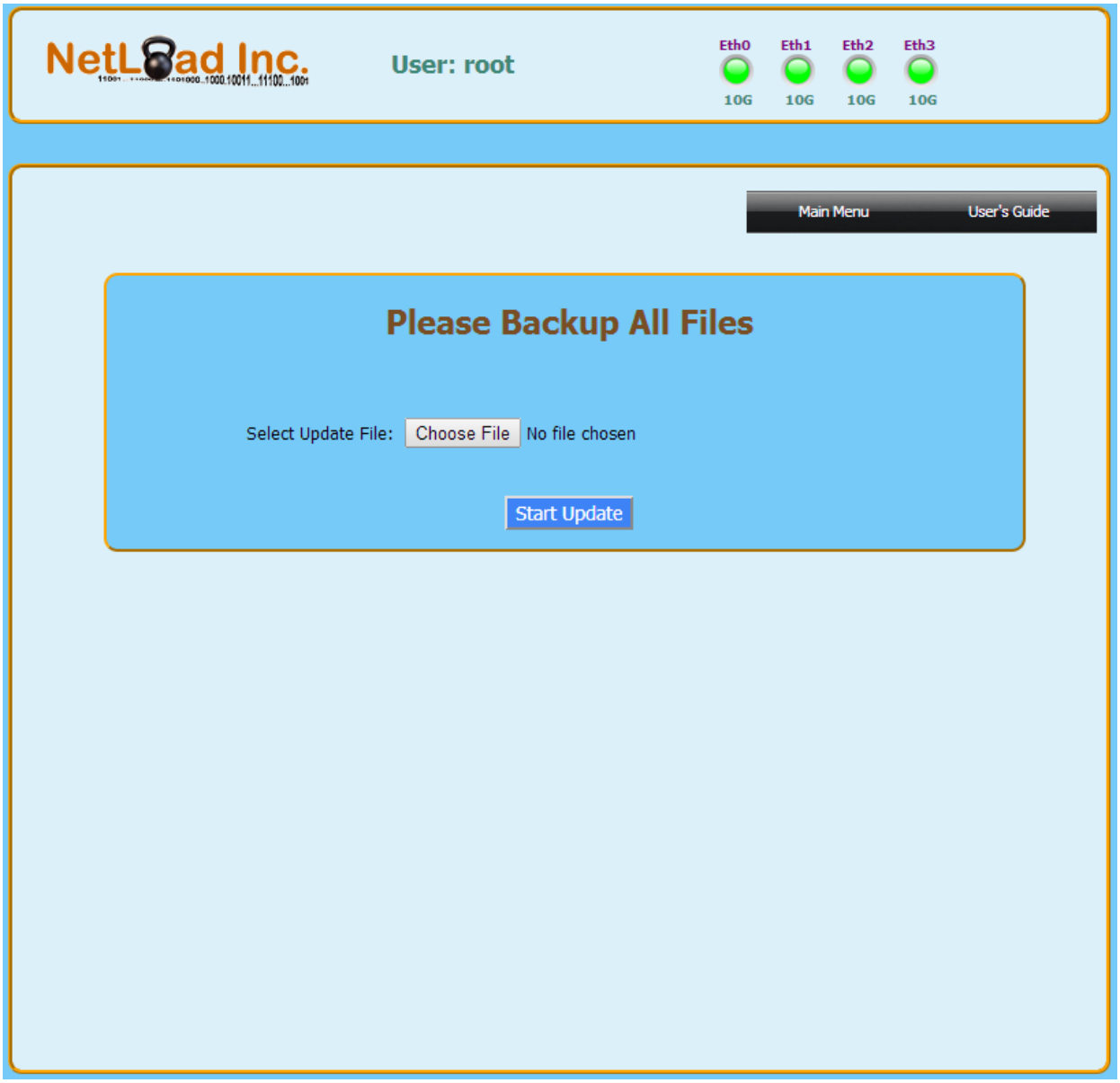


Figure 32 – Update System FW

Download a valid image from NetLoadInc ftp site to a system that is accessible to the NetLoad system. Latest images are available at *www.NetLoadInc.com* with

Username: *netloadsupport* and Password: *Nloadsupport1!*. NetLoad Inc. recommends a tool such as FileZilla to access the FTP site.

1. Select and Download *NetLoadInc*_6Nx .enc* file (where 6Nx is the series of your system). This is an encrypted file that contains the updates for the system.
2. Once the system completes upload and initialization (may take a few minutes), it will reboot.
3. Restart the Web browser once the system is booted
4. Login into the system.

Task 3: Change Login

This allows the user to change the login name and password.

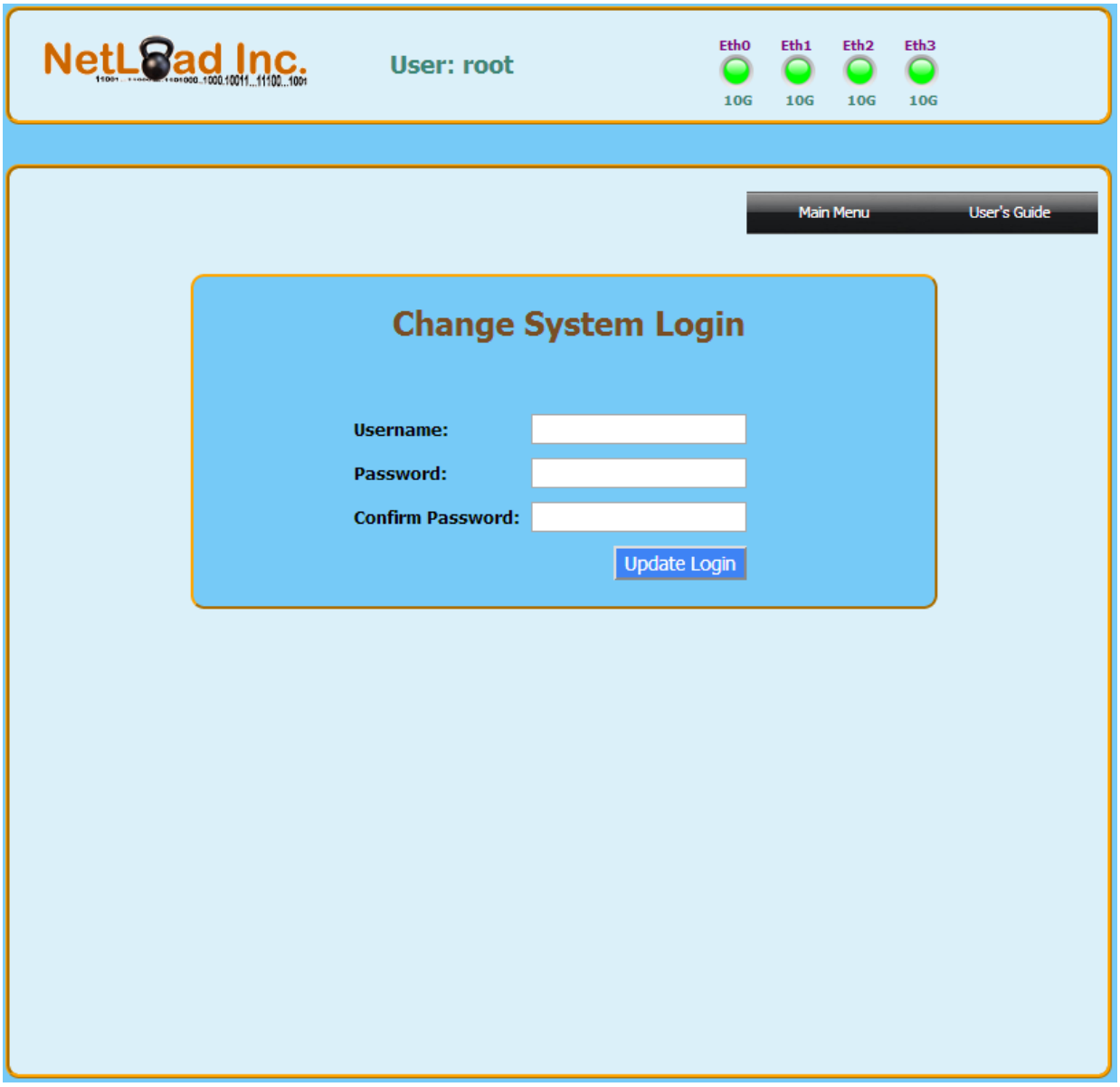


Figure 33 – Change Login

Task 4: System Shutdown

System must be halted before power is turned off to the system.
From “Main Menu” → “Administration” →”Configure System” → ”Halt System”.

Task 5: System Reboot

System reboot can be initiated from the Web GUI.
From “Main Menu” → “Administration” →”Configure System” → ”Reboot System”.

CLI Interface Commands

This section provides the overview of the CLI commands directly on the system.

Task	Description
Task 1	Help
Task 2	List Files
Task 3	Run Test, Run Test with Options, Stop, Abort
Task 4	Modify Rate
Task 5	Show and Clear Statistics, Get Status
Task 6	Configuration File Manipulation

Task 1: CLI: Help

The “nl_cli” command is used to initiate any action for the CLI.

“nl_cli -- help” displays all available CLI commands. Two dashes are used to identify command and/or options.

```
root@NetLoadInc:/home/netload# nl_cli --help
#!/usr/bin/php

--list           List availble tests
--list_batch     List availble batch tests
--run            <Single test  Name>
--run_batch      <Batch test  Name>
--show_config    [Single test  Name] | [current]
--stop           Stop current single test gracefully, or abort Batch or Server-Only Tests
--abort          Abort current test without saving test results
--stats          Show detailed statistics
--status         Show current status
--clear_stats    Clear statistics
--modify_running [--global value] [ [--http value] [--url value] [--replay value] [--udp value]]
--help           help menu
```

Figure 34 – CLI Help

Task 2: CLI: List

"nl_cli -- list" displays all available configuration files.

```
root@NetLoadInc:~# nl_cli --list
#!/usr/bin/php

1024_slow_and_short.cfg
3_http_files.cfg
PCAP_split.cfg
all_traffic_routed.cfg
all_traffic_transparent.cfg
arp_test.cfg
http_1024.cfg
http_1024_separate.cfg
http_1024_transp_vlans.cfg
http_1024_vlans.cfg
http_64.cfg
systestconfig.cfg
udp_and_url.cfg
udp_and_url_and_pcap.cfg
udp_port_0.cfg
url_test.cfg
zero_transaction.cfg

root@NetLoadInc:~#
```

Figure 35 – CLI List

Task 3: CLI: Run Test

"nl_cli – run filename" runs test named "filename".

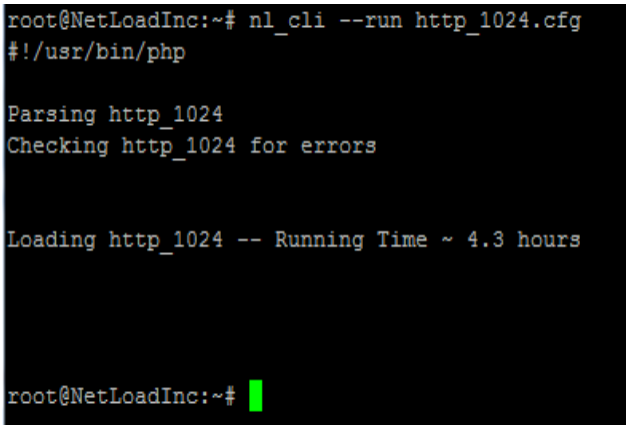


Figure 36a – CLI Run Test

It is useful to enable REST API Logging and/or PCAP capture for debugging purposes prior to generating a Support File. The following command runs the test and enables REST API Logging and PCAP capture for SYN packets.

"nl_cli --run filename --PCAP_capture_on SYN --REST_API_log_on"

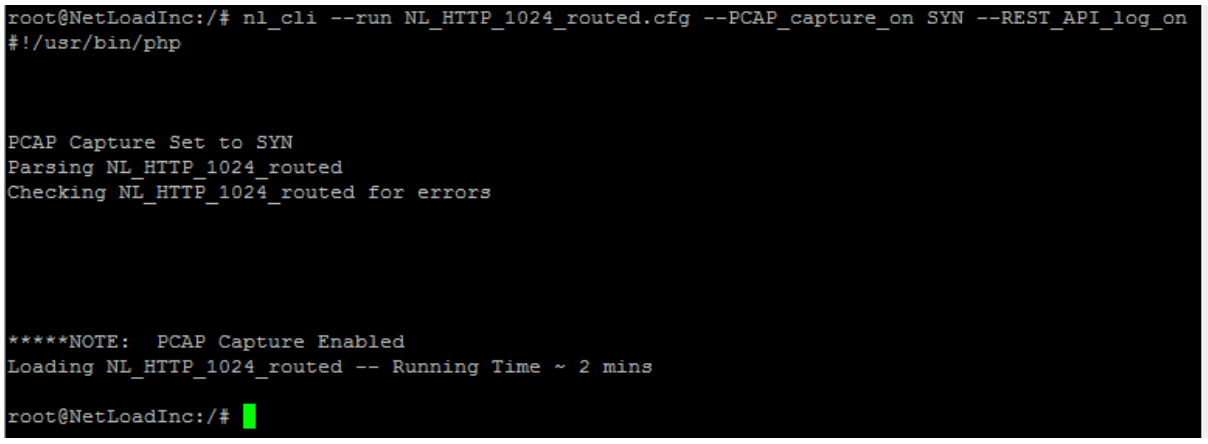


Figure 36b – CLI Run Test with options

Task 4: CLI: Modify Rate of Running Test

"nl_cli – modify_running –option value" changes the rate of existing test by multiplying the rate by a scalar value (0-100,000 in steps of .0001). The "–global value" rate applies to all traffic types.
For Individually Controlled Traffic Rate Setup, "—http value", "—url value", "—udp value", and "—pcap" value can be used to adjust specific traffic rate.

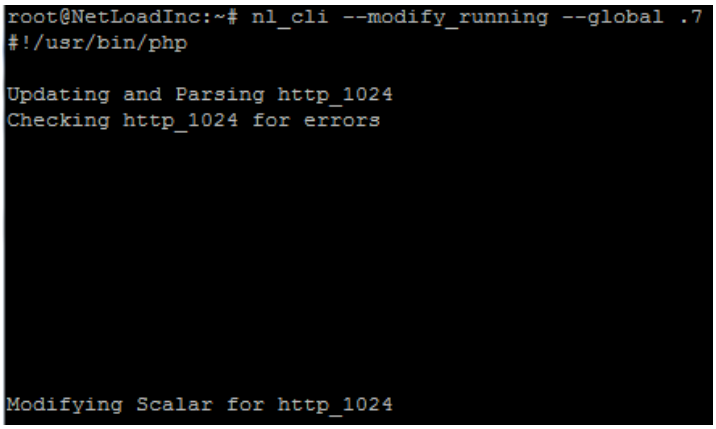


Figure 37 – CLI Modify Rate

Task 5: CLI: Show and Clear Statistics

"nl_cli – stats" displays a full range of statistics for a given test.

"nl_cli – status" displays the current state of running test.

"nl_cli –clear_stats" clears the counters of the running test.

```
root@NetLoadInc:~# nl_cli --stats
#!/usr/bin/php

File Name: < http_64 >  Thu-Sep-18-06:18:40-2014    Test is Complete
 port OUTPUT_PPS OUTPUT_MBPS INPUT_PPS INPUT_MBPS
    eth0          0           0          0
    eth1          0           0          0
    eth2          0           0          0
    eth3          0           0          0

                                        Client Stats                                Server Stats
                                  Total        Rate         Max          Total        Rate         Max

PACKET_SENT                   245759940           0      873907      245759940           0      873905
BYTES_SENT                  29078315120           0   103553846      24453116140           0    86954014
BYTES_AND_IPG_SENT          34976553680  0.000000Gb  0.996214Gb      30351354700  0.000000Gb  0.863420Gb
SYN_SENT                       81919980           0      291323             0           0           0
ACK_SENT                       81919980           0      291313       40959980           0      145662
WGET_SENT                      40960000           0      145662             0           0           0
HTTP_POST_REQUEST_SENT         40959980           0      145659             0           0           0
HTTP_DATA_SENT_BYTES         2621438720           0     9322236       2621440000           0     9322171
PACKET_RCV                    245759940           0      873911      245759940           0      873905
BYTES_RCV                   24453116140           0    86954569      29078315120           0   103554446
BYTES_AND_IPG_RCV           30351354700  0.000000Gb  0.863427Gb      34976553680  0.000000Gb  0.996221Gb
SYN_ACK_RCV                    81919980           0      291311             0           0           0
RST_RCV                        81919980           0      291319             0           0           0
HTTP_POST_RESPONSE_RCV         40959980           0      145663             0           0           0
HTTP_DATA_FINISH_RCV           81919980           0      291313       40959980           0      145662
HTTP_RESPONSE_RCV              40960000           0      145658             0           0           0
HTTP_DATA_RCV_BYTES          2621440000           0     9322161       2621438720           0     9322403
SYN_ACK_SENT                         0           0           0       81919980           0      291313
RST_SENT                             0           0           0       81919980           0      291320
HTTP_RESPONSE_SENT                   0           0           0       40960000           0      145658
HTTP_POST_RESPONSE_SENT              0           0           0       40959980           0      145662
SYN_RCV                              0           0           0       81919980           0      291313
ACK_RCV                              0           0           0       81919980           0      291319
WGET_RCV                             0           0           0       40960000           0      145658
HTTP_POST_REQUEST_RCV                0           0           0       40959980           0      145662

ACTIVE CONNECTIONS                   0
TX-RX PACKET DIFF                    0

LATENCY (us)                   CURRENT         MAX         MIN
SYN_2_SYN_ACK                     3.92        5.25        3.63
GET_POST_2_RESPONSE              20.63       21.59       20.26

root@NetLoadInc:~#
```

Figure 38 – CLI Statistics

Task 6: CLI: Accessing/Copying/Modifying Configuration, Results, and other User-accessible Files

All files accessible by user are stored in **/home/netload** directory. The sub-directories contain user configuration files, PCAP files, transaction/payload files, test results files, and batch files. Care must be taken when working with these files as missing files may cause invalid configurations that will not work properly.
User configuration files are stored in .JSON format in **/home/netload/configs/usertest** directory on the system. Special care must be taken if file modification is done via CLI.
As standard .JSON files, the configuration files can be changed by the user via scripts or CLI to create variants. An example of a full traffic profile configuration file shows all traffic types. Traffic-type specific parameters may not be present if traffic type was not selected during configuration creation.
The naming of parameters follows the GUI naming conventions. Care should be taken to make sure the format, naming, and values are valid. Invalid values will cause the test to be aborted.


```
{
    "global": {
        "File_Name": "all_traffic_routed",
        "File_Description": "all_4_traffic_types",
        "HTTP_Rate_Percentage": 25,
        "PCAP_Rate_Percentage": 25,
        "UDP_Rate_Percentage": 25,
        "URL_Rate_Percentage": 25,
        "Primary_Traffic": "None",
        "Individual_Port_Bandwidth_Control": "Disabled",
        "System_Configuration_Mode": "Single NetLoad Clients-Servers",
        "Port_Configuration_Mode": "Routed Mode",
        "Select_NAT_Type": "None",
        "Total_Port_Bandwidth": 825
    },
    "global_ip_network_info": {
        "TCP_Port_Number_Start": 32768,
        "TCP_Port_Number_End": 57404,
        "IPv4_Address_Sequence": "Increment",
        "Number_of_Servers": 100,
        "Number_of_Clients": 100,
        "Number_of_Client_TCP_Ports": 256,
        "Active_Connections": 10000,
        "Server_IPv4_Pool_Netmask": "255.255.255.128",
        "Client_IPv4_Pool_Netmask": "255.255.255.128",
```

```
        "Virtual_Router_Server_Interface_Netmask": "255.255.255.240",
        "Virtual_Router_Client_Interface_Netmask": "255.255.255.240"
    },
    "ip_port_info": [
        {
            "Server_IPv4_Addr": "5.0.0.1",
            "Virtual_Router_Server_IPv4_Interface_Addr": "16.1.0.1"
        },
        {
            "Virtual_Router_Server_IPv4_Interface_Addr": "16.2.0.1"
        },
        {
            "Client_IPv4_Addr": "172.0.0.1",
            "Virtual_Router_Client_IPv4_Interface_Addr": "16.1.0.2",
            "Virtual_Router_Client_IPv4_Gateway_Interface_Addr": "16.1.0.1",
            "Port_Enable": "Enable"
        },
        {
            "Virtual_Router_Client_IPv4_Interface_Addr": "16.2.0.2",
            "Virtual_Router_Client_IPv4_Gateway_Interface_Addr": "16.2.0.1",
            "Port_Enable": "Disable"
        }
    ],
    "transaction_file_info": [
        {
            "File_Name": "1024_Bytes.txt",
            "Weight": 1,
            "Mime_Type": "application/x-directory; charset=binary"
        }
    ],
    "get_post_global": {
        "Ramp_Time": 15,
        "Burst_Rate": 20,
        "File_Directory": "/var/www/html/transactionfiles/",
        "Number_of_Runs": 1,
        "Zero_Data_Transaction": "Disable",
        "GETs_POSTs_Balance": 50,
        "Session_Termination": "RST",
        "Session_Optimization": "Shortest Session",
        "Session_MSS": 1460,
        "Round_Trip_Delay": 0
    },
    "url": {
        "File_Name": "/var/www/html/urlfiles/url2.txt",
        "Ramp_Time": 15,
        "Burst_Rate": 20,
        "TCP_Port_Number_Start": 57405,
        "TCP_Port_Number_End": 65535
    },
    "replay_global": {
        "Ramp_Time": 15,
        "Burst_Rate": 20,
        "File_Directory": "/var/www/html/pcap/",
        "Time_Stamp": "Rate Control",
        "Time_Stamp_Scaler": 1,
        "Split_Delay": 0,
```

```
        "Max_Packet_Rate": 53694,
        "Final_Packet_Rate": 44297
    },
    "replay_info": [
        {
            "File_Name": "tbot_FC7C3E.pcap",
            "Port_Number": 0,
            "Split_File": "Don't Split",
            "Weight": 1
        }
    ],
    "udp_global": {
        "Ramp_Time": 15,
        "Burst_Rate": 20,
        "File_Directory": "/var/www/html/replay/",
        "Max_Packet_Rate": 118371,
        "Final_Packet_Rate": 97656
    },
    "udp_info": [
        {
            "File_Name": "3000000_Bytes.txt",
            "Port_Number": 0,
            "File_Size": 3000000,
            "Selected_Packet_Sizes": "64:128:256:512",
            "Drop_Fragment_out_of_Number_of_Fragments": 0,
            "Set_UDP_Fragmentation": "Don't Fragment"
        }
    ],
    "run_options": {
        "Enable_Auto": "Disable Auto Search",
        "GR_ARP": "Disable L2 ARP",
        "Acceptable_Loss": 0,
        "Number_of_Runs_HTTP": 1,
        "Number_of_Runs_URL": 1,
        "Number_of_Runs_PCAP": 1,
        "Number_of_Runs_UDP": 1,
        "Stop_on_LOL": "Continue on Loss of Link",
        "Test_Time_Limit": 0,
        "Auto_Scale": "Scale All Traffic",
        "Capture_Direction": "none"
    }
}
```


Figure 39 – JSON Configuration File

REST API Commands (curl examples)

This section provides the commands for REST API.

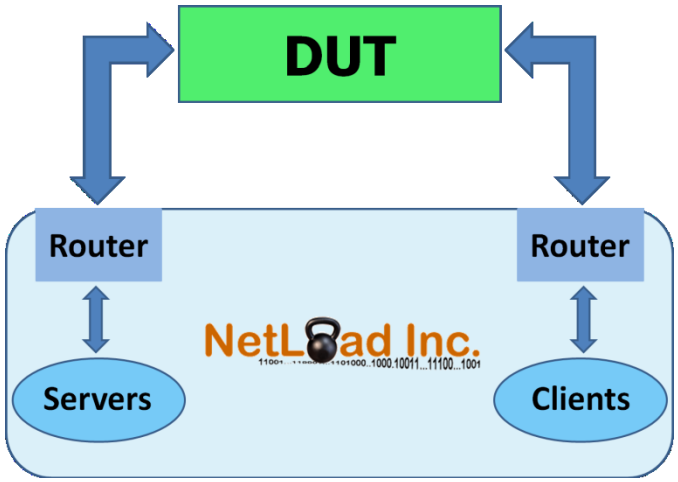
Commands	Description
curl -X GET http://192.168.1.1/nl/Stats *curl -X GET http://192.168.1.1/nl/Stats_JSON*	Show Statistics
curl -X GET http://192.168.1.1/nl/Status	Get Status and State
curl -X GET http://192.168.1.1/nl/Mac_Address *curl -X GET http://192.168.1.1/nl/Mac_Address_JSON*	Get MAC addresses of Test Ports Eth0-Eth3
curl -X GET http://192.168.1.1/nl/Clear_Stats	Clear Statistics
curl -X POST --header "Content_Type: application/json" -d @New_File.cfg http://192.168.1.1/nl/Post_Config/New_File.cfg	Upload Test Configuration file "New_File.cfg" in JSON format
curl -X GET http://192.168.1.1/nl/Get_Config --data "Existing_File.cfg" -o Existing_File.cfg	Download Test Configuration File "Existing_File.cfg" into Existing_File.cfg
curl -X POST --header "Content_Type: application/json" -d @users.json http://192.168.1.1/nl/Post_User_File/users.json	Upload User File "users.json" in JSON format
curl -X GET http://192.168.1.1/nl/Get_User_File --data " users.json *" –o users.json*	Download User File "users.json" into file users.json
curl -X POST --header "Content_Type: image/JPEG" -d @payload.jpg http://192.168.1.1/nl/Post_Transaction_File/payload.jpg	Upload Transaction file "payload.jpg"
curl -X GET http://192.168.1.1/nl/Get_Transaction_File --data "payload.jpg" -o payload.jpg	Download Transaction File "payload.jpg" into payload.jpg
curl -X POST --header "Content_Type: text/plain" –data-binary @urls.txt http://192.168.1.1/nl/Post_URL_File/urls.txt	Upload URL file "urls.txt"
curl -X GET http://192.168.1.1/nl/Get_URL_File --data-binary " urls.txt" -o urls.txt	Download URL file "urls.txt" into urls.txt
curl -X POST --header "Content_Type: text/plain" –data-binary @data.pcap http://192.168.1.1/nl/Post_PCAP_File/data.pcap	Upload PCAP file "data.pcap"
curl -X GET http://192.168.1.1/nl/Get_PCAP_File --data-binary	Download PCAP file

"data.pcap" -o data.pcap	"data.pcap" into data.pcap
curl -X POST http://192.168.1.1/nl/Run --data "Existing_File.cfg"	Run Test "Existing_File.cfg"
curl -X POST http://192.168.1.1/nl/Run --data '{"File_Name":"Existing_File.cfg","Capture_Direction":"RX","REST_API_Log":true} '	Run Test "Existing_File.cfg" with PCAP capture enabled and set to RX and REST API Log enabled
curl -X POST http://192.168.1.1/nl/Stop	Stop Current Test
curl -X POST http://192.168.1.1/nl/Abort	Abort Current Test
curl -X POST http://192.168.1.1/nl/Reload	Re-run Current Test
curl -X POST http://192.168.1.1/nl/Modify_running/global --data "N"	Increase Connection Rate for running test by factor of N
curl -X GET http://192.168.1.1/nl/List_Configs	List Available Test Configurations
curl -X GET http://192.168.1.1/nl/List_Transaction_Files	List Available Transaction Files
curl -X GET http://192.168.1.1/nl/List_URL_Files	List Available URL Files
curl -X GET http://192.168.1.1/nl/List_PCAP_Files	List Available PCAP Files
1. *curl -X GET http://192.168.1.1/nl/Create_Support_Info_File* 2. *curl -X GET http://192.168.1.1/nl/Get_Support_Info_File --data-binary "NetLoad_Support-xxxxxx.gz" -o NetLoad_Support-xxxxxx.gz*	Create and Download Support Info File (2 parts) 1. Create and list support file 2. Use response from part (1) as name of file to download
curl -X GET http://192.168.1.1/nl/Create_and_Get_Support_Info_File -o NetLoad_Support-`date "+-%F-%H:%M"`.tar.gz	Create and Download Support Info File (1 part) with sample name
curl -X POST http://192.168.1.1/nl/Set_Power_On_Config --data "Existing_File.cfg"	Set default configuration that will be executed on power-on

curl -X POST http:// 192.168.1.1/nl/Clear_Power_On_Config	Clear default configuration. Once executed, there will be no configuration executed on power-on
curl -X POST --header "Content_Type: x-gzip" --data-binary @NetLoadInc_revXXX_6NX.11_Oct_2015.2b4567bb3b293fd8386cb05f79d7bc55.tar.gz.enc http:// 192.168.1.1/nl/Firmware_Update/NetLoadInc_revXXX_6NX.11_Oct_2015.2b4567bb3b293fd8386cb05f79d7bc55.tar.gz.enc	Upgrade System Firmware with the latest version

Appendix A: Virtual Router <–> DUT configuration.

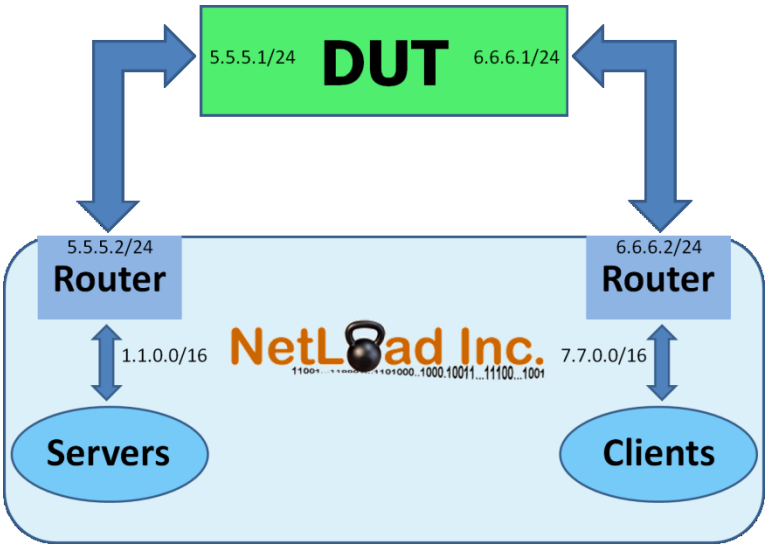
The test setup is as shown below when Virtual Router is enabled (single box scenario shown):



Sample Configuration Scenario:

NetLoad Test Clients	NetLoad Virtual Router	DUT Client Port	DUT Server Port	NetLoad Virtual Router	NetLoad Test Servers
1.1.0.0/16	5.5.5.2/24	5.5.5.1/24	6.6.6.1/24	6.6.6.2/24	7.7.0.0/16

The resulting configuration will look as below:



Static Routes on DUT:

The following Static Routes have to be added to the DUT.

ip route add via 5.5.5.2 1.1.0.0/16
ip route add via 6.6.6.2 7.7.0.0/16

ARP Discovery:

NetLoad Virtual Router 5.5.5.2 → requests DUT 5.5.5.1
DUT 5.5.5.1 → requests NetLoad Virtual Router 5.5.5.2

DUT 6.6.6.1 → requests NetLoad Virtual Router 6.6.6.2
NetLoad Virtual Router 6.6.6.2 (may) request 6.6.6.1

Appendix B: Adding Static Routes to Linux.

How to add Static Routes to Red Hat (RHEL)/CentOS/Fedora Linux

By using the ip command, you can setup and view static route. For example, to display current routing table you can type command:

```
# ip route show
```

Sample output:

```
192.168.2.0/24 dev eth1 proto kernel  scope link  src 192.168.2.1
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.2
default via 192.168.1.254 dev eth0
```

You can add static route using following command:

ip route add {NETWORK} via {IP} dev {DEVICE}

For example network 192.168.55.0/24 available via 192.168.1.254:

```
# ip route add 192.168.55.0/24 via 192.168.1.254 dev eth1
```

Alternatively, you can use old good route command:

```
# route add -net 192.168.55.0 netmask 255.255.255.0 gw 192.168.1.254 dev eth1
```

Linux Persistence Routes

The drawback of 'ip' or 'route' command is that, when Linux reboots it will forget static routes. So store them in configuration file. Static routing describes a system that does not implement adaptive routing. In these systems routes through a data network are described by fixed paths (statically). These routes are usually entered into the router by the system administrator

Red Hat (RHEL) / CentOS / Fedora Linux Persistence Static Routing

You need to open /etc/sysconfig/network-scripts/route-eth0 file to define static routes for eth0 interface:

```
# cat /etc/sysconfig/network-scripts/route-eth0
```

Sample Output:

```
GATEWAY0=192.168.1.254
NETMASK0=255.255.255.0
ADDRESS0=192.168.55.0
GATEWAY1=10.164.234.112
NETMASK1= 255.255.255.240
ADDRESS1=10.164.234.132
```

How do I define static routing for network 10.0.0.0/8 via 10.9.38.65 router?

Open /etc/sysconfig/network-scripts/route-eth0:

```
# vi /etc/sysconfig/network-scripts/route-eth0
```

Append following line:

```
10.0.0.0/8 via 10.9.38.65
```

Save and close the file. Restart networking:

```
# service network restart
```

Verify new routing table:

```
# route -n
```

How to add Static Routes to Debian/Ubuntu Linux

Open configuration file /etc/network/interfaces

```
# cat /etc/network/interfaces
```

Output:

```
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.254
up route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.1
down route del -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.1
```

Debian / Ubuntu Linux static routing for two interfaces:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
        address 10.9.38.76
        netmask 255.255.255.240
        network 10.9.38.64
        broadcast 10.9.38.79
        ### static routing ###
        post-up route add -net 10.0.0.0 netmask 255.0.0.0 gw 10.9.38.65
        pre-down route del -net 10.0.0.0 netmask 255.0.0.0 gw 10.9.38.65
auto eth1
iface eth1 inet static
        address 204.186.149.140
        netmask 255.255.255.240
        network 204.186.149.128
        broadcast 204.186.149.143
        gateway 204.186.149.129
        # dns-* options are implemented by the resolvconf package, if
installed
        dns-nameservers 10.0.80.11 10.0.80.12
        dns-search nixcraft.in
```

For Multicast Routes, the following may be used.

Open /etc/rc.local:

```
vi /etc/rc.local
```

Append command:

```
route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0
```

Reboot system:

```
reboot
```

Verify route

```
route -n
```

Appendix C: Routing with VLAN tagging

Routing with multihomed Linux when using VLAN tagging on RHEL

1 Configure the Interfaces

We want to create VLAN7 tagged interface on our eth0.

First, get the VLAN7 tagged interface (on eth0) up and running, create a `ifcfg-eth0.7` network-scripts file:

```
[root@rhel55 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0.7
DEVICE=eth0.7
BOOTPROTO=static
DHCPCLASS=
ONPARENT=no
NETMASK=255.255.252.0
IPADDR=158.167.99.40
VLAN=yes
USERCTL=NO
PEERDNS=NO
```

2 Default routing configuration on RHEL

Once the configuration has been done, and the interface set up, here is how the IP configuration looks on a RHEL5 server:

```
[root@rhel55 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:F5:50:0C
          inet addr:10.199.99.65  Bcast:10.199.99.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:5161 (5.0 KiB)

eth0.7    Link encap:Ethernet  HWaddr 08:00:27:F5:50:0C
          inet addr:158.167.99.40  Bcast:158.167.99.255  Mask:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:2649 (2.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6338 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6338 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9892090 (9.4 MiB)  TX bytes:9892090 (9.4 MiB)
```

The tagged interface name slightly differs from the one used on Solaris, but that's easy to handle.

The main difference comes with the routes. Here is how they are displayed on RHEL5:

```
[root@rhel55 ~]# netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask          Flags    MSS Window  irtt
Iface
10.199.99.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
158.167.96.0     0.0.0.0          255.255.252.0    U           0 0           0
eth0.7
169.254.0.0      0.0.0.0          255.255.0.0      U           0 0           0
eth0.7
0.0.0.0         10.199.99.254   0.0.0.0         UG        0 0          0 eth0
```

Note: If you want to remove the APIPA 169.254.0.0 route, add `NOZEROCONF=yes` to the `/etc/sysconfig/network` file. This would give the following route configuration:

```
[root@rhel55 ~]# netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask          Flags    MSS Window  irtt
Iface
10.199.99.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
158.167.96.0     0.0.0.0          255.255.252.0    U           0 0           0
eth0.7
0.0.0.0         10.199.99.254   0.0.0.0         UG        0 0          0 eth0
```

So as previously mentioned, the routes on RHEL5 differs from the one you are used to on Solaris 10. And one might be wondering how the traffic is routed with such route definitions. Looking at the output of `netstat -rn` above, one can conclude that any incoming traffic on the `158.167.96.0` interface will be answered back on the `10.199.99.0` network, due to the gateways being set to `0.0.0.0` and the default route being from `0.0.0.0` to `10.199.99.254`.

Unfortunately, this is not exactly how one would expect a multi-homed interface to run 🙂

3 Use of the Linux Advanced Routing capability

In order to get rid of this "unattended" behavior, one has to configure what is called "Linux Advanced Routing"
This will allow to define the VLANs, and to set the routes for each VLAN, specifying the gateway and the interface that is used. In a few words, this is a mean to get a configuration close to one can be used to running Solaris (for this specific need).

3.1 Name the VLANs

The rt_tables is used to map a number (arbitrary) to a string.
This mapping will then be used to identify the routes.

```
[root@rhel55 ~]# cat /etc/iproute2/rt_tables
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
7       vlan7
199     vlan199
```

3.2 Creation of the routes

The routes are created with the definition of two files per route. The first one (`rule-IFNAME`) is the rule file, defining which routing table entry will be used to route from a specific network.

```
[root@rhel55 ~]# cat /etc/sysconfig/network-scripts/rule-eth0
```

```
from 10.199.99.0/24 table vlan199
[root@rhel55 ~]# cat /etc/sysconfig/network-scripts/rule-eth0.7
from 158.167.96.0/22 table vlan7
```

Then the routing table itself will be populated. For each "physical" interface, the outgoing interface as well as the network to reach is defined:

```
[root@rhel55 ~]# cat /etc/sysconfig/network-scripts/route-eth0
table vlan199 to 10.199.99.0/24 dev eth0
table vlan199 to default via 10.199.99.254 dev eth0
[root@rhel55 ~]# cat /etc/sysconfig/network-scripts/route-eth0.7
table vlan7 to 158.167.96.0/22 dev eth0.7
table vlan7 to default via 158.167.96.254 dev eth0.7
```

3.3 Verifying the routes

A `netstat -rn` will show the exact same output as previously. In order to display the routing table, use the following command:

```
[root@rhel55 ~]# ip route show
10.199.99.0/24 dev eth0  proto kernel  scope link  src 10.199.99.66
158.167.96.0/22 dev eth0.7  proto kernel  scope link  src 158.167.98.11
default via 10.199.99.254 dev eth0
```

Appendix D: User Mode (Dynamic and User File)

Note: This Feature is supported for Performer Series Only

User Mode adds a concept of "users" onto an existing TCP/HTTP client-server transaction test. A "user" has an associated set of parameters that are used to generate this specific "user's" transactions.

The "**global_info"** parameter applies to all users. This parameter defines the Maximum bandwidth the user will use for GET (download) and POST (upload) transactions, and PCAP Replay. This parameter is in Mbits/sec and can be included in with user addition/deletion, or by itself. The Maximum bandwidth will be modified for all active users dynamically during test any time this parameter is transferred. Combined with "Weight" attribute for each user, this provides control over bandwidth per groups of users, or even per user.

The following configuration in the "**user_info"** represents two "users" coming from a single source. Each "user" has:

1. "Eth Index" offset parameter to allocate this "user" to a specific client-server port pair. "0" allocates the "user" to port pair 0 (Eth0-servers/Eth2-clients), while "1" will allocate "user" to port pair 1 (Eth1-servers/Eth3-clients).
2. "User ID" that is unique to each user.
3. "User Server IPv4 Address" and "User Client IPv4 Address" are the initial addresses for transactions the "user" will make. The number of transactions is the combination of number of servers, clients, and TCP ports assigned for the test (same as for non-user TCP/HTTP transaction testing).
4. "Client_Tunnel_Info" is used for GTP configurations and must be left blank if GTP is not used.
5. "Server_Tunnel_Info" is used for GTP configurations and must be left blank if GTP is not used.
6. "Weight" is an optional parameter that can allocate total bandwidth to each "user" relative to other "users" on the same client-server port pair.

```
{
    "global_info": {
        "get": {
            "Max_Bandwidth_per_User_Mbits": 0.1
        },
        "post": {
            "Max_Bandwidth_per_User_Mbits": 0.1
        },
        "user_replay": {
            "Max_Bandwidth_per_User_Mbits": 0.3
        }
    },
    "user_info":
      [
        {
            "Eth_Index":   0,
            "User_Id":     5,
            "User_Server_IPv4_Addr":       "2.3.4.5",
            "User_Client_IPv4_Addr":       "6.7.8.9",
            "Weight":          1,
            "Client_Tunnel_Info": {
            },
            "Server_Tunnel_Info": {
            }
        },
        {
            "Eth_Index":   0,
            "User_Id":     6,
            "Replay_File_Name":     "pcap_file.pcap",
            "User_Server_IPv4_Addr":       "8.3.4.5",
            "User_Client_IPv4_Addr":       "9.7.8.9",
            "Weight":          1,
            "Client_Tunnel_Info": {
            },
            "Server_Tunnel_Info": {
            }
        },
        {
            "Eth_Index":   1,
            "User_Id":     7,
            "Type":        "Client",
            "User_Server_IPv4_Addr":       "5.3.4.5",
```

```
                "User_Client_IPv4_Addr":      "9.7.8.9",
                "Weight":         2,
                "Client_Tunnel_Info": {
                },
                "Server_Tunnel_Info": {
                }
            }
        ]
    }
```

Dynamic User Mode

Dynamic User Mode uses a special TCP port/REST API (configured when creating the general test configuration on the NetLoad system) to allow an external system to add/delete "users" on-the-fly into a running test. The external system can be "single-source", where all information about the "users" comes from a single source, or "dual-source" where the information about a "user" comes from two sources, one on the client-side and one on the server-side. Each "user", once added, will start executing TCP/HTTP client-server transactions as specified in the specific "user" parameters passed with the "add user" command and the general parameters pre-configured in the running test. Similarly, "stop user" command will allow the "user" to complete the last TCP/HTTP transaction in progress, while "delete user" will remove the "user" from the test immediately without completing all transactions gracefully. Once all "users" are removed from the test, traffic will stop, but the test does not stop and traffic will resume as more "users" are added any time while the test is running. Due to dynamic nature of this type of testing, the test will continue until "STOP" is issued either via REST API, GUI, or CLI command.

User File Mode uses a pre-configured and pre-loaded file to add/delete "users" into a test. This mode works in conjunction with Dynamic User Mode to pre-load "users" into a test if needed before Dynamic User Mode is executed.

Dual-Source "User" addition/deletion:

The following is an example of “dual-source” user addition where client-side and server-side information for GTP tunnels comes from separate sources.

Server-side info:


```
{
  "user_info": [
    {
      "Eth_Index": 0,
      "User_Id": 1,
      "Server_Tunnel_Info": {
        "Dst_TEID": 1000,
        "Src_TEID": 100
      }
    },
    {
      "Eth_Index": 1,
      "User_Id": 1,
      "Server_Tunnel_Info": {
        "Dst_TEID": 1001,
        "Src_TEID": 101
      }
    }
  ]
}
```


Client-side info:


```
{
  "user_info": [
    {
      "Eth_Index": 0,
      "User_Id": 1,
      "User_Server_IPv4_Addr": "105.0.0.2",
      "User_Client_IPv4_Addr": "172.0.0.2",
      "Client_Tunnel_Info": {
        "Dst_IPv4_Addr": "5.0.0.2",
        "Src_IPv4_Addr": "72.0.0.2",
        "Dst_TEID": 100,
        "Src_TEID": 1000
      }
    },
```

```
   {
    "Eth_Index": 1,
    "User_Id": 1,
    "User_Server_IPv4_Addr": "105.0.0.2",
    "User_Client_IPv4_Addr": "172.0.0.2",
    "Client_Tunnel_Info": {
     "Dst_IPv4_Addr": "5.0.0.2",
     "Src_IPv4_Addr": "72.0.0.2",
     "Dst_TEID": 101,
     "Src_TEID": 1001
    }
   }
  ]
}
```

The following is an example of a “stop” command:

```
{
       "user_info":
        [
           {
                "Eth_Index":    0,
                "Stop" : true,
                "User_Id":     8,
                "Client_Tunnel_Info": {
                   "Dst_TEID":       457,
                   "Src_TEID":       7789
                },
                "Server_Tunnel_Info": {
                    "Dst_TEID":      7789,
                    "Src_TEID":      457
                }
           },
           {
                "Eth_Index":    1,
                "Stop" : true,
                "User_Id":     10,
                "Client_Tunnel_Info": {
                   "Dst_TEID":       556,
                   "Src_TEID":       9789
                },
```

```
            "Server_Tunnel_Info": {
                "Dst_TEID":      9789,
                "Src_TEID":      556
            }
        }
    ]
}
```

The following is an example of a “delete” command:

```
{
    "user_info":
     [
        {
            "Eth_Index":    0,
            "Delete" : true,
            "User_Id":     8,
            "Client_Tunnel_Info": {
                "Dst_TEID":       457,
                "Src_TEID":       7789
            },
            "Server_Tunnel_Info": {
                "Dst_TEID":      7789,
                "Src_TEID":      457
            }
        },
        {
            "Eth_Index":    1,
            "Delete" : true,
            "User_Id":     10,
            "Client_Tunnel_Info": {
                "Dst_TEID":       556,
                "Src_TEID":       9789
            },
            "Server_Tunnel_Info": {
                "Dst_TEID":      9789,
                "Src_TEID":      556
            }
        }
    ]
}
```