Network Performance Testing Appliance Project Report

Author: Robbie Mckennie
Date: 9/11/2025
Course: ITSP7
Supervisor: Emre Erturk

Abstract

This report presents the design and evaluation of a network performance testing appliance developed using Cisco TRex, IPerf3 and a WatchGuard Firebox M4600 firewall appliance. The appliance was designed to generate and measure network traffic in a controlled test environment. Performance testing was conducted on several consumer-grade routers; the MikroTik RB951G-2HnD, Netcomm NF18AC, and Mercku M6A, using both TRex and iPerf. This report outlines the design decisions, system architecture, testing methodology, and results, followed by an analysis of findings and recommendations for future development.

---

1. Introduction

1.1 Project Overview

The goal of this project was to design and implement a dedicated network testing appliance capable of evaluating wired networking performance, both on layer 3 and layer 2 (routed, and switched or bridged respectively). The appliance uses TRex and IPerf3 to generate traffic and measure throughput and packet loss.

1.2 Objectives

- To build a traffic generation and measurement platform.
- To test and compare the network performance of several consumer networking devices.
- To validate manufacturer claims of performance and give data to inform the selection of hardware and the configuration thereof.

1.3 Scope

The project focuses on wired Ethernet performance testing at Layer 2 and Layer 3, using controlled test environments. Wireless testing and large-scale multi-user scenarios are beyond the current scope but may be considered for future work.

---

2. Design Decisions

2.1 Hardware Selection

The **Firebox M4600** was selected as the hardware platform due to its multiple gigabit interfaces and enterprise-grade performance. The device provides sufficient CPU and memory resources to handle basic traffic generation workloads, but is also upgradable using standard server parts.

2.2 Software Stack

- **Traffic Generation:**
  - TRex (stateless mode, Python API)
  - IPerf3 (Python API)
- **Operating Environment:** CentOS Stream 9 Linux
- **Supporting Tools:** Python for test control

2.3 Device Under Test (DUT) Selection

The tested devices represent different market segments: - **MikroTik RB951G-2HnD:** Feature rich SOHO router
- **Netcomm NF18:** Older model mid-range home router with integrated DSL modem
- **Mercku M6a:** Mesh router targeting higher consumer throughput

---

4. Test Methodology

4.1 Performance Measurement

- **NDR:** The goal of this testing is to find the fastest throughput rate while maintaining packet loss under a certain margin, the non-drop rate (NDR)
- **Binary search:** To find the optimal NDR I applied a
- **Varied packet size:** To show how performance can vary by packet size, I've run all tests using packet sizes of 64B, 512B, and 1500B

4.2 Choice of Tool

I was leaning towards using TRex for my testing because of the versatility and performance of the tool, however I found that TRex is not well suited to testing basic SOHO configurations. TRex is designed to simulate a large number of devices behind a router, rather than a number of devices attached to a switch. For these tests I decided to use a simpler tool, IPerf3, in UDP mode.

4.3 TRex for Testing Multi-Port Performance

- Stateless streams configured using the TRex STL Python API
- Speed targets are set as a percentage of the maximum line rate
- Each run at a particular speed is 30 seconds long
- A binary search is used to find a bitrate that produces a drop rate of less than 0.05%

4.4 IPerf3 for Testing Performance Through PPPoE

- IPerf3 tests are managed using a Python library
- Speed targets are set as a target Layer 4 throughput
- Each run at a particular speed is 30 seconds long
- A binary search is used to find a bitrate that produces a drop rate of less than 0.05%

---

4. Network Topologies

4.1 TRex Multi-Port Tests

For mutli-port testing on the RB951G, I connected 4 ports of the router to 4 ports on the Firebox. I wrote 2 different configuration files, corresponding to the 2 modes of operation I was testing, routing and switching/bridging.

In the bridging configuration, the ports on the router don't need IP addresses, and the ports on the firebox are configured in pairs as follows

- Port 5: `IPv4 addr: 1.1.1.1, default GW: 2.2.2.2`
- Port 6: `IPv4 addr: 2.2.2.2, default GW: 1.1.1.1`
- Port 7: `IPv4 addr: 3.3.3.3, default GW: 4.4.4.4`
- Port 8: `IPv4 addr: 4.4.4.4, default GW: 3.3.3.3`

In the routing configuration, each port required their own IP addresses, and the default gateway on the firebox is set to match, as follows.

- Router Port 2: `IPv4 addr: 2.2.2.2`
- Router Port 3: `IPv4 addr: 3.3.3.3`
- Router Port 4: `IPv4 addr: 6.6.6.6`
- Router Port 5: `IPv4 addr: 7.7.7.7`
- Firebox Port 5: `IPv4 addr: 1.1.1.1, default GW: 2.2.2.2`
- Firebox Port 6: `IPv4 addr: 4.4.4.4, default GW: 3.3.3.3`
- Firebox Port 7: `IPv4 addr: 5.5.5.5, default GW: 6.6.6.6`
- Firebox Port 8: `IPv4 addr: 8.8.8.8, default GW: 7.7.7.7`

4.2 IPerf3 PPPoE Tests

For the PPPoE tests on all 3 of my test routers, I configured a MikroTik RB4011iGS as a PPPoE server, set up each router with a default home router config, and arranged them as follows.

```
[Firebox] <-> [DUT] <-> [RB4011iGS] <-> [Firebox]
```

- Firebox Port 5: `IPv4 addr: 192.168.20.10/24, default GW: 192.168.20.1`
- DUT Port 5: `IPv4 addr: 192.168.20.1/24`
- DUT PPPoE Client: `IPv4 addr: 20.20.20.20`
- RB4011 PPPoE Server: `IPv4 addr: 10.10.10.10`
- RB4011 Port 5: `IPv4 addr: 3.3.3.3`
- Firebox Port 6: `IPv4 addr: 4.4.4.4, default GW: 3.3.3.3`

---

5. Test Results

5.1 TRex Results (MikroTik RB951G)

Mode	64B Kpps	64B Mbps	512B Kpps	512B Mbps	1500B Kpps	1500B Mbps
Bridged (Fast path)	2838	1,453	466	1908	164	1,966
Bridged (25 filter rules)	94	48	85	349	66	786
Routed (Fast path)	244	125	238	973	61	736
Routed (25 filter rules)	2.8	1.4	4.6	18.7	4.7	55.8

5.2 iPerf Results

Device	64B Kpps	64B Mbps	512B Kpps	512B Mbps	1500B Kpps	1500B Mbps
RB951G	155	27	157	588	82	957
NF18AC	155	27	156	590	82	956
M6A	150	26	154	578	82	957

5.3 Observations

- Across the board the devices show a degradation of performance for smaller packets, due to needing to process a greater number of packets per second
- The TRex routing results are much lower than I expected. I believe this is due to the traffic being unidirectional, so the connection is never established to be properly accelerated by hardware.
- Between the 3 devices I tested I found very similar performance.

---

6. Discussion

6.1 Tool Comparison

Cisco TRex provided more granular control and consistent results at high packet rates, while IPerf3 was simpler to use. IPerf3 also inherently uses acknowledgements as part of the protocol, which I believe works better with hardware acceleration for routing.

6.2 Challenges

- It was challenging getting TRex working with DPDK
- Difficulty getting my head around bandwidth measured at different layers
- Configuration inconsistencies between DUTs

---

7. Conclusion and Future Work

This project successfully demonstrated a functional network performance testing appliance using Cisco TRex. The system provided reliable and repeatable performance measurements across multiple routers.

Future improvements could include: - Integration with Prometheus and Grafana for real-time visualization.
- Automated testing pipelines for multiple DUTs.
- Wireless and IPv6 performance testing.

For more information see https://github.com/h0dgep0dge/ITSP7

References

Cisco Systems. (n.d.). *TRex traffic generator documentation*. Cisco. https://trex-tgn.cisco.com/trex/doc/index.html

Esnet. (n.d.). *iPerf3 and iPerf2 user documentation*. Energy Sciences Network. https://iperf.fr/iperf-doc.php

MikroTik. (n.d.). *RB951G-2HnD product specifications*. MikroTik. https://mikrotik.com/product/RB951G-2HnD

NetComm Wireless. (n.d.). *NF18ACV Datasheet*. NetComm Wireless. https://support.netcommwireless.com/api/Media/Document/ec1c2eb4-5243-4bcf-adbe-94b0749b1c25?Product=NF18ACV-Datasheet.pdf

Mercku. (n.d.). *M6a Data Sheet*. Mercku. https://www.mercku.com/wp-content/uploads/2022/09/M6a-Data-Sheet.pdf

Bradner, S., & McQuaid, J. (1999). *Benchmarking methodology for network interconnect devices*. Internet Engineering Task Force. https://doi.org/10.17487/RFC2544