**Network Performance Evaluation on   
Linux Based Operating Systems**

**Date: 26/03/2025**

**Version: 1.1**

**Client:**

Raymond Lutui

**Supervisor:**

Daniel Vaipulu

**Team Members:**

Kylie Afable

Zafar Azad

Larissa Goh

Nathan Quai Hoi

Charmi Patel

Win Phyo

Thomas Robinson (Team Lead)

# Executive Summary

This proposal will include

* Scope
  + Evaluating IPv4 and IPv6 network performance on three Linux-based software routers using iPerf/D-ITG.
  + 4 Computers
    - 1 Sender
    - 2 Configured as routers
      * Each with 2 network cards
    - 1 Receiver
  + Static IPs Configured
  + Operating Systems
    - Ubuntu
    - FedoraOS
    - Kali Linux
  + Using D-ITG
* Time
  + The estimated time it will take to complete this project would be 300 to 360 hours.
* Method
  + Performance testing using TCP/UDP over 12 packet sizes, collecting throughput, delay, jitter, and packet loss data.
  + Packet Sizes - 128, 256, 384, 512, 640, 768, 896, 1024, 1152, 1280, 1408, 1536 Bytes.
* Risks
  + Lack of experience with Linux networking, hardware limitations, configuration errors.
* Costs
  + Mentor
    - Based on a conservatively estimated time of one hour per week with the mentor, $3,919.20 is the estimated cost.
  + Equipment
    - 4 Computers
      * 1 Sender
      * 2 Configured as routers
        + Each with 2 network cards
      * 1 Receiver

### Terms of Reference

* Project purpose
  + The core purpose of this project is to evaluate and compare the network performance of IPv4 and IPv6 on Linux-based software routers. This evaluation will be conducted by measuring key metrics such as throughput, delay, jitter, and packet loss across various packet sizes.
* Context of the project
  + IPv4's scalability issues drive the move to IPv6, which has different performance due to higher data overhead. This project aims to measure and provide factual performance results for both protocols within a controlled software router setting.
* Client details
* Problem
  + The core issue is the absence of evidence to accurately assess the performance disparities between IPv4 and IPv6 within software router environments. Despite IPv4's acknowledged scalability constraints and IPv6's intended role as a replacement, a lack of real-world performance data hinders informed decision-making.
* Opportunities
  + This project presents an opportunity to gain valuable insights into IPv6 adoption barriers by providing concrete performance data.
  + It also creates an opportunity to develop expertise in Linux networking, performance testing tools, and data analysis.
  + The project also provides the opportunity to gain experience in creating and running a project, and the creation of a portfolio.

### Rationale

This will include:

* Why is the project needed
  + IPv4 is widely used but has scalability issues; IPv6 is the replacement but comes with larger data overhead.
* Existing systems
  + IPv4 is widely used but has scalability issues; IPv6 is the replacement but comes with larger data overhead.
* Key issues / opportunities
  + IPv6 adoption barriers, need for performance testing.

### Objectives & Scope

Deliverables

* Configured Linux software routers (Fedora, Ubuntu, Kali).
* Performance test results (throughput, delay, jitter, packet loss).
* Statistical analysis (confidence interval re-runs).
* Comparison of IPv4 vs. IPv6 performance.

High level functional and non-functional requirements

* Functional: Must route traffic, measure performance under TCP/UDP.
* Non-Functional: Must handle various packet sizes accurately.

Infostructure needs

* Three Linux-based OS installations.
* Testbed setup for network performance testing.
* Software tools (iPerf, D-ITG).

### Skills Analysis

Required skills to complete the project:

* Networking knowledge:
  + IPv4, IPv6
  + TCP, UDP
  + Packets and packet sizes
  + Performance metrics: throughput, delay, jitter, packet loss
* Understanding of basic Linux operations:
  + Installing and configuring a Linux-based OS
  + Configuring Linux as a router
  + BASH scripting (for automation)
* Performance testing tools (D-ITG, iPerf)
* Data analysis to interpret performance metrics

Existing skills:

* All team members have a basic understanding of how to use Linux.
  + Only some team members are familiar with installing and configuring Linux.
* All team members understand the basics of networking, including IPv4, IPv6, TCP and UDP protocols, packets, and basic performance metrics.
  + Some team members might need to revise the various performance metrics that will be measured during this project.
* All team members have done basic C programming, which will make understanding BASH scripting much easier.

Upskilling needed:

* All of the team will need to learn how to use D-ITG (and/or any other testing tools)
  + This will be done between the acceptance of the proposal and the start of the project’s second phase (Milestone 1).
* Most team members will need to learn basic BASH scripting for automation of testing.
  + Since the scripts theoretically only need to be written once, this can be done by a single team member, so others might not require this skill.
* All of the team will need to learn how to configure Linux as a router
  + Once the proposal is accepted and equipment confirmed, the team will schedule time with the equipment to practice this configuration.

### Team Roles

This will include:

* Responsibilities of each team member
  + Team Leader
    - Thomas Robinson (IPv4 Team)
  + Team Members
    - Kylie Afable (IPv6 Team)
    - Zafar Azad (IPv4 Team)
    - Larissa Goh (IPv6 Team)
    - Nathan Quai Hoi (IPv6 Team)
    - Charmi Patel (IPv6 Team)
    - Win Phyo (IPv4 Team)

### Team Schedule

* Weekly Commitment: 12–15 hours per team member
* Weekly Meeting with Mentor
  + every Thursday from Week 1 – 5, Time: 10 AM/ 1 PM | Location: WZ1101
  + on Thursday every fortnight from Week 6 onwards. Time: 1 PM | Location: WZ1101
* Weekly Team-Only Meeting prior to Mentor meeting on Thursday.
* Additional Team-Only Meeting when necessary/ urgent.

### Project Management Methodology

* Management methodology
* Justification for the project
* Work Breakdown Structure (WBS)
* Key Tasks
* Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Methodologies** | **Core approach** | **Pros** | **Cons** |
| Waterfall | A fixed set of phases, where each phase must be completed before moving to the next phase | Waterfall outlines a clear project structure; this demonstrates clear cost and goals. Due to the fixed nature of the methodology, tracking is linier and easier. The upfront planning approach also minimises risk factors as most of them are accounted for in the initial planning phase. Waterfall approach also expects fewer delays that can occur from additional requirements. | Due to the fixed nature of the methodology its benefit can also be a drawback depending on the nature of the project. Project s that requires continual interactions with end users and the team to review current direction and course correction do not fit in the waterfall approach. Its fixed approach is ultimately limiting flexibility. |
| Agile | Working quickly and collaboratively while promoting an environment that allows changes during the development cycle | The agile framework is the suggested approach for software development projects that prioritises customer needs constantly changing throughout the development lifecycle.  The phases of this approach are broken down into sprints.  Once a sprint is completed review and feedback commence to improve the efficiency of the next sprint. | Due to the agile approaches being constantly changing and reviewed after each sprint, limitation on resource planning can occur due to the unclear end object. In most cases, the dev team can only have sight on a few sprints ahead.  Agile is also difficult to measure due to its nature of change which can also result in scope creep. |
| Lean | A methodology that focuses on maximising efficiency by reducing waste and continually looking for workflow improvements |  |  |

Deliverables

* Results of the performance of IPv6 / IPv4 at the 12 packet sizes. Measuring the throughput, delay, jitter, and packet loss. (Excel File + Log)
* Comparing them

### Risk and Issues Management

Risk Register

* Incorrect router configuration affecting test results.
* Lack of Linux networking expertise.
* Hardware/software compatibility issues.

Issue Log

* D-ITG not available for Fedora.

Mitigation Strategies

* Research best practices for Linux router setup.
* Conduct preliminary tests before the main experiment.

### Project Plan

Gantt chart / timeline

Milestones

* Project Proposal
* First Iteration Testing / Data
* Mid-Term Review

Key assessments

* Project Proposal
* Mid-Term Review
* Poster
* Overall Portfolio

### Estimated Costs

Resource estimates

* Linux operating systems are free
* D-ITG and iPerf are free
* 4x computers
* 6x network cards

Labour Breakdown

* Team’s time (not monetized).
  + 15h per week
  + 12 weeks in a semester
  + 15h x 12 weeks = 144h x 2semesters = 360h (estimate)
  + Average pay for a network engineer in NZ per hour = $30.99 NZD
    - (Network Engineer Salary in New Zealand, n.d.)
  + $30.99 x 360h = $11,156.40 (estimate)

Mentor Costs Breakdown (which is put at $142 + GST per hour)

* $142+GST per hour.
* GST = 15%
* $142 x 15% (1.15) = $163.30
* At about an hour per week
  + 12 weeks x 1h = 12h x 2 semesters = 24h
* 24h x $163.30 = $3,919.20 (estimate)

# Appendix A - Disclaimer

**Auckland University of Technology**

**Bachelor of Computer & Information Sciences**

**Research & Development Project**

**Disclaimer: Network Performance Evaluation on Linux Based Operating Systems**

**Clients should note the general basis upon which the Auckland University of Technology undertakes its student projects on behalf of external sponsors:**

While all due care and diligence will be expected to be taken by the students, (acting in software development, research or other IT professional capacities), and the Auckland University of Technology, and student efforts will be supervised by experienced AUT lecturers, it must be recognised that these projects are undertaken in the course of student instruction. There is therefore no guarantee that students will succeed in their efforts.

This inherently means that the client assumes a degree of risk. This is part of an arrangement, which is intended to be of mutual benefit. On completion of the project, it is hoped that the client will receive a professionally documented and soundly constructed working software application, some part thereof, or other appropriate set of IT artefacts, while the students are exposed to live external environments and problems, in a realistic project and customer context.

In consequence of the above, the students, acting in their assigned professional capacities and the Auckland University of Technology, disclaim responsibility and offer no warranty in respect of the “technology solution” or services delivered, (e.g. a “software application” and its associated documentation), both in relation to their use and results from their use.

# References

Network engineer salary in New Zealand. (n.d.). Indeed. <https://nz.indeed.com/career/network-engineer/salaries>

Atlassian. (n.d.). Waterfall methodology: A comprehensive guide. Atlassian. <https://www.atlassian.com/agile/project-management/waterfall-methodology>

Laoyan, S. (2025, February 20). What is Agile methodology? (A beginner’s guide). Asana. <https://asana.com/resources/agile-methodology>