

**Project Proposal**

Network Performance Evaluation of Linux Based Operating Systems in a Physical Environment

Date:10/04/2025

Version: 2.2

Client: Dr. Raymond Lutui

Prepared By:

Supervisor: Daniel Vaipulu

Project Manager: Thomas Robinson

System Architect: Win Phyo

System Architect: Nathan Quai Hoi

Network Engineer: Zafar Azad

Network Engineer: Larissa Goh

Network Engineer: Charmi Patel

Network Engineer: Kylie Afable

# Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Note** |
| 13/03/2025 | 0.1 | Nathan | Initial document creation. |
| 15/03/2025 | 1.0 | Nathan | Initial versioning and header. |
| 17/03/2025 | 1.01 | Nathan, Zafar | Formatting, basic information for early sections, and some placeholders. |
| 18/03/2025 | 1.02 | Thomas | Update date and name. |
| 20/03/2025 | 1.03 | Zafar, Win | Formatting, start of methodology comparison, new team member, disclaimer, and start of references. |
| 24/03/2025 | 1.04 | Zafar | Addition of methodology information and references. |
| 25/03/2025 | 1.05 | Nathan | Formatting and addition of cost information. |
| 26/03/2025 | 1.1 | Thomas, Nathan, Win | Reformatting of document sections, terms of reference, and upskilling information. |
| 27/03/2025 | 1.2 | Nathan, Win, Zafar | Reformatting sections into paragraphs, documents added to appendix, additions and changes to methodology, and summarising some sections with reference to appendices. |
| 28/03/2025 | 1.3 | Nathan, Thomas | Formatting and placeholders for missing information. Version bumped to 1.2 with proper changelog (should be 1.3 – fixed next day). |
| 30/03/2025 | 1.4 | All Team Members | Completion of draft proposal with revisions of all major sections. |
| 31/03/2025 | 1.5 | All Team Members | Improvement based off feedback, reformatting document and layout. |
| 02/04/2025 | 1.6 | All Team Members | Continued improvement based off feedback, and further reformatting. |
| 04/04/2025 | 1.7 | Zafar | Methodology change based off supervisor’s comments. |
| 05/04/2025 | 1.8 | Larissa, Thomas | Added Quality Assurance Plan based on feedback, new network diagram, some sections moved to more appropriate places. |
| 06/04/2025 | 1.9 | Thomas | Added Change Mgmt Plan and QA Plan. Formatting for appendices. Changed all remaining instances of “Mentor” to “Supervisor”. |
| 06/04/2025 | 2.0 | All Team Members | Added RTM, Client Contract, and Project Charter. Adjusted Milestone Report, Risk Management Plan, Quality Assurance Plan, Scope Statement, and WBS. Removed all references to “sprints”, refined RTM. |
| 09/04/2025 | 2.1 | Thomas, Win | Fixed errors: duplicate data, iPerf over D-ITG, “test” over “evaluation”. Minor reformatting. Added tools comparison and signed client contract. |
| 10/04/2025 | 2.2 | Thomas | Minor grammatical issues and phrasing in QA. |

Table of Contents

[Version Control 2](#_Toc195128844)

[Executive Summary 5](#_Toc195128845)

[Terms of Reference 5](#_Toc195128846)

[Project Objectives 5](#_Toc195128847)

[Rationale 6](#_Toc195128848)

[Project Scope 6](#_Toc195128849)

[Out of Scope 6](#_Toc195128850)

[Key Stakeholders 6](#_Toc195128851)

[Project Feasibility 7](#_Toc195128852)

[Project Plan 8](#_Toc195128853)

[Estimated Costs 10](#_Toc195128854)

[Labour Breakdown 10](#_Toc195128855)

[Skills Analysis 10](#_Toc195128856)

[Upskilling Plan Schedule 10](#_Toc195128857)

[Requirement Traceability Matrix 10](#_Toc195128858)

[Deliverables 11](#_Toc195128859)

[Success Criteria 11](#_Toc195128860)

[Project Management Methodology 11](#_Toc195128861)

[Project Phases 12](#_Toc195128862)

[Team Contract 12](#_Toc195128863)

[Team Schedule 13](#_Toc195128864)

[Risk Register 13](#_Toc195128865)

[Issue Log 14](#_Toc195128866)

[Change Management Plan 14](#_Toc195128867)

[Quality Assurance Plan 14](#_Toc195128868)

[Appendices 15](#_Toc195128869)

[Appendix A – Disclaimer 15](#_Toc195128870)

[Appendix B – Scope Statement 16](#_Toc195128871)

[Appendix C – Communication Stakeholder Register 19](#_Toc195128872)

[Appendix D – Stakeholder Management Plan 20](#_Toc195128873)

[Appendix E – Tools Comparison 22](#_Toc195128874)

[Appendix F – Milestone Report 23](#_Toc195128875)

[Appendix G – Work Breakdown Structure 25](#_Toc195128876)

[Appendix H – Gantt Chart 28](#_Toc195128877)

[Appendix I – Labour Breakdown Table 32](#_Toc195128878)

[Appendix J – Skills Analysis Matrix 33](#_Toc195128879)

[Appendix K – Upskilling Plan Schedule 34](#_Toc195128880)

[Appendix L – Requirement Traceability Matrix 37](#_Toc195128881)

[Appendix M – Project Charter 43](#_Toc195128882)

[Appendix N – Team Contract 45](#_Toc195128883)

[Appendix O – Client Contract 47](#_Toc195128884)

[Appendix P – Risk Register 49](#_Toc195128885)

[Appendix Q – Risk Management Plan 52](#_Toc195128886)

[Appendix R – Issue Log 53](#_Toc195128887)

[Appendix S – Change Management Plan 55](#_Toc195128888)

[Appendix T – Quality Assurance Plan 59](#_Toc195128889)

[References 62](#_Toc195128890)

# Executive Summary

This project evaluates the IPv4 and IPv6 network performance of three Linux-based software routers using D-ITG. Performance evaluation, encompassing throughput, delay, jitter, and packet loss, will be conducted across 12 packet sizes on a network comprised of four computers.

The project, estimated at 300-360 hours, acknowledges potential risks such as Linux networking experience and hardware limitations, and includes a cost analysis for supervisor support and necessary equipment.   
Some of the risks and constraints to consider include lack of team experience with Linux networking, possible hardware failure, and potential budget restraints.   
The total project cost is budgeted at NZD$118,774.71, which covers the hardware, labour, and tools needed for the project.

# Terms of Reference

Our client, Dr. Raymond Lutui, has requested an evaluation of the network performance of three Linux-based operating systems (Fedora, Ubuntu, and Kali Linux) configured as software routers, to determine which has the best performance and to compare the data with his previous evaluations. Despite IPv4's acknowledged scalability constraints and IPv6's intended role as a replacement (Suryaningrat et al., 2016), a lack of real-world performance data hinders network professionals from confidently strategising and executing the migration to IPv6 in their software-defined networks (Narayan et al., 2016).   
This project addresses the lack of real-world data with thorough network evaluation: measuring network performance across multiple different packet sizes using D-ITG, over both IPv4 and IPv6, in a four-PC network topology. The project commences on April 4, 2025, with final deliverables due on October 31, 2025.

# Project Objectives

The goal of this project is to evaluate the performance of 3 latest Linux based Operating Systems distributions – Fedora, Ubuntu, Kali Linux. These operating systems need to be configured as software routers before running the evaluations. Tools such as D-ITG or iPerf are to be employed to generate the evaluation traffics.

Objectives to be achieved include:

* Setup a network with 4 computers, 2 clients and 2 servers (servers configured as routers)
* Use TCP and UDP as transmission protocols.
  + Evaluate on IPv4
  + Evaluate on IPv6
* Performance evaluation:
  + Each evaluation should run a minimum of ten times.
    - Evaluate for throughputs, any delays, jitter, and any packet loss.
  + Any evaluation that falls outside the 95% confidence interval needs to be re-run
* Each evaluation will range from a minimum of 128 through to 1536 Bytes to assess performance under different conditions.
* Set up physical machines with Fedora, Ubuntu, and Kali Linux distributions configured as software routers.
* Conduct performance evaluations to evaluate network throughput, delay, jitters, and packet loss for TCP and UDP transmissions on IPv4 and IPv6 protocols.
* Follow the Software Testing Life Cycle (STLC) methodology for systematically evaluating phases and activities including requirement analysis, evaluation planning, evaluation case development, evaluation environment setup, evaluation execution and evaluation closure.

# Rationale

The transition to IPv6, while crucial for internet expansion and advanced features, introduces increased data overhead. This can create performance bottlenecks, particularly in resource-constrained software routers.   
To address this, we will conduct a controlled experiment measuring the performance differences between IPv4 and IPv6. Using TCP and UDP protocols across 12 packet sizes on four Linux systems configured as software routers.   
We will evaluate on 3 different Linux operating systems which include, Ubuntu, Fedora, and Kali. We will analyse throughput, delay, jitter, and packet loss. The resulting empirical data will provide valuable insights for network performance comparing IPv4 to IPv6.

# Project Scope

This project encompasses the comprehensive performance evaluation of three Linux-based operating systems (Fedora, Ubuntu, and Kali Linux) configured as software routers. The scope includes configuring and evaluating all three operating systems across a four-PC network topology (one sender, two routers, one receiver), with evaluation of both TCP and UDP protocols over IPv4 and IPv6. Performance assessments will measure throughput, delays, jitter, and packet loss across 12 different packet sizes ranging from 128 to 1536 bytes.

Each configuration will undergo 10 evaluation runs to ensure statistical validity, with additional runs conducted for any results falling outside the 95% confidence interval. The evaluation runs involve the sender to initiate the packet sending in specified network topology to receiver. Each evaluation runs use DITG to generate traffic for 10 seconds with different packet rates and logs will be decoded to export the average performance metrics.

The project includes detailed analysis and documentation of all findings, complete data logging, and delivery of comparative performance statistics across all evaluated operating systems. For more complete details on the scope statement, please refer to Appendix B.

## Out of Scope

This project focuses specifically on performance evaluation and does not include hardware procurement, initial operating system installation, nor physical environment setup.

The team will not be responsible for implementing performance improvements based on findings, providing ongoing maintenance, nor supporting the evaluation environment beyond the project duration.

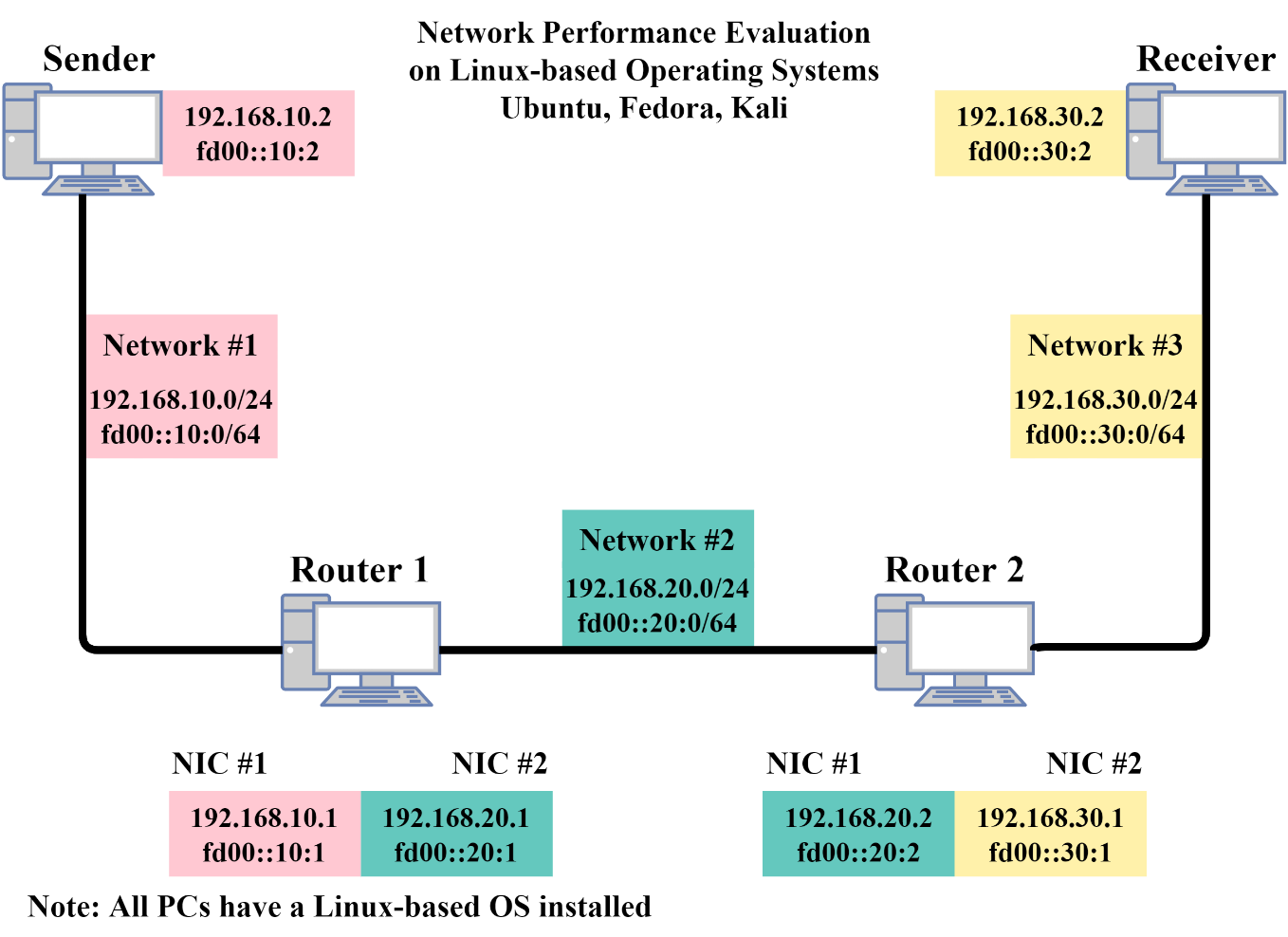
Evaluation is limited to the specified operating systems, network topology, and protocols as outlined in the scope statement.

## Key Stakeholders

The project client is Raymond Lutui, with Daniel Vaipulu acting as supervisor.   
The project team consists of seven members: Kylie Afable, Zafar Azad, Larissa Goh, Nathan Quai Hoi, Charmi Patel, Win Phyo, and team lead Thomas Robinson.   
For a complete and detailed list of stakeholders, please refer to Appendix C. For the Stakeholder Management Plan, please refer to Appendix D.

## Project Feasibility

**Technical Infrastructure:**



The project would involve 4 computer hardware provided by the client, in which two will be configured as routers, while the other two will be used as sender and receiver.

**Tools and Technology:**

Software:

Three Linux-based operating systems will be used, configured as routers. These operating systems are commonly used for networking tasks. We will be using Ubuntu, Fedora, and Kali Linux.

Network Performance Evaluation tools:

The tool ‘D-ITG’ will be used to measure network performance. Please refer to Appendix E for the tools comparison, which explains why D-ITG was chosen. IPv4 and IPv6 will be evaluated using TCP and UDP transmission protocols. 12 variations of packet sizes will be used: 128, 256, 384, 512, 640, 768, 896, 1024, 1152, 1280, 1408, and 1536 Bytes.

**Skills That Are Required:**

* Familiarise themselves with Linux, software routing, and performance evaluation.
* Understanding IPv4 and IPv6 network principles.
* Adequate knowledge in programming languages.
* Problem solving with debugging and troubleshooting.
* Understanding how to configure routers and comprehends the results from the network performance tools.

**Roles:**

Project Manager:

Looks after the project and ensures everyone and the project are all running smoothly.

System Architect:

Creates the general framework and ensures that it aligns with the projects main goal.

Network Engineer

Builds, configures and maintains the network.

**Rationale/Justification:**

Our project aims to evaluate the performance of IPv4 and IPv6 protocols in different Linux-based operating systems configured as routers. With these tools, technologies, and required infrastructure, it helps us understand how both protocols perform in different networking environments. This is important as IPv6 is on the rise.

Please refer to Rationale for further details.

## Project Plan

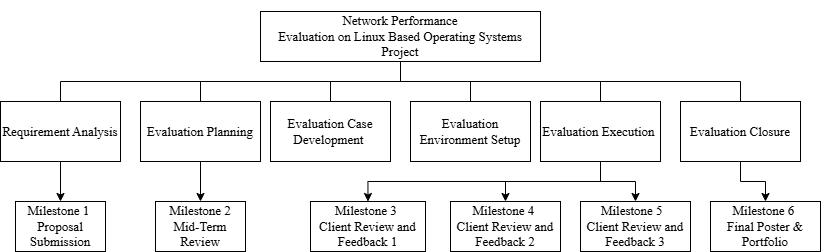
**Milestones**

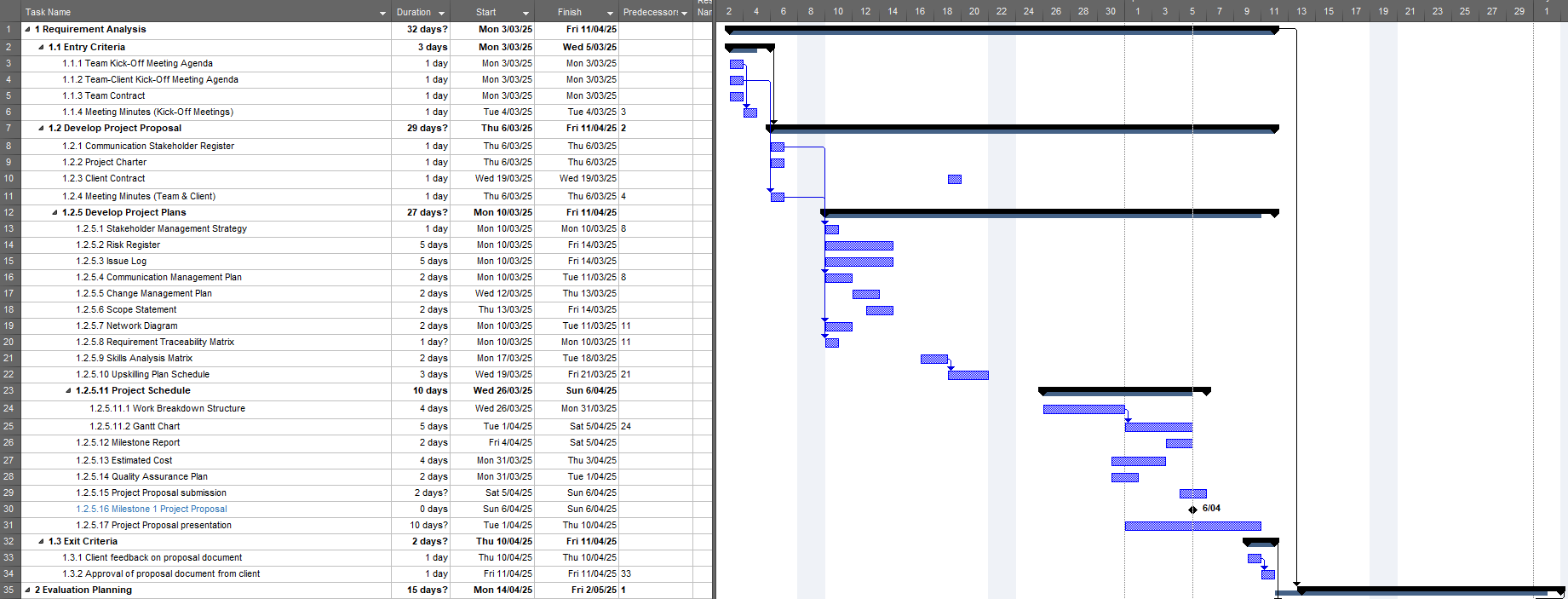
This section includes the main milestones of the project, as well as internal milestones to help track the progress of the project and meet deadlines. The following table includes the important external milestones. For the complete Milestone Report, please refer to Appendix F.

|  |  |  |
| --- | --- | --- |
| **Date** | **Milestone Type** | **Milestone** |
| 06/04/2025 | External | Project Proposal Submission |
| 06/06/2025 | External | Mid-Term Review |
| 30/07/2025 | Internal | Completion of Ubuntu Evaluation |
| 31/07/2025 | External | Client Review and Feedback 1 |
| 04/09/2025 | Internal | Completion of Fedora Evaluation |
| 05/09/2025 | External | Client Review and Feedback 2 |
| 09/10/2025 | Internal | Completion of Kali Evaluation |
| 10/10/2025 | External | Client Review and Feedback 3 |
| 31/10/2025 | External | Final Poster and Portfolio |

**WBS, Tasking, Scheduling**

This section displays a high-level view of the Work Breakdown Structure and Gantt Chart for Requirement Analysis along with the milestones.





For the complete and detailed version of the WBS and Gantt Chart, please refer to the Appendix G and Appendix H, respectively.

## Estimated Costs

The table below shows the estimated cost of the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cost Category** | **Description** | **Quantity** | **Unit Cost (NZD)** | **Total Cost (NZD)** |
| Hardware | HP Z420 Tower Workstation (CORE Technology Brokers, n.d.). | 4 | $950 | $3,800.00 |
| Hardware | Philips 243V7QJAB/79 Monitor | 4 | $161 | $ 644.00 |
| Hardware | ASUS Vivobook Go 15 L510KA-EJ599W Laptop | 7 | $667 | $4,669.00 |
| Hardware | Rapoo X130PRO wired keyboard and mouse combo | 4 | $23.14 | $ 92.56 |
| Network Equipment | Cruxtec 0.5m Cat7 Ethernet Cable RS7-005-BK | 3 | $5.45 | $ 16.35 |
| Network Equipment | TP-Link TG-3468 32-bit Gigabit PCIe Network Adapter | 6 | $22 | $ 132.00 |
| Labour | Refer to labour breakdown table | - | - | $109,420.80 |
| Software Tools | D-ITG | 4 | $0 | $ 0.00 |
| Software Tools | Linux OS (Ubuntu, Fedora, Kali) | 3 | $0 | $ 0.00 |
| **Total Estimated Cost** |  |  |  | **$118,774.71** |

(PB Technologies, 2025).

## Labour Breakdown

We have included a full breakdown of labour costs for this project in Appendix I. It covers each role needed in the project, including a supervisor, project manager, network engineers, and system architects, along with how many hours they will contribute and the associated costs. Based on current industry rates (PayScale, 2025), the total cost for labour sums up to **NZD$109,420.80.**

## Skills Analysis

All team members have existing Linux and networking experience but will need to upskill in specific networking tools such as iPerf and D-ITG. Some team members will need to upskill in certain Linux skill areas such as BASH scripting. Therefore, an upskilling plan schedule has been created.  
For a complete skills analysis matrix, please refer to Appendix J.

## Upskilling Plan Schedule

After the proposal is submitted on Sunday 6th April 2025, and assuming it is accepted at the proposal presentation during the following week, the team will have approximately two weeks of uninterrupted time for upskilling. During that time, it is expected that team members self-study using the resources provided and help each other learn where individual expertise exists. For the complete upskilling plan schedule, please see Appendix K.

## Requirement Traceability Matrix

The Requirement Traceability Matrix provides a distinct link between each project requirement and its corresponding evaluation case. This helps us stay organised and ensures all requirements are complete and thoroughly evaluated. It also supports progress tracking and confirms that all evaluation objectives are met. Please refer to Appendix L for the complete Requirement Traceability Matrix.

## Deliverables

For project-related documents, please refer to the scope statement in Appendix B.

The product-related deliverables include:

* Results of the evaluation for each Linux OS in a spreadsheet.
* Logs for the evaluation
* Results comparison document

## Success Criteria

The project will be considered successful upon delivery of valid performance data for all three operating systems, using both TCP and UDP protocols over IPv4 and IPv6, with all evaluation results falling within the 95% confidence interval. Success includes comprehensive documentation of methodology, complete raw data logs, and a final analysis comparing performance across all evaluated configurations.

# Project Management Methodology

|  |  |  |  |
| --- | --- | --- | --- |
| **Methodology** | **Core Approach** | **Pros** | **Cons** |
| Waterfall | A fixed set of phases, where each phase must be completed before moving to the next phase. (Atlassian, n.d.) | Waterfall outlines a clear project structure - this demonstrates clear cost and goals. Due to the fixed nature of the methodology, tracking is linear 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. (Atlassian, n.d.) | Due to the fixed nature of the methodology, its benefit can also be a drawback depending on the nature of the project. Projects that require 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 ultimately limits flexibility. (Atlassian, n.d.) |
| Scrum | Working quickly and collaboratively while promoting an environment that allows changes during the development cycle. (Asana, 2025) | The Scrum framework is the suggested approach for software development projects that prioritise 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. (Asana, 2025) | Due to the agile approach being constantly changed and reviewed after each sprint, limitation on resource planning can occur due to the unclear end objective.  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. (Asana, 2025) |
| Software testing lifecycle | The core approach (STLC) showcases the sequences of phases that occur during the evaluating process of software. Its main focuses are evaluating and understanding the product to ensure its working as expected  (Testim, 2025) | STLC prioritises identifying early flaws to prevent them becoming serious issues down the line. This reduces the end user faults making the software more stable. Due to the nature of capturing faults early in development results in reduced cost to fix issues as opposed to addressing the issues later in the development. Due to a solid evaluating foundation, a greater user satisfaction can be achieved.  (Dang, 2024) | The STLC process can pan out to be quite resource intensive, these include all the tools, infostructure and personal. There is also a possibility of over documentation which can result in slowing the development process down. STLC can also become difficult to implement for larger scale complex projects. (Just Academy, 2024) |

The project team has chosen to go with Software Testing Life Cycle (STLC) as our chosen methodology. This is due to the focus of our project being systematic evaluation of multiple operating systems, with evaluation confidence and validation.   
Unlike development-heavy methodologies like Waterfall or Scrum, STLC provides a dedicated, structured framework solely for evaluating, which aligns more closely with our project's objectives.   
It allows us to focus on requirement analysis, evaluation planning, evaluation case development, evaluation environment setup, evaluation execution, and evaluation closure in a clear and organised sequence.   
This ensures that each operating system configuration is thoroughly evaluated before moving forward, reducing the risk of undetected issues. By using STLC as our core methodology, we can maintain a high standard of accuracy and reliability throughout the project, while still allowing room for iteration and refinement within the evaluating phases.

## Project Phases

1. **Requirement Analysis**

* Project initiation, project scope, objectives, and deliverables
* Team contract, project charter
* Work Breakdown Structure (WBS)

1. **Evaluation Planning**

* Define evaluation data, scenarios and environment
* Resource planning
* Training requirements.

1. **Evaluation Case Development**

* Approved evaluation plan.
* Defined evaluation case scenarios.
* Validate with client.

1. **Evaluation Environment Setup**

* Establish Hardware and Software
* Prepare Evaluation Data
* Configure Evaluation Environment

1. **Evaluation Execution**

* All evaluations are performed, and results are documented
* Implement evaluation case failure protocol
* Analyse Performance Metrics to determine Operating System Performance

1. **Evaluation Closure**

* Prepare an excel spreadsheet of all the data for each operating system.
* Document Closure Report
* Client Feedback

Please refer to Appendix M for the Project Charter.

# Team Contract

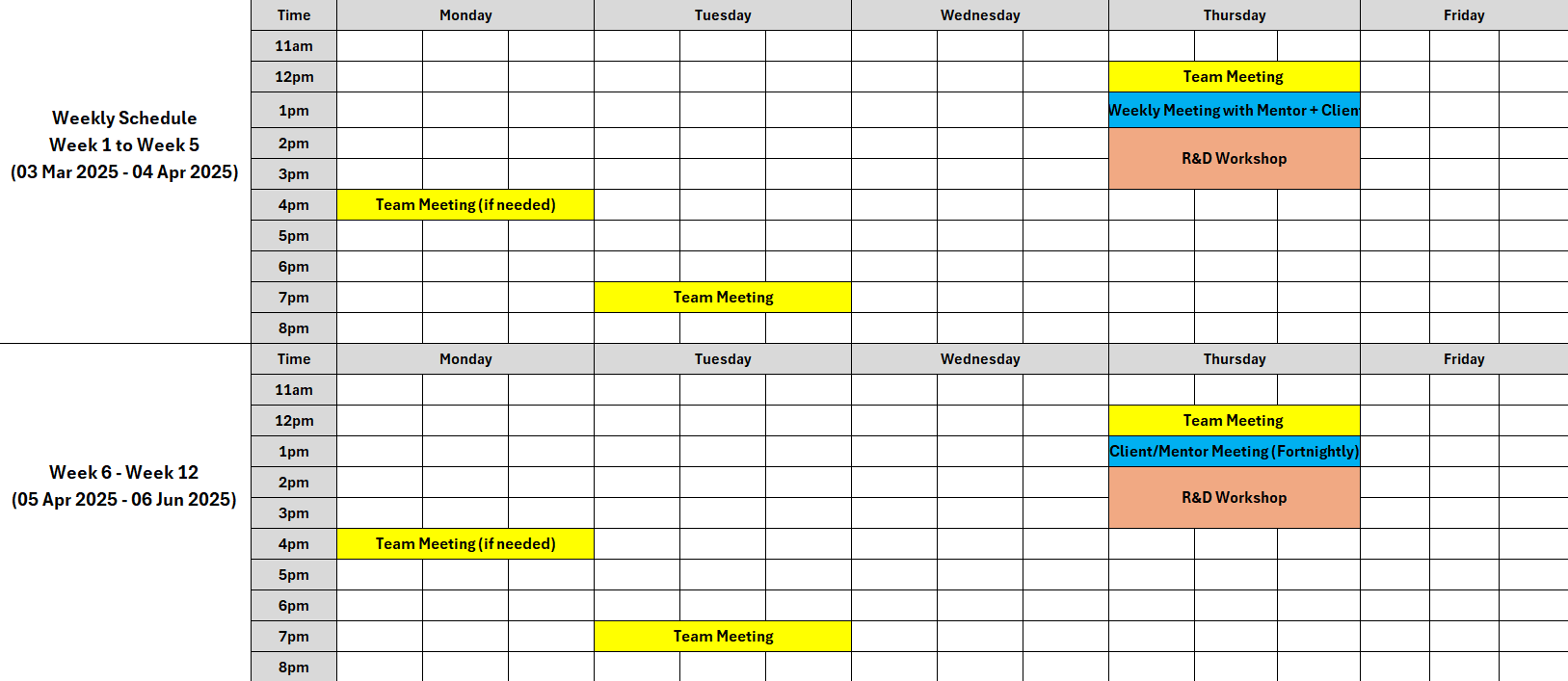
This section covers all the team members and the assigned roles. For complete details on team contract, please refer to Appendix N.

|  |  |  |
| --- | --- | --- |
| Member Name | Team Role | Assigned IP Version |
| Thomas Robinson | Project Manager | IPv4 |
| Win Phyo | System Architect | IPv4 |
| Zafar Afrad | Network Engineer | IPv4 |
| Kylie Afable | Network Engineer | IPv6 |
| Larissa Goh | Network Engineer | IPv6 |
| Nathan Quai Hoi | System Architect | IPv6 |
| Charmi Patel | Network Engineer | IPv6 |

Each Team members are responsible for allocating 12 – 15 hours per week as weekly commitment for the project. All team members are to complete the assigned tasks on timely manner, while constantly communicating with the team while adhering to the code of conduct and ethic.

# Team Schedule

The table below shows the frequency along with the time, date, location for our meetings set for Part 1 of the project. The team schedule for Part 2 has not been finalised due to unknown availability of the team members as of present.



# Risk Register

Risks are identified during planning and reviewed weekly during team meetings. Team members raise or are assigned risks, assess their causes and potential impact of each, and define suitable mitigation strategies. If a risk occurs, it’s escalated to the Issue Log where it’s tracked separately and monitored until it’s closed. Please refer to Appendix P and Appendix Q for the complete Risk Register and Risk Management Plan, respectively.

Below are two scope-related risks that have the potential to impact the project if not managed.

|  |  |  |
| --- | --- | --- |
| Risk | Description | Mitigation |
| **Tool replacement due to compatibility issues** | One of the client’s chosen tools D-ITG, has shown compatibility issues as it is not available in the Fedora package repositories and cannot be compiled from source. This creates a risk of inconsistent data as Fedora will lack D-ITG metrics that will be captured on Ubuntu and Kali. | The team will conduct additional evaluations with D-ITG on Fedora to determine feasibility. If it remains non-functional, iPerf will be used to maintain consistency across all operating systems. This will be documented and communicated to stakeholders to stay within the scope. |
| **Incorrect router configuration affecting evaluation results** | The project requires the configuration of Ubuntu, Fedora, and Kali as routers. Since routing directly affects packet flow, any error from misconfiguring routing tables or IP forwarding can compromise result accuracy. | The team will follow best practices for Linux router setups and validate configurations during the analysis phase. A checklist will be used for consistency, and a peer review process will ensure each router is configured correctly. |

# Issue Log

Issues are logged when risks manifest or when unexpected problems arise during the project. Each issue is assigned to a team member for resolution and tracked until resolved. The issue log is also reviewed during weekly meetings, and priority is escalated where needed to prevent impact on project delivery. Refer to Appendix R for the complete Issue Log.

# Change Management Plan

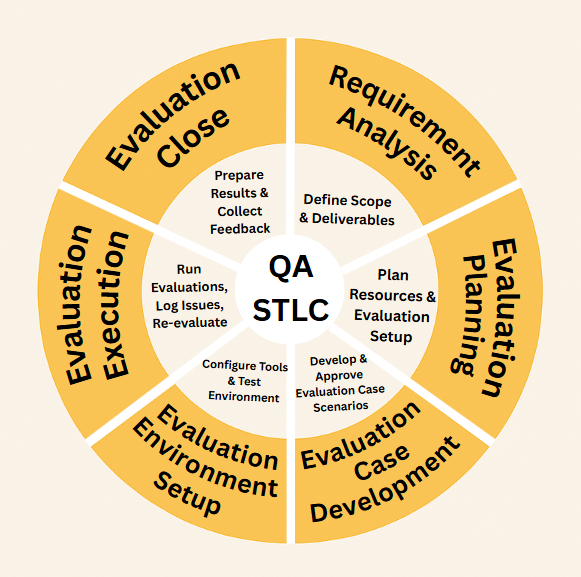
Due to the unpredictability of the project, there may require some changes at some point. This document outlines the approach and essential process to manage the changes. Please refer to Appendix S for the full Change Management Plan.

# Quality Assurance Plan

Quality will be managed through a structured Quality Assurance (QA) Plan which outlines the procedures, metrics, and checklists used to ensure evaluation accuracy, consistency, and compliance with project requirements.

The diagram below shows how quality assurance is applied throughout the project’s phases.

Please refer to Appendix T for the full QA Plan.



# Appendices

## 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.

## Appendix B – Scope Statement

|  |
| --- |
| **Project Title: Network Performance Evaluation on Linux Based Operating Systems**  **Date: 30/03/2025** **Prepared by:** Win Phyo & Thomas Robinson |
| **Project Justification**  This project is designed to evaluate the network performance of IPv4 and IPv6 on the latest versions of three popular Linux-based operating systems, so that the client may compare the data with historical data to observe changes in performance over time and between operating systems.  Both IPv4 and IPv6 have their own benefits and drawbacks, of which this project intends to evaluate.  With this data, the client can have a better understanding and comparison of how each of the chosen operating systems handles network traffic, which ultimately helps to make adequate decisions on optimising network infrastructure. |
| **Product Characteristics and Requirements**  **Functional Requirements:**   1. Configure Fedora, Ubuntu, and Kali Linux as software routers 2. Implement both TCP and UDP transmission protocols 3. Support both IPv4 and IPv6 addressing 4. Set up a four-PC network topology (1 sender, 2 routers, 1 receiver) 5. Configure dual NICs on router PCs and single NICs on sender/receiver PCs 6. Perform three complete evaluation rounds, one for each operating system 7. Generate network traffic using D-ITG or iPerf 8. Evaluate with 12 specified packet sizes (128 to 1536 Bytes) 9. Run each evaluation configuration at least 10 times 10. Identify and re-run evaluations falling outside 95% confidence interval 11. Record throughput performance across all evaluation scenarios 12. All evaluations must be run 10 times each with all evaluations at 95% confidence 13. Measure network delays in all evaluation configurations 14. Calculate jitter values throughout evaluations 15. Track packet loss rates for all evaluation cases 16. Log all evaluations 17. Generate statistical averages using the evaluation tool 18. Compile final statistical data for comparison and analysis   **Non-Functional Requirements:**   1. Results must fall within a 95% confidence interval 2. Statistical validation of evaluation results 3. Consistent evaluation methodology across all platforms and evaluation rounds 4. Evaluation environment must support the full range of packet sizes 5. Network infrastructure must handle required throughput levels 6. Measurement tools must be precise enough to capture microsecond-level jitter 7. Evaluation environment must maintain consistent conditions across evaluation rounds 8. Hardware configurations must remain stable throughout all evaluations 9. Evaluation tools must produce repeatable results 10. Complete logs of all evaluation runs must be preserved 11. Detailed recording of evaluation configurations and parameters 12. Final deliverable must include both raw logs and statistical summaries 13. Evaluation must be organised into three distinct rounds (one per OS) 14. Consistent evaluation procedures must be followed across all operating system rounds. 15. Comparison data must allow for direct analysis between operating systems   **Out of Scope:**   1. Acquisition of the four PCs required for evaluation 2. Physical installation and configuration of hardware 3. Initial operating system installations 4. Network facility and power supply 5. Physical security of the evaluation environment 6. Environmental controls of the evaluation facility 7. Long-term maintenance of the evaluation environment 8. Ongoing support for the configured systems 9. Application of findings to production networks 10. Performance optimisation recommendations beyond evaluation results 11. Evaluation of operating systems beyond the specified three distributions 12. Network configurations other than the specified topology 13. Evaluation of protocols beyond TCP and UDP 14. Performance evaluation at speeds exceeding gigabit Ethernet 15. Addressing performance issues identified during evaluation 16. Implementing changes to improve network performance 17. Troubleshooting underlying hardware issues |
| **Summary of Project Deliverables**  **Project management-related deliverables:**   * Meeting Agendas * Meeting Minutes * Project Charter * Team Contract * Stakeholder Register * Stakeholder Management Strategy * Scope Statement * Project Proposal * Risk Register * Issue Log * Communication Management Plan * Work Breakdown Structure (WBS) * Gantt Chart * Skills Analysis * Upskilling Plan Schedule * Estimated Cost Breakdown * Milestone Report * Lesson-Learned Report * Mid-Term Review * Final Product Presentation   **Product-related deliverables:**   * Complete evaluation plans detailing procedures for all configurations * Configuration documentation for router setup on each operating system * Scripts or configuration files used for evaluation * Spreadsheet of performance evaluation results * Raw evaluation logs from all execution runs (10 runs per configuration) * Data analysis documentation with statistical validation |
| **Project Success Criteria:**   * Delivery of all specified project deliverables * Statistical validity of all evaluation results * Clear presentation of comparative performance across operating systems * Adherence to project timeline and budget constraints * Comprehensive documentation allowing for evaluation reproducibility |

## Appendix C – Communication Stakeholder Register

**Communication Stakeholder Register for** **Network Performance Evaluation on Linux Based Operating Systems - Physical Environment**

**Version: 1.0**

**Prepared by:** Nathan Quai Hoi **Date: 11/03/2025**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Position** | **Internal /External** | **Project Role** | **Contact Information** |
| **Nathan Quai Hoi** | Team Member | Internal | System Architect | wgk6332@autuni.ac.nz |
| **Win Phyo** | Team Member | Internal | System Architect | ddk8093@autuni.ac.nz |
| **Zafar Azad** | Team Member | Internal | Network Engineer | ftk8708@autuni.ac.nz |
| **Thomas Robinson** | Team Leader | Internal | Project Manager | cgr2690@autuni.ac.nz |
| **Larissa Goh** | Team Member | Internal | Network Engineer | xhm5236@autuni.ac.nz |
| **Charmi Patel** | Team Member | Internal | Network Engineer | fhv9735@autuni.ac.nz |
| **Kylie Afable** | Team Member | Internal | Network Engineer | cjq7738@autuni.ac.nz |
| **Daniel Vaipulu** | Project Supervisor | Internal | Project Supervisor | daniel.vaipulu@aut.ac.nz |
| **Raymond Lutui** | Client | Internal | Client | raymond.lutui@aut.ac.nz |
| **Ramon Lewis** | IT Support | Internal | IT Support | ramon.lewis@aut.ac.nz |
| **Dr. Ramesh Lal** | Lecturer | Internal | Lecturer | ramesh.lal@aut.ac.nz |
| **Prof. Catherine Shi** | Lecturer | Internal | Lecturer | catherine.shi@aut.ac.nz |
| **Olivia Tang** | Lecturer | Internal | Lecturer | olivia.tang@aut.ac.nz |
| **Professor Damon Salesa** | Vice-Chancellor AUT | Internal | Senior Manager | vicechancellor@aut.ac.nz |
| **Minh Nguyen** | Head of Computer Sciences | Internal | Senior Manager | minh.nguyen@aut.ac.nz |
| **Tek Tjing Lie** | Interim Head of ECMS | Internal | Senior Manager | tek.lie@aut.ac.nz |
| **Terry Brydon** | School Manager/ Accountant | Internal | Senior Manager | terry.brydon@aut.ac.nz |
| **Matthew Kuo** | BCIS Program Director | Internal | Project Sponsor | matthew.kuo@aut.ac.nz |
| **Roy Cullum** | Head of ICT | Internal | Consultant | roy.cullum@aut.ac.nz |

## Appendix D – Stakeholder Management Plan

**Stakeholder Management Plan for**

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

**Prepared by: Nathan Quai Hoi** **Date: 19/03/2025**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Level of Interest** | **Level of Influence** | **Potential Management Strategies** |
| **Daniel Vaipulu** | High | High | Daniel is very resourceful and relaxed. He makes sure to engage the project at a reasonable pace. We will need to keep him updated with what is happening regularly. |
| **Raymond Lutui** | High | High | Raymond is very relaxed, while also having high expectations. We will need to keep him updated with what is happening. |
| **Thomas Robinson** | High | Medium | Thomas would like to ensure all team members are on the same page and keep track of the project tasks. He develops plans and allocates tasks to achieve goals. |
| **Ramon Lewis** | Low | High | Ramon would like to know details of our setup for evaluation and would like to communicate via Teams chat. |
| **Dr. Ramesh Lal** | High | Low | He would like to know the progress of the project and be up to date with any concerns or problems regarding the project |
| **Prof. Catherine Shi** | High | Low | She would like to know the progress of project and consult to team members to ensure project success. |
| **Olivia Tang** | High | Low | She would like to know the progress of project and consult to team members to ensure project success. |
| **Matthew Kuo** | High | Low | He would like to know the progress of project because if project is successful, BCIS will attract more students each year and reduce the cost of software, increase revenue but he would not come to help team. |
| **Tek Tjing Lie** | High | Low | He would like to stay informed and maintain interest in the project. |

## Appendix E – Tools Comparison

|  |  |  |
| --- | --- | --- |
| **Comparison** | **iPerf** | **D-ITG** |
| Performance | * Measures the maximum achievable bandwidth on IP networks. * Reports throughput, bitrate, loss and other parameters for each test. * Consists of features such as nuttcp and netperf. * Not backwards compatible with original iPerf. | * D-ITG measures delay, jitter, loss throughput. * Uses logs to calculate performance metrics.   D-ITG compromises of different components:   * + ITGSend   + ITGRecv   + ITGLog   + ITGManager   + ITGDec |
| Feature 1 | * Rewritten from scratch * Goal is to have a smaller, simpler code base. * Has a library version for other programs | Supports layer 7 features:   * Predefined stochastic profiles (e.g. Telnet, VolP). * Custom payload (random or a file) * Both PS and IDT support stochastic processes. |
| Feature 2 | * Measures bandwidth and performance * Supports TCP and UDP * Reports packet loss and delay | Packet-level QoS metrics:   * Bitrate, packet rate, one way relay, round trip time, jitter and packet loss. |

## Appendix F – Milestone Report

**Author: Larissa Goh**

**Date: 06/04/2025**

**Version: 1.0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Date** | **Status** | **Responsible** | **Issues/Comments** |
| Team Portfolio Structure | 19/03/25 | Completed | Thomas | Done on time |
| Project Schedule | 25/03/25 | Completed | Win | Done on time |
| Stakeholder Register V1 | 19/03/25 | Completed | Nathan | Done on time |
| Team Contract | 20/03/25 | Completed | Win | Done on time and signed by all team members |
| Client Contract | 02/04/25 | Completed | Thomas | Done on time, not yet signed by client |
| Scope Statement | 25/03/25 | Completed | Nathan | Done on time |
| Project Methodology | 02/03/25 | Completed | Zafar | Done on time, revised multiple times based on feedback. |
| Project Feasibility | 27/03/25 | Completed | Kylie | Done on time |
| Upskilling Recommendation List V1 | 20/03/25 | Completed | Thomas | Done on time |
| Skills Analysis & Matrix | 20/03/25 | Completed | Thomas | Done on time |
| Stakeholder Management Strategy V1 | 20/03/2025 | Completed | Nathan | Done on time |
| Communication Stakeholder Register | 04/04/2025 | Completed | Win | Done on time |
| Project Charter | 04/04/2025 | Completed | Thomas | Done on time |
| Change Management Plan V1 | 02/04/2025 | Completed | Win | Done on time |
| Communication Plan V1 | 20/03/25 | Completed | Charmi | Done on time |
| Gantt Chart V1 | 20/03/25 | Completed | Nathan | Done on time, updated twice to reflect revised timeline and methodology alignment. |
| Milestone Report V1 | 06/04/25 | Completed | Larissa | Done on time |
| Work Breakdown Structure V1 | 20/03/25 | Completed | Win | Done on time, updated twice to reflect revised timeline and methodology alignment. |
| Network Diagram | 31/03/25 | Completed | Nathan | Done on time |
| Issue Log V1 | 05/04/25 | Completed | Larissa | Done on time, populated from early-project challenges. |
| Risk Register V1 | 23/03/25 | Completed | Larissa | Done on time |
| Meeting agenda for kick-off meeting | 19/03/25 | Completed | Zafar | Done on time |
| Meeting agenda for meeting #2 | 25/03/25 | Completed | Thomas | Done on time |
| Meeting agenda for meeting #3 | 30/03/25 | Completed | Thomas | Done on time |
| Meeting minutes for kick-off meeting | 19/03/25 | Completed | Win | Done on time |
| Meeting minutes for meeting #2 | 27/03/25 | Completed | Win | Done on time |
| Meeting minutes for meeting #3 | 03/04/25 | Completed | Win | Done on time |
| Estimated Costs Breakdown | 31/03/25 | Completed | Charmi | Done on time |
| Quality Assurance Plan V1 | 05/04/25 | Completed | Larissa | Done on time |
| Project Proposal Submission | 06/04/25 | Completed | Thomas | Done on time |
| Project Proposal Presentation | 10/04/25 | Pending | Team |  |

## Appendix G – Work Breakdown Structure

**Prepared By:** Win Phyo & Thomas Robinson

**Date:** 02/04/2025

**Project Name:** Linux Network Performance Evaluation

1. Requirement Analysis
   1. Entry Criteria
      1. Team Kick-Off Meeting Agenda
      2. Team-Client Kick-Off Meeting Agenda
      3. Team Contract
      4. Meeting Minutes (Kick-Off Meetings)
   2. Develop Project Proposal
      1. Communication Stakeholder Register
      2. Project Charter
      3. Client Contract
      4. Meeting Minutes (Team & Client)
   3. Develop Project Plans
      1. Stakeholder Management Strategy
      2. Risk Register
      3. Issue Log
      4. Communication Management Plan
      5. Change Management Plan
      6. Scope Statement
      7. Network Diagram
      8. Requirement Traceability Matrix
      9. Skills Analysis Matrix
      10. Upskilling Plan Schedule
      11. Project Schedule
          1. Work Breakdown Structure
          2. Gantt Chart
      12. Milestone Report
      13. Estimated Cost
      14. Quality Assurance Plan
      15. Project Proposal submission
      16. Project Proposal presentation

**Milestone 1 – Project Proposal**

* 1. Exit Criteria
     1. Client feedback on proposal document
     2. Approval of proposal document from client

1. Evaluation Planning
   1. Entry Criteria
      1. Client approval of project proposal
   2. Researching And Upskilling
      1. Team members upskill for necessary skills
      2. Network tools (iPerf & D-ITG)
   3. Define evaluation scenarios
   4. Define evaluation environment
   5. **Milestone 2 – Mid-Term Review**
   6. Exit Criteria
      1. Client and supervisor feedback on evaluation planning
2. Evaluation Case Development
   1. Entry Criteria
      1. Client and supervisor feedback on evaluation planning
   2. Define evaluation scenario
      1. Requirement traceability
   3. Exit Criteria
      1. Review and validate evaluation cases
      2. Validate with client
      3. Client feedback
3. Evaluation Environment Setup
   1. Entry Criteria
      1. Establishment of hardware and software
      2. Prepare evaluation data
      3. Configure evaluation environment
         1. Configure two computers as routers
         2. Configure three separate networks
      4. Write cases for each scenario
   2. Exit Criteria
      1. Fully functional evaluation environment and approved evaluation cases
      2. Client feedback
4. Evaluation Execution
   1. Entry Criteria
      1. All exit criteria from previous steps
         1. Evaluation environment is approved and functional
   2. Exit Criteria
      1. All evaluations are performed, and results are documented
         1. Conduct a minimum of 10 evaluation runs for each operating system
            1. Configure environment
            2. Each evaluation consists of 12 packet sizes
            3. Each evaluation will be run on both IPv4 and IPv6 separately
            4. Each evaluation will be run on both TCP and UDP separately
         2. Implement evaluation case failure protocol
         3. Client feedback
         4. Execute evaluation cases, scripts, pings
         5. Record and document results for each evaluation run
         6. Analyse performance metrics to determine operating system performance
         7. Client feedback

**Milestone 3 – Client Review and Feedback 1**

**Milestone 4 – Client Review and Feedback 2**

**Milestone 5 – Client Review and Feedback 3**

1. Evaluation Closure
   1. Entry Criteria
      1. Compile and compare results of evaluation
         1. Analyse performance pattern
         2. Summary of the network performance evaluation
         3. Create visualisations and reports
         4. Final poster
   2. Exit Criteria
      1. Document closure report
      2. Evaluation results and analysis
      3. Organise folders for portfolio
      4. Client feedback

**Milestone 6 – Final Poster & Portfolio**

## Appendix H – Gantt Chart

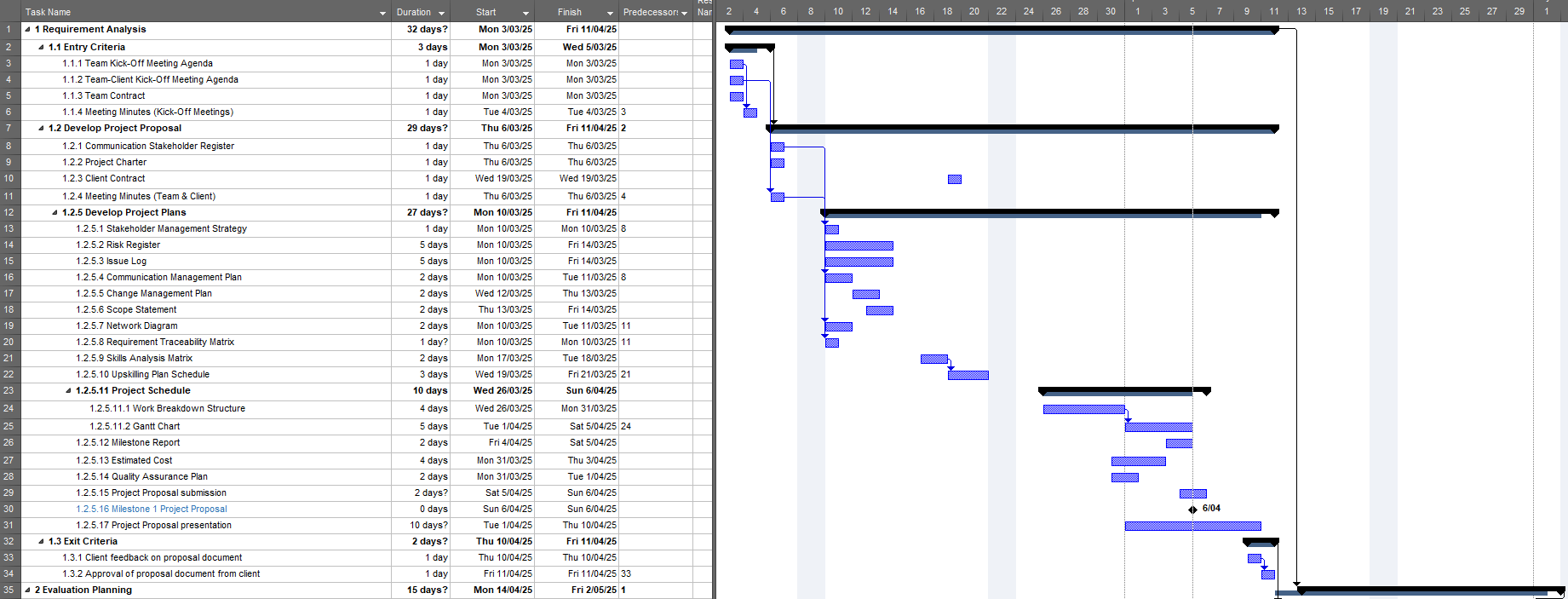


Figure A – Requirement Analysis

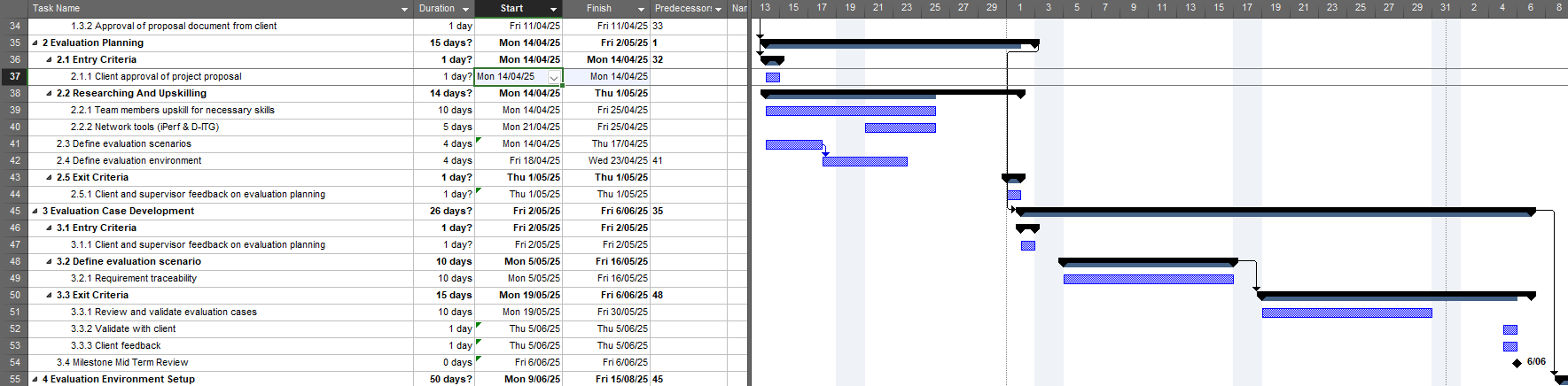


Figure B – Evaluation Planning and Evaluation Case Development

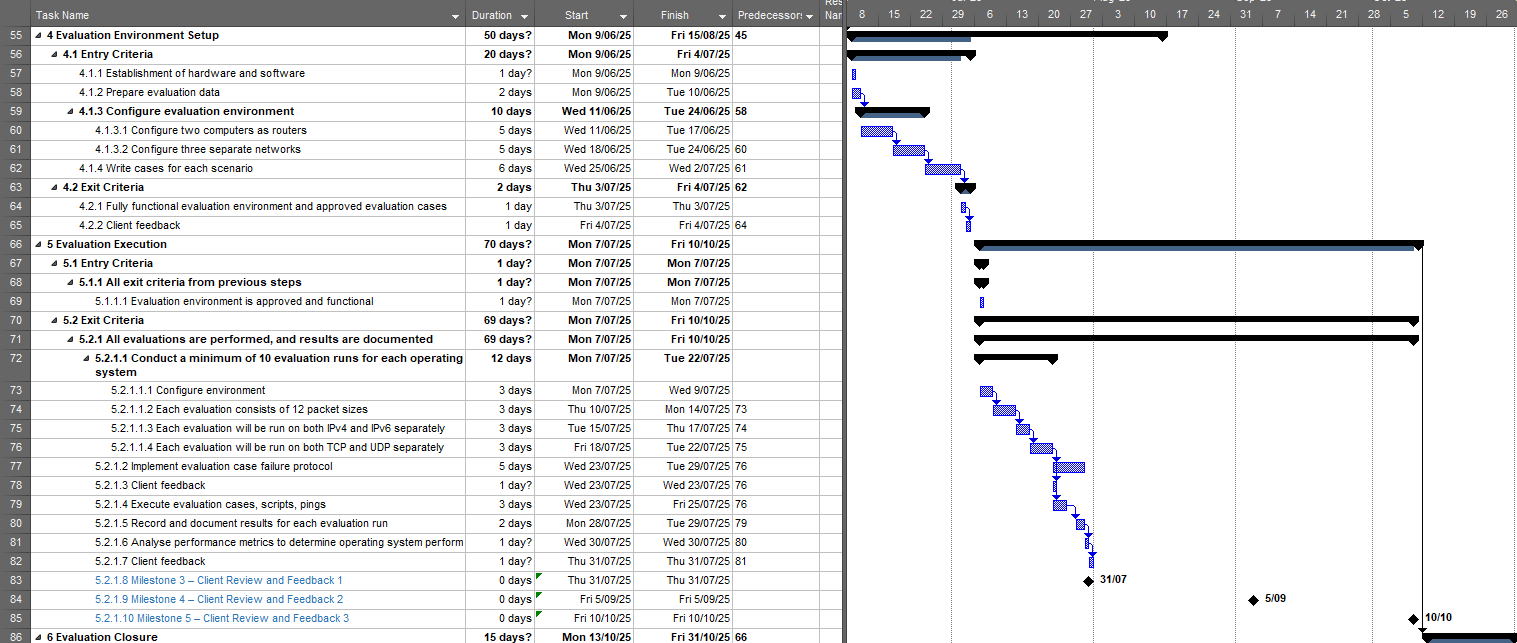


Figure C – Evaluation Environment Setup & Evaluation Execution

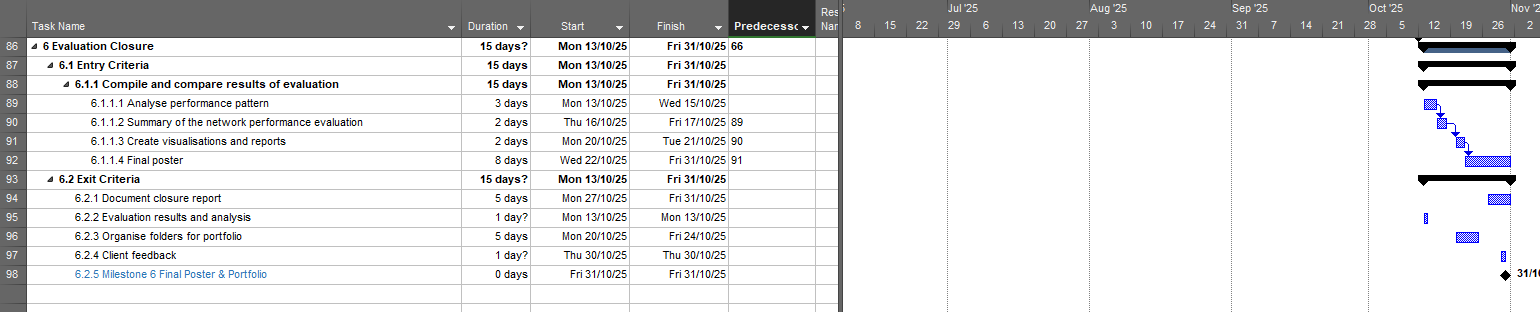


Figure D – Evaluation Closure

## Appendix I – Labour Breakdown Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Cost Breakdown** | **Average pay per hour (inc. GST)** | **Total Hours** | **Total Cost (NZD)** |
| Supervisor | Provided by AUT:  $142 + GST (15%) | $163.30 | 1 hour per week  24 weeks (2 sems)  1 x 24 = **24** hrs | 163.30 x 24  = **$3,919.20** |
| Project Manager | Average yearly salary: $98,461 (PayScale, 2025) | $47.34 | 15 hours per week  24 weeks (2 sems)  15 x 24 = **360** hrs | 360 x 47.24  = **$17,006.40** |
| Network Engineer  (x4) | Average yearly salary: 78,377 (PayScale, 2024) | $37.68 | 15 hours per week  24 weeks (2 sems)  15 x 24 = 360  360 x 4 = **1440** hrs | 1440 x 37.68  = **$54,259.20** |
| System Architect  (x2) | Average yearly salary: $98,895 (PayScale, 2023) | $47.55 | 15 hours per week  24 weeks (2 sems)  15 x 24 = 360hrs  360 x 2 = **720** hrs | 720 x 47.55  = **$34,236.00** |

## Appendix J – Skills Analysis Matrix

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Skills Matrix** | | | | | | | | | | |
| **Team Members** | **Linux Installation and Configuration** | **Linux as Router Configuration** | **BASH Scripting** | **Subnetting IPv4** | **Subnetting IPv6** | **Understanding of Networking Protocols (TCP/UDP)** | **iPerf Usage** | **D-ITG Usage** | **Basic Network Performance Metrics (Throughput, Delay, Jitter, Packet Loss)** | **Basic Data Analysis Skills** |
| **Kylie Afable** | **2** | **0** | **2** | **1** | **1** | **1** | **0** | **0** | **0** | **0** |
| **Zafar Azad** | **2** | **0** | **0** | **3** | **1** | **3** | **0** | **0** | **2** | **1** |
| **Larissa Goh** | **2** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **1** | **1** |
| **Nathan Quai Hoi** | **2** | **0** | **1** | **2** | **1** | **1** | **0** | **0** | **0** | **0** |
| **Charmi Patel** | **2** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **1** |
| **Win Phyo** | **2** | **0** | **0** | **2** | **0** | **1** | **0** | **0** | **1** | **1** |
| **Thomas Robinson** | **3** | **0** | **2** | **2** | **1** | **2** | **0** | **0** | **1** | **2** |
| **Total** | **15** | **0** | **5** | **12** | **4** | **10** | **0** | **0** | **5** | **6** |
| **Average** | **2.14** | **0.00** | **0.71** | **1.71** | **0.57** | **1.43** | **0.00** | **0.00** | **0.71** | **0.86** |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | **Key** | | | |  |  |  |  |
|  |  |  | **0** | **1** | **2** | **3** |  |  |  |  |
|  |  |  | Unskilled | Low-Skilled | Competent | Expert |  |  |  |  |

## Appendix K – Upskilling Plan Schedule

This upskilling plan covers the two-week period following proposal acceptance (approximately 14th-25th April 2025). Team members should self-study using these resources and collaborate to share knowledge where individual expertise exists.

**Weekly Schedule**

**Week 1: Fundamentals & Environment Setup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Day | Focus Area | Resources | Priority | Expected Outcomes |
| 1 | **Environment Setup** | * WSL (Windows users): <https://learn.microsoft.com/en-us/windows/wsl/install> * Homebrew (Mac users): <https://brew.sh/> * Asahi Linux (Mac users): <https://asahilinux.org/> | High | Working Linux environment ready for practice |
| 2-4 | **Linux Fundamentals** | * Linux Journey: <https://linuxjourney.com/> * LinuxCommand.org: <https://linuxcommand.org/> * Command Line for Beginners: <https://www.freecodecamp.org/news/command-line-for-beginners/> | High | Comfortable with basic Linux navigation and commands |
| 5 | **Networking Basics** | * <https://www.freecodecamp.org/news/computer-networking-how-applications-talk-over-the-internet/> * <https://www.freecodecamp.org/news/linux-networking-commands-for-beginners/> * <https://www.freecodecamp.org/news/osi-model-networking-layers-explained-in-plain-english/> | Medium | Revised understanding of basic networking concepts |

**Week 2: Advanced Topics & Tool Familiarisation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Day | Focus Area | Resources | Priority | Expected Outcomes |
| 6-8 | **Linux as a Router** | * <https://fedoramagazine.org/use-fedora-server-create-router-gateway/> * <https://www.ibm.com/docs/en/linux-on-systems?topic=3-linux-as-router> * <https://www.computernetworkingnotes.com/linux-tutorials/how-to-configure-and-use-linux-as-a-router.html> * <https://www.baeldung.com/linux/server-router-configure> * <https://medium.com/@lfoster49203/setting-up-ubuntu-as-a-router-with-advanced-routing-features-4511abc5e1eb> * <https://www.tecmint.com/setup-linux-as-router/> * <https://wiki.archlinux.org/title/Router> | High | Ability to configure basic routing between subnets |
| 9-10 | **Evaluation Tools** | * D-ITG: <https://github.com/jbucar/ditg> * iPerf: <https://iperf.fr/> | High | Familiarity with installing and using network evaluation tools |

**Focus Linux Distributions**

|  |  |  |  |
| --- | --- | --- | --- |
| Distribution | Description | Use Case | Package Manager |
| Ubuntu | Based on Debian Unstable; beginner-friendly | Desktop use, beginners | APT |
| Fedora | Upstream for Red Hat Enterprise Linux; bleeding edge | Desktop use, beginners | DNF |
| Kali | Based on Debian Testing; security-focused, for penetration testing | Penetration testing | APT |

**Extended Learning (Optional)**

For team members who complete the core curriculum ahead of schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| Resource | Description | Value | Time Investment |
| Linux From Scratch | Build your own Linux distro  <https://www.linuxfromscratch.org/lfs/> | Deep understanding of Linux internals | High |
| GNU/Linux Desktop Survival Guide | Desktop Linux usage guide  <https://www.togaware.com/linux/survivor/> | Improved daily workflow | Medium |
| Red Hat Certifications | Industry-recognised qualifications  <https://www.redhat.com/en> | Professional development | High |

**Collaboration Guidelines**

* **Daily Check-ins**: Brief daily updates on progress and challenges
* **Knowledge Sharing**: Schedule sessions where team members can teach others about areas of expertise
* **Troubleshooting**: Document common issues and solutions for team reference
* **Practical evaluating**: Set up small evaluation networks to apply theoretical knowledge

**Progress Tracking**

Team members should track their progress through this plan in their logbook, noting:

* Completed resources
* Skills mastered
* Areas requiring additional focus
* Questions for team discussion

This approach ensures a structured yet flexible learning experience that prepares the team for successful project implementation.

## Appendix L – Requirement Traceability Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Requirement Traceability Matrix** | | | | | | |
| **Project Name:** | **Network Performance Evaluation on Linux Based Operating Systems in a Physical Environment** | | | | | |
| **Prepared By:** | **Thomas Robinson** | | | | | |
| **Traceability #** | **Category** | **Requirement ID** | **Requirement Description** | **Evaluation Case ID** | **Evaluation Case Description** | **Status** |
| 1 | **Functional** | R1 | Configure Fedora, Ubuntu, and Kali Linux as software routers | EC-F1.1 | Verify installation and configuration of routing services on all three Linux distributions (Fedora, Ubuntu, and Kali) | To Do |
| 2 |  | R2 | Implement both TCP and UDP transmission protocols | EC-F1.2 | Configure and validate both ECP and UDP traffic forwarding on all router systems | To Do |
| 3 |  | R3 | Support both IPv4 and IPv6 addressing | EC-F1.3 | Configure dual-stack IPv4/IPv6 addressing and verify connectivity using both protocols | To Do |
| 4 |  | R4 | Set up a four-PC network topology (1 sender, 2 routers, 1 receiver) | EC-F1.4 | Establish physical connections and verify network connectivity across the four-PC topology | To Do |
| 5 |  | R5 | Configure dual NICs on router PCs and single NICs on sender/receiver PCs | EC-F1.5 | Verify correct NIC configuration and interface assignments on all four systems | To Do |
| 6 |  | R6 | Perform three complete evaluation rounds, one for each operating system | EC-F1.6 | Execute full evaluation cycles with identical evaluation parameters for Fedora, Ubuntu, and Kali Linux | To Do |
| 7 |  | R7 | Generate network traffic using iPerf or D-ITG | EC-F1.7 | Install and configure traffic generation tools and verify they produce appropriate network loads | To Do |
| 8 |  | R8 | Evaluate with 12 specified packet sizes (128 to 1536 Bytes) | EC-F1.8 | Configure traffic generator to cycle through all 12 required packet sizes and verify data collection | To Do |
| 9 |  | R9 | Run each evaluation configuration at least 10 times | EC-F1.9 | Execute and document 10 complete evaluation runs for each configuration combination | To Do |
| 10 |  | R10 | Identify and re-run evaluations falling outside 95% confidence interval | EC-F1.10 | Analyse initial results, identify outliers, and verify re-evaluation of anomalous configurations | To Do |
| 11 |  | R11 | Record throughput performance across all evaluation scenarios | EC-F1.11 | Verify throughput data collection and storage for all evaluation combinations | To Do |
| 12 |  | R12 | All evaluations must be run 10 times each with all evaluations at 95% confidence | EC-F1.12 | Validate statistical significance of all collected data across the required number of evaluation iterations | To Do |
| 13 |  | R13 | Record throughput performance across all evaluation scenarios | EC-F1.13 | Verify consistent throughput measurement methodology across all evaluation configurations | To Do |
| 14 |  | R14 | Measure network delays in all evaluation configurations | EC-F1.14 | Validate latency measurement and recording across all evaluation scenarios | To Do |
| 15 |  | R15 | Calculate jitter values throughout evaluations | EC-F1.15 | Verify jitter calculation methodology and confirm data collection in all evaluation cases | To Do |
| 16 |  | R16 | Track packet loss rates for all evaluation cases | EC-F1.16 | Validate packet loss detection and recording mechanisms for all configurations | To Do |
| 17 |  | R17 | Log all evaluations | EC-F1.17 | Confirm comprehensive logging system captures all required evaluation parameters and results | To Do |
| 18 |  | R18 | Generate statistical averages using the evaluation tool | EC-F1.18 | Verify statistical processing capabilities and validate output formats | To Do |
| 19 |  | R19 | Compile final statistical data for comparison and analysis | EC-F1.19 | Validate data aggregation procedures and verify final comparative analysis outputs | To Do |
| 20 | **Non-Functional** | R1 | Results must fall within a 95% confidence interval | EC-NF1.1 | Verify statistical analysis procedures confirm 95% confidence interval for all accepted results | To Do |
| 21 |  | R2 | Statistical validation of evaluation results | EC-NF1.2 | Audit statistical methodology and confirm validation processes for all collected metrics | To Do |
| 22 |  | R3 | Consistent evaluation methodology across all platforms and evaluation rounds | EC-NF1.3 | Verify evaluation procedures are documented and consistently applied across all operating systems | To Do |
| 23 |  | R4 | Evaluation environment must support the full range of packet sizes | EC-NF1.4 | Validate network infrastructure can handle all required packet sizes without fragmentation or loss | To Do |
| 24 |  | R5 | Network infrastructure must handle required throughput levels | EC-NF1.5 | Measure maximum throughput capability of evaluation environment and verify it exceeds evaluation requirements | To Do |
| 25 |  | R6 | Measurement tools must be precise enough to capture microsecond-level jitter | EC-NF1.6 | Calibrate and verify precision of jitter measurement tools against known reference values | To Do |
| 26 |  | R7 | Evaluation environment must maintain consistent conditions across evaluation rounds | EC-NF1.7 | Monitor and document environmental conditions throughout evaluation to verify consistency | To Do |
| 27 |  | R8 | Hardware configurations must remain stable throughout all evaluations | EC-NF1.8 | Implement configuration control procedures and verify hardware stability between evaluation runs | To Do |
| 28 |  | R9 | Evaluation tools must produce repeatable results | EC-NF1.9 | Perform calibration evaluations to verify result consistency with identical input parameters | To Do |
| 29 |  | R10 | Complete logs of all evaluation runs must be preserved | EC-NF1.10 | Audit logging system to verify comprehensive capture and secure storage of all evaluation data | To Do |
| 30 |  | R11 | Detailed recording of evaluation configurations and parameters | EC-NF1.11 | Verify configuration documentation completeness for all evaluation scenarios | To Do |
| 31 |  | R12 | Final deliverable must include both raw logs and statistical summaries | EC-NF1.12 | Validate output format includes both raw data logs and processed statistical summaries | To Do |
| 32 |  | R13 | Evaluation must be organised into three distinct rounds. (one per OS) | EC-NF1.13 | Verify evaluation plan organisation and implementation of separate, controlled evaluation cycles | To Do |
| 33 |  | R14 | Consistent evaluation procedures must be followed across all operating system rounds. | EC-NF1.14 | Audit evaluation execution logs to confirm procedural consistency across all evaluation rounds | To Do |
| 34 |  | R15 | Comparison data must allow for direct analysis between operating systems | EC-NF1.15 | Validate data normalisation and comparison methodologies for cross-platform analysis | To Do |
| 35 | **Out of Scope** | R1 | Acquisition of the four PCs required for evaluation | N/A | Not applicable - outside project scope | To Do |
| 36 |  | R2 | Physical installation and configuration of hardware | N/A | Not applicable - outside project scope | To Do |
| 37 |  | R3 | Initial operating system installations | N/A | Not applicable - outside project scope | To Do |
| 38 |  | R4 | Network facility and power supply | N/A | Not applicable - outside project scope | To Do |
| 39 |  | R5 | Physical security of the evaluation environment | N/A | Not applicable - outside project scope | To Do |
| 40 |  | R6 | Environmental controls of the evaluation facility | N/A | Not applicable - outside project scope | To Do |
| 41 |  | R7 | Long-term maintenance of the evaluation environment | N/A | Not applicable - outside project scope | To Do |
| 42 |  | R8 | Ongoing support for the configured systems | N/A | Not applicable - outside project scope | To Do |
| 43 |  | R9 | Application of findings to production networks | N/A | Not applicable - outside project scope | To Do |
| 44 |  | R10 | Performance optimisation recommendations beyond evaluation results | N/A | Not applicable - outside project scope | To Do |
| 45 |  | R11 | Evaluation of operating systems beyond the specified three distributions | N/A | Not applicable - outside project scope | To Do |
| 46 |  | R12 | Network configurations other than the specified topology | N/A | Not applicable - outside project scope | To Do |
| 47 |  | R13 | Evaluation of protocols beyond TCP and UDP | N/A | Not applicable - outside project scope | To Do |
| 48 |  | R14 | Performance evaluation at speeds exceeding gigabit Ethernet | N/A | Not applicable - outside project scope | To Do |
| 49 |  | R15 | Addressing performance issues identified during evaluation | N/A | Not applicable - outside project scope | To Do |
| 50 |  | R16 | Implementing changes to improve network performance | N/A | Not applicable - outside project scope | To Do |
| 51 |  | R17 | Troubleshooting underlying hardware issues | N/A | Not applicable - outside project scope | To Do |

## Appendix M – Project Charter

Project Charter

**Project Title**:

Network Performance Evaluation on Linux Based Operating Systems – Physical Team

**Project Start Date:** 03/03/2025 **Projected Finish Date:** 31/10/2025

**Budget Information:**

The total estimated cost of the project is $118,774.71.

**Project Manager:**

Thomas Robinson ([cgr2690@autuni.ac.nz](mailto:cgr2690@autuni.ac.nz))

**Project Objectives:**

The goal of this project is to evaluate the performance of 3 latest Linux based Operating Systems distributions – Fedora, Ubuntu, Kali Linux. These operating systems need to be configured as software routers before running the evaluations. Tools such as iPerf or D-ITG are to be employed to generate the evaluation traffics.

**Main Project Success Criteria:**

Successfully gather data comparing IPv4 and IPv6, using TCP and UDP protocols, across 12 given packet sizes on 3 different Linux operating systems configured as software routers.

**Approach:**

The project team has chosen to go with Software Testing Life Cycle (STLC) as our chosen methodology. This is due to the focus of our project being systematic evaluation of multiple operating systems, with evaluation confidence and validation.

Unlike development-heavy methodologies like Waterfall or Scrum, STLC provides a dedicated, structured framework solely for evaluating, which aligns more closely with our project objectives.

It allows us to focus on requirement analysis, evaluation planning, evaluation case development, evaluation environment setup, evaluation execution, and evaluation closure in a clear and organised sequence.

This ensures that each operating system configuration is thoroughly evaluated before moving forward, reducing the risk of undetected issues. By using STLC as our core methodology, we can maintain a high standard of accuracy and reliability throughout the project, while still allowing room for iteration and refinement within the evaluating phases.

**Roles and Responsibilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Name** | **Organisation/**  **Position** | **Contact Information** | **Sign** |
| System Architect | **Nathan Quai Hoi** | System Architect | wgk6332@autuni.ac.nz | **NQH** |
| System Architect | **Win Phyo** | System Architect | ddk8093@autuni.ac.nz |  |
| Network Engineer | **Zafar Azad** | Network Engineer | ftk8708@autuni.ac.nz | **ZA** |
| Project Manager | **Thomas Robinson** | Project Manager | cgr2690@autuni.ac.nz | P2030C25T19#y1 |
| Network Engineer | **Larissa Goh** | Network Engineer | xhm5236@autuni.ac.nz |  |
| Network Engineer | **Charmi Patel** | Network Engineer | fhv9735@autuni.ac.nz |  |
| Network Engineer | **Kylie Afable** | Network Engineer | cjq7738@autuni.ac.nz | **KA** |
| Project Mentor | **Daniel Vaipulu** | Project Mentor | daniel.vaipulu@aut.ac.nz |  |
| Client | **Raymond Lutui** | Client | raymond.lutui@aut.ac.nz |  |

## Appendix N – Team Contract

**Project Name: Network Performance Evaluation on Linux Based Operating Systems**

**Project Team Member Names and Sign-off:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Date** | **Sign-off on Team Contract** |
| **Nathan Quai Hoi** | **18/03/2025** |  |
| **Win Phyo** | **18/03/2025** |  |
| **Zafar Azad** | **18/03/2025** | **ZA** |
| **Thomas Robinson** | **18/03/2025** | P2088C15T20#y1 |
| **Larissa Goh** | **18/03/2025** |  |
| **Charmi Patel** | **18/03/2025** |  |
| **Kylie Afable** | **20/03/2025** | **KA** |

**Code of Conduct:** As a project team, we will:

* Work proactively, anticipating potential problems and working to prevent them
* Keep other team members informed of information related to the project
* Focus on what is best for the whole project team
* See the team project through to completion

**Code of Ethics:**

* Maintain integrity, transparency, and accountability in all actions.
* Respect and value diversity, fostering an inclusive team environment.
* Commit to excellence and collaborative efforts for project success.

**Participation:**

* Be honest and open during all project activities
* Encourage diversity in teamwork
* Provide the opportunity for equal participation
* Be open to new approaches and consider new ideas
* Have one discussion at a time
* Let the team know well in advance if a team member must miss a meeting or may have trouble meeting a deadline for a given task

**Communication:**

* Decide as a team on the best way to communicate various information
* Focus on solving problems, not blaming people
* Present ideas clearly and concisely
* Meet and communicate frequently to discuss project progress
* Arrange additional meetings as needed
* Keep discussions on track
* Honor meeting timeframes
* Read communications (emails, meeting minutes, action items, etc.) from each other
* Respond to each other in a timely manner (i.e., within 24 hours)

**Problem Solving:**

* Encourage everyone to participate in solving problems
* Only use constructive criticism
* Hold each other accountable for meeting the standards
* Consequences for behaviours impacting Team performance

**Meeting Guidelines:**

* It is mandatory to attend meetings every Thursday with the supervisor/client.
* Any additional meeting date and time must be scheduled in advance.
* Create and distribute an agenda before each meeting.
* Set expectations for meeting schedules and respect allocated time for each agenda item.
* Assign responsibilities for documenting meeting minutes, tracking action items, and following up on tasks between meetings.
* Encourage feedback on meeting processes and suggest periodic reviews to assess the effectiveness of meetings and adjust guidelines as needed.

**Consequences for behaviour impacting Team performance:**

* Tasks will not be accomplished on time due to constant absence.
* Poor communication among team members can impact work efficiency.
* Project will not meet the deadline if individual tasks are not completed on time.
* Teams cannot work effectively if team members do not respect other team members or constantly have conflicts.

## Appendix O – Client Contract

**Network Performance Evaluation of Linux Based Operating Systems (Physical Team)**

**Contract/Service Agreement**

**Author:** Thomas Robinson

**Date:** 31/03/2025

**Title of Work:** Network Performance Evaluation of Linux Based Operating Systems

This is an Agreement made as of 31/03/2025 by Thomas Robinson (on behalf of the “Project Team”), and Dr. Raymond Lutui (the “Client”).

The Project Team consists of: Thomas Robinson, Win Phyo, Zafar Azad, Nathan Quai Hoi, Larissa Goh, Charmi Patel, Kylie Afable, and Daniel Vaipulu.

THE PROJECT TEAM AND THE CLIENT AGREE THAT:

1. The Work: The Project Team will create the Work as set forth in Exhibit A hereto. The Client will provide the Project Team with the format and specifications in which each element of the Work is to be submitted. The Project Team agrees to conform to such format and specifications.
2. Delivery of the Work: The Project Team agrees to deliver to the Client the Work in form and content acceptable to the Client on or before the dates outlined in Exhibit B of this Agreement, time being of the essence to the Client.
3. Right to Terminate: If the Project Team materially departs from the agreed-upon schedule or if the Work is not satisfactory to the Client (based on reviews of drafts, market conditions, and/or other criteria as determined by the Client), the Client may at its option:
4. Allow the Project Team to finish, correct, or improve the Work by a date specified by the Client;
5. Terminate this Agreement by giving written notice to the Project Team.
6. Payments: The Client will pay the Project Team $1 upon accepted completion of the Work.
7. Exhibit: The following Exhibit is hereby incorporated by reference into this Agreement:

Exhibit A: Statement of Work

The Team will evaluate network performance on three Linux-based operating systems. The Team will deliver a comprehensive performance evaluation package including complete evaluation logs, statistical analysis of performance metrics, comparative analysis across all evaluated operating systems, and a final report detailing findings and methodology.   
Success includes comprehensive documentation of methodology, complete raw data logs, and a final analysis comparing performance across all evaluated configurations.

Exhibit B: Schedule

Full Project: 03/03/2025 – 31/10/2025

Requirement Analysis: 03/03/2025 – 11/04/2025  
Evaluation Planning: 14/04/2025 – 02/05/2025  
Evaluation Case Development: 02/05/2025 – 15/08/2025

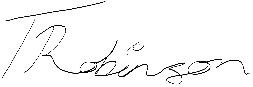
Evaluation Environment Setup: 21/07/2025 – 28/09/2025  
Evaluation Execution: 07/07/2025 – 10/10/2025

Evaluation Closure: 13/10/2025 – 31/10/2025

IN WITNESS WHEREOF, THE PARTIES HERETO HAVE EXECUTED THIS Agreement as a sealed instrument as of the date first above written.

Client Project Team

By: Raymond Lutui Thomas Robinson

Date: 08/04/2025 31/03/2025

P2192#y1Signature:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Appendix P – Risk Register | | | | | | | | | | | |
| **Prepared by: Larissa Goh** | | |  |  | **Date: 19/03** |  |  |  |  |  |
| **No.** | **Rank** | **Risk** | **Description** | **Category** | **Root Cause** | **Triggers** | **Mitigating Action** | **Risk Owner** | **Probability** | **Impact** |
| 1 | Medium | Long distance between members | The huge distances between majority of the team may create challenges when organising meetups. | Logistical | Members located in different regions | Meetup planning, travel expenses, travel time, unreliable transport | Advanced weekly schedules in place, online meetings scheduled where needed | Project Manager | Medium | High |
| 2 | Medium | Varying levels of technical knowledge among members | All members have varying levels of technical skill required for this project. This disparity may create workload imbalances and affect team cohesion. | Training & Knowledge | Different backgrounds and technical experience | Imbalanced task completion. Reliance on members | Ongoing commitment to upskilling. Creating a skill matrix and detailed upskilling plan. | Team | High | High |
| 3 | High | Unsuitable operating system | Selecting an unsuitable operating system for our project. This may lead to major issues during our evaluation phase, cause compatibility issues, or have performance limitations. | Technical | Lack of research | software and hardware incompatibility | In depth research and assessing project requirements. Comparing OS options | System Architect | Medium | High |
| 4 | High | Group member quitting | Group member potentially quitting during any stage of the project. | Team Members | Personal reasons | Personal reasons | Urgent meeting to discuss changed task delegations and effect on project timeline. | Project Manager | Medium | High | |
| 5 | Low | Poor communication | Miscommunication or lack of clarity among group members | Team Members | Missed updates, unclear expectations, lack of structure | Low morale, unclear leadership, confusing task interpretation, failing to include others or themselves | Create a team contract, encourage communication, weekly high priority meetings that requires all members to share ideas and speak. | Project Manager | Low | High | |
| 6 | Medium | File loss & Data corruption | Important data is lost or corrupted due to hardware issues or human error. | Data | Poor backup strategy, carelessness. | System crashes, storage issues, overwriting files | Implement regular backup on cloud or local drives. Use version control such as Git. | Network Engineer | Low | Medium | |
| 7 | Medium | Sickness | Team member, supervisor, and client may be unavailable due to sickness, especially going into colder months. | Health & Safety | COVID-19, infection | COVID and sickness increasing. | Enable remote collaboration is available when needed e.g. Discord, Teams. Flexibility to reschedule to suit client when needed. | Project Manager | High | Medium | |
| 8 | High | D-ITG Compatibility Issues | D-ITG may not function correctly or be supported on all selected operating systems. | Technical | Software incompatibility | Errors during installation | Early evaluation on each operating system | Network Engineer | High | Medium | |
| 9 | High | Hardware Issues | Failure of critical hardware such as computers and network cards can delay evaluation. | Technical | Improper set up or handling, physical damage, firmware issues, | Loss of connectivity, overheating, inability to boot operating system | Perform early hardware evaluation, document all hardware used for quick replacement, spare equipment, in contact with IT technician | System Architect | Medium | High | |
| 10 | Low | Incomplete evaluation due to time constraints | Large number of evaluation combinations may cause the team to run out of time. | Schedule | Time-consuming evaluation and set up | Delay in router configuration, tool issues | Ensure all tools and configurations are fully evaluated during analysis phase and divide responsibilities. | Project Manager | Low | High | |
| 11 | Medium | Inconsistent evaluation results | Differing results between operating systems | Technical | Misconfigured routers, hardware issues | Differing results for identical evaluations | Standardise configuration steps, peer review setup. | System Architect | Medium | High | |
| 12 | Low | Misinterpretation of data | Team may incorrectly analyse performance data due to unfamiliarity with tools, software, or hardware | Technical | Lack of network experience | Unexpected output formats, inconsistent logs | Research tool output structures, consult supervisor if unclear, align on analysis methods as a team | Network Engineer | Low | Medium | |

## Appendix Q – Risk Management Plan

Our project follows the Software Testing Life Cycle (STLC) methodology, where risk and issue management are handled through structured planning and continuous review. Risks are identified and documented during the planning phase through past experiences, client input, and technical research.

This involves reviewing:

* The project scope
* Objectives
* Timeline
* Tools
* Costs
* Team structure

However, as conditions change throughout a project, especially during the analysis and development phases, the team must be ready to adapt quicky. To manage this, risks will be monitored regularly and escalated to active issues if they occur.

* + - 1. Identify risk
      2. Access risk
         1. Likelihood
         2. Impact
      3. Record in risk register
      4. Assign owner and plan mitigation
      5. Monitor during weekly meetings

**If a risk has occurred…**

* + - 1. Move to issue log
      2. Assign owner and track resolution
      3. Resolve or escalate as needed
      4. Continue monitoring risks

The team acknowledges that not all problems can be predicted. Unexpected problems that arise will be added directly into the Issue Log. This ensures both anticipated and unexpected challenges are managed effectively.

## Appendix R – Issue Log

**Prepared by: Larissa** **Date: 19/03/2025**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Issue #** | **Issue Description** | **Impact on Project** | **Date Reported** | **Reported By** | **Assigned To** | **Priority M/H/L** | **Due Date** | **Status** | **Comments** |
| 01 | Delayed meeting with client and supervisor | Delayed start on entirety of project as we need more information and details from our client. | 17/03/25 | Zafar | Group | H | NA | Closed | First meeting with client has been pushed to 20/03/25 |
| 02 | D-ITG isn’t available in Fedora repos; it won’t be compiled from source | If the source code can’t be fixed, another tool will have to be chosen for evaluation, despite the client’s request. | 25/03/25 | Thomas | Thomas | H | 27/03/25 | Closed | Client isn’t aware of this issue yet – it needs to be raised at 27/03 meeting |
| 03 | Cancelled team meeting | Due to multiple members facing unforeseen transport issues, our team meeting before our meeting with client and supervisor had to be cancelled, limiting preparation time | 27/03/25 | Larissa | Thomas | H | 29/03/25 | Closed | Quick overview over Discord and in-depth debrief after to ensure everyone is on the same page |
| 04 | Management methodology didn't align with our project | Team roles and documentation process needed to be reviewed to indicate which protocol we will be following | 27/03/35 | Zafar | Group | H | 29/03/25 | Closed | A hybrid model approach has been selected to better suit the project |
| 05 | Team members absent due to illness and unforeseen circumstances | Two members were unable to attend the scheduled meeting with the client and supervisor, which affected team presence and contribution. | 03/04/25 | Larissa | Group | M | 03/04/25 | Closed | Presentation tasks were reassigned, and absent members were updated via Discord to ensure they were informed of all meeting outcomes. |
| 06 | Methodology still does not align with our project scope | Feedback from our client and supervisor indicated that our hybrid methodology was still not suitable for our project. | 03/04/25 | Larissa | Group | H | 06/04/25 | Closed | The team adopted the STLC methodology and updated all necessary documentation. |

## Appendix S – Change Management Plan

**Date: 02/04/2025**

**Author: Win Phyo**

**Version: 1.0**

**Project Name: Network Performance Evaluation of Linux Based Operating Systems in a Physical Environment**

**Background**

Be sure to plan for change at the beginning of the project. All projects will require some sort of changes during their lifetime. Plan by having the process already defined and clearly documented. Change management is critical to completing your projects on schedule and on budget.

**Introduction**

Change Management is an important part of any project. Changes must be vetted and managed to ensure that they are within the scope of the project and are communicated to all stakeholders if they are approved. The process for submitting, reviewing, and approving changes must also be communicated to all stakeholders to properly set expectations. If changes are allowed to be submitted or are implemented in and unorganised way, any project is sure to fail. All projects must include a Change Management Plan as part of the overall Project Plan, it can either be included as a section in the Project Plan or as an appendix as a subsidiary management plan.

The Change Management Plan sets expectations on how the approach to changes will be managed, what defines a change, the purpose and role of the change control board, and the overall change management process. All stakeholders will be expected to submit or request changes to in accordance with this Change Management Plan and all requests and submissions will follow the process detailed herein.

**Change Management Approach**

This section of the Change Management Plan describes the approach the organisation will use for managing change throughout the project. Throughout a project’s lifecycle there may only be a few or many changes submitted. The approach taken to manage these changes must be consistent and repeatable to provide a quality change management plan and process.

The Change Management approach ensures that all proposed changes are defined, reviewed, and agreed upon so they can be properly implemented and communicated to all stakeholders - and importantly that the implications of the change on the scope, budget and timeline are accounted for. This approach will also ensure that only changes within the scope of the project are approved and implemented or are clearly agreed to by those responsible for the timeline and budget for the overall project.

The Change Management approach is not to be confused with the Change Management Process which should be detailed later in this plan. The Change Management approach consists of three areas:

* Ensure changes are within scope and beneficial to the project
* Determine how the change will be implemented
* Manage the change as it is implemented

The Change Management process has been designed to make sure this approach is followed for all changes. By using this approach, the team will prevent unnecessary change from occurring and focus its time and resources only on beneficial changes within the project scope.

**Definitions of Change**

This section of the Change Management Plan defines the different types of changes that may be requested and considered for the project. These changes may include schedule change, budget change, scope change, or project document changes. Most changes will impact at least one of these areas and it is important to consider these impacts and how they will affect the project.

There are several types of changes which may be requested and considered for a project. Depending on the extent and type of proposed changes, changes to project documentation and the communication of these changes will be required to include any approved changes into the project plan and ensure all stakeholders are notified. Types of changes include:

**Scheduling Changes**: changes which will impact the previously approved project schedule. These changes may require fast tracking, crashing, or re-baselining the schedule depending on the significance of the impact.

**Budget Changes**: changes which will impact the approved project budget. These changes may require requesting additional funding, releasing funding which would no longer be required, or adding to project or management reserves. These may require changes to the cost baseline.

**Scope Changes**: changes which are necessary and impact the project’s scope which may be the result of unforeseen requirements which were not initially planned for. These changes may also impact budget and schedule. These changes may require revision to the WBS, the project scope statement, and other project documentation as necessary.

The project manager typically ensures that any approved changes are communicated to the project stakeholders. Additionally, as changes are approved, the project manager will ensure that the changes are captured in the project documentation where necessary. These document updates must then be communicated to the wider project team and potentially the stakeholders as well.

**Change Control Board**

Here the Change Management Plan describes the Change Control Board, the purpose of the board, and the members and their roles on the board. The change control board is the approval authority for all proposed project changes. If a change is not approved by the control board, then it will not be implemented with the project. The size and function of change control boards may vary depending on the organisation, but their purpose and the roles and responsibilities are typically consistent.

The Change Control Board is the approval authority for all proposed change requests pertaining to a Project. The purpose of the Change Control Board is to review all change requests, determine their impacts on the project risk, scope, cost, and schedule, and to approve or deny each change request.

The following chart provides a list of the Change Control Board members for a typical project:

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Role** |
| Raymond Lutui | Client | Client |
| Daniel Vaipulu | Supervisor | Supervisor |
| Thomas Robinson | Project Manager | Team Leader |
| Ramon Lewis | IT Support | IT Support |
| Dr. Ramesh Lal | Lecturer | Lecturer |
| Prof. Catherine Shi | Lecturer | Lecturer |
| Olivia Tang | Lecturer | Lecturer |

As change requests are submitted using a Change Request Form to the Project Manager by the project team or via stakeholders, the Project Manager will log the requests in a Change Log and the Change Control Board will convene frequently to review all change requests. For a change request to be approved, all Change Control Board members must vote in favour. In the event more information is needed for a particular change request, the request will be deferred and sent back to the requestor for more information or clarification. If a change is deemed critical, an ad hoc Change Control Board meeting may be called to review the change prior to the next scheduled Change Control Board meeting.

**Roles and Responsibilities**

This section of the Change Management Plan describes the roles and responsibilities of project team members regarding the change management process. It is important that everyone understands these roles and responsibilities as they work through the change management process. These roles and responsibilities must be communicated as part of the change management plan to all project stakeholders.

The following are roles and responsibilities for all change management efforts related to a typical project:

Project Sponsor:

* Approve all changes to budget/funding allocations
* Approve all changes to schedule baseline
* Approve any changes in project scope
* Chair the Change Control Board

Project Manager:

* Receive and log all change requests from project stakeholders
* Conduct preliminary risk, cost, schedule, scope analysis of change prior to Change Control Board
* Seek clarification from change requestors on any open issues or concerns
* Make documentation revisions/edits as necessary for all approved changes
* Participate on Change Control Board

Project Team / Stakeholders:

* Submit all change requests on standard organisational change request forms
* Provide all applicable information and detail on change request forms
* Be prepared to address questions regarding any submitted change requests
* Provide feedback as necessary on impact of proposed changes

**Change Control Process**

This part of the Change Management Plan should describe the change control process from beginning to end. Typically, a change control process would be an organisational standard and repeatable. This process is the tool which is used to ensure adherence to the organisation’s change management approach. By following all the steps, the project team can successfully incorporate approved changes, communicate the changes, and update project documentation.

The Change Control Process for the project will follow the organisational standard change process for all projects. The project manager has overall responsibility for executing the change management process for each change request.

1. Identify the need for a change (Stakeholders) – Change requestor will submit a completed change request form to the project manager.
2. Log change in the change request register (Project Manager) – The project manager will keep a log of all submitted change requests throughout the project’s lifecycle.
3. Evaluate the change (Project Manager, Team, Requestor) – The project manager will conduct a preliminary analysis on the impact of the change to risk, cost, schedule, and scope and seek clarification from team members and the change requestor.
4. Submit change request to Change Control Board (Project Manager) – The project manager will submit the change request, as well as the preliminary analysis, to the Change Control Board for review.
5. Obtain Decision on change request (Change Control Board) – The Change Control Board will discuss the proposed change and decide whether it will be approved based on all submitted information.
6. Implement change (Project Manager) – If a change is approved by the Change Control Board, the project manager will update and re-baseline project documentation as necessary.

## Appendix T – Quality Assurance Plan

**1.0 Introduction**

This QA Plan outlines how Quality Assurance (QA) will be used across all phases of the *Network Performance Evaluation of Linux Based Operating Systems in a Physical Environment* R&D project. The plan details walkthroughs, QA/QC responsibilities, metrics, tools, and templates to maintain confidence in our results.

**2.0 QA Walkthrough Procedures by Project Phase**

|  |  |
| --- | --- |
| **Phase** | **Walkthrough Procedure** |
| **Requirement Analysis** | |  | | --- | | The team reviews all project documentation (proposal, WBS, risk register, etc.) to ensure that scope, evaluation metrics (throughput, delay, jitter, loss), and deliverables are clearly defined and feasible. |  |  | | --- | |  | |
| **Evaluation Planning** | The team confirms tool compatibility (D-ITG/iPerf), defines evaluation scenarios and environments, and ensures upskilling align with planning needs. Feedback from the client and supervisor is used to validate planning outcomes. |
| **Evaluation Case Development** | Evaluation cases are reviewed against project requirements. Evaluation scripts are peer reviewed, and evaluation cases are validated with the client. Templates for logs, bug reports, and scenario documentation are also prepared during this phase. |
| **Evaluation Environment Setup** | The team sets up the evaluation hardware and software, verifies router configurations and network topology, and ensures a consistent environment for each operating system. Setup is reviewed before any execution begins. |
| **Evaluation Execution** | Each scenario (OS, 12x packet sizes, 2x IP versions, 2x protocols) is executed 10 times. Results are logged and reviewed daily. Evaluations outside the 95% confidence range are re-run. Issues and bugs are tracked, and regular peer review ensures consistency. |
| **Evaluation Closure** | Results and analysis are verified and compared across all operating systems. Reports, graphics, and the final poster are created. Closure documentation is reviewed and submitted with feedback from the client and supervisor. |

**3.0 Quality Assurance Metrics**

|  |  |
| --- | --- |
| **Metric** | **Description** |
| **Evaluation completion rate** | % of all planned evaluations completed |
| **Re-run frequency** | % of evaluations outside 95% confidence interval |
| **Setup bug count** | Number of environment/configuration-related issues encountered per OS |
| **Execution bug count** | Number of issues during or after evaluation runs |

**4.0 Quality Control Activities**

**Unit Evaluation**

* Verify NIC configurations and IP addresses.
* Ensure IP forwarding is enabled and functioning on routers.
* Confirm D-ITG and iPerf tools run properly on each OS.

**Integration Evaluation**

* Evaluate full packet flow across sender → router 1 → router 2 → receiver
* Validate correct routing and subnet communication between networks.

**System Evaluation**

* Execute full evaluation scenarios with 12 packet sizes per OS.
* Evaluate IPv4 and IPv6 separately across TCP and UDP.
* Review logs and verify data consistency across 10 repeated runs.

**Bug Tracking & Re-evaluating**

* Log bugs and produce daily and weekly bug reports (on each evaluation).
* Apply 95% confidence interval rule and re-run failed scenarios.
* Track and document issue resolution.

**5.0 Templates & Checklists (to be developed in Phase 3)**

|  |  |
| --- | --- |
| Item | Purpose |
| Evaluation case template | Define evaluate setup, packet size, OS, and expected output |
| Router configuration checklist | Ensure consistent dual NIC setup, IP forwarding, routing |
| Evaluation log template | Record evaluation attempt details, logs, tool used, results |
| Bug report form | Track defect info, reproduction steps, screenshots |
| Peer review checklist | Verify evaluation cases/scripts/configs before execution |

**6.0 Review & Update Policy**

This QA plan is a living document. It will be reviewed:

* After Evaluation Planning (Phase 2)
* After first full OS evaluation
* After Evaluation Closure (Phase 6)

These three review points were chosen to match key moments in the project: once planning is complete, after running the first full evaluation, and at the very end of the project. They give the team a chance to reflect on what’s working, adjust the QA approach as needed, and improve the process as we go. All updates will be version-controlled and documented.

**7.0 Ensuring Quality**

The following quality attributes are prioritised in this project:

|  |  |
| --- | --- |
| Quality Attribute | Applied in project |
| Understandability | Evaluate plans, logs, and scripts are clearly documented and reviewed by all team members. |
| Correctness | 95% confidence threshold and re-run policy ensure that only accurate data is accepted. |
| Testability | Scenarios are repeatable across 3 OSs × 2 protocols × 2 IP versions × 12 packet sizes. |
| Learnability | Clear direction, training plan, and step-by-step evaluation to improve accessibility for team members. |
| Usability | Evaluation tools (iPerf/D-ITG) are user-friendly for automated evaluating. |
| Reliability | Same hardware/software setup used per OS to ensure consistent performance evaluation. |
| Portability | Evaluation cases are designed to run identically across Ubuntu, Fedora, and Kali. |
| Efficiency | Scripts automate evaluations and logging, reducing manual effort and execution time. |
| Maintainability | Evaluation scripts and configuration files are version-controlled and peer reviewed. |
| Flexibility | Fallback evaluating tools (iPerf if D-ITG fails) are pre-planned. |

# References

Atlassian. (n.d.). *Waterfall Methodology: A Comprehensive Guide*. <https://www.atlassian.com/agile/project-management/waterfall-methodology>

Botta, A., de Donato, W., Dainotti, A., Avallone, S., & Pescapé, A. (2013, October 28). *D-ITG 2.8.1 Manual*. COMICS Group. <https://traffic.comics.unina.it/software/ITG/manual/>

Chandra, D. G., Kathing, M., & Kumar, D. P. (2013). A Comparative Study on IPv4 and IPv6. *2013 International Conference on Communication Systems and Network Technologies*, 286-289. <http://doi.org/10.1109/CSNT.2013.67>

CORE Technology Brokers. (n.d.). *HP Z420 Tower Workstation*. <https://shop.core.net.nz/products/hp-z420-tower-workstation>

Dang, D. (2024, May 31). *A comprehensive guide to Software Testing Life Cycle (STLC)*. Relia Software. <https://reliasoftware.com/blog/software-testing-life-cycle>

Gamess, E., & Velásquez, K. (2008). *IPv4 and IPv6 Forwarding Performance Evaluation on Different Operating Systems*. CLEI. <https://clei.org/proceedings_data/CLEI2008/Anales/pdf/CLEI/CLEI_2008_118.pdf>

Just Academy. (2024, May 2). *Software testing advantages and disadvantages*. Just Academy. <https://www.justacademy.co/blog-detail/software-testing-advantages-and-disadvantages>

Katalon. (n.d.). *Software Testing Life Cycle: A Definitive Guide.* Katalon. <https://katalon.com/resources-center/blog/software-testing-life-cycle>

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

Narayan, S., Sodhi, S. S., Lutui, P. R., & Vijayakumar, K. J. (2010). Network performance evaluation of routers in IPv4/IPv6 environment. *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, *2*, 707–710. <https://doi.org/10.1109/WCINS.2010.5541871>

PayScale. (2024, May 21). *Average Network Engineer Salary in New Zealand*. <https://www.payscale.com/research/NZ/Job=Network_Engineer/Salary>

PayScale. (2025, January 05). *Average Project Manager, Information Technology (IT) with Project Management Skills Salary in New Zealand*. <https://www.payscale.com/research/NZ/Job=Project_Manager%2C_Information_Technology_(IT)/Salary/ac095581/Project-Management>

PayScale. (2023, October 19). *Average Systems Architect Salary in New Zealand*. <https://www.payscale.com/research/NZ/Job=Systems_Architect/Salary>

PB Technologies. (n.d.). *ASUS Vivobook Go 15 L510KA 15.6" FHD Intel Pentium Silver N6000 - 8GB... ( L510KA-EJ599W )*. <https://www.pbtech.co.nz/product/NBKASU510599/ASUS-Vivobook-Go-15-L510KA-156-FHD-Intel-Pentium-S>

PB Technologies. (n.d.). *Philips 243V7QJAB/79 24" FHD Monitor*. <https://www.pbtech.co.nz/product/MONPHS2438/Philips-243V7QJAB79-24-FHD-Monitor-1920x1080---IPS>

PB Technologies. (n.d.). *A4Tech Fstyler F1010 Multimedia Keyboard & Mouse Combo USB Wired -... ( F1010 )*. <https://www.pbtech.co.nz/product/KEYA4T1001/A4Tech-Fstyler-F1010-Multimedia-Keyboard--Mouse-Co>

PB Technologies. (n.d.). *Cruxtec 0.5m Cat7 Ethernet Cable - Black Color -- 10Gb / SFTP Triple... ( RS7-005-BK )*. <https://www.pbtech.co.nz/product/CABCXT920050/Cruxtec-05m-Cat7-Ethernet-Cable---Black-Color----1>

PB Technologies. (n.d.). *TP-Link TG-3468 32-bit Gigabit PCIe Network Adapter, Realtek RTL8168B... ( TG-3468 )*. <https://www.pbtech.co.nz/product/NETTPL3468/TP-Link-TG-3468-32-bit-Gigabit-PCIe-Network-Adapte>

Quintero, A., Sans, F., & Gamess, E. (2016). Performance evaluation of IPv4/IPv6 transition mechanisms. *International Journal of Computer Network and Information Security*, *8*(2), 1–14. <https://doi.org/10.5815/ijcnis.2016.02.01>

Taffer, M. (2024, December 9). *Lean Project Management: Definition, Principles, & Drawbacks*. The Digital Project Manager. <https://thedigitalprojectmanager.com/projects/pm-methodology/lean-project-management/>

Testim. (2025, February 3). *What is the Software Testing Life Cycle? A Complete Guide*. Testim. <https://www.testim.io/blog/software-testing-life-cycle/>