Smart Computer Program for Network Validation & Optimization

Final Report-Cisco Virtual Internship

Candidate Name: P Chandra Hasini

Institute: KL University

# Abstract

This project presents a Smart Computer Program designed for automated network validation and optimization. The system parses networking device configurations, generates topologies, validates configurations, analyzes network loads, and provides recommendations for optimization. Through integration with Cisco Packet Tracer, the solution enables network simulation, fault injection, and performance validation. Results demonstrate its ability to detect loops, overloaded links, and recommend alternate routing paths, thereby improving network reliability and efficiency.

# Acknowledgement

We would like to express our sincere gratitude to Cisco for providing the problem statement and guidance throughout the project. We also extend our thanks to our institute, faculty mentors, and peers who supported us during the development of this solution.

# Table of Contents

1. Abstract  
2. Acknowledgement  
3. Introduction  
4. Problem Statement  
5. Objectives  
6. Proposed Solution  
7. System Design & Architecture  
8. Methodology  
9. Implementation & Results  
10. Results & Discussion  
11. Conclusion  
12. Limitations  
13. Future Scope  
14. References

# Introduction

Modern enterprise networks are becoming increasingly complex, requiring automated validation and optimization tools to ensure smooth operations. Manual validation is time-consuming and error-prone, which can result in misconfigurations, network loops, and congestion. This project addresses these challenges by providing a smart solution that integrates configuration parsing, topology generation, validation, load analysis, and simulation.

# Problem Statement

The official Cisco problem statement outlines the need for a system capable of parsing network device configurations, generating topologies, validating configurations, analyzing loads, and suggesting optimizations. The solution must be scalable, reliable, and efficient, ensuring better fault tolerance and resource utilization.

# Objectives

- Automate the generation of network topologies from configuration files.  
- Validate network configurations and detect issues like duplicate IPs, VLAN mismatches, MTU mismatches, and loops.  
- Analyze bandwidth usage and suggest load balancing paths.  
- Simulate network performance and resilience using Cisco Packet Tracer.  
- Provide a scalable foundation for future network optimization.

# Proposed Solution

The solution was divided into four main parts:  
  
Part 1: Network Topology Generation  
Objective: Automatically generate network topology from configuration files.  
Approach: Parse device configs, build JSON, and generate graph-based topology. Topology is also implemented in Cisco Packet Tracer (.pkt).

A diagram of a network

AI-generated content may be incorrect.  
  
Part 2: Network Performance and Load Management  
Objective: Analyze bandwidth & load, suggest balancing.  
Approach: Bandwidth analyzer detects overloaded links and proposes alternate paths. Helps prevent congestion and optimize utilization.

A screenshot of a computer

AI-generated content may be incorrect.  
  
Part 3: Configuration Validation and Optimization  
Objective: Detect missing configurations and errors.  
Approach: Validation engine checks duplicate IPs, VLAN mismatches, MTU mismatches, and network loops. Optimizations include protocol-level improvements (e.g., OSPF vs BGP).

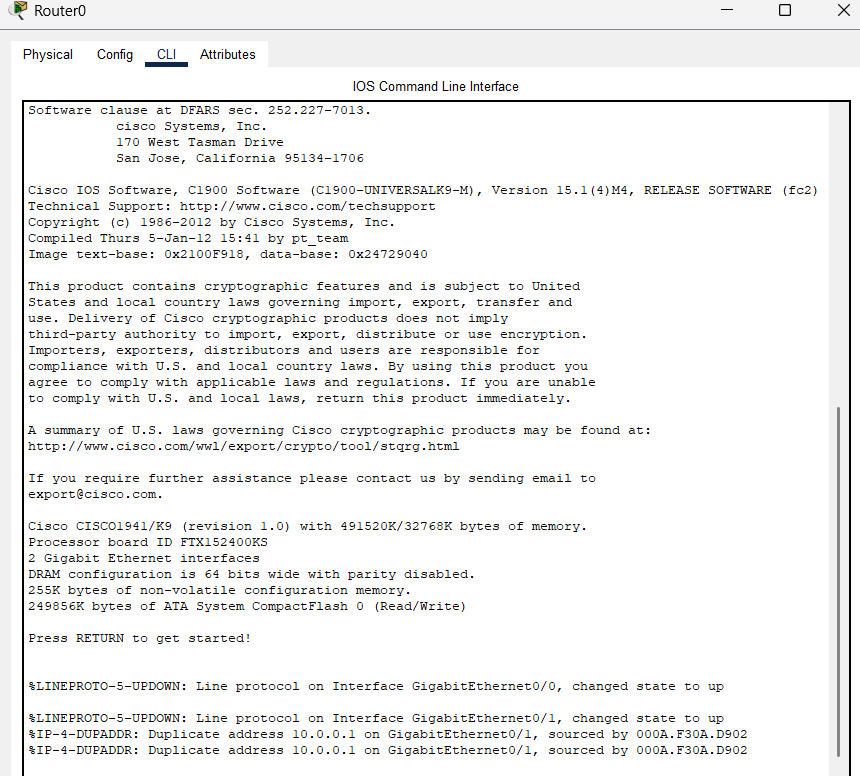


FIG: CLI SCREENSHOT OF ROUTER 0

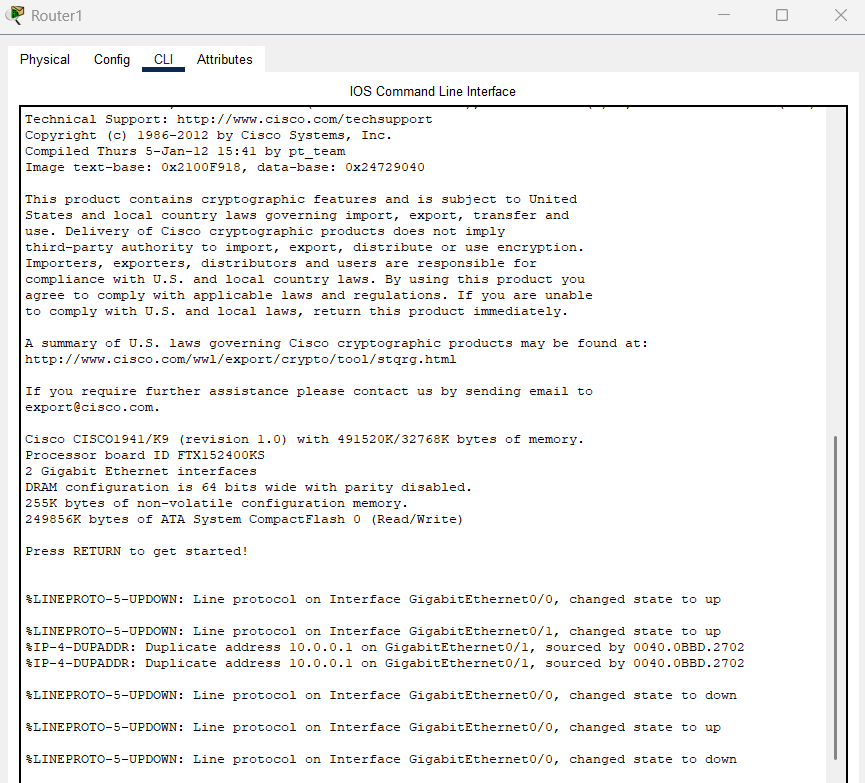


FIG: CLI SCREENSHOT OF ROUTER 1

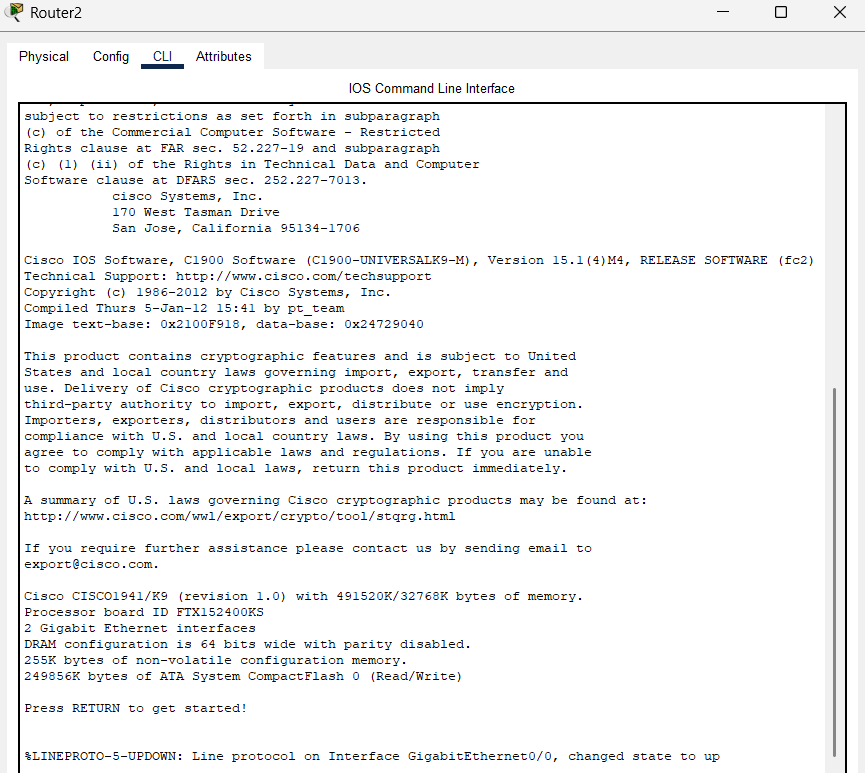


FIG: CLI SCREENSHOT OF ROUTER 2  
  
Part 4: Network Simulation and Fault Injection  
Objective: Simulate network performance and fault scenarios.  
Approach: Packet Tracer is used to simulate failures and validate system resilience. The topology is saved as a .pkt file for Cisco Packet Tracer.

A diagram of a network

AI-generated content may be incorrect.

# System Design & Architecture

# CISCO.PKT FILE IS HERE:

The system is designed with modular components:  
1. Configuration Parser (Conf.zip → JSON)  
2. Topology Builder (NetworkX Graph → Packet Tracer)  
3. Validation Engine (check IPs, VLANs, MTUs, loops)  
4. Load Analyzer (detect overloaded links, suggest alternate paths)  
5. Report Generator (PDF report with validation results)  
6. Simulation (Packet Tracer .pkt for hands-on testing)

# Methodology

The methodology involves parsing configuration files, validating data, analyzing loads, and generating visual representations. Algorithms such as shortest path (Dijkstra) were applied to suggest alternate paths. The validation engine was implemented in Python using NetworkX, while visualization and reporting were done with Matplotlib and ReportLab.

# Implementation & Results

The system was implemented in Python using NetworkX, ReportLab, and Matplotlib. Google Colab was used for development. The tool accepts either Conf.zip or JSON input.  
  
Validation results:  
- ✅ No duplicate IPs found  
- ✅ No VLAN mismatches  
- ✅ No MTU mismatches  
- ⚠ Network loops detected: [['R2', 'S1', 'S2'], ['R2', 'R1', 'S1']]  
- ⚠ Link R3 <-> S2 overloaded: 104 Mbps / 100 Mbps (excess 4 Mbps)  
  
Alternate Path Suggestions:  
- S1 → S2 → R2 with available 58 Mbps capacity.  
  
The final topology has been implemented in Cisco Packet Tracer (.pkt) for practical simulation and testing.

A graph of a network link bandwidth usage

AI-generated content may be incorrect.

A green graph with white text

AI-generated content may be incorrect.

# Results & Discussion

The proposed system successfully automated the validation of configurations and identified key issues. Network loops and overloaded links were accurately detected, and alternate paths were suggested. This demonstrates that the system not only validates but also provides actionable recommendations. Graphs and simulation outputs confirm its effectiveness in improving reliability and performance.

A diagram of a network

AI-generated content may be incorrect.

A close-up of a results

AI-generated content may be incorrect.

A table with numbers and a number of data

AI-generated content may be incorrect.

A grey and white text

AI-generated content may be incorrect.

# Conclusion

The developed system automates the process of validating and analyzing network configurations. It ensures early detection of misconfigurations, detects loops and overloaded links, and suggests alternate routing paths to improve reliability and performance. The solution aligns with Cisco's problem statement and provides a scalable foundation for network optimization.

# Limitations

1. Currently limited to static configuration files (Conf.zip/JSON) without real-time monitoring.  
2. Protocol support is basic (OSPF, BGP not fully integrated).  
3. Automated remediation scripts not yet implemented.  
4. AI/ML predictive analysis remains as future scope.

# Future Scope

1. Integration with real-time SNMP/NetFlow monitoring.  
2. Extending to multi-protocol analysis (OSPF, BGP).  
3. Automated remediation scripts for detected issues.  
4. AI/ML-based predictive analytics for network performance.

# References

- Cisco Networking Problem Statement PDF  
- Networking Stream Solution Draft PDF  
- GeeksforGeeks – Hybrid Topology  
- Cisco Multi-cloud Networking Guide  
- GitHub Repositories for Packet Tracer Labs  
- Python Libraries: NetworkX, Matplotlib, ReportLab