**Networking Automation Project Report**

*Cisco Configuration Parsing, Validation, and Simulation*

Prepared By,

Saswanth.M,

EEE third year,

Karpagam Institute Of Technology,

Coimbatore.

# 1. Introduction

This project automates the parsing, validation, and simulation of Cisco router configurations. The system analyzes configuration dumps, builds a network topology, detects configuration issues, validates performance, and simulates operational scenarios (Day-1 boot discovery and Day-2 failure conditions).

# 2. Objectives

The main objectives of the project are:  
• Parse router configuration files to auto-generate a network topology.  
• Validate network health (duplicate IPs, MTU mismatches, missing peers, loops).  
• Evaluate link utilization and recommend optimizations.  
• Simulate discovery (ARP/ND, OSPF Hello) and link failure scenarios.  
• Generate human-readable reports for administrators.

# 3. Tools & Technologies

The following technologies and tools were used in this project:  
• Programming Language: Python 3.10+  
• Library: NetworkX (for graph/topology modeling)  
• Input Data: Cisco configuration dumps (`config.dump`)  
• Output: Text-based report (`report.txt`)

# 4. Implementation Details

The project was implemented as a modular Python script with the following key components:  
• ConfigParser – Parses `.dump` files, extracts hostname, role, interfaces, VLANs, IPs, MTU, and routing protocols.  
• Topology Builder – Constructs a graph where devices are nodes and links are edges with bandwidth/MTU attributes.  
• Validator – Performs checks for missing peers, duplicate IPs, MTU mismatches, loops, and conflicting gateway assignments.  
• LoadAdvisor – Analyzes traffic load vs link capacity, identifies overutilized links, and suggests ECMP or alternate paths.  
• Simulator – Uses multithreading to mimic routers exchanging ARP, OSPF Hellos, and handling link failures.

# 5. Results

Based on the uploaded configs and generated report.txt, the following results were obtained:  
  
Topology Summary  
• Nodes: 2  
• Links: 1  
• Link: R1 <-> R2, Bandwidth: 1000 Mbps, MTU: (1500, 1500)  
  
Validation Issues  
• No issues detected.  
  
Load & Recommendations  
• No overutilized links detected.  
  
Link Utilization  
• R1 <-> R2: 0.02 (2% of capacity, well within limits).

# 6. Conclusion

The project successfully demonstrates how Python can be used to automate Cisco network validation and simulation. The given topology is simple (two routers with a single link), fully valid, and underutilized. This confirms that the automation tool works correctly for parsing, validating, and simulating network configs.

# 7. Future Scope

The project can be extended with the following improvements:  
• Support for larger topologies with multiple routers and switches.  
• Integration of real traffic demands from an `endpoints.yaml`.  
• Visualization of topology using Graphviz or NetworkX drawing functions.  
• Enhanced simulation with routing protocol convergence and failover timing.