NETWORKING STREAM: TECHNICAL REPORT

OVERVIEW:

This report outlines the development of an auto-topology generation and network simulation's as per the Networking Problem Statement of the Cisco Virtual Internship Program 2025. The automatically parses router configuration files, constructs a network topology, validates setting and simulates performance and failures.

MAIN-PROJECT:

GITHUB -LINK: roshankumar0036singh/CISCO-VIP-NETWORKING-2025

INPUT CONFIGURATION-FILES:

https://drive.google.com/drive/folders/1IpQ6TzIeMt7BoVMD8mypGwCoBCGKeIBG?usp=drive_link

- *Config/R1.txt*   - *Config/S1.txt*   - *Config/PC1.txt*   - *Config/PC4.txt*

- *Config/R2.txt*   - *Config/S2.txt*   - *Config/PC2.txt*   - *Config/PC5.txt*

- *Config/R3.txt*   - *Config/S3.txt*   - *Config/PC3.txt*   - *Config/PC6.txt*

Each file includes interface settings, IP addresses, bandwidth, routing protocols (OSPF/BGP) VLANs, and more.

AUTO TOPOLOGY-GENERATION:

- The system parses config files to extract link relationships, interface details, and bandwidth.

```
PS D:\download> & D:/download/.venv/Scripts/Activate.ps1
(.venv) PS D:\download> python src/main_integration.py
🚀 Cisco Virtual Internship - Complete Network Analysis Tool
===============================================================================
📋 Step 1: Parsing device configurations with comprehensive validation...
   ✅ Parsed 12 configurations
```

- A hierarchical topology is generated connecting routers, switches, and end devices.

```
PS D:\download> & D:/download/.venv/Scripts/Activate.ps1
(.venv) PS D:\download> python src/main_integration.py
🚀 Cisco Virtual Internship - Complete Network Analysis Tool
================================================================================
📄 Step 1: Parsing device configurations with comprehensive validation...
    ✅ Parsed 12 configurations

📊 Step 2: Constructing hierarchical network topology...
    ✅ Built topology: 12 nodes, 24 links
```

- Visual layout auto-generates using extracted metadata.
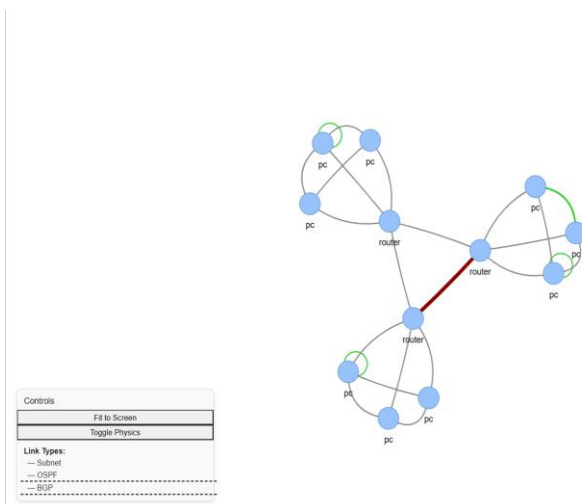
CONFIGURATION VALIDATION & OPTIMIZATION:

The tool checks for:

- Missing configuration files (e.g., a switch config for an endpoint)

- Duplicate IPs in the same subnet

- Incorrect VLAN tags or gateway assignments

- MTU mismatches

- Potential network loops

- Suggestions to replace OSPF with BGP when scalability is needed

```
🔍 Step 3: Running comprehensive network validation...
   🟥 Validation Results:
      ❌ missing_components: 9 issues found
         - PC S1 appears to be missing associated switch configuration
         - PC S2 appears to be missing associated switch configuration
         - PC S3 appears to be missing associated switch configuration
         ... and 6 more
      ✅ duplicate_ips: No issues
      ✅ vlan_issues: No issues
      ✅ gateway_issues: No issues
      ✅ routing_recommendations: No issues
      ✅ mtu_mismatches: No issues
      ❌ network_loops: 2 issues found
         - Potential network loop detected: R3 -> R1 -> R2 -> R3
         - Potential network loop detected: S3 -> R3 -> PC5 -> S3
      ✅ aggregation_opportunities: No issues
```

LOAD-MANAGEMENT & TRAFFIC AWARENESS:

- Parses bandwidth details from configs to estimate capacity.

- Models expected traffic per application type (e.g., video conferencing vs file transfer).

- If a link is overloaded, recommends load balancing or path offloading.

- Provides fallback routing paths for low-priority traffic.

```
🟥 Step 4: Analyzing traffic patterns and capacity...
   🔗 Link Utilization Analysis:
      ⚠️ Link R1-S1 is heavily utilized (100.0%)
      ⚠️ Link R1-PC1 is heavily utilized (100.0%)
      ⚠️ Link R1-PC2 is heavily utilized (100.0%)
   💡 Load Balancing Recommendations:
      - Activate alternative paths for R1-S1 to distribute load. Found 4 alternative routes.
      - Consider implementing ECMP (Equal-Cost Multi-Path) routing for R1-S1
      - URGENT: Implement traffic shaping on R1-S1 to prioritize critical applications
```

STIMULATION & FAULT-INJECTION:

- Day-1 simulation includes ARP, OSPF discovery, and neighbor formation.

- Impact on endpoints

- Routing table reconvergence

- MTU issue effect on data delivery

```
🌅 Step 6: Running Day-1 simulation scenarios...
    🔌 Bringing up all network devices...
✅ All interfaces set to up
    ⏳ Running 60-second network stabilization...
⏳ Waiting 60s for Day 1 network stabilization…
✅ Stabilization complete
    🔍 Populating ARP tables and discovering neighbors...
✅ ARP tables populated
✅ OSPF adjacencies formed: {}
✅ BGP sessions established: {}
✅ Day 1 neighbor validation passed
```

- Day-2 testing includes link failure simulation and behavior analysis:

```
📝 Step 8: Running Day-2 comprehensive testing...
2025-08-15 19:12:08,215 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,215 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,215 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,215 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,216 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,216 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,216 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,217 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,217 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,217 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,217 - day2_testing - INFO - Checking configuration best practices
2025-08-15 19:12:08,217 - day2_testing - INFO - Checking configuration best practices
    📊 Day-2 Test Summary:
       Total tests: 31
       Passed: 85
       Failed: 10
       Warnings: 5
```

```
💥 Step 7: Testing link failure scenarios...
    🔗 Simulating failure: R1 <-> R2
2025-08-15 19:12:04,203 - SimulationEngine - INFO - Link failure injected: R1 <-> R2
       ✅ Network maintained connectivity
2025-08-15 19:12:06,204 - SimulationEngine - INFO - Link restored: R1 <-> R2
    🔧 Restored link: R1 <-> R2
    🔗 Simulating failure: R1 <-> S1
2025-08-15 19:12:06,205 - SimulationEngine - INFO - Link failure injected: R1 <-> S1
       ✅ Network maintained connectivity
2025-08-15 19:12:08,208 - SimulationEngine - INFO - Link restored: R1 <-> S1
    🔧 Restored link: R1 <-> S1
```

- Simulation can be paused, edited, and resumed.

```
⏸️Step 11: Demonstrating pause/resume capabilities...
    ⏸️ Pausing simulation...
2025-08-15 19:12:09,795 - Node-R1 - INFO - Node R1 paused
2025-08-15 19:12:09,795 - Node-R2 - INFO - Node R2 paused
2025-08-15 19:12:09,795 - Node-R3 - INFO - Node R3 paused
2025-08-15 19:12:09,795 - Node-S1 - INFO - Node S1 paused
2025-08-15 19:12:09,795 - Node-S2 - INFO - Node S2 paused
2025-08-15 19:12:09,796 - Node-S3 - INFO - Node S3 paused
2025-08-15 19:12:09,796 - Node-PC1 - INFO - Node PC1 paused
2025-08-15 19:12:09,796 - Node-PC2 - INFO - Node PC2 paused
2025-08-15 19:12:09,796 - Node-PC3 - INFO - Node PC3 paused
2025-08-15 19:12:09,796 - Node-PC4 - INFO - Node PC4 paused
2025-08-15 19:12:09,796 - Node-PC5 - INFO - Node PC5 paused
2025-08-15 19:12:09,796 - Node-PC6 - INFO - Node PC6 paused
2025-08-15 19:12:09,796 - SimulationEngine - INFO - Simulation paused
    ▶️ Resuming simulation...
2025-08-15 19:12:11,799 - Node-R1 - INFO - Node R1 resumed
2025-08-15 19:12:11,800 - Node-R2 - INFO - Node R2 resumed
2025-08-15 19:12:11,800 - Node-R3 - INFO - Node R3 resumed
2025-08-15 19:12:11,800 - Node-S1 - INFO - Node S1 resumed
2025-08-15 19:12:11,801 - Node-S2 - INFO - Node S2 resumed
2025-08-15 19:12:11,801 - Node-S3 - INFO - Node S3 resumed
2025-08-15 19:12:11,801 - Node-PC1 - INFO - Node PC1 resumed
2025-08-15 19:12:11,801 - Node-PC2 - INFO - Node PC2 resumed
2025-08-15 19:12:11,801 - Node-PC3 - INFO - Node PC3 resumed
2025-08-15 19:12:11,802 - Node-PC4 - INFO - Node PC4 resumed
2025-08-15 19:12:11,802 - Node-PC5 - INFO - Node PC5 resumed
2025-08-15 19:12:11,802 - Node-PC6 - INFO - Node PC6 resumed
2025-08-15 19:12:11,802 - SimulationEngine - INFO - Simulation resumed
```

SYSTEM ARCHITECTURE:

- IPC (FIFO/TCP sockets) used to exchange metadata packets.

- Logs maintained per thread to simulate MAC/IP layer activity.

- Each router/switch is represented as a multithreaded object.

```
🔧 Step 5: Initializing multithreaded simulation engine with IPC...
2025-08-15 19:11:04,192 - SimulationEngine - INFO - IPC server listening on port 54024
2025-08-15 19:11:04,193 - SimulationEngine - INFO - Starting network simulation
2025-08-15 19:11:04,193 - Node-R1 - INFO - Node R1 started
2025-08-15 19:11:04,194 - Node-R2 - INFO - Node R2 started
2025-08-15 19:11:04,195 - Node-R3 - INFO - Node R3 started
2025-08-15 19:11:04,195 - Node-S1 - INFO - Node S1 started
2025-08-15 19:11:04,196 - Node-S2 - INFO - Node S2 started
2025-08-15 19:11:04,196 - Node-S3 - INFO - Node S3 started
2025-08-15 19:11:04,196 - Node-PC1 - INFO - Node PC1 started
2025-08-15 19:11:04,197 - Node-PC2 - INFO - Node PC2 started
2025-08-15 19:11:04,197 - Node-PC3 - INFO - Node PC3 started
2025-08-15 19:11:04,197 - Node-PC4 - INFO - Node PC4 started
2025-08-15 19:11:04,197 - Node-PC5 - INFO - Node PC5 started
2025-08-15 19:11:04,198 - Node-PC6 - INFO - Node PC6 started
    ✅ Simulation engine started with IPC capabilities
```

CONCLUSION:

- The project successfully automates network topology generation and simulation, streamlining configuration validation and performance testing. By minimizing manual effort and improving accuracy, it offers a reliable, efficient, and scalable solution for modern network design and troubleshooting.