CS352 - Report

Arun Felix + Bhavya Patel

May 9, 2025

1 Team Details

Arun Felix and Bhavya Patel

2 Collaboration

Bhavya handled the virtual machine environment, since he had an x86 64 machine. I primarily handled the report.

3 Implementation details

To set up the interface configurations, we used the `ip addr add` command with the specified IP addresses for each device interface (e.g., `h1 ip addr add 10.0.0.1 dev h1-eth0`). For the four host interfaces and four router interfaces, we assigned the required IP addresses as per the project specifications. To establish default routes for the hosts, we implemented the `ip route add default dev` command for each host, directing all outbound traffic through their respective interfaces (e.g., `h1 ip route add default dev h1-eth0`). This ensured that hosts could communicate with the router acting as their gateway. Finally, for setting up specific routes on the router, we configured direct routes to each destination host using the `ip route add` command with the `via` parameter to specify the appropriate next-hop (e.g., `r1 ip route add 10.0.0.1 via 10.0.0.2`). This created the necessary forwarding table entries on router r1 to correctly route packets between all hosts in the network topology.

4 Implmentation

To our best of our knowledge, we have implemented the project as per the requirements. We have set up the network topology with 4 hosts and 2 routers. The hosts are connected to the routers, and we have configured the routing tables accordingly. We have also tested the connectivity between the hosts and verified that they can communicate with each other through the routers.

5 Difficulties

We primarily had issues with the Virtual Machine. Bhavya and I had a problem because we both had macbooks, which uses the aarch64 architecture. In the end, Bhavya used his home PC for the VM instead.

6 Technical Observation

While implementing this project, we observed the critical difference between configuring routes with `ip route add default` for hosts versus explicit destination routing on the router. Specifically, we discovered that endpoint hosts require only simple default routes since they have single network interfaces, while routers demand more precise routing table entries with explicit next-hop information. When configuring the router, we noted that the syntax `r1 ip route add 10.0.0.1 via 10.0.0.2` creates a routing entry that instructs the router to forward packets destined for 10.0.0.1 through the interface with IP 10.0.0.2. This demonstrates the network layer principle of next-hop routing, where forwarding decisions are made based on destination IP addresses matched against routing table entries. Furthermore, we observed through successful `traceroute` outputs that TTL decrementing functions correctly at each hop, confirming that our router implementation was properly handling the IP header's TTL field as packets traversed through the network. This reinforces the theoretical concept of hop-by-hop forwarding that prevents packets from circulating indefinitely in routing loops while providing valuable diagnostic information about the network path.