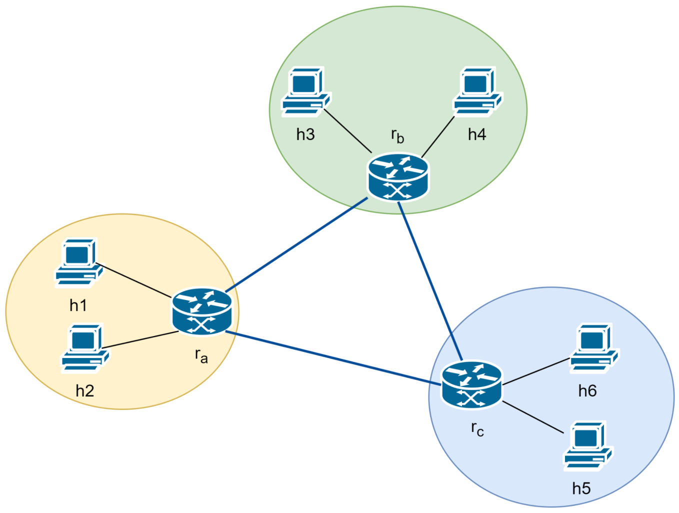
Assignment-02 **CS 612 Computer Systems(2023-24)** **50 points**
Submission Deadline: 16-Nov-2023 11:59 PM.

Instructions:

1. Assignment can be done in pairs. (At most 2 students together, an individual is also fine).
2. All the programs must work on the Linux operating system (Debian/Ubuntu/Kali/CentOS).
3. Submit the zipped file or link to the GitHub repository containing the source code. Include a readme.txt containing the information of the team member and how to run your code.
4. Also, submit a write-up (PDF) containing all the references. Explain the implementation in brief and include your observations with sufficient screenshots.
5. Only1 submission per team.

Part I: Implement the routing functionality in Mininet. (25 points)

1. Implement the below network topology in mininet. Here, the h1, h2, h3, h4, h5, and h6 are hosts, and r_a, r_b, and r_c are the routers connected to each other. You are required to implement the given network topology in mininet (using mininet's python API).



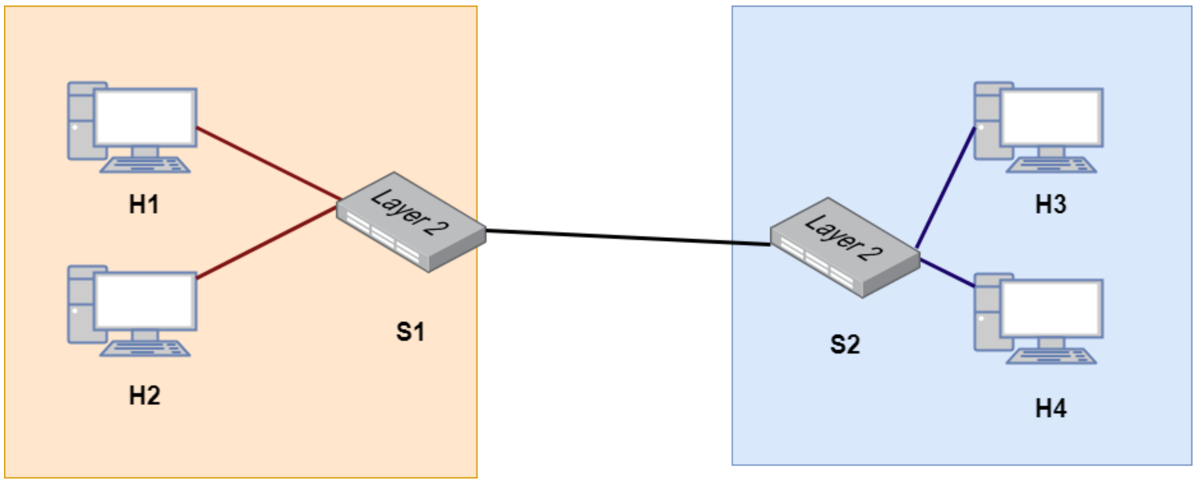
Note: As the mininet doesn't support routers natively, you can implement routers as a combination of a switch and a host. Refer to this mininet example as to how to implement a simple router.

The network is supposed to be divided into three subnets connected by the routers. Thus, you will need to assign the ip addresses to the hosts and routers/switches appropriately, maintaining the isolation.

a. **Implementation** of the custom topology using mininet. ***(10 pts)*** (Every host should be able to send packets to every other host. You are to submit the code, as well as provide screenshots as proof of its working)
b. **Observations**: Capture and show the wireshark/tcpdump (packets) for the route setup on any one of the routers.***(5 pts)***
c. Vary the default routing and measure the latency difference. (i.e. the default route for a packet from h1 to h6 would be h1 -> r_a -> r_c -> h6. Try to vary the path so that it takes the path h1-> r_a -> r_b -> r_c -> h6. Provide the screenshots as proof for the latency difference, both ping as well as iperf) ***(5 pts)***
d. Dump the routing tables for all the routers for both of the above questions. ***(5 pts)***

Part II: Analysis of different TCP congestion control schemes. (25 points)

Create a mininet topology as shown in the diagram below. H1 to H4 are four hosts connected to switches S1 and S2. Run a TCP Server that accepts data in the H4 system. H1, H2, H3 are the TCP clients that connect with the server (H4) and send data packets.



For the congestion control schemes: Vegas, Reno, Cubic, and BBR, identify the throughput (graph) observed. You can use wireshark to identify the throughput.

You can implement a custom client-server program that generates TCP data packets for the required analysis. Ex: a File transfer program, HTTP server or simple load generator tools like iperf.

Throughput analysis can also be done using the iPerf tool. Explore how to set up a client–server for analysis of TCP connections and configuring Congestion control schemes in the official documentation from the iPerf website: https://iperf.fr/.

a. Implementation of the mininet topology and TCP client-server program (for packet generation).

   The TCP client-server program should run with the required configuration for question (b) when `--config=b` and with required configuration for question (c) when `--config=c`

   Other configurable parameters like congestion control scheme, link loss should also be taken as command line arguments to the program. **(5 points)**
b. Run the client on H1 and the server on H4. Report and reason the throughput over time for each congestion control scheme. **(10 points)**
c. Run the client on H1, H2, H3 simultaneously and the server on H4. Report and reason the throughput over time for each host, each congestion control scheme. **(5 points)**
d. Configure the link loss parameter of the middle link (s1 - s2) to 1% and 3% and run the experiment in (b). Report and compare the throughput over time for each congestion control scheme. **(5 points)**