**Melanie Prettyman**
**CS6014 HW4**
**02/2024**

## Question 1: Reliable Data Transfer

### Sender S State Machine:

- State: WAIT_FOR_CALL_0_FROM_ABOVE (application layer)
    - After receiving data from the application layer in state 0:
        - **udt_send(data, seq_num=0)** to R1 and R2.
        - Start timer
        - Transition to state WAIT_FOR_ACK_0.

- State: WAIT_FOR_ACK_0
    - After receiving ACK0 from both R1 and R2:
        - Stop timer
        - Transition to WAIT_FOR_CALL_1_FROM_ABOVE
    - On timer timeout:
        - **udt_send(data, seq_num=0)** to R1 and R2.
        - Restart timer.

- State: WAIT_FOR_CALL_1_FROM_ABOVE (application layer)
    - After receiving data from the application layer when in state 1:
        - **udt_send(data, seq_num=1)** to R1 and R2.
        - Start timer.
        - Transition to state WAIT_FOR_ACK_1.

- State: WAIT_FOR_ACK_1
    - After receiving ACK1 from both R1 and R2:
        - Stop timer.
        - Transition to WAIT_FOR_CALL_0_FROM_ABOVE (application layer).
    - On timer timeout:
        - udt_send(data, seq_num=1) to R1 and R2.
        - Restart timer

### R (Receiver, applies to both R1 and R2) Protocol State Machine

- State: WAIT_FOR_0
    - On receiving a packet with seq_num=0:
        - If packet is not corrupted:
            - Deliver data to the application layer.
            - **udt_send(ACK0).**

                ■ Transition to WAIT_FOR_1.

- State: WAIT_FOR_1
    - On receiving a packet with seq_num=1:
        - If packet is not corrupted:
            - Deliver data to the application layer.
            - **udt_send(ACK1).**
    - Transition to WAIT_FOR_0.

- Sender S sends data packets with sequence numbers (0 or 1) to both receivers R1 and R2.
- Receivers R1 and R2 acknowledge receipt of the data packet with the corresponding sequence number.
- If an ACK is not received within a timeout period, the sender resends the data packet.
- The sender transitions between states based on receiving ACKs or timeouts.
- The receivers wait for the expected sequence number and send ACKs. They switch between waiting for sequence number 0 and 1 upon receiving the correct data packet.

## Question 2: Throttling

Flow control and congestion control are both mechanisms used in protocols like TCP, to manage the transmission of data.

Flow control is used to prevent a sender from overwhelming a receiver with data ( which would prevent the receiver from being able to process the data). It maintains a balance between the rate data is sent by the sender and the rate it can be processed by the receiver.

In TCP, flow control is implemented with a sliding window mechanism. The receiver informs its receive window size (amount of data that the receiver is willing to accept) to the sender in the TCP header fields. The sender then adjusts its transmission rate based on this window size, thus preventing it from sending more data than the receiver can handle.

Congestion control deals with the overall network congestion, which happens when there is more demand for network resources (bandwidth, buffer space) than the network can handle. Congestion can lead to packet loss, increased latency, and decreased network throughput.

TCP implements congestion control through algorithms such as TCP Tahoe, TCP Reno, and TCP New Reno. These algorithms use mechanisms such as slow start, congestion avoidance, and fast retransmit to control the rate data is transmitted into the network. TCP monitors the network conditions by observing packet loss and round-trip time and adjusts its sending rate to avoid congestion.

## Question 3: NAT

When communication occurs between hosts behind a NAT router and an external host, the NAT router performs address translation to allow communication between the internal hosts and the external host. Here are the possible values for source and destination addresses and ports for packets in each scenario:

- From A to X behind the NAT:
    - Source IP: 10.0.0.1
    - Source port: random high port assigned by NAT
    - Destination IP: 1.2.3.4
    - Destination port: 80

- From B to X behind the NAT:
    - Source IP: 10.0.0.2
    - Source port: random high port assigned by NAT
    - Destination IP: 1.2.3.4
    - Destination port: 80

- From A to X between the NAT and X:
    - Source IP: 5.6.7.8 (NAT translates the internal private IP addresses and ports to its public IP address with unique ports)
    - Source port: random high port assigned by NAT
    - Destination IP: 1.2.3.4
    - Destination port: 80

- From B to X between the NAT and X:
    - Source IP: 5.6.7.8
    - Source port: random high port assigned by NAT
    - Destination IP: 1.2.3.4
    - Destination port: 80

- From X to A between X and the NAT:
    - Source IP: 1.2.3.4
    - Source port: 80
    - Destination IP: 5.6.7.8
    - Destination port: port used by A in the NAT translation table

- From X to A between the NAT and A:
    - Source IP: 1.2.3.4
    - Source port: 80
    - Destination IP: 10.0.0.1
    - Destination port: port used by A in the NAT translation table

| Direction | Internal IP:Port | External IP:Port | Description |
|---|---|---|---|
| Outgoing | 10.0.0.1:1024 | 5.6.7.8:50001 | Packets from A to X |
| Outgoing | 10.0.0.2:2025 | 5.6.7.8:50002 | Packets from B to X |
| Incoming | 5.6.7.8:50001 | 10.0.0.1:1024 | Packets to A |
| Incoming | 5.6.7.8:50002 | 10.0.0.2:2025 | Packets to B |

## Question 4: Routers

**Part 1: Subnets and their smallest IP prefix:**

There are 6 total subnets

1. Group A Subnet: 1.1.1.0/24
2. Group B Subnet: 1.1.2.0/24
3. Group C Subnet: 1.1.3.0/24
4. A-B Link Subnet: Assuming the description means a point-to-point link between A and B, typically this would use a /30 subnet to minimize address waste, thus: 1.1.4.0/30
5. A-C Link Subnet: Similarly, for the A-C link: 1.1.5.0/30
6. B-C Link Subnet: And for the B-C link: 1.1.6.0/30

Groups A, B, C already has the smallest prefix since they are /24 (supports 254 host addresses minus 2 reserved addresses for the network and broadcast addresses).
Each link subnet (A-B, A-C, B-C) has a minimum of /30, which is the smallest for supporting 1-1 connection of 2 host addresses.

**Part 2: Cheapest IP prefix for internet connection:**

They would need to combine all subnets into a single prefix that can cover all in a larger subnet that starts at 1.1.1.0 and goes up to at least 1.1.6.3 .The smallest prefix that can cover this range is 1.1.0.0/21 (covers addresses from 1.1.0.0 to 1.1.7.255).

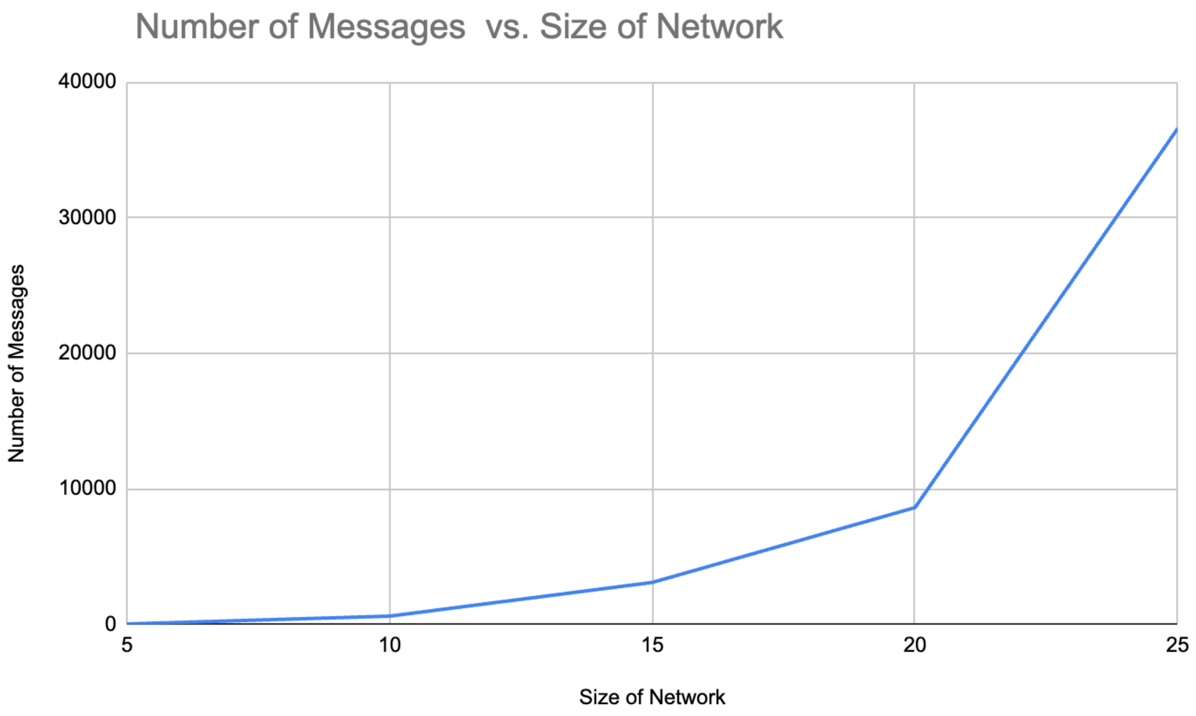
**Part 3: Router A's forwarding table:**

Port 1: Group A subnet (1.1.1.0/24)
Port 2: Link to Router B (1.1.4.0/30)
Port 3: Link to Router C (1.1.5.0/30)
Port D: Connection to ISP

| Destination | Subnet Mask | Next Hop |
|---|---|---|
| 1.1.1.0 | 255.255.255.0 | destination is directly connected to the port |
| 1.1.2.0 | 255.255.255.0 | 1.1.4.1 |
| 1.1.3.0 | 255.255.255.0 | 1.1.5.1 |
| 1.1.4.0 | 255.255.255.25 | destination is directly connected |

| | | |
|---|---|---|
| | 2 | to the port |
| 1.1.5.0 | 255.255.255.252 | destination is directly connected to the port |
| 1.1.6.0 | 255.255.255.252 | 1.1.5.1 (or 1.1.4.1) |

## Question 5: Routing


Number of Messages vs. Size of Network
Number of Messages
Size of Network