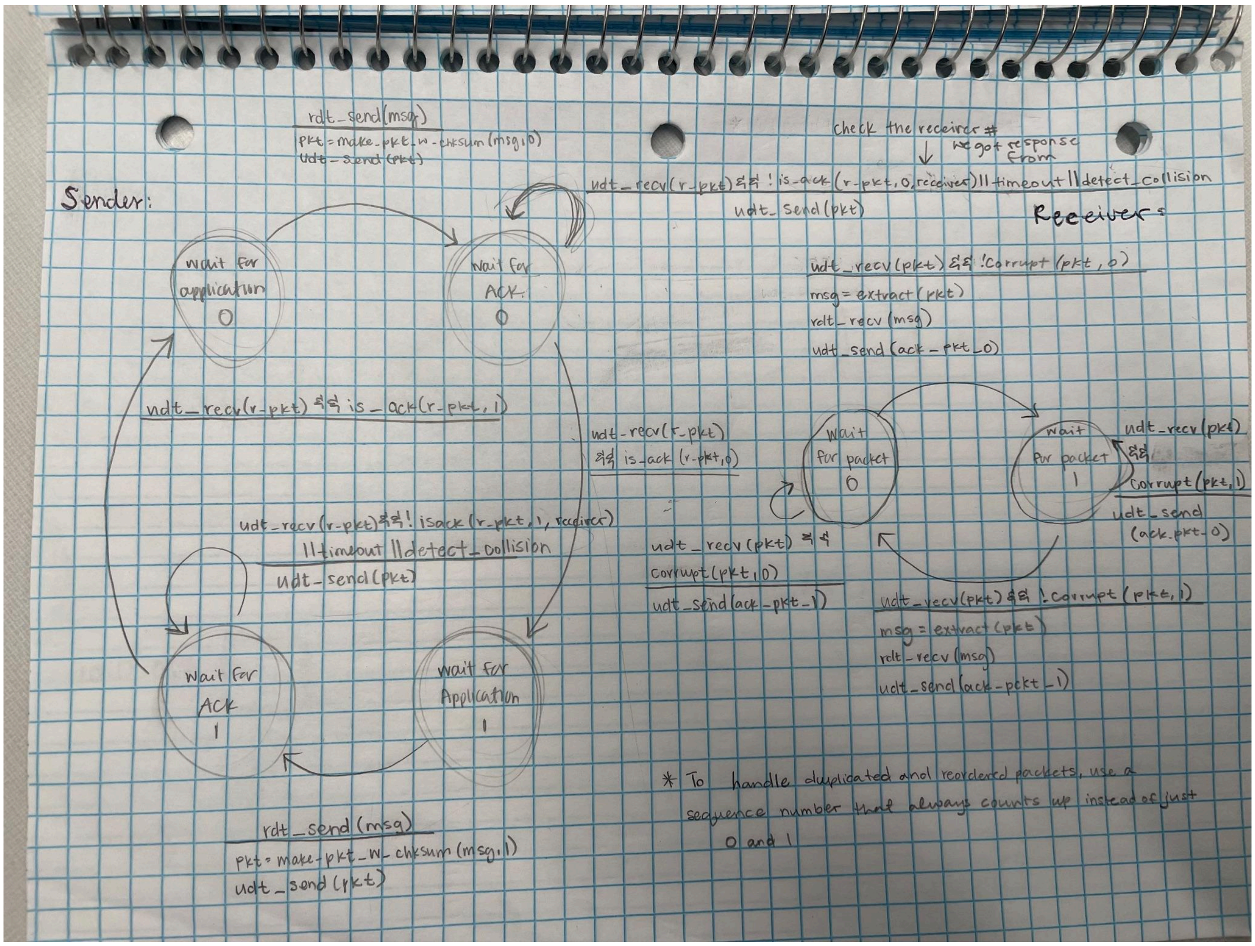
Elisabeth Frischknecht - CS 6014 HW 3

# Question 1: Reliable Data Transfer

NOTE: I added detect_collision conditions and a parameter that would check the receivers that we received the acknowledgement from




# Question 2: Throttling

What is the difference between flow control and congestion control?

Flow control prevents a host from overwhelming another host, but ignores the network. It limits the speed based on what the receiver can handle from the sender. Congestion control tries to modulate sending rate based off the estimated congestion in the network, but since we only receive messages through the endpoints and don't have data in the middle this must be done through receiving information through the ack we get back.

Describe the way TCP implements each of these features.

Flow Control:

- Flow control is something we do when a receiver can't keep up with the speed that we are transmitting data to it. It is supposed to help the sender moderate their sending speed.
- In TCP, the receiver will send an ack back to the sender when it has received something. In the header of the ack, the receiver sends the amount of free space they have available to receive messages (the receive window–rwnd). The sender have more data "in flight" than the last rwnd it has received (lastByteSent - lastByteAcked must be less than the last rwnd received).
- If the receive buffer is completely full, there will be a rwnd of 0. The sender can't send any more data, but the receiver also won't send any messages to the sender to let them know they have room to receive. So if the sender gets a rwnd of 0 but still has data to send, they will send 1 byte messages to the receiver. Some of those messages may get dropped, and the sender will resend them after a timeout. Once the rwnd is available again, the sender will transmit data.

Congestion control:

- Congestion control is used to try to mitigate the effects of network congestion–congestion can slow things down, and cause packets to be lost.
- When a TCP connection starts it begins with a small "congestion window" (cwnd) size. While starting out, TCP will increase the cwnd exponentially with each received ack to speed up the transmission rate–the "slow start" phase.
- TCP tracks a threshold (ssthresh) to determine when to switch from the "slow start" phase to a "congestion avoidance" phase
    - If a packet is lost (timeout), ssthresh is set to cwnd/2, then cwnd is reset to 1 and the slow start process starts over
    - If cwnd is larger than ssthresh, then we switch to congestion avoidance
    - If we get 3 duplicate ACKs, we will switch to fast recovery
- During congestion avoidance, TCP increases the cwnd linearly for each RTT (round-trip-time) until it reaches the ssthresh value. (linearly by MSS size– do cwnd += (MSS/cwnd)*MSS  where MSS is the Maximum Segment Size)
    - If there is ever a loss (timeout or triple duplicate ACK) then the ssthresh will be set to cwnd/2
        - If there's a timeout, cwnd is set to MSS and the sender switches back to slow start
        - If it's a triple duplicate ACK, cwnd is set to ssthresh + 3MSS and switches to fast recovery phase

- Fast recovery mode is what we use whenever we get a triple duplicate ACK, indicating that there has been a loss. The triple duplicate indicates that the receiver is still getting packets but there is a loss. (I believe that the timeout condition implies that the receiver has stopped receiving packets entirely)
    - Since the receiver is still receiving packets, we don't need to dramatically reduce the sending rate because they only missed one.
    - While we are receiving duplicate acks we will increase the cwd for each ack:  cwnd += MSS
    - As soon as we get a good ack we switch to congestion avoidance
    - On a timeout, we set ssthresh to cwnd/2 and switch to slow start with cwnd 1.

# Question 3: NAT

Two hosts (IPs A: 10.0.0.1 and B: 10.0.0.2) sit behind a NAT enabled router (public IP 5.6.7.8). They're both communicating with a remote host X, 1.2.3.4 on port 80. What are *possible* values for the source and destination addresses and ports for packets:

- from A to X behind the NAT
    - Source: 10.0.0.1, port 7743 (could be basically any port that isn't used)
    - Dest: 1.2.3.4 , port 80
- from B to X behind the NAT
    - Source: 10.0.0.2, port 6650 (another unused port)
    - Dest: 1.2.3.4, port 80
- from A to X between the NAT and X
    - Source: 5.6.7.8, port 11115 (decided by the translation table, preset)
    - Dest: 1.2.3.4, port 80
- from B to X between the NAT and X
    - Source: 5.6.7.8, port 11114 (decided by the translation table)
    - Dest: 1.2.3.4, port 80
- from X to A between X and the NAT
    - Source: 1.2.3.4, port 80
    - Dest: 5.6.7.8, port 11115
- from X to A between the NAT and A
    - Source: 1.2.3.4, port 80
    - Dest: 5.6.7.8, port 11114

What are the corresponding contents of the router's NAT translation table?

| Inside | Outside |
|---|---|

| | |
|---|---|
| 10.0.0.1 7743 | 11115 |
| 10.0.0.2 6650 | 11114 |

# Question 4: Routers

A company has 3 groups that each have a subnet on the corporate network. Group A uses subnet 1.1.1.0/24. Group B uses 1.1.2.0/24. Group C uses subnet 1.1.3.0/24.

Each group has a router. There is a link between each pair of routers.
A and B have a link: 1.1.4.0 (on A) to 1.1.4.1 (on B) A and C have a link: 1.1.5.0 (on A) to 1.1.5.1 (on C) B and C have a link: 1.1.6.0 (on B) to 1.1.6.1 (on C)

- How many subnets are a part of this network, and what is the smallest IP prefix (i.e. most fixed bits) that can be used to describe each one?
  - There are 6 subnets
  - Since the number after each subnet is /24, the first 24 bits (the first three digits separated by ".") are fixed, and the last digit can be varied. So the smallest "fixed" one is.
    - 1.1.1.0
    - 1.1.2.0
    - 1.1.3.0
    - 1.1.4.0
    - 1.1.5.0
    - 1.1.6.0
- If this network is somehow connected to the internet, what is the cheapest (i.e. smallest number of address) IP prefix the company could have purchased (without using NAT)?
  - 1.1.0.0/21
  - For this particular group of addresses, we go up to 6 in the third integer. The number 6 must be written with the three least significant bytes, so we would need to purchase after that point. So we would need 11 bytes of information saved for us. This would give us 2^11 addresses which is 2048.
- Assume the router for group A has 4 ports: port 1 is connected to the group subnet, port 2 is connected to router B, port 3 is connected to router C, and port D is connected to the ISP. Write out router A's forwarding table.

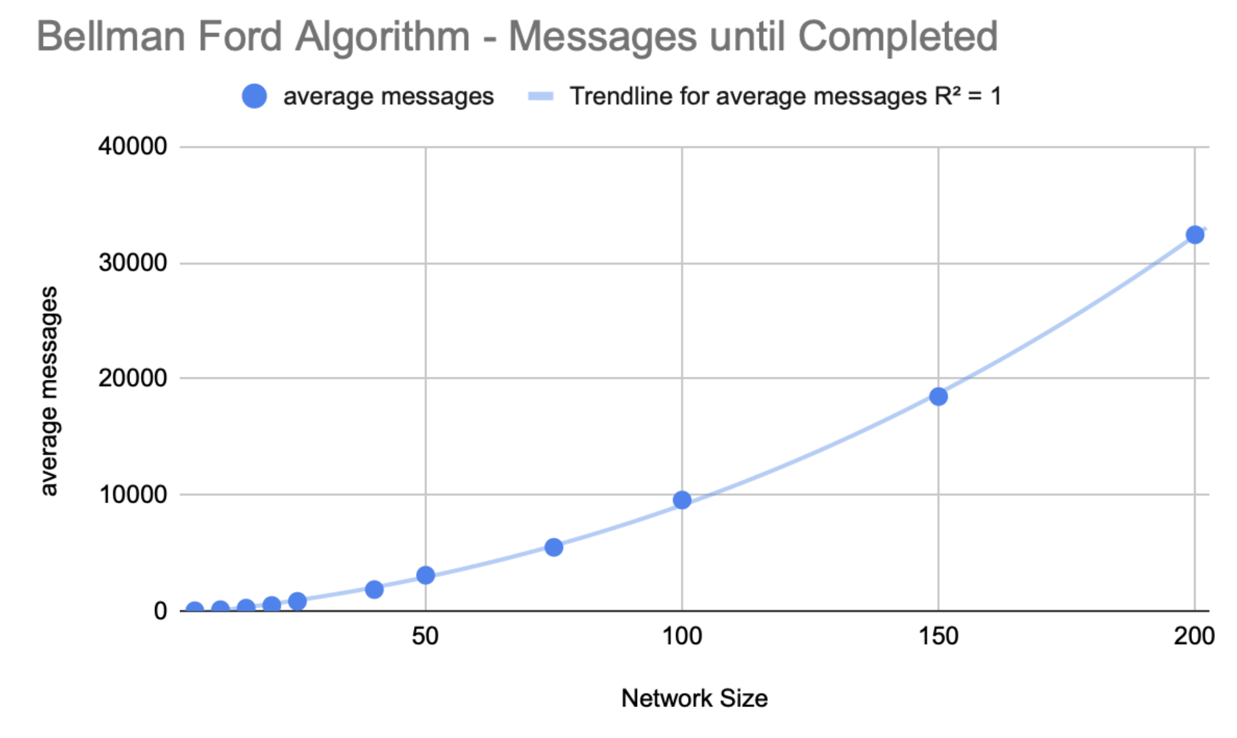
| port | IP address |
|---|---|
| 1 | 1.1.1.0/24 |
| 2 | 1.1.4.1 |
| 3 | 1.1.5.1 |
| 4 | 0.0.0.0 //this would be the ISP address, which we aren't given… |

# Question 5: Routing

To determine the average number of steps per network to complete the Bellman Ford algorithm, I created a probabilistic network for a variety of network sizes. I ran the experiment 20 times for each network size, and then averaged the number of messages needed to build the final tables.

The table shown below is using a cubic fit, which fits the O(M N^3) behavior expected from this algorithm.

| Network Size | average messages |
|---|---|
| 5 | 20 |
| 10 | 109 |
| 15 | 244 |
| 20 | 474 |
| 25 | 827 |
| 40 | 1839 |
| 50 | 3077 |
| 75 | 5480 |
| 100 | 9556 |
| 150 | 18472 |
| 200 | 32409 |


Bellman Ford Algorithm - Messages until Completed
average messages
Trendline for average messages R² = 1
40000
30000
20000
10000
0
average messages
50
100
150
200
Network Size