Noah Hanks
ECEn 426

Networking Homework

1. Did you do the Getting Started Wireshark lab?

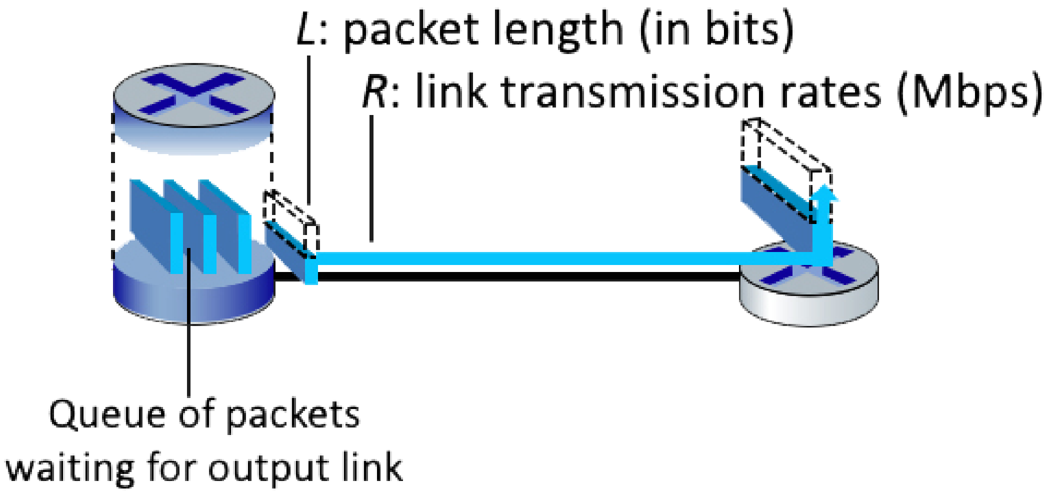
   Yes
2. What are the benefits and drawbacks of using packet-switching vs circuit switching?

   With packet switching, the packet can find its own way to the destination without a special channel to change the path. This causes packet switching to be more efficient when transporting data. The downside of packet switching is that it isn't optimal for things that require low latency. It's good for surfing the web or watching video but a video call would have laggy audio and video. With circuit switching, dedicating one channel to a particular service makes it so you can't use the channel for anything else but keeping users connected in a live call would work great.
3. Describe a human analogy for a protocol. What happens when someone does not follow the protocol?

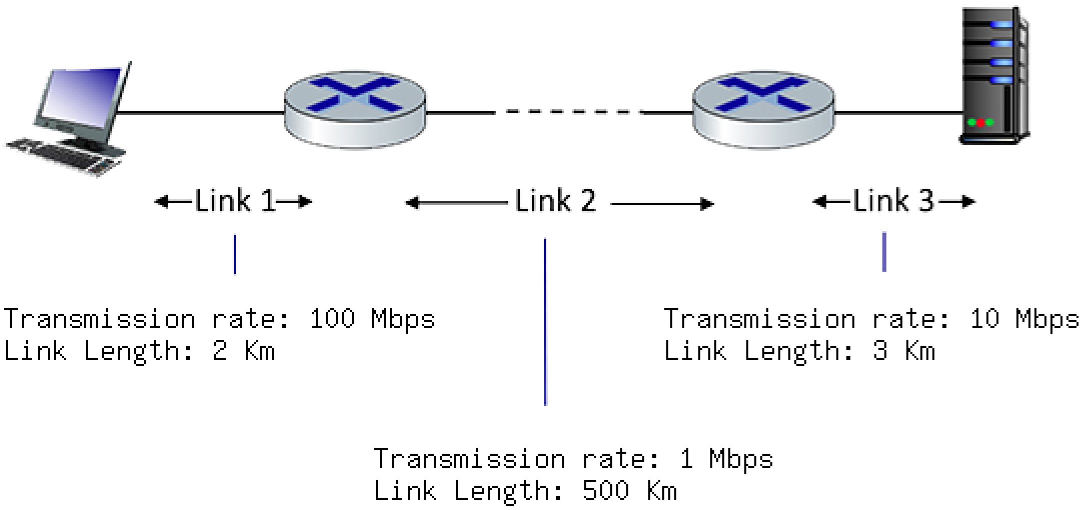
   Humans have many protocols such as when we greet each other. We usually start with a greeting and a response of another greeting. Someone may then ask a question such as the time and the other person would answer it if they knew the answer. When someone doesn't follow these protocols, the conversation gets messed up and things don't make sense.
4. What should a networking protocol define?

   A networking protocol should define how different devices communicate with each other and how information should be transferred. If a computer is requesting information from a server, it should ask in the right way and the information from the server should be given in a way that the computer can accept.
5. What problems are introduced by using a store-and-forward transmission technique for routers and switches?

   When devices use store-and-forward technology, it may take a log longer to send a packet since the device has to wait for the whole packet to arrive before its next hop. As such, this can add a lot of delay and latency and decrease overall responsiveness.

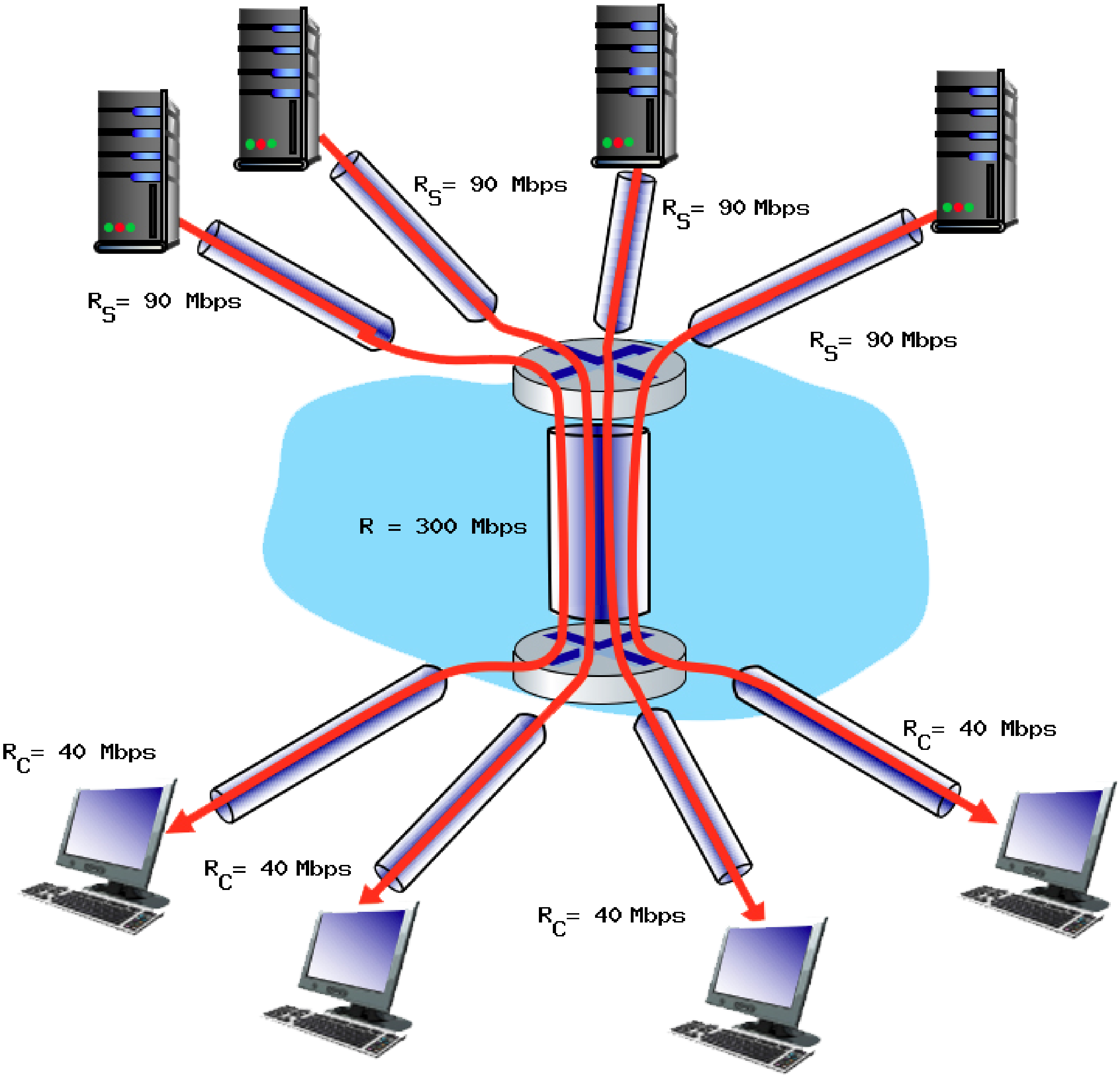

L: packet length (in bits)
R: link transmission rates (Mbps)
Queue of packets waiting for output link

6. Suppose that the packet length is L = 16000 bits, and that the link transmission rate along the link to router on the right is R = 100 Mbps.
    a. What is the transmission delay?
       L/R = 16000/(100*10^6) = 1.6*10^-4 seconds or 0.16 milliseconds
    b. What is the maximum number of packets per second that can be transmitted by this link?
       R/L = (100*10^6)/16000 = 6250 packets


Link 1
Link 2
Link 3
Transmission rate: 100 Mbps
Link Length: 2 Km
Transmission rate: 10 Mbps
Link Length: 3 Km
Transmission rate: 1 Mbps
Link Length: 500 Km

7. Assume the length of a packet is 4000 bits. The speed of light propagation delay on each link is 3x108 m/sec. What is the transmission delay and propagation delay for links 1, 2, and 3? What is the total end-to-end delay?
    Transmission Delay(1): 4000/(100*10^6) = 0.04 milliseconds
    Propagation Delay(1): 2*10^3/(3*10^8) = 0.0067 milliseconds
    Transmission Delay(2): 4000/(1*10^6) = 4 milliseconds
    Propagation Delay(2): 500*10^3/(3*10^8) = 1.667 milliseconds
    Transmission Delay(3): 4000/(10*10^6) = 0.4 milliseconds

Propagation Delay(3): 3*10^3/(3*10^8) = 0.01 milliseconds
End-to-End Delay: 6.1237 milliseconds



8.

a. What is the maximum achievable end-end throughput (in Mbps) for each of four client-to-server pairs, assuming that the middle link is fairly shared (divides its transmission rate equally)?

   40Mbps

b. Which link is the bottleneck link?

   Rs vs Rc vs R/4
   Rc is the smallest

c. Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the server links (RS)?

    R_bottleneck/Rs = 40/90 = 0.44

d. Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the client links (RC)?

    R_bottleneck/Rc = 40/40 = 1

e. Assuming that the servers are sending at the maximum rate possible, what is the link utilizations for the shared link (R)?

    R_bottleneck/(R/4) = 40/(300/4) = 40/75 = 0.533

1. **Did you do the HTTP Wireshark lab?**

    i. Yes

2. **Suppose you and four of your friends are trying to load the same webpage.**

    1. **You decide to open multiple parallel non-persistent HTTP connections. Will this help you load the webpage more quickly than your friends? Why or why not?**

        a. Opening multiple connections will help load the webpage more quickly because it allows your computer to send more requests to the server and receive multiple things at once rather than having to wait for each item to load one after another

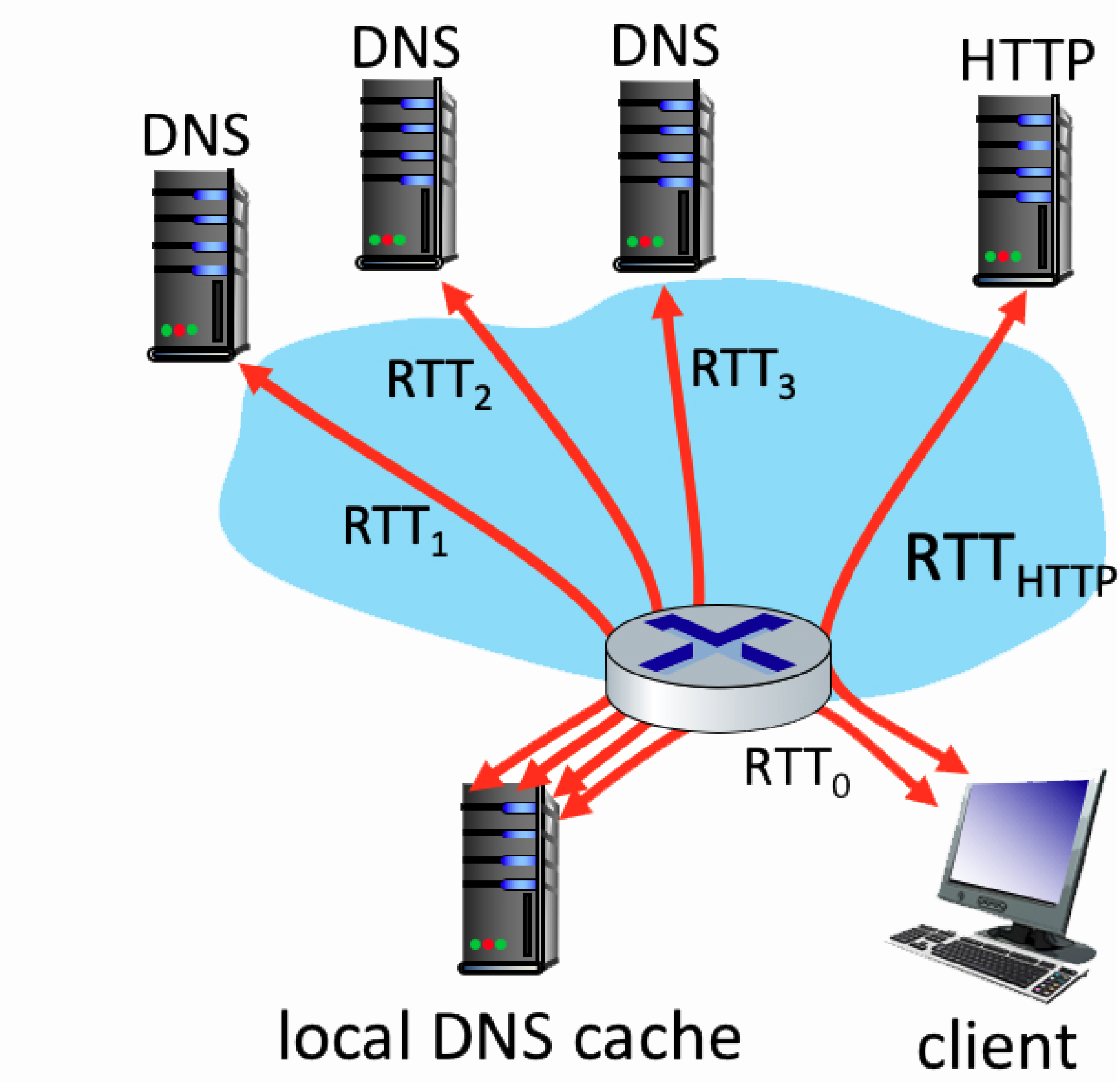
    2. **If all of your friends do the same thing and open the same number of parallel non-persistent HTTP connections as you, will your parallel connections still be beneficial? Why or why not?**

        a. The parallel connections will still be beneficial because the server is made to send out many responses to requests all at once. If five people are opening multiple connections to the server, all will be able to make multiple requests for items on the page to load.

3. **What is the difference between non-persistent HTTP connections and persistent HTTP connections? Which one is better and why?**

    i. Non-persistent connections have to make a new TCP connection for each request that they want to send. Persistent uses the same connection and will save time because the round trip time will be lower. It can send requests back to back without having to open a new TCP connection to the server each time.

4.



Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that four DNS servers are visited before your host receives the IP address from DNS. The first DNS server visited is the local DNS cache, with an RTT delay of RTT_0 = 5 ms. The second, third and fourth DNS servers contacted have RTTs of 35, 30, and 1 ms, respectively. Initially, let's suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Suppose the RTT between the local host and the Web server containing the object is RTT_{HTTP} = 6 ms.

1. **Assuming zero transmission time for the HTML object, how much time (in ms) elapses from when the client clicks on the link until the client receives the object?**
    a. (5+35+30+1)+(2*6) = 83ms
2. **Now suppose the HTML object references 2 very small objects on the same server. Neglecting transmission times, how much time (in ms) elapses from when the client clicks on the link until the base object and all 2 additional objects are received from web server at the client, assuming non-persistent HTTP and no parallel TCP connections?**
    a. (5+35+30+1)+(2*6)*3 = 107ms
3. **Suppose the HTML object references 2 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with non-persistent HTTP.**
    a. (5+35+30+1)+(2*6)*2 = 95ms
4. **Suppose the HTML object references 2 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with persistent HTTP.**
    a. (5+35+30+1)+(3*6) = 89ms
5. **What's the fastest method we've explored: Nonpersistent-serial, Nonpersistent-parallel, or Persistent-parallel?**
    a. Persistent-parallel

5.

How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain.

i. SMTP uses a period to mark the end of the message body. Http uses a "content-Length header field." HTTP can't use the period because it could have binary data whereas SMTP uses a 7 bit ASCII character to represent the period character.

6.

Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

i. By accessing the local DNS servers in the department, you can see which web servers are requested more often than others. This will cache in the local DNS and show which site are the most popular in the department.

7. **Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e., not a network administrator). Can you determine if an external website was likely accessed recently from a computer in your department? Explain.**

   i. Through the command line, you can use dig followed by an external site to see the query time. This is stored in the local DNS cache. If the query time is low, then the site was recently accessed from a computer in the department.

8. **Suppose you join a BitTorrent torrent, but you do not want to upload any data to any other peers (free-riding). Can you receive a complete copy of a file that is shared by the swarm. Why or why not?**

   i. Possibly but probably not. Whenever you torrent something, you usually automaticcaly start slowly uploading the chunks of the files you already have so new people can start downloading it too. If you aren't uploading, or seeding, your IP address can be flagged and you could be marked leecher, or someone who hits-and-runs, meaning that you selfishly downloaded files but didn't give back to the community by uploading in return. As such, you may be blacklisted from torrent trackers and unable to torrent future files.

1. Did you do the TCP and UDP Wireshark labs?
    i. **Yes**

2.
    1. Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these 2 bytes?
        a. 01011100 +
        01100101
        11000001
        00111110

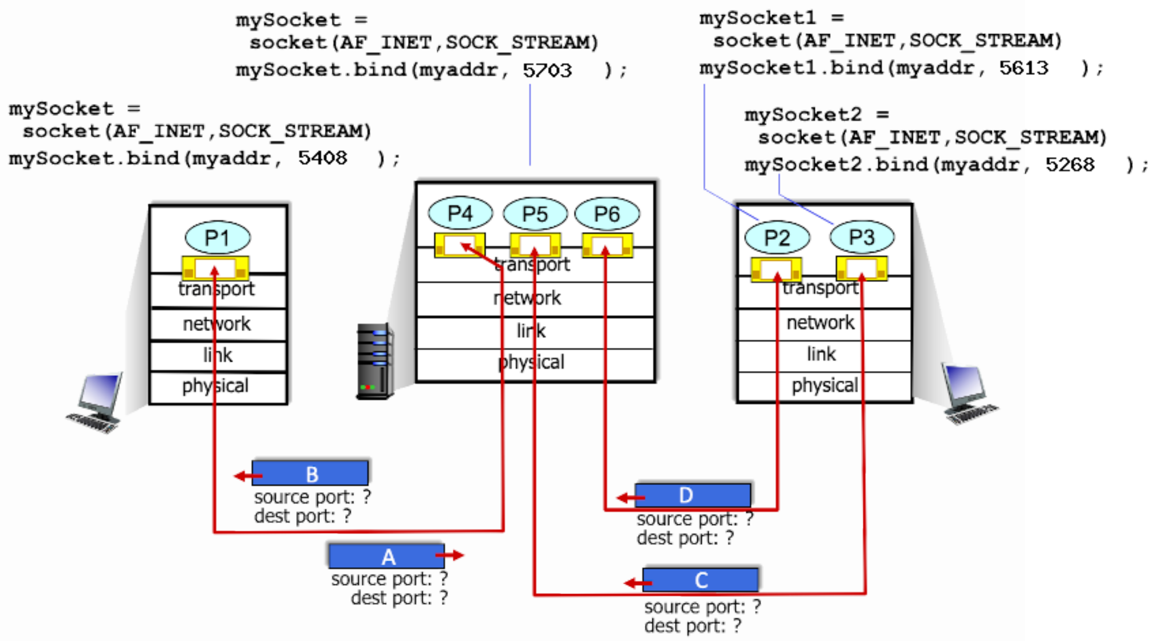
```
  11111
  01011100
+ 01100101
  11000001
```

```
11
11011010
01100101
00111111
01000000
10111111
```

    2. Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1s complement of the sum of these 2 bytes?
        a. 11011010 +
        0110010101000000
        10111111
    3. For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1s complement doesn't change.
        a. 01011101 +
        01100100
        11000001
        00111110

3.
    1. What is the UDP/TCP checksum protecting against?
        a. **Used to protect against errors such as flipped bits**
    2. What is the UDP/TCP checksum not protecting against?
        a. **Does not protect against times when two bits may have been flipped but the checksum yields the same result**
    3. Why should you include the header in the checksum and not just the payload?
        a. **The header includes important information such as the source and destination port, length, and checksum.**

4.



```
mySocket =
 socket(AF_INET,SOCK_STREAM)
mySocket.bind(myaddr, 5408   );
```

```
mySocket =
 socket(AF_INET,SOCK_STREAM)
mySocket.bind(myaddr, 5703   );
```

```
mySocket1 =
 socket(AF_INET,SOCK_STREAM)
mySocket1.bind(myaddr, 5613   );
```

```
mySocket2 =
 socket(AF_INET,SOCK_STREAM)
mySocket2.bind(myaddr, 5268   );
```

1. What is the source port # for packet B?
   - **a. 5703**
2. What is the destination port # for packet B?
   - **a. 5408**
3. What is the source port # for packet A?
   - **a. 5408**
4. What is the destination port # for packet A?
   - **a. 5703**
5. What is the source port # for packet D?
   - **a. 5613**
6. What is the destination port # for packet D?
   - **a. 5703**
7. What is the source port # for packet C?
   - **a. 5268**
8. What is the destination port # for packet C?
   - **a. 5703**

5.

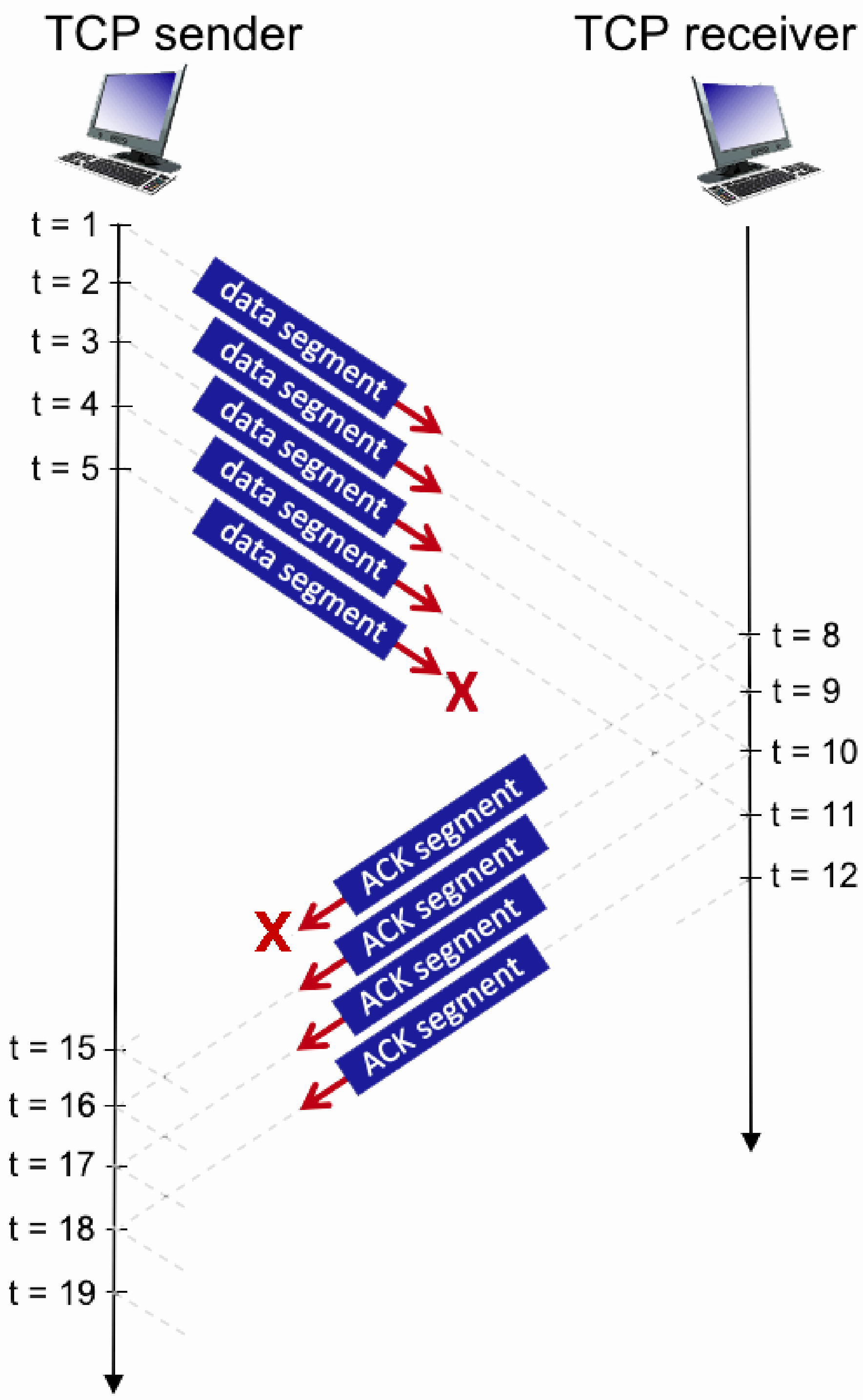
Suppose that TCP’s current estimated values for the round trip time (estimatedRTT) and deviation in the RTT (DevRTT) are 290 ms and 11 ms, respectively. Suppose that the next measured value of the RTT is 220 ms. For the following questions, use the values of α = 0.125, and β = 0.25.

1. What is the estimatedRTT after the new RTT?

    a. $(1 - \alpha)*\text{estimatedRTT} + \alpha*\text{RTT}$
       $(1-.125)*290 + .125*220 = 281.25\text{ms}$

2. What is the RTT Deviation for the the new RTT?
    a. $(1 - \beta)*\text{DevRTT} + \beta*|\text{estimatedRTT - sampleRTT}|$
       $(1-.25)*11 + .25*|290-220| = 25.75\text{ms}$
3. What is the TCP timeout for the new RTT?
    a. $\text{estimatedRTT} + (4*\text{DevRTT})$
       $281.25 + 4*25.75 = 384.25\text{ms}$

6.



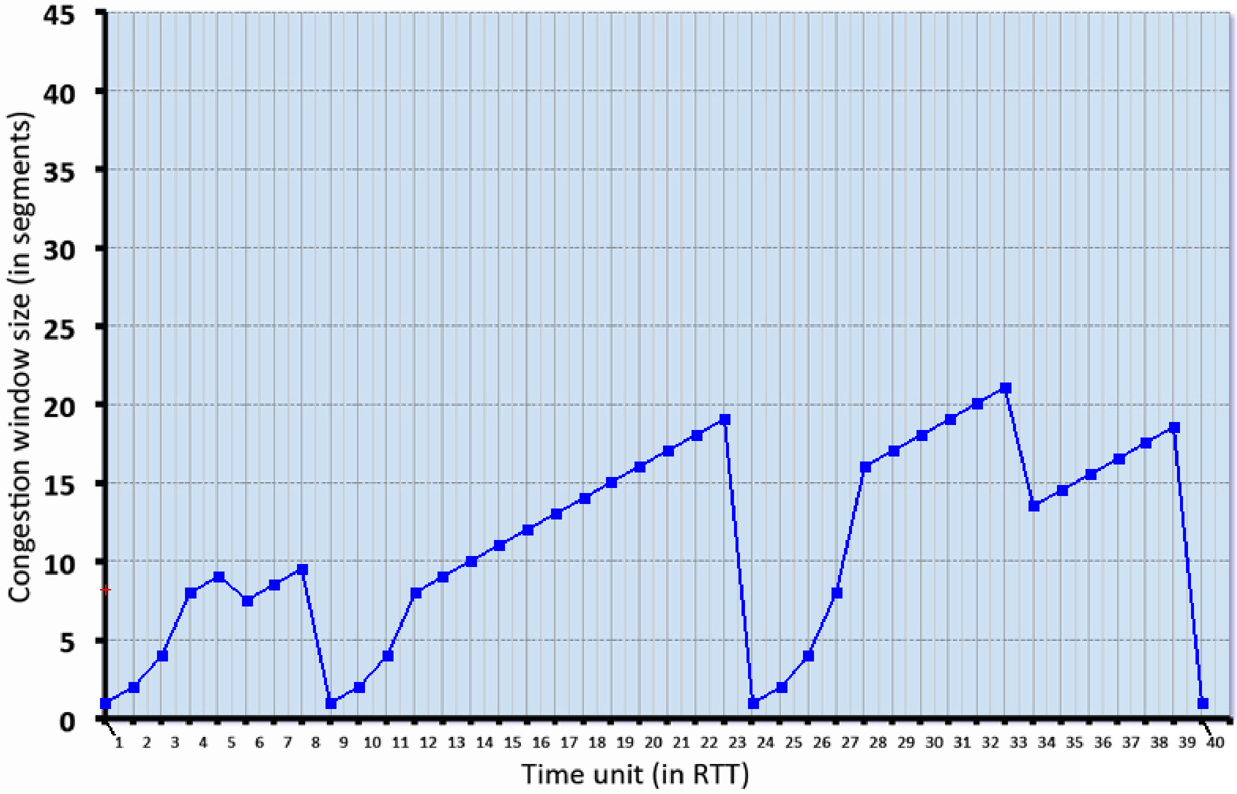
Consider the figure above in which a TCP sender and receiver communicate over a connection in which the segments can be lost. The TCP sender wants to send a total of 10 segments to the receiver and sends an initial window of 5 segments at t = 1, 2, 3, 4, and 5, respectively. Suppose the initial value of the sequence number is 60 and every segment sent to the receiver each contains 818 bytes. The delay between the sender and receiver is 7 time units, and so the first segment arrives at the receiver at t = 8, and an ACK for this segment arrives at t = 15. As shown in the figure, 1 of the 5 segments is lost between the sender and the receiver, but one of the ACKs is lost. Assume there are no timeouts and any out of order segments received are thrown out. For each of the time periods (t = 1 to t = 19), list the corresponding sequence number or acknowledgement number. If for a time period, no data or acknowledgement was received or no data was sent, mark with an x.

i.

time	**Sequence number from sender**	**ACK sent by receiver**
t = 1	**60**	
t = 2	**878**	
t = 3	**1696**	
t = 4	**2514**	
t = 5	**3332**	
t = 6		
t = 7		
t = 8		**878**
t = 9		**1696**
t = 10		**2514**
t = 11		**3332**
t = 12		**X**
t = 13		
t = 14		
t = 15	**X**	

t = 16	**4150**	
t = 17	**4968**	
t = 18	**5786**	
t = 19	**6604**	

7.



Consider the figure above, which plots the evolution of TCP’s congestion window at the beginning of each time unit (where the unit of time is equal to the RTT). In the abstract model for this problem, TCP sends a “flight” of packets of size cwnd at the beginning of each time unit. The result of sending that flight of packets is that either (i) all packets are ACKed at the end of the time unit, (ii) there is a timeout for the first packet, or (iii) there is a triple duplicate ACK for the first packet. In this problem, you are asked to reconstruct the sequence of events (ACKs, losses) that resulted in the evolution of TCP’s cwnd shown above. Consider the evolution of TCP’s congestion window in the example above and answer the following questions. The initial value of cwnd is 1 and the initial value of ssthresh (shown as a red +) is 8.

1. Give the time ranges at which TCP is in slow start.
    a. **1-3, 9-11, 24-27, 40**

2. Give the time ranges at which TCP is in congestion avoidance.
    a. **4-8, 12-23, 28-39 (technically 6 and 34 are fast recovery)**
3. Give the times at which packets are lost via timeout.
    a. **8, 23, 39**
4. Give the times at which packets are lost via triple ACK.
    a. **5, 33**
5. Give the times at which the value of `ssthresh` updated.
    a. **6, 9, 24, 34, 40**

Network Layer Homework

1. Did you do the IP Wireshark lab?
   i. **Yes**
2.

Prefix Match	Interface
10	1
00	2
100	3
001	4
010	5
otherwise	6

Consider a datagram network using 8-bit host addresses. Suppose a router uses longest-prefix matching with the above forwarding table. Suppose a datagram arrives at the router, with the following destination addresses. Specify which interface will the datagram be forwarded using longest-prefix matching.

1. `01110001`
   `a. Interface 6`
2. `10011101`
   `a. Interface 3`
3. `01011011`
   `a. Interface 5`

3.

Consider sending a 2400-byte datagram into a link that has an MTU (maximum transmit unit) of 700 bytes. Suppose the original datagram is stamped with the identification number 422. You might have to look up the specifics of each field to answer this question. Remember, the IP header takes up 20 bytes of the MTU.

1. How many fragments are generated?
    a. $\left[\frac{(2400-20)}{(700-20)}\right] =$ **4 fragments**
2. List the (1) identification field, (2) fragment offset field, (3) more fragments flag, and (4) length field for each of the fragments generated.

	Fragment 1	**Fragment 2**	**Fragment 3**	**Fragment 4**
ID	**422**	**422**	**422**	**422**
Offset	**0**	**680/8 = 85**	**85+85 = 170**	**170+85 = 270**
Flag	**1**	**1**	**1**	**0**
Length	**700**	**700**	**700**	**360**

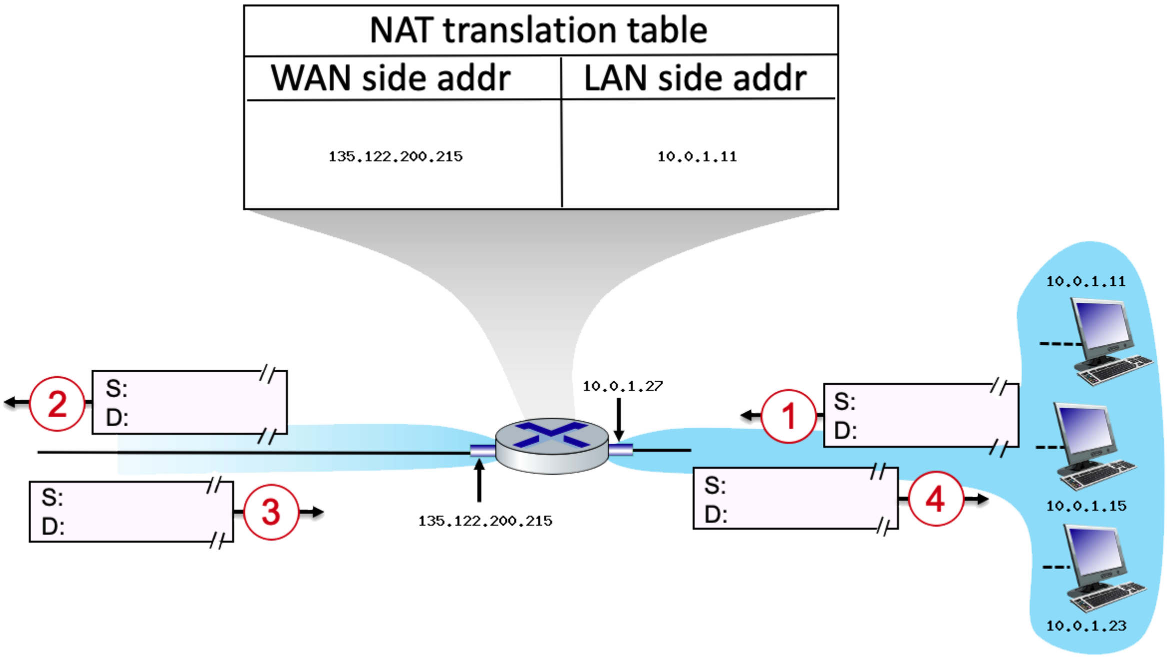
2400-700-700-700+20+20+20=360

4.

Assume you are setting up subnets on your home network, using the address space 192.168.0.0/16. You are trying to make subnets as small as possible but still accommodate the necessary number of hosts. What should the subnet (in CIDER notation) be if you want to support at most 61 hosts?

   i. **192.168.0.0/26**

5.



NAT translation table	
WAN side addr	LAN side addr
135.122.200.215	10.0.1.11

Consider the scenario below in which three hosts, with private IP addresses 10.0.1.11, 10.0.1.15, 10.0.1.23 are in a local network behind a NAT'd router that sits between these three hosts and the larger Internet. IP datagrams being sent from, or destined to, these three hosts must pass through this NAT router. The router's interface on the LAN side has IP address 10.0.1.27, while the router's address on the Internet side has IP address 135.122.200.215. Suppose that the host with IP address 10.0.1.11 sends an IP datagram destined to host 128.119.175.183. The source port is 3415, and the destination port is 80.

1. What is the source IP address, source port, destination address, and destination port for datagram 1?
    a. **S: 10.0.1.11, 3415**
    b. **D: 128.119.175.183, 80**
2. What is the source IP address, source port, destination address, and destination port for datagram 2?
    a. **S: 135.122.200.215, 5001**
    b. **D: 128.119.175.183, 80**
3. What is the source IP address, source port, destination address, and destination port for datagram 3?
    a. **S: 128.119.175.183, 80**
    b. **D: 135.122.200.215, 5001**

4. What is the source IP address, source port, destination address, and destination port for datagram 4?
    a. **S: 128.119.175.183, 80**
    b. **D: 10.0.1.11, 3415**

6.

Suppose you are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a random number, and the identification numbers of the subsequent IP packets are sequentially assigned. Assume all IP packets generated by hosts behind the NAT are sent to the outside world.

1. Based on this observation, and assuming you can sniff all packets sent by the NAT to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.
    a. **Just by using the packet sniffer, you can see the ID numbers of the packets and group the ones that are close together since they are sent sequentially. Thus, count how many different groups of packets being sent and that will determine the hose count.**
2. If the identification numbers are not sequentially assigned but randomly assigned, would your technique work? Justify your answer.
    a. **If the ID numbers aren't sequential, this technique wouldn't work since you couldn't identify any clusters sent by unique hosts.**

7.

In this problem we'll explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

i. **There is a method called NAT P2P hole punching. Basically, there is a third server involved where both Arnold and Bernard connect to. The server is able to get the IP's of Arnold and Bernard then send the IP info to both of them. They are then able to start a UDP p2p connection with each other and thus**

know the correct address to send requests. They can then make a TCP connection with each other.

Link Layer Homework

1. Did you do the Ethernet and ARP Wireshark lab?
    - i. **Yes**
2. Read about ARP spoofing online.
    1. Describe how an attacker would perform ARP spoofing to perform a person-in-the-middle attack.
        - a. **First, the person-in-the-middle sends a spoofed ARP message onto a LAN. Their goal is to get their MAC address to act as the IP address of another host so that traffic meant for the true IP is directed toward them instead.**
    2. Assuming a device has been on a network for awhile, devise a scheme where that device could detect that ARP spoofing is happening on the network and alert the network administrator.
        - a. **You would just need software that relies on some form of verification when using ARP responses where messages without certification are blocked.**
    3. If you could make any modifications to the link layer, what would you change or add to protect against ARP spoofing? Would that approach be feasible in real use?
        - a. **You could add some kind of cryptography encryption in the NIC that would have a public and private key for talking to routers and other devices on the LAN. It may add cost but would add protection against spoofing.**
3. List at least 7 protocols and the order they would be used when a user connects a new device to a network and goes to a webpage on their web browser for the first time. You can assume that no caching has previously been performed.
    - i. **DHCP, UDP, IP, Eth, Phy, ARP, DNA, HTTP**
4. In this problem, we explore the use of small packets for Voice-over-IP applications.
    1. Consider sending a digitally encoded voice source directly. Suppose the source is encoded at a constant rate of 128 kbps. Assume each packet is entirely filled before the source sends the

packet into the network. The time required to fill a packet is the packetization delay. What is the packetization delay in milliseconds, assuming that the packet is L bytes long?

   a. **(L bytes * 8)/128000kbps = L/16 msec**

2. Packetization delays greater than 20 msec can cause a noticeable and unpleasant echo. Determine the packetization delay for L = 1,500 bytes (roughly corresponding to a maximum-sized Ethernet packet) and for L = 50 (corresponding to an ATM packet).
   a. **1500 bytes/16msec = 93.75msec**
   b. **50 bytes/16msec = 3.125msec**
3. What is the percent overhead associated with packets L = 1,500 bytes long and for L = 50 bytes long when the packet header is 20 bytes? Assume that L includes the header.
   a. **Header bits/Transmitted bits * 100%**
   b. **(20*8)/(1500*8) * 100% = 1.33%**
   c. **(20*8)/(50*8) * 100% = 40%**
4. Calculate the transmission delay at a single switch for a link rate of R = 600 Mbps for L = 1,500 bytes, and for L = 50 bytes.
   a. **L/R = (1500*8)/(600*10^6) = 0.02 msec**
   b. **(50*8)/(600*10^6) = 0.000666 msec**
5. What are the advantages/disadvantages of using a small packet size?
   a. **Smaller packets take less time for the computer to make and decode them allowing for less end-to-end delay. There would be more overhead however when trying to transmit large files which would benefit from larger packet sizes.**

5.

What are the advantages and drawbacks of the following multiple access protocols?

1. Channel partitioning
   a. **Advantages:**
      i. **No collisions, perfect fairness**
   b. **Disadvantages:**
      i. **Max rate is R/N bps, even when only one node has frames to send**
      ii. **Fixed latency before a node can send data**
2. Random access
   a. **Advantages:**

        - i. **single active node can continuously transmit at full rate**
        - ii. **of channel**
        - iii. **highly decentralized: only slots in nodes need to be in sync**
        - iv. **simple**
    - b. **Disadvantages:**
        - i. **collisions, wasting slots**
        - ii. **idle slots**
        - iii. **nodes may be able to detect collision in**
        - iv. **less than time to transmit packet**
        - v. **clock synchronization**

3. Taking turns
    - a. **Advantages:**
        - i. **No collisions, fairness**
        - ii. **Fully decentralized, efficient**
    - b. **Disadvantages:**
        - i. **Rate less than R bps if only one node active;**
        - ii. **Master is single point of failure**
        - iii. **Token could get lost**