CMPE150 Midterm Solutions

Question 1 Packet switching and circuit switching:

(a) Is the Internet a packet switching or circuit switching network? Justify your answer.
The Internet is a packet switching network. It does not reserve resources (by establishing physical circuits) before sending data. It uses statistical multiplexing to more efficiently share its resources among users.

(b) If your answer for (a) was "packet switching", provide an example of a network that uses circuit switching technology; if your answer for (a) was "circuit switching", provide an example of a packet-switching network. In both cases, justify your answer.
The Plain Old Telephone System (POTS) is an example of a circuit-switching network. In the POTS, when a user calls another user, before any data (voice) can be sent, a physical circuit is established and resources along the circuit are reserved for the call ahead of time. While the call is taking place, the resources reserved for that call cannot be used by any other call and may stay idle for extended periods of time.

(c) Suppose you were recruited to take part in the group who is designing the Future Internet. To do that, you and your colleagues are considering that the main applications driving the Internet of the Future will be real-time services such as audio and video streaming, distributed games, etc. Would you propose a design based on packet switching or circuit switching? Justify your answer.
This answer can be either packet switching or circuit switching, depending on the justification. But a justification for circuit switching should discuss the pitfalls of using such a system for the future internet.

Question 2 Network performance:

(a) One metric of network performance we covered in class is data loss. Explain how data loss can be used as an indicator of how the network is performing.
Data can be lost due to transmission errors or data being dropped by routers. If losses caused by transmission errors are high, it usually means that the physical communication medium is unreliable (due, for example, to interference in case of wireless transmission) and therefore stronger reliability measures at higher layers of the protocol stack are needed.

In the case of losses due to data drops at routers, they usually mean that the network is

congested. To combat network congestion, routers can send congestion notifications to the end points so they can control the amount of data injected into the network accordingly. Alternatively, using a pure end-to-end approach, hosts, upon detecting data loss, can back off their transmission rate.

(b) In class, we discussed different ways loss can occur as data is transferred over the network. List and provide a brief explanation of the different types of data loss we discussed.
[See answer for letter (a)]

(c) Latency is another way to measure network performance. One way to account for network latency is to measure the round-trip time (RTT). What is the round-trip time?
The round-trip time is the interval of time between sending data and receiving the acknowledgment for it.

(d) Is the RTT constant or variable? Explain your answer.
The RTT is variable; its value depends on network load. The higher the load on the network, the higher the RTT as routers take longer to service data.

Question 3 List one advantage and one disadvantage of:

(a) Peer-to-peer model for networked applications (when compared to the client-server model). Explain.

In the peer-to-peer model, an end-host can be both a client and a provider of information. This means that peer-to-peer applications exhibit a higher degree of decentralization and distribution which typically increase robustness and fault-tolerance. One main disadvantage of peer-to-peer services is increased security, especially in terms of data integrity and authenticity as data sources are frequently not authoritative.

(b) DNS caching. Explain.

In DNS caching, a mapping that is requested by a DNS client can be serviced by the local DNS server. This usually results in faster name resolution and shorter response times. The main disadvantage of DNS caching (and caching in general) is that the cached information may be out of date.

(c) Layering. Explain.
The main advantage of a layered system design is ease of implementation, as well as maintaining and evolving the system. The system is architected as a set of layers, where each layer builds upon the services provided by the layer below and offers its own service to the layer above.
The main disadvantage of layered systems is higher overhead as well as possible duplication of effort across layers.

(d) DNS' distributed database (compared to a centralized name service). Explain.

The main advantages of implementing a distributed name service for the Internet is robustness, high availability, as well as support for administrative autonomy and decentralization.

The main disadvantage is higher complexity when compared to a centralized service.

Question 4 Alice was studying for her midterm exams in the library when her friend Bob calls to tell her he just posted a collection of her favorite video clips on his Web site hosted at www.coolsites.com.

(a) Alice immediately tries to access Bob's Web site. Describe the steps that need to happen before Alice's machine at QueensAcademy.edu can issue a request for Bob's Web site hosted at www.coolsites.com. Assume that this is the first time content from coolsites.com is requested by a someone at QueensAcademy.edu. Explain your answer.

ITERATED name resolution: Alice's web client would send out its request to the DNS server on its local LAN and that server would make all the requests thereafter.

1. The DNS resolver at Alice's QueensAcademy.edu machine queries the local DNS server for www.coolsites.com.
2. QueensAcademy's local DNS server checks its cache and doesn't find the mapping for www.coolsites.com. It then queries the root server for www.coolsites.com, who returns the IP address of the top level domain for .com.
3. The local DNS queries the .com DNS server who returns the IP address for the authoritative DNS server for coolsites.com.

4. The local DNS queries the authoritative DNS server for www.coolsites.com who then returns the ip address of www.coolsites to Alice.

RECURSIVE name resolution: Alice's web client would send out its request to the DNS server on its local LAN and each DNS server involved in the name resolution process will forward the query to the next server.

1. The DNS resolver at Alice's QueensAcademy.edu machine queries the local DNS server for www.coolsites.com.
2. QueensAcademy's local DNS server checks its cache and doesn't find the mapping for www.coolsites.com. It then queries the root name server for www.coolsites.com.
3. The DNS root name server forwards the query to the name server for .com.
4. The name server for .com forwards the query to the authoritative name server for coolsites.com, which returns to the .com name server the IP address for www.coolsites.comBob's.
5. The name server for .com forwards the mapping to the root name server.
6. The root name server returns the IP address for www.coolsites.com to the local DNS server for QueensAcademy.edu, who sends the information to the resolver on Alice's machine.

(b) Soon thereafter, Bob calls his other friend Carla, who also goes to Queen's Academy. Carla quickly tries to watch Bob's video from her machine at QueensAcademy.edu by issuing a request to Bob's Web site at www.coolsites.com. What steps need to happen before Carla's request is sent to www.coolsites.com. Assume Alice's request has already been issued. Explain your answer.

Before the Web request for Bob's Web site at www.coolsites.com can be issued by Carla's browser, the name www.coolsites.com needs to be resolved to iits IP address. To this end, the DNS resolver on Carla's machine issues a DNS query to QueensAcademy.edu local DNS server which will check its cache for the mapping being requested. Since the information had been recently used by Alice, it is still on the QueensAcademy.edu DNS server and therefore the mapping can be returned to Carla's DNS client without the need to query other DNS servers.

(c) Alice is finally able to download Bob's Web site. There are seven videos embedded in it. The total processing/service time within the network is 20ms and the one-way propagation delay is 120ms. Assume that transmission delay is negligible. What is the response time, i.e., the time between when the browser on Alice's machine requests the videos from www.coolsites.com and when they are delivered, assuming Alice's browser uses non-persistent HTTP? Explain your answer and show your work.

The processing/service time of 20ms combined with the 120ms propagation delay results in a total of 140ms one-way delay (latency). So 1 RTT = 280ms.

Since Alice's browser uses non-persistent HTTP, transferring each object will require a separate TCP connection. Thus, for each object, the transfer time $T_{transfer}$ is:

$T_{transfer}$ = 1 RTT (for connection establishment) +
1 RTT (for requesting and transferring object) = 2*RTT = 560ms

For all the objects (page plus 7 embedded videos), the total transfer time $TT_{transfer}$ is:

$TT_{transfer}$ = 8 * 560ms = 4480ms

(d) What would be the response time if Alice's browser uses persistent HTTP? Explain and show your work.

If persistent HTTP is used, all objects can be transferred using a single TCP connection. Thus, in this case, TTtransfer is:

TTtransfer = 1RTT (for connection establishment) +
1RTT (for each object)*8
= 280ms + 280*8ms = 2520 ms

(e) Now, it's Carla's turn to download Bob's videos. Assuming the steps in (b) have already been executed and that the hit ratio for QueensAcademy.edu's cache is 50%, what is the average response time Carla experiences for each object in Bob's Web page? Suppose that the delay to access an object from within Queen's Academy is 15ms.

Assuming Carla's browser uses persistent HTTP and that the TCP connection has

already been open, the average response time Carla experiences to retrieve each object is:
$T_{average}$ = .50(280ms) + .50(15ms) = 147.5ms.

Question 5 Reliable data delivery transport protocols employ several mechanisms including feedback, checksums, sequence numbers, retransmissions, and retransmission timers.

(a) What is the specific problem retransmission timers try to address when used to accomplish reliable data transfer? Explain.

Retransmission timers are needed to recover from losses due to congestion, i.e., packets being dropped by routers due to their queues overflowing (as opposed as losses due to transmission errors). After a certain period of time (i.e., the retransmission timeout), if the sender does not receive an acknowledgment from the receiver, it will retransmit the lost segment .

(b) What is the retransmission timeout?

As discussed in (a), the retransmission timeout (or RTO) is the interval of time the transport-layer sender will wait before retransmitting a segment. When the retransmission timer reaches the value of the retransmission timeout, the sender retransmits the segment in question.

(c) What is the trade-off in defining the value of the retransmission timeout?

If we set the timeout too high, the TCP sender will wait too long before retransmitting a segment that may have been lost, increasing response time and decreasing throughput. If we set the timeout too short, the sender may be retransmitting segments unnecessarily.

(d) Protocols that employ negative acknowledgements (NACKs) should also use positive acknowledgments (ACKs). Why is that the case?

In high loss scenarios, for example, when the receiver is disconnected from the network, no data arrives at the receiver who does not know it is supposed to receive data and thus does not send negative acknowledgements informing the sender that it has not received any information. If ACKs were also used (e.g., by having receivers confirm receipt of data periodically), then the situation described above would not

happen, i.e., if a sender stops hearing from a receiver, it will assume the receiver is having problems and will act accordingly.

(e) Why do reliable data transfer protocols need sequence numbers?

Both end points need to be able to identify (generally - uniquely) the data that has been sent and received. This way data that has not been received can be retransmitted and if data duplicate are received at the receiver, it can also detect that.

Question 6 Suppose that Alice is sending Bob a message over the Internet. Alice's computer is directly connected to *router1* and Bob's to *router2*. *Router1* and *router2* are directly connected. Thus, Alice's communication with Bob goes to *router1*, then to *router2*, and finally to Bob's computer.

Assume the message is small enough that it does not need to be broken down into smaller units as it is processed by the lower layers. Illustrate your answer using a diagram.

(a) What happens to Alice's message as it gets ready to be transmitted, i.e, as it goes down the protocol stack on Alice's computer? Show what happens to the message at each protocol layer. Is this process called encapsulation or de-encapsulation?

The process of going down the Internet protocol stack is called *encapsulation*. At Alice's computer, the application-layer message is passed on to the transport layer which processes the message and encapsulates it with the transport-layer header. The resulting segment is passed to the network layer which does its own processing and encapsulates the segment with the network-layer header (depending on the segment size and the size of the packet, the segment has to be broken down into more than one packet). The resulting packet (or packets) is (are) passed to the data link layer (DLL) which encapsulates it (them) with its header (and sometime trailer, depending on the data link layer protocol being used) into DLL frames. Then the data finally passed to the physical layer (PHY) for transmission.

(b) What happens to the message as it is processed by *router1* ? Show what happens at each layer.

The process of having data being received by a host or router and going up the protocol stack is referred to as *de-encapsulation*. As *router1* receives data from Alice's computer, the PHY at *router1* passes the received frames to the DLL at *router1* which processes the frames and removes its header (and trailer) passing the resulting packets

to *router1*'s network layer. The network layer header at *router1* is processed and based on the destination address specified in the header, *router1* forwards the packet onto the appropriate output interface, in this case the one corresponding to the link with *router 2*. The packet will then be re-encapsulated with the network header, then with the DLL header, and then transmitted by the PHY to *router2*. Note that routers typically only run up to the network layer protocols.

(c) And at *router2* ? Show what happens at each layer.

The sequence of steps at *router2* is similar to *router1*. The only difference is that router2 will forward the packet onto Bob's computer.

(d) What happens to the message when it arrives at Bob's computer as it goes up the protocol stack? Is this process called encapsulation or de-encapsulation? Show what happens at each layer.

At Bob's computer, the data will be received at the PHY and passed to the DLL for processing. As the data moves up the Internet protocol stack, it is de-encapsulated at each layer. Headers (and trailers) are processed and removed from each frame at the DLL. The resulting packet (or packets) are passed to the network layer which inspects the destination address in the header and decides the packet (s) are destined to Bob's host. So it passes the resulting segments to the transport layer, which doest its own processing and delivers the message to the application layer process which is supposed to receive the messages.

Question 7 "Pipelined" protocols, also known as "sliding window" protocols, are a type of ARQ (Automatic Repeat Request) protocols that use a "window" to control the amount of data they inject into the network.

(a) Stop-and-Wait is one type of ARQ protocol. What is the window size of Stop-and-Wait? Explain.

In Stop-and-Wait, the window size is 1. By design, in Stop-and-Wait ARQ, only one segment is sent "at a time", i.e., before an acknowledgment is received.

(b) Based on your answer for (a), how many unique sequence numbers does Stop-and-Wait need? How many bits are needed to represent Stop-and-Wait's unique sequence numbers? Explain.

For Stop-and-Wait, we only need 2 unique sequence numbers since there can be only one “outstanding” segment at any given point in time. Thus, in order to minimize overhead, one can use sequence numbers “0” and “1”, and thus only one bit is needed to represent sequence numbers.

(c) What is the main advantage of requiring a smaller number of bits to represent the range of unique sequence numbers employed by a protocol? What is the main disadvantage?

As discussed in (b), less bits needed to represent sequence numbers means reduced overhead incurred by the protocol. The main disadvantage is protocol efficiency in terms of throughput and network utilization since only one segment can be sent every RTT.

(d) For a 100Mbits/sec channel with 100ms propagation delay, what is the channel utilization when sending 2KByte segments if Stop-and-Wait is used? Show your work.

Dtrans = L/R = 16000 bits/100,000,000 bits/sec
= 160 microseconds = .16 milliseconds

$$\text{Utilization} = \frac{(L/R)}{RTT + (L/R)} = \frac{.16 \text{ ms}}{200 \text{ ms} + .16 \text{ ms}} = .000799 \text{ or } .08\% \text{ utilization}$$

(e) How can you increase channel utilization? Explain using a numeric example to support your solution.

We could increase L, decrease R, and/or RTT, which will likely incur in additional hardware costs. The other alternative is to move away from Stop-and-Wait into pipelined protocols by allowing window sizes greater than 1.

$$\frac{X * (L/R)}{RTT + (L/R)}$$

where X is the number of additional packets in flight, i.e., the window size. So for X=10, the utilization will be increased 10 times.

(f) Describe the additional complexity of your solution compared to Stop-and-Wait.

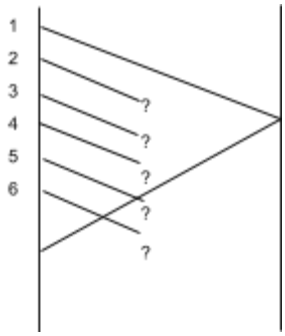
The complexity that we add to our solution as compared to Stop-and-Wait is that now we must increase the sequence number space to keep track of the X segments that can be in flight, which of them were received or not received, etc. If for (e) we chose to use another approach other than a windowed protocol, we do not actually add complexity to the the protocol.

Question 8 Suppose that a sender and a receiver are using ARQ to perform reliable data delivery.

(a) How many sequence numbers are needed to implement Stop-and-Wait?

Two sequence numbers.

(b) In a Go-Back-N ARQ protocol, the window size is 6. Segments with sequence numbers 1, 2, 3, 4 and 5 have been sent. The sender just received an ACK for segment 1. Segments 6, 7, 8, 9 and 10 are waiting to be sent. Draw the time diagram showing this scenario.



(c) Which segment(s) can the sender send before it must wait for the next ACK from the receiver? Explain.

Assuming segments 1 through 5 have been sent, the window size of 6 still allows one more segment (segment 6) to be transmitted. When the ACK for segment 1 is received, that opens up the window by an additional 1 segment which means that segment 7 can also be sent.

(d) Some time later, the sender transmitted segments 20, 21, 22, 23, 24, and 26; however, segment 22 got lost. If Go-Back-N is used, what segment(s) would the sender have to retransmitted? Explain.

Assuming that the ACKs for 20 and 21 are received, the sender would have to retransmit 22, 23, 24, (25), and 26.

(e) Suppose the same situation as above but sender and receiver use Selective-Repeat ARQ. What segment(s) would the sender need to retransmit? Explain.

Again, assuming that ACKs for 20 and 21 are received, only the lost segment will need to be retransmitted in Selective-Repeat ARQ. (25 and/or 22)

(f) Can Selective-Repeat ARQ use cumulative ACKs? Explain.

No since the receiver needs to be able to signal to the sender which segments have been received.

(g) What are the trade-offs between Go-Back-N ARQ and Selective-Repeat ARQ?

In Go-Back-N ARQ, the receiver does not need to keep state about “out-of-order” sequence numbers. It can just discard them and wait for the sender to retransmit them. That is not the case for Selective-Repeat ARQ.