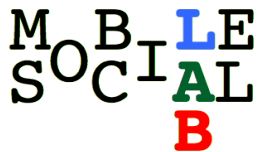


Lecture 28: What's new?

CSEE 4119: Computer Networks
Thursday April 26th



A couple of things

- * HW5 will be graded after final
- * Final Tuesday May 8th, at 1-4pm in Pupin 428
 - Come early!
 - No document, calculators are allowed
 - Cumulative (about everything, from class 1 to 28!)
- * Evaluation of the class (before May 5th!)
 - Please speak-up **especially** if you did not like it!
 - You can also speak if you like it, we will appreciate it, but the interpretation of feedback is most important!

The midterm review

* Exercise 1:

Exercise 1: Layers (20 pt)

1. Which one(s) of the following layer(s) of the Internet protocol stack can possibly provide end-to-end reliability (*i.e.*, there either exists a functionality that provides it or it could be easily added)? Which one provides it usually most of the time? (Justify your answer)
(a) Routing (b) Application (c) What is a “layer”? (d) Transport (e) What is the “Internet”?
2. Same question as above to provide communication security using a form of data encryption.

The midterm review

- * Exercise 1:
- * 1) Easy part: include transport
- * Easy mistake: forget application, or too much details
- * Real mistake: include anything else
- * 2) Easy part: include application
- * Easy mistake: anything else, misinterpret security implemented in a layer

The midterm review

* Exercise 2:

Exercise 2: Pipelining and utilization (20 pt)

Consider a 100-Mbps link of 10-km in length with 5 nanoseconds per meter propagation delay, constant 4,000 bit data packets, and negligible processing and queuing delays at both the sender and receiver. Assume the sender always has data to send, that there are no losses or corruptions, and $1\text{Mbps} = 10^6\text{bps}$.

1. If pipelining is used with a window N packets, what is the smallest value of N that ensures at least a 80% utilization of the link.

The midterm review

- * Exercise 2:
- * Here are the incorrect formulas for utilization people used.
- * 1) $N * d_t / (d_t + d_p)$
- * 2) $N * d_t / (N d_t + d_p)$
- * 3) $N * d_t / (N d_t + 2 d_p)$
- * 4) $N * d_t / d_p$
- * 5) $N * d_t / (2 d_p)$

- * A bunch of people also did: $N = .8 R/L$

The midterm review

* Exercise 3

Exercise 3: Delay for Web-traffic (30 pt)

Consider a client that wants to retrieve a WWW document at a given URL using HTTP. The IP address of the HTTP server is initially unknown. The WWW object at the URL has one embedded GIF image that resides at the same server as the original object.

1. Provide a list of transport and application layer protocols used besides HTTP in this scenario. Give a short description of their functions.
2. Suppose that the time needed to contact and receive a reply from any server (for any protocol) is RTT. How many RTTs are needed from when the user first enters the URL until the complete document is displayed? Assume that non-persistent http is used, and that another computer in the same local network recently queried it. Consider the delays of all protocols in your answer, not just those of HTTP.

The midterm review

* Exercise 3

- * - Here a bunch of people forgot to mention UDP
 - some people calculated RTTs with persistent connections...
 - some people forgot to include DNS RTT,
 - some people included RTTs for closing connections...

The midterm review

Exercise 4: Designing a reliability protocol for two receivers (80 pt)

* Ex
4

A host wishes to transmit message over a broadcast channel to two receivers. Unfortunately, losses may occur on this channel for any of the two receivers independently, so the sender will have to ensure reliability by receiving some signal from each receiver separately.

More precisely, consider a scenario in which a Host A wants to simultaneously send messages to Hosts B and C. A is connected to B and C via a broadcast channel: a packet sent by A (e.g., in a single `unreliable_send()` operation) is carried by the channel to both B and C.

We make the following assumption on the channel connecting A, B, and C:

- It can independently lose and corrupt messages from A to B and C (and so, for example, a message sent by A might be correctly received at B but not at C).
 - It has a maximum bounded delay of D (i.e., if a message is sent by A, it will either be lost or arrive at B and/or C within D time units).
 - any control message (e.g., an ACK or NAK) sent by B or C to A will only be received by A and not any other node, but like others they can be lost or corrupted.
 - We do not use any pipelining in this channel.
1. Design a stop-and-wait-like error-control protocol for reliably transferring a packet from A to B and C, such that A will not get new data from the upper layer until it knows that both B and C have correctly received the current packet. Give a FSM description for A and C (assuming the FSM for B is similar, if it is not similar give the FSM for B as well). Also, give a description of the packet format used.
 2. If we assume that each channel ($A \rightarrow B, A \rightarrow C, B \rightarrow A, C \rightarrow A$) may create a loss or a corruption independently from the others with probability p_e , and has the same propagation delay d_p . We assume that the packet has a transmission delay d_t and we neglect the transmission delay of the ACK, compute the utilization of this channel.

The midterm review

- * Exercise 4
- * Easy part: figure out that both receivers should ACK before proceedings, include a time-out
 - Figure out that SSN are needed, and ACK are signed.
- * Easy mistake: neglect that ACK retransmission, don't go in sufficient details to fix that case.
- * Real mistake: not really an alternative solution, so it's mostly for those who did not answer

The midterm review

* Exercise 5

Exercise 5: Difference of GO-BACK-N and SELECTIVE-REPEAT (50 pt)

In the diagram below, one row corresponds to a time slot during which one packet can be sent. Packet and ACKs propagates so that they are received in the following time slot in the other end. The RTT is hence 2 time slots. We assume that the timeout value is set to 5.5 (*i.e.*, after a packet is transmitted, unless an ACK is received the time out is triggered just before the 6th time slot). We assume Pipelining is used (window size 4, each packet size 1). Sequence numbers starts at 0. ACK indicates the sequence number of the packet they acknowledge. All transmissions are successful and without corruption except:

- The first packet with SSN=1 sent by the sender is lost.
- The first ACK sent by the receiver with SSN=3 is lost.

1. Assuming the Sender and Receiver implements GO-BACK-N, represent on the following diagram, for the first 11 time slots:

- The packet and ACK sent and received by each host during each time slot, with their SSN.
- The set of packets currently in the window maintained by each host.

The midterm review

- * Exercise 5-1
- * Easy part: numbering the ACK correctly
- * Easy mistake: assume fast retransmission
- * Real mistake: assuming that packets are ACKed but discarded

The midterm review

- * Exercise 5-2
- * Easy part: creating ACK for the correct packet
- * Easy mistake: misinterpret the window
- * Real mistake: wrong ACK numbering

The midterm review

* Exercise 6

Exercise 6: TCP and window evolution (50 pt)

We assume that the Maximum Segment Size is MSS (in bytes), and that the current value of `wnd` is W (in bytes).

1. What will be the value of `wnd` after the next new ACK (acknowledging new data) is received (we assume that no time-out or other event occurs before)? (Justify your answer!)
(a) $2 \times W$ (b) $W + MSS$ (c) $W + 1$ (d) W (e) It depends. (f) What is an ACK?
2. During the last hour, the evolution of the window was monitored and it has been observed that on average it is equal to $34 \times MSS$, while $MSS=500$ bytes. Knowing that the average throughput of the TCP connection has been 50 kilobyte per second, can you estimate the RTT of this connection?
3. Why would the RTT estimation obtained by TCP at the end of this hour be different from this estimation? Would you rather use this estimation or the estimation of TCP to set the time-out value?

The midterm review

* Exercise 6-1

- Easy part: not (f) or (d)
- Easy mistake: misunderstanding the unit,
- Real mistake:
 - * forget that it depends on the slowstart threshold
 - * Assuming that $W+1$ leads to linear increase!

The midterm review

* Exercise 6-2

- Easy part: something linear in W and $1/\text{rate}$
- Easy mistake: use a 0.8 coefficient (confusing W and $W\text{-max}$)
- Real mistake: not really

* Exercise 6-3

- Easy part: different because of time-scale
- Easy mistake: think more data is always better
- Real mistake: being lost in detail

The midterm review

* Exercise 7

Exercise 7: Efficiency of P2P (50 pt)

Please indicate TRUE or FALSE for each of the following statements and provide one element from the course to justify your answer:

1. Structured P2P architecture like DHTs are useful to avoid congestion at the server and serve a large number of clients.
2. P2P architecture such as DHTs and Bittorrent are ideally designed to find the content I need from a nearby peer, allowing for low delay.
3. Some P2P architecture allows to find efficiently an object among a very large number of peers.
4. Bittorrent is so efficient because although a single peer is not likely to have information useful to me, I can contact a very large number of peers.
5. P2P architectures are the only solution for a content provider to make a large number of clients accessing large objects such as videos.

The midterm review

* Exercise 7

- * 1. most people got this right
- * 2. A lot of people said true here, not realizing the "nearby peer" and "low delay" aspects, which are what CDNs try to do.
- * 3. Many people didn't realize this was referring to DHTs with shortcuts and $O(\log N)$ lookup times.
- * 4. Many people got this wrong, not remembering the result from the HW.
- * 5. most people got this right.