Review Materials: Set B

EE122 Fall 2012

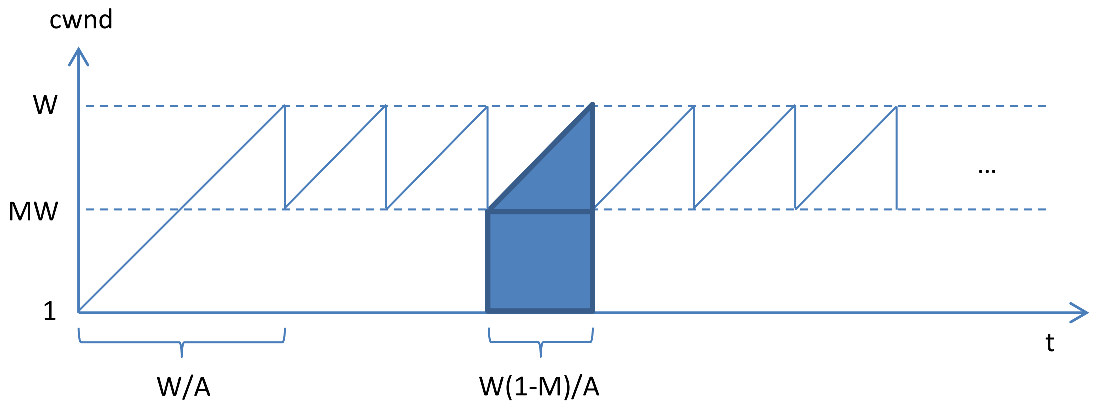
1. Lucky you're with AIMD

Consider a generalized version of AIMD, where:

- for every window of data ACKed, the window size increases by a constant A
- when the window size reaches W, a loss occurs, and the window size is multiplied by a constant M < 1

For simplicity, assume that W(1-M) is divisible by A. Thus, the window sizes will cycle through {WM, WM + A, WM + 2A, …, W}. Use RTT to denote the packet round trip time.

The window size over time is:



1. What is the average throughput (in # of packets)?

$$T = \frac{MW + W}{2RTT} = \frac{W(M+1)}{2RTT}$$

Sanity check:

- for regular AIMD (M = 0.5): T = 3W/4RTT

- for buggy AIMD (M = 0; see Section 11): T = W/2RTT

2. What proportion of packets is dropped?

In every sawtooth, there are

$$\frac{(MW + W)}{2} * \frac{W(1 - M)}{A} = \frac{W^2(1 - M^2)}{2A}$$

packets sent. One of these packets is lost, so the loss rate

$$p = \frac{2A}{W^2(1 - M^2)}$$

Sanity check:

- for regular AIMD (M = 0.5, A = 1): $p = 8/(3W^2)$
- for buggy AIMD (M = 0, A = 1): $p = 2/(W^2)$

3. What is the average throughput, as a function of the drop rate, RTT, A, and M?

Inverting the proportion of packets dropped:

$$W^2 = \frac{2A}{p(1 - M^2)}$$

$$W = \sqrt{\frac{2A}{p(1 - M^2)}}$$

Now substituting into the average throughput (in # of packets) equation:

$$T = \frac{(M + 1)\sqrt{\frac{2A}{p(1 - M^2)}}}{2RTT}$$

To get the average throughput in bytes, multiply by the MSS.

Sanity check:

- for regular AIMD (M = 0.5, A = 1):

$$T = \frac{(0.5+1)\sqrt{\frac{2}{p(1-0.5^2)}}}{2RTT} = \frac{1}{RTT} * \sqrt{\frac{3}{2p}}$$

- for buggy AIMD (M = 0, A = 1):

$$T = \frac{(0+1)\sqrt{\frac{2}{p(1-0^2)}}}{2RTT} = \frac{1}{RTT} * \sqrt{\frac{1}{2p}}$$

2. TCP

Panda has an infinite number of packets to send to Anand. Their point-to-point link has a propagation delay of 4ms, the transmission delay for a data packet is 1ms, and the propagation delay for an ACK packet is 1ms (the path from Anand to Panda is asymmetric). Assume there is no processing delay.

Implementation details:

- Panda is using simple fast retransmission
- during congestion avoidance, CWND += MSS / Int(CWND / MSS)
- the algorithm leaves slow-start when CWND > SSTHRESH (not >=)
- Panda's retransmission timer is 20ms, and is reset whenever new data is ACKed.

Assume that SSTHRESH starts at 10, and that the first transmissions of data packet #5 and #10 are lost. Assume that Anand uses traditional cumulative ACKs.

Determine what Panda's CWND is, and what packet(s) are sent at each time:

Time	Events for Panda	Panda's CWND	Events for Anand
0	Sends D1	1	
1			
2			
3			
4			
5			Receives D1; Sends A2
6	Receives A2; Sends D2	2	
7	Sends D3		
8			
9			
10			
11			Receives D2; Sends A3
12	Receives A3; Sends D4	3	Receives D3; Sends A4
13	Receives A4; Sends D5	4	
14	Sends D6		
15	Sends D7		
16			
17			Receives D4; Sends A5
18	Receives A5; Sends D8	5	<D5 lost>
19	Sends D9		Receives D6; Sends A5
20	Receives A5		Receives D7; Sends A5
21	Receives A5		

22			
23			Receives D8; Sends A5
24	Receives A5; Sends D5	2.5	Receives D9; Sends A5
25	Receives A5		
26			
27			
28			
29			Receives D5; Sends A10
30	Receives A10; Sends D10	3.0	
31	Sends D11		
32	Sends D12		
33			
34			
35			<D10 lost>
36			Receives D11; Sends A10
37	Receives A10		Receives D12; Sends A10
38	Receives A10		
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50	Sends D10		

3. Cheating in TCP

It takes a long time to receive an infinite number of packets, so Anand would like to speed things up. For the TCP example above:

4. What would likely happen if Anand had sent A2 at t = 1?

Panda would not realize anything was wrong, unless they fastidiously checked the end-to-end delay (which is fraught with stochasticity). Panda would probably increase the window size.

5. What would likely happen if Anand had sent A2 at t = 1, and A3 at t = 2?

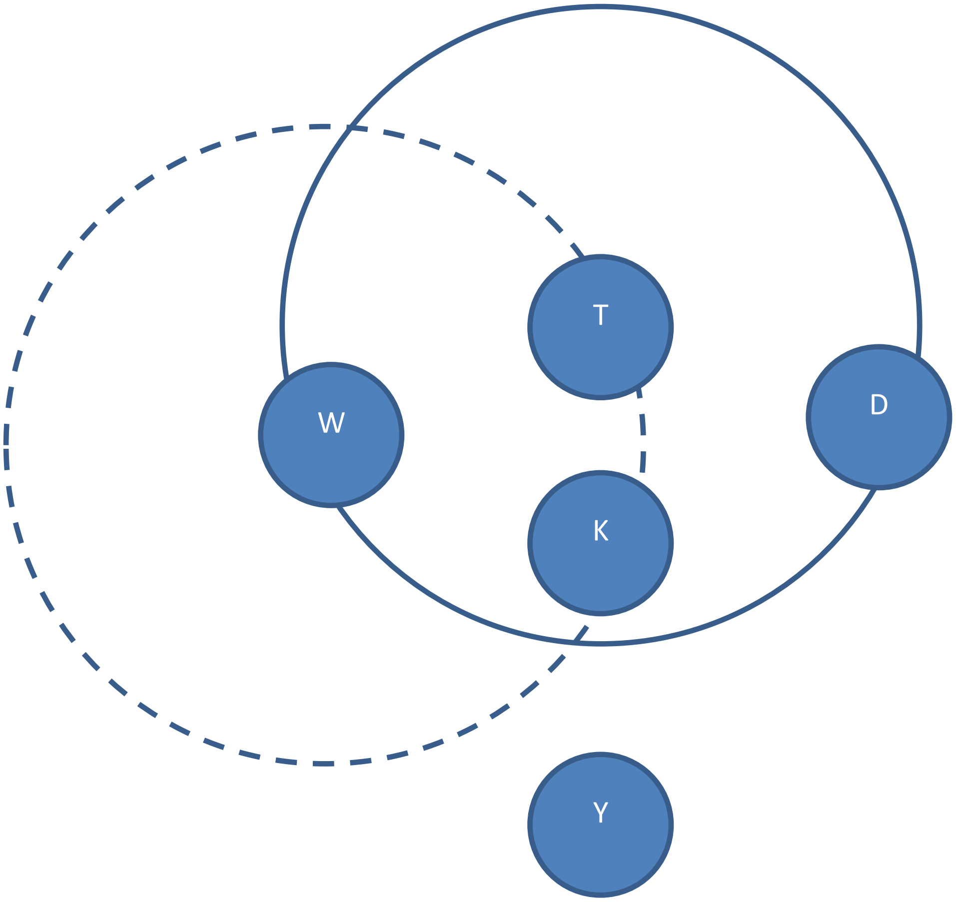
Panda has not yet sent packet 2, so they could detect that cheating had occurred.

6. What would likely happen if Anand had sent A6 at t = 14?

Panda would probably increase the window size. However, Panda would also assume that Anand has received all packets up to and including D5. Unfortunately, D5 becomes lost, and there would be no way for Anand to ask for Panda to retransmit it!

4. Wireless Etiquette

The following hosts are on the same wireless channel. Each host has the same receive/transmit radius (radii shown for two hosts).



Initially, there is no communication.

7. Taylor (T), who uses the classic MACA protocol (RTS-CTS-DATA-ACK), wishes to send data to W, and sends an RTS to W. Which hosts hear the RTS? Which hosts are allowed to send?

D, K, and W hear the RTS. All hosts are currently allowed to send.

8. W sends a CTS to Taylor. Which hosts hear the CTS? Which hosts are allowed to send?

T and K hear the CTS. T, D and Y are currently allowed to send.

9. Upon receiving the CTS, Taylor starts sending DATA to W. Kanye (K), having overheard the CTS, decides to ignore the protocol and immediately transmit. What happens?

A collision would occur at W.

The nodes within range of Kanye are:

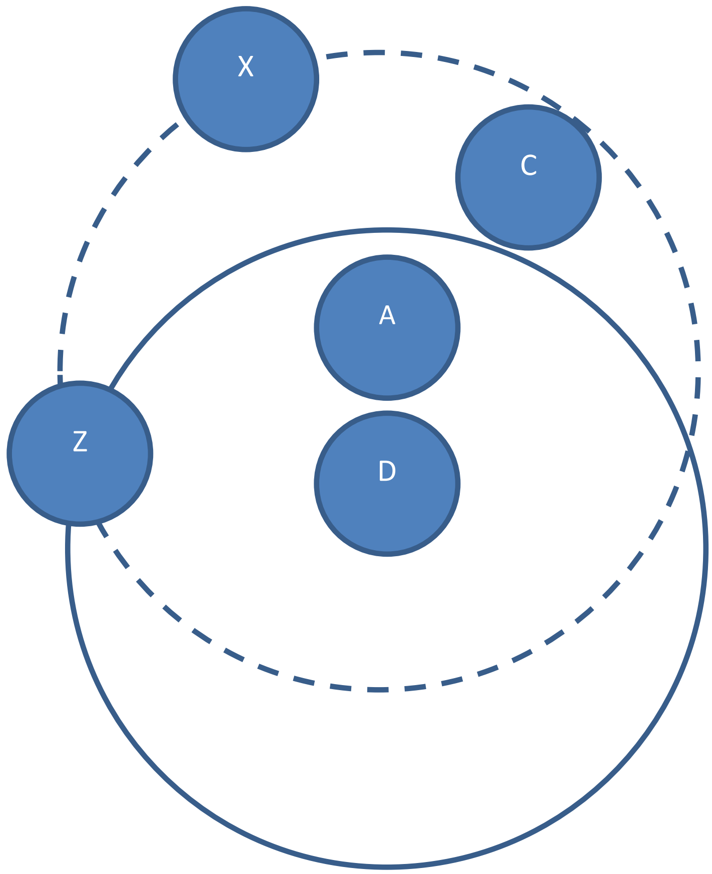
- W,D: Kanye can't successfully send, due to collision with Taylor's transmission
- T: Kanye can't successfully send, because Taylor can't receive while sending
- Y: Kanye can successfully send an RTS. However, Taylor's transmission would prevent Kanye from hearing a CTS! (exposed terminal problem)

  (Of course, Kanye doesn't really care about CTS, and could send DATA.)

10. What would have happened if Kanye had started transmitting before Taylor did?

- If Kanye completes an RTS/CTS exchange with W or D: Taylor would hear the CTS, and defer transmission
- If Kanye sends DATA immediately (without RTS/CTS), or completes an RTS/CTS exchange with Y: Taylor wouldn't hear a CTS, hence would send an RTS, but no one can receive it. *This question might be more realistic if it was the simplified 802.11 protocol.*

Now consider the following, similar scenario:



11. Cruz (C), who is using the simplified 802.11 protocol (carrier sense + RTS/CTS), completes the RTS/CTS handshake with X, and starts sending DATA. Arnold (A) ignores the CTS he heard from X, and begins sending DATA to Z. What happens?

(Essentially Q9 – the carrier sense doesn't matter here)

Arnold and Cruz's transmission collide at X. Arnold's DATA is successfully received by Z.

12. Suppose Cruz hadn't started transmitting, and that instead Arnold had been continuously transmitting DATA for the previous two hours. When should Cruz send?

By the rules of carrier sense, Cruz would need to wait until Arnold stops sending, even if this is unreasonably long. Furthermore, there is nothing Cruz can do to tell Arnold to shut up – Arnold is not listening when he's sending.

13. Arnold's twin, Danny, also ignores RTS/CTS and carrier sense, and hence Arnold

and Danny are both sending to Z. What happens?

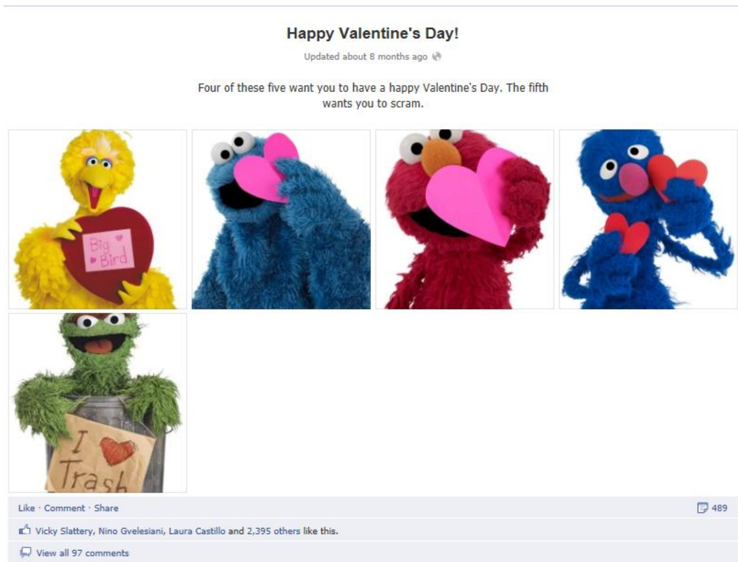
Collisions happen at Z. In the absence of cooperation, they are both going to collide almost always.

5. HTTP

We would like to download all the thumbnails from one of the TA's list of "Top Ten Sesame Street Photo Albums". This involves the following steps:

- download a master index page, of size I. This page contains links to ten album pages.

- download the album pages, each of size A. Each album page contains five photos.



An example of an album page.

- download the photos (thumbnails), each of size P.

We can't start downloading the album pages before we finish downloading the entire master index page. We can't start downloading the images from album *i* before we finish downloading album page *i* (but we don't need to wait for album page *j*).

Assumptions **(read these carefully!)**:

- the transmission delay for any one album page is larger than an RTT

- every photo is larger than any one album page

- HTTP request packets, TCP SYNs and ACKs are negligibly small

- the connections can each achieve throughput T (but if there are multiple connections they must share the throughput)

- we don't need to wait for the HTTP responses to be acknowledged, nor for TCP connections to terminate

- the only files we need to download are those explicitly stated above in steps i-iii (i.e., the total size of the HTTP responses is [I + 10(A + 5P)])

- all files are hosted on the same web server

For each of the following scenarios, calculate the **minimum** total time to download the pages and photos.

14. Sequential requests with non-persistent TCP connections:

    _____ x RTTs + _____ x I/T + _____ x A/T + _____ x P/T

Content	RTTs		HTTP transmission	#	Totals
	TCP	HTTP			
Master index page	1	1	I/T	1	2 RTT + I/T
Album page	1	1	A/T	10	20 RTT + 10A/T
Photo	1	1	P/T	50	100 RTT + 50P/T

The grand total is 122 RTT + I/T + 10A/T + 50P/T.

15. Concurrent with non-persistent TCP connections

    Additional assumptions:

    - all album pages are requested concurrently

    - TCP connections are not opened pre-emptively (e.g., we do not open a TCP connection for use with the album pages, until we are ready to make a HTTP request)

    _____ x RTTs + _____ x I/T + _____ x A/T + _____ x P/T

The assumption that the album pages are requested (and therefore received simultaneously), means we don't need to worry about downloading some album pages and some photos concurrently.

For each of the album pages, the 2 RTTs (for TCP and HTTP) occur concurrently (i.e., not 10x20 RTTs). Likewise, for each of the photos, the 2 RTTs occur concurrently. The throughput is shared between concurrent downloads, so the transmission time is not affected. Hence:

Total = (2 RTT + I/T) + (2 RTT + 10A/T) + (2 RTT + 50P/T)

= 6 RTT + I/T + 10A/T + 50P/T

16. Sequential with a single persistent TCP connection

_____ x RTTs + _____ x I/T + _____ x A/T + _____ x P/T

This is the same as "Sequential requests with non-persistent TCP connections", except that we don't need any RTTs for TCP, other than the first one i.e.,

Content	**RTTs**		**HTTP**	**#**	**Totals**
	TCP	**HTTP**	**transmission**		
Master index page	1	1	I/T	1	2 RTT + I/T
Album page	0	1	A/T	10	10 RTT + 10A/T
Photo	0	1	P/T	50	50 RTT + 50P/T

The grand total is 62 RTT + I/T + 10A/T + 50P/T.

17. Pipelined within a single persistent TCP connection

_____ x RTTs + _____ x I/T + _____ x A/T + _____ x P/T

The steps are:

- download the master index page (1 RTT for TCP + 1 RTT for HTTP + I/T).
- download the album pages (1 RTT for HTTP + 10A/T).
- download the images (*0* RTT for HTTP + 50P/T).

Once the first album page is received, we place HTTP requests for the first batch of images. Our assumptions imply that, while the HTTP request for any batch of albums is being propagated, we are receiving more album pages and/or images at the same time.

Total = (2 RTT + I/T) + (1 RTT + 10A/T) + (0 RTT + 50P/T)

= 3 RTT + I/T + 10A/T + 50P/T

6. You Down with BGP?

The domains A, B, C, … , Z are fully interconnected (i.e., they each have a direct link to all other domains), and use BGP. We do not know their selection or export policies, which can be **arbitrary**. We do know that, in the steady state, the following routes have been exported:

- ABCDEF
- LMNOP
- PQ
- and some others that we don't know about.

Indicate whether, in the steady state, the following route advertisements:

- are **guaranteed** to be exported regardless of the choice of selection or export policies
- are **guaranteed** to be exported **if standard** policies are used, but not necessarily true in general
- are **not guaranteed** to be exported, even if standard policies are used

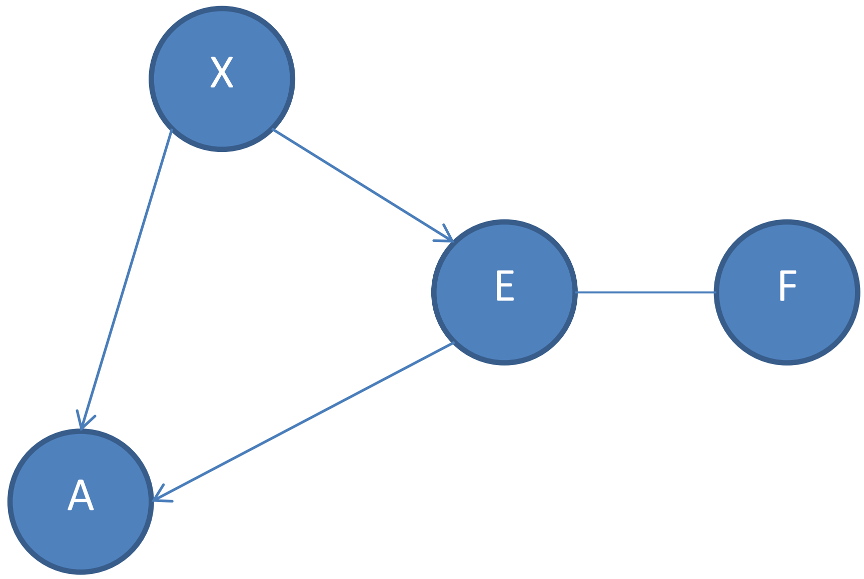
1. ABCDE ~~Guaranteed if standard~~ **Not guaranteed**

The path A-F implies that B, C and D find it profitable to transit traffic to E. Their profitability (and thereby willingness to provide transit) is not affected by whether E decides to move the data to F.

However, there may be a more preferable route: in the following diagram, suppose X is the primary provider of A[1], and E is a backup provider.

- AXEF is not a valid route, so A must choose AEF
- For A to E, AXE is preferred (not AE).

[1] For simplicity, we have omitted nodes B, C and D; if B is the provider of A, C is the provider of B, D is the provider of C, and E is the provider of D, then the same reasoning applies.



Since ABCDE is not guaranteed under standard policies, it is not guaranteed in general.

We can also note, albeit redundantly, that ASes may make arbitrary decisions based on the entire path, thereby allowing ABCDEF but not ABCDE.

2. ABCDEFG Not guaranteed

See (L, Q)

3. BCDE ~~Guaranteed if standard~~ **Not guaranteed**

See (A, E)

4. BCDEF Guaranteed

"Any traffic I carry will follow the same path my traffic takes" (p. 34, lecture 16)

5. LMNOPQ Not guaranteed

Even though we know LMNOP and PQ are preferred, we cannot assume that paths can be stitched together. e.g., suppose P is a customer of O, and PQ is a peer link: under standard policies, P would not allow transit.