

Worksheet #1

EE122 Fall 2011

You are not required to do these problems, and we are not asking you to hand them in. However, if you can't do these problems, then you will have trouble on the final. In fact, many of these questions came from last year's final. As the saying goes, **"Know This"**.

1. Acronyms

Consider the following acronyms: A, AIMD, BGP, CDN, CIDR, CNAME, DHCP, DNS, ECN, FIN, HTTP, ICANN, LPM, MSS, MTU, MX, NAT, NS, PMTU, PTR, RFC, RIP, RST, RTO, SSTHRESH, STFU, SYN, TCP, TLD, UDP.

Match the acronyms to the descriptions below, using each acronym exactly once. Supply exactly one answer for each item below. In some cases, an acronym might fit more than one question, but there is only one way to answer these so that every acronym is used exactly once. For terms that aren't familiar to you, use the book or the web.

- i) A mechanism that allows the network to indicate congestion to TCP without having to drop the packet. **ECN**
- ii) The DNS record type that provides the IP address of a given hostname. **A**
- iii) The DNS record type that provides the canonical name associated with a given hostnam. **CNAME**
- iv) The DNS record type that provides the authoritative name server associated with a given domain. **NS**
- v) The DNS record type that provides the mail server associated with a given domain. **MX**
- vi) The DNS record type that provides a hostname associated with a given IP address **PTR**
- vii) The name of the message used to initiate a TCP connection. **SYN**
- viii) A message (control flag) used to terminate a TCP connection abruptly. **RST**
- ix) A message (control flag) used to terminate a TCP connection smoothly. **FIN**
- x) The kind of window adjustment algorithm used in TCP. **AIMD**
- xi) The size of the largest IP packet that can be sent end-to-end along a path without fragmentation. **PMTU**
- xii) The largest sized segment that can be sent by TCP. **MSS**
- xiii) The largest size IP packet that can be sent across a particular link. **MTU**
- xiv) The most common examples of this acronym are .com, .net, and .org. **TLD**

- xv) The documents used by the IETF to describe protocol standards. **RFC**
- xvi) An in-line network device that rewrites IP addresses (and often transport ports). **NAT**
- xvii) A widely used routing protocol that does not necessarily compute lowest-cost paths. **BGP**
- xviii) The organization that is ultimately responsible for address allocation and domain name management. **ICANN**
- xix) A method for matching packets to routing entries that determines which among multiple matching entries determines the packet's destination. **LPM**
- xx) The length of time TCP waits for an acknowledgement before timing out and initiating a retransmission. **RTO**
- xxi) The most widely used reliable transport protocol. **TCP**
- xxii) An unreliable transport protocol. **UDP**
- xxiii) The protocol that provides newly arrived hosts with their own IP addresses (and also gives them other information). **DHCP**
- xxiv) An IP addressing scheme that does not involve the traditional A, B, and C address classes. **CIDR**
- xxv) Akamai is an example of one. **CDN**
- xxvi) The TCP congestion control state variable that determines when to switch from slow-start to congestion avoidance. **SSTHRESH**
- xxvii) An application-level protocol that typically runs over TCP. **HTTP**
- xxviii) An application-level protocol that typically runs over UDP. **DNS**
- xxix) A distance-vector routing protocol. **RIP**
- xxx) The appropriate response to Stanford students when they bring up the 2010 Big Game. **STFU**

2. A TCP Sob Story

Due to budget cuts, the CS department is forced to buy cheaper networking equipment. It turns out that one of the switches has a defective port that has a strange dropping behavior. On a particular port, it drops every fourth packet, in both the sending and receiving directions. To be precise, if one only looks at the packets being received, it drops the 4th, 8th, 12th, ... packet. Similarly, if one only looks at the packets being sent, it drops the 4th, 8th, 12th, etc.

This problematic port is the port that attaches the department chair's computer, host A, to the network, so every fourth packet sent to A is dropped, and every fourth packet sent by A is dropped.

Suspecting a problem, the CS department runs a test by initiating a file transfer from host A to an offsite host B. The RTT of the packets is 1 msec, the packet transmission time is .01msec, and the RTO (timeout value) in the TCP connection is 100msec (you won't need these exact numbers, they just set the relative magnitude of the various timescales). Host A uses a transport protocol with a sliding window flow control with a constant window size of 5 packets and a duplicate ACK threshold of 3 packets (3 duplicate ACKs lead to a retransmission). The test consists of sending a series of data packets (labeled D1, D2, ...) from A to B.

We want to describe the transmissions *from* A (by writing Dx for the appropriate x). The first five data packets are shown, in addition to the starting SYN and ACK packets, just to get you started (and just to clarify, we assume that the packet count, in terms of dropping, starts with the SYN; therefore, in these first few packets, D2 and only D2 is dropped). The packets from A that will be dropped are marked in bold. Please fill in the remaining 15 packets sent from A. (And remember that the packets coming *to* A are also dropped in a similar one-every-four pattern, with the packet count starting with the SYN-ACK, so the first packet dropped in the receiving direction is the ACK sent in response to D4.)

SYN, ACK, D1, **D2**, D3, D4, D5, **D6**, **D2 (TO)**, D7, D8, **D9**, **D10**, **D6 (TO)**, **D11**, **D12**, **D13**,
D9 (TO), **D14**, **D15**, **D16**, **D12 (TO)**,

How many fast retransmits occur during the transmission of this set of packets? 0

How many timeouts occur during the transmission of this set of packets? 4

3. TCP Congestion Control Details

In the following, assume that the $MSS=1000$ bytes and that in all computations $CWND$ is rounded down to the nearest integer.

During the congestion avoidance phase, TCP updates the window following each ACK by the following equation:

$$CWND += MSS / \text{Int}(CWND / MSS)$$

Where $\text{Int}(x)$ is the integer part of x .

In the slow-start phase, TCP updates the window following each ACK by

$$CWND += MSS$$

The algorithm leaves slow-start when $CWND > SSThresh$ (note the strict inequality, which is implementation dependent rather than a property of the TCP spec.). When undergoing a Fast Retransmit, assume the TCP uses the “advanced” version discussed in the congestion control lecture (slide 38).

In the answers on the next page, denote the packet containing bytes 1401 to 1500 as the 15th MSS, and denote the ACK expecting the next byte of 15001 as an ACK for the 16th MSS.

Consider a TCP connection with a $CWND$ of 12000 bytes and then undergoes a timeout. Assume all data up to 16000 have been ACKed (i.e., the last ACK's expected next byte was 16001). What is the value of $CWIN$, and which packets does TCP send, at each step in the following process? The first two lines are filled out, to indicate the required form of the answers.

<u>ACK received or timeout</u>	<u>cwin</u>	<u>Packets sent (MSS #'s)</u>
Timeout	1000	17
ACK for 17 th MSS	2000	18, 19
ACK for 18 th MSS:	3000	20, 21
ACK for 19 th MSS:	4000	22, 23
ACK for 20 th MSS:	5000	24, 25
ACK for 21 st MSS:	6000	26, 27
ACK for 22 nd packet:	7000	28, 29
ACK for 23 rd packet:	7142	30
ACK for 24 th packet:	7284	31
ACK for 25 th packet:	7426	32
ACK for 26 th packet:	7568	33
ACK for 27 th packet:	7710	34
ACK for 28 th packet?	7852	35
ACK for 29 th packet:	7994	36
ACK for 30 th packet:	8136	37, 38
ACK for 31 st packet:	8261	39
ACK for 31 st packet?	8261	
ACK for 31 st packet:	8261	
ACK for 31 st packet:	7130	32
ACK for 31 st packet:	8130	
ACK for 31 st packet?	9130	40
ACK for 31 st packet:	10130	41
ACK for 31 st packet:	11130	42
ACK for 39 th packet?	4130	43

4. HTTP and Caching

i) Consider a case where a client A is retrieving files F and G from web site B. F and G are both 125KB (i.e., one megabit). The RTT between A and B is 10msec (note, these are round-trip-times, not one-way latencies), and the bandwidth between the sites is 10Mbps. Assume all TCP SYN/ACK packets and HTTP request packets are negligible in size. How long does it take A to retrieve both files under the following circumstances:

Sequential (one-at-a-time) requests with nonpersistent TCP connections: **240**_

Concurrent requests with nonpersistent TCP connections: **220**_

Sequential requests within a single persistent TCP connection: **230**_

Pipelined requests within a single persistent TCP connection: **220**_

ii) Consider the same situation as in i), but assume that rather than a dedicated link there is a large shared link with many flows traversing it, and each TCP connection gets 10Mbps (adding additional flows does not significantly change the bandwidth per TCP connection, because there are thousands of flows on the link). Now, how long does it take A to retrieve both files under the following circumstances:

Sequential (one-at-a-time) requests with nonpersistent TCP connections: **240**_

Concurrent requests with nonpersistent TCP connections: **120**

Sequential requests within a single persistent TCP connection: **230**

Pipelined requests within a single persistent TCP connection: **220**

iii) Consider the same situation as in i), except that A is only downloading file F and there is now a cache C between A and B. All requests from A to B go through cache C, and assume the bandwidth along the path from A to C is 1gbps and the RTT between A and C is negligible, while the bandwidth along the path from C to B is 10mbps with an RTT of 10msec. Note, these are round-trip-times, not one-way latencies. As above, assume that the file is 125KB (i.e., one megabit) and that all TCP SYN/ACK packets and HTTP request packets are negligible in size.

Assume the cache operates as follows: (where the origin server refers to the site named in the URL)

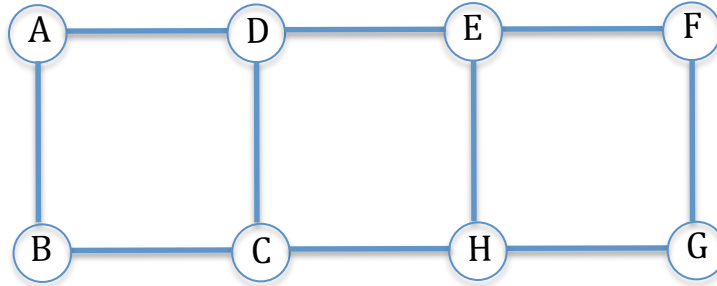
- If the object is not in the cache, the request is forwarded to the origin server
- If the object is in the cache, and the cache entry has not timed out (i.e., the cache TTL has not expired), the object is returned to the client
- If the object is in the cache, but the cache entry has timed out, the cache issues a conditional-GET to the origin server, asking if the object has changed since this object was cached: if the origin server responds that it hasn't, the cache returns the cached object, otherwise the origin server responds with the updated object which the cache forwards to the client.

How long does it take for A to receive the file under the following circumstances:

- The file is not in the cache. 121
- The file is in the cache and the TTL has not expired. 1
- The file is in the cache, the TTL has expired, but the file has not been changed. 21
- The file is in the cache, the TTL has expired, and the file has changed. 121

5. BGP Policy Extravaganza

i) Length-based policies:



Consider the interdomain connectivity graph above, where each of the ASes A through H are running BGP. Every domain advertises its chosen routes to all its neighbors (i.e., there is no limiting export policy). Assume that in this case all the domains want shortest paths (as measured in terms of the number of ASes traversed) and, when there is a tie, they prefer to route advertised with the letter closest to the beginning of the alphabet.

What path do packets take from A to G? **A-B-C-H-G**

What path do packets take from G to A? **G-F-E-D-A**

ii) Export policies: Using the same graph, now assume that nodes only advertise routes to the vertically connected node and a horizontal peer if they have one: assume that A-D are peers, E-F are peers, and C-H are peers. When advertising routes, nodes either advertise all or none. When selecting a route, ASes prefer the shortest path and, when there is a tie, they prefer to route advertised with the letter closest to the beginning of the alphabet.

What path do packets take from A to G? **A-D-C-H-E-F-G**

What path do packets take from G to A? **G-F-E-H-C-D-A**

iii) Nonstandard policies: Using the same graph, now assume that each domain's route preferences can be expressed in terms of their preferences over their neighbors (i.e., a domain prefers any route coming from one neighbor over any route coming from another neighbor, no matter what their relative lengths are). Assume the preferences are as follows (where $x > y$ means the domain prefers routes from x over routes from y).

Domain A: D>B

Domain B: A>C

Domain C: H>B>D

Domain D: C>E>A

Domain E: H>F>D

Domain F: E>G

Domain G: F>H
Domain H: C>E>G

What path do packets take from A to G? **A-D-C-H-G**
What path do packets take from G to A? **G-F-E-H-C-B-A**

Note: there is not a unique right answer to this question: BGP could reach several different fixed points. All you need to do is give an answer that is stable; that is, your answer should reflect a set of routes that, if BGP were to start with these routes as its initial condition, BGP would continue to retain that set of routes.