Homework Assignment #1 Solutions

EE122: Introduction to Communication Networks

(Fall 2007)

Department of Electrical Engineering and Computer Sciences
College of Engineering
University of California, Berkeley

Vern Paxson / Jorge Ortiz / Lisa Fowler / Daniel Killebrew

1. **Kurose & Ross**
   Chapter 1, p. 71: P8, P18, and P24.
   Chapter 4, p. 422: P11, P14, and P16.

Chapter 1, P8

(a) **[5 points]** With circuit switching you can support up to 10^4 users, since you need to allocate 100 kbps of capacity for each user, and 1 Gbps / 100 kbps = 10^4.

(b) **[5 points]** Given M total users, each using the network at a given point of time with probability p, then the probability that exactly K are using the network at a given time is:

$$\binom{M}{K} p^K (1-p)^{M-K}$$

Therefore, the probability that *at least* $N+1$ users are active at the same time is given by:

$$\sum_{i=N+1}^{M} \binom{M}{i} p^i (1-p)^{M-i}.$$

Chapter 1, P18

(a) **[2 points]** The delay is:

$$\frac{10^4\text{km} \cdot 10^3\text{m/km}}{2.5 \cdot 10^8\text{m/s}} = 0.04\text{s}.$$

Therefore the bandwidth-delay product is:

$$1\text{Mbps} \cdot 0.04\text{s} = 40,000 \text{ bits}.$$

(b) **[2 points]** The link can carry data at 1 Mbps. Consider a single bit propagating over the link. By the time it reaches the other end, 40 msec have elapsed. By the time it exits, another 1 Mbps · 0.040s = 40,000 bits have entered the link, so at any given time the link can hold at most 40,000 bits.

(c) **[2 points]** The bandwidth-delay product of a link is the maximum number of bits that can be in the link. We will see later that this generalizes to multi-hop paths.

(d) **[2 points]** If 40,000 bits completely fill a link that is 10,000 km long, then during propagation each bit spans 10,000 km / 40,000 bits = 0.25 km = 250 m. A football field is 100 yds $\approx$ 90 m, so a bit is longer.

(e) **[2 points]** The total number of bits a link can carry at one time is the bandwidth-delay product. The delay of a link is its length m divided by the propagation speed s, so the bandwidth delay product is $R \cdot \frac{m}{s}$.
The width of each bit is the length of the link divided by the number of bits it can carry, so

$$\frac{m}{R \cdot \frac{m}{s}} = s/R.$$

Chapter 1, P24

(a) **[2 points]** Time to send message from source to first packet switch = $\frac{7.5x10^6}{1.5x10^6}$ sec = 5 sec. With store-and-forward switching, the total time to move a message from source host to destination host = 5 sec x 3 hops = 15 sec.

(b) **[2 points]** Time to send first packet from source to first packet switch = $\frac{1.5x10^3}{1.5x10^6}$ sec = 1 ms. Time at which second packet is received at the first switch = 2 x 1 ms = 2 ms.

(c) **[3 points]** Time at which first packet is received at the destination host = 1 ms x 3 hops = 3 ms. After this, every 1 ms one packet will be received; thus the time at which last (5000^{th}) packet is received = 3 ms + $4999 * 1$ ms = 5.002 ms. It can be seen that delay in using message segmentation is significantly less (almost $1/3^{rd}$).

(d) **[3 points]** Drawbacks:

i. Packets have to be put in sequence at the destination.

ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

Chapter 4, P11 [10 points]

For Subnet 1 you need at least 125 addresses, so you could allocate 223.1.17.0/25.

Subnet 2 needs at least 60 addresses, so you could allocate 223.1.17.128/26.

Finally for Subnet 3, where you also need at least 60 address, you could assign 223.1.17.192/26.

Another possible set of answers are 223.1.17.128/25 for Subnet 1; 223.1.17.0/26 Subnet 2; and 223.1.17.64/26 for Subnet 3.

Chapter 4, P14 [10 points]

We need to include the first 26 bits of 101.101.101.64, so any address from 101.101.101.65 to 101.101.101.127 is a valid address in this network. The four equal-sized subnets that span it are 101.101.128.0/19, 101.101.160.0/19, 101.101.192.0/19, 101.101.224.0/19.

Chapter 4, P16 [10 points]

The maximum size of data in each fragment is 480 bytes (500 bytes total minus 20-byte IP header). Thus, the number of required fragments = $\lceil \frac{3000-20}{500-20} \rceil$ = 7.

The reason you also subtract 20 from 3000 is because the problem mentions that it's a 3000 byte *datagram* (so it includes an IP header), not that there's 3000 bytes of data to send.

Each fragment will have an Identification number 422. Each fragment except the last one will be of size 500 bytes (including the IP header). The amount of data carried in the last datagram will be (3000-20) - 6 x 480 = 2980 - 2880 = 100 bytes. It will require a 20-byte IP header, so it will total 120 bytes in all.

The offsets of the 7 fragments will be 0, 60, 120, 180, 240, 300, 360 (each offset reflecting another 480 bytes, but expressed in 8-byte units).

Each of the first 6 fragments will have the **More Fragments** flag set to 1; the last fragment will have it set to 0.

*Note #1: the slides presented in lecture to illustrate fragmentation are **wrong**, for which Prof. Paxson offers his apologies :-(. Because of this, we also accept answers computed without factoring in the IP header sizes, namely that each fragment except the last carries 496 bytes (needs to be a multiple of 8), and the last holds* $3000 - 496 \cdot 6 = 24$ *bytes.*

Note #2: when fragmenting an oversized datagram, an IP node is in fact free to split it up as it wishes. So if you noted this fact and used a different allocation than those given above, that was acceptable.

2. **File Transfer**

Calculate the total time to transmit a 1500 KB file over a link in the following cases, assuming the one way delay in either direction is 40ms, and an initial RTT of "handshaking" before any data is sent. (Note: 1 KB = 2^{10} bytes, 1 Mbps = 10^6 bits/s).

(a) **[10 points]** The bandwidth is 1 Mbps, the packet size including the header is 1 KB of which the header is 40 bytes, and the data packets are sent continuously and never lost.

Answer: Given that the maximum size of the packet is 2^{10} bytes, we can only fit $2^{10} - 40$ bytes in the payload. Therefore we need to send $\lceil \frac{1500 * 2^{10}}{2^{10}-40} \rceil$ packets, which is 1,561 packets. (Note, the last packet is less than 1 KB in size.)

We then have the following:

$\underbrace{(1560 * 2^{10})}_{TotalFullPacket} \underbrace{+(40 + 960)}_{LastPacket}$ = 1,598,440 bytes to send = 12,787,520 bits.

With the bandwidth set at 10^6 bits/sec, the transmission time of the data is:

$$\frac{12787520}{10^6} = 12.78752 \text{ sec.}$$

But we're not done: we have to add the RTT of handshaking and the one-way delay, so the total time is $12.78752 + 0.08 + 0.04$ sec = 12.90752 sec (for which it's reasonable to round up to 12.91 sec).

(b) Same as (1), but after we finish transmitting a packet we must wait for one RTT before transmitting the next packet.

Answer: [5 points] The number of packets we wish to send is the same as in part (a), except now we wait an RTT for each packet that we send. This leaves us with the following:

$$\underbrace{\frac{1560 * 2^{10} * 8}{10^6}}_{TotalFullPacket} + \underbrace{\frac{1000 * 8}{10^6}}_{LastPacket} + \underbrace{1561 * 0.08}_{RTT/packet} + \underbrace{0.08}_{Handshake} = 137.74752 \text{ sec}$$

(c) Same as (1), but the link bandwidth is "infinite" (the transmission time is assumed to be zero). We start by sending one packet in the first RTT (2^0), during the second RTT we send two (2^1) packets, during the third RTT we send four (2^2), and so on. (Any idea why we might want to do something like this?)

Answer: [5 points] The main idea here to count the minimum number of terms in the following summation: $\sum_{i=0}^{n} 2^i$ that yields a sum ≥ 1561.

When expanded we see that there are 11 terms. (We could also get this directly as $\lceil \log_2 1561 \rceil$.) This means that the total amount of time to send the file is the initial handshake, plus the 11 RTTs to send the groups, plus a final one-way propagation for the last group to reach the destination: $0.08 + 11 * 0.08 + 0.04 = 1.0$ sec.

(Note: we also accepted an answer of 0.96 sec which left out the final 40 ms of propagation time.)

This scheme might be used when we want to maximize the number of packets in flight but we are trying to determine how much the network and receiving host(s) could handle (which it turns out TCP does when starting up).

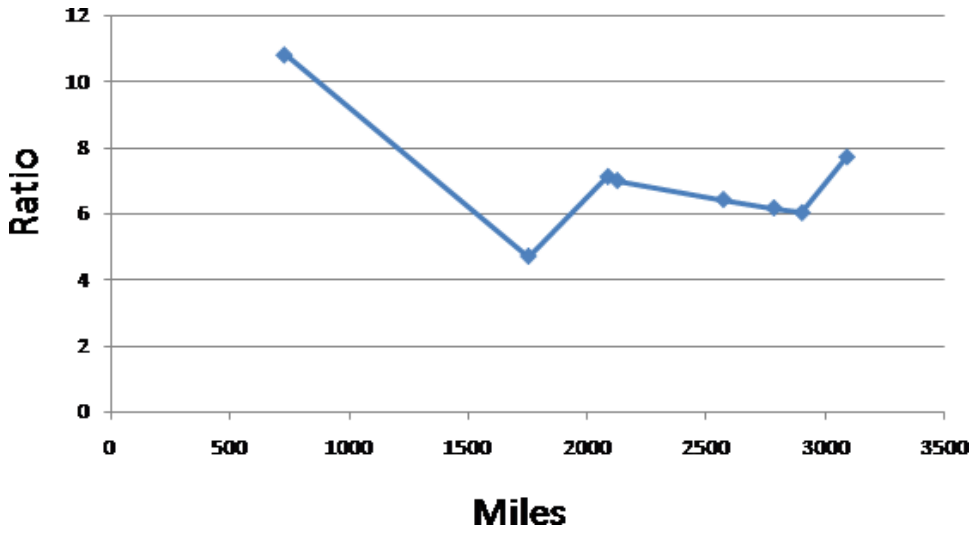
3. **Ping**

The ping program determines the round-trip-time (RTT) to any host on the Internet. Using a computer on campus, ping the following hosts: mit.edu (Cambridge, MA), cornell.edu (Ithaca, NY), princeton.edu (Princeton, NJ), cmu.edu (Pittsburgh, PA), wisc.edu (Madison, WI), uchicago.edu (Chicago, IL), utexas.edu (Austin, TX), utah.edu (Salt Lake City, UT). For each of these locations find the physical distance from Berkeley using google maps (maps.google.com).

(a) Plot a graph (using your favorite plotting program like gnuplot or excel) where the X-axis represents the distance to each city, and the Y-axis represents the ratio between the RTT as measured by the ping program and the shortest possible time T along the driving route returned by Google maps.

[3 points] Ping data
[3 points] Ratio calculation and graph

Dest	Google Distance (mi)	Ping RTT (ms)	Light T (ms)	Ratio
mit.edu	3086	127.66	16.566	7.706
cornell.edu	2780	91.84	14.924	6.154
princeton.edu	2898	93.49	15.557	6.001
cmu.edu	2568	88.36	13.786	6.4009
wisc.edu	2084	79.52	11.187	7.107
uchicago.edu	2131	78.18	11.187	6.988
utexas.edu	1750	44.01	9.394	4.684
utah.edu	726	42.13	3.897	10.81



(b) **[4 points]** 2 points per reason.
Give two reasons why the Y-values you plot are larger than 2.

Answer: At least two of the following:

- Packets might take routes that are longer (less direct) than the shortest path.
- Packets can be delayed due to queuing (waiting behind other packets for transmission).
- Packet can travel along links for which the propogation speed is slower than that of light in a vacuum.
- Packets will encounter store-and-forward delays. (They have to entirely arrive at a given hop before they can proceed with transmission to the next hop).
- Packets can traverse low-bandwidth links such that it takes considerable extra time for the full packet to transit the link.

4. **RFCs**

The Request for Comments (RFC) documents define and standardize the bulk of the protocols used in the Internet. They are published on behalf of the Internet Engineering Task Force (IETF), which

forms the main standards body of the Internet. If you want to understand a protocol in full detail (especially to write your own implementation of it), these are the documents to refer to.

(a) In Bellovin's April Fool's RFC, he describes a method for identifying malicious packets in the network and the actions that a system should take if this packet type is identified.

Search http://rfc-editor.org to locate this document

Once you find the document, read it and answering the following:

i. **[1 point]** How many IP header bits does the mechanism require?

Answer: The mechanism only requires 1 header bit.

ii. **[1 point]** How does an end-system react to the settings?

Answer: If the end-system is an attacker, it may use the given API to request that the Evil bit be set in outgoing packets. Multi-level insecure operating systems must set the Evil bit for all outgoing packets. Also, individual fragments that are malicious by themselves must have the Evil bit set.

End-system hosts should process the packet with the evil bit set according to their nature. This behavior is operating-system dependent. When devices receive a packet with the Evil bit set, they must drop the packet. If the Evil bit is not set, the packet must not be dropped.

(b) A great strength of the IP protocol is how it provides the "glue" to communicate data over a series of widely varying, lower-layer network ("link") technologies. Using http://rfc-editor.org (or whatever search means you want to use, providing you work by yourself), locate RFCs that refer to transmitting IP over <XXXX>, where <XXXX> is some link technology. For example, RFC 201 is "Transmitting IP traffic over ARCNET networks".

List all the different lower-layer networking technologies for which you can find an RFC whose title indicates it defines how to send IP packets (or datagrams, or traffic) over that technology. Give both the name of the technology and the RFC defining the transmission. Can you find more than 20?

[8 points]

RFC	Title
877	A Standard for the Transmission of IP Datagrams Over Public Data Networks
894	A Standard for the Transmission of IP Datagrams Over Ethernet Networks
895	A Standard for the Transmission of IP Datagrams Over Experimental Ethernet Networks
1042	A Standard for the Transmission of IP Datagrams Over IEEE 802 Networks
1044	Internet Protocol on Network System's HYPERchannel Protocol Specification
1055	A NONSTANDARD FOR TRANSMISSION OF IP DATAQGRAMS OVER SERIAL LINES: SLIP
1088	A Standard for the Transmission of IP Datagrams over NetBIOS Networks
1209	The Transmission of IP Datagrams over the SMDS Service
1390	Transmission of IP and ARP over FDDI Networks
1577	Classical IP and ARP over ATM
2023	IP Version 6 over PPP
2067	IP over HIPPI
2470	Transmission of IPv6 Packets over Token Ring Networks
2491	IPv6 over Non-Broadcast Multiple Access (NBMA) networks
2590	Transmission of IPv6 Packets over Frame Relay Networks Specification
2625	IP and ARP over Fibre Channel
2734	IPv4 over IEEE 1394
3572	Internet Protocol Version 6 over MAPOS (Multiple Access Protocol Over SONET/SDH)
3717	IP over Optical Networks: A Framework
4259	A Framework for Transmission of IP Datagrams over MPEG-2 Networks
4391	Transmission of IP over InfiniBand (IPoIB)