Solutions for Homework #1

EE122: Introduction to Communication Networks

(Fall 2006)

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

Vern Paxson / Sukun Kim / Dilip Antony Joseph

1. Problems from Peterson & Davie.

   (a) **Exercise 1.13** (11 points)
   The width of a bit:

   $$\frac{1\text{ bit}}{1\text{ Gbit/s}} = 1\text{ ns} \qquad \text{(5 points)}$$

   The length of a bit is equal to the distance the signal representing it propagates in the given medium (copper wire, in this case) during the time it takes to transmit the bit:

   $$1\text{ ns} \times (2.3 \cdot 10^8\text{ m/s}) = 0.23\text{ m} \qquad \text{(6 points)}$$

   (b) **Exercise 1.15** (20 points)

   i. (5 points) Here the size of packet is assumed to be 0, so the RTT is twice the one-way propagation delay:

   $$\begin{aligned} RTT &= \text{(Propagation Delay)} \times 2 \\ &= (385,000\text{ km}/(3 \cdot 10^8\text{ m/s}))\ \times 2 \\ &= 2.57\text{ s.} \end{aligned}$$

   ii. (5 points) The delay-bandwidth product is:

   $$\begin{aligned} \text{delay } \times \text{ bandwidth} &= 2.57\text{ s} \times 100\text{ Mbit/s} \\ &= 257\text{ Mb} \\ &= 32.125\text{ MB} \end{aligned}$$

iii. (5 points) The large delay-bandwidth product means that filling the communication "pipe" (fully utilizing the link) requires a very large amount of data in flight simultaneously. We might also (or instead) express this that the path requires a very large amount of buffering to hold unacknowledged data.

iv. (5 points) Let us assume that one megabyte is 10^6 bytes. (Alternatively, using 1 MB $= 2^{20}$ bytes is also fine.) The total latency is the latency of sending the request plus latency of receiving the reply (i.e., receiving the data).
Given we assume the size of the request is very small, we ignore its transmission time. Therefore:

$$\begin{aligned}\text{Minimum Time Needed} &= \text{Latency of Request} + \text{Latency of Data}\\ &= \text{Propagation of Request} + \text{Transmission of Request}\\ &\quad + \text{Propagation of Data} + \text{Transmission of Data}\\ &= \frac{385,000\text{ km}}{3\cdot 10^8\text{ m/s}} + 0\text{ s} + \frac{385,000\text{ km}}{3\cdot 10^8\text{ m/s}} + \frac{25\text{ MB}}{100\text{ Mbit/s}}\\ &= 4.57\text{ s}\end{aligned}$$

(c) **Exercise 1.18** (21 points)

i. (7 points) Since we're using a store-and-forward switch, we can divide the path into the two links on either side of the switch, and we obtain the total latency by adding the latencies of two links:

$$\begin{aligned}\text{Total Latency} &= (\text{Latency at Link 1}) + (\text{Latency at Link 2})\\ &= 2\times(\text{Latency at Link 1})\\ &= 2\times((\text{Propagation at Link 1}) + (\text{Transmit at Link 1}))\\ &= 2\times\left[10\mu\text{s} + \frac{5000\text{ bits}}{10\text{ Mbit/s}}\right]\\ &= 1.020\text{ ms.}\end{aligned}$$

ii. (7 points) Similar to (a), we can divide the path into four links. In the textbook, it is not clear whether the latency between any two switches remains $10\mu s$, or if a new switch divides an existing link. Therefore, you could use either a total propagation delay of $20\mu s$ or $40\mu s$. Here, we assume the propagation delay between two switches remains the same ($10\mu s$):

$$\begin{aligned}\text{Total Latency} &= (\text{Latency at Link 1}) + (\text{Latency at Link 2}) +\\ &\quad (\text{Latency at Link 3}) + (\text{Latency at Link 4})\end{aligned}$$

$$
\begin{aligned}
&= 4 \times (\text{Latency at Link 1}) \\
&= 4 \times ((\text{Propagation at Link 1}) + (\text{Transmit at Link 1})) \\
&= 4 \times \left[10\mu\text{s} + \frac{5000 \text{ bits}}{10 \text{ Mbit/s}}\right] \\
&= 2.040 \text{ ms.}
\end{aligned}
$$

iii. (7 points) When using a cut-through switch, the switch introduces a transmission delay of 200 bits, but *the rest of the transmission delay overlaps with the transmission delay at the first link*:

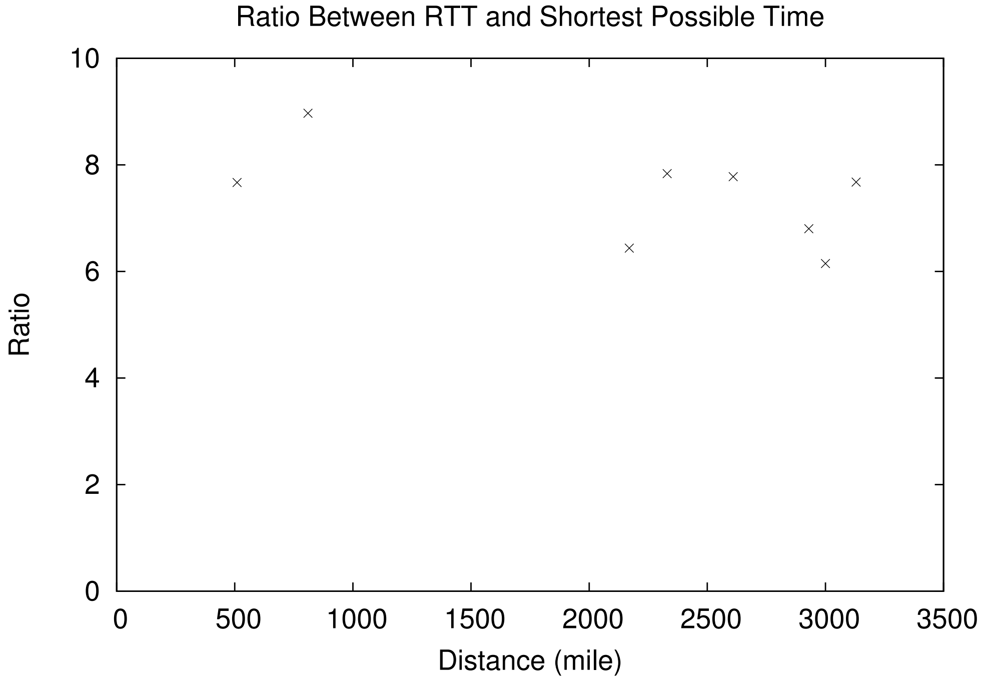
$$
\begin{aligned}
\text{Total Latency} &= (\text{Latency at Link 1}) + (\text{Latency at Link 2}) \\
&= ((\text{Propagation at Link 1}) + (\text{Transmit at Link 1})) \\
&\quad +((\text{Propagation at Link 2}) + (\text{Delay at Link 2})) \\
&= \left[10\mu\text{s} + \frac{5000 \text{ bits}}{10 \text{ Mbit/s}}\right] + \left[10\mu\text{s} + \frac{200 \text{ bits}}{10 \text{ Mbit/s}}\right] \\
&= 20\mu s + 500\mu s + 20\mu s \\
&= 540\mu s.
\end{aligned}
$$

2. **Ping** (16 points)

(a) (8 points) Here are the values we measured:

Host	**RTT**	**City**	**Distance**	**Shortest Possible Time**
cmu.edu	109 ms	Pittsburgh, PA	2,610 miles	14.0 ms
mit.edu	129 ms	Boston, MA	3,130 miles	16.8 ms
washington.edu	39 ms	Seattle, WA	810 miles	4.35 ms
ucsd.edu	21 ms	San Diego, CA	510 miles	2.74 ms
uchicago.edu	75 ms	Chicago, IL	2,170 miles	11.6 ms
columbia.edu	107 ms	New York, NY	2,930 miles	15.7 ms
odu.edu	99 ms	Norfolk, VA	3,000 miles	16.1 ms
www.vanderbilt.edu	98 ms	Nashville, TN	2,330 miles	12.5 ms

For a plot of the ratio of the times versus geographic distance, see the next page.



(b) (8 points) 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 queueing (waiting behind other packets for transmission).
- Packets can travel along links for which the propagation speed is slower than that of light-in-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.

3. **RFCs** (16 points)

   (a) (8 points) RFC 1149—*A Standard for the Transmission of IP Datagrams on Avian Carriers*
   The RFC states that the MTU (Maximum Transmission Unit) varies among carriers. Generally, as the carrier gets older, the MTU also increases (as the bird grows). A typical MTU value is 256 mg (weight, rather than bits!).

   (b) (8 points) Here are the RFCs we readily found:

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 DATAGRAMS 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)*

   Extra Credit for finding appropriate RFCs not listed above: 4 points

4. **End-to-End Arguments** (16 points)

(a) (8 points) The probabilities for a package to make it pass stages 1–7 are 97%, 98%, 99%, 98%, 99%, 98%, and 97%, respectively. For a package to arrive at the destination, it must pass every stage successfully:

$$\begin{aligned} P(\text{arrives at destination}) &= \prod_{i=1}^{7} P(\text{passes through stage } i) \\ &= 0.97 \cdot 0.98 \cdot 0.99 \cdot 0.98 \cdot 0.99 \cdot 0.98 \cdot 0.97 \\ &\approx 87\% \end{aligned}$$

Note: it's possible to instead interpret the wording of this problem as stating that *in total* the probability of losses at the different types of stages are 3%, 2%, and 1%, and thus:

$$\begin{aligned} P(\text{arrives at destination}) &= \prod_{i=1}^{3} P(\text{passes through all stages of type } i) \\ &= 0.97 \cdot 0.98 \cdot 0.99 \\ &\approx 94\% \end{aligned}$$

(b) (8 points) If a functionality can be implemented at end hosts, do not implement it within the lower-level network unless it is both a performance enhancement *and* does not impose a burden on applications that do not need the functionality.

   i. In the first case, the required functionality is reliable delivery. Customers (end hosts) can implement this functionality themselves by repeatedly attempting delivery in the face of apparent failure. Therefore, BearsEx (lower level network) should not spend money to implement reliable delivery. (4 points)

   ii. In the second case, the required functionality is quick delivery the next day (low latency). The customers (end hosts) cannot implement this functionality themselves, since they lack control over the speed with which packages are delivered. Therefore, BearsEx should spend money to better achieve the required functionality. (4 points)