Prof. Athina Markopoulou
Fall 2016

Intro to Computer Networks
EECS148/CompSci132

Homework 1 - Solution Sketch

Topics: Packet vs. Message vs. Circuit Switching; Delay and Throughput.

Problem 1

(a) $\frac{3Mbps}{150kbps} = 20$ users can be supported using circuit switching.

(b) A user is transmitting 10% of the time or w.p. $p = 0.1$ at a given time.

(c) $Pr\{n$ out of 120 users transmit at a given time$\} = \binom{120}{n} p^n (1-p)^{120-n}$.

(d) $Pr\{21$ or more users are active at the same time$\} = 1 - \sum_{n=0}^{20} \binom{120}{n} p^n (1-p)^{120-n} \simeq 0.003$.

You can compute the above number using a program or calculator. Alternatively, you can use the central limit theorem to approximate this probability.

Problem 2

1. **Circuit Switching.**

The transmission and propagation delay of a single message of H bits over a single link is $H/W + \tau$. In order to establish a circuit we need the `Disconnect Request` and the `Connect Accepted` messages to cross k links each to reach the destination and source, expectively. This takes $2k(H/W + \tau)$ time. Then the message is transmitted over the circuit, which uses only a fraction of the speed, $\frac{W}{n}$, i.e, the transmission time of the message is nL/W and the propagation from the source to the destination is the same $k\tau$. After the sender finishes the transmission of the message, it sends a `Disconnect` signaling message over the circuit, which takes $nH/W + k\tau$ to be completely received at the destination. It takes another $nH/W + k\tau$ time for the `Disconnect Accepted` message to reach the source. At that point (t_2), the source can create another circuit to send another message, if it so desires. Figure 1 shows this scenario.

The end-to-end delay includes the time to establish the circuit plus the time to transmit the message plus the propagation delay of the message; according to the definition, it does not include the disconnect delay. So, the message delay is the following:

$$D_{CS} = t_1 - t_0 = 2k(\frac{H}{W} + \tau) + n\frac{L}{W} + k\tau \tag{1}$$

The time it took the source to send L useful bits before it can considering the next message is

$$t_2 - t_0 = 2k(\frac{H}{W} + \tau) + \frac{L}{\frac{W}{n}} + 2(\frac{H}{\frac{W}{n}} + k\tau) = \frac{2(n+k)H + nL}{W} + 4k\tau$$

Therefore, the throughput (in bits per sec) is:

$$T_{CS} = \frac{L}{t_2 - t_0} = \frac{L}{2(n+k)\frac{H}{W} + n\frac{L}{W} + 4k\tau} \tag{2}$$

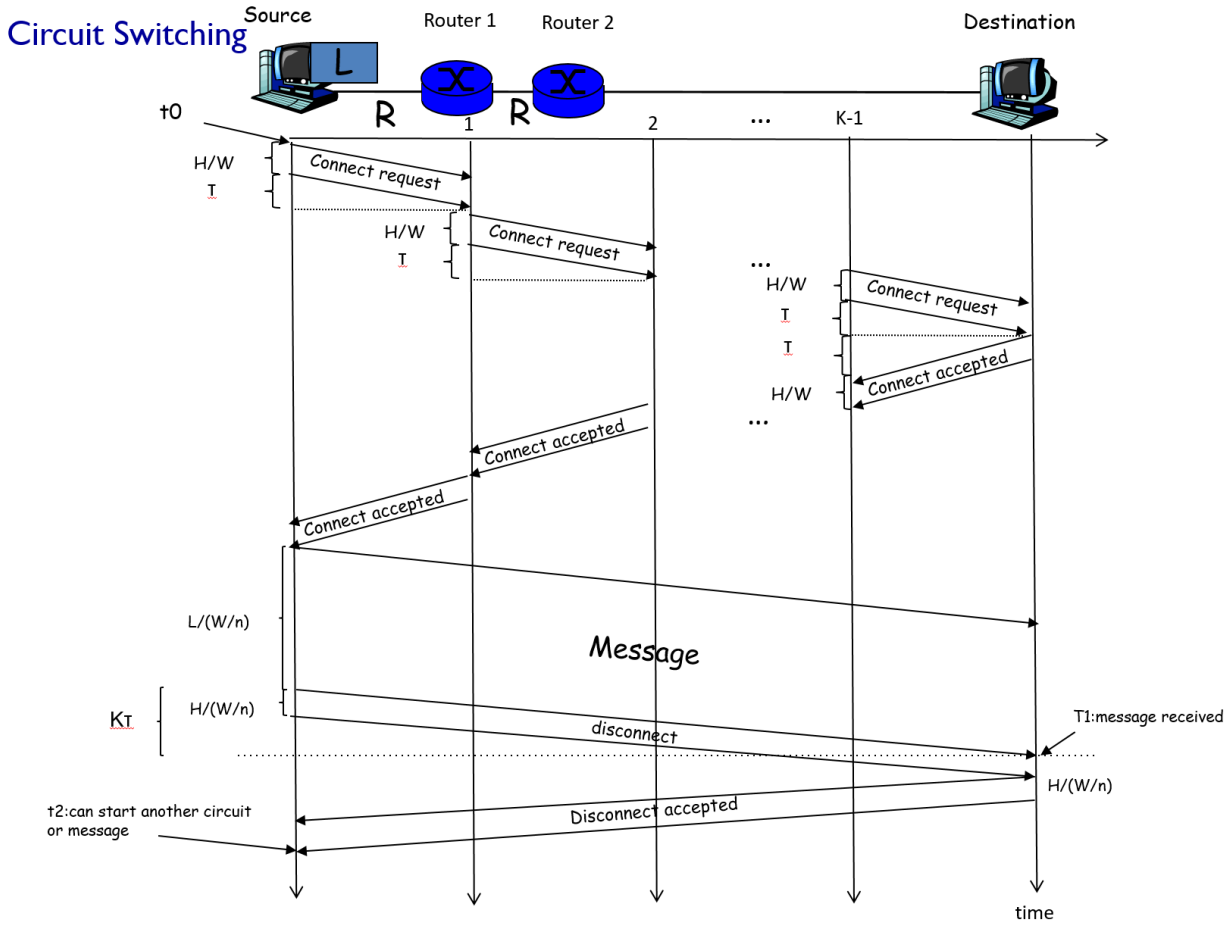

Figure 1: Circuit Switching

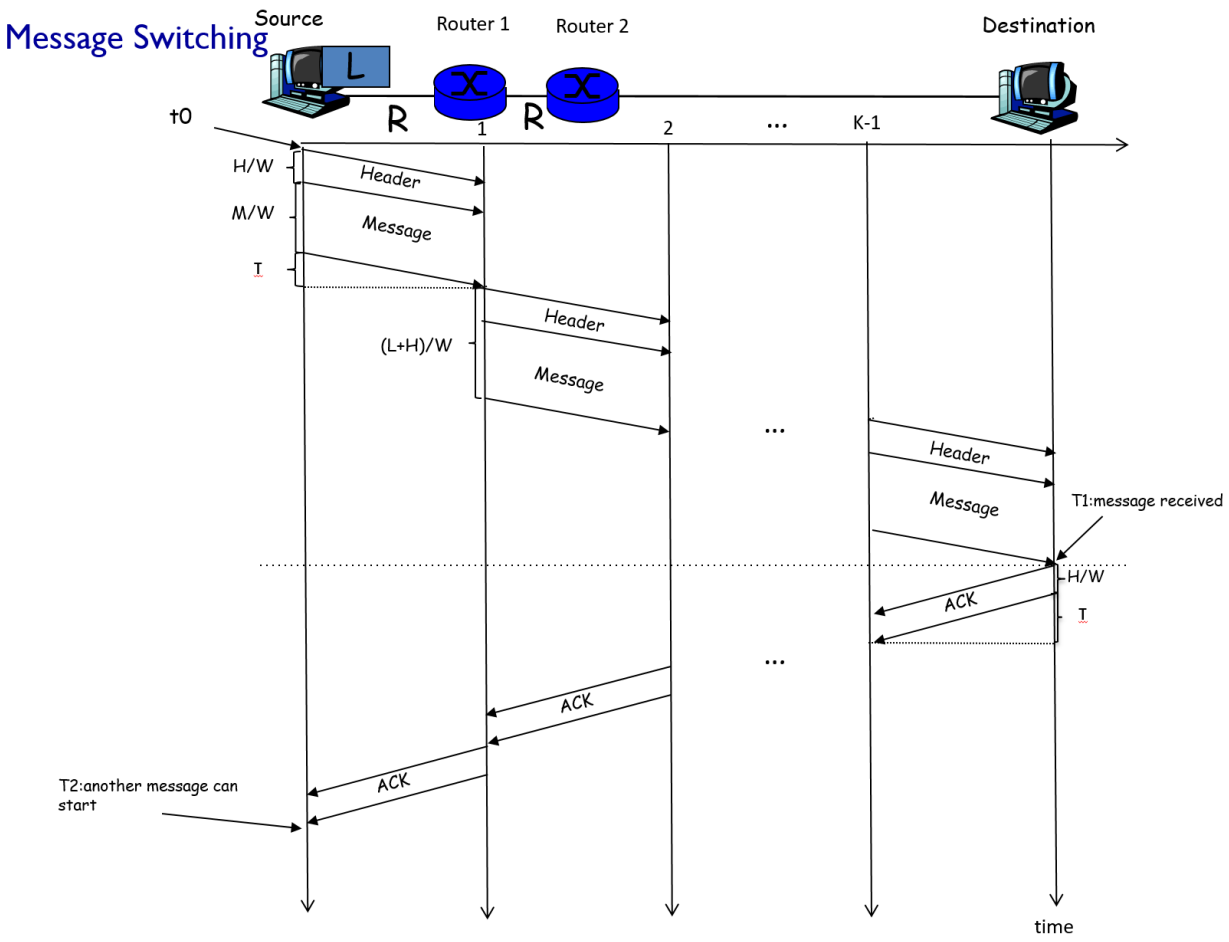

Figure 2: Message Switching

2. **Message Switching.**

   The time for the transmission and propagation of the message and its header across a single link is $\frac{L+H}{W} + \tau$. So the end-to-end delay across k such hops is

$$D_{MS} = t_1 - t_0 = k\left(\frac{L+H}{W} + \tau\right) \tag{3}$$

   Upon successful reception of the first message at the receiver, the destination sends an ACK, which takes $k(H/W + \tau)$ to be received by the source. At that time, t_2, the source can start transmitting the second message. Figure2 shows this scenario. Therefore, the source transmits L useful bits every $t_2 - t_0$ time:

$$T_{MS} = \frac{L}{t_2 - t_0} = \frac{L}{2k\frac{H}{W} + k\frac{L}{W} + 2k\tau} \tag{4}$$

3. **Packet Switching.**

   The message is split into $l = \frac{L}{P}$ messages, which is assumed to be an integer number. A header of H bits is added to the P bits of data, thus every packet is $(H + P)$ bits long. The packets are sent in a store-and-forward pipelined way as shown in the Figure 3.

Packet Switching

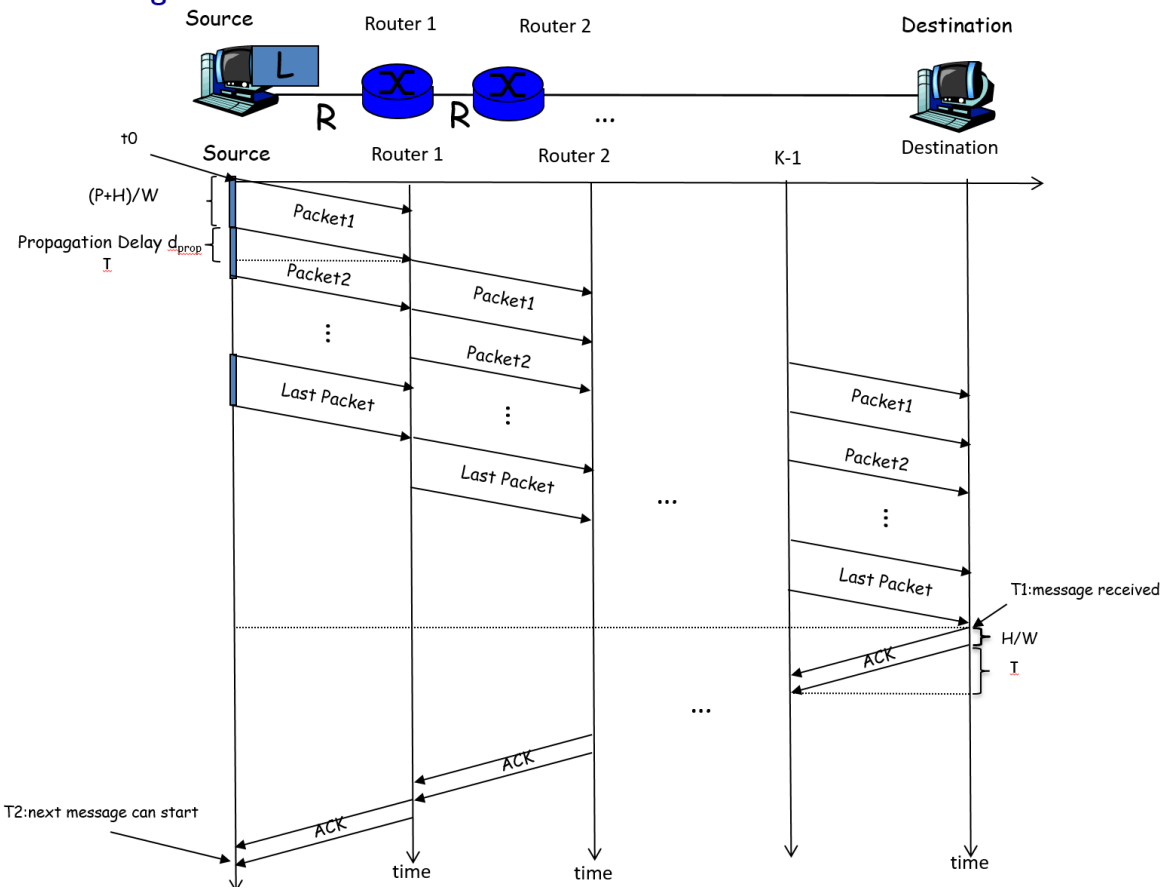

Figure 3: Packet Switching

The first packet is received at the first router after delay that consists of transmission $\frac{H+P}{W}$ and propagation τ across that link. The time it takes for the first packet to be received at the destination, i.e. after k hops, is $k(\frac{H+P}{W}+\tau)$. After the first packet is received, the rest of the $(\frac{L}{P}-1)$ packets are also received at the destination, each after transmission time $\frac{H+P}{W}$. Therefore the total end-to-end delay until the whole message is received at the destination is:

$$D_{PS} = t_1 - t_0 = k(\frac{H+P}{W}+\tau) + (\frac{L}{P}-1)\frac{H+P}{W} = (k+\frac{L}{P}-1)\frac{P+H}{W} + k\tau \qquad (5)$$

At that time (t_1), the destination sends an ACK, which received by the source at $t_2 = t_1 + k(\frac{H}{W}+\tau)$. After receiving the ACK, the source is ready to transmit the second message. Therefore the source sends L useful bits every $t_2 - t_0$ interval and the throughput is:

$$T_{PS} = \frac{L}{t_2 - t_0} = \frac{L}{(2k+\frac{L}{P}-1)\frac{H}{W} + (k+\frac{L}{P}-1)\frac{P}{W} + 2k\tau}$$

4. **Delay Comparison.**

   PS vs MS. $D_{PS} < D_{MS} \iff ....\frac{H}{P} < k-1$.[1] This is true for a large number of hops k and a large enough packet so that $H < P$: $\frac{H}{P} < 1 < k-1$. However it is not true when e.g., $k = 2$ and $H = P$. Take a moment to reflect upon the intuition behind this statement: when the message goes across many hops (k), packetization and pipelining help decrease the delay, as long as we don't overdo it, i.e. we don't break the message L into too small packets (P useful bits) compared to the packet header H (not useful from the source's point of view).

   MS vs CS. Similarly it is $D_{MS} < D_{CS} \iff ....\frac{k}{n} > \frac{L}{H+2\tau \cdot W-L}$. This is also a tradeoff involving pipelining and overhead (for MS) vs. the time to set up the circuit and the speed of the reserved circuit (for CS), but it is more difficult to interpret.

   Let us consider the simplifying, but reasonable, assumptions made in the problem: $\tau = 0, n = 1, H < P, k \to \infty$. Since k is large (thus $k >> \frac{L}{P} - 1$ in D_{PS}) the dominant delay terms are

$$D_{CS} = \frac{2kH + L}{W}$$

$$D_{MS} = \frac{kH + kL}{W}$$

$$D_{PS} = \frac{kH + kP}{W}$$

   Since $P < L$ always, we get $D_{MS} > D_{PS}$ in this case.

   The comparison between CS and MS still involves a tradeoff: $D_{CS} < D_{MS} \iff .....\frac{k}{k-1} < \frac{L}{H}$ which is <u>true</u> since for large k and $L >> H$: $\frac{k}{k-1} \simeq 1 << \frac{L}{H}$. Basically if the number of hops is large, storing and forward the whole message is too expensive and it is worth reserving a circuit, unless the overhead of storing-and-forwarding the signaling message (H bits) is comparable to the storing-and-forward of the message itself (L bits).

   CS vs PS. $D_{CS} > D_{PS} \iff .....L < k(P - H)$, which is <u>true</u> when $H < P, k \to \infty$.

   In summary, for $\tau = 0, n = 1, H < P, k \to \infty$ the order is the following:

$$D_{MS} > D_{CS} > D_{PS}$$

5. **Throughput Comparison.**

   MS vs PS. The denominator of the throughput is $t_2 - t_0 = (t_2 - t_1) + (t_1 - t_0)$. The second term $(t_1 - t_0)$ is the end-to-end delay we previously computed and showed that: $D_{PS} < D_{MS}$ (in general, if and only if $\frac{H}{P} < k-1$, but true for large k). The first term $(t_2 - t_1)$ is the time for the ACK to be received by the source, which is the same $t_2 - t_1 = k(\frac{H}{W} + \tau)$ for both PS and MS. Therefore, both schemes transmit the same number of useful bits (L): MS over time $t_2^{MS} - t_0 = D_{MS} + k(\frac{H}{W} + \tau)$, PS over time $t_2^{PS} - t_0 = D_{PS} + k(\frac{H}{W} + \tau)$. Therefore, it is $T_{PS} > T_{MS}$, as intuitively expected due to pipelining.

   MS vs CS. Following similar steps, for $\tau = 0, n = 1, k \to \infty, H < P$ we find that: $T_{CS} > T_{MS} \iff .....(k-1)\frac{L}{W} < 2\frac{H}{W} + k\tau$. This is intuitive: MS is better than CS if the store-and-forward transmission of the entire message in k intermediate hops takes less time than the propagation+transmission for setting up the circuit.

[1] Just substitute the two formulas for D_{PS} and D_{MS} into $D_{PS} < D_{MS}$ and do algebra. We omit the intermediate steps and show only the the simplified equivalent statement at the end. It is easy to asses whether this final statement is true or false, which also implies that the original statement is true or false.

In summary, for $\tau = 0, n = 0, k \to \infty, P < H$, the order is the following:

$$T_{CS} < T_{MS} < T_{PS}$$

6. **Choosing the Packet Size to Minimize Delay.**

We already computed the end-to-end delay of packet switching, repeated here:

$$D_{PS}(P) = (k + \frac{L}{P} - 1)\frac{P+H}{W} + k\tau = (k-1)\frac{P}{W} + \frac{L}{P}\frac{H}{W} + (k-1)\frac{H}{W} + \frac{L}{W} + k\tau$$

Since other variables are fixed, this is a function of a single variable P. You can minimize it by taking the first derivative and finding the optimal packet size P^* that makes the derivative zero.[2]

$$\frac{dD}{dP} = \frac{k-1}{W} - \frac{LH}{WP^2} = 0 \Longleftrightarrow P = P^* = \sqrt{\frac{LH}{k-1}}$$

Alternatively, if you are not comfortable taking derivatives, you could simply plot $D_{PS}(P)$ as a function of P for fixed values of the other parameters. You would then see that it has a nice convex shape and a minimum at the value P^* computed above.

Finally, this value $P^* = \sqrt{\frac{LH}{k-1}}$ makes intuitive sense. If there is only one hop, $k = 1$, there is no benefit from packetization and pipelining: the larger the P the better, i.e., $P^* = L$. However, if $k > 1$, the packet size should be chosen to balance the tradeoff between paying overhead H multiple times (once per packet), but also saving time from pipelining multiple times (once at each of the $k - 1$ intermediate hops).

[2]You can also easily check that $\frac{d^2D}{dP^2} > 0$, therefore the function is convex and $D_{PS}(P^*)$ is indeed a minimum.