CS/ECE 438: Communication Networks **Fall 2007**
Problem Set 2 **Solutions**

1. **Ethernet Timing**

   This problem is about the Ethernet/IEEE 802.3 access protocol. To be definite, suppose that if a host detects a transmission while it is transmitting a frame, then: (i) if the host has already transmitted the 64-bit preamble, the host stops transmitting the frame and sends a 32-bit jamming sequence; (ii) else the host finishes transmitting the 64-bit preamble and then sends a 32-bit jamming sequence. Suppose the packets are 512 bits long, which is the minimum length allowed. Hosts A and B are the only active hosts on a 10 Mbps Ethernet and the propagation time between them is 25 μs, or 250 bit durations. Suppose A begins transmitting a frame at time $t = 0$, and just before the beginning of the frame reaches B, B begins sending a frame, and then almost immediately B detects a collision. NOTE: For simplicity, please ignore the 9.6 μs inter-frame wait time in the Ethernet protocol.

   (a) Does A finish transmitting the frame before it detects that there was a collision? Explain. *No, it does not. A's transmission reaches B at 25 μs, and the beginning of the frame from B arrives back at A at 50 μs. A does not finish transmitting its own frame until 51.2s μs, hence it will detect a collision before the frame is done.*

   (b) What time does A finish sending the jamming signal? What time does B finish sending the jamming signal? *A hears a collision at 50 μsand then jams for 3.2 μs, for a total of 53.2 μs. B hears the collision at 25 μsbut has just started sending the preamble, so it sends a total of 96 bits and finishes at 34.6 μs.*

   (c) What time does A hear an idle channel again? What time does B first hear an idle channel again? *A hears an idle channel 25 μsafter B finishes transmitting, at 59.6 μs. Likewise, B hears an idle channel 25 μsafter A finishes transmitting, at 78.2 μs.*

   (d) Suppose each host next decides to retransmit immediately after hearing the channel idle. After the resulting (second) collision: When does A next hear the channel idle? When does B next hear the channel idle? *A transmits at 59.6 μs, and B hears this transmission at 84.6 μs. B itself started transmitting at 78.2 μsand so it has just finished transmitting the preamble, so it jams for another 3.2 μs. A hears an idle channel at 84.6 + 3.2 + 25 = 112.8 μs. On the other hand, A hears B's transmission at 78.2 + 25 = 103.2 μsand jams until 106.4 μs. B hears the channel as idle at 106.4 + 25 = 131.4 μs.*

   (e) Suppose after the second collision, A decides to wait 512 bit durations to retransmit (if it hears silence after that long) and B decides to retransmit immediately after hearing a silent channel. Is the transmission of host B successful? *A is waiting 512 bit durations, so it will not start transmitting until 112.8 + 51.2 = 164 μs. B transmits at 131.4 μs, and the signal reaches A at 156.4 μs, while A is still waiting. Therefore, B's transmission is successful.*

2. **Multiple Access**

   Suppose nodes A and B are ready to send a packet at the same time a third node ends transmission on a 10 Mbps Ethernet. In the ith round after $i-1$ collisions have already occurred, the two nodes wait $0, 1, \ldots, 2^{i-1}-1$ slots until the next attempt, all 2^{i-1} choices having equal probability.

   (a) Find the probability q_i of a collision in the ith round, given that there are collisions in the previous $i-1$ rounds (i.e. $q_1 = 1, q_2 = 1/2$), for all $i \geq 1$.

   $$q_i 1/2^{i-1}$$

   (b) Find the probability p_i that exactly i rounds are needed for the first success, and compute $p_1, p_2, \ldots, p_4$.

   $$p_i = \left(\prod_{j=1}^{i-1} q_j \right) \cdot (1 - q_i)$$

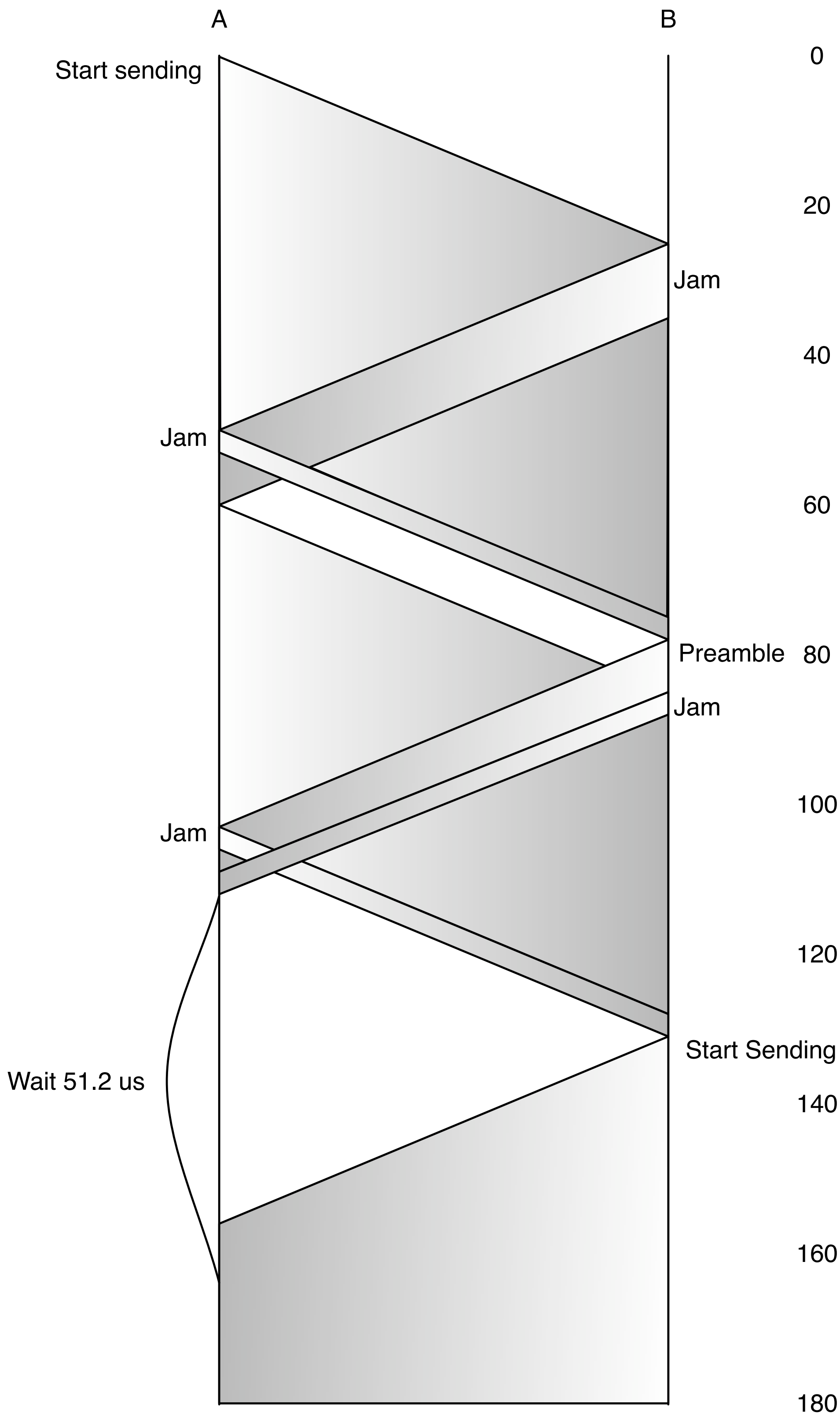

Figure 1: Ethernet Diagram for Problem 1

$$\begin{aligned} p_1 &= 0 \\ p_2 &= 0.5 \\ p_3 &= 0.375 \\ p_4 &= 0.109 \end{aligned}$$

(c) Now assume that after the first collision, node A "wins" the backoff and transmits successfully. After it is finished, both nodes try to transmit again (A has an infinite amount of traffic to send), causing a collision. After this collision, the A's collision counter is at 1 and B's is at 2. Compute the probability that A wins again.

If A picks 0 (with probability 1/2), it wins if B does not pick 0, which happens with probability 3/4. If A picks 1, it wins if B picks 2 or 3, which happens with probability 1/2. So A wins with probability $1/2 * 3/4 + 1/2 * 1/2 = 5/8$.

(d) Given that A "won" the first round, compute the probability that A captures the network for the next 5 frames.

In general, we can see that A wins after i collisions with probability:

$$w_i = \frac{2^i - 1 + 2^i - 2}{2^{i+1}}$$

The chance of winning all 5 frames is surprisingly high:

$$\prod_{i=2}^{6} w_i \approx 0.43$$

3. **Token Ring Networks**

In a token ring network, a station is allowed to hold the token for some period of time, the *token holding time*, THT. Let RingLatency denote the time it takes the token to make one complete rotation around the network when none of the stations have any data to send.

(a) In terms of THT and RingLatency, express the efficiency of the network when only one station is active. Assume early release for the next few questions. *The station sends data for THT time and then waits for RingLatency for the token to circle around, resulting in an efficiency of:*

$$\frac{\text{THT}}{\text{THT} + \text{RingLatency}}$$

(b) What setting of THT would be optimal for a network that only had one station active (with data to send) at a time? *Infinite (or as large as possible).*

(c) In the case where N stations are active, give an upper bound on the token rotation time, TRT, for the network.

$$\text{TRT} \leq N \cdot \text{THT} + \text{RingLatency}$$

(d) Let $N = 100$, THT $= 1000$ μs, and RingLatency $= 200\mu$s. Compute the efficiency of this network if all N nodes are active and are using early release.

Using the above equation, TRT is $100 * 1000 + 200 = 100200$ μs. *Out of this, 100000 $\mu s$is used for productive transmission, so the efficiency is 100000/100200 ≈ 99.8% .*

(e) Compute the efficiency of the above network if delayed release is used.

With delayed release, each node transmits for THT, then waits RingLatency before releasing the token. So in this case:

$$\text{TRT} = N(\text{THT} + \text{RingLatency}) + \text{RingLatency} = 100 \cdot 1200 + 200 = 120200\mu s$$

The useful transmission time is still 100000 so the efficiency is 100000 / 120200 ≈ 83%.

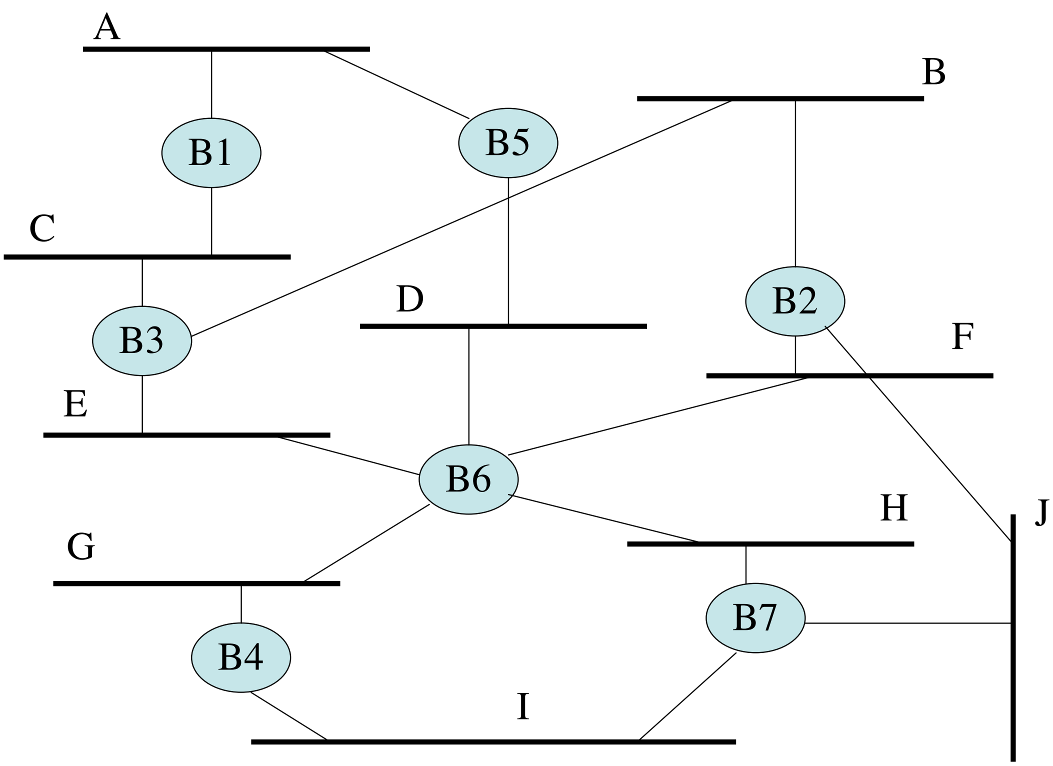

Figure 2: Bridged Network Configuration

4. **Bridges**

Consider the network configuration in Figure 1 for this question.

(a) What ports would be *not* selected by the spanning tree algorithm? (The ID of each bridge is its number, i.e. $B1$ has ID 1.) List ports as pairs (bridge, network), i.e. (B1,A).

0: B1 1: B5, B3 2: B6, B2 3: B4, B7 *(B6,D), (B6,F), (B6,H), (B7,I)*

(b) To simplify thinking about the spanning tree problem, suppose that communication among bridges happens in rounds. In each round, a bridge will forward the configuration message it received from the root in the previous round to all its other links, or, if it had not received this message, become root and send a new configuration message.

Suppose that bridge $B1$ fails at round 0 (i.e. it sent out its last message in round -1). How long will it take for the network of bridges to form a new tree?

This table lists the configuration messages sent in each round. Messages are of the form (own id, distance from root, root id).

round	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**
-1	(B1, 0, B1)	(B2, 2, B1)	(B3, 1, B1)	(B4, 3, B1)	(B5, 1, B1)	(B6, 2, B1)	(B7, 3, B1)
0		(B2, 2, B1)	(B3, 1, B1)	(B4, 3, B1)	(B5, 1, B1)	(B6, 2, B1)	(B7, 3, B1)
1		(B2, 2, B1)	(B3, 0, B3)	(B4, 3, B1)	(B5, 0, B5)	(B6, 2, B1)	(B7, 3, B1)
2		(B2, 0, B2)	(B3, 0, B3)	(B4, 3, B1)	(B5, 0, B5)	(B6, 1, B3)	(B7, 3, B1)
3		(B2, 0, B2)	(B3, 1, B2)	(B4, 4, B1)	(B5, 2, B3)	(B6, 1, B2)	(B7, 1, B2)
4		(B2, 0, B2)	(B3, 1, B2)	(B4, 2, B2)	(B5, 2, B2)	(B6, 1, B2)	(B7, 1, B2)

Note that in round

(c) List the ports that are not selected by the new spanning tree.