SE331: Introduction to Computer Networks	Recitation 5
Semester 1 5785	**4 Dec 2024**
Lecturer: Michael J. May	**Kinneret College**

Sliding Window, Effective Bandwidth, Ethernet, Datagram Switching

1 Effective Bandwidth

The *effective bandwidth* of a conversation is the total amount of information sent divided by the total time the sending took.

- On a 1 Mbps link with a 10ms RTT: A sends B a 1MB file continuously. The total sending time is: $8.394s$. The effective bandwidth is therefore: $\frac{1\times 2^{20}\times 8b}{8.394s} = 999,357.636b/s = 0.999Mbps$
- On a 1 Mbps link with a 10ms RTT: A sends B a 1MB file in 16KB packets with 100B ACK packets sent back in a stop-and-wait protocol. There are 64 packets total. Total sending time is: $9.079808s$. The effective bandwidth is: $\frac{1\times 2^{20}\times 8b}{9.079808s} = 923,875.042b/s = 0.923875Mbps$

  Note that since the transmission times are greater than the RTT, we still have a very high effective bandwidth, even with stop and wait.

Calculate the effective bandwidth for the following scenarios:

(a) On a 10Mbps link with a 150ms RTT: A sends B a 5MB file in 8KB packets with 150B ACK packets using the stop-and-wait protocol.

(b) On a 10Mbps link with a 150ms RTT: A sends B a 5MB file in 8KB packets with 150B ACK packets using the sliding window protocol with SWS=4 and RWS=4. No packets are dropped.

2 Ethernet Propagation

Suppose the round-trip propagation delay for Ethernet is 46.4 μs. This yields a minimum packet size of 512 bits (464 bits corresponding to propagation delay + 48 bits of jam signal).

(a) What happens to the minimum packet size if the delay time is held constant, and the signalling rate rises to 100 Mbps?

(b) What are the drawbacks to so large a minimum packet size?

(c) If compatibility were not an issue, how might the specifications be written so as to permit a smaller minimum packet size?

3 Ethernet Capture

Let A and B be two stations attempting to transmit on an Ethernet. Each has a steady queue of frames ready to send; A's frames will be numbered A_1, A_2, and so on, and B's similarly. Let $T = 51.2\mu$s be the exponential backoff base unit. Suppose A and B simultaneously attempt to send frame 1, collide, and happen to choose backoff times of $0 \times T$ and $1 \times T$, respectively, meaning A wins the race and transmits A_1 while B waits. At the end of this transmission, B will attempt to retransmit B_1 while A will attempt to transmit A_2. These first attempts will collide, but now A backs off for either $0 \times T$ or $1 \times T$, while B backs off for time equal to one of $0 \times T, \ldots, 3 \times T$.

(a) Give the probability that A wins this second backoff race immediately after this first collision; that is, A's first choice of backoff time $k \times 51.2$ is less than B's.

(b) Suppose A wins this second backoff race. A transmits A_3, and when it is finished, A and B collide again as A tries to transmit A_4 and B tries once more to transmit B_1. Give the probability that A wins this third backoff race immediately after the first collision.

(c) Give a reasonable lower bound for the probability that A wins all the remaining backoff races.

(d) What then happens to the frame B_1?

This scenario is known as the *Ethernet capture* effect.

4 Ethernet Address Collisions

Suppose Ethernet physical addresses are chosen at random (using true random bits).

(a) What is the probability that on a 1024-host network, two addresses will be the same?

(b) What is the probability that the above event will occur on some one or more of 2^{20} networks?

(c) What is the probability that of the 2^{30} hosts in all the networks of (b), some pair has the same address?

Hint: The calculation for (a) and (c) is a variant of that used in solving the so-called Birthday Problem:

Given N people, what is the probability that two of their birthdays (addresses) will be the same? The second person has probability $1 - \frac{1}{365}$ of having a different birthday from the first, the third has probability $1 - \frac{2}{365}$ of having a different birthday from the first two, and so on. The probability all birthdays are different is thus

$$\left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \ldots \times \left(1 - \frac{N-1}{365}\right)$$

which for smallish N is about

$$\frac{1 + 2 + \ldots + (N-1)}{365}$$

5 Learning Switches with Datagram Switching

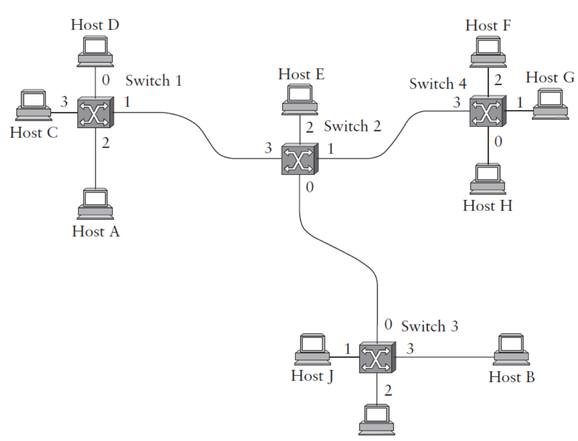

Figure 1: Example Network

Using the example network given above, show how the switches will learn about the locations of the hosts in the network using datagram routing. Assume that the sequence of connections is cumulative; that is, that is the switches learn about each sending in the order shown below and don't lose any information between steps.

(a) Host D sends a packet to host H. Host H responds with an ACK.

(b) Host B sends a packet to host G. Host G responds with an ACK.

(c) Host F sends a packet to host A. Host A responds with an ACK.

(d) Host H sends a packet to host C. Host C responds with an ACK.

(e) Host I sends a packet to host E. Host E responds with an ACK.

(f) Host H sends a packet to host J. Host J responds with an ACK.

6 Unweighted Datagram Routing

Consider the example network given in the figure below. Write down the full forwarding tables for Switches 1–4.

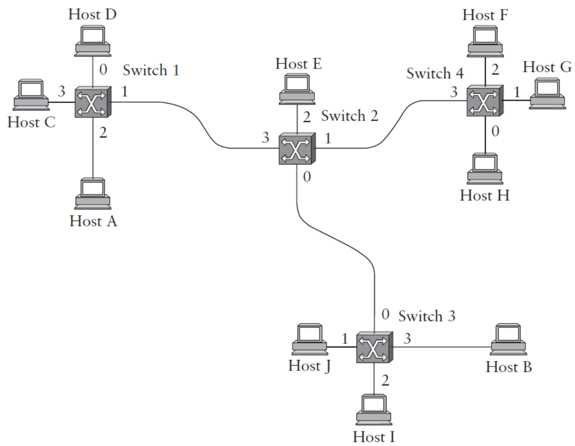

Figure 2: Example Network

7 More Complex Sliding Window: Fast Recovery

Draw a timeline diagram for the sliding window algorithm with SWS = 5 and RWS = 2 with 8 packets for the following situations. Assume each packet is 10KB, each ACK is 100B, the bandwidth is 2Mbps, and the round trip time (RTT) is 100ms. Use a timeout interval of 2× RTT.

Assume that when given a chance to send packets, priority is given to *older packets which have just timed out* over packets which are available to send. For example, if packet 2 just timed out, it has priority over later packets 6, 7, and 8 (which may not have been sent yet).

7.1 Regular Sliding Window

Show a timeline of events using the format shown in the recitation for the following situation:

(a) Packets 2, 4, 6 are lost the first time they are sent.

Use the regular sliding window assumptions, including cumulative acknowledgements (ACKs).

7.2 Sliding Window with Fast Recovery

Show a time lines of events using the format shown in the recitation for the same situations above, but assume that the sender uses the *fast recovery* algorithm shown in class:

> The receiver sends a duplicate acknowledgement if it does not receive the expected packet. For example, it sends `DUPACK[1]` when it expects to see `PACKET[2]` but receives `PACKET[3]` instead. When the sender receives a `DUPACK` message, it immediately sends the (assumed) lost packet.

Explain whether the total transfer time is improved and why.