

Problem 1 [10 Pts]

Peterson and Davie, Chapter 2, Exercise 5.

Solution 1

Insert a 0 after every 5 consecutive 1s.

The transmitted sequence is: (the stuffed bits are in bold font)

1101 0111 11**0**01 0111 11**0**10 1011 11**1**01 10

Problem 2 [15 Pts]

Peterson and Davie, Chapter 2, Exercise 24.

Solution 2

$$\text{RTT} = 2 \times 1.25 \text{ s} = 2.5 \text{ s}$$

$$\text{Bandwidth} \times \text{RTT} = 1 \text{ Mbps} \times 2.5 \text{ s} = 2.5 \text{ Mbits}$$

$$\text{Minimum SWS} = \text{No. of frames in the bandwidth-RTT product} = 2.5 \text{ Mbits} / 1 \text{ KB} = 306$$

$$\text{If RWS} = 1, \text{ then max sequence no.} = 306 + 1 = 307 \Rightarrow \text{no. of bits} = \lceil \log_2(307) \rceil = 9$$

Therefore, min. no. of bits needed for sequence no. = 9 (will need 10 bits if RWS = SWS, but we are just interested in the min. no. of bits)

Problem 3 [15 Pts]

Peterson and Davie, Chapter 2, Exercise 26.

Solution 3

The following problems may arise:

- The sender times out more often on the ACKs, resulting in more retransmissions.
- In case of a lost ACK, the sender has no way of knowing that it did not receive the ACK because it was lost, and not because the receiver's buffer was full. If the sender was using the ACKs to control the rate of data frames being sent, then it might wait for a long time, even though the receiver has space for more data frames.

Problem 4 [20 Pts]

Peterson and Davie, Chapter 2, Exercise 28(a) [10 Pts], and 28(b) [10 Pts]

Solution 4

(a) Figure 1 shows a timeline for the given scenario. Note that the sending of duplicate frames and ACKs continues forever.

(b) Figure 2 shows a timeline for a possible scenario to trigger the Sorcerer's Apprentice bug.

Problem 5 [20 Pts]

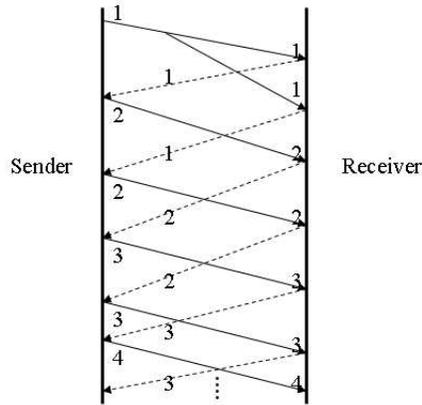


Fig. 1. Sorcerer's Apprentice bug. The solid lines represent the data frames and the dashed lines represent the ACKs. The sending of duplicate frames and ACKs continues forever.

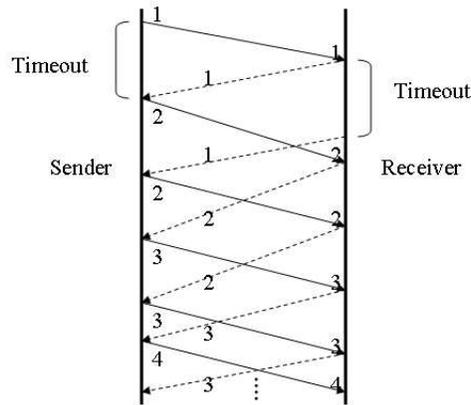


Fig. 2. A possible timeline showing the Sorcerer's Apprentice bug being triggered by a delayed data frame.

Peterson and Davie, Chapter 2, Exercise 36(a) [10 Pts], and 36(b) [10 Pts].

Solution 5

- (a) Each packet takes 1 second to be transmitted from one node to another. Figure 3 shows the timeline.
- (b) All packets in the send window can be transmitted at once (bandwidth is infinite). Figure 4 shows the timeline.

Problem 6 [20 Pts]

Peterson and Davie, Chapter 2, Exercise 49(a) [10 Pts], and 49(b) [10 Pts]

Solution 6

(a) Ethernet physical addresses are 48 bits. Therefore, no. of possible addresses = 2^{48} .

Using the hint, the probability that on a 1024-host network, two addresses are the same =

$$1 - \left(1 - \frac{1}{2^{48}}\right) \times \left(1 - \frac{2}{2^{48}}\right) \times \dots \times \left(1 - \frac{1024 - 1}{2^{48}}\right) \approx \frac{1 + 2 + \dots + (1024 - 1)}{2^{48}} = \frac{1023 \times 1024}{2 \times 2^{48}}$$

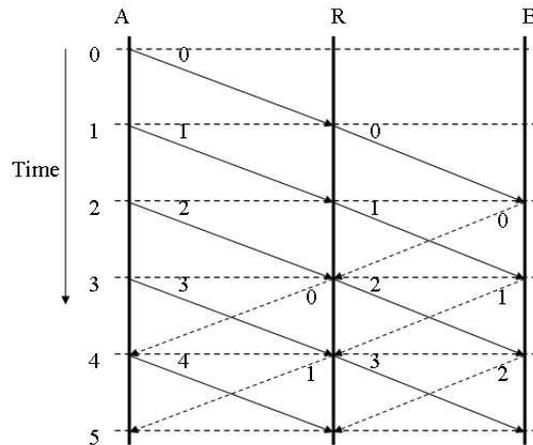


Fig. 3. Timeline. The solid lines represent the data frames and the dashed lines represent the ACKs.

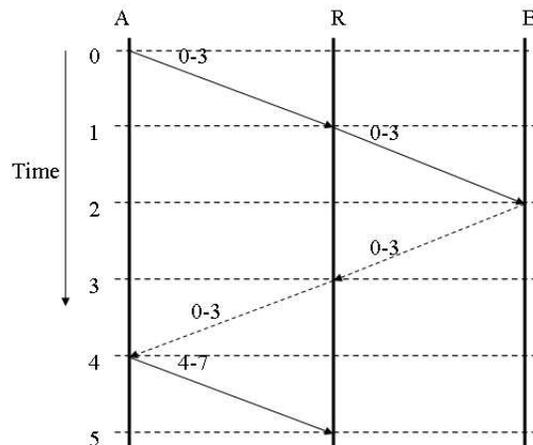


Fig. 4. Timeline. The solid lines represent the data frames and the dashed lines represent the ACKs.

$$= 1.86 \times 10^{-9}$$

(b) Probability of two addresses being the same in a 1024-host network $= p = 1.86 \times 10^{-9}$.

Probability of the event in (a) occurring in none of the 2^{20} networks $= (1 - p)^{2^{20}} \approx 1 - p \times 2^{20} = 0.9980496$

Therefore, probability of the event in (a) occurring in at least one of the 2^{20} networks $\approx 1 - 0.9980496$

$$= 1.95 \times 10^{-3}$$

Problem 7 [20 Pts]

Consider the scenario where Host A is sending three packets to Host B. Each packet is 1500 bits long. Assume that all the delay from Host A to Host B is due to queueing delay at a bottleneck first-come-first-serve store-and-forward queue along the path and transmission delay from that queue (i.e., all other contributions to the end-to-end delay are negligible). This queue is served by a 1 Mbps link on the first-come-first-served basis.

Assume that the bottleneck queue is empty initially. The three packets from Host A are inserted in this queue at 0 ms, 2 ms, and 4 ms, respectively. Some other ongoing connections to destinations other than Host B share

the bottleneck queue with the connection from Host A to Host B. These other connections insert in the bottleneck queue a 4000 bit packet at 0.5 ms and a 2000 bit packet at 3 ms.

(a) Find the times when the packets from Host A are delivered to Host B. [10 points]

(b) Suppose that the Host B wants to forward these packets to the associated display system with the spacing between consecutive packets same as at the time of origination (i.e., 2 ms). A commonly implemented strategy for forwarding packets at a regular interval calls for applying a “build-out delay” to the first packet received such that the packets are forwarded as early as possible while meeting the requirement of the fixed spacing between consecutive packets. Under this strategy, for the packets receive from Host A, what “build-out delay” should Host B apply? [10 points]

Solution 7

(a) We label the packets as 1 (0 ms), 2 (0.5 ms), 3 (2 ms), 4 (3 ms), and 5 (4 ms) in the order of times when the packets were inserted in the queue. Packets 1, 3, and 5 are the ones we are concerned. The queue being first-come-first-served, the packets are sent in the order 1, 2, 3, 4, and then, 5.

Transmission delay of packets 1, 3, and 5 = $\frac{1500}{1 \times 10^6} = 1.5$ ms

Transmission delay of packet 2 = $\frac{4000}{10^6} = 4$ ms

Transmission delay of packet 4 = $\frac{2000}{10^6} = 2$ ms

Since major delay components are transmission delay and queuing delay, packet 1 reaches B at time = arrival time + transmission delay = 0 ms + 1.5 ms

Packet 2 completes transmission at time = 1.5 ms + transmission delay = 5.5 ms

Packet 3 reaches B at time = 5.5 ms + transmission delay = 7 ms

Packet 4 completes transmission at time = 7 ms + transmission delay = 9 ms

Packet 5 reaches B at time = 9 ms + transmission delay = 10.5 ms

(b) For equally spaced time interval, need a delay of 2 ms between packets. Last packet arrives at 10.5 ms. Therefore, packet 1 should be sent at 10.5 ms - (2 × 2 ms) = 6.5 ms. Therefore, the build-out delay = 6.5 ms - 1.5 ms = 5 ms.