

ECE 465: Computer Network Protocols and Applications
Homework 7
Solution
Dr. Paris

Chapter 3 Problems

Problem 1.

	source port numbers	destination port numbers
a) $A \rightarrow S$	467	23
b) $B \rightarrow S$	513	23
c) $S \rightarrow A$	23	467
d) $S \rightarrow B$	23	513

e) Yes.

f) No.

Problem 2.

Suppose the IP addresses of the hosts A, B, and C are a, b, c, respectively. (Note that a,b,c are distinct.)

To host A: Source port =80, source IP address = b, dest port = 26145, dest IP address = a

To host C, left process: Source port =80, source IP address = b, dest port = 7532, dest IP address = c

To host C, right process: Source port =80, source IP address = b, dest port = 26145, dest IP address = c

Problem 3.

$$\begin{array}{r}
 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\
 + \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 \hline
 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\
 + \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \\
 \hline
 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1
 \end{array}$$

One's complement = 1 1 1 0 1 1 1 0.

To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error. All one-bit errors will be detected, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

Problem 4.

Suppose the sender is in state “Wait for call 1 from above” and the receiver (the receiver shown in the homework problem) is in state “Wait for 1 from below.” The sender sends a packet with sequence number 1, and transitions to “Wait for ACK or NAK 1,” waiting for an ACK or NAK. Suppose now the receiver receives the packet with sequence number 1 correctly, sends an ACK, and transitions to state “Wait for 0 from below,” waiting for a data packet with sequence number 0. However, the ACK is corrupted. When the rdt2.1 sender gets the corrupted ACK, it resends the packet with sequence number 1. However, the receiver is waiting for a packet with sequence number 0 and (as shown in the home work problem) always sends a NAK when it doesn't get a packet with sequence number 0. Hence the sender will always be sending a packet with sequence number 1, and the receiver will always be NAKing that packet. Neither will progress forward from that state.

Problem 8.

Here, we add a timer, whose value is greater than the known round-trip propagation delay. We add a timeout event to the “Wait for ACK or NAK0” and “Wait for ACK or NAK1” states. If the timeout event occurs, the most recently transmitted packet is retransmitted. Let us see why this protocol will still work with the rdt2.1 receiver.

- Suppose the timeout is caused by a lost data packet, i.e., a packet on the sender-to-receiver channel. In this case, the receiver never received the previous transmission and, from the receiver's viewpoint, if the timeout retransmission is received, it look *exactly* the same as if the original transmission is being received.
- Suppose now that an ACK is lost. The receiver will eventually retransmit the packet on a timeout. But a retransmission is exactly the same action that is take if an ACK is garbled. Thus the sender's reaction is the same with a loss, as with a garbled ACK. The rdt 2.1 receiver can already handle the case of a garbled ACK.