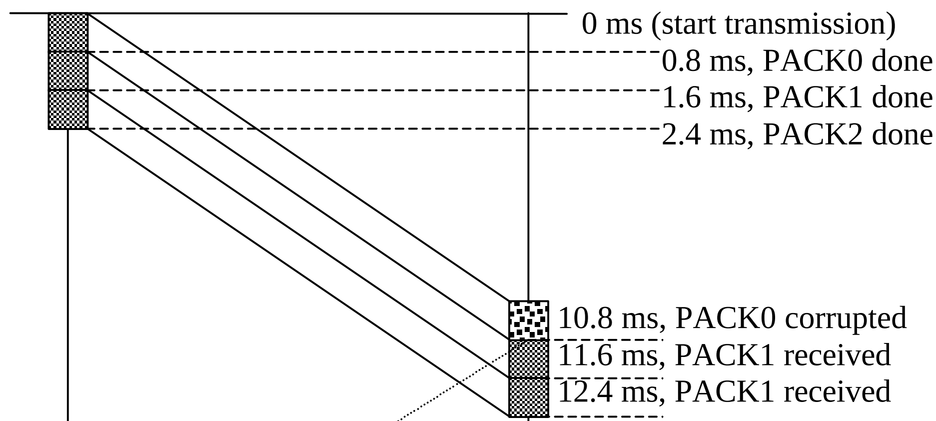
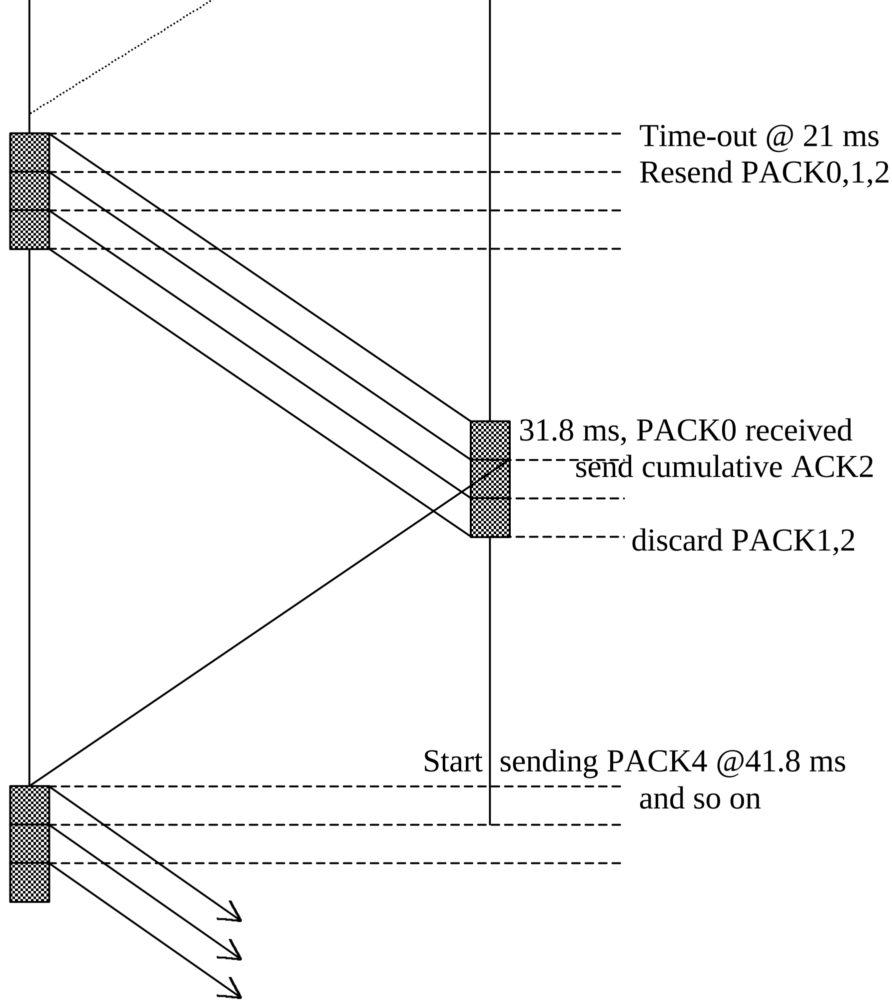
2.993: Principles of Internet Computing
Homework #4 Solutions

1.(a) Cumulative ACKs

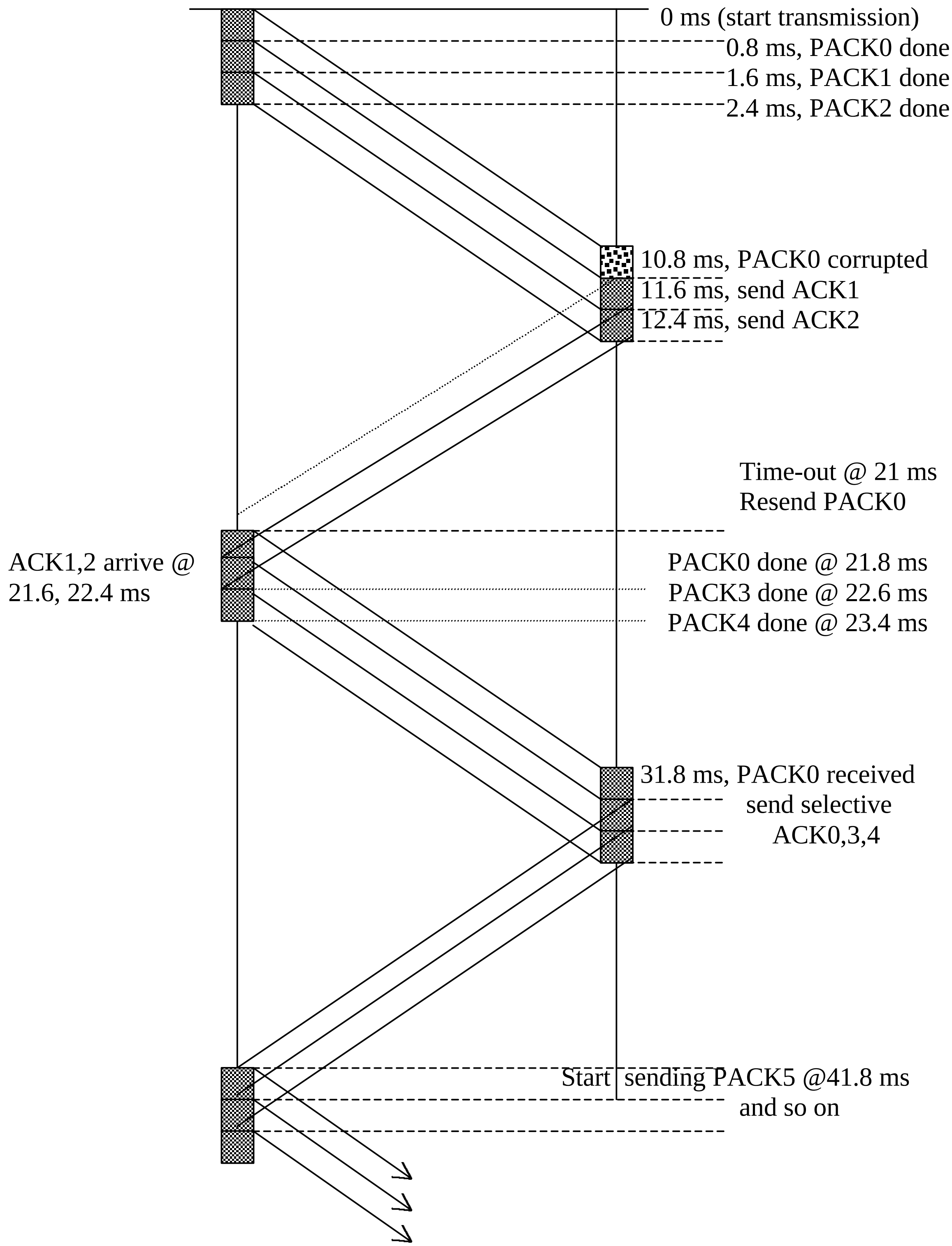
propagation delay = 10 ms, packet transmission time ~ 0.8 ms



We assume ACK size is small relative to PACK, and its transmission time is neglected.
We pick time-out = 21 ms > RTT = 20.8 ms



(b) selective ACKs



2. Cumulative ACKs has less complexity because ACKs are not sent for every packet. Selective ACKs can improve the throughput.

3.

(a) For Go-Back-N, RWS = 1.

(b) the minimum required sequence size for a sliding window is (SWS + RWS).

If SWS=N and RWS=M, consider the following worst-case scenario: The sender sends N packets. All are received in order at the receiver, but all of its ACKs are corrupted. The receiver is expecting packets from (N+1) to (N+M). After time-out, the sender repeats the same packets, but the receiver is expecting new ones. For RWS=M, the confusion is avoided if the (minimum) total number of SeqNum is (N+M).

i) 6 ii) 5 iii) X+Y

4. We know RTT $\times$ bandwidth = window size
Thus, throughput = window size/RTT

i) (a) 1 KB/40 ms (b) 1KB/20ms
(Keep in mind though that the actual throughput is the same.)

ii) This question is related to a comparison between end-to-end and link-to-link flow controls. As you know, the Internet (with IP) uses datagram (connectionless) communication and end-to-end sliding window (TCP). But a virtual-circuit (connection-oriented) communication is used for such networks as X.25 or ATM. The latter has the advantage of providing some sort of quality of service guarantee (throughput, delay). This, however, requires coordination among routers within the network, which increases complexity. For the Internet, where many different networks may be interconnected, the function of flow control is left to the end user.

We will revisit this topic later in the semester. You can also read 4.1.2 and 4.1.3 from the text.

5.

During 1st RTT, PACK=1 is sent. The snd_wnd=1. After receiving ACK=2 at the end of 1st RTT, snd_wnd is incremented to 2, and PACK=2,3 are sent at the start of 2nd RTT. After ACK=3 is received at the end of 2nd RTT, snd_wnd=3. During 3rd RTT, after receiving ACK=4, snd_wnd=4. Thus, during 3rd RTT, there are four outstanding packets, PACK=4,5,6,7. When ACK=5 is received at the end of 3rd RTT, the slow-start threshold begins. ACKs=5,6,7,8 contribute 1/4 each. ACKs=6,7,8 arrive during 4th RTT. When ACK=8 (for PACK=7) is received at the end of 4th RTT, snd_wnd=5. Thus, during 4th RTT, there are five outstanding packets, PACK=8,9,10,11,12. The acknowledgments, ACK=9,10,11,12,13 which arrive during 5th RTT, each contribute 1/5 to the window size.

Also, during 5th RTT, PACK=13,14,15,16,17 are sent. After snd_wnd=6, at the start of 6th RTT, 6 packets can be outstanding. Thus, PACK=18 is also sent. But delay-bandwidth product is only 3, and the bottleneck buffer size is 2. Thus, PACK=18 is dropped, and it causes fast retransmission.

RTT#	window range	packets sent	ssthresh	buffer size
1	1 - 2	1	4	0
2	2 - 4	2,3	4	0
3	4 - 5	4,5,6,7	4	1
4	5 - 6	8,9,10,11,12	4	2
5	6 -	13,14,15,16,17,18	4	2/drop