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

Start sending PACK4 @41.8 ms and so on
(b) selective ACKs

0 ms (start transmission)
- 0.8 ms, PACK0 done
- 1.6 ms, PACK1 done
- 2.4 ms, PACK2 done

10.8 ms, PACK0 corrupted
11.6 ms, send ACK1
12.4 ms, send ACK2

Time-out @ 21 ms
Resend PACK0

ACK1,2 arrive @ 21.6, 22.4 ms
PACK0 done @ 21.8 ms
PACK3 done @ 22.6 ms
PACK4 done @ 23.4 ms

31.8 ms, PACK0 received
send selective
ACK0,3,4

Start sending PACK5 @ 41.8 ms
and so on
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 $\text{RTT} \times \text{bandwidth} = \text{window size}$

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 1\textsuperscript{st} RTT, PACK=1 is sent. The snd\_wnd=1. After receiving ACK=2 at the end of 1\textsuperscript{st} RTT, snd\_wnd is incremented to 2, and PACK=2,3 are sent at the start of 2\textsuperscript{nd} RTT. After ACK=3 is received at the end of 2\textsuperscript{nd} RTT, snd\_wnd=3. During 3\textsuperscript{rd} RTT, after receiving ACK=4, snd\_wnd=4. Thus, during 3\textsuperscript{rd} RTT, there are four outstanding packets, PACK=4,5,6,7. When ACK=5 is received at the end of 3\textsuperscript{rd} RTT, the slow-start threshold begins. ACKs=5,6,7,8 contribute 1/4 each. ACKs=6,7,8 arrive during 4\textsuperscript{th} RTT. When ACK=8 (for PACK=7) is received at the end of 4\textsuperscript{th} RTT, snd\_wnd=5. Thus, during 4\textsuperscript{th} RTT, there are five outstanding packets, PACK=8,9,10,11,12. The acknowledgments, ACK=9,10,11,12,13 which arrive during 5\textsuperscript{th} RTT, each contribute 1/5 to the window size.
Also, during 5\textsuperscript{th} RTT, PACK=13,14,15,16,17 are sent. After snd wnd=6, at the start of 6\textsuperscript{th} 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.

<table>
<thead>
<tr>
<th>RTT#</th>
<th>window range</th>
<th>packets sent</th>
<th>ssthresh</th>
<th>buffer size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 - 2</td>
<td>1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2 - 4</td>
<td>2,3</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4 - 5</td>
<td>4,5,6,7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5 - 6</td>
<td>8,9,10,11,12</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>6 -</td>
<td>13,14,15,16,17</td>
<td>4</td>
<td>2/drop</td>
</tr>
</tbody>
</table>