1. (10pt.)
   a) TCP opens a connection using an initial sequence number of 22,156. The other party opens
      the connection with an initial sequence number of 34,425. Show the three-way hand shaking
      during the connection establishment.
   b) TCP is sending data at 4 mega bytes per second (1 byte = 8 bits). If the sequence number
      starts with 6000 in decimal, how long does it take before the sequence number goes back to
      zero? (hint: in TCP, every byte has a sequence number)

2. (10pt.) Consider two nodes are linked by a 4Mbps channel and RTT is 0.04 sec. Assume the
   size of each packet is 1024 bits. Answer the following questions for ARQ schemes:
   a) Assume that the link is error-free, what is the possible maximum rate of transmission for
      Stop-and-wait, GBN, and SR, respectively? Why?
   b) For GBN and SR, in order to allow sender to continuously send packets without any
      waiting, what is the minimum window size in terms of the number of packets?
   c) For b), what should be the minimum number of bits for the sequence number for GBN
      and SR, respectively?
   d) Suppose that we are continuously transmitting packets end-to-end start from the 1st
      packet, and the 4th packet is lost. Assume there is no other packet lost or ACK lost. For
      stop-and-wait, GBN, and SR, which packets need to be retransmitted?

3. (10pt.) Suppose TCP Tahoe is used over a lossy link that loses one segment in every 7th RTT
   (for example, starting from RTT #1, one segment will be lost during RTT #7, and the
   congestion window and threshold should be adjusted accordingly at RTT #7. The loss will
   repeat during RTT #14, RTT #21, …). Show how congestion window varies over time by
   filling in the following table. Assume that initially the congestion window is 2 segment and
   the threshold is 32. Also, while computing threshold, round it up to the nearest integer.

   To simply this question, we assume after a burst is sent, the sender will wait for ACKs to all the
   segments before the congestion window is adjusted. In addition, we assume the ACKs will return
   to the sender every RTT time, if not lost.

<table>
<thead>
<tr>
<th>RTT #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Congwin</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RTT #</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Congwin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. (10pt.) Consider following network:
   a) Suppose that routing information is exchanged using link state routing protocol and the shortest paths are computed using Dijkstra's algorithm. What will be the shortest paths spanning tree computed by E? (Show the steps in your computation, follow the example in slide 4a-13, and draw the spanning tree.) What will be the routing table computed by E?
b) Suppose distance vector routing is used instead. What will be the distance table computed by E? What will be the routing table computed by E? You don't need to show the steps here.

5. (10pt.) As networks become a critical infrastructure that many businesses depend on, reliability is an important consideration in today's networks. To ensure reliability in the face of link failure, one solution is to establish a back-up path between two nodes in addition to a primary path (e.g., the shortest path). This back-up path should not share any link with the primary path, i.e., they are disjoint.
   a) Suppose that the nodes in a network (with any arbitrary topology) exchange routing information using a link state routing protocol. Design a simple heuristic algorithm that will find two disjoint paths from a given source node to any destination node. You need to argue that your algorithm is correct, i.e., the two paths your algorithm finds are indeed disjoint.
   b) Suppose the nodes use a distance vector routing protocol to exchange routing information. Is it possible to find two disjoint paths from a given source to any destination node? Briefly explain your answer.