Chapter 4   Problems

1.

a) With a connection-oriented network, every router failure will involve the routing of that connection. At a minimum, this will require the router that is “upstream” from the failed router to establish a new downstream part of the path to the destination node, with all of the requisite signaling involved in setting up a path. Moreover, all of the router on the initial path that are downstream from the failed node must take down the failed connection, with all of the requisite signaling involved to do this.

With a connectionless datagram network, no signaling is required to either set up a new downstream path or take down the old downstream path. We have seen, however, that routing tables will need to be updated (e.g., either via a distance vector algorithm or a link state algorithm) to take the failed router into account. We have seen that with distance vector algorithms, this routing table change can sometimes be localized to the area near the failed router. Thus, a datagram network would be preferable. Interesting, the design criteria that the initial ARPAnet be able to function under stressful conditions was one of the reasons that a datagram architecture was chosen for this Internet ancestor.

b) In order for a router to determine the delay (or a bound on delay) along an outgoing link, it would need to know the characteristics of the traffic from all sessions passing through that link. That is, the router must have per-session state in the router. This is possible in a connection-oriented network, but not with a connectionless network. Thus, a connection-oriented network would be preferable.
5.

The distance table in E is:

<table>
<thead>
<tr>
<th>via</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>to</td>
<td>a</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

7.

The distance table in X is:

<table>
<thead>
<tr>
<th>via</th>
<th>W</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>to</td>
<td>W</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>6</td>
</tr>
</tbody>
</table>

Note that there is not enough information given in the problem (purposefully!) to determine the distance table entries D(W,Y) and D(Y,W). To know these values, we would need to know Y's minimum cost to W, and vice versa.

Since X's least cost path to A goes through W, a change in the link cost c(X,W) will cause X to inform its neighbors of a new minimum cost path to A. Since X's least cost path to A does not go through Y, a change in the link cost c(X,Y) will not cause X to inform its neighbors of a new minimum cost path to A.

10.

We show below a simplified finite state machine for the case that the DHCP client specifically requests DHCP service via the DHCP discover message. (Not shown below and not discussed in the text is the case where DHCP servers active
advertise their services by sending DHCP server advertisements, see RFC 2131 for details).

In our FSM below, the notation DHCPdiscover(transactionID) refers to sending a DHCP discover message with the specified transaction ID. In the “gather offers” state the DHCP client maintains two timers: a “fast timer” that is used to retransmit DHCP discover messages if no replies are received, and a “slow timer” that causes the DHCP client to leave the “gather offers” state (to either the failure state, if no DHCP server offers have been received, or to the “wait ACK” state) when it times out.

```
Λ
DHCPdiscover(transactionID)

noffers = 0
start slowtimer
start fasttimer

fail
slow timer timeout && noffers==0
Λ

wait
ack
fast timer timeout && noffers==0
resend DHCPdiscover
restart fasttimer

pick among DHCP offers
DHCPrequest(transactionID)
restart both timers

DHCPack(rcvd_transactionID) &&
(transactionID != rcvd_transactionID)
Λ

DHCPack(rcvd_transactionID) &&
(transactionID == rcvd_transactionID)
Λ

done
```
12. One way for C to force B to hand over all of B’s traffic to D on the east coast is for C to only advertise its route to D via its east coast peering point with C.

21. See Figure 5.