P2. Consider a virtual-circuit network. Suppose the VC number is an 8-bit field.
   a. What is the maximum number of virtual circuits that can be carried over a
      link?
   b. Suppose a central node determines paths and VC numbers at connection
      setup. Suppose the same VC number is used on each link along the VC's
      path. Describe how the central node might determine the VC number at con-
      nection setup. Is it possible that there are fewer VC's in progress than the
      maximum as determined in part (a) yet there is no common free VC number?
   c. Suppose that different VC numbers are permitted in each link along a
      VC's path. During connection setup, after an end-to-end path is determined,
      describe how the links can choose their VC numbers and configure their for-
      warding tables in a decentralized manner, without reliance on a central node.

P3. A bare-bones forwarding table in a VC network has four columns. What is
the meaning of the values in each of these columns? A bare-bones forwarding
table in a datagram network has two columns. What is the meaning of the
values in each of these columns?

P4. Consider the network below.
   a. Suppose that this network is a datagram network. Show the forwarding
      table in router A, such that all traffic destined to host H3 is forwarded
      through interface 3.
   b. Suppose that this network is a datagram network. Can you write down a
      forwarding table in router A, such that all traffic from H1 destined to host
      H3 is forwarded through interface 3, while all traffic from H2 destined to
      host H3 is forwarded through interface 4? (Hint: this is a trick question.)
   c. Now suppose that this network is a virtual circuit network and that there is
      one ongoing call between H1 and H3, and another ongoing call between
      H2 and H3. Write down a forwarding table in router A, such that all traffic
      from H1 destined to host H3 is forwarded through interface 3, while all
      traffic from H2 destined to host H3 is forwarded through interface 4.
   d. Assuming the same scenario as (c), write down the forwarding tables in
      nodes B, C, and D.

P5. Consider a VC network with a 2-bit field for the VC number. Suppose that
the network wants to set up a virtual circuit over four links: link A, link B,
be transferred to a given output port in a time slot, but different output ports can receive datagrams from different input ports in a single time slot. What is the minimal number of time slots needed to transfer the packets shown from input ports to their output ports, assuming any input queue scheduling order you want (i.e., it need not have HOL blocking)? What is the largest number of slots needed, assuming the worst-case scheduling order you can devise, assuming that a non-empty input queue is never idle?

---

P10. Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111</td>
<td>0</td>
</tr>
<tr>
<td>11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111</td>
<td>1</td>
</tr>
<tr>
<td>11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

a. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.

b. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

11000000 00100001 01010001 01010010 1101101
11100001 01100000 11000011 00111100
11100001 10000000 00010001 10110111
P11. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
</tr>
</tbody>
</table>

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

P12. Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

P13. Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support at least 60 interfaces, Subnet 2 is to support at least 90 interfaces, and Subnet 3 is to support at least 12 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

P14. In Section 4.2.2 an example forwarding table (using longest prefix matching) is given. Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

P15. In Problem P10 you are asked to provide a forwarding table (using longest prefix matching). Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.
P22. Suppose you are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a random number, and the identification numbers of the subsequent IP packets are sequentially assigned. Assume all IP packets generated by hosts behind the NAT are sent to the outside world.

a. Based on this observation, and assuming you can sniff all packets sent by the NAT to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.

b. If the identification numbers are not sequentially assigned but randomly assigned, would your technique work? Justify your answer.

P23. In this problem we'll explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

P24. Looking at Figure 4.27, enumerate the paths from $v$ to $u$ that do not contain any loops.

P25. Repeat Problem P24 for paths from $x$ to $z$, $z$ to $u$, and $z$ to $w$.

P26. Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from $x$ to all network nodes. Show how the algorithm works by computing a table similar to Table 4.3.

-- Diagram of a network with labeled edges --

P27. Complete the following.

a.

b.

c.

d.

e.

f. I

P28. Complete the following.

P29. Complete the following.

P30. Consider the network with vertices $x$, $y$, and $z$. Assume that $x$, $y$, and $z$ are connected by edges with costs as shown in the table.

--- Network diagram with labeled edges ---
P27. Consider the network shown in Problem P26. Using Dijkstra's algorithm, and showing your work using a table similar to Table 4.3, do the following:
   a. Compute the shortest path from \( i \) to all network nodes.
   b. Compute the shortest path from \( u \) to all network nodes.
   c. Compute the shortest path from \( v \) to all network nodes.
   d. Compute the shortest path from \( w \) to all network nodes.
   e. Compute the shortest path from \( y \) to all network nodes.
   f. Compute the shortest path from \( z \) to all network nodes.

P28. Consider the network shown below, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node \( z \).

![Network Diagram](image)

P29. Consider a general topology (that is, not the specific network shown above) and a synchronous version of the distance-vector algorithm. Suppose that at each iteration, a node exchanges its distance vectors with its neighbors and receives their distance vectors. Assuming that the algorithm begins with each node knowing only the costs to its immediate neighbors, what is the maximum number of iterations required before the distributed algorithm converges? Justify your answer.

P30. Consider the network fragment shown below. \( x \) has only two attached neighbors, \( w \) and \( y \). \( w \) has a minimum-cost path to destination \( u \) (not shown) of 5, and \( y \) has a minimum-cost path to \( u \) of 6. The complete paths from \( w \) and \( y \) to \( u \) (and between \( w \) and \( y \)) are not shown. All link costs in the network have strictly positive integer values.

![Network Diagram](image)