Practice Problem 1) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 P1)

In this question, we consider some of the pros and cons of virtual-circuit and datagram networks.

a. Suppose that routers were subjected to conditions that might cause them to fail fairly often. Would this argue in favor of a VC or datagram architecture? Why?

b. Suppose that a source node and a destination require that a fixed amount of capacity always be available at all routers on the path between the source and destination node, for the exclusive use of traffic flowing between this source and destination node. Would this argue in favor of a VC or datagram architecture? Why?

c. Suppose that the links and routers in the network never fail and that routing paths used between all source/destination pairs remains constant. In this scenario, does a VC or datagram architecture have more control traffic overhead? Why?

Practice Problem 2) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 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 connection setup. Is it possible that there are fewer VCs 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 forwarding tables in a decentralized manner, without reliance on a central node.

Practice Problem 3) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 P6)

In the text we have used the term connection-oriented service to describe a transport-layer service and connection service for a network-layer service. Why do we use different terminology?

Practice Problem 4) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 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:

|  |  |
| --- | --- |
| Destination Address Range | Outgoing Link Interface |
| 11100000 00000000 00000000 00000000through11100000 00111111 11111111 11111111 | 0 |
| 11100000 01000000 00000000 00000000through11100000 01000000 11111111 11111111 | 1 |
| 11100000 01000001 00000000 00000000through11100001 01111111 11111111 11111111 | 2 |
| Otherwise | 3 |

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:

11001000 10010001 01010001 01010101

11100001 01000000 11000011 00111100

11100001 10000000 00010001 01110111

Practice Problem 5) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 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.

Practice Problem 6) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 P15)

In Problem 4 (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.

Practice Problem 7) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 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.



Practice Problem 8) (Computer Networking: A Top-Down Approach 6th Edition: Chapter 4 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.

