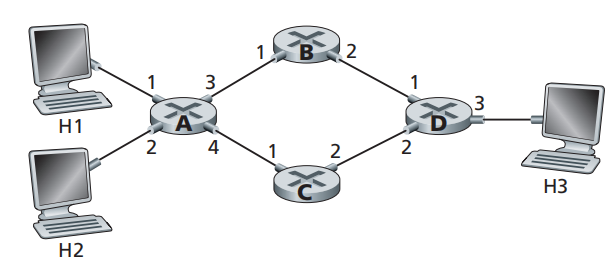
P1. Consider the network below.

a. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3.

b. 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.)



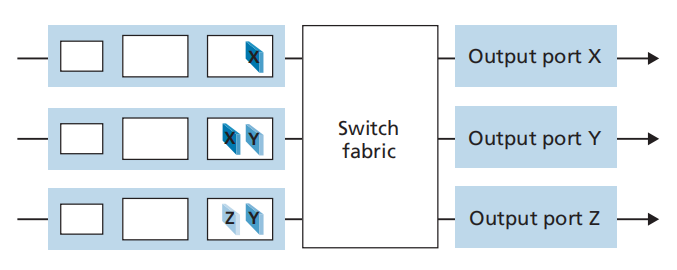
Answer:

1. Header Value Destination H3

Output Interface Link 3

1. No, this is impossible. A datagram bases what path to take based off of the destination. If the output link from router A is determined to be link 3 as denoted in the routing table above, then this will be the only path that H2 will be able to send on. H2 can only tell the datagram to send to router A, from there router A will pick the link with the lowest cost, which is link 3.

P4. Consider the switch shown below. Suppose that all datagrams have the same fixed length, that the switch operates in a slotted, synchronous manner, and that in one time slot a datagram can be transferred from an input port to an output port. The switch fabric is a crossbar so that at most one datagram can 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?



Answer:

The minimal number of time slots needed is 3. The scheduling is as follows.

Slot 1: send X in top input queue, send Y in middle input queue.

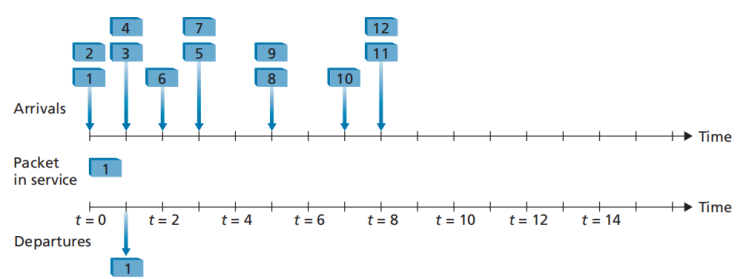
Slot 2:  send X in middle input queue, send Y in bottom input queue

Slot 3: send Z in bottom input queue.

Largest number of slots is still 3.  Actually, based on the assumption that a non-empty input queue is never idle, we see that the first time slot always consists of sending in the top input queue and Y in either middle or bottom input queue, and in the secong time slot, we can always send two more datagram, and the last datagram can be sent in the third slot.

NOTE: Actually, if the first datagram in the bottom input queue is X, then the worst case would require 4 time slots.

P6. Consider the figure below. Answer the following questions:



a. Assuming FIFO service, indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the slot in which it is transmitted? What is the average of this delay over all 12 packets?

b. Now assume a priority service, and assume that odd-numbered packets are high priority, and even-numbered packets are low priority. Indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and the beginning of the slot in which it is transmitted? What is the average of this delay over all 12 packets?

c. Now assume round robin service. Assume that packets 1, 2, 3, 6, 11, and 12 are from class 1, and packets 4, 5, 7, 8, 9, and 10 are from class 2. Indicate the time at which packets 2 through 12 each leave the queue. For each packet, what is the delay between its arrival and its departure? What is the average delay over all 12 packets?

d. Now assume weighted fair queueing (WFQ) service. Assume that oddnumbered packets are from class 1, and even-numbered packets are from class 2. Class 1 has a WFQ weight of 2, while class 2 has a WFQ weight of 1. Note that it may not be possible to achieve an idealized WFQ schedule as described in the text, so indicate why you have chosen the particular packet to go into service at each time slot. For each packet what is the delay between its arrival and its departure? What is the average delay over all 12 packets?

e. What do you notice about the average delay in all four cases (FIFO, RR, priority, and WFQ)?

Answer:

|  |  |  |  |
| --- | --- | --- | --- |
| Packet | Arrival time | Leaving time | Delay (leaving-arrival time) |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 2 | 1 |
| 4 | 1 | 3 | 2 |
| 5 | 3 | 4 | 2 |
| 6 | 2 | 5 | 2 |
| 7 | 3 | 6 | 3 |
| 8 | 5 | 7 | 2 |
| 9 | 5 | 8 | 3 |
| 10 | 7 | 9 | 2 |
| 11 | 8 | 10 | 2 |
| 12 | 8 | 11 | 3 |
| Average delay: | | | 1.91 |

|  |  |  |  |
| --- | --- | --- | --- |
| Packet | Arrival time | Leaving time | Delay (leaving-arrival time) |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 2 | 2 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 6 | 5 |
| 5 | 3 | 4 | 1 |
| 6 | 2 | 7 | 5 |
| 7 | 3 | 3 | 0 |
| 8 | 5 | 9 | 4 |
| 9 | 5 | 5 | 0 |
| 10 | 7 | 10 | 3 |
| 11 | 8 | 8 | 0 |
| 12 | 8 | 11 | 3 |
| Average delay: | | | 1.91 |

1. 100
2. 222
3. 310
4. 465
5. 531
6. 675
7. 740
8. 894
9. 950
10. 10 10 3
11. 11 8 0
12. 12 11 3
13. Average Delay 1.91

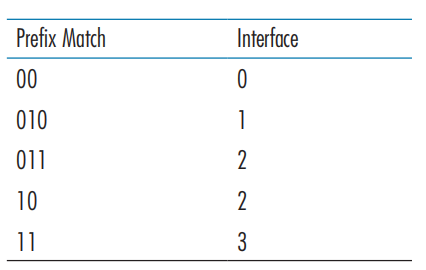
|  |  |  |  |
| --- | --- | --- | --- |
| Packet | Arrival time | Leaving time | Delay (leaving-arrival time) |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 2 | 2 |
| 3 | 1 | 4 | 3 |
| 4 | 1 | 1 | 0 |
| 5 | 3 | 3 | 0 |
| 6 | 2 | 6 | 4 |
| 7 | 3 | 5 | 2 |
| 8 | 5 | 7 | 2 |
| 9 | 5 | 9 | 4 |
| 10 | 7 | 11 | 4 |
| 11 | 8 | 8 | 0 |
| 12 | 8 | 10 | 2 |
| Average delay: | | | 1.91 |

D.

|  |  |  |  |
| --- | --- | --- | --- |
| Packet | Arrival time | Leaving time | Delay (leaving-arrival time) |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 2 | 2 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 5 | 4 |
| 5 | 3 | 3 | 0 |
| 6 | 2 | 7 | 5 |
| 7 | 3 | 4 | 1 |
| 8 | 5 | 9 | 4 |
| 9 | 5 | 6 | 1 |
| 10 | 7 | 10 | 3 |
| 11 | 8 | 8 | 0 |
| 12 | 8 | 11 | 3 |
| Average delay: | | | 1.91 |

P9. Consider a datagram network using 8-bit host addresses. Suppose a router

uses longest prefix matching and has the following forwarding table:



For each of the four interfaces, give the associated range of destination host

addresses and the number of addresses in the range.

Answer:

|  |  |
| --- | --- |
| **Destination Address Range** | **Link Interface** |
| 00000000 – 00111111 | 0 |
| 01000000 – 01011111 | 1 |
| 01100000 – 01111111 | 2 |
| 10000000 – 10111111 | 2 |
| 11000000 – 11111111 | 3 |

number of addresses for interface 0 = 26=64

number of addresses for interface 1 = 25 =32

number of addresses for interface 2 = 25 +26 =32+64 =96

number of addresses for interface 3 = 26 =64