# Computer Networking

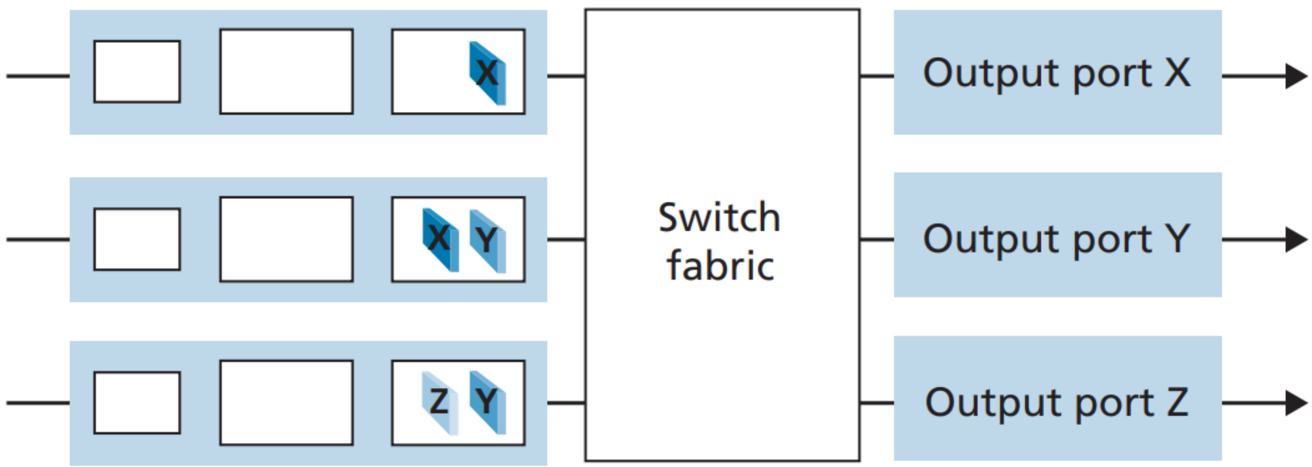
# Assignment 3

# Deadline Dec. 19, 2023

**Submission format: submit your solution in a single file to blackboard under Assignment Section, choose whatever format you like (jpg, pdf, word) as long as it is clear enough for grading, compress into a single .zip file if there are multiple files). Comment your name and student number in the submission**

## Problem 1: Switching Fabrics (10 points)

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 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?



Solution 1:

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 X in the top input queue and Y in either middle or bottom input queue, and in the second time slot, we can always send two more datagram, and the last datagram can be sent in third time slot.

## Problem 2: Packet Scheduling (10 points)

Suppose that the WFQ scheduling policy is applied to a buffer that supports three classes, and suppose the weights are 0.5, 0.25, and 0.25 for the three classes.

(1) Suppose that each class has a large number of packets in the buffer. In what sequence might the three classes be served in order to achieve the WFQ weights? (For round robin scheduling, a natural sequence is 123123123 . . .).

(2) Suppose that classes 1 and 2 have a large number of packets in the buffer, and there are no class 3 packets in the buffer. In what sequence might the three classes be served in to achieve the WFQ weights?

## Solution 2:

(1) The WFQ (Weighted Fair Queuing) scheduling is a circular order a round roubin strategy in ascending order depends on weights. Assume the three classes weights are 0.5, 0.25, and 0.25 respectively. If same weight have for two or more classes then give highest order is the first preference class.

The sequence order that the three classes served is :1 2 1 3 1 2 1 3….

Another possible sequence order that the three class are: 1 1 2 3 1 1 2 3…

(2) Consider the classes 1 and 2 have a large number of packets in the buffer, and there are no class 3 packets in the buffer. The sequence might the three classes be served in to achieve the WFQ weights are 1 1 2 1 1 2 1 1 2….

Problem 3: IP Addressing (15 points)

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** | **Link Interface** |
|---|---|
| 11100000 00000000 00000000 00000000<br>through<br>11100000 00111111 11111111 11111111 | 0 |
| 11100000 01000000 00000000 00000000<br>through<br>11100000 01000000 11111111 11111111 | 1 |
| 11100000 01000001 00000000 00000000<br>through<br>11100001 01111111 11111111 11111111 | 2 |
| otherwise | 3 |

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

(2) 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

(3) 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.

Solution 3:

(1)

| Prefix Match | Link Interface |
|---|---|
| 11100000 00 | 0 |
| 11100000 01000000 | 1 |
| 1110000 | 2 |
| 11100001 1 | 3 |
| otherwise | 3 |

(2)

Prefix match for first address is 5th entry: link interface 3

Prefix match for second address is 3nd entry: link inte15+1283rface 2

Prefix match for third address is 4th entry: link interface 3

(3)

| Destination Address | Link Interface |
|---|---|
| 11100000 00 (224.0.0.0/10) | 0 |
| 11100000 01000000 (224.64.0.0/16) | 1 |
| 1110000 (224.0.0.0/7) | 2 |
| 11100001 1 (225.128.0.0/9) | 3 |
| otherwise | 3 |

Problem 4: Subnet (10 points)

Consider the topology shown in Figure 1. Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

(1) Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support 250 interfaces; Subnet B should have enough addresses to support 120 interfaces; and Subnet C should have enough addresses to support 120 interfaces. Of course, subnets D, E and F should each be able to support two interfaces. For each subnet, the assignment should take the form a.b.c.d/x or a.b.c.d/x – e.f.g.h/y.

(2) Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.

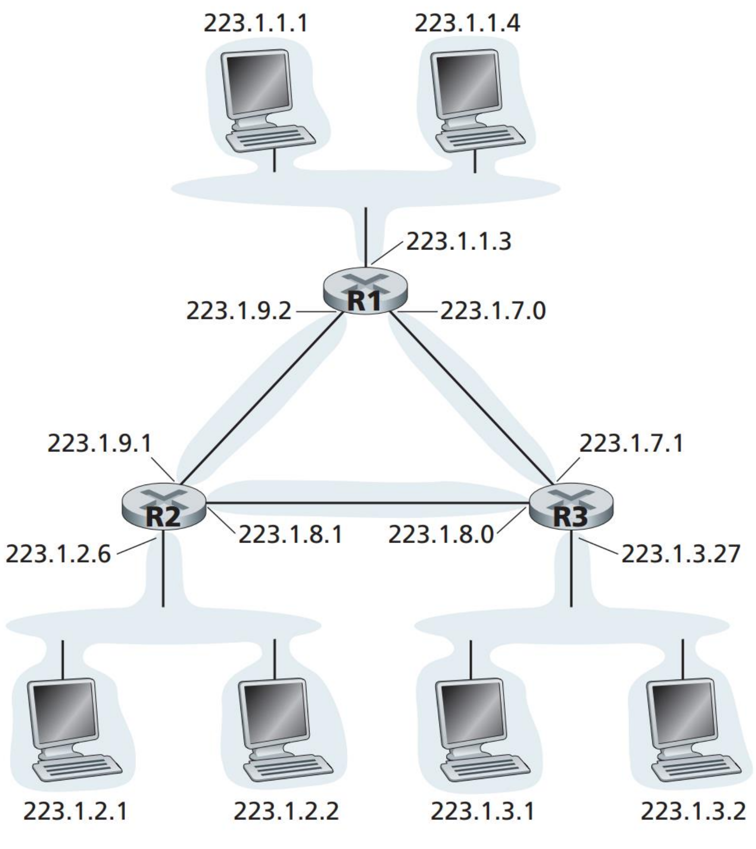



Figure 1 Three routers interconnecting six subnets.

Solution 4:

(1) From 214.97.254/23, possible assignments are

Subnet A: 214.97.255/24 (256 addresses)

Subnet B: 214.97.254.0/25 - 214.97.254.0/29 (128-8 = 120 addresses)

Subnet C: 214.97.254.128/25 (128 addresses)

Subnet D: 214.97.254.0/31 (2 addresses)

Subnet E: 214.97.254.2/31 (2 addresses)

Subnet F: 214.97.254.4/30 (4 addresses)

(2)

Router 1

| Longest Prefix Match | Outgoing Interface |
|---|---|
| 11010110 01100001 11111111 | Subnet A |
| 11010110 01100001 11111110 0000000 | Subnet D |
| 11010110 01100001 11111110 000001 | Subnet F |

Router 2

| Longest Prefix Match | Outgoing Interface |
|---|---|
| 11010110 01100001 11111111 000001 | Subnet F |
| 11010110 01100001 11111110 0000001 | Subnet E |
| 11010110 01100001 11111110 1 | Subnet C |

Router 3

| Longest Prefix Match | Outgoing Interface |
|---|---|
| 11010110 01100001 11111111 0000000 | Subnet D |
| 11010110 01100001 11111110 0 | Subnet B |
| 11010110 01100001 11111110 0000001 | Subnet E |

Problem 5: SDN (15 points)

Consider the SDN OpenFlow network shown in Figure 2. Suppose that the desired forwarding behavior for datagrams arriving at s2 is as follows:

(1) any datagrams arriving on input port 1 from hosts h5 or h6 that are destined to hosts h1 or h2 should be forwarded over output port 2;

(2) any datagrams arriving on input port 2 from hosts h1 or h2 that are destined to hosts h5 or h6 should be forwarded over output port 1;

(3) any arriving datagrams on input ports 1 or 2 and destined to hosts h3 or h4 should be delivered to the host specified;

(4) hosts h3 and h4 should be able to send datagrams to each other.

Specify the flow table entries in s2 that implement this forwarding behavior.

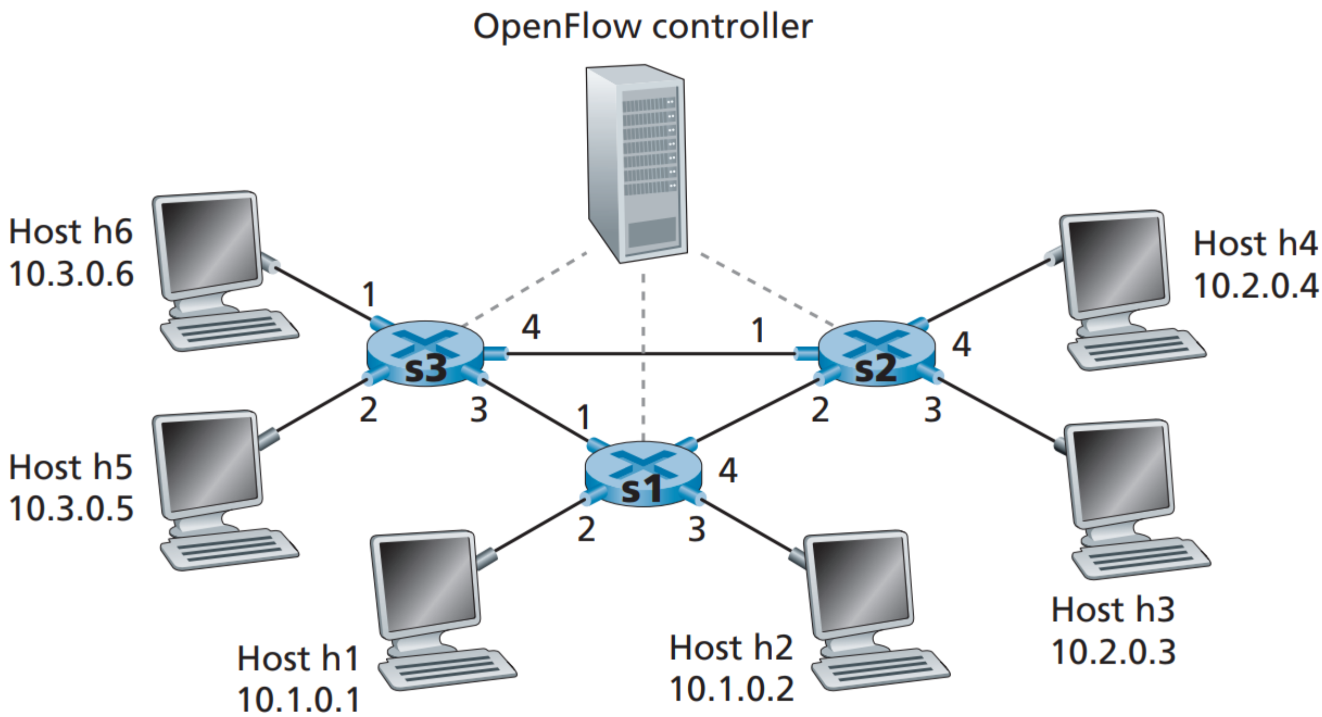


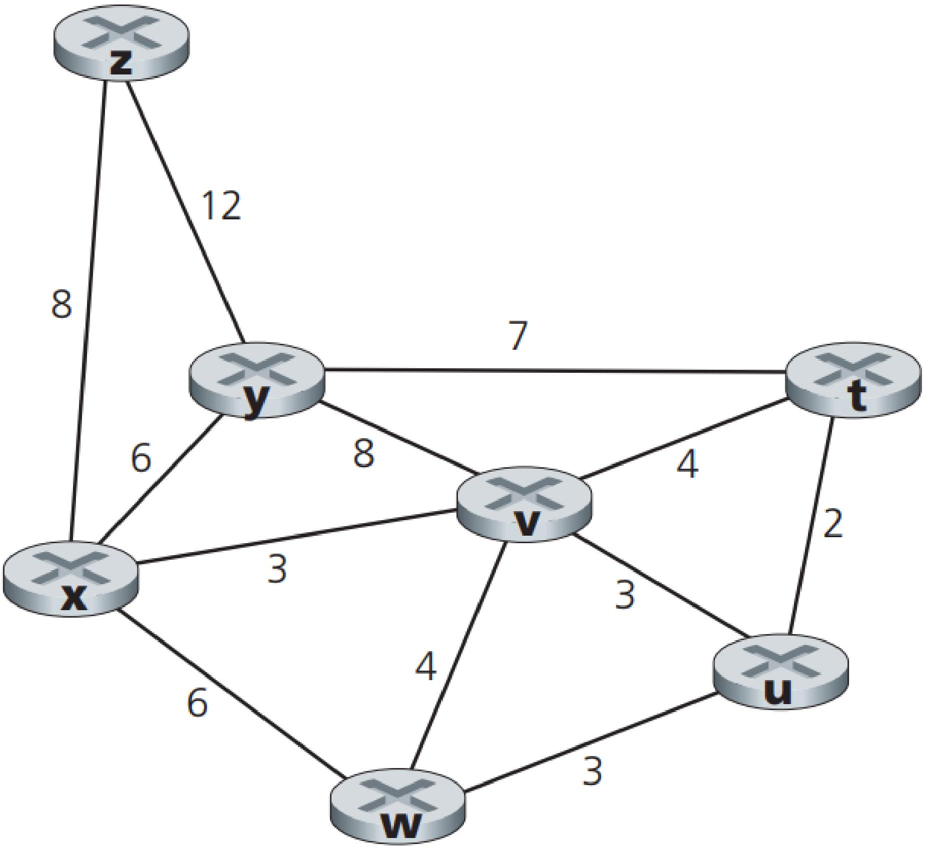

Figure 2 OpenFlow match-plus-action network with three packet switches, 6 hosts, and an OpenFlow controller

Solution 5:

| S2 Flow Table | |
|---|---|
| **Match** | **Action** |
| Ingress Port = 1; IP Src = 10.3.*.*; IP Dst = 10.1.*.* | Forward (2) |
| Ingress Port = 2; IP Src = 10.1.*.*; IP Dst = 10.3.*.* | Forward (1) |
| Ingress Port = 1; IP Dst = 10.2.0.3<br>Ingress Port = 2; IP Dst = 10.2.0.3<br>Ingress Port = 1; IP Dst = 10.2.0.4<br>Ingress Port = 2; IP Dst = 10.2.0.4 | Forward (3)<br>Forward (3)<br>Forward (4)<br>Forward (4) |
| Ingress Port = 4<br>Ingress Port = 3 | Forward (3)<br>Forward (4) |

Problem 6: Link State Routing (10 points)

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 1.




| step | N' | D(v), p(v) | D(w), p(w) | D(x), p(x) | D(y), p(y) | D(z), p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2, u | 5, u | 1,u | ∞ | ∞ |
| 1 | ux | 2, u | 4, x | | 2, x | ∞ |
| 2 | uxy | 2, u | 3, y | | | 4, y |
| 3 | uxyv | | 3, y | | | 4, y |
| 4 | uxyvw | | | | | 4, y |
| 5 | uxyvwz | | | | | |

Table 1. Running the link-state algorithm

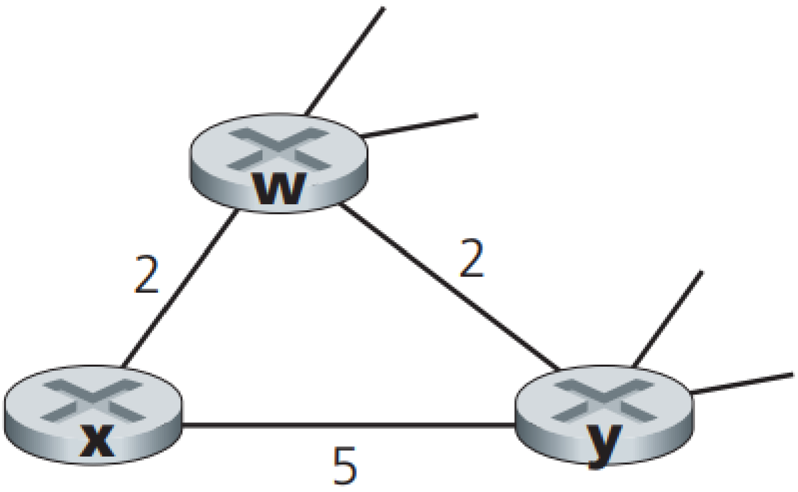
## Solution 6:

**Problem 3**

| *Step* | *N'* | *D(t),p(t)* | *D(u),p(u)* | *D(v),p(v)* | *D(w),p(w)* | *D(y),p(y)* | *D(z),p(z)* |
|---|---|---|---|---|---|---|---|
| 0 | x | ∞ | ∞ | 3,x | 6,x | 6,x | 8,x |
| 1 | xv | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 2 | xvu | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 3 | xvuw | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 4 | xvuwy | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 5 | xvuwyt | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |
| 6 | xvuwytz | 7,v | 6,v | 3,x | 6,x | 6,x | 8,x |

## Problem 7: Distance Vector Routing (10 points)

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.




(1) Give *x*'s distance vector for destinations *w, y,* and *u*.

(2) Give a link-cost change for either $c(x,w)$ or $c(x,y)$ such that *x* will inform its neighbors of a new minimum-cost path to *u* as a result of executing the distance-vector algorithm.

(3) Give a link-cost change for either $c(x,w)$ or $c(x,y)$ such that *x* will *not* inform its neighbors of a new minimum-cost path to *u* as a result of executing the distance-vector algorithm.

Solution 7:

(1) Dx(w) = 2, Dx(y) = 4, Dx(u) = 7

(2) (3)

First, let’s consider the change of c(x, w):

If c(x, w)>6, x will inform its neighbors a new minimum-cost path to u.

If c(x, w)=<6, x will **not** inform its neighbors a new minimum-cost path to u.

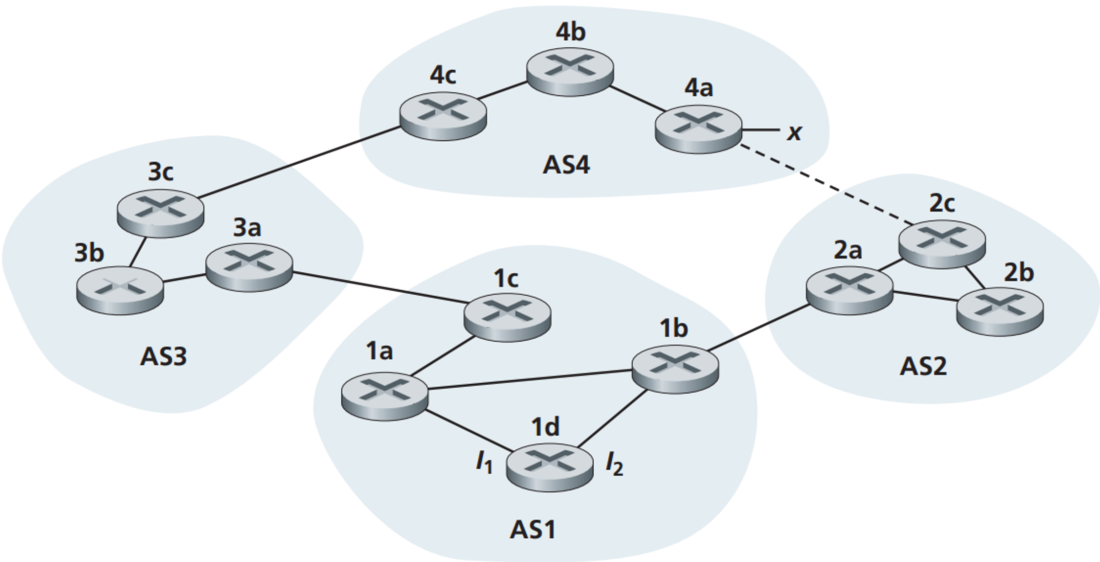
Then, let’s consider the change of c(x, y):

Because c(y, u)=6, c(x, w)=2 and c(w, u)=5, the minimum-cost path from x to u is x→w→u, even if c(x, y) change to minimum cost, i.e. c(x, y)=1.

Therefore, any change in link cost c(x,y) (and as long as c(x,y) >=1) will not cause x to inform its neighbors of a new minimum-cost path to u.

## Problem 8: Border Gateway Protocol (20 points)

Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is *no* physical link between AS2 and AS4.




(1) Router 3c learns about prefix *x* from which routing protocol: OSPF, RIP, eBGP, or iBGP?

(2) Router 3a learns about *x* from which routing protocol?

(3) Router 1c learns about *x* from which routing protocol?

(4) Router 1d learns about *x* from which routing protocol?

Referring to the previous problem, once router 1d learns about *x* it will put an entry $(x, I)$ in its forwarding table.

(5) Will $I$ be equal to $I_1$ or $I_2$ for this entry? Explain why in one sentence.

(6) Now suppose that there is a physical link between AS2 and AS4, shown by the dotted line. Suppose router 1d learns that *x* is accessible via AS2 as well

as via AS3. Will $I$ be set to $I_1$ or $I_2$? Explain why in one sentence.

(7) Now suppose there is another AS, called AS5, which lies on the path between AS2 and AS4 (not shown in diagram). Suppose router 1d learns that $x$ is accessible via AS2 AS5 AS4 as well as via AS3 AS4. Will $I$ be set to $I_1$ or $I_2$? Explain why in one sentence.

Solution 8:

(1) eBGP

(2) iBGP

(3) eBGP

(4) iBGP

(5) $I_1$. because this interface begins the least cost path from 1d towards the gateway router 1c..

(6) $I_2$. Both routes have equal AS-PATH length but $I_2$ begins the path that has the closest NEXT-HOP router.

(7) $I_1$. $I_1$ begins the path that has the shortest AS-PATH.