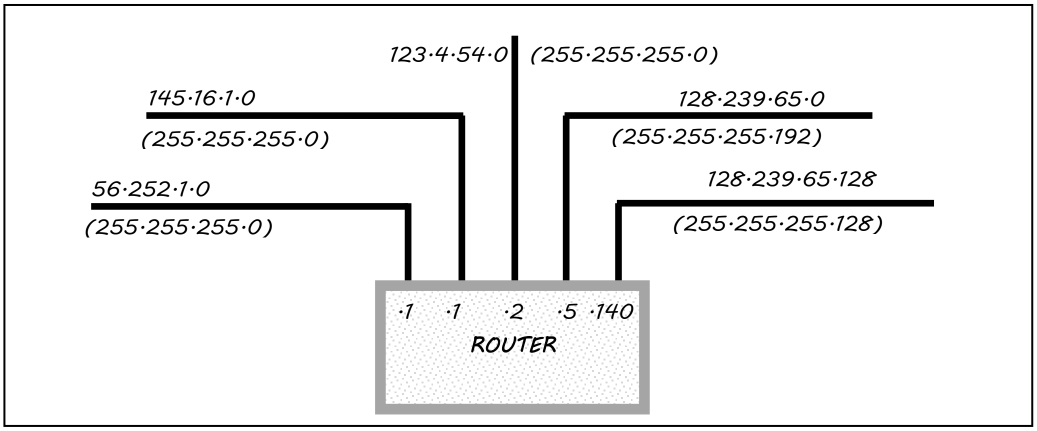
EL5373 Internet Architecture and Protocols

Review Problems – Set Three

Question 1

Consider the network shown below. The subnet masks of the Ethernet segments are given in parenthesis. The IP Addresses of the Ethernet segments are also given.



An IP Datagram arrives at the IP Router shown in the network, on the Ethernet 56.252.1.0. A couple of fields of the IP Header in the datagram are given below.

Source IP	192.16.42.2
Destination IP	178.24.1.56

The Routing Table at the Router is given below.

Destination IP Address	Subnet Mask	Next Hop IP Address	Flags
192.16.42	255.255.255.0	56.252.1.2	H=0;G=1
178.24.1	255.255.255.0	145.16.1.2	H=0;G=1
145.16.1	255.255.255.0	145.16.1.1	H=0;G=0
56.252.1	255.255.255.0	56.252.1.1	
123.4.54	255.255.255.0	123.4.54.2	H=0;G=0
128.239.65.0	255.255.255.192	128.239.65.5	H=0;G=0
128.239.65.128	255.255.255.192	128.239.65.140	H=0;G=0
128.239.65.64	255.255.255.192	123.4.54.1	H=0;G=1
128.239.65.200	255.255.255.255	128.239.65.200	H=1;G=0
default	-	56.252.1.2	G=1

Answer the following questions.

a. On which link is the datagram forwarded?
b. Fill in the shaded cell in the Routing Table above.
c. If a datagram arrives at the Router with the destination address 128.239.65.70, which way is it routed?
d. If a datagram arrives at the Router with the destination address 128.239.65.200, which way is it routed?

Solution

The *Destination IP Address*, together with the *Subnet Mask* in each row of the routing table represents a pool of IP addresses. The row corresponding to the pool, in which the destination IP address of an incoming packet falls, is selected by the router. The incoming packet is then forwarded to the *Next Hop IP Address* in the selected row.

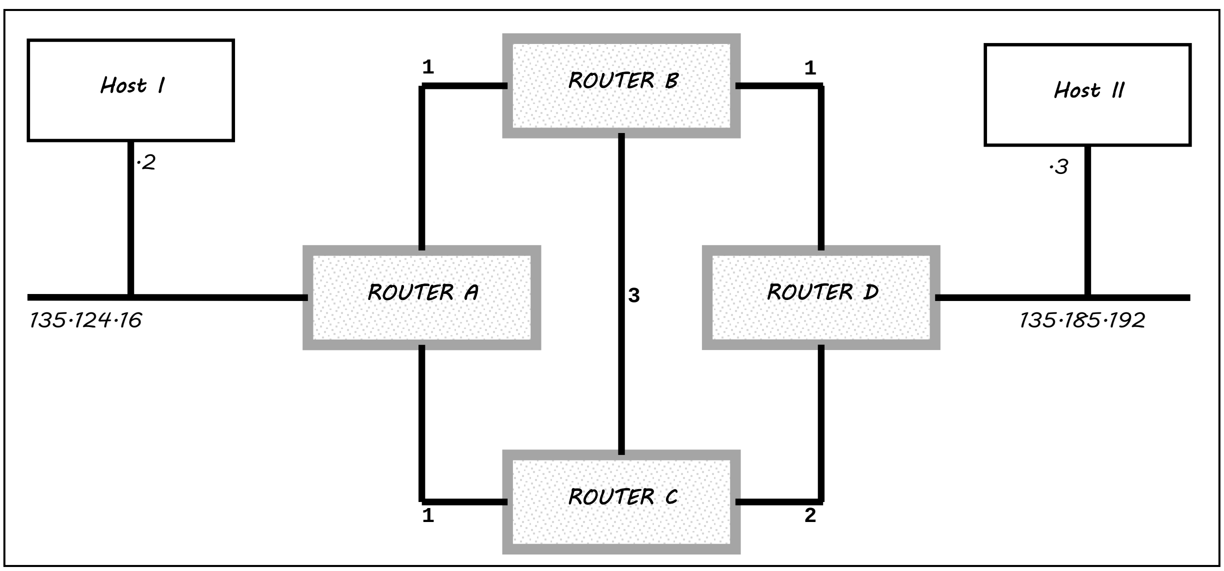
In other words, for every incoming packet, the row in the routing table for which the *Destination IP Address* has the longest matching prefix with the destination IP address of the packet, is selected, provided that the number of matching prefix bits is not less than the number of set bits in the corresponding *Subnet Mask*. If the first n bits of two addresses are the same, it is said that the length of their prefix match is n. This n is calculated for every row, and the one which has the largest n is selected. Also note that this value of n cannot be **less** than the number of *set* bits in the subnet mask of the same row.

With these rules, the following results can be arrived at.

a. The datagram is forwarded to the 145.16.1.0 subnet. The routing entry referred is the one corresponding to 178.24.1.
b. H=0 since the destination address is a subnet, not a host; and G=0 since the interface is directly connected.
c. It is sent on the 123.4.54.0 subnet.
d. It is sent on the 128.239.65.128 subnet.

Question 2

Consider the network given below.



a. Show the distance and routing tables at Router A and Router B if a Distance Vector Routing Algorithm is used.
b. Show the steps of Dijkstra's Algorithm applied to the above network to find the shortest paths from Router B to the other routers. Show the routing tables at Router B. Note that you are using a Link State Routing Algorithm.
c. Is the final routing table at Router B obtained in each of the above cases the same?
d. Compare the two routing algorithms. Give one advantage and one disadvantage of a Link State Routing Algorithm over a Distance Vector Routing Algorithm.

Solution

a. Note that none of the routers has a complete overview of the network. Every router makes its calculations only based on what it hears from its neighbors.

Distance and Routing Tables at node A

Distance Table			Routing Table	
to/via	B	C	via	cost
B	1	3	B	1
C	3	1	C	1
D	2	3	B	2

Distance and Routing Tables at node B

Distance Table				Routing Table	
to/via	A	C	D	via	cost
A	1	4	3	A	1
C	2	3	3	A	2
D	3	5	1	D	1

b. Dijkstra's Algorithm implemented below:

Iteration	M	D_a	D_b	D_c	D_d
1	{B}	1	0	3	1
2	{B, D}	1	0	3	1
3	{B, A, D}	1	0	2	1
4	{B, A, D, C}	1	0	2	1

Paths given below:
to A: B → A
to C: B → A → C
to D: B → D

Thus, Routing table at node B given below:

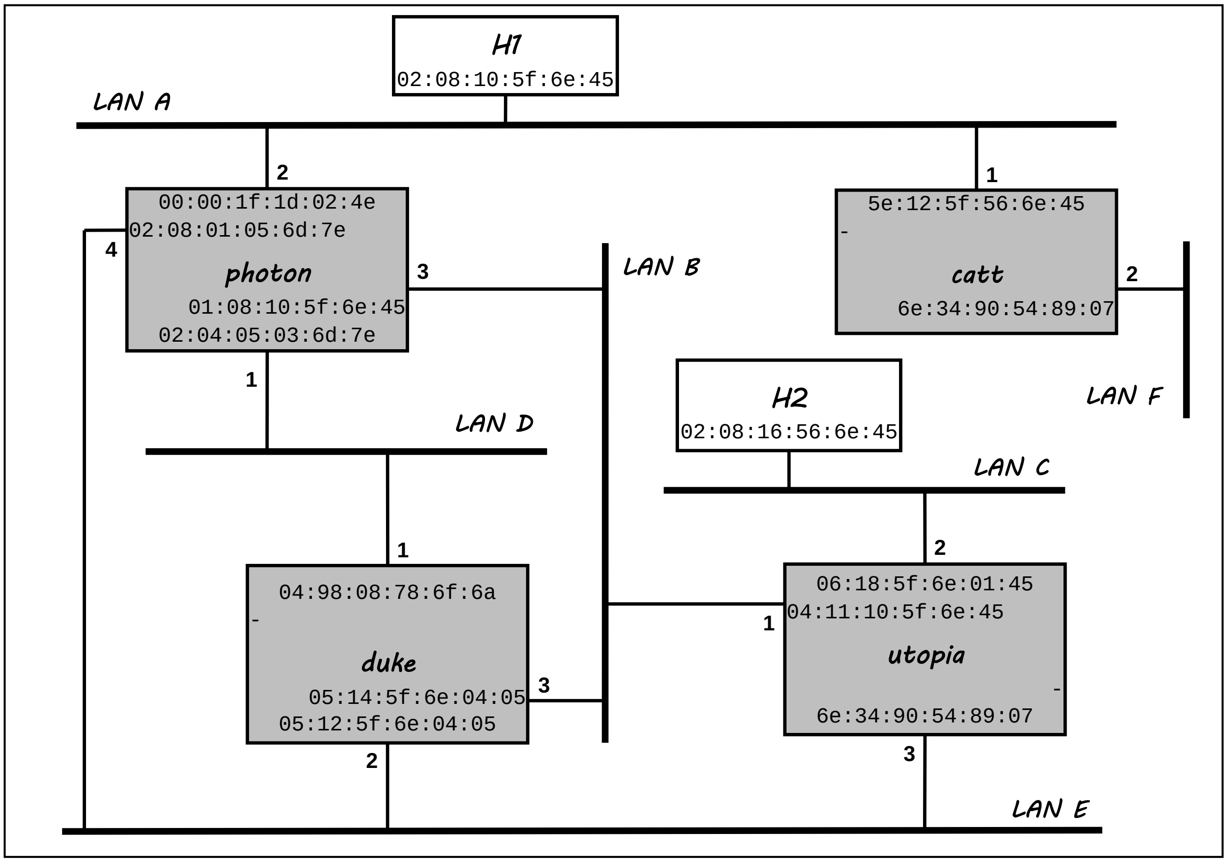
Routing Table		
to	via	cost
A	A	1
C	A	2
D	D	1

c. Yes, they are the same. Since both are Shortest Path Algorithms, they will always end up with the same result as long as there is a unique shortest path.

d. Unlike Distance Vector Routing Algorithms, Link State Routing Algorithms do not have convergence problems. However Link State Routing Algorithms are more complex than Distance Vector Routing Algorithms.

Question 3

Consider the Network of LANs and MAC Bridges shown below.



Assume all LANs are Ethernet LANs. The priorities of the bridges in the above figure are shown below.

Bridge	Priority (0x)
photon	*cd:12*
duke	*ac:12*
utopia	*ac:12*
catt	*bc:15*

Answer the following questions.

a. What are the Bridge Identifiers of the four bridges?
b. Order the four bridges based on their Bridge Identifiers (answered in a.) and assign them logical numbers 1, 2, 3, and 4 in order, with logical number 1 being assigned to the bridge with the lowest Bridge Identifier. Note that now, each bridge has three attributes – the Name, the Bridge Identifier and the Logical Number.
c. What are the initial BPDUs that the bridges generate at $t = 0$? Use the notation for BPDUs used in the example in the slides (<`root id, root path cost, bridge id`>). Use the bridge Logical Numbers instead of Bridge Identifiers for the `root id` and `bridge id` fields of the BPDUs. Note that this is just a short hand to avoid writing the whole Bridge Identifiers all the time. In the actual BPDUs exchanged by the bridges, the whole Bridge Identifiers are used.

d. Assume that the bridges send out BPDUs at $t = 0$, $t = 1$, $t = 2$,... and that the bridges receive the corresponding BPDUs at $t = 0 + \epsilon$, $t = 1 + \epsilon$, $t = 2 + \epsilon$,... respectively. When the bridges receive BPDUs, they compute five parameters. Fill in the shaded regions of the tables below using values of these five parameters generated at each time $t = n + \epsilon$ until the network reaches a stable point (i.e., when there are no further changes in the five parameters computed by each bridge).

Parameters computed at $t = n + \epsilon$

photon		*duke*		*utopia*		*catt*	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
Root Bridge		Root Bridge		Root Bridge		Root Bridge	
Root Port		Root Port		Root Port		Root Port	
Root Path Cost		Root Path Cost		Root Path Cost		Root Path Cost	
New BPDU		New BPDU		New BPDU		New BPDU	
Designated Port(s)		Designated Port(s)		Designated Port(s)		Designated Port(s)	

BPDUs sent at $t = n$

Bridge	BPDU
photon	
duke	
utopia	
catt	

e. Mark the state of each port in the Network figure: use **b** for blocked state and **f** for forwarding state.

f. Draw the spanning tree.

g. If host ***H1*** sends a MAC frame to host ***H2***, list the bridges that receive and/or forward the MAC frame with the ports at which they receive or forward frames respectively.

h. Subsequently, if host ***H2*** sends a MAC frame to host ***H1***, list the bridges that receive and/or forward the MAC frame with the ports at which they receive or forward frames respectively.

Solution

a and b.

Bridge	Identifier	Logical Number
photon	cd:12:02:04:05:03:6d:7e	4
duke	ac:12:04:98:08:78:6f:6a	2
utopia	ac:12:04:11:10:5f:6e:45	1
catt	bc:15:5e:12:5f:56:6e:45	3

c.

BPDUs sent at t=0

Bridge	BPDU
photon	$[4, 0, 4]$
duke	$[2, 0, 2]$
utopia	$[1, 0, 1]$
catt	$[3, 0, 3]$

d. *Parameters computed at t=0+ε, after BPDUs sent at t=0 are received.*

photon		*duke*		*utopia*		*catt*	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
Root Bridge	1	Root Bridge	1	Root Bridge	1	Root Bridge	3
Root Port	3	Root Port	3	Root Port	$-$	Root Port	$-$
Root Path Cost	1	Root Path Cost	1	Root Path Cost	0	Root Path Cost	0
New BPDU	$[1, 1, 4]$	New BPDU	$[1, 1, 2]$	New BPDU	$[1, 0, 1]$	New BPDU	$[3, 0, 3]$
Designated Port(s)	$1, 2$	Designated Port(s)	1	Designated Port(s)	$1, 2, 3$	Designated Port(s)	$1, 2$

BPDUs sent at t=1

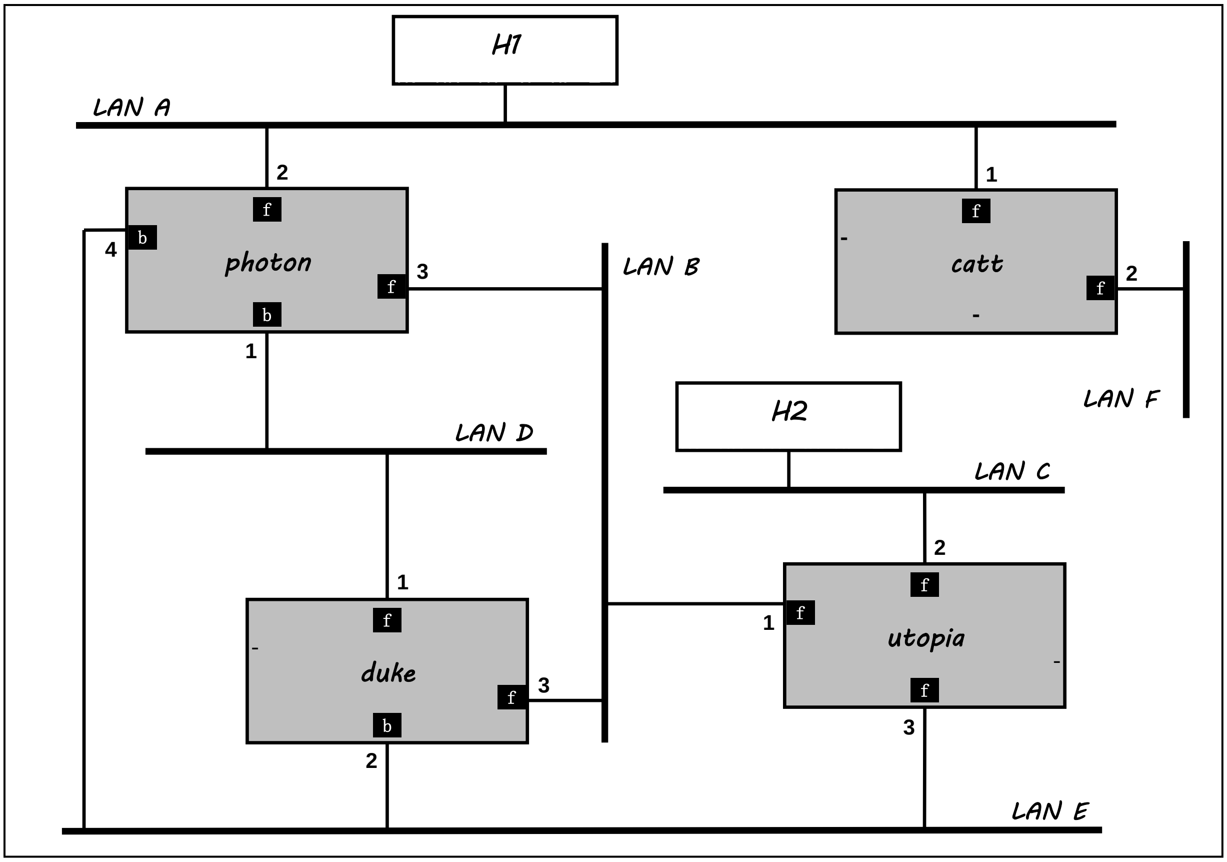
Bridge	**BPDU**
photon	$[1, 1, 4]$
duke	$[1, 1, 2]$
utopia	$[1, 0, 1]$
catt	$[3, 0, 3]$

Parameters computed at t=1+ε, after BPDUs sent at t=1 are received.

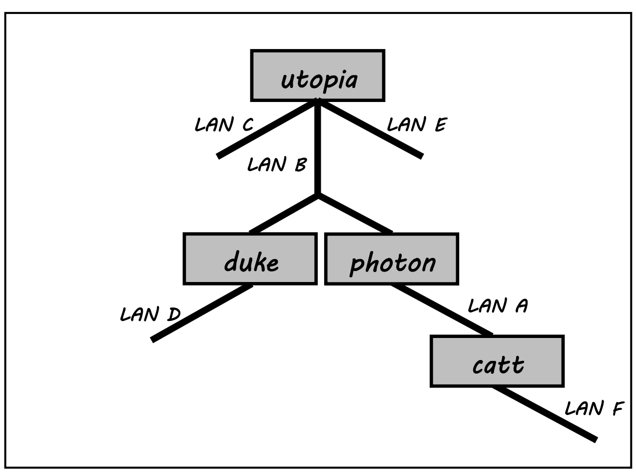
photon		*duke*		*utopia*		*catt*	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
Root Bridge	1	Root Bridge	1	Root Bridge	1	Root Bridge	1
Root Port	3	Root Port	3	Root Port	$-$	Root Port	1
Root Path Cost	1	Root Path Cost	1	Root Path Cost	0	Root Path Cost	2
New BPDU	$[1, 1, 4]$	New BPDU	$[1, 1, 2]$	New BPDU	$[1, 0, 1]$	New BPDU	$[1, 2, 3]$
Designated Port(s)	2	Designated Port(s)	1	Designated Port(s)	$1, 2, 3$	Designated Port(s)	2

Note, this is a stable point, because the next iteration will not change anything.

e. Ports marked below. All Root Ports and all Designated Ports are forwarding. All other ports are Blocking.



f. The Spanning Tree formed by removing the links connected to Blocking ports. Note that the Bridges are the nodes of the spanning tree, and the links are its edges. Hosts simply sit on the links, and are not part of the tree structure. The purpose of the entire process of a. through f. is to arrive at this spanning tree, hence cutting off all loops in the network (tree property) while keeping it still intact (spanning property).



g.

Bridge/Port on which frame is received	Bridge/Port on which frame is forwarded
Photon / 2	Photon / 3
Duke / 3	Duke / 1
Utopia / 1	Utopia / 2,3
Catt / 1	Catt / 2

h.

Bridge/Port on which frame is received	Bridge/Port on which frame is forwarded
Photon / 3	Photon / 2
Duke / 3	DOES NOT FORWARD
Utopia / 2	Utopia / 1
Catt / 1	DOES NOT FORWARD