Computer Networks - HW5

Alireza Rostami
Student Number: 9832090

Part 1

Question 1

With the shadow copy, the forwarding decision is made locally, at each input port, without invoking the centralized routing processor. Such decentralized forwarding avoids creating a forwarding processing bottleneck at a single point within the router.

Question 2

Route summarization, also known as route aggregation, is a method for aggregating multiple specific routes and summarizing them into a singular route. In this method, the network layer addresses are arranged in a hierarchical fashion and each ISP is assigned a continuous block of IP addresses.

It is useful to perform route aggregation in order to further subnet any addresses and assign them to organization in an organized manner. It reduces the number of routes used to access the network, the size of routing tales, and depletion of routes.

Question 3

a. No, you can only transmit one packet at a time over a shared bus.

b. No, as discussed in the text, only one memory read/write can be done at a time over the shared system bus.

c. No, in this case the two packets would have to be sent over the same output bus at the same time, which is not possible.

Question 4

a. eBGP; Router 3c learns about x from eBGP.

b. iBGP; Router 3a learns about x from iBGP.

c. eBGP; Router 1c learns about x from eBGP.

d. iBGP; Router 1d learns about x from iBGP.

Question 5

The below table is the computation of shortest path from source x to all other nodes using the *Dijkstra's algorithm.*

Here,

$$S' = \text{subset of nodes}$$
$$c(i) = \text{Current path of node } i$$
$$l(i) = \text{Least cost path of node } i$$

S'	$l(t)$, $c(t)$	$l(u)$, $c(u)$	$l(v)$, $c(v)$	$l(w)$, $c(w)$	$l(y)$, $c(y)$	$l(z)$, $c(z)$
$\boldsymbol{x}$	∞	∞	3, x	6, x	6, x	8, x
$x\boldsymbol{v}$	7, v	6, v	**3, $\boldsymbol{x}$**	6, x	6, x	8, x
$xv\boldsymbol{u}$	7, v	**6, $\boldsymbol{v}$**	3, x	6, x	6, x	8, x
$xvu\boldsymbol{w}$	7, v	6, v	3, x	**6, $\boldsymbol{x}$**	6, x	8, x
$xvuw\boldsymbol{y}$	7, v	6, v	3, x	6, x	**6, $\boldsymbol{x}$**	8, x
$xvuwy\boldsymbol{t}$	**7, $\boldsymbol{v}$**	6, v	3, x	6, x	**6, $\boldsymbol{x}$**	8, x
$xvuwyt\boldsymbol{z}$	7, v	6, v	3, x	6, x	6, x	**8, $\boldsymbol{x}$**

Table 1 – Table of computation using Dijkstra's algorithm.

Thus, the following are the shortest paths from x and their costs:

Destination	Path	Cost
t	$x \to v \to t$	7
u	$x \to v \to u$	6
v	$x \to v$	3
w	$x \to w$	6
y	$x \to y$	6
z	$x \to z$	8

Table 2 – Table of paths and their costs.

Question 6

Distance vector routing algorithm exchanges the information with the neighbors and works asynchronously.

According to the distance vector algorithm, any node m computes the distance vector using the following formulas:

$$D_m(m) = 0$$
$$D_m(n) = min\{c(m,n) + D_n(n),\ c(m,n) + D_o(n)\}$$
$$D_m(o) = min\{c(m,n) + D_n(o),\ c(m,n) + D_o(o)\}$$

Note: *NA* is used when there is no distance value.
Constructing the distance vector table for node z from the network diagram, we would get:

Node	u	v	x	y	z
v	*NA*	*NA*	*NA*	*NA*	*NA*
x	*NA*	*NA*	*NA*	*NA*	*NA*
z	*NA*	6	2	*NA*	0

Table 3 – Initial distance vector table.

Now, we update the table with costs of all the neighboring nodes.

Node	u	v	x	y	z
v	1	0	3	*NA*	6
x	*NA*	3	0	3	2
z	*NA*	6	2	*NA*	0

TABLE 4 – Updated distance vector table.

Now, we update the table with minimum costs using the distance vector routing algorithm. Example: v to y, two paths are available. $v \rightarrow u \rightarrow y$ and $v \rightarrow x \rightarrow y$ with costs 3 and 6 respectively. So, $v \rightarrow u \rightarrow y$ is the path with minimum cost. Hence, we update the table with this value.

We would finally get the following table:

Node	u	v	x	y	z
v	1	0	3	3	5
x	4	3	0	3	2
z	6	5	2	5	0

TABLE 5 – Final distance vector table.

Part 2

Question 1

TCP slow start.

Question 2

Option 4 is the correct answer.

Question 3

- *Server*:
  - divides video file into multiple chunks.
  - each chunk stored, encoded at different rates.
  - manifest file: provides URLs for different chunks.
- *Client*:
  - periodically measures server to client bandwidth.
  - consulting manifest, requests one chunk at a time:
    - chooses maximum coding rate sustainable given current bandwidth.
    - can choose different coding rates at different points in time (depending on available bandwidth at time).

Question 4

There are three types of switching fabrics:

1. memory
2. bus
3. crossbar

Question 5

Option 2 is the correct answer. `POST` is not used in *HTTP requests*.

Question 5

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$
$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * \mid \text{SampleRTT} - \text{EstimatedRTT} \mid$$
$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

Initially, EstimatedRTT would be equal to the SampleRTT. Thus, we would get the following table:

Segment	*SampleRTT*	*EstimatedRTT*	*DevRTT*	*TimeoutInterval*
1	50	50	25	150
2	100	$0.8 * 50 + 0.2 * 100 = 60$	$0.9 * 25 + 0.1 * 40 = 26.5$	$60 + 4 * 26.5 = 166$
3	200	$0.8 * 60 + 0.2 * 200 = 88$	$0.9 * 26.5 + 0.1 * 112 = 35.05$	$88 + 4 * 35.05 = 228.2$

Table 6 – Computation of TimoutInterval.

Question 6

Option 4 is the correct answer.

Question 7

Interleaving.

- Pros: (i) If some packet is lost, we still have *most* of every original chunk. (ii) No redundancy overhead.
- Cons: (i) Increased playout delay.

Question 8

The below table is the computation of shortest path from source c to all other nodes using the *Dijkstra's algorithm*.

Here,

$$S' = \text{subset of nodes}$$
$$c(i) = \text{Current path of node } i$$
$$l(i) = \text{Least cost path of node } i$$

S'	$l(A)$, $c(A)$	$l(B)$, $c(B)$	$l(D)$, $c(D)$	$l(E)$, $c(E)$	$l(F)$, $c(F)$	$l(G)$, $c(G)$	$l(H)$, $c(H)$
C	∞	7, C	**3, *C***	∞	3, C	∞	∞
CD	∞	7, C		∞	**3, *C***	∞	5, D
CDF	∞	7, C		**5, *F***		∞	5, D
$CDFE$	∞	7, C				6, E	**5, *D***
$CDFEH$	∞	7, C				**6, *E***	
$CDFEHG$	12, G	**7, *C***					
$CDFEHGB$	**9, *B***						
$CDFEHGBA$							

Table 7 – Table of computation using Dijkstra's algorithm.

Thus, the shortest path from C to G is $C \rightarrow F \rightarrow E \rightarrow G$ with the cost 6.

Question 9

Assumptions:

avg object size $= 100K$ bits
avg request rate from browsers to origin servers $= 20$ req/sec
avg data rate to browsers $= 100K$ bits $*$ 20 req/sec $= 2$ Mbps
RTT from institutional router to any origin server $= 2$ sec
cache hit ratio $= 0.3$
access link rate $= 1.2$ Mbps

Thus:

A.

70% of requests use access link.
Data rate to browsers over access link $= 0.7 * 2$ Mbps $= 1.4$ Mbps.
Utilization$_{\text{access link}} = 1.4 / 1.2 = 1.16 > 1$. Thus, the access link cannot satisfy the requests, therefore, we need a fatser access link.

B.

Total delay $= 0.7 *$ (delay from origin servers) $+ 0.3 *$ (delay when satisfied at cache) $= 0.7 * 2 + 0.3 * (\sim 0.001) = 1.4003$ secs.

Question 10

If no advertisement heard after 180 seconds, then neighbor/link is declared dead.

Question 11

Hierarchical OSPF divides the network into two levels: 1. local area, 2. backbone.