

# ICN Written Assignment 2

Student ID: b10201054

May 25, 2025

## Problem 1

(a) The forwarding table is:

Destination Address Range	Link Interface
11100000 00	0
11100000 01000000	1
11100000	2
11100001 0	2
otherwise	3

Table 1: Forwarding Table

(b) 11001000 10010001 01010001 01010101 doesn't match any prefix in destination address range, so the packet would be sent to Link 3.

11100001 01000000 11000011 00111100 matches matches the prefix of Link 1 longer than Link 2, so the packet would be sent to Link 1.

11100001 10000000 00010001 01110111 doesn't matches any prefix in destination address range, so the packet would be sent to Link 3.

## Problem 2

(a) 214.97.254/23 to 32-bit binary is 11010110 01100001 11111110 \*\*\*\*\*.

The first 23 bits is the subnet part, so we can only use the last 9 digits.

Subnet A: 214.97.255/24 Subnet B: 214.97.254.0/25 - 214.97.254.0/29

Subnet C: 214.97.254.128/25

Subnet D: 214.97.254.0/31

Subnet E: 214.97.254.2/31

Subnet F: 214.97.254.4/30

(b) Forwarding table:

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

Table 2: Router 1 Forwarding Table

Longest Prefix Match	Outgoing Interface
11010110 01100001 11111110 1	C
11010110 01100001 11111110 0000001	E
11010110 01100001 11111110 000001	F

Table 3: Router 2 Forwarding Table

Longest Prefix Match	Outgoing Interface
11010110 01100001 11111110 0	B
11010110 01100001 11111110 0000000	D
11010110 01100001 11111110 0000001	E

Table 4: Router 3 Forwarding Table

## Problem 3

- (a) The header of each packet has 20 bytes, MTU = 700 bytes means for each packet we have header 20 + content 680 bytes. Then the 2400 bytes datagram = header 20 + content 2380. The link will cut the datagram in 680, 680, 680, 340 four fragments.
- (b) Identification: 1 1 1 1 fragflag: 1 1 1 0 fragment offset: 0 85 170 255 length: 700 700 700 360

## Problem 4

- (a) We can record all packets sent from the NAT, since the hosts send the packet with sequential numbers and a random initial ID number, we can group the consecutive ID number to a cluster. Then we can find the initial number of the cluster. The number of cluster means the number of distinct initial number, so we will know the number of host through the distinct initial ID number.
- (b) The technique used in (a) is based on the sequential number. Because of the sequential assigned number, we can find the initial ID number in technique (a). If the ID number is randomly assigned, we cannot find the assigned rule of the ID number. Hence, we cannot know which number is the initial ID number of the host.

## Problem 5

## Problem 6

Dijkstra's algorithm table:

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	$\infty$	$\infty$	3,x	6,x	6,x	6,x
1	xv	7,v	6,v	-	6,x	6,x	6,x
2	xvw	7,v	6,v	-	-	6,x	6,x
3	xvwy	7,v	6,v	-	-	-	6,x
4	xvwyz	7,v	6,v	-	-	-	-
5	xvwyzu	7,v	-	-	-	-	-
6	xvwyzut	-	-	-	-	-	-

Table 5: Dijkstra's algorithm table

## Problem 7

- (a)  $D_x(w) = 2, D_x(y) = 4, D_x(u) = 7$
- (b)  $D_x(u) = \min((c(x, w) + D_w(u)), (c(x, y) + D_y(u)))$  If  $c(x, w)$  changes to 1,  $D_x(u) = \min(1+5, 3+6) = 6 < 7$ . Then x will inform its neighbors of a new minimum-cost path to u.
- (c)  $D_x(u) = \min((c(x, w) + D_w(u)), (c(x, y) + D_y(u)))$  If  $c(x, y)$  changes to 5,  $D_x(u) = \min(2+5, 5+6) = 7 = 7$ . Then x will not inform its neighbors of a new minimum-cost path to u since the distance-vector did not change.

## Problem 8

- (a) The time it costs from the source host to the first packet switch is

$$d_1 = \frac{8 \cdot 10^6}{2 \cdot 10^6} = 4 \text{ sec}$$

Since each switch uses store-and-forward packet switching, the message start transmitting after the last bit of the message arrived. Every link costs  $d_1$  sec to transmit the message. Hence, the total time cost is  $3d_1 = 12$  sec.

- (b) The time cost from the source host to the first packet switch is

$$d_1 = \frac{10^4}{2 \cdot 10^6} = 5 \cdot 10^{-3} \text{ sec}$$

The second packet is being sent from the source host to the first switch while the first packet is being sent from the first switch to the second switch. The second

packet has the size 10000 bits. The transmit rate is  $2 \cdot 10^6$  bps. Hence the time cost is

$$d = d_1 + \frac{10^4}{2 \cdot 10^6} = 10^{-3} \text{ sec}$$

After  $10^{-3}$  sec, the second packet is fully received at the first switch.

- (c) Each packet from a switch to the next switch cost  $d = \frac{10^4}{2 \cdot 10^6} = 5 \cdot 10^{-3}$  sec. By (b). we know that we can send packet from the source host to the first switch and from the first switch to the second switch simultaneously.

Hence, when  $t = 2d$ , the first packet arrived the second switch, and the second packet arrived the first switch.

When  $t = 3d$ , the first packet arrived the destination, the second packet arrived the second switch, and the third packet arrived the first switch. Then we can observe that the first packet cost  $3d$  and the second packet will soon arrive destination after  $d$  sec.

The total time cost is

$$d_{total} = 3d + d \cdot (800 - 1) = 802 \cdot 5 \cdot 10^{-3} = 4.01 \text{ sec}$$

To compare the result of (a)., (c). has reserved about 8 sec in transmitting. The reason is because in (a)., we need to wait until the last bit of the data arrived, the first bit can start being transmitted to the next switch. But in (c)., the first packet didn't need to wait the second packet, which reserved the delay in waiting all data transmitted completely.

- (d) We can avoid congestion when transmitting. If we don't use message segmentation, the switch needs 4 sec to send a data in this example, if there is another data which is sent to this switch in 4 sec. The data would stay at input queue. If there are too much data transmitted from other switches, data loss would happen. If we use message segmentation, we can divided the data into several packets, and send them to different switch to transmit, so that each packet would not cost too long in transmitting. The congestion might not happen.