**CSCE 5580 Computer Networks  
Assignment – 3**

1. What is packet scheduling? Explain different types in detail with diagrams and why is it important in the network layer? (5 points)

Text, letter

Description automatically generated

Text, letter

Description automatically generated

2. Give the IP datagram format for IPv4 and IPv6. What is the difference between IPv4 and  
IPv6 with structures? Which is faster and why? (5 points)

Text, letter

Description automatically generated

Text, letter

Description automatically generated

3. What is NAT? Explain in detail how it works with changes in addresses in the router.  
(5 points)

Text, letter

Description automatically generated

Text, letter

Description automatically generated  
4. Explain the Tunneling and encapsulation mechanism in IPv6? (5 points)

Text, letter

Description automatically generated5. What is software-defined networking, and What are the benefits of software-defined  
networking? (5 Points)

Text

Description automatically generated

Text, letter

Description automatically generated

6. Consider the following network. With the indicated link costs, use Dijkstra’s shortest-path  
algorithm to compute the shortest path to all the nodes. Show how the algorithm works by  
computing a table below (Look slides for table Example). (10 Points)  
Diagram, engineering drawing

Description automatically generated

Diagram

Description automatically generated with medium confidence

**3**

**2**

**2**

**1**

**1**

**2**

7. Consider the network shown below and assume that each node initially knows the costs  
to each of its neighbors. Consider the distance-vector algorithm and show the distance  
table entries at: (10 Points)

Diagram

Description automatically generated with low confidence

Method 1:

Text

Description automatically generated

Method 2

My name starts with “N” hence I am going to find distance table entries at Y

The routing table at Y initially will be :

|  |  |  |
| --- | --- | --- |
| **Network** | **Cost** | **Next router** |
| u | 2 | u |
| v | ∞ | - |
| x | 3 | x |
| z | ∞ | - |
| y | 0 | y |

Z's updated table after receiving x and u table can be calculated by first calculating u and x initial tables :

u's initial table will be:

|  |  |  |
| --- | --- | --- |
| Network | Cost | Next router |
| u | 0 | u |
| v | 1 | v |
| x | ∞ | - |
| z | ∞ | - |
| y | 2 | y |

x's initial table will be :

|  |  |  |
| --- | --- | --- |
| Network | Cost | Next router |
| u | ∞ | - |
| v | 3 | v |
| x | 0 | x |
| z | 2 | z |
| y | 3 | y |

To calculate Y's updated table:

Y will receive the distance - vector from u and x as :

Distance from y to u

Min { y to u, y to x -> x to v -> v to u}

Min { 2 , 3 + 3 +1} = Min { 2 , 7}

= 2

Distance from y to v

Min { y to u -> u to v, y to x -> x to v }

Min { 2 + 1 , 3 + 3 } = Min { 3 , 6}

= 3

Distance from y to x

Min { y to x, y to u -> u to v -> v to x}

Min { 3, 2 + 1 + 3 } = Min { 3 , 6}

= 3

Distance from y to z

Min { y to x -> x to z, y to u -> u to v -> v to z}

Min { 3 + 2, 2 + 1 + 6 } = Min { 5 , 9}

= 5

Distance from y to y is 0

So final distance routing table of y after receiving distances from u and x will be:

|  |  |  |
| --- | --- | --- |
| Network | Cost | Next router |
| u | 2 | u |
| v | 3 | u |
| x | 3 | x |
| z | 5 | x |
| y | 0 | y |

8. 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: (5  
Points)  
Calendar

Description automatically generated with medium confidence  
a. Provide a forwarding table that has four entries, uses the longest prefix matching, and  
forward packets to the correct link interfaces.

|  |  |
| --- | --- |
| Destination Address Range (longest prefix matching) | Outgoing Link Interface |
| 11110000 00\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* | 3 |
| 11100000 0100000\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* | 2 |
| 1110000\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* | 1 |
| Otherwise | 0 |

b. Describe how your forwarding table determines the appropriate link interface for datagrams  
with destination addresses:

11100001 01000000 11000011 00111100

link interface: 1

11100001 11110000 00010001 01110111

link interface: 0

11110000 00010001 01010001 01010101

link interface: 3

11100000 01000000 00010000 00100100

link interface: 2

00000000 00000000 00000000 00000000

link interface: 0

c. Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

|  |  |
| --- | --- |
| Destination Address Range (longest prefix matching) | Outgoing Link Interface |
| 240.0.0.0/10 | 3 |
| 224.64.0.0/16 | 2 |
| 224.0.0.0/8 | 1 |
| 225.128.0.0/9 | 0 |
| Otherwise | 0 |

The above-highlighted row can be excluded.