King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Department of Computer Engineering

COE 344 – Computer Networks (T172)

Homework # 04 (*due date & time: Tuesday 10/04/2018 during class period*)

Late homework submission will NOT be accepted

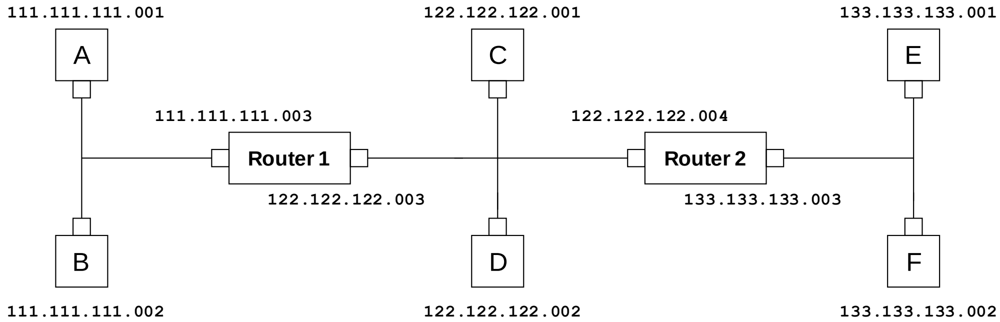
***** Show all your work. No credit will be given if work is not shown! *****

Problem # 1 (12 points): Consider a datagram network using IPv4 addresses. Suppose a router has five links, numbered 0 through 4, and datagrams are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 00001111 11111111	1
11100000 01000000 00010000 00000000 through 11100000 01000000 11111111 11111111	2
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	3
otherwise	4

a. Provide an equivalent forwarding table that translates each of the given ranges into a prefix of the form ***a.b.c.d/x*** along with the associated link interface. The forwarding table should have six entries (2 for the "otherwise" range, and 4 for the other ranges).
b. Determine the appropriate link interface for forwarding datagrams with the following destination addresses:
    i. 11100000 01010000 11000011 00111100
    ii. 11100001 10000000 00010001 01110111
    iii. 11100000 01000000 00001000 11111111

Problem # 2 (5 points): Consider the following IP-based network with the assigned IP addresses as shown. For each of the following cases, complete the table regarding the datagram as it is forwarded from the source to the destination.



1. Assume that host ***A*** sends an IP datagram to host ***C***.

Source IP address	Destination IP address	Receiving interface IP address that was passed down to the Data Link layer to be used for forwarding

2. Assume that host ***A*** sends an IP datagram to host ***E***.

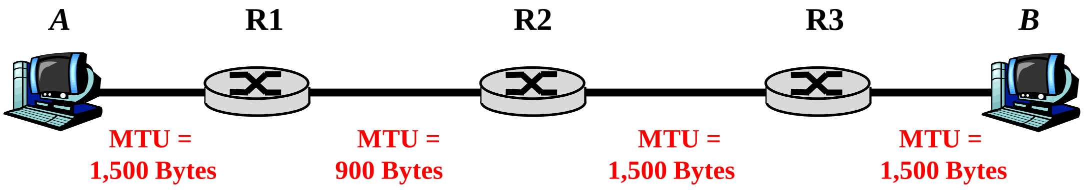
Source IP address	Destination IP address	Receiving interface IP address that was passed down to the Data Link layer to be used for forwarding

---

Problem # 3 (6 points): Consider a router that interconnects 3 subnets: **Subnet 1**, **Subnet 2**, and **Subnet 3**. Suppose all of the interfaces in each of these 3 subnets are required to have the prefix **`224.1.18/24`**. Also suppose that **Subnet 1** is required to support 60 interfaces, **Subnet 2** is to support 11 interfaces, and **Subnet 3** is to support 91 interfaces. Provide three network addresses of the form ***`a.b.c.d/x`*** that satisfy these constraints.

Problem # 4 (3 points):

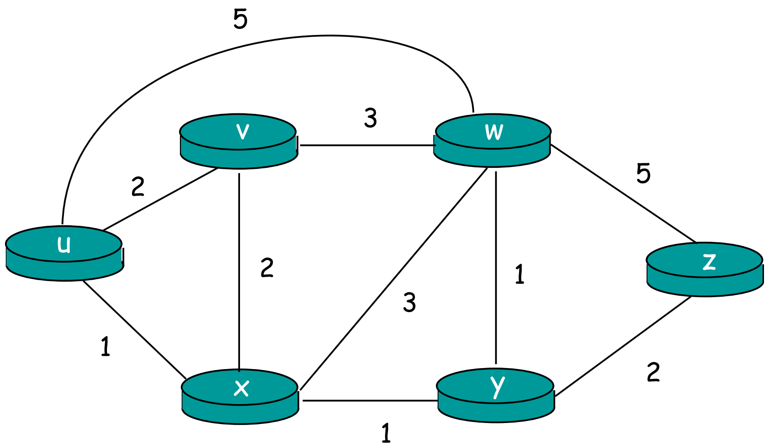
Consider sending a **1500**-byte IP datagram (inclusive of a minimum size header of 20 bytes) from host ***A*** to host ***B*** along the path shown in the following figure. Assume that the MTU of each of the shown links is inclusive of a minimum size header of 20 bytes.



1. **(1 point)** Find the total number of fragments generated by router **R1**.
2. **(2 points)** Specify the ***header's "offset" field value*** of each fragment that is generated by router **R1**.

Problem # 5 (12 points):

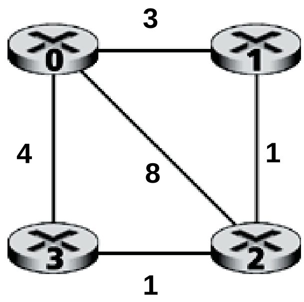
Consider the following network.



With the indicated link costs, use *Dijkstra*'s shortest-path algorithm, as discussed in class, to compute the shortest path from ***w*** to all network nodes using the table given below.

N'	*D(u),p(u)*	*D(v),p(v)*	*D(x),p(x)*	*D(y),p(y)*	*D(z),p(z)*

Problem # 6 (12 points): Consider the following network.



Suppose that the link cost $c(0,2)$ has changed from 8 to 1, re-compute the distance tables for nodes 0, 1, 2, and 3 after only two iterations of a synchronous version of the distance vector algorithm. Note that the current tables' values *prior* to the link cost change are as shown in the leftmost column of the tables.

cost to (columns), from (rows)

D^0	0	1	2	3
0	0	3	4	4
1	3	0	1	2
2	4	1	0	1
3	4	2	1	0

cost to (columns), from (rows)

D^0	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^0	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^1	0	1	2	3
0	0	3	4	4
1	3	0	1	2
2	4	1	0	1
3	∞	∞	∞	∞

cost to (columns), from (rows)

D^1	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^1	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^2	0	1	2	3
0	0	3	4	4
1	3	0	1	2
2	4	1	0	1
3	4	2	1	0

cost to (columns), from (rows)

D^2	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^2	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^3	0	1	2	3
0	0	3	4	4
1	∞	∞	∞	∞
2	4	1	0	1
3	4	2	1	0

cost to (columns), from (rows)

D^3	0	1	2	3
0				
1				
2				
3				

cost to (columns), from (rows)

D^3	0	1	2	3
0				
1				
2				
3				