HY-335a Computer Networks
Χρηστος Παπασταμος csd4569
Assignment 4

Part 1

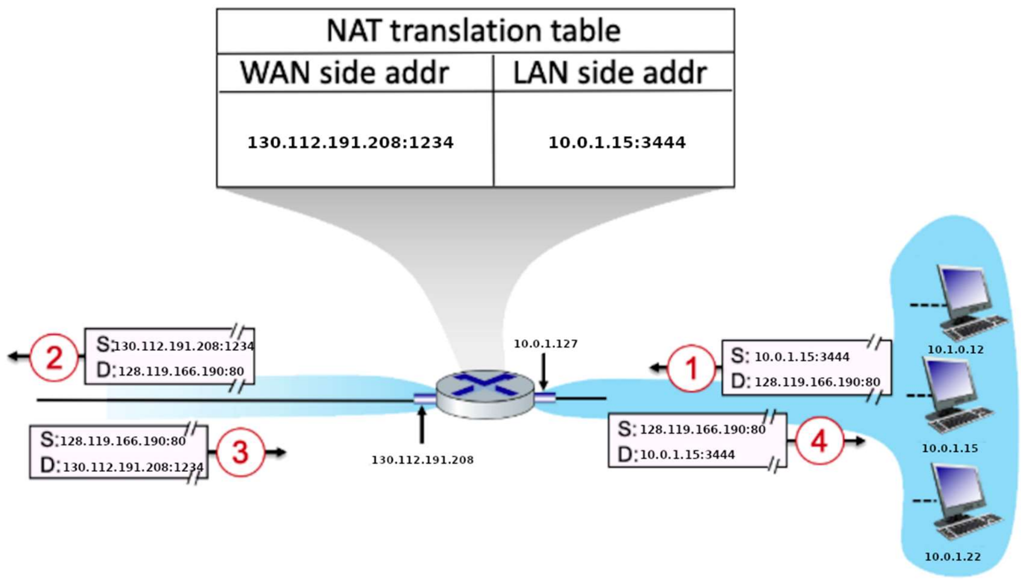
Question 1

a) The main deference between link state and distance vector is that on distance vector the algorithm selects the path with less hops (or with less distance if it is measured) where as in link state, the algorithm selects the fastest route depending on the speed (state) of each link.

b)BGP is kind of a distance vector protocol, as it calculates its paths based mostly on the hops to each destination.

Question 2

Nowadays the number of the devices that need to be connected to the internet is well above the max capacity of the ipv4 protocol. Therefore, there had to be a way to decrease this number of devices. NAT (short for Network Address Translation) is used to assign local ip addresses to all devices connected to one router, and "make them all" use one public IP address visible to the global internet. Like the example, the NAT table redirects all incoming traffic (from a port, not in the example tho) to a local IP address, and same goes for outgoing traffic.



NAT translation table

WAN side addr	LAN side addr
130.112.191.208:1234	10.0.1.15:3444

Question 3

A) 1,3
B) 2,4
C) 1
D) 1,2,5
E) 1,2,4,5

Question 4

a. Time: t=1 Packet:1
Time: t=2 Packet: 2
Time: t=3 Packet: 3
Time: t=4 Packet: 4
Time: t=5 Packet: 5
Time: t=6 Packet: 6
Time: t=7 Packet: 7
Time: t=8 Packet: 8

b. Time: t=1 Packet:1
Time: t=2 Packet: 2
Time: t=3 Packet: 3
Time: t=4 Packet: 4
Time: t=5 Packet: 5
Time: t=6 Packet: 8
Time: t=7 Packet: 6
Time: t=8 Packet: 7

Question 5

a.

Destination Address	Link
11100000 00****** ******** ********	0
11100000 01000000 ******** ********	1
11100000 01000001 ******** ********	2
11100000 01****** ******** ********	2
11100001 0******* ******* ********	2
default	3

b. 1. This datagram will be forwarded to link 3 because the 3rd LS bit is 0 while on all entries of our forwarding table the first 3 bits are 1. That is why this datagram will be forwarded to the default route

2. This datagram will be forwarded to link 1 because, according to the longest prefix match, the b part of its address matches only to the second line of the forwarding table

3. This datagram will be forwarded to link 2 because due to the 8th LS bit, it only matches to the 5th line of the forwarding table

4. This datagram will be forwarded to link 3 due to the 9th LS bit being a 1, which makes it not match any line of the forwarding table. In that case it goes on the default link 3

5. This datagram will be forwarded to link 1 because the 16 LS bits match only to the second line of the forwarding table

Part 2

Part 3

Processed nodes	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(Z),p(Z)
A	3,A	4,A	∞	∞	32,A
AB	-	4,A	9,B	8,B	32,A
ABC	-	-	9,B	5,C	32,A
ABCE	-	-	7,E	-	17,E
ABCED	-	-	-	-	14,D
ABCEDZ	-	-	-	-	-

PART 4

1. The main reason a router will retransmit its distance-vector table is if it finds a shortest path to a router and updates its path

2. The nodes of the network will exchange vectors 3 times, concluding to the following vectors

A:

	A	B	C	D
A	0	6	2	3
B	6	0	4	3
C	2	4	0	1

Paths from A:
B: A→C→D→B
C: A→C
D: A→C→D

B:

	A	B	C	D
A	0	6	2	3
B	6	0	4	3
D	3	3	1	0

Paths from B:
A: B→D→C→A
C: B→D→C
D: B→D

C:

	A	B	C	D
A	0	6	2	3
C	2	4	0	1
D	3	3	1	0

Paths from C:
A: C→A
B: C→D→B
D: C→D

D:

	A	B	C	D
B	6	0	4	3
C	2	4	0	1
D	3	3	1	0

Paths from D:
A: D→C→A
B: D→B
C: D→C

3. The count to infinity problem will occur in our example if the nodes C and D get disconnected. In that case, router D will try to calculate a path to C by adding c(D,B) + D_B(C)=3+4=7. Then D will retransmit its d-v table and B will update its value to C witch c(B,D) + D_D(C)=3+7=10. The retrasmition will repeat forever counting to infinity. This problem can be resolved if node D would set its value to node C as infinite. That would cause a retransmition of its table, informing anyone having a route to C trough D that this route is no longer available (distance is infinite).

PART 5

First of all, lets subtract the header length from all the MTUs and the datagram size:

- Datagram size=5000Bytes-20B header=4980Bytes
- L_a MTU=6000B-20B header=5980B
- L_b MTU=1500B-20B header=1480B
- L_c MTU=580B-20B header=560B

Link **a** doesn't need any fragmentation, the packet is transmitted as one as it is smaller than the link's MTU

Link **b** needs to slice the datagram in datagram size/L_c MTU= 4980/1480=3,3=**4 fragments**

- F_b1 = 1480B+20B header, offset=0
- F_b2 = 1480B+20B header, offset=1480
- F_b3 = 1480B+20B header, offset=2960
- F_b4 = 540B+20B header, offset=4440

Link **c** needs to slice the first three packets that came from Link B in 1480/580=2,5=**3 fragments each**, the packet F_b4 doesn't need any fragmentation since it is smaller than the link's MTU

F_b1 will get sliced into:

- F_c1=560B+20B header, offset=0
- F_c2=560B+20B header, offset=560
- F_c3=360B+20B header, offset=1120

F_b2 will get sliced into:

- F_c4=560B+20B header, offset=1480
- F_c5=560B+20B header, offset=2040
- F_c6=360B+20B header, offset=2600

F_b3 will get sliced into:

- F_c7=560B+20B header, offset=2960
- F_c8=560B+20B header, offset=3520
- F_c9=360B+20B header, offset=4080

And last of all F_b4 which will get transmitted without fragmentation:

- F_c10=540B+20B header, offset=4440