Ans 1

(a) There are following reasons:

(*) AS numbers are not included in IP header

(*) And routing is still done towards a network number, not towards an AS#.

→ So main routers should be aware of all network numbers.

(b) OSPF has some limitations handling a large number of networks (scaling issues).

→ Also, to reach all network outside the AS, all network prefixes should be have to be in either BGP or OSPF.

→ If, only border router speaks BGP, then we have to inject all network prefixes to the OSPF, which is not at all efficient.

→ So, all router speaks BGP in core.

(c) First and main reason is: to avoid loops.

Second reason: Many times, we require flexible routing policies (i.e flexible routes).

→ By knowing the path you can decide if you like or not to go to which path.

→ ex: Let say I don't want to go through AS owned by XYZ company. In that case, whole path bol

(d) To do ARP.

→ To translate from IP to physical address we need broadcast.

Q.2

Ans-2 Here we first address issue about u_0's existance

→ Consider any node n that is not fixed (its path changes). Let path P in values (C, w)

→ As we know C is cycle. $P = (n, \ldots, w, v, \ldots 0)$

v is first fix value. w is not fixed.

→ Also $n = w$ and $v = 0$ can be possible.

(*) As lemma says, w can take some time during cycle the path $= (w, v, \ldots 0)$. Thus w is the u_0, because one of its paths in values (C, w) goes directy to a fixed node

(*) Now, $Q_0 = (u_0 v_0 \ldots 0)$ where v_0 is fixed. Assume we had another path $P = (u_0, w_1 \ldots)$ where w is also fixed. So both path should be accible to u_0. So one must be ranked higher than other and u_0 would pick only one of these, that says another will never get picked up. Hence both Q_0 and P cannot be these, so Q_0 is unique.

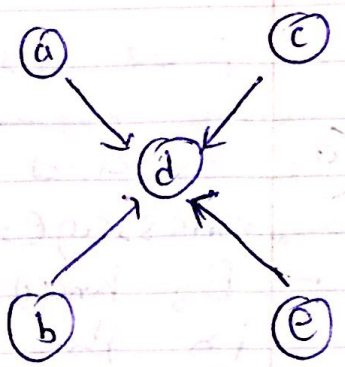
(*) Now, $Q_0 = (u_0 v_0 \ldots)$ goes through a fixed nodes. So it is always available. the only way another path R will be taken by u_0 is if R is ranked higher than Q_0.

→ Therefore Q_0 is ranked lowest in all path. // Ans

A-3

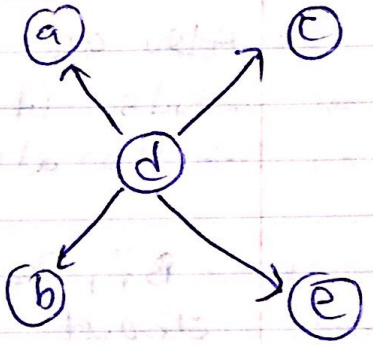
(i)

→ a, c, b, e all are service provider to d
∴ d can reach all.



→ But a, c, b and e cannot reach each other. They can reach only d.

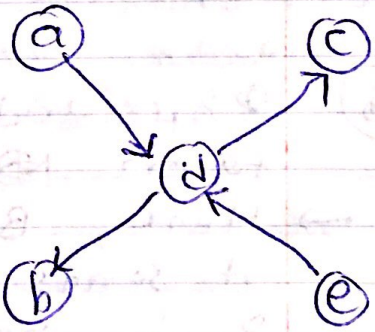
(ii) a, b, c and e all are customer of d.
∴ everyone can reach each other's AS in the system



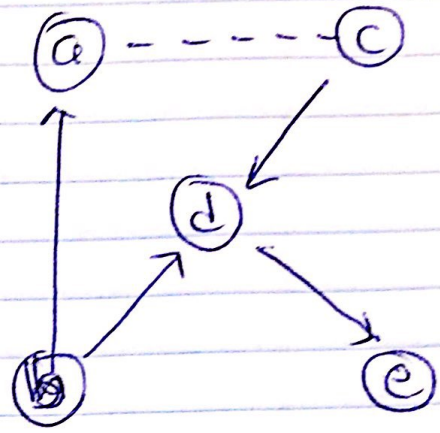
(iii) a, e are service provider to d, while b and c are customer to d.

∴ b and c can reach everyone else

Also, a, e can reach ~~each other~~ b, d, c but not each other.



Q3 (iv) AQ



- a can reach b, c directly
- c can export its customes to a. So a can reach d and e.

∴ a can reach all.

* b

b can reach a, d, e as customers.
b can't reach c, ~~with neighbor~~ via a or d.

* c

c can reach a (peer), d, e (customes)
c can't reach b.

* d

d can reach b, c (provider), e (customers) and a (peer of c ~~as~~ as well as customer of b)

d can reach all.

* e

a is ~~not~~ neighter peer of d, ~~and p~~ nor provider of d.
∴ ~~even if~~ d knows about a, it can't export to e.
~~→ e knows about b, c and d only.~~
→ ~~But~~ Also a is customer of b, so as e. therefore e knows about a via b, d.
∴ e can also reach all.

A-5

(a) B_2 and B_3 will ~~[illegible]~~ listens to the message.

→ B_3 knows about K, so it forwards it in the direction of B_3, which forwards it to B_4, which forwards it to B_5 as all knows about K.

→ Also as B_2 don't know about K, it sends it in all direction.
∴ B_1 also learns about N.

∴ B_1, B_2, B_3, B_4 and B_5 learns about N.

(b) B_1 and B_2 listen to the message from M.

→ Here, B_1 don't know about N, so it sends in all direction. But as B_1 is not connected to any other bridge, only B_1 learns about M.

→ As B_2 know about N, it sends it to Lan where N is located.

→ B_3 is also on same LAN. So, B_3 also learns about M.

∴ B_1, B_2 and B_3 learns about M.

(c) I will be sending message to J on its previous location, ~~until one of following happens.~~ but as J not available there any more messages will be lost.

→ In case of one of following event, situation will change:

(+) If J sends enough message, that can reach to location of I - then I will get aware of new location of J.

(+) If the information of J times out from the cache of bridge, then only all bridge will forget about J.

→ In that case I will flood messages to get to the J again.

A-6 Yes all this subnet can be part of same company.

→ Here we can divide section of network 129.56.0.0 as below:

(i) 129.56.0.0 to 129.56.3.255 range
- which can be subdivided as below

A → 129.56.1.0 - 129.56.1.255
B → 129.56.2.0 - 129.56.2.255
C → 129.56.3.0 - 129.56.3.255

Each A, B and C here gives one of our required subnet range.

→ A can be represented as 129.56.1.0/24 which gives 129.56.1.0 subnet
→ B can be represented as 129.56.2.0/24 which gives 129.56.2.0 subnet
→ C can be represented as 129.56.3.0/24 which gives 129.56.3.0 subnet.

And A, B and C do not override any network/host Id with each other.

(ii) 129.56.4.0 - 129.56.7.255 range

→ This can be represented as 129.56.4.0/22, which gives subnet 129.56.4.0.

(iii) 129.56.8.0 - 129.56.11.255 range.

→ This can be represented as 129.56.8.0/22, which gives subnet 129.56.8.0.

(iv) 129.56.12.0 - 129.56.15.255 range

→ This can be represented as 129.56.12.0/22, which gives subnet 129.56.12.0.

⇒ Here any of A, B, C, (ii), (iii) and (iv) do not override any of network/host id in each other.

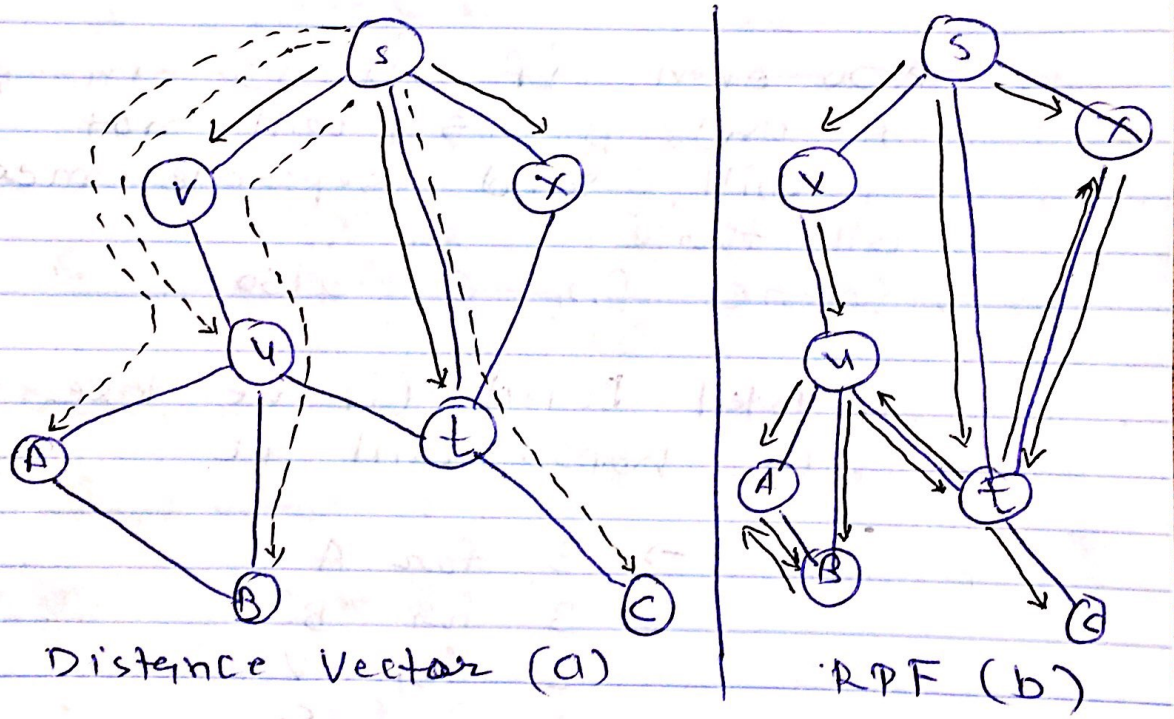
→ Also if we go minimum in any subnet mask, like 24 → 23 or 22 → 21, then that subnet will surely override networkId of any of other subnet. So finilized subnet with mask are as below:

129.56.4.0 /22
129.56.2.0 /24
129.56.12.0 /22
129.56.8.0 /22
129.56.3.0 /24
129.56.1.0 /24.

Ans

A-7 If nodes use distance vector routing, in my opinion it is not possible to get reduced number of messages.

→ I will try to explain using example used in slide "DVMRP and MOSPF", slide no: 26.



Distance Vector (a) RPF (b)

→ Let say as we know RPF takes

$2E - (n-1) = 2 * 10 - (8-1)$

$= 20 - 7$

$= 13$ message to broadcast msg in given figure (b)

→ If we will replace RPF with

distance vector routing, then messages sent to all node can be counted as below:

→ S knows path only about its neighbour V, X and t.

→ For others (u, A, B, C) it does not know about path, as their neighbour will give direct cost to them.

→ So even if u is in path of A and B, S will not know and S will send separate messages for all three.

→ Same for C also.

∴ total count (if we take measurement in hops) will be:

→ 3 for A
3 for B
2 for V
2 for C
1,1,1 for V, X and t.

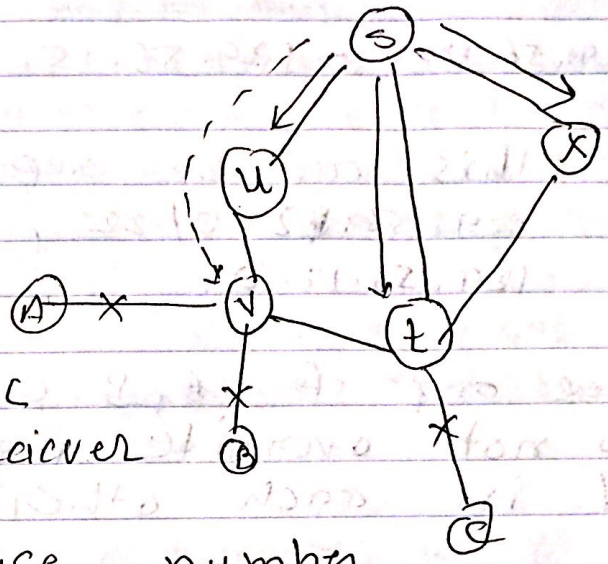
Total = 3+3+2+2+3 = 13 which is same as RPF.

→ But when system will grow, this will grow even more with distance vector, as there will be duplicates for all non-neighbour elements.

→ ∴ RPF is better than Distance Vector.

* But if we use DVMRP, then we can reduce message count.

→ With DVMRP, we remove leaf nodes and in that case all end nodes which don't have receiver can be removed.

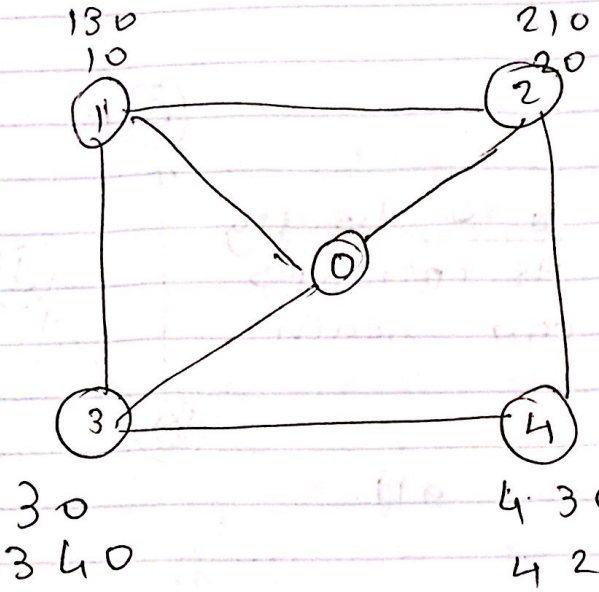


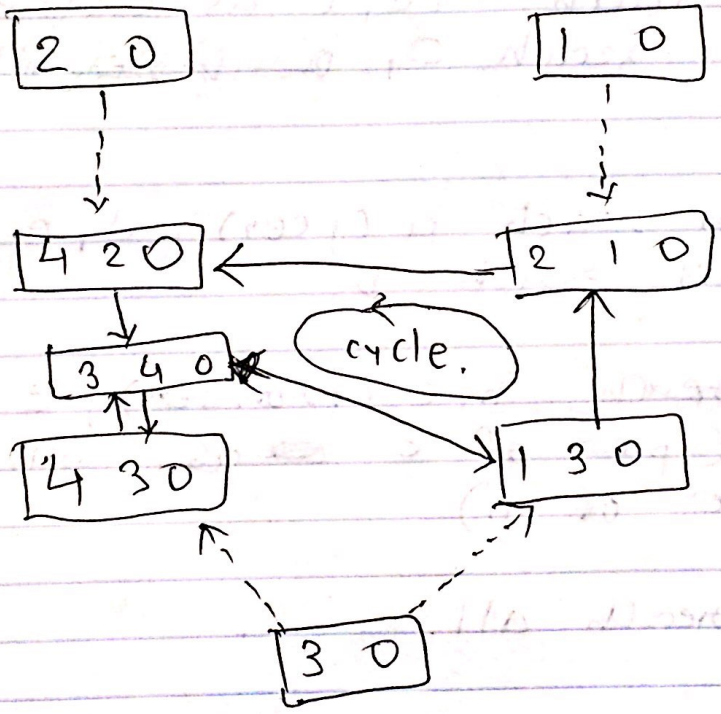
→ Here we assume A, B, C don't have reciever

⇒ In that case number of message done using simple Distance vector routing can be limited to s, u, v, t and x.

⇒ So DVMRP can be better than RPF here in terms of number of messages.

A-6



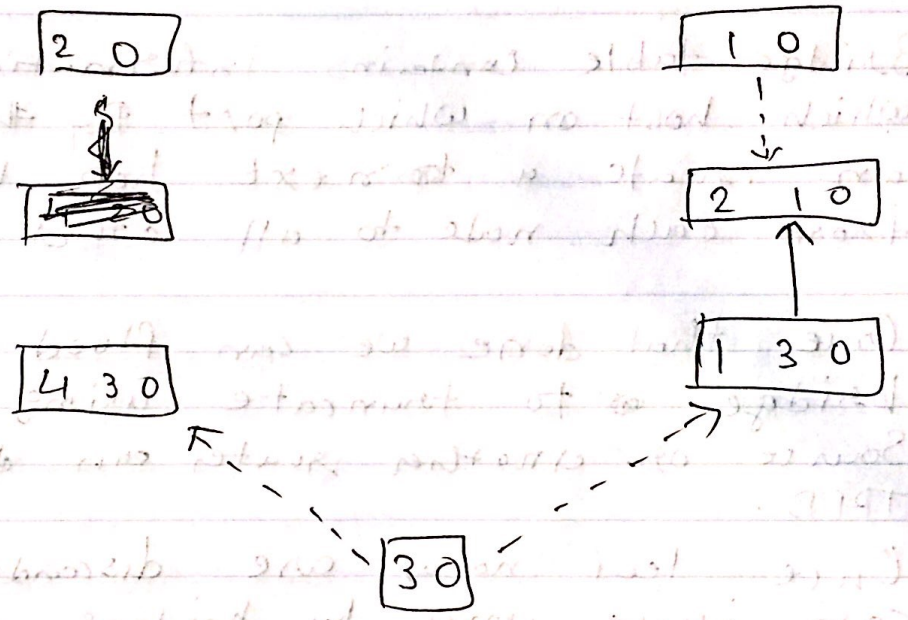


* Here 3 4 0 and 2 1 0 are main reason for cycle.
→ And there are like 6 conflict arcs.

* Now if there are only one conflict

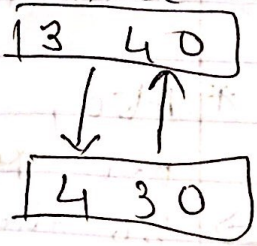
are, we can say that there will be no node which will have one incoming and one outgoing transition arc.

→ Let say we remove 340 and 420



* So here there is only one node, which will have one incoming and one outgoing arc.

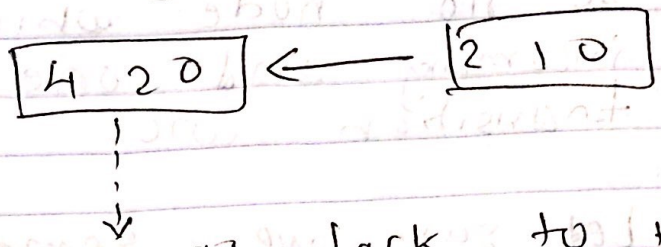
→ But to form cycle, you need atleast 2 such nodes, like



→ So, with only one conflict arck circle is impossible, because any node having outgoing going transmission arc will not have

another incoming transmission arc.

like:



→ So it will never go back to the point origin with only one conflict arc.

Ans:4 Bridge table contains information of which host on which port. By this we can create a next hop vector from each node to all other nodes.

* Once that done we can flood each bridge to truncate using TRPB.

→ Source or another router can do this TRPB.

→ Once leaf nodes are discarded, we can issue MMR by bridges and prune LANS.

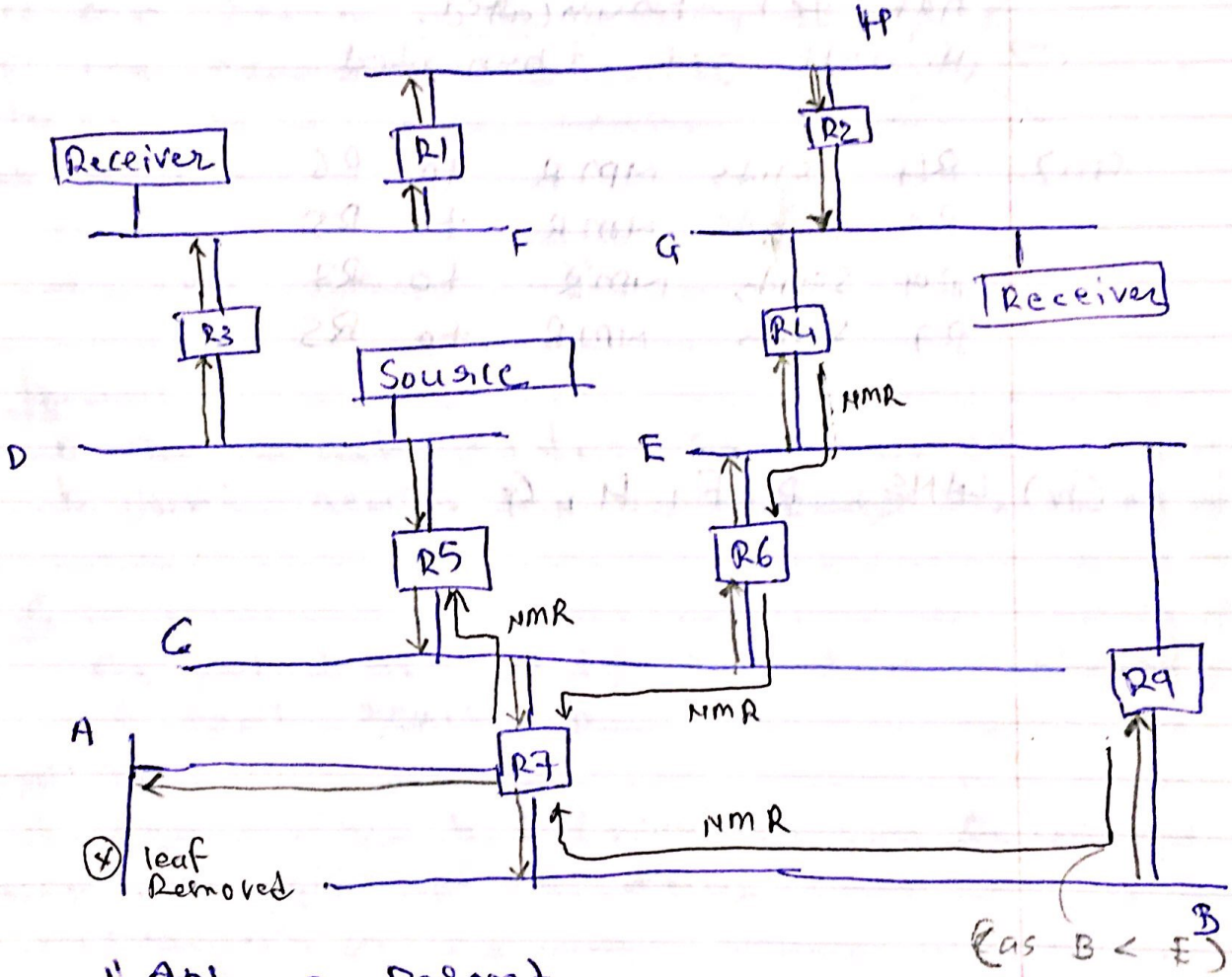
⇒ This works same as routers, it gets next hop vector from each node to all other nodes.

* Also, bridge will have to store entry per MMR request.

⇒ So that is how this can be done.

A-9

(i).



(i) LAN - parent

A	R7
B	R7
C	R5
D	Source itself
E	R6
F	R3
G	R2 (as R2 < R4 alphabatically)
H	R1

(ii) A and G are leaf LANS.

→ As G has receiver on it, it will not get truncated.

→ A will get truncated.

(iii) R4 sends NMR to R6
R6 sends NMR to R5
R4 sends NMR to R7
R7 sends NMR to R5

(iv) LANS: D, F, H, G.