Network Layer

Evripidis Tzamousis

28/04/2017

Address Aggregation

- One of the problems in route finding is the scale of the address space (millions of addresses)
  - Memory Limitation = How to store all
  - Speed limitation = How to find the exit towards one of them in the time scale of a packet interarrival
- Solution : forwarding based on group of addresses (network address) or group of networks.
  - (Going to Heraklion from NY we are not interested in local streets)
- => IP address hierarchy using subnet mask (like a wildcard)

- First : Routing table (Routing Information Base) has ALL topology (routes)
- Routing table created using OSPF , RIP etc which collect/distribute topology info.

- Then: Forwarding table has best routes e.g. using Dijkstra (Has next hop to destination)
- Forwarding: the process of moving packets from input to output based on the forwarding table.

P4

Forwarding Table

Prefix	**Out Interface**
10*	3
1011*	9
011*	8
010110*	5
001*	4
101101*	2
011010*	6
011100*	1
10111*	8
00101*	7

Forwarding Table

RRAS-ROUTER1 - IP Routing Table

Destination	Network mask	Gateway	Interface	Metric	Protocol
10.57.76.0	255.255.255.0	10.57.76.1	Local Area C...	1	Local
10.57.76.1	255.255.255.255	127.0.0.1	Loopback	1	Local
10.255.255.255	255.255.255.255	10.57.76.1	Local Area C...	1	Local
127.0.0.0	255.0.0.0	127.0.0.1	Loopback	1	Local
127.0.0.1	255.255.255.255	127.0.0.1	Loopback	1	Local
192.168.45.0	255.255.255.0	192.168.45.1	Local Area C...	1	Local
192.168.45.1	255.255.255.255	127.0.0.1	Loopback	1	Local
224.0.0.0	224.0.0.0	192.168.45.1	Local Area C...	1	Local
224.0.0.0	224.0.0.0	10.57.76.1	Local Area C...	1	Local
255.255.255.255	255.255.255.255	192.168.45.1	Local Area C...	1	Local
255.255.255.255	255.255.255.255	10.57.76.1	Local Area C...	1	Local

255.255.255.0 = **11111111.11111111.11111111**.00000000

netmask

10.57.76.0 AND 255.255.255.0 = 10.57.76.*

netmask

- the netmask (eg /24) is not part of the ip header, and is not transmitted on the wire. no-one knows the netmask of a device on the network, except the device itself.

11010110.01100001.11111111.00000011 / 25
Ποιές διευθύνσεις αντιπροσωπεύει :

11010110.01100001.11111111.00000011

11010110.01100001.11111111.0*

(Αρχίζουμε απο την δοθείσα διεύθυνση (32 bit) και πάνω χωρίς να πειράζουμε τα bit που είναι στο netmask)

P15

- In Problem P10 you are asked to provide a forwarding table (using longest prefix matching). Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

11100000 00*

11100000 01000000*

11100000 *

11100001 0*

Μετατρέπουμε δυαδικούς σε δεκαδικούς και μετρούμε πόσα ψηφία πριν το αστεράκι

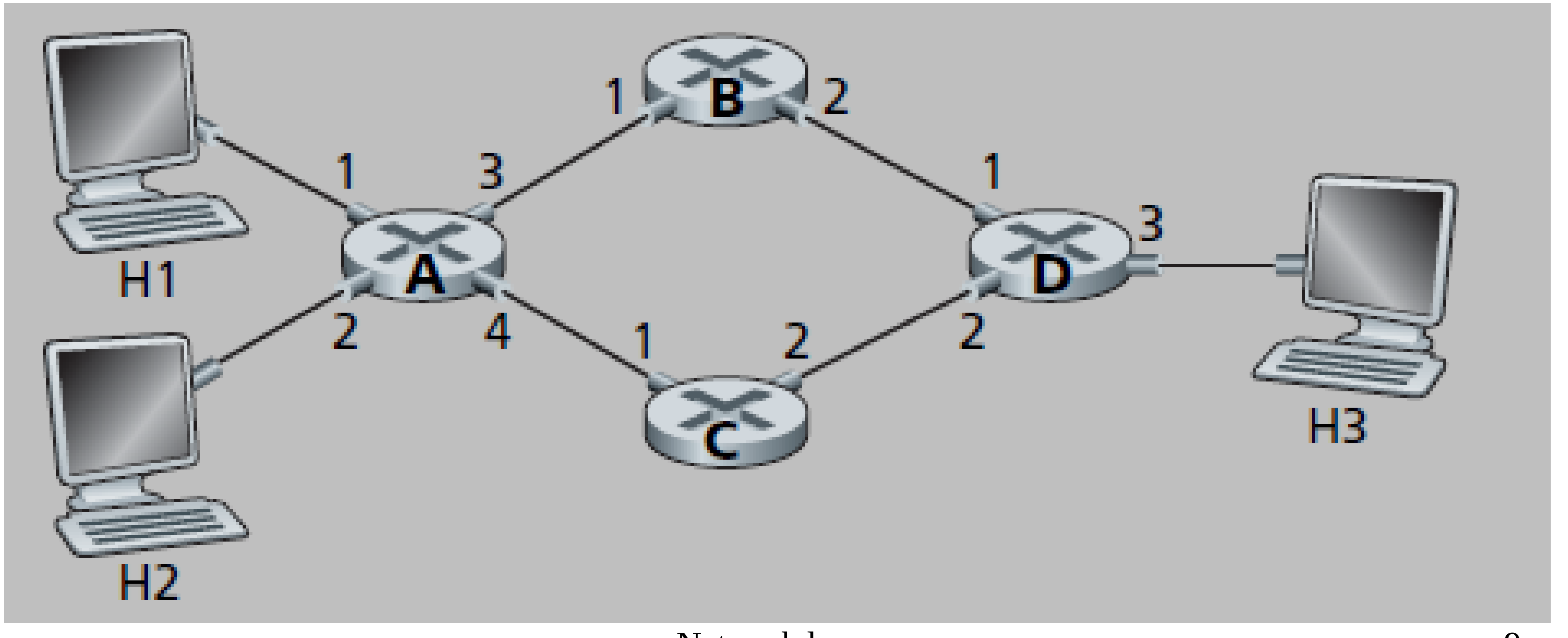
P15

- In Problem P10 you are asked to provide a forwarding table (using longest prefix matching). Rewrite this forwarding table using the a.b.c.d/x notation instead of the binary string notation.

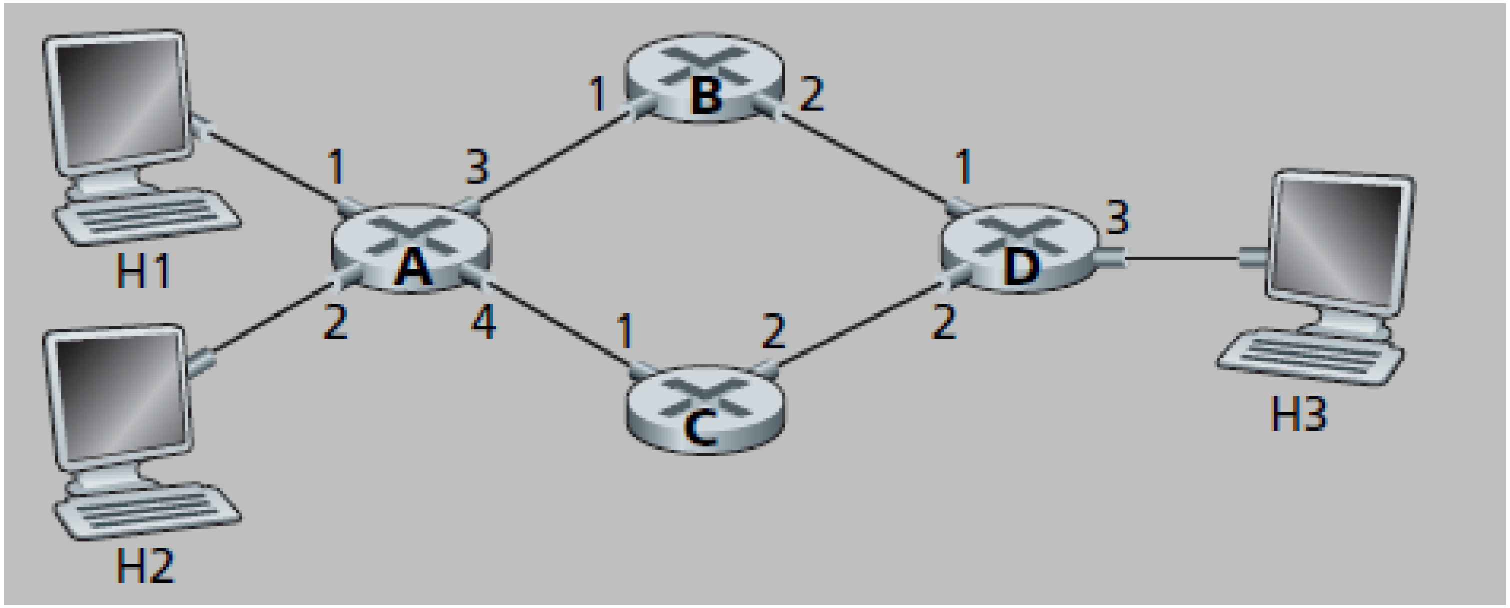
- **Destination Address**
- 11100000 00 = (224.0.0.0/10)
- 11100000 01000000 = (224.64.0.0/16)
- 11100000 = (224.0.0.0/8)
- 11100001 0 = (225.0.0.0/9)

P4

- Suppose that this network is a datagram network. Show the forwarding table in router A, such that all traffic destined to host H3 is forwarded through interface 3



P4 a. : Forwarding Table

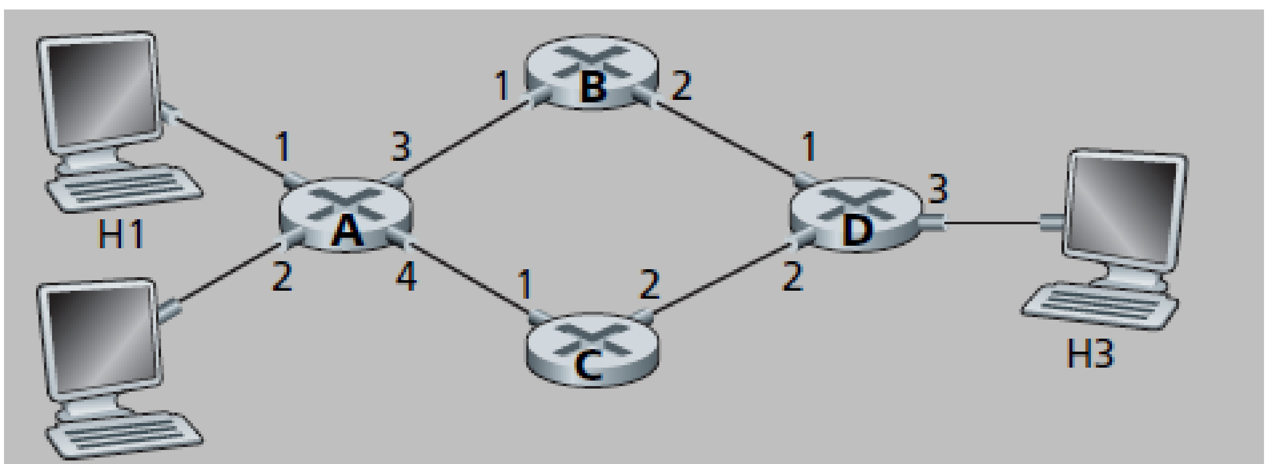


forwarding table in router A

Prefix	mask	out
IP address of H3 -	Subnet Mask –	Out interface

P4 b.

Suppose that this network is a datagram network. Can you write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4? (Hint: this is a trick question.)

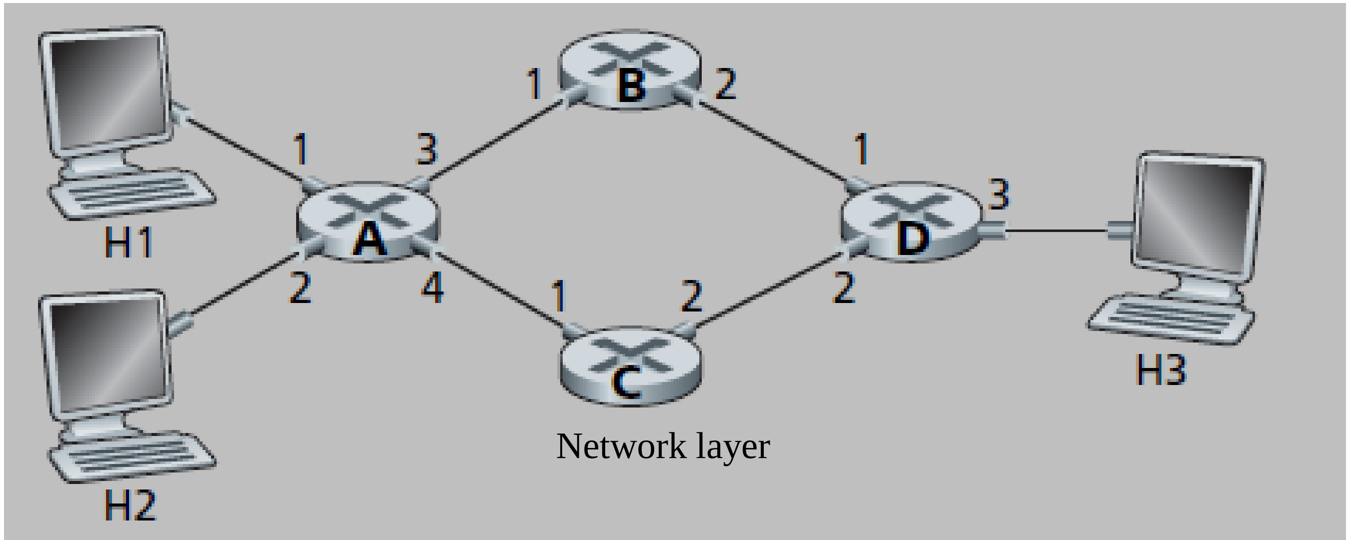


P4 b.

- No, because forwarding rule is only based on destination address.

P4 c.

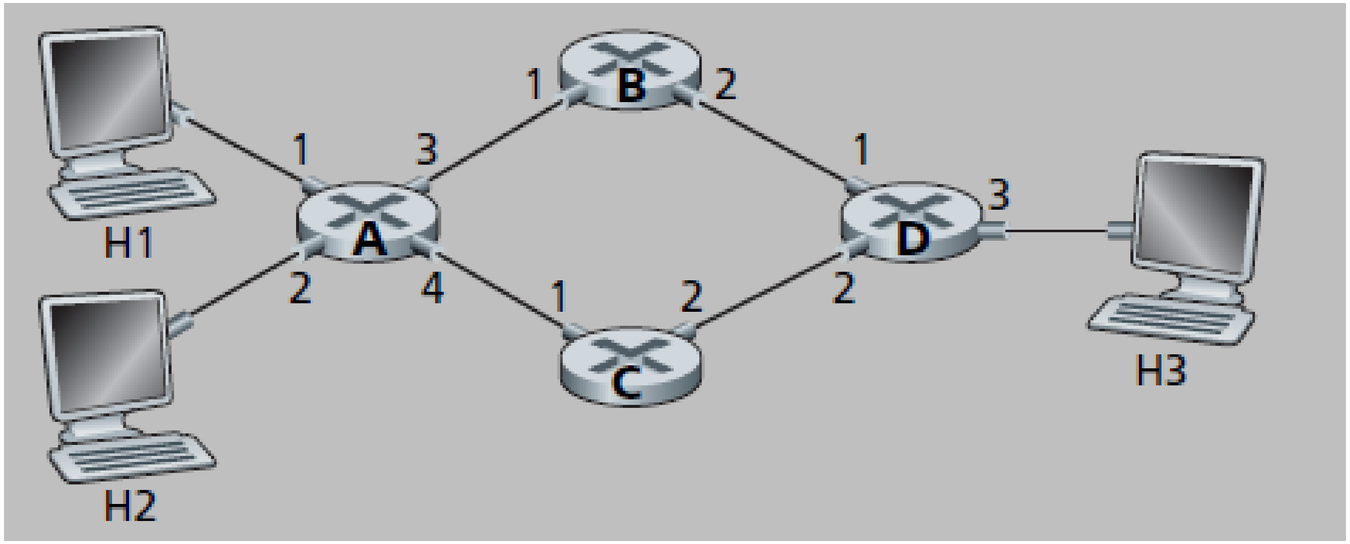
- Now suppose that this network is a virtual circuit network and that there is one ongoing call between H1 and H3, and another ongoing call between H2 and H3. Write down a forwarding table in router A, such that all traffic from H1 destined to host H3 is forwarded through interface 3, while all traffic from H2 destined to host H3 is forwarded through interface 4.



P4 c.

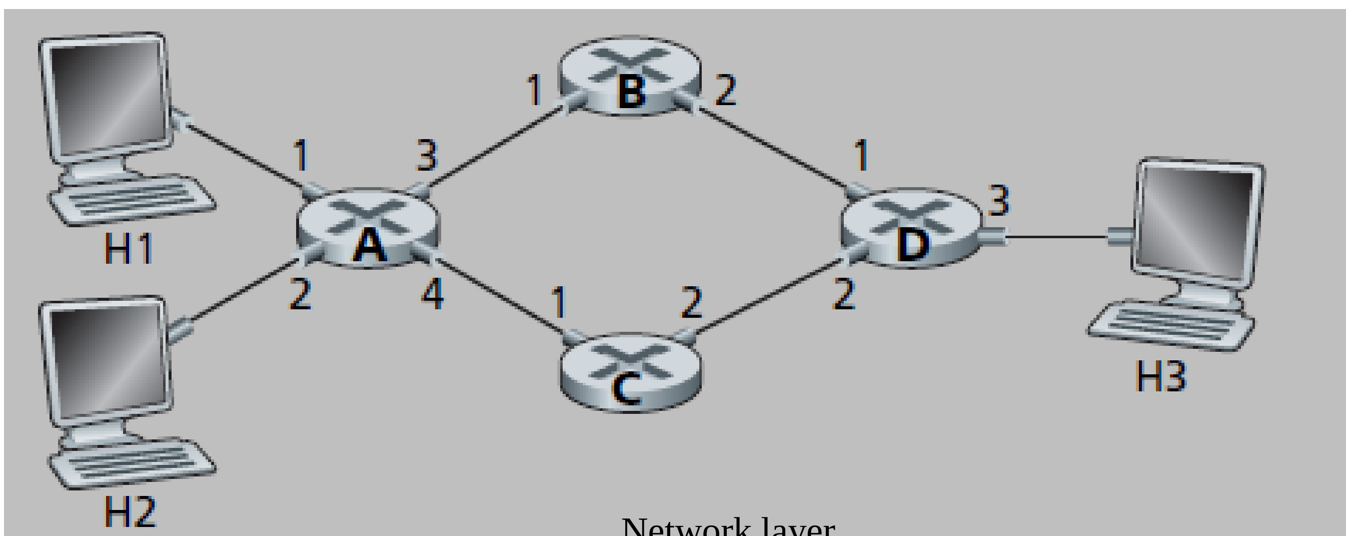
forwarding table (VC Table) in router A

Incoming interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	12	3	22
2	63	4	18



P4 d.

- Assuming the same scenario as (c), write down the forwarding tables in nodes B, C, and D.



d).

Router B.

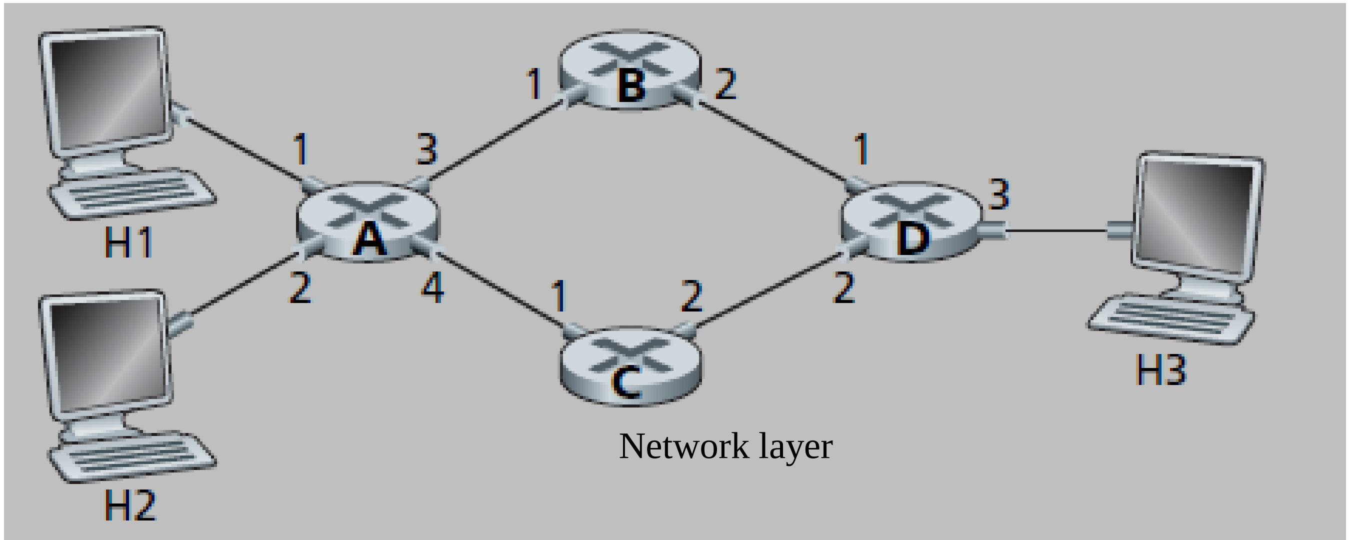
Incoming interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	22	2	24

Router C.

Incoming interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	18	2	50

Router D.

Incoming interface	Incoming VC#	Outgoing Interface	Outgoing VC#
1	24	3	70
2	50	3	76



P16

- Consider a subnet with prefix 128.119.40.128/26. Give an example of one IP address (of form xxx.xxx.xxx.xxx) that can be assigned to this network.

128.119.40.128/26

10000000.01110111.00101000.10000000 / 26

to

10000000.01110111.00101000.10111111 / 26

P16

- Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create **four subnets** from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

128.119.40.64/26

10000000.01110111.00101000.01000000 / 26

P16

- Suppose an ISP owns the block of addresses of the form 128.119.40.64/26. Suppose it wants to create **four subnets** from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

10000000.01110111.00101000.01000000 / 26

10000000.01110111.00101000.01000000 / 26 128.119.40.64/28

10000000.01110111.00101000.01010000 / 26 128.119.40.80/28

10000000.01110111.00101000.01100000 / 26 128.119.40.96/28

10000000.01110111.00101000.01110000 / 26 128.119.40.112/28

P17

- Consider the topology shown in Figure 4.17. Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

- Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support **250 interfaces**; Subnet B should have enough addresses to support **120 interfaces**; and Subnet C should have enough addresses to support **120 interfaces**. Of course, subnets D, E and F should each be able to support **two interfaces**.

P17

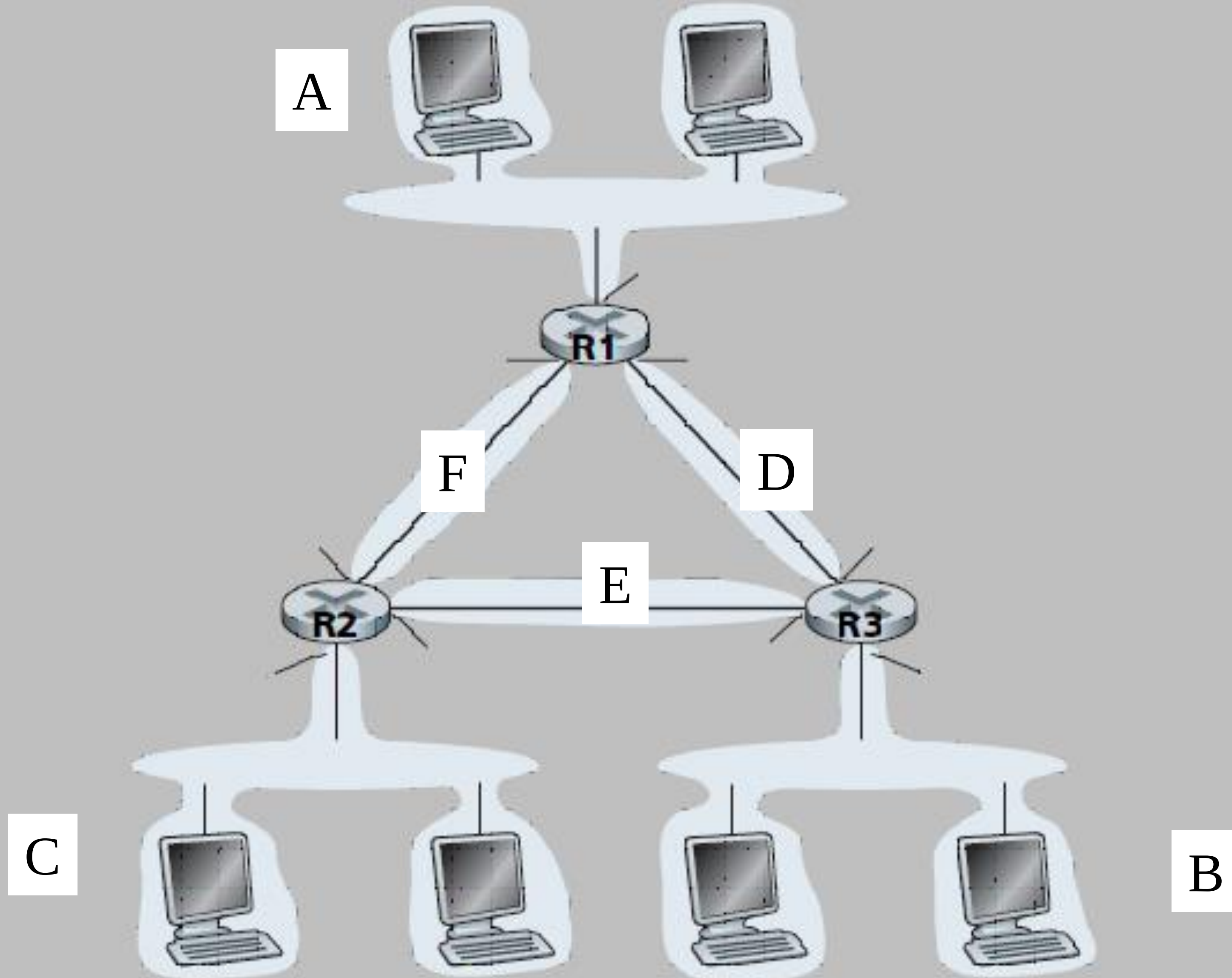


Figure 4.17 ♦ Three routers interconnecting six subnets

Allocating IPs to subnets - Steps

- Sort address groups in descending order
- Assign IP numbers to the largest, then go to smaller
- Try to give as continuous as possible IP addresses to subnets which are on same router (so as to be able to do address aggregation)

P17

given 214.97.254/23 =

11010110.01100001.11111110.00000000 /23

For 250 hosts we need 8 bits

11010110.01100001.11111110.00000011 /24 -

11010110.01100001.11111110.11111111 /24

A (250)

= 214.97.254.3 /24

One entry in routing table for all 250 addresses

The link between routers is a subnet with two addresses

11010110.01100001.11111110.00000000

11010110.01100001.11111110.00000001

D

= 214.97.254.0 /31

One entry in routing table for the link between routers e.g as an exit point

P17

214.97.254/23

11010110.01100001.11111110.00000000 /23

11010110.01100001.11111111.00000011 / 25 -

B 120

11010110.01100001.11111111.01111111 / 25

214.97.255.3 /25

11010110.01100001.11111111. 00000000

E

11010110.01100001.11111111. 00000001

214.97.255.0 /31

P17

214.97.254/23

11010110.01100001.11111110.00000000 /23

11010110.01100001.11111111.10000011 / 25 -

11010110.01100001.11111111.11111111 / 25

C 120

214.97.255.131 /25

11010110.01100001.11111111.10000001

11010110.01100001.11111111.10000000

F

214.97.255.129 /31

(P17 b)

- **Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.**

P17

net	Router 1	Outgoing Interface
A	**11010110.01100001.11111110**.00000000 ⇨	A
B	**11010110.01100001.11111111.0**0000000 ⇨	D
C	**11010110.01100001.11111111.**10000000 ⇨	F

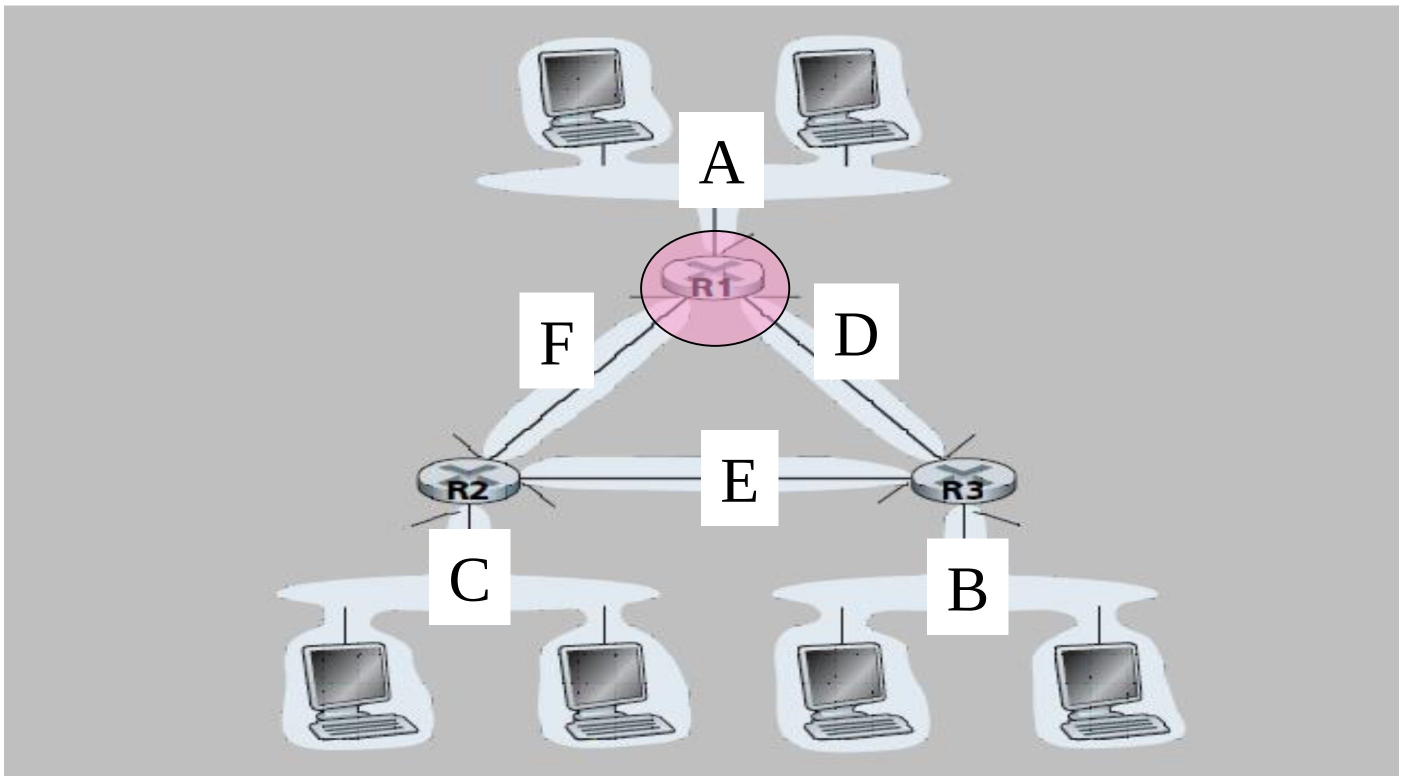


Figure 4.17 ◆ Three routers interconnecting six subnets

P10

- 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:

P10

Destination Address Range	Out Link Interface
11100000 00000000 00000000 00000000 - **11100000 00**111111 11111111 11111111 =4194304 routes	**0**
11100000 01000000 00000000 00000000 - **11100000 01000000** 11111111 11111111 = 65536 routes	**1**
11100000 01000001 00000000 00000000 - 11100001 01111111 11111111 11111111	**2**
otherwise	**3**

P10 a

- Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.
- (generally could have more than five entries but the less entries the more efficient forwarding is)

Forwarding table aggregation methodology

P10

- If addresses in decimal form, convert to binary
- Group addresses by interface / next hop
- Find identical/common bits in grouped addresses => find subnet mask
- Aggregated address is based on common bits. (best if aggregated address contain exactly the addresses we want to represent – no more no less)
- Crucial that there is no overlap between our aggregated addresses found! => routing errors
- No overlap =
  - Either same prefix but different bits (naturally no overlap)
  - Or different prefix and same bits (differentiate by choosing longest prefix)

P10

Destination Address Range	Out Link Interface	
11100000 00000000 00000000 00000000 - **11100000 00**111111 11111111 11111111 =4194304 routes	⇨ **0**	√
11100000 01000000 00000000 00000000 - **11100000 01000000** 11111111 11111111 = 65536 routes	⇨ **1**	√
11100000 01000001 00000000 00000000 - 11100001 01111111 11111111 11111111 ?	⇨ **2**	
otherwise	⇨ **3**	

11100000 ⇨ 2 ?

More specific (longer) entry wins!

Out Link Interface

11100000 00000000 00000000 00000000 -
11100000 00111111 11111111 11111111 ⇨ **0**

11100000 01000000 00000000 00000000 -
11100000 01000000 11111111 11111111 ⇨ **1**

11100000 01000001 00000000 00000000 -
11100000 11111111 11111111 11111111
+
11100001 00000000 00000000 00000000
11100001 01111111 11111111 11111111 ⇨ **2**

otherwise ⇨ **3**

P10

11100000 00000000 00000000 00000000 -
11100000 00111111 11111111 11111111

11100000 01000000 00000000 00000000 -
11100000 01000000 11111111 11111111

11100000 01000001 00000000 00000000 -
11100000 11111111 11111111 11111111

+

11100001 00000000 00000000 00000000
11100001 01111111 11111111 11111111

Aggregated routes

11100000 00* ⇨ 0

11100000 01000000* ⇨ 1

11100000 * ⇨ 2

11100001 0* ⇨ 2

otherwise ⇨ 3

P10

Another Solution

11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

P 10 b.

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

11001000 10010001 01010001 01010101
11100001 01000000 11000011 00111100
11100001 10000000 00010001 01110111

Που ταιριάζουν περισσότερο?

11100000 00* ⇨ 0

11100000 01000000* ⇨ 1

11100000 * ⇨ 2

11100001 0* ⇨ 2

otherwise to interface 3

P 10 b.

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

11001000 10010001 01010001 01010101 ⇨ 3
11100001 01000000 11000011 00111100 ⇨ 2
11100001 10000000 00010001 01110111 ⇨ 3

11100000 00* ⇨ 0

11100000 01000000* ⇨ 1

11100000 * ⇨ 2

11100001 0* ⇨ 2

otherwise to interface 3

P21

- Consider the network setup in Figure 4.22 Suppose that the ISP assigns the router the address 24.34.112.235 and that the network address of the home network is 192.168.1/24.

- a. Assign addresses to all interfaces in the home network.

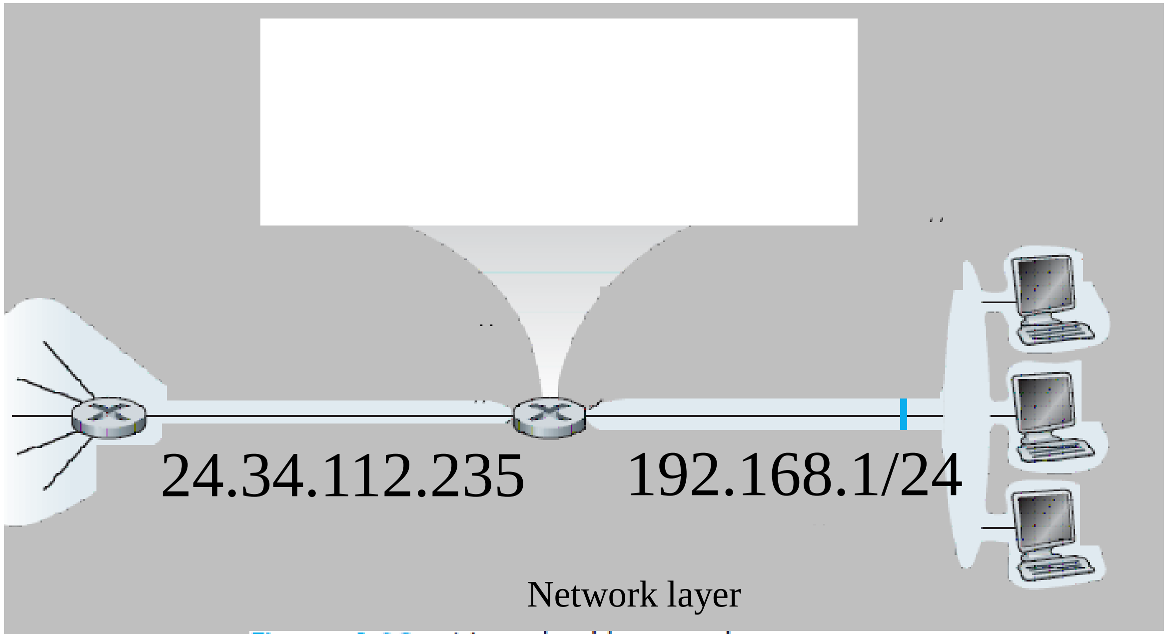


Figure 4.22 ♦ Network address translation

P21

NAT translation table	
WAN side	LAN side
138.76.29.7, 5001	10.0.0.1, 3345
. . .	. . .

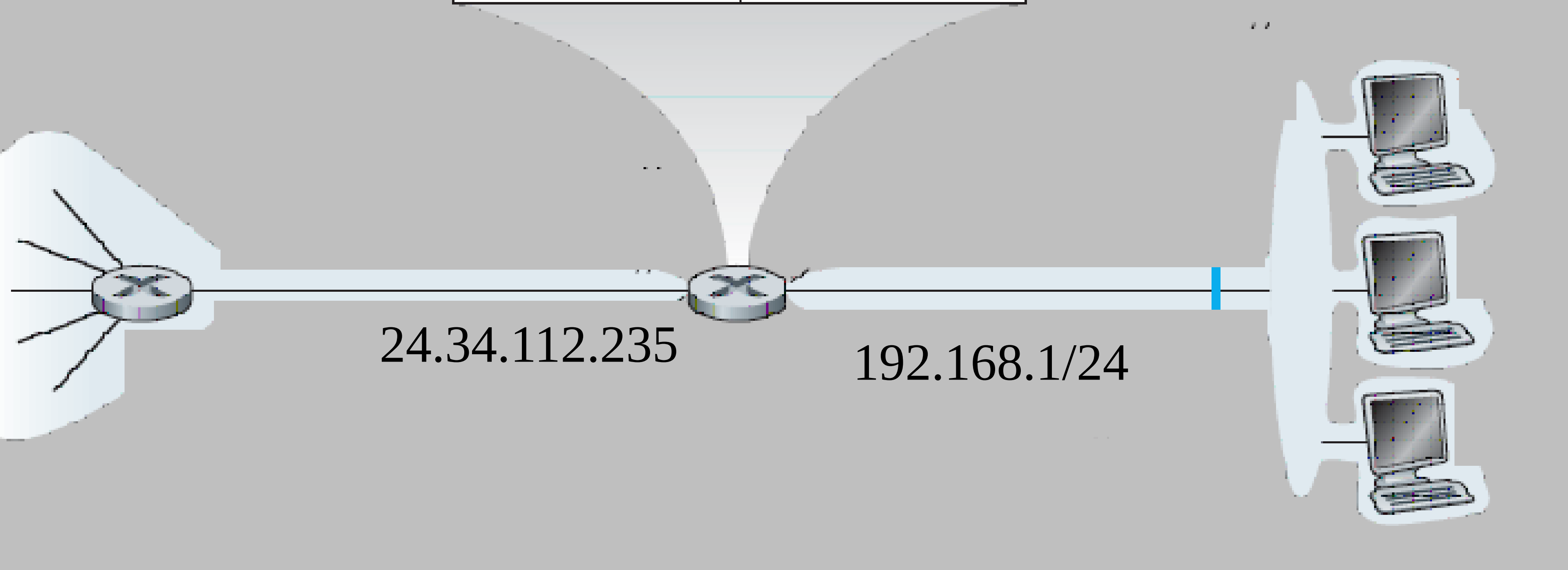


Figure 4.22 ♦ Network address translation

a) Home addresses: 192.168.1.1, 192.168.1.2, 192.168.1.3 with the router interface being 192.168.1.4

P.21

b) Suppose each host has two ongoing TCP connections, all to port 80 at host 128.119.40.86. Provide the six corresponding entries in the NAT translation table.

NAT Translation Table

WAN Side	LAN Side
24.34.112.235, 4000	192.168.1.1, 3345
24.34.112.235, 4001	192.168.1.1, 3346
24.34.112.235, 4002	192.168.1.2, 3445
24.34.112.235, 4003	192.168.1.2, 3446
24.34.112.235, 4004	192.168.1.3, 3545
24.34.112.235, 4005	192.168.1.3, 3546

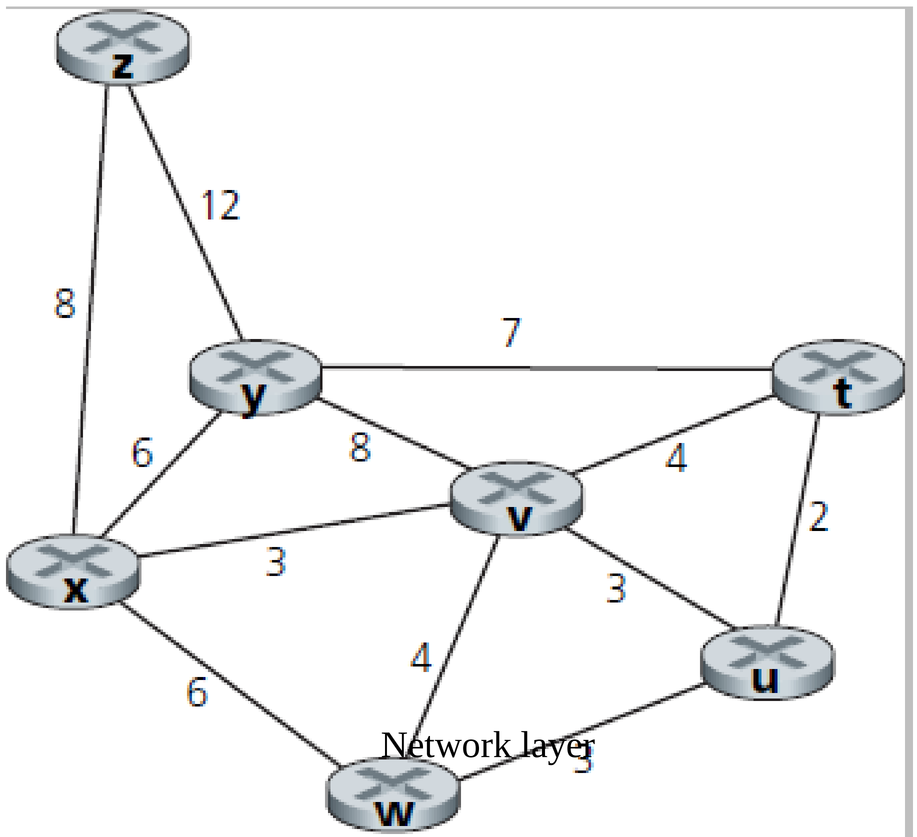
P21

What is NAT ?

Different LAN addresses are mapped to one WAN address but on different ports.

P26

- Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 4.3.

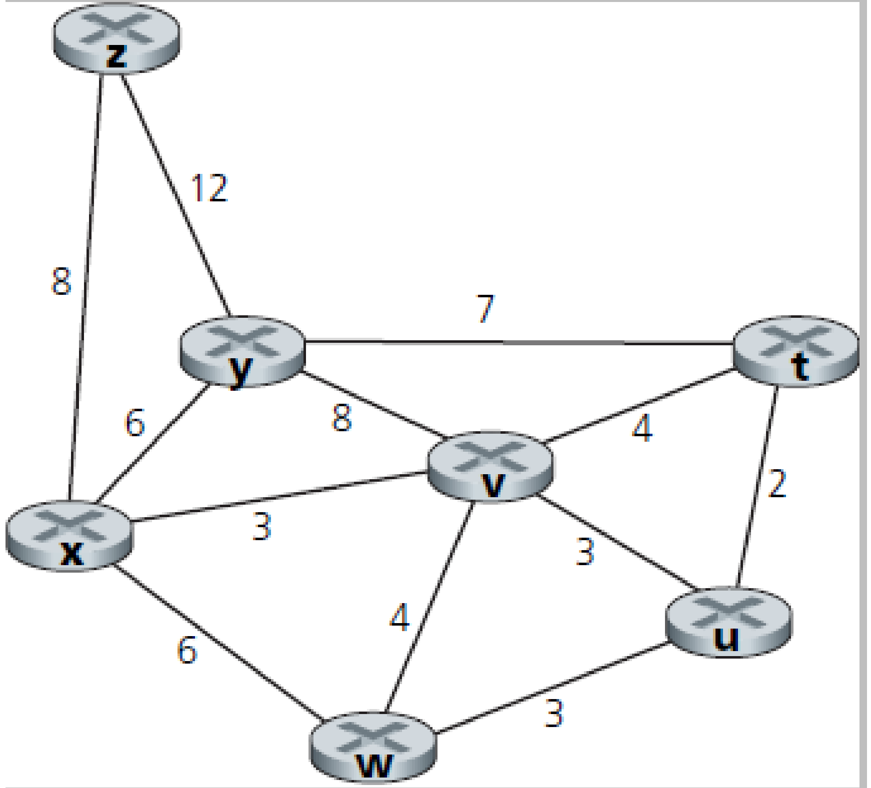


P26

DIJKSTRA

- Group N has chosen nodes-it gets filled up at each step. We work on un-chosen.
- We are at source. N has source only. (non direct neighbors have cost ∞) Add in N the node X (not in N) with the least cost path to source. Then calculate new costs to other nodes (neighbors of X) not in N if going though X. (if smaller cost than previously, then set node X as previous node to them)
- REPEAT : add node with less distance to source –as previously set, until N contain all nodes

P26



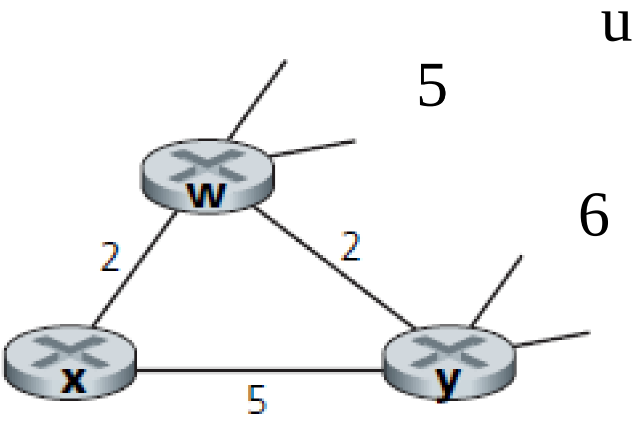
	y	v	w	u	t	z
x	6x	3x	6x	∞	∞	8x
v	6x	**3x**	6x	6v	7v	8x
y	**6x**	**3x**	6x	6v	7v	8x
w	**6x**	**3x**	**6x**	6v	7v	8x
u	**6x**	**3x**	**6x**	**6v**	7v	8x
t	**6x**	**3x**	**6x**	**6v**	**7v**	8x
z	**6x**	**3x**	**6x**	**6v**	**7v**	**8x**

P27

- Consider the network shown in Problem P26. Using Dijkstra's algorithm, and showing your work using a table similar to Table 4.3, do the following:
- a. Compute the shortest path from t to all network nodes.
- b. Compute the shortest path from u to all network nodes.
- c. Compute the shortest path from v to all network nodes.
- d. Compute the shortest path from w to all network nodes.
- e. Compute the shortest path from y to all network nodes.
- f. Compute the shortest path from z to all network nodes.

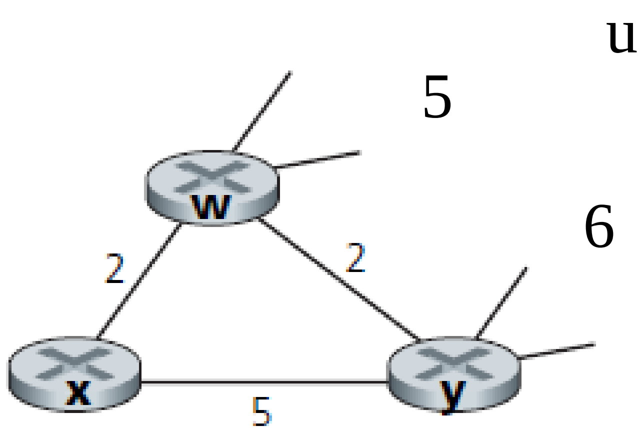
P30

- Consider the network fragment shown below. X has only two attached neighbors, w and y. w has a minimum-cost path to destination u (not shown) of 5, and y has a minimum-cost path to u of 6. The complete paths from w and y to u (and between w and y) are not shown. All link costs in the network have strictly positive integer values.



P30 a

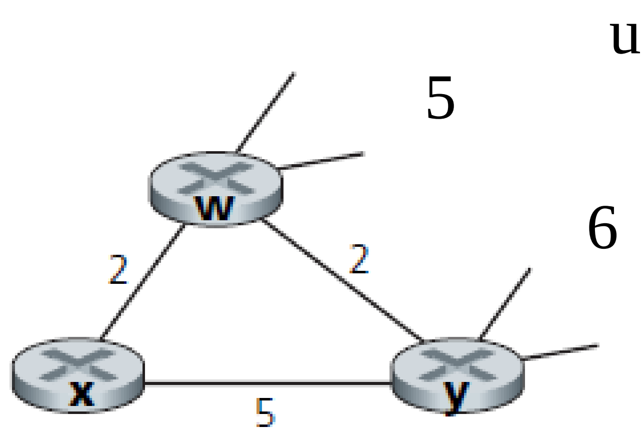
Give *x*'s distance vector for destinations *w*, *y*, and *u*.



P30 a

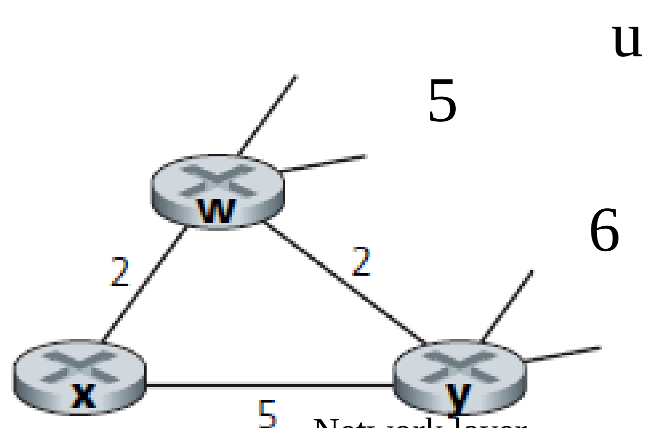
Give *x*’s distance vector for destinations *w*, *y*, and *u*.

Dx(w) = 2, Dx(y) = 4, Dx(u) = 7



P30 b

- Give a link-cost change for either c(x,w) or c(x,y) such that **x will inform** its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.



P30 b

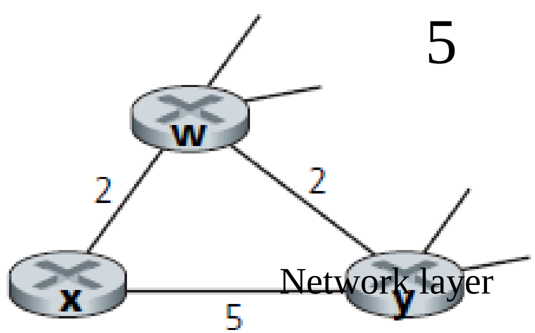
- First consider what happens if c(x,y) changes. If c(x,y) becomes larger or smaller (as long as c(x,y) >=1) , the least cost path from x to u will still have cost at least 7. Thus a change in c(x,y) (if c(x,y)>=1) will not cause x to inform its neighbors of any changes.
- If **c(x,y)= δ<1**, then the least cost path now passes through y and has cost δ+6. x will inform its neighbors of this new cost.
- Now consider if c(x,w) changes. If **c(x,w) < 2,** then the least-cost path to u continues to pass through w and its cost changes to 5 + δ;

  x will inform its neighbors of this new cost.
- If **c(x,w) > 6,** then the least cost path now passes through y and has cost 11; again x will inform its neighbors of this new cost.

u

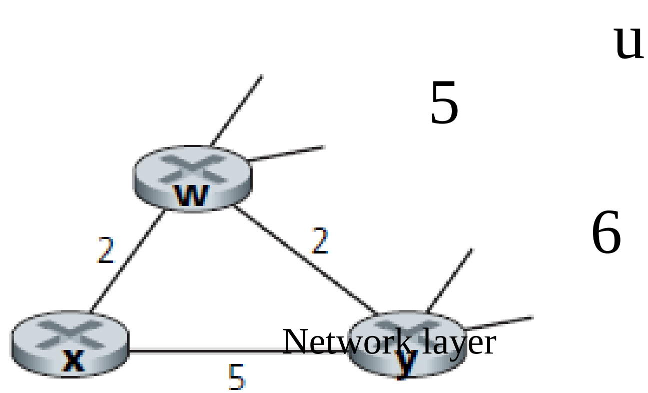
5

6



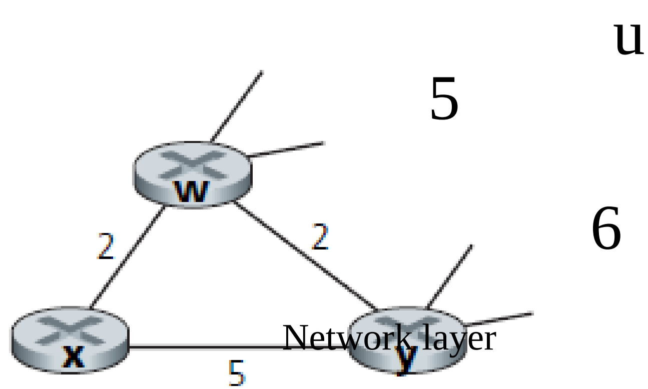
P30 c

- Give a link-cost change for either c(x,w) or c(x,y) such that **x will not inform** its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.



P30 c

- Any change in link cost c(x,y) (and as long as c(x,y) >=1) will not cause x to inform its neighbors of a new minimum-cost path to u



P36

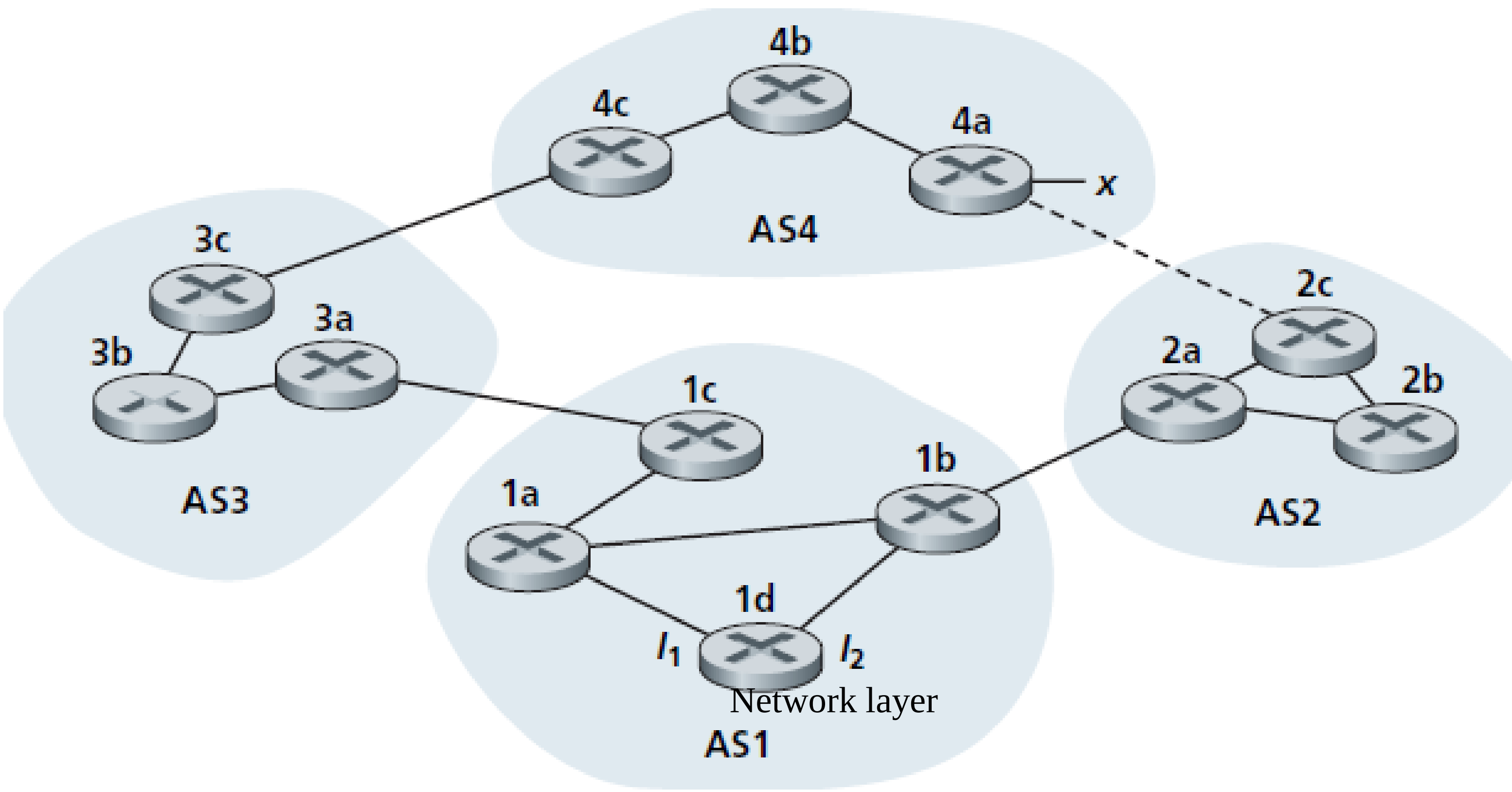
- Will a BGP router always choose the loop-free route with the shortest AS path length? Justify your answer.

P36

- The chosen path is not necessarily the shortest AS-path. Recall that there are many issues to be considered in the route selection process. It is very likely that a longer loop-free path is preferred over a shorter loop-free path due to economic reason. For example, an AS might prefer to send traffic to one neighbor instead of another neighbor with shorter AS distance.

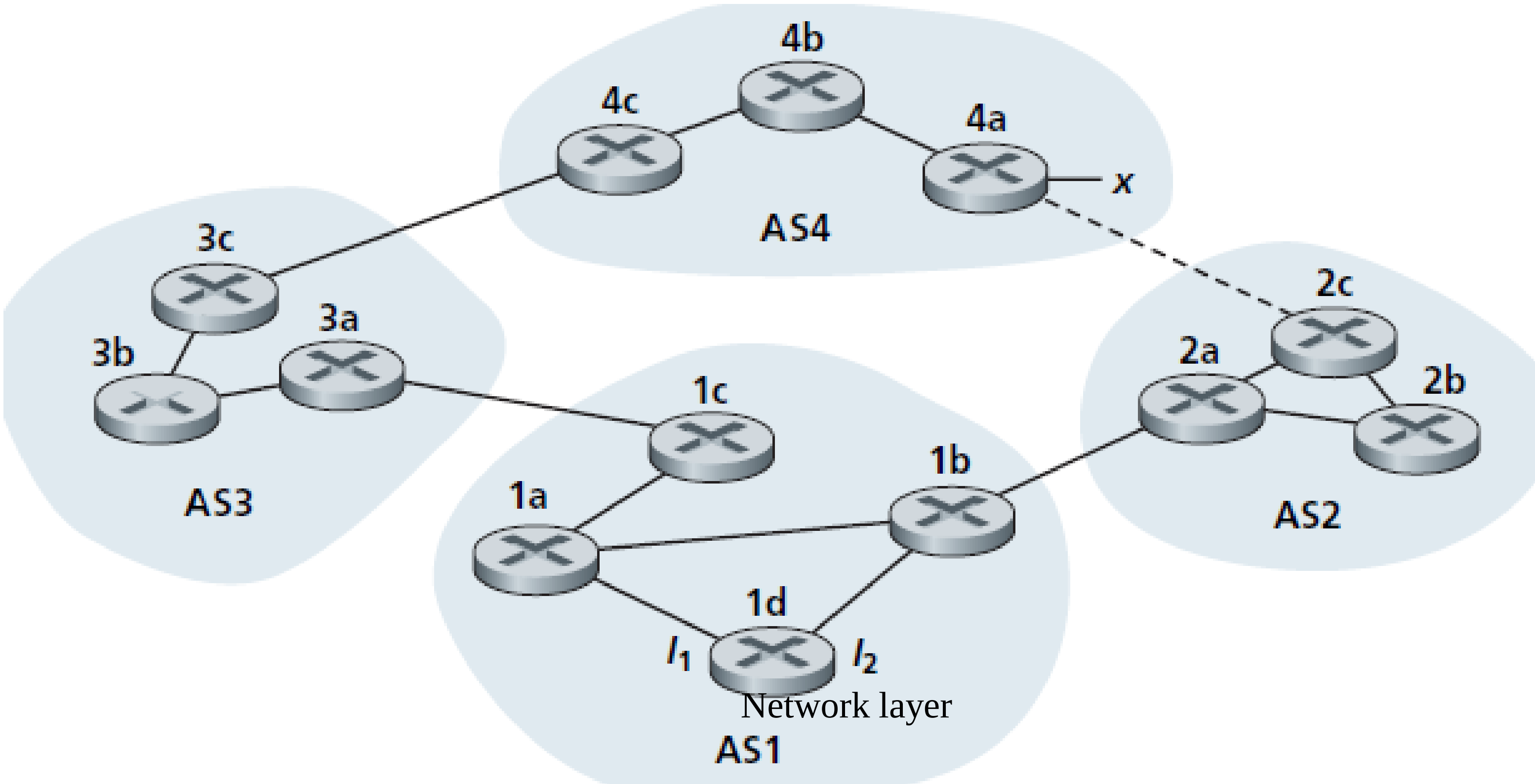
P37

- Consider the network shown below. Suppose **AS3 and AS2** are running **OSPF** for their intra-AS routing protocol. Suppose **AS1 and AS4 are running RIP** for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4.



P37

- a. Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?
- b. Router 3a learns about x from which routing protocol?
- c. Router 1c learns about x from which routing protocol?
- d. Router 1d learns about x from which routing protocol?



- a. Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?
- eBGP

- b. Router 3a learns about x from which routing protocol?
- iBGP

- c. Router 1c learns about x from which routing protocol?
- eBGP

- d. Router 1d learns about x from which routing protocol?
- iBGP