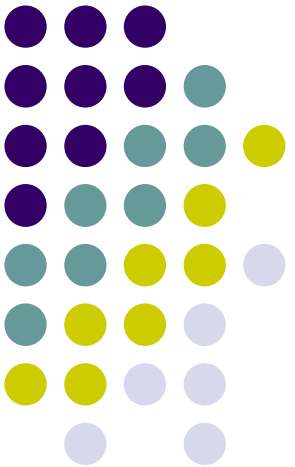
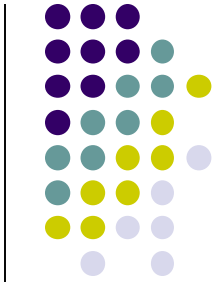
Lecture 7: Routing

Reading 5.2

Computer Networks, Tanenbaum



Contents



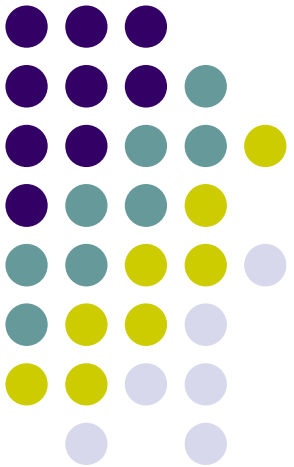
- What is routing?
- Static routing and dynamic routing
- Routing algorithms and protocols

What is routing?

Routing principals

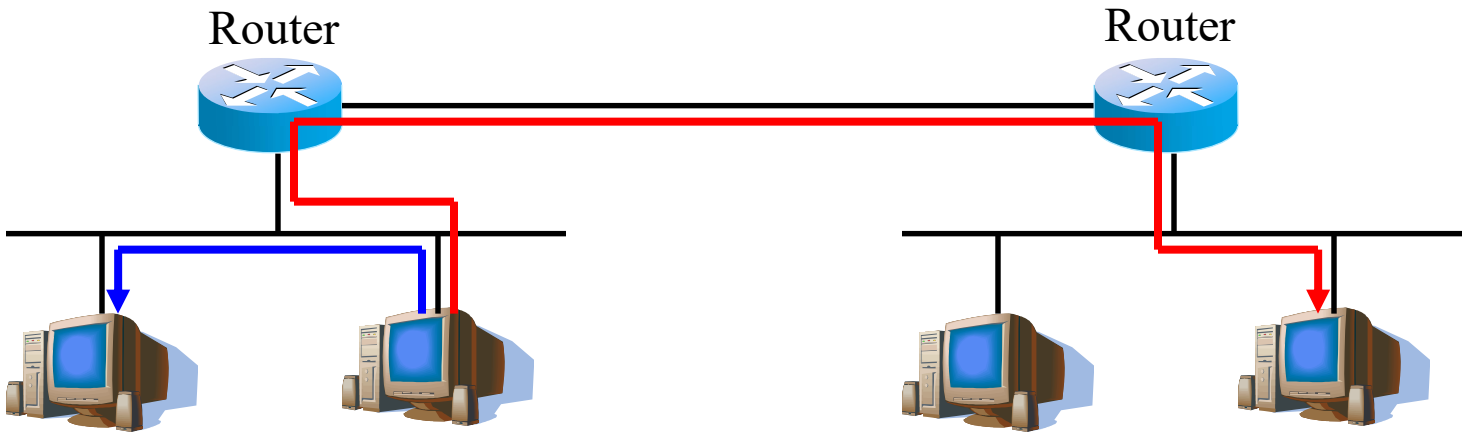
Forwarding mechanism

“Longest matching” rule

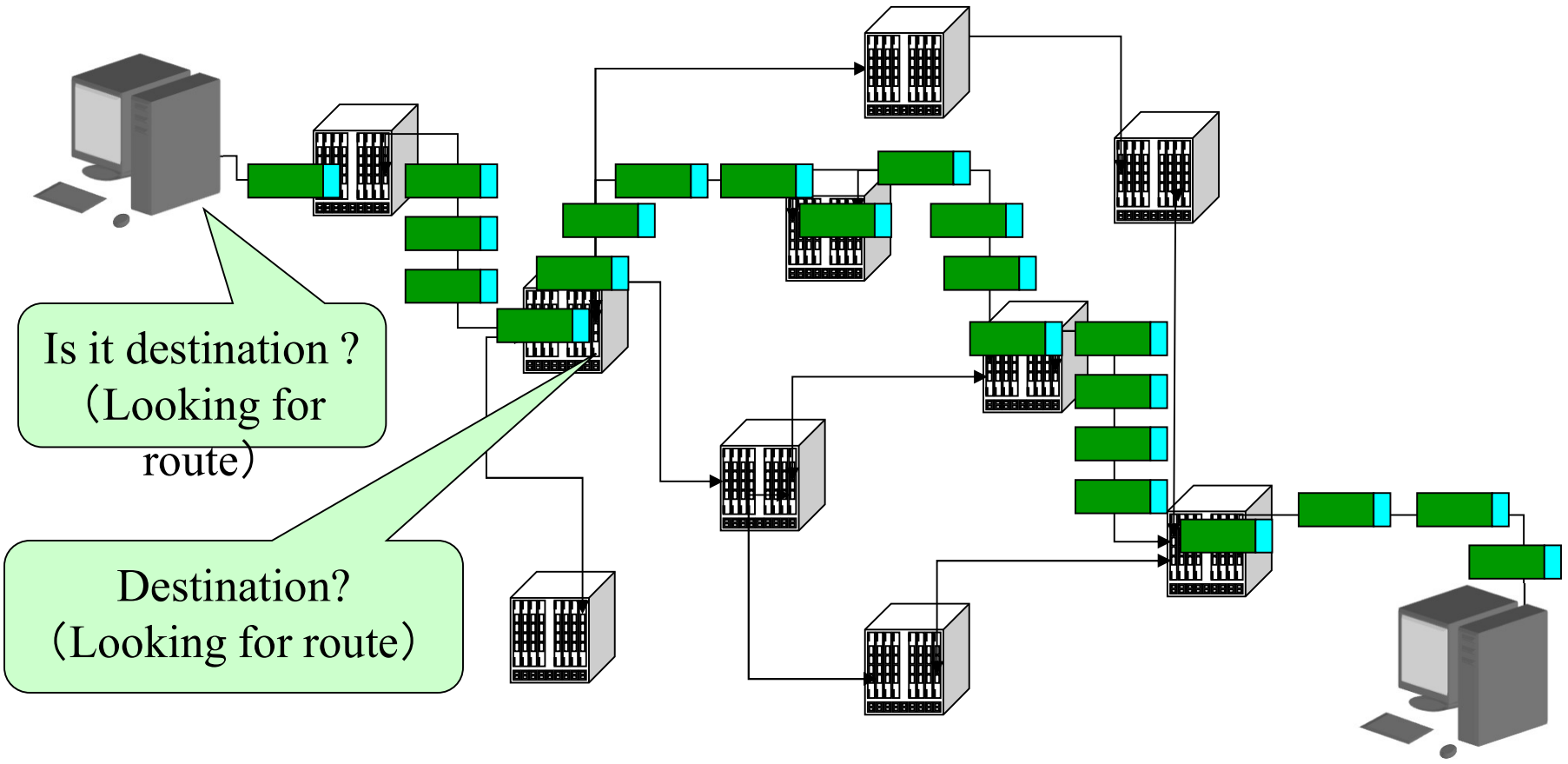


Routing principles (1)

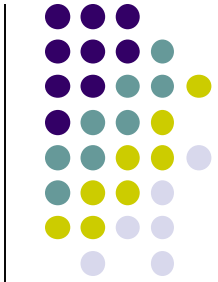
- When a host send an IP packet to another host
  - If the destination and the source are in the same physical medium: Transfer directly
  - If the destination is in a different network with the source: Send through some other routers (need to choose route)



Routing principles (2)

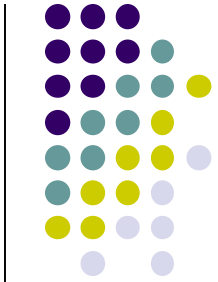


What is routing?



- A mechanism so that a host or a router decides how to forward a packet from source to destination.
- Result of the routing is a routing table
- What to consider in routing
  - Building routing table
  - Information need to calculating route
  - Routing algorithm and protocol.

What is a router?



- Router is the device that forwards data between networks
  - Is a computer with particular hardware
  - Connects multiple networks together, has multiple network interfaces
  - Forward packets according to routing table

Some examples of routers…

BUFFALO
BHR-4RV

PLANEX
GW-AP54SAG

YAMAHA
RTX-1500

Cisco 2600

Router ngoại vi

Cisco CRS-1

Router mạng trục

Hitachi
GR2000-1B

Juniper M10

Foundry Networks
NetIron 800

Cisco 3700

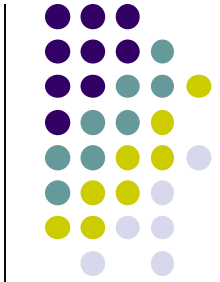
Router cỡ trung

http://www.cisco.com.vn

http://www.juniper.net/

http://www.buffalotech.com

Routing table

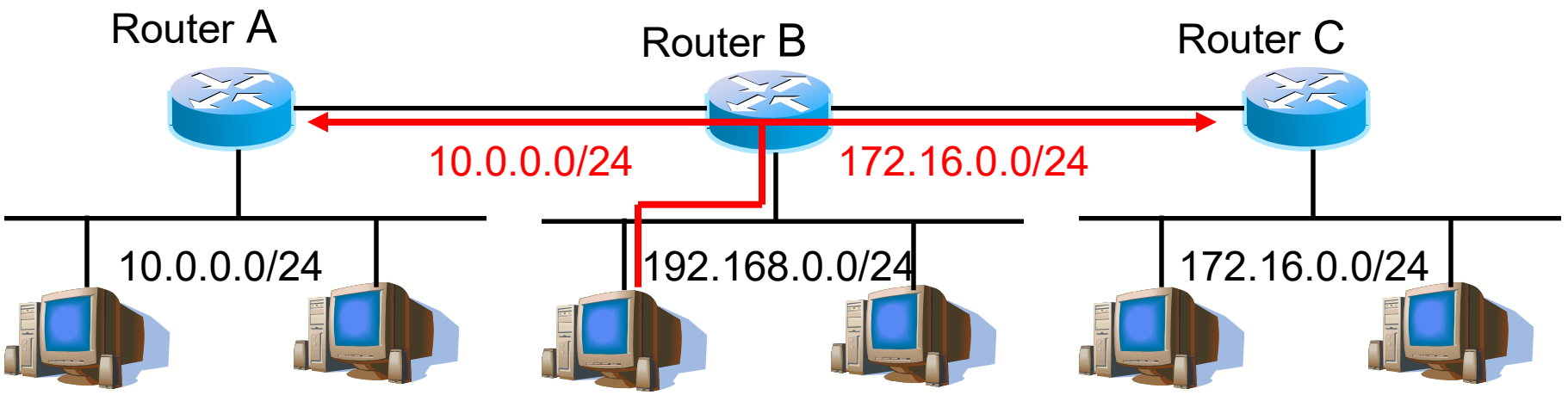


- Lists of possible routes, saved in the memory of router
- Main components of routing table
  - Destination network address/network mask
  - Next router

```
#show ip route
Prefix                Next Hop
203.238.37.0/24 via 203.178.136.14
203.238.37.96/27 via 203.178.136.26
203.238.37.128/27 via 203.178.136.26
203.170.97.0/24 via 203.178.136.14
192.68.132.0/24 via 203.178.136.29
203.254.52.0/24 via 203.178.136.14
202.171.96.0/24 via 203.178.136.14
```

Routing table and forwarding mechanism (1)

Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C



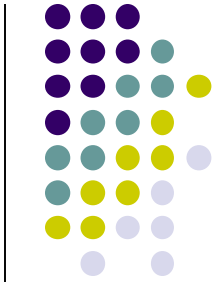
Rule: ***No routes, no reachability*****!**

“Longest matching” rule (1)

- Assume that there are more than one entry matching with a destination network in routing table.
- Destination address: 11.1.2.5
- What should be chosen as the next hop?

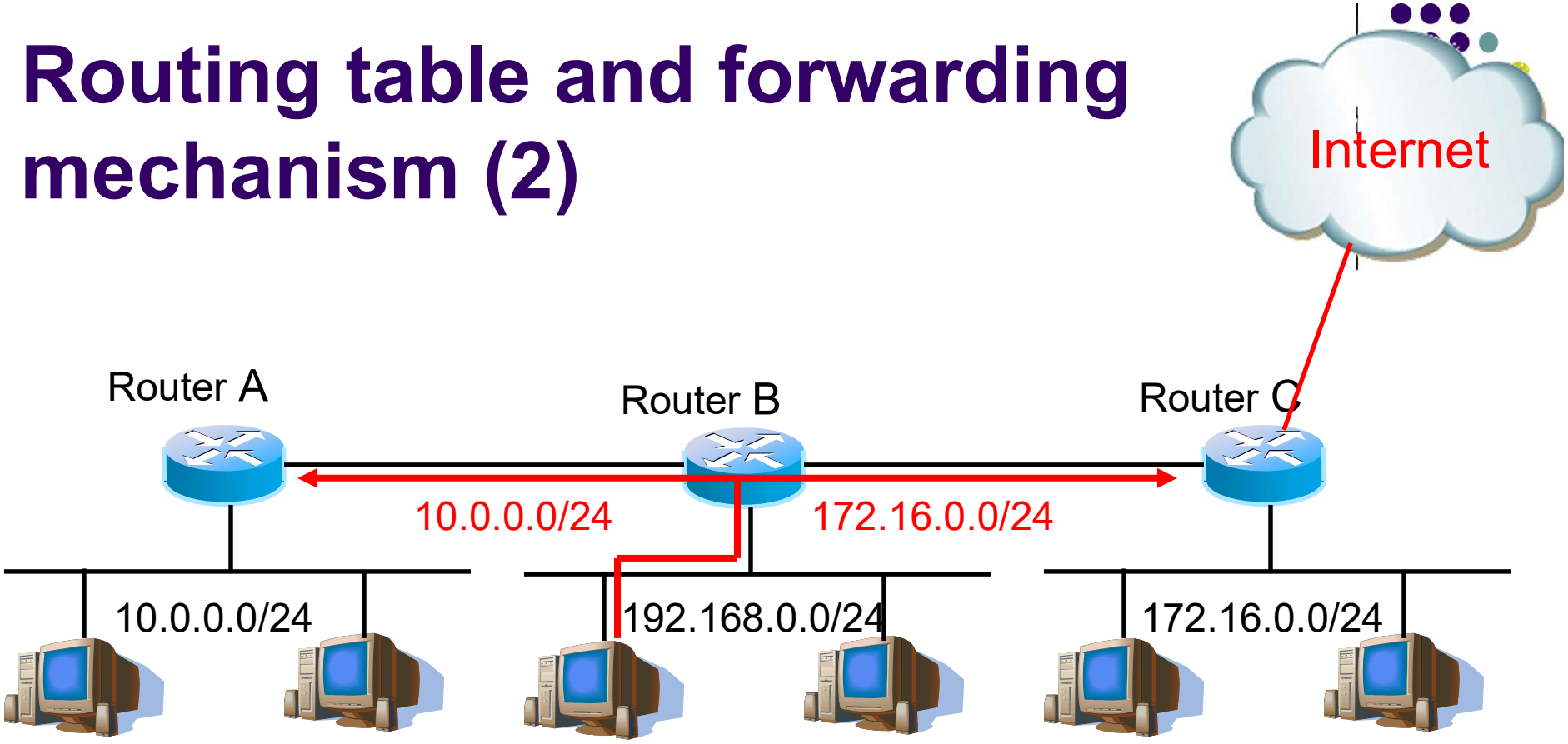
Network	Next hop
11.0.0.0/8	A
11.1.0.0/16	B
11.1.2.0/24	C

“Longest matching” rule (2)



Destination address:
11.1.2.5 = 00001011.00000001.00000010.00000101
Route 1:
11.1.2.0/24 = 00001011.00000001.00000010.00000000
Route 2:
11.1.0.0/16 = 00001011.00000001.00000000.00000000
Route 3:
11.0.0.0/8 = 00001011.00000000.00000000.00000000

Routing table and forwarding mechanism (2)



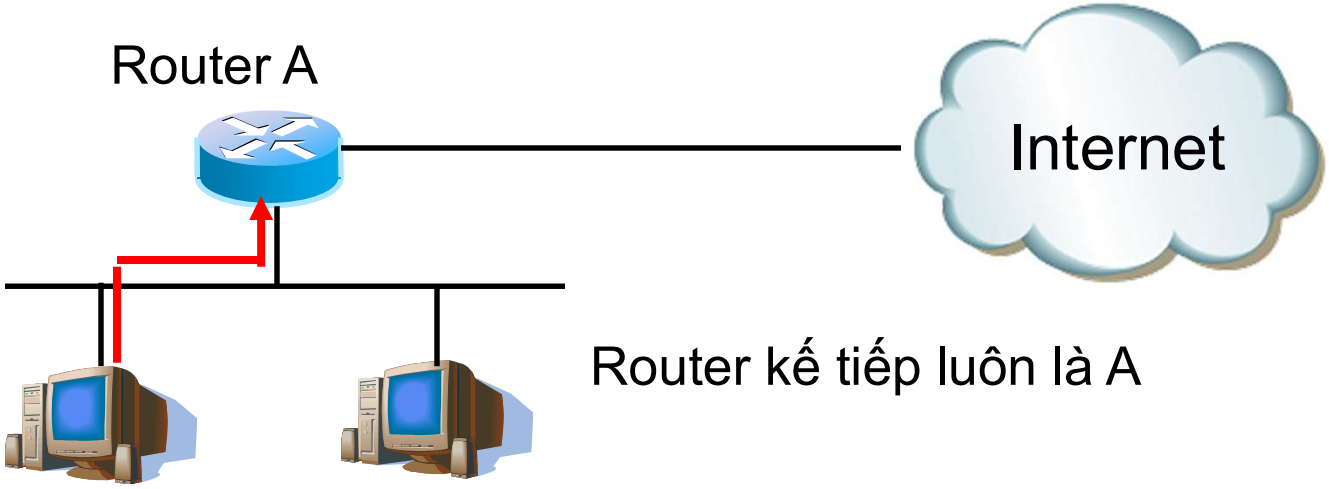
Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C
192.168.0.0/24 0.0.0.0/0	Direct C

Q. What is the routing table in C?

Q: What if C is connected to the Internet?

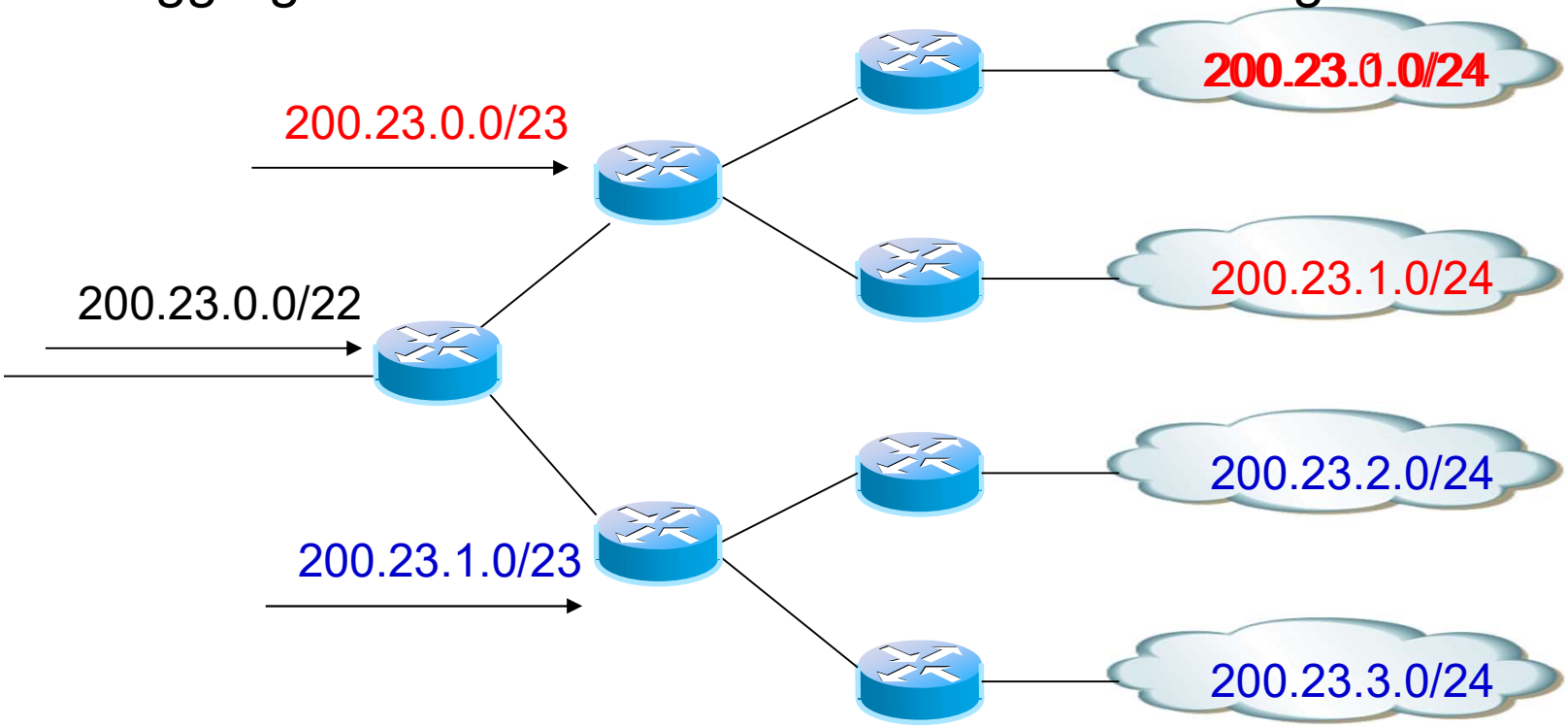
Default route

- If router does not find a route to a destination in its routing table, default route is necessary
  - Default route is defined for all destination networks that are not figured in the routing table.
- 0.0.0.0/0
  - Is a special notation for all destination networks

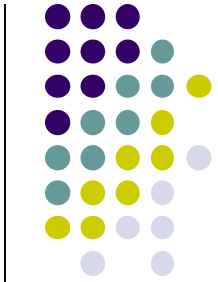


Route aggregation

- How many networks in the Internet?
- There will be a lot of entries in the routing table?
- The entries to sub-networks of the same “big” network can be aggregated inorder to reduce the size of routing table.

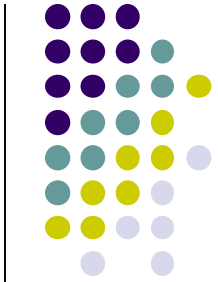


Route aggregation (2)



- Example of Viettel network
  - Viettel own a big IP address space
    - 203.113.128.0-203.113.191.255
  - For connecting to a subnet (client) of Viettel, routing table needs only to have a route to Viettel network.
- Default route is a type of route aggregation
  - 0.0.0.0/0

Exercises



- A router has the following (CIDR) entries in its routing table:

Address/mask Next hop

135.46.56.0/22 Interface 0 0011 1000

135.46.60.0/22 Interface 1 0011 1100

192.53.40.0/23 Router 1 0010 1000

default Router 2

- For each of the following IP addresses, what does the router do if a packet with that address arrives?

(a) 135.46.63.10 0011 1111

(b) 135.46.57.14

(c) 135.46.52.2

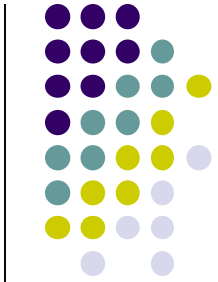
(d) 192.53.40.7

(e) 192.53.56.7 0011 1000

Solution:

Apply longest matching rule.

Solution



Apply longest matching rule.

(students should explain why by matching binary form of the addresses)

(a) 135.46.63.10 → Interface 1

(b) 135.46.57.14 → Interface 0

(c) 135.46.52.2 → Router 2 (default route)

(d) 192.53.40.7 → Router 1

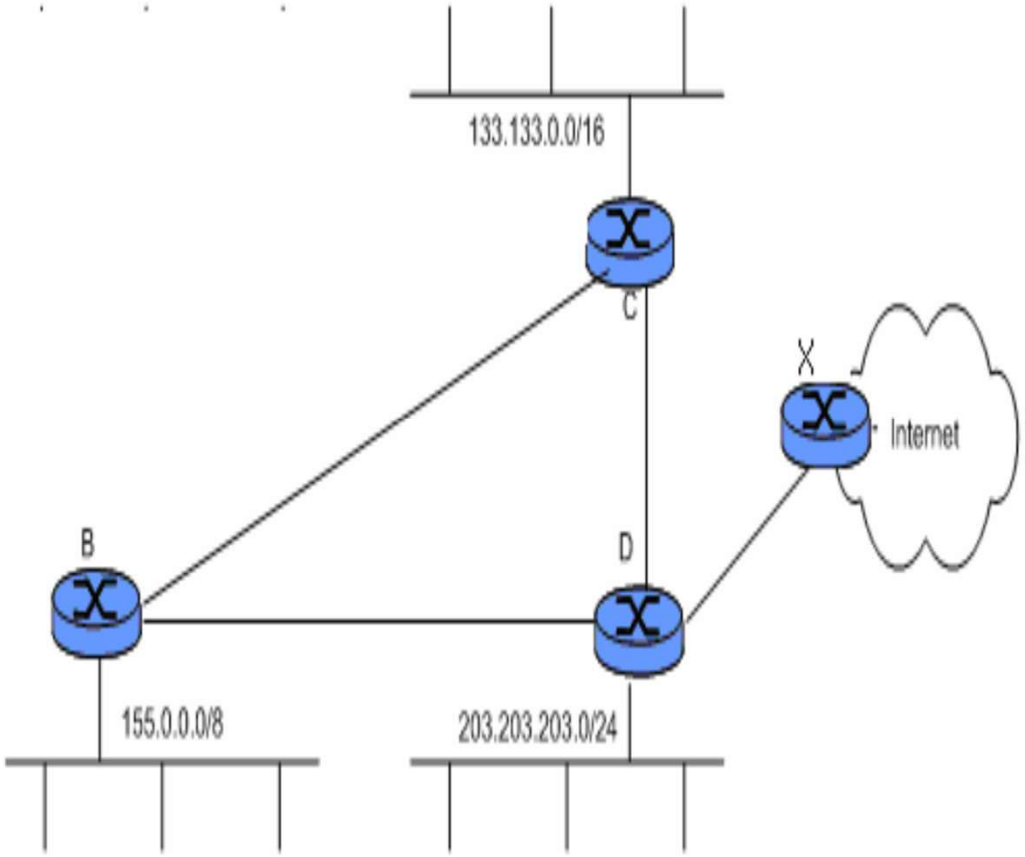
(e) 192.53.56.7 → Router 2 (default route)

40 = 0010 1000

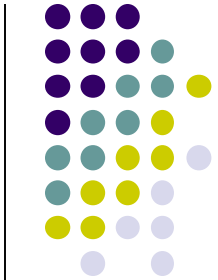
56 = 0011 1000

Exercise

- Assume that we have a network with following topology. What should be routing table of routers B, C, D in order to assure that all hosts can send data to each other and to the Internet.



Solution



- Routing table on B

Network	Next hop
133.133.0.0/16	C
155.0.0.0/8	Direct
203.203.203.0/24	D
0.0.0.0/0	D

- Routing table on C

Network	Next hop
133.133.0.0/16	Direct
155.0.0.0/8	B
203.203.203.0/24	D
0.0.0.0/0	D

- Routing table on D

Network	Next hop
133.133.0.0/16	C
155.0.0.0/8	B
203.203.203.0/24	Direct
0.0.0.0/0	X

Example of routing table on a host

C:\Documents and Settings\hongson>netstat -rn
Route Table
==========
Interface List
0x1 ..... ……MS TCP Loopback interface
0x2 ...08 00 1f b2 a1 a3 ..... Realtek RTL8139 Family PCI Fast Ethernet NIC -
==========

Active Routes:

Network	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.34	20
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.1.0	255.255.255.0	192.168.1.34	192.168.1.34	20
192.168.1.34	255.255.255.255	127.0.0.1	127.0.0.1	20
192.168.1.255	255.255.255.255	192.168.1.34	192.168.1.34	20
224.0.0.0	240.0.0.0	192.168.1.34	192.168.1.34	20
255.255.255.255	255.255.255.255	192.168.1.34	192.168.1.34	1

Default Gateway: 192.168.1.1
==========

Example of routing table in a Router

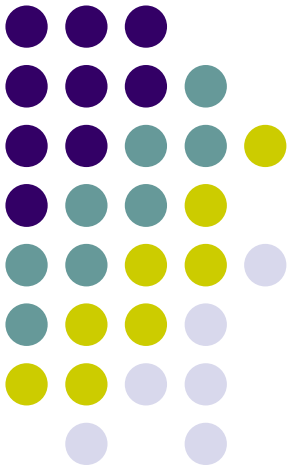
```
#show ip route
Prefix                    Next Hop
203.238.37.0/24 via 203.178.136.14
203.238.37.96/27 via 203.178.136.26
203.238.37.128/27 via 203.178.136.26
203.170.97.0/24 via 203.178.136.14
192.68.132.0/24 via 203.178.136.29
203.254.52.0/24 via 203.178.136.14
202.171.96.0/24 via 203.178.136.14
```

Static and dynamic routing

Static routing

Dynamic routing

Advantage – Weakness



Updating routing table

- Network structure may change
  - New network is added.
  - Router failure due to power corruption …
- It's necessary to update routing table
  - of all nodes (theory)
  - in pratice: some nodes

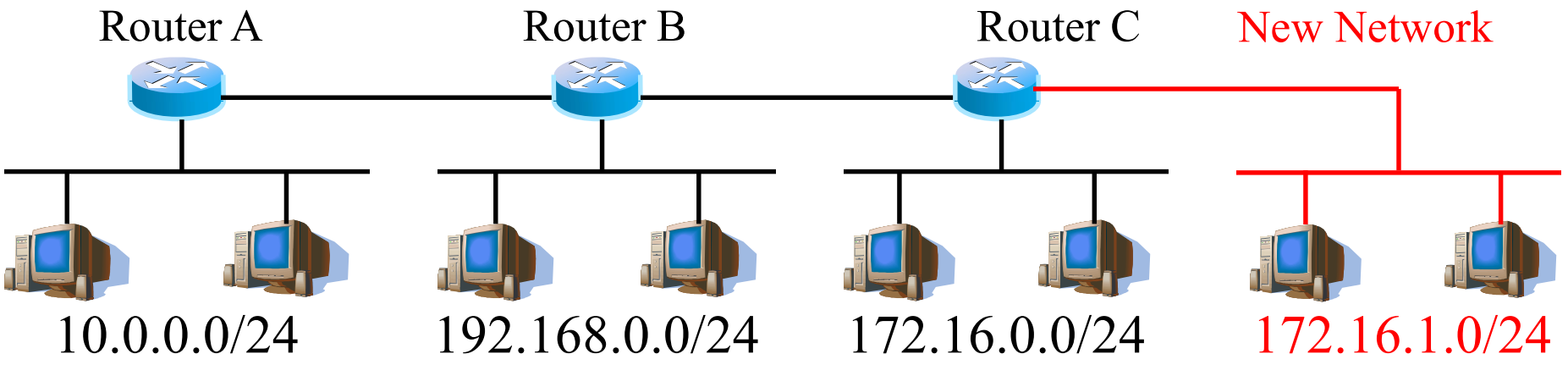
Network	Next-hop
192.168.0.0/24	B
172.16.0.0/24	B

172.16.1.0/24 B

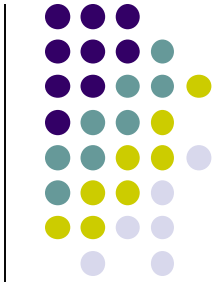
Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C

172.16.1.0/24 C

Network	Next-hop
10.0.0.0/24	B
192.168.0.0/24	B



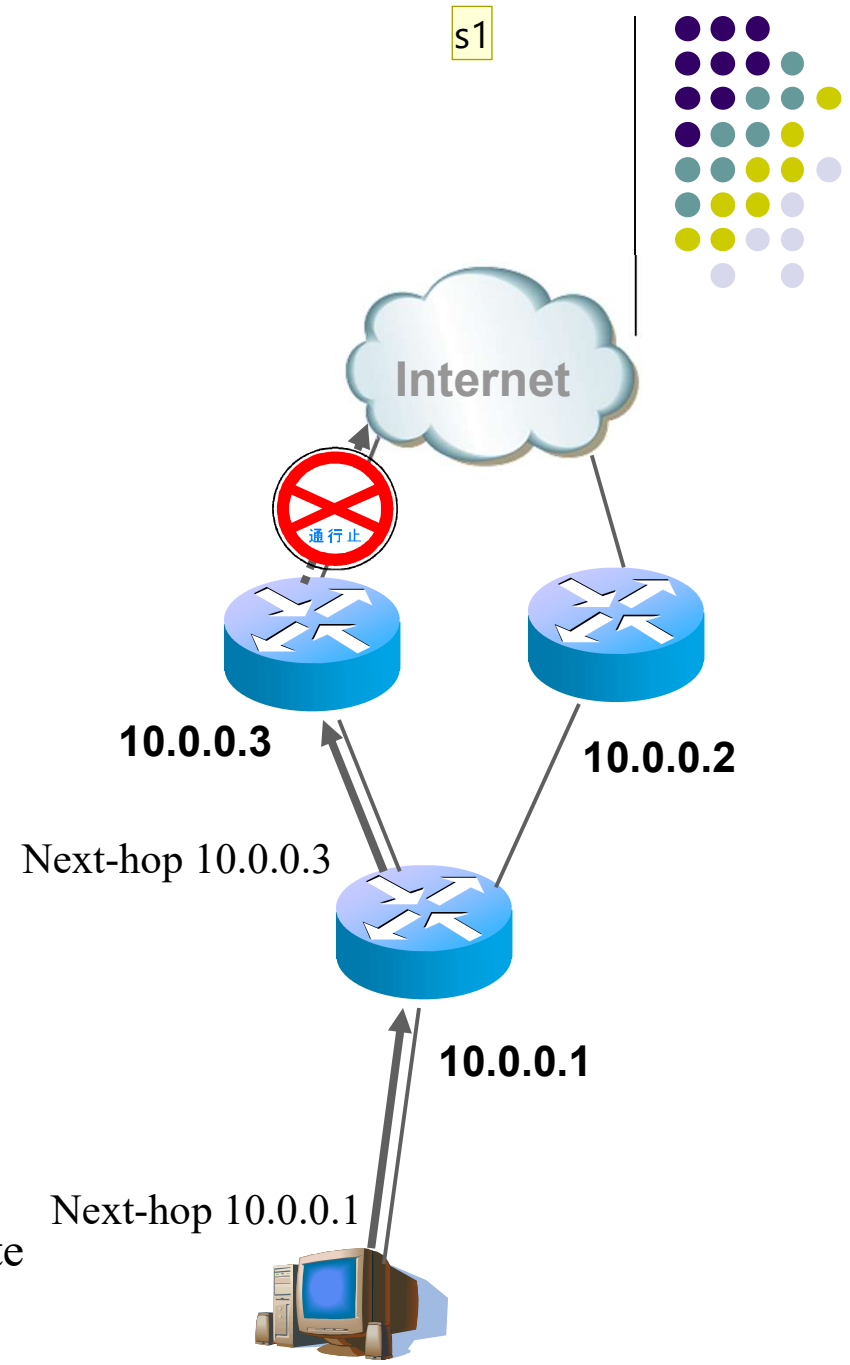
How to update routing table?



- Static routing
  - Entries in routing table are added manually by administrator
- Dynamic routing
  - Automatically update routing table
  - By mean of routing protocols

Static routing

- When there is a failure:
  - Problem happens even there are alternative routes.
  - Network administrator needs to change setting



Routing table of 10.0.0.1 (extract)

Prefix	Next-hop
~~0.0.0.0/0~~	~~10.0.0.3~~

Unreachable route

s1 Fix the ip address

sonnh, 8/03/2008

Dynamic Routing

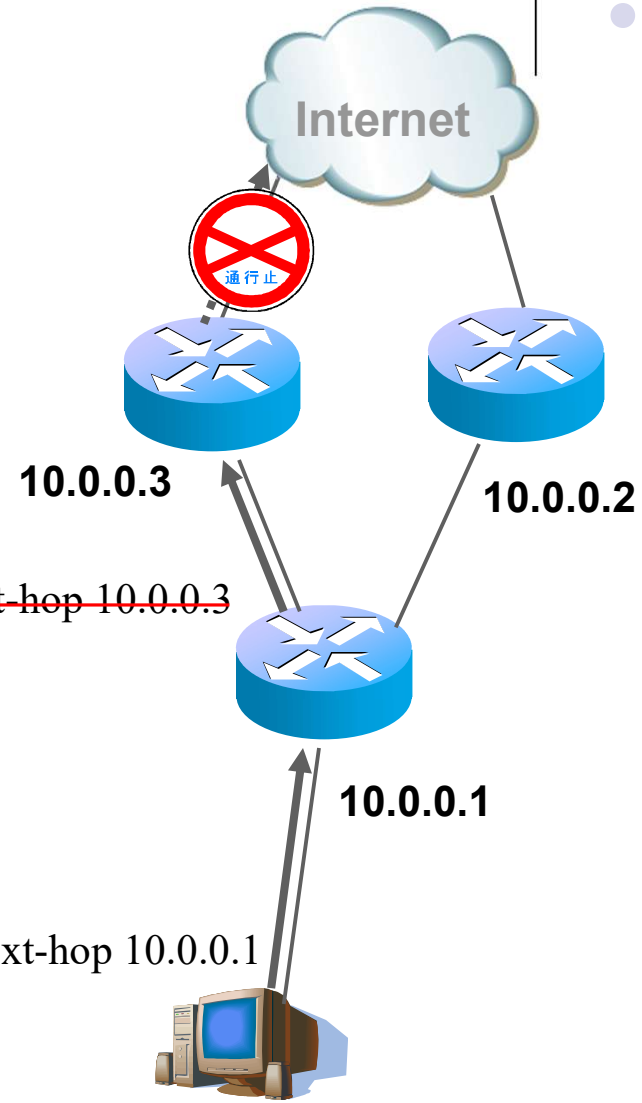
- When there is a failure:
  - Alternative route is added automatically

Routing table of 10.0.0.1 (extract)

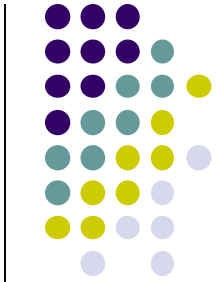
Prefix	Next-hop
~~0.0.0.0/0~~	~~10.0.0.3~~

Alternative route

Unreachable route



s2



Static routing

- Pros
  - Stable
  - Secure
  - It won’t be effected by other factors

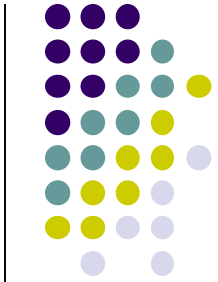
- Cons
  - Very stubborn
  - Back up link cannot be used
  - Difficult to manage

Slide 28

s2 If one ISP annouce the wrong routing information, so one part of Internet mis-operate

sonnh, 8/03/2008

Dynamic routing

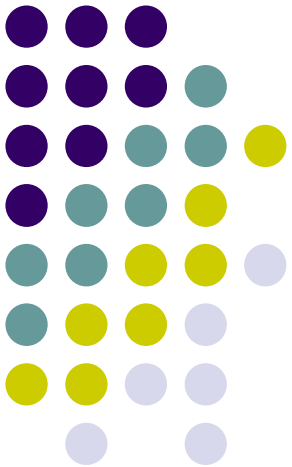


- Pros
  - Easy to manage
  - Backup link can be utilized
- Cons
  - Insecure
  - Difficult to understand the routing protocols

Routing algorithm and protocols

Dijkstra and Bellman-Ford Algo

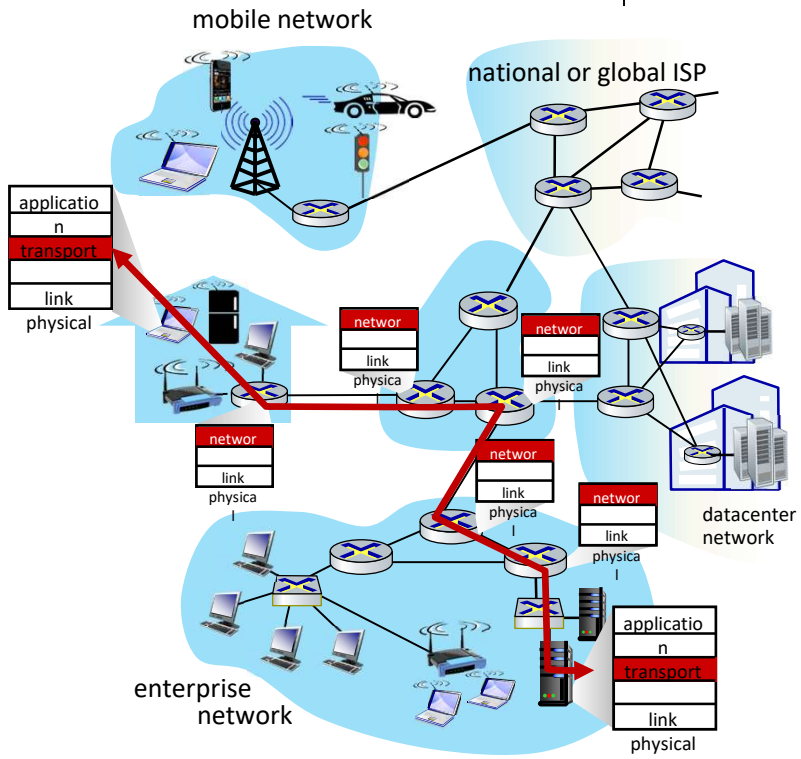
link-state and distance-vector protocols



Routing protocols

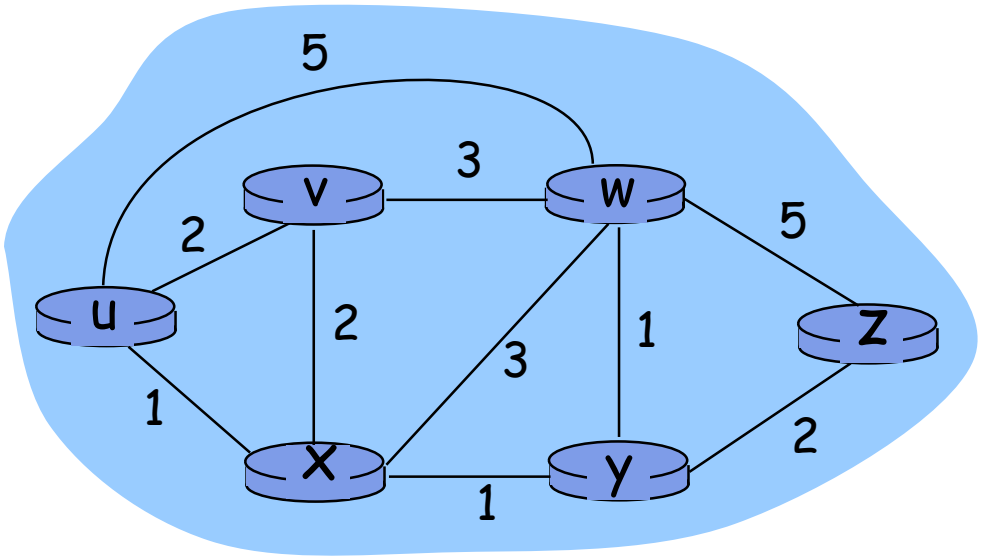
Routing protocol goal: determine "good" paths (equivalently, routes), from sending hosts to receiving host, through network of routers

- path: sequence of routers packets traverse from given initial source host to final destination host
- "good": least "cost", "fastest", "least congested"
- routing: a "top-10" networking challenge!

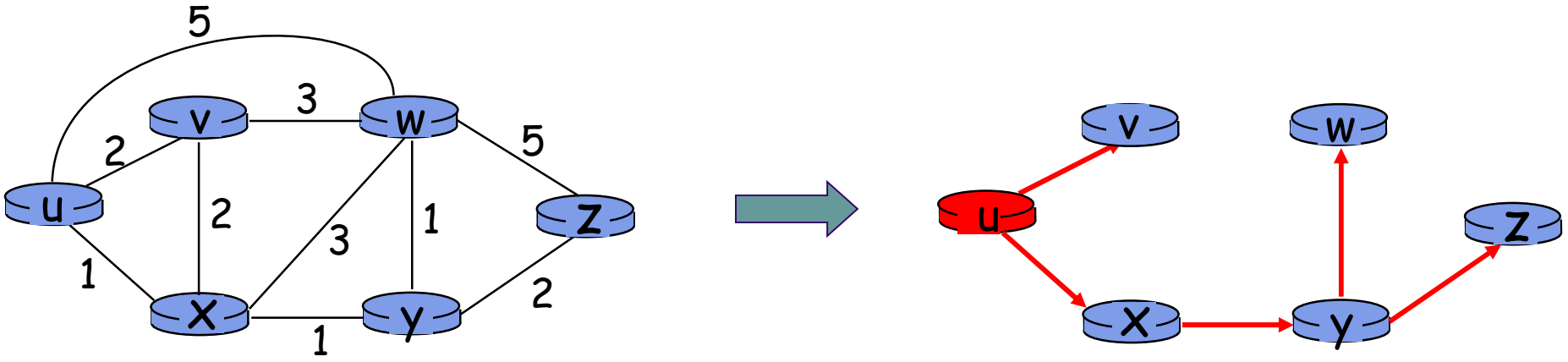


Network as a graph

- Graph with nodes (routers) and edges (links)
- Link “cost” c(x,y)
  - Bandwidth, delay, cost, congestion level…
- Determine least cost path from every node to every other node

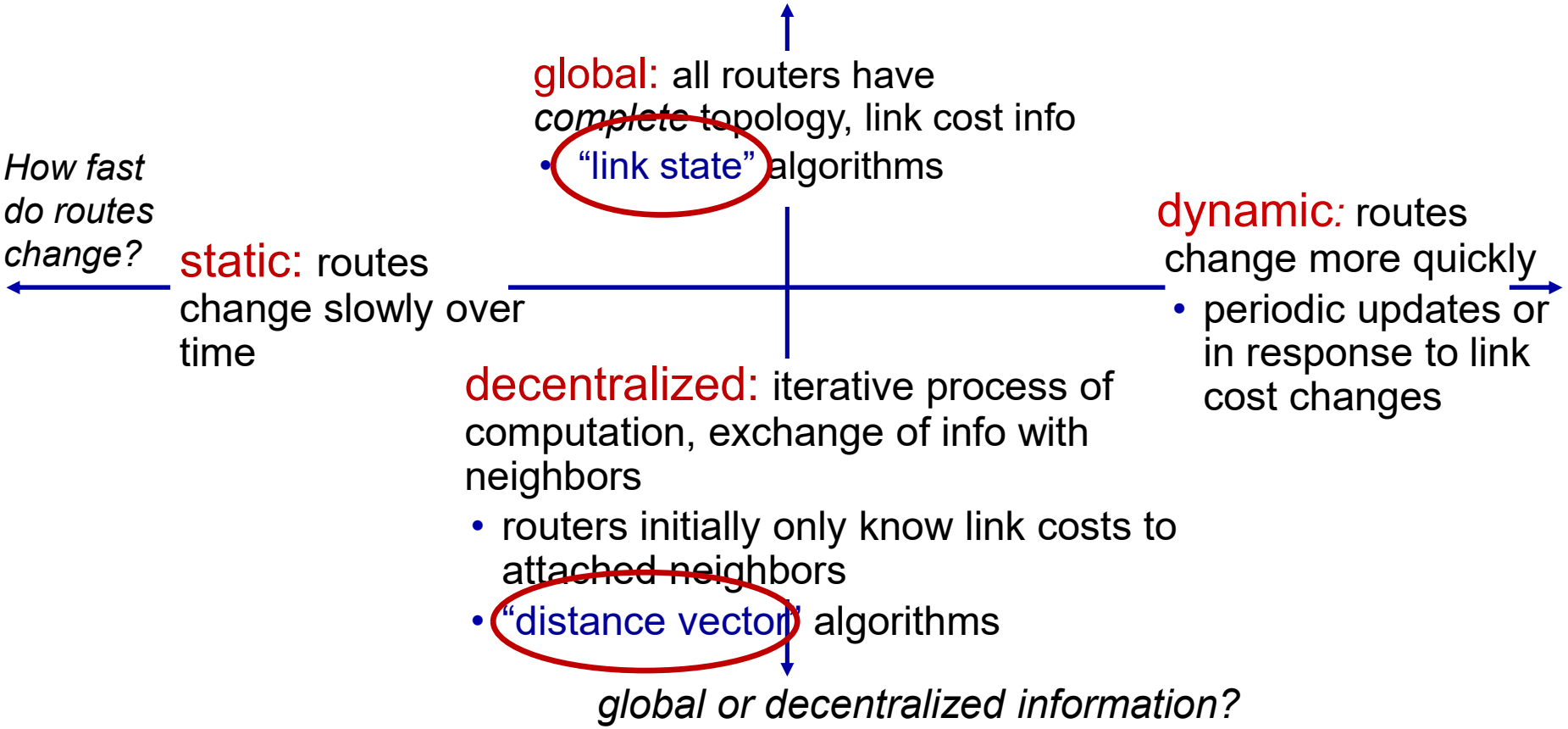


Shortest path tree - SPT

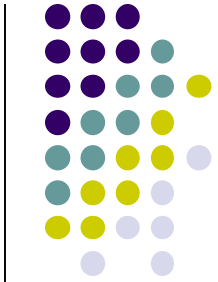


- A tree that links go out from root to leaves
- The unique path from root to any node v is the shortest path from the root to v
- Each node has a different SPT

Routing algorithm classification



Dijkstra's link-state routing algorithm

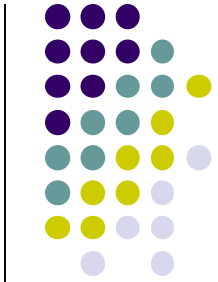


- centralized: network topology, link costs known to *all* nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
  - gives *forwarding table* for that node
- iterative: after *k* iterations, know least cost path to *k* destinations

notation

- $c_{x,y}$: direct link cost from node *x* to *y*; = ∞ if not direct neighbors
- *D(v): current* estimate of cost of least-cost-path from source to destination *v*
- *p(v):* predecessor node along path from source to *v*
- *N':* set of nodes whose least-cost-path *definitively* known

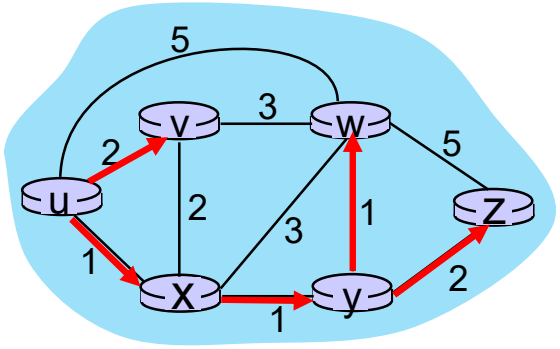
Dijkstra's link-state routing algorithm



```
1  Initialization:
2    N' = {u}                          /* compute least cost path from u to all other nodes */
3    for all nodes v
4      if v adjacent to u              /* u initially knows direct-path-cost only to direct
neighbors    */
5          then D(v) = c_{u,v}         /* but may not be minimum cost!
*/
6      else D(v) = ∞
7
8   Loop
9      find w not in N' such that D(w) is a minimum
10     add w to N'
11     update D(v) for all v adjacent to w and not in N' :
12         D(v) = min ( D(v),  D(w) + c_{w,v} )
13     /* new least-path-cost to v is either old least-cost-path to v or known
14     least-cost-path to w plus direct-cost from w to v */
15  until all nodes in N'
```

Dijkstra's algorithm: an example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

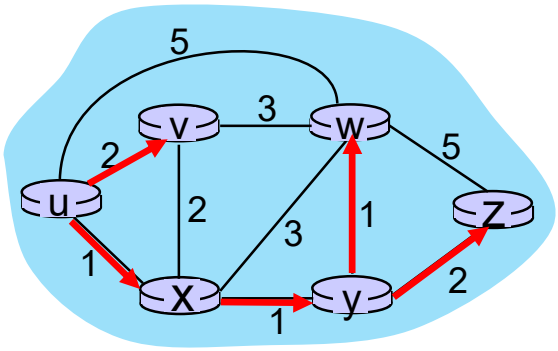


Initialization (step 0): For all a: if *a* adjacent to then $D(a) = c_{u,a}$

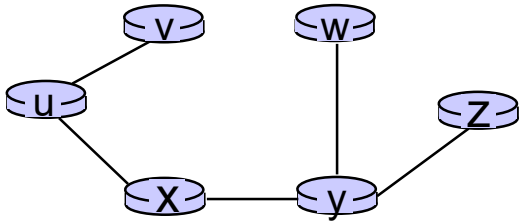
find *a* not in *N'* such that *D(a)* is a minimum
add *a* to *N'*
update *D(b)* for all *b* adjacent to *a* and not in *N'* :

$$D(b) = \min (D(b), D(a) + c_{a,b})$$

Dijkstra's algorithm: an example



resulting least-cost-path tree from u:



resulting forwarding table in u:

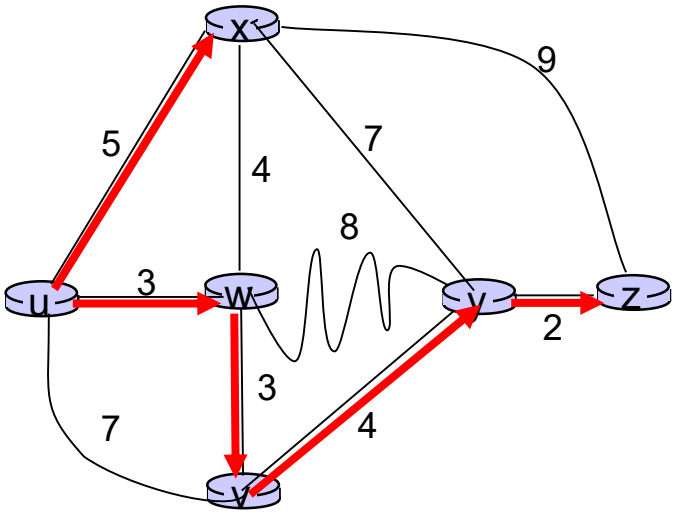
destination	outgoing link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

route from *u* to *v* directly

route from u to all other destinations via *x*

Dijkstra's algorithm: another example

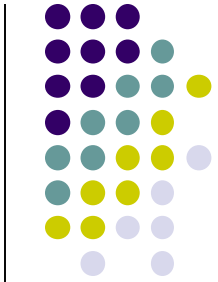
Step	N′	v D(v), p(v)	w D(w), p(w)	x D(x), p(x)	y D(y), p(y)	z D(z), p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					



notes:

- construct least-cost-path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)

Dijkstra's algorithm: discussion



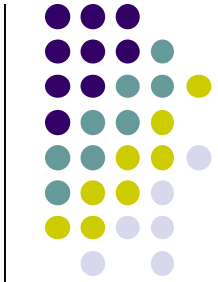
algorithm complexity: *n* nodes

- each of *n* iteration: need to check all nodes, *w*, not in *N*
- $n(n+1)/2$ comparisons: $O(n^2)$ complexity
- more efficient implementations possible: $O(n\log n)$

message complexity:

- each router must *broadcast* its link state information to other *n* routers
- efficient (and interesting!) broadcast algorithms: $O(n)$ link crossings to disseminate a broadcast message from one source
- each router's message crosses $O(n)$ links: overall message complexity: $O(n^2)$

Distance vector algorithm



Based on *Bellman-Ford* (BF) equation (dynamic programming):

Bellman-Ford equation

Let $D_x(y)$: cost of least-cost path from x to y.
Then:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

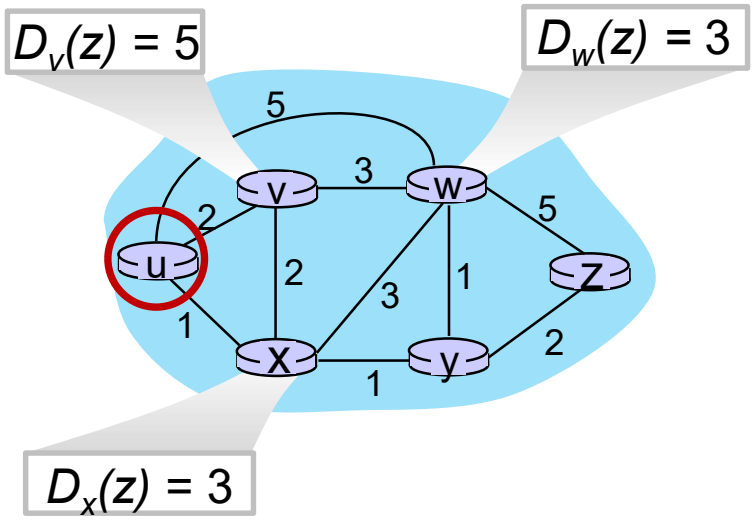
v's estimated least-cost-path cost to y

min taken over all neighbors v of x

direct cost of link from x to v

Bellman-Ford Example

Suppose that *u*'s neighboring nodes, *x,v,w,* know that for destination *z*:

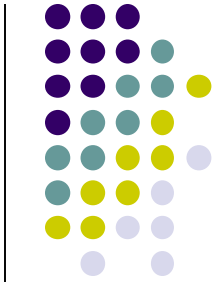


Bellman-Ford equation says:

$$D_u(z) = \min \{ c_{u,v} + D_v(z), \; c_{u,x} + D_x(z), \; c_{u,w} + D_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

node achieving minimum (x) is next hop on estimated least-cost path to destination (z)

Distance-vector algorithm (2)



Main ideas:

- Distance vector: vector of all distance from the current node to all other nodes
- Each node send periodically the its distance vector to its adjacent nodes
- When a node x receives a distance vector, it updates its distance vector by using equation Bellman-ford
- With some condition, the distance $D_x(y)$ *in each vector will converge to the smallest value of* $d_x(y)$

At each node:

Wait for a DV from neighbor

Re-calculate its DV

If DV changes, *Inform its neighbor*

$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

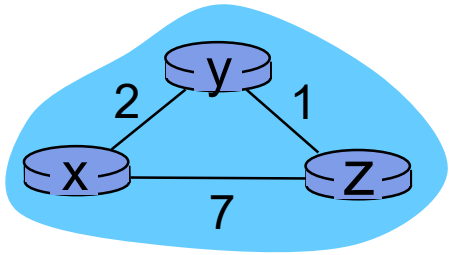
$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



time

$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

from	cost to x	cost to y	cost to z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

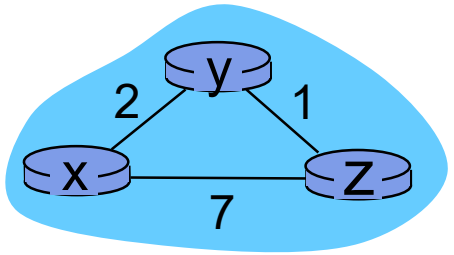
from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	7	1	0

node z table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	3	1	0



$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

from	cost to x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from	cost to x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

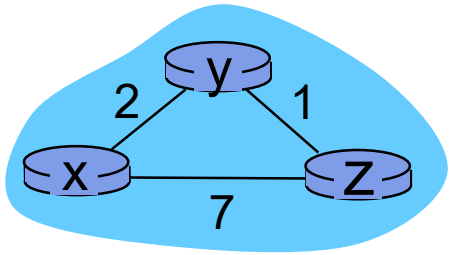
from	cost to x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from	cost to x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

node z table

from	cost to x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from	cost to x	y	z
x	0	2	7
y	2	0	1
z	3	1	0



time

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

node x table

from	cost to x	cost to y	cost to z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

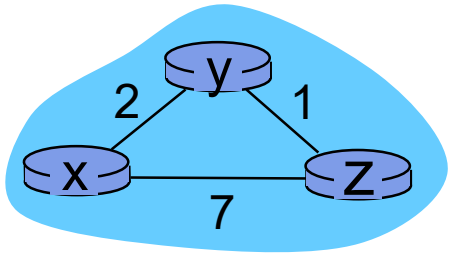
node z table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	3	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

time

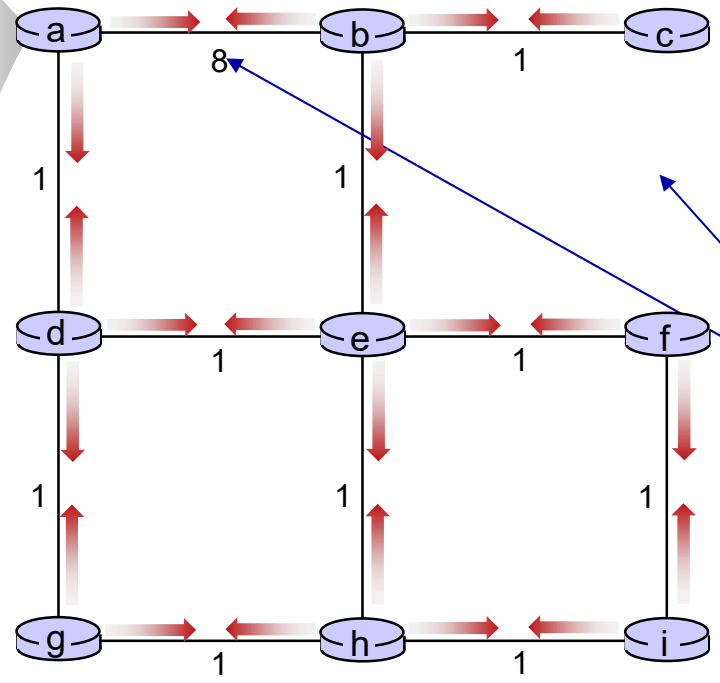


Distance vector: example

t=0

DV in :

$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$



- All nodes have distance estimates to nearest neighbors (only)
- All nodes send their local distance vector to their neighbors

A few asymmetries:

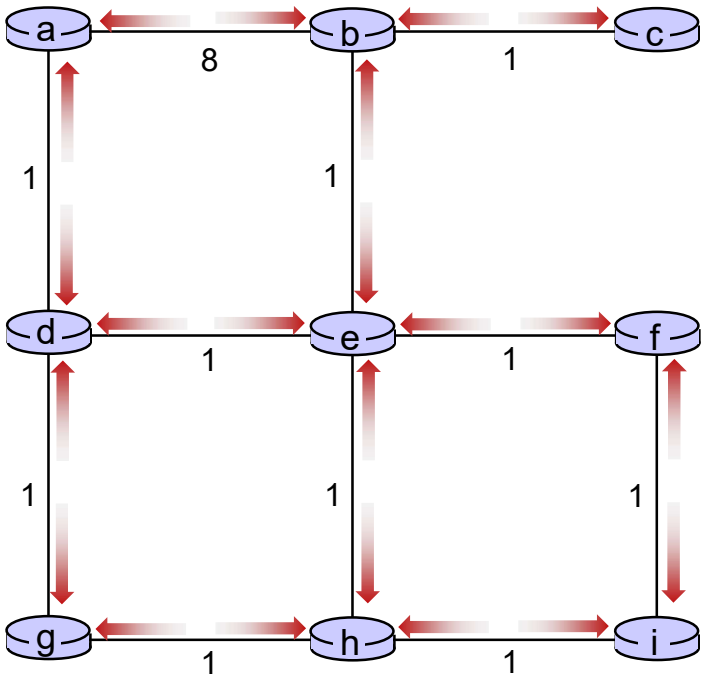
- missing link
- larger cost

Distance vector example: iteration

t=1

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

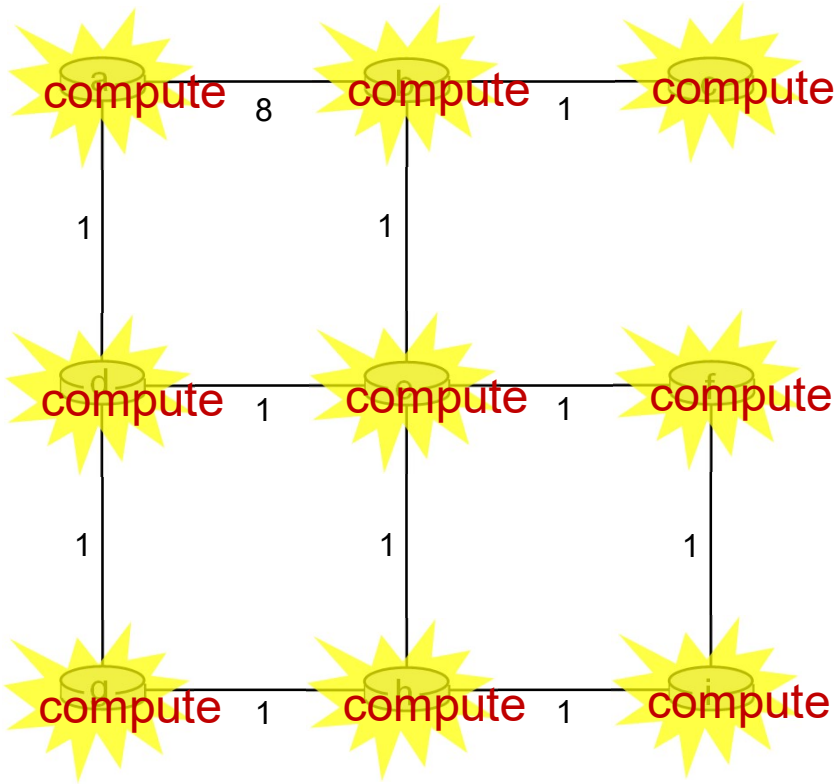


Distance vector example: iteration

t=1

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

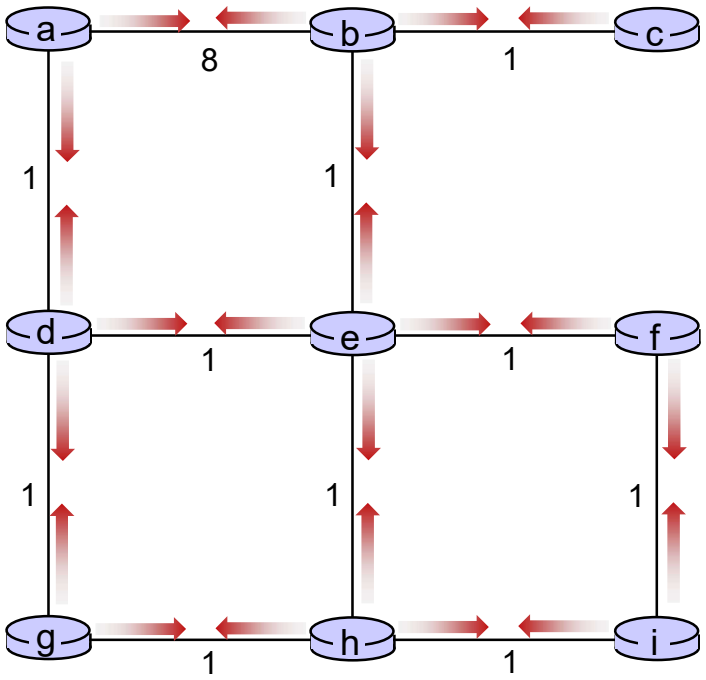


Distance vector example: iteration

t=1

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

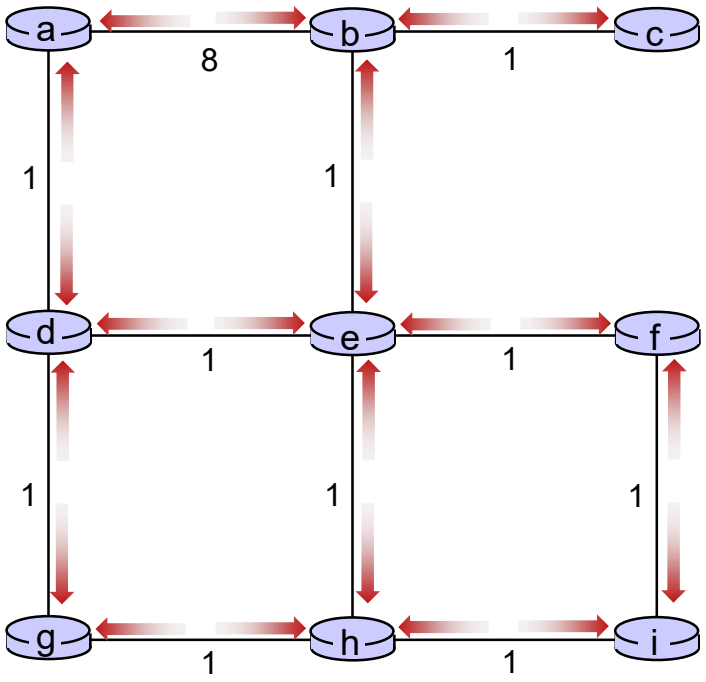


Distance vector example: iteration

t=2

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

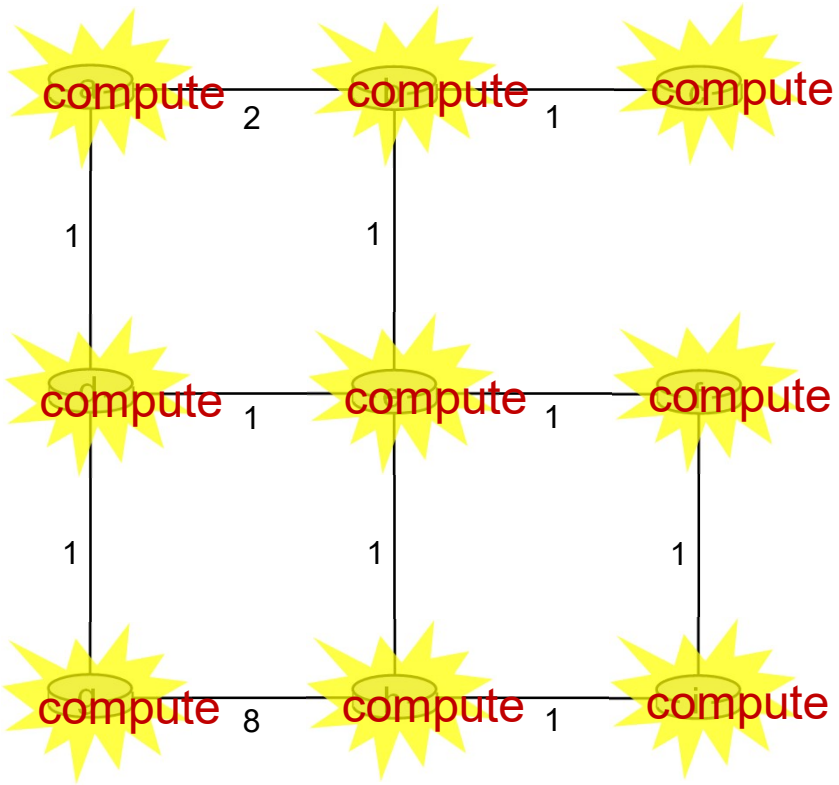


Distance vector example: iteration

t=2

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

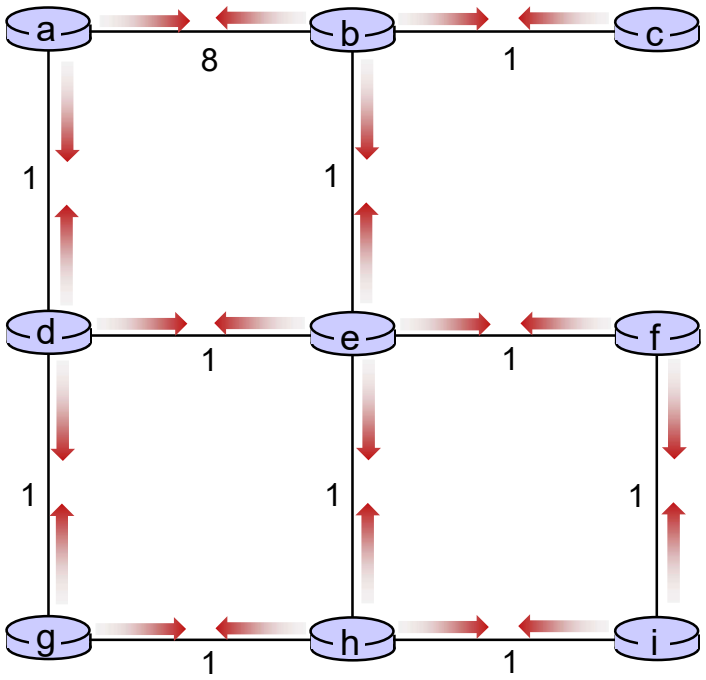


Distance vector example: iteration

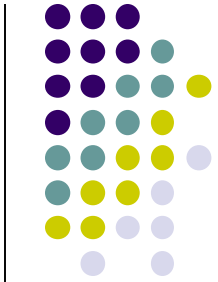
t=2

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



Distance vector example: iteration



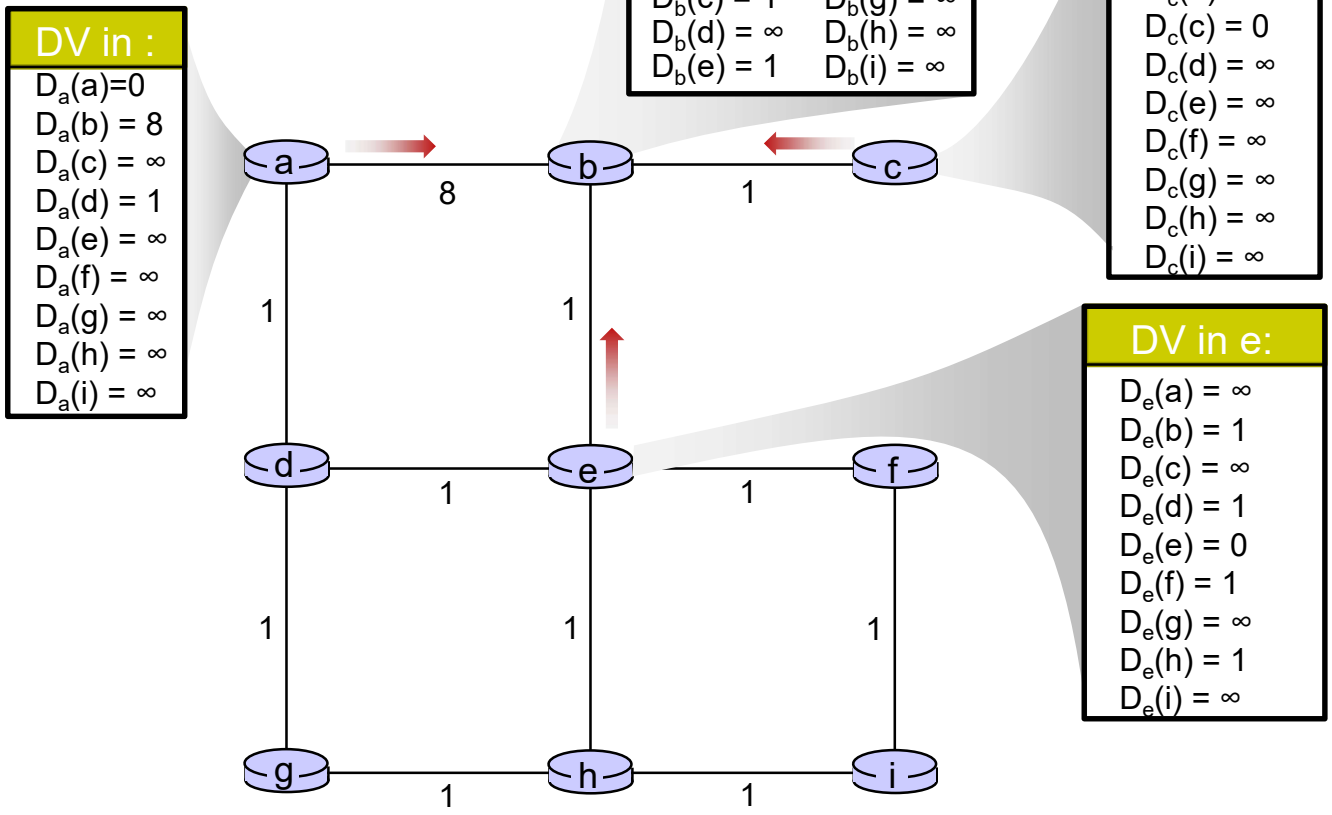
…. and so on

Let's next take a look at the iterative *computations* at nodes

Distance vector example: computation

t=1

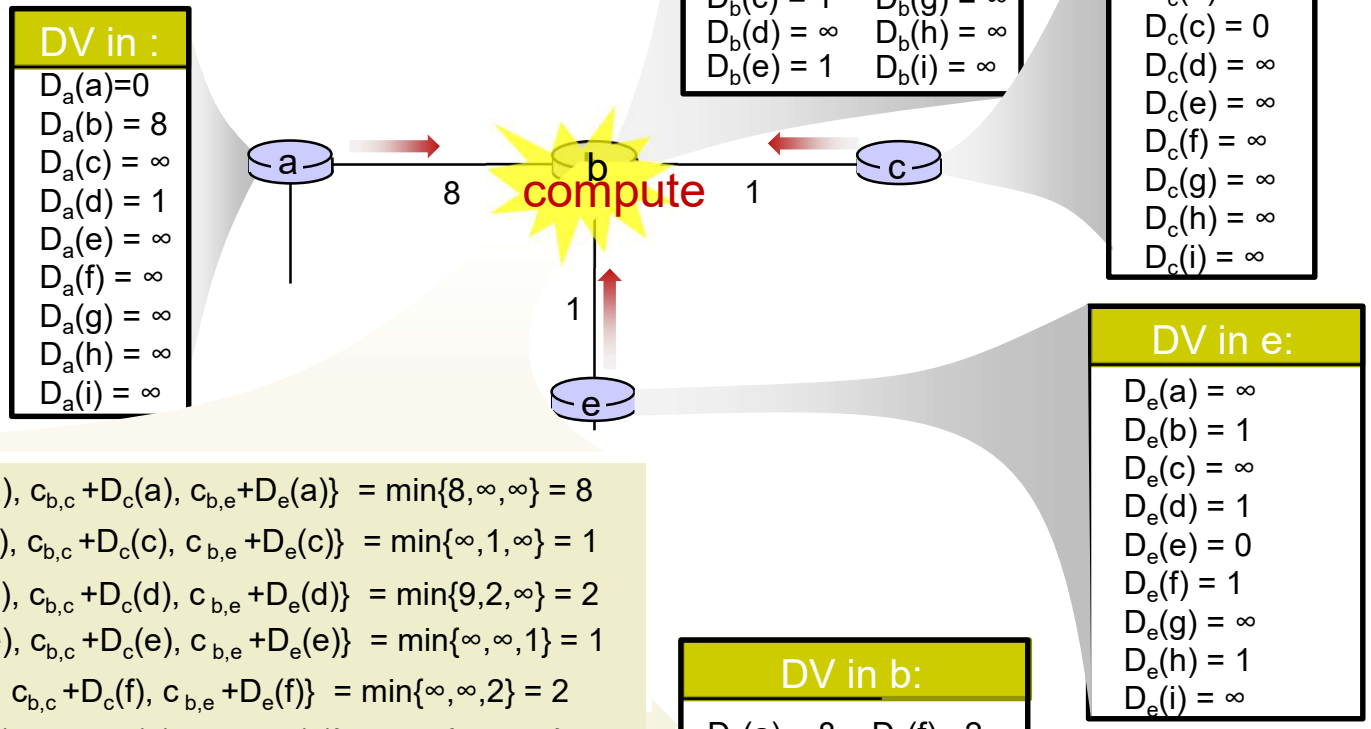
- b receives DVs from a, c, e



Distance vector example: computation

t=1

- b receives DVs from a, c, e, computes:



$D_b(a) = \min\{c_{b,a}+D_a(a), c_{b,c}+D_c(a), c_{b,e}+D_e(a)\} = \min\{8,\infty,\infty\} = 8$

$D_b(c) = \min\{c_{b,a}+D_a(c), c_{b,c}+D_c(c), c_{b,e}+D_e(c)\} = \min\{\infty,1,\infty\} = 1$

$D_b(d) = \min\{c_{b,a}+D_a(d), c_{b,c}+D_c(d), c_{b,e}+D_e(d)\} = \min\{9,2,\infty\} = 2$

$D_b(e) = \min\{c_{b,a}+D_a(e), c_{b,c}+D_c(e), c_{b,e}+D_e(e)\} = \min\{\infty,\infty,1\} = 1$

$D_b(f) = \min\{c_{b,a}+D_a(f), c_{b,c}+D_c(f), c_{b,e}+D_e(f)\} = \min\{\infty,\infty,2\} = 2$

$D_b(g) = \min\{c_{b,a}+D_a(g), c_{b,c}+D_c(g), c_{b,e}+D_e(g)\} = \min\{\infty, \infty, \infty\} = \infty$

$D_b(h) = \min\{c_{b,a}+D_a(h), c_{b,c}+D_c(h), c_{b,e}+D_e(h)\} = \min\{\infty, \infty, 2\} = 2$

$D_b(i) = \min\{c_{b,a}+D_a(i), c_{b,c}+D_c(i), c_{b,e}+D_e(i)\} = \min\{\infty, \infty, \infty\} = \infty$

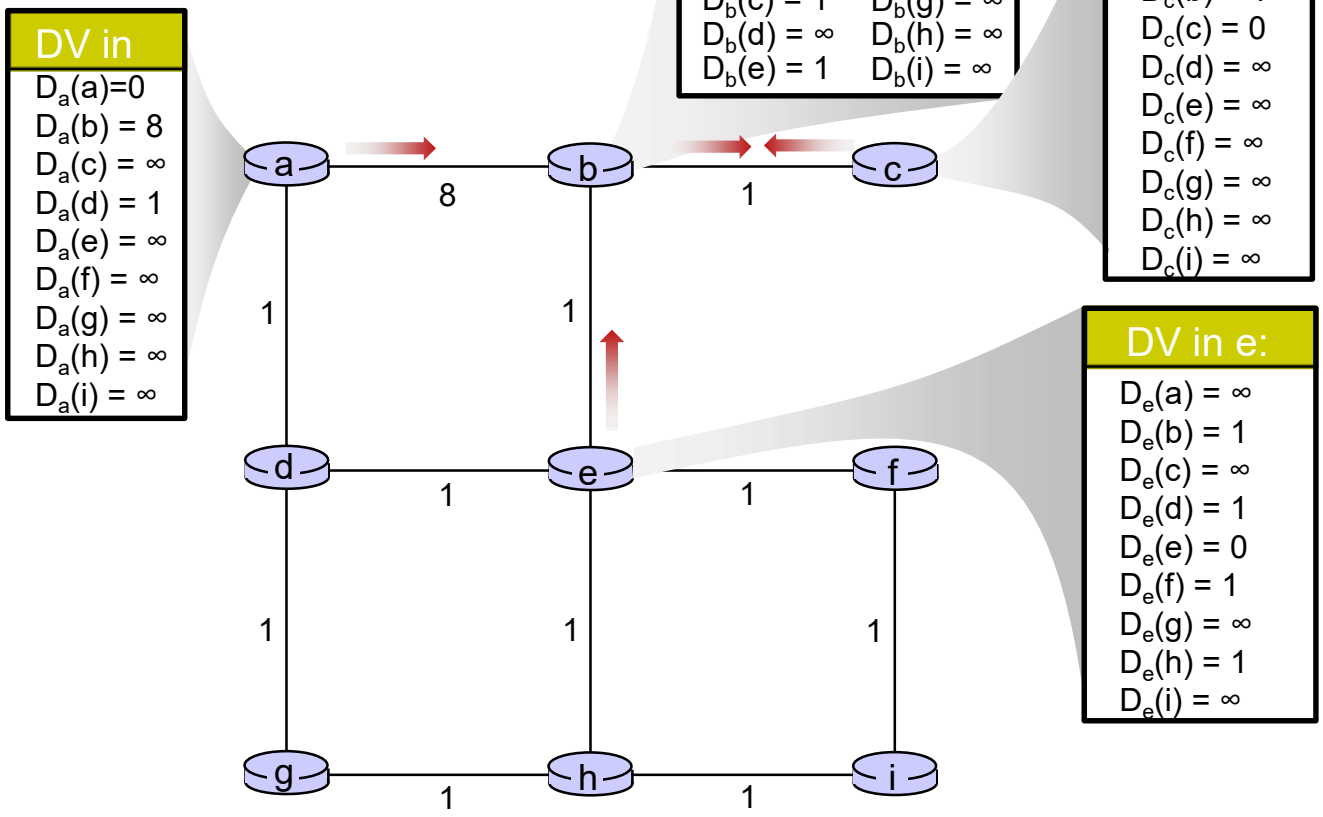
DV in b:

$D_b(a) = 8$	$D_b(f) = 2$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = 2$	$D_b(h) = 2$
$D_b(e) = 1$	$D_b(i) = \infty$

Distance vector example: computation

t=1

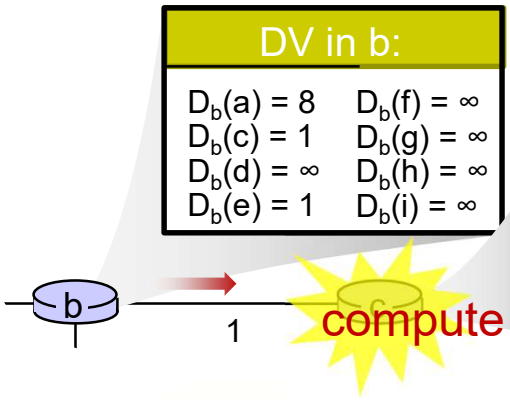
- c receives DVs from b



Distance vector example: computation

t=1

- c receives DVs from b computes:



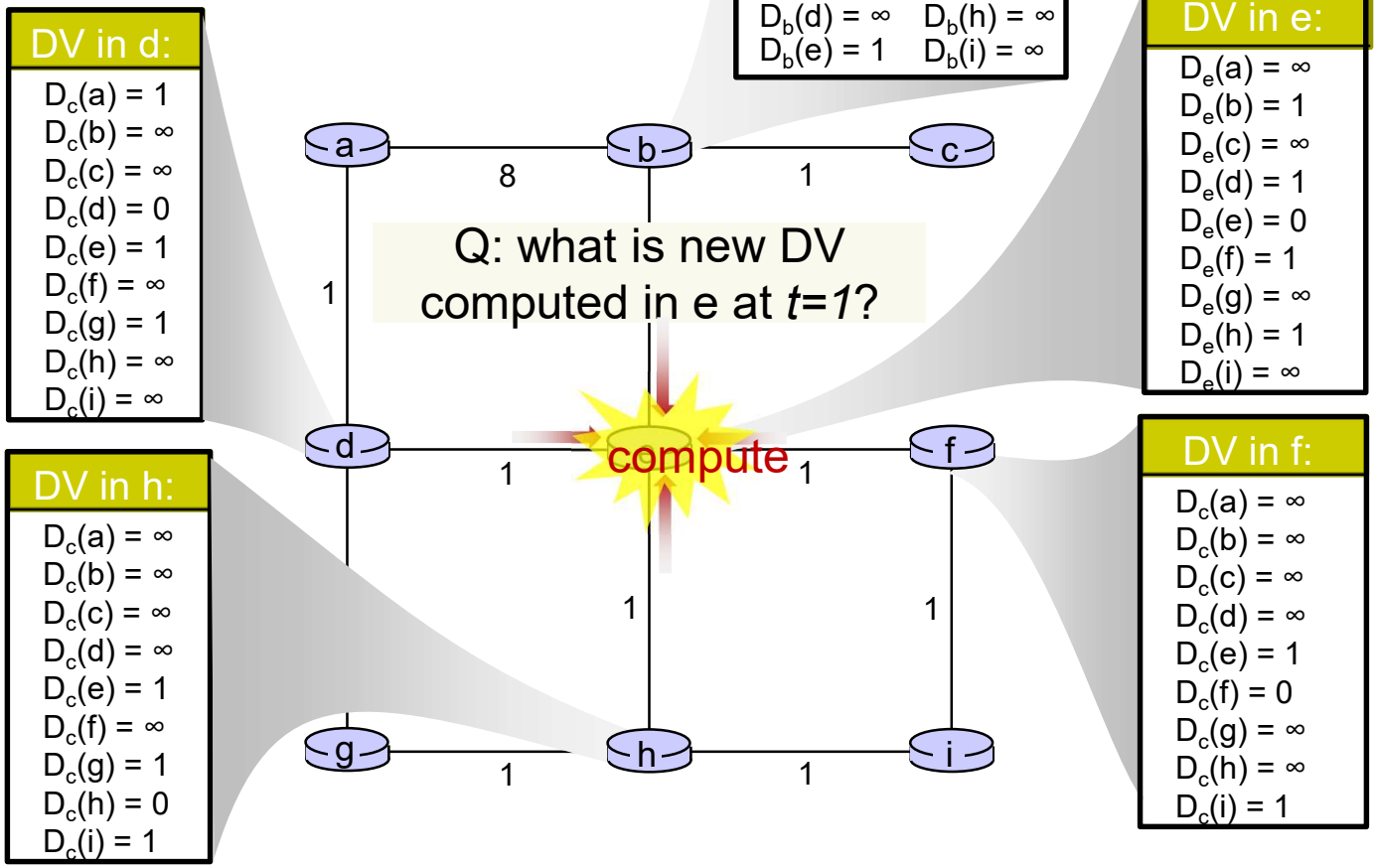
$D_c(a) = \min\{c_{c,b}+D_b(a)\} = 1 + 8 = 9$
$D_c(b) = \min\{c_{c,b}+D_b(b)\} = 1 + 0 = 1$
$D_c(d) = \min\{c_{c,b}+D_b(d)\} = 1+ \infty = \infty$
$D_c(e) = \min\{c_{c,b}+D_b(e)\} = 1 + 1 = 2$
$D_c(f) = \min\{c_{c,b}+D_b(f)\} = 1+ \infty = \infty$
$D_c(g) = \min\{c_{c,b}+D_b(g)\} = 1+ \infty = \infty$
$D_c(h) = \min\{c_{bc,b}+D_b(h)\} = 1+ \infty = \infty$
$D_c(i) = \min\{c_{c,b}+D_b(i)\} = 1+ \infty = \infty$

DV in c:
$D_c(a) = 9$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = 2$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

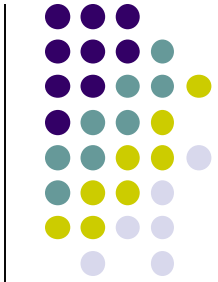
Distance vector example: computation

t=1

- e receives DVs from b, d, f, h



Distance vector: state information diffusion



Iterative communication, computation steps diffuses information through network:

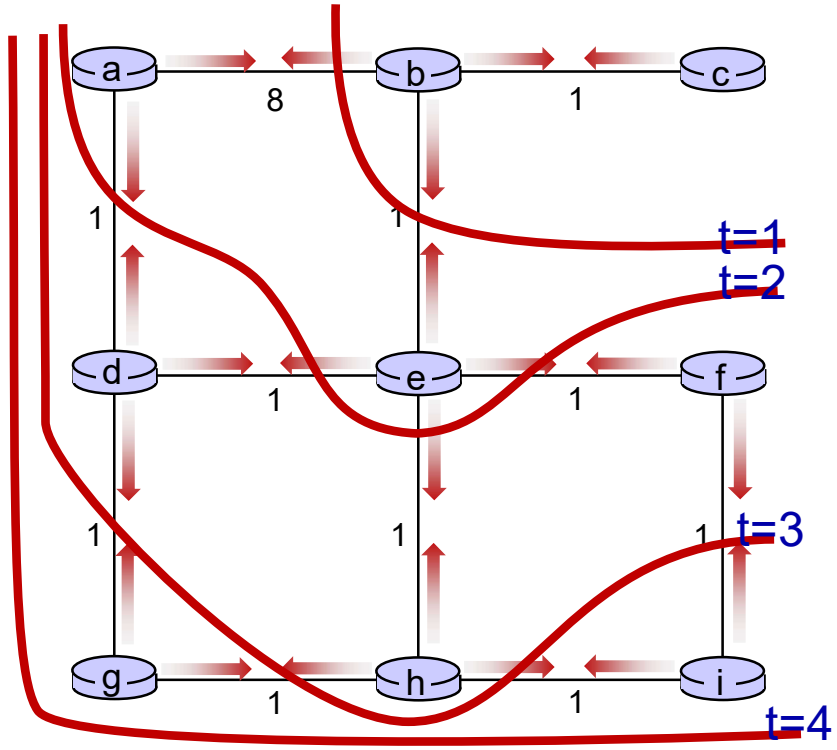
t=0 c's state at t=0 is at c only

t=1 c's state at t=0 has propagated to b, and may influence distance vector computations up to **1** hop away, i.e., at b

t=2 c's state at t=0 may now influence distance vector computations up to **2** hops away, i.e., at b and now at a, e as well

t=3 c's state at t=0 may influence distance vector computations up to **3** hops away, i.e., at b,a,e and now at c,f,h as well

t=4 c's state at t=0 may influence distance vector computations up to **4** hops away, i.e., at b,a,e, c, f, h and now at g,i as well



Comparison of Link-state and Distance vector

Number of exchange messages

- LS: n nodes, E links, O(nE) messages
- DV: Exchange only with neighbor

Convergent time

- LS: Complexity $O(n^2)$
- DV: Varies

Reliability: If one routers provide incorrect information

LS:

- The router may send out incorrect cost
- Each node calculate its own routing table

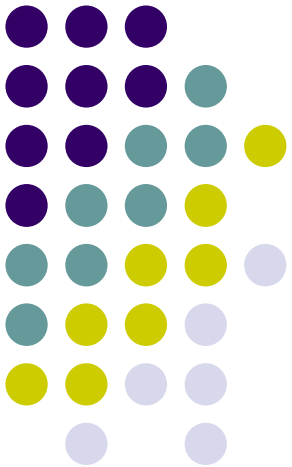
DV:

- Incorrect distance vector may be sent out
- Each node calculate its DV based to what receives from the neighbor
  - Error propagates in the network.

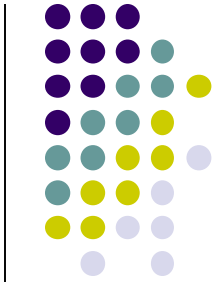
Hierarchical routing

Autonomous System

Intra and Inter domain routing



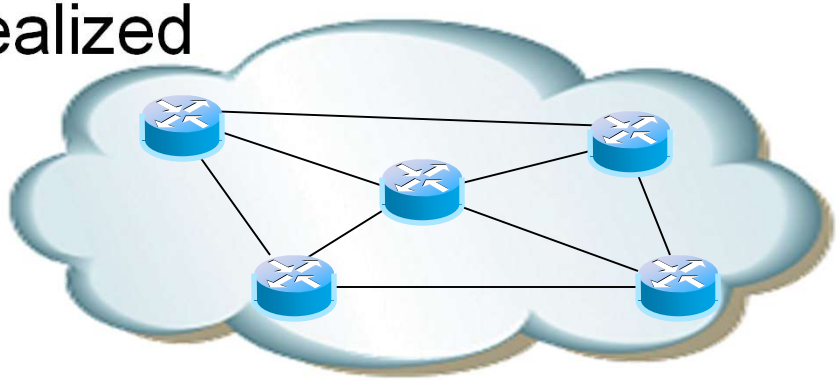
Making routing scalable



our routing study thus far - idealized

- all routers identical
- network “flat”

… not true in practice



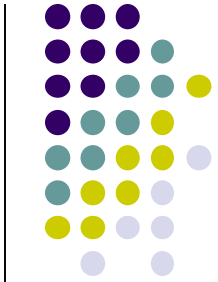
scale: billions of destinations:

- can’t store all destinations in routing tables!
- routing table exchange would swamp links!

administrative autonomy:

- Internet: a network of networks
- each network admin may want to control routing in its own network

Internet approach to scalable routing



aggregate routers into regions known as "autonomous systems" (AS) (a.k.a. "domains")

intra-AS (aka "intra-domain"): routing among *within same AS ("network")*

- all routers in AS must run same intra-domain protocol
- routers in different AS can run different intra-domain routing protocols
- gateway router: at "edge" of its own AS, has link(s) to router(s) in other AS'es

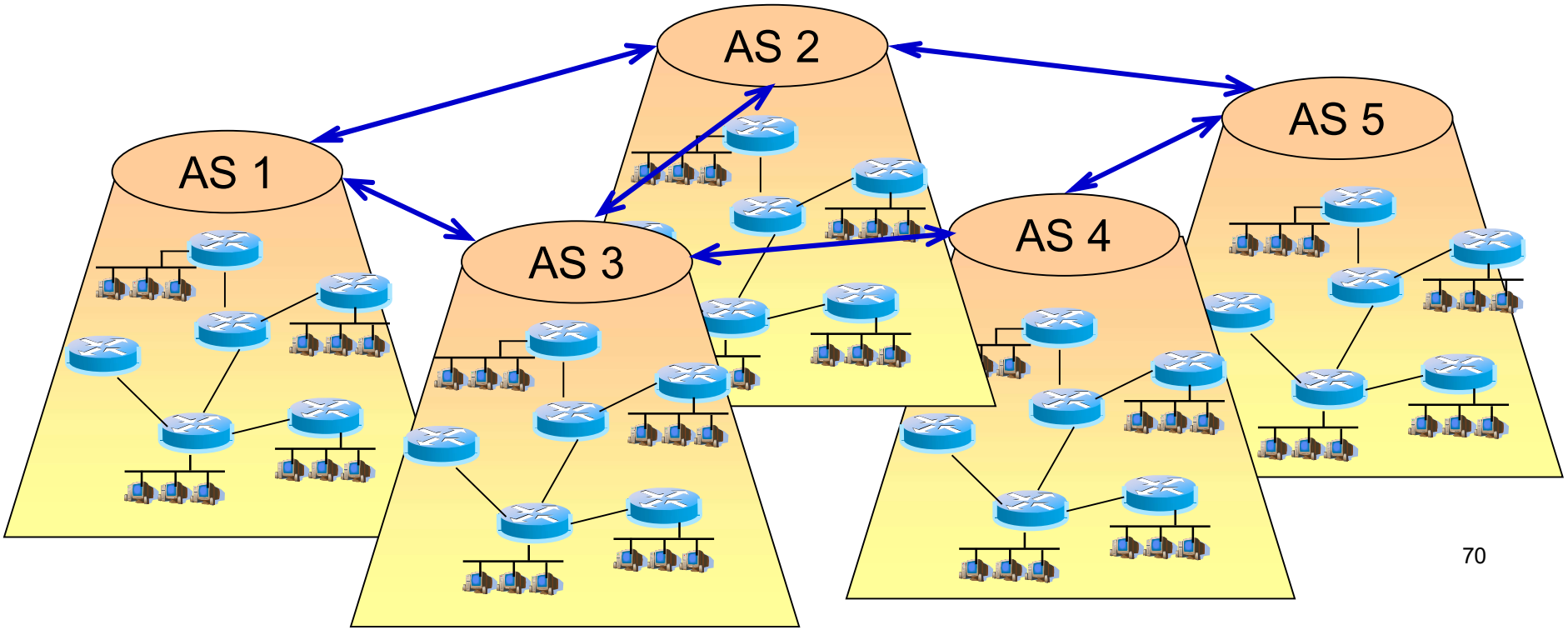
inter-AS (aka "inter-domain"): routing *among* AS'es

- gateways perform inter-domain routing (as well as intra-domain routing)

s3

Hierarchical structure of the Internet

- Internet = network of networks
- Such networks can select its own routing policy (routing domain)
- Such networks are called autonomous system (AS)

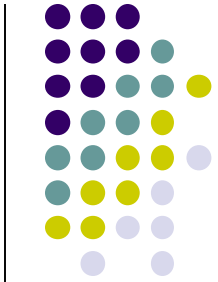


s3 Combine 5 and 6

sonnh, 8/03/2008

s4

Autonomous System (AS)



- A set of routers with the same routing policy (routing protocol, metric…) is aggregated into an AS
- Gateway: router connect between two ASes
- Each AS has an unique number (ASN - 16 bits or 32 bits).

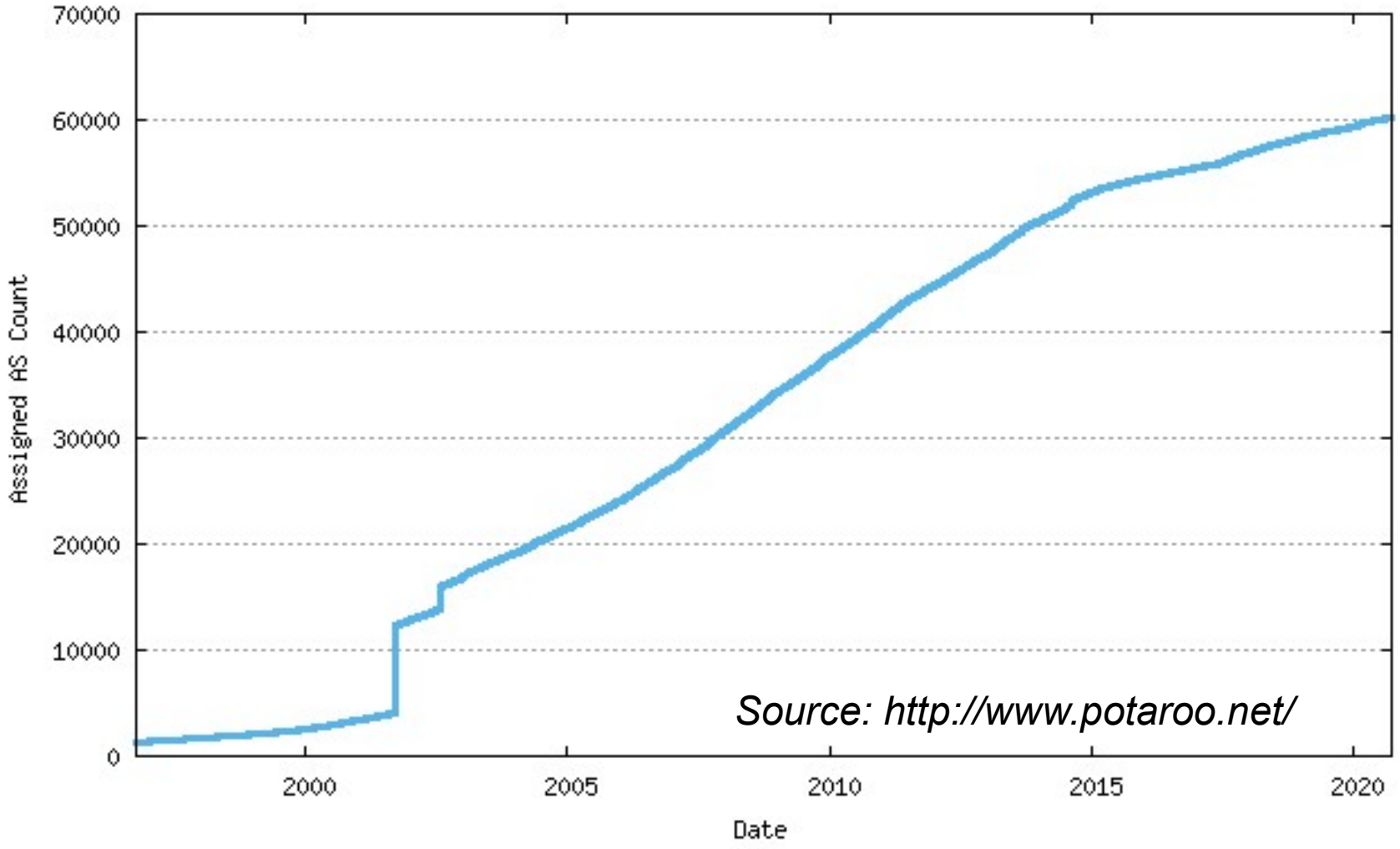
2914 NTT-COMMUNICATIONS-2914 - NTT America, Inc.
3491 BTN-ASN - Beyond The Network America, Inc.
4134 CHINANET-BACKBONE No.31,Jin-rong Street
6453 GLOBEINTERNET Teleglobe America Inc.
24087 VNGT-AS-AP Vietnam New Generation Telecom
24066 VNNIC-AS-VN Vietnam Internet Network Information Center
17981 CAMBOTECH-KH-AS ISP Cambodia
.....

Source: http://www.cidr-report.org

s4 Explain about AS

sonnh, 8/03/2008

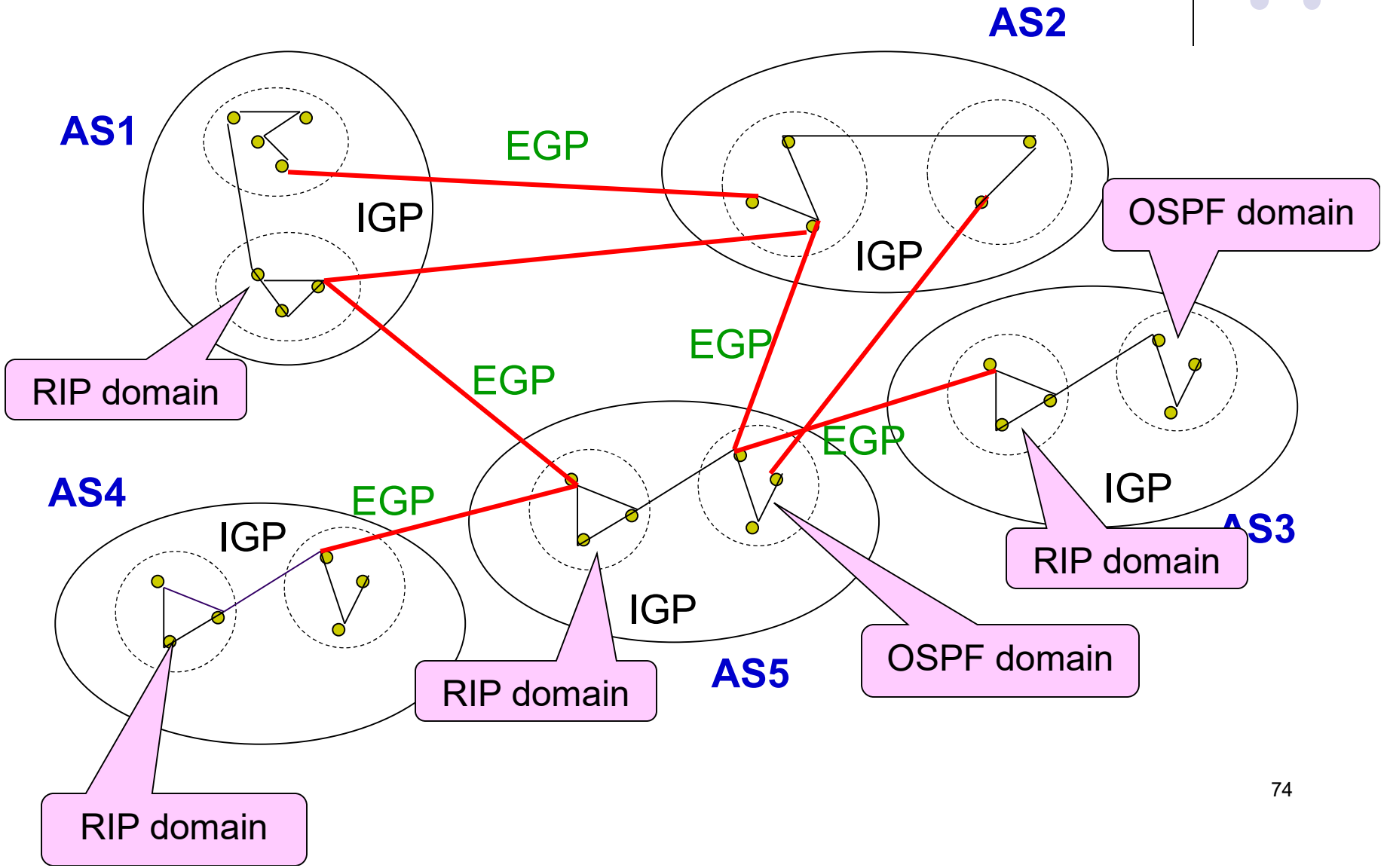
Number of AS by time



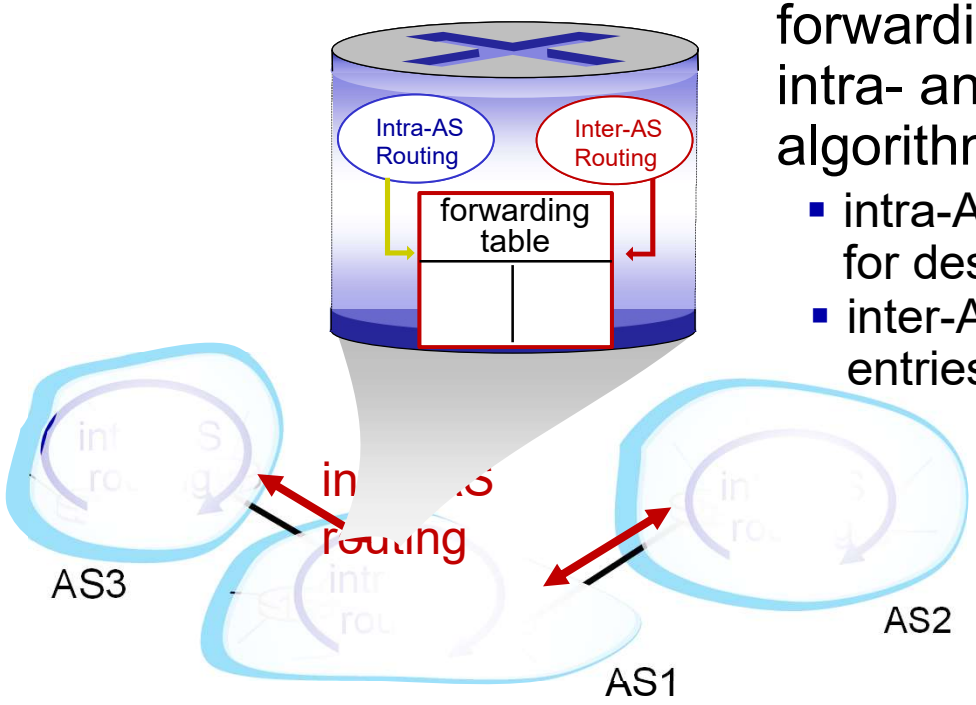
Hierarchical routing protocols

- Inside an AS: Intra-domain routing protocols
  - Also named IGP: *Interior Gateway Protocol*
    - RIP: Routing Information Protocol
    - OSPF: Open Shortest Path First
    - IS-IS, IGRP, EIGRP (Cisco)…
- Among ASes: Inter-domain routing protocols
  - Also named EGP: *Exterior Gateway Protocol*
    - BGP (v4): Border Gateway Protocol

Intra-domain and Inter-domain routing



Interconnected ASes



forwarding table configured by intra- and inter-AS routing algorithms

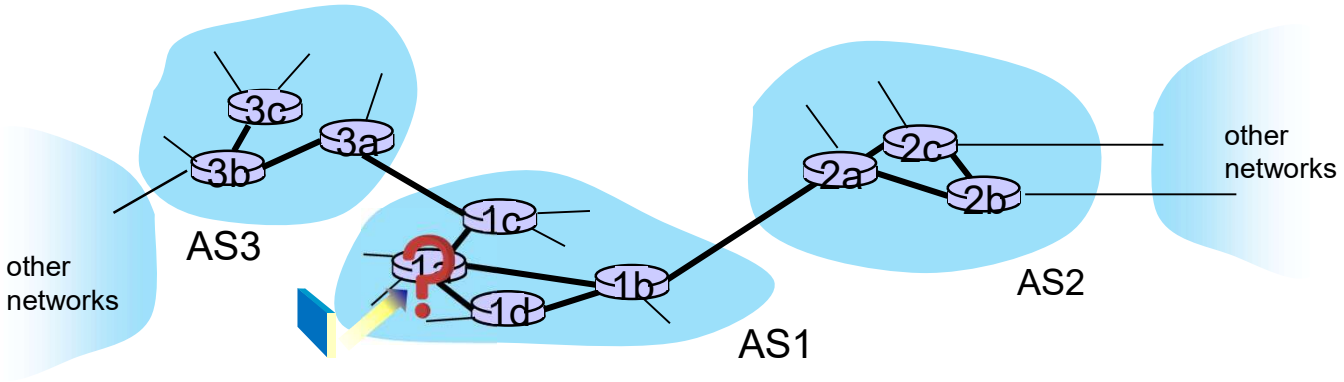
- intra-AS routing determine entries for destinations within AS
- inter-AS & intra-AS determine entries for external destinations

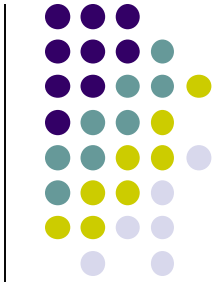
Inter-AS routing: a role in intradomain forwarding

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router in AS1, but which one?

AS1 inter-domain routing must:

1. learn which destinations reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1





Inter-AS routing: routing within an AS

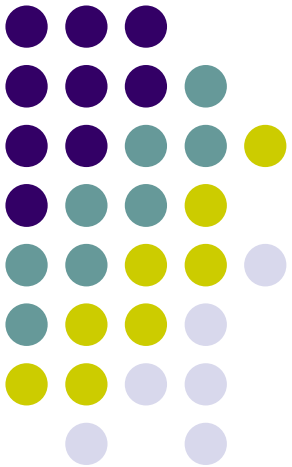
most common intra-AS routing protocols:

- RIP: Routing Information Protocol [RFC 1723]
  - classic DV: DVs exchanged every 30 secs
  - no longer widely used
- OSPF: Open Shortest Path First [RFC 2328]
  - link-state routing
  - IS-IS protocol (ISO standard, not RFC standard) essentially same as OSPF
- EIGRP: Enhanced Interior Gateway Routing Protocol
  - DV based
  - formerly Cisco-proprietary for decades (became open in 2013 [RFC 7868])

Intra-domain routing protocol

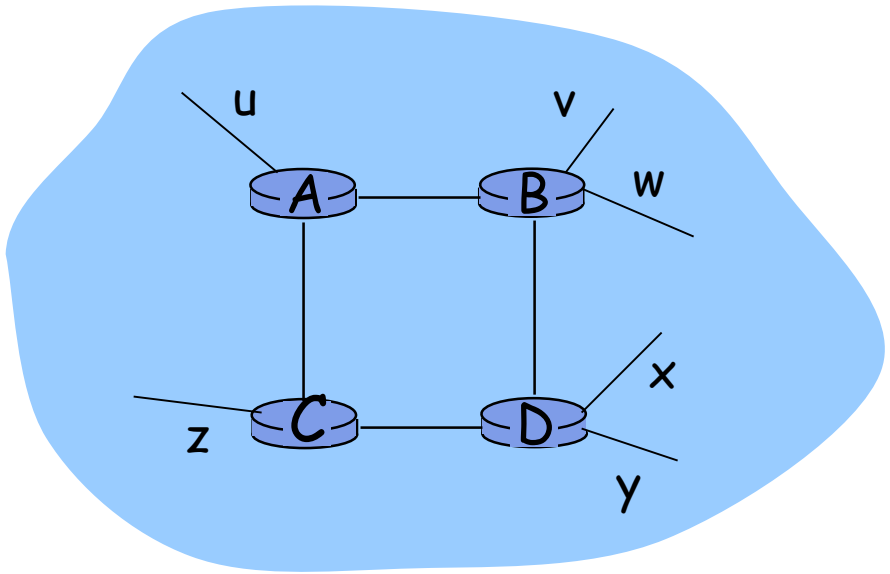
RIP

OSPF



RIP (Routing Information Protocol)

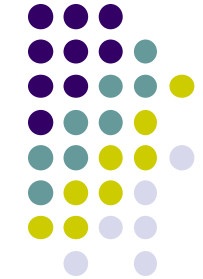
- IGP
- RIP v.1, currently use RIP v.2
- Distance-vector algorithm
- Routing metric: # of hops (max = 15 hops)



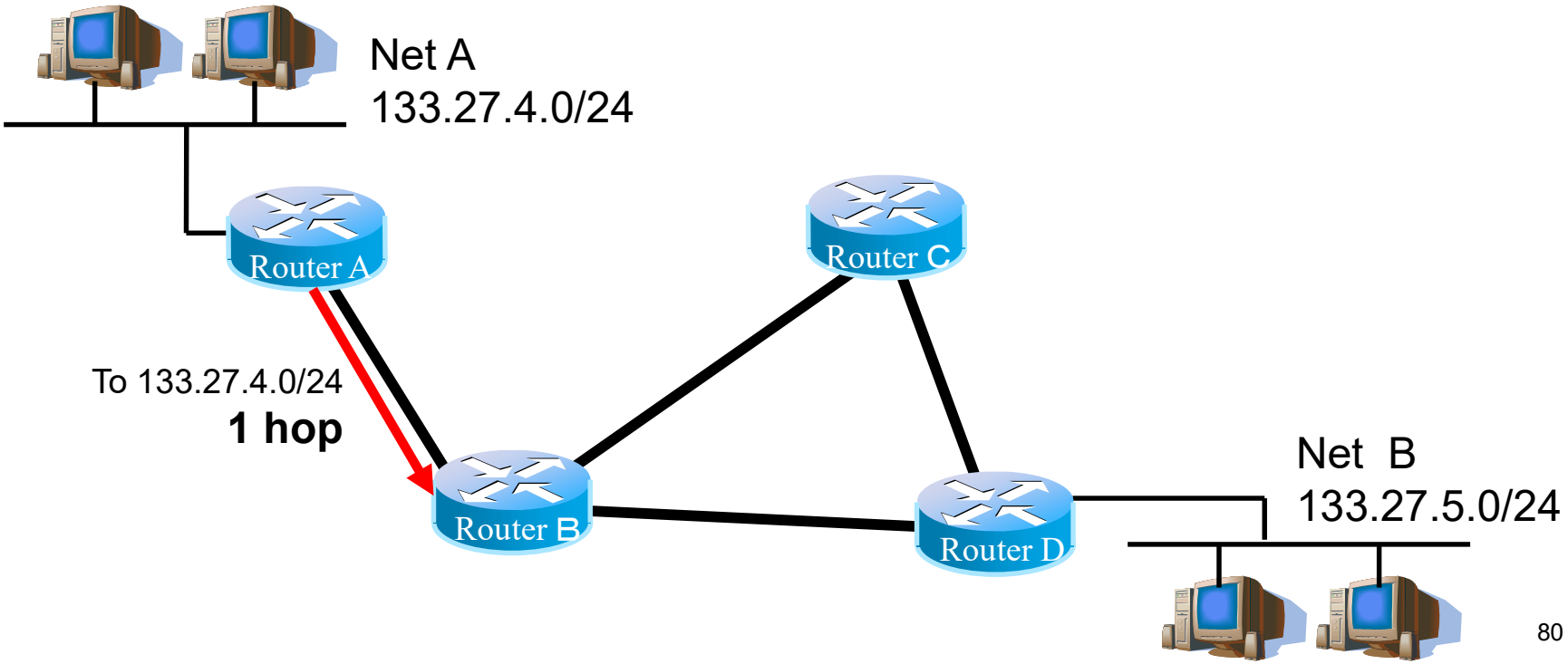
From router A to subsets:

destination	hops
u	1
v	2
w	2
x	3
y	3
z	2

Review: DV routing (1)

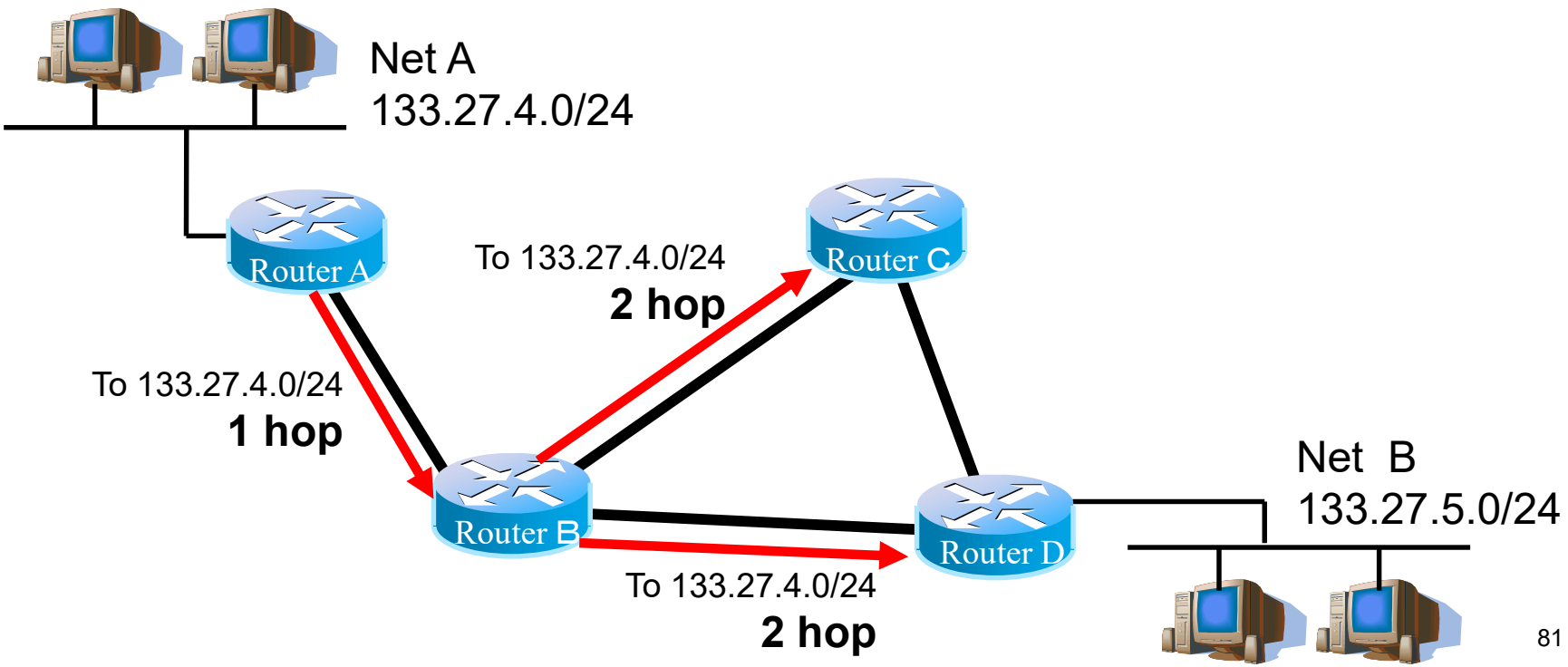


- Friend of friend is friend



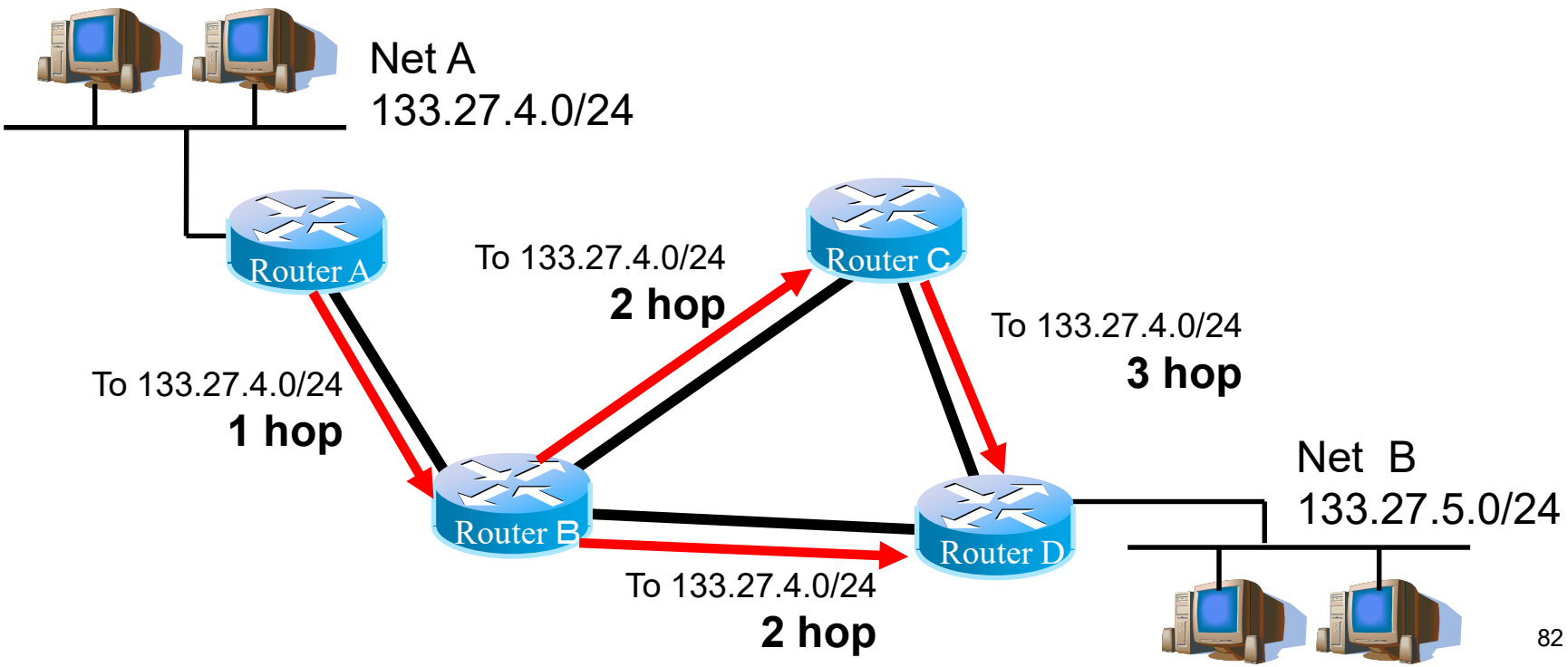
Review: DV routing (2)

- Friend of friend is friend



Review: DV routing (3)

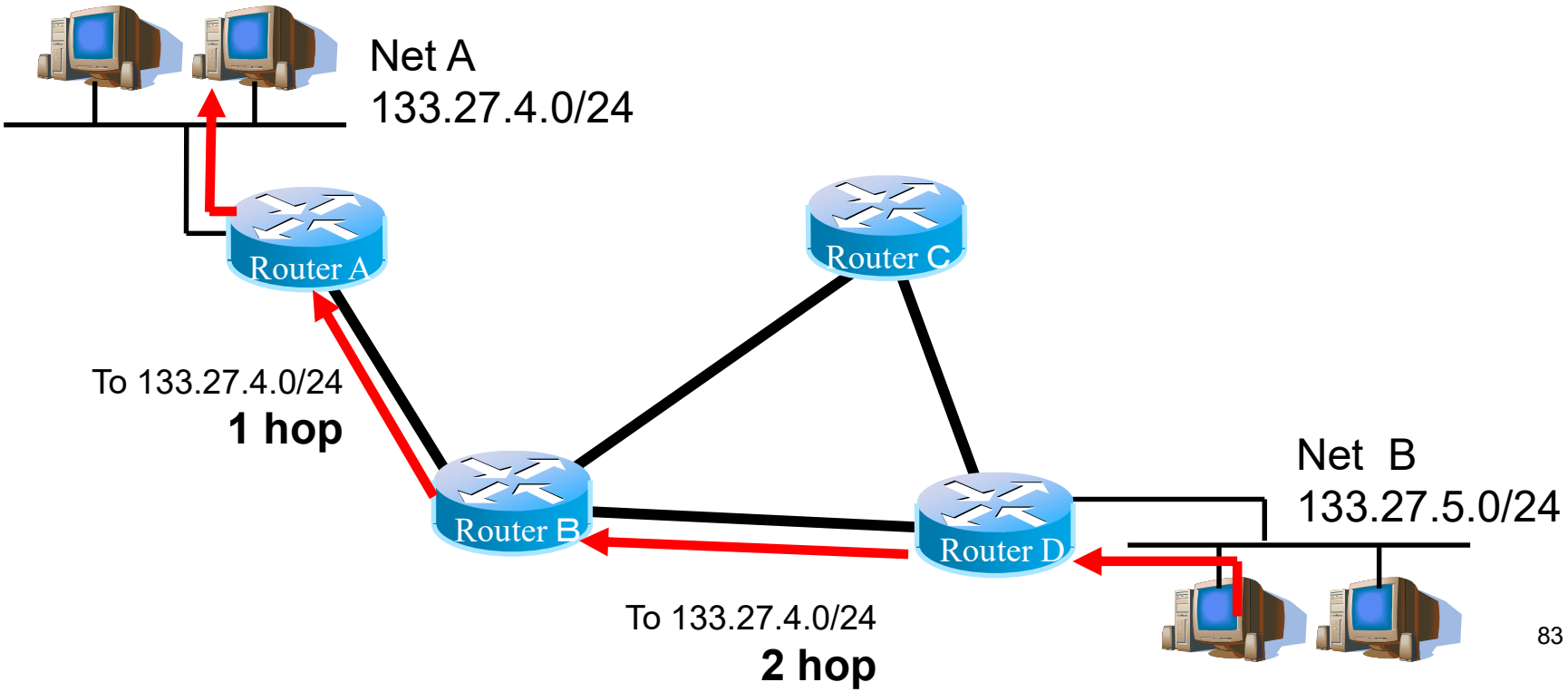
- Friend of friend is friend



s5
s6

Review: DV routing (4)

- Friend of friend is friend

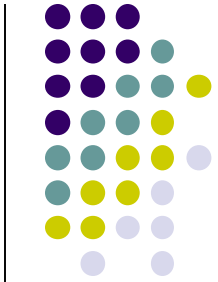


Slide 83

s5 Explain in opposite way: How B is annouced
sonnh, 8/03/2008

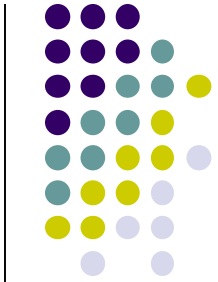
s6 Expain that we announce networks address. not router id
sonnh, 8/03/2008

RIP: Exchange information



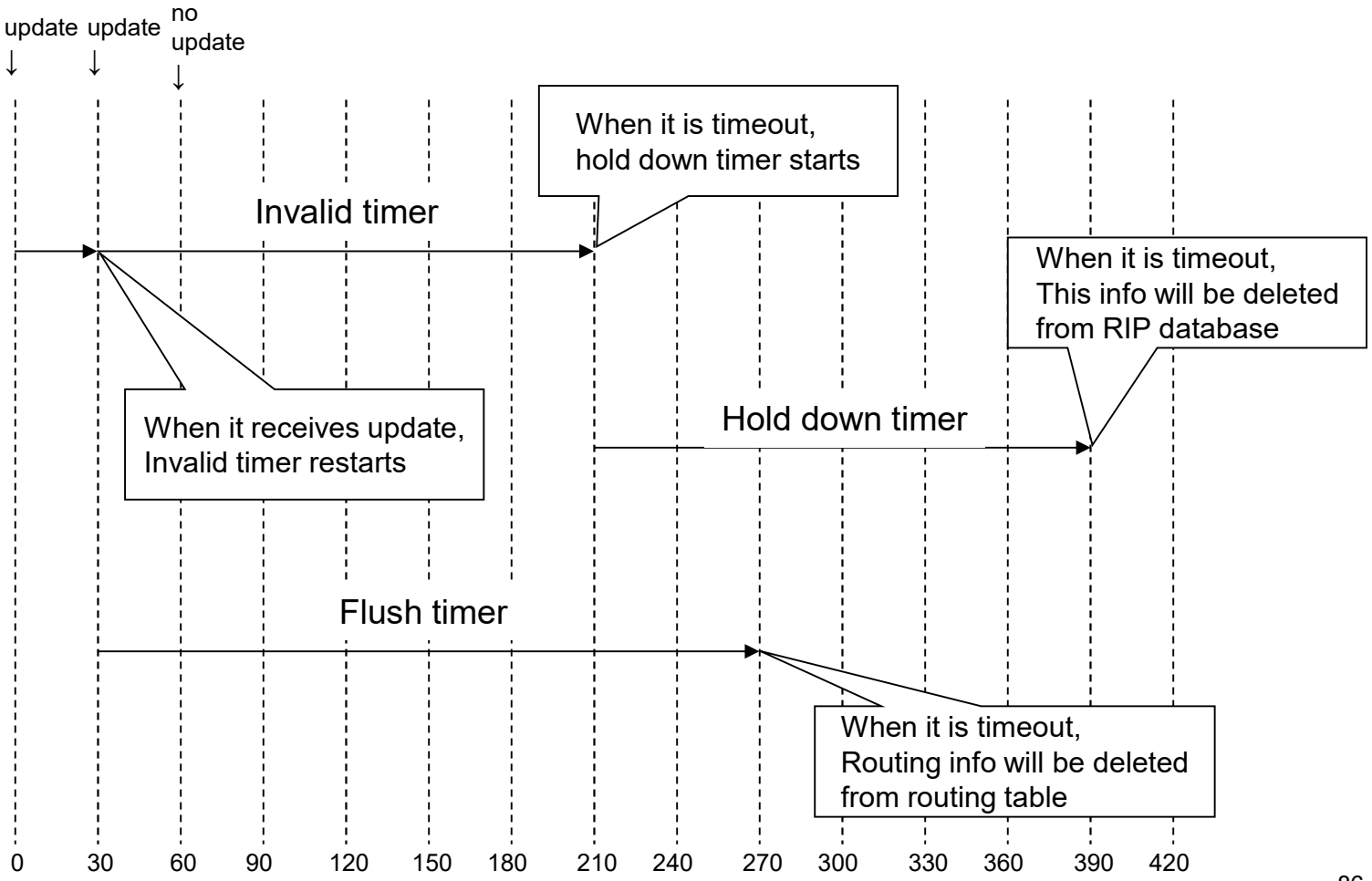
- Routing table of router is exchanged
- Periodic
  - Node advertise its distance-vector with neighbors every 30s
  - Each message contains up to 25 routing table entries
  - In practice, multiple messages are sent
- Triggered
  - When every entry changes, send copy of entry to neighbors
  - Neighbors use to update their tables

RIP timer (1)



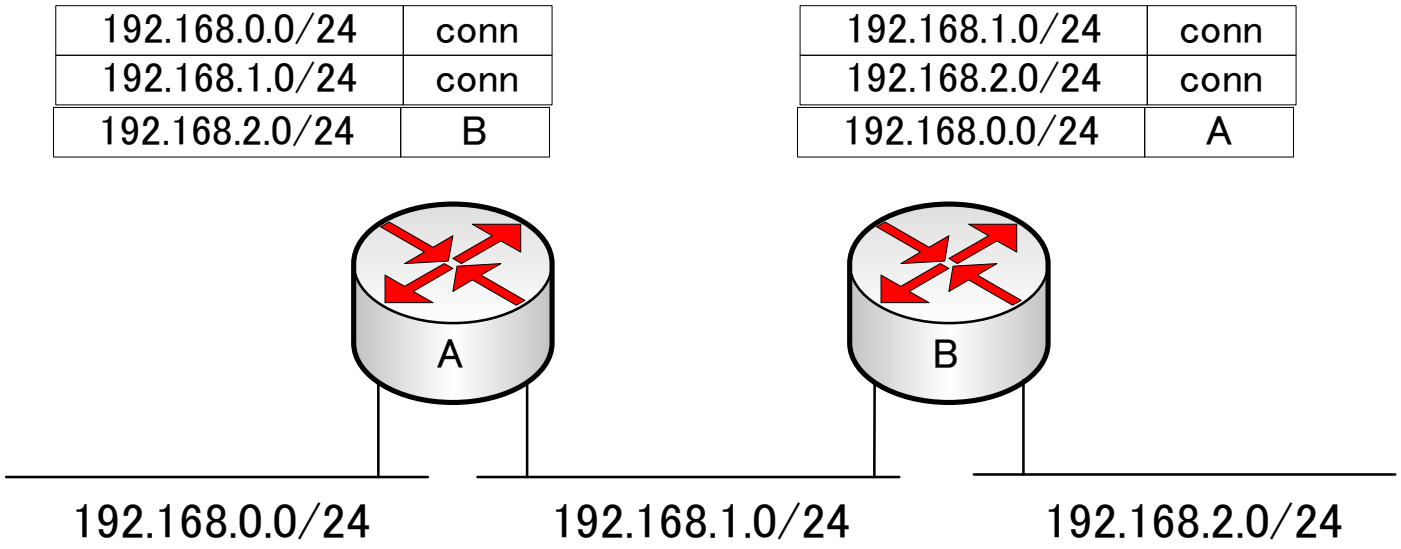
- Update timer
  - Exchange routing table every 30 sec
- Invalid timer
  - Updated every time router receives information
  - If it is time out (180sec), it becomes hold down status
- Hold down timer
  - Router keeps routing information for 180 sec
  - Not refer the worse update (to avoid the loop)
  - Possibly down status
- Flush timer
  - Update every time router receives information
  - If 240sec passed, routing entry will be deleted

RIP timer (2)

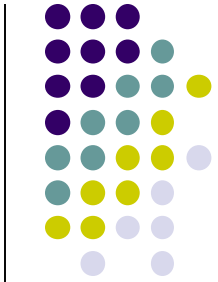

update
update
no update
When it is timeout,
hold down timer starts
Invalid timer
When it is timeout,
This info will be deleted
from RIP database
When it receives update,
Invalid timer restarts
Hold down timer
Flush timer
When it is timeout,
Routing info will be deleted
from routing table
0
30
60
90
120
150
180
210
240
270
300
330
360
390
420

Ping-pong failure

- If 192.168.0.0/24 is down…
  - B can update 192.168.0.0 info to A
  - Packets to 192.168.0.0/24 become loop status
- A will update 192.168.0.0 info to B
  - Count up to infinity!



s7



RIP: to avoid this loop

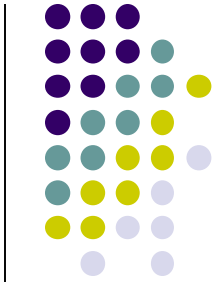
- Limit the maximum hop count
  - 16
- Split horizon
  - The routing info will not return back to the sender
- Poison reverse
  - When network is down, send update with metric 16
  - That routing information become Hold-down status

s7 16 TTL vs. this?
sonnh, 8/03/2008

OSPF (Open Shortest Path First) routing

- *IGP*
- "open": publicly available - standard by IETF (current version, version 3, defined in RFC 2740)
- classic link-state
  - each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS
  - multiple link costs metrics possible: bandwidth, delay
  - each router has full topology, uses Dijkstra's algorithm to compute forwarding table (*Shortest Path First)*
- *Advanced features*
  - *security:* all OSPF messages authenticated (to prevent malicious intrusion)
  - Large AS: Hierarchical OSPF
  - Classless routing (able to use Variable-Length Subnet Masking -VLSM)
  - Different metric for each link based on TOS (is not used in practice)

s8



Hierarchical OSPF

- Why we have to divide the network into small area?
- If we have more than 100 routers….
  - Link state update is delivered all the time
  - Number of re-calculation increase
  - Need more memory, need more CPU power
  - Number of link state update become large
  - Routing table become large
- Area
  - Group of routers which share the same LSA

Slide 91

s8 Explain why we need to reduce the calculaiton

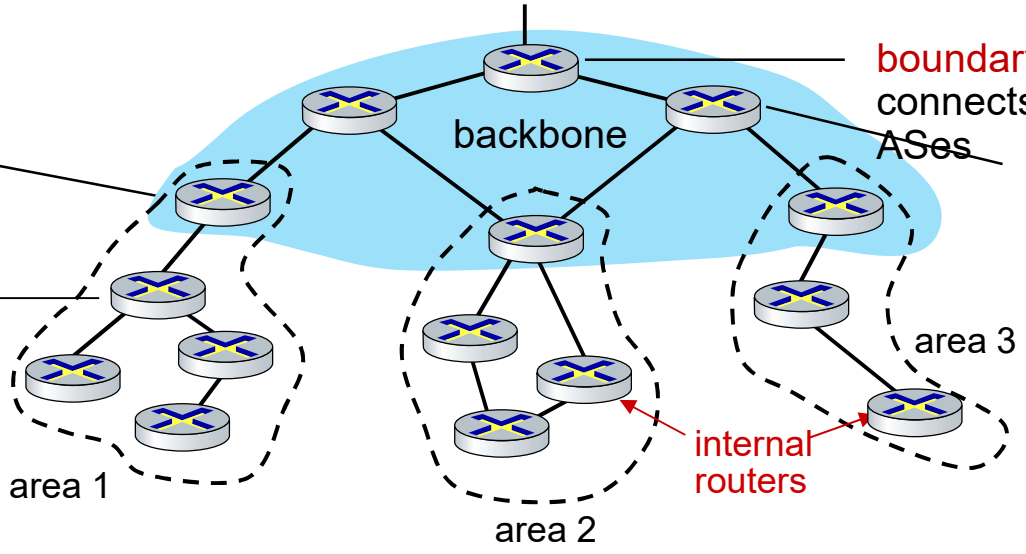
sonnh, 8/03/2008

Hierarchical OSPF

- two-level hierarchy: local area, backbone.
  - link-state advertisements flooded only in area, or backbone
  - each node has detailed area topology; only knows direction to reach other destinations

area border routers:
"summarize" distances to destinations in own area, advertise in backbone

local routers:
- flood LS in area only
- compute routing within area
- forward packets to outside via area border router

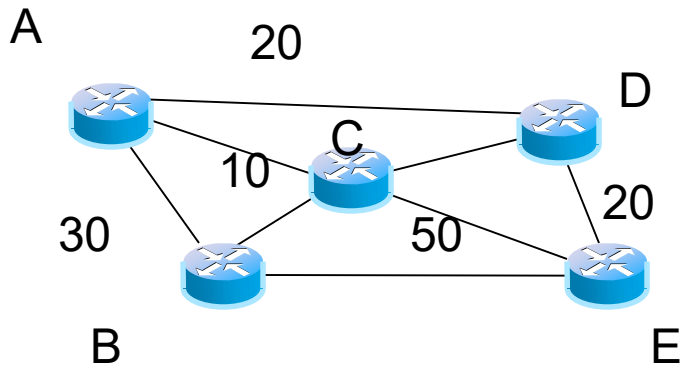


boundary router: connects to other ASes

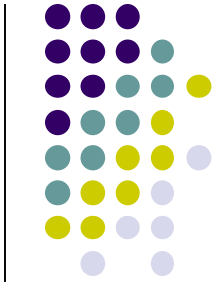
backbone router: runs OSPF limited to backbone

Which information is exchanged among routers?

- Link-State Advertisement (LSA): Contain the link and the cost to the neighbor
- For example, node A
  - link to B, cost 30
  - link to D, cost 20
  - link to C, cost 10
- For example, node D
  - link to A, cost 20
  - link to E, cost 20
  - link to C, cost 50



OSPF metric

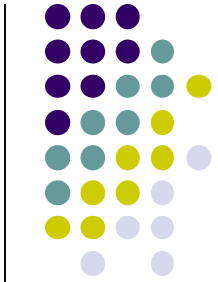


- Default value is existing

  100Mbps / bandwidth of interface

  - But these days administrator assign the original value
- During the calculation of routing table
  - Smallest cost to the one path will be selected
- If cost are same
  - Router will do load balancing

OSPF default cost

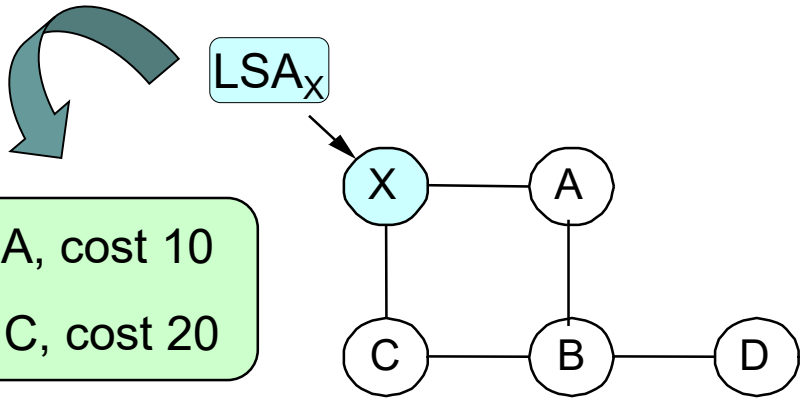


Link Bandwidth	Default OSPF cost
56Kbps serial link	1785
64Kbps serial link	1562
T1 (1.544Mbps) serial link	65
E1 (2.048Mbps) serial link	48
4Mbps Token Ring	25
Ethernet	10
16Mbps Token Ring	6
FDDI or Fast Ethernet	1
Gigabit Ethernet / 10G network	1

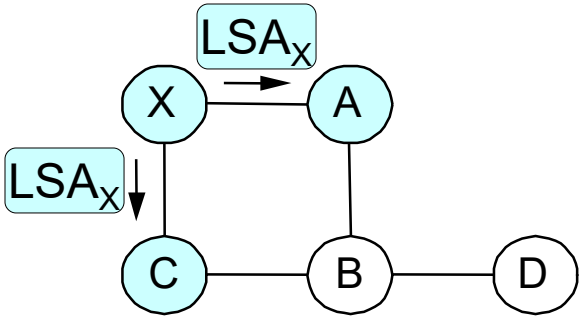
Link state flooding

X has link to A, cost 10

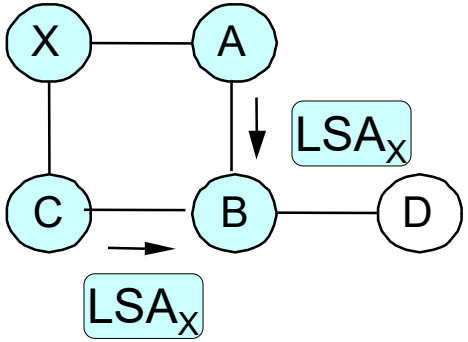
X has link to C, cost 20



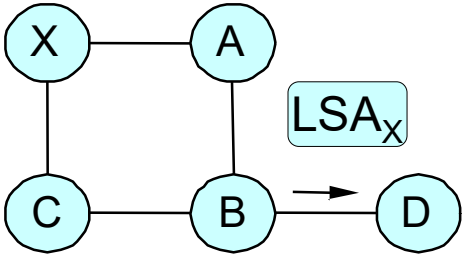
(a)



(b)



(c)



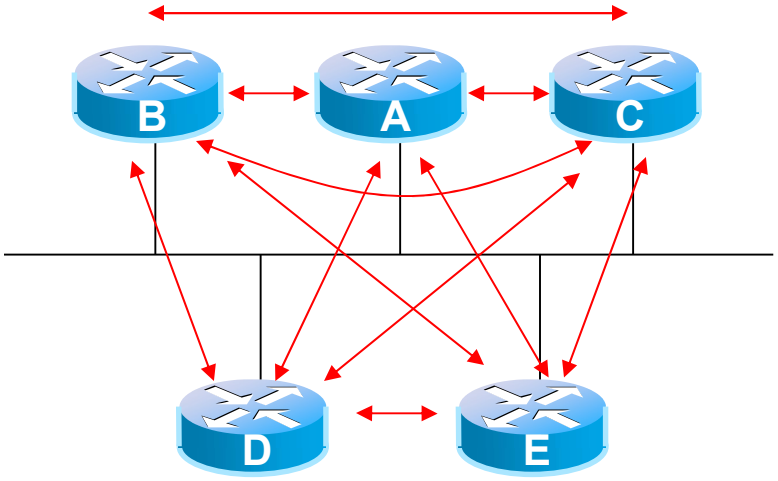
(d)

s9 What information is flood

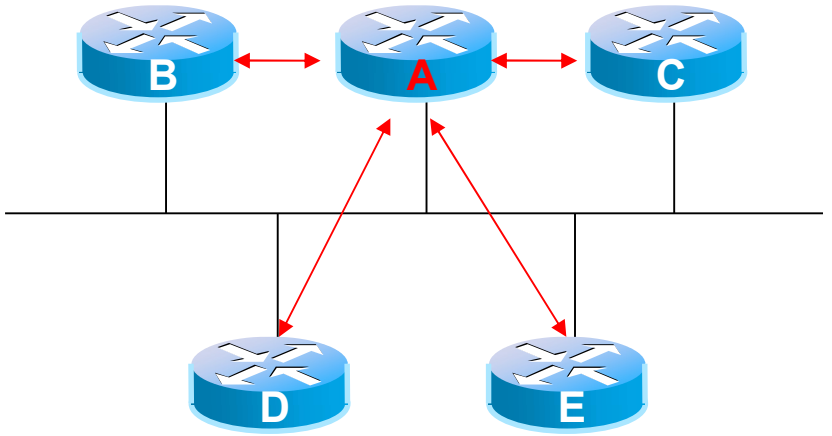
sonnh, 8/03/2008

Designated router (DR)

- To improving efficiency on exchanging link state
- Each router forms an adjacency with the designated router (DR)
  - Exchanging link state through DR
  - If DR fails, use a BDR (Backup DR)
- How to select DR and BDR?



Do not use DR



Use DR

s10

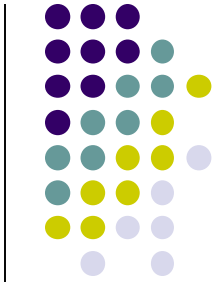
Neighbor & Adjacency

- Neighbor and adjacency are different concept!
  - Adjacency: router which exchange the routing information each other
  - Neighbor: routers have a direct link
- Broadcast multi-access network (e.g Ethernet)
  - Neighbor != Adjacency
- Point-to-point network
  - Neighbor == Adjacency
- Non Broadcast Multi-access network (e.g. ATM)
  - Exchanging data using unicast

s10 Chang the order

sonnh, 8/03/2008

s11



RIP vs. OSPF

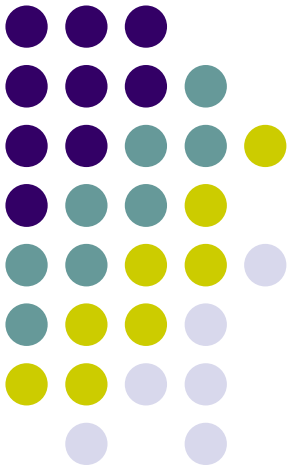
	RIP	OSPF
Characteristics	• Flat relation between routers • Implementation is simple • Small-scale network	• Support hierarchy • Implementation is complicated • Middle and large-scale network
Scalability	x	o
Computational complexity	little	many
Convergence time	Low speed	high speed
Exchanged information	Routing table	Link-state information
Algorithm	Distant vector type	Link-state type
Neighbor discovery	30s	10s (Hello packet)
Metric	number of hops	Cost (band width)

s11

Exchanged informatio
Rip:
OSPF

sonnh, 8/03/2008

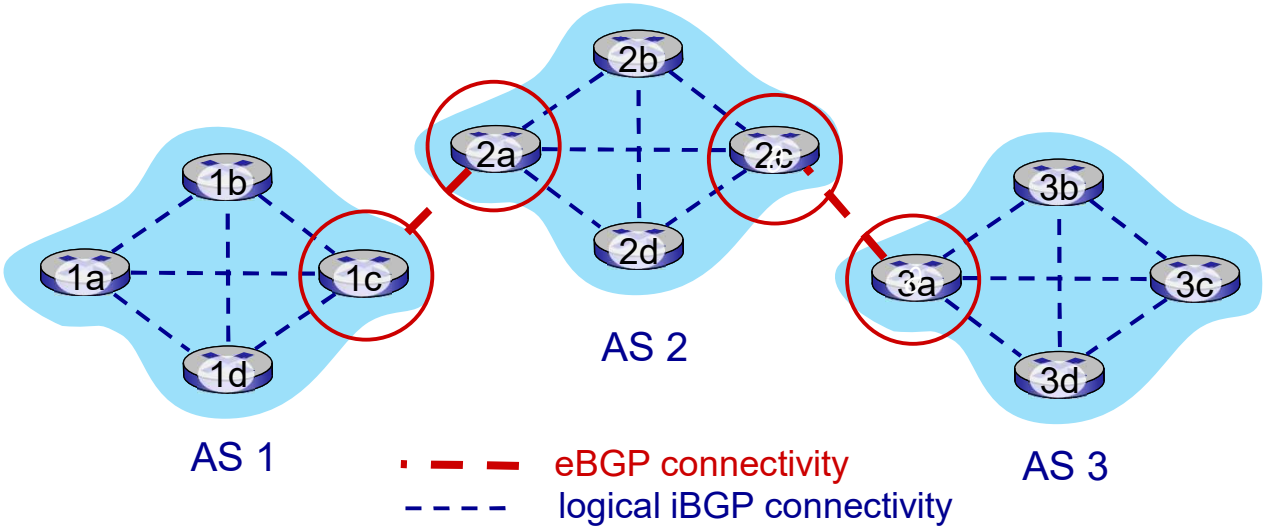
Inter-domain routing protocol



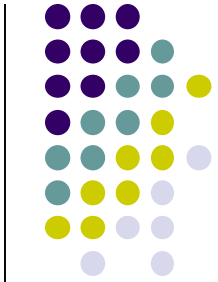
Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- allows subnet to advertise its existence, and the destinations it can reach, to rest of Internet: *"I am here, here is who I can reach, and how"*
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASes
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and *policy*

eBGP, iBGP connections

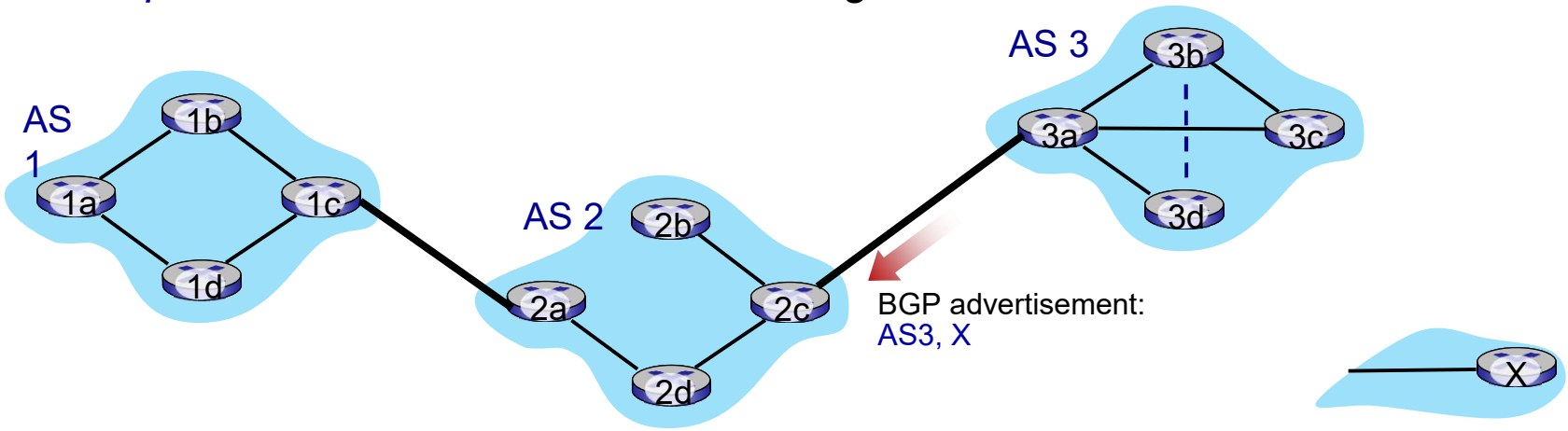


gateway routers run both eBGP and iBGP protocols



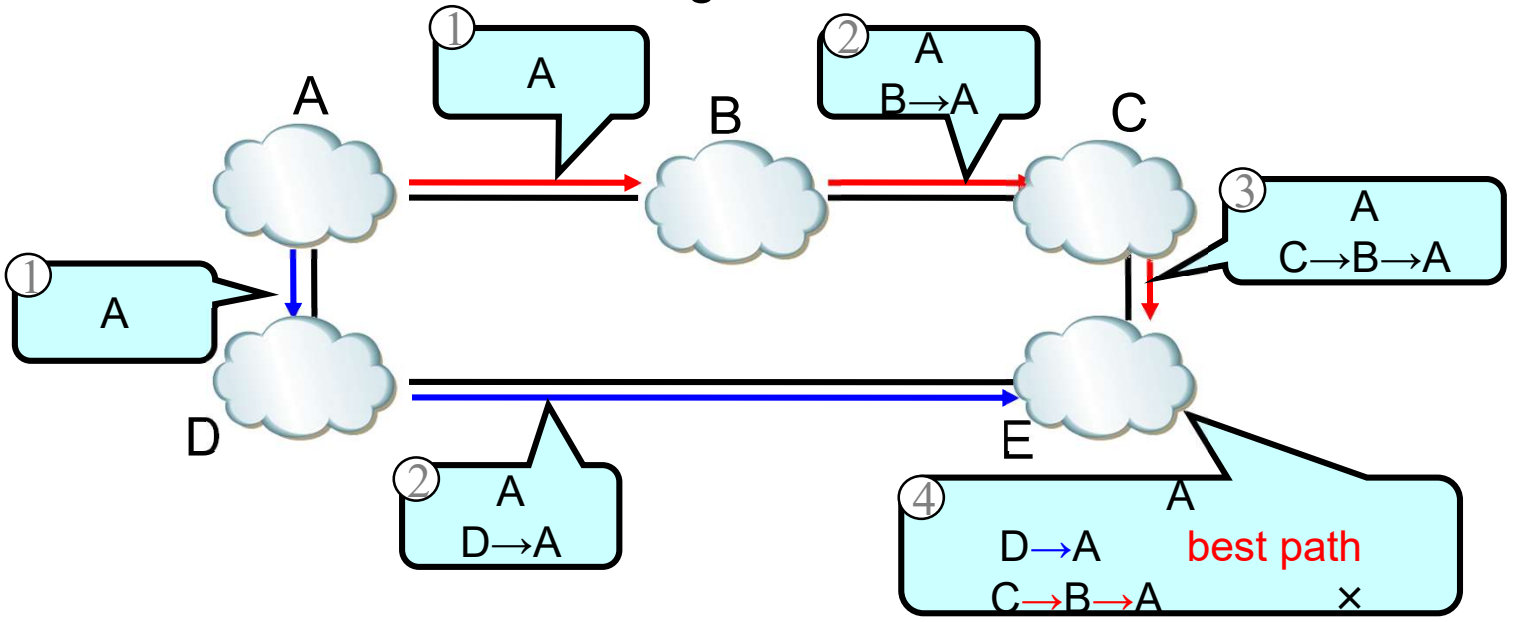
BGP basics

- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a "path vector" protocol)
- when AS3 gateway 3a advertises path AS3,X to AS2 gateway 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X



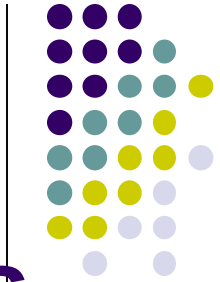
BGP: Path vector routing

- Which routing protocol can be used to connect multiple ASes?
  - No universal metric – policy decisions
  - LS: No, Metric are not the same, LS database too large – entire Internet
  - DV: Bellman-Ford algorithm may not converge
- Solution: Path vector routing



s12 Change the symbol of router into AS

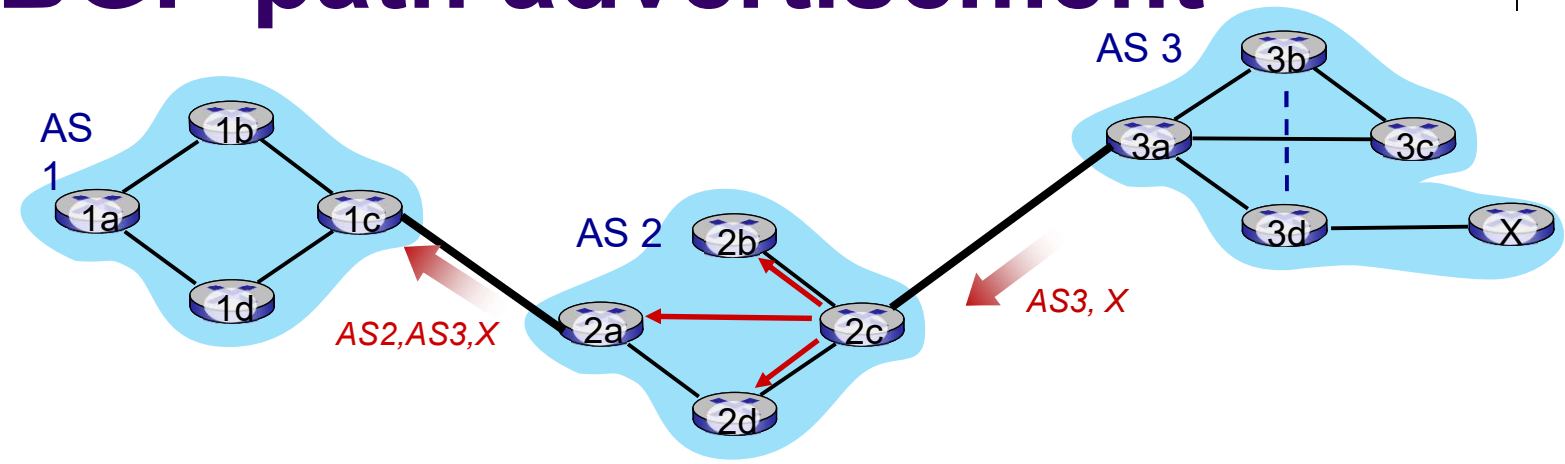
sonnh, 29/02/2008



Path attributes and BGP routes

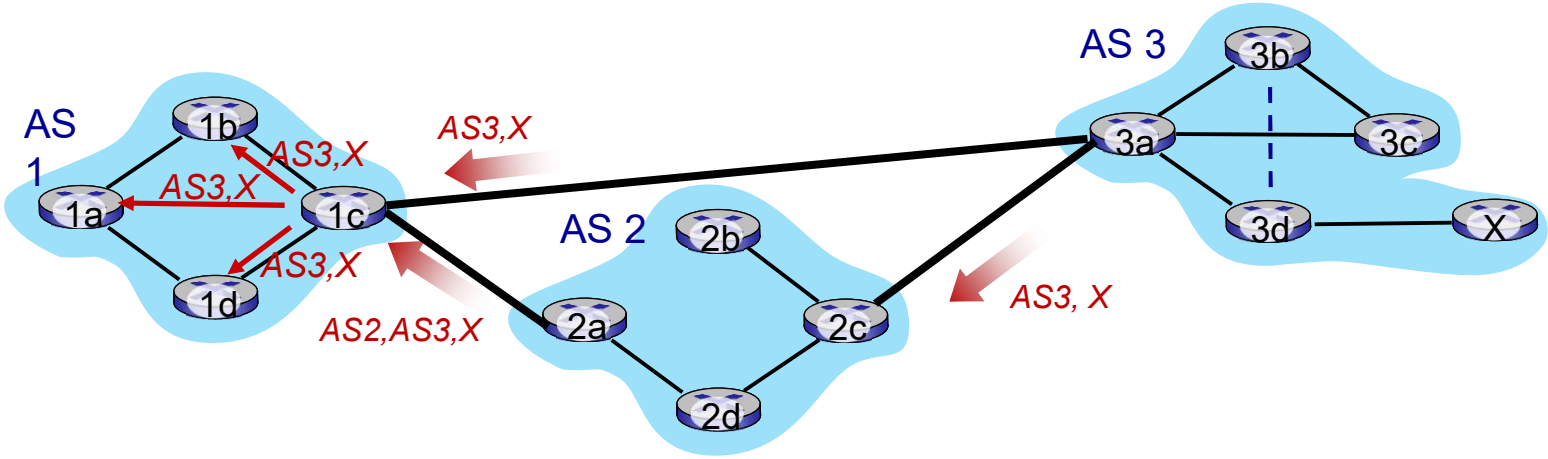
- BGP advertised route: prefix + attributes
  - prefix: destination being advertised
  - two important attributes:
  - AS-PATH: list of ASes through which prefix advertisement has passed
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- policy-based routing:
  - gateway receiving route advertisement uses *import policy* to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to *advertise* path to other other neighboring ASes

BGP path advertisement



- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c

BGP path advertisement (more)

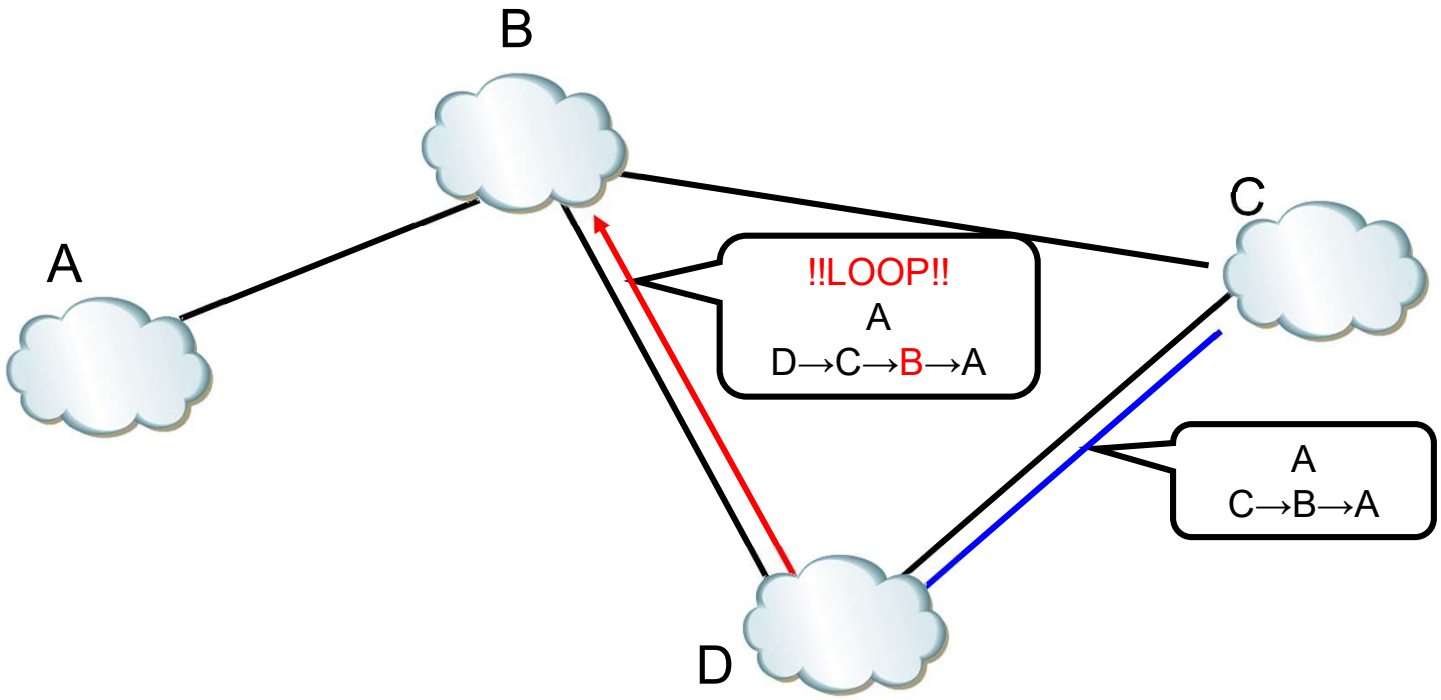


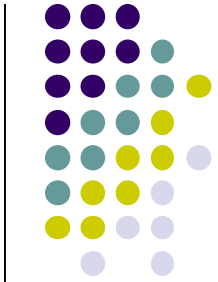
gateway router may learn about multiple paths to destination:

- AS1 gateway router 1c learns path *AS2,AS3,X* from 2a
- AS1 gateway router 1c learns path *AS3,X* from 3a
- based on *policy,* AS1 gateway router 1c chooses path *AS3,X* and advertises path within AS1 via iBGP

Loop free mechanism

- Detecting loop depending on whether that router is included in path of received routing information or not
  - B will cancel the route to A, which B includes in path

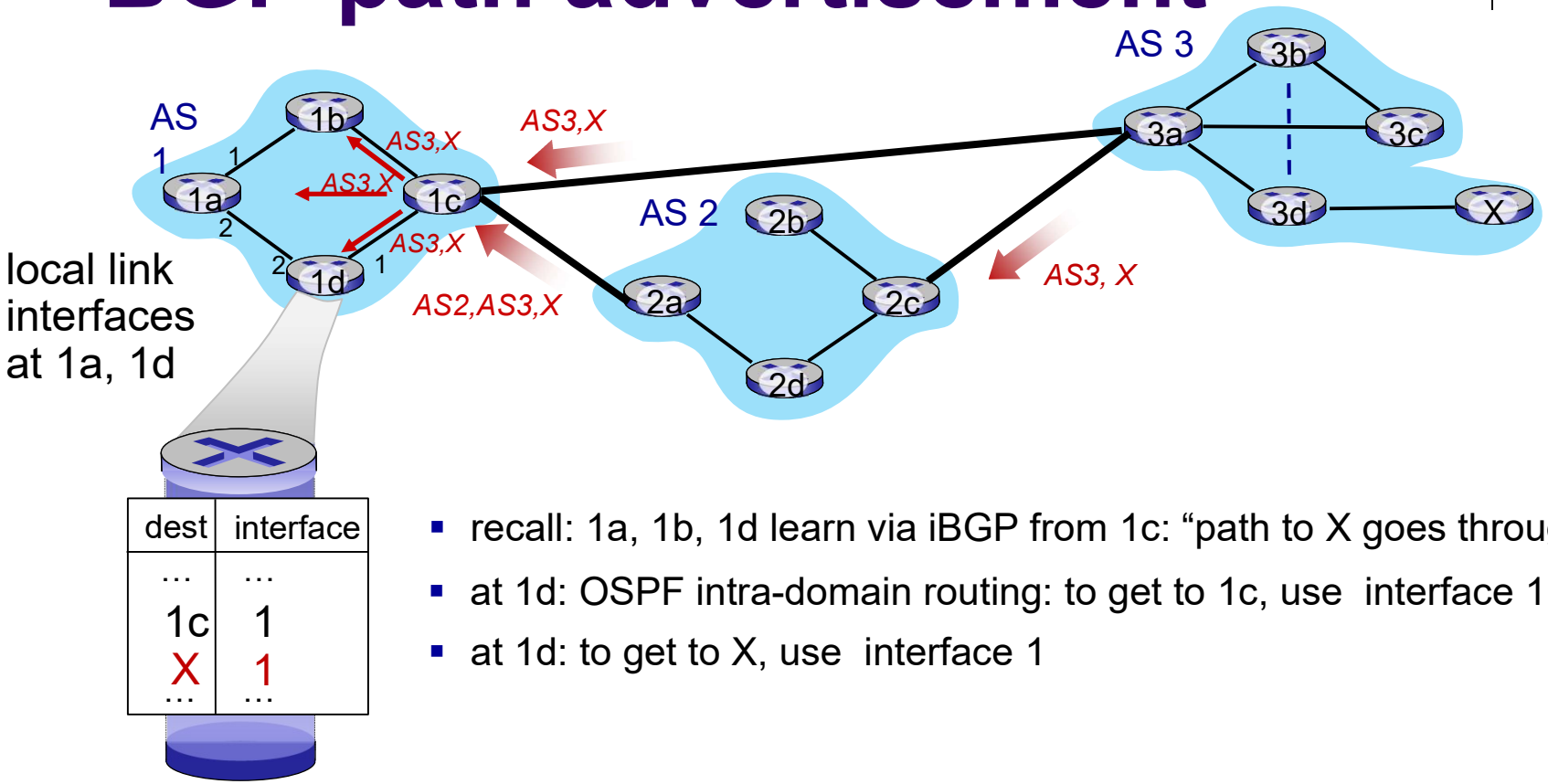




BGP messages

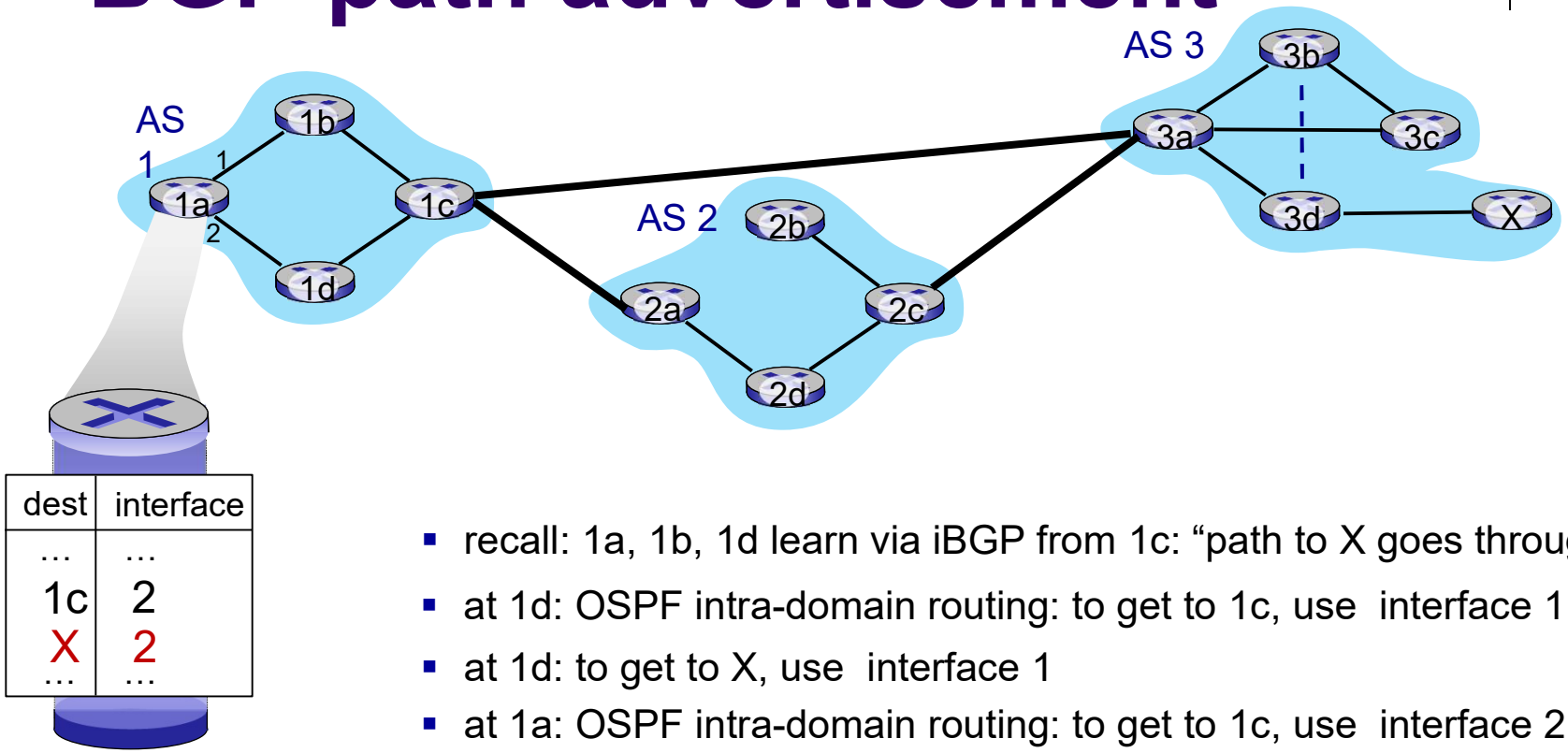
- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - OPEN: opens TCP connection to remote BGP peer and authenticates sending BGP peer
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP path advertisement

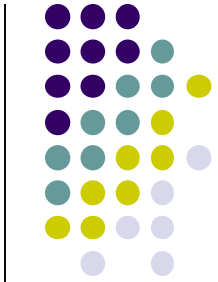


- recall: 1a, 1b, 1d learn via iBGP from 1c: “path to X goes through 1c”
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1

BGP path advertisement



- recall: 1a, 1b, 1d learn via iBGP from 1c: “path to X goes through 1c”
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1
- at 1a: OSPF intra-domain routing: to get to 1c, use interface 2
- at 1a: to get to X, use interface 2



Why different Intra-, Inter-AS routing ?

policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its network
- intra-AS: single admin, so policy less of an issue

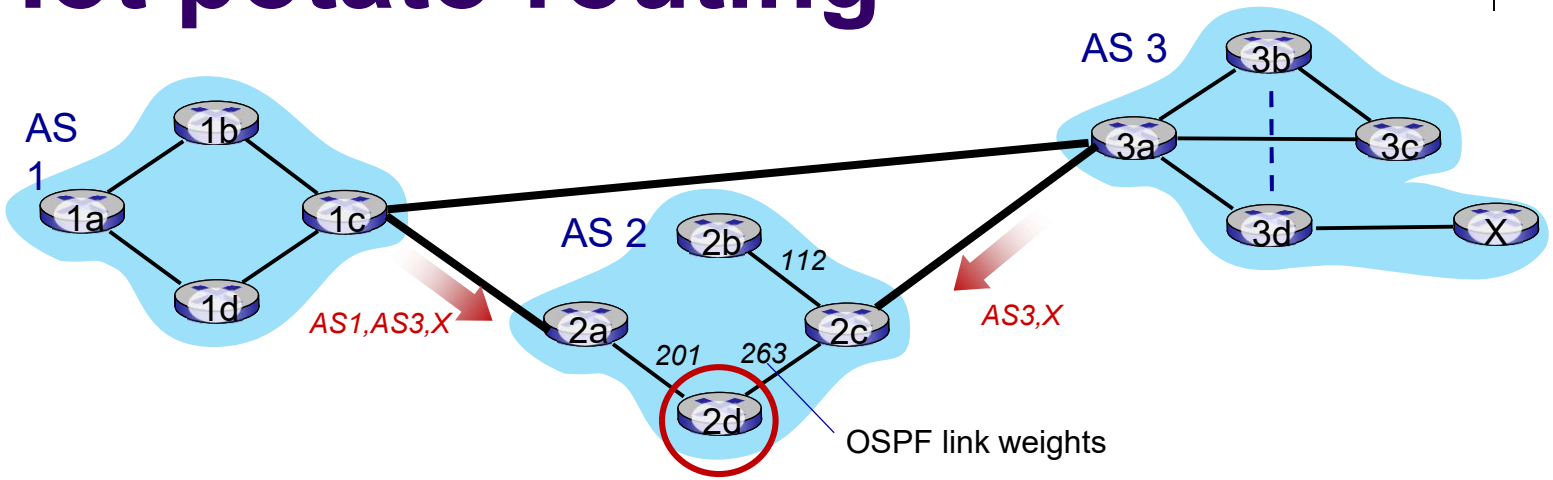
scale:

- hierarchical routing saves table size, reduced update traffic

performance:

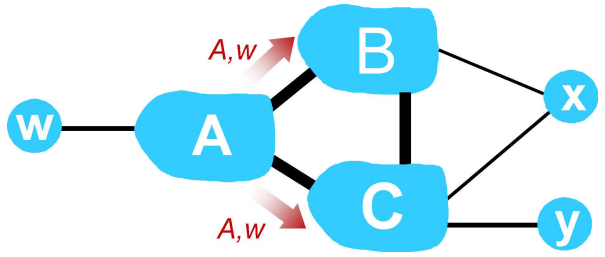
- intra-AS: can focus on performance
- inter-AS: policy dominates over performance

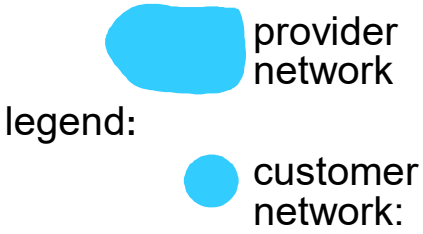
Hot potato routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- hot potato routing: choose local gateway that has least *intra-domain* cost (e.g., 2d chooses 2a, even though more AS hops to *X*): don't worry about inter-domain cost!

BGP: achieving policy via advertisements

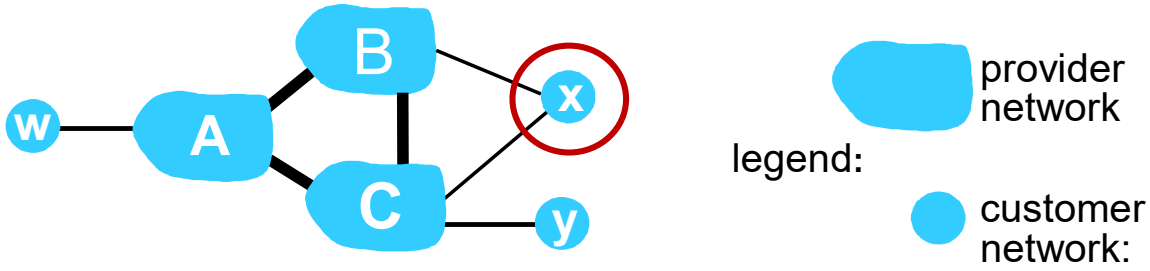




ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical "real world" policy)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C!
  - B gets no "revenue" for routing CBAw, since none of C, A, w are B's customers
  - C does *not* learn about CBAw path
- C will route CAw (not using B) to get to w

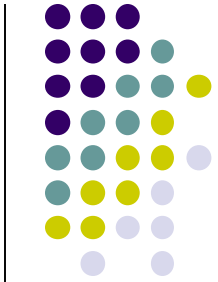
BGP: achieving policy via advertisements (more)



ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical "real world" policy)

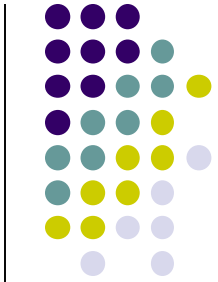
- A,B,C are provider networks
- x,w,y are customer (of provider networks)
- x is dual-homed: attached to two networks
- *policy to enforce:* x does not want to route from B to C via x
  - .. so x will not advertise to B a route to C

BGP route selection



- router may learn about more than one route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. additional criteria

Các thuộc tính của đường đi



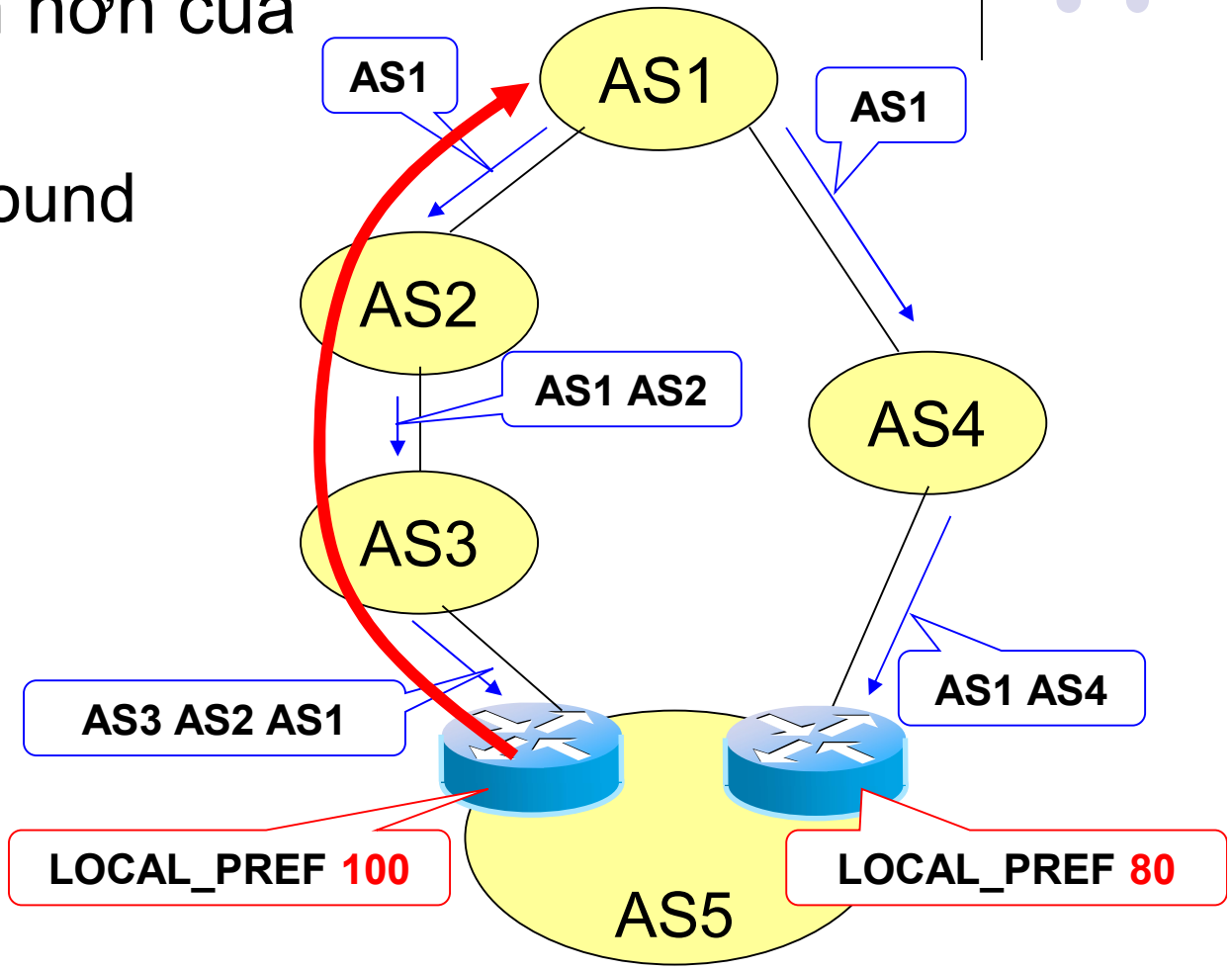
- ORIGIN
  - Source of the information (IGP/EGP/incomplete)
- AS_PATH
- NEXT_HOP
- MED (MULTI_EXIT_DISCRIMINATOR)
- LOCAL_PREF
- ATOMIC_AGGREGATE
- AGGREGATOR
- COMMUNITY

Steps to select the path

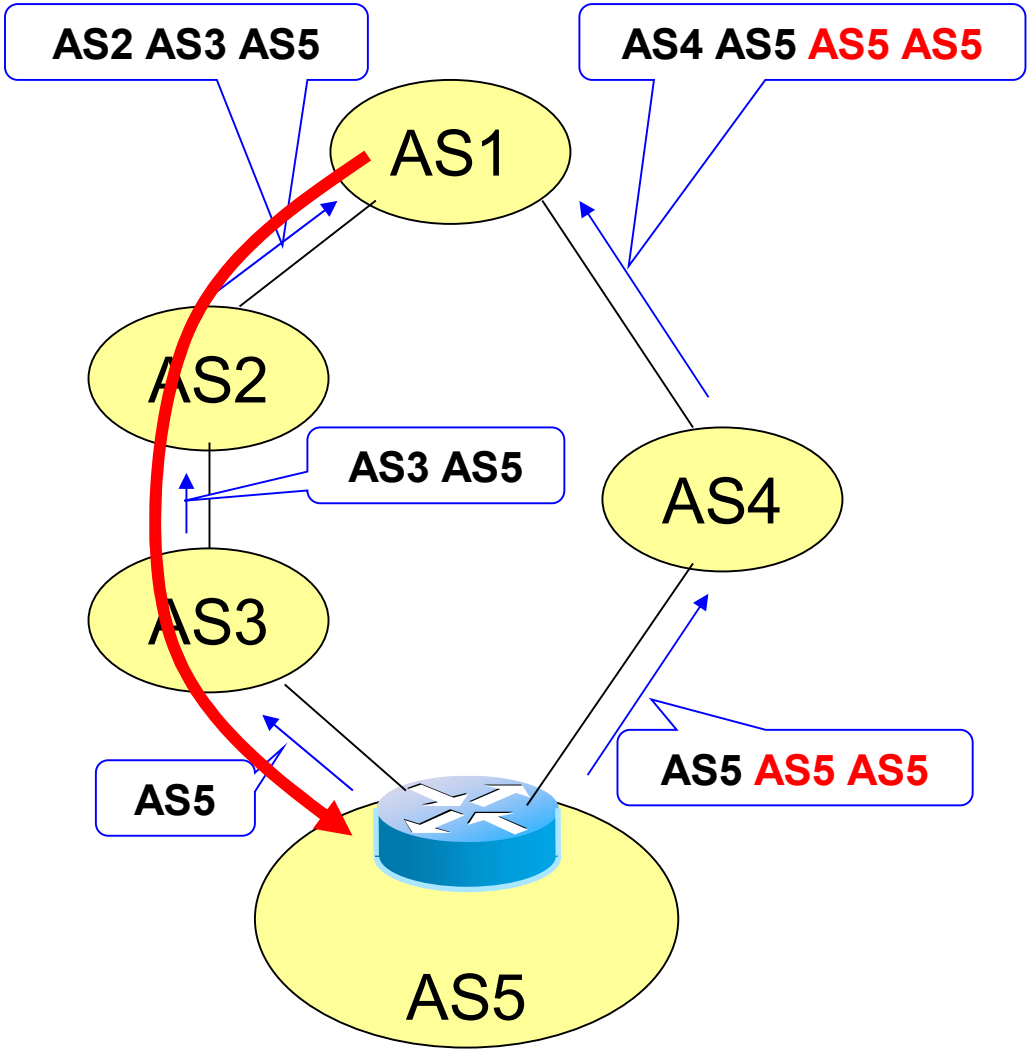
- Step 1: Compare LOCAL_PREF
- Step 2: Compare AS_PATH
- Step 3: Compare ORIGIN
- Step 4: Compare MED
- Step 5: Compare EBGP/IBGP
- Step 6: Compare cost to NEXT_HOP
- Step 7: Compare Router ID

Sử dụng LOCAL_PREF

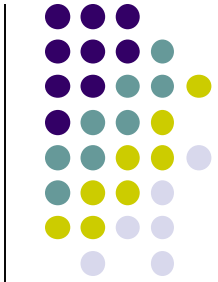
- Chọn giá trị lớn hơn của LOCAL_PREF
- Control the upbound bandwidth



Routing with AS_PATH Prepend

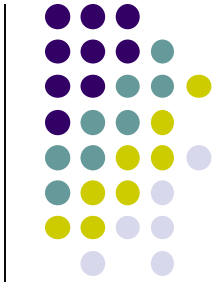


Example of AS PATH



Network	Next Hop	Metric	LocPrf	Weight	Path
4.79.201.0/26	203.178.136.29	700	500	0	7660 22388 11537 10886 40220
	203.178.136.29	700	500	0	7660 22388 11537 10886 40220
	203.178.136.29	700	500	0	7660 22388 11537 10886 40220
6.1.0.0/16	203.178.136.29	700	500	0	7660 22388 11537 668
	203.178.136.29	700	500	0	7660 22388 11537 668
	203.178.136.29	700	500	0	7660 22388 11537 668
6.2.0.0/22	203.178.136.29	700	500	0	7660 22388 11537 668

Example of AS PATH prepend

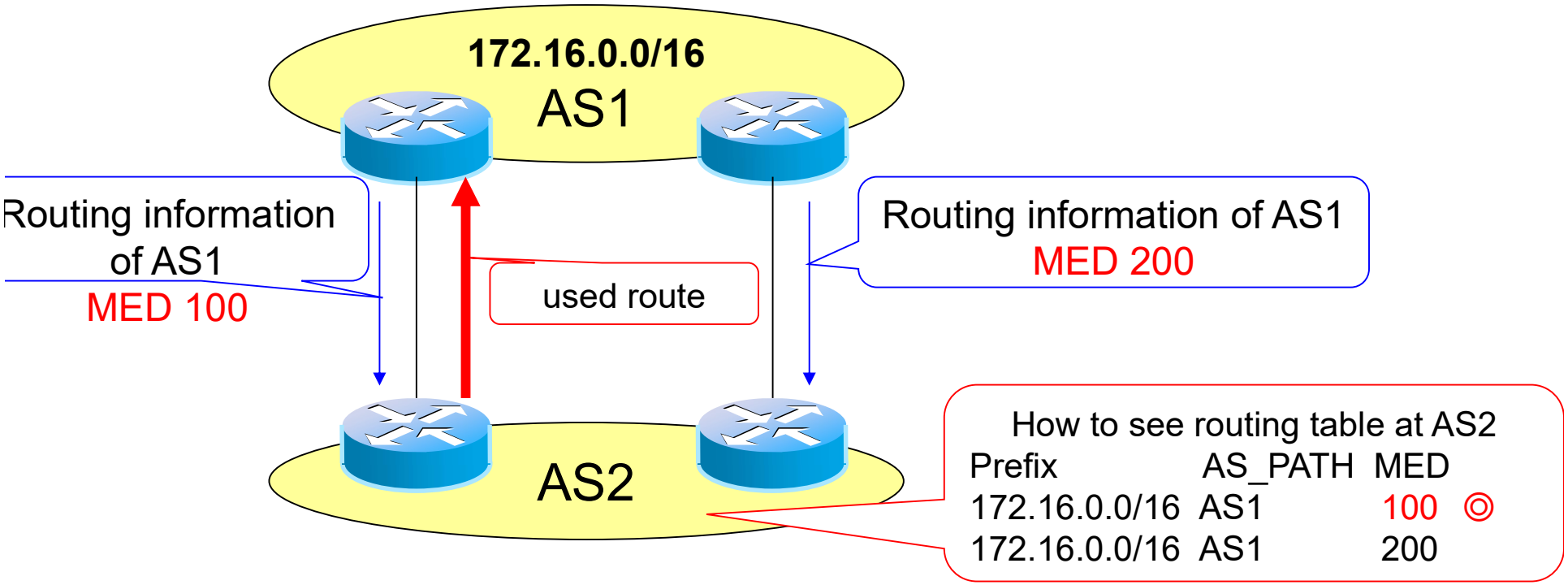


Network	Next Hop	Metric	LocPrf	Weight	Path
8.5.192.0/22	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
8.5.196.0/24	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
8.5.200.0/22	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989

Some AS are repeated on the path to make it longer and not being selected

Routing with MED

- In case of 2 AS with many links
- Choose smaller MED
- Apply in controlling bandwidth



Load balancing with MED

- Set MED for different paths
- Also control the bandwidth