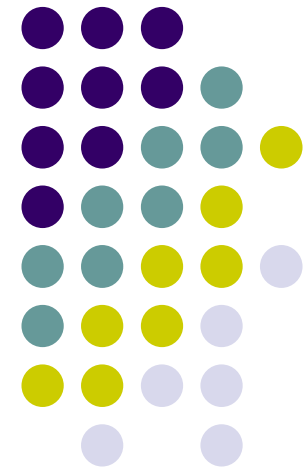
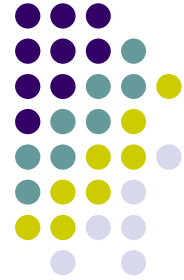


# Chương 4: Chọn đường - Routing

Dự án HEDSPI  
Khoa CNTT- ĐHBK Hà Nội

Giảng viên: Ngô Hồng Sơn  
Bộ môn Truyền thông và Mạng máy tính





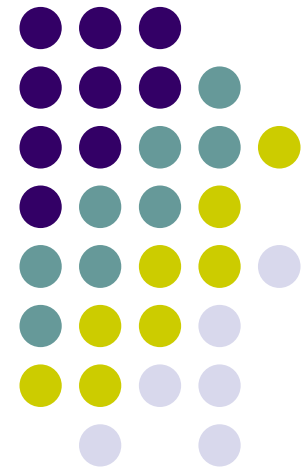
# Tổng quan

- Tuần trước
  - Giao thức IP
  - Địa chỉ IP và cấu trúc gói tin IP
  - Giao thức ICMP
- Tuần này: Tiếp tục về tầng mạng
  - Thế nào là chọn đường?
  - Chọn đường tĩnh và chọn đường động
  - Giải thuật và giao thức chọn đường

# Chọn đường là gì?

---

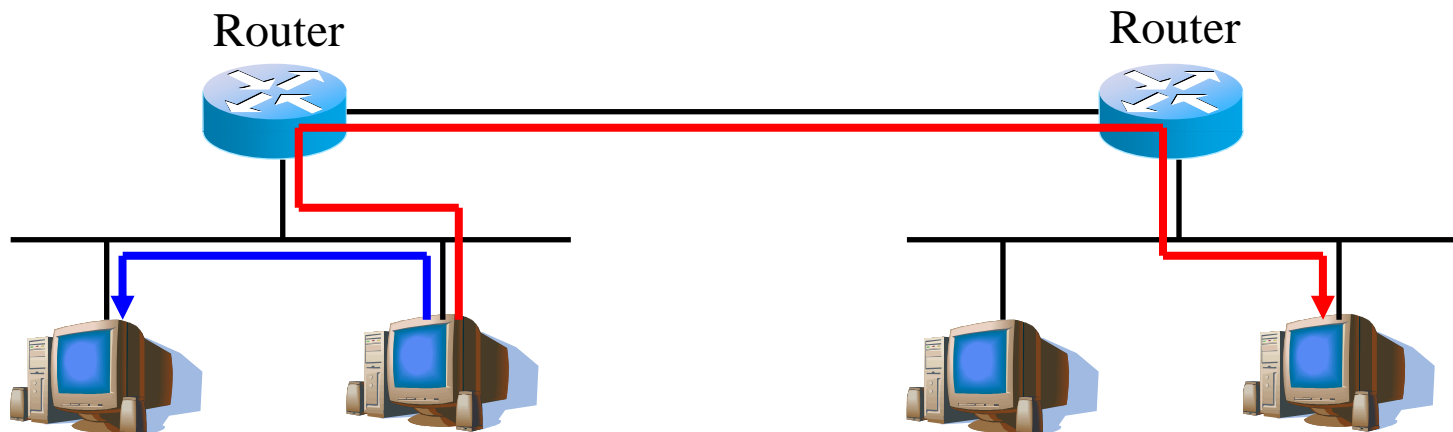
Các nguyên lý chọn đường  
Cơ chế chuyển tiếp gói tin  
Quy tắc “Longest matching”



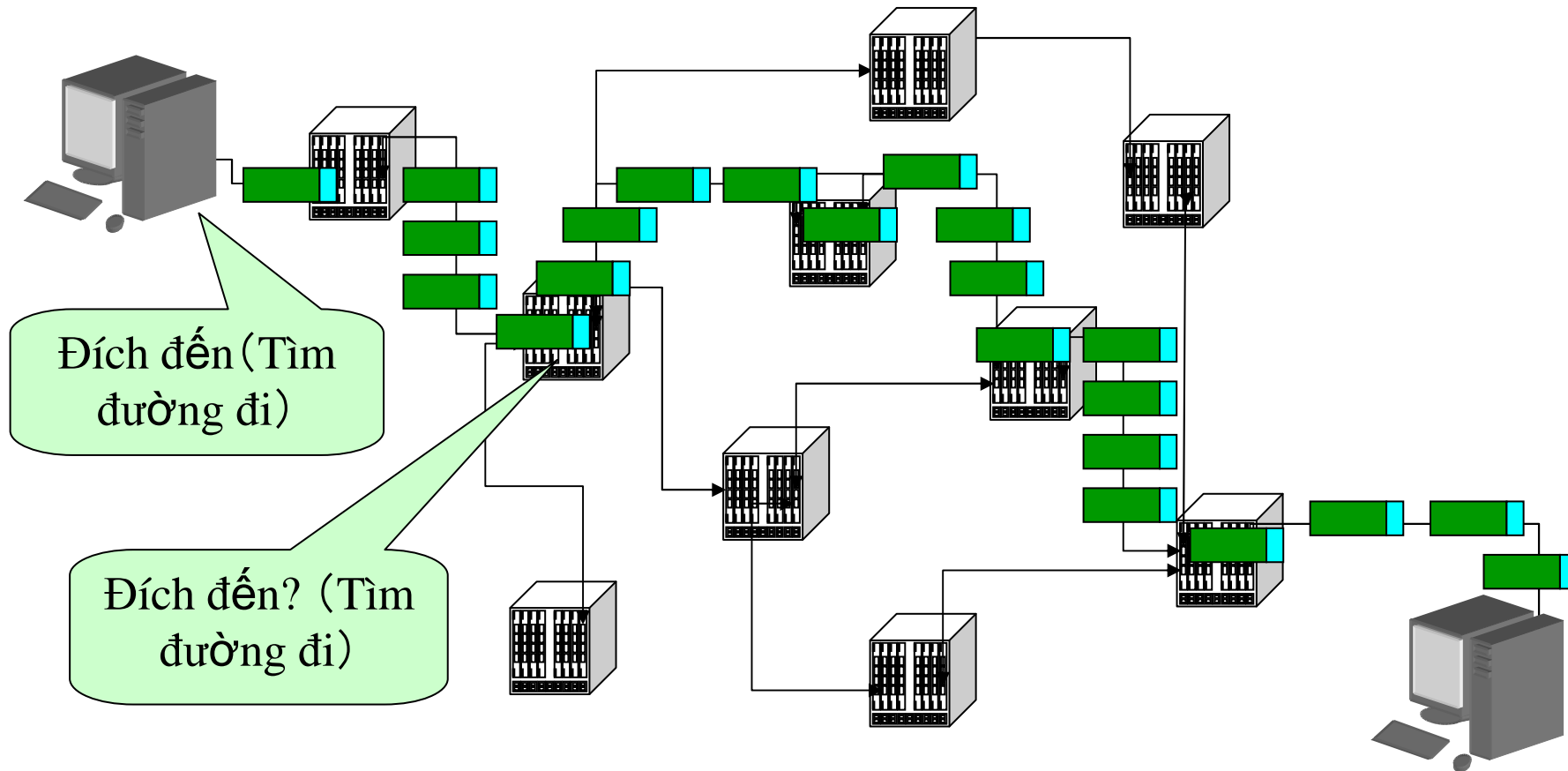
# Cơ bản về chọn đường (1)



- Khi một máy trạm gửi một gói tin IP tới một máy khác
  - Nếu địa chỉ đích nằm trên cùng một đường truyền vật lý: Chuyển trực tiếp
  - Nếu địa chỉ đích nằm trên một mạng khác: Chuyển gián tiếp qua bộ định tuyến (chọn đường)



# Cơ bản về chọn đường (2)





# Chọn đường là gì?

- Cơ chế để máy trạm hay bộ định tuyến chuyển tiếp gói tin từ nguồn đến đích
- Các thành phần của chọn đường
  - Bảng chọn đường
  - Thông tin chọn đường
  - Giải thuật, giao thức chọn đường



# Bộ định tuyến?

- Thiết bị chuyển tiếp các gói tin giữa các mạng
  - Là một máy tính, với các phần cứng chuyên dụng
  - Kết nối nhiều mạng với nhau
  - Chuyển tiếp gói tin dựa trên bảng chọn đường
- Có nhiều giao diện
- Phù hợp với nhiều dạng lưu lượng và phạm vi của mạng

# Một số ví dụ...



BUFFALO  
BHR-4RV



PLANEX  
GW-AP54SAG



YAMAHA  
RTX-1500



Cisco 2600

## Router ngoại vi



Cisco CRS-1

## Router mạng trực



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>





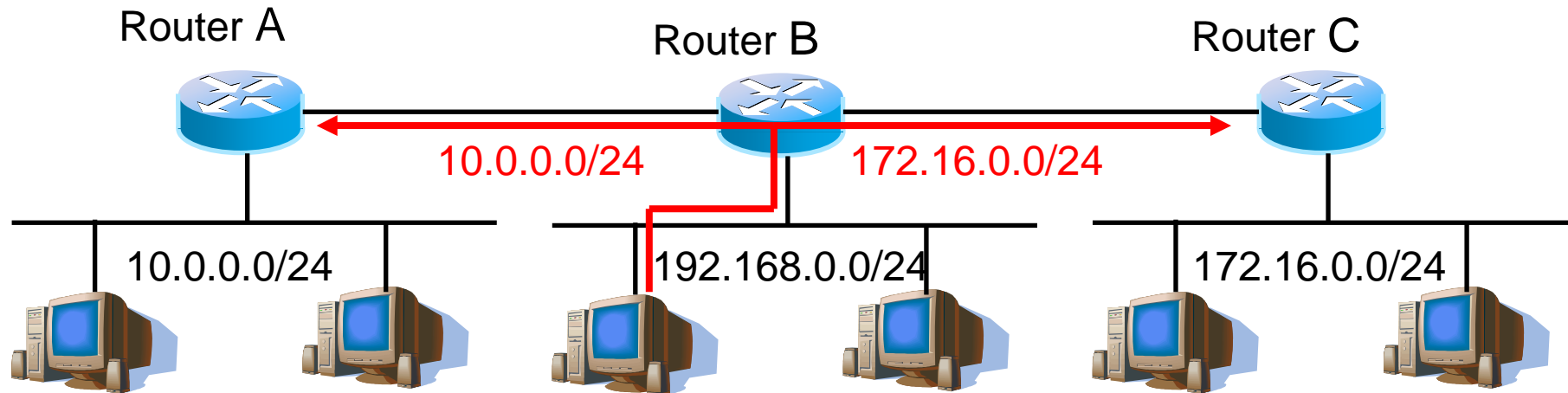
# Bảng chọn đường

- Chỉ ra danh sách các đường đi có thể, được lưu trong bộ nhớ của router
- Các thành phần chính của bảng chọn đường
  - Địa chỉ đích/mặt nạ mạng
  - Router kế tiếp

# Bảng chọn đường và cơ chế chuyển tiếp (1)

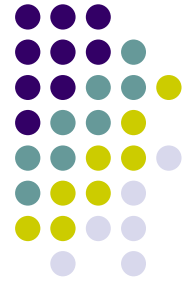


Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C



Lưu ý quy tắc: **No routes, no reachability!**

# Quy tắc “Longest matching”(1)



- Giả sử một địa chỉ mạng đích lại có nhiều hơn một mục trong bảng chọn đường
- Địa chỉ đích : 11.1.2.5
- Router kế tiếp nào sẽ được sử dụng?

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

# Quy tắc “Longest matching”(2)



Địa chỉ đích:

11.1.2.5 = 00001011.00000001.00000010.00000101

Đường đi 1:

11.1.2.0/24 = 00001011.00000001.00000010.00000000

Đường đi 2:

11.1.0.0/16 = 00001011.00000001.00000000.00000000

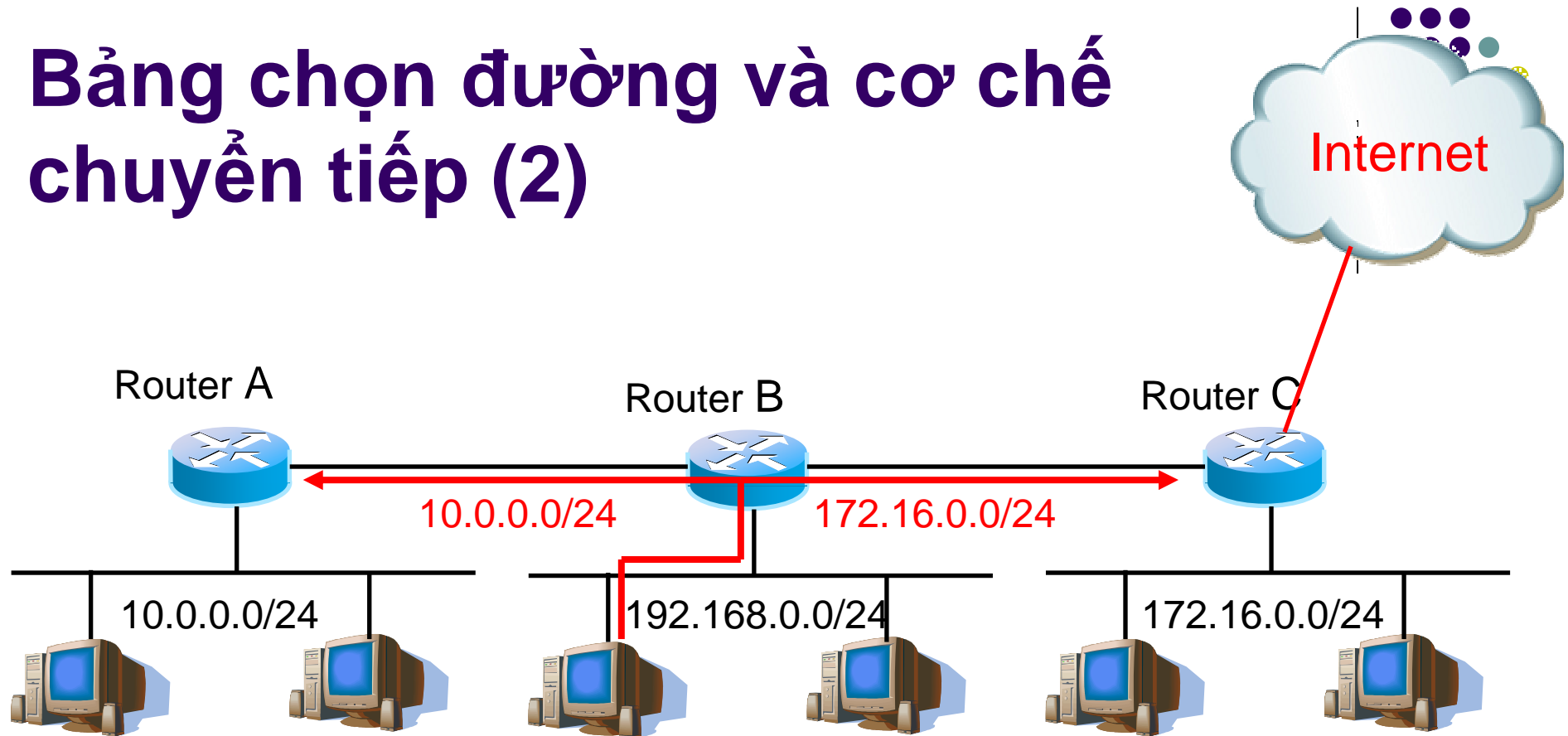
Đường đi 3:

11.0.0.0/8 = 00001011.00000000.00000000.00000000

“Longest matching” là gì?

Tại sao phải cần quy tắc này?

# Bảng chọn đường và cơ chế chuyển tiếp (2)



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

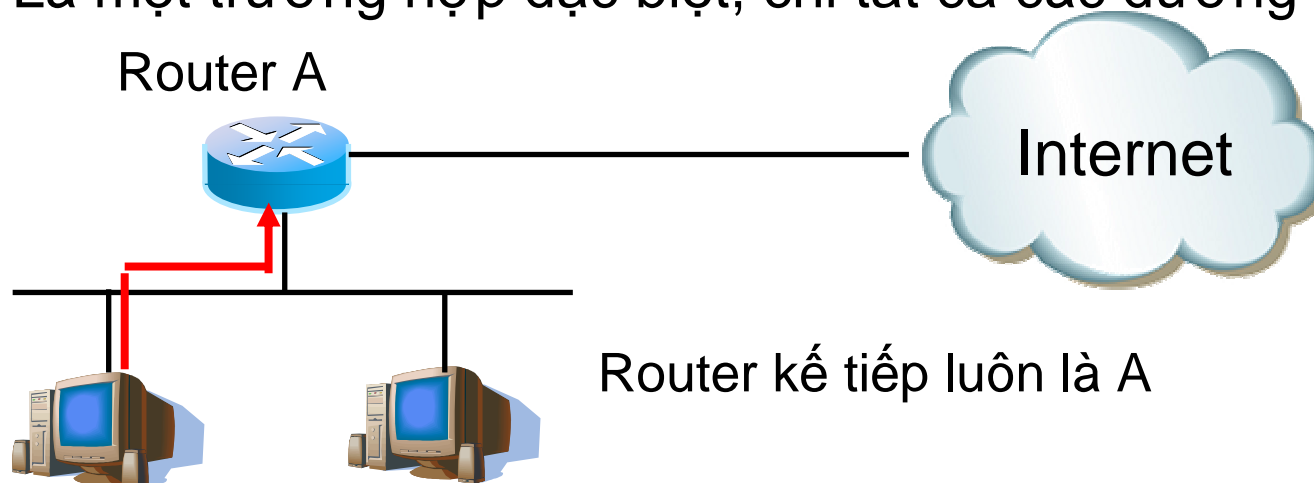
Q. Mô tả bảng chọn đường trên C

Nếu C nối vào Internet?



# Đường đi mặc định

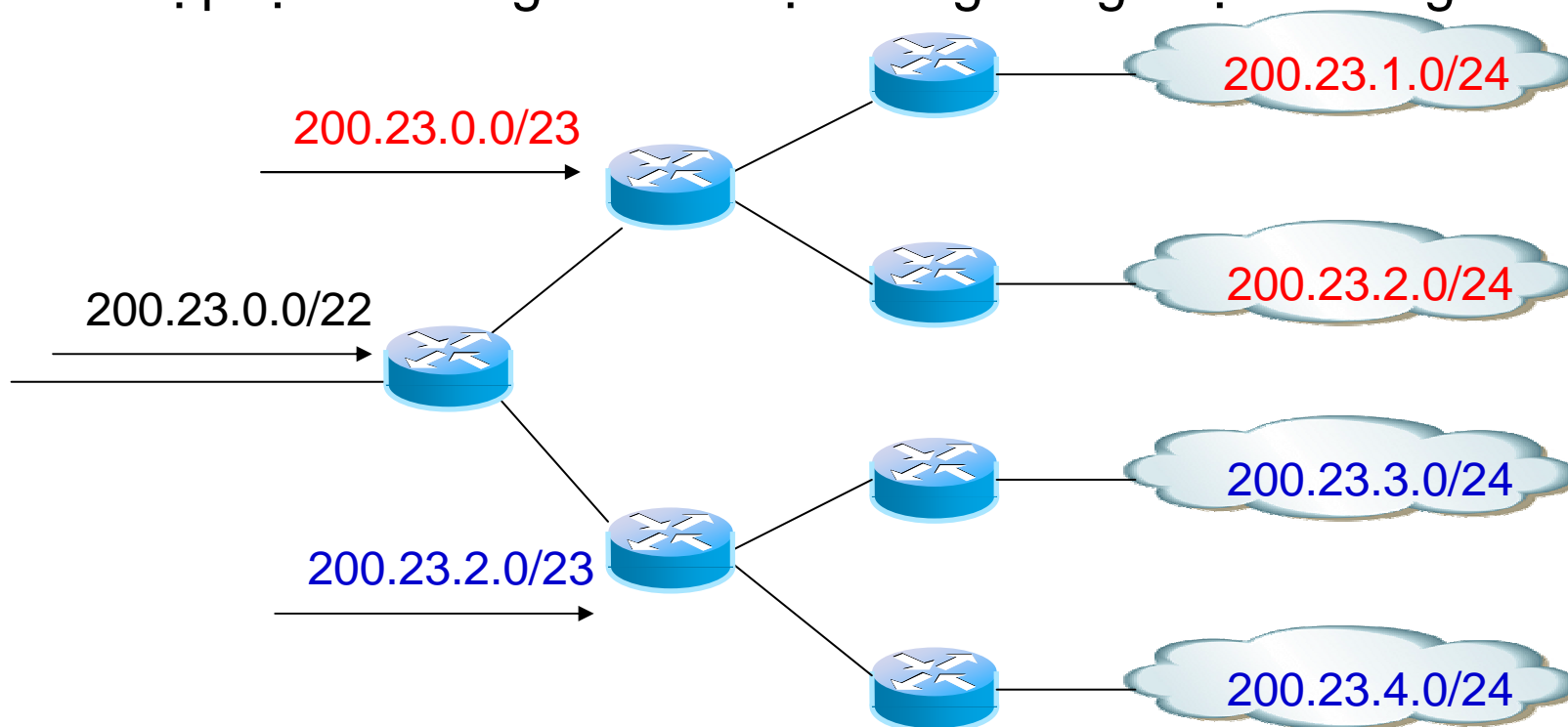
- Nếu đường đi không tìm thấy trong bảng chọn đường
  - Đường đi mặc định trở đến một router kết tiếp
  - Trong nhiều trường hợp, đây là đường đi duy nhất
- 0.0.0.0/0
  - Là một trường hợp đặc biệt, chỉ tất cả các đường đi



# Kết hợp đường đi (Routing aggregation)



- Có bao nhiêu mạng con trên mạng Internet?
- Sẽ có rất nhiều mục trong bảng chọn đường?
- Các mạng con kế tiếp với cùng địa chỉ đích có thể được tổng hợp lại để làm giảm số mục trong bảng chọn đường.





# Kết hợp đường đi (2)

- Ví dụ về Viettel
  - Không gian địa chỉ IP: khá lớn
    - 203.113.128.0-203.113.191.255
  - Để kết nối đến một mạng con của Viettel (khách hàng): Chỉ cần chỉ ra đường đi đến mạng Viettel
- Đường đi mặc định chính là một dạng của việc kết hợp đường
  - 0.0.0.0/0



# Ví dụ về bảng chọn đường – máy trạm



```
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
```

```
=====
```

# Ví dụ về bảng chọn đường – Router (trích)

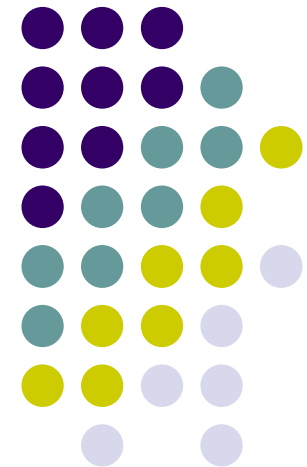


```
#show ip route
```

```
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
```

# Chọn đường tĩnh và chọn đường động

Chọn đường tĩnh  
Chọn đường động  
Ưu điểm – nhược điểm



# Vấn đề cập nhật bảng chọn đường



- Sự thay đổi cấu trúc mạng: thêm mạng mới, một nút mạng bị mất điện
- Sự cần thiết phải cập nhật bảng chọn đường
  - Cho tất cả các nút mạng (về lý thuyết)
  - Thực tế, chỉ một số nút mạng phải cập nhật

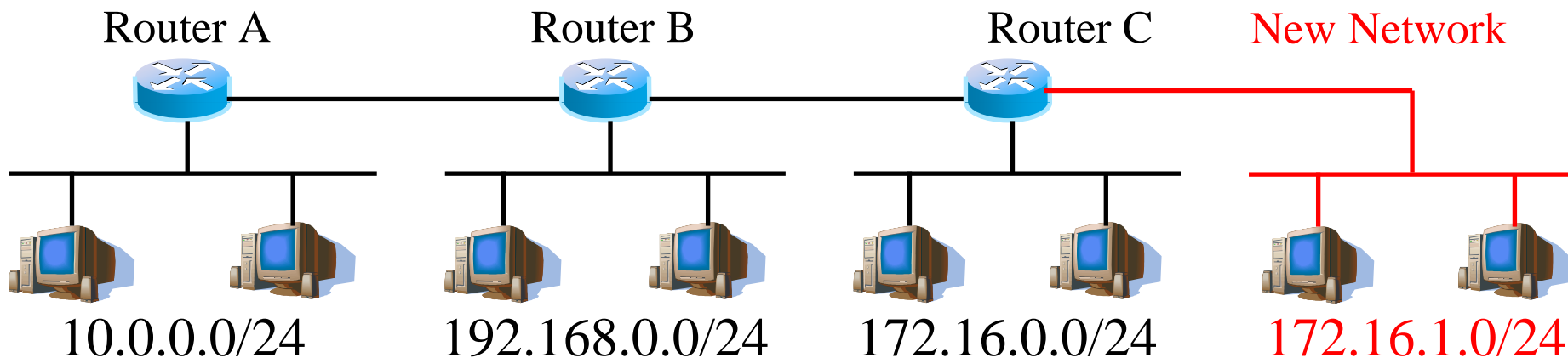
Network	Next-hop
192.168.0.0/24	B
172.16.0.0/24	B

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



# Làm thế nào để cập nhật?



- Chọn đường tĩnh
  - Các mục trong bảng chọn đường được sửa đổi thủ công bởi người quản trị
- Chọn đường động
  - Tự động cập nhật bảng chọn đường
  - Bảng các giao thức chọn đường

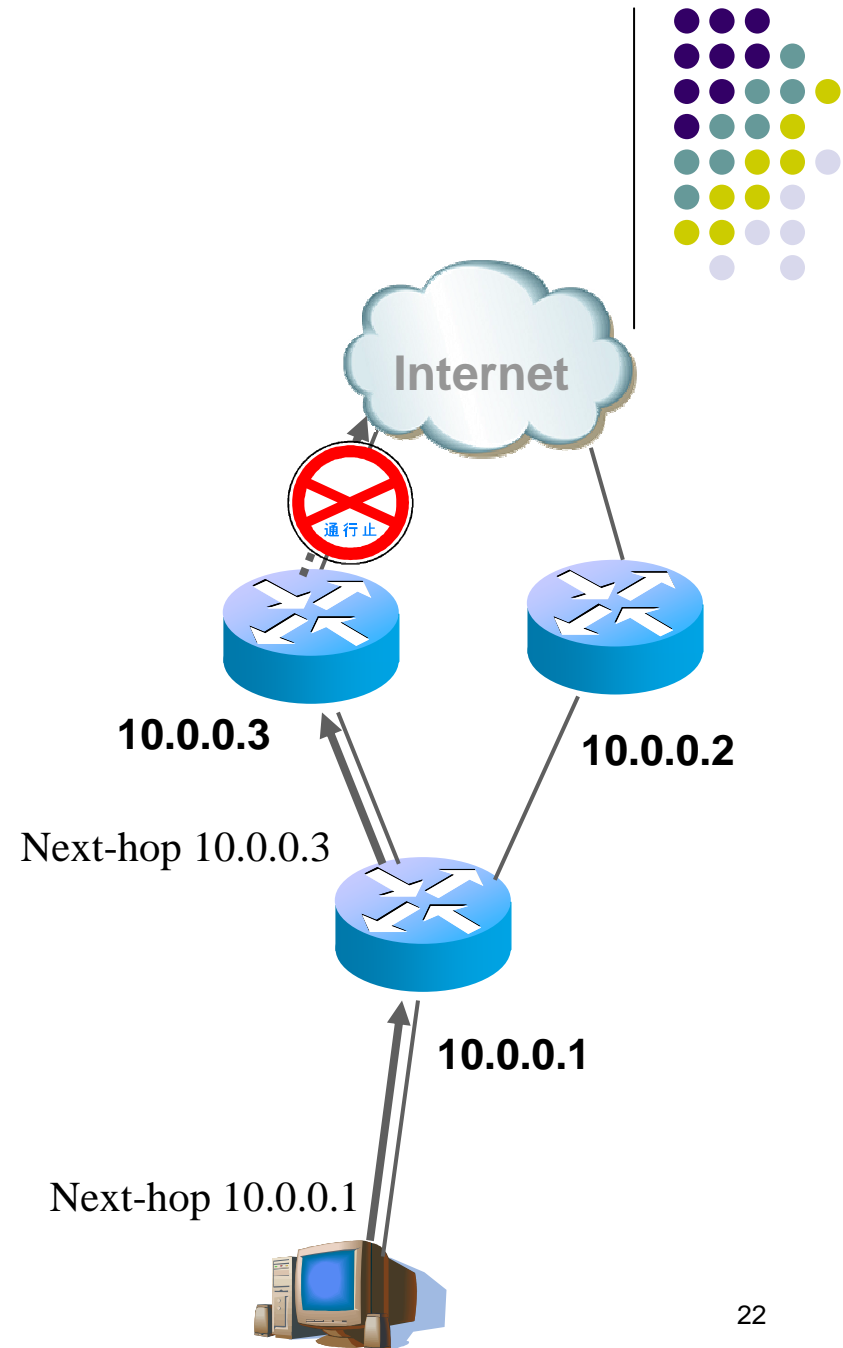
# Chọn đường tĩnh

- Khi có sự cố:
  - Không thể nối vào Internet kể cả khi có tồn tại đường đi dự phòng
  - Người quản trị mạng cần thay đổi

Bảng chọn đường của 10.0.0.1 (1 phần)

Prefix	Next-hop
<del>0.0.0.0/0</del>	<del>10.0.0.3</del>

Kết nối bị lỗi



# Chọn đường động

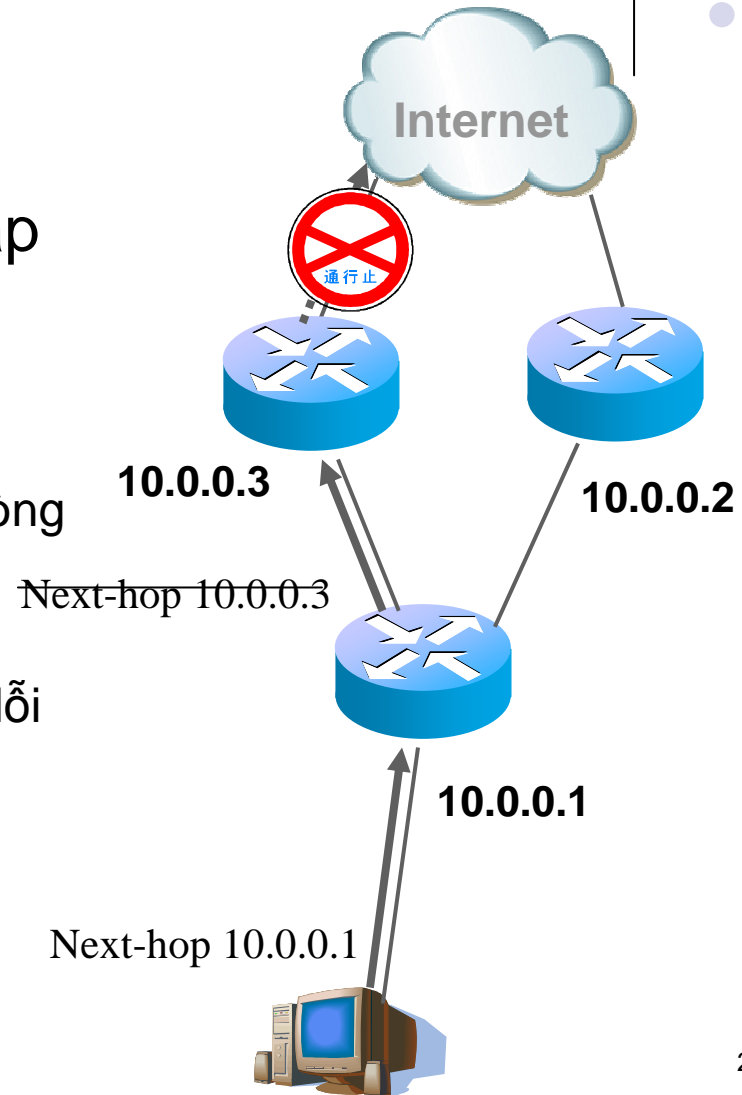
- Khi có sự cố:
  - Đường đi thay thế được cập nhật một cách tự động

Bảng chọn đường của 10.0.0.1 (1 phần)

Prefix	Next-hop
<b>0.0.0.0/0</b>	<b>10.0.0.2</b>
<del>0.0.0.0/0</del>	<del>10.0.0.3</del>

Kết nối dự phòng

Kết nối bị lỗi

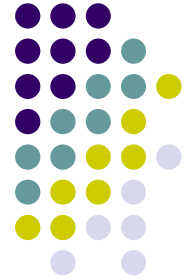


# Đặc điểm của chọn đường tĩnh



- Ưu
  - Ổn định
  - An toàn
  - Không bị ảnh hưởng bởi các yếu tố tác động
- Nhược
  - Cứng nhắc
  - Không thể sử dụng tự động kết nối dự phòng
  - Khó quản lý



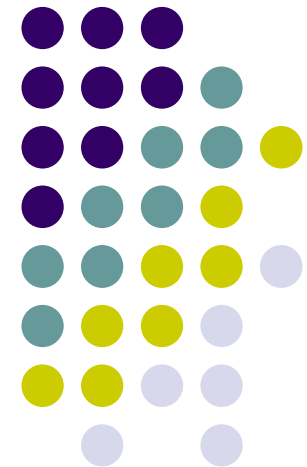


# Chọn đường động

- Ưu
  - Dễ quản lý
  - Tự động sử dụng kết nối dự phòng
- Nhược
  - Tính an toàn
  - Các giao thức chọn đường phức tạp và khó hiểu
  - Khó quản lý

# Các giải thuật và giao thức chọn đường

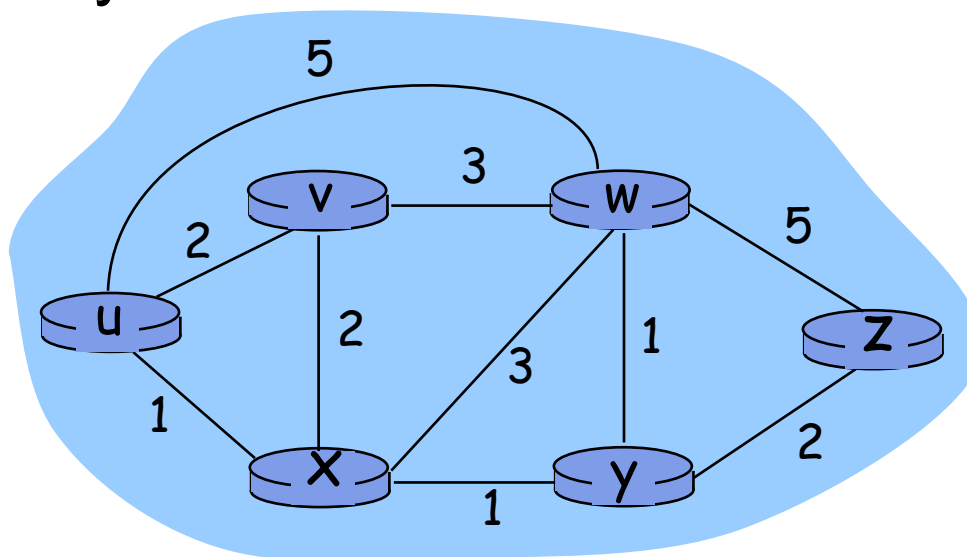
Giải thuật Dijkstra và Bellman-Ford  
Giao thức dạng link-state và dạng distance-vector



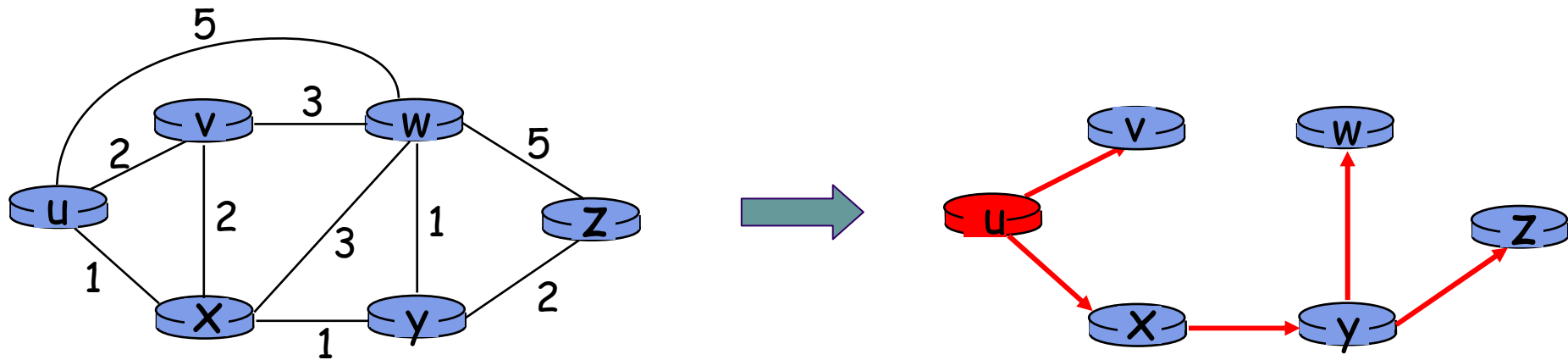
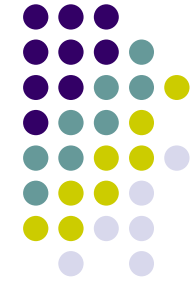


# Biểu diễn mạng bởi đồ thị

- Đồ thị với các nút (bộ định tuyến) và các cạnh (liên kết)
- Chi phí cho việc sử dụng mỗi liên kết  $c(x,y)$ 
  - Băng thông, độ trễ, chi phí, mức độ tắc nghẽn...
- Giải thuật chọn đường: Xác định đường đi ngắn nhất giữa hai nút bất kỳ



# Cây đường đi ngắn nhất - SPT



- SPT – Shortest Path Tree
- Các cạnh xuất phát từ nút gốc và tới các lá
- Đường đi duy nhất từ nút gốc tới nút v, là đường đi ngắn nhất giữa nút gốc và nút v
- Mỗi nút sẽ có một SPT của riêng nút đó



# Tập trung hay phân tán

- Tập trung
  - Thu thập thông tin vào một nút mạng
  - Sử dụng các **giải thuật tìm đường đi trên đồ thị**
  - Phân bổ băng chọn đường từ nút trung tâm tới các nút
- Phân tán
  - Mỗi nút tự xây dựng bảng chọn đường riêng
  - **Giao thức chọn đường**: Link-state hoặc distance-vector
  - Được sử dụng phổ biến trong thực tế



# Tập trung hay phân tán

- Thông tin chọn đường là cần thiết để xây dựng bảng chọn đường
- Tập trung hay phân tán?
  - Tập trung:
    - Mỗi router có thông tin đầy đủ về trạng thái của mạng
    - Giải thuật dạng “link state”
  - Phân tán:
    - Các nút chỉ biết được trạng thái của liên kết vật lý tới nút kế bên
    - Liên tục lặp lại việc tính toán và trao đổi thông tin với nút kế bên
    - Giải thuật dạng “distance vector”
    - “Bạn của bạn cũng là bạn”



# Giải thuật dạng link-state

## Giải thuật Dijkstra's

- Mỗi nút đều có sơ đồ và chi phí mỗi link
  - Quảng bá “Link-state”
  - Mỗi nút có cùng thông tin
- Tìm đường đi chi phí nhỏ nhất từ một nút (‘nguồn’) tới tất cả các nút khác
  - dùng để xây dựng bảng chọn đường



# Ký hiệu

- $G = (V, E)$  : Đồ thị với tập đỉnh  $V$  và tập cạnh  $E$
- $c(x, y)$ : chi phí của liên kết  $x$  tới  $y$ ;  $= \infty$  nếu không phải 2 nút kề nhau
- $d(v)$ : chi phí hiện thời của đường đi từ nút nguồn tới nút đích.  $v$
- $p(v)$ : nút ngay trước nút  $v$  trên đường đi từ nguồn tới đích
- $T$ : Tập các nút mà đường đi ngắn nhất đã được xác định





# Các thủ tục

- **Init():**

Với mỗi nút  $v$ ,  $d[v] = \infty$ ,  $p[v] = \text{NIL}$

$d[s] = 0$

- **Improve( $u, v$ )**, trong đó  $(u, v)$   $u, v$  là một cạnh nào đó của  $G$

if  $d[v] > d[u] + c(u, v)$  then

$d[v] = d[u] + c(u, v)$

$p[v] = u$



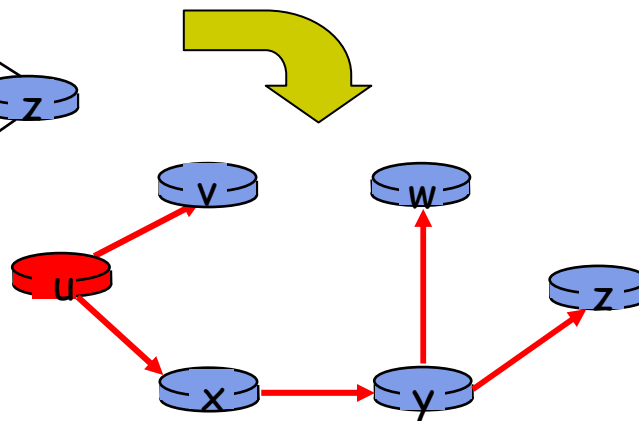
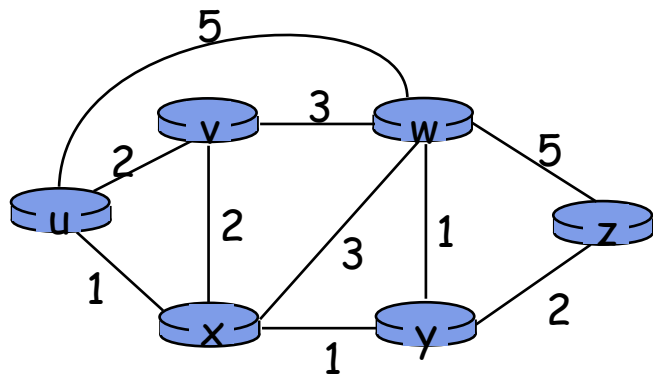
# Dijkstra's Algorithm

1. **Init()** ;
2.  $T = \Phi$ ;
3. **Repeat**
4.      $u: u \in T \mid d(u)$  là bé nhất ;
5.      $T = T \cup \{u\}$ ;
6.     for all  $v \in \text{neighbor}(u)$  và  $v \notin T$
7.          $\text{update}(u, v)$  ;
8. **Until**  $T = V$

# Dijkstra's algorithm: Ví dụ



Step	T	$d(v), p(v)$	$d(w), p(w)$	$d(x), p(x)$	$d(y), p(y)$	$d(z), p(z)$
0	u	2,u	5,u	1,u	$\infty$	$\infty$
1	ux	2,u	4,x		2,x	$\infty$
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



SPT của u:

Bảng chọn đường của u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x) <sup>35</sup>

# Giải thuật dạng distance-vector (1)



## Phương trình Bellman-Ford (quy hoạch động)

Định nghĩa

$d_x(y)$  := chi phí của đường đi ngắn nhất  
từ  $x$  tới  $y$

Ta có

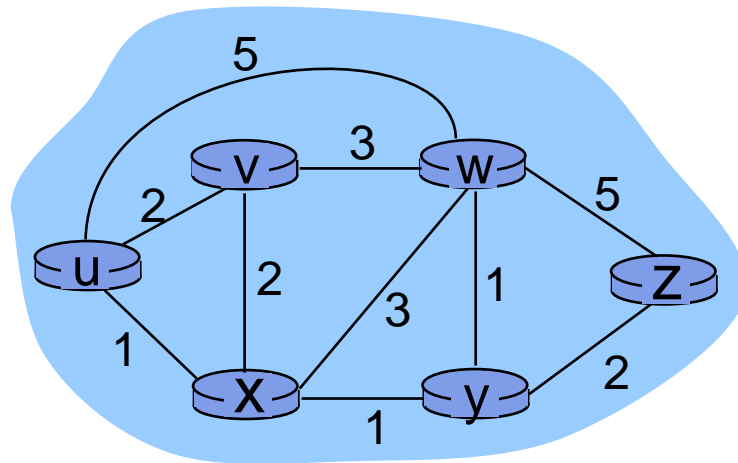
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

cho tất cả các  $v$  là hàng xóm của  $x$

# Minh họa Bellman-Ford Eq.



Dễ thấy,  $d_v(z) = 5$ ,  $d_x(z) = 3$ ,  $d_w(z) = 3$

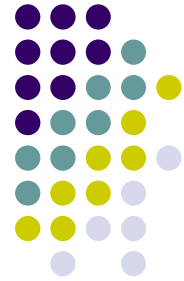


B-F eq. cho ta biết:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

Nút nào làm giá trị trên nhỏ nhất  $\rightarrow$  Lựa chọn là nút kế tiếp trong bảng chọn đường

# Giải thuật dạng distance-vector (2)



## ý tưởng cơ bản:

- DV: Vector khoảng cách, tạm coi là đường đi ngắn nhất của từ một nút tới những nút khác
- Mỗi nút định kỳ gửi DV của nó tới các nút bên cạnh
- Khi nút x nhận được 1 DV, nó sẽ cập nhật DV của nó qua pt Bellman-ford
- Với một số điều kiện, ước lượng  $D_x(y)$  sẽ hội tụ dần đến giá trị nhỏ nhất  $d_x(y)$

## Mỗi nút:

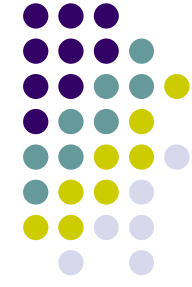
*Chờ* (Thay đổi trong DV của nút bên cạnh)

*Tính lại* ước lượng DV

Nếu DV thay đổi, *Báo* cho nút bên cạnh

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

$$= \min\{2+0, 7+1\} = 2$$



nút x

		chi phí tới		
		x	y	z
tù	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		chi phí tới		
		x	y	z
tù	x	0	2	3
	y	2	0	1
	z	7	1	0

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

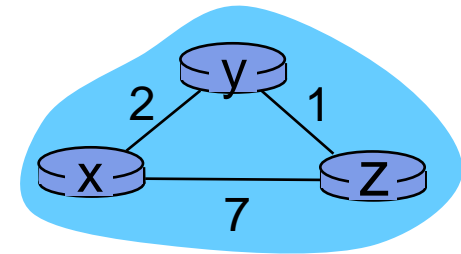
$$= \min\{2+1, 7+0\} = 3$$

nút y

		chi phí tới		
		x	y	z
tù	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

nút z

		chi phí tới		
		x	y	z
tù	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0



► thời gian

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

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

**nút x**

chi phí tới

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

**nút y**

chi phí tới

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

chi phí tới

	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

**nút z**

chi phí tới

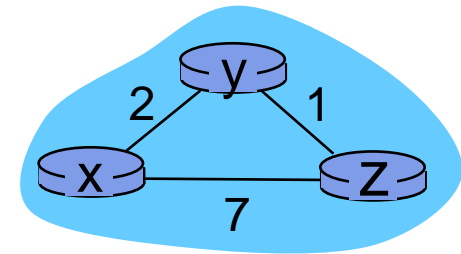
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

chi phí tới

	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



▶ thời gian



# So sánh các giải thuật LS và DV



## Thông điệp trao đổi

- LS: n nút, E cạnh,  $O(nE)$  thông điệp
- DV: Chỉ trao đổi giữa các hàng xóm
  - Thời gian hội tụ thay đổi

## Tốc độ hội tụ

- LS: Thuật toán:  $O(n^2)$  cần  $O(nE)$  thông điệp
- DV: Thay đổi

**Sự chắc chắn:** Giải sử một router hoạt động sai

### LS:

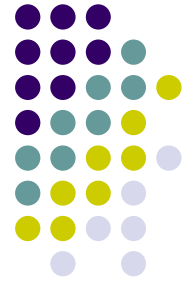
- nút gửi các chi phí sai
- Mỗi nút tính riêng bảng chọn đường -> có vẻ chắc chắn hơn

### DV:

- DV có thể bị gửi sai
- Mỗi nút tính toán dựa trên các nút khác
  - Lỗi bị lan truyền trong mạng

# Tóm tắt

- Nguyên lý của bài toán chọn đường
- Tĩnh vs. động, tập trung vs. phân tán
- Link-state vs. distance-vector



# Tuần tới: Các giao thức chọn đường trên Internet

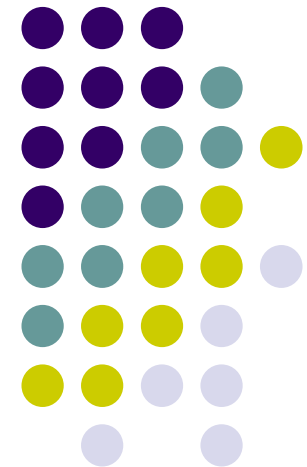


- Chọn đường phân cấp
- RIP
- OSPF
- BGP

# Chương 5: Các giao thức chọn đường

Dự án HEDSPI  
Khoa CNTT- ĐHBK Hà Nội

Giảng viên: Ngô Hồng Sơn  
Bộ môn Truyền thông và Mạng máy tính



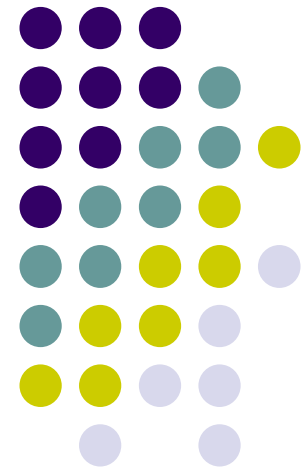


# Tổng quan

- Tuần trước: Khái niệm chung về chọn đường.
  - Chọn đường tĩnh và chọn đường động
  - Các giao thức chọn đường
- Tuần này: Các giao thức chọn đường cụ thể trên Internet
  - Vấn đề phân cấp trong chọn đường
  - Chọn đường nội vùng (Intra-domain): RIP, OSPF
  - Chọn đường liên vùng (Inter-domain): BGP

# Phân cấp trong chọn đường

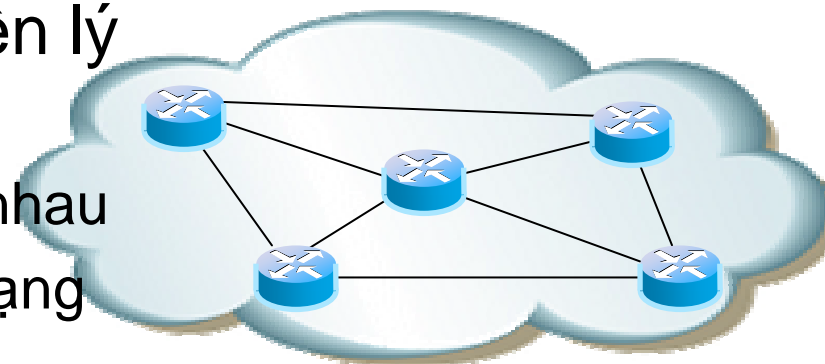
Các hệ tự trị  
Chọn đường nội vùng  
Chọn đường liên vùng



# Tổng quan



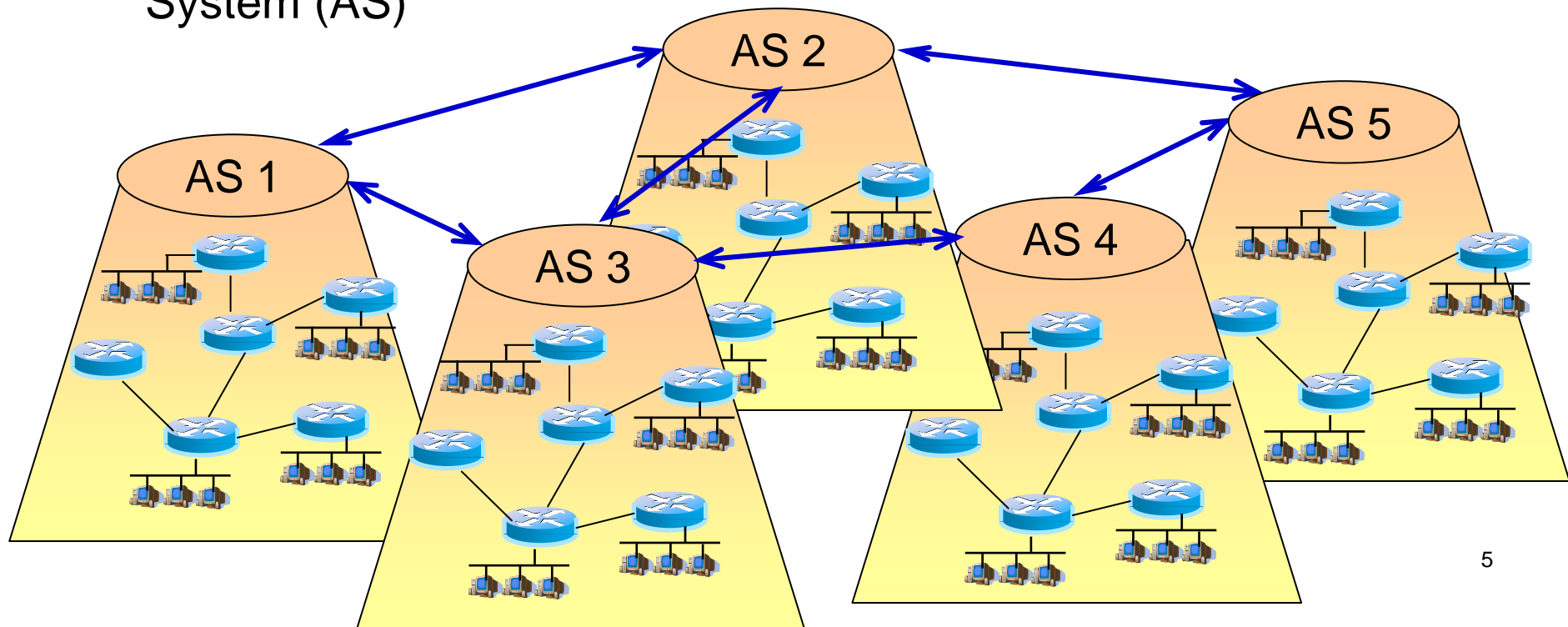
- Vấn đề chọn đường đã học được xem xét trong điều kiện lý tưởng
  - Các nút mạng có vai trò như nhau
  - Chỉ có một mạng duy nhất, mạng “phẳng”
- Thực tế không giống như vậy
- Tính mở rộng: Internet có hàng triệu (tỷ) máy trạm, chọn đường bằng LS hay DV?
  - LS: Quá tải thông tin chọn đường
  - DV: Có hội tụ được không?





# Kiến trúc phân cấp của Internet

- Internet = Mạng của các mạng
- Mỗi mạng có thể lựa chọn riêng cho mình một chiến lược chọn đường riêng.
- Mỗi mạng như vậy có thể gọi là một hệ tự trị - Autonomous System (AS)





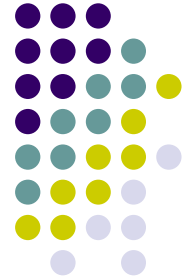
**Slide 5**

---

**s1**

**Combine 5 and 6**

sonnh, 3/8/2008



# Khái niệm hệ tự trị - AS

- Tập hợp các nút mạng có cùng chính sách chọn đường (Giao thức, quy ước chi phí...)
- Các ASes được nối kết thông qua các router hay gateway
- Mỗi hệ tự trị có một số hiệu riêng – AS number (ASN - 16 bits hay 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

.....

**Slide 6**

---

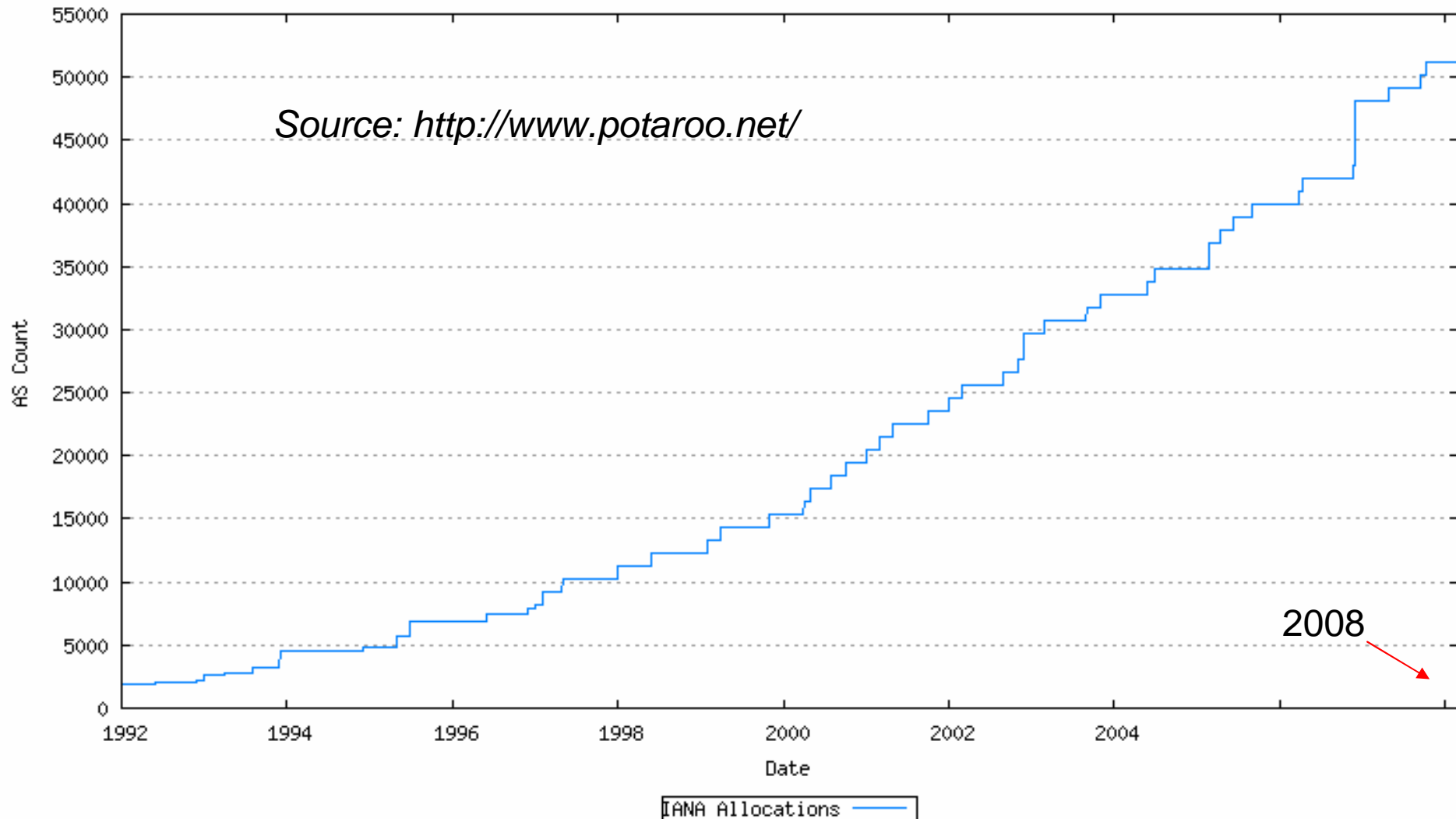
**s2**

**Explain about AS**  
sonnh, 3/8/2008

# Số lượng ASN cấp phát bởi IANA



Time Series of IANA AS Allocations

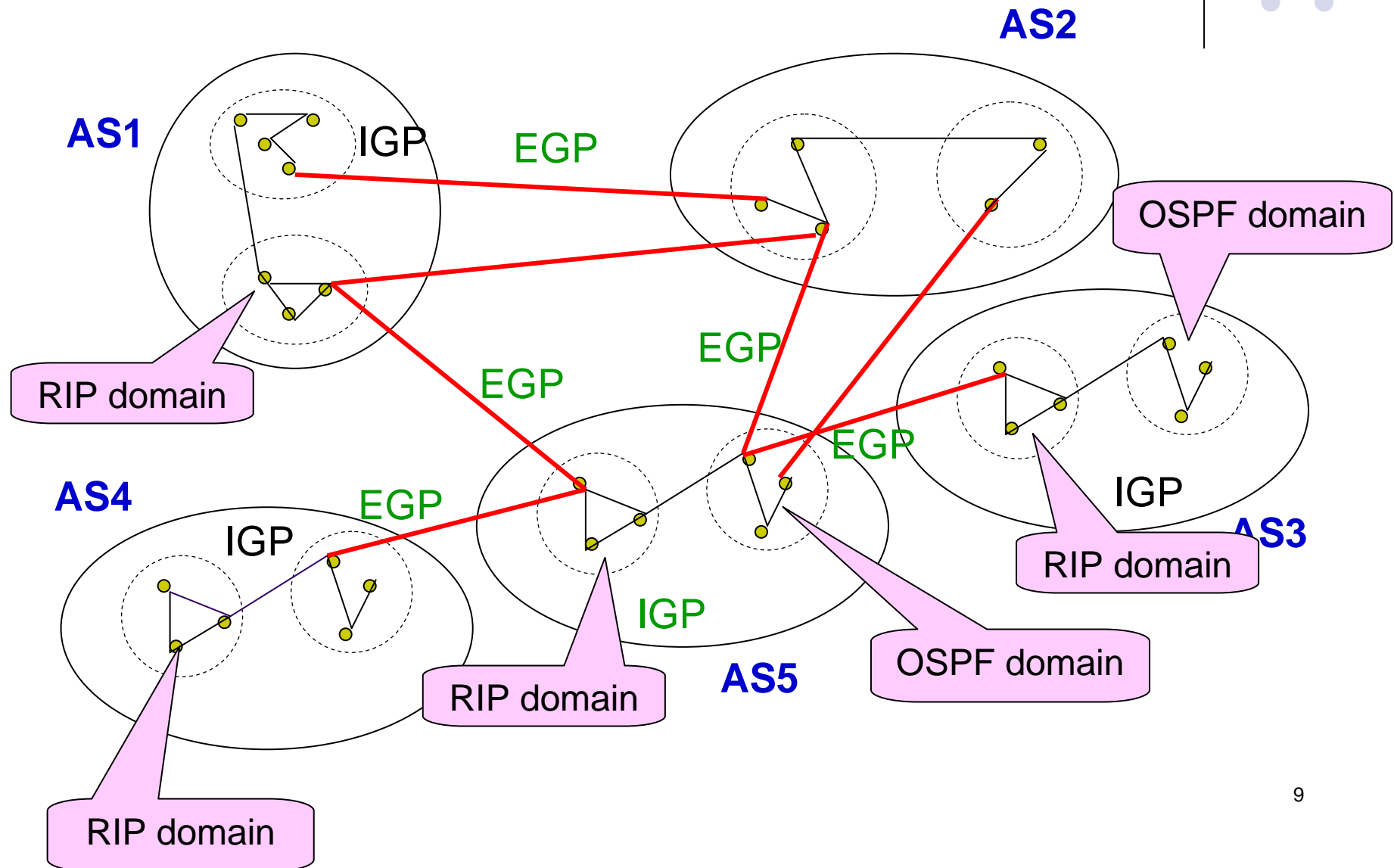
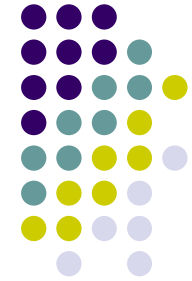


# Phân cấp giao thức chọn đường



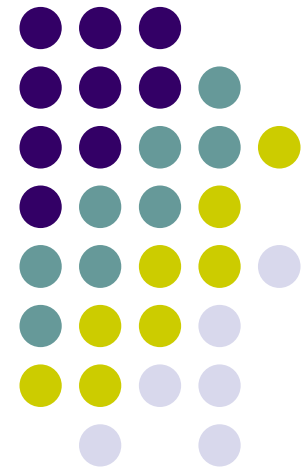
- **Trong một hệ tự trị:** Giao thức chọn đường nội vùng
  - **IGP:** *Interior Gateway Protocol*
    - RIP: Routing Information Protocol
    - OSPF: Open Shortest Path First
    - IS-IS, IGRP, EIGRP (Cisco)...
- **Giữa các hệ tự trị:** Giao thức chọn đường liên vùng
  - **EGP:** *Exterior Gateway Protocol*
    - BGP (v4): Border Gateway Protocol

# Intra-domain và Inter-domain routing



# Chọn đường nội vùng

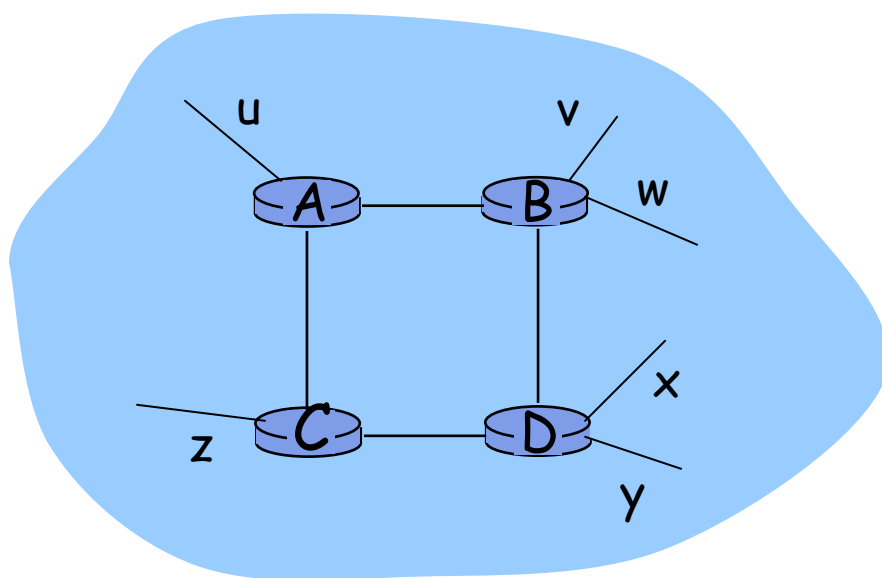
RIP  
OSPF





# RIP ( Routing Information Protocol)

- IGP
- RIP v.1, phiên bản mới RIP v.2
- Giao thức dạng vector khoảng cách
- Chọn đường đi theo số nút mạng đi qua (# of hops, max = 15 hops)



Từ nút A:

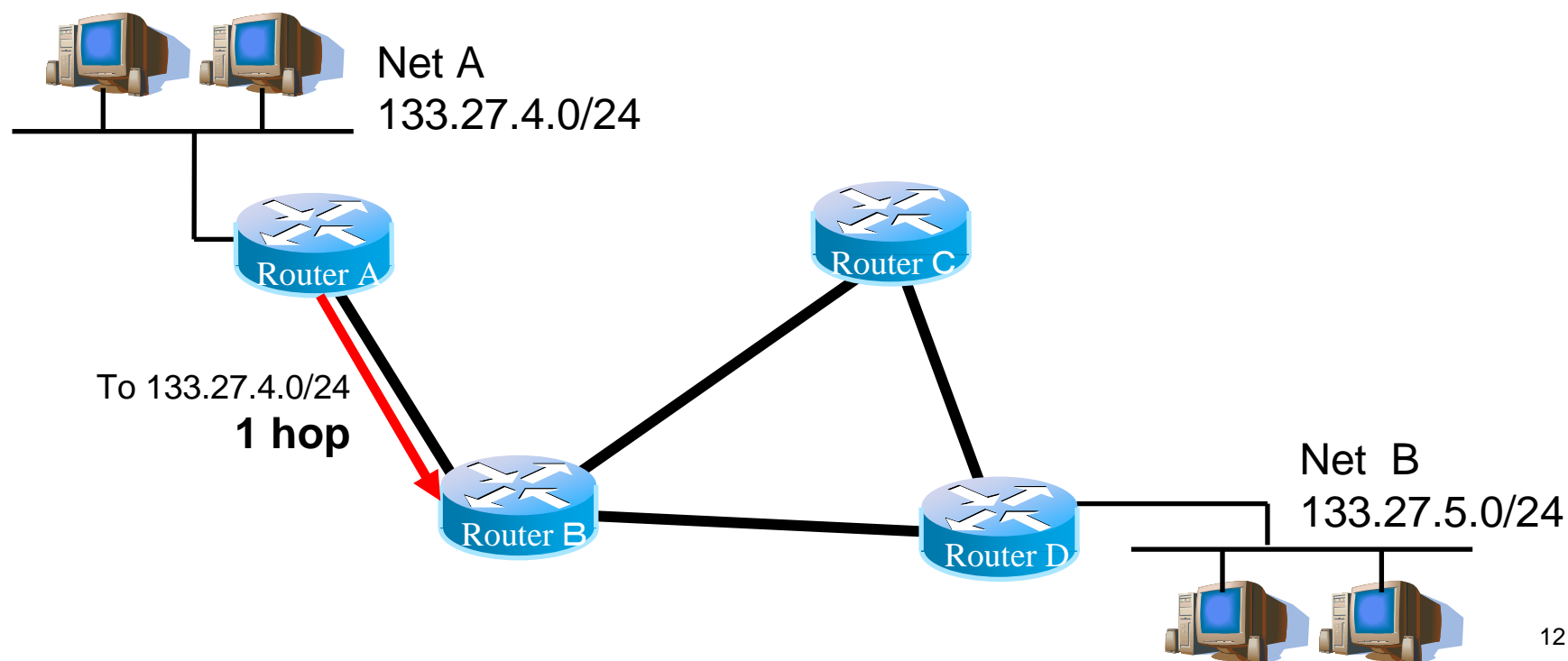
<u>Đích đến</u>	<u>số nút</u>
u	1
v	2
w	2
x	3
y	3
z	2





# Nhắc lại: Chọn đường dạng DV (1)

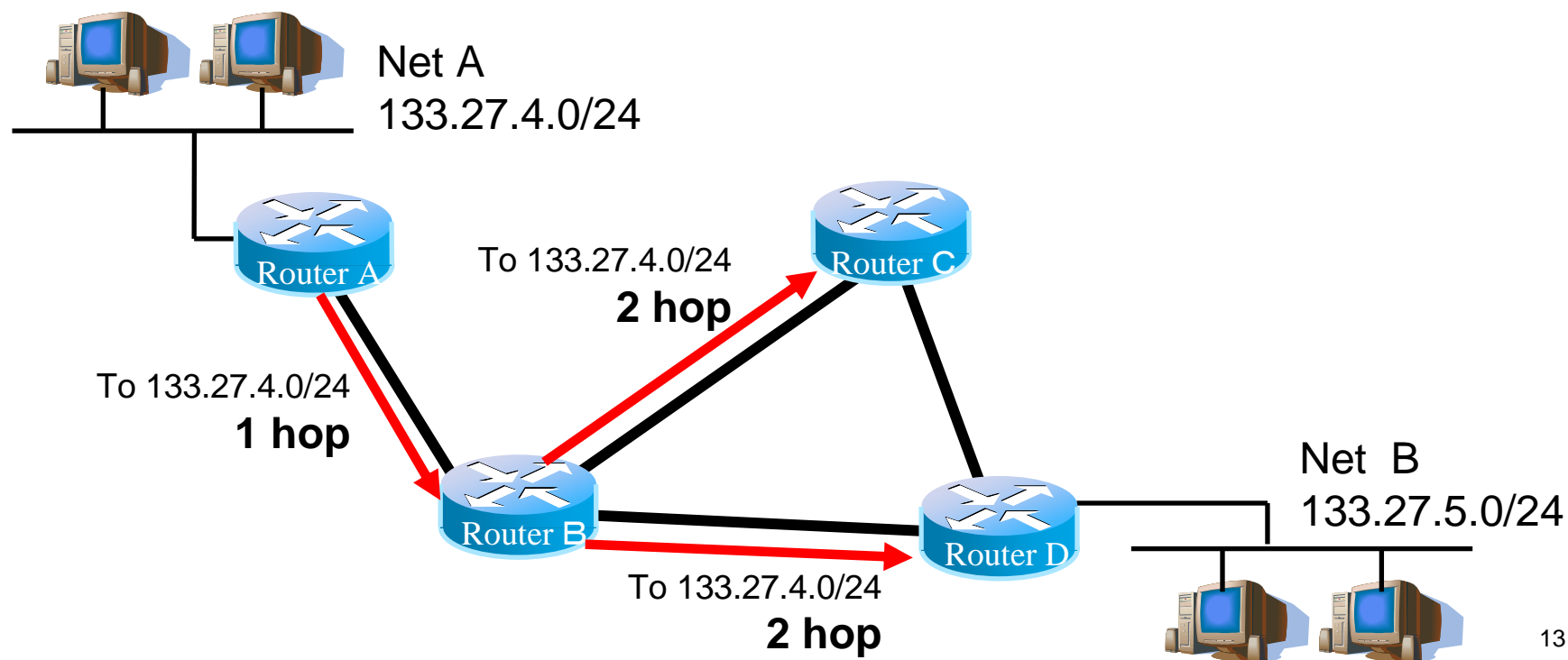
- Bạn của bạn là bạn





# Nhắc lại: Chọn đường dạng DV (2)

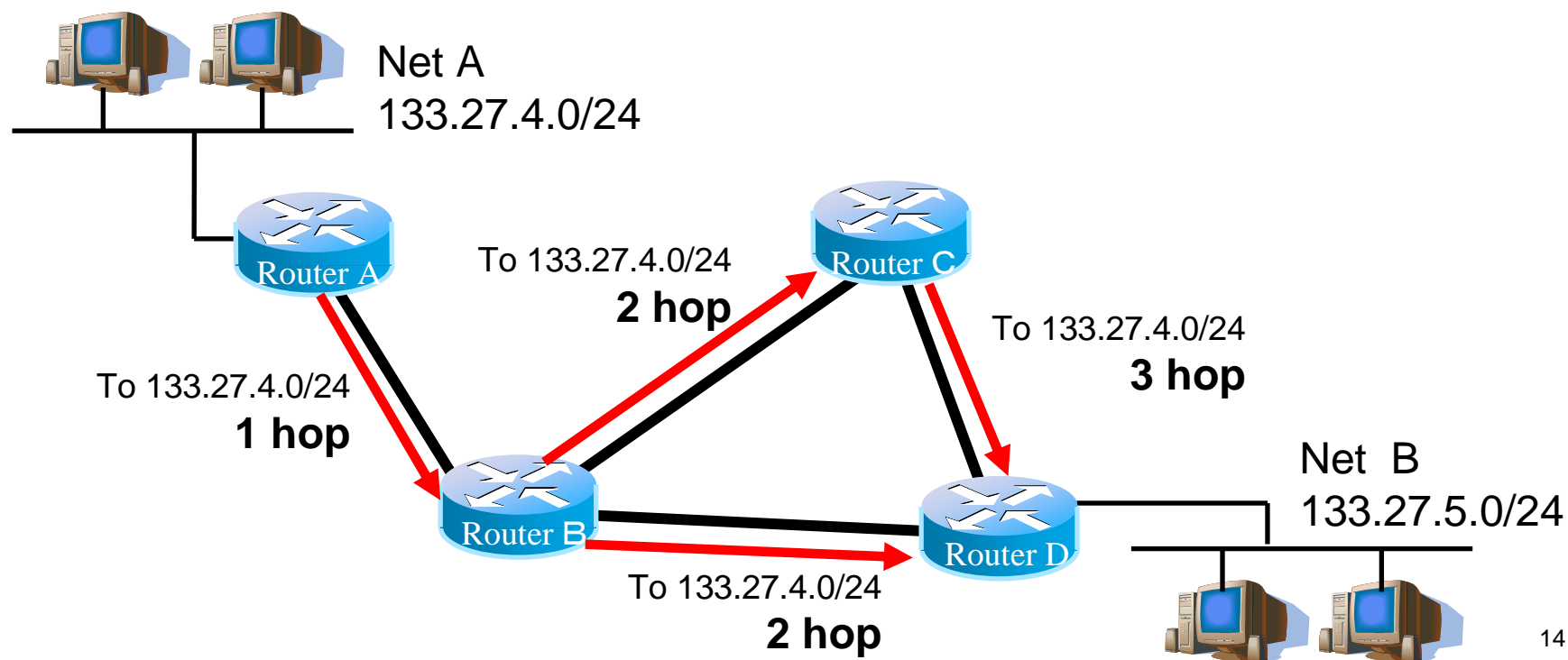
- Bạn của bạn là bạn





# Nhắc lại: Chọn đường dạng DV (3)

- Bạn của bạn là bạn

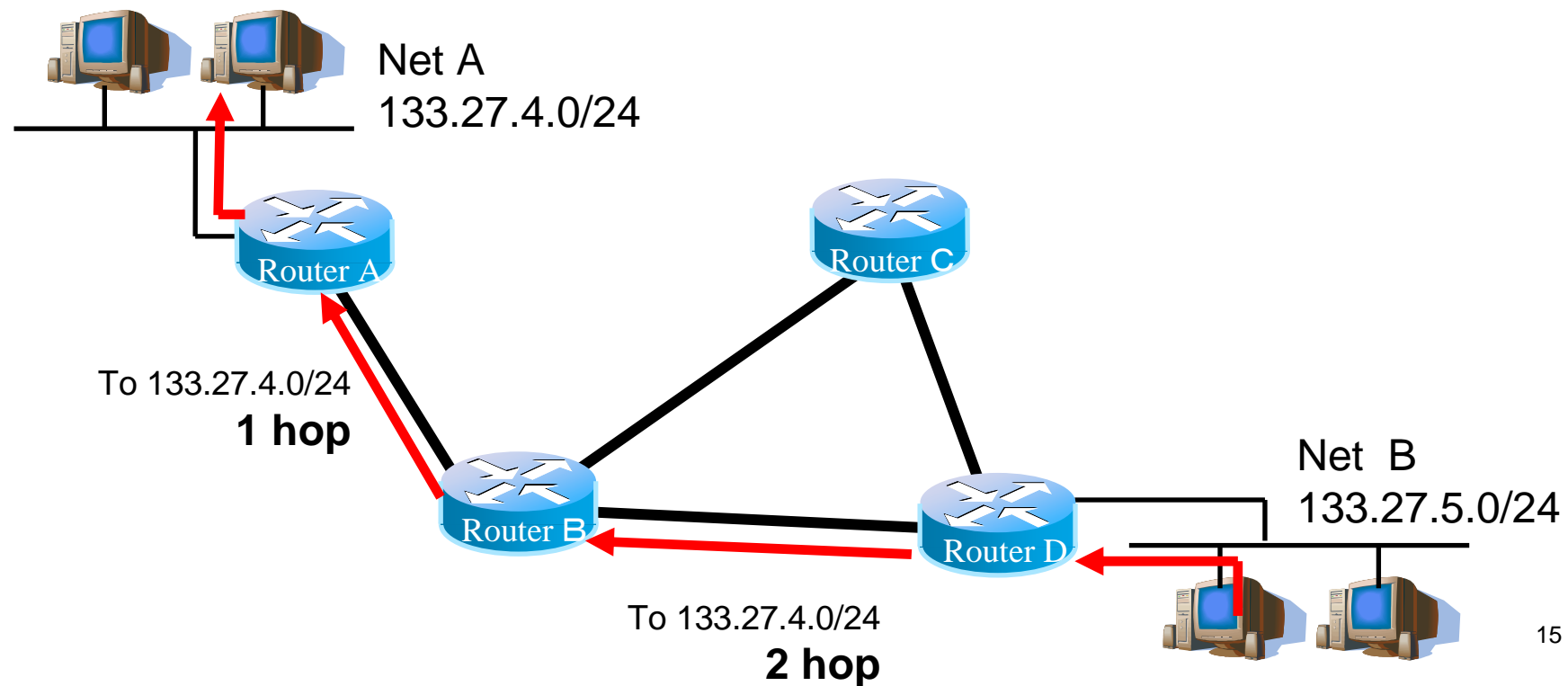




# Nhắc lại: Chọn đường dạng DV (4)

- Bạn của bạn là bạn

Lưu ý: Tên của router



## Slide 15

---

**s3**

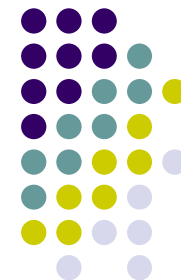
Explain in opposite way: How B is annouced

sonnh, 3/8/2008

**s4**

Explain that we announce networks address. not router id

sonnh, 3/8/2008



# RIP: Trao đổi thông tin

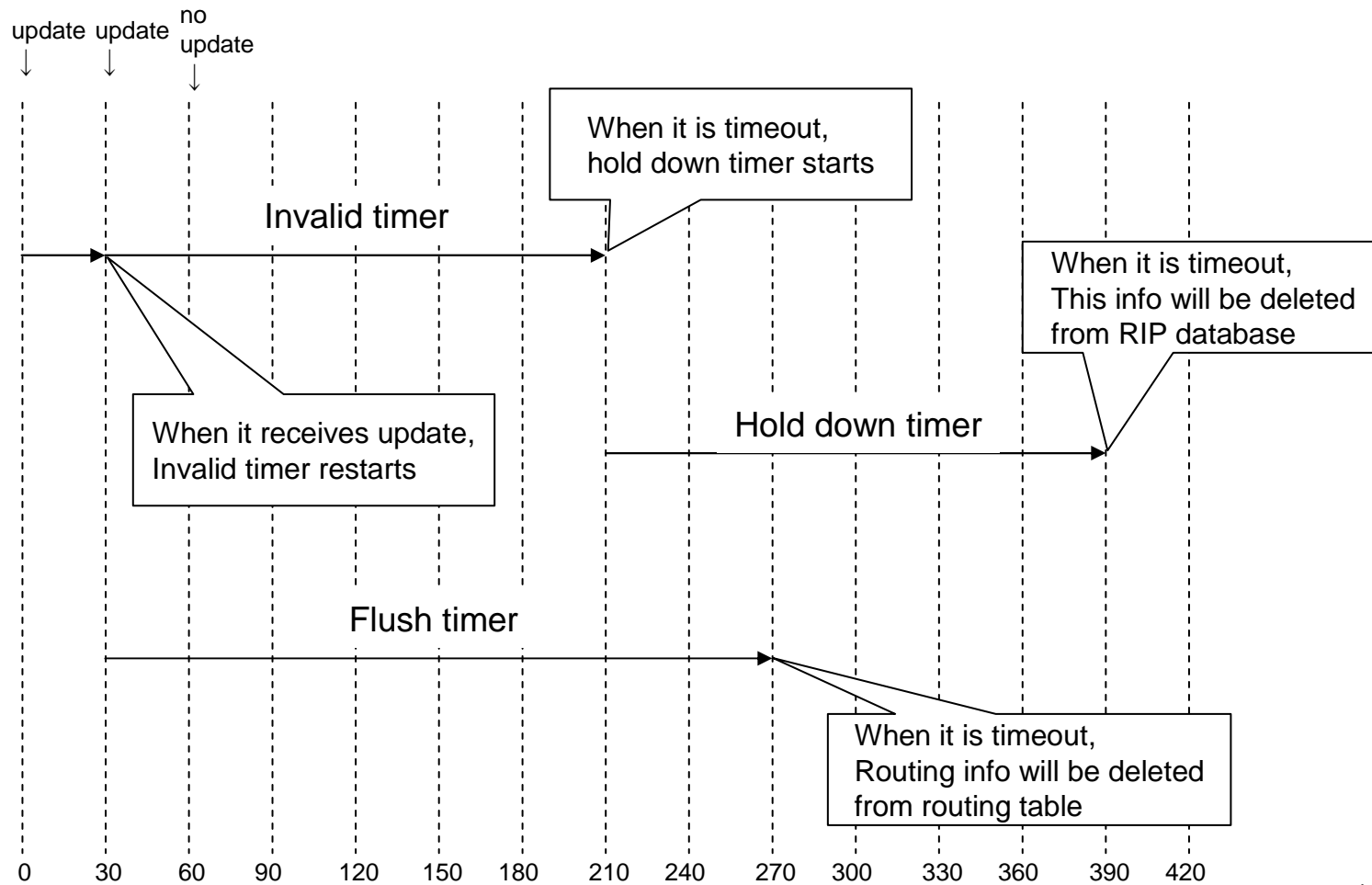
- Trao đổi bảng chọn đường
- Định kỳ
  - Các vector khoảng cách được trao đổi định kỳ - 30s
  - Mỗi thông điệp chứa tối đa 25 mục
  - Trong thực tế, nhiều thông điệp được sử dụng
- Sự kiện
  - Gửi thông điệp cho nút hàng xóm mỗi khi có thay đổi
  - Nút hàng xóm sẽ cập nhật bảng chọn đường của nó

# Các bộ đếm thời gian - RIP timer (1)



- Update timer
  - Dùng để trao đổi thông tin cứ 30s
- Invalid timer
  - Khởi tạo lại mỗi khi nhận được thông tin chọn đường
  - Nếu sau 180s không nhận được thông tin -> trạng thái hold-down
- Hold down timer
  - Giữ trạng thái hold-down trong 180s
  - Chuyển sang trạng thái down
- Flush timer
  - Khởi tạo lại mỗi khi nhận được thông tin chọn đường
  - Sau 120s, xóa mục tương ứng trong bảng chọn đường

# RIP timer (2)

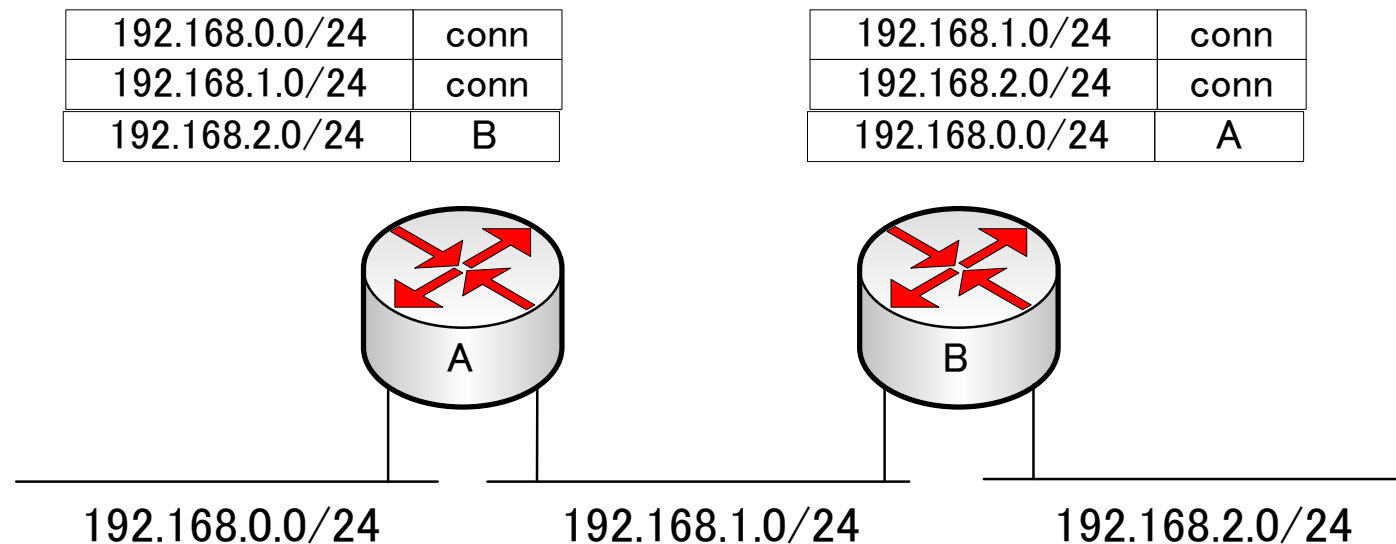




# Lỗi lặp vô hạn (Ping-pong failure)



- Nếu 192.168.0.0/24 bị lỗi...
  - B cập nhật thông tin về 192.168.0.0 cho A
  - Các gói tin đến 192.168.0.0/24 sẽ bị quẩn
- A cập nhật thông tin về 192.168.0.0 cho B
  - Luẩn quẩn, vô hạn





# OSPF: Đề tránh lỗi lặp vô hạn

- Giới hạn số hop tối đa
  - 16
- “Split horizon”
  - Thông tin chọn đường không được quay về nút nguồn
- “Poison reverse”
  - Khi liên kết bị lỗi, gửi giá trị của chi phí là 16
  - Liên kết chuyển sang trạng thái hold-down

**Slide 20**

---

**s6**

**16 TTL vs. this?**  
sonnh, 3/8/2008



# OSPF: Open Shortest Path First

- IGP
- *Open*: Chuẩn mở của IETF (phiên bản 3, định nghĩa trong [RFC 2740](#))
- *Shortest Path First*: Cài đặt giải thuật Dijkstra.
- Thông tin về trạng thái liên kết - LSA (link state advertisement) được quảng bá “tràn ngập” trên toàn AS

**Slide 21**

---

**s7**

**Tittle slide**  
sonnh, 3/8/2008



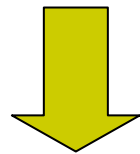
# Một số đặc điểm của OSPF

- An toàn: thông điệp OSPF được bảo vệ
- Với các AS lớn: OSPF phân cấp
- Địa chỉ không phân lớp (Variable-Length Subnet Masking -VLSM )
- Mỗi link sẽ có nhiều giá trị về chi phí khác nhau dựa trên TOS (tuy nhiên hơi phức tạp và chưa được sử dụng)



# Phân cấp OSPF

- Trong việc chọn đường, tại sao phải chia mạng thành các vùng nhỏ hơn?
- Nếu có quá nhiều router
  - Thông tin trạng thái liên kết được truyền nhiều lần hơn
  - Phải liên tục tính toán lại
  - Cần nhiều bộ nhớ hơn, nhiều tài nguyên CPU hơn
  - Lượng thông tin phải trao đổi tăng lên
  - Bảng chọn đường lớn hơn



- Vùng
  - Nhóm các router có cùng thông tin LSA

## Slide 23

---

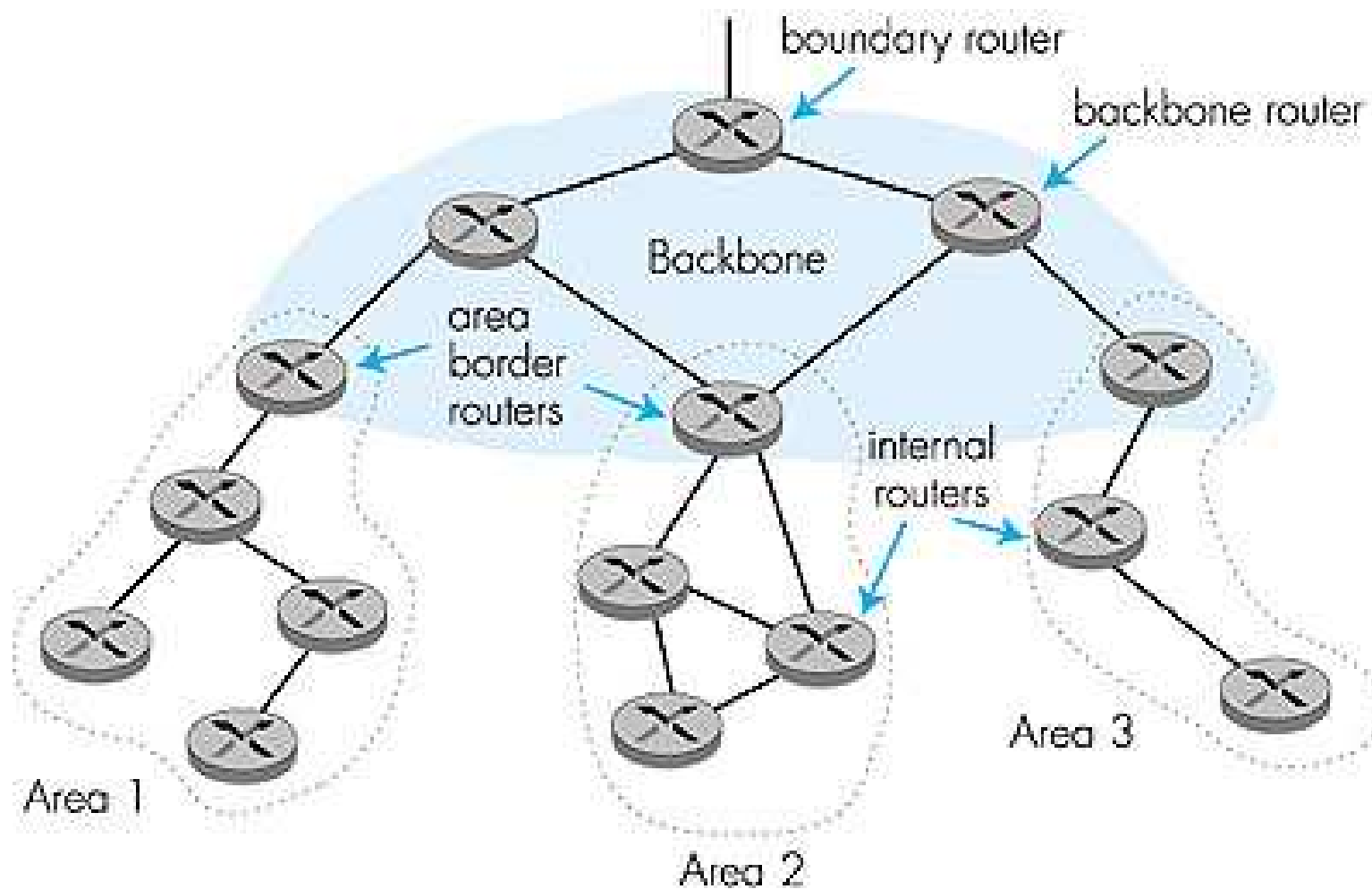
s9

Explain why we need to reduce the calculaiton  
sonnh, 3/8/2008





# Phân cấp OSPF





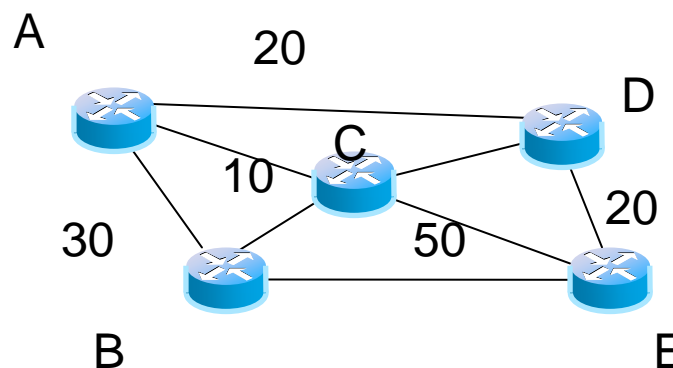
# Các dạng router

- **ABR - Area border routers:** Quản lý 1 vùng và kết nối đến các vùng khác
- **ASBR - Autonomous system boundary router:** Nối đến các AS khác
- **BR - backbone routers:** thực hiện OSPF routing trong vùng backbone
- **Internal Router** – Thực hiện OSPF bên trong một vùng



# Thông tin chọn đường?

- Link-State Advertisement (LSA): Chỉ ra một nút được nối tới nút nào (link) và chi phí (cost) tương ứng
- Ví dụ: nút A
  - link to B, cost 30
  - link to D, cost 20
  - link to C, cost 10
- Ví dụ: nút D
  - link to A, cost 20
  - link to E, cost 20
  - link to C, cost 50



# Chi phí trong giao thức OSPF - metric



- Giá trị mặc định  
100Mbps / bandwidth of interface
  - Hiện nay người quản trị có thể gán giá trị này
- Khi tính toán bảng chọn đường
  - Chọn đường đi chi phí nhỏ nhất
- Chi phí bằng nhau
  - Có thể thực hiện cân bằng tải

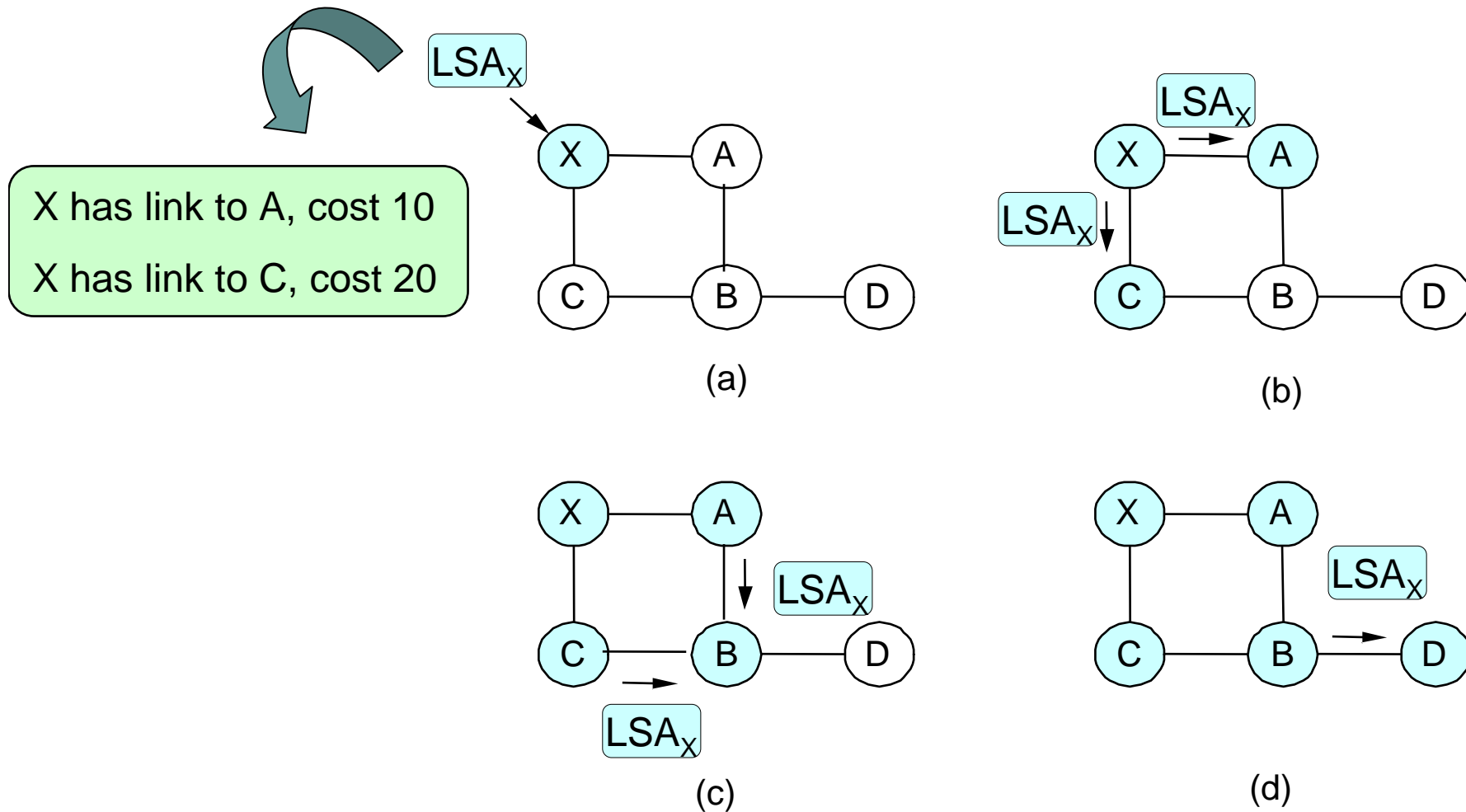


# Chi phí mặc định của OSPF

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



# Quảng bá thông tin LSA



**s8**

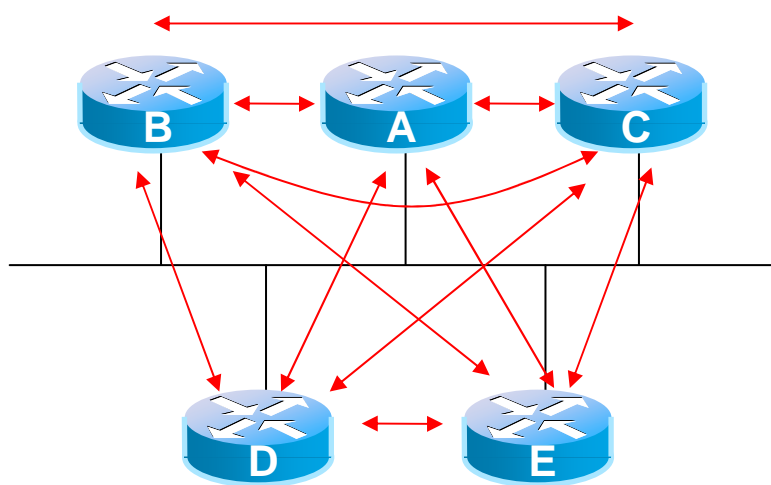
**What information is flood**

sonnh, 3/8/2008

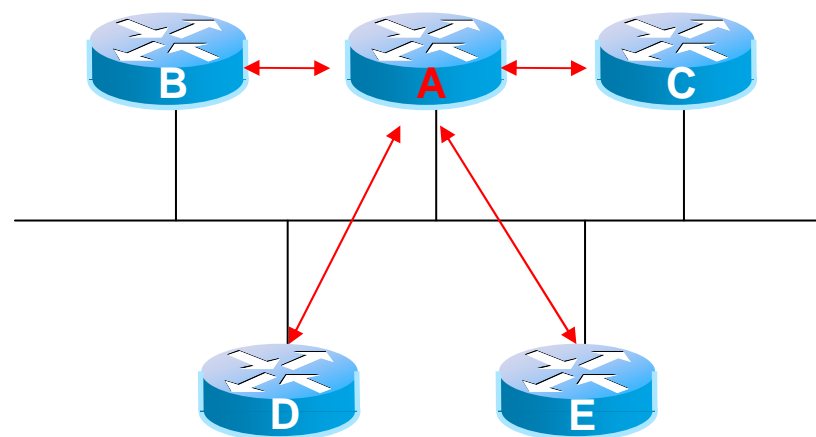


# Router đại diện - DR

- Để tăng hiệu quả của việc quảng bá LSA
- Mỗi router phải lập quan hệ với router đại diện - designated router (DR)
  - Trao đổi thông tin thông qua DR
  - DR dự phòng
- Chọn DR và BDR?



Không có DR



Có DR



# Neighbor & Adjacency



- “Neighbor” và “adjacency” là các k/n khác nhau!
  - Adjacency: có trao đổi thông tin
  - Neighbor: có đường nối trực tiếp
- Mạng quảng bá đa truy cập (e.g Ethernet)
  - Neighbor != Adjacency
- Mạng điểm-nối-điểm
  - Neighbor == Adjacency

**Slide 31**

---

**s10**

**Chang the order**  
sonnh, 3/8/2008



# RIP vs. OSPF

	RIP	OSPF
Đặc điểm	<ul style="list-style-type: none"> <li>• Router bình đẳng</li> <li>• Cấu hình dễ dàng</li> <li>• Mạng cỡ nhỏ</li> </ul>	<ul style="list-style-type: none"> <li>• Phân cấp</li> <li>• Cấu hình phức tạp</li> <li>• Mạng cỡ vừa và lớn</li> </ul>
Khả năng mở rộng	X	O
Độ phức tạp tính toán	Nhỏ	Lớn
Hội tụ	Chậm	Nhanh
Trao đổi thông tin	Bảng chọn đường	Trạng thái liên kết
Giải thuật	<b>Distant vector</b>	<b>Link-state</b>
Cập nhật hàng xóm	30s	10s (Hello packet)
Đơn vị chi phí	Số nút mạng	Băng thông

## Slide 32

---

**s11**

Exchanged informatio

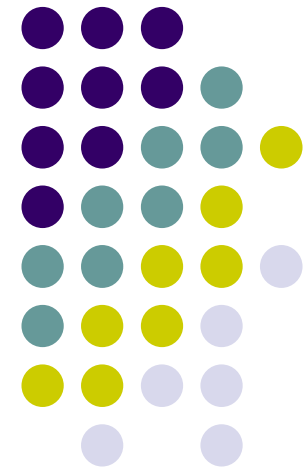
Rip:

OSPF

sonnh, 3/8/2008

# Giao thức chọn đường liên vùng

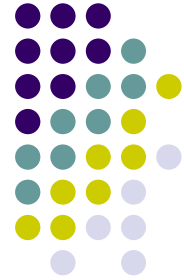
---



# BGP – Border Gateway Protocol

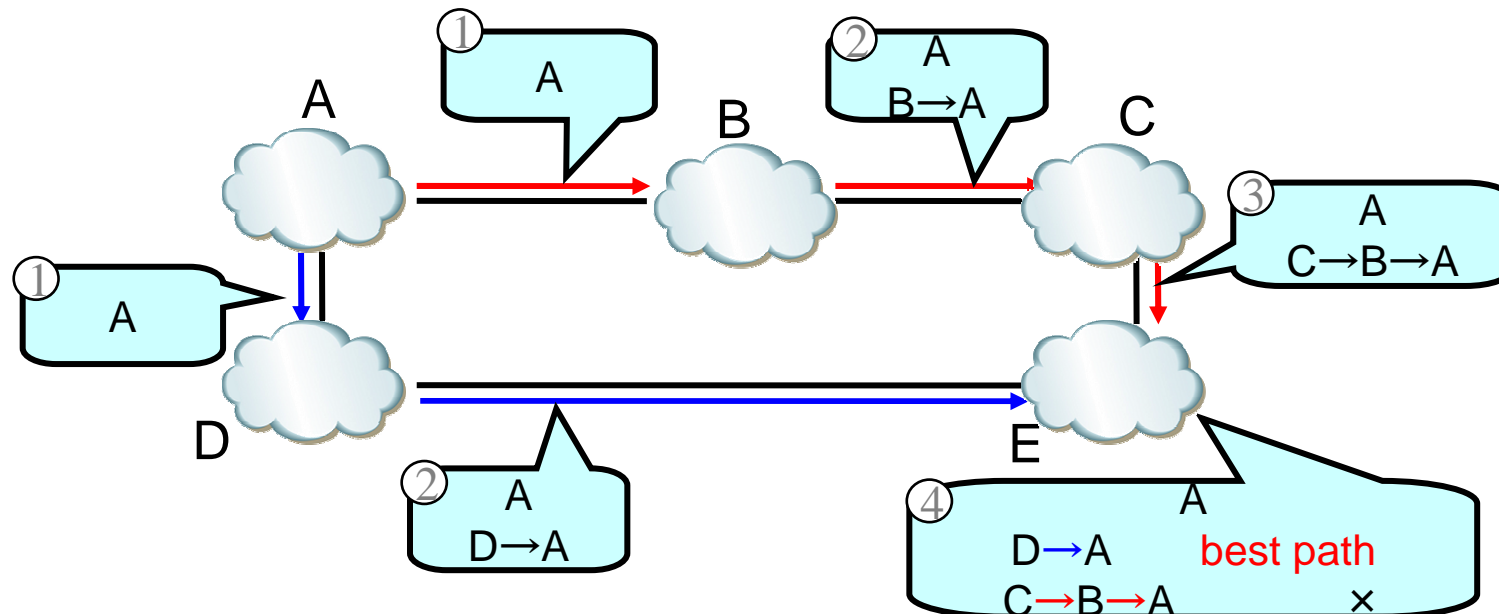


- Yếu tố gắn kết của Internet, kết nối các hệ tự trị
- Trao đổi thông tin NLRI (Network Layer Reachability Information)
  - Cho phép một AS biết được thông tin đi đến AS khác
  - Gửi thông tin này vào bên trong AS đó
  - Xác định đường đi tốt nhất dựa trên thông tin đó và các chính sách chọn đường
- Cho phép thiết lập các chính sách
  - Chọn đường ra
  - Quảng bá các đường vào



# BGP: Path vector routing

- Giữa các AS nên dùng giao thức nào?
  - Khó có một chính sách và đơn vị chi phí chung
  - LS: Chi phí không đồng nhất, CSDL quá lớn
  - DV: Mạng quá rộng, khó hội tụ
- Giải pháp: Chọn đường theo path-vector



**Slide 35**

---

**s12**

Change the symbol of router into AS

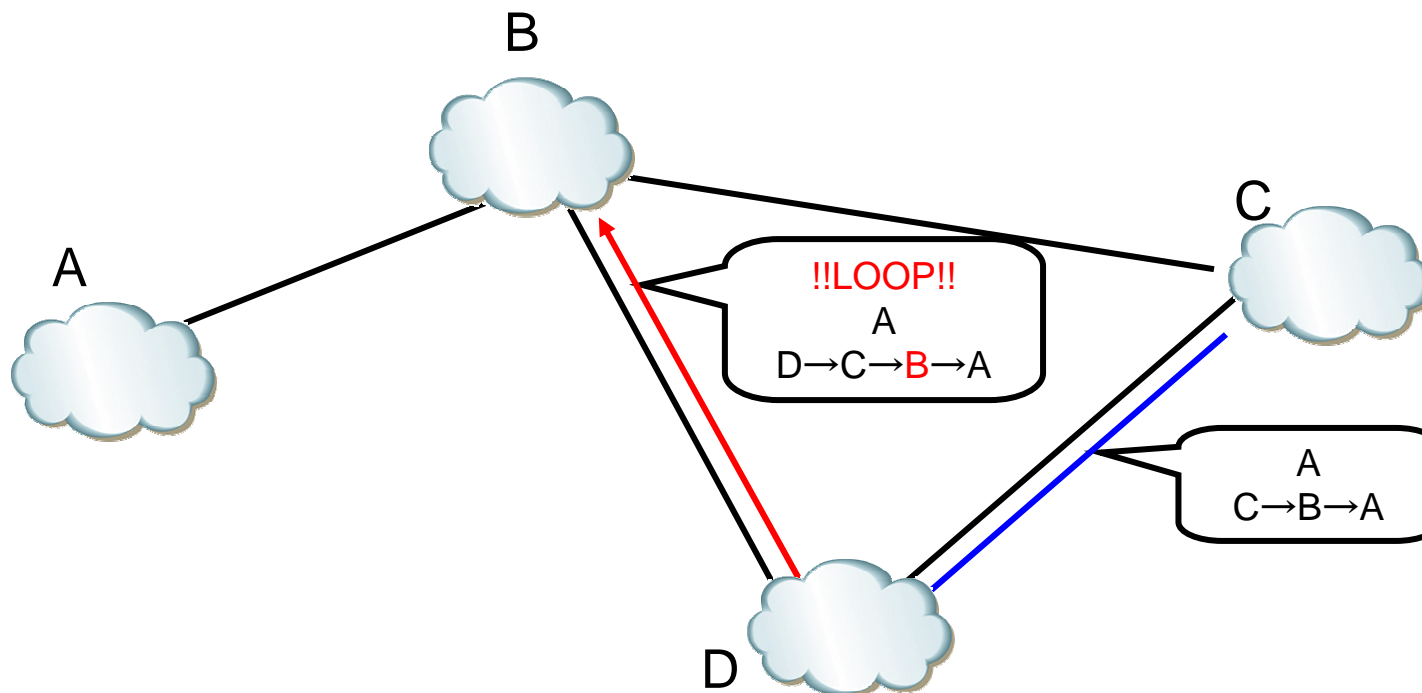
sonnh, 2/29/2008

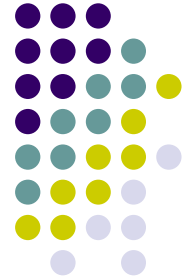


# Cơ chế tránh vòng lặp



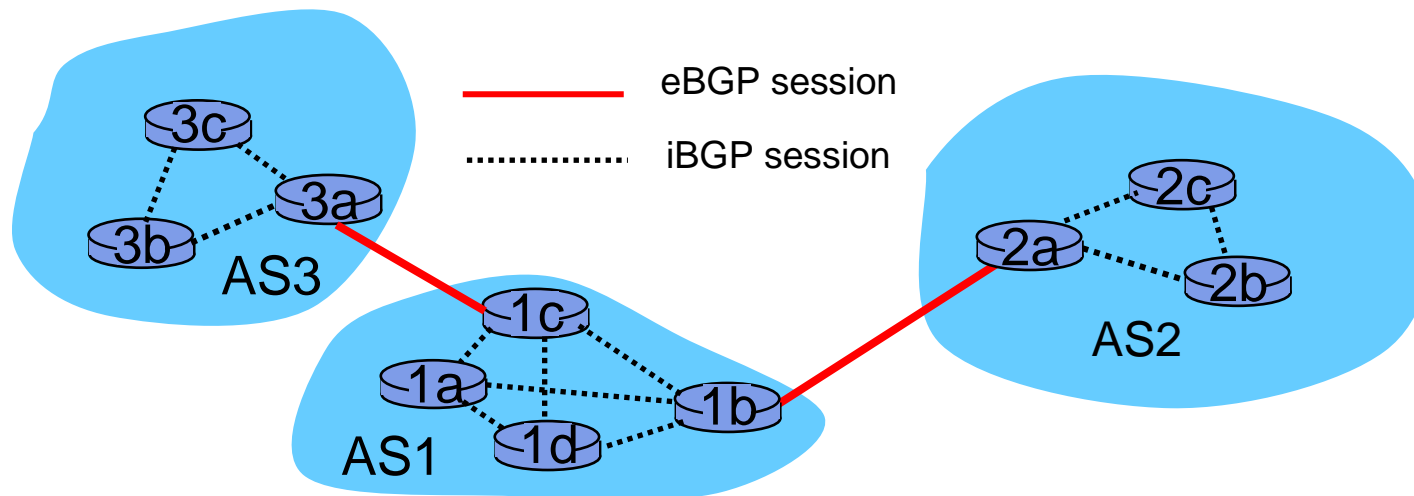
- Dò lại xem router đã có trên path-vector hay chưa
  - B hủy đường đi tới A





# eBGP và iBGP

- External BGP vs. Internal BGP
- Phân tán thông tin chọn đường
  1. 3a gửi tới 1c bằng
  2. 1c gửi thông tin nội bộ tới (1b, 1d, ...) trong AS1 bằng iBGP
  3. 2a nhận thông tin từ 1b bằng eBGP



## Slide 37

---

**s13**

- We have to explain why we have two kind of BGP

(Or we ask student)

- iBGP: internal

-e: external

sonnh, 2/29/2008

# BGP có thể cài đặt các policy



- Khi các router gửi và nhận thông tin chọn đường
  - BGP có thể đặt các chính sách
    - Cho đường vào
    - Cho đường ra



# Các thuộc tính của đường đi

- **ORIGIN**
  - Nguồn của thông tin (IGP/EGP/incomplete)
- **AS\_PATH**
- **NEXT\_HOP**
- **MED (MULTI\_EXIT\_DISCRIMINATOR)**
- **LOCAL\_PREF**
- **ATOMIC\_AGGREGATE**
- **AGGREGATOR**
- **COMMUNITY**

## Slide 39

---

**s15**

Add a new slide to explain about policy  
Explain only attributes here  
and then, explain the step

Put the slide 103 of Keio slide just before slide about policy: howto implement policy

Add AS prepend just after filtering  
and others come following  
sonnh, 2/29/2008



# Các bước chọn đường đi

- Bước 1: NEXT\_HOP?
- Bước 2: So sánh LOCAL\_PREF
- Bước 3: So sánh độ dài AS\_PATH
- Bước 4: So sánh ORIGIN
- Bước 5: So sánh MED
- Bước 6: So sánh EBGP/IBGP
- Bước 7: So sánh chi phí tới NEXT\_HOP
- Bước 8: So sánh Router ID

**Slide 40**

---

**s14**

**Priority of route selection**

sonnh, 2/29/2008



# Cài đặt các chính sách ntn?

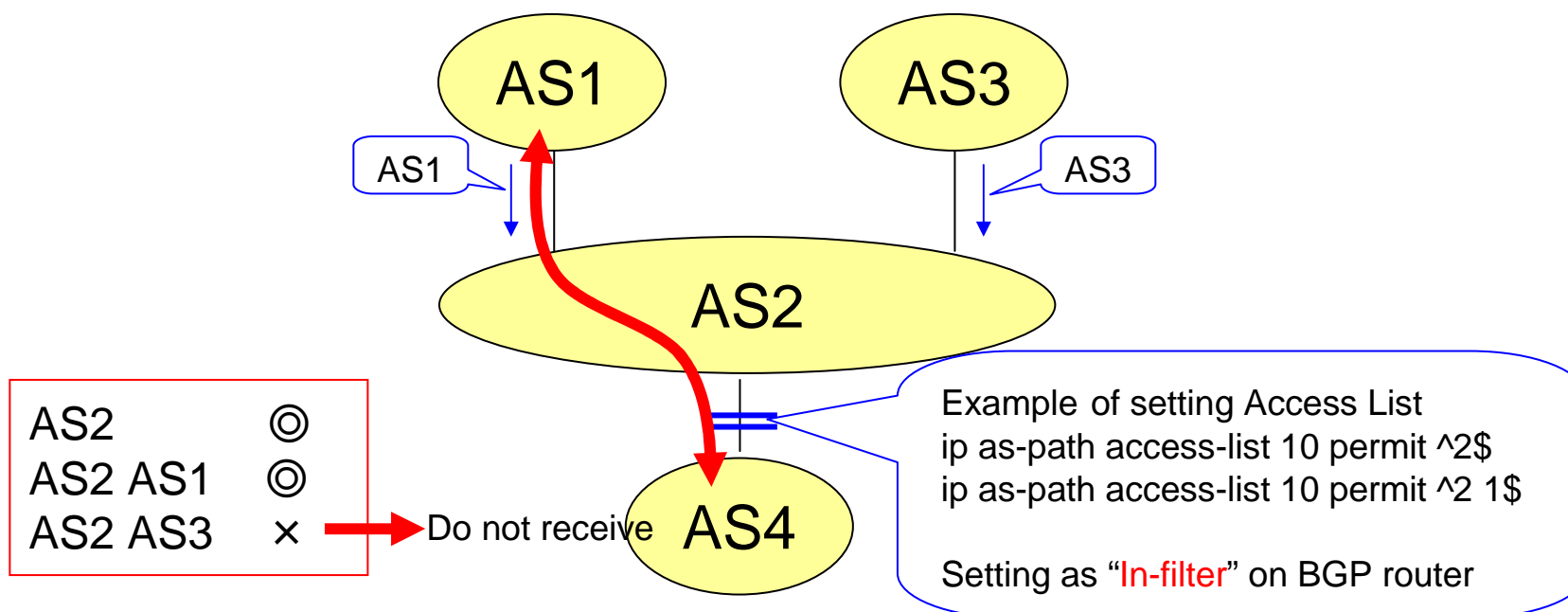


- Bộ lọc
- Các thuộc tính
  - AS\_PATH PREPEND
  - MED
  - LOCAL\_PREF



# Bộ lọc

- Chỉ trao đổi đường đi nào đã được đăng ký
  - Hạn chế thông qua bộ lọc vào - In-filter
  - Thông qua bộ lọc ra - Out-filter



**Slide 42**

---

**s16**

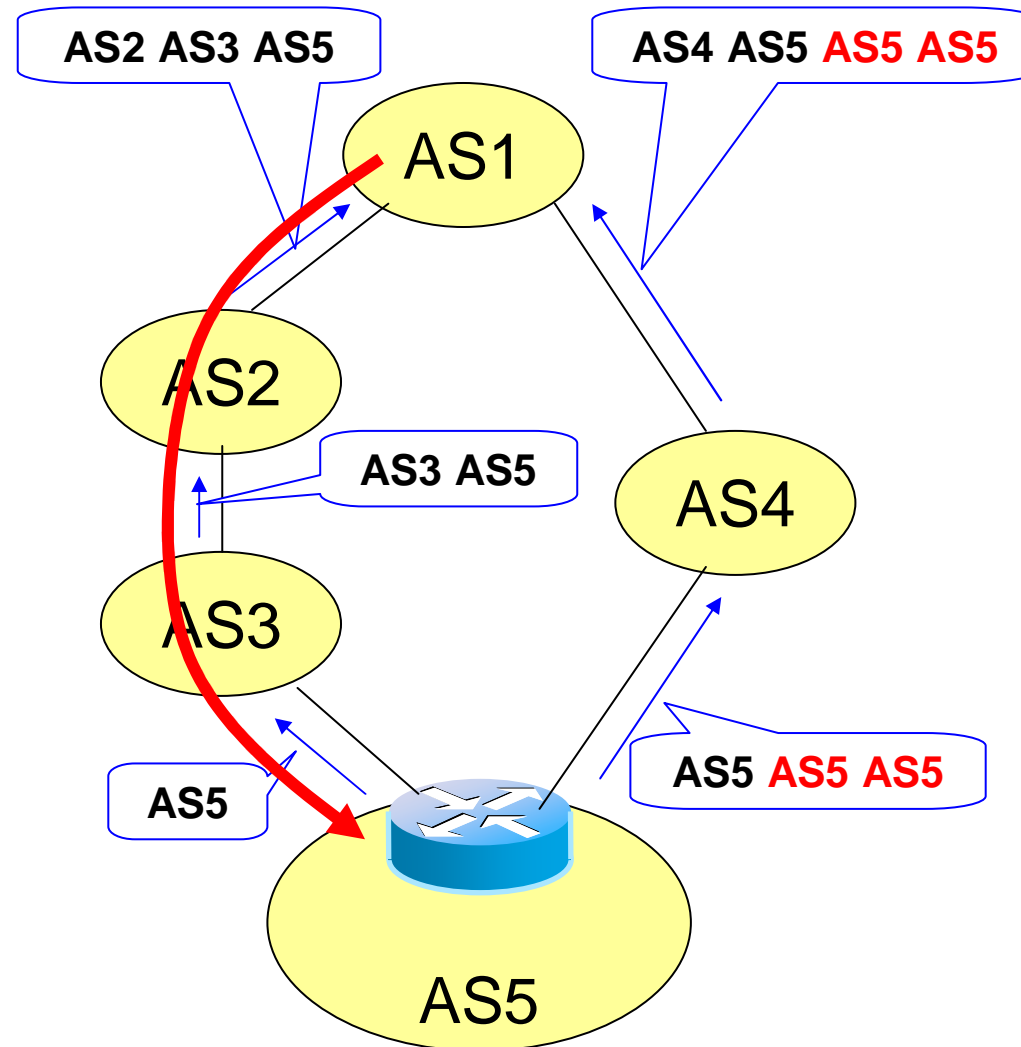
Move just after policy  
sonnh, 2/29/2008

# Ví dụ về AS PATH



<b>Network</b>	<b>Next Hop</b>	<b>Metric</b>	<b>LocPrf</b>	<b>Weight</b>	<b>Path</b>
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

# Chọn đường với AS\_PATH Prepend



# Ví dụ về AS PATH prepend

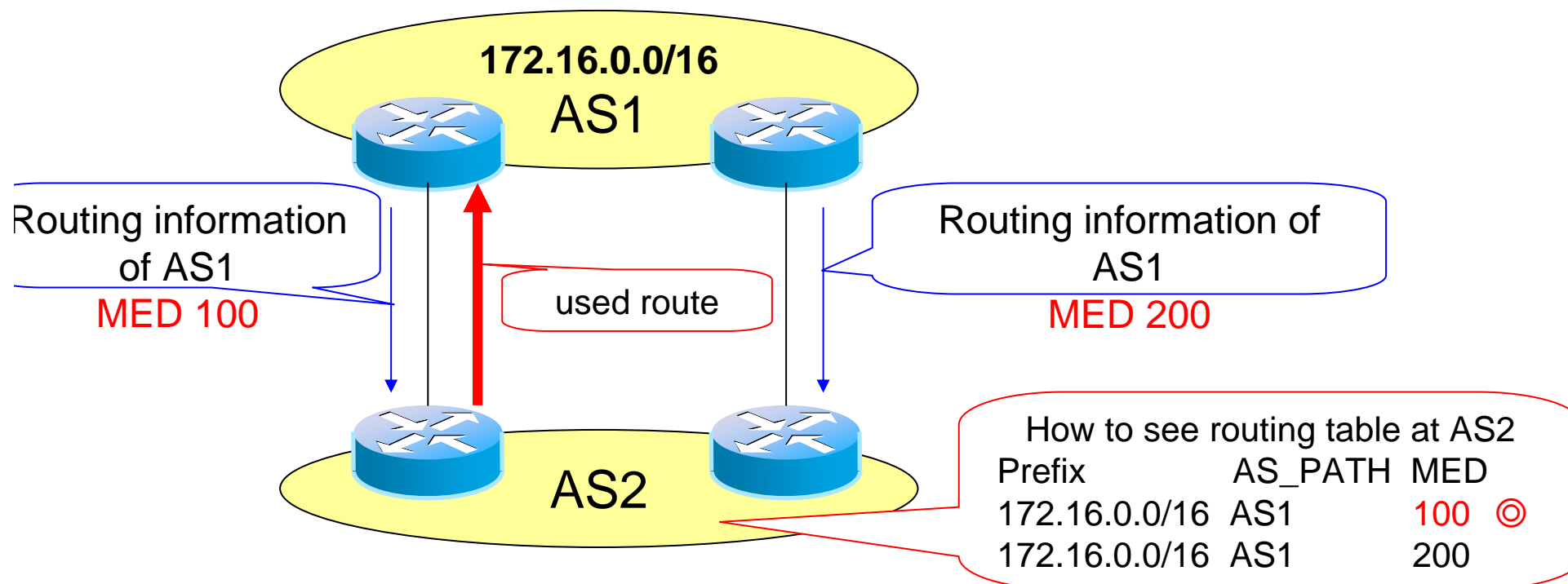


Network	Next Hop	Metric	LocPrf	Weight	Path
8.5.192.0/22	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
8.5.196.0/24	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14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



# Chọn đường với MED

- trong trường hợp 2 AS với nhiều link
- Chọn MED nhỏ hơn
- Áp dụng trong điều khiển lưu lượng



## Slide 46

---

**s17**

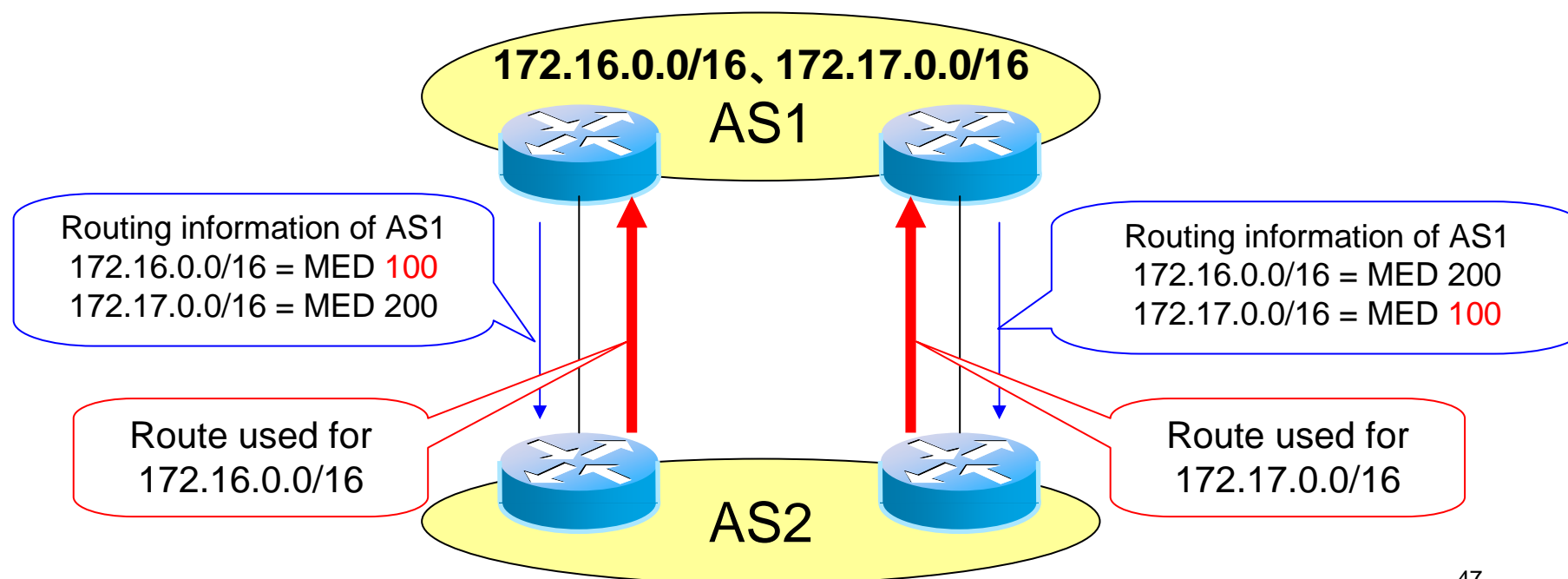
MED is set by distance AS  
Local-pref set by local AS  
so local-pre is higher  
sonnh, 2/29/2008





# Phân tải với MED

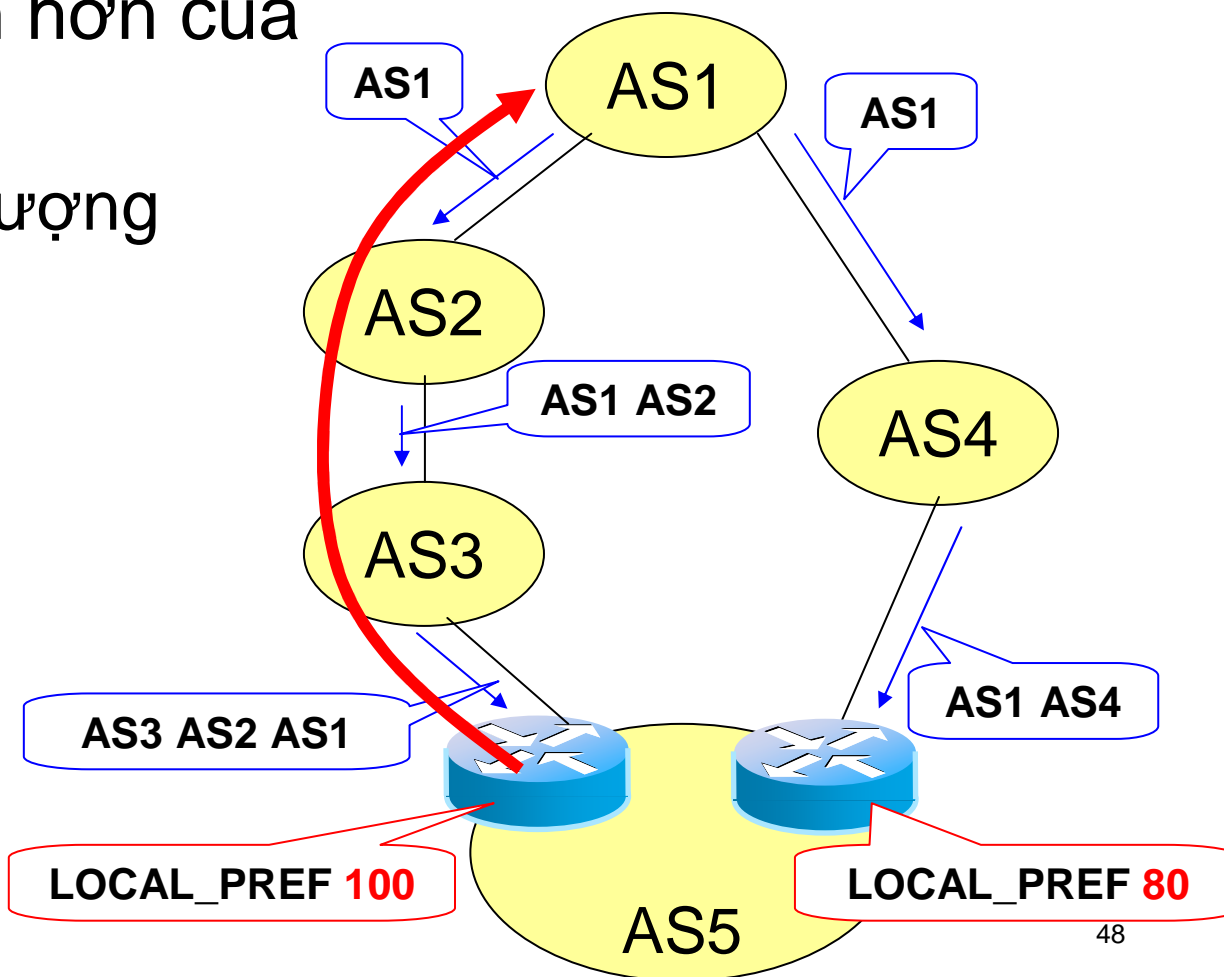
- Đặt giá trị MED khác nhau cho mỗi đường
- Cũng điều khiển lưu lượng





# Sử dụng LOCAL\_PREF

- Chọn giá trị lớn hơn của LOCAL\_PREF
- Điều khiển lưu lượng upbound



# Tóm tắt

- Hierarchical routing
- RIP
- OSPF
- BGP



# Tuần tới: Transport Layer

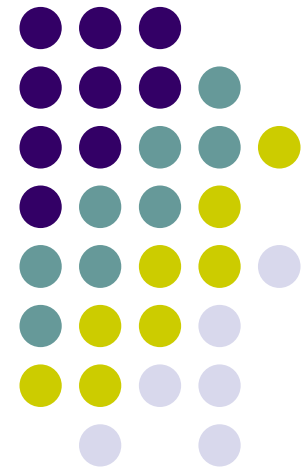


- Nguyên lý tầng giao vận
- UDP và TCP
- Điều khiển luồng
- Điều khiển tắc nghẽn

# Chương 6: Tầng giao vận

Giảng viên: Ngô Hồng Sơn

Bộ môn Truyền thông và Mạng máy tính  
Khoa CNTT- ĐHBK Hà Nội





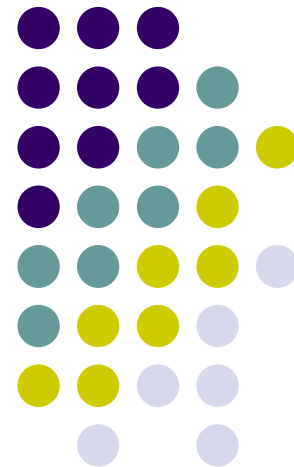
# Tổng quan

- Các tuần trước : Giao thức IP
  - Địa chỉ, gói tin IP
  - ICMP
  - Chọn đường
- Hôm nay: Tầng giao vận
  - Nguyên lý tầng giao vận
  - Giao thức UDP
  - Giao thức TCP

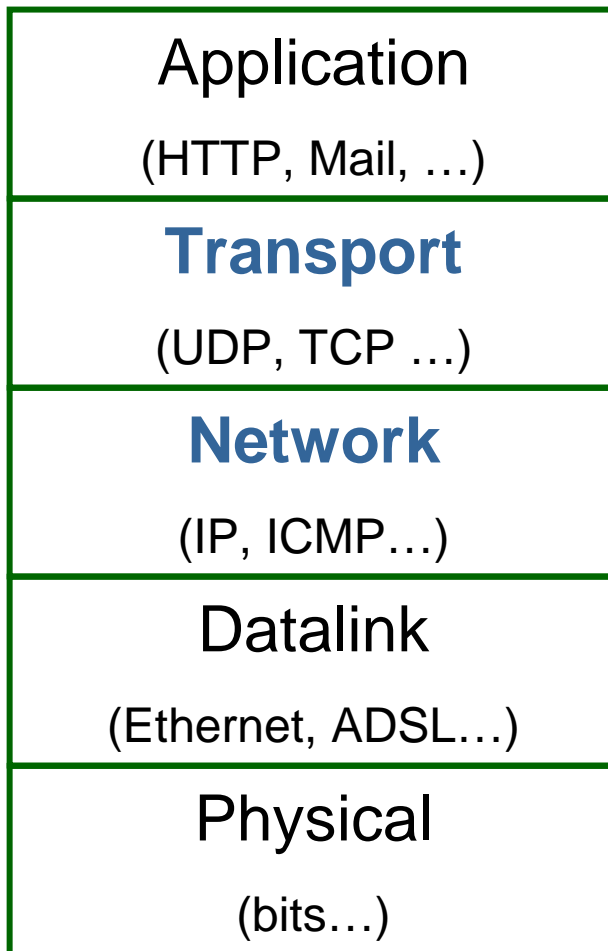
# Các khái niệm cơ bản

---

Nhắc lại kiến trúc phân tầng  
Hướng liên kết vs. Không liên kết  
UDP & TCP



# Nhắc lại về kiến trúc phân tầng



Hỗ trợ các ứng dụng trên mạng

**Truyền dữ liệu giữa các ứng dụng**

Chọn đường và chuyển tiếp gói tin giữa các máy, các mạng

Hỗ trợ việc truyền thông cho các thành phần kế tiếp trên cùng 1 mạng

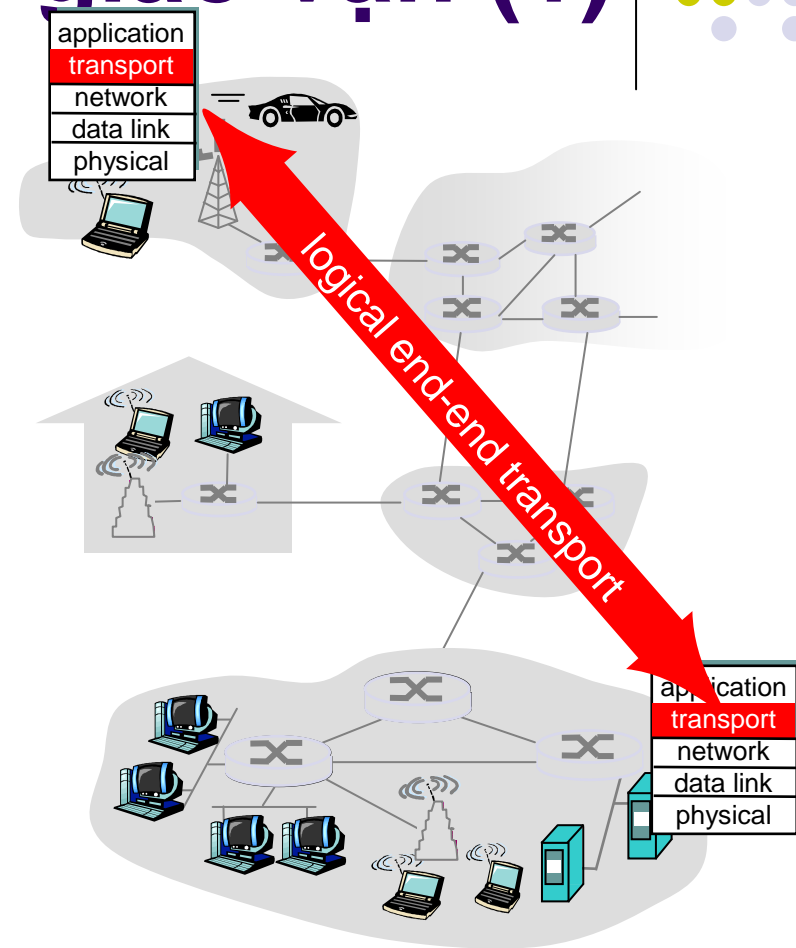
Truyền và nhận dòng bit trên đường truyền vật lý



# Tổng quan về tầng giao vận (1)



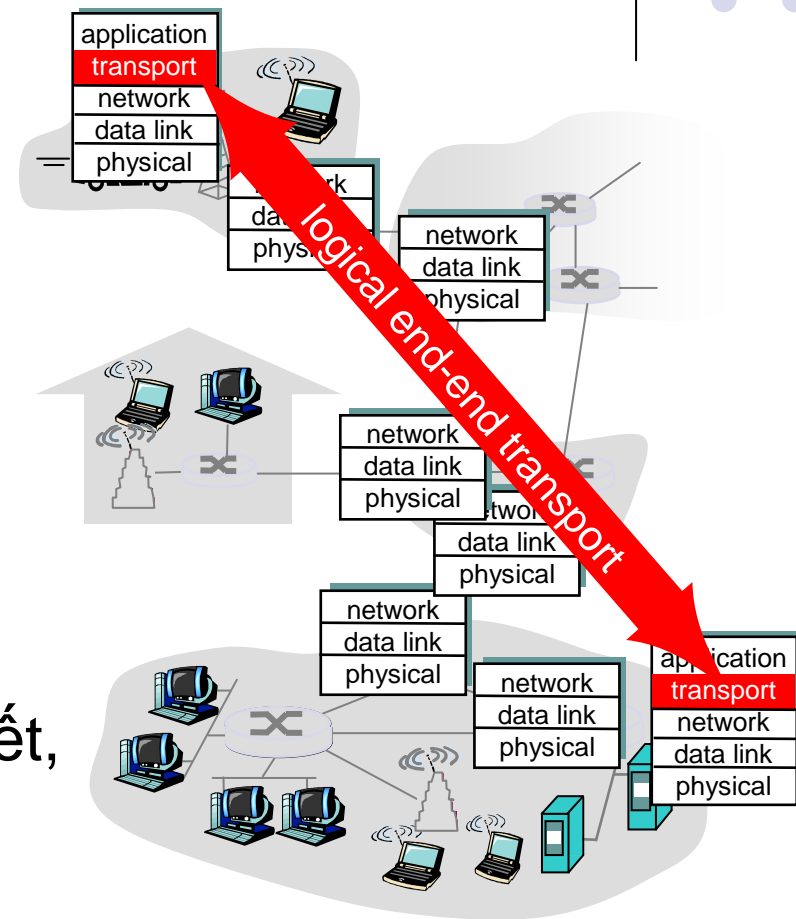
- Cung cấp phương tiện truyền giữa các ứng dụng cuối
- Bên gửi:
  - Nhận dữ liệu từ ứng dụng
  - Đặt dữ liệu vào các đoạn tin và chuyển cho tầng mạng
  - Nếu dữ liệu quá lớn, nó sẽ được chia làm nhiều phần và đặt vào nhiều đoạn tin khác nhau
- Bên nhận:
  - Nhận các đoạn tin từ tầng mạng
  - Tập hợp dữ liệu và chuyển lên cho ứng dụng



# Tổng quan về tầng giao vận (2)



- Được cài đặt trên các hệ thống cuối
  - Không cài đặt trên các routers, switches...
- Hai dạng dịch vụ giao vận
  - Tin cậy, hướng liên kết, e.g. TCP
  - Không tin cậy, không liên kết, e.g. UDP





# Tại sao lại cần 2 loại dịch vụ?

- Các yêu cầu đến từ tầng ứng dụng là đa dạng
- Các ứng dụng cần dịch vụ với 100% độ tin cậy như mail, web...
  - Sử dụng dịch vụ của TCP
- Các ứng dụng cần chuyển dữ liệu nhanh, có khả năng chịu lỗi, e.g. VoIP, Video Streaming
  - Sử dụng dịch vụ của UDP



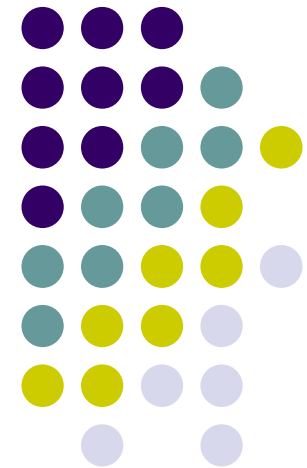
# Ứng dụng và dịch vụ giao vận

<b>Ứng dụng</b>	<b>Giao thức ứng dụng</b>	<b>Giao thức giao vận</b>
e-mail	SMTP	TCP
remote terminal access	Telnet	TCP
Web	HTTP	TCP
file transfer	FTP	TCP
streaming multimedia	giao thức riêng (e.g. RealNetworks)	TCP or UDP
Internet telephony	giao thức riêng (e.g., Vonage, Dialpad)	thường là UDP

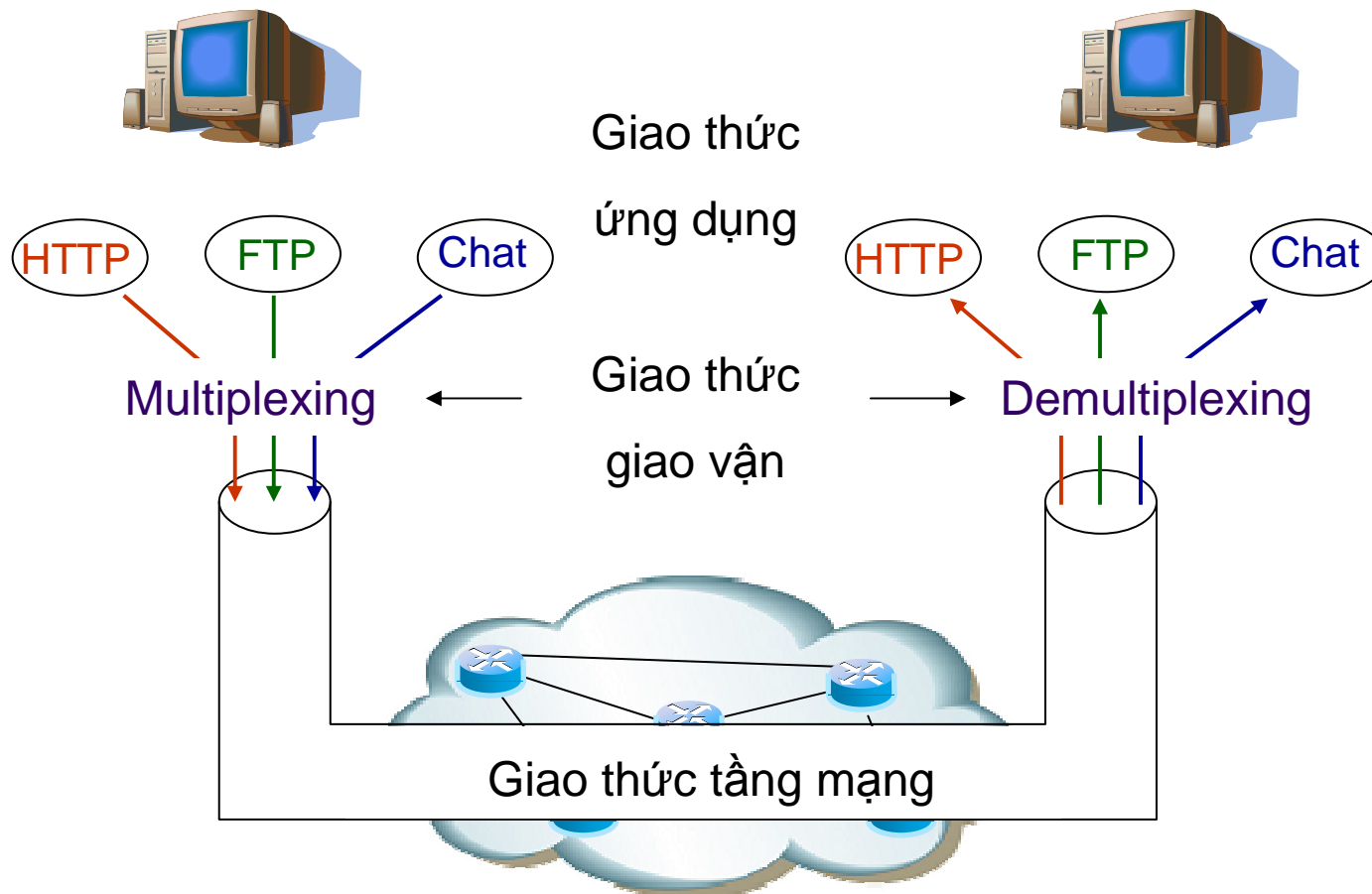
# Các chức năng chung

---

Dồn kênh/phân kênh  
Mã kiểm soát lỗi



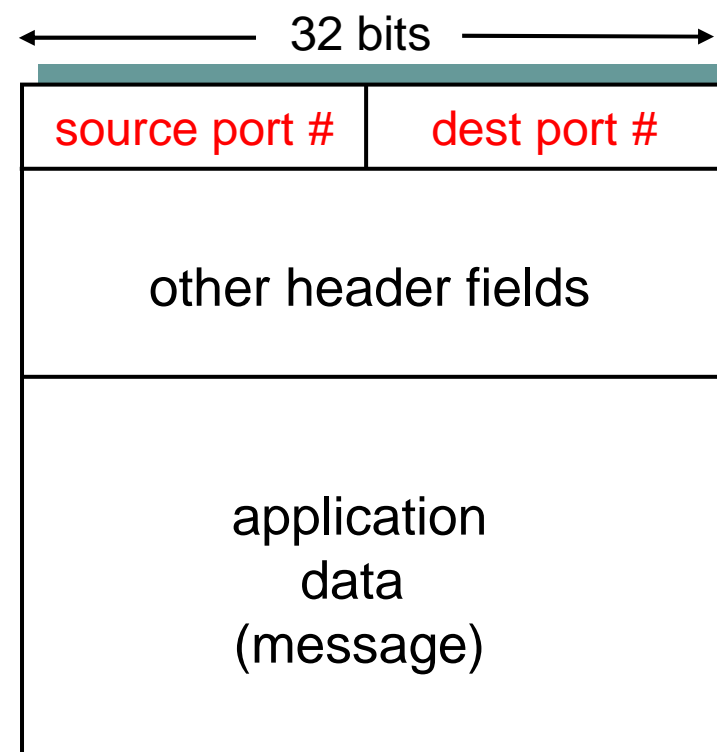
# Dồn kênh/phân kênh - Mux/Demux





# Mux/Demux hoạt động ntn?

- Tại tầng mạng, gói tin IP được định danh bởi địa chỉ IP
  - Để xác định máy trạm
- Làm thế nào để phân biệt các ứng dụng trên cùng một máy?
  - Sử dụng số hiệu cổng (16 bits)
  - Mỗi tiến trình ứng dụng được gán 1 cổng
- **Socket**: Một cặp địa chỉ IP và số hiệu cổng



TCP/UDP segment format



# Checksum

- Phát hiện lỗi bit trong các đoạn tin/gói tin
- Nguyên lý giống như checksum (16 bits) của giao thức IP
- Ví dụ:

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
Tổng	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
Checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

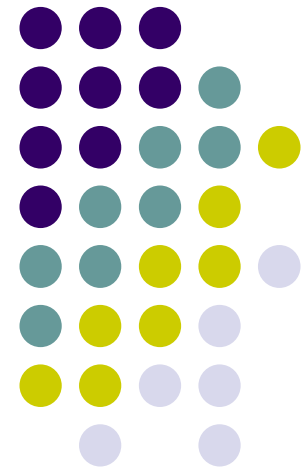


# UDP

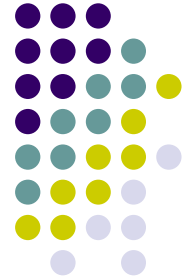
## User Datagram Protocol

---

Tổng quan  
Khuôn dạng gói tin



# Giao thức dạng “Best effort”



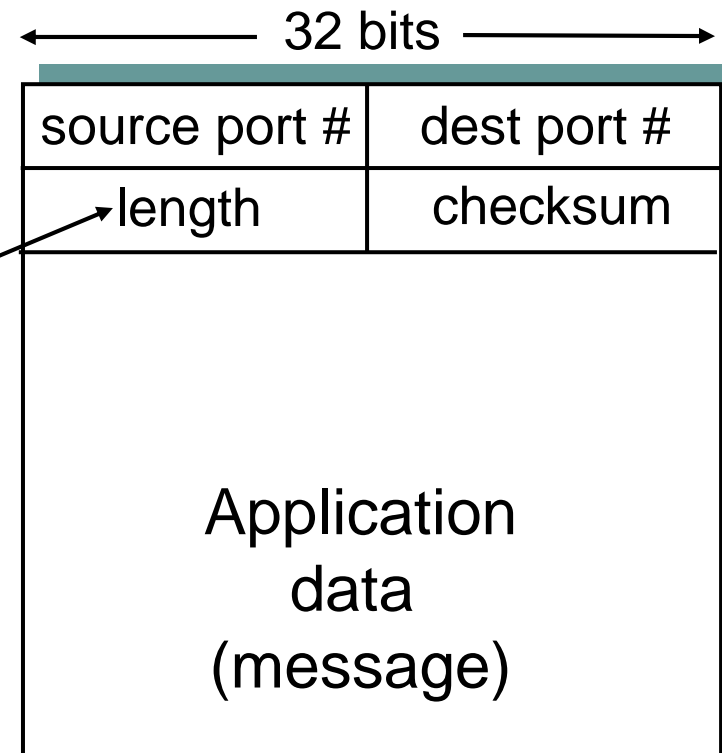
- Vì sao cần UDP?
  - Không cần thiết lập liên kết (tăng độ trễ)
  - Đơn giản: Không cần lưu lại trạng thái liên kết ở bên gửi và bên nhận
  - Phần đầu đoạn tin nhỏ
  - Không có quản lý tắc nghẽn: UDP cứ gửi dữ liệu nhanh nhất, nhiều nhất nếu có thể
- UDP có những chức năng cơ bản gì?
  - Dồn kênh/phân kênh
  - Phát hiện lỗi bit bằng checksum

# Khuôn dạng bức tin (datagram)



- UDP sử dụng đơn vị dữ liệu gọi là – datagram (bức tin)

Độ dài toàn bộ bức tin tính theo byte



Khuôn dạng đơn vị dữ liệu của UDP

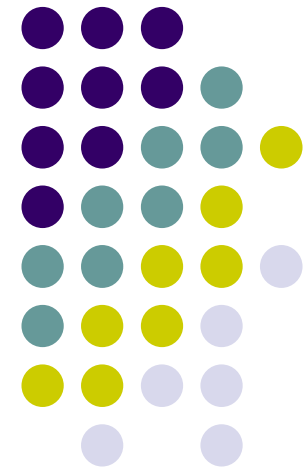


# Các vấn đề của UDP

- Không có kiểm soát tắc nghẽn
  - Làm Internet bị quá tải
- Không bảo đảm được độ tin cậy
  - Các ứng dụng phải cài đặt cơ chế tự kiểm soát độ tin cậy
  - Việc phát triển ứng dụng sẽ phức tạp hơn

# Khái niệm về truyền thông tin cậy

---

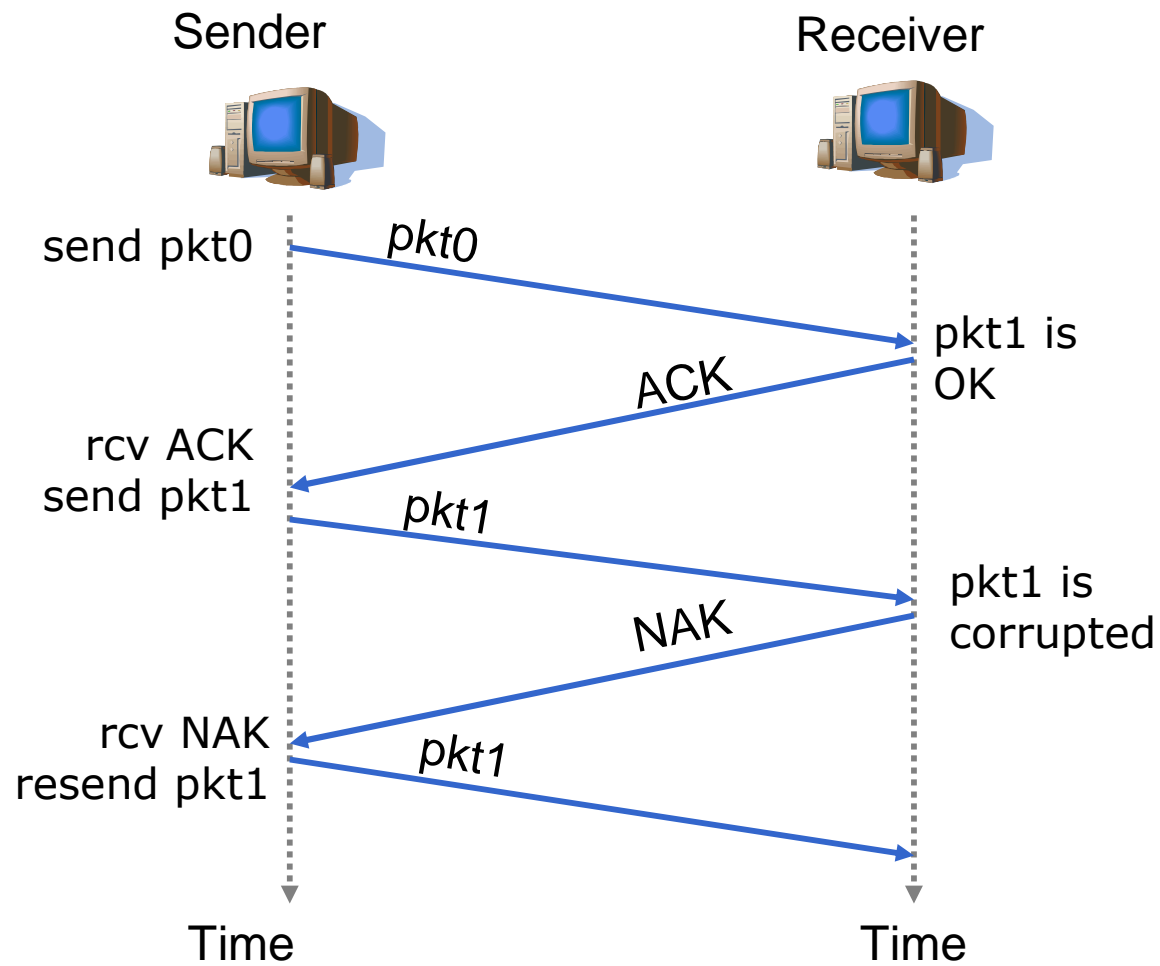
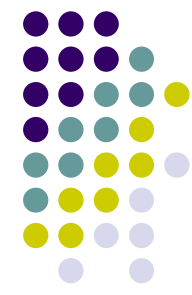


# Kênh có lỗi bit, không bị mất tin

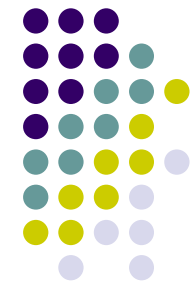


- Phát hiện lỗi?
  - Checksum
- Làm thế nào để báo cho bên gửi?
  - ACK (*acknowledgements*):
  - NAK (*negative acknowledgements*): báo cho bên nhận rằng pkt bị lỗi
- Phản ứng của bên gửi?
  - Truyền lại nếu là NAK

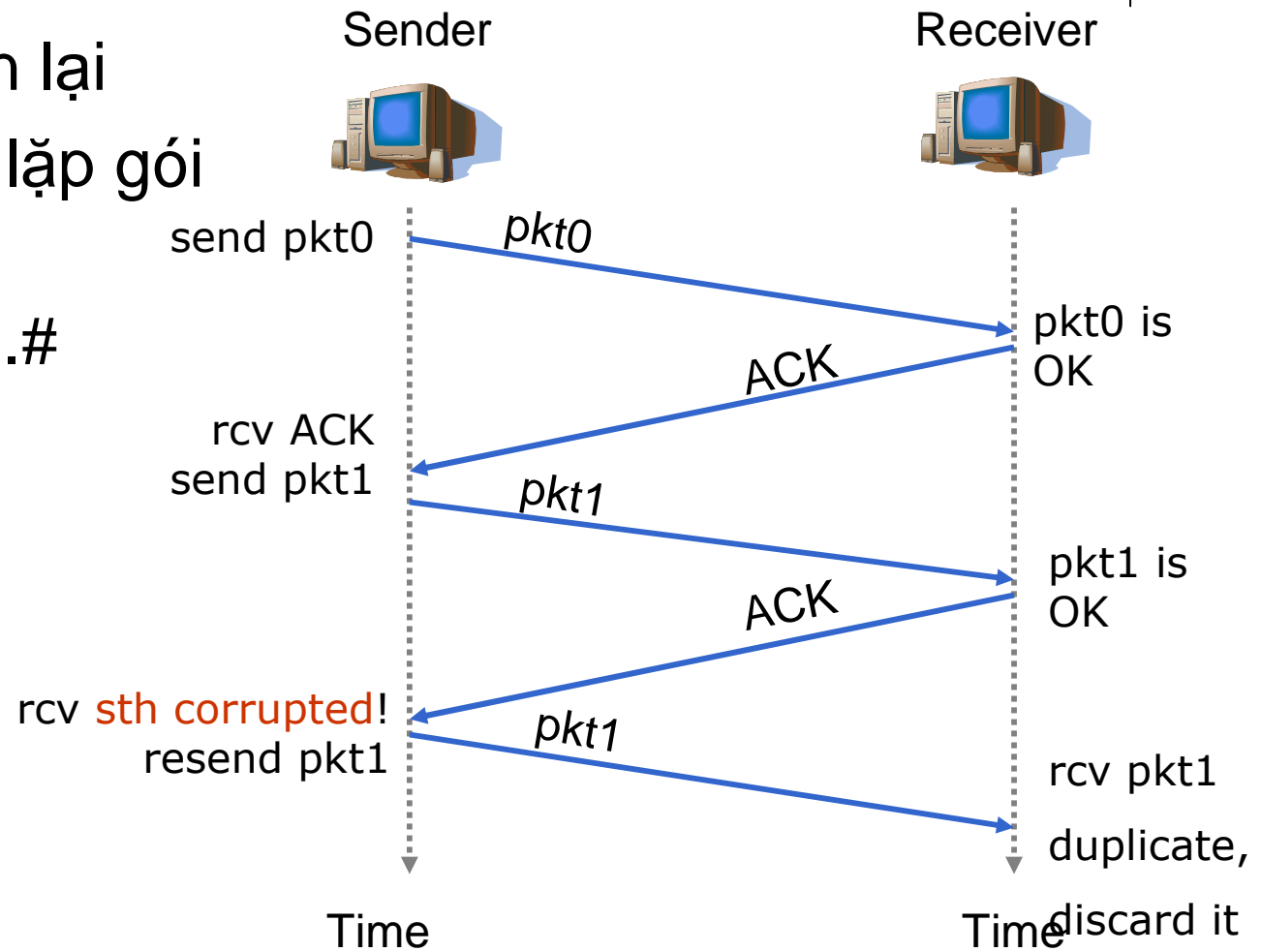
# Hoạt động



# Lỗi ACK/NAK

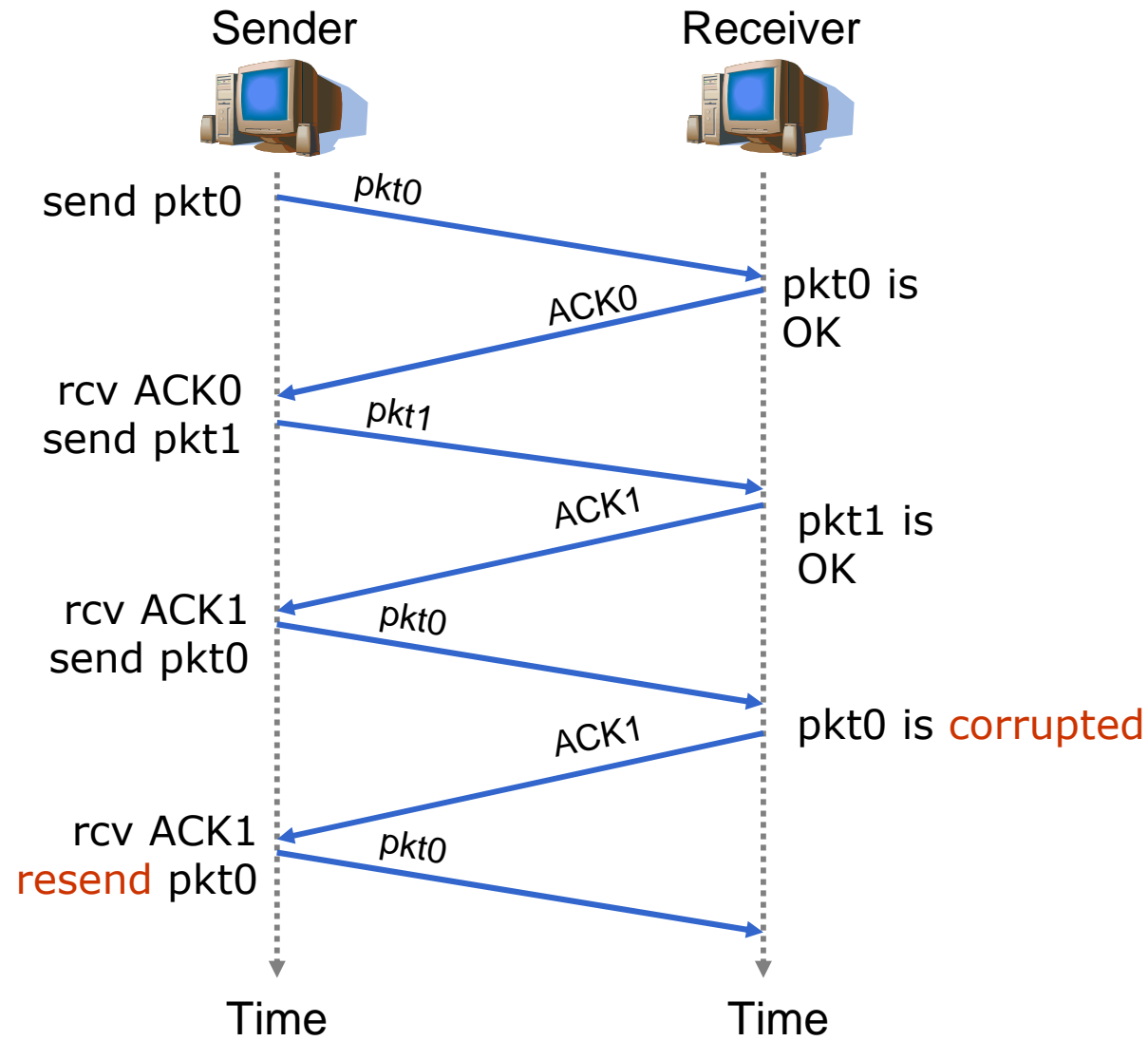
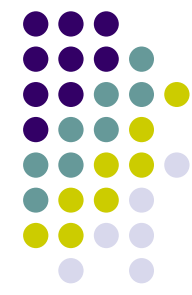


- Cần truyền lại
- Xử lý việc lặp gói tin ntn?
- Thêm Seq.#

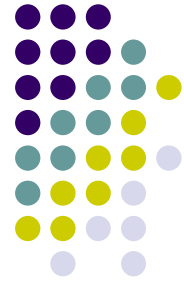




# Giải pháp không dùng NAK

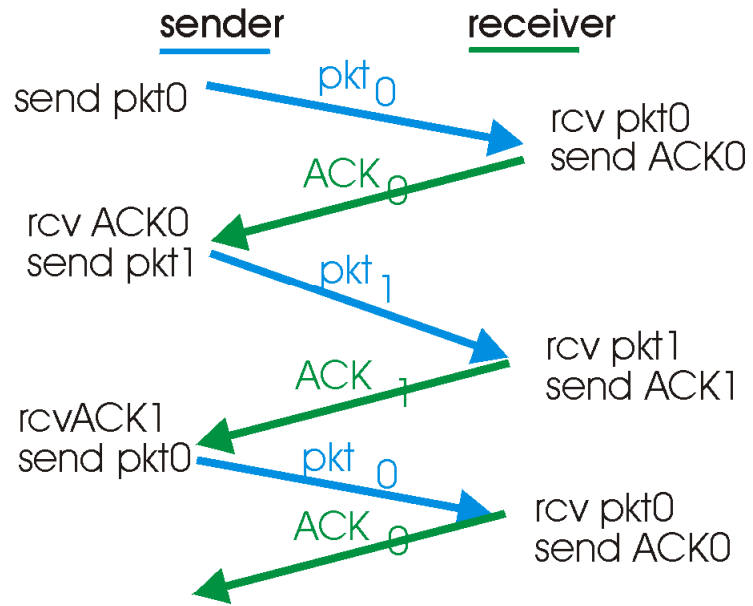


# Kênh có lỗi bit và mất gói tin

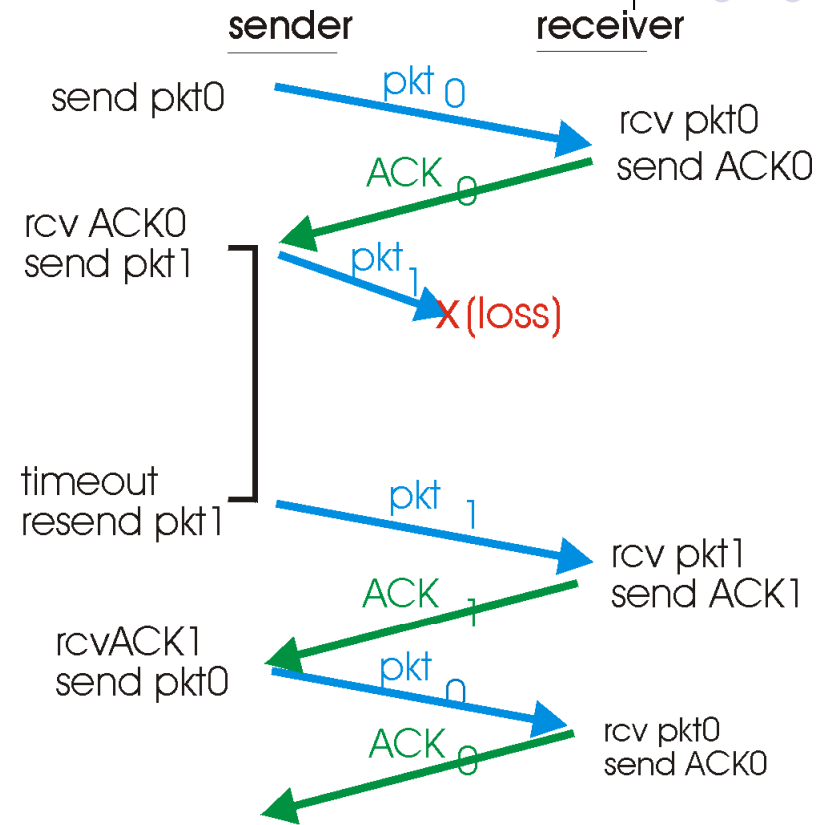


- Dữ liệu và ACK có thể bị mất
  - Nếu không nhận được ACK?
  - Truyền lại như thế nào?
  - Timeout!
- Thời gian chờ là bao lâu?
  - Ít nhất là 1 RTT (Round Trip Time)
  - Mỗi gói tin gửi đi cần 1 timer
- Nếu gói tin vẫn đến đích và ACK bị mất?
  - Dùng số hiệu gói tin

# Minh họa

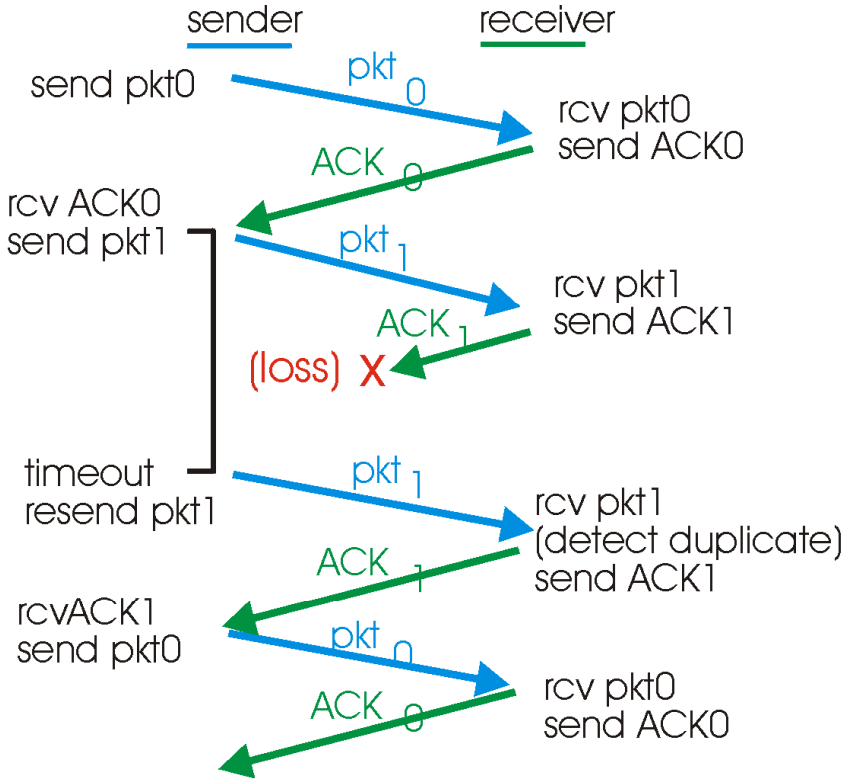


(a) operation with no loss

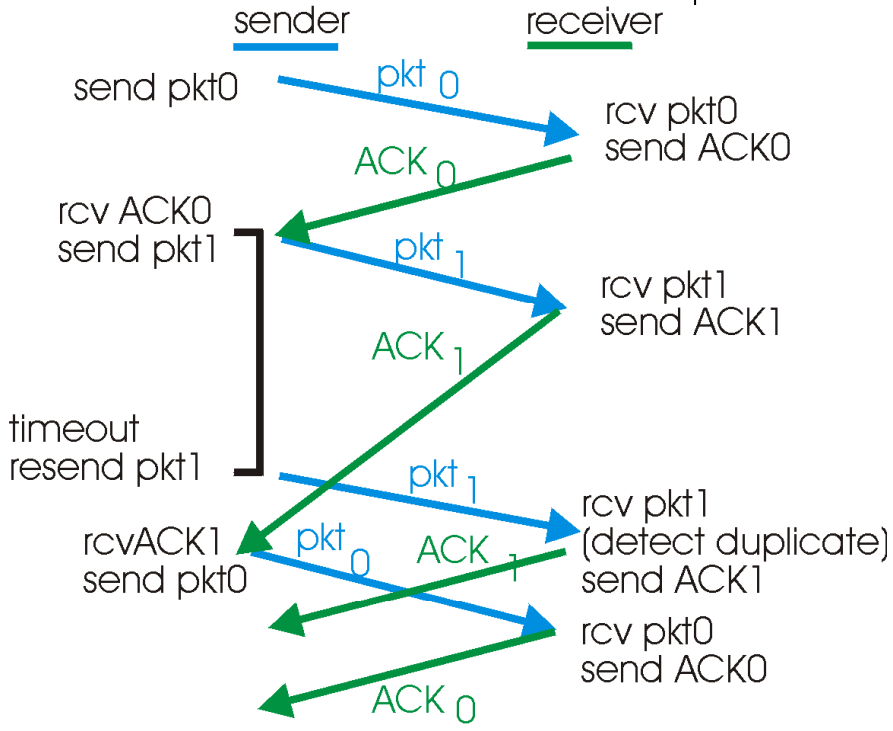


(b) lost packet

# Minh họa

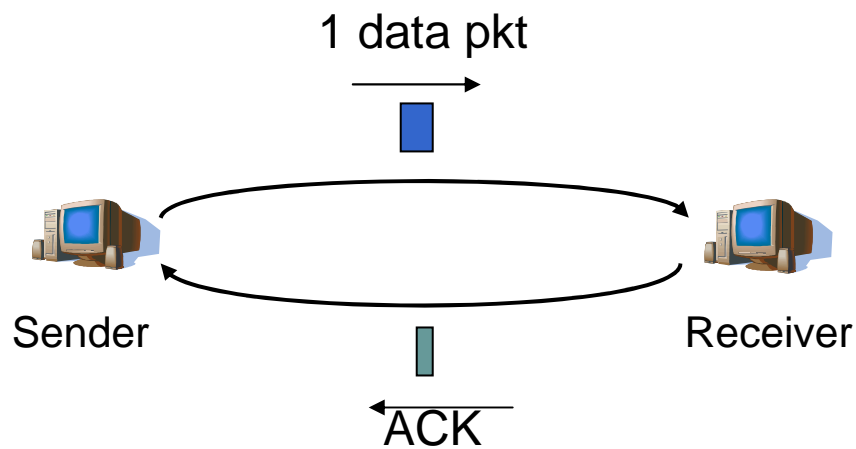


(c) lost ACK

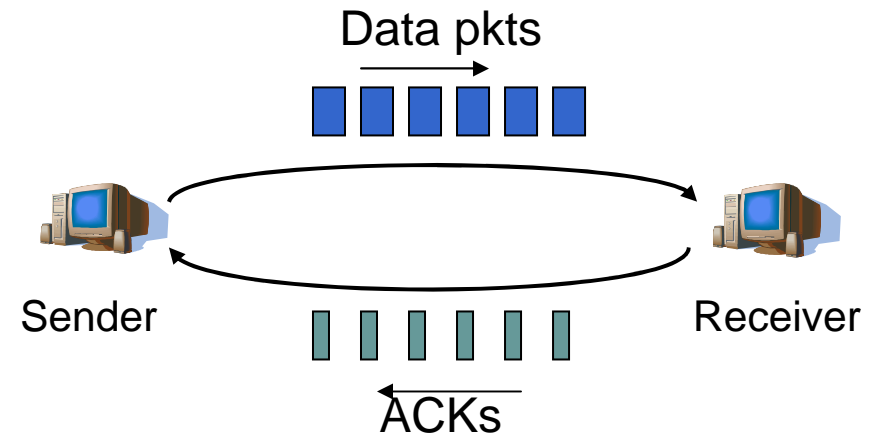


(d) premature timeout

# Truyền theo kiểu pipeline



**stop-and-wait**

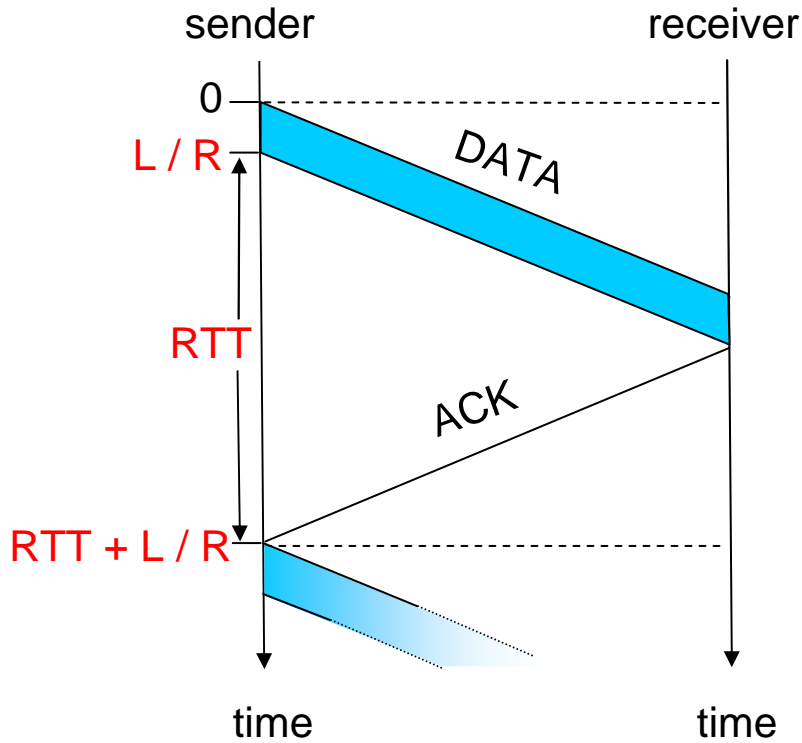


**Pipeline**

# So sánh hiệu quả



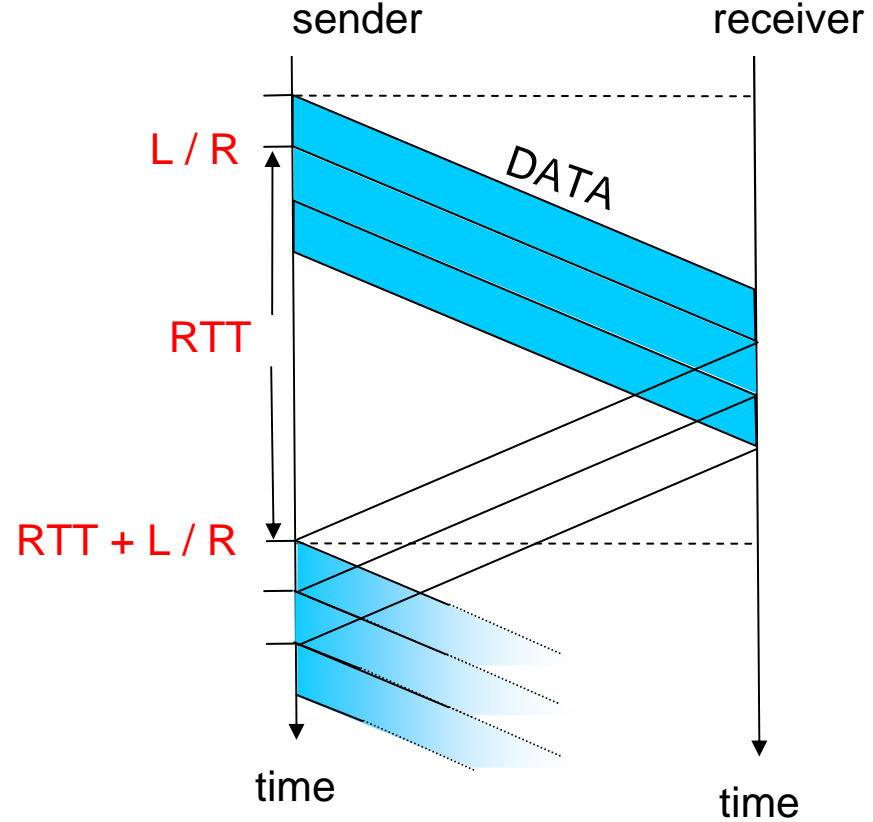
## stop-and-wait



L: Size of data pkt  
 R: Link bandwidth  
 RTT: Round trip time

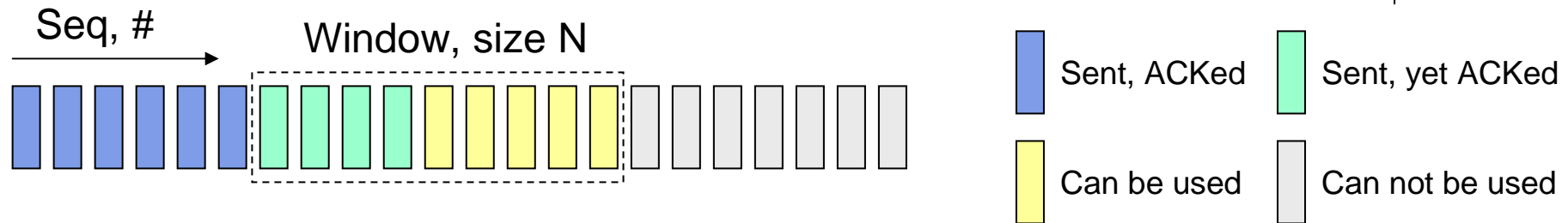
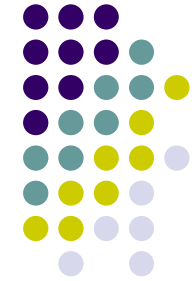
$$\text{Performance} = \frac{L/R}{RTT + L/R}$$

## Pipeline



$$\text{Performance} = \frac{3 * L/R}{RTT + L/R}$$

# Go-back-N



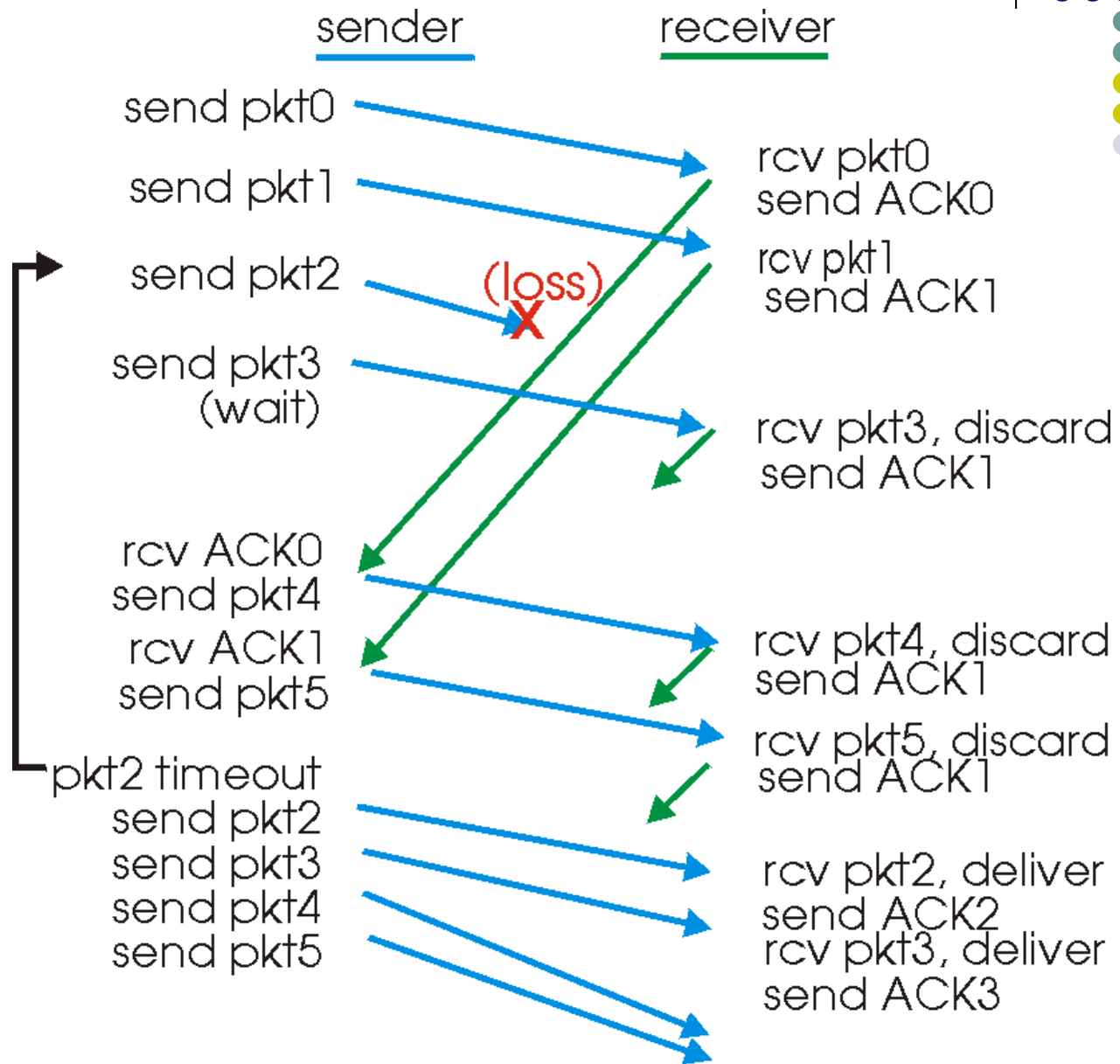
## Sender

- Chỉ gửi các gói tin với số hiệu trong cửa sổ, “dịch” cửa sổ sang phải mỗi khi nhận được ACK
- ACK(n): xác nhận cho các gói tin với số hiệu cho đến n
- Khi có timeout: truyền lại tất cả các gói tin có số hiệu lớn hơn n trong cửa sổ.

## Receiver

- Chỉ gửi 1 xác nhận ACK cho gói tin có số hiệu lớn nhất đã nhận được theo đúng thứ tự.
- Với các gói tin không theo thứ tự:
  - Hủy bỏ -> không lưu vào vùng đệm
  - Xác nhận lại gói tin với số hiệu lớn nhất còn đúng thứ tự

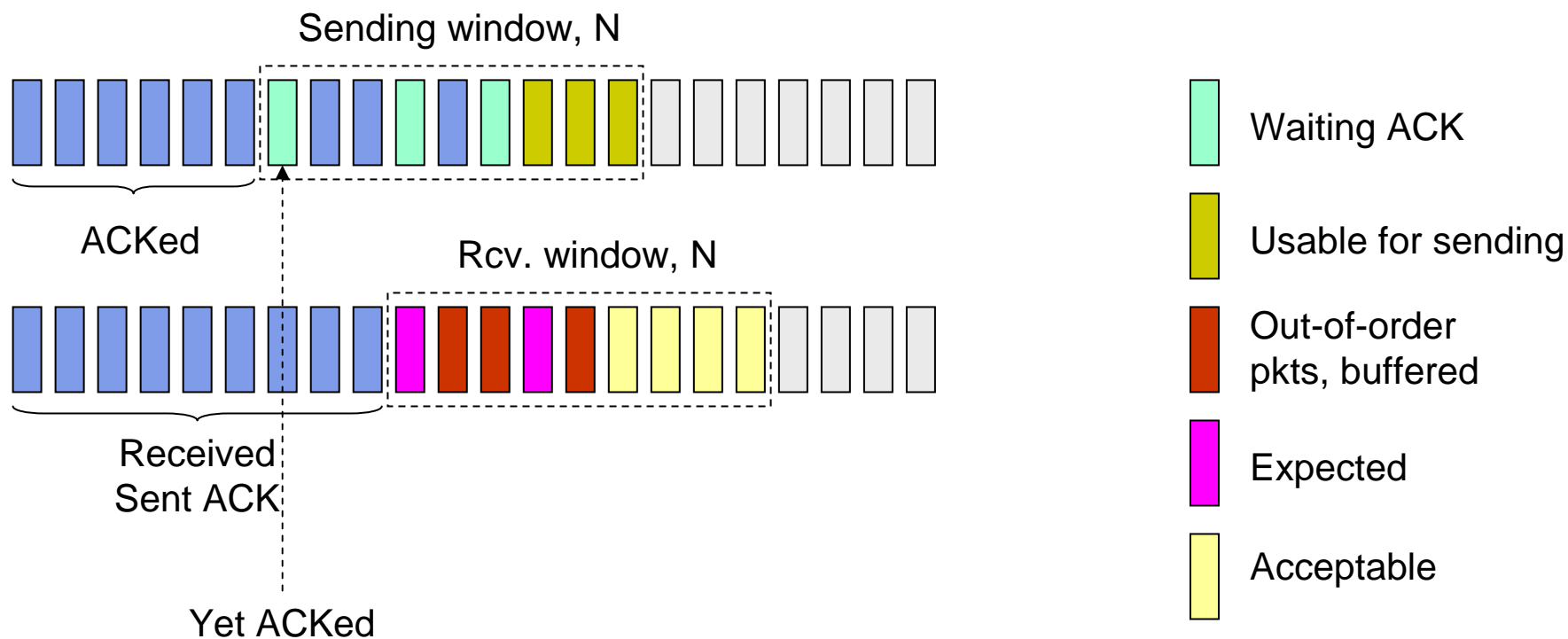
# Ví dụ về GBN





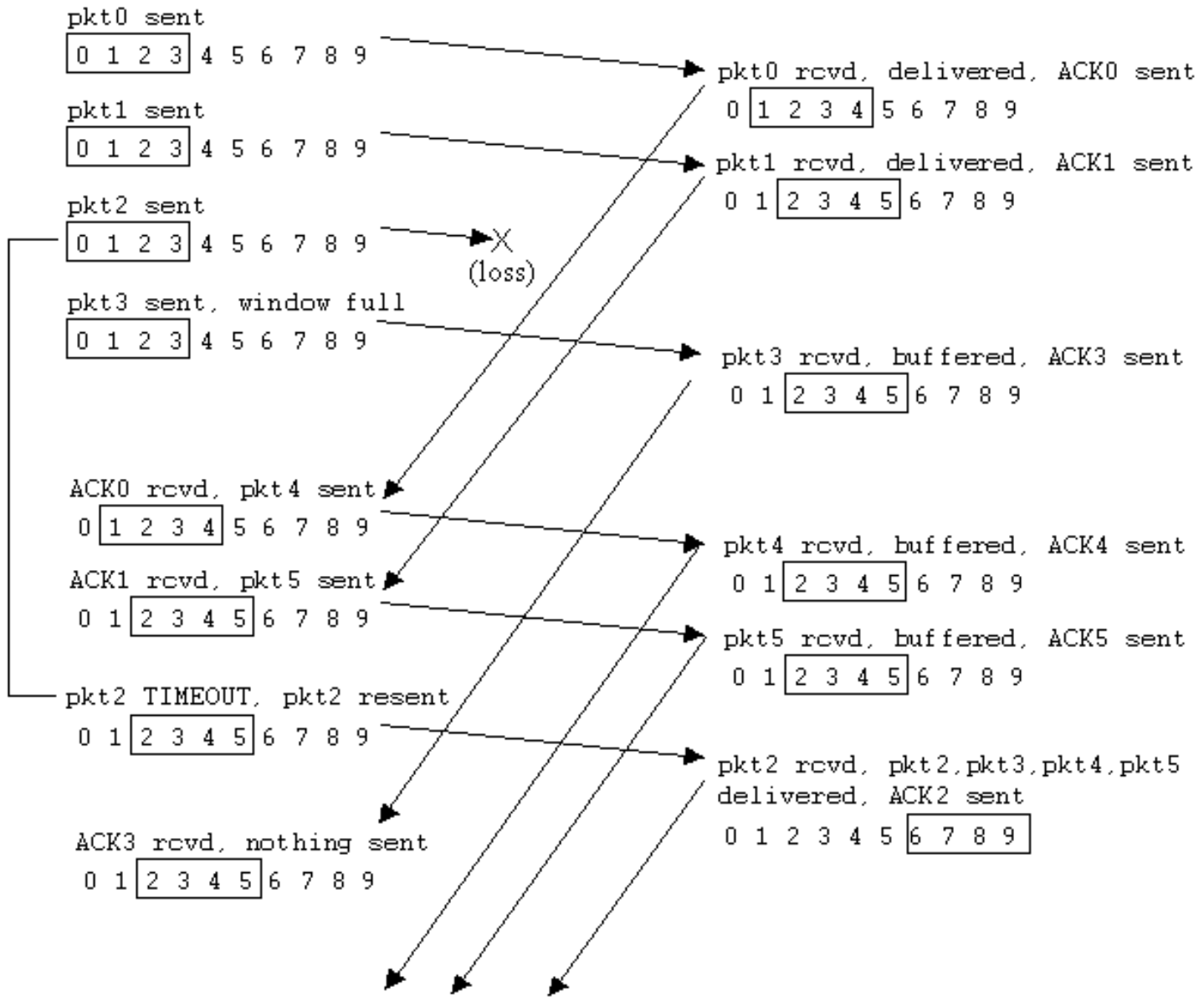
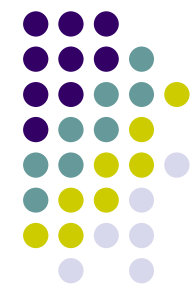


# Selective Repeat



- Bên nhận xác nhận riêng rẽ cho từng gói tin
- Chỉ gửi lại các gói tin chưa được xác nhận bị timeout
- Tổ chức vùng đệm để sắp xếp các gói tin theo đúng thứ tự để chuyển cho tầng trên

# Ví dụ về Selective Repeat



# Q. Hãy nhận xét về 2 phương pháp?



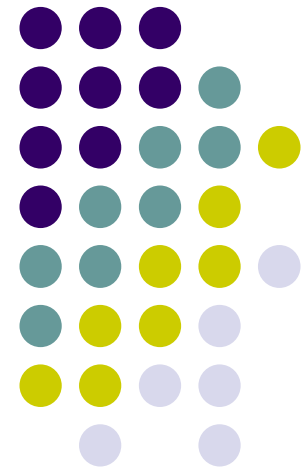
- Ưu điểm
- Nhược điểm

# TCP

## Transmission Control Protocol

---

Cấu trúc đoạn tin TCP  
Quản lý liên kết  
Kiểm soát luồng  
Kiểm soát tắc nghẽn

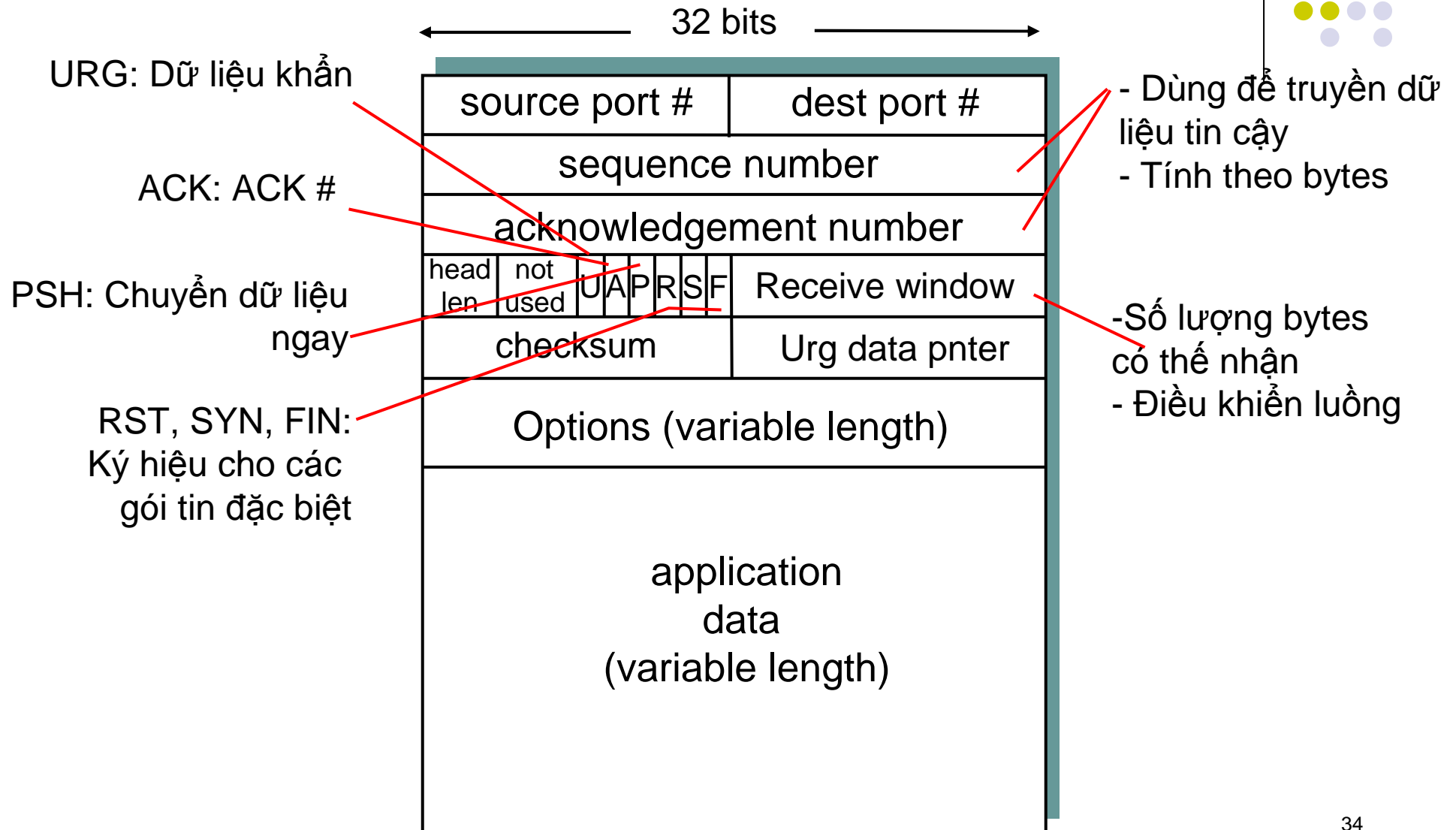




# Tổng quan về TCP

- Giao thức hướng liên kết
  - Bắt tay ba bước
- Giao thức truyền dữ liệu theo dòng byte, tin cậy
  - Sử dụng vùng đệm
- Truyền theo kiểu pipeline
  - Tăng hiệu quả
- Kiểm soát luồng
  - Bên gửi không làm quá tải bên nhận (thực tế: quá tải)
- Kiểm soát tắc nghẽn
  - Việc truyền dữ liệu không nên làm tắc nghẽn mạng (thực tế: luôn có tắc nghẽn)

# Khuôn dạng đoạn tin - TCP segment



# TCP cung cấp dịch vụ tin cậy ntn?



- Kiểm soát dữ liệu đã được nhận chưa:
  - Seq. #
  - Ack
- Chu trình làm việc của TCP:
  - Thiết lập liên kết
    - Bắt tay ba bước
  - Truyền/nhận dữ liệu
  - Đóng liên kết

# Cơ chế báo nhận trong TCP

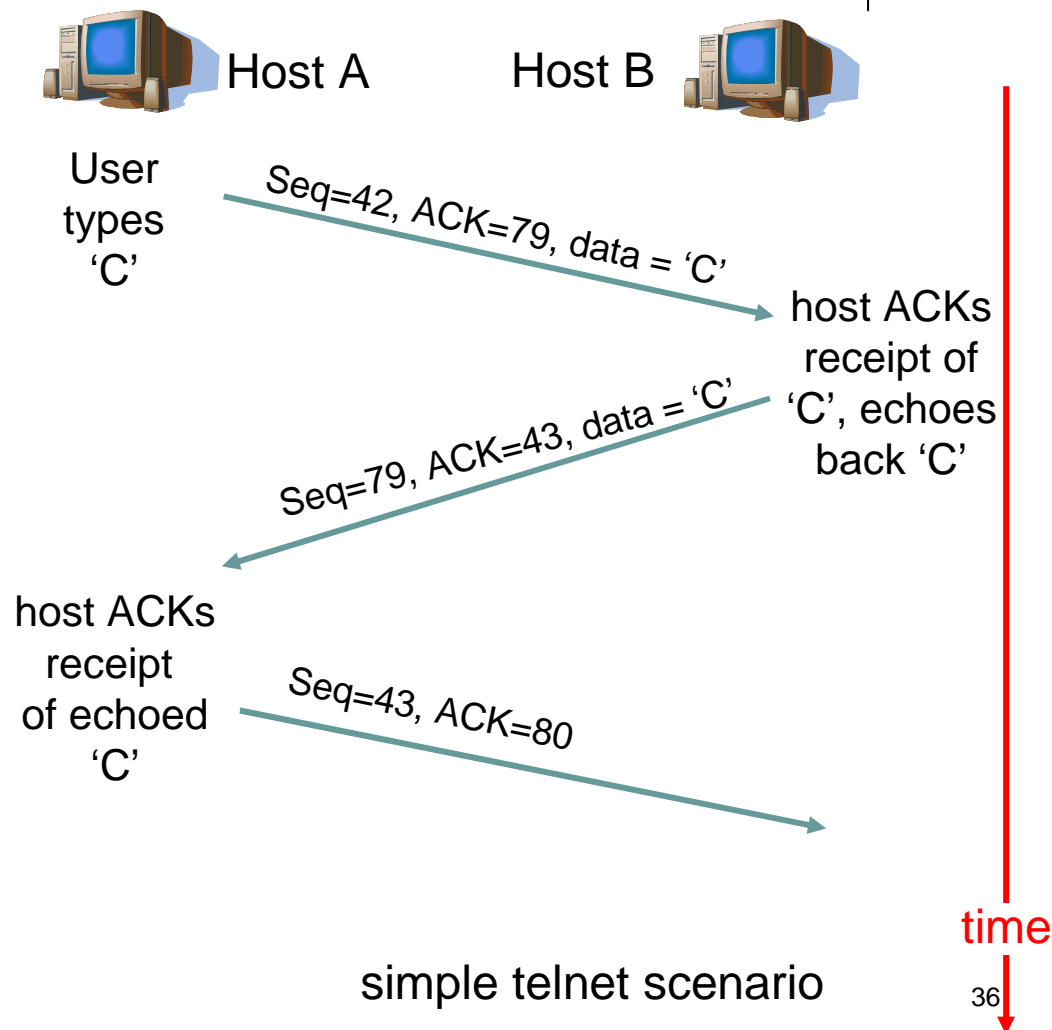


## Seq. #:

- Số hiệu của byte đầu tiên của đoạn tin trong dòng dữ liệu

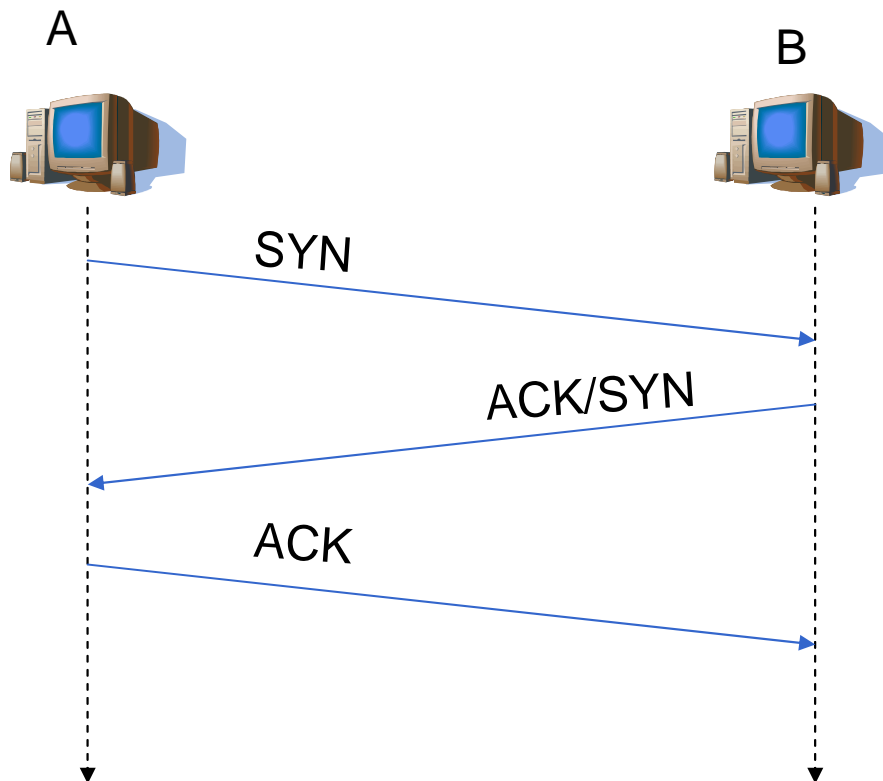
## ACK:

- Số hiệu byte đầu tiên mong muốn nhận từ đối tác





# Thiết lập liên kết TCP : Giao thức bắt tay 3 bước



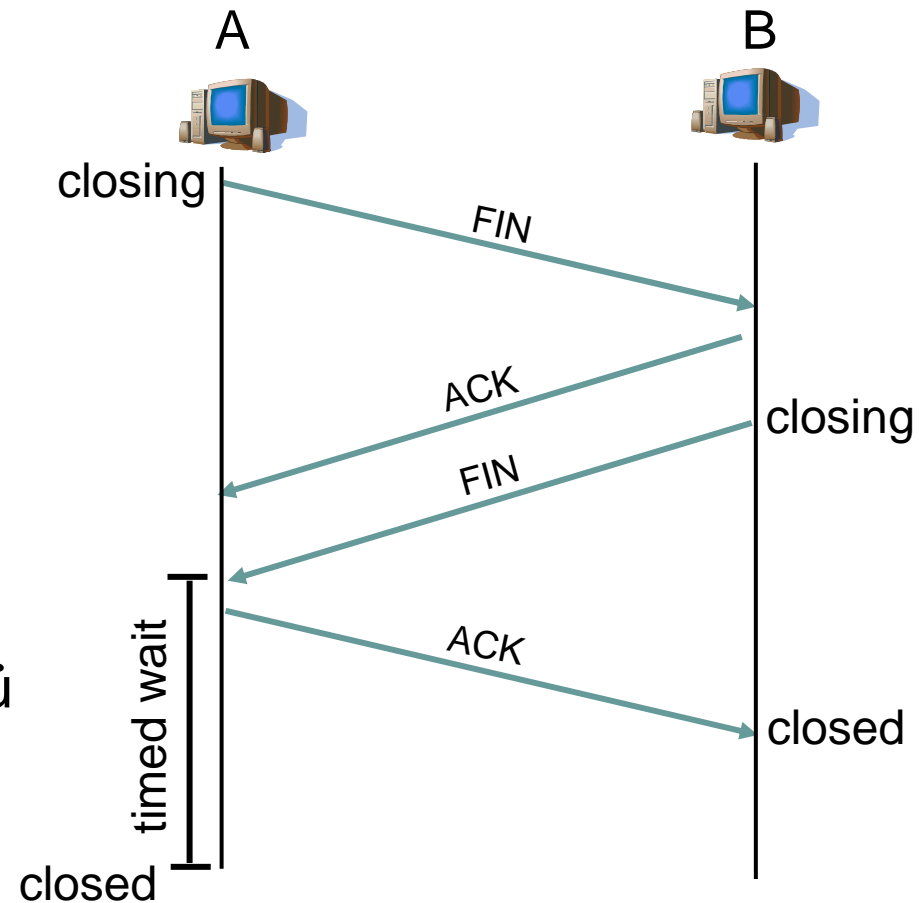
- **Bước 1:** A gửi SYN cho B
  - chỉ ra giá trị khởi tạo seq # của A
  - không có dữ liệu
- **Bước 2:** B nhận SYN, trả lời bằng SYNACK
  - B khởi tạo vùng đệm
  - chỉ ra giá trị khởi tạo seq. # của B
- **Bước 3:** A nhận SYNACK, trả lời ACK, có thể kèm theo dữ liệu

# Ví dụ về việc đóng liên kết

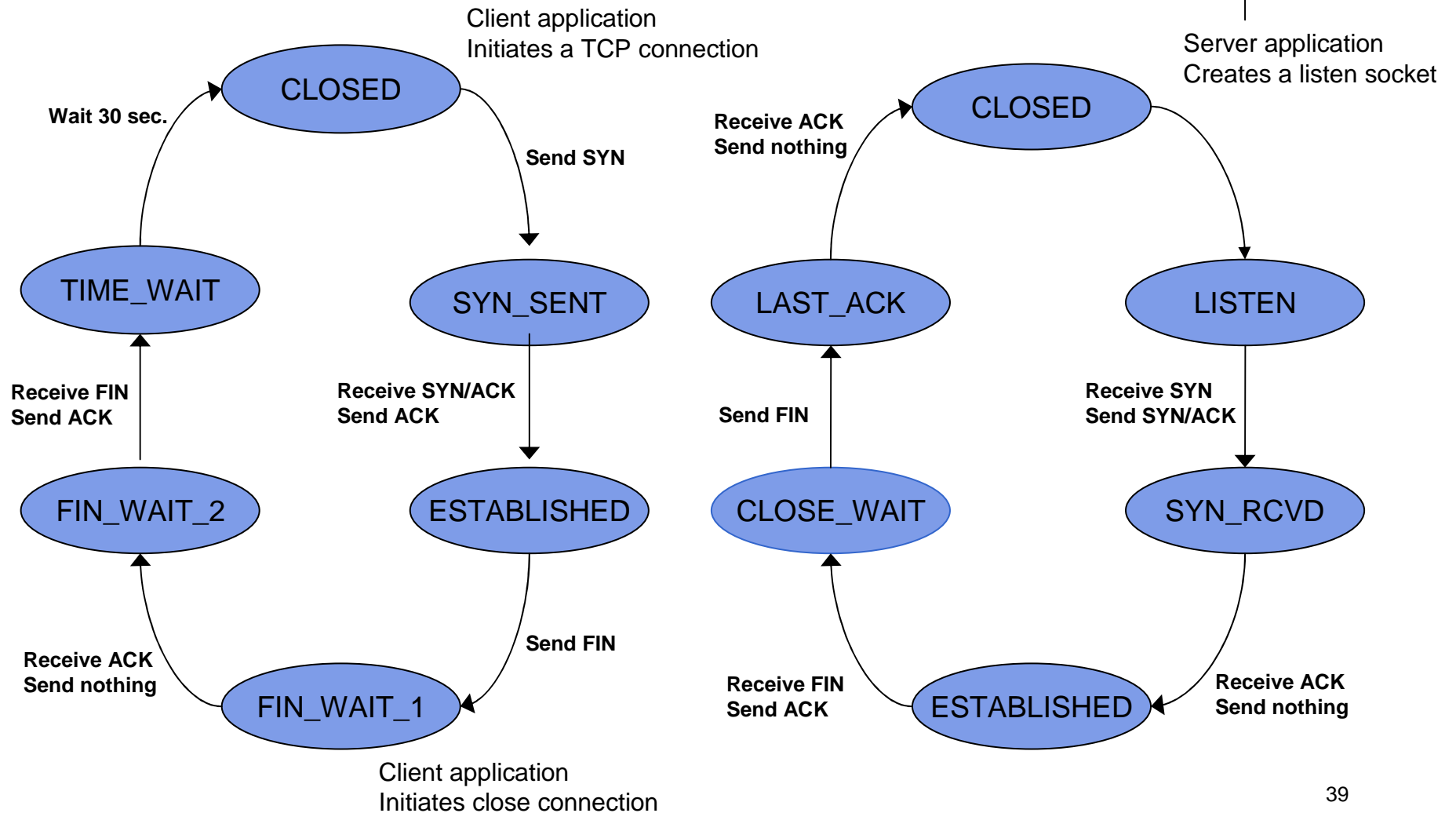
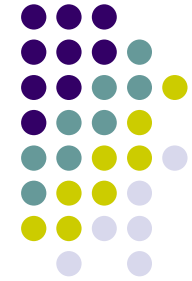


- **Bước 1:** Gửi FIN cho B
- **Bước 2:** B nhận được FIN, trả lời ACK, đồng thời đóng liên kết và gửi FIN.
- **Bước 3:** A nhận FIN, trả lời ACK, vào trạng thái “chờ”.
- **Bước 4:** B nhận ACK. đóng liên kết.

Lưu ý: Cả hai bên đều có thể chủ động đóng liên kết

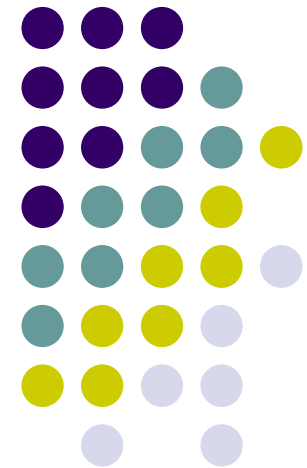


# Chu trình sống của TCP (đơn giản hóa)

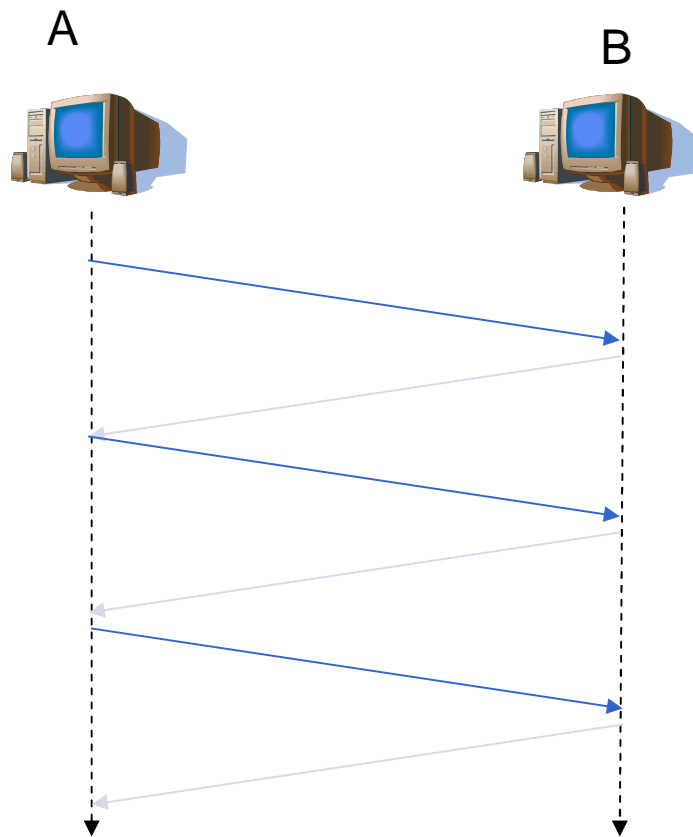


# Kiểm soát luồng

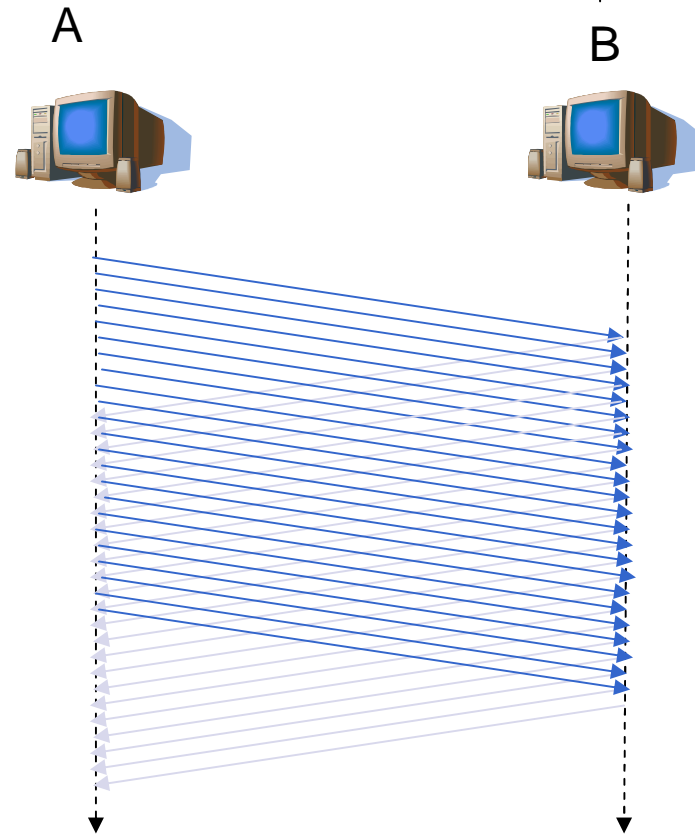
---



# Kiểm soát luồng (1)



**Chậm**



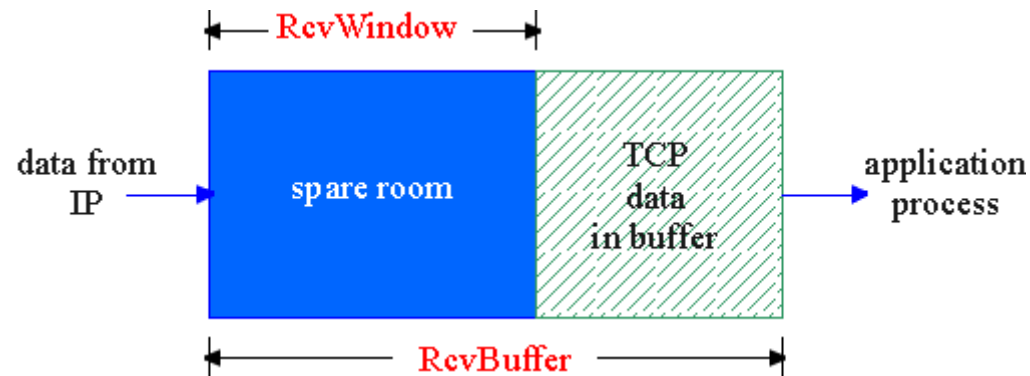
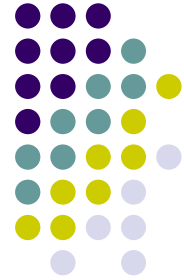
**Quá tải**



## Kiểm soát luồng (2)

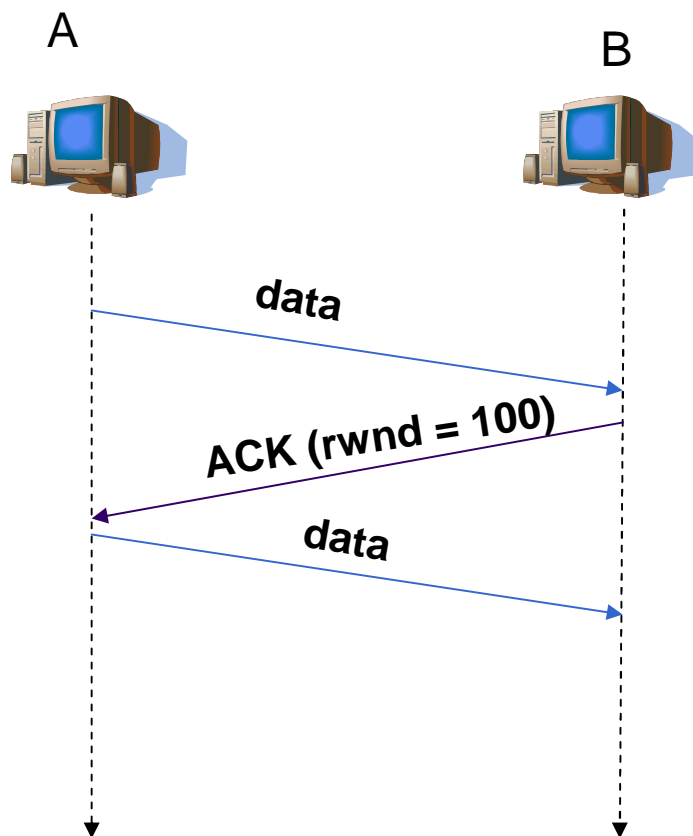
- Điều khiển lượng dữ liệu được gửi đi
  - Bảo đảm rằng hiệu quả là tốt
  - Không làm quá tải các bên
- Các bên sẽ có cửa sổ kiểm soát
  - Rwnd: Cửa sổ nhận
  - CWnd: Cửa sổ kiểm soát tắc nghẽn
- Lượng dữ liệu gửi đi phải nhỏ hơn  $\min(Rwnd, CWnd)$

# Kiểm soát luồng trong TCP



- Kích thước vùng đệm trống  
=  $Rwnd$   
=  $RcvBuffer - [LastByteRcvd - LastByteRead]$

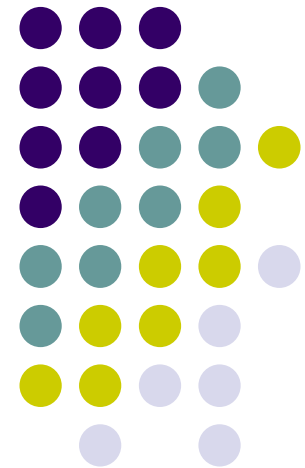
# Trao đổi thông tin về Rwnd



- Bên nhận sẽ báo cho bên gửi biết Rwnd trong các đoạn tin
- Bên gửi đặt kích thước cửa sổ gửi theo Rwnd



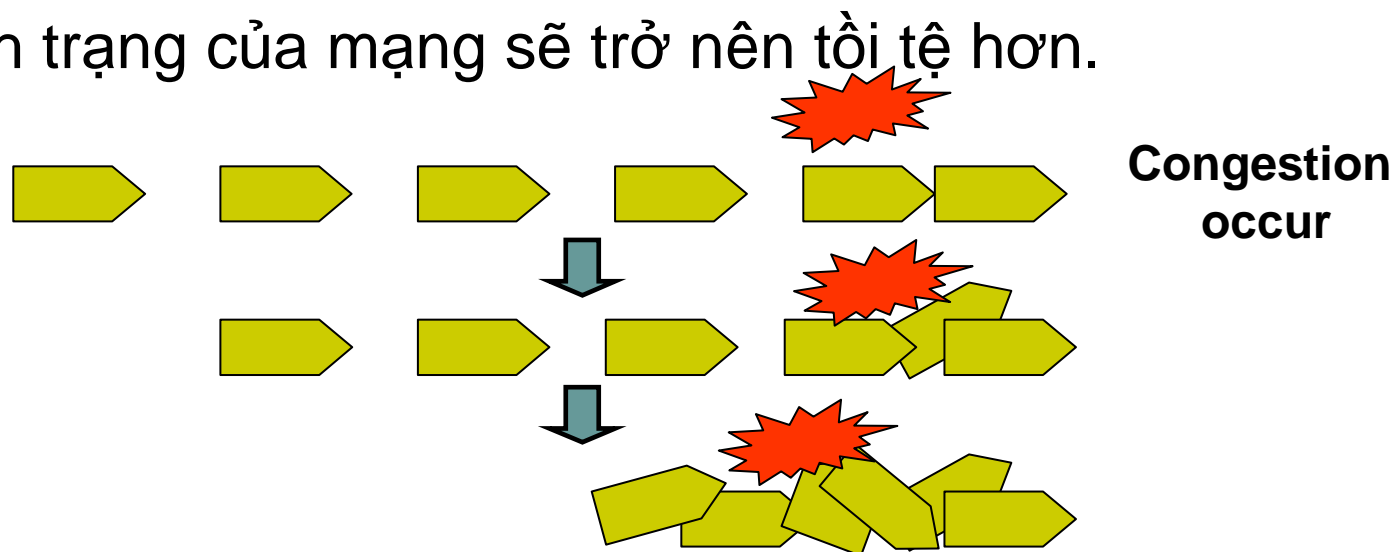
# Điều khiển tắc nghẽn trong TCP

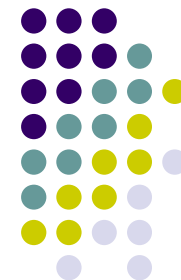




# Tổng quan về tắc nghẽn

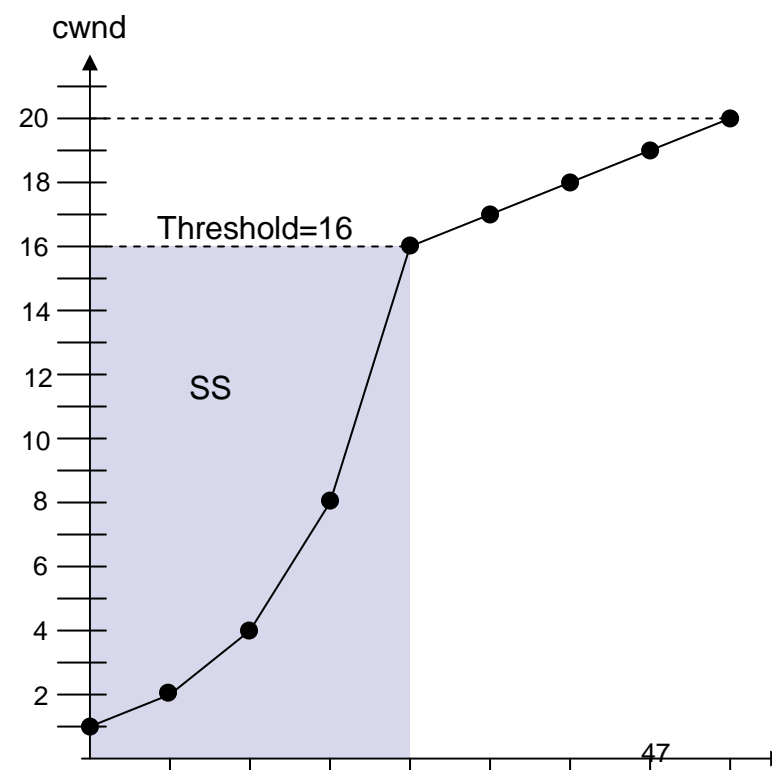
- Khi nào tắc nghẽn xảy ra ?
  - Quá nhiều cặp gửi-nhận trên mạng
  - Truyền quá nhiều làm cho mạng quá tải
- Hậu quả của việc nghẽn mạng
  - Mất gói tin
  - Thông lượng giảm, độ trễ tăng
  - Tình trạng của mạng sẽ trở nên tồi tệ hơn.

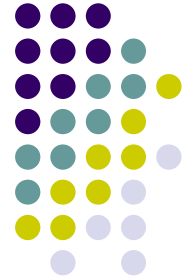




# Nguyên lý kiểm soát tắc nghẽn

- Slow-start
  - Tăng tốc độ theo hàm số mũ
  - Tiếp tục tăng đến một ngưỡng nào đó
- Tránh tắc nghẽn
  - Tăng dần tốc độ theo hàm tuyến tính cho đến khi phát hiện tắc nghẽn
- Phát hiện tắc nghẽn
  - Nếu gói tin bị mất

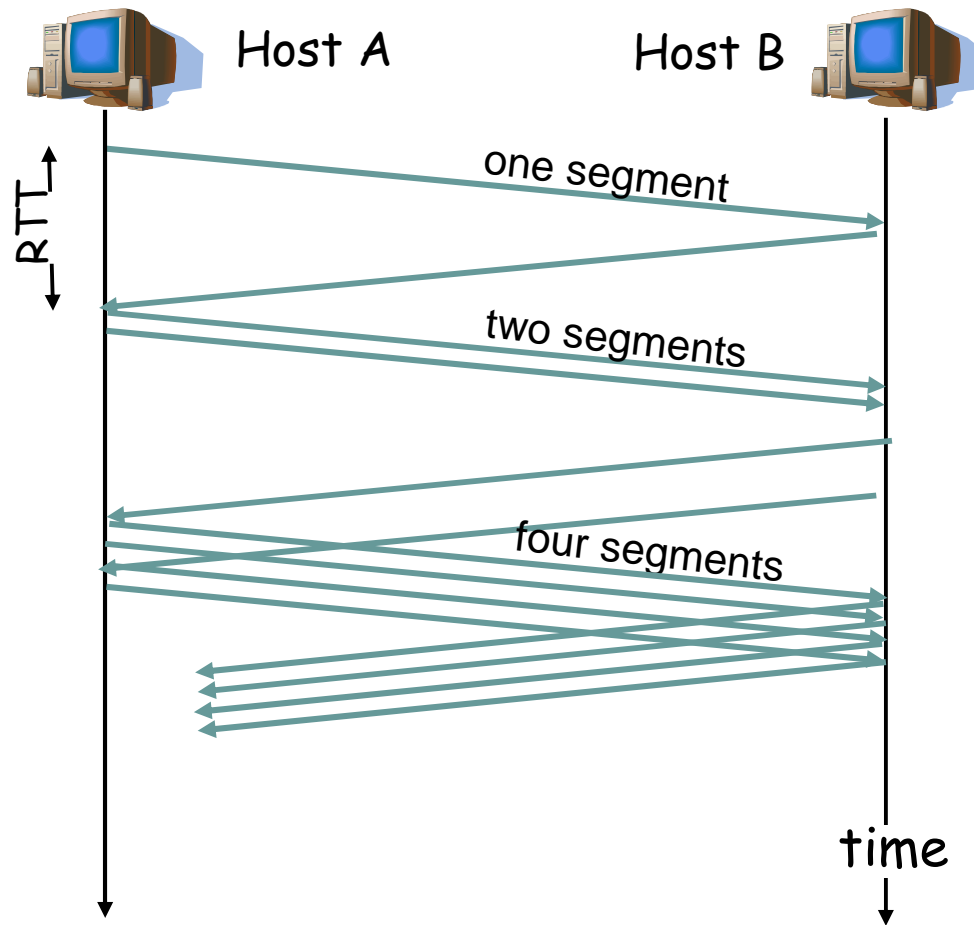




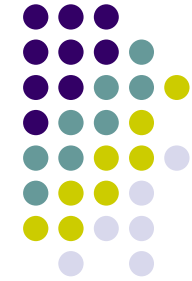
# TCP Slow Start (1)

- Ý tưởng cơ bản
  - Đặt cwnd bằng 1 MSS (Maximum segment size)
  - Tăng cwnd lên gấp đôi
    - Khi nhận được ACK
  - Bắt đầu chậm, nhưng tăng theo hàm mũ
- Tăng cho đến một ngưỡng: ssthresh
  - Sau đó, TCP chuyển sang trạng thái tránh tắc nghẽn

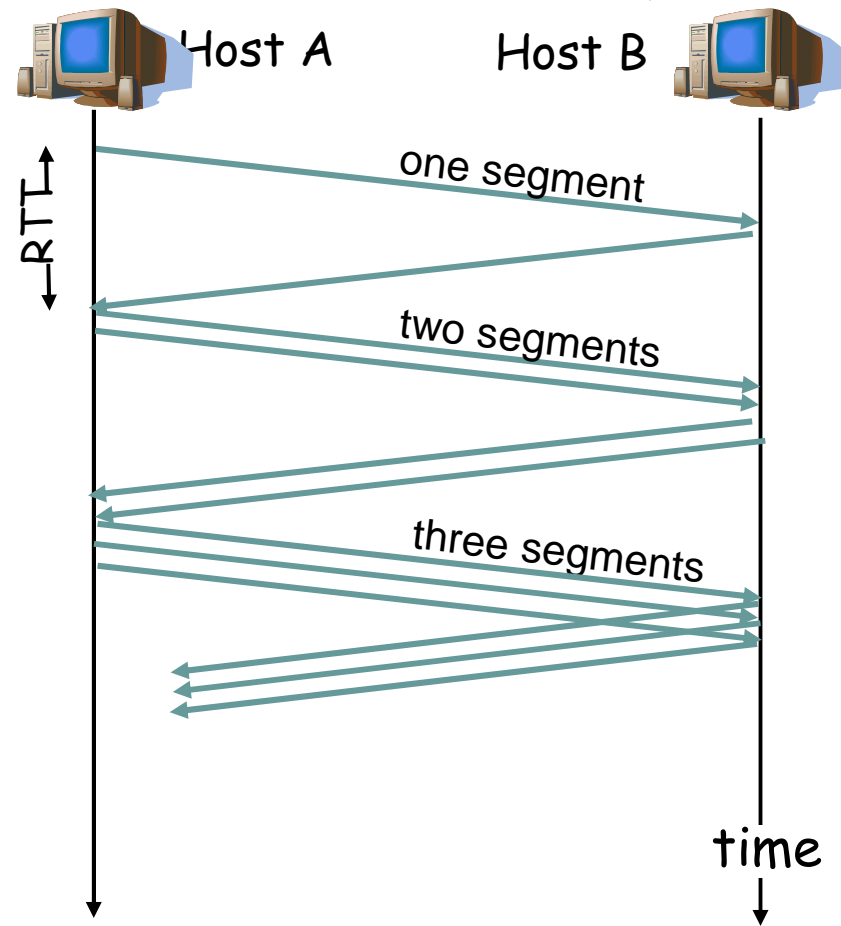
# TCP Slow Start (2)



# Tránh tắc nghẽn - Congestion avoidance



- ý tưởng cơ bản
  - Tăng cwnd theo cấp số cộng sau khi nó đạt tới ssthresh
  - Khi bên gửi nhận được ACK
    - Tăng cwnd thêm 1 MSS





# Phản ứng của TCP (1)

- Giảm tốc độ gửi
- Phát hiện tắc nghẽn?
  - Nếu như phải truyền lại
    - Có thể suy ra là mạng “tắc nghẽn”
- Khi nào thì phải truyền lại?
  - Timeout!
  - Cùng một gói tin số hiệu gói tin trong ACK

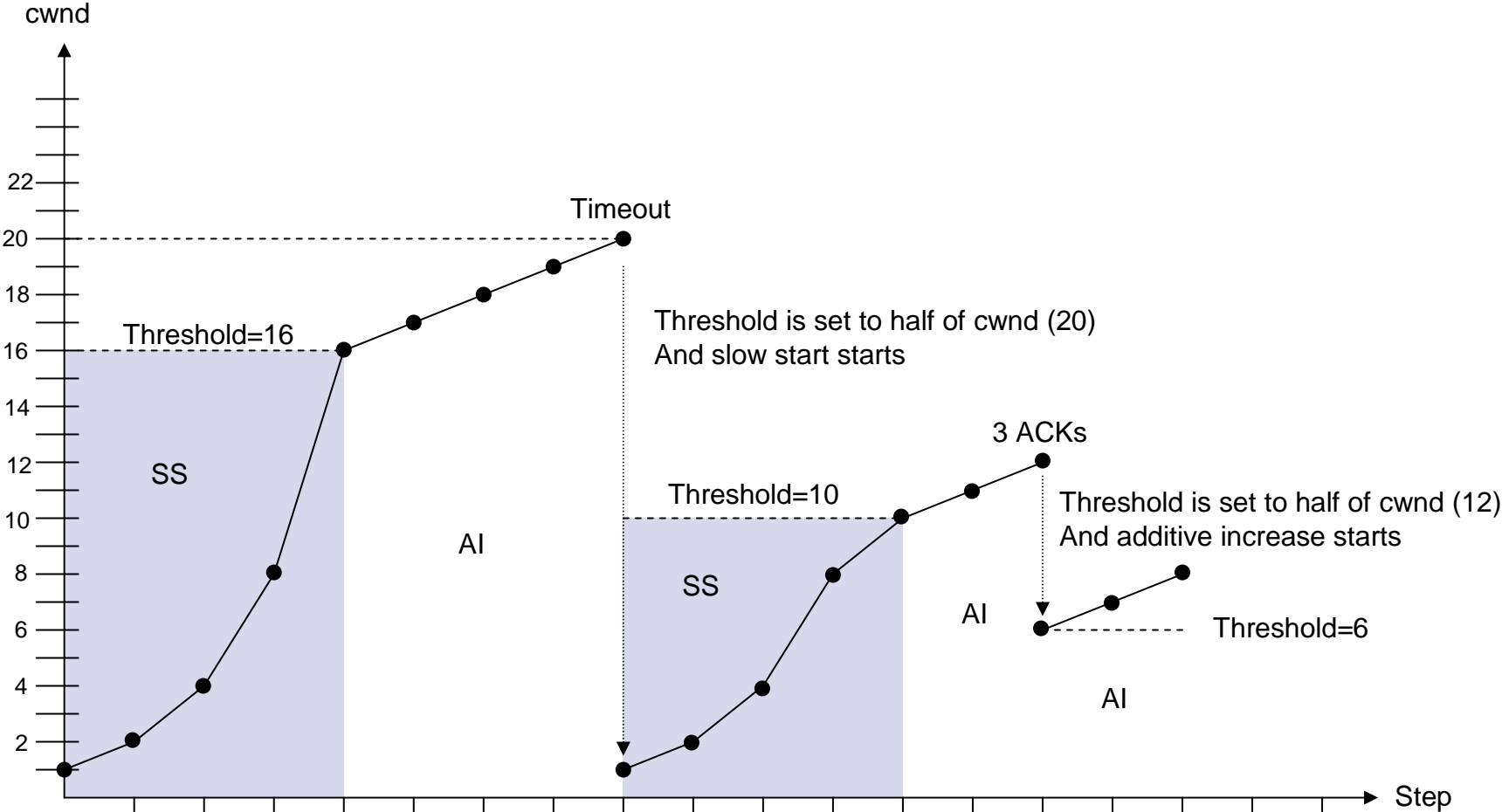


## Phản ứng của TCP (2)

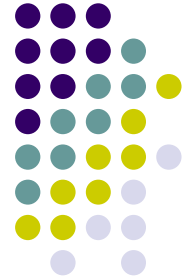
- Khi có timeout của bên gửi
  - TCP đặt ngưỡng xuống còn một nửa giá trị hiện tại của cwnd
  - TCP đặt cwnd về 1 MSS
  - TCP chuyển về slow start
- Nếu nhận được 3 ACK giống nhau
  - TCP đặt ngưỡng xuống còn một nửa giá trị hiện tại của cwnd
  - TCP đặt cwnd về giá trị hiện tại của ngưỡng cũ
  - TCP chuyển trạng thái “congestion avoidance”



# Kiểm soát tắc nghẽn – minh họa



# Tổng kết

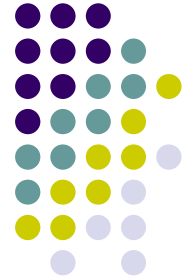


- Còn rất nhiều chi tiết về TCP!
- Có hai dạng giao thức giao vận
  - UDP và TCP
  - Best effort vs. reliable transport protocol
- Các cơ chế bảo đảm độ tin cậy
  - Báo nhận
  - Truyền lại
  - Kiểm soát luồng và kiểm soát tắc nghẽn



# Tuần tới: Application Layer

- Application service model
  - Client-server vs. P2P
- Typical applications and protocols
  - HTTP
  - Mail
  - FTP
  - P2P file sharing
  - .....
  - and your applications?



# Acknowledgment

- Bài giảng có sử dụng các hình vẽ từ
  - Tài liệu của trường đại học Keio và Ritsumeikan
  - Tài liệu “Computer Network, a top down approach” của J.F Kurose và K.W. Ross