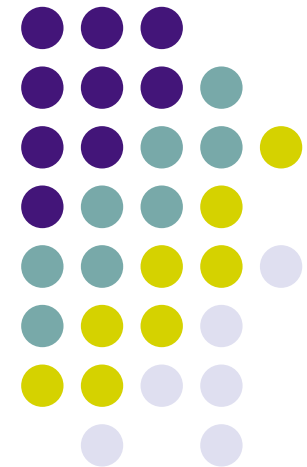


# Chương 7: Chọn đường – Routing

---



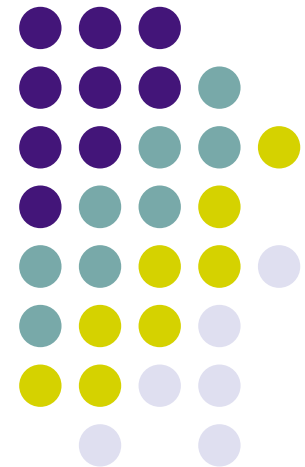


# Tổng quan

- 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?

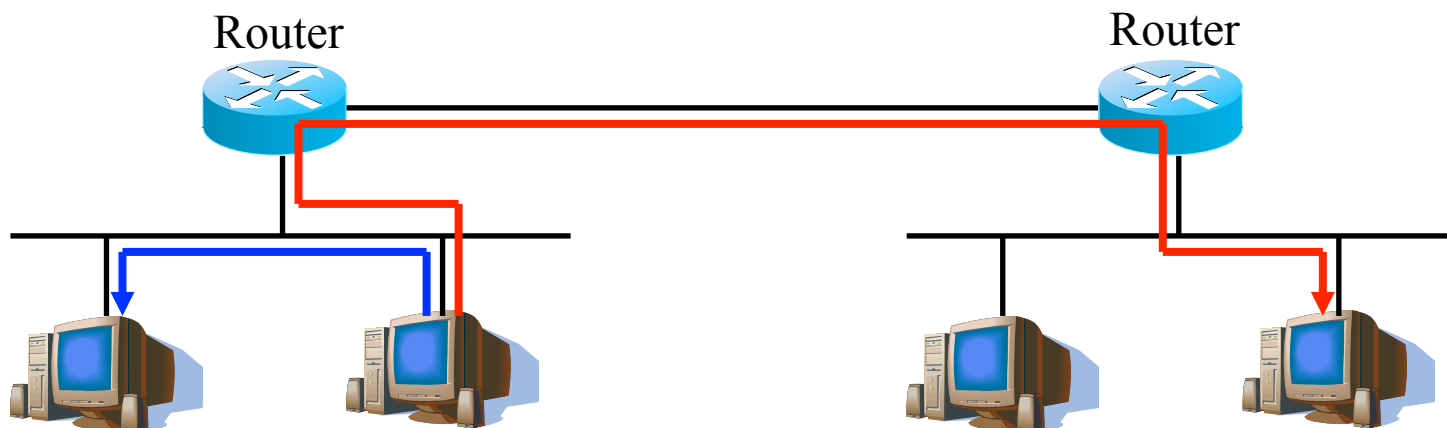
Các nguyên lý chọn đường  
Cơ chế chuyển tiếp gói tin  
Quy tắc “Longest matching”





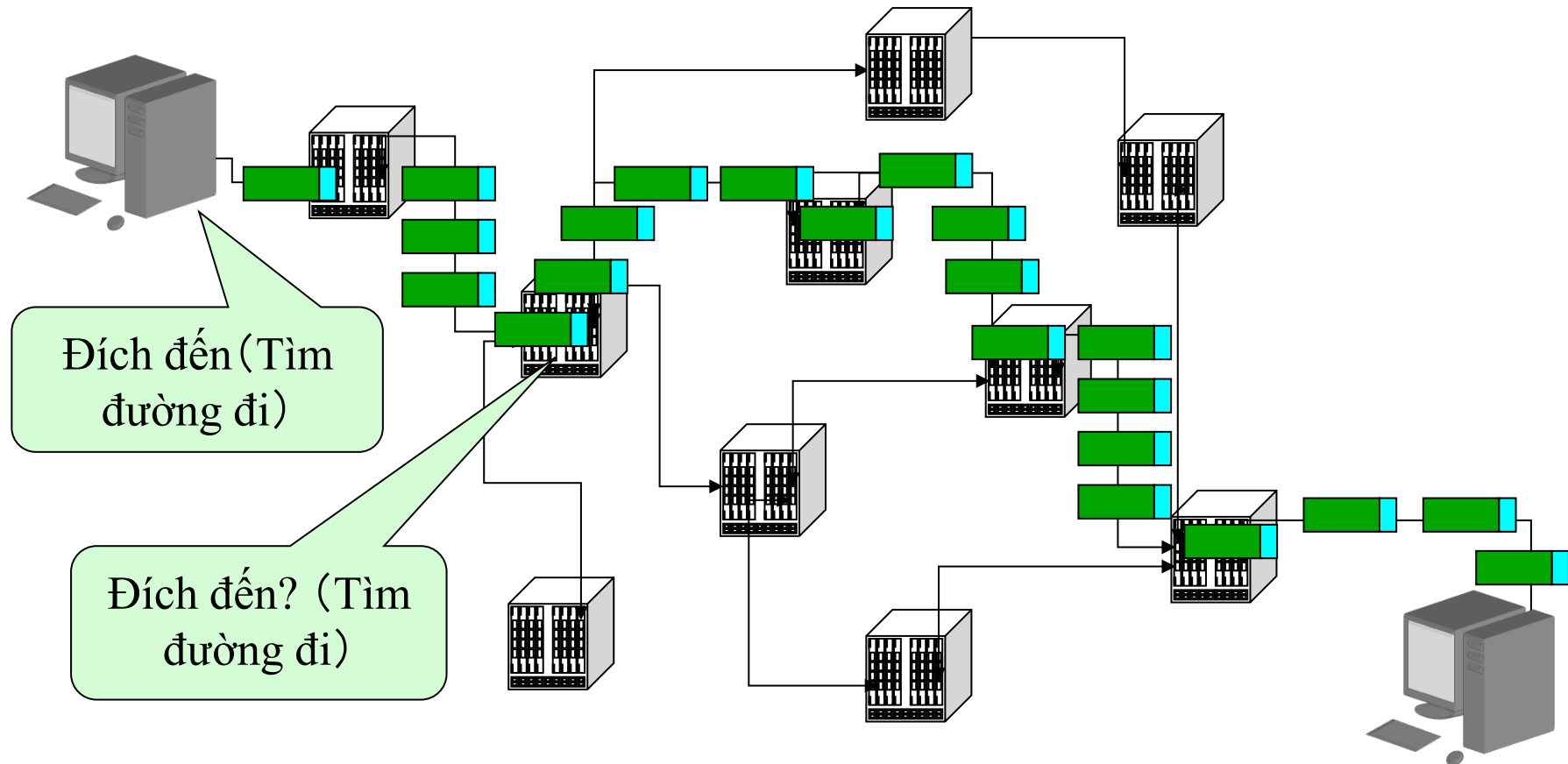
# Cơ bản về chọn đường (1)

- 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



Hitachi  
GR2000-1B



Juniper M10



Foundry Networks  
NetIron 800



Cisco 3700

## Router cỡ trung



Cisco CRS-1

## Router mạng trục

<http://www.cisco.com.vn>

<http://www.juniper.net/>

<http://www.buffalotech.com>





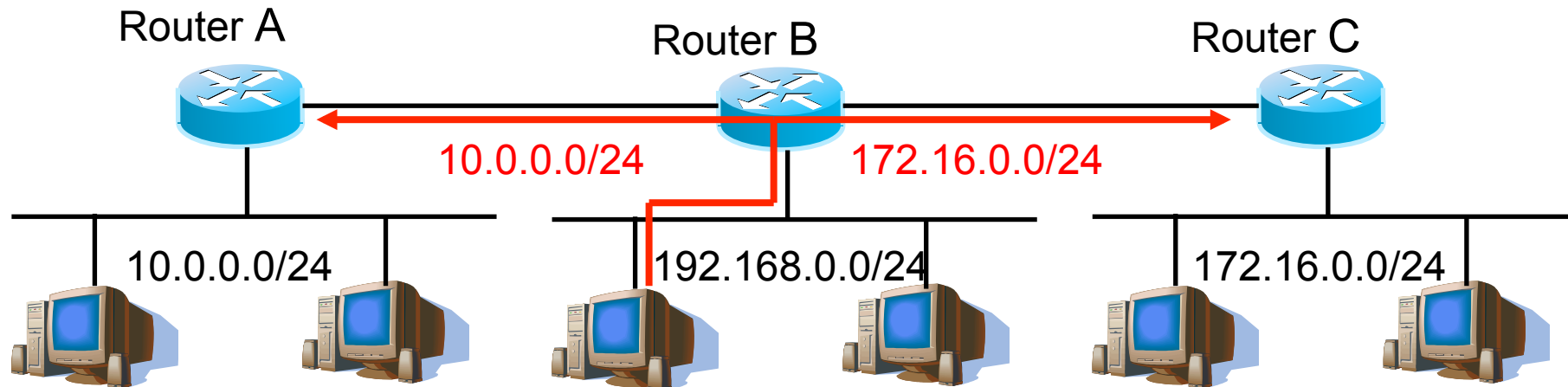
# Bảng chọn đường

- 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)

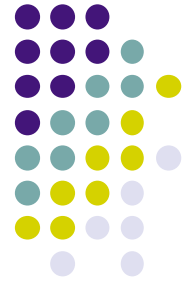


Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C
192.168.0.0/24	Direct



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

11.1.3.6 = 00001011.00000001.00000011.00000110

Đường đi 1:

11.1.2.0/24 = 00001011.00000001.00000010.00000000

Đường đi 2:

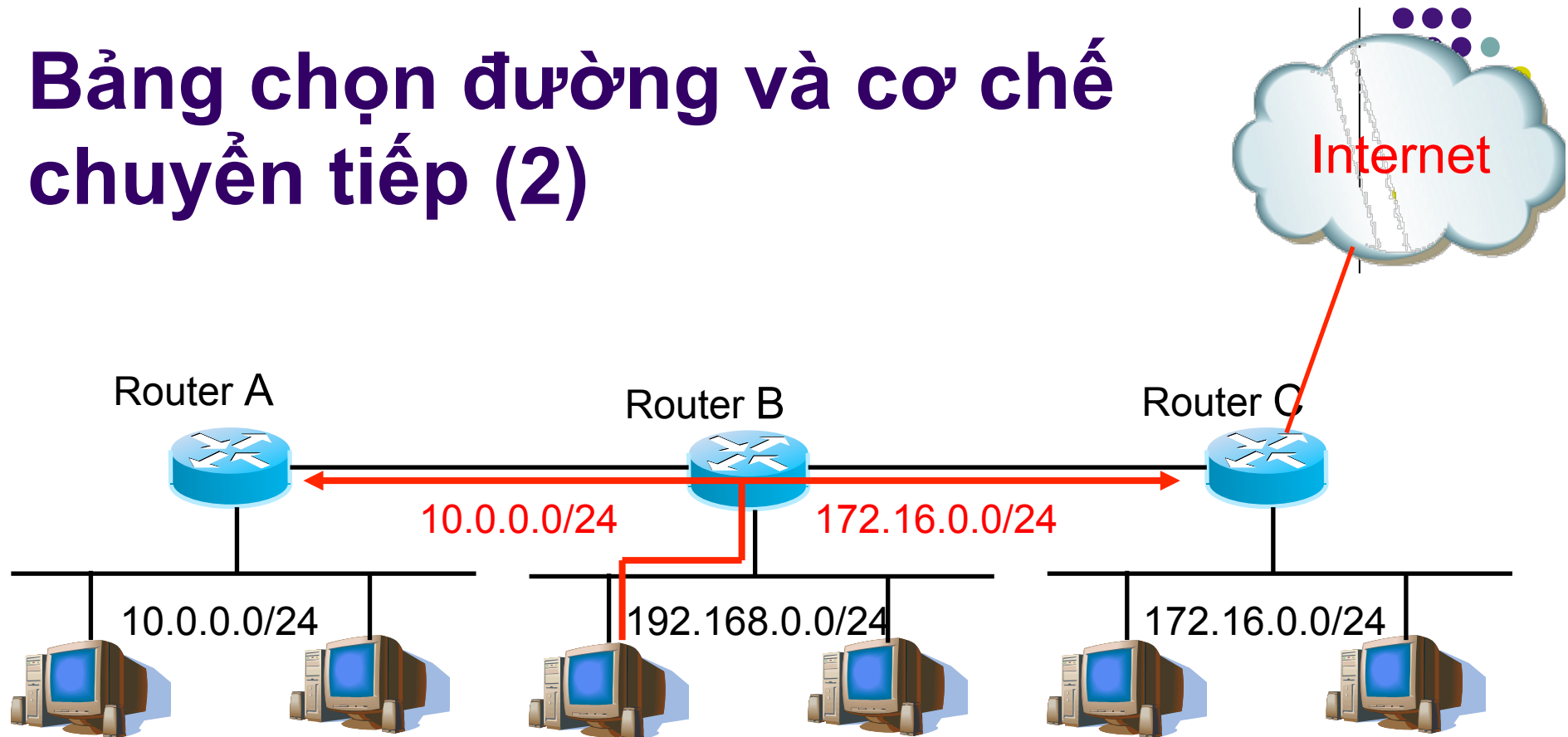
11.1.0.0/16 = 00001011.00000001.00000000.00000000

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

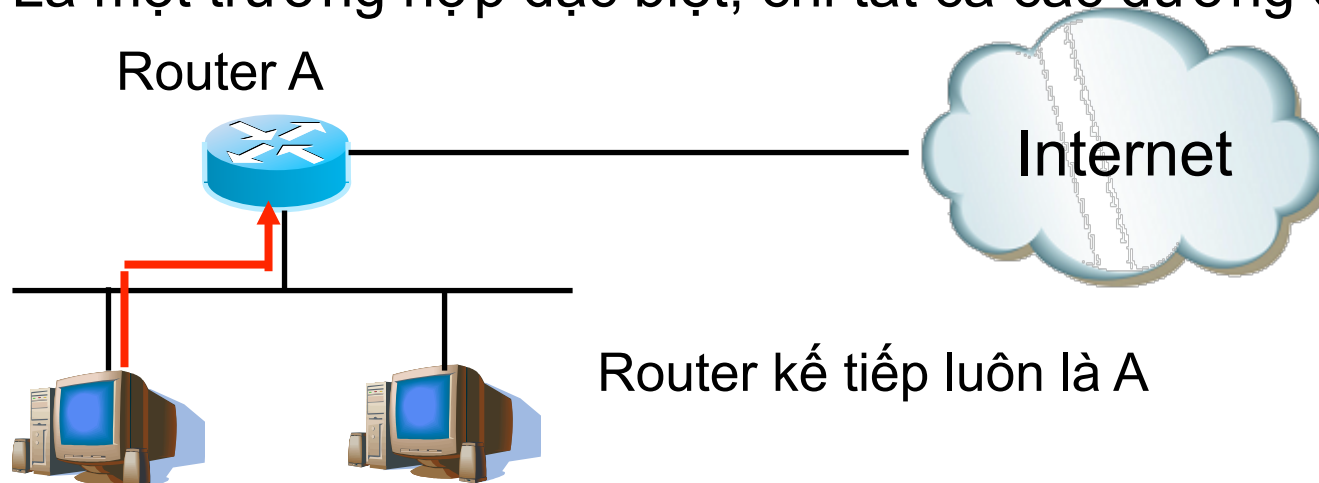
Q. Mô tả bảng chọn đường trên C

Nếu C nối vào Internet?



# Đường đi mặc định

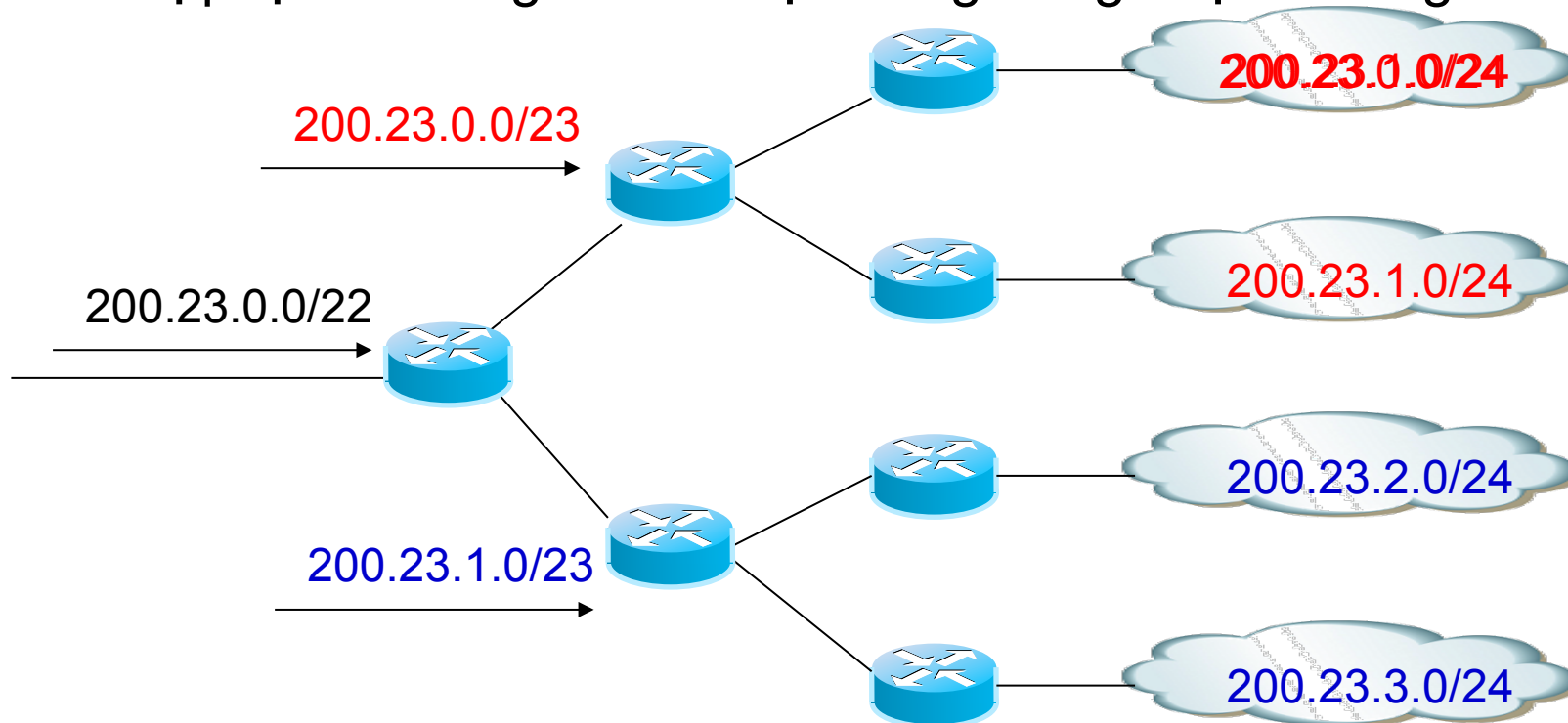
- Nếu đường đi không tìm thấy trong bảng chọn đường
  - Đường đi mặc định trở đến một router kết tiếp
  - Trong nhiều trường hợp, đây là đường đi duy nhất
- 0.0.0.0/0
  - Là một trường hợp đặc biệt, chỉ tất cả các đường đi

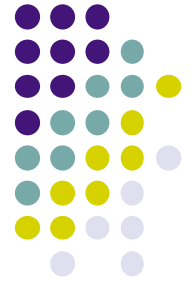


# Kết hợp đường đi (Routing aggregation)



- 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)



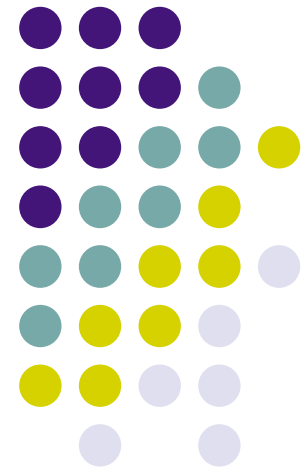
```
#show ip route
```

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

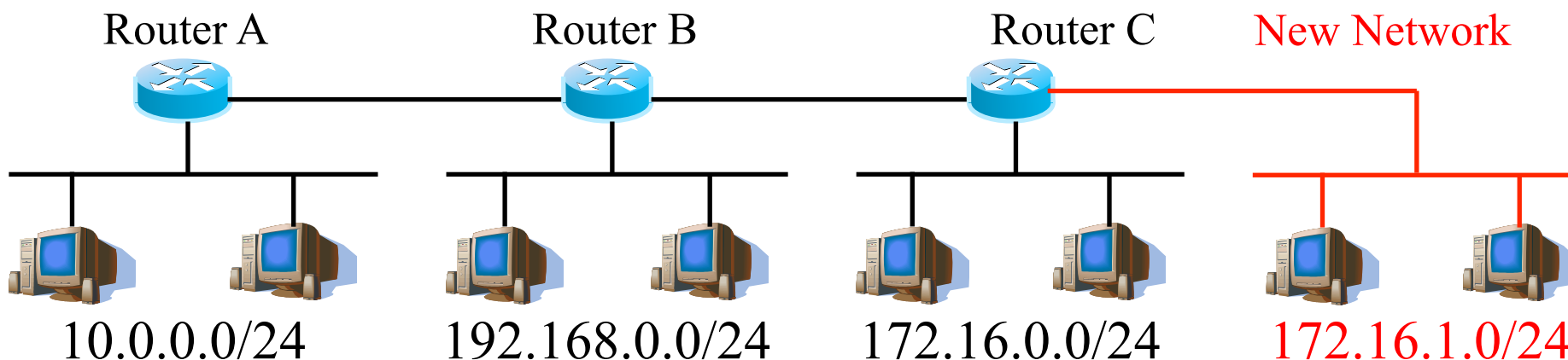
Network	Next-hop
192.168.0.0/24	B
172.16.0.0/24	B

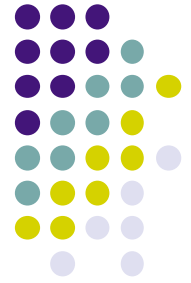
172.16.1.0/24      B

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

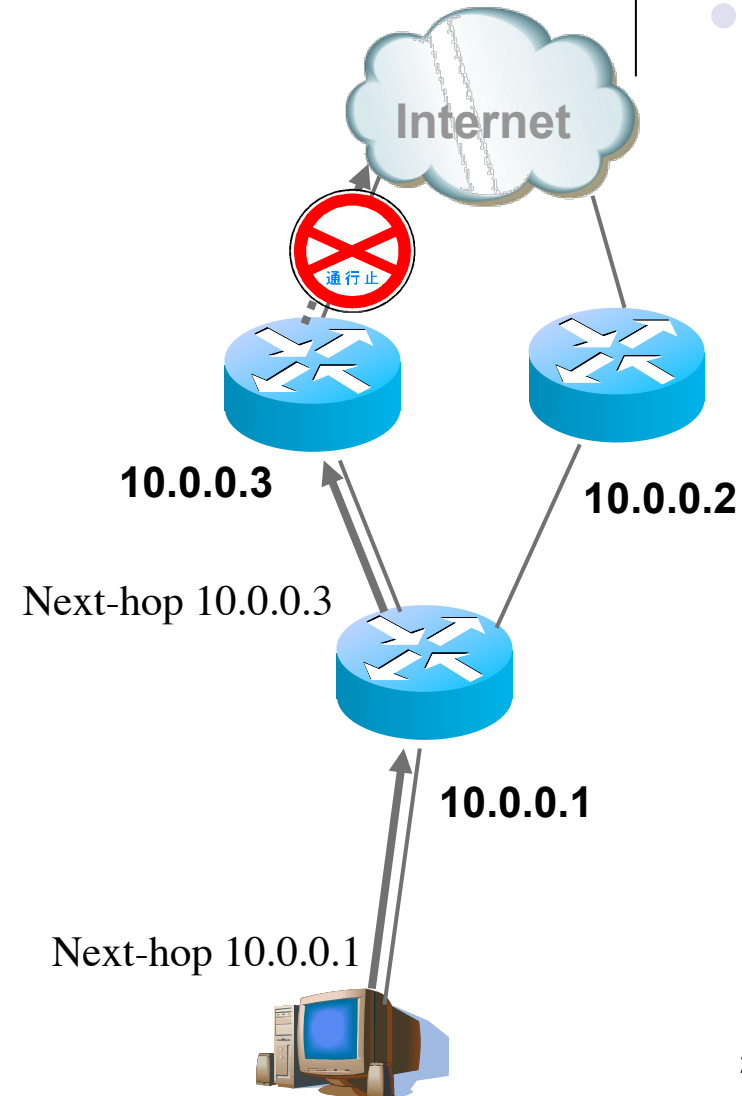
# Chọn đường tĩnh

- 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

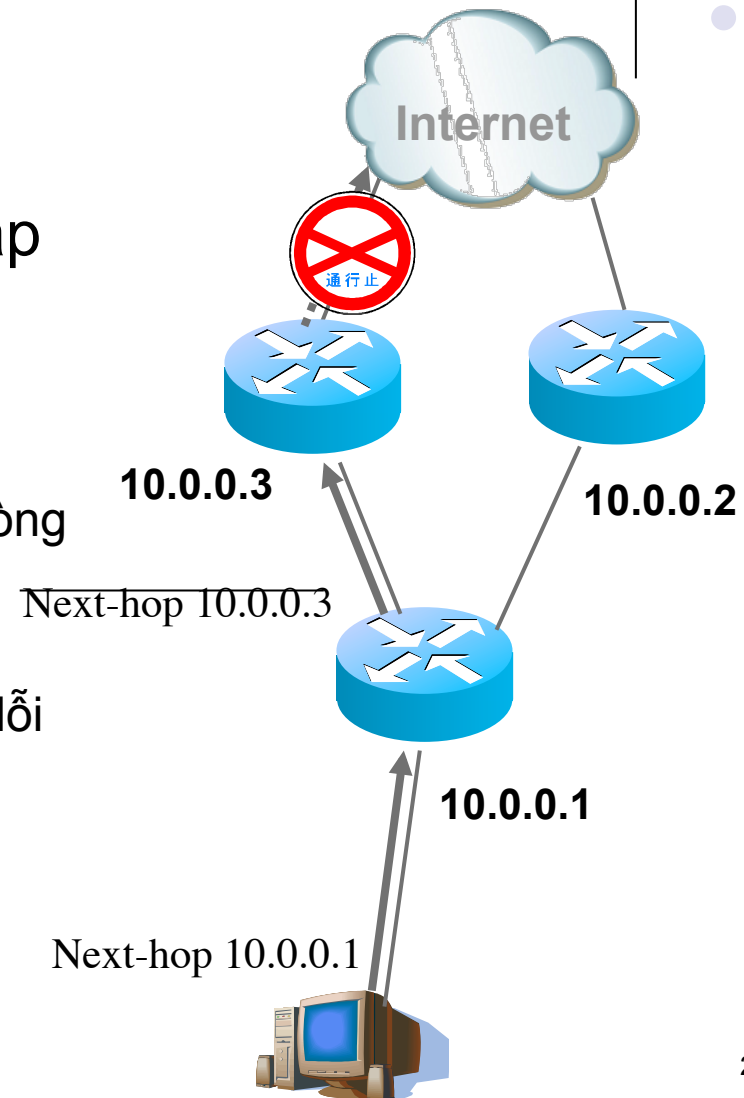
- Khi có sự cố:
  - Đường đi thay thế được cập nhật một cách tự động

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



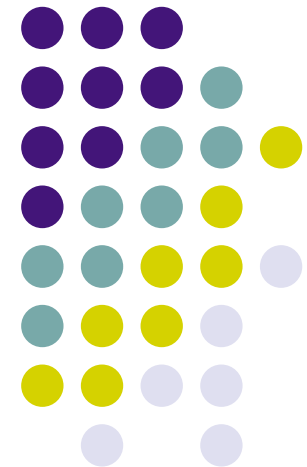


# Chọn đường động

- 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

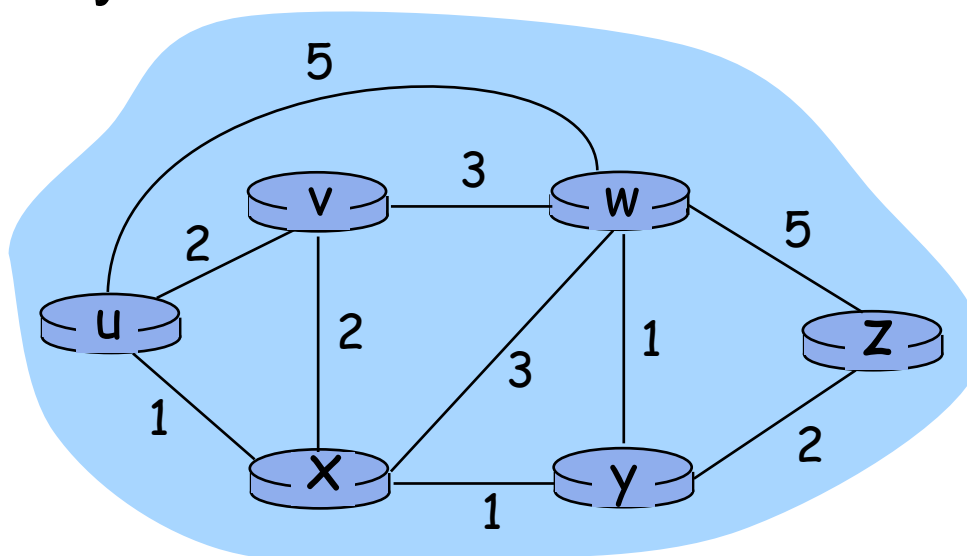
Giải thuật Dijkstra và Bellman-Ford  
Giao thức dạng link-state và dạng  
distance-vector



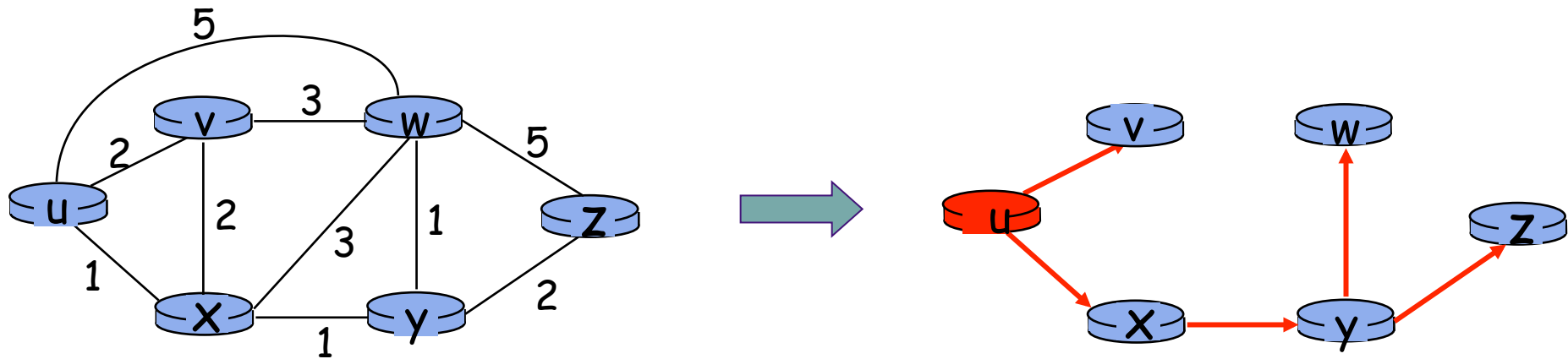


# Biểu diễn mạng bởi đồ thị

- 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**: distance-vector
  - Được sử dụng phổ biến trong thực tế



# Tập trung hay phân tán

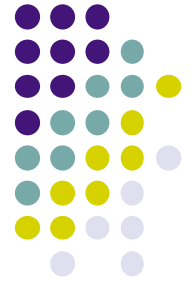
- Thông tin chọn đường là cần thiết để xây dựng bảng chọn đường
- Tập trung hay phân tán?
  - Tập trung:
    - Mỗi router có thông tin đầy đủ về trạng thái của mạng
    - Giải thuật dạng “link state”
  - Phân tán:
    - Các nút chỉ biết được trạng thái của liên kết vật lý tới nút kế bên
    - Liên tục 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

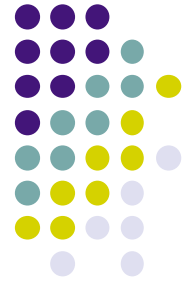
- Mỗi nút đều có sơ đồ và chi phí mỗi link
  - Quảng bá “Link-state”
  - Mỗi nút có cùng thông tin
- Tìm đường đi chi phí nhỏ nhất từ một nút ( ‘nguồn’ ) tới tất cả các nút khác
  - dùng để xây dựng bảng chọn đường



# 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

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

$p[v] = u$



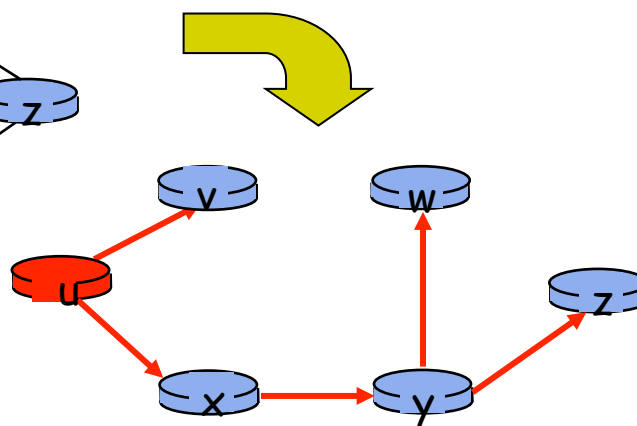
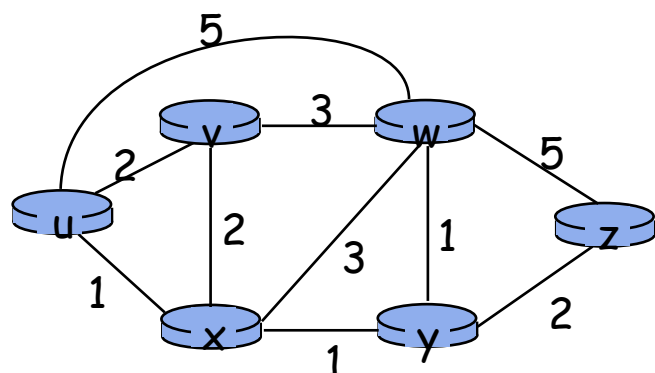
# Dijkstra's Algorithm

1. **Init()** ;
2.  $T = \Phi$ ;
3. **Repeat**
4.      $u: u \notin T \mid d(u)$  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

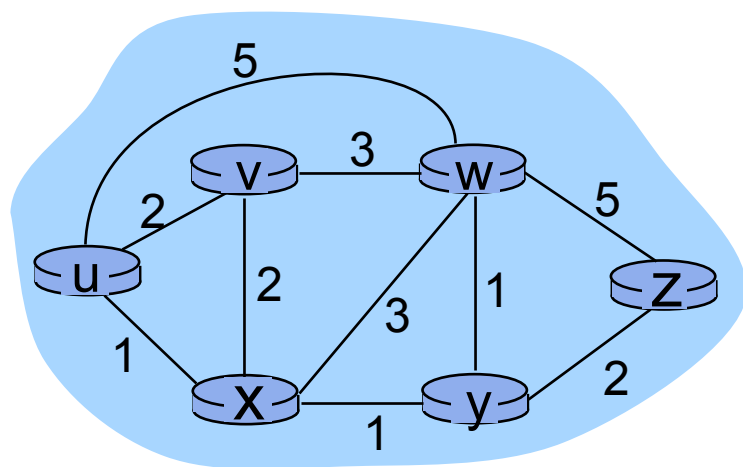
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

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 → 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
x	0	2	7
y	$\infty$	$\infty$	$\infty$
z	$\infty$	$\infty$	$\infty$

chi phí tới

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

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

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

nút y

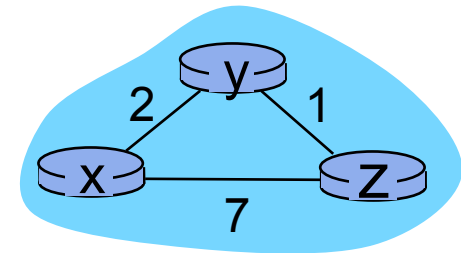
chi phí tới

	x	y	z
x	$\infty$	$\infty$	$\infty$
y	2	0	1
z	$\infty$	$\infty$	$\infty$

nút z

chi phí tới

	x	y	z
x	$\infty$	$\infty$	$\infty$
y	$\infty$	$\infty$	$\infty$
z	7	1	0



thờigian



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

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

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

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

**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

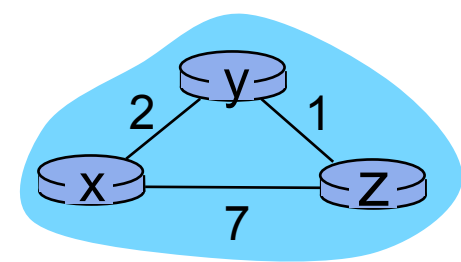
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

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
  - thay đổi

## Tốc độ hội tụ

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

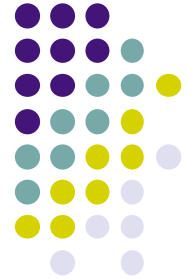
**Sự chắc chắn:** Giải sử một router hoạt động sai

### LS:

- nút gửi các chi phí sai
- Mỗi nút tính riêng bảng chọn đường  $\rightarrow$  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