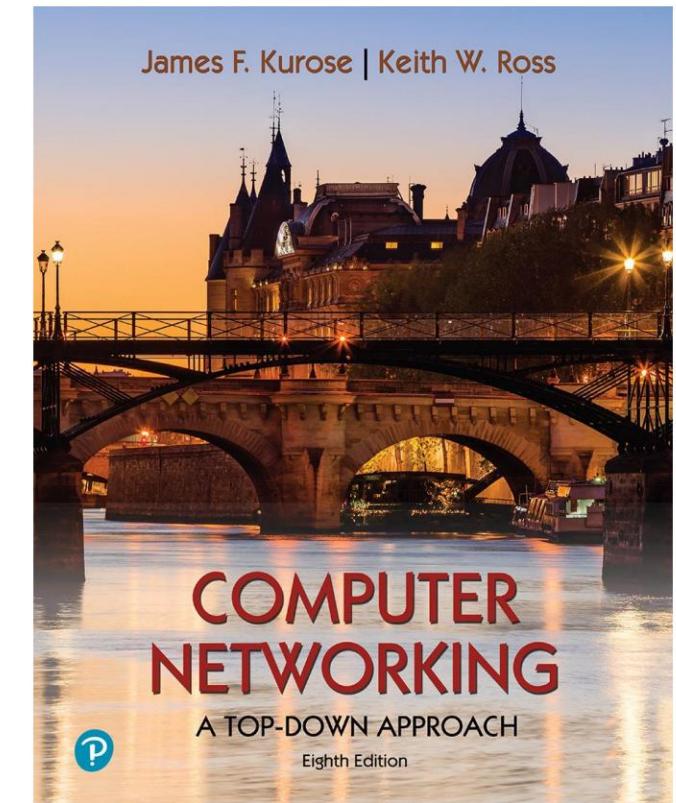


Chương 5

Lớp mạng: máy bay điều khiển



Mạng máy tính: A
Cách tiếp cận từ trên xuống
phiên bản thứ 8
Jim KuroseKeith Ross
Pearson, 2020

Mặt phẳng điều khiển lớp mạng: mục tiêu của chúng tôi

Hiểu các nguyên tắc đằng sau mặt phẳng điều khiển

mạng: • thuật toán định tuyến truyền thông • Bộ điều khiển SDN

• quản lý mạng, cấu hình

khởi tạo, triển khai trên mạng:

- OSPF, BGP
- Bộ điều khiển OpenFlow, ODL và ONOS
- Thông báo Kiểm soát Internet Giao thức: ICMP
- SNMP, YANG/NETCONF

Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

trạng thái liên kết

vector khoảng cách

định tuyến trong ISP : OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Chức năng tầng mạng

chuyển tiếp: di chuyển các gói từ đầu vào của bộ định tuyến đến đầu ra của bộ định tuyến thích hợp

mặt phẳng dữ liệu

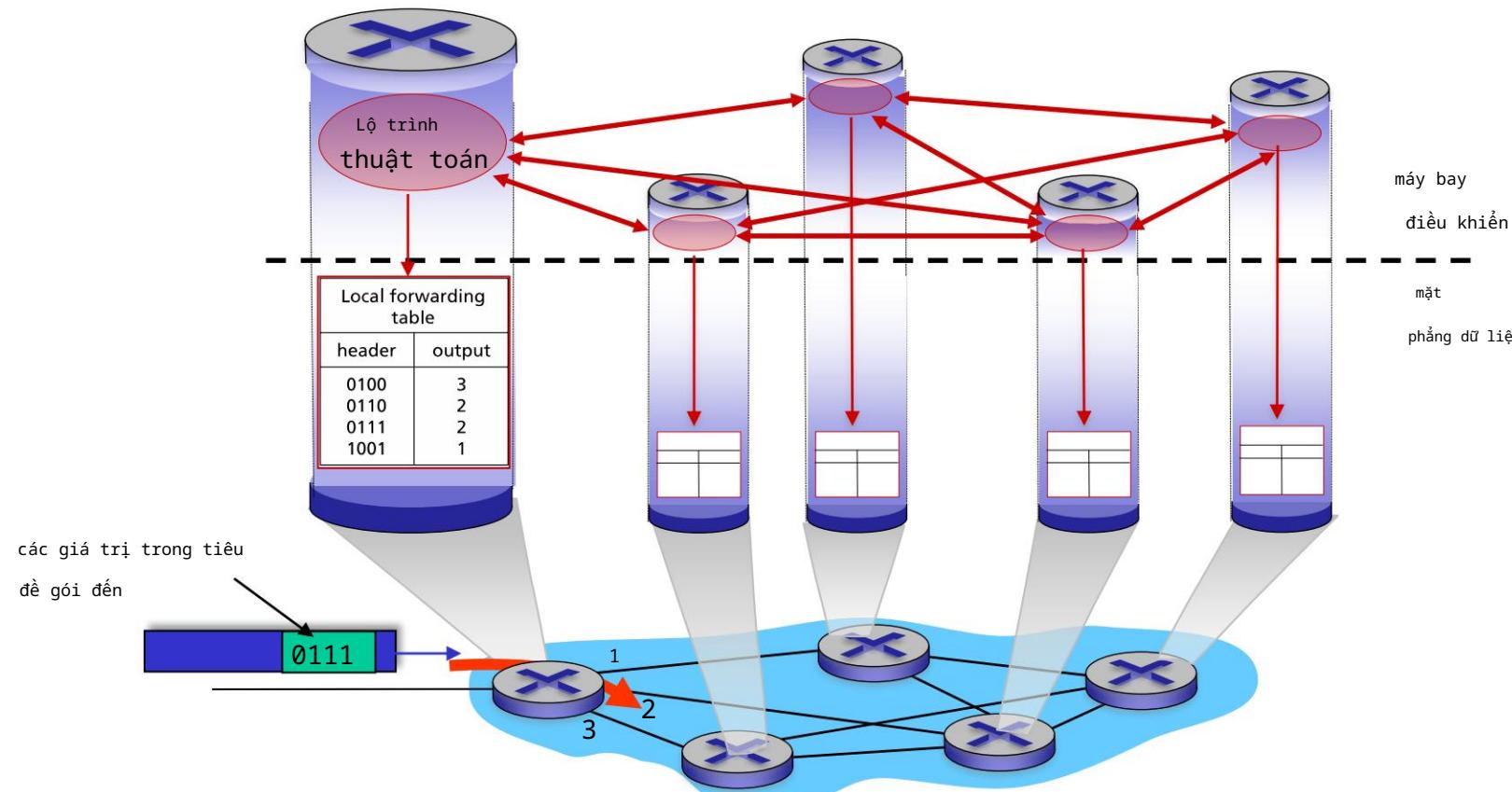
định tuyến: xác định đường đi của các gói tin từ nguồn đến đích

máy bay điều khiển

Hai cách tiếp cận để cấu trúc mặt phẳng điều khiển mạng: điều khiển theo từng bộ định tuyến (truyền thống) điều khiển tập trung về mặt logic (mạng được xác định bằng phần mềm)

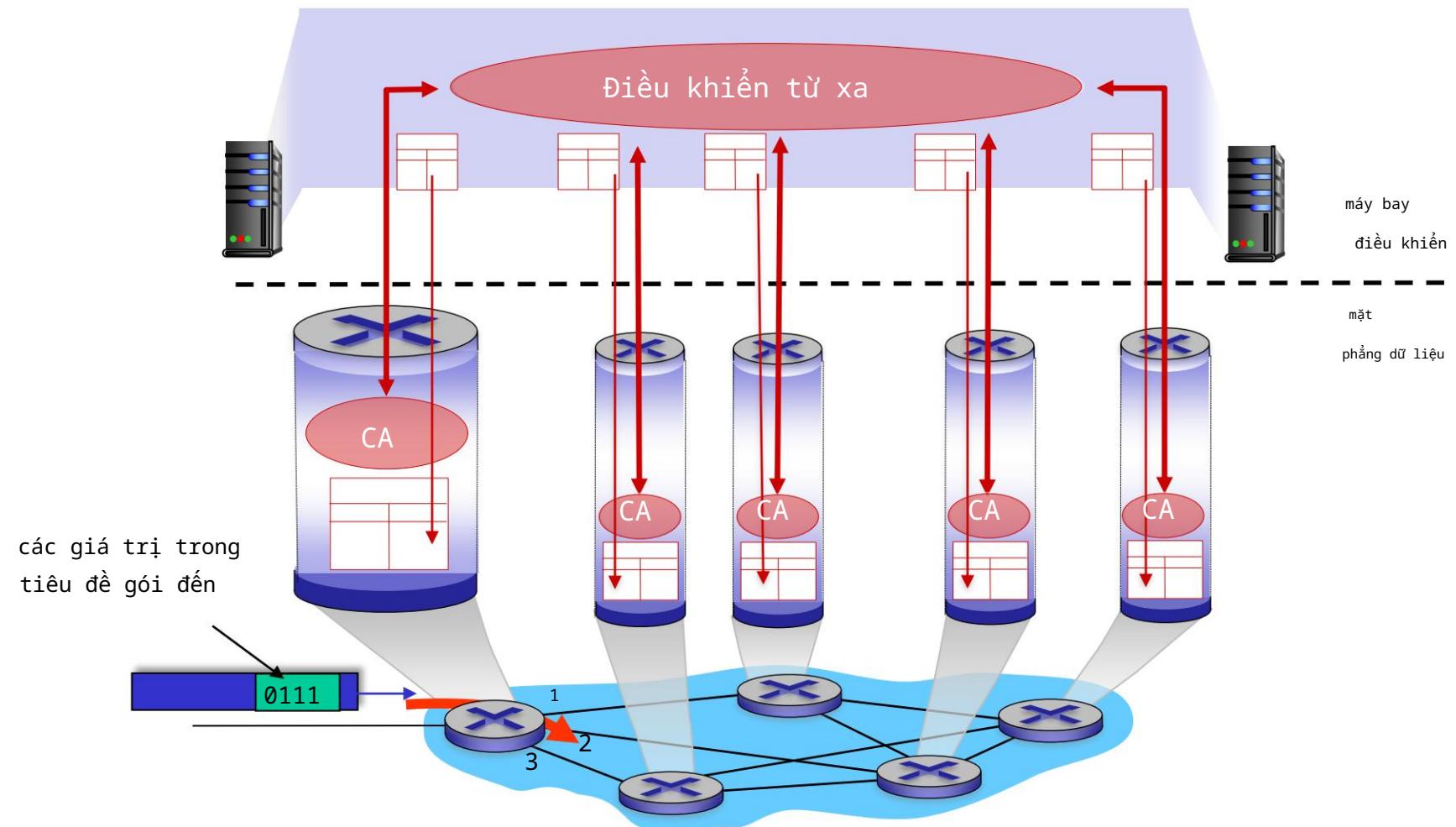
Mặt phẳng điều khiển mỗi bộ định tuyến

Các thành phần thuật toán định tuyến riêng lẻ trong mỗi và mọi bộ định tuyến tương tác trong mặt phẳng điều khiển



Mặt phẳng điều khiển Mạng được xác định bằng phần mềm (SDN)

Bộ điều khiển từ xa tính toán, cài đặt bảng chuyển tiếp trong bộ định tuyến



Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

trạng thái liên kết

vectơ khoảng cách

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo điều khiển Internet
giao thức



quản lý mạng, cấu hình

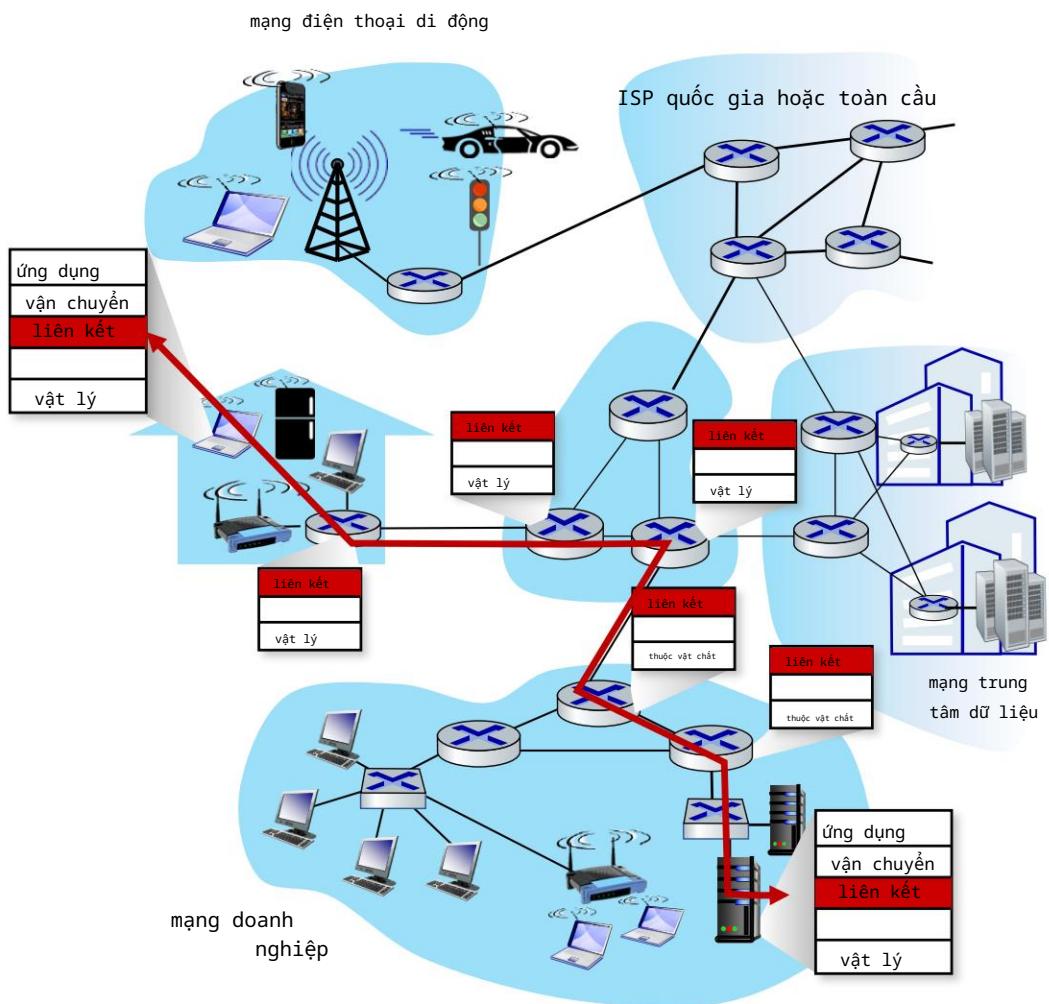
- SNMP
- NETCONF/YANG

giao thức định tuyến

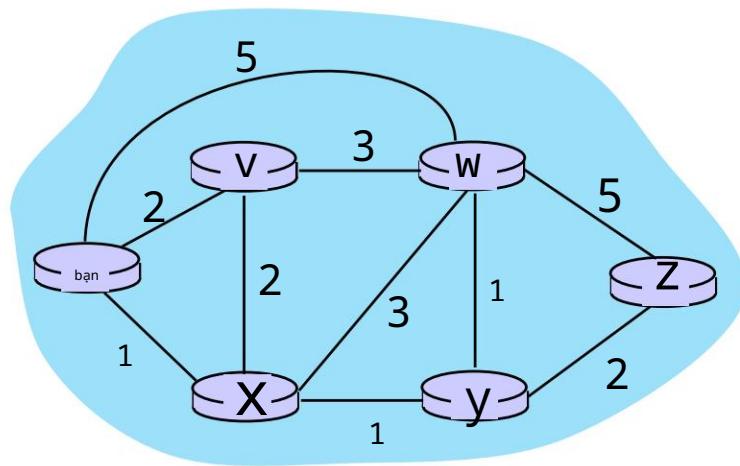
Mục tiêu của giao thức **định tuyến** : xác định các đường dẫn “tốt” (tư ơng ứng đường, các tuyến đường), từ máy chủ gửi đến máy chủ nhận, thông qua mạng các bộ định tuyến

đường dẫn: chuỗi các gói của bộ định tuyến đi từ máy chủ nguồn ban đầu nhất định đến máy chủ đích cuối cùng

“tốt”: “chi phí” thấp nhất, “nhanh nhất”, “ít tắc nghẽn nhất”
định tuyến: thách thức mạng “top 10” !



Trừu tượng hóa đồ thị: chi phí liên kết



đồ thị: $G = (N, E)$

N : bộ định tuyến = { u, v, w, x, y, z }

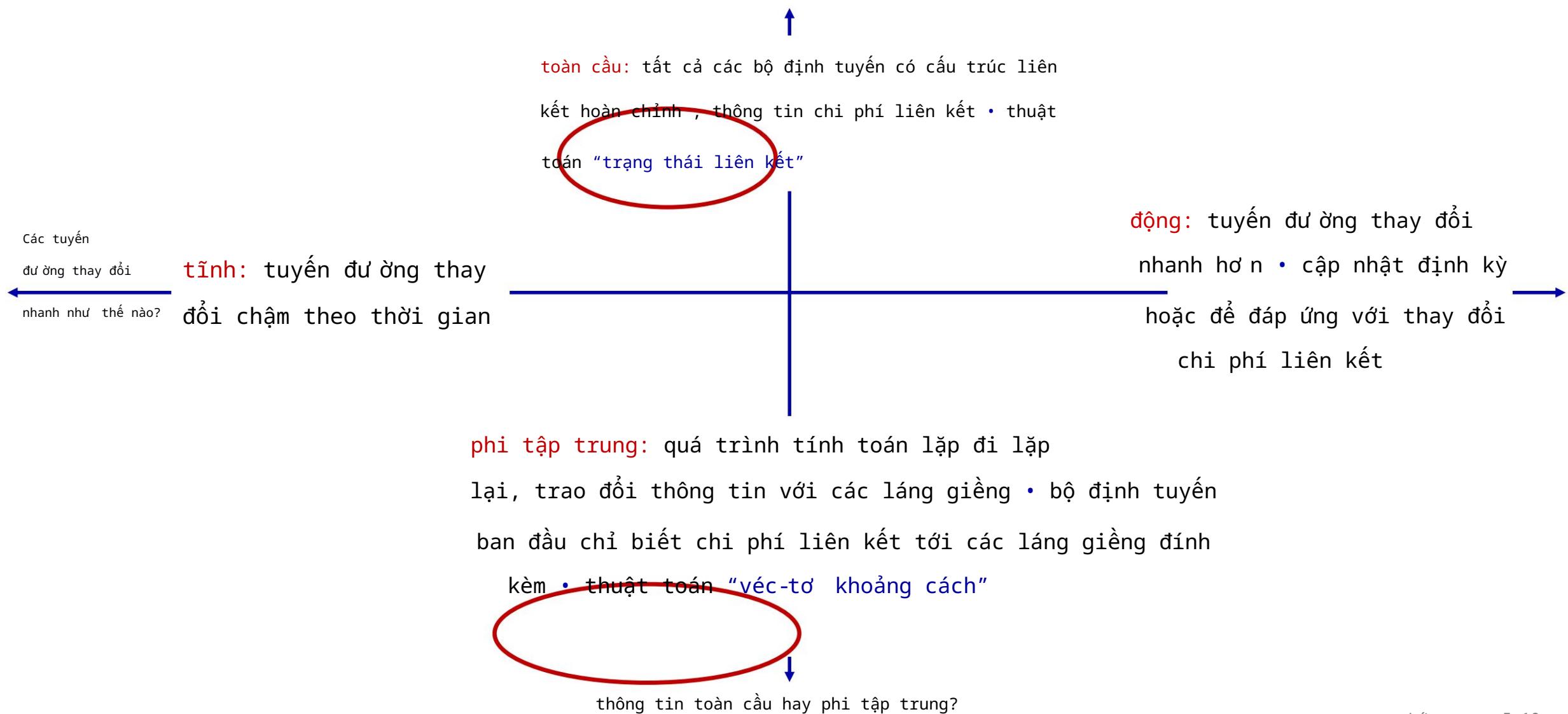
E : tập hợp các liên kết ={ $(u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z)$ }

ca, b : chi phí của liên kết **trực tiếp** kết nối a và b

ví dụ: $c_{w,z} = 5, c_{u,z} = \infty$

chi phí do nhà điều hành mạng xác định: có thể luôn bằng 1 hoặc tỷ lệ nghịch với băng thông hoặc tỷ lệ nghịch với tắc nghẽn

Phân loại thuật toán định tuyến



Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

trạng thái liên kết

vectơ khoảng cách

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Thuật toán định tuyến trạng thái liên kết của Dijkstra

tập trung: cấu trúc liên kết mạng, chi phí liên kết
đư ợc biết đến với tất cả các nút • đư ợc thực hiện

thông qua "phát trạng thái liên kết" • tất cả các nút
có cùng thông tin

tính toán **các đường dẫn có chi phí thấp nhất** từ một nút ("nguồn") đến tất cả các nút
khác • cung cấp **bảng chuyển tiếp** cho nút đó

lặp lại: sau k lần lặp lại, biết đư ợc
đư ờng đi có chi phí thấp nhất **tới k đích**

ký hiệu

cx,y : chi phí liên kết trực
tiếp từ nút x đến y ; $= \infty$ nếu không
phải là hàng xóm trực tiếp $D(v)$:

ước tính hiện tại về chi phí của đư ờng
đi có chi phí thấp nhất từ nguồn đến
đích v

$p(v)$: nút tiền nhiệm dọc theo đư ờng
dẫn từ nguồn tới v N' : tập hợp các
nút có đư ờng dẫn chi phí nhỏ nhất đã
biết rõ ràng

Thuật toán định tuyến trạng thái liên kết của Dijkstra

1 Khởi tạo:

2 N' = {u}

/* tính toán đư ờng đi có chi phí thấp nhất từ u đến tất cả các nút khác */

3 với mọi nút v

4 nếu v liền kề với

/* u ban đầu chỉ biết chi phí đư ờng dẫn trực tiếp cho các hàng xóm trực tiếp */ /*

cu, v 5 ~~khác~~ $D(v) = \infty$

như ng có thể không phải là chi phí tối thiểu ! */

6

8 Vòng

9 thêm ~~vào~~ vào N' không hiện tại $D(w)$ chia cho $D(w)$ là $\min_{v \in N'} D(v)$ nhỏ nhất

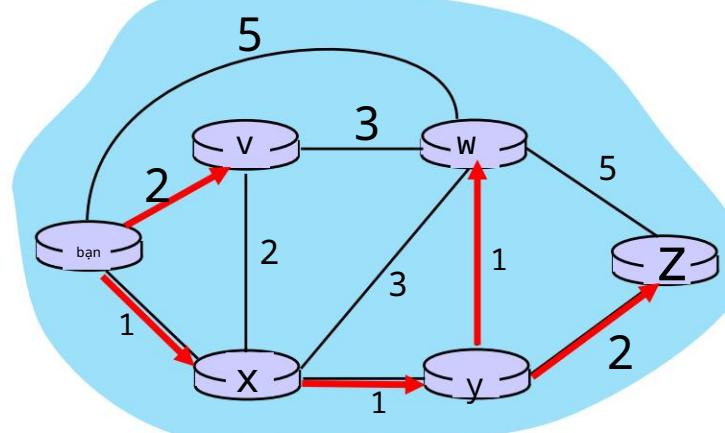
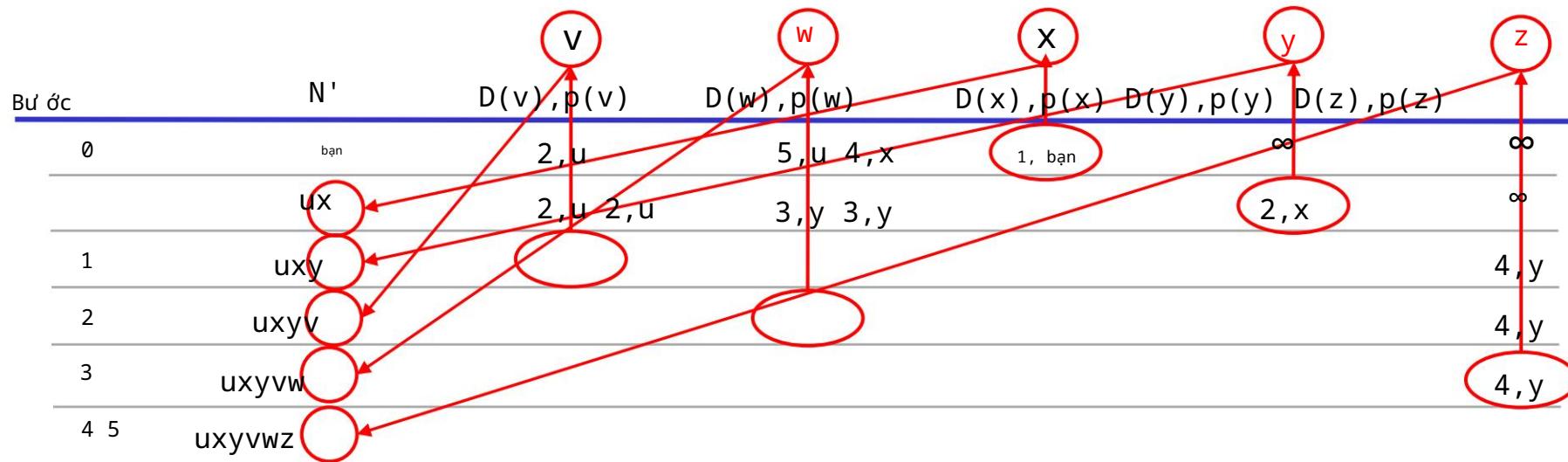
10 không thuộc N' :

11

12 $D(v) = \text{tối thiểu} (D(v), D(w) + cw, v)$

13 /* chi phí đư ờng đi nhỏ nhất mới đến v là đư ờng đi có chi phí nhỏ nhất cũ đến v hoặc đư ờng đi
cả các nút thông qua thấp nhất đã biết đến w cộng với chi phí trực tiếp từ w đến v */ 14 15 cho đến khi tất

Thuật toán Dijkstra: một ví dụ

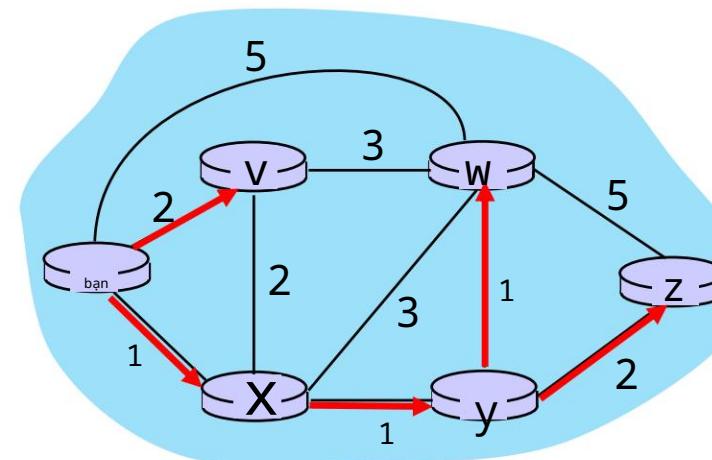


Khởi tạo (bu ớc 0): Với mọi a : nếu a kề với u thì $D(a) = c_{u,a}$

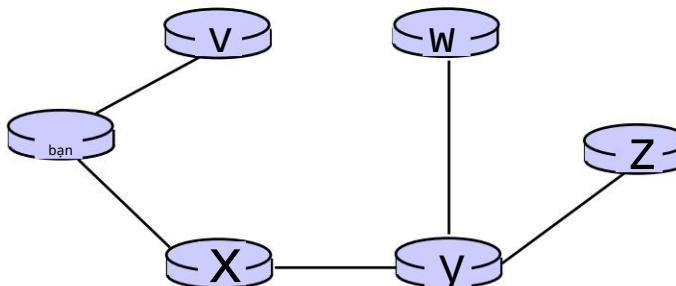
↓
tìm a không thuộc N' sao cho $D(a)$ là số nhỏ nhất thêm a vào N' cập nhật $D(b)$ cho tất cả b liền kề với a và không thuộc N' :

$$D(b) = \text{tối thiểu} (D(b), D(a) + c_{a,b})$$

Thuật toán Dijkstra: một ví dụ



kết quả cây đư ờng dẫn chi phí thấp nhất từ u:



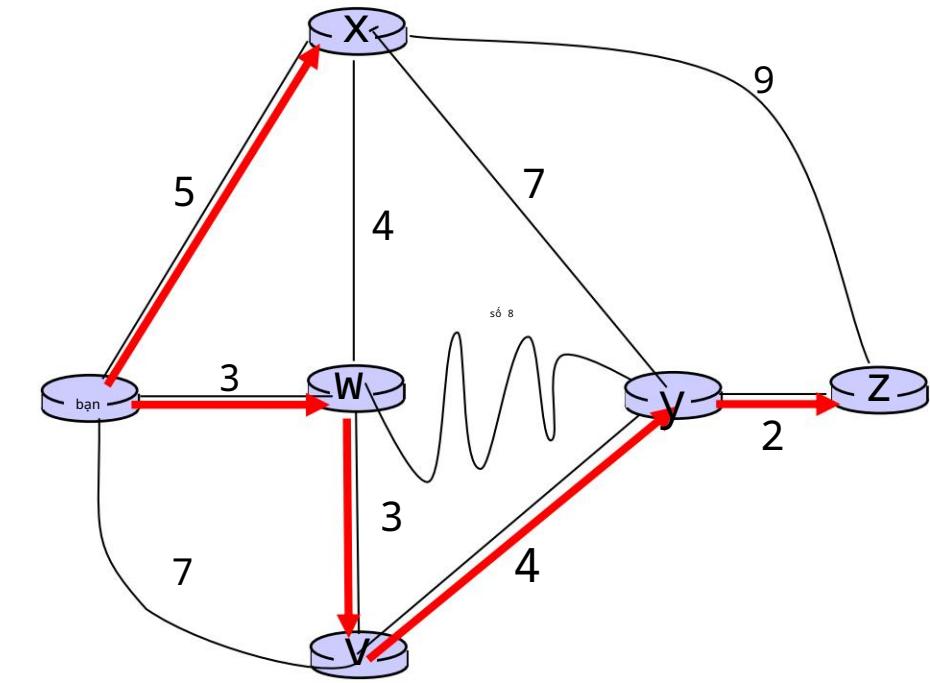
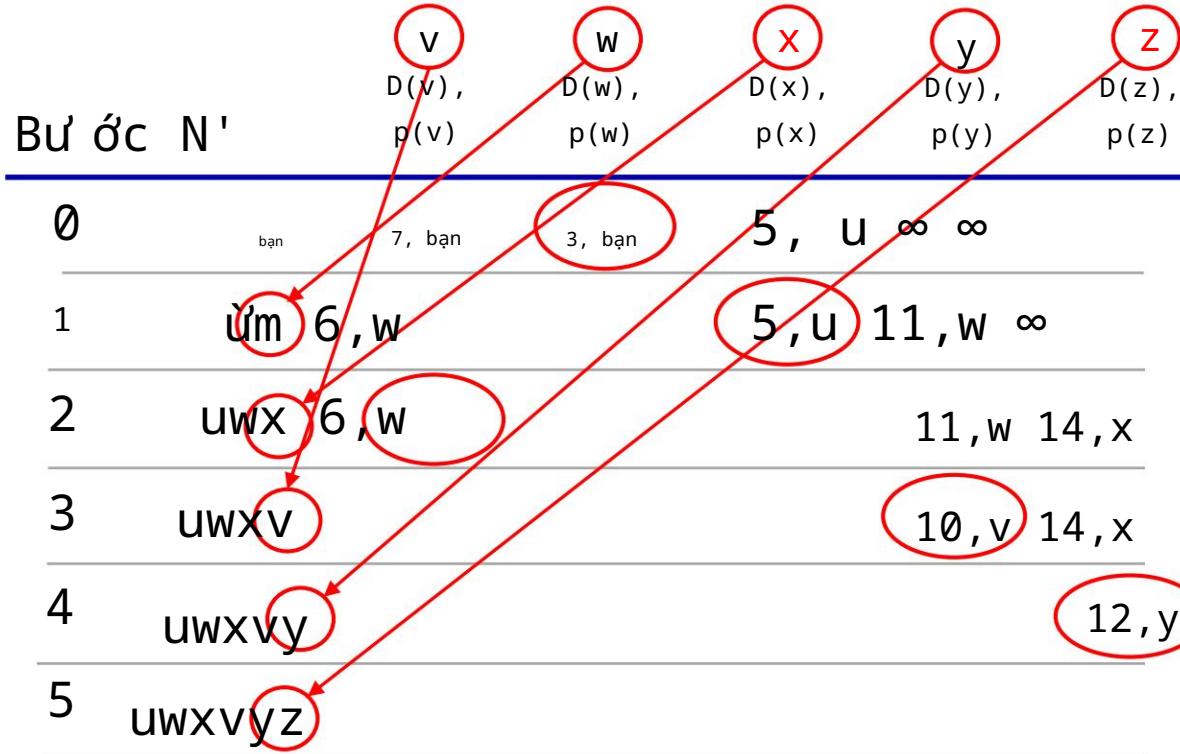
bảng chuyển tiếp kết quả trong u:

liên kết đi đích	
v	(u, v)
x	(u, x)
y	(u, x)
w	(u, x)
x	(u, x)

tuyến đư ờng từ u đến v trực tiếp

định tuyến từ u đến
tất cả các đích khác
qua x

Thuật toán Dijkstra: một ví dụ khác



ghi chú:

xây dựng cây đường đi có chi phí thấp nhất bằng cách lần theo các nút tiền thân

Có thể tồn tại các mối ràng buộc (có thể bị phá vỡ tùy ý)

Thuật toán Dijkstra: thảo luận

độ phức tạp của thuật toán: n nút

mỗi n lần lặp: cần kiểm tra tất cả các nút, w , không thuộc N

so sánh $n(n+1)/2$: $O(n^2)$ độ phức tạp

thể triển khai hiệu quả hơn: $O(n \log n)$

Độ phức tạp của thông điệp:

mỗi bộ định tuyến phải quảng bá thông tin trạng thái liên kết của nó tới n bộ định tuyến khác

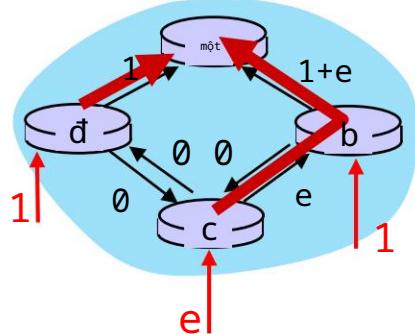
thuật toán quảng bá hiệu quả (và thú vị!): $O(n)$ giao cắt liên kết để phổ biến một thông báo quảng bá từ một nguồn mỗi thông báo của bộ định tuyến đi qua $O(n)$ liên kết: độ phức tạp tổng thể của thông báo: $O(n^2)$

Thuật toán Dijkstra: dao động có thể

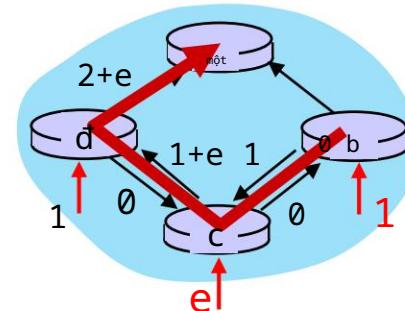
khi chi phí liên kết phụ thuộc vào lưu lượng, tuyến đường có thể **dao động**

kịch bản mẫu:

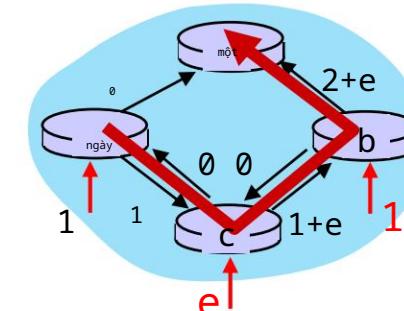
- định tuyến đến đích a, lưu lượng đi vào tại d, c, b với tốc độ 1, e (<1), 1
- chi phí liên kết có **hư ớng** và **lưu lượng- sự phụ thuộc**



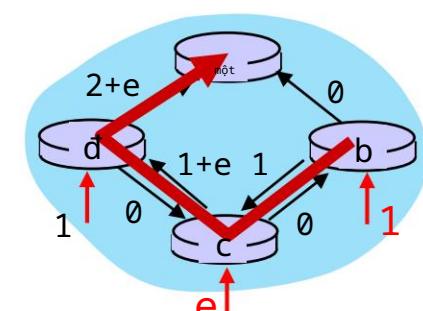
ban đầu



đưa ra các chi phí
này, tìm định tuyến
mới.. dẫn đến chi phí mới



đưa ra các chi phí
này, tìm định tuyến
mới.. dẫn đến chi phí mới



đưa ra các chi phí
này, tìm định tuyến
mới.. dẫn đến chi phí mới

Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

trạng thái liên kết

vectơ khoảng cách

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Thuật toán vectơ khoảng cách

Dựa trên phư ơng trình Bellman-Ford (BF) (lập trình động):

phư ơng trình Bellman-Ford

Cho $D_x(y)$: chi phí của **đường đi có chi phí nhỏ nhất** từ x đến y .

Sau đó:

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

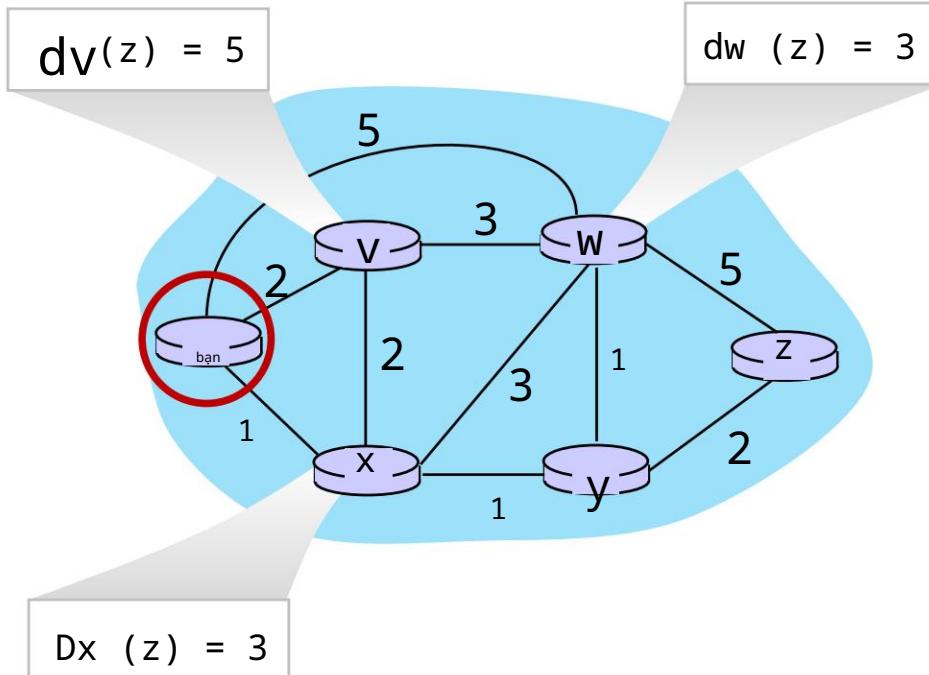
\min chiếm tất cả các lân cận v của x

chi phí trực tiếp của liên kết từ x đến v

chi phí đường dẫn chi phí nhỏ nhất ước tính của v đến y

Ví dụ Bellman-Ford

Giả sử rằng các nút lân cận của u, x, v, w , biết rằng đối với đích z :



Phương trình Bellman-Ford nói:

$$Du(z) = \min \{ cu,v + Dv(z),$$

$$cu,x + Dx(z),$$

$$cu,w + Dw(z) \}$$

$$5, 1 + 3, 5 + 3 \} = 4$$

nút đạt được giá trị tối thiểu (x) là
bước nhảy tiếp theo trên đường ước
tính có chi phí thấp nhất tới đích (z)

Thuật toán vectơ khoảng cách

ý chính:

theo thời gian, mỗi nút gửi ước tính véc-tơ khoảng cách của riêng mình đến hàng xóm

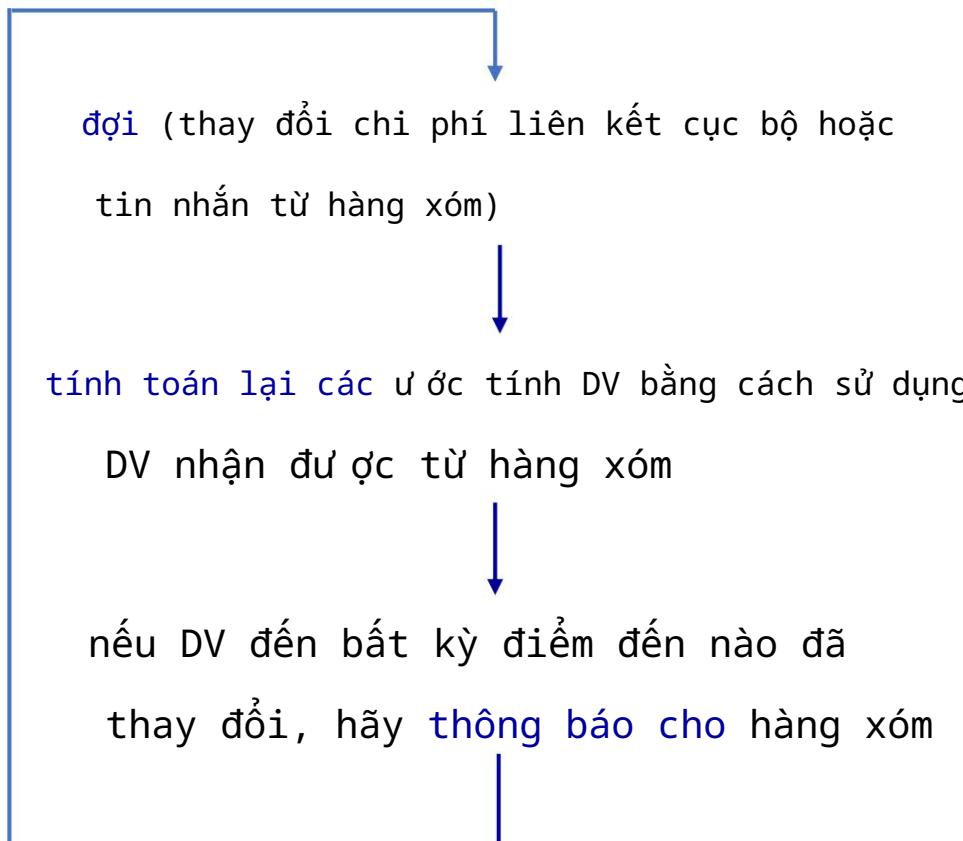
khi x nhận được ước tính DV mới từ bất kỳ hàng xóm nào, nó sẽ cập nhật DV của chính nó bằng phư ơng trình BF:

$$Dx(y) = \min_v \{cx, v + Dv(y)\} \text{ cho mỗi nút } y \in N$$

trong các điều kiện tự nhiên nhỏ, ước tính $Dx(y)$ hội tụ về chi phí thực tế thấp nhất $dx(y)$

Thuật toán vectơ khoảng cách:

mỗi nút:



lặp đi lặp lại, không đồng bộ: mỗi cục bộ

lặp lại do: thay

đổi chi phí liên kết

cục bộ Bản tin cập nhật DV từ hàng xóm

phân phối, tự dừng: mỗi nút chỉ thông báo cho các nút lân cận khi DV của nó thay đổi các nút lân cận sau đó thông báo

hang xom - chỉ khi cần thiết

không nhận được thông báo; không có hành động nào được thực hiện!

Vector khoảng cách: ví dụ

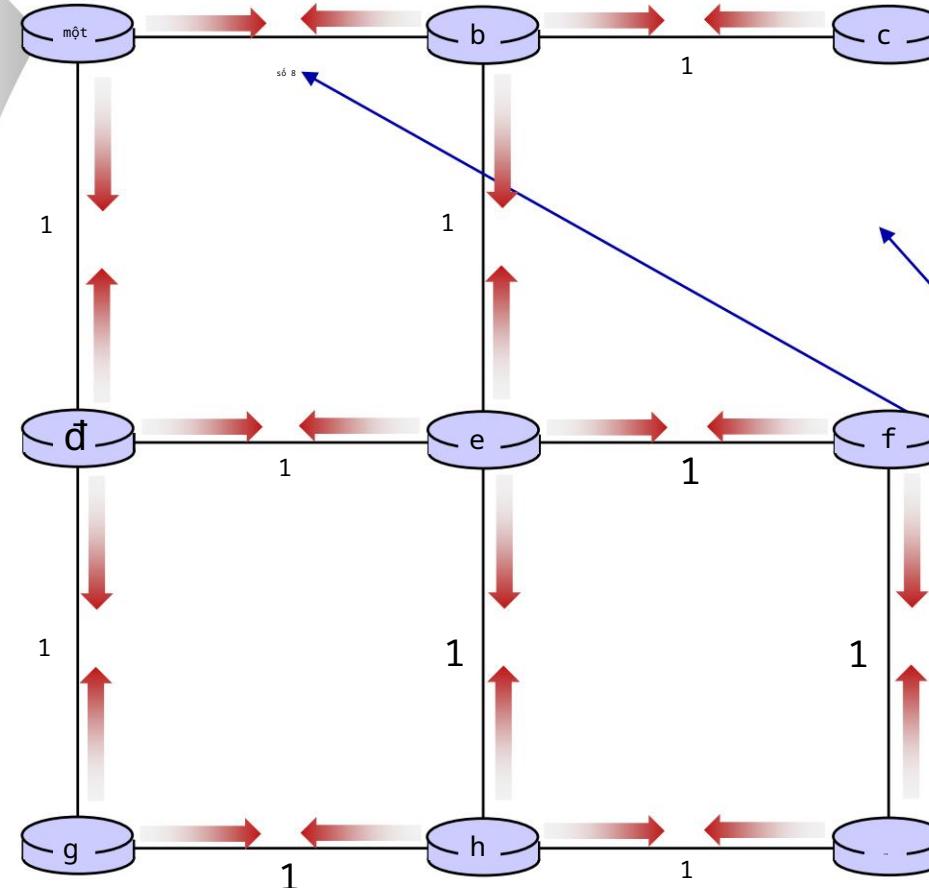


$t=0$

Tất cả các nút đều
có [khoảng cách ước tính](#)
đến hàng
xóm gần nhất (chỉ)

Tất cả các nút [gửi](#)
[vectơ khoảng cách](#)
[cục bộ của chúng](#) tới
các nút lân cận

DV trong mót:
$\text{Đ}_{\text{mót}}(a) = 0$
$\text{Đ}_{\text{mót}}(b) = 8$
$\text{Đ}_{\text{mót}}(c) = \infty$
$\text{Đ}_{\text{mót}}(d) = 1$
$\text{Đ}_{\text{mót}}(e) = \infty$
$\text{Đ}_{\text{mót}}(f) = \infty$
$\text{Đ}_{\text{mót}}(g) = \infty$
$\text{Đ}_{\text{mót}}(h) = \infty$
$\text{Đ}_{\text{mót}}(i) = \infty$

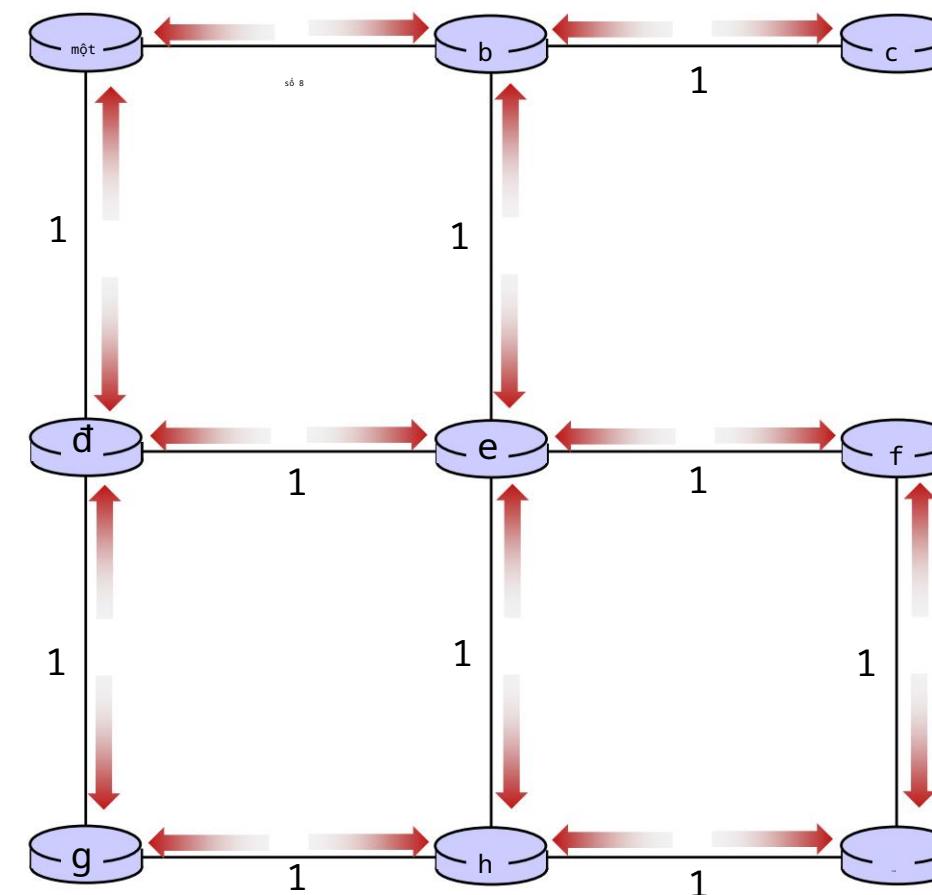


Ví dụ về vectơ khoảng cách: phép lặp



$t=1$

Tất cả các
nút: nhận vectơ
khoảng cách từ các
nút lân cận tính
tổng vectơ khoảng cách
cục bộ mới của chúng
gửi vectơ khoảng cách
cục bộ mới của chúng tới
các nút lân cận



Ví dụ về vectơ khoảng cách: phép lặp



$t=1$

Tất cả các

nút: nhận vectơ

khoảng cách từ các

nút lân cận tính

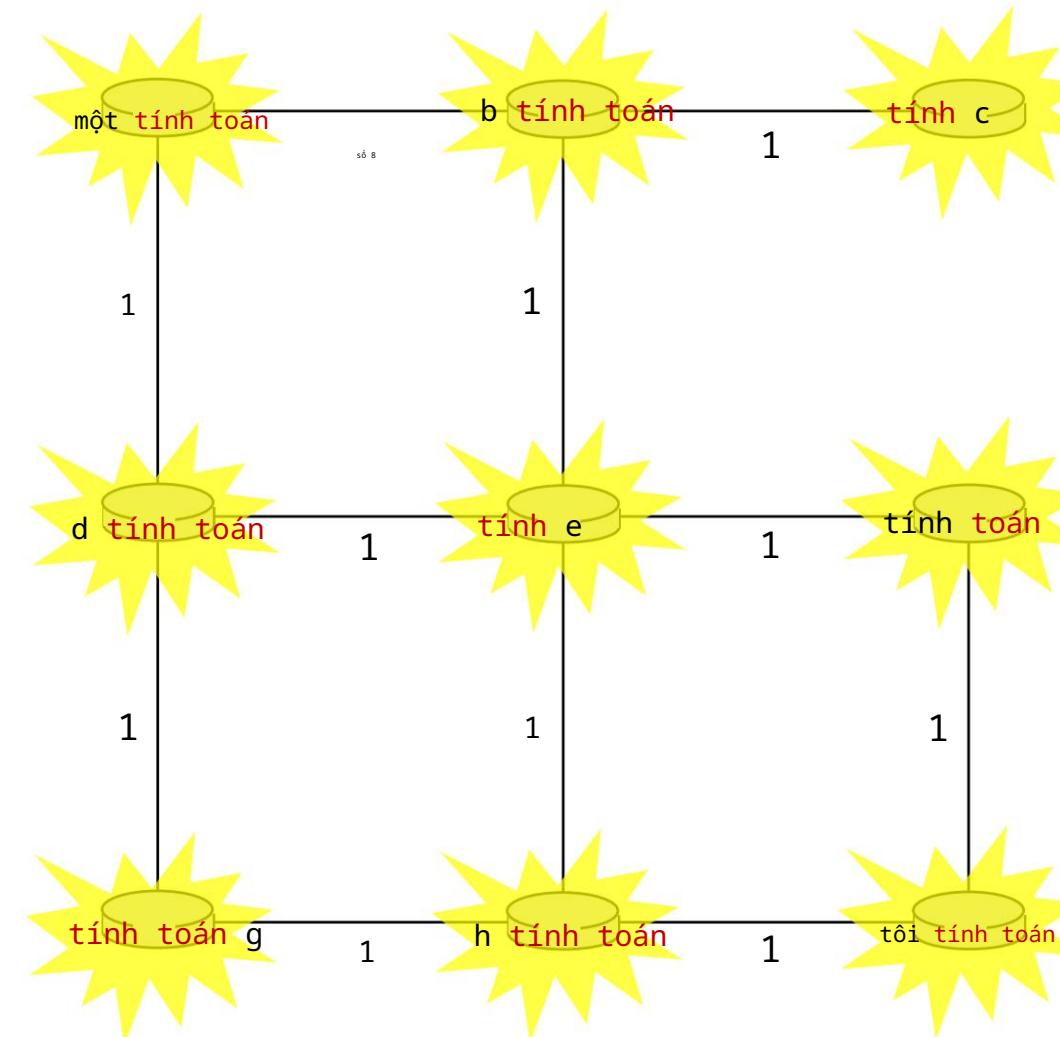
toán vectơ khoảng cách

cục bộ mới của chúng

gửi vectơ khoảng cách

cục bộ mới của chúng tới

các nút lân cận



Ví dụ về vectơ khoảng cách: phép lặp



$t=1$

Tất cả các

nút: nhận vectơ

khoảng cách từ các

nút lân cận tính

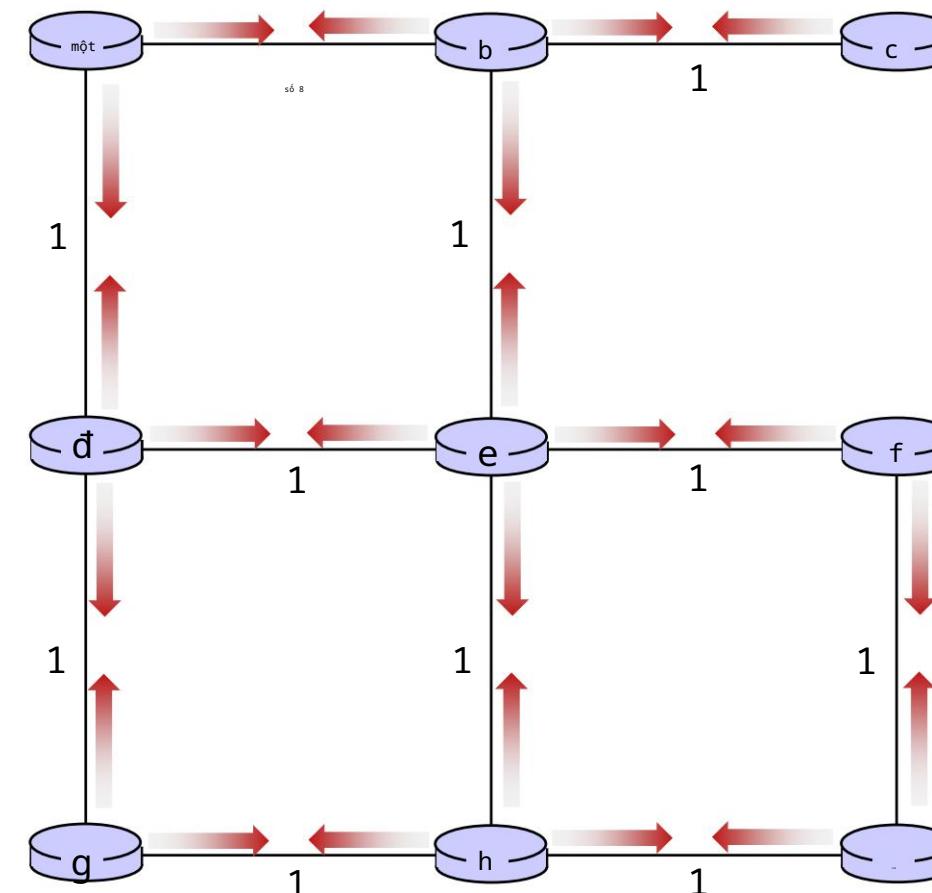
toán vectơ khoảng cách

cục bộ mới của chúng

gửi vectơ khoảng cách

cục bộ mới của chúng tới

các nút lân cận



Ví dụ về vectơ khoảng cách: phép lặp



$t=2$

Tất cả các

nút: nhận vectơ

khoảng cách từ các

nút lân cận tính

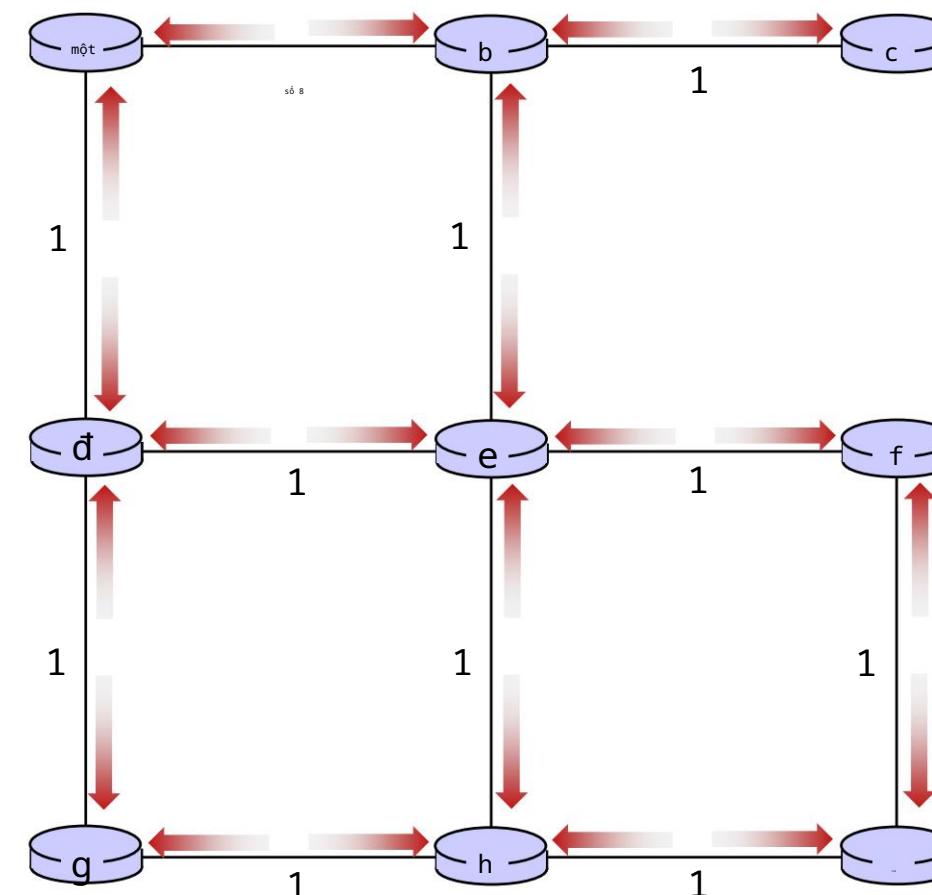
toán vectơ khoảng cách

cục bộ mới của chúng

gửi vectơ khoảng cách

cục bộ mới của chúng tới

các nút lân cận



Ví dụ về vectơ khoảng cách: phép lặp



t=2

Tất cả các

nút: nhận vectơ

khoảng cách từ các

nút lân cân tính

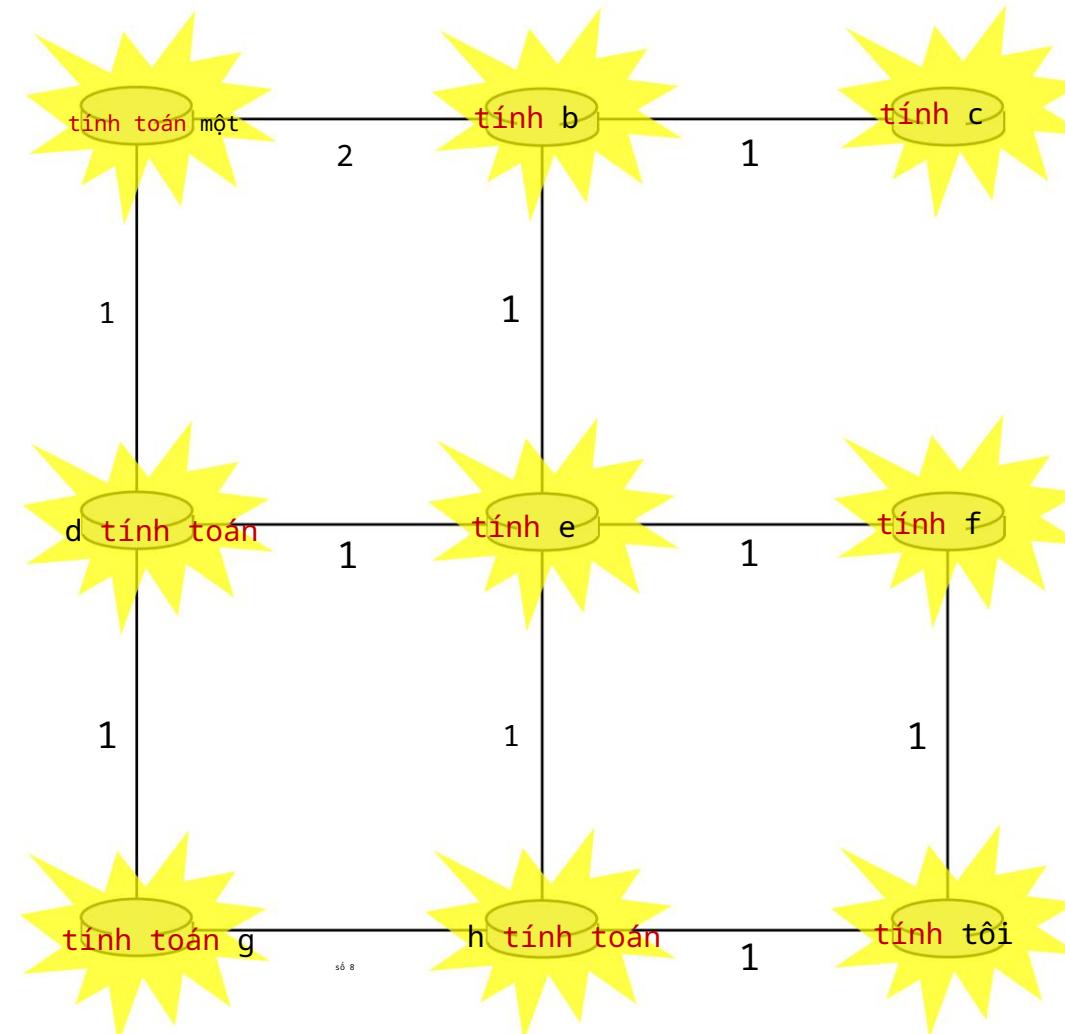
toán vectơ khoảng cách

cục bộ mới của chúng

gửi vectơ khoảng cách

cục bộ mới của chúng tôi

các nút lân cận



Ví dụ về vectơ khoảng cách: phép lặp



$t=2$

Tất cả các

nút: nhận vectơ

khoảng cách từ các

nút lân cận tính

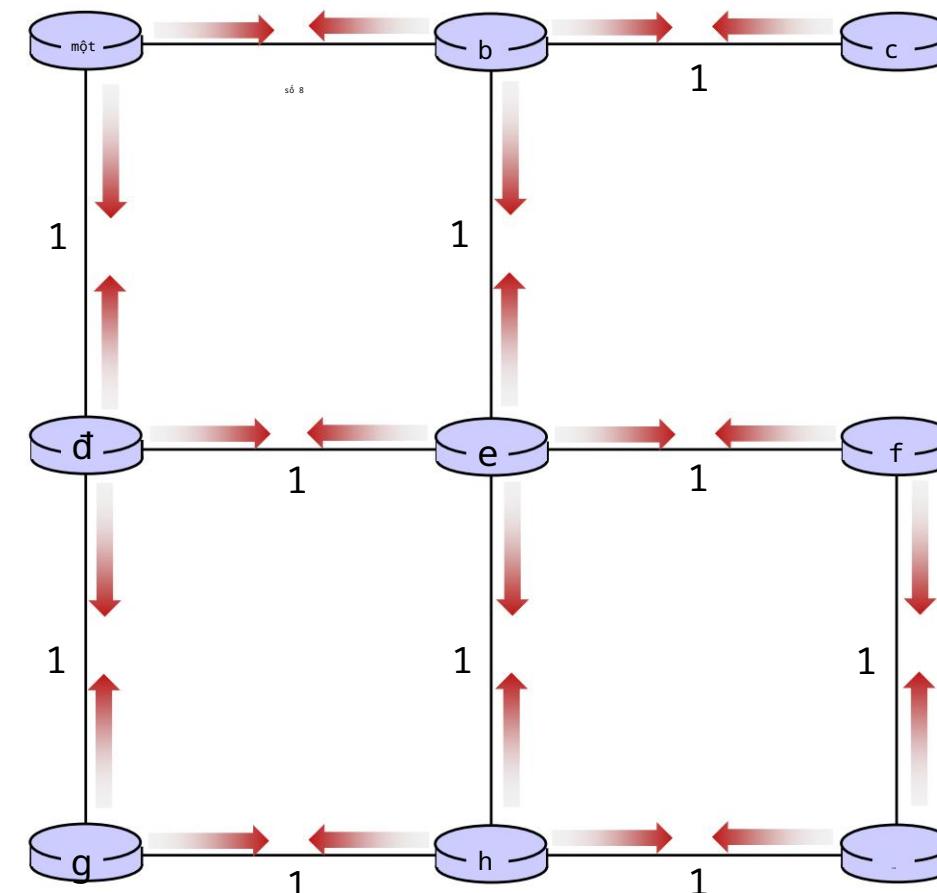
toán vectơ khoảng cách

cục bộ mới của chúng

gửi vectơ khoảng cách

cục bộ mới của chúng tới

các nút lân cận



Ví dụ về vectơ khoảng cách: phép lặp

.. và như thế

Tiếp theo chúng ta hãy xem các phép tính lặp tại các nút

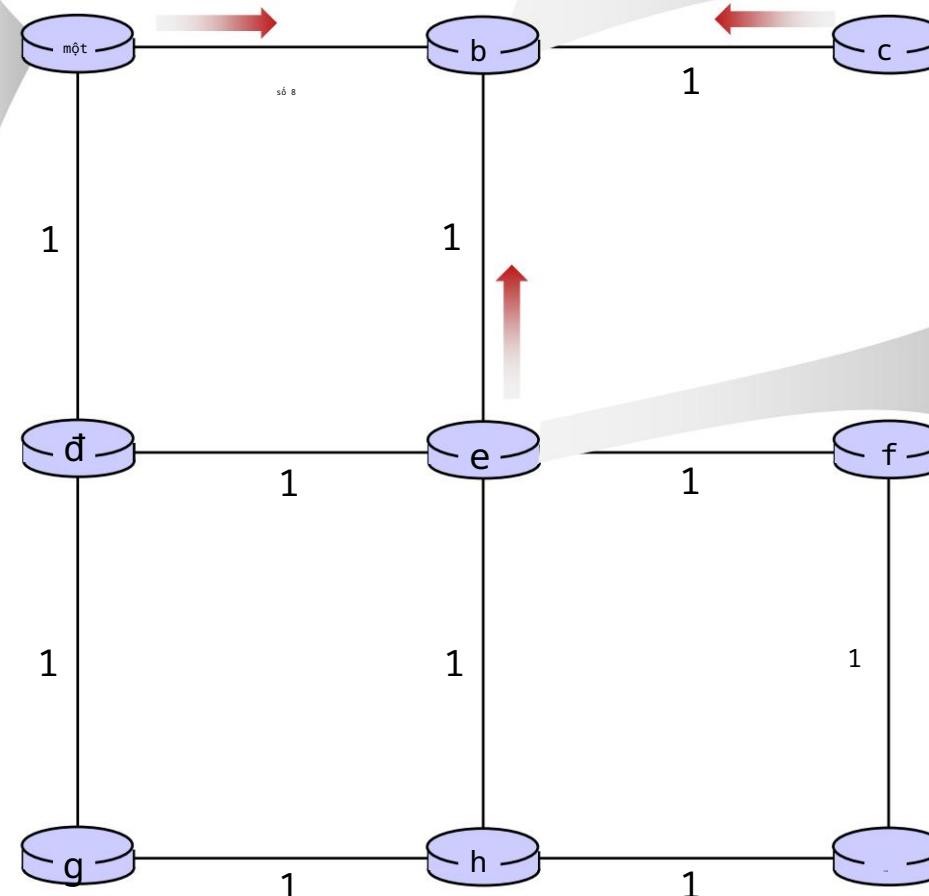
Ví dụ về vectơ khoảng cách: tính toán



$t=1$

b nhận DV
từ a, c, e

DV trong m ^ô t:
$D_m(a) = 0$
$D_m(b) = 8$
$D_m(c) = \infty$
$D_m(d) = 1$
$D_m(e) = \infty$
$D_m(f) = \infty$
$D_m(g) = \infty$
$D_m(h) = \infty$
$D_m(i) = \infty$



DV trong b:	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

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

DV trong e:
$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

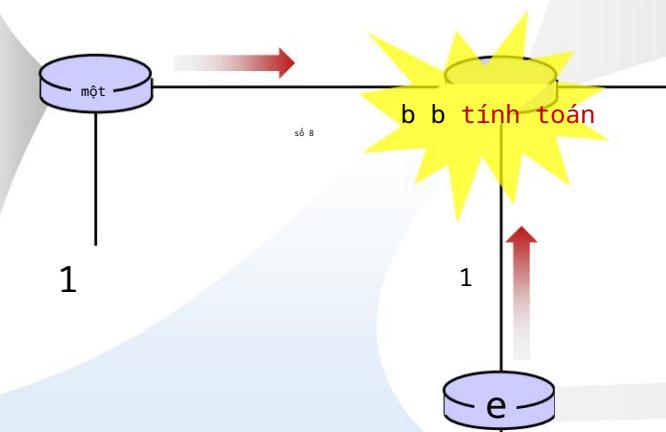
Ví dụ về vectơ khoảng cách: tính toán



t=1

b nhận DV
từ a, c,
e, tính:

DV trong m
Đà (a)=0
Đà (b) = 8
Đa (c) = ∞
Đà (d) = 1
Đa (e) = ∞
Đa (f) = ∞
Đa (g) = ∞
Đa (h) = ∞
Đa (i) = ∞



DV trong b:	
Db (a) = 8	Db (f) = ∞
Db (c) = 1	Db (g) = ∞
Db (d) = ∞	Db (h) = ∞
Db (e) = 1	Db (i) = ∞

DV trong c:
dc (a) = ∞
dc (b) = 1
dc (c) = 0
dc (d) = ∞
dc (e) = ∞
dc (f) = ∞
dc (g) = ∞
dc (h) = ∞
dc (i) = ∞

DV trong e:
De (a) = ∞
De (b) = 1
De (c) = ∞
De (d) = 1
De (e) = 0
De (f) = 1
De (g) = ∞
De (h) = 1
De (i) = ∞

DV trong b:	
Db (a) = 8	Db (f) = 2
Db (c) = 1	Db (g) = ∞ tôi
Db (d) = 2	Db (h) = 2
Db (e) = 1	Db (i) = ∞

$$\begin{aligned}
 Db(a) &= \min\{cb, a+Da(a), cb, c+Dc(a), cb, e+De(a)\} = \min\{8, \infty, \infty\} = 8 \\
 Db(c) &= \min\{cb, a+Da(c), cb, c+Dc(c), cb, b, e+De(c)\} = \min\{\infty, 1, \infty\} = 1 \\
 Db(d) &= \min\{cb, a+Da(d), cb, c+Dc(d), cb, b, e+De(d)\} = \min\{9, 2, \infty\} = 2 \\
 Db(e) &= \min\{cb, a+Da(e), cb, c+Dc(e), cb, b, e+De(e)\} = \min\{\infty, \infty, 1\} = 1 \\
 Db(f) &= \min\{cb, a+Da(f), cb, c+Dc(f), cb, b, e+De(f)\} = \min\{\infty, \infty, 2\} = 2 \\
 Db(g) &= \min\{cb, a+Da(g), cb, c+Dc(g), cb, b, e+De(g)\} = \min\{\infty, \infty, \infty\} = \infty \\
 Db(h) &= \min\{cb, a+Da(h), cb, c+Dc(h), cb, b, e+De(h)\} = \min\{\infty, \infty, 2\} = 2 \\
 Db(i) &= \min\{cb, a+Da(i), cb, c+Dc(i), cb, b, e+De(i)\} = \min\{\infty, \infty, \infty\} = \infty
 \end{aligned}$$

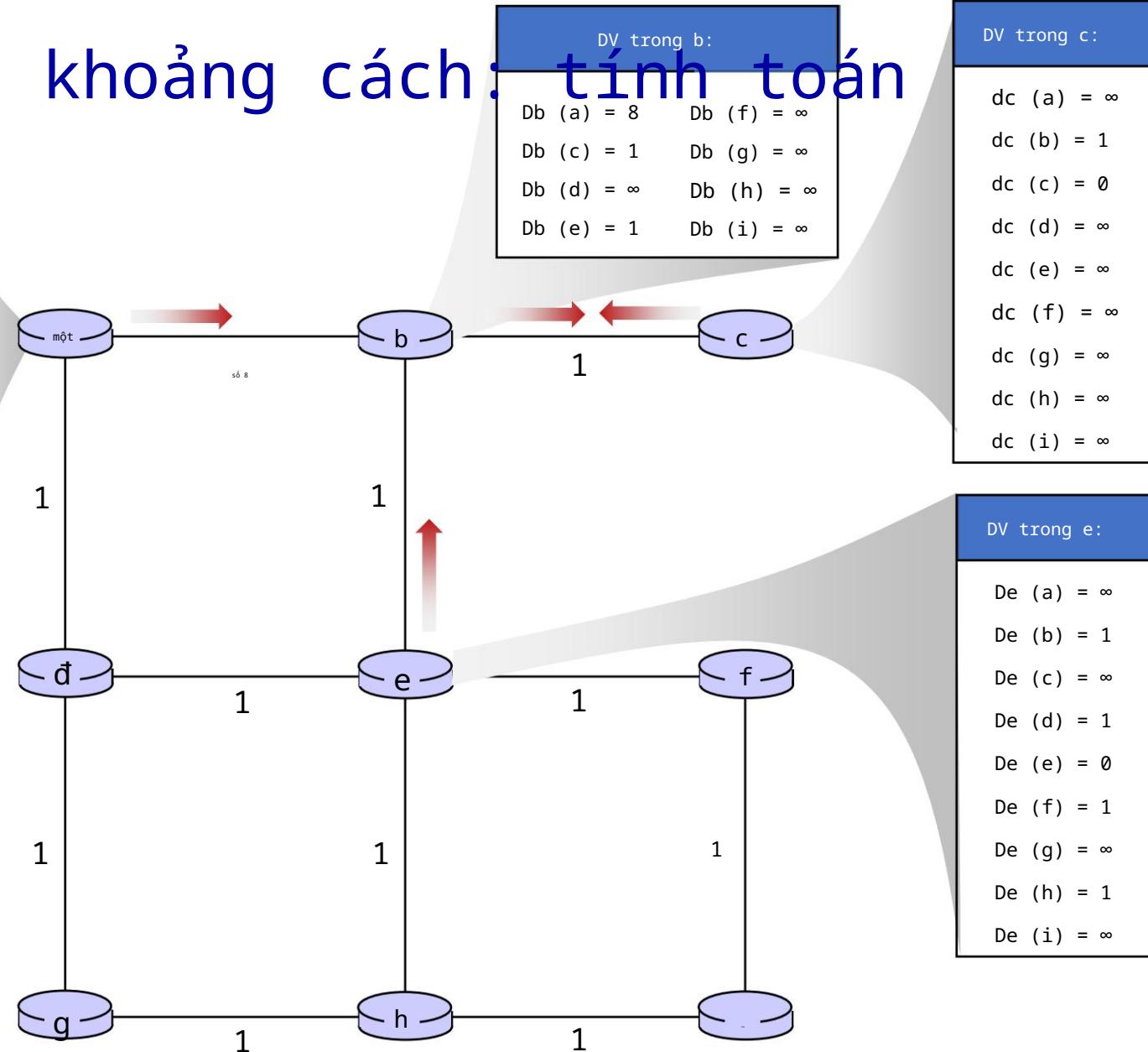
Ví dụ về vectơ khoảng cách: tính toán



$t=1$

c nhận DV từ
b

DV trong m:
$D_m(a) = 0$
$D_m(b) = 8$
$D_m(c) = \infty$
$D_m(d) = 1$
$D_m(e) = \infty$
$D_m(f) = \infty$
$D_m(g) = \infty$
$D_m(h) = \infty$
$D_m(i) = \infty$



Ví dụ về vectơ khoảng cách: tính toán



$t=1$

c nhận DV

từ b tính toán:

một

số 8

1

1

1



DV trong b:

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

DV trong c:

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

$$dc(a) = \min\{cc, b+Db(a)\} = 1 + 8 = 9$$

$$dc(b) = \min\{cc, b+Db(b)\} = 1 + 0 = 1$$

$$dc(d) = \min\{cc, b+Db(d)\} = 1 + \infty = \infty$$

$$dc(e) = \min\{cc, b+Db(e)\} = 1 + 1 = 2$$

$$dc(f) = \min\{cc, b+Db(f)\} = 1 + \infty = \infty$$

$$dc(g) = \min\{cc, b+Db(g)\} = 1 + \infty = \infty$$

$$dc(h) = \min\{cc, b+Db(h)\} = 1 + \infty = \infty$$

$$dc(i) = \min\{cc, b+Db(i)\} = 1 + \infty = \infty$$

e

h

f

DV trong c:

$dc(a) = 9$
$dc(b) = 1$
$dc(c) = 0$
$dc(d) = 2$
$dc(e) = \infty$
$dc(f) = \infty$
$dc(g) = \infty$
$dc(h) = \infty$
$dc(i) = \infty$

* Xem các bài tập tư duy tác trực tuyến để biết thêm ví dụ: http://gaia.cs.umass.edu/kurose_ross/interactive/

Ví dụ về vectơ khoảng cách: tính toán

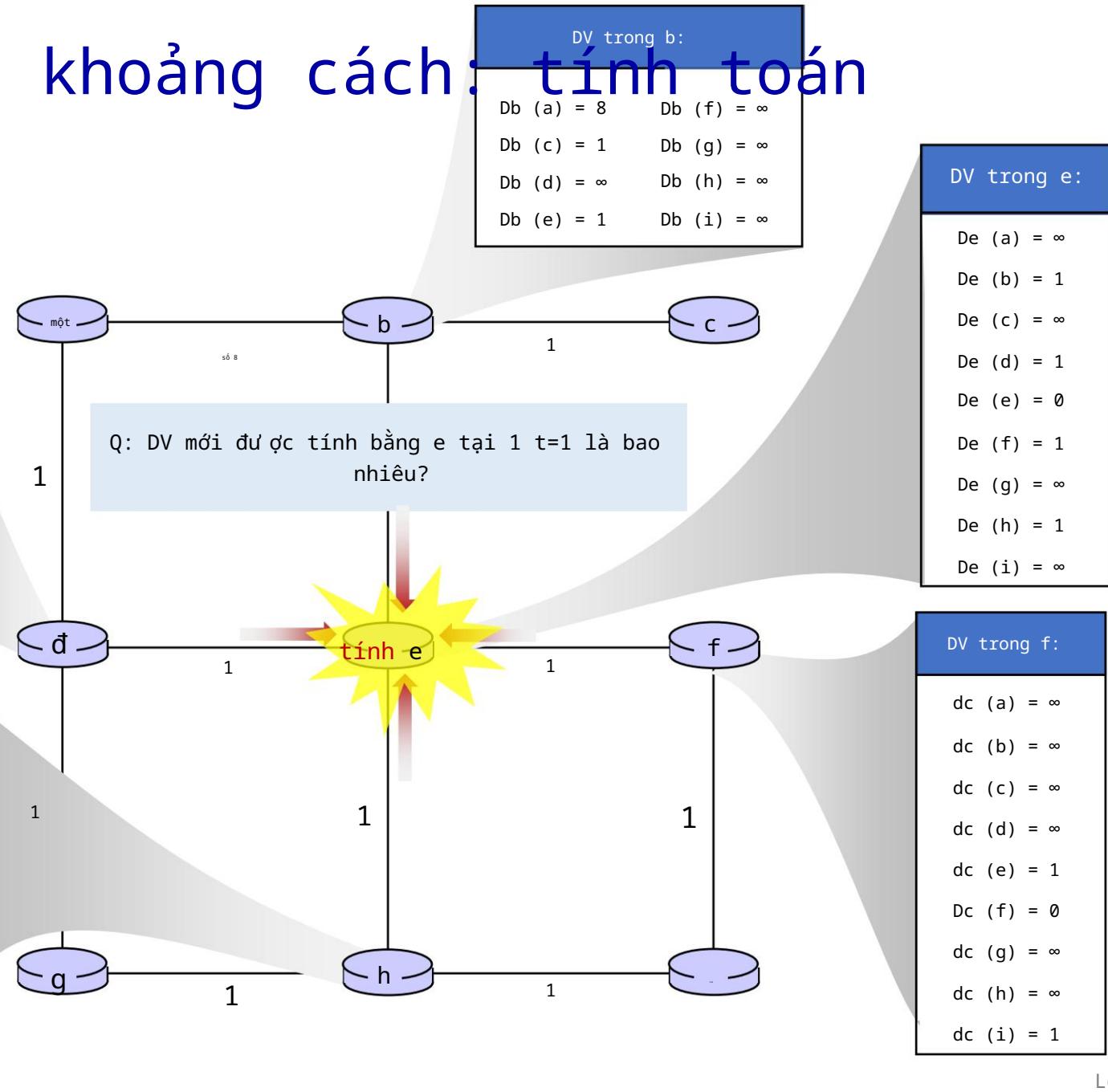


t=1

e nhận DV
từ b, d, f, h

DV trong d:
dc (a) = 1
dc (b) = ∞
dc (c) = ∞
dc (d) = 0
dc (e) = 1
dc (f) = ∞
dc (g) = 1
dc (h) = ∞
dc (i) = ∞

DV trong giờ:
dc (a) = ∞
dc (b) = ∞
dc (c) = ∞
dc (d) = ∞
dc (e) = 1
dc (f) = ∞
dc (g) = 1
dc (h) = 0
dc (i) = 1



Vectơ khoảng cách: khuếch tán thông tin trạng thái

Giao tiếp lặp, các bước tính toán khuếch tán thông tin qua mạng:



$t=0$ trạng thái của c tại $t=0$ chỉ tại **c**



trạng thái của c tại $t=0$ đã lan truyền tới b và có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 1 bước nhảy, tức là tại **b**



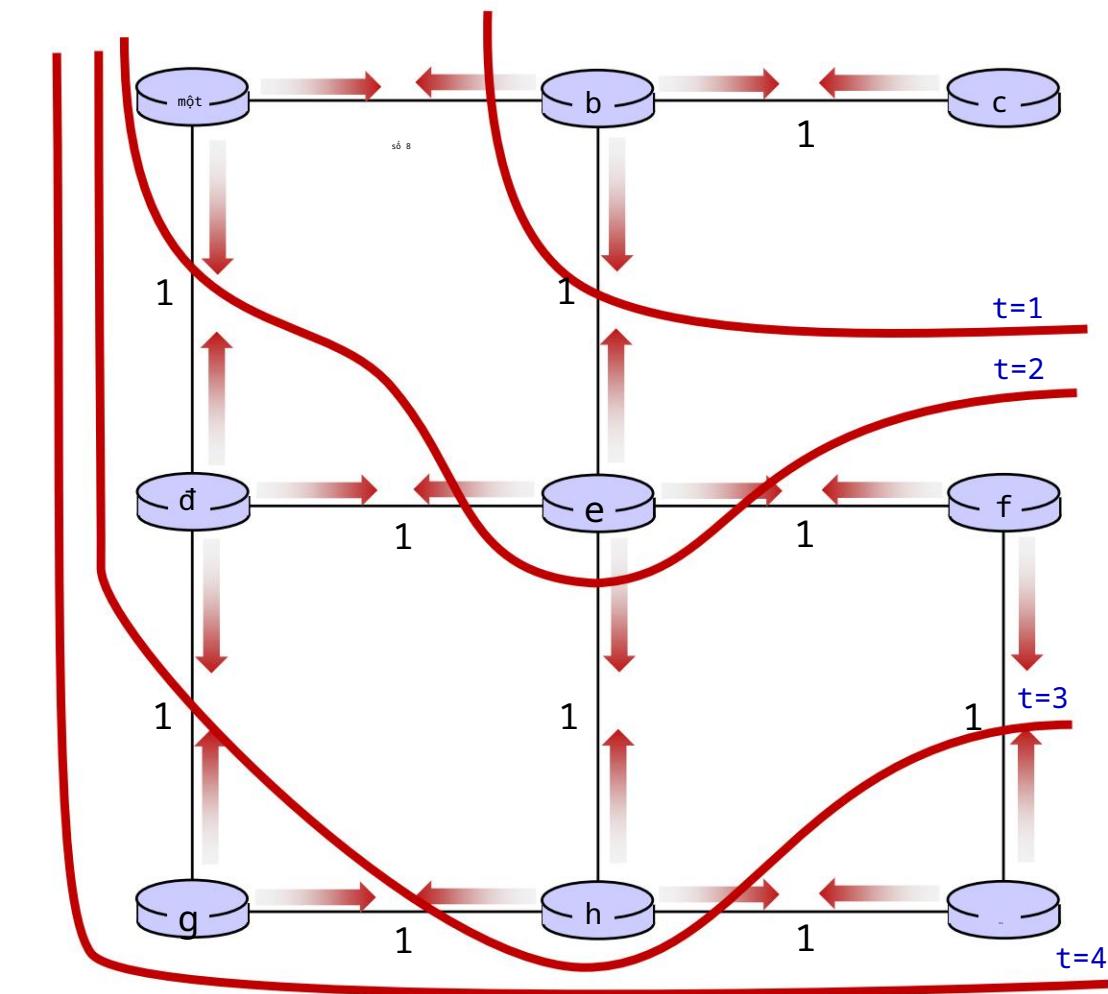
trạng thái của c tại $t=0$ bây giờ có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 2 bước nhảy, nghĩa là tại **b** và bây giờ tại a, e cũ ngay



trạng thái của c tại $t=0$ có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 3 bước nhảy, nghĩa là tại **b,a,e** và bây giờ là tại d,f,h



trạng thái của c tại $t=0$ có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 4 bước nhảy, tức là tại **b,a,e, d, f, h** và bây giờ là cả g,i



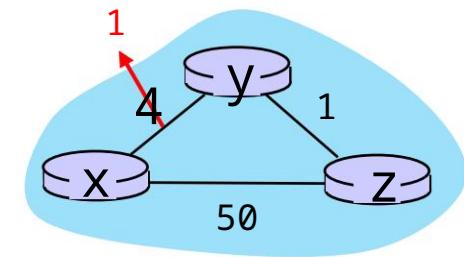
Vectơ khoảng cách: thay đổi chi phí liên kết

chi phí liên kết thay đổi:

nút phát hiện **thay đổi chi phí liên kết cục**

bộ cập nhật thông tin định tuyến, tính toán lại DV cục

bộ nếu DV thay đổi, **Thông báo cho hàng xóm**



t0 : y phát hiện thay đổi chi phí liên kết, cập nhật DV của nó, thông báo cho các láng giềng của nó.

**"tin tốt truyền
đi nhanh chóng"**

t1 : z nhận cập nhật từ y, cập nhật bảng của nó, tính giá trị nhỏ nhất mới

chi phí cho x, gửi cho các láng giềng của nó DV của nó.

t2 : y nhận cập nhật của z, cập nhật bảng khoảng cách của nó. Chi phí nhỏ nhất của y không thay đổi, vì vậy y không gửi tin nhắn cho z.

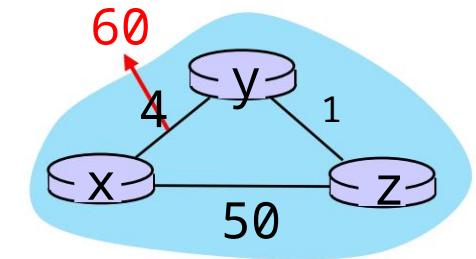
Vectơ khoảng cách: thay đổi chi phí liên kết

thay đổi chi phí liên kết:

nút phát hiện **thay đổi chi phí liên kết cục bộ** “tin

xấu lan truyền chậm” - vấn đề **đếm đến vô cùng** : y thấy liên kết trực tiếp tới x

- có chi phí mới là 60, như ng z đã nói rằng nó có một đường dẫn với chi phí là 5. Vì vậy, y tính “chi phí mới của tôi cho x sẽ là 6, thông qua z); thông báo cho z về chi phí mới từ 6 đến x. • z biết rằng đường dẫn tới x qua y có chi phí mới là 6, vì vậy z tính toán “chi phí mới của tôi tới x sẽ là 7 qua y), thông báo cho y về chi phí mới từ 7 đến x.
- y biết rằng đường dẫn đến x qua z có chi phí mới là 7, vì vậy y tính toán “chi phí mới của tôi tới x sẽ là 8 qua y), thông báo cho z về chi phí mới từ 8 đến x.
- z biết rằng đường dẫn tới x qua y có chi phí mới là 8, vì vậy z tính toán “chi phí mới của tôi tới x sẽ là 9 qua y), thông báo cho y về chi phí mới từ 9 đến x.
-



xem văn bản để biết giải pháp. Các thuật toán phân tán rất phức tạp!

So sánh thuật toán LS và DV

độ phức tạp của thông điệp

LS: n bộ định tuyến, $O(n^2)$ tin nhắn đã gửi

DV: trao đổi hàng xóm với nhau; thời gian hội tụ thay đổi

tốc độ hội tụ

LS: $O(n^2)$, thuật toán $O(n^2)$ tin nhắn

- có thể có **dao động**

DV: thời gian hội tụ khác nhau

- có thể có **các vòng lặp định tuyến** • ván đè đêm đến vô cùng

mạnh mẽ: điều gì xảy ra nếu bộ định tuyến gặp **trục trặc** hoặc **bị xâm phạm**?

LS:

- bộ định tuyến có thể quảng cáo **chi phí liên kết không chính xác**
- mỗi bộ định tuyến **chỉ tính riêng** của nó **cái bàn**

DV:

- Bộ định tuyến DV có thể quảng cáo **chi phí đường dẫn không chính xác** ("Tôi có một đường dẫn chi phí thực sự thấp đến mọi nơi"): blackholing
- mỗi bảng của bộ định tuyến được sử dụng bởi những người khác: **lỗi lan truyền qua mạng**

Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN Thông

báo Điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Làm cho định tuyến có thể mở rộng

nghiên cứu định tuyến của chúng tôi cho đến nay - đưa ra lý thuyết

hóa tất cả các bộ định tuyến giống nhau mạng "phẳng"

- . không đúng trong thực tế

quy mô: hàng tỷ điểm đến:

không thể lưu trữ tất cả các điểm đến trong

bảng định tuyến!

trao đổi bảng định tuyến sẽ tràn

ngập các liên kết!

tự chủ hành chính:

Internet: mạng của các mạng

mỗi quản trị viên mạng có thể muốn kiểm soát định tuyến trong mạng riêng của mình

Cách tiếp cận Internet để định tuyến có thể mở rộng

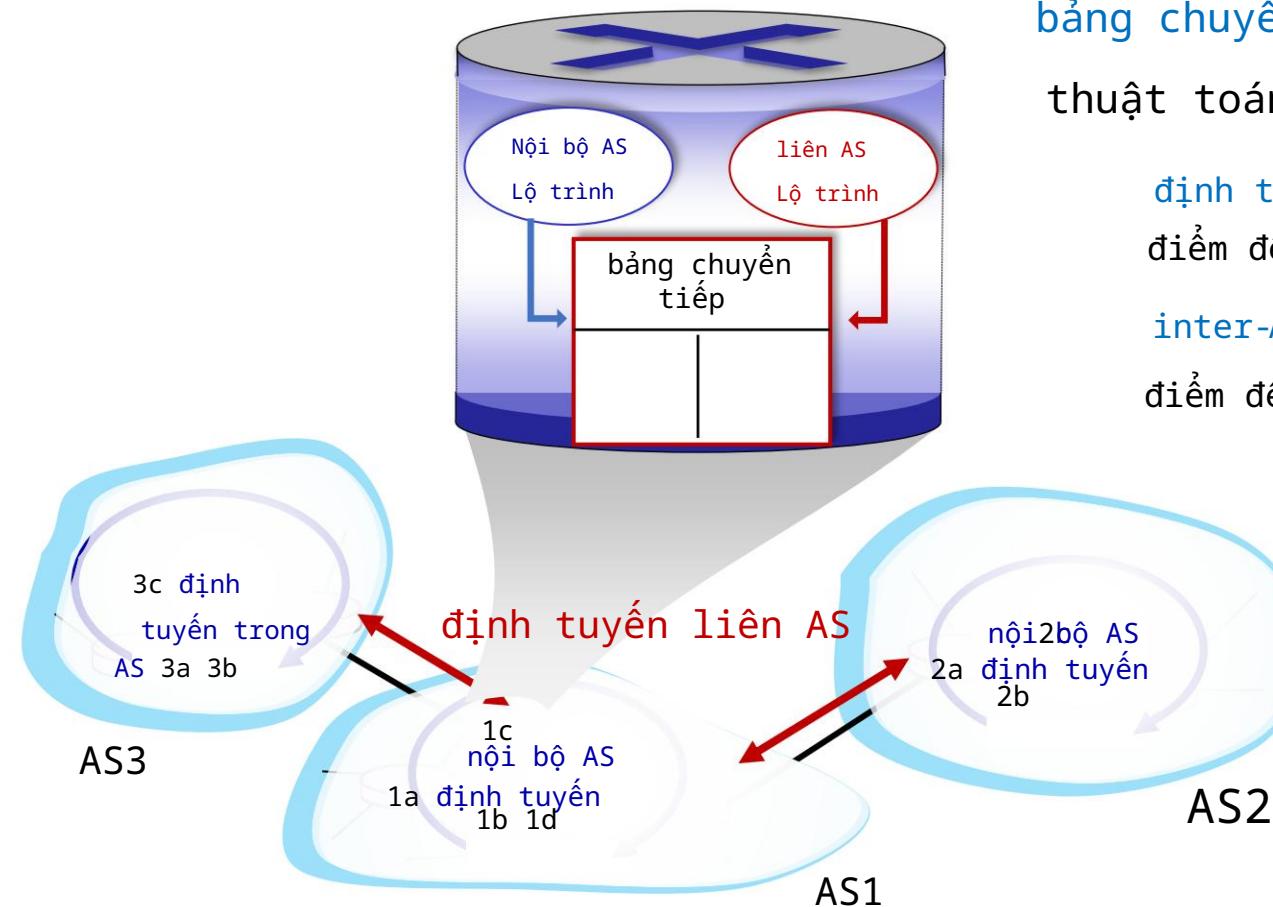
tổng hợp các bộ định tuyến thành **các vùng** đư ợc gọi là “hệ thống tự trị” (AS) (còn gọi là “miền”)

intra-AS (còn gọi là “nội miền”): định tuyến giữa các bên trong cùng một AS (“mạng”) tất cả các bộ định tuyến trong AS phải chạy **cùng một giao thức nội miền**

- các bộ định tuyến trong các AS khác nhau có thể chạy các giao thức định tuyến nội miền khác nhau
- bộ định tuyến **cổng**: ở “biên” của AS riêng, có (các) liên kết đến (các) bộ định tuyến trong các AS khác

liên AS (hay còn gọi là “liên miền”): định tuyến giữa các AS'es **cổng** thực hiện định tuyến liên miền (cũ ng như định tuyến nội miền)

Các AS đư ợc kết nối với nhau



bảng chuyển tiếp đư ợc định cấu hình bởi các thuật toán định tuyến nội bộ và liên AS

định tuyến trong AS xác định các mục nhập cho điểm đến trong AS

inter-AS & intra-AS xác định mục nhập cho các điểm đến bên ngoài

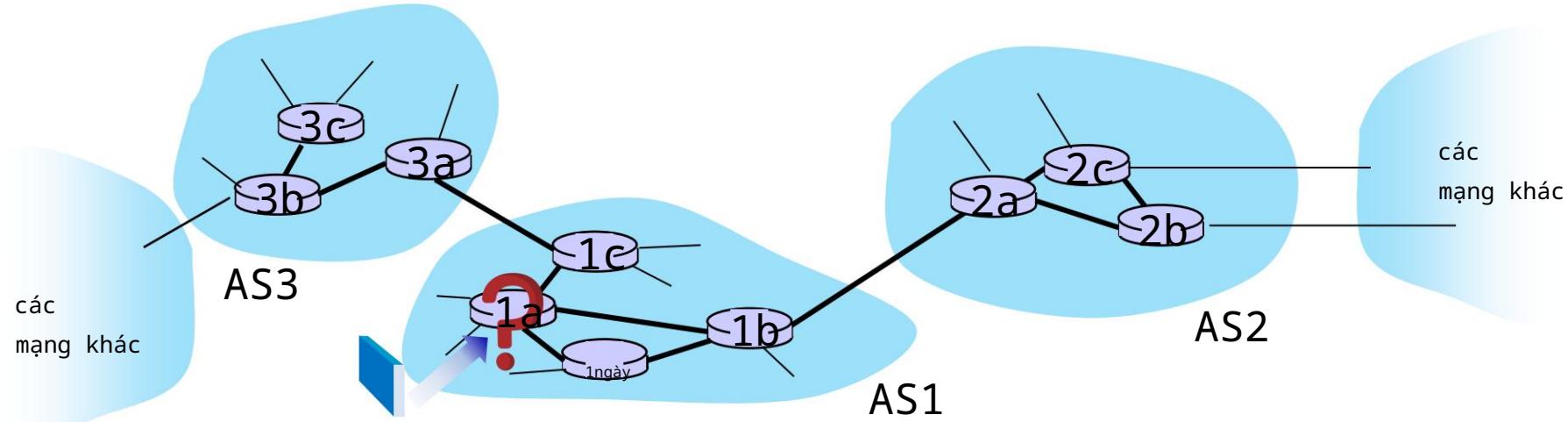
Định tuyến giữa các AS: vai trò trong chuyển tiếp nội miền

ví dụ: giả sử bộ định tuyến trong **AS1** nhận được gói dữ liệu được định sẵn bên ngoài **AS1**: • bộ

? định tuyến nên chuyển gói tới **bộ định tuyến cổng** trong **AS1**, như ng bộ định tuyến nào?

Định tuyến liên miền AS1 phải:

1. tìm hiểu đích đến nào có thể đến được thông qua **AS2**, đích đến nào thông qua **AS3**
2. truyền thông tin về khả năng đến được này đến tất cả các bộ định tuyến trong **AS1**



Định tuyến giữa các AS: định tuyến trong một AS

các giao thức định tuyến nội bộ AS phổ biến nhất : RIP: Giao

thức thông tin định tuyến [RFC 1723] • DV cổ điển : DV đư ợc

trao đổi cứ sau 30 giây • không còn đư ợc sử dụng rộng rãi

EIGRP: Giao thức định tuyến cổng nội bộ nâng cao

- Dựa trên DV
- trư ớc đây là công ty độc quyền của Cisco trong nhiều thập kỷ (đư ợc mở vào năm 2013 [RFC 7868])

OSPF: Mở đư ờng đi ngắn nhất trư ớc [RFC 2328] • định

tuyến trạng thái liên kết • Giao thức IS-IS (tiêu chuẩn

ISO, không phải tiêu chuẩn RFC) về cơ bản giống như OSPF

Định tuyến OSPF (Mở đư ờng dẫn ngắn nhất trư ớc)

“mở”: có sẵn công khai trạng

thái liên kết cổ điển • mỗi bộ định

tuyến tràn ngập quảng cáo trạng thái liên kết OSPF (trực tiếp qua IP thay vì sử dụng TCP/UDP) tới tất cả các bộ định tuyến khác trong toàn bộ AS

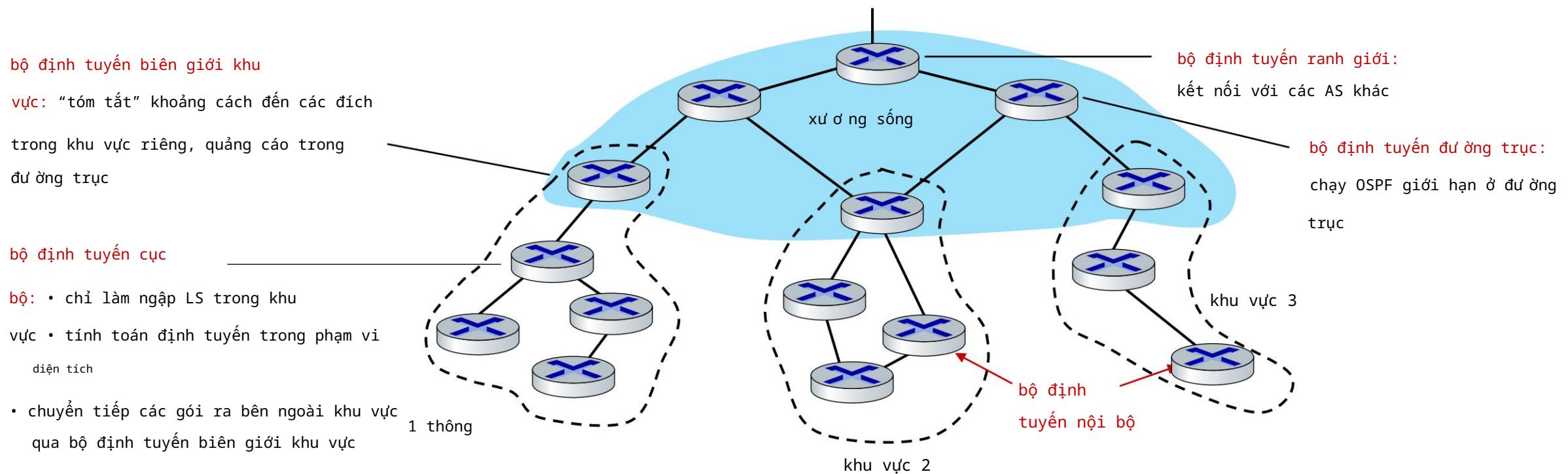
• có thể đo lư ờng nhiều chi phí liên kết : băng thông, độ trễ •

mỗi bộ định tuyến có cấu trúc liên kết đầy đủ, sử dụng thuật toán Dijkstra để tính toán bảng chuyển tiếp bảo mật: tất cả các thông báo OSPF đư ợc xác thực (để ngăn chặn xâm nhập độc hại)

OSPF phân cấp

phân cấp hai cấp: khu vực cục bộ, đư ờng

- quảng cáo trạng thái liên kết chỉ tràn ngập trong khu vực hoặc đư ờng trực
- mỗi nút có cấu trúc liên kết khu vực chi tiết; chỉ biết hứ ớng đến điểm đến khác



Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN Thông

báo Điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Định tuyến giữa các AS Internet: BGP

BGP (Border Gateway Protocol): định tuyến liên miền trên thực tế giao thức

- “chất keo dính kết Internet”

cho phép mạng con quảng cáo sự tồn tại của nó và các đích mà nó có thể tiếp cận, đến phần còn lại của Internet: “Tôi ở đây, đây là người tôi có thể tiếp cận và bằng cách nào”

BGP cung cấp cho mỗi AS một phư ơng tiện

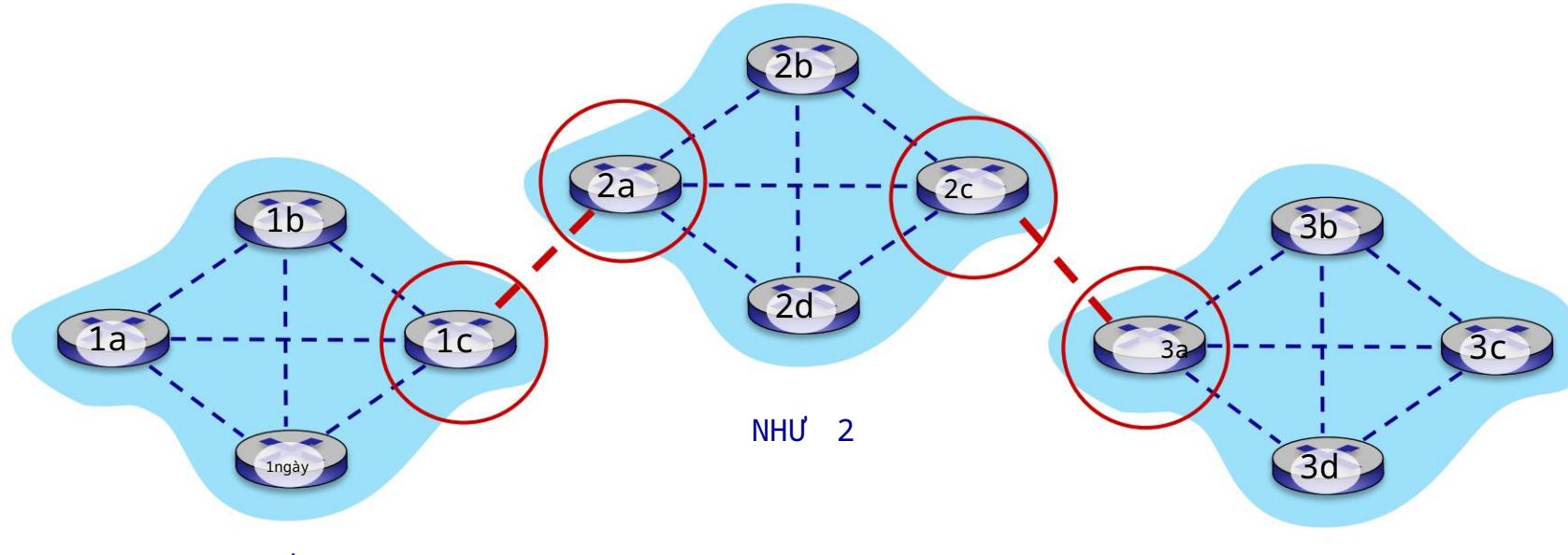
để:

- eBGP: lấy thông tin về khả năng tiếp cận mạng con từ các AS lân cận •

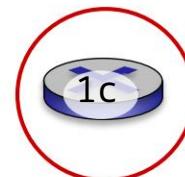
iBGP: truyền thông tin về khả năng tiếp cận tới tất cả các bộ định tuyến bên trong AS.

- xác định các tuyến “tốt” đến các mạng khác dựa trên thông tin về khả năng tiếp cận và chính sách

Kết nối eBGP, iBGP



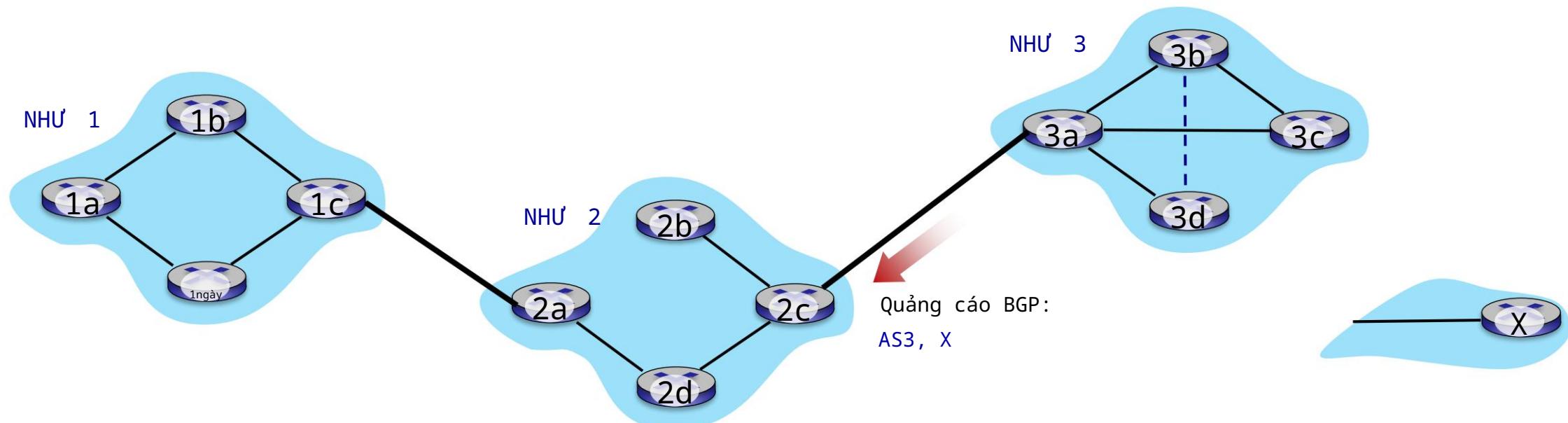
— — — kết nối eBGP logic
- - - - - kết nối iBGP



bộ định tuyến cổng chạy cả giao thức eBGP và iBGP

Khái niệm cơ bản về BGP

Phiên BGP: hai bộ định tuyến BGP ("ngang hàng") trao đổi thông báo BGP qua kết nối TCP bán cố định: • quảng cáo đường dẫn đến các tiền tố mạng đích khác nhau (BGP là giao thức "véc -tơ đường dẫn") , ví dụ: khi cổng AS3 3a quảng cáo đường dẫn AS3, Cổng X tới AS2 2c: • AS3 hứa với AS2 rằng nó sẽ chuyển tiếp các datagram tới X



Thuộc tính đường dẫn và tuyến đường BGP

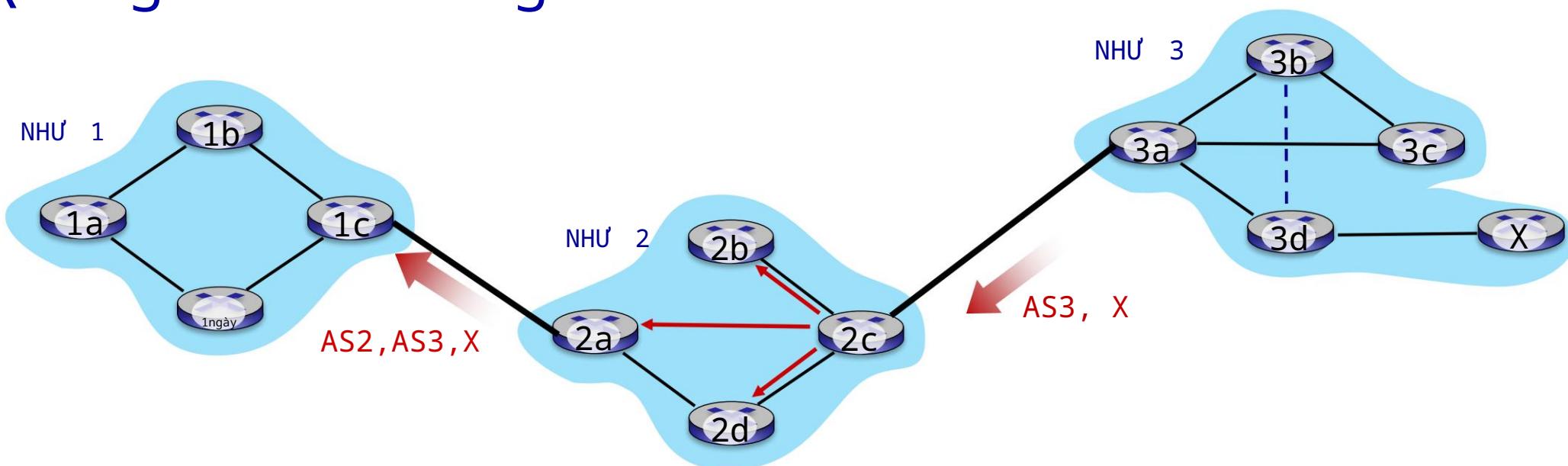
Tuyến đường được quảng cáo BGP = tiền tố +
thuộc tính • tiền tố: đích được quảng cáo • hai
thuộc tính quan trọng:

- AS-PATH: danh sách các AS mà quảng cáo tiền tố đã đi qua • NEXT-HOP:
chỉ ra bộ định tuyến AS bên trong cụ thể tới AS next-hop

định tuyến dựa trên chính

sách: • Quảng cáo tuyến nhận **cổng** sử dụng **chính sách nhập** để chấp nhận/
từ chối **đường** (ví dụ: không bao giờ định tuyến qua AS Y). • **Chính**
sách AS cũng xác định xem có **quảng bá** đường dẫn đến các AS lân cận
khác hay không

Quảng cáo đư ờng dẫn BGP

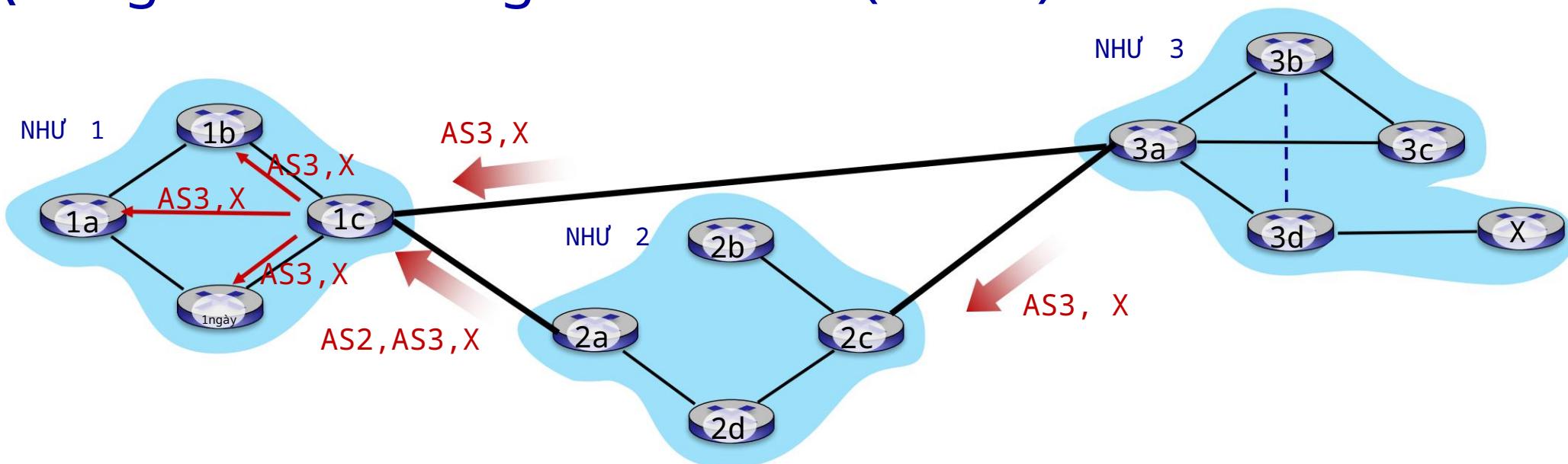


Bộ định tuyến AS2 2c nhận quảng cáo đư ờng dẫn AS3,X (through eBGP) từ bộ định tuyến AS3 3a

dựa trên chính sách AS2, bộ định tuyến AS2 2c chấp nhận đư ờng dẫn AS3,X, truyền (through iBGP) tới tất cả bộ định tuyến AS2

dựa trên chính sách AS2, bộ định tuyến AS2 2a quảng cáo (through eBGP) đư ờng dẫn AS2, AS3, X tới bộ định tuyến AS1 1c

Quảng cáo đư ờng dẫn BGP (thêm)



bộ định tuyến cổng có thể tìm hiểu về nhiều đư ờng dẫn đến đích:

Bộ định tuyến cổng AS1 1c học đư ờng AS2, AS3, X từ 2a

Bộ định tuyến cổng AS1 1c học đư ờng AS3, X từ 3a

dựa trên chính sách, bộ định tuyến cổng AS1 1c chọn đư ờng AS3, X và quảng bá đư ờng trong AS1 thông qua iBGP

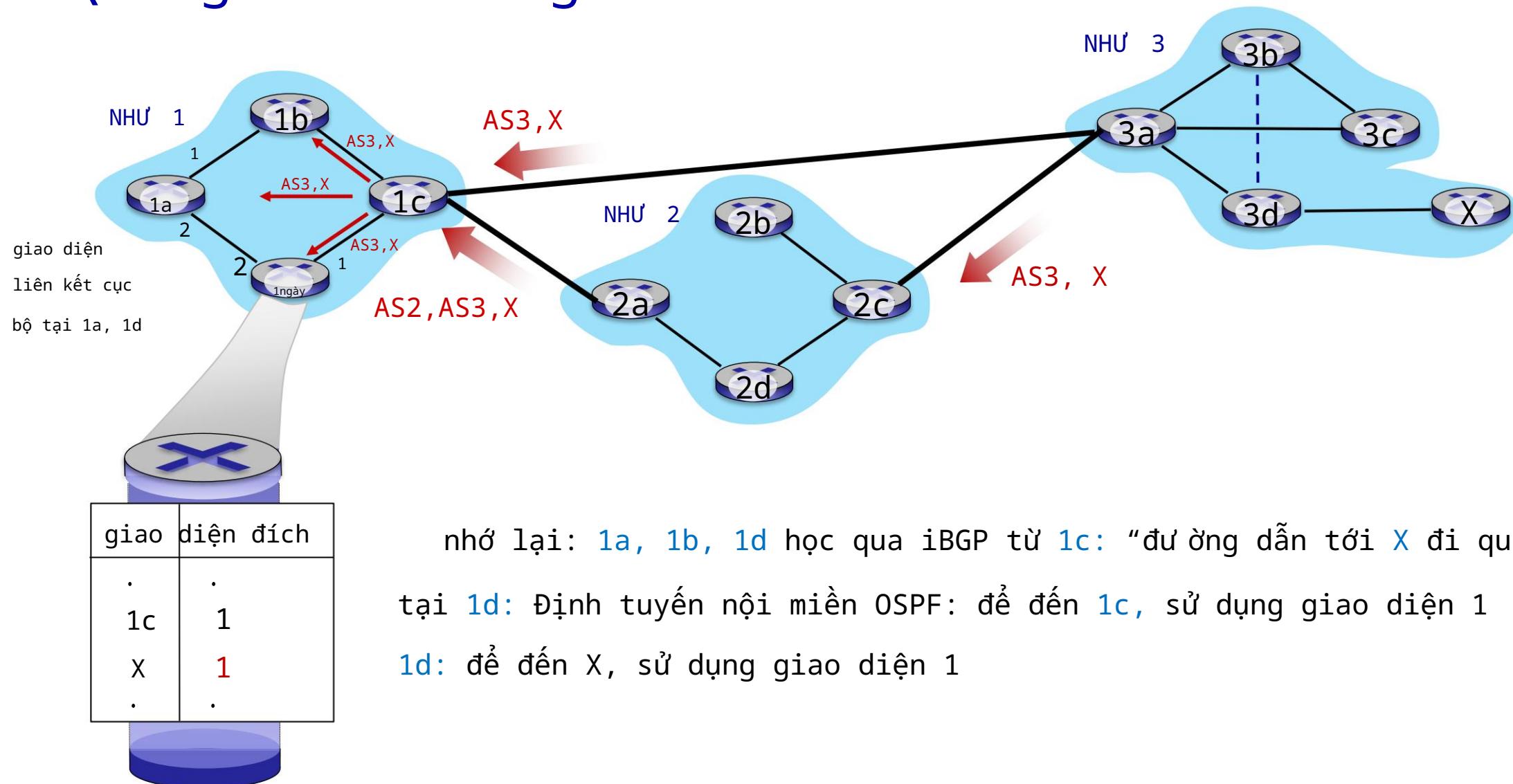
tin nhắn BGP

Các bản tin BGP được trao đổi giữa các đồng nghiệp qua kết nối TCP

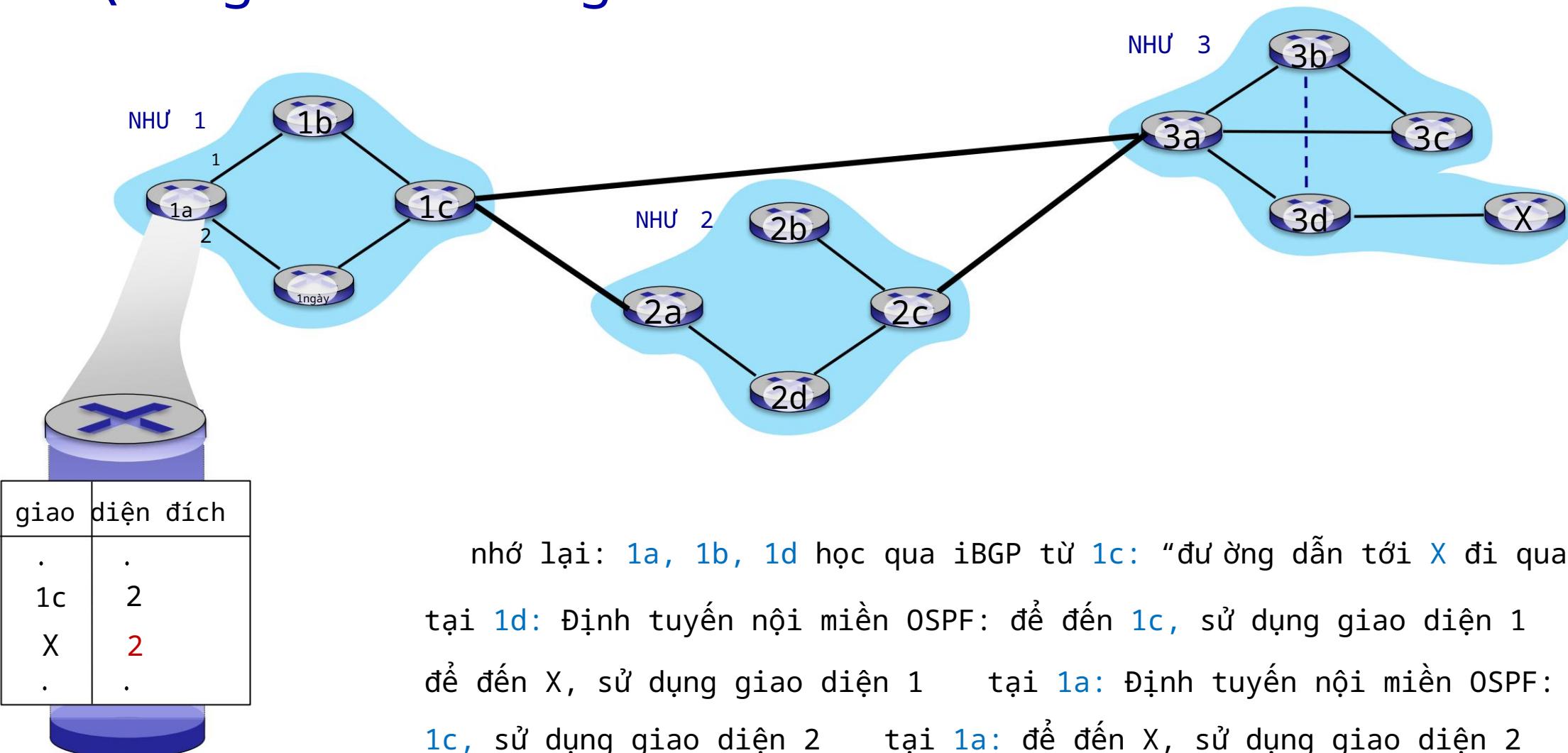
Các bản tin BGP:

- OPEN: mở kết nối TCP tới BGP ngang hàng từ xa và xác thực gửi BGP ngang hàng
- CẬP NHẬT: quảng cáo đường dẫn mới (hoặc rút lại đường dẫn cũ)
- KEEPALIVE: duy trì kết nối khi không có CẬP NHẬT; Ngoài ra, ACK Mở yêu cầu
- NOTIFICATION: thông báo lỗi trong tin nhắn trước đó; Ngoài ra, được sử dụng để đóng sự liên quan

Quảng cáo đư ờng dẫn BGP



Quảng cáo đư ờng dẫn BGP



Tại sao định tuyến Intra-, Inter-AS khác nhau?

chính

sách: **liên AS**: quản trị viên muốn kiểm soát cách lưu luợng được định tuyến, ai định tuyến qua mạng của nó

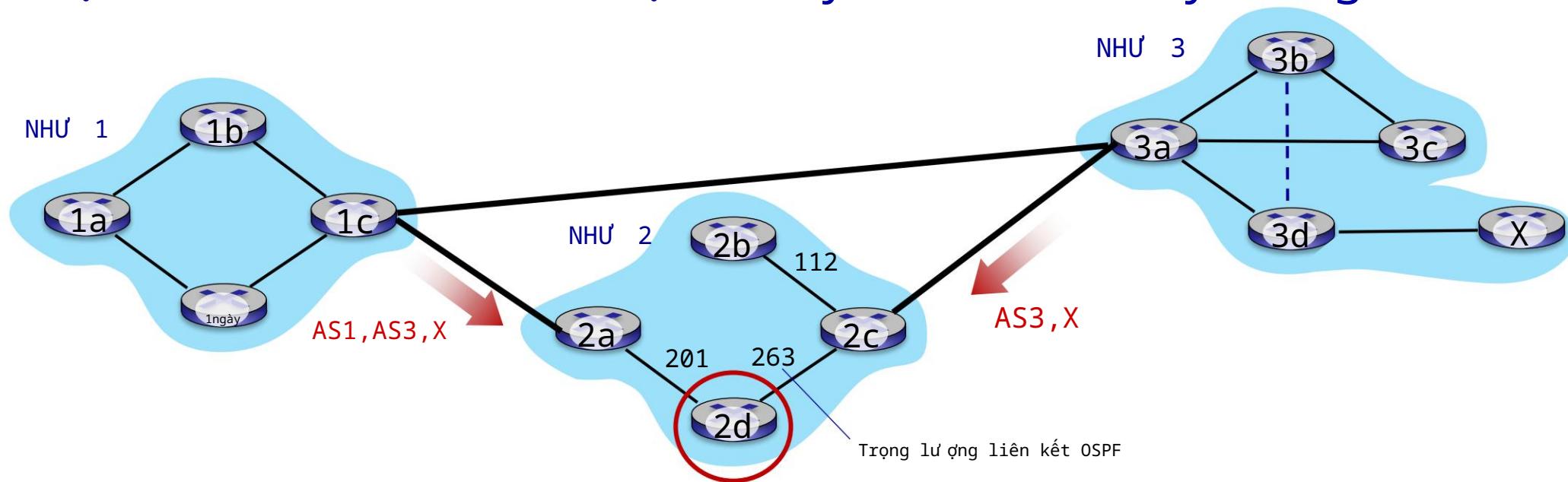
intra-AS: quản trị viên duy nhất, vì vậy chính sách ít gặp vấn đề hơn

tỉ lệ:

định tuyến phân cấp giúp tiết kiệm kích thư ớc bảng, giảm **hiệu suất**

lưu luợng cập nhật: **nội bộ AS**: có thể tập trung vào **hiệu suất liên AS**: **chính sách** chi phối hiệu suất

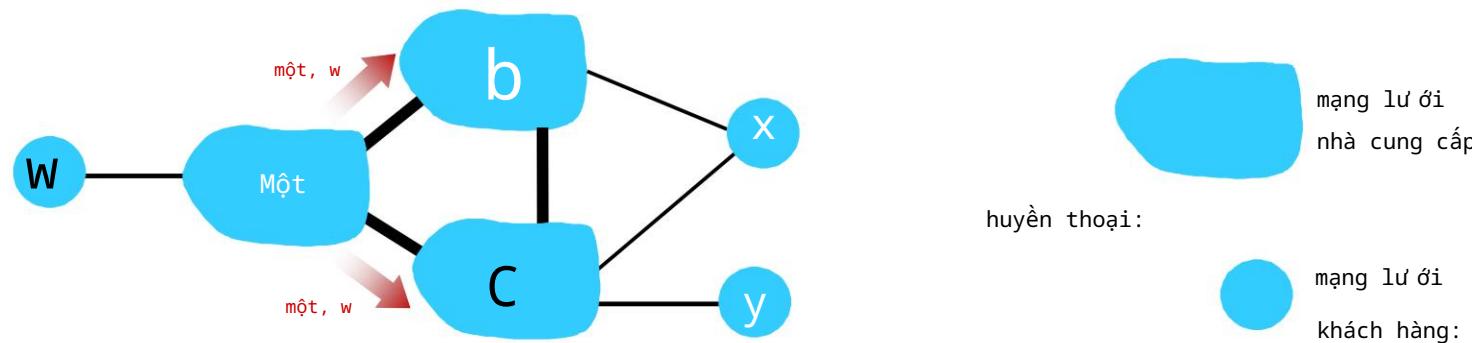
Thuộc tính NEXT-HOP: Định tuyến khoai tây nóng



2d học (thông qua iBGP) nó có thể định tuyến đến X thông qua 2a hoặc 2c

định tuyến khoai tây nóng: chọn cổng cục bộ có ít miền nội bộ nhất chi phí (ví dụ: 2d chọn 2a, mặc dù có nhiều AS nhảy tới X): đừng lo lắng về chi phí liên miền!

BGP: đạt được chính sách thông qua quảng cáo

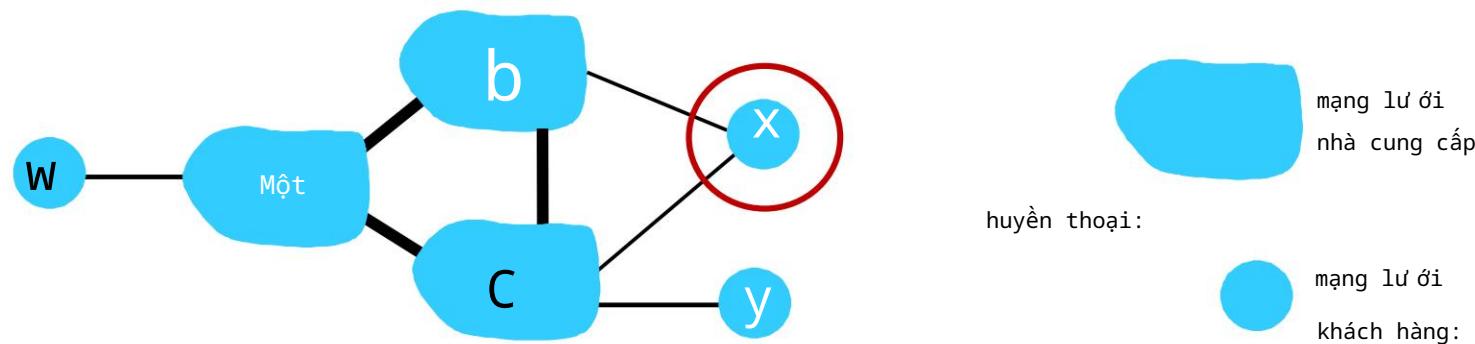


ISP chỉ muốn định tuyến lưu lượng đến/từ mạng khách hàng của mình (không muốn **chuyển lưu lượng** giữa các ISP khác - một chính sách “thế giới thực” điển hình)

- A quảng cáo đường dẫn Aw đến B và
- C B **chọn không quảng cáo BAw** đến C!

B không nhận được “doanh thu” cho việc định tuyến CBAw, vì không ai trong số C, A, w là khách hàng của B C không **tìm hiểu** về đường dẫn CBAw C sẽ định tuyến CAw (không sử dụng B) để đến w

BGP: đạt được chính sách thông qua quảng cáo (thêm)



ISP chỉ muốn định tuyến lưu lượng đến/từ mạng khách hàng của mình (không muốn **chuyển lưu lượng** giữa các ISP khác - một chính sách “thế giới thực” điển hình)

A,B,C là **các mạng** của nhà cung

cấp x,w,y là **khách hàng** (của các mạng của nhà
cung cấp) x là **dual-homed**: **đư ợc gắn vào hai mạng**

chính sách để thực thi: x không muốn định tuyến từ B đến C qua
x . .vì vậy, x sẽ không quảng cáo tới B một lô trình tới C

Lựa chọn tuyến đư ờng BGP

bộ định tuyến có thể tìm hiểu về **nhiều tuyến đư ờng** đến AS đích, chọn tuyến đư ờng dựa trên: 1. thuộc tính giá trị **ưu tiên cục bộ** : quyết định chính sách 2. **AS-PATH ngắn nhất** 3. bộ định tuyến **NEXT-HOP gần nhất** : định tuyến củ khoai tây nóng 4. **tiêu chí bổ sung**

Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo Điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Mạng được xác định bằng phần mềm (SDN)

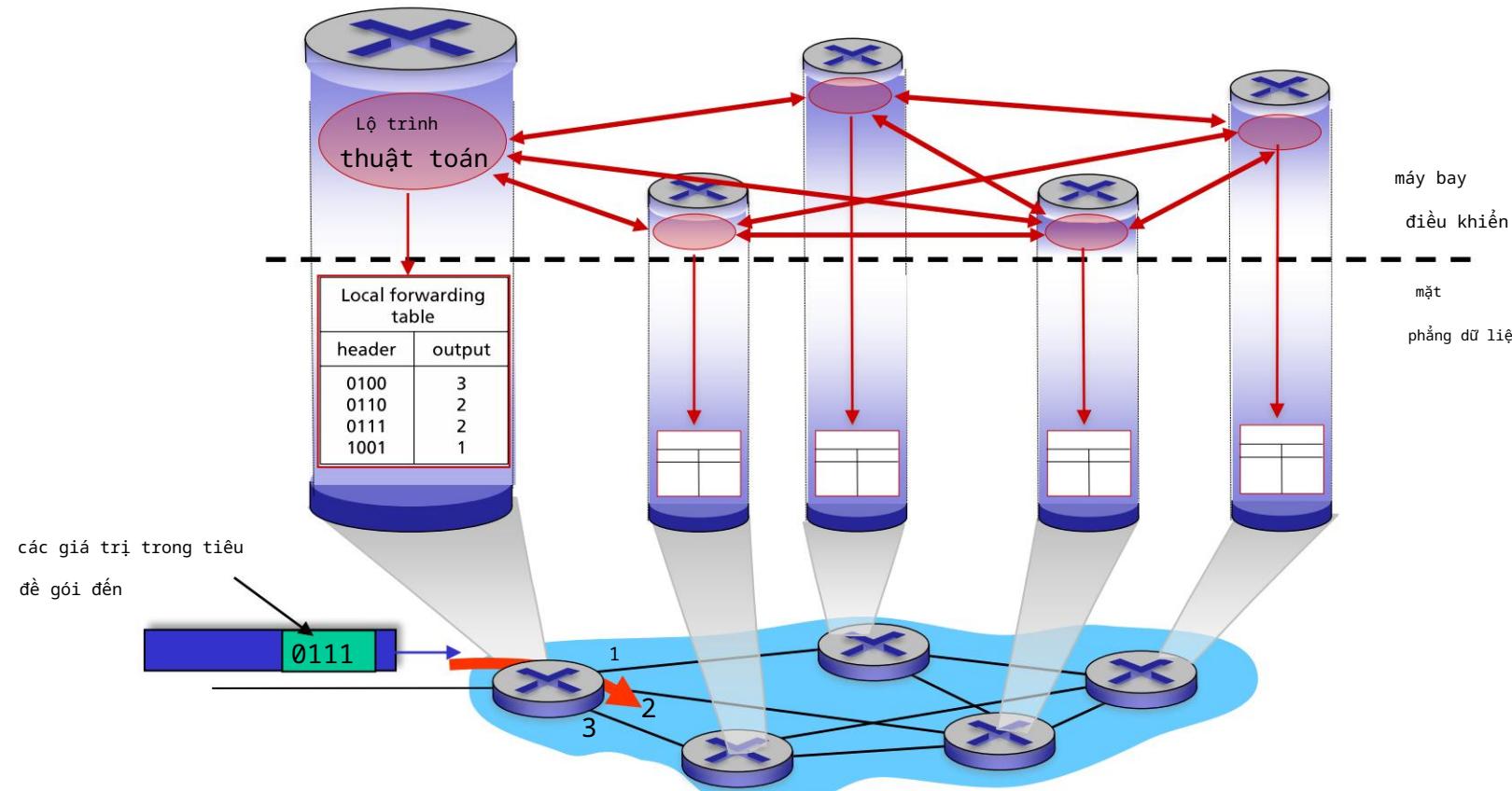
Lớp mạng Internet: trong lịch sử, được triển khai thông qua
phương pháp **điều khiển phân tán**, cho mỗi bộ định

tuyến: • bộ định tuyến nguyên khối chứa **phần cứng chuyển mạch**, chạy triển
khai độc quyền các giao thức chuẩn Internet (IP, RIP, IS-IS, OSPF, BGP)
trong **hệ điều hành bộ định tuyến độc quyền** (ví dụ: Cisco IOS) • các “hộp
trung gian” khác nhau cho các chức năng lớp mạng khác nhau: tư ờng lửa, cân
bằng tải, hộp NAT, ..

~2005: lại quan tâm đến việc xem xét lại mặt phẳng điều khiển mạng

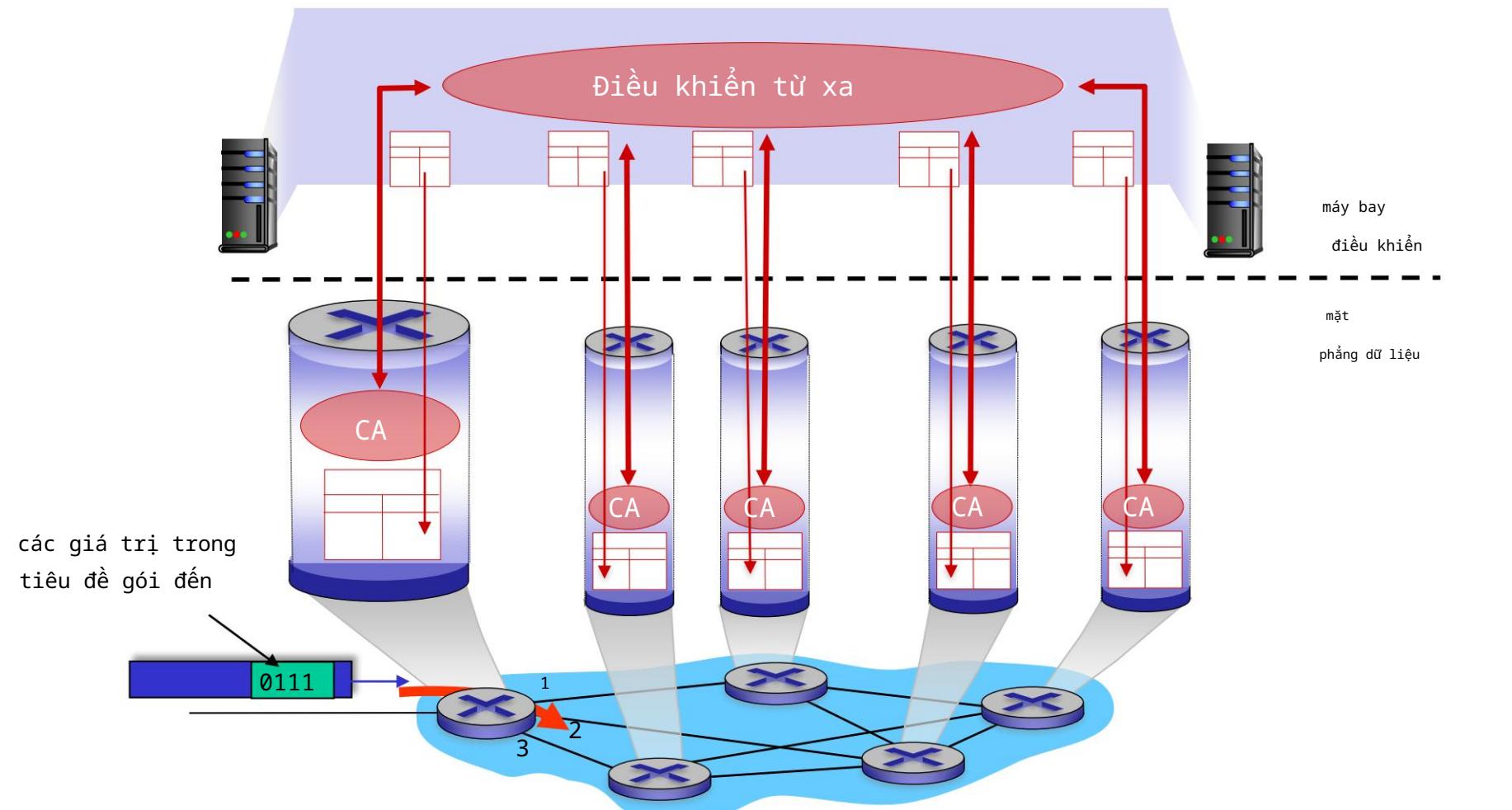
Mặt phẳng điều khiển mỗi bộ định tuyến

Các thành phần **thuật toán định tuyến riêng lẻ** trong mỗi và mọi bộ định tuyến tương tác trong mặt phẳng điều khiển để tính toán các bảng chuyển tiếp



Mặt phẳng điều khiển Mạng được xác định bằng phần mềm (SDN)

Bộ điều khiển từ xa tính toán, cài đặt bảng chuyển tiếp trong bộ định tuyến



Mạng được xác định bằng phần mềm (SDN)

Tại sao một mặt phẳng điều khiển tập trung hợp lý ?

quản lý mạng dễ dàng hơn: tránh cấu hình sai bộ định tuyến, linh hoạt hơn của luồng giao thông

chuyển tiếp dựa trên bảng (nhờ lại API OpenFlow) cho phép các bộ định tuyến “lập trình” • “lập trình” tập trung dễ dàng

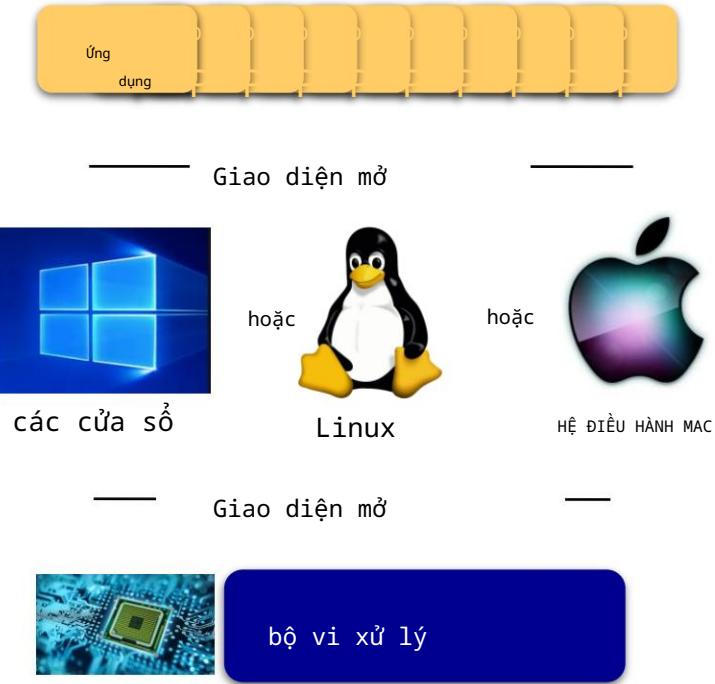
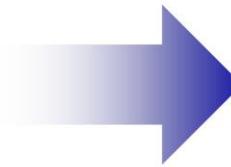
hơn: tính toán các bảng tập trung và phân phối • “ lập trình” phân tán khó khăn hơn: tính toán các bảng do thuật toán (giao thức) phân tán đư ợc triển khai trong mỗi và -mọi bộ định tuyến

triển khai mở (không độc quyền) mặt phẳng điều khiển • thúc đẩy đổi mới: để 1000 bông hoa cùng nở

Tư ơng tự SDN: cuộc cách mạng từ máy tính lớn đến PC

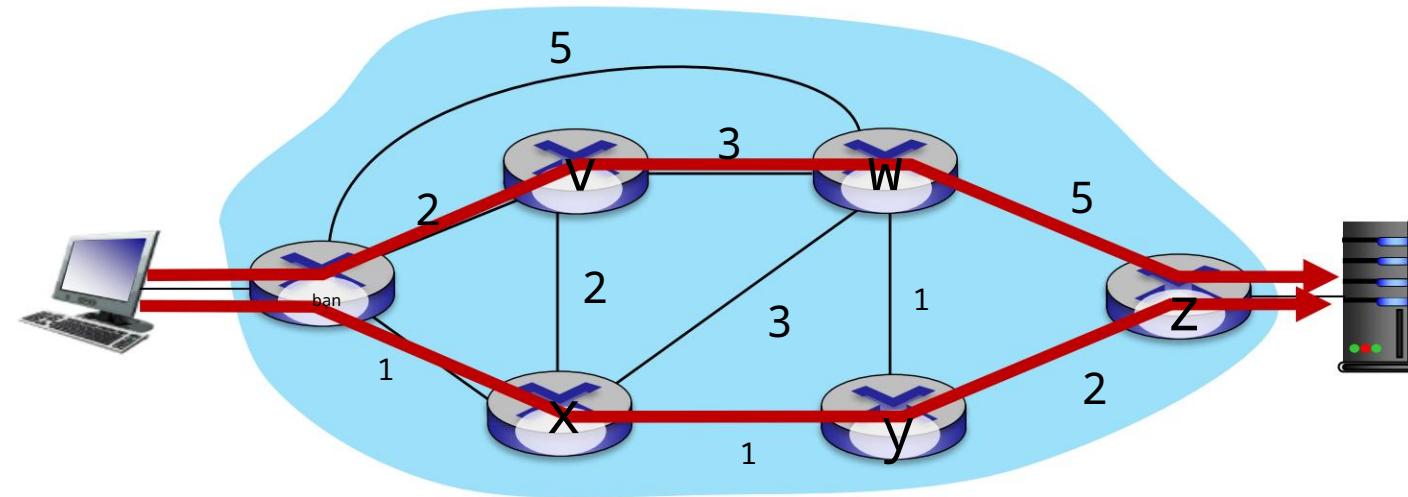


Tích hợp theo chiều dọc
Đóng cửa, độc quyền
Chậm đổi mới
công nghiệp nhỏ



Nằm ngang
mở giao diện
Đổi mới nhanh chóng
ngành công nghiệp khổng lồ

Kỹ thuật lưu u lưu ợng: khó khăn với định tuyến truyền thống

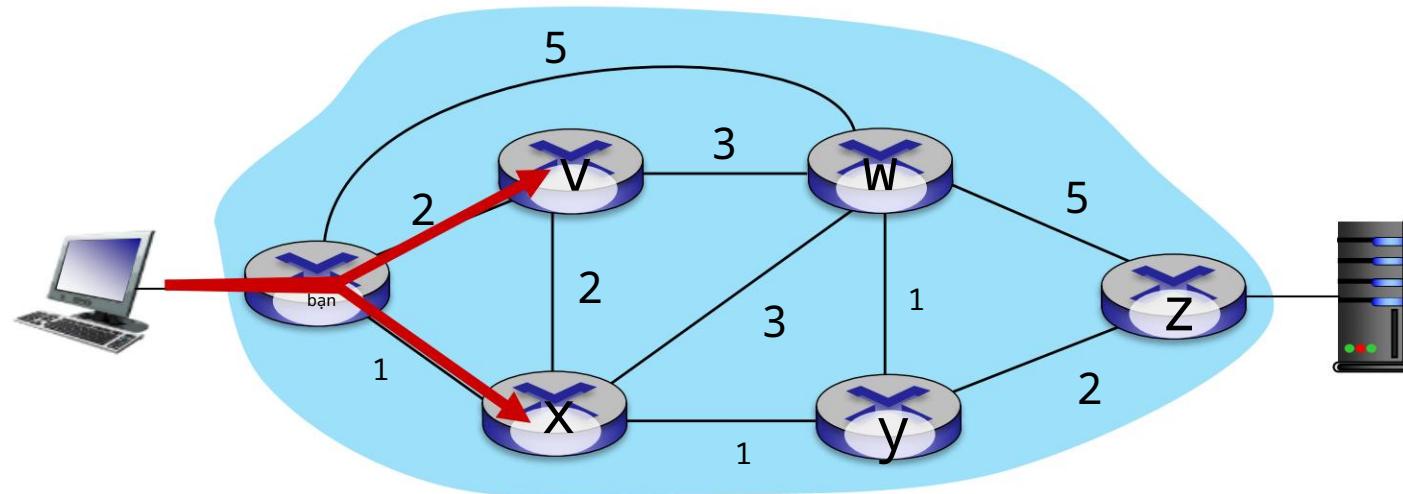


Câu hỏi: nếu nhà điều hành mạng muốn lưu u lưu ợng từ u đến z chảy dọc theo $uvwz$, thay vì $uxyz$ thì sao?

Trả lời: cần xác định lại **trọng số liên kết** để thuật toán định tuyến lưu u lưu ợng tính toán các tuyến tư ứng (hoặc cần một thuật toán định tuyến mới)!

trọng số liên kết chỉ là “nút” điều khiển: không kiểm soát nhiều!

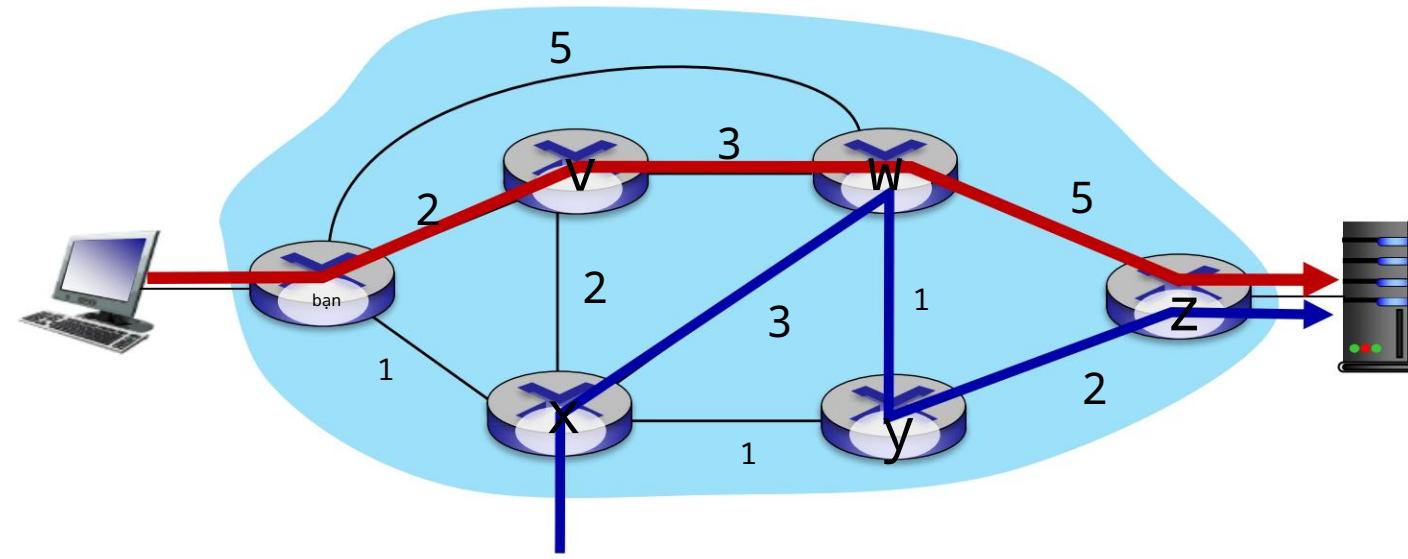
Kỹ thuật lưu u lưu ợng: khó khăn với định tuyến truyền thống



Câu hỏi: nếu nhà điều hành mạng muốn **phân chia** lưu u lưu ợng u-to-z dọc theo uvwz và uxyz (cân bằng tải) thì sao?

A: không làm đư ợc (hoặc cần thuật toán định tuyến mới)

Kỹ thuật lưu u lưu ợng: khó khăn với định tuyến truyền thống



H: nếu w muốn định tuyến lưu u lưu ợng màu xanh và đỏ khác nhau từ w đến z thì sao?

Trả lời: không thể thực hiện đư ợc (với chuyển tiếp dựa trên đích và định tuyến LS, DV)

Chúng ta đã học trong Chương 4 rằng **chuyển tiếp tổng quát** và **SDN** có thể đư ợc sử dụng để đạt đư ợc bất kỳ định tuyến mong muôn nào .

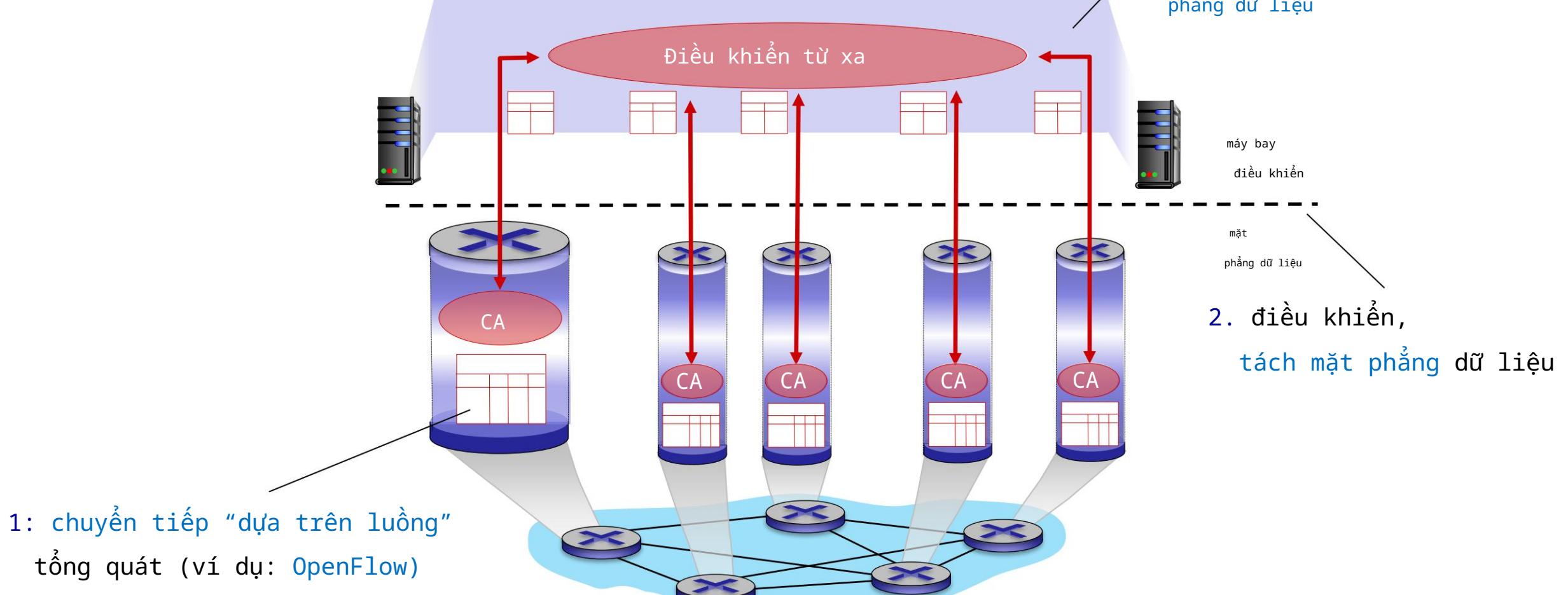
Mạng được xác định bằng phần mềm (SDN)

4. Ứng dụng điều khiển lập trình

định tuyến
quyền kiểm soát

cân bằng tải

3. các chức năng của mặt phẳng điều khiển bên ngoài các công tắc **mặt phẳng dữ liệu**



Mạng được xác định bằng phần mềm (SDN)

Chuyển mạch mặt phẳng dữ liệu:

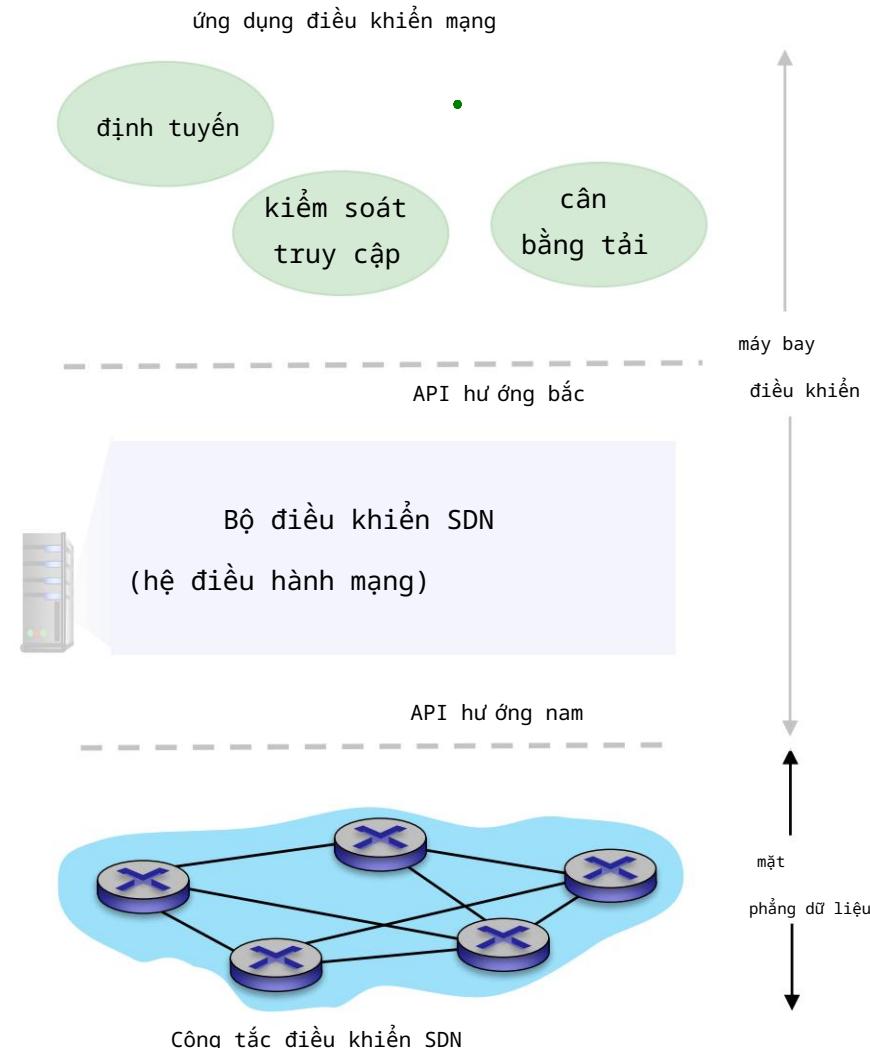
chuyển mạch hàng hóa nhanh, đơn giản

triển khai chuyển tiếp mặt phẳng dữ liệu tổng quát (Phần 4.4) trong phần cứng

bảng lưu trữ (chuyển tiếp) được tính toán, cài đặt dữ liệu sự giám sát của bộ điều khiển API cho điều khiển chuyển đổi dựa trên bảng

(ví dụ: OpenFlow) •

xác định cái gì có thể kiểm soát được, cái gì không giao thức giao tiếp với bộ điều khiển (ví dụ: OpenFlow)



Mạng được xác định bằng phần mềm (SDN)

Bộ điều khiển SDN (hệ điều hành mạng):

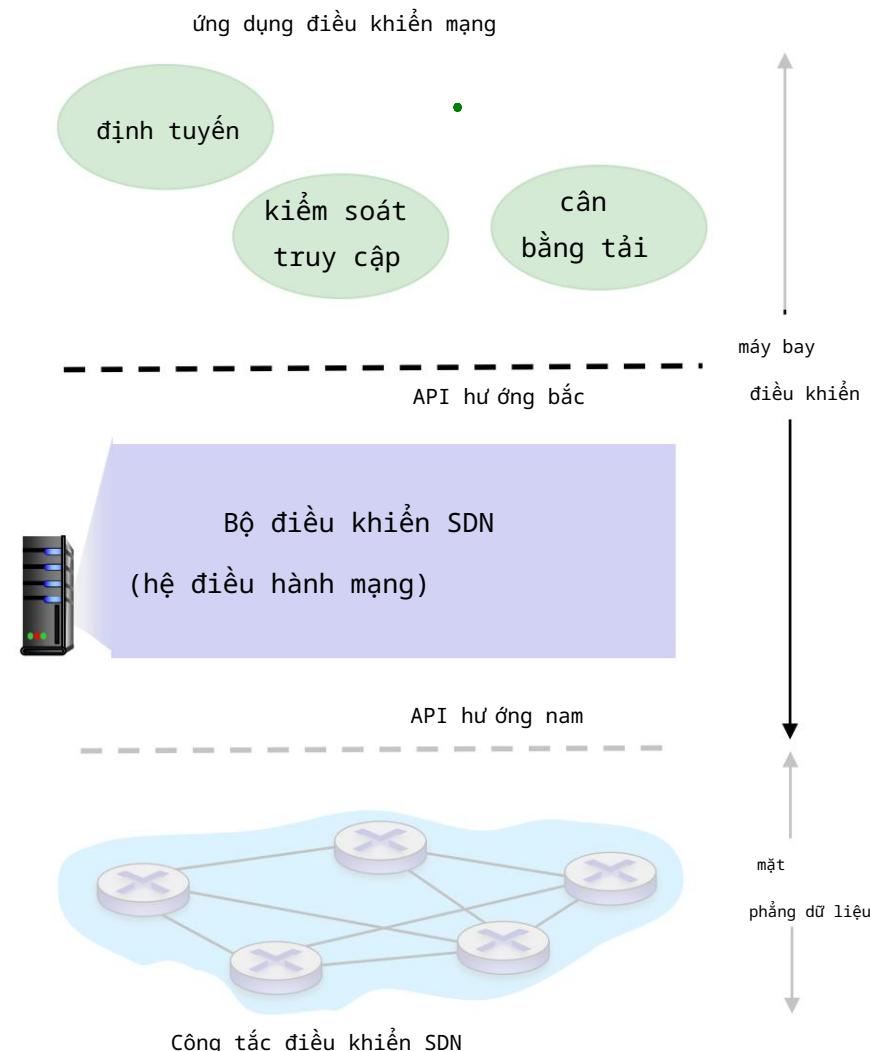
điều khiển trạng thái mạng

thông tin

tươn tác với các ứng dụng điều khiển mạng “ở trên” thông qua **API hướn bắc**

tươn tác với các bộ chuyển mạch mạng “bên dưới” thông qua **API hướn nam**

được triển khai dưới dạng **hệ thống phân tán** về hiệu suất, khả năng mở rộng, khả năng chịu lỗi, độ bền

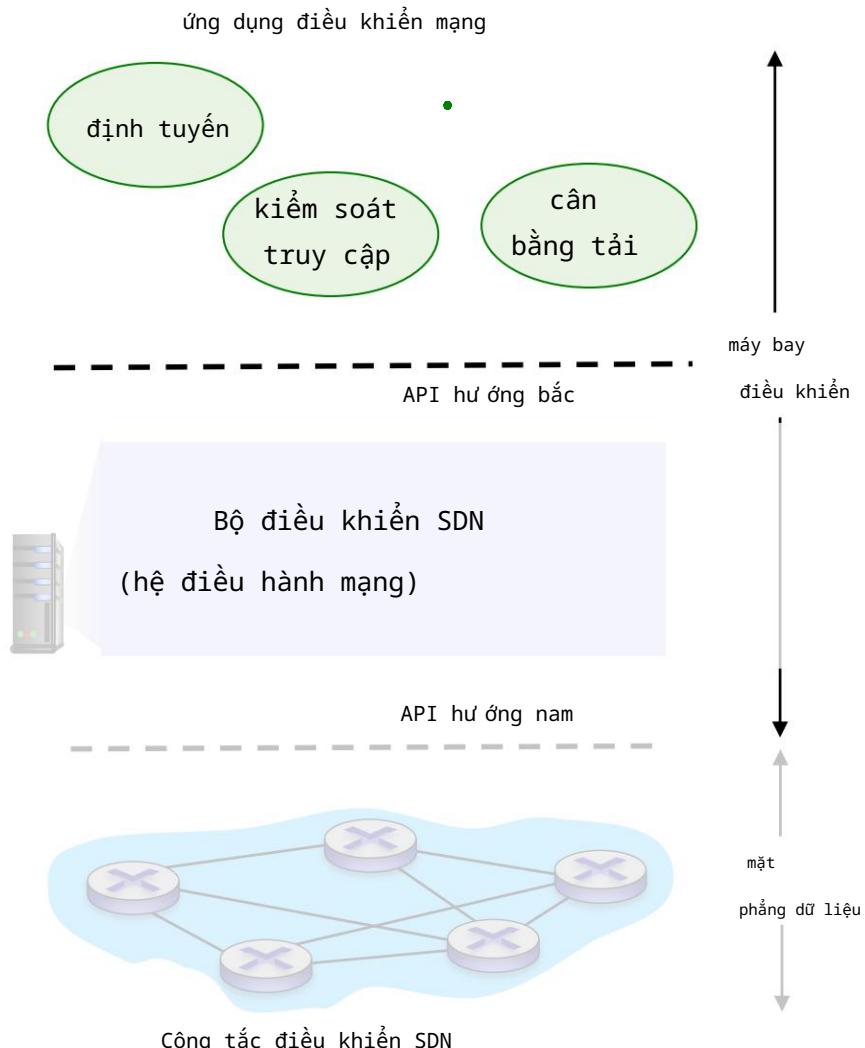


Mạng được xác định bằng phần mềm (SDN)

các ứng dụng điều khiển mạng:

“bộ não” điều khiển: thực hiện các chức năng điều khiển bằng cách sử dụng các dịch vụ cấp thấp hơn, API được cung cấp bởi bộ điều khiển SDN

không được nhóm: có thể được cung cấp bởi bên thứ 3: [khác với nhà cung cấp](#)
định tuyến: hoặc bộ điều khiển SDN

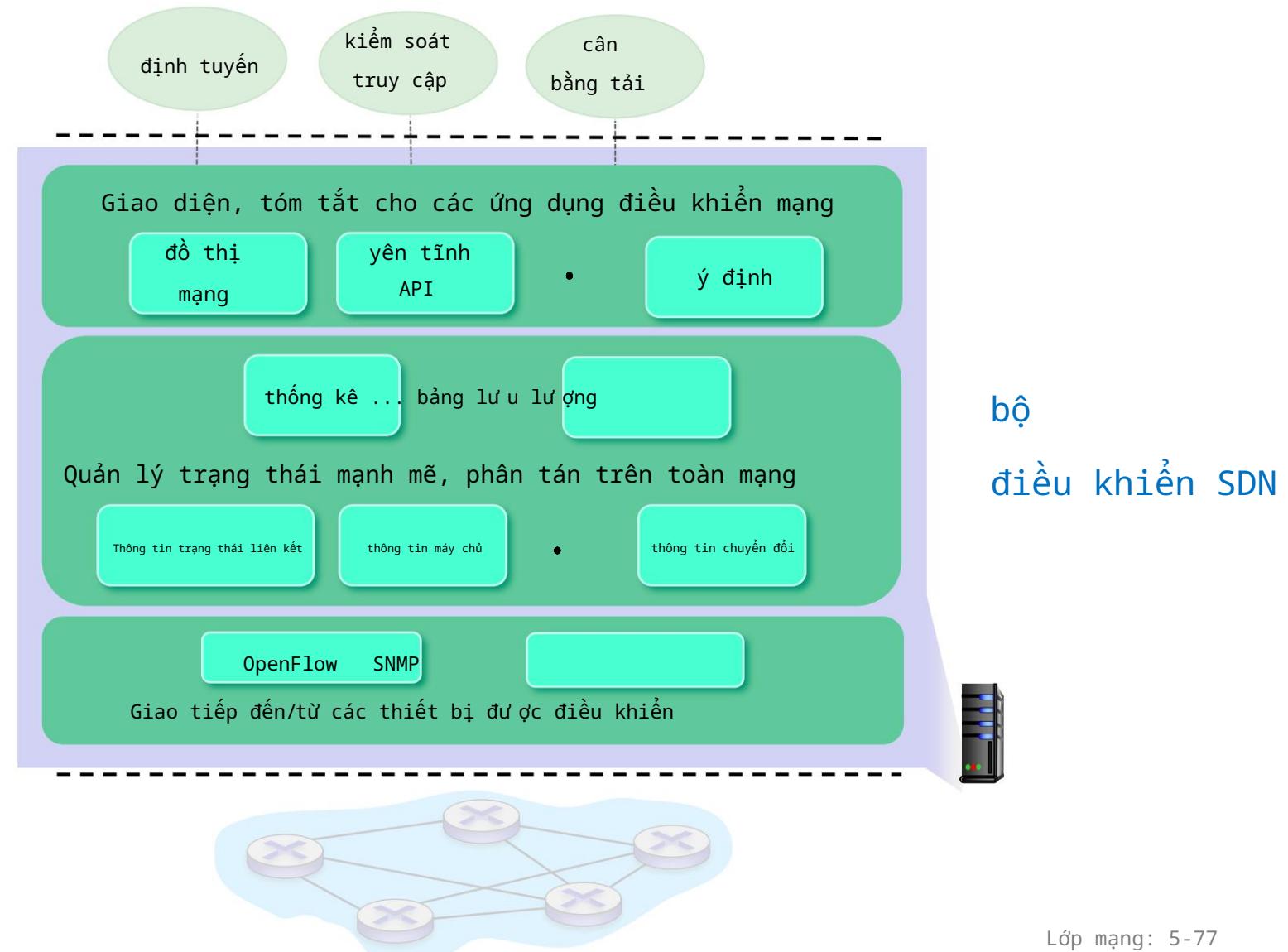


Các thành phần của bộ điều khiển SDN

**lớp giao diện cho các ứng dụng
điều khiển mạng:** API trừu tư ợng

**quản lý trạng thái toàn
mạng:** trạng thái của các
liên kết mạng, chuyển mạch,
dịch vụ: cơ sở dữ liệu phân tán

giao tiếp: giao tiếp giữa bộ điều khiển
SDN và các công tắc được điều khiển



Giao thức OpenFlow

hoạt động giữa bộ điều khiển, switch
TCP được sử dụng để trao đổi thông điệp

- mã hóa tùy chọn

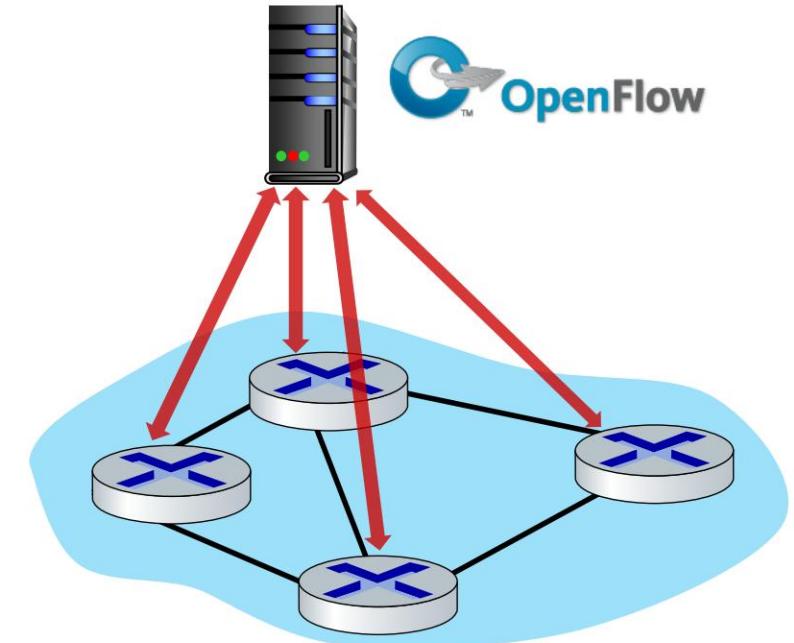
ba loại thông điệp OpenFlow:

- bộ điều khiển chuyển đổi
- không đồng bộ (chuyển sang bộ điều khiển)
- đối xứng (linh tinh)

khác với OpenFlow API

- API được sử dụng để chỉ định các hành động chuyển tiếp tổng quát

Bộ điều khiển OpenFlow

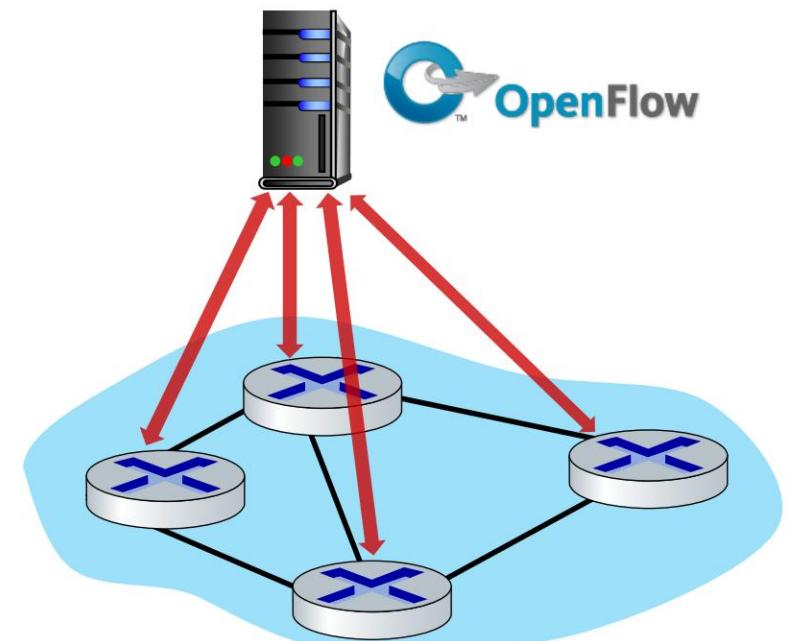


OpenFlow: thông báo từ bộ điều khiển đến chuyển đổi

Các thông báo chính từ bộ điều khiển đến bộ

chuyển mạch tính năng: truy vấn bộ điều
 khiển tính năng chuyển đổi, phản hồi chuyển
 đổi
định cấu hình: bộ điều khiển truy vấn/
 đặt tham số cấu hình chuyển đổi **sửa đổi**
trạng thái: thêm, xóa, sửa đổi **các mục nhập**
luồng trong bảng OpenFlow **gửi gói ra:** bộ
 điều khiển có thể gửi gói này ra khỏi cổng
 chuyển đổi cụ thể

Bộ điều khiển OpenFlow



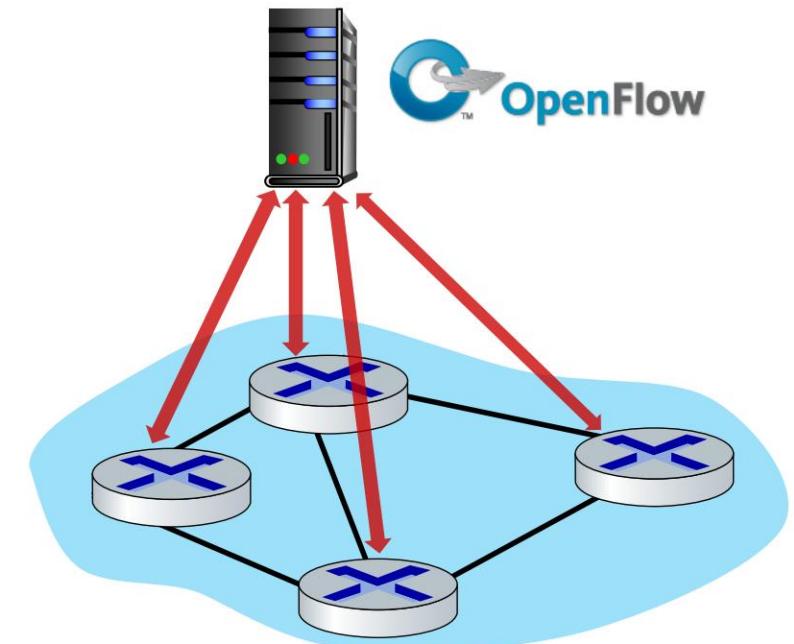
OpenFlow: thông báo chuyển sang bộ điều khiển

Các thông báo chuyển mạch chính sang bộ điều

khiển **packet-in**: truyền gói tin (và
điều khiển) đến bộ điều khiển. Xem thông
báo gói tin từ bộ điều khiển **loại bỏ**
luồng: mục nhập bảng lưu trữ bị xóa tại công
tắc

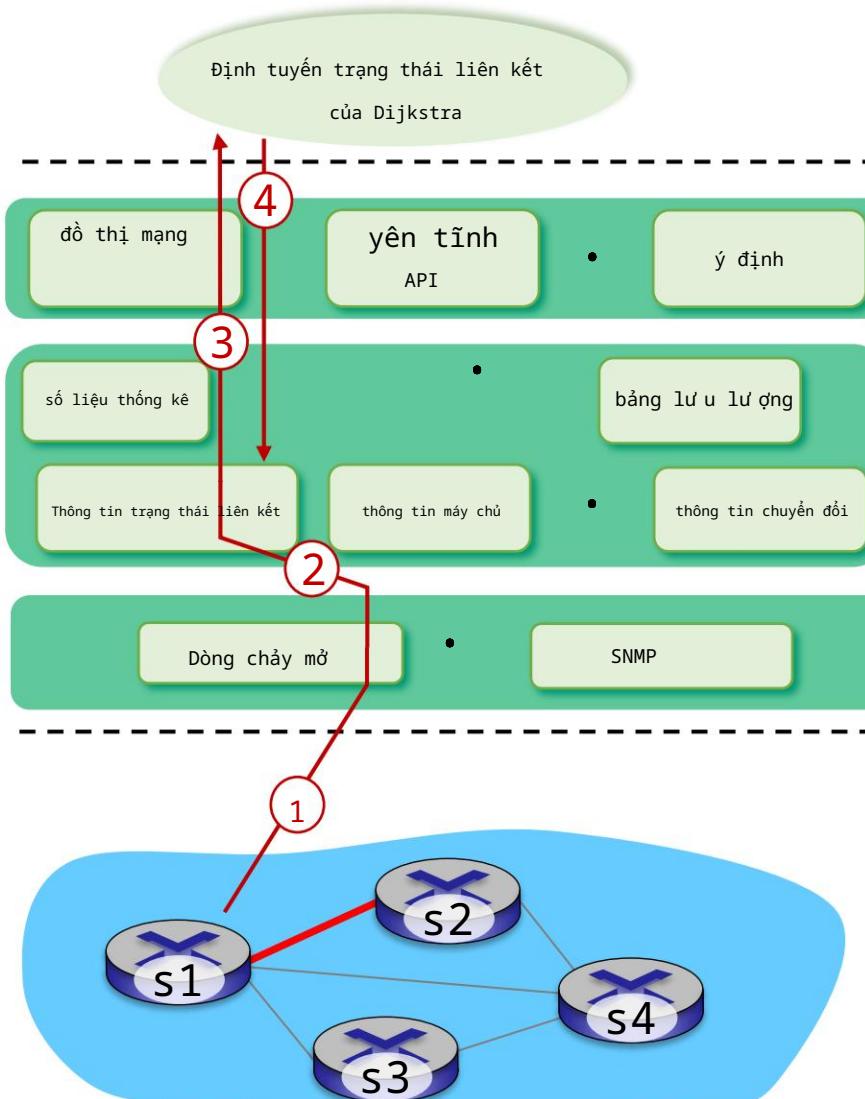
trạng thái cổng: thông báo cho bộ điều khiển về một
thay đổi trên một cổng.

Bộ điều khiển OpenFlow



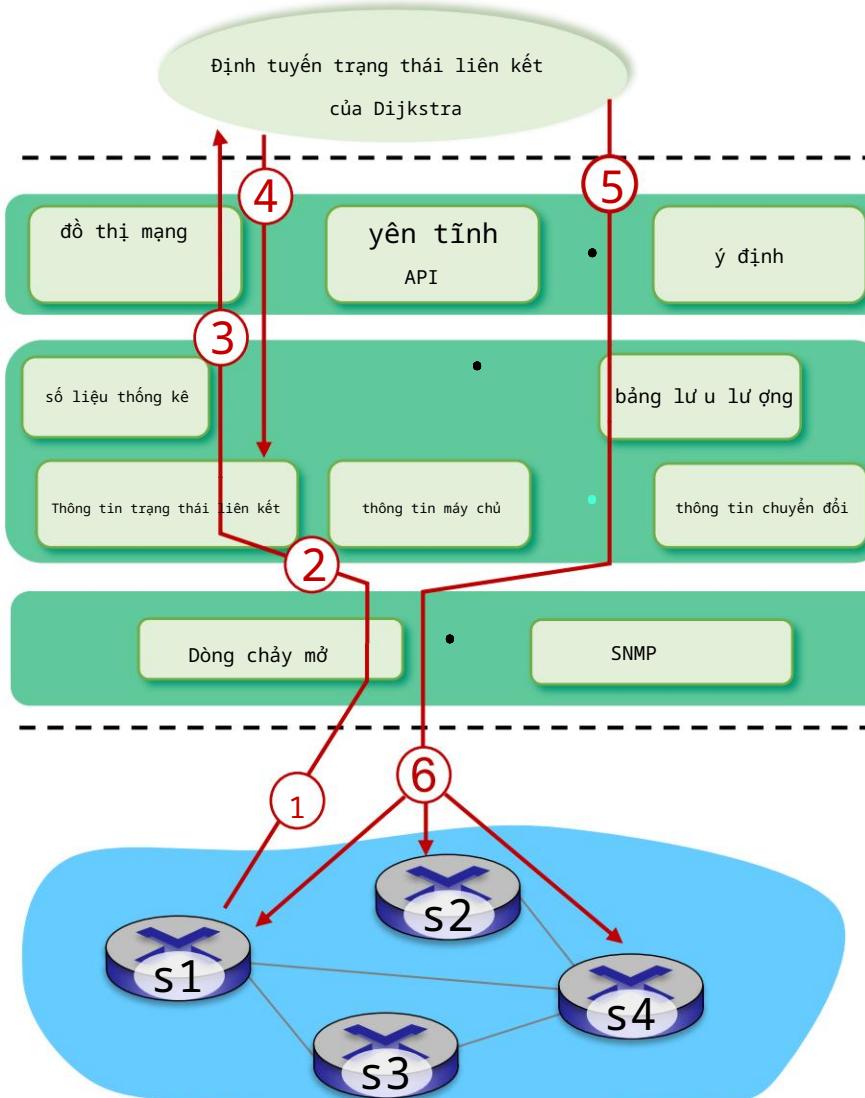
May mắn thay, các nhà khai thác mạng không "lập trình" chuyển mạch bằng cách tạo/gửi trực tiếp các tin nhắn OpenFlow. Thay vào đó, hãy sử dụng **trừu tượng hóa cấp cao hơn** tại bộ điều khiển

SDN: ví dụ tư ơng tác măt phẳng điều khiển/dữ liệu



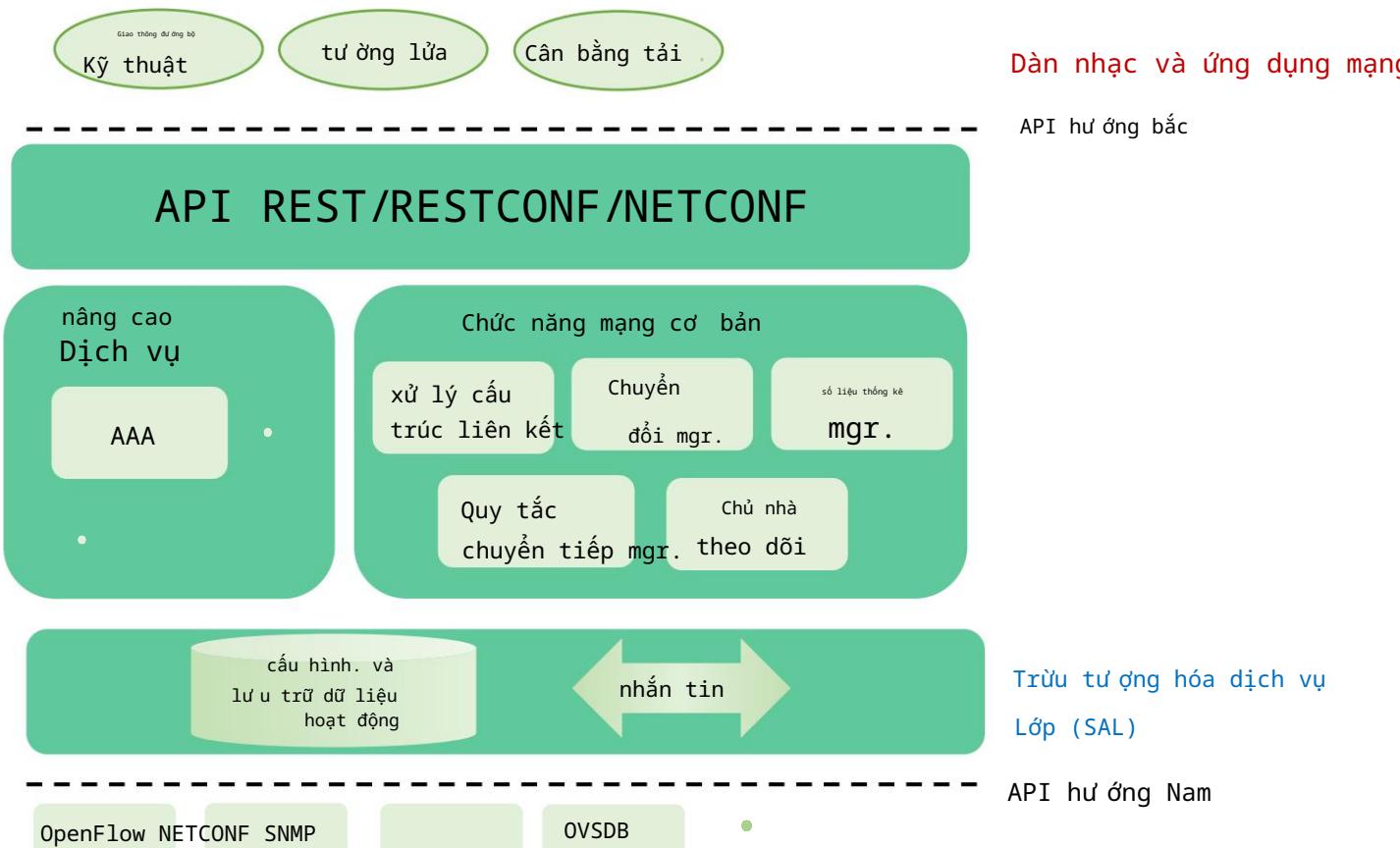
- ① S1, gặp lỗi sử dụng liên kết
Thông báo trạng thái cổng OpenFlow để thông báo cho bộ điều khiển
- ② Bộ điều khiển SDN nhận thông báo OpenFlow, cập nhật thông tin trạng thái liên kết
- ③ Ứng dụng thuật toán định tuyến của Dijkstra trứ ớc đây đã đăng ký để đư ợc gọi khi trạng thái liên kết thay đổi. Nó đư ợc gọi là.
- ④ Thuật toán định tuyến của Dijkstra truy cập thông tin biểu đồ mạng, thông tin trạng thái liên kết trong bộ điều khiển, tính toán các tuyến mới

SDN: ví dụ tư ơng tác măt phẳng điều khiển/dữ liệu



- ⑤ **Ứng dụng định tuyến trạng thái liên kết** tư ơng tác với thành phần tính toán bảng lưu trữ trong bộ điều khiển SDN, **tính toán** các bảng lưu trữ mới cần thiết
- ⑥ **bộ điều khiển** sử dụng OpenFlow để cài đặt các bảng mới trong các công tắc cần cập nhật

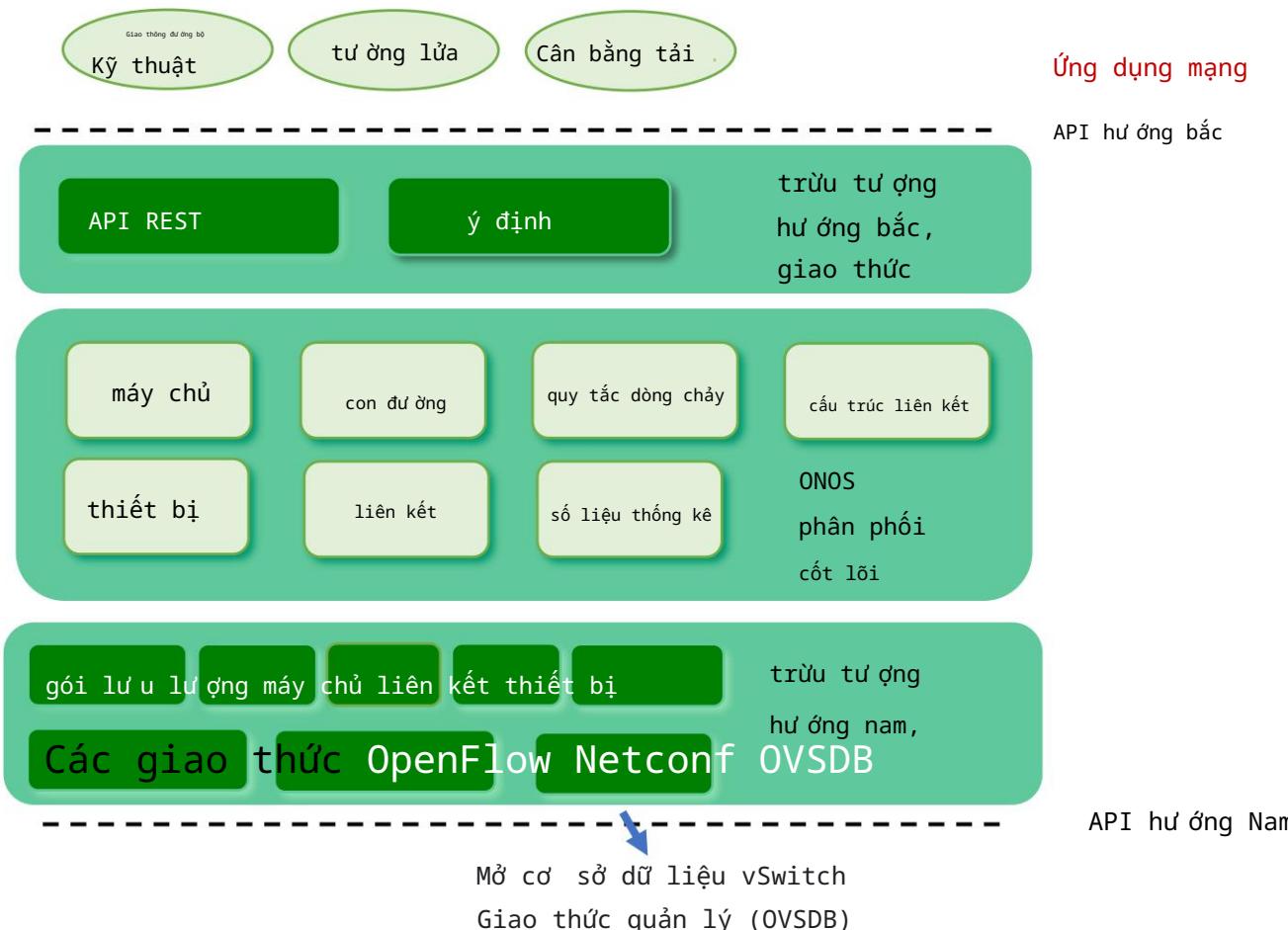
Bộ điều khiển OpenDaylight (ODL)



Tầng Trùu tu ợng Dịch vụ:

kết nối các ứng dụng
và dịch vụ bên trong,
bên ngoài

Bộ điều khiển ONOS



Ứng dụng mạng

API hứa ờng bắc

API hứa ờng Nam

Ứng dụng điều khiển tách biệt với bộ điều khiển

khung ý định: đặc điểm kỹ thuật cấp cao của dịch vụ: **cái gì** hơn là **nó thế nào**

nhấn mạnh đáng kể vào lõi phân tán: độ tin cậy của dịch vụ, nhân rộng hiệu suất sao chép

SDN: thử thách đã chọn

làm cứng mặt phẳng điều khiển: đáng tin cậy, đáng tin cậy, hiệu suất hệ thống phân tán an toàn, có khả năng mở rộng

- mạnh mẽ đối với các lỗi: tận dụng lý thuyết mạnh mẽ về phân phối đáng tin cậy hệ thống điều khiển máy bay
- Độ tin cậy, bảo mật: “đư ợc nung nấu” ngay từ ngày đầu tiên?

mạng, giao thức đáp ứng các yêu cầu dành riêng cho nhiệm vụ , ví dụ: thời

- gian thực, cực kỳ đáng tin cậy, cực kỳ an toàn

Mở rộng quy mô Internet: ngoài một AS duy nhất SDN quan trọng trong mạng di động 5G

SDN và tương lai của các giao thức mạng truyền thống

Bảng chuyển tiếp do SDN tính toán so với do bộ định tuyến tính toán : • chỉ là một ví dụ về tính toán tập trung logic so với giao thức tính toán

ví dụ: người ta có thể tư ởng tư ợng điều khiển tắc nghẽn do SDN tính toán:

- bộ điều khiển đặt tốc độ gửi dựa trên mức độ tắc nghẽn do bộ định tuyến báo cáo (đến bộ điều khiển)



Việc triển khai chức năng mạng (SDN so với các giao thức) sẽ phát triển như thế nào?



Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo Điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

ICMP: giao thức thông báo điều khiển internet

đư ợc sử dụng bởi các máy chủ và bộ
định tuyến để truyền đạt thông tin cấp
độ mạng • báo cáo lỗi: máy chủ, mạng,
cổng, giao thức không thể truy cập đư ợc •
yêu cầu/trả lời phản hồi (đư ợc sử dụng
bằng ping)

IP “phía trên” tầng mạng: • Các bản tin ICMP được mang trong **các datagram IP**

Thông báo ICMP: loại, mã cộng với 8 byte đầu tiên của gói dữ liệu IP gây ra lỗi

Loại Mô tả mã 0 0 trả lời tiếng

vang (ping) è 3 tuy đ ica p ña ña ña ña ña

truy cập thẻ ợt truy cập thẻ 3G và 4G không bị lỗi 3G 6 mạng

dịch không xác định khẩn trương (khoa học) không xác định

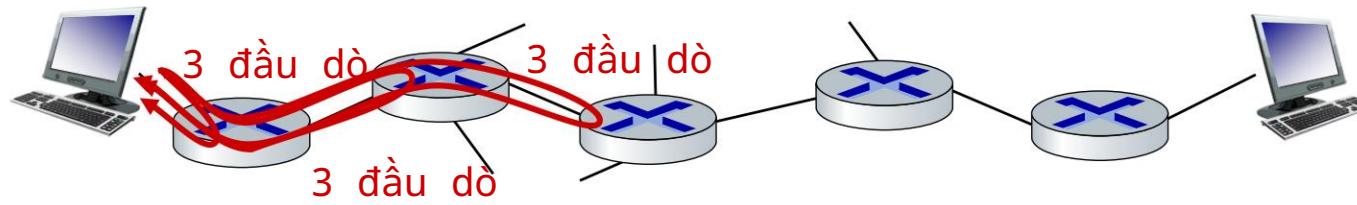
vang (pintu) và nút đai (đầu) màu vàng đồng kèm đinh bô định

8 0 9

10 0 1

0 12 0

Traceroute và ICMP



nguồn gửi tập hợp các phân đoạn UDP đến đích

Bộ 1 bộ có TTL =1, bộ thứ 2 có TTL=2, v.v. •

datagram trong n bộ **thứ** đến n bộ định tuyến:

- bộ định tuyến loại bỏ datagram và gửi nguồn **Thông báo ICMP** (loại 11, mã 0) •

Thông báo ICMP có thể bao gồm tên của bộ định tuyến & địa chỉ IP

khi tin nhắn ICMP đến nguồn: **ghi RTT**

tiêu chí dừng:

Đoạn UDP cuối cùng
đến máy chủ đích
đích trả về thông báo
không thể truy cập cổng
ICMP (loại 3, mã 3)
nguồn dừng

Lớp mạng: lộ trình “mặt phẳng điều khiển”

giới thiệu

giao thức định tuyến

định tuyến trong ISP: OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo Điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF /YANG

quản lý mạng là gì

các hệ thống tự trị (hay còn gọi là “mạng”): hàng nghìn lần tương tác thành phần phần cứng/phần mềm

các hệ thống phức tạp khác yêu cầu giám sát, cấu hình, điều khiển:

- máy bay phản lực, nhà máy điện hạt nhân, những thứ khác?



"Quản lý mạng bao gồm việc triển khai, tích hợp và phối hợp các yếu tố phần cứng, phần mềm và con người để theo dõi, kiểm tra, thăm dò ý kiến, định cấu hình, phân tích, đánh giá và kiểm soát tài nguyên mạng và yếu tố để đáp ứng hiệu suất hoạt động, thời gian thực và Yêu cầu về chất lượng dịch vụ với chi phí hợp lý."

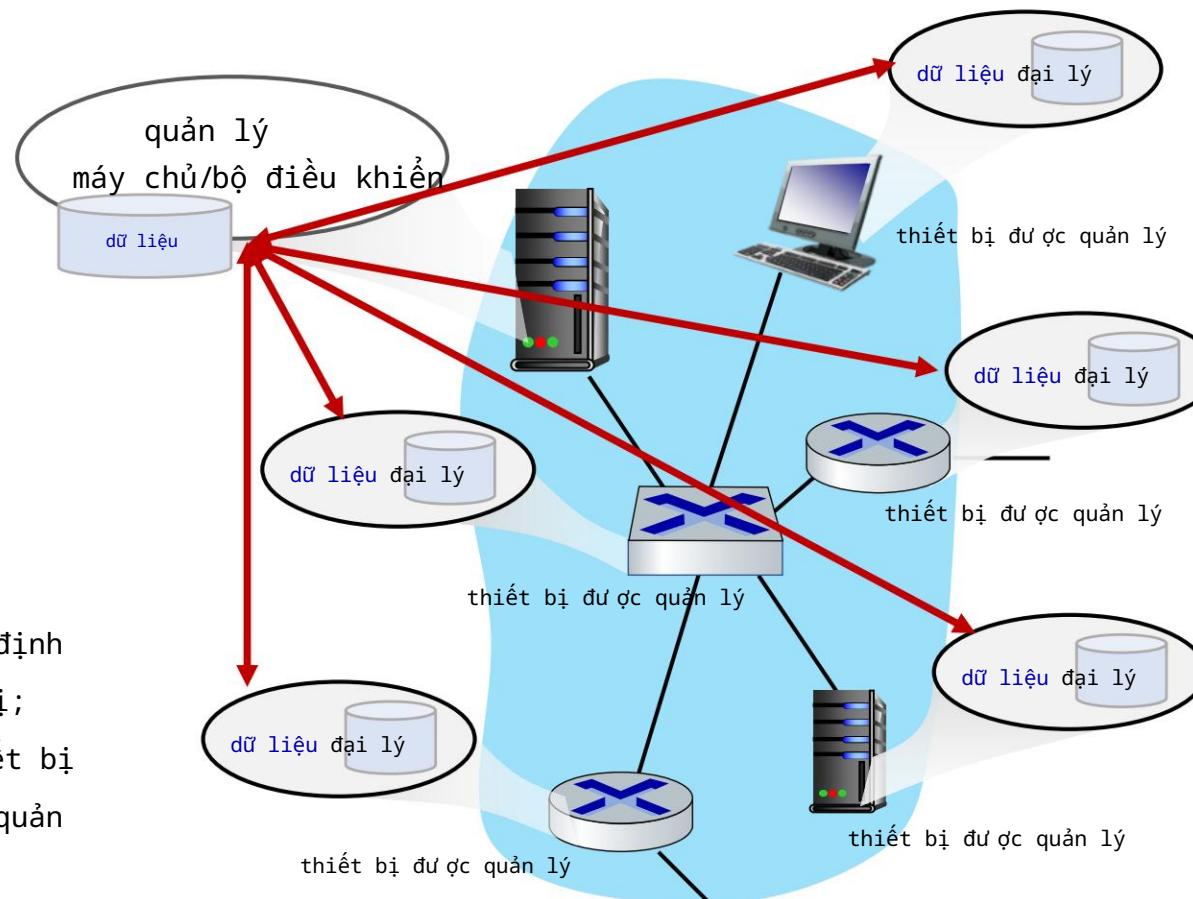
Các thành phần quản lý mạng

Quản lý máy chủ: Ứng dụng, thư ờng có ngư ời quản lý mạng (con ngư ời) trong vòng lặp

Giao thức

quản lý **mạng** :

được sử dụng bởi máy chủ quản lý để truy vấn, định cấu hình, quản lý thiết bị; được sử dụng bởi các thiết bị để thông báo cho máy chủ quản lý dữ liệu, sự kiện.



Thiết bị được quản lý:

thiết bị có các thành phần phần mềm, phần cứng có thể quản lý, có thể định cấu hình

Dữ liệu: dữ liệu cấu hình

"trạng thái" của thiết bị, dữ liệu vận hành, thống kê thiết bị

Phư ơ ng pháp tiếp cận của nhà điều hành mạng để quản lý

CLI (Giao diện dòng lệnh)

- các vấn đề về người vận hành (loại, tập lệnh) trực tiếp đến từng thiết bị (ví dụ: qua ssh)

SNMP/MIB • người

vận hành truy vấn/thiết lập dữ liệu thiết bị
(MIB) sử dụng Mạng đơn giản

Giao thức quản lý (SNMP)

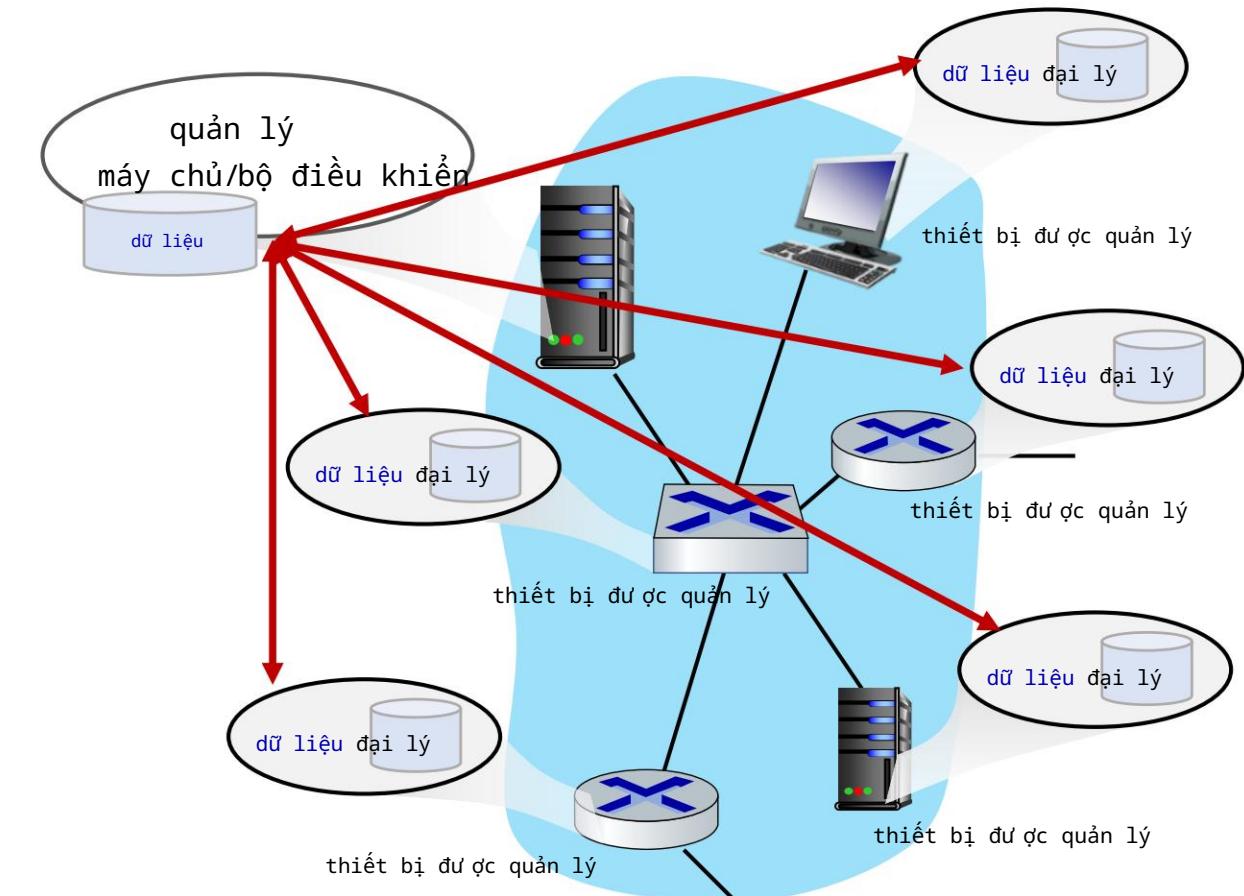
NETCONF/YANG • trừu tượng

hơn, toàn mạng, toàn diện hơn • nhấn mạnh vào
cấu hình đa thiết bị
sự quản lý.

YANG: ngôn ngữ mô hình hóa dữ liệu •

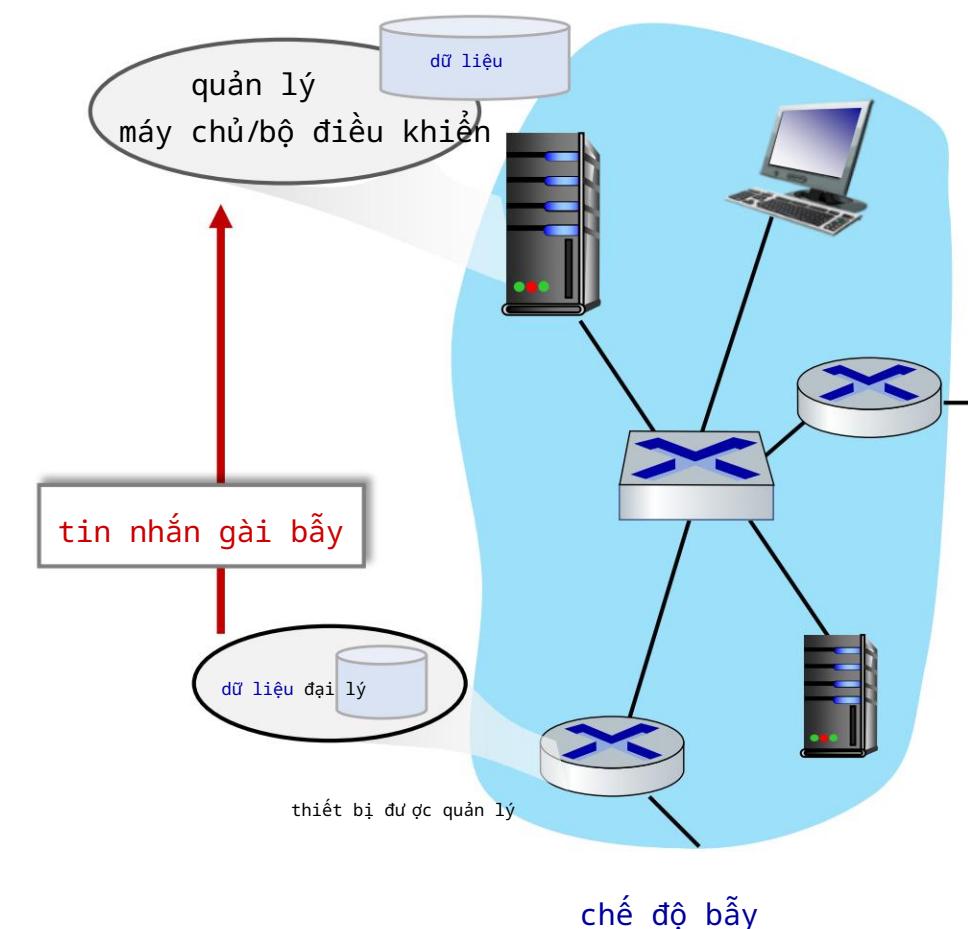
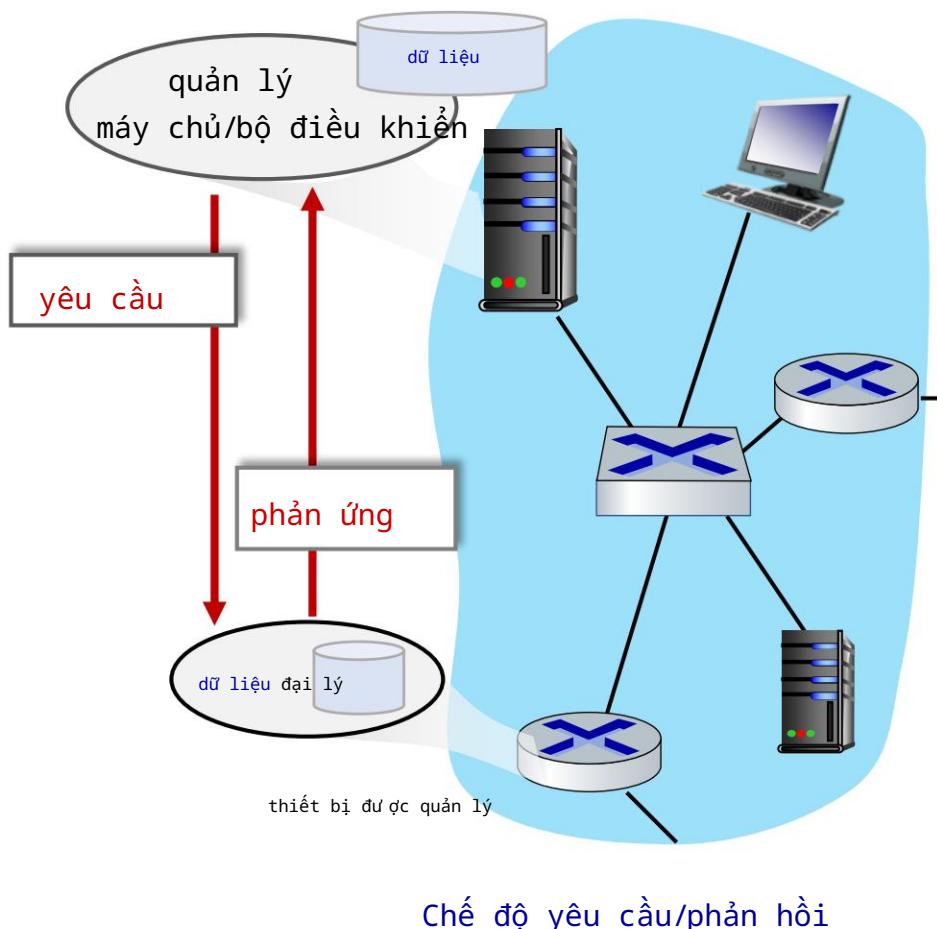
NETCONF: giao tiếp tương thích với YANG

hành động/dữ liệu đến/từ/giữa các thiết bị từ xa



giao thức SNMP

Hai cách để truyền tải thông tin MIB, lệnh:



Giao thức SNMP: các loại thông báo

Loại tin nhắn Chức năng	
Nhận yêu cầu GetNextRequest GetBulkRequest	manager-to-agent: "lấy dữ liệu cho tôi" (ví dụ dữ liệu, dữ liệu tiếp theo trong danh sách, khôi dữ liệu).
Đặt yêu cầu	manager-to-agent: đặt giá trị MIB
Phản ứng	Đại lý đến người quản lý: giá trị, phản hồi Yêu cầu
Bảng	Agent-to-manager: thông báo cho người quản lý về sự kiện đặc biệt

Giao thức SNMP: định dạng tin nhắn

loại tin nhắn 0-3



tin nhắn loại 4



SNMP PDU

SNMP: Cơ sở thông tin quản lý (MIB)

dữ liệu hoạt động (và một số cấu hình) của thiết bị đư ợc quản lý



tập hợp thành **mô-đun MIB** của thiết

bị • 400 mô-đun MIB đư ợc định nghĩa trong RFC's; nhiều MIB dành

riêng cho nhà cung cấp khác **Cấu trúc thông tin quản lý (SMI):** ngôn ngữ định
nghĩa dữ liệu Các biến MIB ví dụ cho giao thức UDP:

ID đối tượng	Tên	Loại	Bình luận
1.3.6.1.2.1.7.1	UDPIInDatagrams	Tổng bộ đếm 32 bit # gói dữ liệu đư ợc phân phối	
1.3.6.1.2.1.7.2	UDPNoPorts	Bộ đếm 32 bit # gói dữ liệu không gửi đư ợc (không có ứng dụng tại cổng)	
1.3.6.1.2.1.7.3	UIInErrors	Bộ đếm 32 bit # gói dữ liệu không gửi đư ợc (tất cả các lý do khác)	
1.3.6.1.2.1.7.4	UDPOutDatagrams	Tổng bộ đếm 32 bit # datagram đã gửi 1.3.6.1.2.1.7.5	
udpTable	SỰ NỐI TIẾP	một mục nhập cho mỗi cổng hiện đang đư ợc sử dụng	

Tổng quan về NETCONF

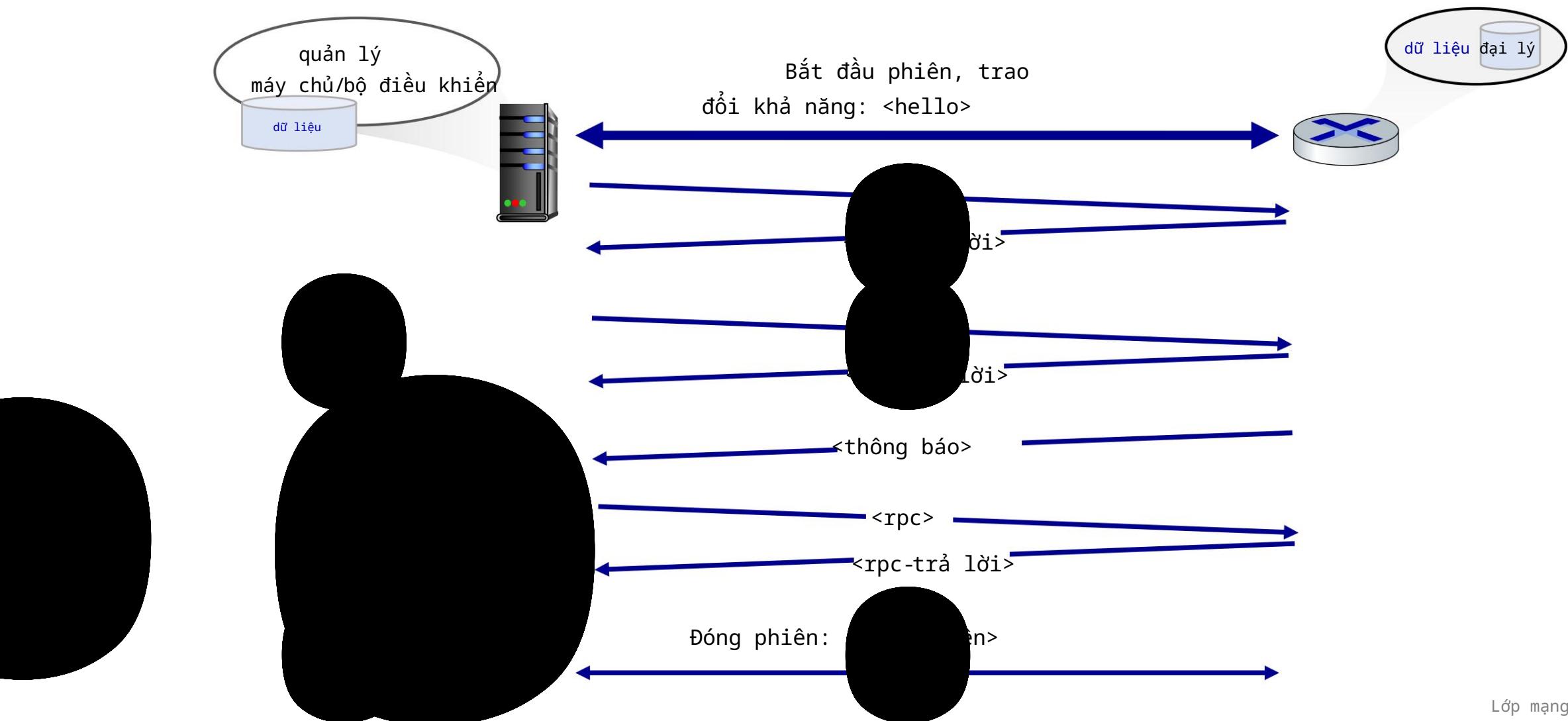
mục tiêu: chủ động quản lý/cấu hình thiết bị trên toàn mạng hoạt động giữa máy chủ quản lý và thiết bị mạng được quản lý

- hành động: truy xuất, đặt, sửa đổi, kích hoạt cấu hình
- hành động cam kết nguyên tử trên nhiều thiết bị truy vấn dữ liệu vận
- hành và thống kê
- đăng ký nhận thông báo từ thiết bị

mô hình gọi thủ tục từ xa (RPC)

- Thông báo giao thức NETCONF được mã hóa bằng XML
- được trao đổi qua giao thức vận chuyển an toàn, đáng tin cậy (ví dụ: TLS)

NETCONF khởi tạo, trao đổi, đóng



Hoạt động NETCONF đã chọn

NETCONF	hoạt động Mô tả
<get-config>	Truy xuất tất cả hoặc một phần của cấu hình đã cho. Một thiết bị có thể có nhiều cấu hình.
<lấy>	Truy xuất tất cả hoặc một phần của cả trạng thái cấu hình và dữ liệu trạng thái hoạt động.
<sửa-cấu hình>	Thay đổi cấu hình đã chỉ định (có thể đang chạy) tại thiết bị được quản lý. Thiết bị được quản lý <rpc-reply> chứa <ok> hoặc <rpcerror> có khôi phục.
<khóa>, <mở khóa>	Khóa (mở khóa) kho dữ liệu cấu hình tại thiết bị được quản lý (để khóa các lệnh NETCONF, SNMP hoặc CLI từ các nguồn khác).
<tạo-đăng ký>, Bật đăng ký thông báo sự kiện từ thiết bị được quản lý <thông báo>	

Thông báo NETCONF RPC mẫu

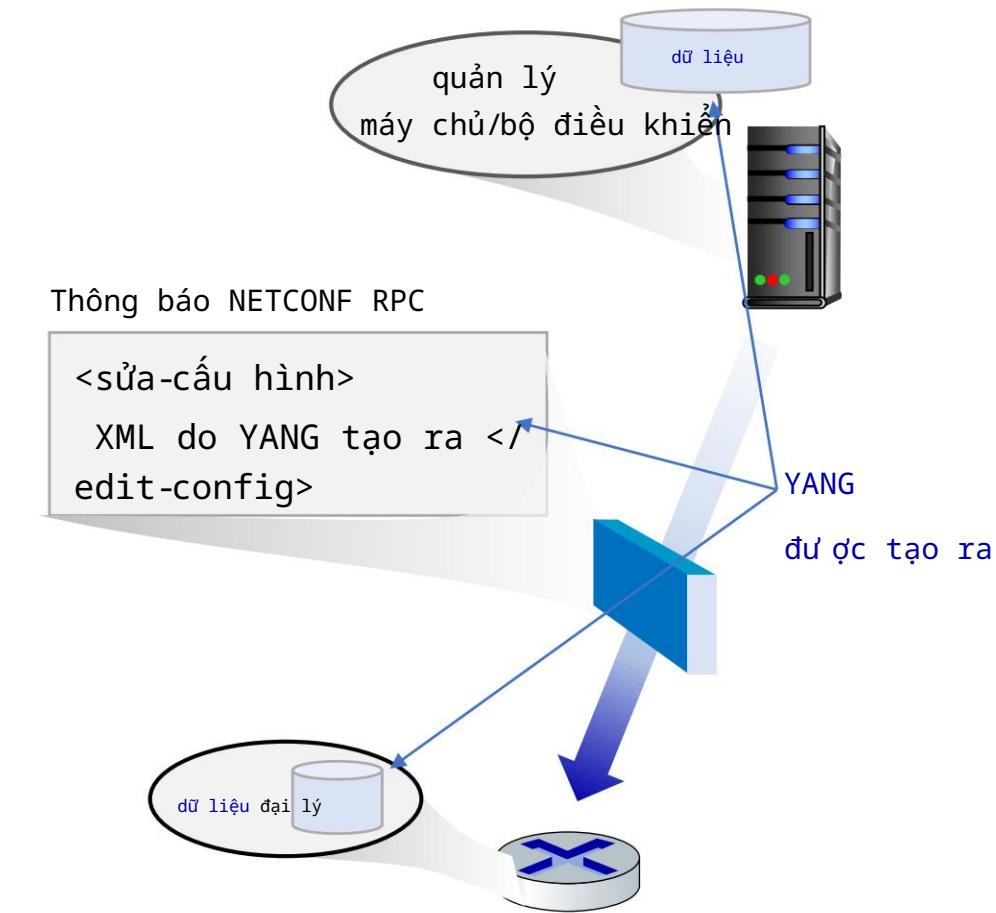
```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <rpc message-id="101" id tin nhắn lưu ý
03   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
04     <edit-config>    thay đổi cấu hình
05       <target>
06         <running/> thay đổi cấu hình đang chạy
07       </target>
08     <config>
09       <top xmlns="http://example.com/schema/
10         1.2/config">
11           <interface>
12             <name>Ethernet0/0</name> thay đổi MTU của giao diện Ethernet 0/0 thành 1500
13             <mtu>1500</mtu>
14           </interface>
15         </top>
16       </config>
17     </edit-config>
18   </rpc>
```

DƯ ƠNG

ngôn ngữ mô hình hóa dữ liệu được sử dụng để xác định **cấu trúc, cú pháp, ngữ nghĩa** của dữ liệu quản lý mạng NETCONF • các kiểu dữ liệu tích hợp, như SMI

Tài liệu XML mô tả thiết bị, khả năng có thể được tạo ra từ Mô tả YANG có

thể hiện **các ràng buộc** giữa các dữ liệu phải được thỏa mãn bởi một cấu hình NETCONF hợp lệ • đảm bảo các cấu hình NETCONF đáp ứng các ràng buộc về tính chính xác, nhất quán



Lớp mạng: Tóm tắt

chúng tôi đã học được rất nhiều!

tiếp cận mặt phẳng điều khiển mạng • điều khiển
theo bộ định tuyến (truyền thông) • điều khiển
tập trung về mặt logic (mạng được xác định bằng phần mềm) thuật
toán định tuyến truyền thông

- triển khai trên Internet: OSPF, BGP

Bộ điều khiển SDN

- triển khai trong thực tế: ODL, ONOS Giao

thức thông báo điều khiển Internet (ICMP) quản lý

mạng (SNMP/SMI, NETCONF/YANG)

điểm dừng tiếp theo: lớp liên kết!

Lớp mạng, mặt phẳng điều khiển: Xong!

giới thiệu

giao thức định tuyến

trạng thái liên kết

vector khoảng cách

định tuyến trong ISP : OSPF

định tuyến giữa các ISP: BGP

Mặt phẳng điều khiển SDN

Thông báo điều khiển Internet
giao thức



quản lý mạng, cấu hình

- SNMP
- NETCONF/YANG

Các slide Chư ơ ng 5 bổ sung

Vector khoảng cách: một ví dụ khác

chi Đx ()		
x	y	z
0	2	7
∞	∞	∞
∞	∞	∞

phi cho xyz

chi Đx ()		
x	y	z
0	2	3
2	0	1
7	1	0

phi cho xyz

$$\begin{aligned}Dx(z) &= \min\{cx, y + Dy(z), cx, z + Dz(z)\} = \min\{2+1, 7+0\} = 3 \\Dx(y) &= \min\{cx, y + Dy(y), cx, z + Dz(y)\} \\&= \min\{2+0, 7+1\} = 2\end{aligned}$$

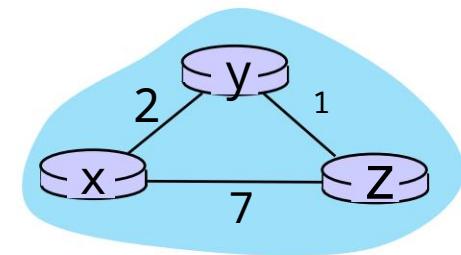
chi đê ()		
x	y	z
∞	∞	∞
2	0	1
∞	∞	∞

phi cho xyz

chi Dz ()		
x	y	z
∞	∞	∞
∞	∞	∞
7	1	0

phi cho xyz

thời gian



Vector khoảng cách: một ví dụ khác

