

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Kiến trúc của bộ định tuyến

4.3 Giao thức mạng Internet (IP): IPv4 và IPv6

4.3.1. Cấu trúc gói tin IPv4

4.3.2. Định địa chỉ IPv4

4.3.3. NAT: dịch chuyển địa chỉ mạng

4.3.4. IPv6

4.4 Các giải thuật định tuyến

4.4.1. Link state

4.4.2. Distance vector

4.5 Định tuyến trên mạng Internet:

RIP, OSPF, BGP

Tăng mạng 4-74

Một giải thuật định tuyến Link-State

Giải thuật Dijkstra

- Tất cả các nút đều biết được cấu trúc mạng và chi phí của các liên kết trên mạng
 - Được thực hiện bằng cách "quảng bá trạng thái liên kết"
 - Tất cả các nút có thông tin giống nhau
- Tính toán chi phí thấp nhất đường đi từ một nút ("nguồn") đến tất cả các nút khác.
 - Cho *bảng chuyển tiếp* của nút đó
- Lặp: sau k lần duyệt, sẽ biết được chi phí thấp nhất tới k đích

Ký hiệu:

- $c(x,y)$: chi phí liên kết từ nút x tới y ; $= \infty$ nếu không có kết nối trực tiếp đến nút lân cận
- $D(v)$: giá trị hiện tại của chi phí đường đi từ nguồn đến đích
- $p(v)$: nút trước nằm trên đường đi từ nguồn đến v
- N' : tập các nút mà chi phí đường đi thấp nhất đã được xác định

Tăng mạng 4-75

Giải thuật Dijkstra

1 Khởi tạo:

2 $N' = \{u\}$

3 for tất cả các nút v

4 if v kề với u

5 then $D(v) = c(u,v)$

6 else $D(v) = \infty$

7

8 Lặp

9 tìm w không thuộc N' mà $D(w)$ là nhỏ nhất

10 thêm w vào N'

11 Cập nhật $D(v)$ cho tất cả v kề với w và không thuộc N' :

12 $D(v) = \min(D(v), D(w) + c(w,v))$

13 /* chi phí mới đến v là chính giá trị cũ hoặc chi phí đường đi ngắn nhất đã tính tới w cộng với chi phí từ w tới v */

15 cho đến khi tất cả các nút nằm trong N'

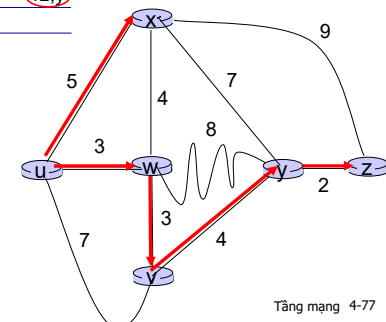
Tăng mạng 4-76

Giải thuật Dijkstra: Ví dụ

Bước	N'	$D(v)$ $p(v)$	$D(w)$ $p(w)$	$D(x)$ $p(x)$	$D(y)$ $p(y)$	$D(z)$ $p(z)$
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwvx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxyvz					

Chú ý:

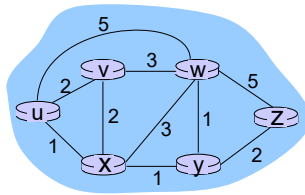
- Xây dựng cây đường đi ngắn nhất bằng cách đi tìm các nút trước.
- Các mối quan hệ có thể tồn tại (có thể được chia tùy ý)



Tăng mạng 4-77

Giải thuật Dijkstra: Một ví dụ khác

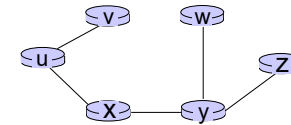
Bước	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x	2,x	∞	∞
2	uxy	2,u	3,y		4,y	
3	uxyv		3,y		4,y	
4	uxyvw				4,y	
5	uxyvwz					



Tăng mạng 4-78

Giải thuật Dijkstra: Ví dụ (2)

Kết quả cây đường đi ngắn nhất từ u:



Kết quả bảng chuyển tiếp trong u:

Đích	Liên kết
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Tăng mạng 4-79

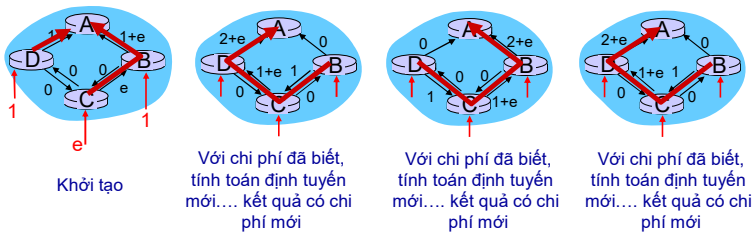
Giải thuật Dijkstra: Thảo luận

Độ phức tạp của giải thuật: n nút

- ❖ Mỗi lần duyệt: cần kiểm tra lại tất cả các nút w không thuộc N
- ❖ $n(n+1)/2$ phép so sánh: $O(n^2)$
- ❖ Có thể cài đặt hiệu quả hơn: $O(n \log n)$

Có thể dao động:

- ❖ Ví dụ: chi phí kết nối bằng tổng lưu lượng mạng:



Tăng mạng 4-80

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Kiến trúc của bộ định tuyến

4.3 Giao thức mạng Internet (IP): IPv4 và IPv6

- 4.3.1. Cấu trúc gói tin IPv4
- 4.3.2. Định địa chỉ IPv4
- 4.3.3. NAT: dịch chuyển địa chỉ mạng
- 4.3.4. IPv6

4.4 Các giải thuật định tuyến

4.4.1. Link state

4.4.2. Distance vector

4.5 Định tuyến trên mạng Internet:

RIP, OSPF, BGP

Tăng mạng 4-81

Giải thuật distance vector

Công thức Bellman-Ford

Cho

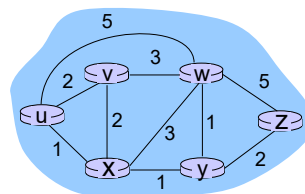
$d_x(y) :=$ chi phí thấp nhất đường đi từ x đến y
thì

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

\min được tính trên tất cả các lân cận v của x
Chi phí tới lân cận v
Chi phí từ lân cận v tới đích y

Tăng mạng 4-82

Ví dụ Bellman-Ford



Có: $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Công thức B-F cho:

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

Giá trị nhỏ nhất đạt được của nút là hop kế tiếp trong đường đi ngắn nhất, được dùng trong bảng chuyển tiếp.

Tăng mạng 4-83

Giải thuật distance vector

- ❖ $D_x(y)$ = ước lượng chi phí thấp nhất từ x đến y
 - x duy trì véc-tơ khoảng cách $D_x = [D_x(y): y \in N]$
- ❖ Nút x:
 - Biết chi phí đến mỗi nút lân cận v: $c(x, v)$
 - Duy trì véc-tơ khoảng cách của các nút lân cận của. Với mỗi nút lân cận v, x duy trì $D_v = [D_v(y): y \in N]$

Tăng mạng 4-84

Giải thuật distance vector

Ý tưởng chính:

- ❖ Mỗi nút định kỳ gửi ước lượng véc-tơ khoảng cách của nó đến các nút lân cận
- ❖ Khi x nhận được ước lượng DV mới từ nút lân cận, nó sẽ cập nhật DV của nó bằng cách dùng công thức B-F:

$$D_x(y) \leftarrow \min_v \{ c(x, v) + D_v(y) \} \text{ với mỗi nút } y \in N$$

- ❖ Dưới các điều kiện tự nhiên, ước lượng $D_x(y)$ sẽ hội tụ về giá trị chi phí thấp nhất thực tế $d_x(y)$

Tăng mạng 4-85

Giải thuật distance vector

Lập, không đồng bộ: mỗi lần lập cục bộ có nguyên nhân từ:

- ❖ Thay đổi chi phí liên kết cục bộ
- ❖ Thông báo cập nhật DV từ nút lân cận

Phân bố:

- ❖ Mỗi nút *chỉ* thông báo đến các nút lân cận khi DV của nó thay đổi
 - Các lân cận sau đó sẽ thông báo đến các lân cận khác của nó nếu cần

Mỗi nút:

Chờ cho (thay đổi trong chi phí liên kết cục bộ hoặc thông báo từ nút lân cận)

Tính toán lại các ước lượng

Nếu DV tới bất kỳ đích nào có thay đổi, thì **thông báo** cho các nút lân cận

Tăng mạng 4-86

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

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

Bảng nút x

Chi phí tới	x	y	z
x	0	2	7
từ y	∞	0	∞
z	∞	∞	0

Chi phí tới

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

Bảng nút y

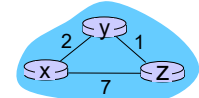
Chi phí tới	x	y	z
x	∞	0	∞
từ y	2	0	1
z	∞	∞	0

Bảng nút z

Chi phí tới	x	y	z
x	∞	∞	∞
từ y	∞	∞	0
z	7	1	0

Thời gian

Tăng mạng 4-87



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

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

Bảng nút x

Chi phí tới	x	y	z
x	0	2	7
từ y	∞	0	∞
z	∞	∞	0

Chi phí tới

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

Bảng nút y

Chi phí tới	x	y	z
x	∞	0	∞
từ y	2	0	1
z	∞	∞	0

Chi phí tới

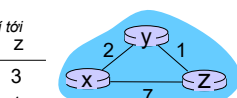
x	y	z
0	2	3
2	0	1
7	1	0

Bảng nút z

Chi phí tới	x	y	z
x	∞	∞	∞
từ y	∞	∞	0
z	7	1	0

Chi phí tới

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



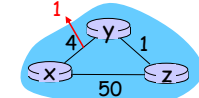
Thời gian

Tăng mạng 4-88

Distance vector: Chi phí kết nối thay đổi

Chi phí kết nối thay đổi:

- ❖ Nút kiểm tra thay đổi chi phí kết nối cục bộ
- ❖ Cập nhật thông tin tìm đường, tính toán lại véc-tơ khoảng cách
- ❖ Nếu DV thay đổi, thông báo đến các lân cận



“thông tin tốt truyền đi nhanh”

t_0 : y kiểm tra thay đổi chi phí kết nối, cập nhật lại DV của nó, và thông báo cho các lân cận.

t_1 : z nhận thông tin cập nhật từ y, cập nhật lại bảng của nó, tính toán chi phí thấp nhất mới tới x, và gửi DV của nó tới các lân cận.

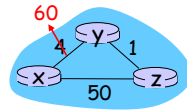
t_2 : y nhận cập nhật của z, cập nhật lại bảng khoảng cách của nó. Chi phí thấp nhất của y *không* thay đổi, nên y *không* cần gửi thông báo tới z.

Tăng mạng 4-89

Distance vector: Chi phí kết nối thay đổi

Chi phí kết nối thay đổi:

- ❖ Nút kiểm tra thay đổi chi phí cục bộ
- ❖ **Tin xấu truyền đi chậm** – vấn đề “đếm vô hạn”!
- ❖ 44 lần duyệt trước khi thuật toán ổn định: (xem thêm trong tài liệu)



poisoned reverse:

- ❖ Nếu Z định tuyến Y đi thẳng tới X :
 - Z thông báo cho Y khoảng cách (của Z) tới X là vô hạn (để Y không đi tới X qua Z)
- ❖ Sẽ giải quyết triệt để vấn đề đếm vô hạn?

Tăng mạng 4-90

So sánh các giải thuật LS và DV

Độ phức tạp thông báo

- ❖ **LS:** với n nút, E liên kết, $O(nE)$ thông báo được gửi đi
- ❖ **DV:** chỉ trao đổi giữa các nút lân cận
 - Thời gian hội tụ thay đổi

Tốc độ hội tụ

- ❖ **LS:** Thuật toán $O(n^2)$ yêu cầu $O(nE)$ thông báo
 - Có thể có dao động
- ❖ **DV:** Thời gian hội tụ thay đổi
 - Có thể lập định tuyến
 - Vấn đề đếm vô hạn

Sự linh hoạt: điều gì sẽ xảy ra nếu router hoạt động sai chức năng?

LS:

- Nút có thể thông báo chi phí **kết nối** không chính xác
- Mỗi nút chỉ tính toán bằng **riêng** của nó

DV:

- Nút có thể thông báo chi phí **đường đi** không chính xác
- Mỗi bảng của nút được dùng bởi các nút khác
 - Lỗi lan truyền thông qua mạng

Tăng mạng 4-91