



NHẬP MÔN MẠNG MÁY TÍNH

Chương 4: Tầng Network

I. Tổng quan:

- Tầng Network: Cung cấp kênh truyền logic giữa các **hosts**. Các segment của tầng vận chuyển được đóng gói (encapsulation) và gửi từ host đến host.

⇒ **Các giao thức của tầng mạng hoạt động trong mọi **host**, mọi **router****

- Router sẽ xem xét các trường trong header trong tất cả các gói tin (IP datagram) đi qua nó.

- Chức năng chính:

+ Routing: Xác định đường đi cho các gói từ nguồn đến đích.

+ Forwarding: chuyển các gói tin (packet) từ đầu vào đến đầu ra thích hợp của router.

⇒ **Bảng Forwarding:** Routing xác định đường đi, Forwarding thực hiện việc chuyển gói bên trong một router.

- Thiết lập kết nối: ATM, Frame Relay, X.25 ⇒ Mạng mạch ảo.

- Mô hình dịch vụ mạng:

Kiến trúc mạng	Mô hình dịch vụ	Đảm bảo băng thông	Đảm bảo không mất dữ liệu	Thứ tự	Đảm bảo thời gian	Bảo hiệu nhện
Internet	Best effort	Không	Không	Không	Không	Không
ATM	CBR	Tốc độ không đổi	Có	Có	Có	Không tắc nhện
ATM	VBR	Có	Có	Có	Có	Không tắc nhện
ATM	ABR	Bảo đảm tối thiểu	Không	Có	Không	Có
ATM	UBR	Không	Không	Có	Không	Không



II. Mạng mạch ảo và mạng chuyển gói:

- Mạng mạch ảo (Virtual circuits): Thiết lập kết nối ảo. Một kết nối ảo bao gồm:
 - + Đường đi (path) từ nguồn tới đích.
 - + Số hiệu nhận dạng kết nối ảo (VC numbers).
 - + Các mục trong các bảng forwarding ở trong các router dọc theo đường đi.
- Mạng chuyển gói (Datagram network):
 - + Không thiết lập kết nối ảo.
 - + Gói tin được đưa lên mạng và định tuyến nhờ router.
 - + Thêm địa chỉ nguồn đích vào gói tin.

III. So trùng phần đầu dài nhất (Longest prefix matching):

- Kiểm tra phần đầu dài nhất được cho trong bảng.
- Nếu trùng thì đưa vào interface tương ứng.

VD: Cho bảng forwarding như sau: Giả sử router tiếp nhận và chuyển tiếp các gói tin có địa chỉ đích là 8 bits và sử dụng phương pháp “Longest match prefix” – So sánh phần đầu dài nhất. Cho địa chỉ đích và xác định interface tương ứng cho các trường hợp dưới đây

Prefix match	Interface
11	1
01	2
110	3
011	4
100	5
Otherwise	6

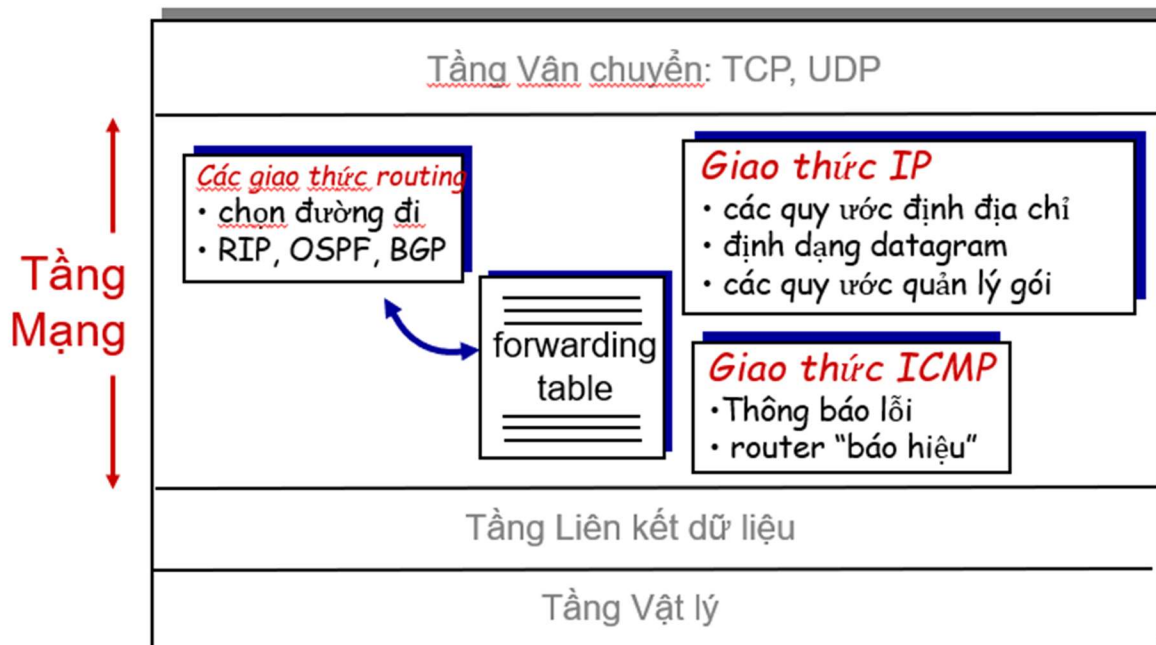


01011011	2
01110111	4
10100111	6

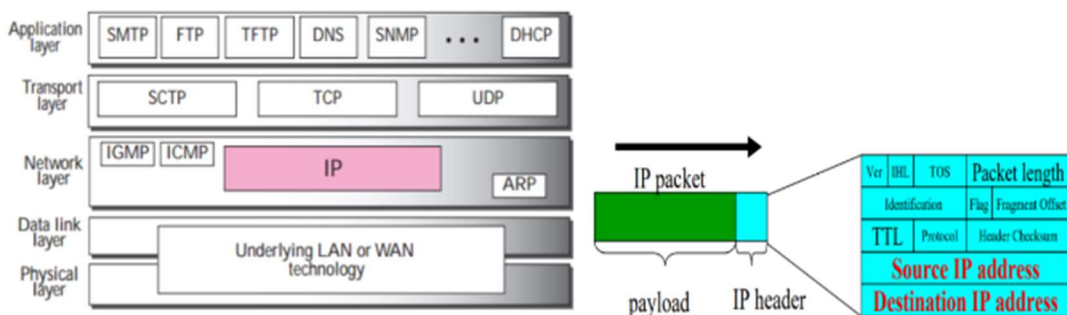
IV. IP (Internet Protocol)

1. Tầng Mạng Internet:

- Các chức năng tầng Mạng của host và router:

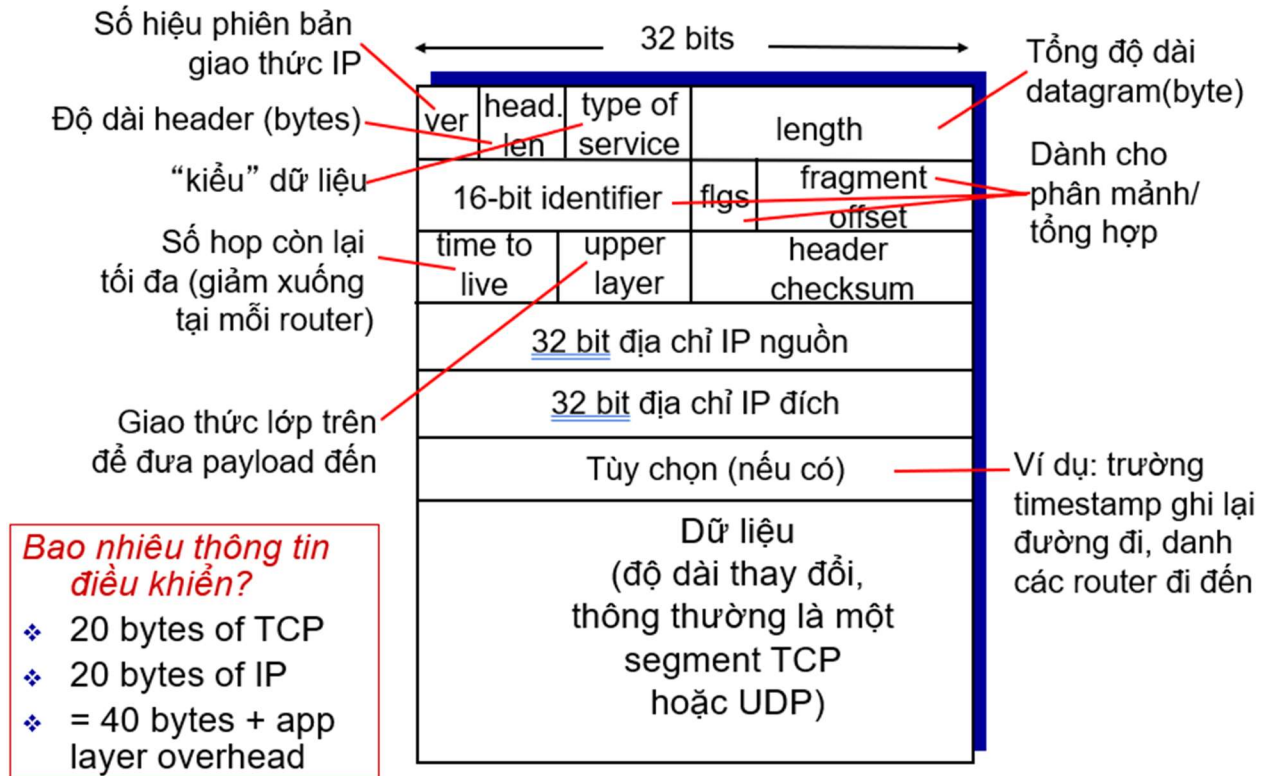


- Vị trí của IP:



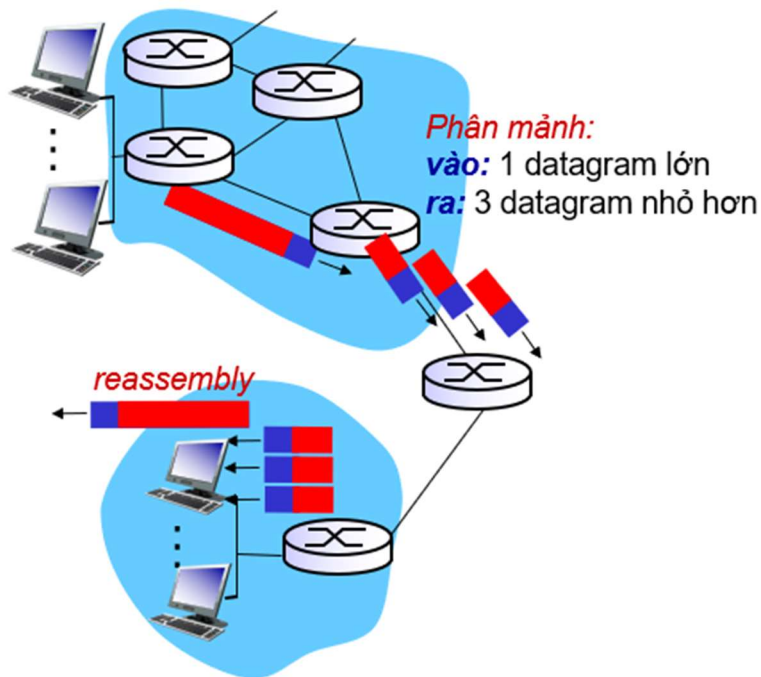


2. Định dạng IP Datagram:



3. Phân mảnh và tổng hợp IP:

- Các đoạn kết nối mạng có MTU (max.transfer size) – frame lớn nhất có thể truyền trên kết nối:
- + Các kiểu kết nối khác nhau có các MTU khác nhau
- Các gói IP datagram lớn được chia (“fragmented”) bên trong mạng:
- + 1 datagram thành 1 vài datagram
- + “Tổng hợp” chỉ được thực hiện ở đích cuối cùng
- + Các bit của IP header được sử dụng để xác định, xếp thứ tự các fragment liên quan



Quá trình phân mảnh

Ví dụ:

- ❖ 4000 byte datagram
- ❖ MTU = 1500 bytes

1480 bytes
trong trường dữ liệu

offset =
 $1480/8$

length	ID	fragflag	offset
=4000	=x	=0	=0

1 datagram lớn thành vài datagram nhỏ hơn

length	ID	fragflag	offset
=1500	=x	=1	=0

length	ID	fragflag	offset
=1500	=x	=1	=185

length	ID	fragflag	offset
=1040	=x	=0	=370



VD: Khi gửi một gói tin IPV4 có kích thước là 8000 bytes vào một mạng có kích thước của MTU là 2000 byte. Biết kích thước của phần header gói tin là 20 bytes, kích thước của gói tin cuối cùng là?

$(2000 - 20) * 4 + 80 = 8000 \Rightarrow$ **5 mảnh, mảnh cuối cùng 680 bytes**

Mảnh	Length	ID	Fraglag	Offset	Còn lại
1	2000	X	1	0	6020
2	2000	X	1	$(1980*1)/8$	4040
3	2000	X	1	$(1980*2)/8$	2060
4	2000	X	1	$(1980*3)/8$	80
5	80	X	0	$(1980*4)/8$	0

V. IP Address (Địa chỉ IP)

1. IPv4 addressing:

- **Địa chỉ IP:** gồm 32-bit, được chia thành 4 cụm 8 bit (gọi là các octet), nhận dạng cho host, router interface.
 - **Interface:** Kết nối giữa host/router và đường kết nối vật lý.
 - + Router thường có nhiều interface
 - + Host thường có 1 hoặc 2 interface (ví dụ wired Ethernet, wireless 802.11)
- \Rightarrow Mỗi địa chỉ IP được kết nối với mỗi interface.

2. Các subnet (mạng con):

- **Địa chỉ IP:**
 - + Subnet: Các bit bên trái.
 - + Host: Các bit bên phải.
- **Subnet:** Kết nối giữa host/router và đường kết nối vật lý.
 - + Các interface của thiết bị có phần subnet của địa chỉ IP giống nhau
 - + Có thể giao tiếp vật lý với nhau mà không cần **router trung gian can thiệp**.



- Phương pháp:

- + Để xác định các subnet, tách mỗi interface từ host hoặc router của nó, tạo vùng các mạng độc lập
- + Mỗi mạng độc lập được gọi là một **subnet**.

3. Phân lớp địa chỉ IPv4:

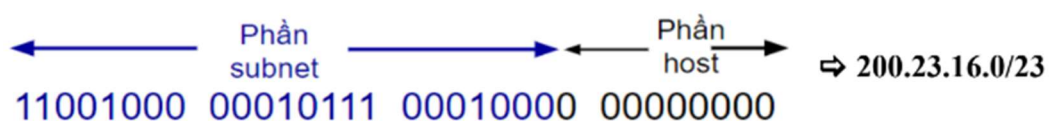
Lớp	Octet thứ 1 hệ thập phân	Network/Host (N = Network, H = Host)	Subnet Mask mặc định	Số mạng	Số mạng mỗi mạng
A	1 - 126	N.H.H.H	255.0.0.0	126 ($2^7 - 2$)	16,777,214 ($2^{24} - 2$)
B	128 - 191	N.N.H.H	255.255.0.0	16,382 ($2^{14} - 2$)	65,534 ($2^{16} - 2$)
C	192 - 223	N.N.N.H	255.255.255.0	2,097,150 ($2^{21} - 2$)	254 ($2^8 - 2$)
D	224 - 239	Multicast			
E	240 - 254	Nghiên cứu (Không sử dụng)			

4. Định địa chỉ IP: CIDR

- CIDR: Classless InterDomain Routing

+ **Phần subnet:** độ dài bất kỳ

+ **Định dạng địa chỉ:** a.b.c.d/x, trong đó x là số các bits trong phần subnet của địa chỉ.





5. Phân loại địa chỉ IP:

a. Phân loại theo **phạm vi hoạt động**:

- **Private IP**: sử dụng trong mạng LAN, có thể sử dụng lặp lại ở các mạng LAN khác nhau.

+ Lớp A: Từ 10.0.0.0 → 10.255.255.255

+ Lớp B: Từ 172.16.0.0 → 172.31.255.255

+ Lớp C: Từ 192.168.0.0 → 192.168.255.255

- **Public IP**: sử dụng trong mạng WAN (A, B, C), dùng để định tuyến trên Internet, và là duy nhất cho mỗi host tham gia vào Internet

- **Loopback IP**: Là các địa chỉ được chọn cho localhost.

Ví dụ: Dải địa chỉ 127.0.0.1 → 127.255.255.254

b. Phân loại trong **quá trình truyền thông**:

- Địa chỉ mạng (Network): Tất cả bit HostID = 0

- Địa chỉ quảng bá (Broadcast): Tất cả bit HostID = 1

- Địa chỉ dùng cho host: Trường hợp còn lại

+ Default Gateway thường là địa chỉ đầu tiên của mạng.

Ví dụ: Với địa chỉ mạng 192.168.1.0/24 thì 192.168.1.1/24 là địa chỉ Default Gateway.

- ❖ Ngoài ra, 0.0.0.0 là địa chỉ đặc biệt. Không có định tuyến. Được gán cho trường src của DHCP khi chưa có máy đích. Địa này có nghĩa là “Không có địa chỉ nào cụ thể”. Tức là khi chưa biết IP thì thay vì để trống thì máy sẽ đặt 0.0.0.0 vào đấy

6. Quy tắc đặt địa chỉ IP

- Các **bit** phần mạng **không** được phép **đồng thời bằng 0**.

VD: địa chỉ 0.0.0.1/24 với phần mạng là 0.0.0 (24 bit 0) là không hợp lệ.

- Nếu các bit **host đồng thời bằng 0**, ta có một **địa chỉ mạng**. **Địa chỉ mạng không** được dùng gán cho host.



Ví dụ:

+ Địa chỉ 192.168.1.0/24 là địa chỉ mạng, vì 8 bit host đồng thời bằng 0.

+ **Hỏi:** Địa chỉ 172.16.1.0/16 có phải địa chỉ mạng không? Nếu không, thì đâu là địa chỉ mạng của nó? \Rightarrow Không là địa chỉ mạng, địa chỉ mạng của nó là 172.16.0.0/16.

- Nếu các bit **host đồng thời bằng 1**, ta có một **địa chỉ quảng bá (broadcast)**. **Địa chỉ quảng bá** cũng **không** được gán cho host.

Ví dụ:

+ Địa chỉ 192.168.1.255/24 (8 bit host đều bằng 1) là một địa chỉ broadcast (quảng bá) cho mạng 192.168.1.0/24.

+ **Hỏi:** Địa chỉ 172.16.1.255/16 có phải địa chỉ broadcast không? Nếu không, thì đâu là địa chỉ broadcast của nó? \Rightarrow Không là địa chỉ broadcast, địa chỉ broadcast của nó là 172.16.255.255/16.

7. Chia mạng con trong IPv4

- Mượn thêm một số bit bên phần host để làm phần mạng.

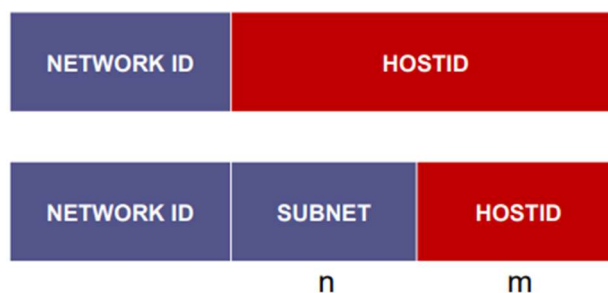
- Các bit mượn được gọi là các bit subnet.

- Gọi n là số bit mượn và m là số bit host còn lại

+ Số subnet có thể chia được: 2^n

+ Số host có trên mỗi subnet: $2^m - 2$ (*)

+ $n + m =$ số bit phần host của mạng ban đầu.





VD: Cho mạng ban đầu là 192.168.1.0/24. Nếu chia mạng này ra 4 mạng, hãy cho biết thông tin của mỗi mạng.

- Xác định số bit mượn **n**: Số subnet $2^n = 4 \Leftrightarrow n = 2$.
- Xác định số bit **m** còn lại và **số host** trong mỗi mạng con
 - Số bit host ban đầu: $32 - 24 = 8$
 - Số bit m: $8 - n = 8 - 2 = 6$
 - Số host/mỗi mạng con: $2^m - 2 \Rightarrow 2^6 - 2 = 62$
- Các mạng được chia là: 4 mạng, mỗi mạng 64 địa chỉ (62 địa chỉ có thể gán cho Host)

STT	Địa chỉ mạng	Địa chỉ host đầu	Địa chỉ host cuối	Địa chỉ broadcast
1	192.168.1.0/26	192.168.1.1	192.168.1.62	192.168.1.63
2	192.168.1.64/26	192.168.1.65	192.168.1.126	192.168.1.127
3	192.168.1.128/26	192.168.1.129	192.168.1.190	192.168.1.191
4	192.168.1.192/26	192.168.1.193	192.168.1.254	192.168.1.255

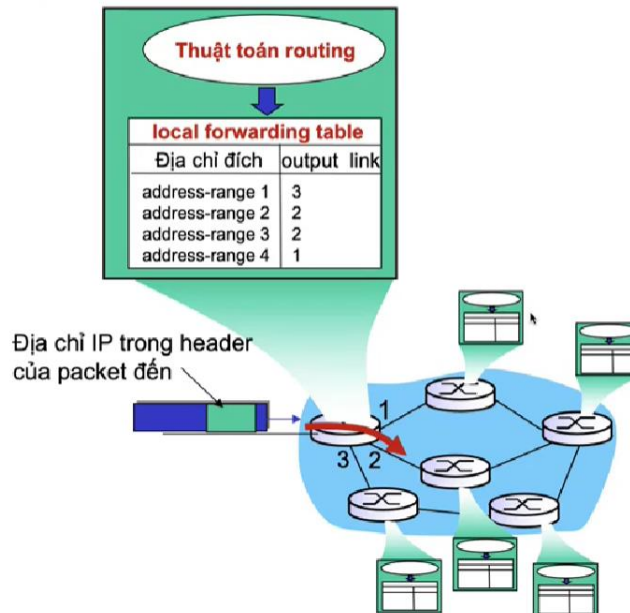


NHẬP MÔN MẠNG MÁY TÍNH

Chương 4: Tầng Mạng

(Tiếp Theo)

* Tác động lẫn nhau giữa routing và forwarding



Routing: mang tính global (toàn cục), xác định được đường đi có “chi phí” thấp nhất, hay còn gọi là tốt nhất cho gói tin từ đầu cuối này đến đầu cuối khác trong Mạng. Sử dụng thuật toán routing ta thu được output là bảng Forwarding (Forwarding Table), dựa vào bảng này ta có thể biết được Gói tin có địa chỉ bao nhiêu sẽ đi ra ở đâu ra nào.

Forwarding: sau khi gói tin đi đến interface của router thì việc đưa nó đến interface output phù hợp sẽ dựa vào forwarding table.

4.5. Các Thuật Toán Routing

Phân loại thuật toán Routing

- Nút (node) là router
- Neighbor là nút láng giềng, kết nối trực tiếp với router, 1 router có thể có nhiều neighbors.

Toàn cục hay Phân cấp?

- Toàn cục (global):

+ Tất cả các router trong mạng thu thập toàn bộ thông tin kết nối của toàn bộ mạng.

+ Sử dụng giải thuật tìm đường đi trên đồ thị (Dijkstra) \Rightarrow Thuật toán Link State.

+ Phân phối bảng định tuyến từ trung tâm tới các routers khác trong mạng.

- Phân cấp (Decentralized):

+ Mỗi router tự xây dựng bảng chọn đường riêng dựa trên kết nối với các nút láng giềng với nó \rightarrow lặp đi lặp lại quy trình tính toán và trao đổi thông tin với neighbor \rightarrow dần tìm ra con đường có chi phí ngắn nhất tới đích.

+ Sử dụng giải thuật định tuyến: Distance vector.

+ Được sử dụng phổ biến hơn trong thực tế.

4.5.1. Link State

\rightarrow Sử dụng thuật toán Dijkstra

- Các nút biết các thông tin của tất cả các nút trong mạng bằng cách quảng bá các gói tin trạng thái (link state broadcast) chứa các thông tin định danh, **chi phí của các liên kết với neighbor của nó.**

- Kết quả của việc này là tất cả các nút đều có cái nhìn giống nhau và toàn diện về cấu trúc mạng. Mỗi nút sau đó có thể tự chạy giải thuật **Link State** và tính toán đường đi ngắn nhất giống như các nút khác.

- **Giải thuật Dijkstra** tính toán con đường ngắn nhất từ 1 nút nguồn đến tất cả các nút khác trong mạng bằng việc lặp đi lặp lại k lần, sẽ biết được đường đi có chi phí thấp nhất của k đích.

- Các ký hiệu:

+ **c(x, y)**: chi phí kết nối từ nút x đến nút y, $= \infty$ nếu x không trực tiếp kết nối với y.

+ **D(v)**: chi phí của đường đi tốt nhất từ nút nguồn tới đích v sau khi thực hiện giải thuật.

+ **p(v)**: nút trước của v trên đường đi ngắn nhất từ nguồn tới đích.

+ **N'**: tập các nút mà đường đi ngắn nhất đã được biết.

- *Giải thuật Link State cho nút nguồn u:*

+ Bước khởi tạo ban đầu.

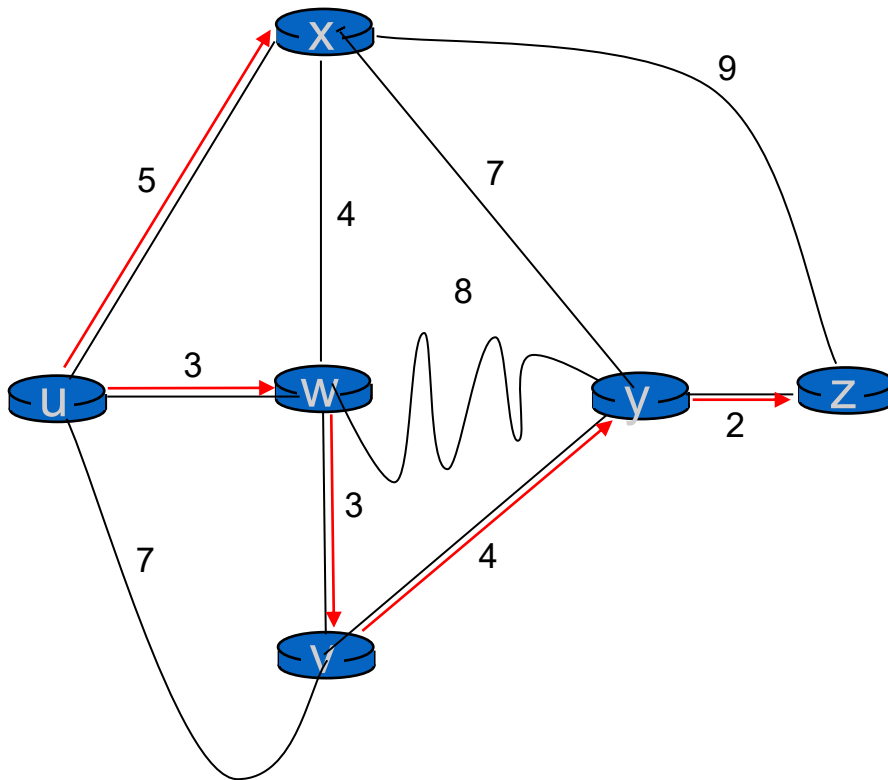
+ Sau đó là 1 vòng lặp. Số lần lặp bằng số nút trong mạng



→ Kết quả: các con đường ngắn nhất từ u đến mỗi nút trong mạng.

- 1 Khởi tạo
- 2 $N' = \{u\}$
- 3 Với nút v bất kì
- 4 Nếu v là láng giềng của u
- 5 Thì $D(v) = c(u, v)$
- 6 Ngược lại $D(v) = \infty$
- 7
- 8 Lặp
- 9 Tìm w không trong N' mà $D(w)$ có giá trị nhỏ nhất đến nút trong N'
- 10 Cập nhật $D(v)$ cho mỗi láng giềng v của w mà không nằm trong N' :
- 11 $D(v) = \min(D(v), D(w) + c(w, v))$
- 12 Đến khi hết tất cả các nút trong N'

Ví dụ:



Bước	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
Khởi tạo	u	7, u	3, u	5, u	∞	∞
1	uw	6, w		5, u	11, w	∞
2	uwx	6, w			11, w	14, x
3	uwxv				10, v	14, x
4	uwxvy					12, y
5	uwxvzy					

Giải thích: - Trong bước khởi tạo, các chi phí đường đi ngắn nhất từ nút nguồn u tới các láng giềng liên kết trực tiếp với nó (v, w, x) lần lượt là (7; 3; 5). Các chi phí tới (y, z) được thiết lập là ∞ vì nó không kết nối trực tiếp tới u.

- Ở vòng lặp đầu tiên, trong số các nút v, w, x, y, z, chưa có nút nào được thêm vào N'; và thấy rằng nút có đường đi với chi phí nhỏ nhất là w với $D(w) = 3$ (kết quả của vòng lặp tìm chi phí trước), vì vậy w được thêm vào N'.

- Sau đó, ta sẽ cập nhật D cho tất cả các nút không nằm trong N' còn lại giống như dòng 13 trong giải thuật phía trên: Chi phí tới v (ban đầu là 7, trước nút đó là u) được cập nhật lại thành 6 thông qua nút trước đó được tìm thấy với chi phí thấp hơn là w. Chi phí tới nút x thì không thay đổi. Tương tự như v, x đối với lần lượt y và z, chi phí đến y (thông qua w) được cập nhật lại thành 11, và z giữ nguyên ∞ .

- Ở vòng lặp thứ 2, nút x được tìm thấy là có chi phí đường đi nhỏ nhất (5), nên ta đưa x vào N'. Tập N' lúc này gồm (u, w, x). Chi phí của các nút còn lại chưa có trong N' (v, y, z) được cập nhật qua dòng 13 trong giải thuật ở trên, kết quả được chỉ ra trong dòng bước 2 ở bảng tính toán trên.

- Tiếp tục cho đến khi N' tập hợp tất cả các nút.

→ Khi thuật toán LS kết thúc, chúng ta biết được nút trước của mỗi nút trên con đường đi ngắn nhất từ nút nguồn. Với mỗi nút trước, ta lại

có nút trước của nó, vì vậy chúng ta có thể xây dựng một đường đi hoàn chỉnh từ nguồn cho tới mọi đích.

- **Độ phức tạp** khi tính toán với n nút là **$O(n^2)$** với $n(n+1)/2$ phép so sánh.

- Có một vấn đề nảy sinh, đó là sự dao động của đường đi tốt nhất, khi chi phí dựa trên lưu lượng của đường đi có thể thay đổi khiến cho giải thuật Link State tạo ra kết quả khác. Sự dao động này có thể xảy ra ở bất kỳ giải thuật nào.

+ Một giải pháp đề ra đó là các chi phí liên kết không dựa vào lưu lượng mạng, nhưng đây là một giải pháp không thể chấp nhận được khi mục tiêu của định tuyến là tránh sự tắc nghẽn ở các liên kết.

+ Một giải pháp khác hợp lý hơn đó là đảm bảo các bộ định tuyến không cùng chạy thuật toán Link State trong cùng một thời điểm. Tuy nhiên, nghiên cứu cho thấy các bộ định tuyến có thể tự đồng bộ hóa với nhau → Một cách xử lý đó là cho các bộ định tuyến ngẫu nhiên thời gian nó gửi ra thông tin quảng bá đường liên kết.

4.5.2. Distance Vector

Ý tưởng:

- Mỗi nút định kỳ nhận ước lượng distance vector từ một hoặc **nhiều neighbor nối trực tiếp với nó.**
- Sau khi nhận DV (Distance Vector) mới, nó cập nhật lại DV cũ dùng công thức Bellman-Ford:

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

Với:

- o $d_x(y)$: Chi phí đường đi nhỏ nhất từ x đến y .
 - o $c(x, v)$: Chi phí từ x đến v (lân cận của x).
 - o $d_v(y)$: Chi phí đường đi nhỏ nhất từ v đến y (lân cận của v)
- Chỉ khi DV thay đổi, nút mới thông báo đến các nút láng giềng.
 - Các nút lặp đi lặp lại quá trình này cho đến khi không còn thông tin trao đổi giữa các láng giềng.
- Giải thuật là thuật toán lặp đi lặp lại, bất đồng bộ, và phân tán



Giải thuật:

- Mỗi nút bắt đầu với $D = [D_x(y): y \in N]$ là vector của ước lượng chi phí từ x tới các nút y trong N . Với giải thuật DV, mỗi nút x duy trì các thông tin định tuyến sau:
 - o Đối với mỗi láng giềng x , chi phí $c(x, v)$ từ x với láng giềng nối trực tiếp, v .
 - o $D_x = [D_x(y): y \in N]$, là vector khoảng cách của nút x chứa chi phí ước lượng của x tới tất cả các đích y trong N .
 - o $D = [D_v(y): y \in N]$, là vector khoảng cách của mỗi láng giềng v thuộc N của x .
- Từ thời điểm này tới thời điểm khác, mỗi nút gửi một bảng các vector khoảng cách tới các láng giềng của nó. Khi một nút x nhận vector khoảng cách từ bất kì láng giềng v nào của nó, nó cập nhật lại vector khoảng cách của nó bằng công thức Bellman-Ford như sau:
$$D_x(y) = \min_v \{c(x, v) + D_v(y)\}$$
- Nếu như sau khi cập nhật mà vector khoảng cách của x thay đổi thì nút x , sau đó, sẽ gửi bảng vector khoảng cách vừa cập nhật của nó cho các láng giềng của nó. Các láng giềng này cũng sẽ cập nhật các vector khoảng cách của nó.



NHẬP MÔN MẠNG MÁY TÍNH

Chương 4: Tầng Network (tiếp theo)



VII. Định tuyến trên Internet

Định tuyến Intra – AS: còn gọi là IGP (Interior Gateway Protocol). Có các giao thức phổ biến:

- + RIP: Routing Information Protocol (công bố năm 1982 trong BSD – UNIX).
- + OSPF: Open Shortest Path First (công khai cho mọi đối tượng).
- + IGRP: Interior Gateway Routing Protocol. (Độc quyền của Cisco)

1. Định tuyến nội vùng RIP: Routing Information Protocol (công bố năm 1982 trong BSD – UNIX).

- Là giao thức vector khoảng cách (**Distance vector**):

+ Tính chi phí bằng số lượng hop (**hop count**) = số lượng các mạng con đi qua trên đường ngắn nhất từ bộ định tuyến nguồn đến bộ định tuyến đích, bao gồm mạng con đích.

+ **Maximum hop count = 15 hop**

+ Neighbors trao đổi bảng định tuyến mỗi **30s** trong thông điệp phản hồi (còn gọi là advertisement)

* Các router gửi thông điệp RIP yêu cầu và RIP phản hồi cho nhau sử dụng **UDP** và cổng **520**. Chúng còn được gọi là thông điệp quảng bá RIP (RIP advertisements)

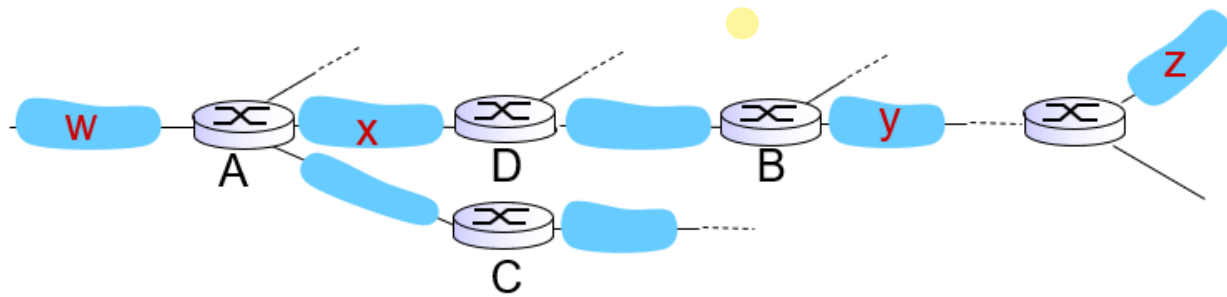
* Bảng định tuyến của router bao gồm cả vector khoảng cách của router và bảng chuyển tiếp của router.

+ Mỗi advertisement: liệt kê lên đến **25** subnet đích

+ Hủy đường kết nối với neighbors không tương tác sau **180s**.

=> Thông tin về lỗi đường kết nối nhanh chóng lan truyền trên toàn mạng

+ poison reverse được dùng để ngăn chặn vòng lặp ping-pong (khoảng cách vô hạn = 16 hops)



Bảng định tuyến trong router D

Subnet đích	router kế tiếp	số lượng hop đến đích
W	A	2
y	B	2
Z	B	7
X	--	1
....

2. Định tuyến nội vùng OSPF: Open Shortest Path First (công khai cho mọi đối tượng).

- Là giao thức định tuyến theo trạng thái đường liên kết (**Link State**) sử dụng giải thuật tìm đường **Dijkstra**:

+ OSPF phát tán advertisements đến toàn bộ vùng mạng tự trị (AS) định kỳ mỗi **30s** dù trạng thái mạng không thay đổi.

+ Xây dựng bản đồ đường đi tới các mạng từ các node.

+ Thông điệp OSPF được đóng gói trong **gói tin IP, cổng 89**.

* **Ích lợi của OSPF (mà RIP không có):**

• Tránh tắc nghẽn: Cho phép sử dụng nhiều đường để chuyển gói tin nếu chúng có chi phí bằng nhau. (RIP chỉ cho 1)



- Bảo mật: Cơ chế xác thực đơn giản và MD5 (sử dụng hàm băm) giúp các OSPF messages được chứng thực, tránh bị tấn công.
- Hỗ trợ tích hợp Unicast và Multicast.
- Khả năng xây dựng mô hình phân cấp tự trị.

Bảng so sánh 2 giao thức định tuyến

	RIP	OSPF
Giải thuật định tuyến	Distance vector	Link state
Chu kỳ hoạt động	30 giây	30 giây
Tương tác với các node lân cận	Trao đổi bảng định tuyến	Phát tán advertisements
Hủy kết nối	Sau 180s im lặng	Tính toán lại đường đi định kì
Phương tiện	UDP, port 520	IP, port 89
Nội dung lưu trữ ở router	Bảng định tuyến (bảng chuyển tiếp và distance vector)	Bản đồ đường đi