



Tầng mạng (Network Layer)

Một số hạn chế của tầng liên kết dữ liệu

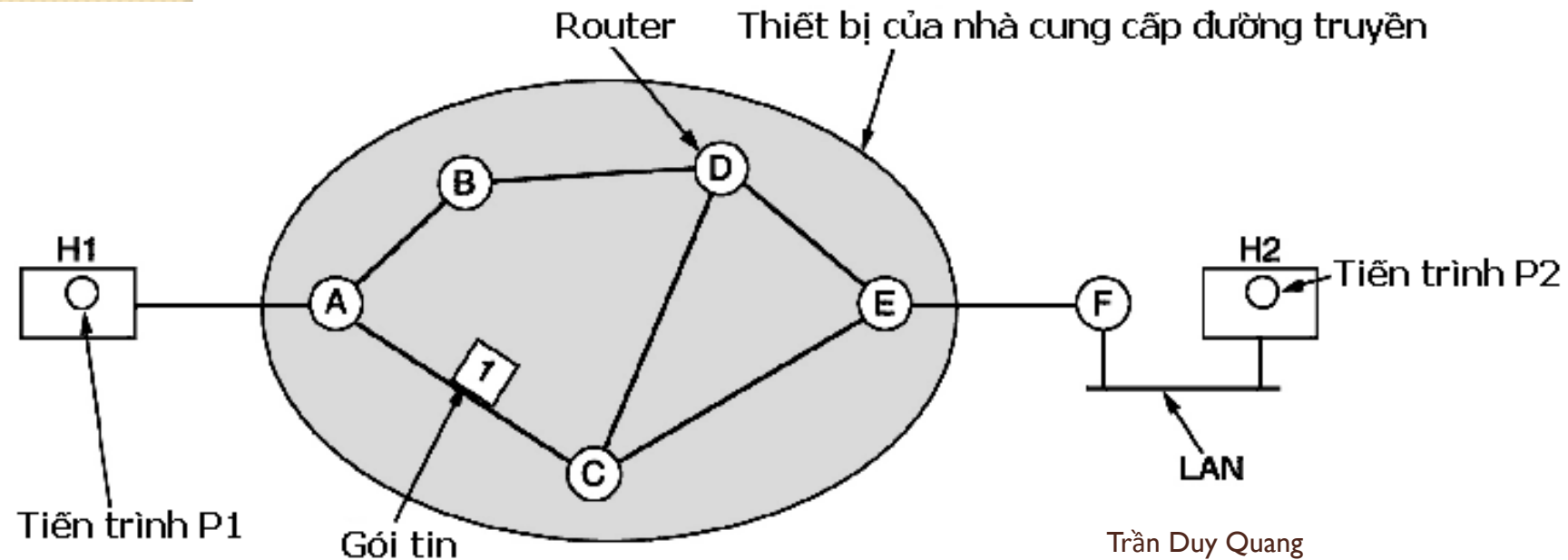
- Chỉ đảm bảo truyền tải thông tin giữa các máy tính có đường kết nối trực tiếp (truyền từ đầu dây dẫn này đến đầu dây dẫn kia)
- Bị giới hạn về số lượng máy tính và kích thước mạng
- Khó khăn trong việc nối kết các mạng sử dụng kỹ thuật chia sẻ đường truyền khác nhau – mạng không đồng nhất

Vai trò của tầng mạng

- Cung cấp cho người dùng một dịch vụ nối kết host-host trên một hệ thống mạng diện rộng, không đồng nhất một cách dễ dàng
- Đưa các gói tin từ máy gửi qua các chặn đường để đến được máy nhận
- Chọn đường đi cho gói tin để tránh tắc nghẽn

Các vấn đề liên quan đến việc thiết kế tầng mạng

- Kỹ thuật hoán chuyển lưu và chuyển tiếp (Store-and-Forward Switching)



Các vấn đề liên quan đến việc thiết kế tầng mạng

- Các dịch vụ cung cấp cho tầng vận chuyển
 - Mục tiêu thiết kế: Các dịch vụ cần độc lập với kỹ thuật của các router.
 - Tầng vận chuyển cần được độc lập với số lượng, kiểu và hình trạng của các router hiện hành.

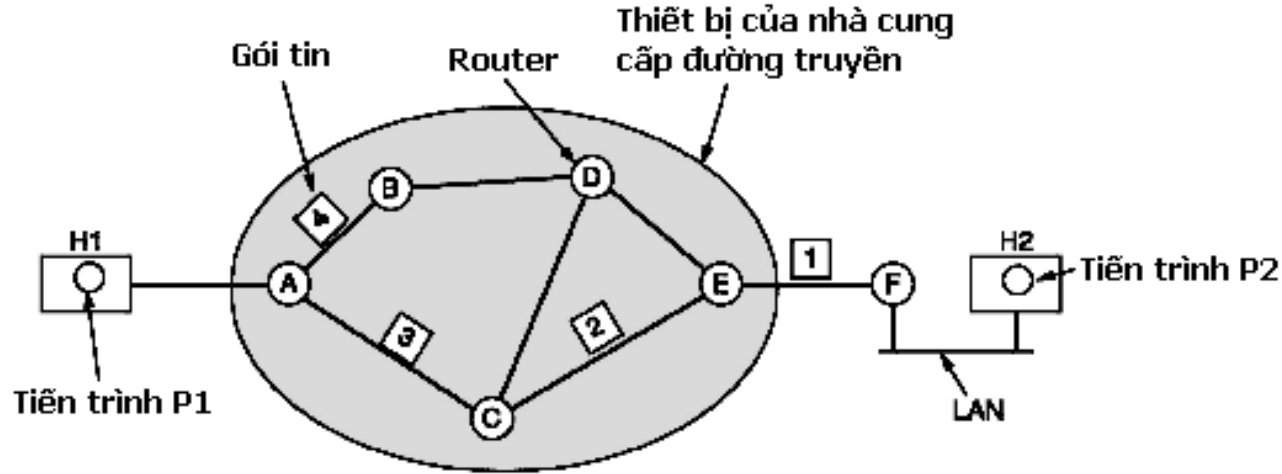
Các vấn đề liên quan đến việc thiết kế tầng mạng

- Địa chỉ mạng cung cấp cho tầng vận chuyển phải có sơ đồ đánh số nhất quán cho dù chúng là LAN hay WAN
- Hai dịch vụ cơ bản:
 - Dịch vụ không nối kết (Connectionless Service)
 - Dịch vụ hướng nối kết (Connection – Oriented Service)

Dịch vụ không nối kết

- Các gói tin được đưa vào subnet một cách riêng lẻ và được chọn đường một cách độc lập nhau.
- Không cần thiết phải thiết lập nối kết trước khi truyền tin.
- Các gói tin được gọi là thư tín (Datagram) và subnet được gọi là Datagram Subnet.

Cài đặt dịch vụ không nối kết (Implementation of Connectionless Service)



Bảng vạch đường của nút A
lúc đầu lúc sau

A	-	A	-
B	B	B	B
C	C	C	C
D	B	D	B
E	C	E	B
F	C	F	B

Nút C

A	A
B	A
C	-
D	D
E	E
F	E

Nút E

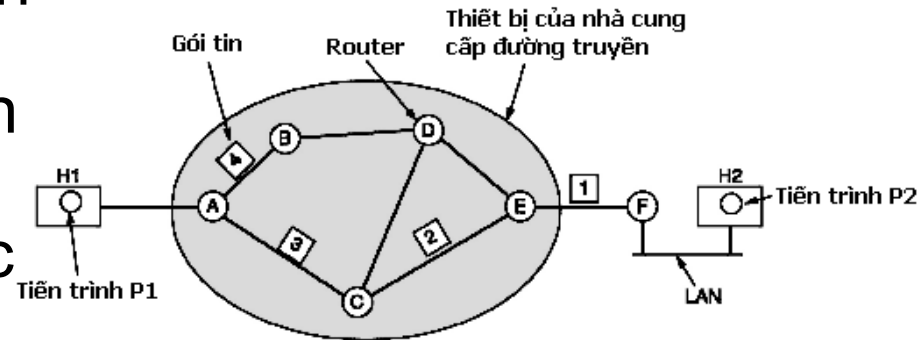
A	C
B	D
C	C
D	D
E	-
F	F

Đích đến

Bước kế tiếp

Cài đặt dịch vụ không nối kết

- Giả sử rằng tiến trình P1 có nhiều thông điệp cần gửi cho tiến trình P2.
- Khi đó P1 sẽ gửi các thông điệp này cho tầng vận chuyển và yêu cầu tầng vận chuyển truyền sang tiến trình P2 trên máy tính H2.



Bảng vạch đường của nút A lúc đầu lúc sau

	lúc đầu	lúc sau
A	-	-
B	B	B
C	C	C
D	B	B
E	C	B
F	C	B

Nút C
A A
B A
C -
D D
E E
F E

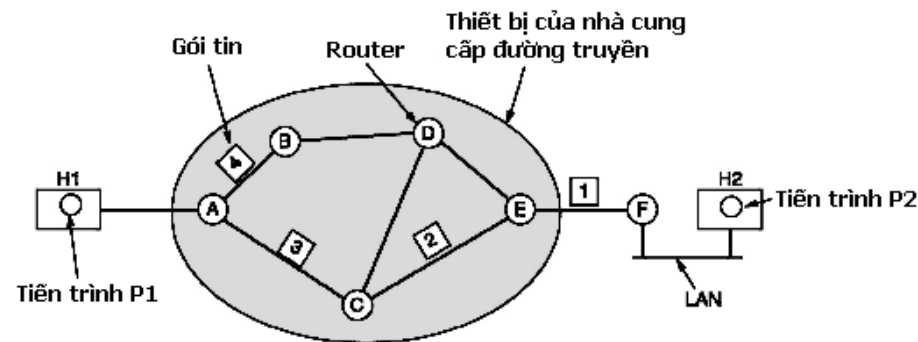
Nút E
A C
B D
C C
D D
E -
F F

Cài đặt dịch vụ không nối kết

- Tầng vận chuyển sẽ gắn thêm tiêu đề (header) của nó vào thông điệp và chuyển các thông điệp xuống tầng mạng.
- Giả sử tầng mạng chia thông điệp ra thành 4 gói tin 1,2,3 và 4, và lần lượt gửi từng gói một đến router A bằng một giao thức điểm nối điểm như PPP chẳng hạn.

Cài đặt dịch vụ không nối kết

- Mỗi router có một bảng chọn đường (Routing Table) chỉ ra nơi nào có thể gửi các gói tin để có thể đến được những đích đến khác nhau trên mạng.



Bảng vạch đường của nút A
lúc đầu lúc sau

	lúc đầu	lúc sau
A	-	-
B	B	B
C	C	C
D	B	B
E	C	B
F	C	B

Nút C

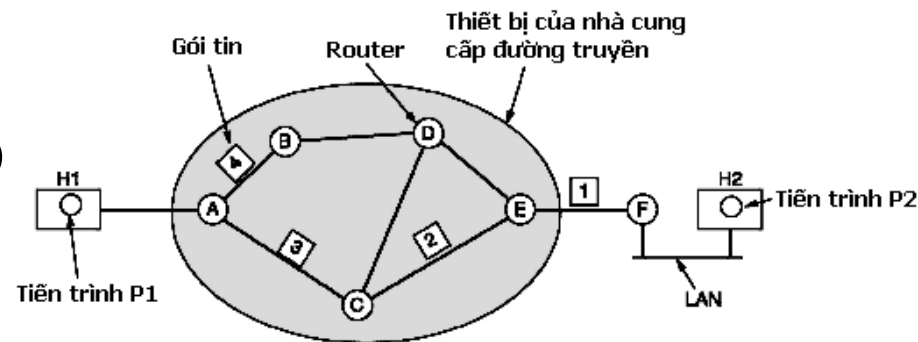
A	A
B	A
C	-
D	D
E	E
F	E

Nút E

A	C
B	D
C	C
D	D
E	-
F	F

Cài đặt dịch vụ không nối kết

- Mỗi mục của bảng chứa 2 thông tin quan trọng nhất đó là Đích đến (Destination) và ngõ ra kế tiếp (Next Hop) là nơi cần chuyển gói tin đến để gói tin đến được đích đến.



Bảng vạch đường của nút A
lúc đầu lúc sau

	lúc đầu	lúc sau
A	-	-
B	B	B
C	C	C
D	B	B
E	C	B
F	C	B

Nút C

A	A
B	A
C	-
D	D
E	E
F	E

Nút E

A	C
B	D
C	C
D	D
E	-
F	F

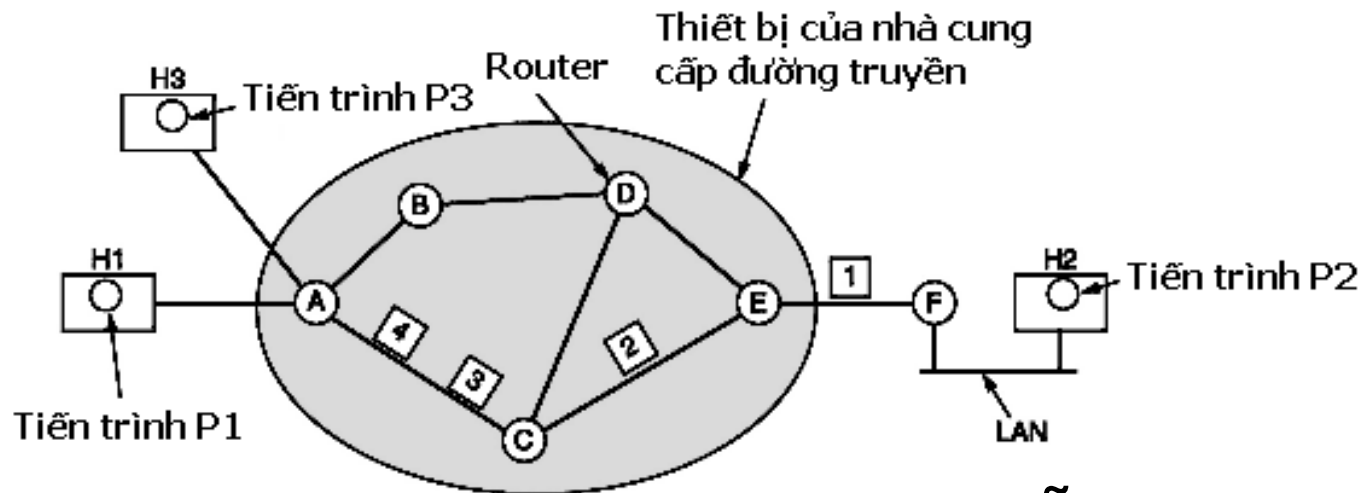
Dịch vụ định hướng nối kết

- Một đường nối kết giữa bên gửi và bên nhận phải được thiết lập trước khi các gói tin có thể được gửi đi.
- Nối kết này được gọi là mạch ảo (Virtual Circuit (**VC**)) tương tự như mạch vật lý được nối kết trong hệ thống điện thoại và subnet trong trường hợp này được gọi là virtual circuit subnet.

Dịch vụ định hướng nối kết

- Mục đích của việc sử dụng mạch ảo là để tránh việc chọn lại đường đi mới cho mỗi gói tin có cùng đích đến.

Cài đặt dịch vụ có nối kết



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2

Vào Ra

Bảng vạch đường của C

A	1	E	1
A	2	E	2

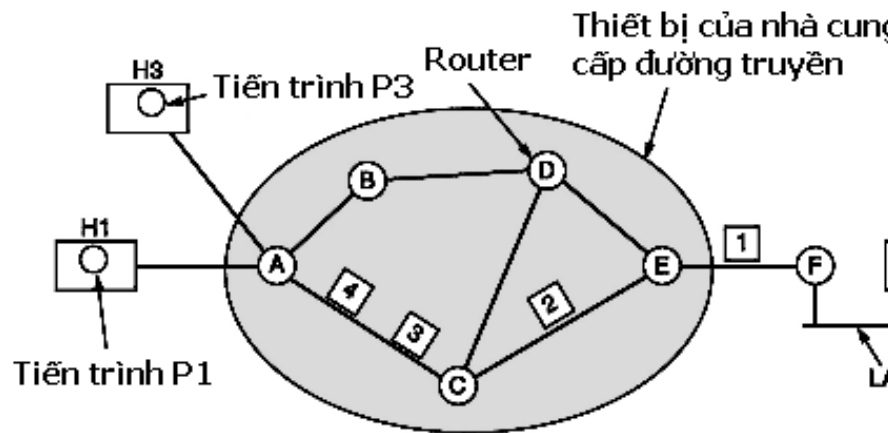
Bảng vạch đường của E

C	1	F	1
C	2	F	2

Mỗi gói tin có mang một số nhận dạng để xác định mạch ảo mà nó thuộc về

Cài đặt dịch vụ có nối kết

- Máy tính H1 thực hiện một nối kết với máy tính H2 qua nối kết số 1. Nối kết này được ghi nhận trong mục từ đầu tiên trong bảng chọn đường của các router.



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2

Vào Ra

Bảng vạch đường của C

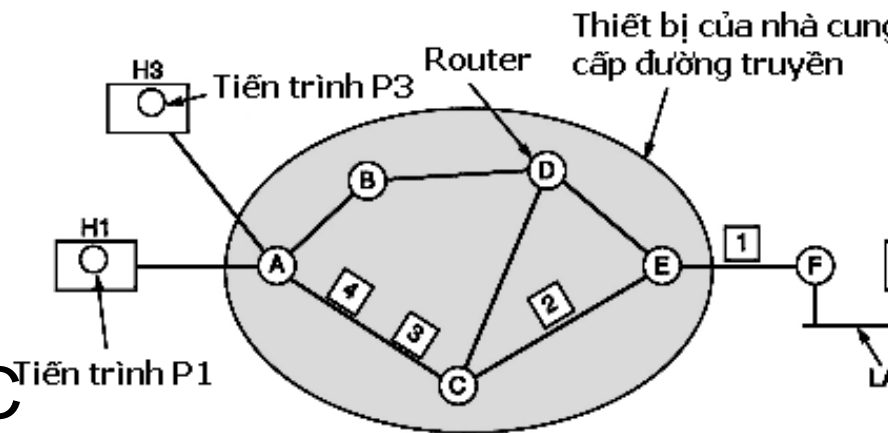
A	1	E	1
A	2	E	2

Bảng vạch đường của E

C	1	F	1
C	2	F	2

Cài đặt dịch vụ có nối kết

- Dòng đầu tiên trong bảng chọn đường của router A nói rằng: những gói tin mạng số nhận dạng nối kết số 1 đến từ máy H1 phải được gửi sang router C với số nhận dạng nối kết là 1. Tương tự, cho các mục đầu tiên của router C và E.



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2

Vào Ra

Bảng vạch đường của C

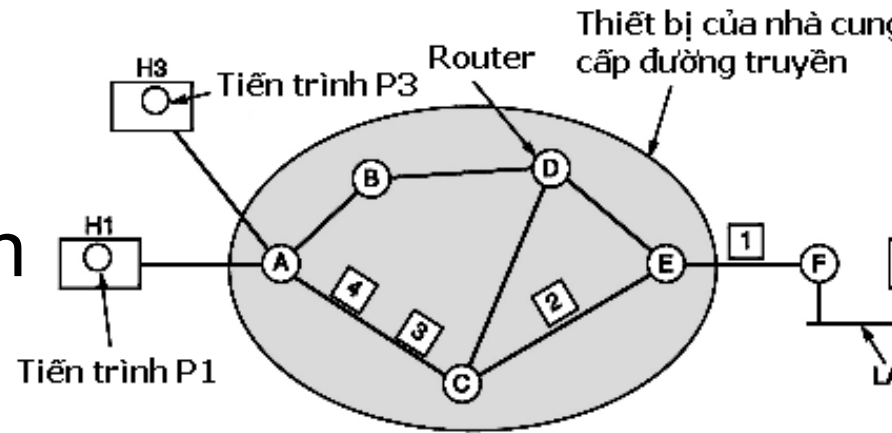
A	1	E	1
A	2	E	2

Bảng vạch đường của E

C	1	F	1
C	2	F	2

Cài đặt dịch vụ có nối kết

- Điều gì xảy ra nếu máy tính H3 muốn nối kết với máy tính H2. Nó chọn số nhận dạng nối kết là 1, vì đây là nối kết đầu tiên đối với H3, và yêu cầu subnet thiết lập mạch ảo.



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2

Vào Ra

Bảng vạch đường của C

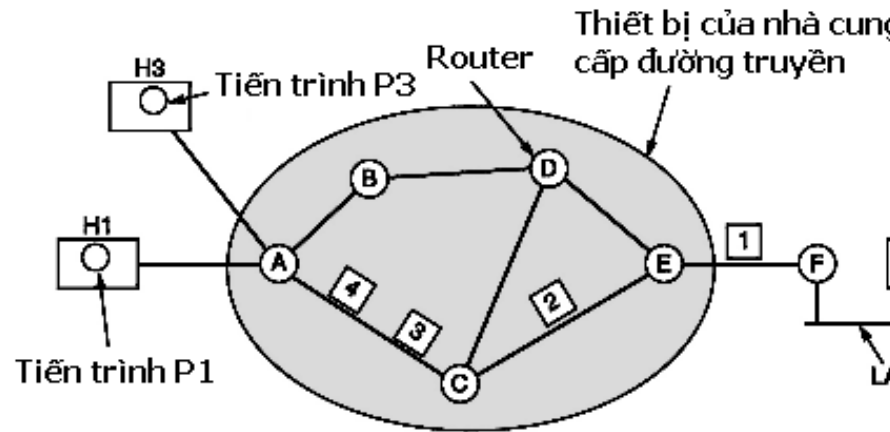
A	1	E	1
A	2	E	2

Bảng vạch đường của E

C	1	F	1
C	2	F	2

Cài đặt dịch vụ có nối kết

- Điều này đã làm cho các router phải thêm mục thứ hai vào bảng chọn đường.



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2

Vào Ra

Bảng vạch đường của C

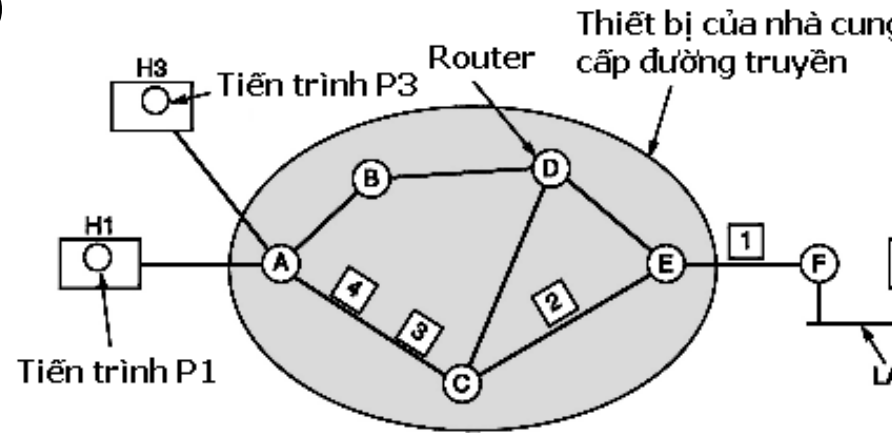
A	1	E	1
A	2	E	2

Bảng vạch đường của E

C	1	F	1
C	2	F	2

Cài đặt dịch vụ có nối kết

- Đối với router A, số nhận dạng nối kết với H3 là 1, trùng với nối kết với H1, không làm router A lẫn lộn vì A có thêm thông tin máy gọi là H1 hay H3.



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2
Vào		Ra	

Bảng vạch đường của C

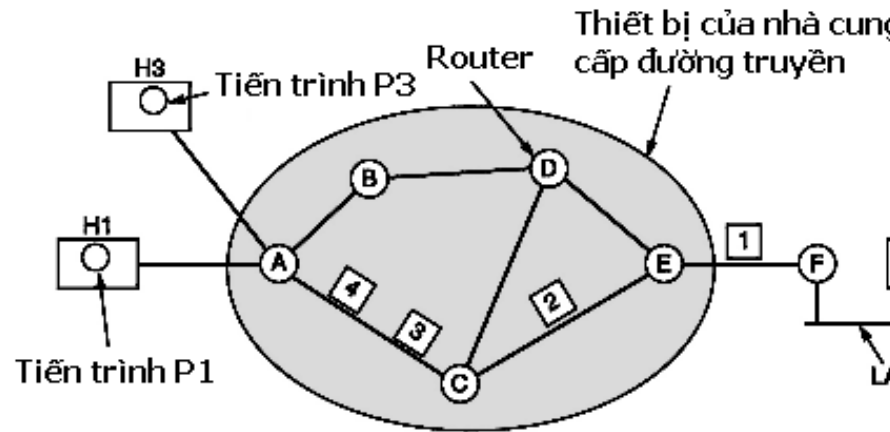
A	1	E	1
A	2	E	2

Bảng vạch đường của E

C	1	F	1
C	2	F	2

Cài đặt dịch vụ có nối kết

- Tuy nhiên, đối với các router C, E và F thì không thể phân biệt được nên A đã gán một số nhận dạng khác, là số 2, cho các gói tin gửi đến C có nguồn gốc từ H3.



Bảng vạch đường của A

H1	1	C	1
H3	1	C	2

Vào

Ra

Bảng vạch đường của C

A	1	E	1
A	2	E	2

Bảng vạch đường của E

C	1	F	1
C	2	F	2

So sánh giữa Datagram subnet và Virtual-Circuit subnet

Vấn đề	Datagram Subnet	Circuit Subnet
Thiết lập nối kết	Không cần	Cần thiết
Đánh địa chỉ	Mỗi gói tin chứa đầy đủ địa chỉ gửi và nhận	Mỗi gói tin chỉ chứa số nhận dạng nối kết có kích thước nhỏ
Thông tin trạng thái		

So sánh giữa Datagram subnet và Virtual-Circuit subnet

Vấn đề	Datagram Subnet	Circuit Subnet
Thông tin trạng thái	Router không cần phải lưu giữ thông tin trạng thái của các nối kết	Mỗi nối kết phải được lưu lại trong bảng chọn đường của router
Chọn đường		

So sánh giữa Datagram subnet và Virtual-Circuit subnet

Vấn đề	Datagram Subnet	Circuit Subnet
Chọn đường	Mỗi gói tin có đường đi khác nhau	Đường đi được chọn khi mạch ảo được thiết lập, sau đó tất cả các gói tin đều đi trên đường này

So sánh giữa Datagram subnet và Virtual-Circuit subnet

Vấn đề	Datagram Subnet	Circuit Subnet
Ảnh hưởng khi router bị hỏng	Không bị ảnh hưởng, ngoại trừ gói tin đang trên đường truyền bị hỏng	Tất cả các mạch ảo đi qua router hỏng đều bị kết thúc

So sánh giữa Datagram subnet và Virtual-Circuit subnet

Vấn đề	Datagram Subnet	Circuit Subnet
Chất lượng dịch vụ	Khó đảm bảo	Có thể thực hiện dễ dàng nếu có đủ tài nguyên gán trước cho từng mạch ảo (VC)

So sánh giữa Datagram subnet và Virtual-Circuit subnet

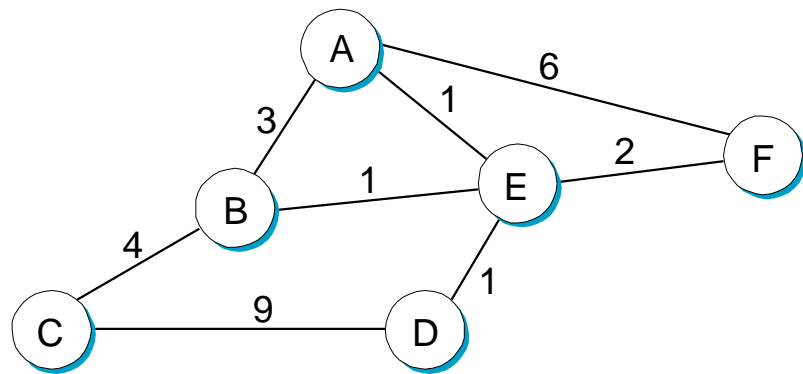
Vấn đề	Datagram Subnet	Circuit Subnet
Điều khiển tắc nghẽn	Khó điều khiển	Có thể thực hiện dễ dàng nếu có đủ tài nguyên gán trước cho từng mạch ảo (VC)



Giải thuật chọn đường

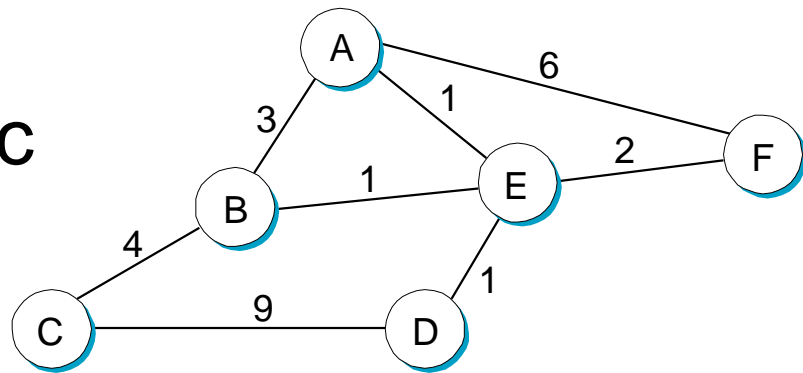
Chọn đường (Routing) [định tuyến]

- Mục tiêu: là xác định một đường đi tốt (chuỗi các router) trên mạng từ máy gửi đến máy nhận thông tin
- Cần đồ thị hóa hệ thống mạng cho các giải thuật chọn đường:



Chọn đường (Routing) [định tuyến]

- Nút là các host, switch, router hoặc là các mạng con.
- Cạnh của đồ thị tương ứng với các đường nối kết mạng.
- Mỗi cạnh có một chi phí đính kèm



Chọn đường (Routing)

- Chọn đường là tìm ra đường đi có chi phí thấp nhất giữa hai nút bất kỳ.
- Chi phí của đường đi là tổng chi phí khi đi qua tất cả các cạnh làm thành đường đi đó.
- Nếu không có một đường đi giữa hai nút; chi phí là vô cùng.

Mục tiêu của giải thuật chọn đường

- Xác định đường đi nhanh chóng, chính xác.
- Khả năng thích nghi được với những thay đổi về hình trạng mạng.
- Khả năng thích nghi được với những thay đổi về tải đường truyền.
- Khả năng tránh được các nút kết bị tắc nghẽn tạm thời.
- Chi phí tính toán để tìm ra được đường đi phải thấp.

Phân loại giải thuật chọn đường

- Chọn đường tập trung (Centralized routing): Trong mạng có một Trung tâm điều khiển mạng (Network Control Center) chịu trách nhiệm tính toán và cập nhật thông tin về đường đi đến tất cả các điểm khác nhau trên toàn mạng cho tất cả các router.
- Chọn đường phân tán (Distributed routing): Mỗi router phải tự tính toán tìm kiếm thông tin về các đường đi đến những điểm khác nhau trên mạng. Để làm được điều này, các router cần phải trao đổi thông tin qua lại với nhau.

Phân loại giải thuật chọn đường

- Chọn đường tĩnh (Static routing): Các router không tự động cập nhật thông tin về đường đi khi hình trạng mạng thay đổi. Thông thường nhà quản trị mạng sẽ là người cập nhật thông tin về đường đi cho router.
- Chọn đường động (Dynamic routing): Các router sẽ tự động cập nhật lại thông tin về đường đi khi hình trạng mạng bị thay đổi.

Đường đi tối ưu

- Đường đi tối ưu từ A đến B là đường đi “ngắn” nhất trong số các đường đi có thể. Tuy nhiên khái niệm “ngắn” tùy thuộc vào đơn vị dùng để đo chiều dài đường đi.
- Đối với các router, các đại lượng sau có thể được sử dụng để đo độ dài đường đi:
 - Số lượng các router trung gian phải đi qua (HOP)
 - Độ trì hoãn trung bình của các gói tin
 - Chi phí truyền tin

Giải thuật tìm đường đi ngắn nhất

Dijkstra

- Mục đích là để tìm đường đi ngắn nhất từ một nút cho trước trên đồ thị đến các nút còn lại trên mạng
- Thuộc loại giải thuật tìm đường đi tối ưu tập trung
- *Giao thức OSPF sử dụng giải thuật này*

Giải thuật tìm đường đi ngắn nhất

Dijkstra

- Gọi:
 - S là nút nguồn cho trước.
 - N: là tập hợp tất cả các nút đã xác định được đường đi ngắn nhất từ S.
 - D_i : là độ dài đường đi ngắn nhất từ nút nguồn S đến nút i.
 - L_{ij} : là giá của cạnh nối trực tiếp nút i với nút j, sẽ là ∞ nếu không có cạnh nối trực tiếp giữa i và j.
 - P_j là nút cha của nút j.

Giải thuật tìm đường đi ngắn nhất

Dijkstra

- Bước 1: Khởi tạo
 - $N = \{S\}; D_S = 0;$
 - Với $\forall i \neq S: D_i = L_{Si}, P_i = S$
- Bước 2: Tìm nút gần nhất kế tiếp
 - Tìm nút $i \notin N$ thoả $D_i = \min (D_j)$ với $j \notin N$
 - Thêm nút i vào N .
 - Nếu N chứa tất cả các nút của đồ thị thì dừng. Ngược lại sang Bước 3

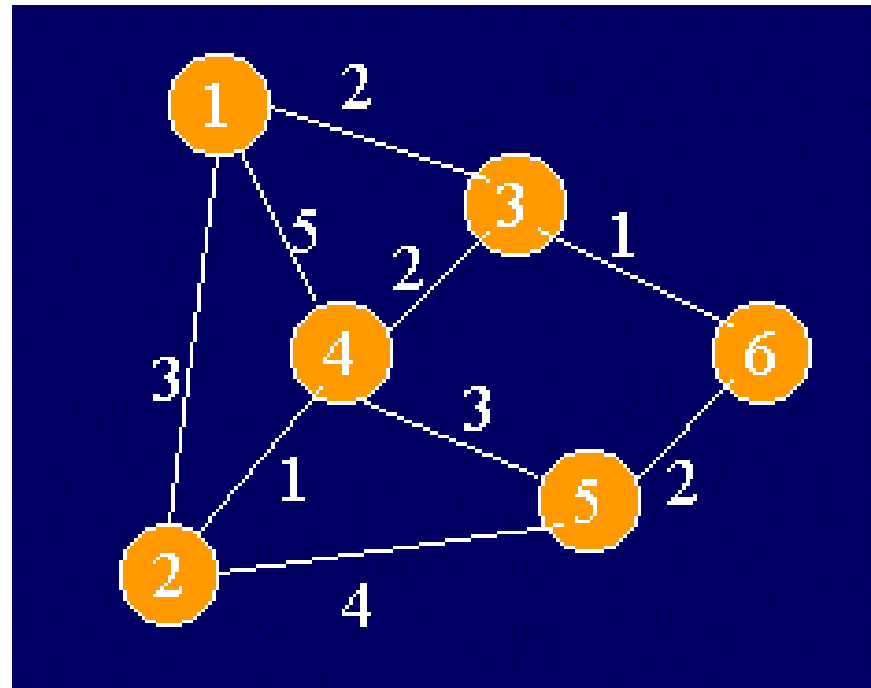
Giải thuật tìm đường đi ngắn nhất Dijkstra

- Bước 3: Tính lại giá đường đi nhỏ nhất
 - ❖ Với mỗi nút $j \notin N$: Tính lại $D_j = \min\{D_j, D_i + L_{ij}\}$; $P_j = i$;
 - ❖ Trở lại Bước 2

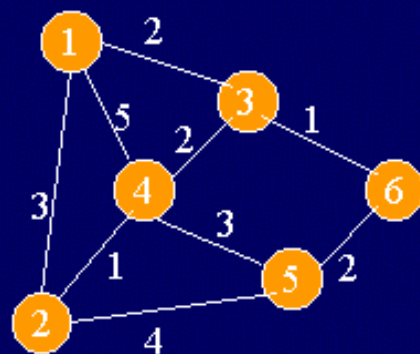
Giải thuật tìm đường đi ngắn nhất

Dijkstra

- Ví dụ: Cho mạng có hình trạng như đồ thị: Tìm đường đi ngắn nhất từ nút 1 đến các nút còn lại.
- Áp dụng giải thuật ta có:



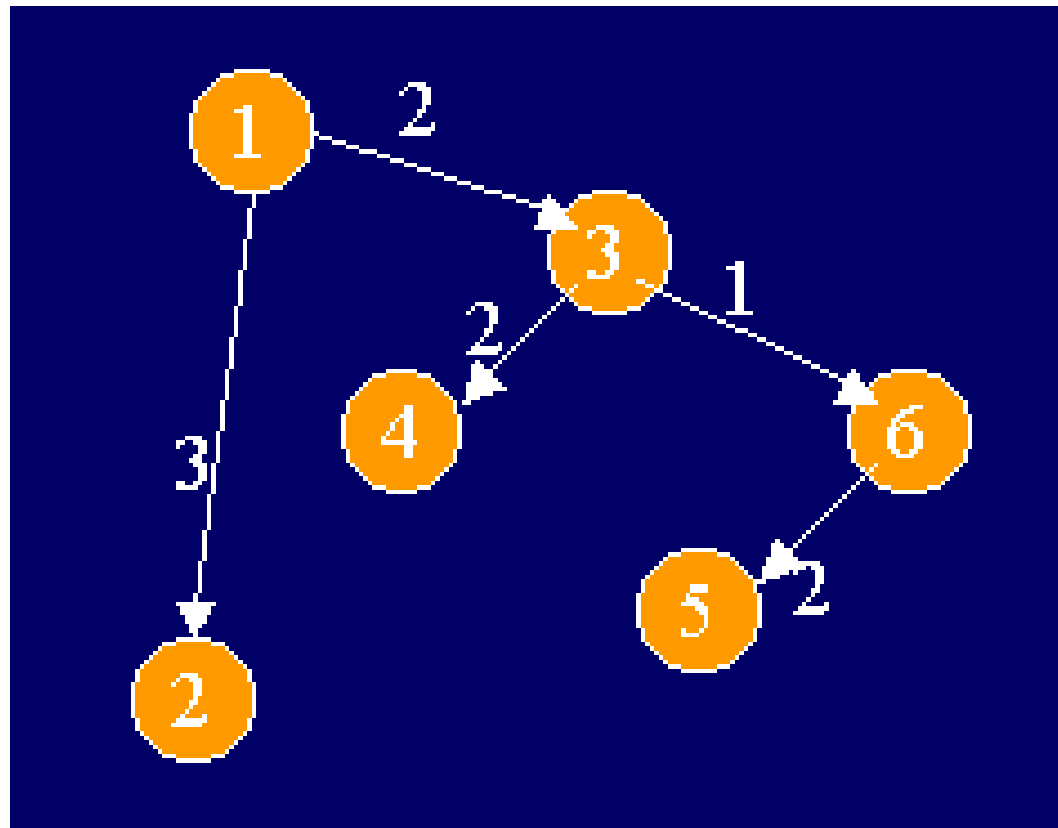
Giải thuật Dijkstra



Lần lặp	N	D ₂	D ₃	D ₄	D ₅	D ₆	P ₂	P ₃	P ₄	P ₅	P ₆
Khởi tạo	{1}	3	2	5	∞	∞	1	1	1	1	1
1	{1,3}	3	<u>2</u>	4	∞	3	1	1	3	1	3
2	{1,3,2}	<u>3</u>		4	7	3	1		3	2	3
3	{1,3,2,6}			4	5	<u>3</u>			3	6	3
4	{1,3,2,6,4}			<u>4</u>	5				3	6	
5	{1,3,2,6,4,5}				<u>5</u>					6	

Giải thuật Dijkstra

- Kết quả



Giải thuật chọn đường tối ưu Ford-Fulkerson

- Mục đích của giải thuật này là để tìm đường đi ngắn nhất từ tất cả các nút đến một nút đích cho trước trên mạng
- Thuộc loại giải thuật tìm đường đi tối ưu – phân tán

Giải thuật chọn đường tối ưu Ford-Fulkerson

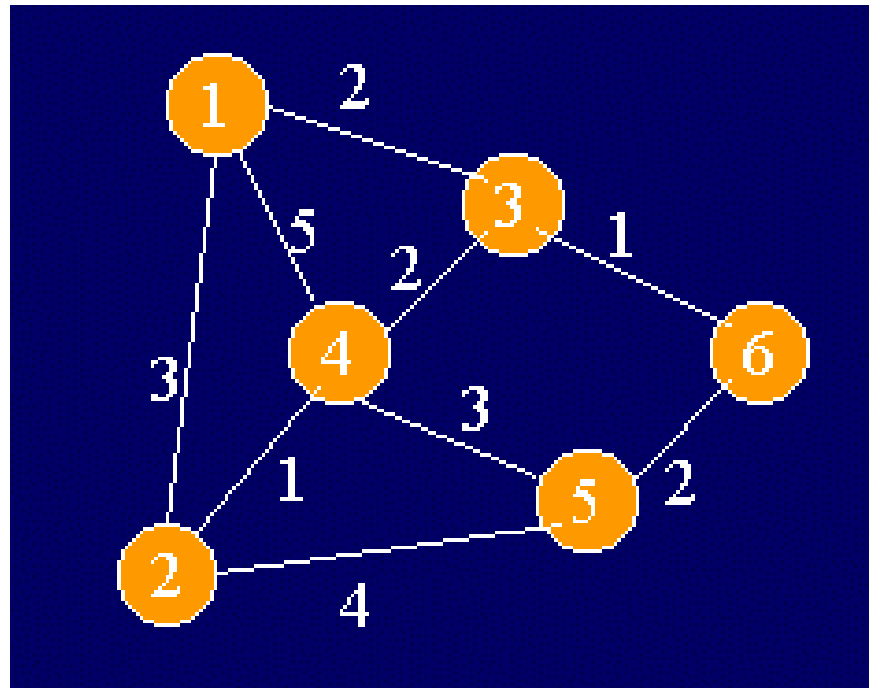
- Gọi
 - d là nút đích cho trước.
 - D_i là chiều dài đường đi ngắn nhất từ nút i đến nút d .
 - C_i là nút con của nút i
- Bước 1: Khởi tạo:
 - Gán $D_d = 0$;
 - Với $\forall i \neq d$: gán $D_i = \infty$; $C_i = -1$;

Giải thuật chọn đường tối ưu Ford-Fulkerson

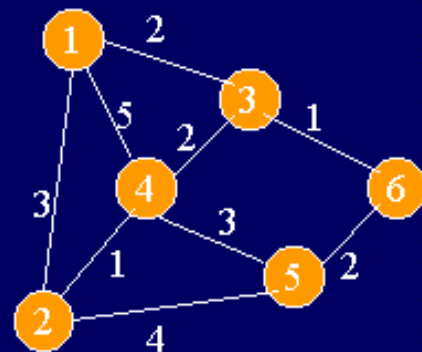
- Bước 2: Cập nhật giá đường đi ngắn nhất từ nút i đến nút d
 - $D_i = \min\{ L_{ij} + D_j \}$ với $\forall j \neq i \Rightarrow C_i = j$;
 - Lặp lại cho đến khi không còn D_i nào bị thay đổi giá trị

Giải thuật chọn đường tối ưu Ford-Fulkerson

- Ví dụ, cho sơ đồ mạng có hình trạng như đồ thị. Hãy tìm đường đi ngắn nhất từ nút khác trên đồ thị đến nút 6. Áp dụng giải thuật ta có:



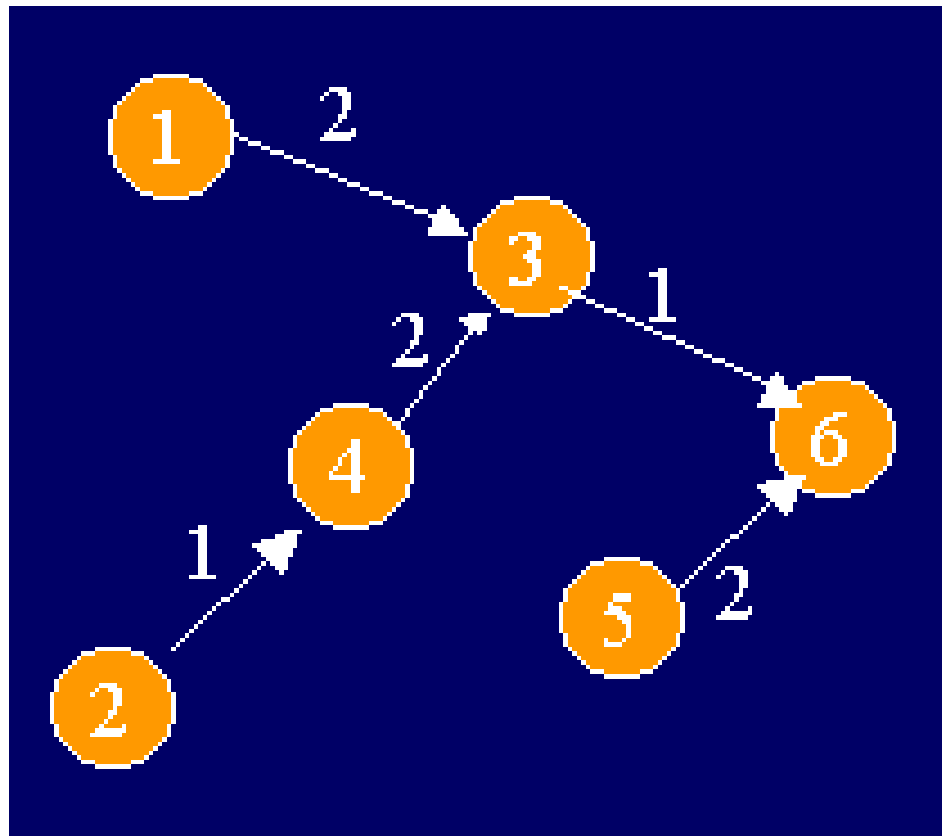
Giải thuật Ford-Fulkerson



Lần lặp	D_1	D_2	D_3	D_4	D_5	C_1	C_2	C_3	C_4	C_5
Khởi tạo	∞	∞	∞	∞	∞	-1	-1	-1	-1	-1
1	∞	∞	1	3	2	-1	-1	6	3	6
2	3	4	1	3	2	3	4	6	3	6
3	3	4	1	3	2	3	4	6	3	6

Giải thuật Ford-Fulkerson

- Kết quả



Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

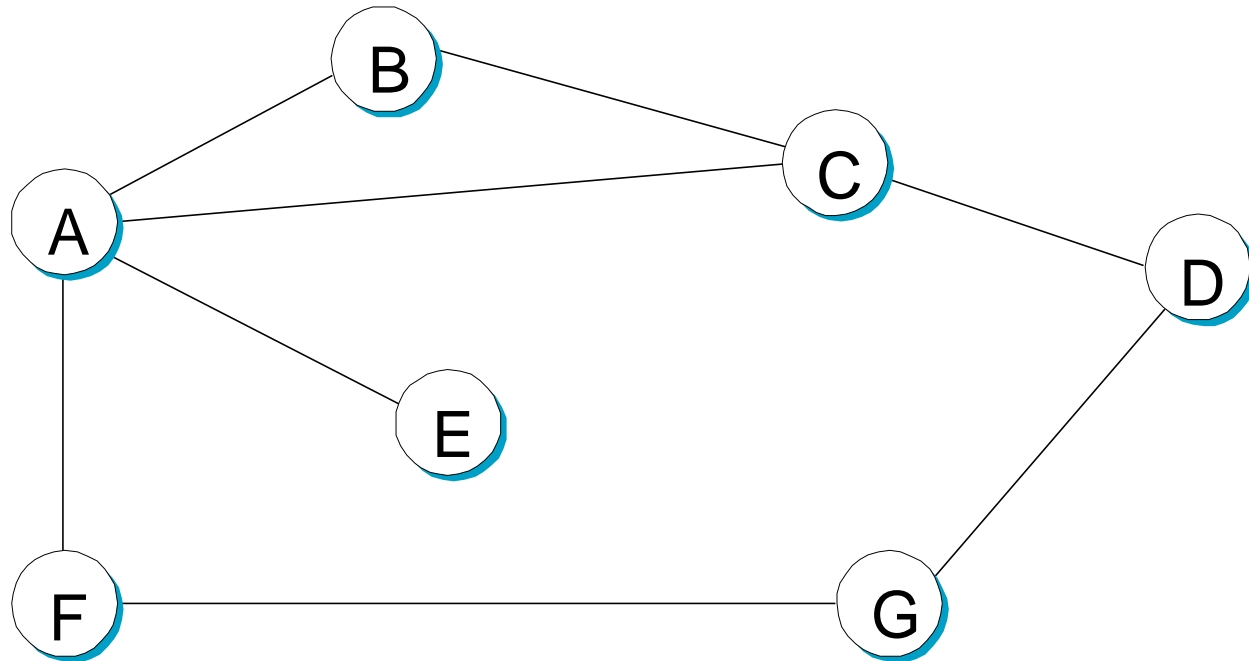
- Mỗi nút thiết lập một mảng một chiều (vector) chứa khoảng cách (chi phí) từ nó đến tất cả các nút còn lại và sau đó phát vector này đến tất cả các nút láng giềng của nó.
- *Giao thức RIP sử dụng giải thuật này*

Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

- Giả thiết
 - Mỗi nút phải biết được chi phí của các đường nối từ nó đến tất cả các nút láng giềng.
 - Một nối kết bị đứt (down) sẽ được gán cho chi phí có giá trị vô cùng.

Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

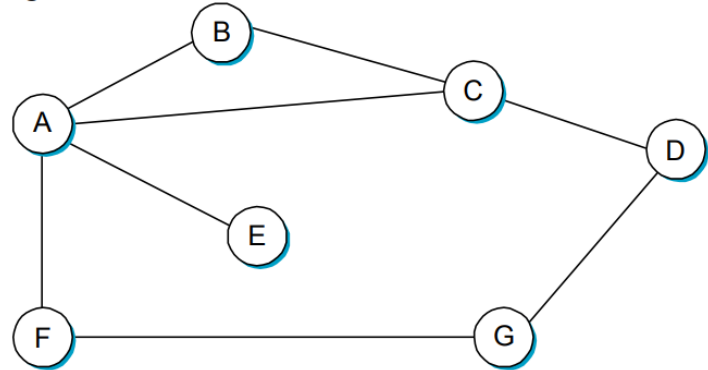
- Ví dụ:



Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

- Khởi đầu, mỗi nút đặt giá trị 1 cho đường nối kết đến các nút láng giềng kề nó, ∞ cho các đường nối đến tất cả các nút còn lại.

Distance Vector



Thông tin được lưu tại các nút	Khoảng cách đến nút						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

- Bảng chọn đường khởi đầu tại nút A

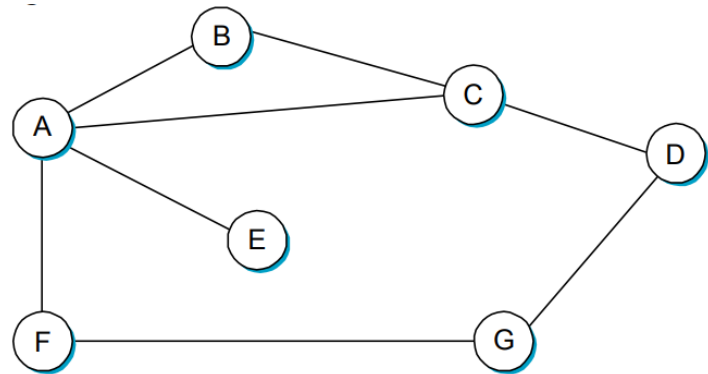
Đích (Destination)	Chi phí (Cost)	Nút kế tiếp (Next Hop)
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

Distance Vector

- Mỗi nút sẽ gửi một thông điệp đến các láng giềng liền kề nó, chứa danh sách các khoảng cách mà cá nhân nút tính được.

Distance Vector

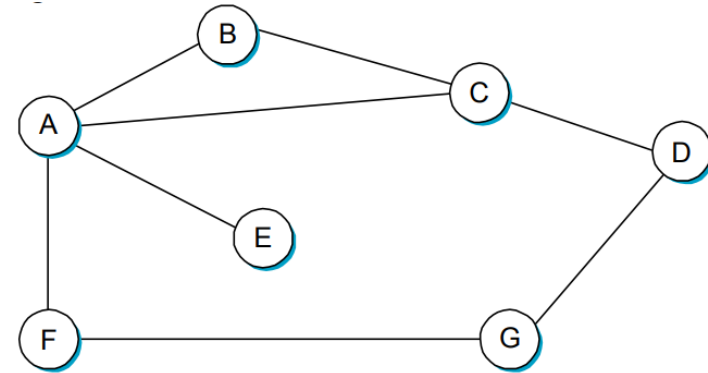
- Ví dụ, nút F bảo nút A rằng F có thể đi đến nút G với chi phí là 1; A cũng biết được rằng nó có thể đến F với chi phí là 1, vì thế A cộng các chi phí lại thành chi phí đi đến G là 2 thông qua F. Tương tự, A có thể học từ các láng giềng khác.



Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

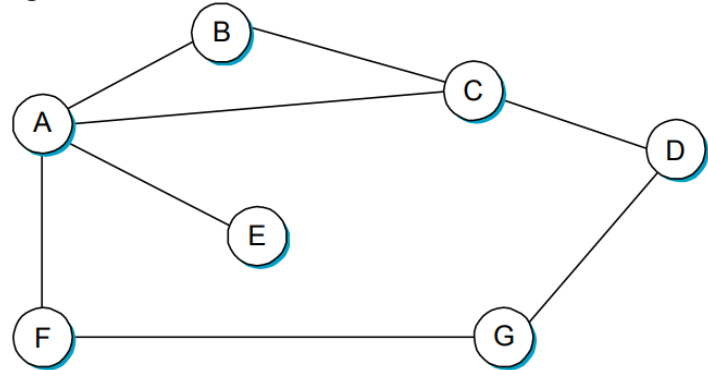
- Bảng chọn đường cuối cùng tại nút A

Đích (Destination)	Chi phí (Cost)	Nút kế tiếp (Next Hop)
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F



Distance Vector

- Kết quả



Thông tin được lưu tại các nút	Khoảng cách đến nút						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Giải thuật chọn đường Vector Khoảng cách (Distance Vector)

- Một số vấn đề:
 - Thời điểm gửi thông tin chọn đường của mình cho các nút láng giềng:
 - ❖ Cập nhật theo chu kỳ
 - ❖ Cập nhật do bị *kích hoạt* khi có sự thay đổi thông tin trong bảng chọn đường của nút

Giải thuật chọn đường Vector

Khoảng cách (Distance Vector)

- Kiểm tra sự hiện diện của láng giềng
 - Gửi thông điệp hỏi thăm láng giềng định kỳ
- Khi phát hiện đường truyền bị sự cố:
 - Router sẽ cập nhật đường đi tương ứng với giá vô cùng và gửi bảng chọn đường mới sang láng giềng
- Vấn đề đếm đến vô cùng (Count-to-Infinity), vấn đề vòng quần

Giải thuật chọn đường “Trạng thái nối kết” (Link State)

- Mỗi nút được giả định có khả năng tìm ra trạng thái đường nối của nó đến các nút láng giềng và chi phí trên mỗi đường nối đó.

Giải thuật chọn đường “Trạng thái nối kết” (Link State)

- Mọi nút đều biết đường đi đến các nút láng giềng kề bên chúng và nếu chúng ta đảm bảo rằng tổng các kiến thức này được phân phối cho mọi nút thì mỗi nút sẽ có đủ hiểu biết về mạng để dựng lên một bản đồ hoàn chỉnh của mạng
- Mỗi nút sẽ chạy các giải thuật tìm đường đi trên hình trạng của toàn mạng để tìm đường đi

Giải thuật chọn đường “Trạng thái nối kết” (Link State)

- Làm ngập một cách tin cậy (Reliable Flooding)
 - Đảm bảo tất cả các nút tham gia vào giao thức chọn đường đều nhận được thông tin về trạng thái nối kết từ tất cả các nút khác.

Giải thuật chọn đường “Trạng thái nối kết” (Link State)

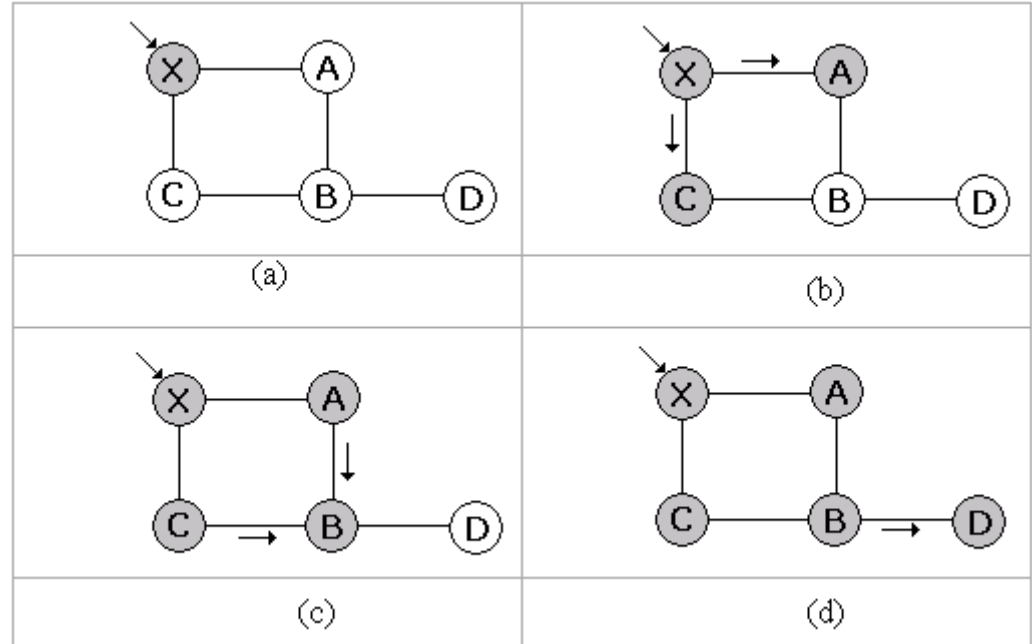
- Một nút phát thông tin về trạng thái nối kết của nó với mọi nút láng giềng liền kề, đến lượt mỗi nút nhận được thông tin trên lại chuyển phát thông tin đó ra các nút láng giềng của nó. Tiến trình này cứ tiếp diễn cho đến khi thông tin đến được mọi nút trong mạng.

Giải thuật chọn đường “Trạng thái nối kết” (Link State)

- Mỗi nút tạo ra gói tin cập nhật, còn được gọi là gói tin trạng thái nối kết (link-state packet – LSP), chứa :
 - ID của nút đã tạo ra LSP
 - Một danh sách các nút láng giềng có đường nối trực tiếp tới nút đó, cùng với chi phí của đường nối đến mỗi nút.
 - Một số thứ tự
 - Thời gian sống (time to live) của gói tin này

Giải thuật chọn đường “Trạng thái nối kết” (Link State)

- Làm ngập một cách tin cậy (Reliable Flooding)



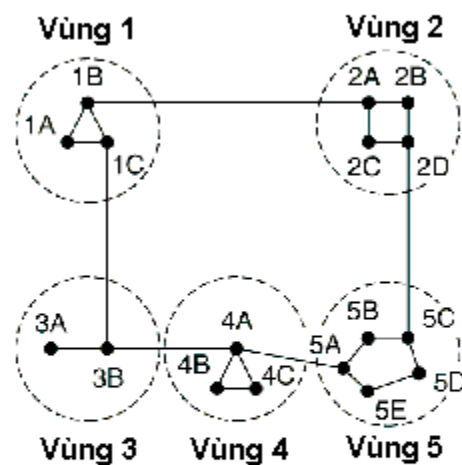
Chọn đường phân cấp (Hierarchical Routing)

- Khi mạng tăng kích thước:
 - Tăng kích thước bảng chọn đường của các router
 - Tăng kích thước bộ nhớ
 - Tăng thời gian tìm kiếm đường đi
 - Cần thực hiện chọn đường phân cấp

Chọn đường phân cấp (Hierarchical Routing)

- Trong chọn đường phân cấp:
 - Các router được chia thành những vùng (domain).
 - Router biết cách chọn đường bên trong vùng, nhưng không biết gì về cấu trúc bên trong của các vùng khác.

Chọn đường phân cấp (Hierarchical Routing)



(a)

Bảng vạch đường
đầy đủ của nút 1A

Đích Lỗi ra Chi phí

1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Bảng vạch đường
phân cấp của host 1A

Đích Lỗi ra Chi phí

1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

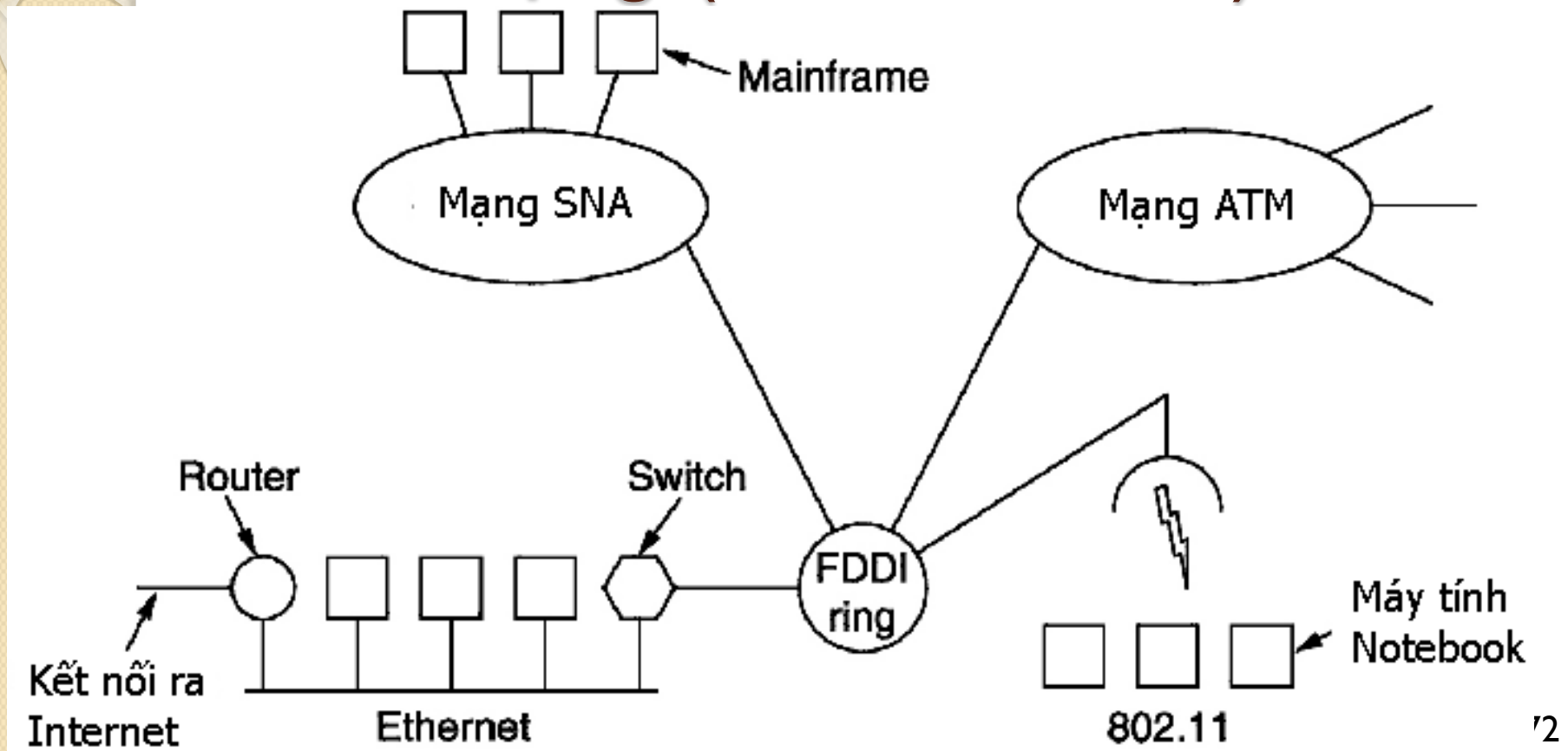


Liên mạng và bộ giao thức IP

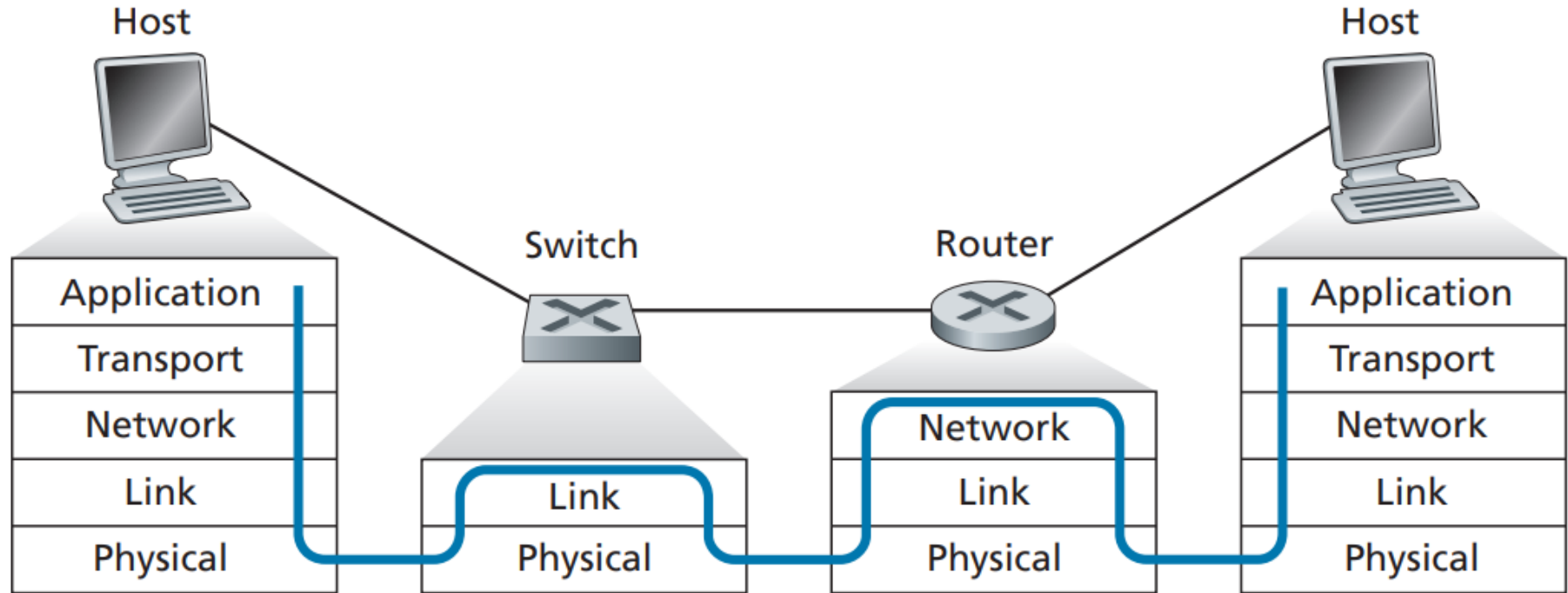
Liên mạng (Internetwork)

- Liên mạng: Mạng được hình thành từ việc liên nối kết nhiều mạng lại với nhau
- Các mạng thành phần là không đồng nhất (homogeneous): khác nhau về phần cứng, phần mềm, giao thức
- Mục tiêu của việc xây dựng liên mạng là cho phép người dùng trên một mạng con có thể liên lạc được với người dùng trên các mạng con khác

Liên mạng (Internetwork)



Liên mạng (Internetwork)



Các hình thức xây dựng liên mạng

- Ở tầng vật lý: Các mạng có thể được nối kết bằng các repeater hoặc hub, những thiết bị chỉ đơn thuần làm nhiệm vụ di chuyển các bit từ mạng này sang mạng kia.

Các hình thức xây dựng liên mạng

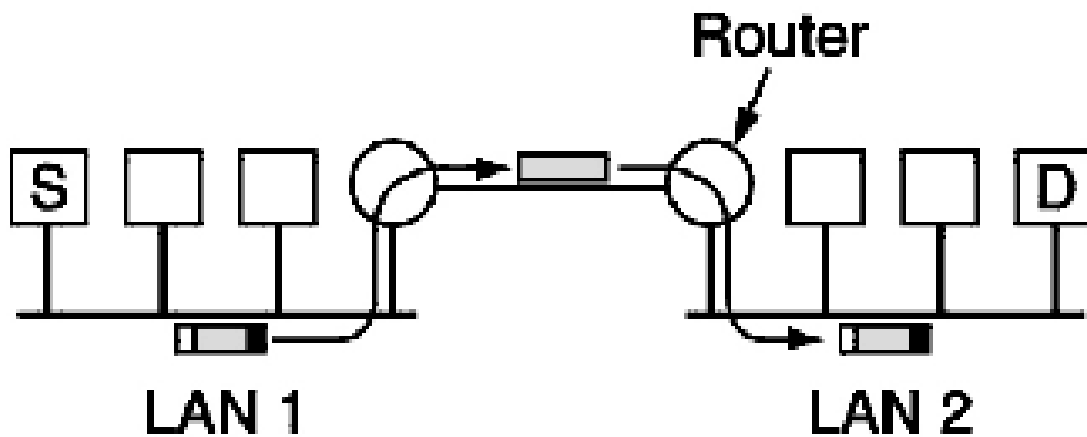
- Ở tầng LKDL: Người ta dùng các cầu nối (bridges) hoặc switches. Chúng có thể nhận các khung, phân tích địa chỉ MAC và cuối cùng chuyển khung sang mạng khác trong khi song song đó, chúng vừa làm nhiệm vụ giám sát quá trình chuyển đổi giao thức, ví dụ như từ Ethernet sang FDDI hoặc 802.11.

Các hình thức xây dựng liên mạng

- Ở tầng mạng: Người ta dùng các router để nối kết các mạng với nhau. Nếu hai mạng có tầng mạng khác nhau, router có thể chuyển đổi khuôn dạng gói tin, quản lý nhiều giao thức khác nhau trên các mạng khác nhau.

Liên mạng ở tầng mạng

- Hai router được nối với nhau bằng đường nối điểm-điểm,
- Máy S muốn gửi cho máy D một gói tin,
- S đóng gói gói tin này thành một khung và gửi lên đường truyền.

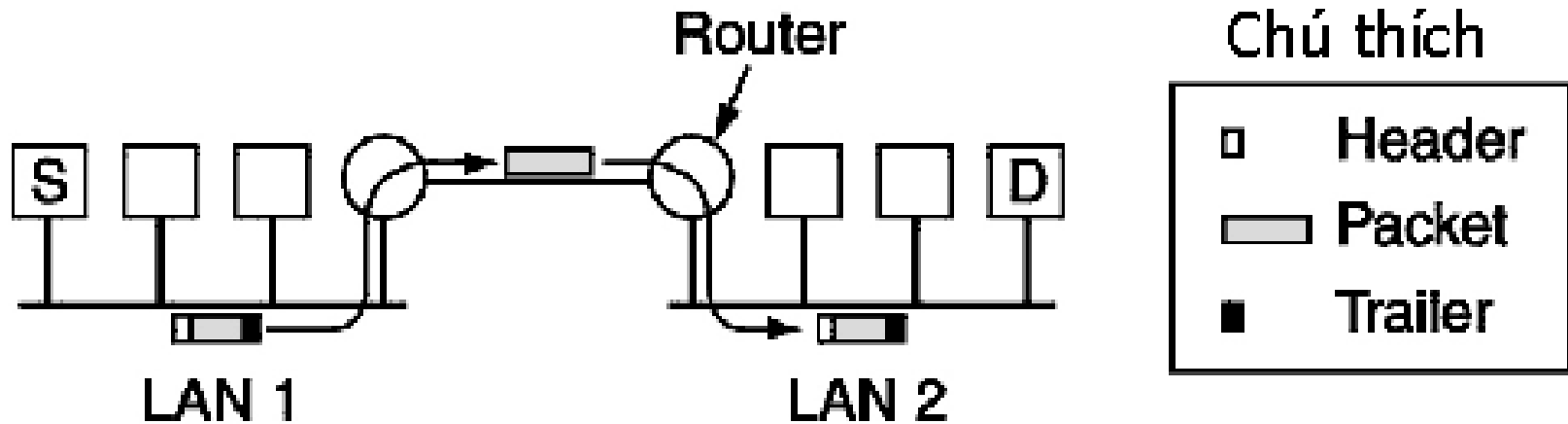


Chú thích

- | | |
|---|---------|
| □ | Header |
| ▬ | Packet |
| ■ | Trailer |

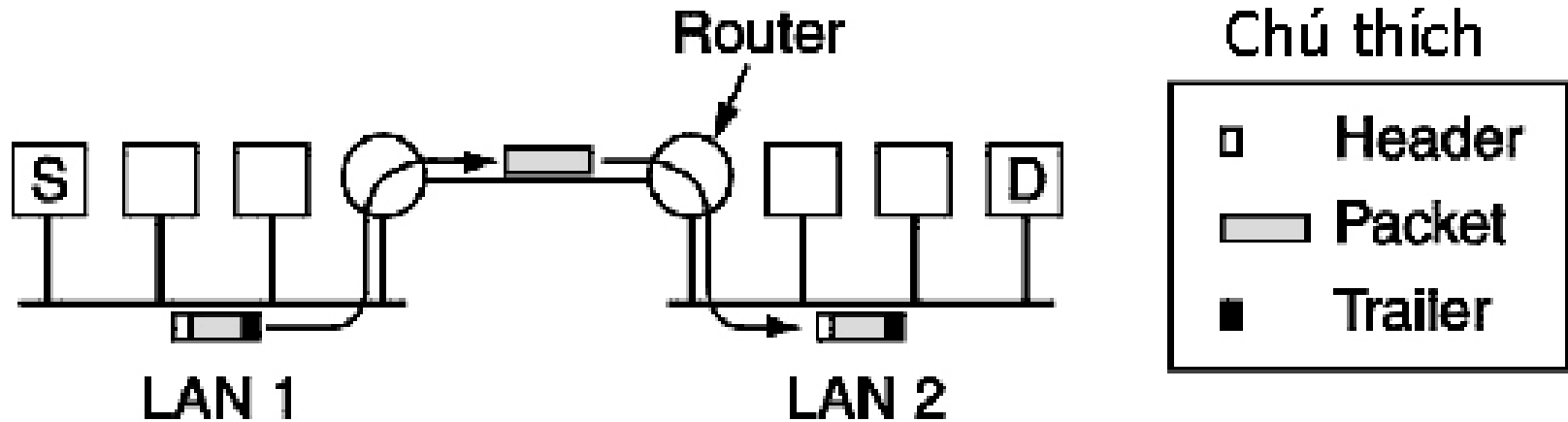
Liên mạng ở tầng mạng

- Khung đến được router của LAN1, router này liền bóc vỏ khung, lấy gói tin ra, tìm ra địa chỉ mạng (IP) của máy đích, địa chỉ này sẽ được tham khảo trong bảng chọn đường của router LAN1 để tìm đường đi đến LAN 2



Liên mạng ở tầng mạng

- router LAN1 quyết định chuyển gói sang router LAN2 bằng cách đóng thành khung gởi cho router LAN2.



Bộ giao thức liên mạng (IPs - Internet Protocols)

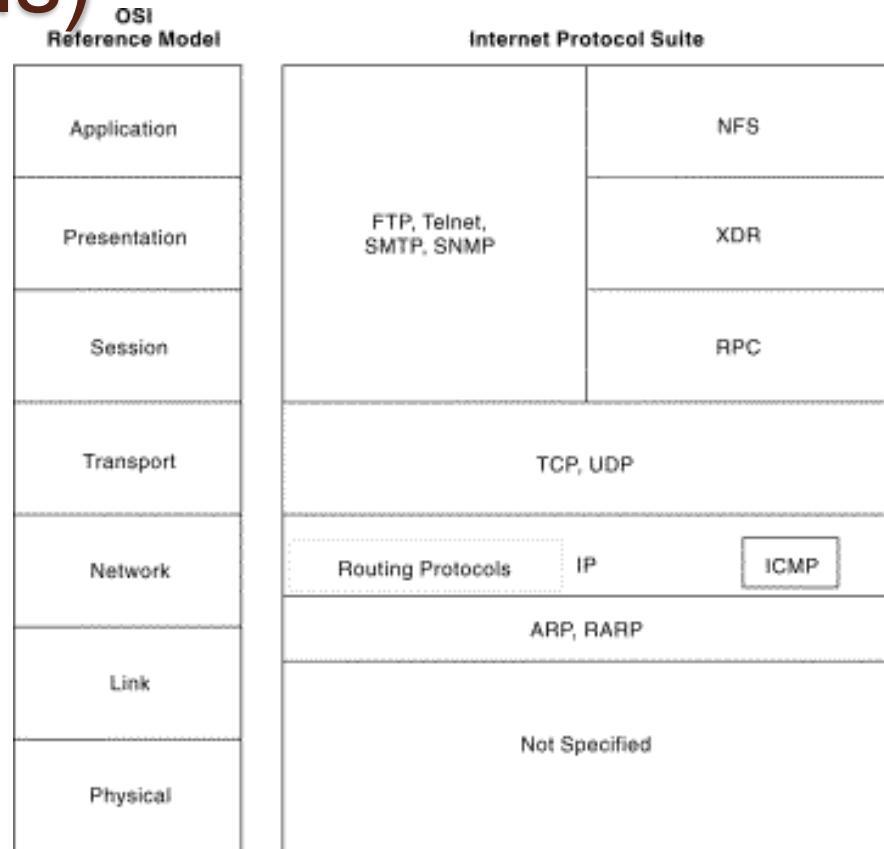
- Là bộ giao thức liên mạng cho các hệ thống mở nổi tiếng nhất trên thế giới
- Được sử dụng để giao tiếp qua bất kỳ các liên mạng nào cũng như thích hợp cho các giao tiếp trong mạng LAN và mạng WAN.

Bộ giao thức liên mạng (IPs - Internet Protocols)

- Bao gồm một bộ giao thức:
 - Tầng ứng dụng: SMTP, FTP, TELNET, HTTP, ...
 - Tầng 4 :
 - ❖ TCP (Transmission Control Protocol)
 - ❖ UDP (User Datagram Protocol)
 - Tầng 3: IP (Internet Protocol)
 - Và các giao thức khác: ARP, RARP, ICMP, ...

Bộ giao thức liên mạng (IPs - Internet Protocols)

- TCP/IP được tích hợp vào hệ điều hành UNIX phiên bản BSD (Berkeley Software Distribution)
- Trở thành nền tảng cho mạng Internet và dịch vụ WWW (World Wide Web)



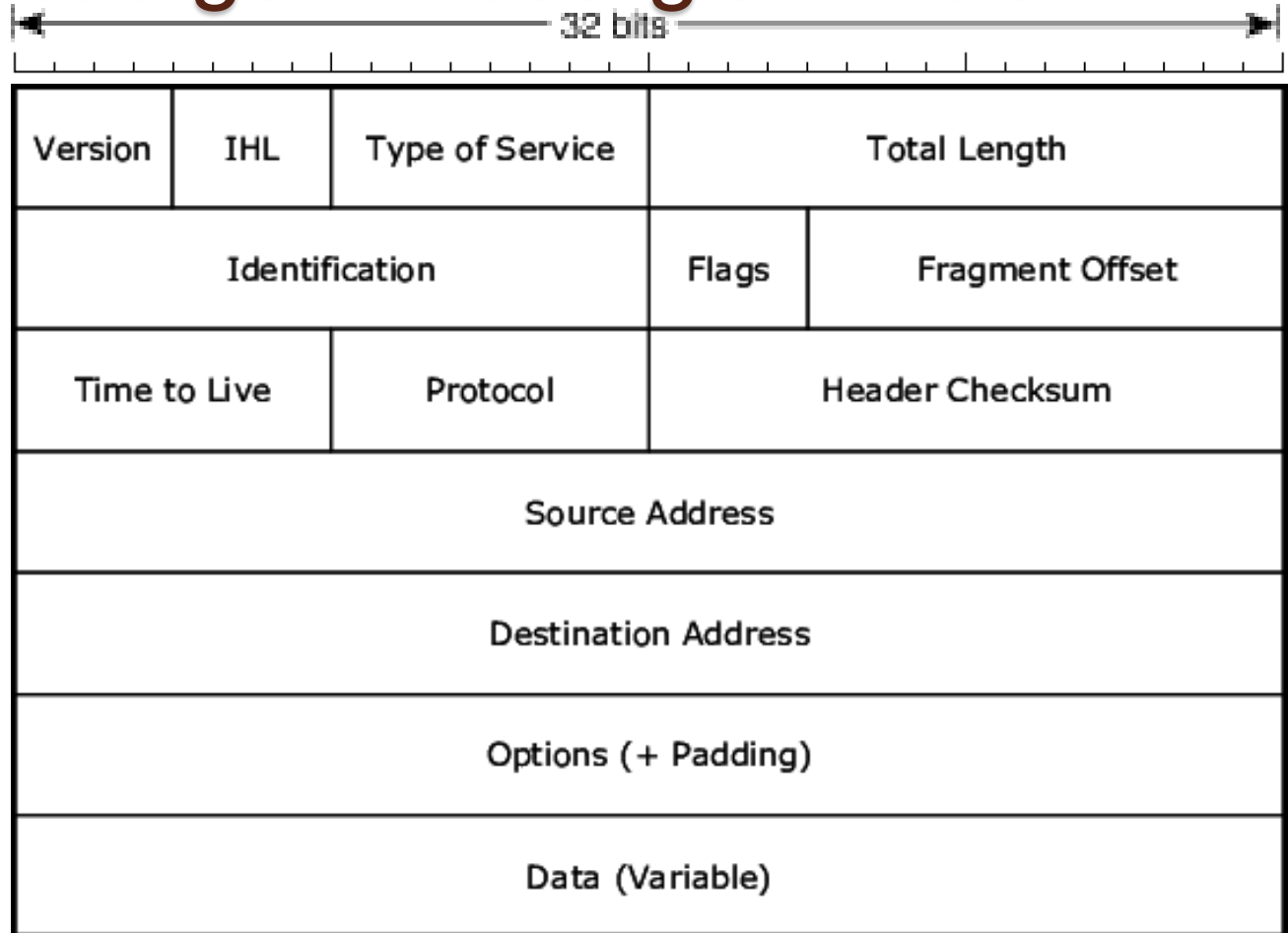
Giao thức IP (Internet protocol)

- Hoạt động ở tầng 3 của mô hình OSI
- Qui định *cách thức định địa chỉ* các máy tính và cách thức *truyền* các gói tin qua một liên mạng
- Được đặc tả trong RFC 791
- Cùng với giao thức TCP, IP trở thành trái tim của Bộ giao thức Internet

Giao thức IP (Internet protocol)

- Có hai chức năng chính
 - Cung cấp *dịch vụ dạng không nối kết* để chuyển tải các gói tin qua một liên mạng.
 - Phân mảnh cũng như tập hợp lại các gói tin để hỗ trợ cho tầng liên kết dữ liệu với kích thước đơn vị truyền dữ liệu là khác nhau.

Cấu trúc gói tin của giao thức IPv4



IPv4 datagram

- (sinh viên tự dịch sang tiếng Việt)
- An IPv4 datagram consists of a header part and a body or payload part
- The header has a 20-byte fixed part and a variable-length optional part
- The bits are transmitted from left to right and top to bottom (“big-endian” network byte order)

IPv4 datagram

- **Version:** version of the protocol.
- **IHL:** IP header length (in 32-bit words), value from 5 -> 15 or 20 -> 60 bytes. The minimum value is 5 (no *Options* field).
- **Type of service (Differentiated services):** It was and still is intended to distinguish between different classes of service. Various combinations of reliability and speed are possible.

IPv4 datagram

- ***Total length:*** includes everything in the datagram—both header and data. The maximum length is 65,535 bytes (or larger in the future).
- ***Identification:*** is needed to allow the destination host to determine which packet a newly arrived fragment belongs to. All the fragments of a packet contain the same Identification value.

IPv4 datagram

- **Flags:** 3 bits = {unused bit, DF, MF}
 - Unused bit: must be zero.
 - DF stands for Don't Fragment. It is an order to the routers not to fragment the packet.
 - MF stands for More Fragments. All fragments except the last one have this bit set (last fragment, this bit is 0).

IPv4 datagram

- ***Fragment offset***: This field specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram. The fragmentation offset value for the first fragment is always 0.
- ***TTL (Time to live)***: is a counter used to limit packet lifetimes (to prevent routing loop). In practice, it just counts hops (not seconds). When the TTL field hits zero, the router discards the packet and typically sends an "ICMP time exceeded" message to the sender.

IPv4 datagram

- **Protocol:** defines the protocol used in the data portion of the IP datagram (tells it which transport process to give the packet to, e.g. TCP, UDP, ICMP...).
- **Header checksum:** The 16-bit IPv4 header checksum field is used for error-checking of the header. When a packet arrives at a router, the router decreases the TTL field in the header. Consequently, the router must calculate a new header checksum.

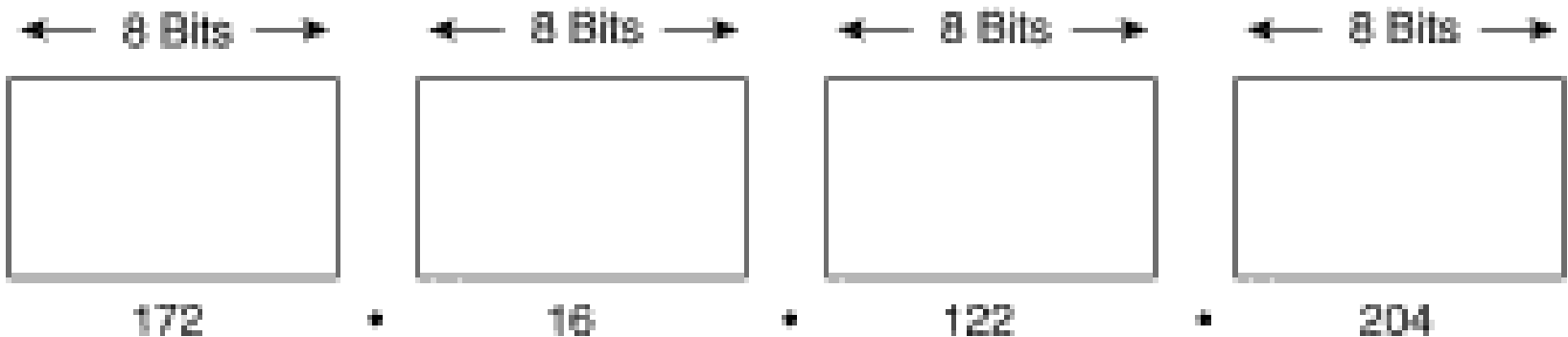
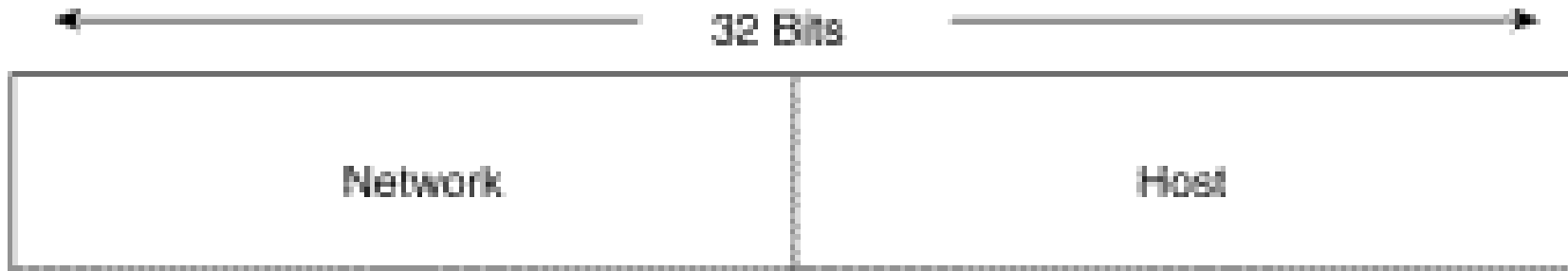
IPv4 datagram

- **Source address** and **Destination address**: indicate the IPv4 address
- **Options**: The options field is not often used. It supports information not present in the original header design. Note that packets containing some options may be considered as dangerous by some routers and be blocked.
- **Data**: The packet payload.

Cấu trúc địa chỉ IP (IPv4)

- Địa chỉ IP gồm 32 bit (4 byte), do đó có 2^{32} (gần 4 tỷ) giá trị địa chỉ khác nhau.
- Mỗi byte địa chỉ được viết dạng thập phân và phân cách với các byte khác bằng dấu chấm (dotted-decimal notation).

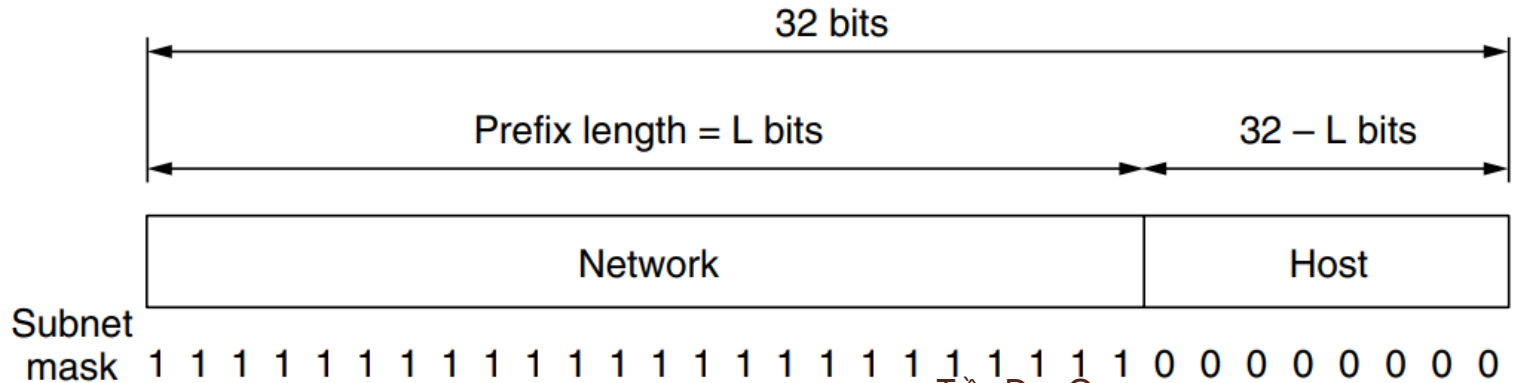
Cấu trúc địa chỉ IP



Dotted
Decimal
Notation

Cấu trúc địa chỉ IP

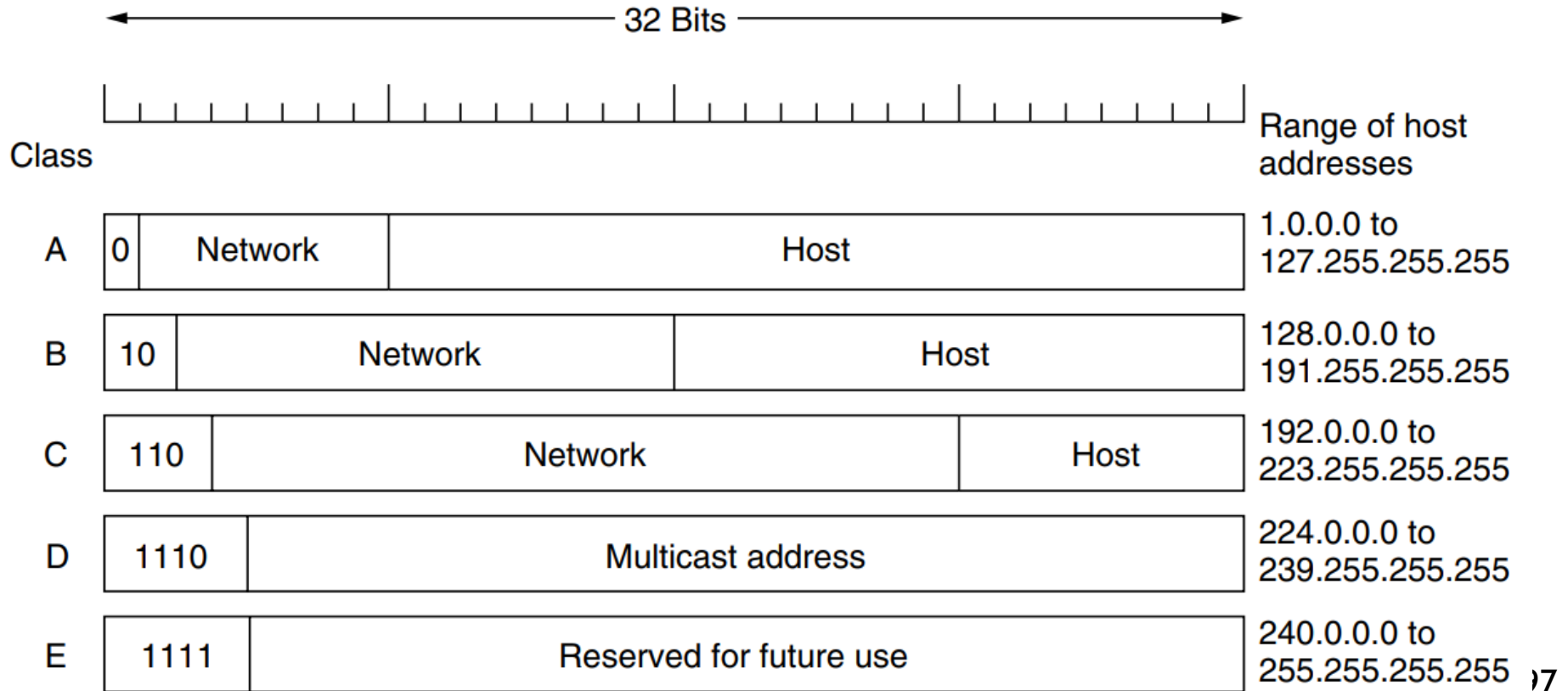
- Địa chỉ IP gồm hai phần: nhận dạng mạng (Network ID) gồm các bit đầu và phần nhận dạng máy tính/interface (Host ID) gồm các bit còn lại.



Định địa chỉ IP dạng Classful

- Định địa chỉ dạng Classful (classful addressing): phần nhận dạng mạng (Network ID) của địa chỉ IP bị ràng buộc với chiều dài là 8, 16, 24 bit (hay 1, 2, 3 byte).
- Trước năm 1993, chưa có net mask (subnet mask), địa chỉ IP được chia thành các lớp sau:

Định địa chỉ IP dạng Classful



Một số địa chỉ IP đặc biệt

- Địa chỉ mạng (Network Address): là địa chỉ IP mà giá trị của tất cả các bits ở phần *Host ID* đều là 0, được sử dụng để xác định một mạng.
 - Ví dụ : 10.0.0.0; 172.18.0.0; 192.1.1.0

Một số địa chỉ IP đặc biệt

- Địa chỉ quảng bá (Broadcast Address):
Là địa chỉ IP mà giá trị của tất cả các bits ở phần *Host ID* đều là 1, được sử dụng để chỉ tất cả các máy tính trong mạng.
 - Ví dụ : 10.255.255.255,
172.18.255.255, 192.1.1.255
 - *Không được dùng địa chỉ này để đặt cho các máy tính*

Một số địa chỉ IP đặc biệt

- Mặt nạ mạng chuẩn (Netmask): Là địa chỉ IP mà giá trị của các bits ở phần nhận dạng mạng (Network ID) đều là 1, các bits ở phần nhận dạng máy tính (Host ID) đều là 0.

Một số địa chỉ IP đặc biệt

- Như vậy khi dùng mặt nạ mạng để thể hiện 3 lớp mạng A, B và C, ta có:
 - Mặt nạ mạng lớp A : 255.0.0.0
 - Mặt nạ mạng lớp B : 255.255.0.0
 - Mặt nạ mạng lớp C : 255.255.255.0

Ta gọi chúng là các mặt nạ mạng mặc định (Default Netmask)

Một số địa chỉ IP đặc biệt

- Địa chỉ mạng 127.0.0.0 là địa chỉ được dành riêng để đặt cho từng máy tính. Nó chỉ có giá trị cục bộ (trong phạm vi một máy tính).
- Thông thường khi cài đặt giao thức IP thì máy tính sẽ được gán địa chỉ 127.0.0.1 (Loopback). Địa chỉ này thông thường để kiểm tra xem giao thức IP trên máy hiện tại có hoạt động không.

Mặt nạ mạng (Netmask)

- Ý nghĩa: Dùng để tính ra địa chỉ mạng từ địa chỉ IP và mặt nạ mạng (Netmask) bằng phép AND từng bit.
- Network address = IP address & Netmask
- VD: cho địa chỉ IP = 198.53.147.45 và Netmask = 255.255.255.0
=> Network address = 198.53.147.45 & 255.255.255.0 = 198.53.147.0

Mặt nạ mạng (Netmask)

- Biểu diễn thập phân

- 198.53.147.45

&

- 255.255.255.0

=198.53.147.0

- Biểu diễn nhị phân

- 11000110 00110101 10010011 00101101

&

- 11111111 11111111 11111111 00000000

- 11000110 00110101 10010011 00000000

Địa chỉ IP public và IP private

- Địa chỉ IP public (công cộng) dùng để xác định máy tính trong mạng internet.
- Nếu ta muốn sử dụng địa chỉ này, ta phải đăng ký với ISP và phải trả phí.
 - Vấn đề: số lượng địa chỉ public không đủ dùng => một giải pháp là dùng địa chỉ private.

Địa chỉ IP public và IP private

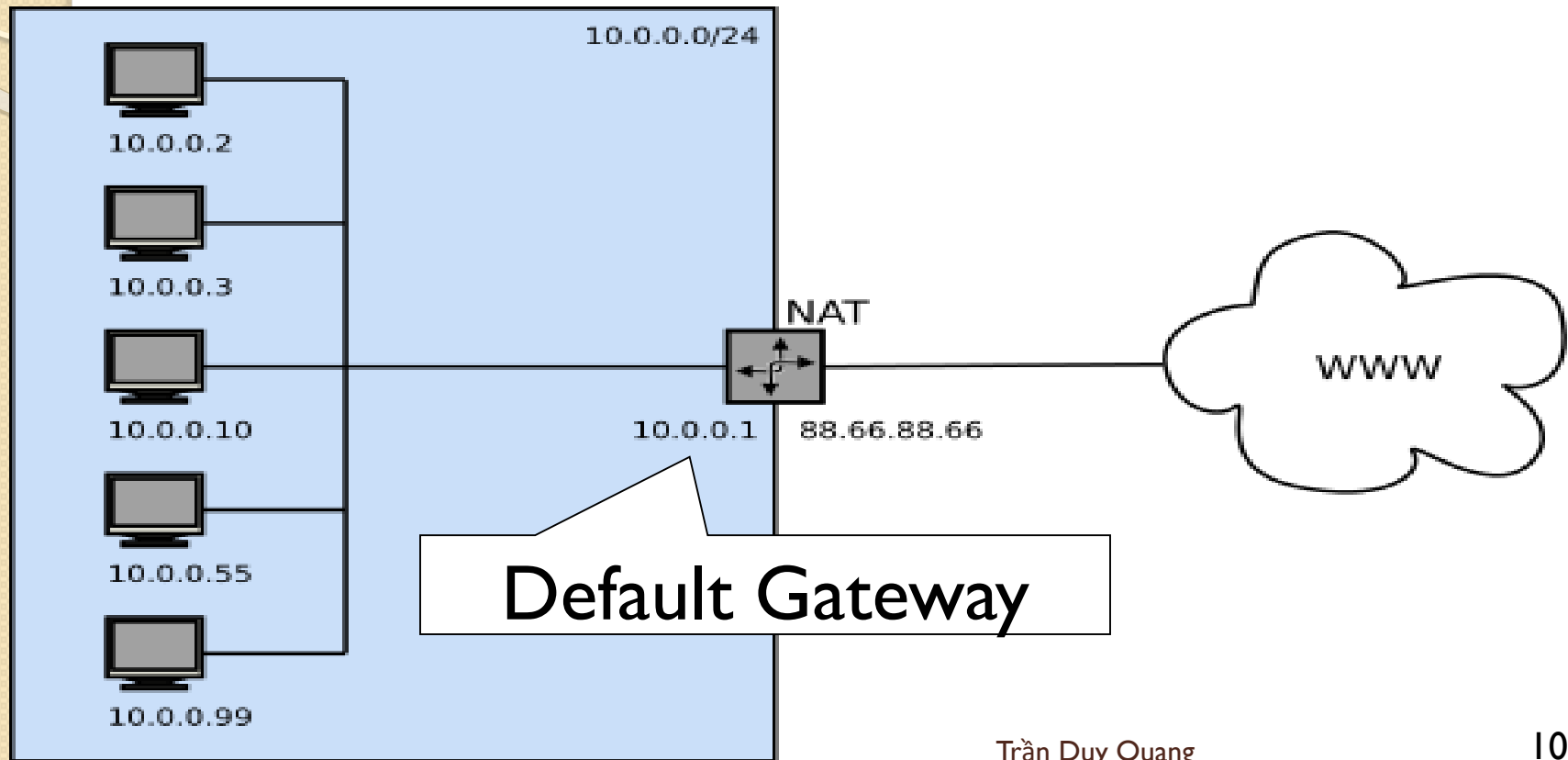
- Địa chỉ IP private (riêng) dùng để xác định máy tính trong mạng cục bộ.
- Nếu ta muốn gửi dữ liệu từ mạng cục bộ ra mạng internet?

=> chuyển đổi địa chỉ mạng: – NAT (Network Address Translation). NAT được cấu hình ở Router biên (Router kết nối trực tiếp internet) để chuyển đổi: IP private <-> IP public .

Địa chỉ IP public và IP private

- Địa chỉ private:
 - 10.0.0.0 đến 10.255. 255. 255 (Lớp A)
 - 172.16.0.0 đến 172.31.255.255 (Lớp B)
 - 192.168.0.0 đến 192.168.255.255 (Lớp C)
- Địa chỉ public là địa chỉ nằm ngoài các khoảng địa chỉ này

NAT



Chia mạng con

Việc chia mạng con có các tiện lợi sau:

- Đơn giản hóa việc quản trị
- Có thể thay đổi cấu trúc bên trong của mạng mà không ảnh hưởng đến các mạng bên ngoài
- Ẩn địa chỉ của các mạng con trong hệ thống
- Cô lập các luồng giao thông trên mạng – giảm tối đa đụng độ

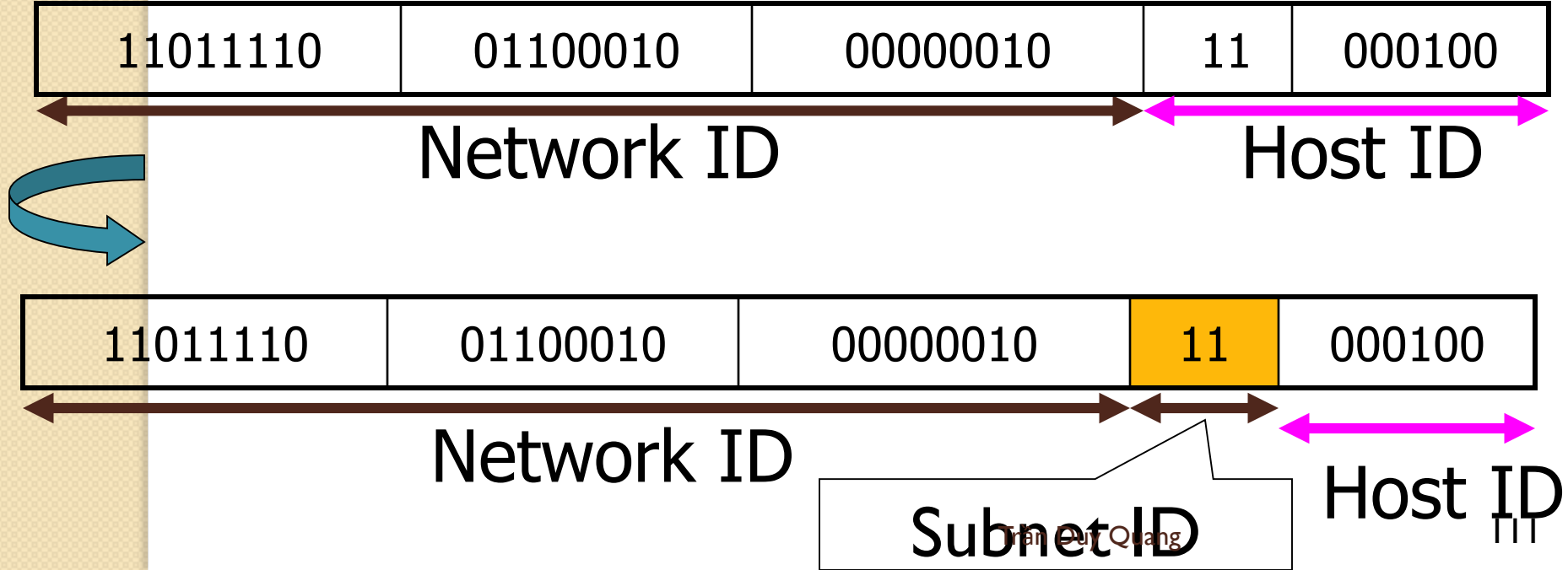
(Thường dùng Router để kết nối mạng con)

Chia mạng con

- Nguyên tắc phân mạng con:
 - Phần nhận dạng mạng (Network ID) của địa chỉ mạng ban đầu được giữ nguyên
 - Mượn thêm vài bit ở phần nhận dạng máy tính (Host ID) để làm phần nhận dạng mạng con (Subnet ID) [từ trái sang phải]

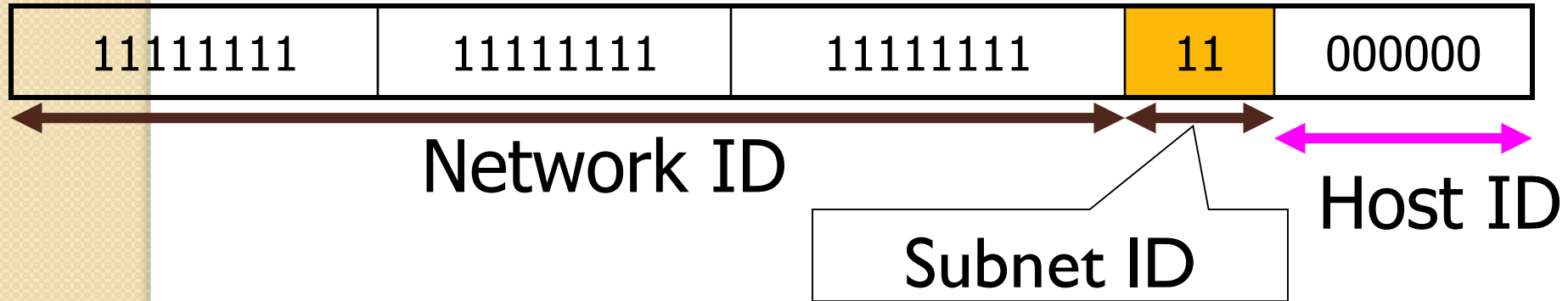
Chia mạng con

- Ví dụ dạng Fixed Length Subnet Mask (FLSM):



Chia mạng con

- Mặt nạ mạng con (Subnetmask) có phần Network ID và Subnet ID là các bit 1, có phần Host ID là các bit 0.



Chia mạng con

- Do đó, địa chỉ mạng con (Subnetwork address) là:

Subnetwork address = IP &
Subnetmask

- &: phép toán AND từng bit

Chia mạng con

- Số lượng bit thuộc phần nhận dạng mạng con (subnet ID) xác định số lượng mạng con.
- Số lượng bit còn lại xác định số lượng Host ID. Nếu dùng Router cho mạng con này, ta cần loại 3 địa chỉ trong Host ID: 1 địa chỉ mạng và 1 địa chỉ quảng bá và 1 địa chỉ (Host ID) dùng để đặt cho 1 cổng/interface của Router (default gateway).

Chia mạng con

- Gọi m là số bit mượn của phần Host ID ban đầu
 - Gọi n là số bit còn lại của phần Host ID sau khi bị mượn m bit
- => Số subnet ID được tạo ra: 2^m
- => Số host ID / subnet (bao gồm interface của router): $2^n - 2$
- (Nếu không bao gồm interface của router là $2^n - 3$)*

Chia mạng con

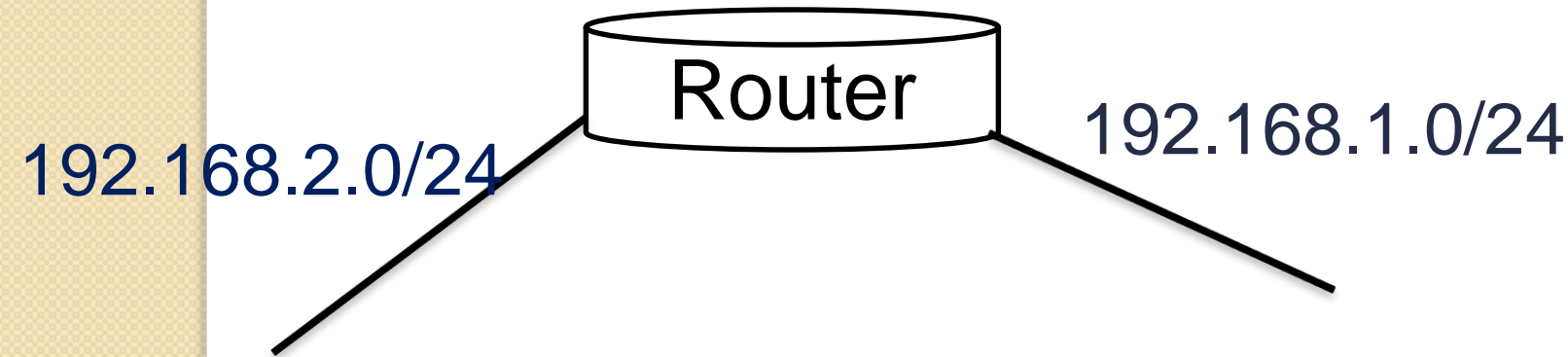
- Ví dụ: cho địa chỉ mạng lớp C: 192.168.1.0 với mặt nạ mạng mặc định là 255.255.255.0
 - Xét trường hợp ta mượn 2 bit trong Host ID ban đầu để làm phần nhận dạng mạng con (subnet ID) (có dùng Router để kết nối với mạng khác).
- => Mặt nạ mạng con là 255.255.255.192

Chia mạng con

Địa chỉ IP	Dạng thập phân	Dạng nhị phân			
Mạng ban đầu	192.168.1.0	1100 0000	1010 1000	0000 0001	0000 0000
Mạng con 1	192.168.1.0	1100 0000	1010 1000	0000 0001	0000 0000
Mạng con 2	192.168.1.64	1100 0000	1010 1000	0000 0001	0100 0000
Mạng con 3	192.168.1.128	1100 0000	1010 1000	0000 0001	1000 0000
Mạng con 4	192.168.1.192	1100 0000	1010 1000	0000 0001	1100 0000

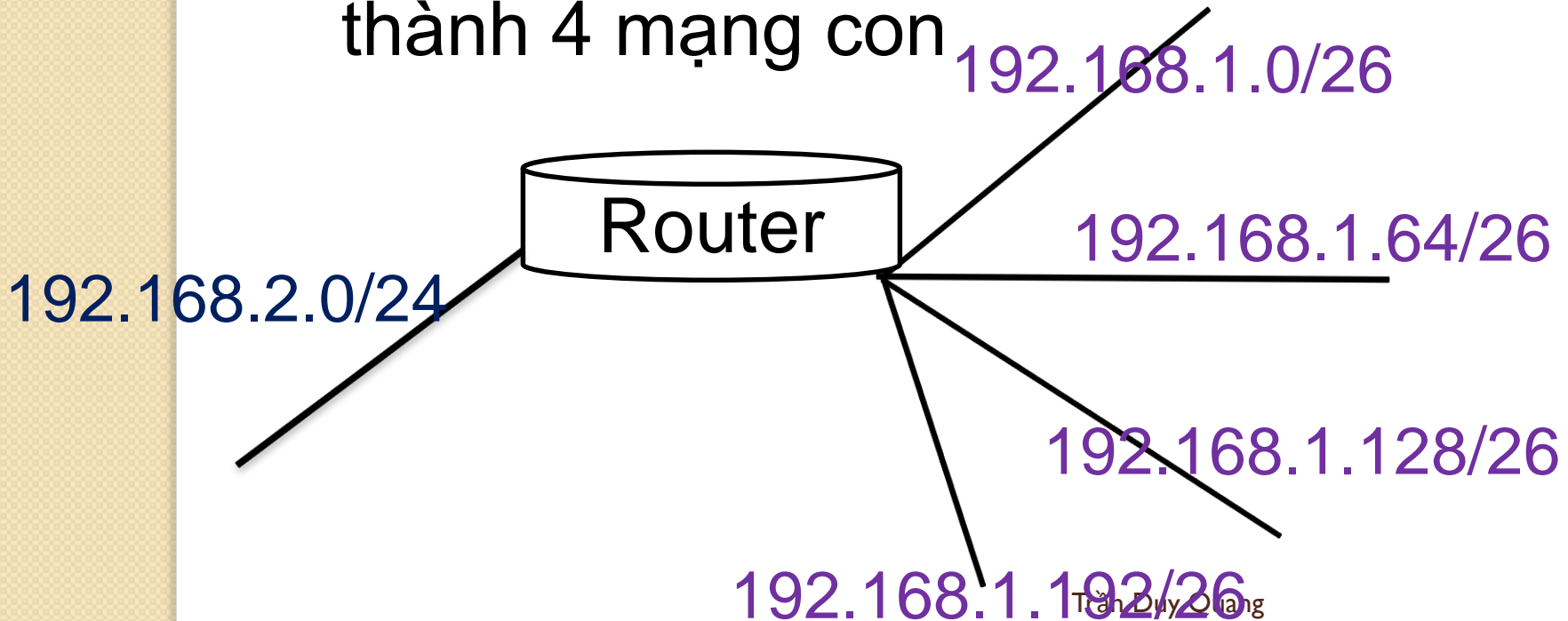
Chia mạng con

- Ví dụ: trước khi chia mạng
192.168.1.0/24



Chia mạng con

- Sau khi chia mạng 192.168.1.0/24 thành 4 mạng con



Chia mạng con

- Số lượng Subnet ID: $2^2 = 4$
- Số lượng Host ID sẵn dùng cho mỗi Subnet ID (trừ địa chỉ dành cho Router): $2^6 - 3 = 64 - 3 = 61$
- Vậy ở mỗi mạng con này ta có thể dùng 61 địa chỉ Host ID để đặt cho máy tính.

Phương pháp chọn đường liên miền không phân lớp CIDR

- CIDR (Classless Inter-Domain Routing) là một sơ đồ đánh địa chỉ mới cho mạng Internet hiệu quả hơn nhiều so với sơ đồ đánh địa chỉ cũ theo kiểu phân lớp A, B và C.
- CIDR ra đời để giải quyết hai vấn đề bức xúc đối với mạng Internet là:
 - Thiếu địa chỉ IP.
 - Vượt quá khả năng chứa đựng của các bảng chọn đường.

Phương pháp chọn đường liên miền không phân lớp CIDR

- Cấu trúc địa chỉ CIDR:
 - Không sử dụng cơ chế phân lớp A,B,C,E,D
 - Phần nhận dạng mạng: từ 13 đến 27 bits
- Một địa chỉ theo cấu trúc CIDR:
 - Bao gồm 32 bits của địa chỉ IP chuẩn và thông tin về số lượng bit được sử dụng cho phần nhận dạng mạng (Network ID)
 - Ví dụ : 206.13.01.48/25

Phương pháp chọn đường liên miền không phân lớp CIDR

- Kết hợp việc chọn đường có phân cấp để giảm tối đa số lượng mục từ trong bảng chọn đường của các Router
 - CIDR cho phép kết hợp các đường đi
 - ❖ Mục từ trong bảng chọn đường ở mức cao có thể đại diện cho nhiều router ở mức thấp hơn trong các bảng chọn đường tổng thể

Phương pháp chọn đường liên miền không phân lớp CIDR

- Tương tự kiến trúc phân cấp của mạng điện thoại
 - Một router ở mức cao (quốc gia), chỉ quan tâm đến mã quốc gia trong số điện thoại, sau đó nó sẽ vạch đường cho cuộc gọi đến router đường trục phụ trách mạng quốc gia tương ứng với mã quốc gia đó.

Phương pháp chọn đường liên miền không phân lớp CIDR

- Router nhận được cuộc gọi nhìn vào phần đầu của số điện thoại, mã tỉnh, để vạch đường cho cuộc gọi đến một mạng con tương ứng với mã tỉnh đó, và cứ như thế.
- Trong sơ đồ này, các router đường trục chỉ lưu giữ thông tin về mã quốc gia cho mỗi mục từ trong bảng chọn đường của mình, mỗi mục từ này đại diện cho một số khổng lồ các số điện thoại riêng lẻ chứ không phải là một số điện thoại cụ thể.

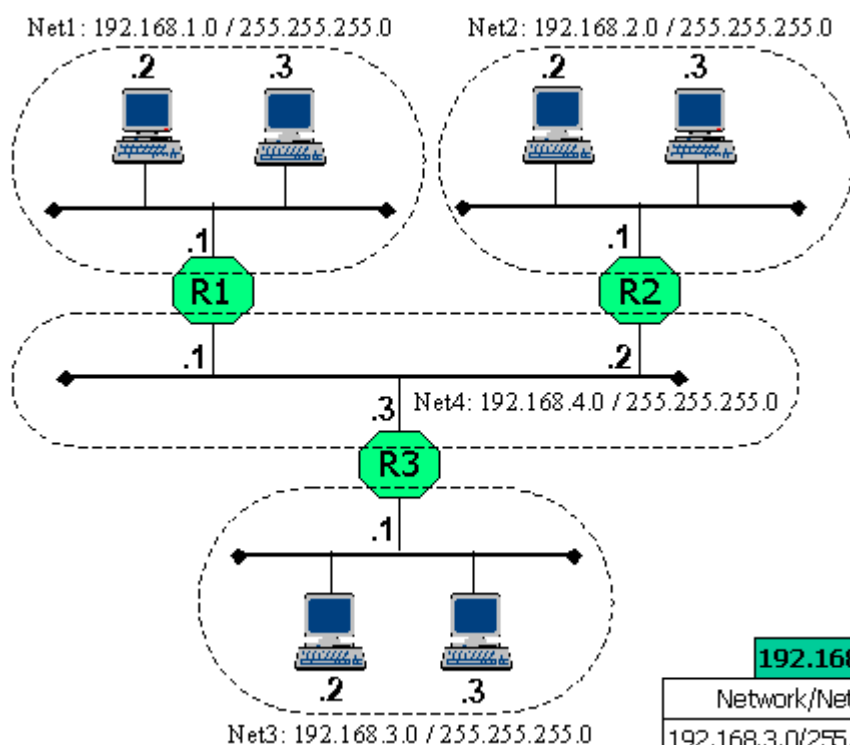
Phương pháp chọn đường liên miền không phân lớp CIDR

- Thông thường, các khối địa chỉ lớn được cấp cho các nhà cung cấp dịch vụ Internet (IP-Internet Service Providers) lớn, sau đó họ lại cấp lại các phần trong khối địa chỉ của họ cho các khách hàng của mình.
- Hiện tại, mạng Internet sử dụng cả hai sơ đồ cấp phát địa chỉ Classfull và Classless (CIDR). Hầu hết các router mới đều hỗ trợ CIDR và những nhà quản lý Internet thì khuyến khích người dùng sử dụng CIDR.

Phương pháp chọn đường liên miền không phân lớp CIDR

- Tham khảo thêm về CIDR ở địa chỉ
- <http://www.rfc-editor.org/rfcsearch.html> với các RFC liên quan sau:
 - RFC 1517: Applicability Statement for the Implementation of CIDR
 - RFC 1518: An Architecture for IP Address Allocation with CIDR
 - RFC 1519: CIDR: An Address Assignment and Aggregation Strategy
 - RFC 1520: Exchanging Routing Information Across Provider Boundaries in the CIDR Environment

Chọn đường trong giao thức IP



192.168.3.3 - Routing table		
Network/Netmask	NextHop	Interface
192.168.3.0/255.255.255.0	local	local
default	192.168.3.1	local

R1-Routing table		
Network/Netmask	NextHop	Interface
192.168.1.0/255.255.255.0	local	local
192.168.2.0/255.255.255.0	192.168.4.2	192.168.4.1
192.168.3.0/255.255.255.0	192.168.4.3	192.168.4.1
192.168.4.0/255.255.255.0	local	local

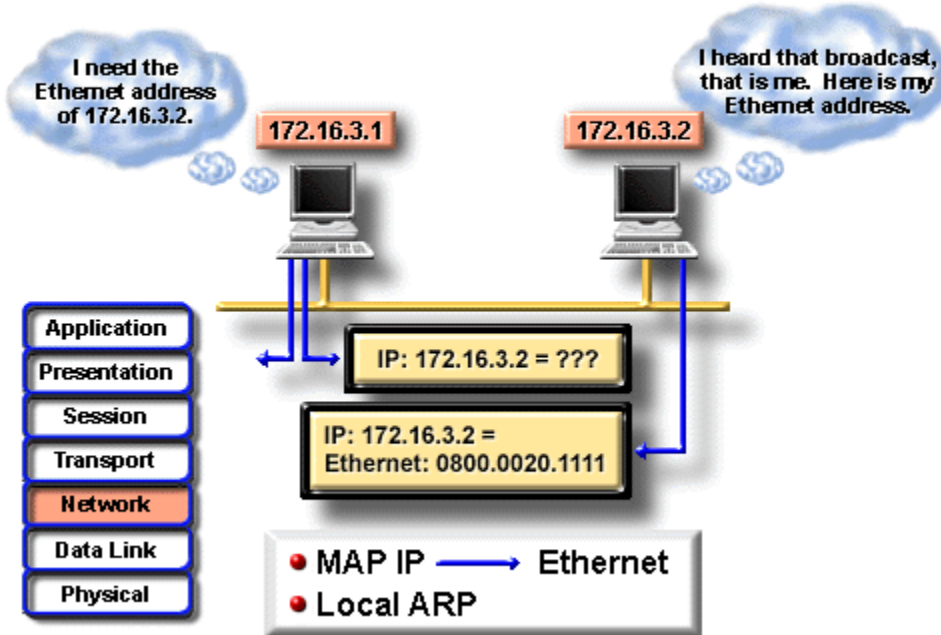
R2-Routing table		
Network/Netmask	NextHop	Interface
192.168.1.0/255.255.255.0	192.168.4.1	192.168.4.2
192.168.2.0/255.255.255.0	local	local
192.168.3.0/255.255.255.0	192.168.4.3	192.168.4.2
192.168.4.0/255.255.255.0	local	local

R3-Routing table		
Network/Netmask	NextHop	Interface
192.168.1.0/255.255.255.0	192.168.4.1	192.168.4.3
192.168.2.0/255.255.255.0	192.168.4.2	192.168.4.3
192.168.3.0/255.255.255.0	local	local
192.168.4.0/255.255.255.0	local	local

Giao thức phân giải địa chỉ (ARP - Address Resolution Protocol)

- Dùng để tìm địa chỉ MAC (physical address) từ một địa chỉ IP (phân giải từ địa chỉ IP sang địa chỉ MAC).

Address Resolution Protocol (ARP)



Giao thức phân giải địa chỉ ngược RARP

- Giao thức RARP (Reverse Address Resolution Protocol) dùng để phân giải ngược từ địa chỉ MAC sang địa chỉ IP.
- Dùng trong các hệ thống trạm làm việc không đĩa cứng (Diskless workstation). Các máy trạm cần có một địa chỉ IP để giao tiếp với server.
- Trên server duy trì một bảng mô tả mối tương quan giữa địa chỉ vật lý và địa chỉ IP của các máy trạm.
- Khi nhận được yêu cầu RARP, server tìm trong bảng địa chỉ và trả về địa chỉ IP tương ứng cho máy trạm đã gửi yêu cầu.

Giao thức thông điệp điều khiển Internet ICMP

- ICMP (Internet Control Message Protocol)
- Các thông điệp của giao thức được gói đi trong các gói tin IP và được dùng để gửi đi các báo lỗi hay các thông tin điều khiển.

Giao thức thông điệp điều khiển Internet ICMP

- ICMP tạo ra nhiều loại thông điệp hữu ích như :
 - Đích đến không tới được (Destination Unreachable),
 - Thăm hỏi và trả lời (Echo Request and Reply) **[lệnh ping]**
 - Chuyển hướng (Redirect),
 - Vượt quá thời gian (Time Exceeded),
 - Quảng bá bộ chọn đường (Router Advertisement)
 - Cô lập bộ chọn đường (Router Solicitation)
 -