

Chương 5:

# **MẠNG THÔNG TIN DỮ LIỆU**

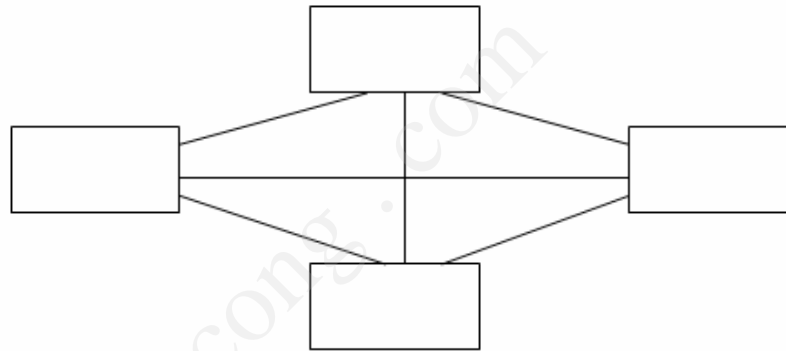
# NỘI DUNG

---

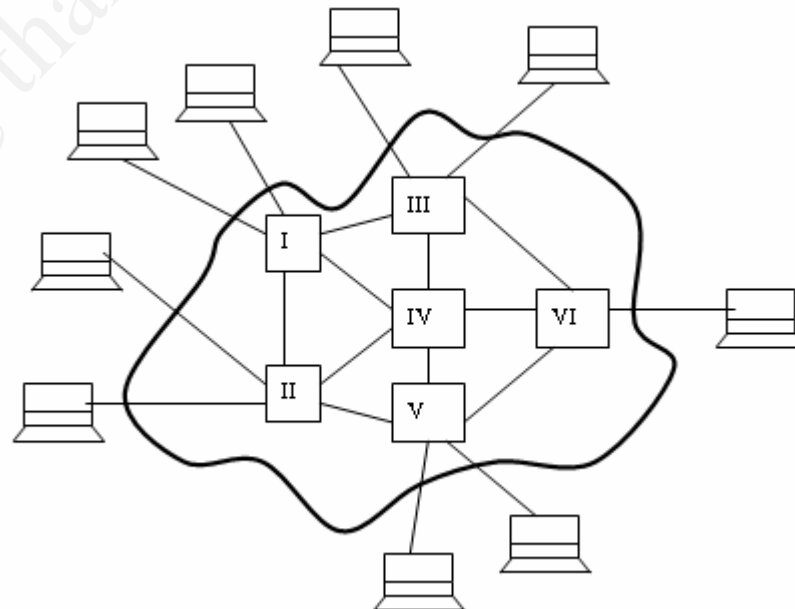
- ◆ Chuyển mạch
- ◆ Định tuyến (routing)
- ◆ X.25
- ◆ Frame Relay
- ◆ ISDN
- ◆ ATM
- ◆ SONET/SDH

## 5.1 CHUYỂN MẠCH

Không dùng  
chuyển mạch: tốn  
kém, không hiệu  
quả



Sử dụng chuyển  
mạch

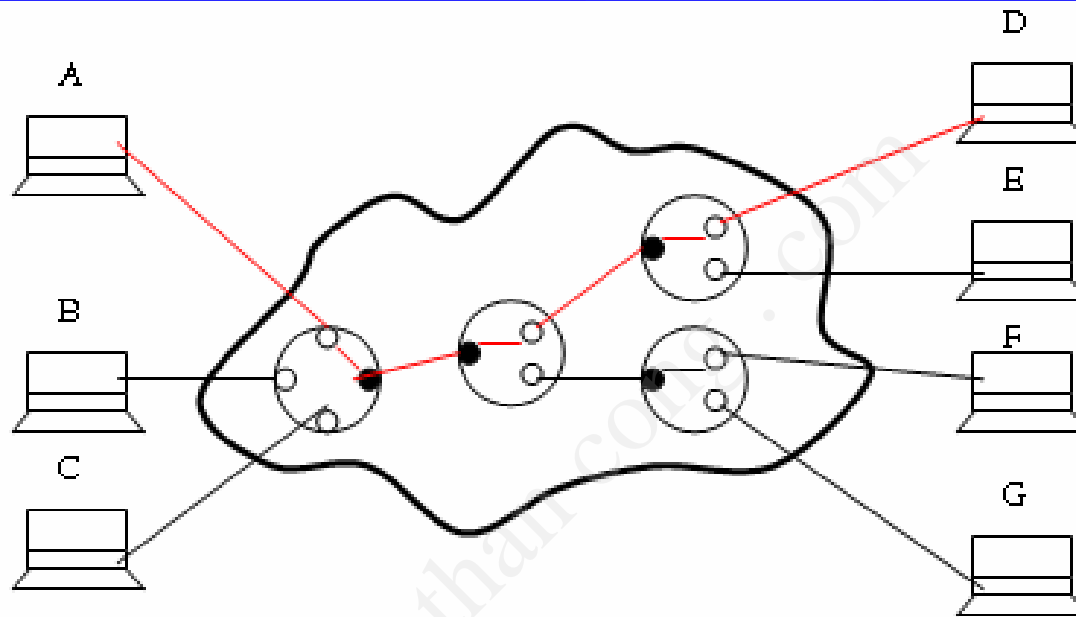


# CHUYỂN MẠCH

---

- ◆ Mạng chuyển mạch gồm một chuỗi các bộ chuyển mạch được kết nối với nhau
- ◆ Bộ chuyển mạch là các thiết bị (gồm cả phần cứng và mềm) có khả năng tạo các kết nối tạm thời giữa 2 hay nhiều thiết bị được kết nối đến bộ chuyển mạch.
- ◆ Các phương pháp chuyển mạch
  - Chuyển mạch mạch (circuit switching)
  - Chuyển mạch gói (packet switching)
  - Chuyển mạch thông điệp (message switching)

# CHUYỂN MẠCH MẠCH

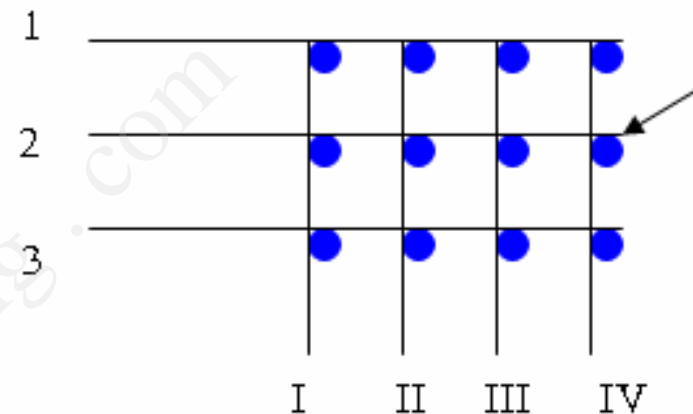


- Tạo kết nối vật lý trực tiếp giữa 2 thiết bị
- Sử dụng trong mạng điện thoại công cộng PSTN
- Kênh truyền khi đã kết nối được **dành riêng** cho cuộc gọi.
- Trạm nguồn và trạm đích phải hoạt động ở cùng tốc độ
- Dữ liệu truyền đúng thứ tự.
- Delay thấp (constant) -> realtime
- Cước phí được xác định tương ứng **thời lượng** user chiếm kênh truyền, không dựa vào khối lượng dữ liệu truyền qua kênh truyền
- Chuyển mạch mạch không phù hợp với dữ liệu phi thoại, có khuynh hướng bùng phát

# BỘ CHUYỂN MẠCH KHÔNG GIAN

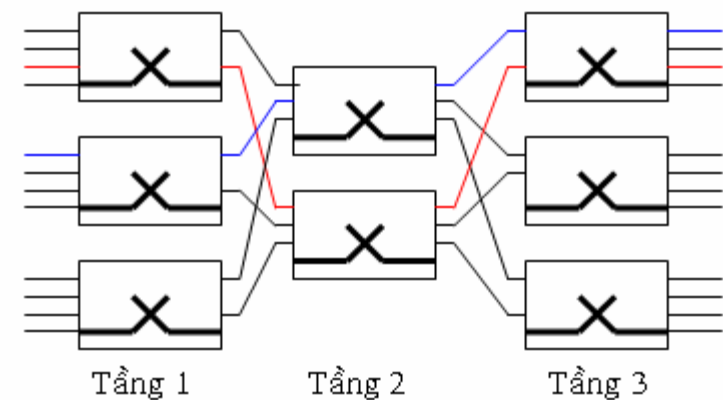
## Bộ chuyển mạch ngang dọc

- Nối  $m$  ngõ vào và  $n$  ngõ ra cần  $n \times m$  giao điểm (các khóa điện tử)
- Không bị nghẽn

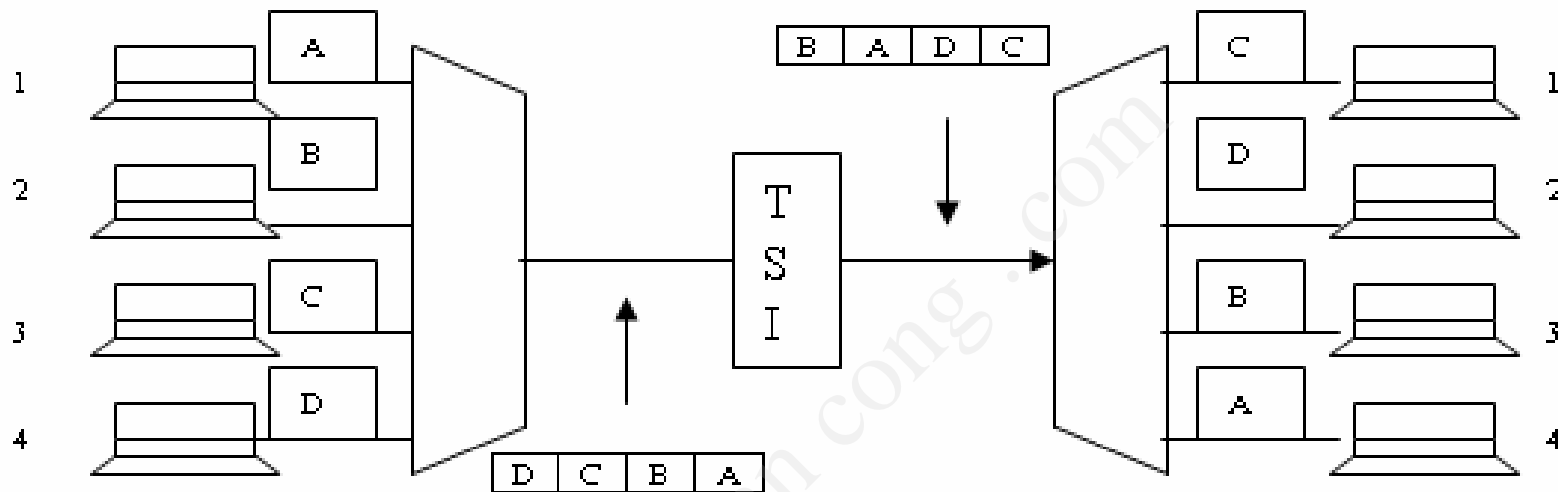


## Bộ chuyển mạch đa tầng

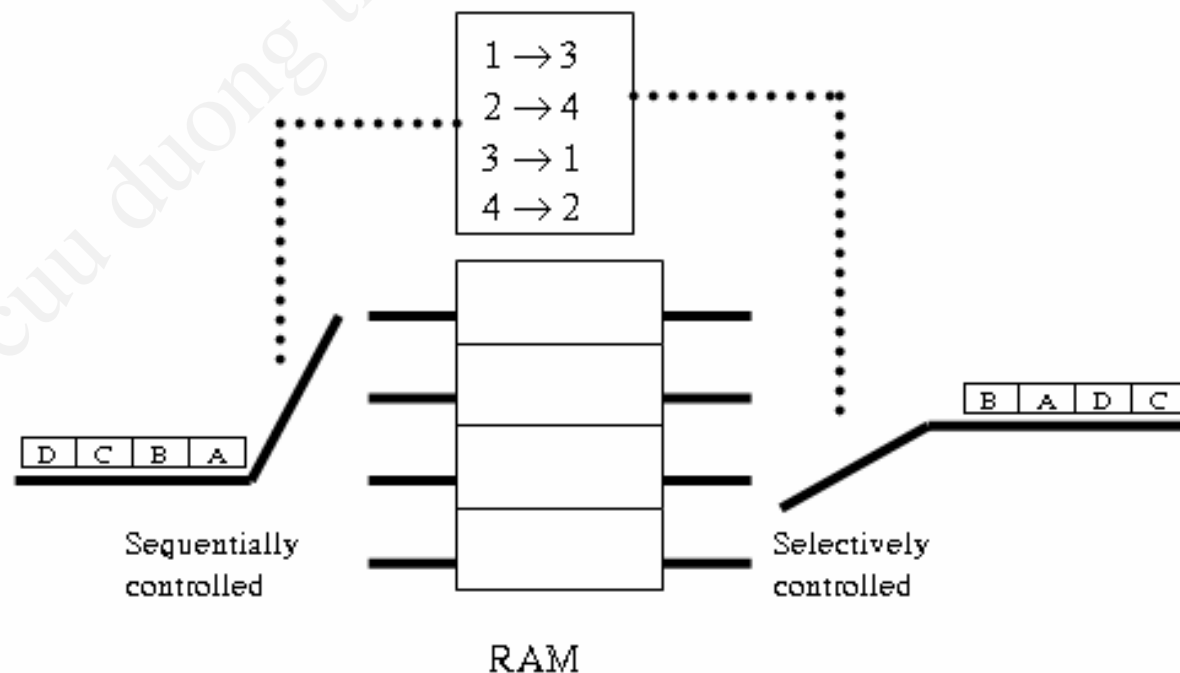
- Số tiếp điểm giảm so với chuyển mạch ngang dọc
- Có thể bị nghẽn (đầu vào không thể nối đến đầu ra vì không còn đường kết nối)
- Số tiếp điểm sẽ là tổng số tiếp điểm của tất cả các tầng



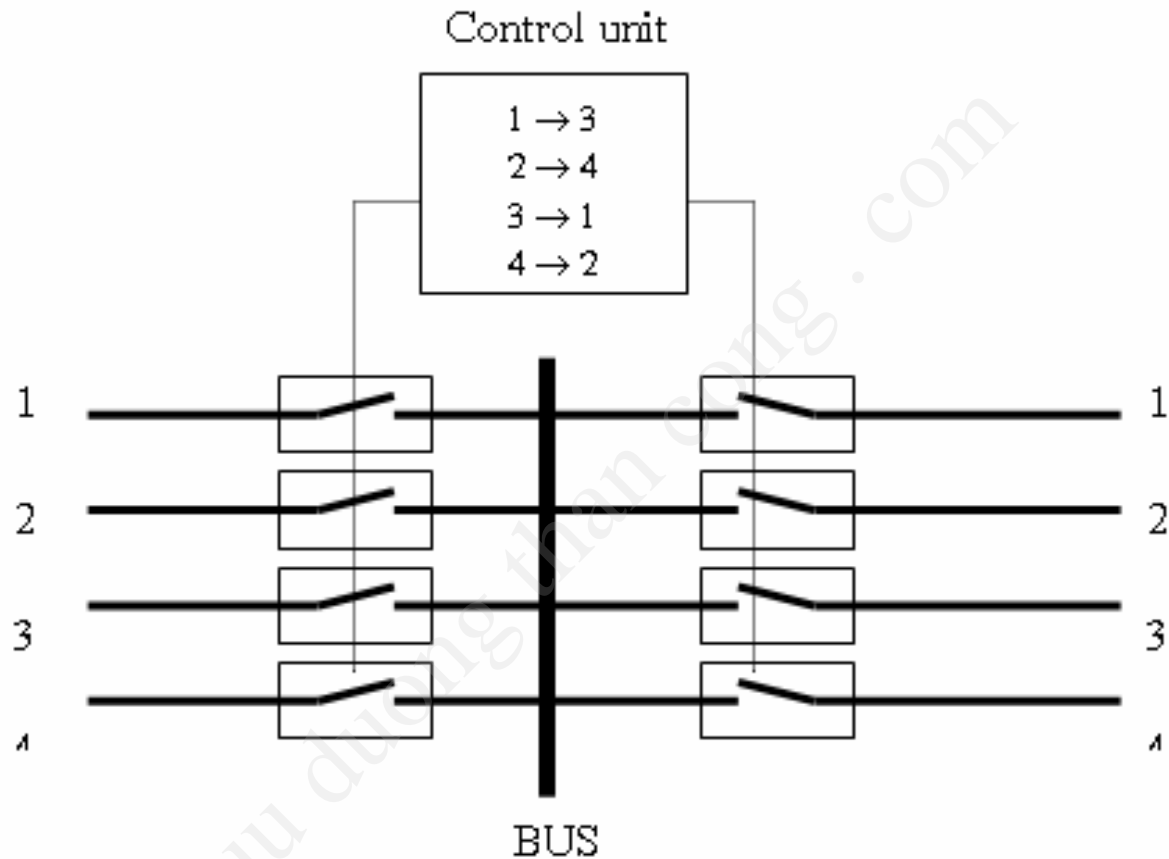
# BỘ CHUYỂN MẠCH THỜI GIAN



**TSI:** Time Slot  
Interchange



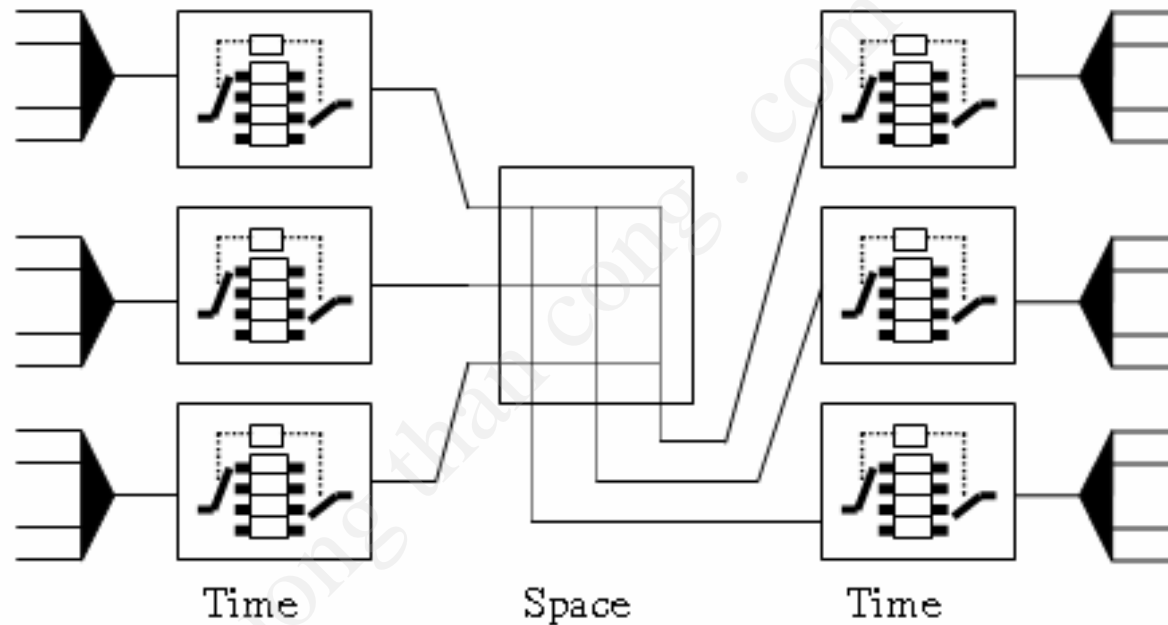
# BỘ CHUYỂN MẠCH THỜI GIAN



Bộ chuyển mạch TDM bus



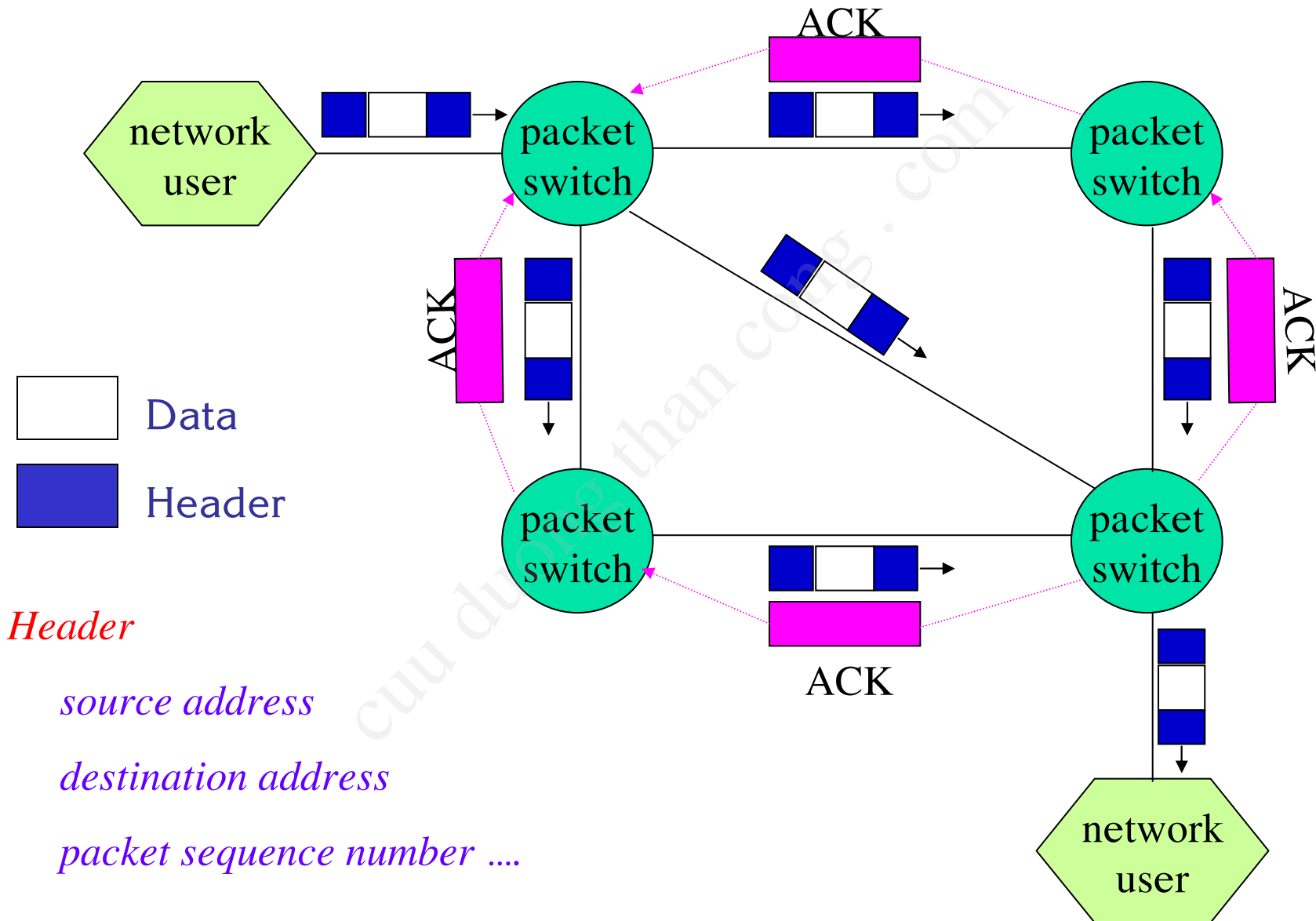
# **BỘ CHUYỂN MẠCH KẾT HỢP**



Bộ chuyển mạch TST

Giảm số tiếp điểm và thời gian trễ

# CHUYỂN MẠCH GÓI



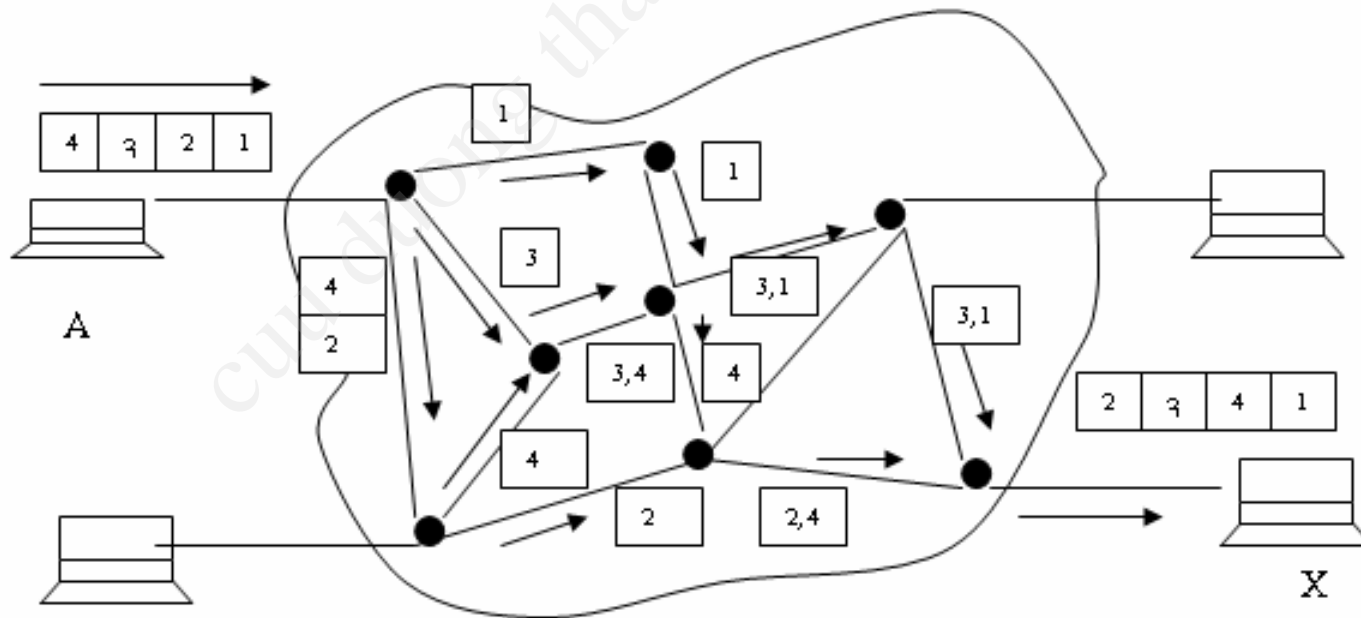
# CHUYỂN MẠCH GÓI

---

- Kênh truyền được khai thác tốt hơn so với chuyển mạch mạch
- Dữ liệu được phân thành các gói (packet), kích thước tối đa mỗi gói phụ thuộc vào mạng
- Các gói được truyền trên mạng theo kiểu store and forward từ nút này sang nút khác.
- Tại mỗi nút trung gian, gói dữ liệu được lưu lại thời gian ngắn rồi chuyển đi tiếp theo thông tin của header
- Thời gian trễ lớn hơn so với chuyển mạch mạch
- Có 2 loại chuyển mạch gói là **datagram** và **chuyển mạch ảo** (virtual circuit)

# DATAGRAM

- Các gói (datagram) được xem xét một cách độc lập, mạng sẽ xử lý gói như một thực thể độc lập
- Các gói có thể đến đích không đúng thứ tự (sẽ được lớp transport sắp xếp lại)
- Tại mỗi nút trung gian, gói dữ liệu được lưu lại thời gian ngắn rồi chuyển đi tiếp theo thông tin của header
- Có 2 loại chuyển mạch gói là **datagram** và **chuyển mạch ảo** (virtual circuit)

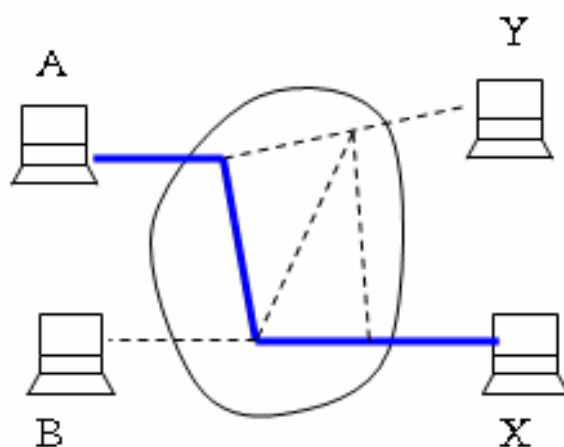


# **MẠCH ẢO (VIRTUAL CIRCUIT)**

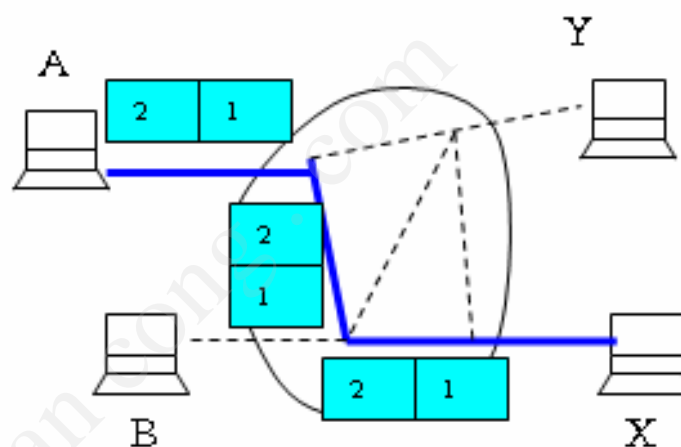
---

- Một kết nối được thiết lập giữa phía gửi và phía nhận trước khi gửi dữ liệu
- Các gói được truyền đến phía thu theo cùng kết nối đã thiết lập -> đến phía thu đúng thứ tự
- Mạch ảo được thiết lập trước khi thực sự truyền data, sau khi việc truyền data kết thúc, mạch ảo được giải tỏa
- Có 2 loại mạch ảo là mạch ảo chuyển mạch (SVC Switched Virtual Circuit) và mạch ảo thường xuyên (PVC Permanent Virtual Circuit)
- Mạch ảo chuyển mạch: mạch ảo chỉ là kết nối tạm thời trong quá trình truyền dữ liệu
- Mạch ảo thường xuyên: kết nối thường trực giữa 2 trạm

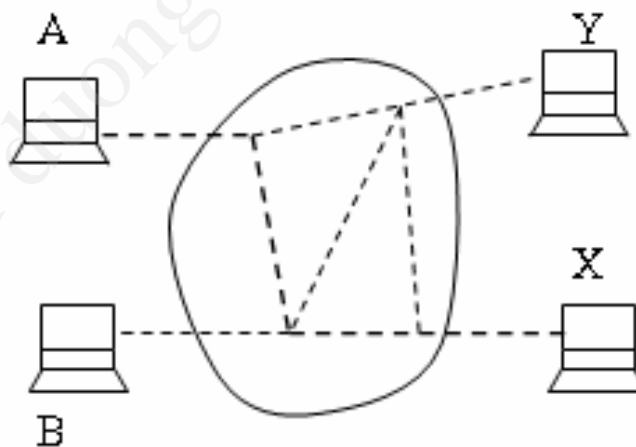
# MẠCH ẢO (VIRTUAL CIRCUIT)



(a) Thiết lập kết nối

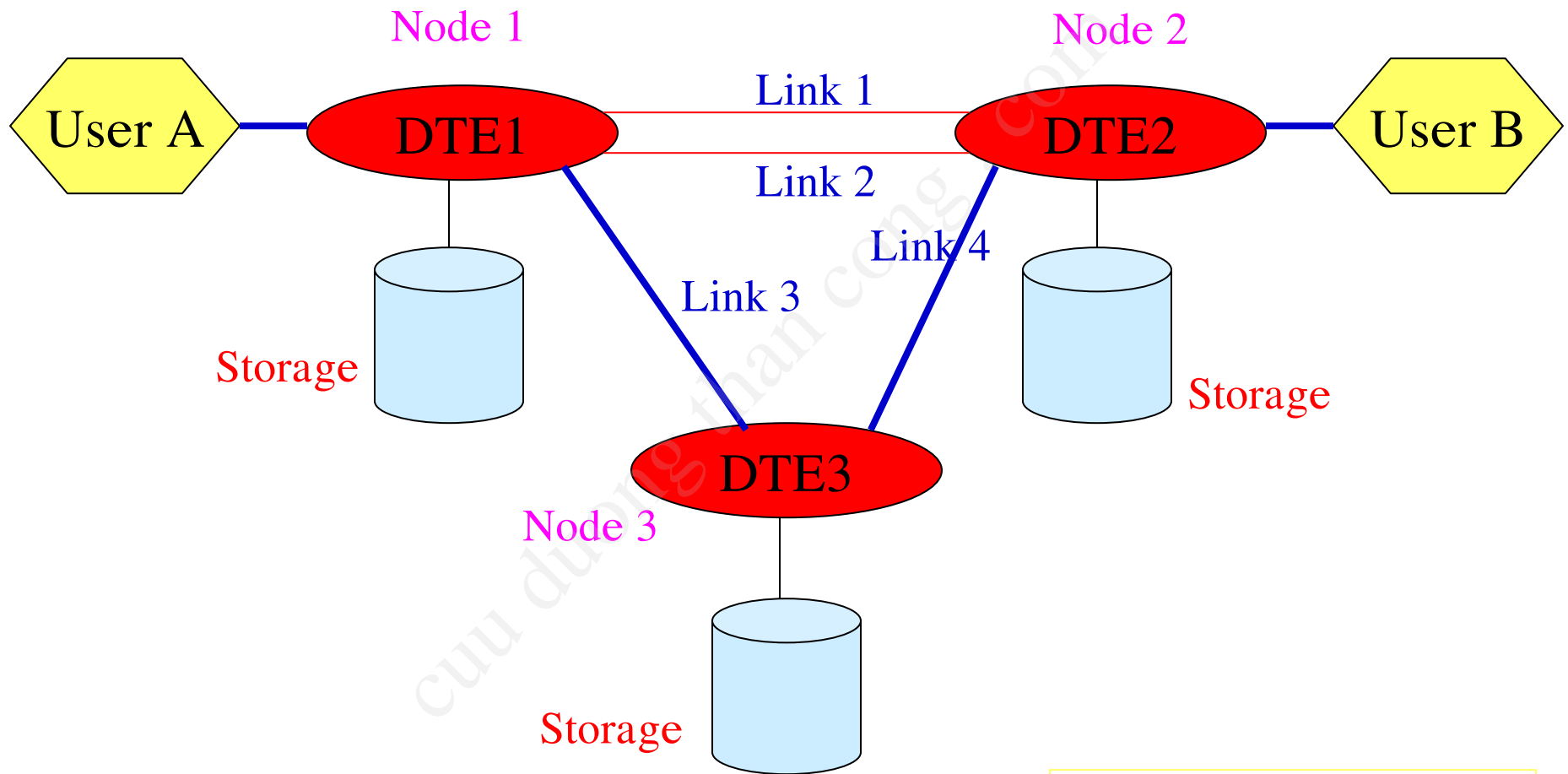


(b) Chuyển dữ liệu



(c) Giải tỏa kết nối

# MESSAGE-SWITCHING

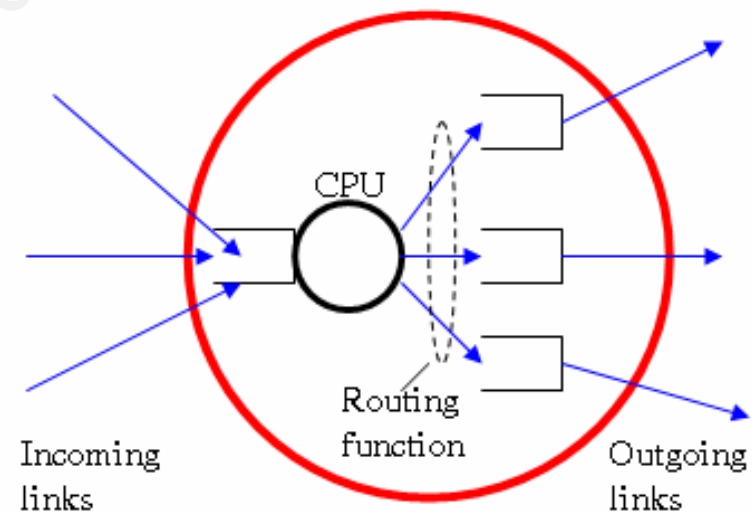


Hình: Chuyển mạch thông điệp

Store and forward

# ĐỊNH TUYẾN (ROUTING)

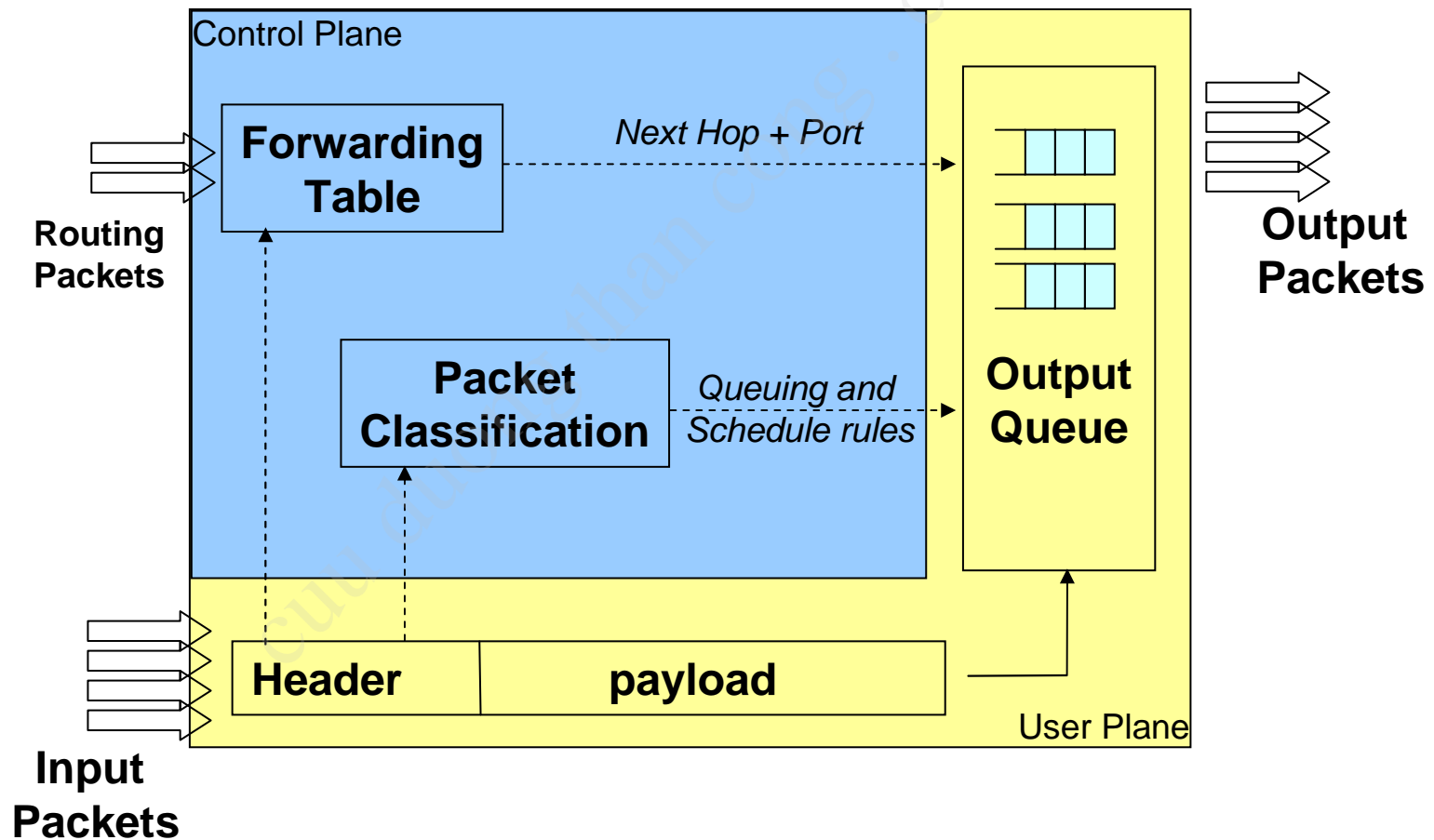
- Các nút chuyển mạch trong mạng chuyển mạch gói có nhiệm vụ nhận các gói dữ liệu từ nguồn chuyển đến trạm đích
- Khi các gói dữ liệu đi vào nút chuyển mạch, chúng được kiểm tra bởi CPU của nút (**địa chỉ đích lớp mạng của gói**). Dựa vào đó gói sẽ đưa đến hàng đợi của ngõ ra thích hợp. Chức năng này được gọi là **định tuyến** (routing)
- Việc định tuyến tại mỗi nút sẽ gây ra thời gian trễ
  - ❑ Trễ do xếp hàng trong CPU và hàng đợi liên kết ra
  - ❑ Trễ do thời gian xử lý của CPU
  - ❑ Trễ do thời gian truyền gói
- Lưu ý sự khác nhau giữa chuyển mạch gói và chuyển mạch mạch



Hình: Nút chuyển mạch



# ĐỊNH TUYẾN (ROUTING)



- Định tuyến
  - Nhận gói dữ liệu từ nguồn và phân phối đến đích
- Nút chuyển mạch
  - Thực hiện việc chuyển mạch
- Định tuyến
  - Định tuyến không bảng:
    - Ngẫu nhiên
    - Nguồn
    - Tính toán
  - Định tuyến theo bảng

# Định tuyến theo bảng

- Xác định con đường đến đích tối ưu
- Khi tính toán tối ưu, có thể xem xét đến:
  - Tổng thời gian truyền
  - Thời gian xử lý và xếp hàng tại mỗi nút
  - Mức độ ưu tiên của mỗi gói
  - Hoạch định của admin
- Thủ tục định tuyến
  - Tính chất:
    - Tĩnh & động
    - Tập trung & phân bố
  - Thủ tục tập trung: Shortest Forward Path Tree (SFPT), Shortest Backward Path Tree (SBPT)
  - Thủ tục phân bố: Thủ tục trao đổi khoảng cách

# THUẬT TOÁN SFPT

## (SHORT FORWARD PATH FIRST)

### ■ **Giới thiệu thuật toán SPF :**

- Xác định đường đi ngắn nhất từ 1 nguồn tới tất cả các nút trên mạng
- Vị trí trung tâm cần phải thực hiện việc tính toán này một lần cho mỗi node để xác định con đường ngắn nhất tới mỗi node còn lại

# THUẬT TOÁN SFPT

## (SHORT FORWARD PATH FIRST)

### ■ **Mô tả thuật toán SPF :**

- Trong cây Shortest Path First (SPF), Dijkstra đã định nghĩa ra ba loại nhánh : I, II và III. Và trong router, sẽ có ba database đại diện cho tập ba loại nhánh đó, gồm có :
  - Tree database : đại diện cho nhánh loại I. Những link (nhánh) nào được thêm vào cây SPF sẽ được thêm vào đây. Khi thuật toán SPF hoàn tất, thì database này chính là toàn bộ cây SPF.
  - Candidate database : database này đại diện cho nhánh loại II. Các link được copy từ link state database sang database này theo một trật tự nhất định. Các link trong database này sẽ lần lượt được khảo sát để thêm vào cây SPF.
  - Link state database : chứa tất cả các link. Đại diện cho loại nhánh III.
- Ngoài ra Dijkstra cũng định nghĩa ra hai loại node, A và B. Loại A, bao gồm tất cả các router nối với các link trong cây SPF. Loại B, gồm tất cả các router khác. Khi thuật toán SPF kết thúc thì các router loại B sẽ không còn.

# THUẬT TOÁN SFPT

## (SHORT FORWARD PATH FIRST)

Thuật toán SPF của Dijkstra như sau :

- **Bước 1** : Chọn một router làm gốc.
- **Bước 2** : tất cả các link nối giữa router gốc với neighbor của nó sẽ được đưa vào candidate database.
- **Bước 3** : cost của mỗi link trong candidate database sẽ được tính toán. Link nào có cost nhỏ nhất sẽ được thêm vào Tree database. Nếu như có hai link có cost nhỏ nhất bằng nhau thì sẽ chọn một.
- **Bước 4** : router neighbor nối với đường link đó sẽ được kiểm tra. Tất cả các đường link nối với router đó sẽ được đưa vào candidate database, ngoài trừ đường link đã có trong Tree database.
- **Bước 5** : Nếu vẫn còn giá trị trong candidate database thì thuật toán sẽ quay lại bước 3 thực hiện tiếp. Nếu không còn, thì dừng thuật toán.

# THUẬT TOÁN SFPT

## (SHORT FORWARD PATH FIRST)

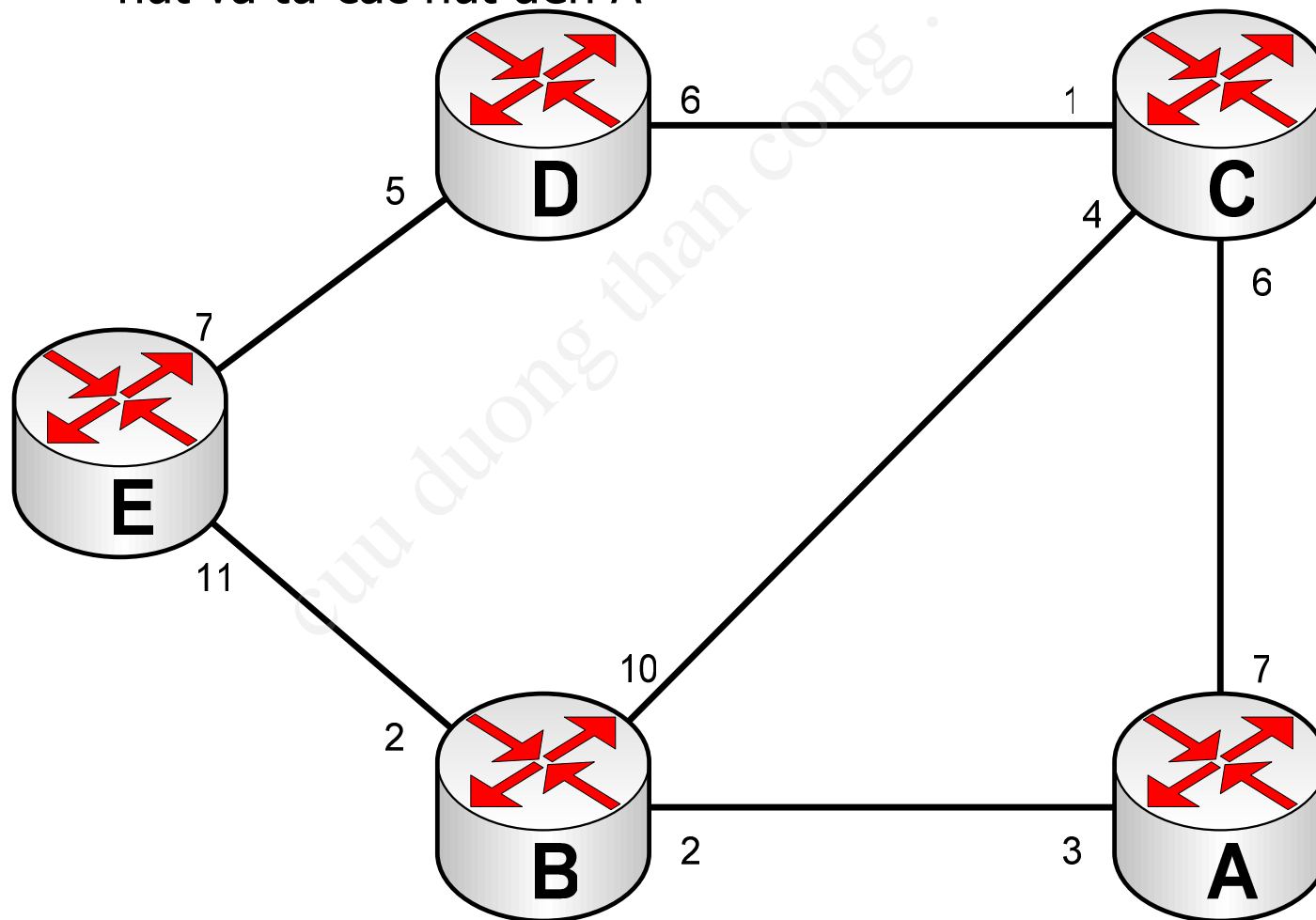
### Giải thuật :

$N = \{s\}$  *(với  $s$  là nút nguồn)*  
For all node  $v \neq s$   
Begin  
     $\epsilon_{sv} = C_{sv}$   
If  $\epsilon_{sv} < \infty$  then  $L_{sv} = (s, v)$   
End  
Do while *( $N$  không chứa tất cả các node)*  
    Find  $w \notin N$  for which  $\epsilon_{sw} = \min \epsilon_{sv}$   
     $N = N \cup \{w\}$  *( $N$  hợp với  $w$ , tức bổ sung thêm  $w$  vào  $N$ )*  
For all  $v \notin N$   
Begin  
    Temp  $= \epsilon_{sv}$   
     $\epsilon_{sv} = \min(\epsilon_{sv}, \epsilon_{sw} + \epsilon_{wv})$   
    if  $\epsilon_{sv} < \text{temp}$  then  
         $L := L_{sw} \cup \{v\}$   
End  
End

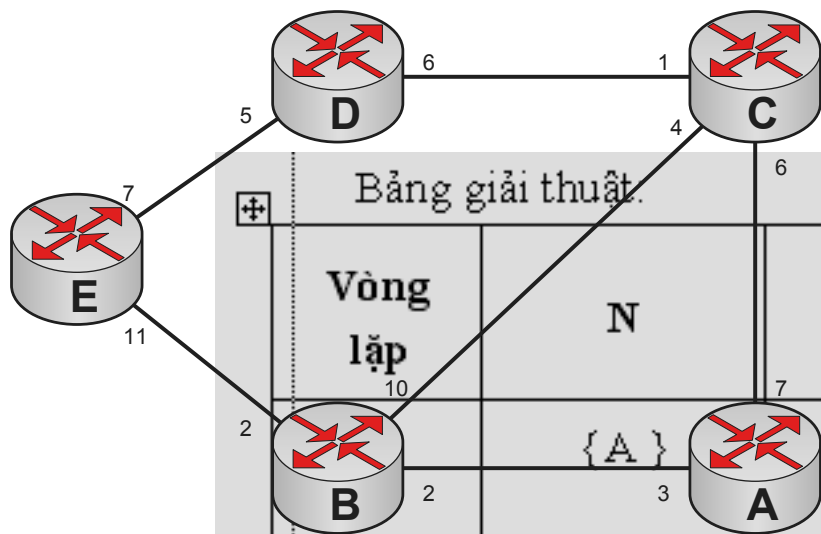
# THUẬT TOÁN SFPT

## (SHORT FORWARD PATH FIRST)

- **Ví dụ thuật toán** : Tìm đường đi ngắn nhất từ A đến tất cả các nút và từ các nút đến A





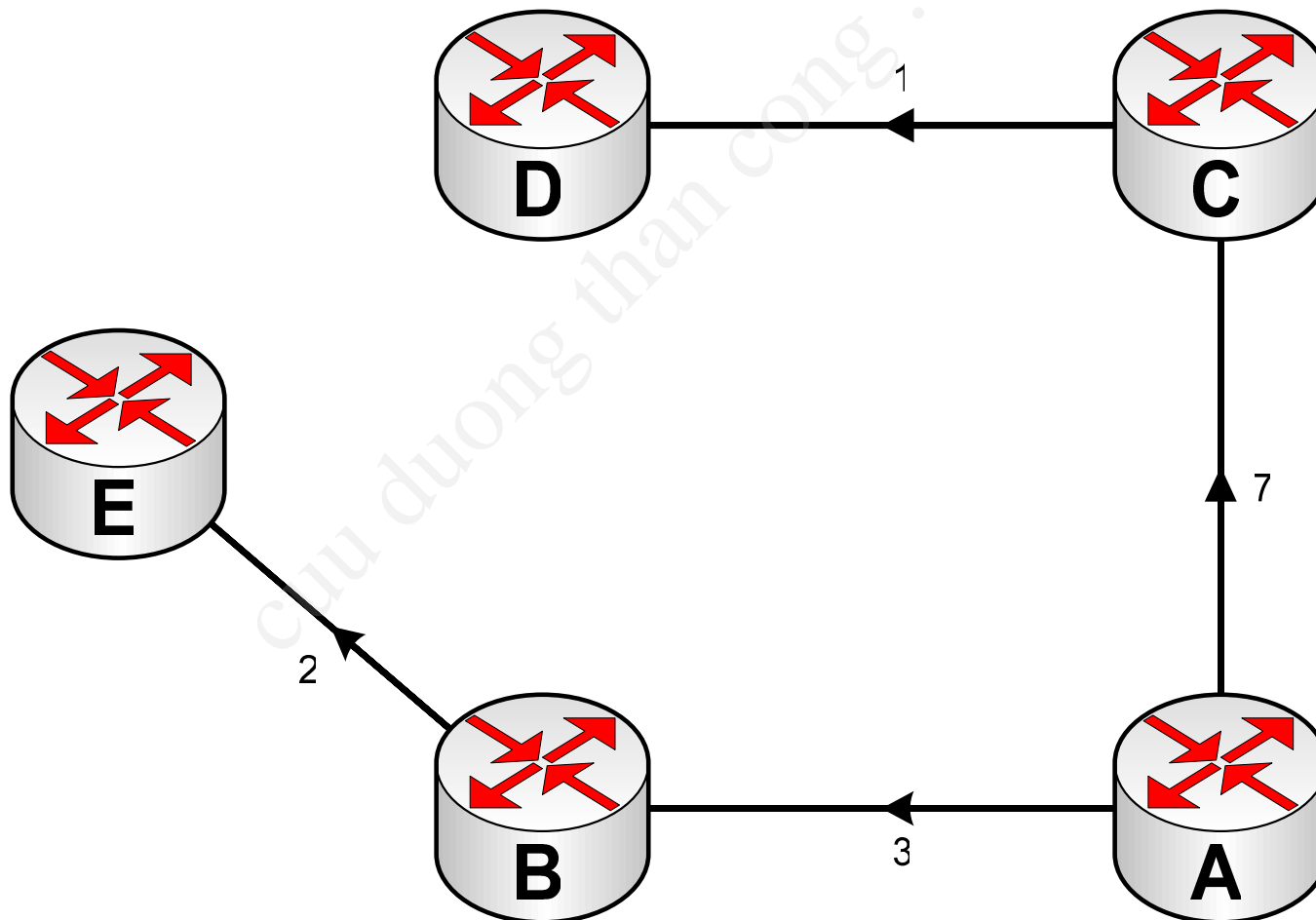


**Bảng giải thuật.**

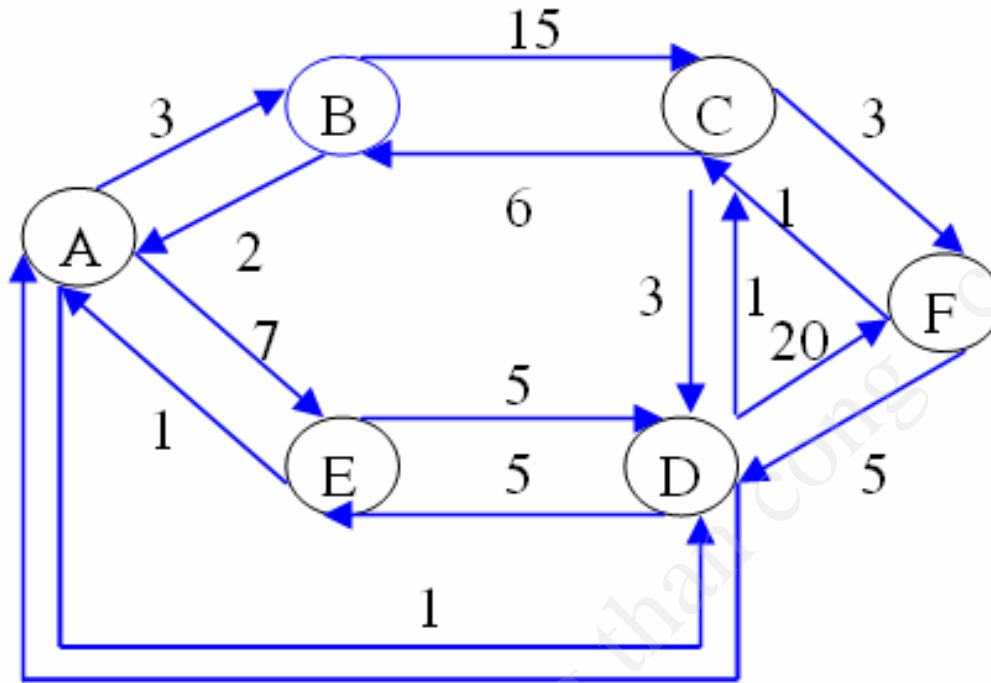
Vòng lặp	N	$L_{AB}$ $\epsilon_{AB}$	$L_{AC}$ $\epsilon_{AC}$	$L_{AD}$ $\epsilon_{AD}$	$L_{AE}$ $\epsilon_{AE}$
0	{A}	{A,B} 3	{A,C} 7	- $\infty$	- $\infty$
1	{A,B}	{A,B} 3	{A,C} 7	- $\infty$	{A,B,E} 5
2	{A,B,E}	{A,B} 3	{A,C} 7	{A,B,E,D} 12	{A,B,E} 5
3	{A,B,E,C}	{A,B} 3	{A,C} 7	{A,C,D} 8	{A,B,E} 5
4	{A,B,E,D,C}	{A,B} 3	{A,C} 7	{A,C,D} 8	{A,B,E} 5

# THUẬT TOÁN SFPT (SHORT FORWARD PATH FIRST)

Kết quả con đường đi ngắn nhất từ A đến các nút còn lại



# **SFPT (SHORT FORWARD PATH FIRST)**



**Hình: Sơ đồ mạng**

$P_{XY}$  đường đi từ X đến Y

$C_{XY}$  chi phí từ X đến Y

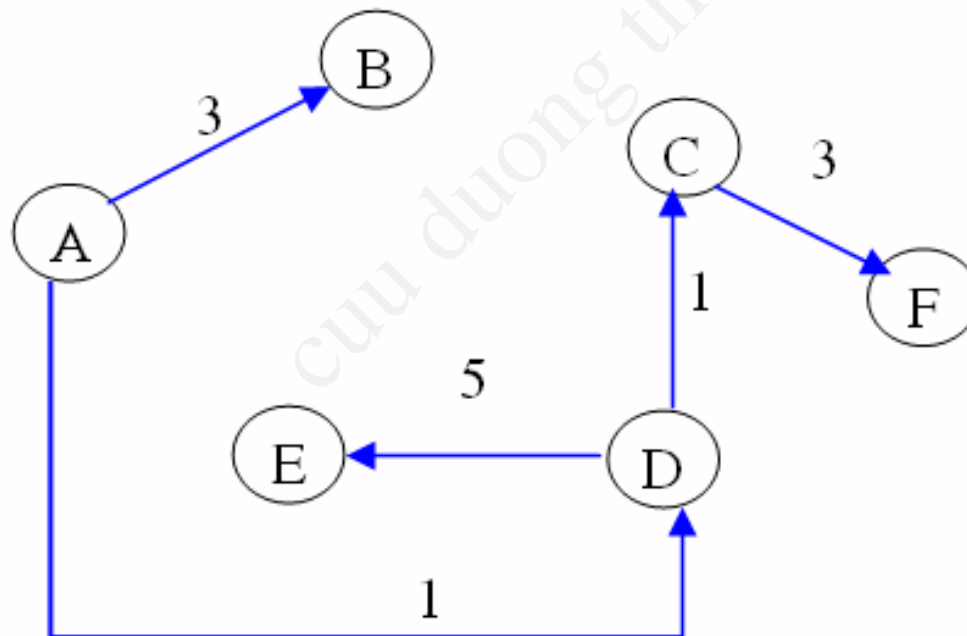
Iter.	N	$P_{AB}$	$C_{AB}$	$P_{AC}$	$C_{AC}$	$P_{AD}$	$C_{AD}$	$P_{AE}$	$C_{AE}$	$P_{AF}$	$C_{AF}$
0	{A}	(A,B)	3	-	$\infty$	(A,D)	1	(A,E)	7	-	$\infty$
1	{A,D}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,F)	21
2	{A,D,C}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5
3	{A,D,C,B}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5
4	{A,D,C,B,F}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5
5	{A,D,C,B,F,E}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5

**Hình: Tính toán SFPT bắt đầu từ nút nguồn A**

# SFPT (SHORT FORWARD PATH FIRST)

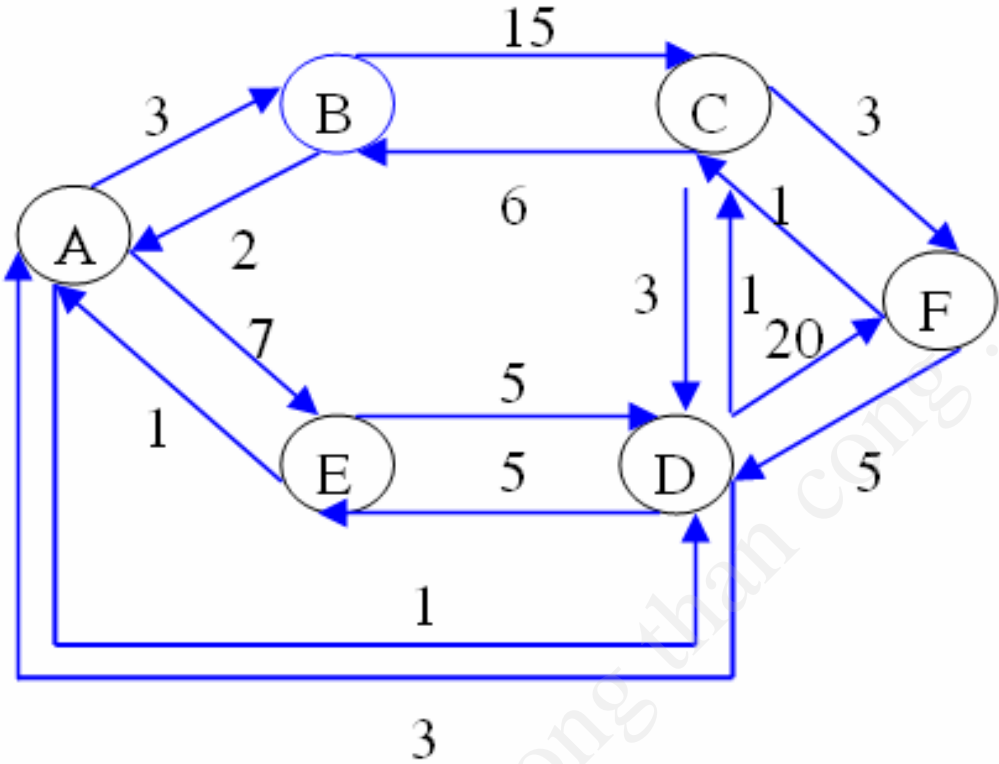
Iter.	N	$P_{AB}$	$C_{AB}$	$P_{AC}$	$C_{AC}$	$P_{AD}$	$C_{AD}$	$P_{AE}$	$C_{AE}$	$P_{AF}$	$C_{AF}$
0	{A}	(A,B)	3	-	$\infty$	(A,D)	1	(A,E)	7	-	$\infty$
1	{A,D}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,F)	21
2	{A,D,C}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5
3	{A,D,C,B}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5
4	{A,D,C,B,F}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5
5	{A,D,C,B,F,E}	(A,B)	3	(A,D,C)	2	(A,D)	1	(A,D,E)	6	(A,D,C,F)	5

Hình: Tính toán SFPT bắt đầu từ nút nguồn A



Hình: Các đường ngắn nhất từ nút A

# SBPT (SHORT BACKWARD PATH FIRST)



Hình: Sơ đồ mạng

Xác định các đường ngắn nhất đến A, cập nhật theo thứ tự B,C,D,E,F

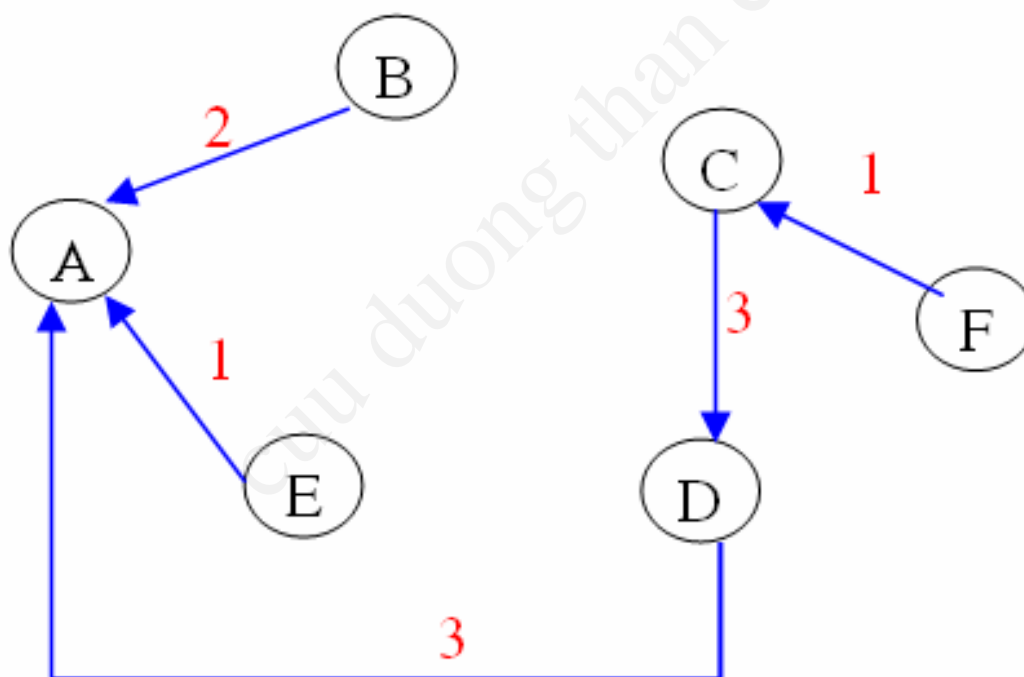
Cycles	A	B	C	D	E	F
0	(●,0)	(●, ∞)	(●, ∞)	(●, ∞)	(●, ∞)	(●, ∞)
1	(●,0)	(A,2)	(B,8)	(A,3)	(A,1)	(D,8)
2	(●,0)	(A,2)	(D,6)	(A,3)	(A,1)	(C,7)
3	(●,0)	(A,2)	(D,6)	(A,3)	(A,1)	(C,7)

Hình: Tính toán SBPT đến nút đích A

# SBPT (SHORT BACKWARD PATH FIRST)

Cycles	A	B	C	D	E	F
0	(●,0)	(●, ∞)	(●, ∞)	(●, ∞)	(●, ∞)	(●, ∞)
1	(●,0)	(A,2)	(B,8)	(A,3)	(A,1)	(D,8)
2	(●,0)	(A,2)	(D,6)	(A,3)	(A,1)	(C,7)
3	(●,0)	(A,2)	(D,6)	(A,3)	(A,1)	(C,7)

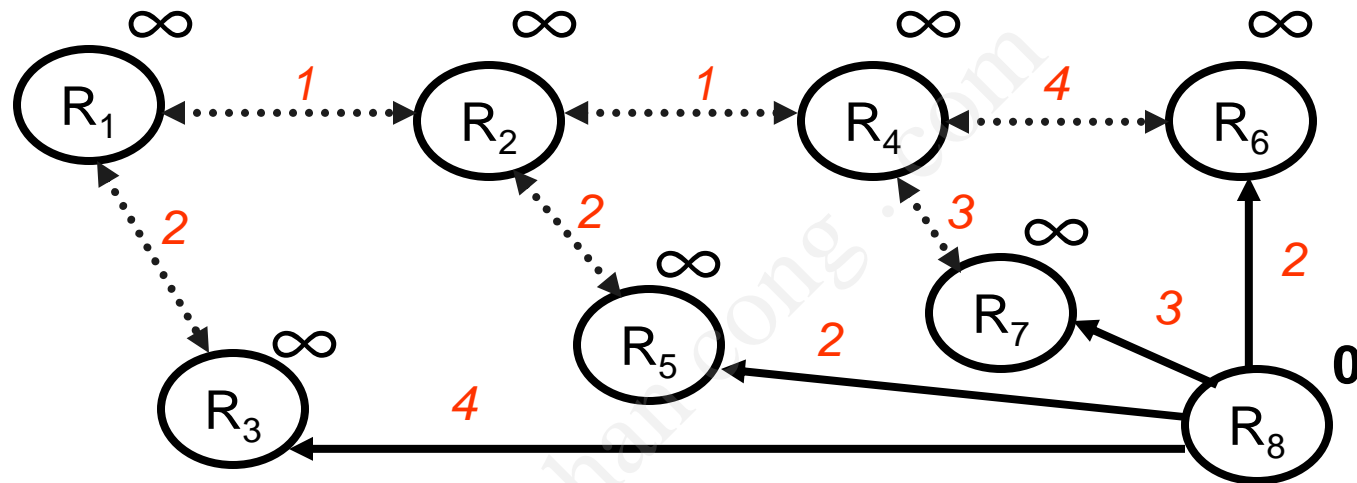
Hình: Tính toán SBPT đến nút đích A



Hình: SBPT đến nút A

# Distributed Bellman-Ford Algorithm

## Example

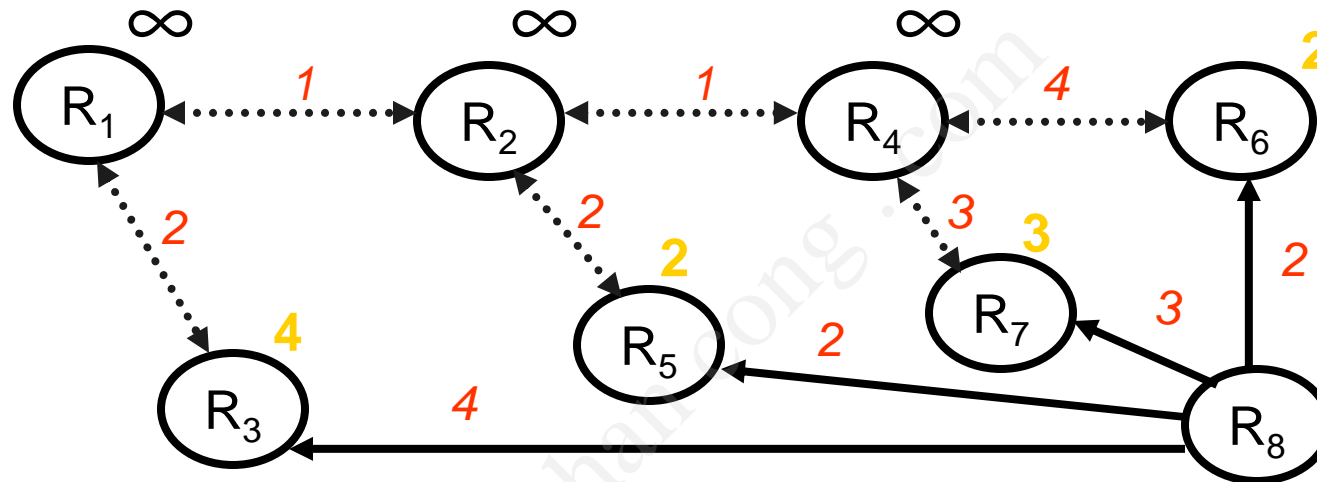


Mỗi node nhận thông tin routing đến một đích đến từ các node lân

Trạng thái khởi động: Tất cả nodes trừ R8 có cost đến R8 là  $\infty$ . R8 đặt đường đi đến chính nó là 0

R <sub>1</sub>	Inf
R <sub>2</sub>	Inf
R <sub>3</sub>	4, R <sub>8</sub>
R <sub>4</sub>	Inf
R <sub>5</sub>	2, R <sub>8</sub>
R <sub>6</sub>	2, R <sub>8</sub>
R <sub>7</sub>	3, R <sub>8</sub>

### Example



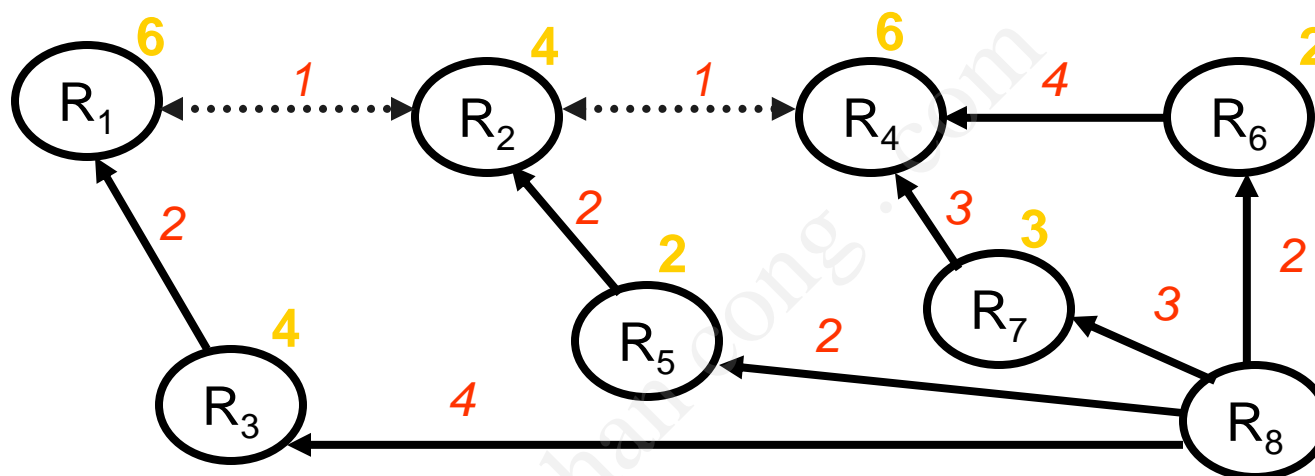
- ❖ Mỗi T seconds (hoặc khi phát hiện ra thay đổi trong bảng routing của mình), mỗi node sẽ thông báo đến những node lân cận giá trị đường đi đến R8
- ❖ Mỗi node sẽ tự cập nhật thông tin về đường ngắn nhất  $\min(\text{current cost}, \text{received cost} + \text{link cost})$
- ❖ Đặt node nào có giá trị nhỏ nhất là node kế tiếp

**Routing tables have both the next-hop and the cost**



$R_1$	6, $R_3$
$R_2$	4, $R_5$
$R_3$	4, $R_8$
$R_4$	6, $R_7$
$R_5$	2, $R_8$
$R_6$	2, $R_8$
$R_7$	3, $R_8$

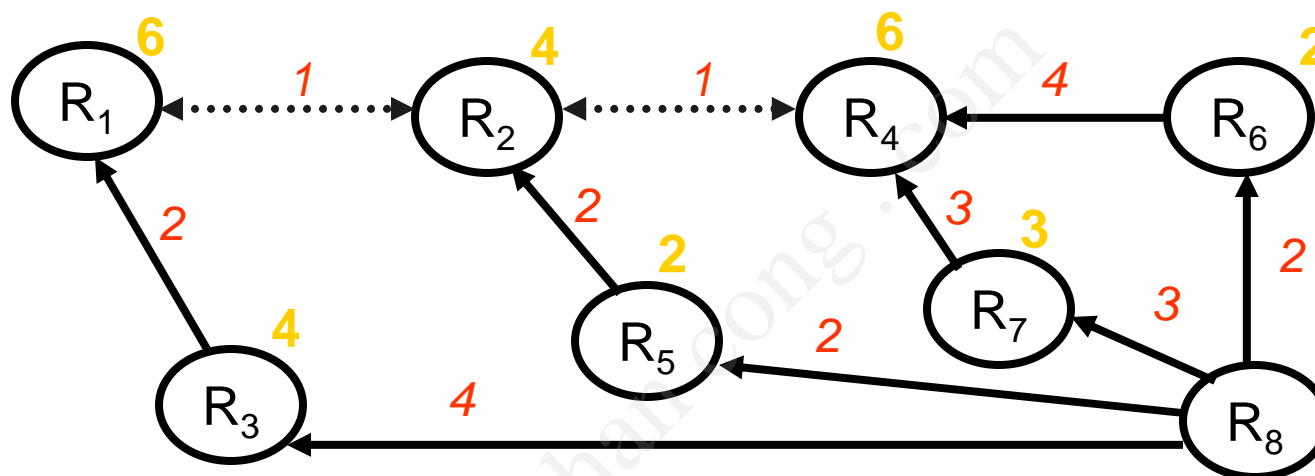
## Example



Lặp lại cho đến khi không có sự thay đổi

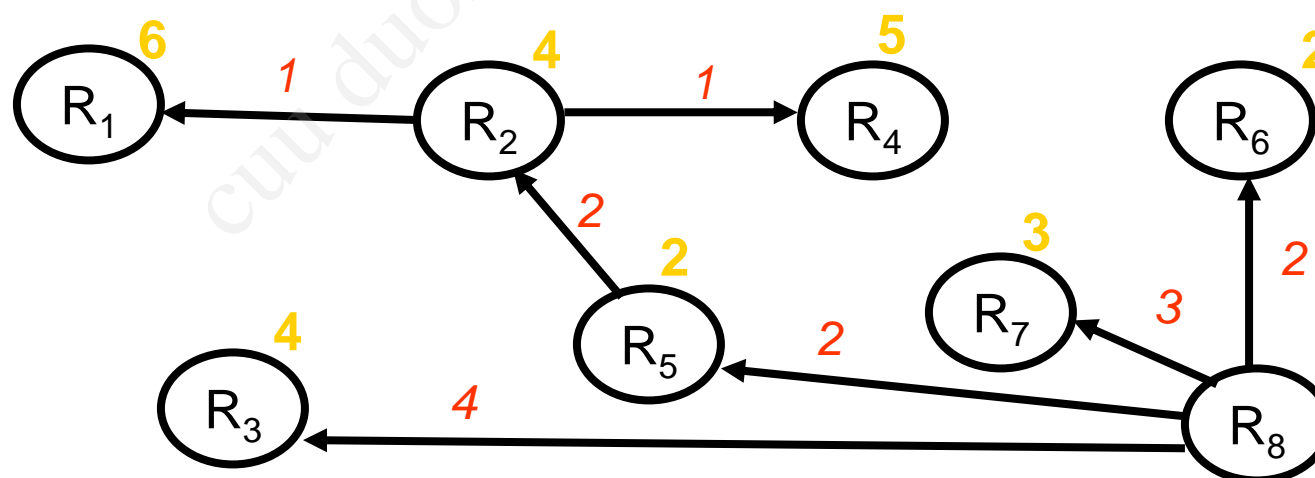
R <sub>1</sub>	6, R <sub>3</sub>
R <sub>2</sub>	4, R <sub>5</sub>
R <sub>3</sub>	4, R <sub>8</sub>
R <sub>4</sub>	6, R <sub>7</sub>
R <sub>5</sub>	2, R <sub>8</sub>
R <sub>6</sub>	2, R <sub>8</sub>
R <sub>7</sub>	3, R <sub>8</sub>

## Example



R <sub>1</sub>	5, R <sub>2</sub>
R <sub>2</sub>	4, R <sub>5</sub>
R <sub>3</sub>	4, R <sub>8</sub>
R <sub>4</sub>	5, R <sub>2</sub>
R <sub>5</sub>	2, R <sub>8</sub>
R <sub>6</sub>	2, R <sub>8</sub>
R <sub>7</sub>	3, R <sub>8</sub>

## Solution



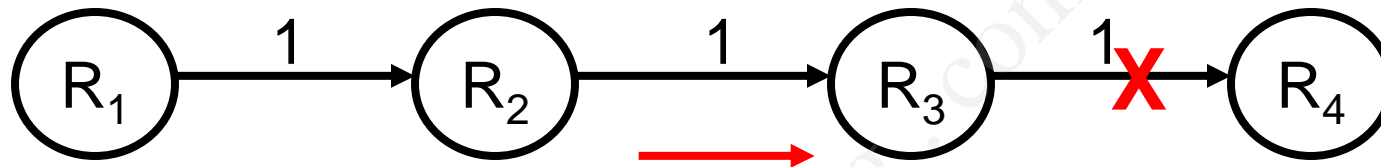
# Distributed Bellman-Ford Algorithm

Vấn đề:

1. Bao lâu thì thuật toán hội tụ?
2. Làm sao chứng minh thuật toán luôn hội tụ?
3. Nếu có sự thay đổi (ví dụ đường truyền bị hư)?

# Vấn đề Bellman-Ford

“Hội tụ chậm”



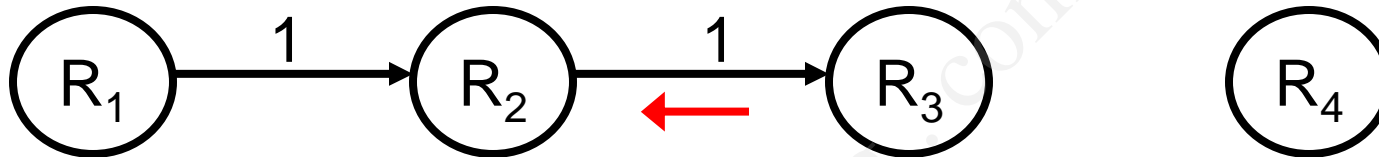
Giả sử link đến R4 bị đứt :

Time	$R_1$	$R_2$	$R_3$
0	3, $R_2$	2, $R_3$	1, $R_4$
1	3, $R_2$	2, $R_3$	<b>3, <math>R_2</math></b>
2			
3			
4			

$R_3 \leftrightarrow R_4$  đứt

# Vấn đề Bellman-Ford

“Hội tụ chậm”



Time	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>
0	3, R <sub>2</sub>	2, R <sub>3</sub>	1, R <sub>4</sub>
1	3, R <sub>2</sub>	2, R <sub>3</sub>	<b>3, R<sub>2</sub></b>
2	3, R <sub>2</sub>	<b>4, R<sub>3</sub></b>	3, R <sub>2</sub>
3	<b>5, R<sub>2</sub></b>	4, R <sub>3</sub>	<b>5, R<sub>2</sub></b>
4	5, R <sub>2</sub>	<b>6, R<sub>3</sub></b>	5, R <sub>2</sub>

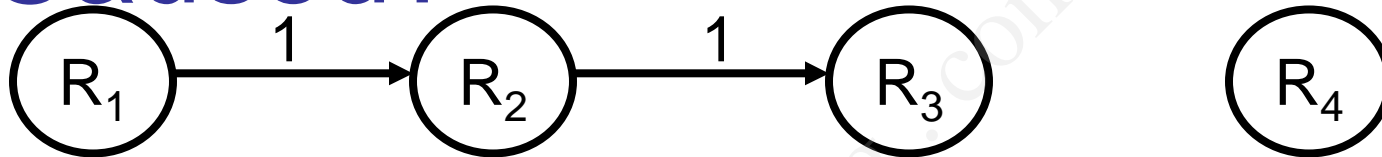
← R<sub>3</sub>-R<sub>4</sub> fails

“Đếm đến vô cực”

...

...

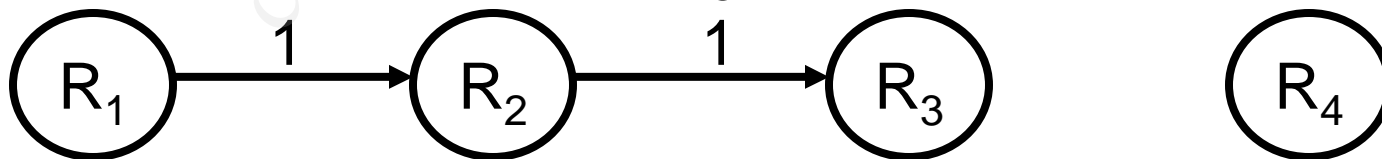
# How are These Loops Caused?



- Nhận xét 1:
  - R3's metric tăng
- Nhận xét 2:
  - R2 lấy R3 như là node kế tiếp đến R4
  - Nhưng đường đi đến R4 bao gồm chính đường đi qua R2

# Giải pháp cho việc đếm đến vô cùng

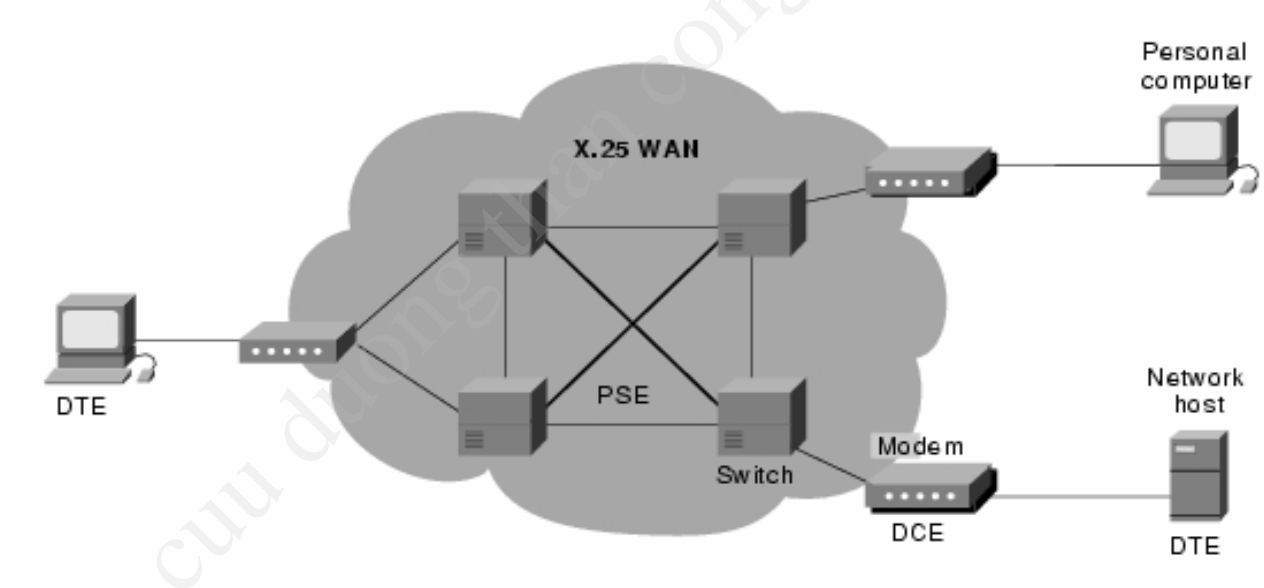
- Đặt infinity = “một số hữu hạn” (e.g. 16). Dừng lại khi count = 16.
- Split Horizon: Vì  $R_2$  có đường ngắn nhất qua  $R_3$ , nên nó không thông báo cập nhật cho  $R_3$
- Split-horizon with poison reverse:  $R_2$  cập nhật vô cùng cho  $R_3$



# X.25

- Được ITU phát triển và chuẩn hóa từ năm 1970 nhằm tạo một giao thức cho mạng WAN
- X.25 định nghĩa 3 loại phần tử:
  - DTE ( Data Terminal Equipment): thiết bị đầu cuối như terminal, PC
  - DCE (Data-circuit terminating equipment): thiết bị giao tiếp giữa thiết bị đầu cuối (DTE) và thiết bị chuyển mạch (PSE)
  - PSE (Packet-switching exchange): vai trò chuyển mạch
  - PAD (Packet Assembler/Disassembler): là thiết bị phụ được hỗ trợ thiết bị đầu cuối trong việc sử dụng X.25, thực hiện 3 chức năng chính: buffer, phân đoạn và ghép nối các gói dữ liệu
- X.25 thực hiện việc truyền dữ liệu theo mode connection-oriented

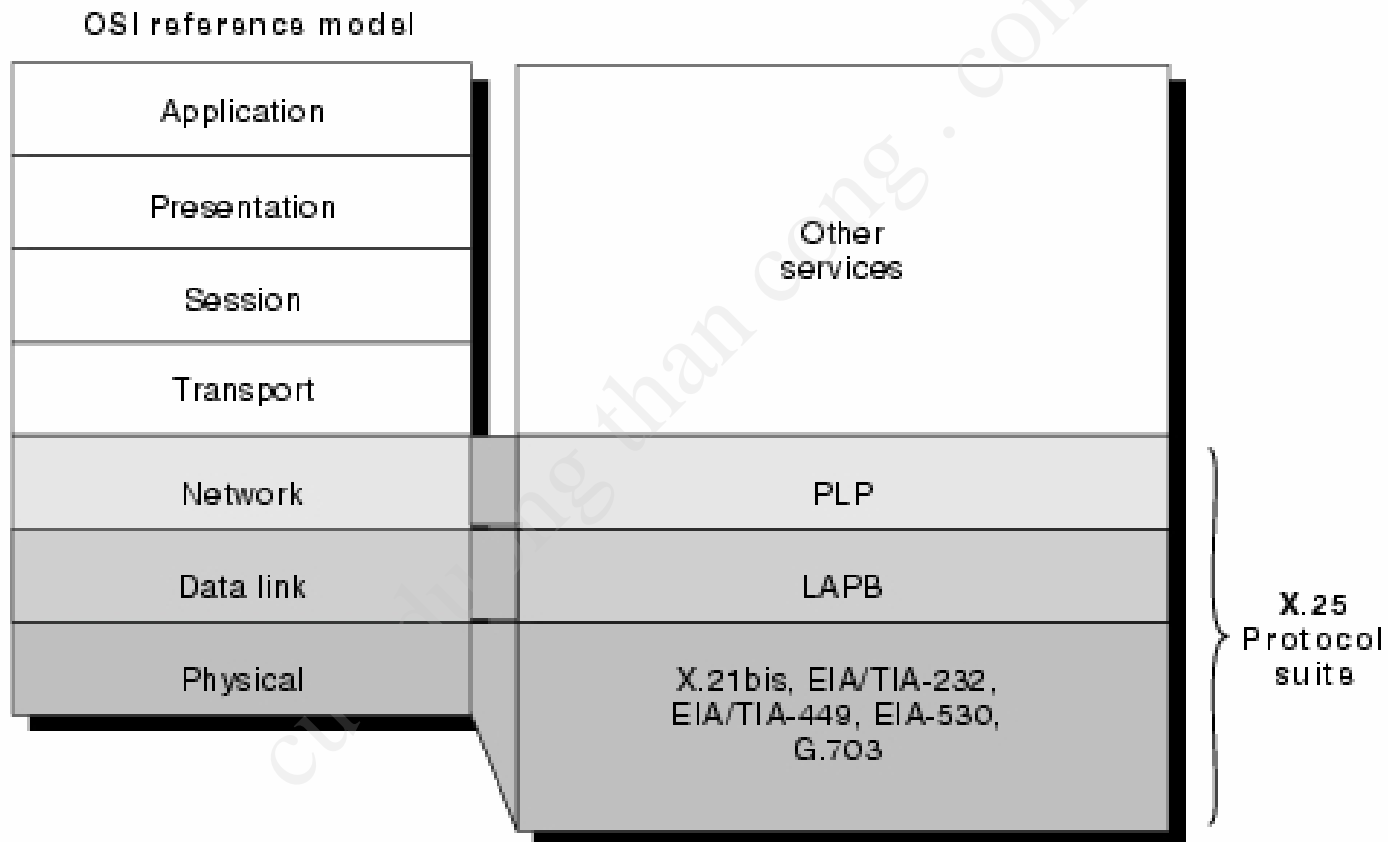




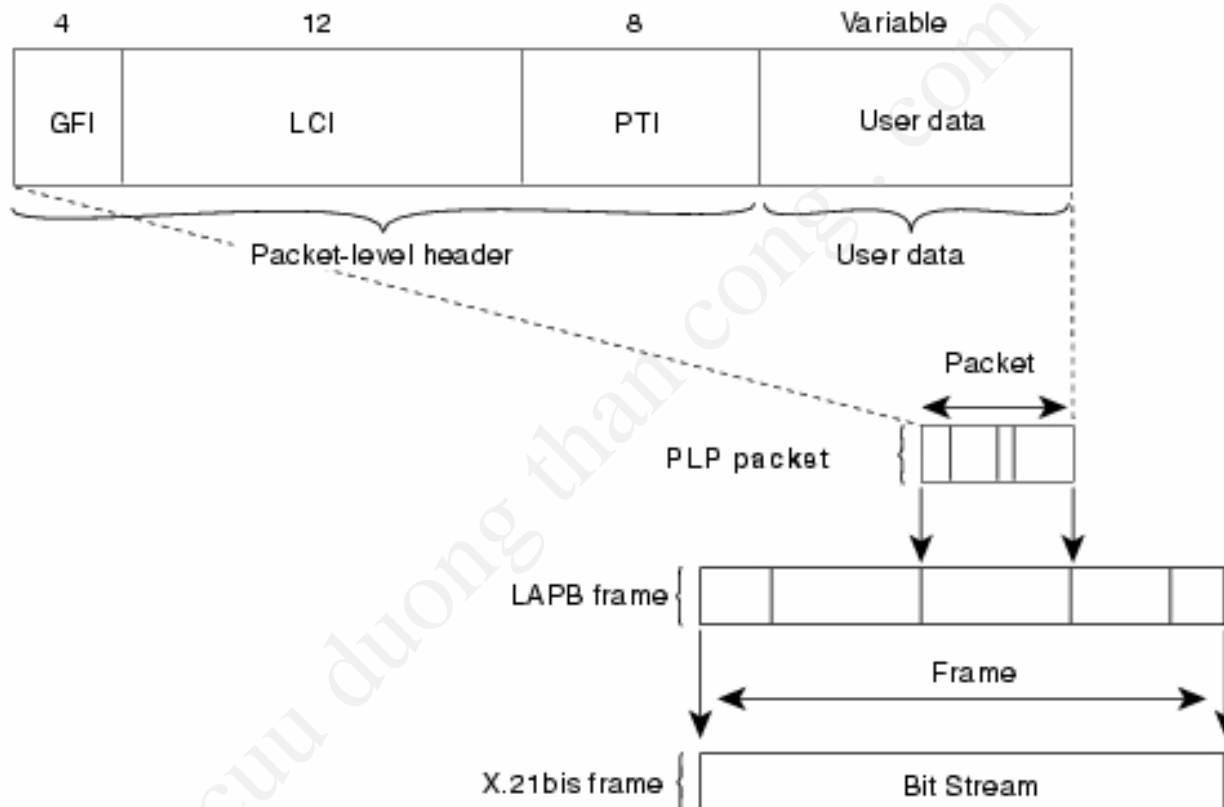
# Virtual circuit (VC)

- Virtual circuit: một liên kết luận lý 2 chiều nhằm bảo đảm việc truyền thông tin tin cậy giữa 2 thiết bị đầu cuối
- Có thể qua nhiều thiết bị chuyển mạch
- Có thể có nhiều VC trên 1 đường truyền vật lý
- 2 loại VC: Switched VC (SVC) và Permanent VC (PVC)
- Cho phép 4095 VC trên một X.25 interface

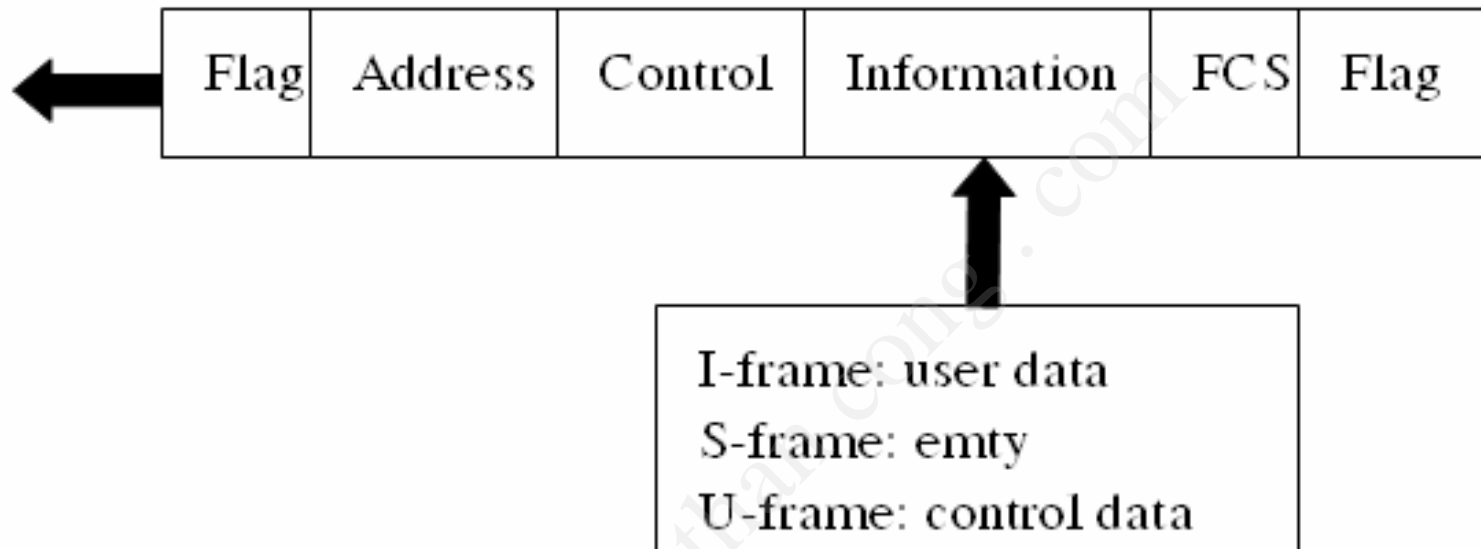
# OSI layers



Field length,  
in bits



# ĐỊNH DẠNG KHUNG CỦA LỚP B

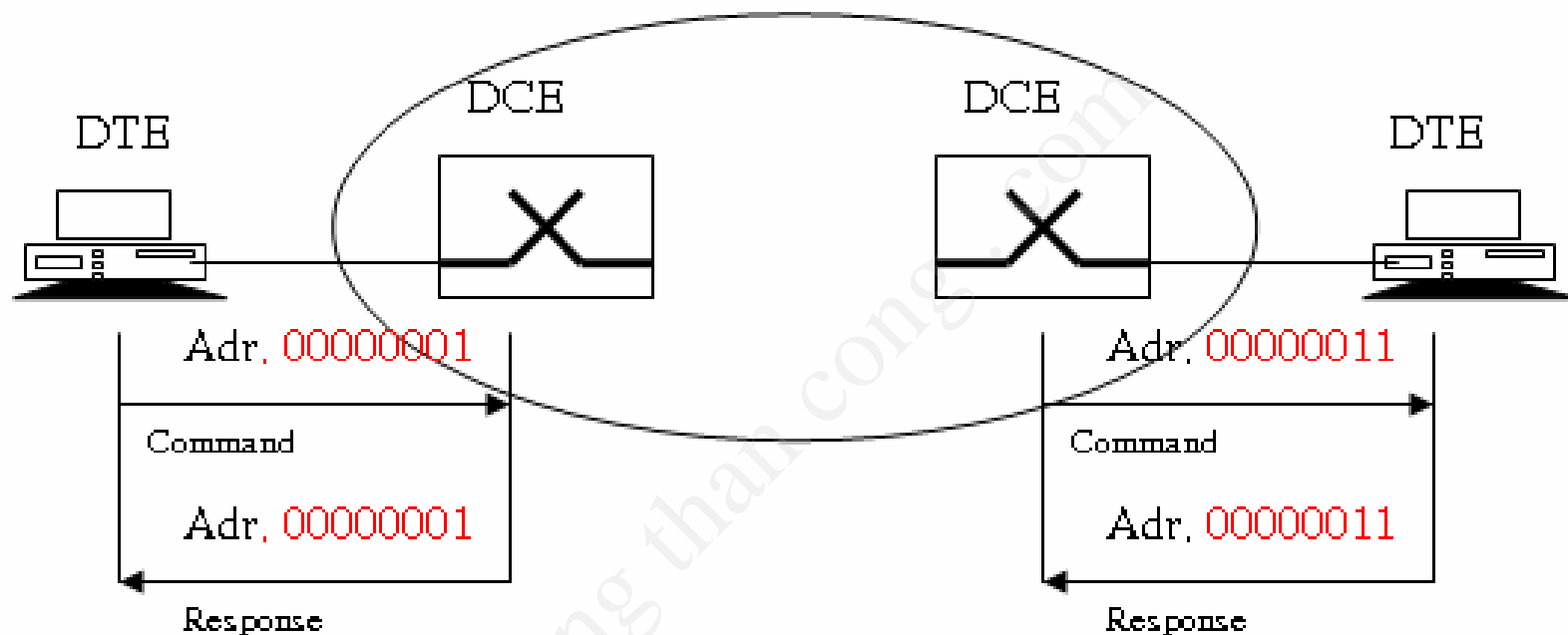


I-frame, các khung dữ liệu từ lớp mạng

S-frame, kiểm soát lỗi và kiểm soát luồng trong lớp khung

U-frame: thiết lập và giải tỏa kết nối giữa DTE và DCE

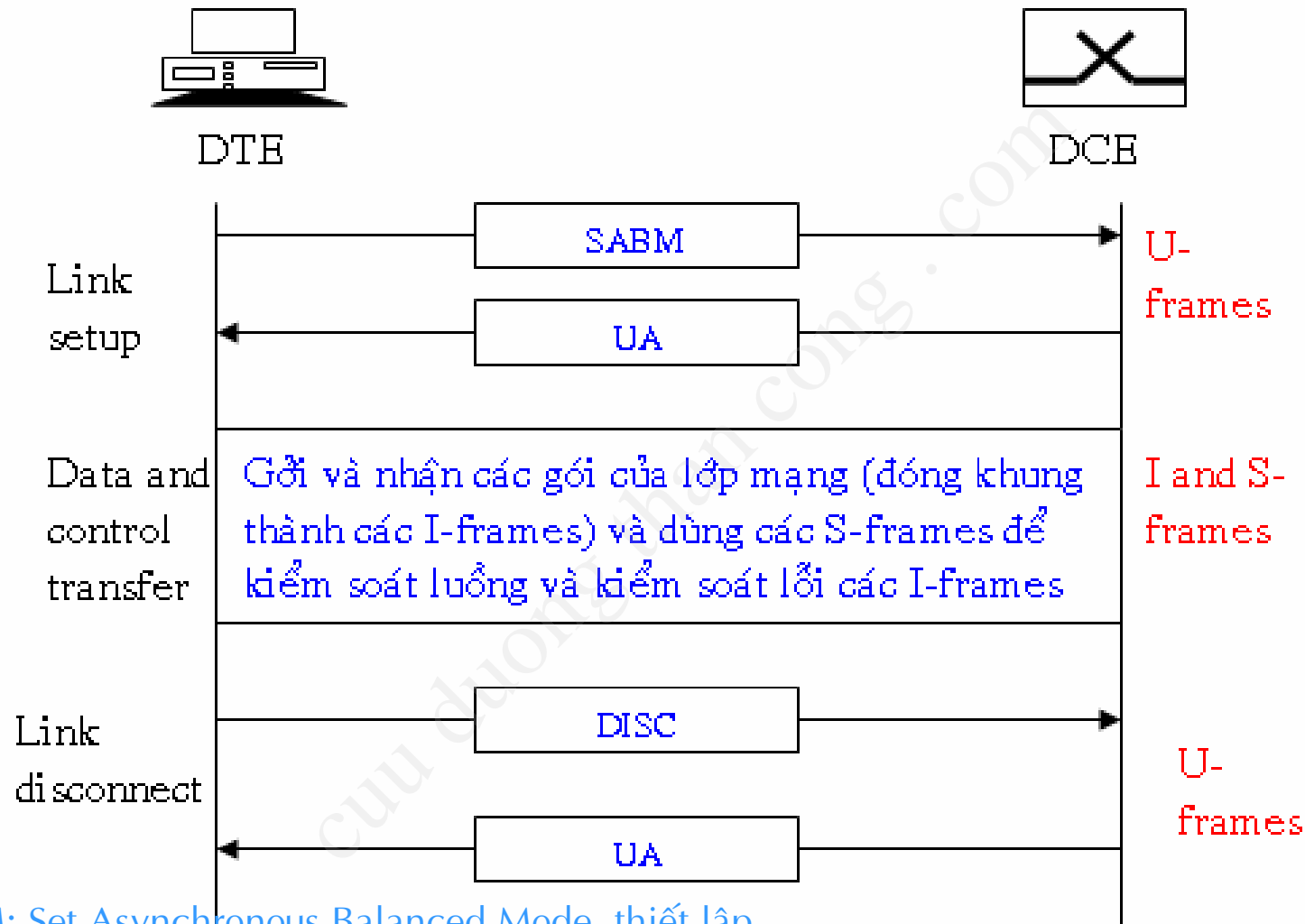
# **ĐỊNH DẠNG KHUNG CỦA LAPB**



Địa chỉ 00000001 sử dụng cho lệnh phát từ DTE và đáp ứng phát từ DCE

Địa chỉ 00000011 sử dụng cho lệnh phát từ DCE và đáp ứng phát từ DTE

# TRUYỀN DỮ LIỆU LẬP B



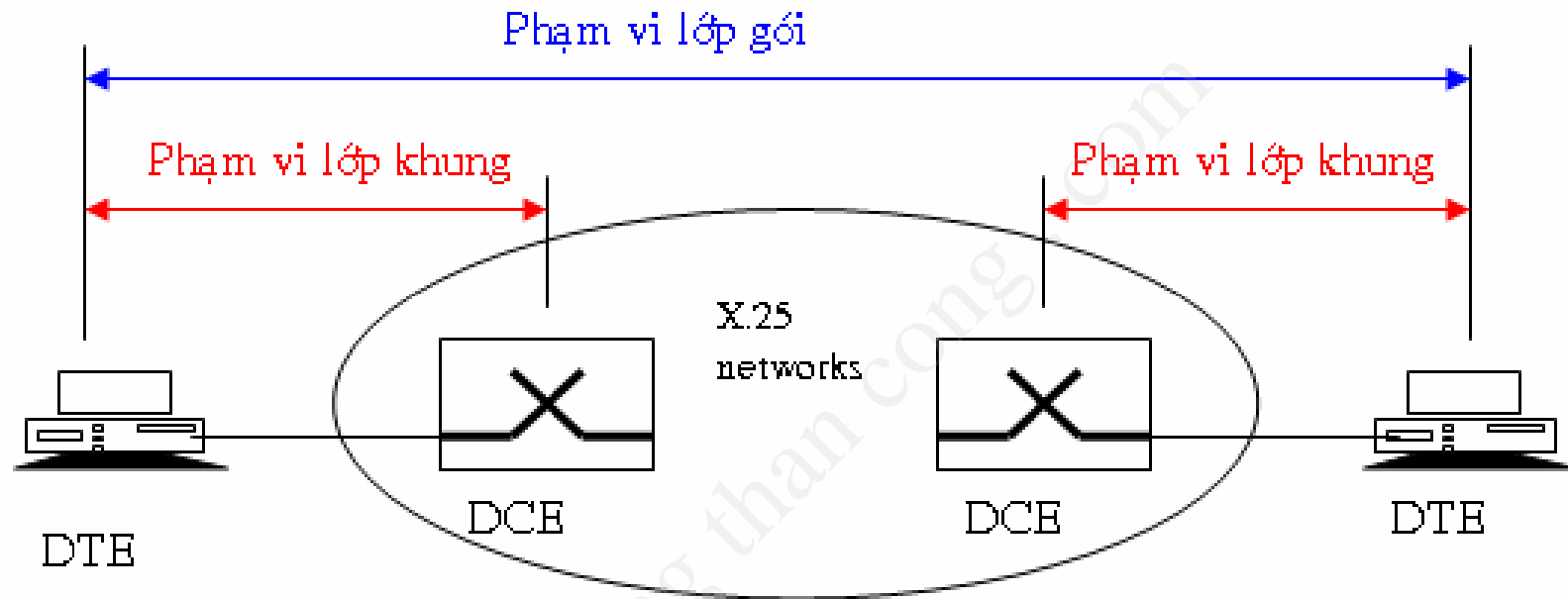
# Packet-layer Protocol (PLP)

- Quản lý việc gửi các gói qua các VCs:
  - Cho phép 2 DTE có thể trao đổi dữ liệu
  - Cho phép 2 DTE có thể sử dụng một phần tài nguyên của mạng tạm thời hay trong một khoảng thời gian dài
  - Khắc phục những lỗi ở lớp mạng
  - Kiểm soát luồng trên một VC
- Có 5 modes hoạt động:
  - Call setup: thiết lập SVC
  - Data transfer: truyền dữ liệu giữa các DTE, thực hiện phân-đoạn/ghép-nối dữ liệu, chèn bit, phát hiện lỗi, và kiểm soát luồng
  - Idle mode: SVC được thiết lập nhưng chưa truyền dữ liệu.
  - Call clearing: kết thúc sự trao đổi dữ liệu giữa 2 DTE và kết thúc SVC
  - Restarting: đồng bộ truyền giữa một DTE và một DCE



- PLP có thể mang các loại thông tin sau:
  - General Format Identifier (GFI): loại thông tin được mang (control/data), loại cửa sổ (window), thông tin xác nhận về việc nhận thông tin.
  - Logical Channel Identifier (LCI): xác định VC
  - Packet Type Identifier (PTI): xác định loại gói (17 loại)
  - User data

# Packet-layer Protocol (PLP)



## Nhiệm vụ của lớp gói (packet layer)

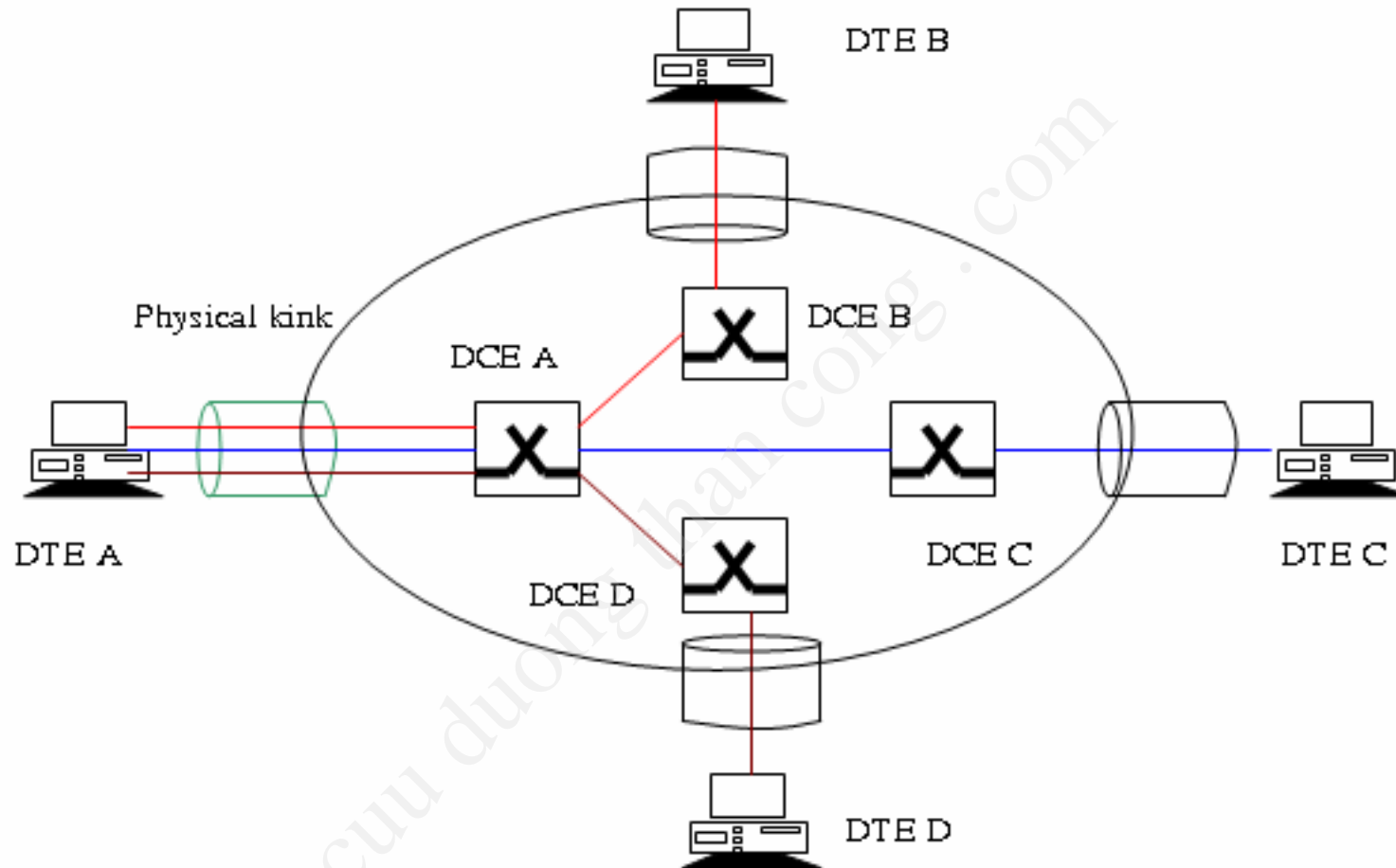
Thiết lập, kết nối, chuyển dữ liệu và kết thúc kết nối giữa 2 DTE

Lưu ý: X.25 kiểm soát luồng và kiểm soát lỗi tại 2 lớp là lớp khung và lớp gói, do đó X.25 truyền dữ liệu tin cậy (ít lỗi) nhưng tốc độ chậm

Lớp khung kiểm soát luồng và kiểm soát lỗi giữa DCE và DTE

Lớp gói kiểm soát lỗi và kiểm soát luồng giữa 2 DTE

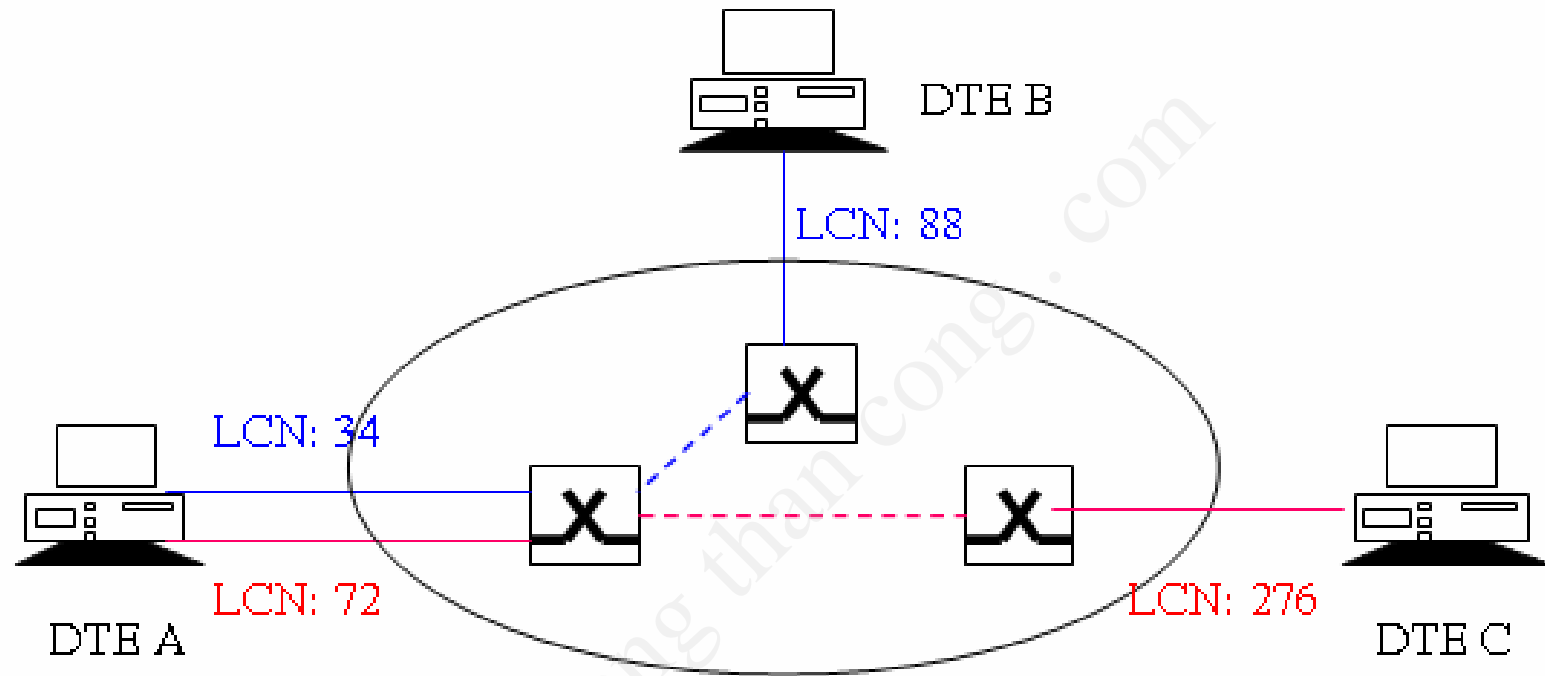
# Packet-layer Protocol (PLP)



*Mạch ảo được tạo ra giữa DTE A và các DTE B, DTE C, DTE D*

➤ X.25 là mạng mạch ảo chuyển mạch gói

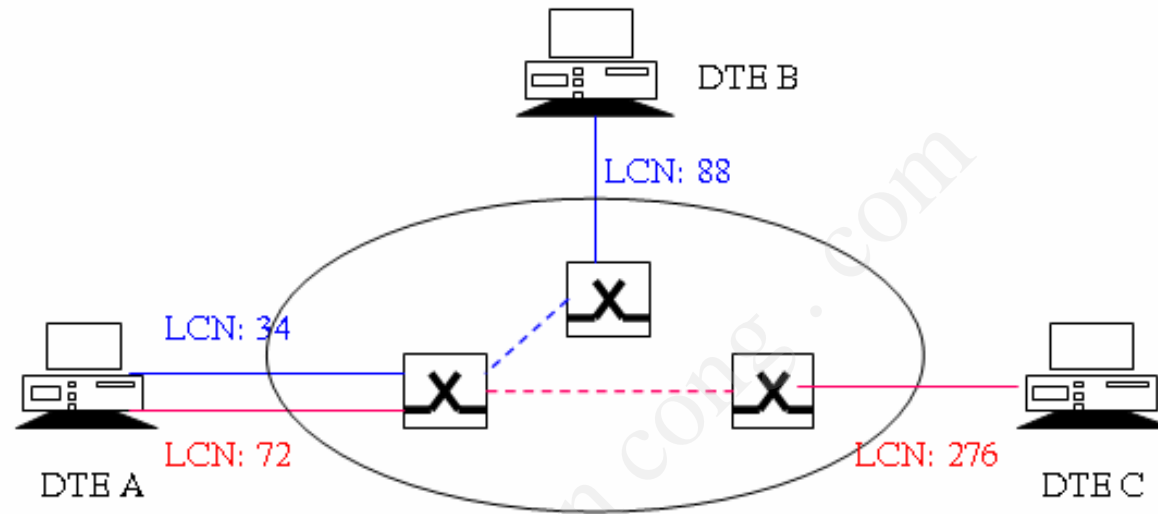
# Packet-layer Protocol (PLP)



*Định danh mạch ảo trong X.25*

- LCN (Logical Channel Number) là các định danh mạch ảo trong X.25 được dùng cho các gói.
- Một mạch ảo được thiết lập luôn có một cặp LCN, LCN 1 DTE-DCE cục bộ và LCN 2 giữa DTE-DCE đầu xa
- X.25 dùng cả SVC và PVC

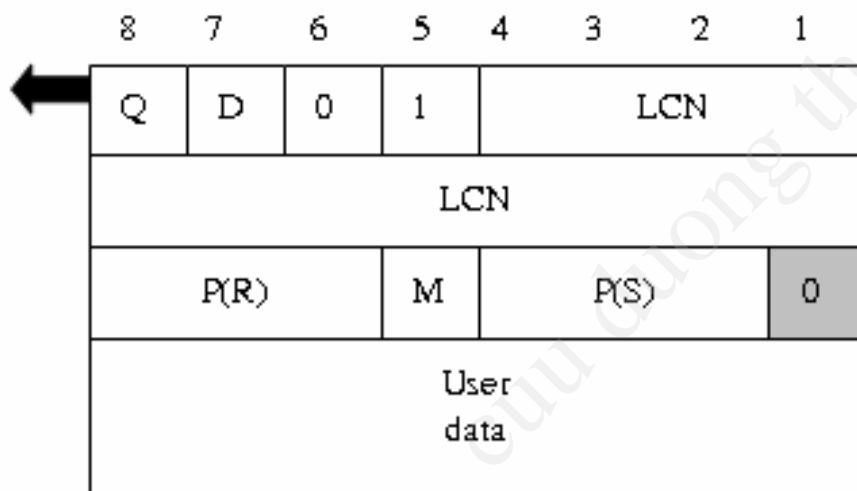
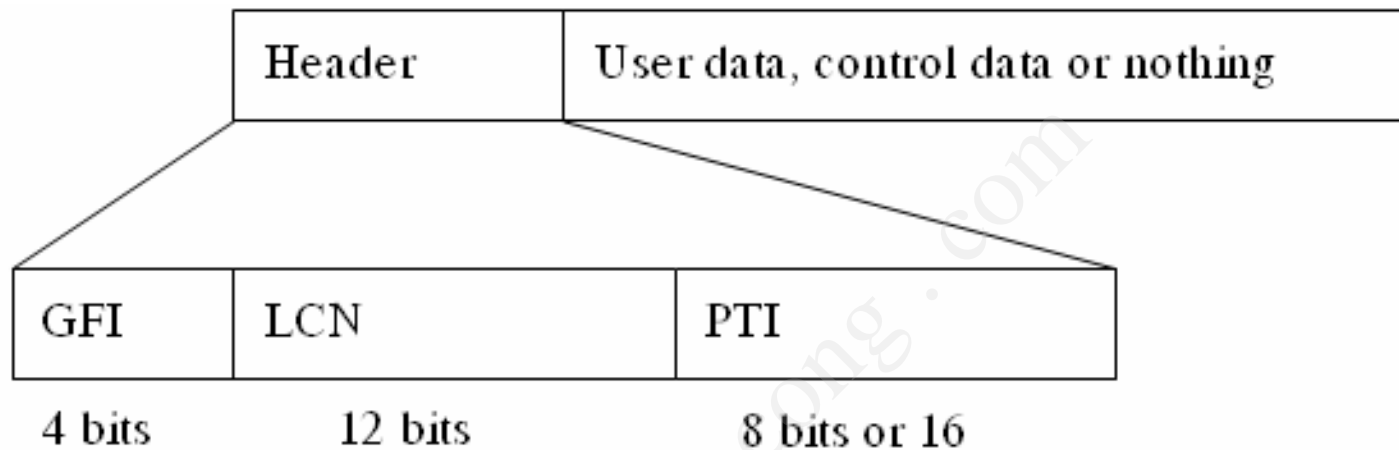
# QUÁ TRÌNH TRUYỀN TRÒNG X.25



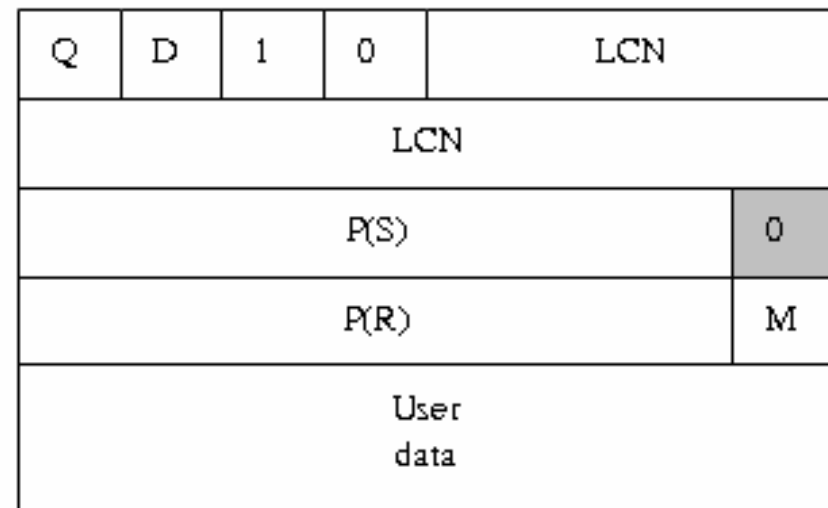
## *Định danh mạch ảo trong X.25*

- Một liên kết được thiết lập giữa DTE-DCE cục bộ và DTE-DCE cục bộ
- Một mạch ảo được thiết lập giữa 2 DTE
- Dữ liệu được chuyển giữa 2 DTE
- Mạch ảo được giải tỏa
- Liên kết được dùng kết nối

# CẤU TRÚC GÓI TRONG PLP

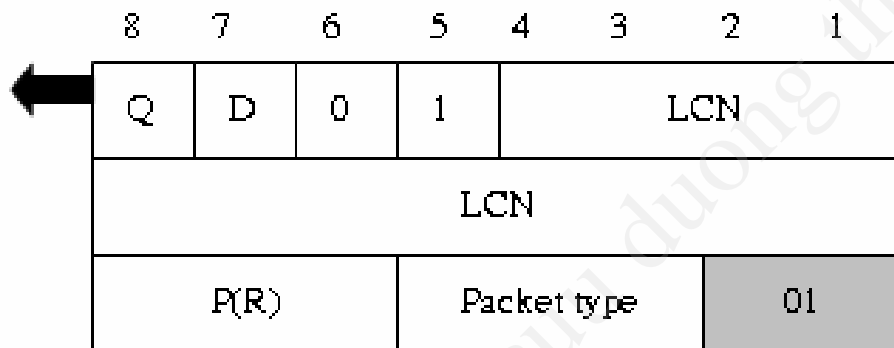
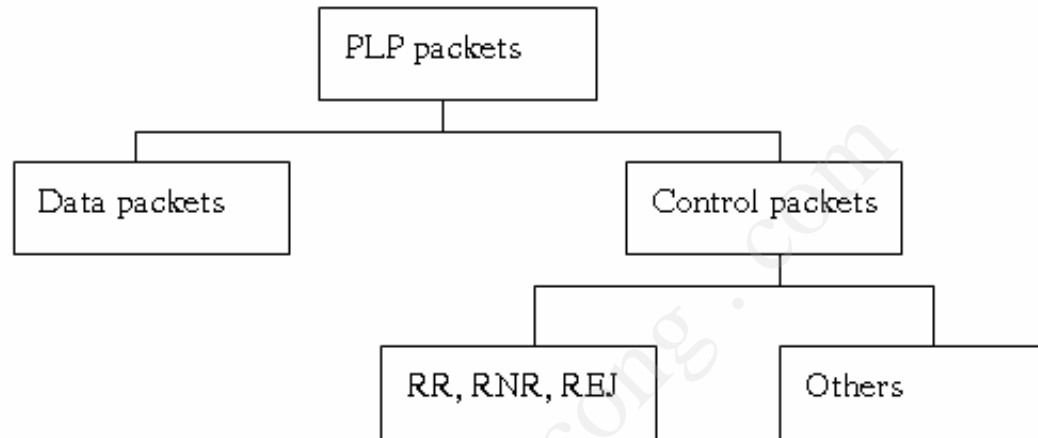


(a) Đánh số 3 bits

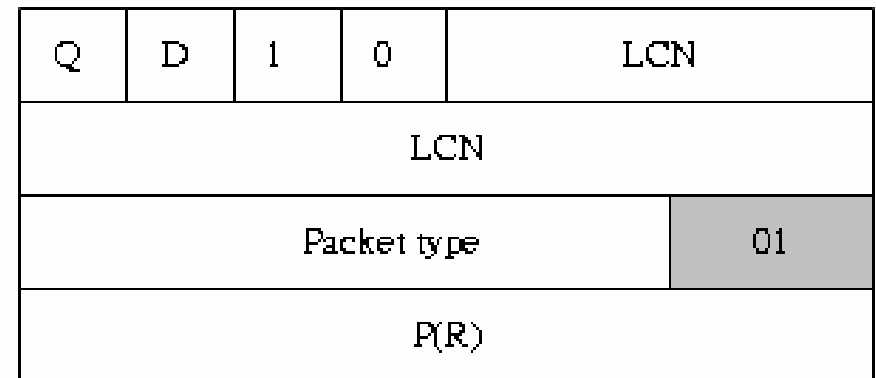


(b) Đánh số 7 bits

# CÁC LOẠI GÓI TRONG X.25



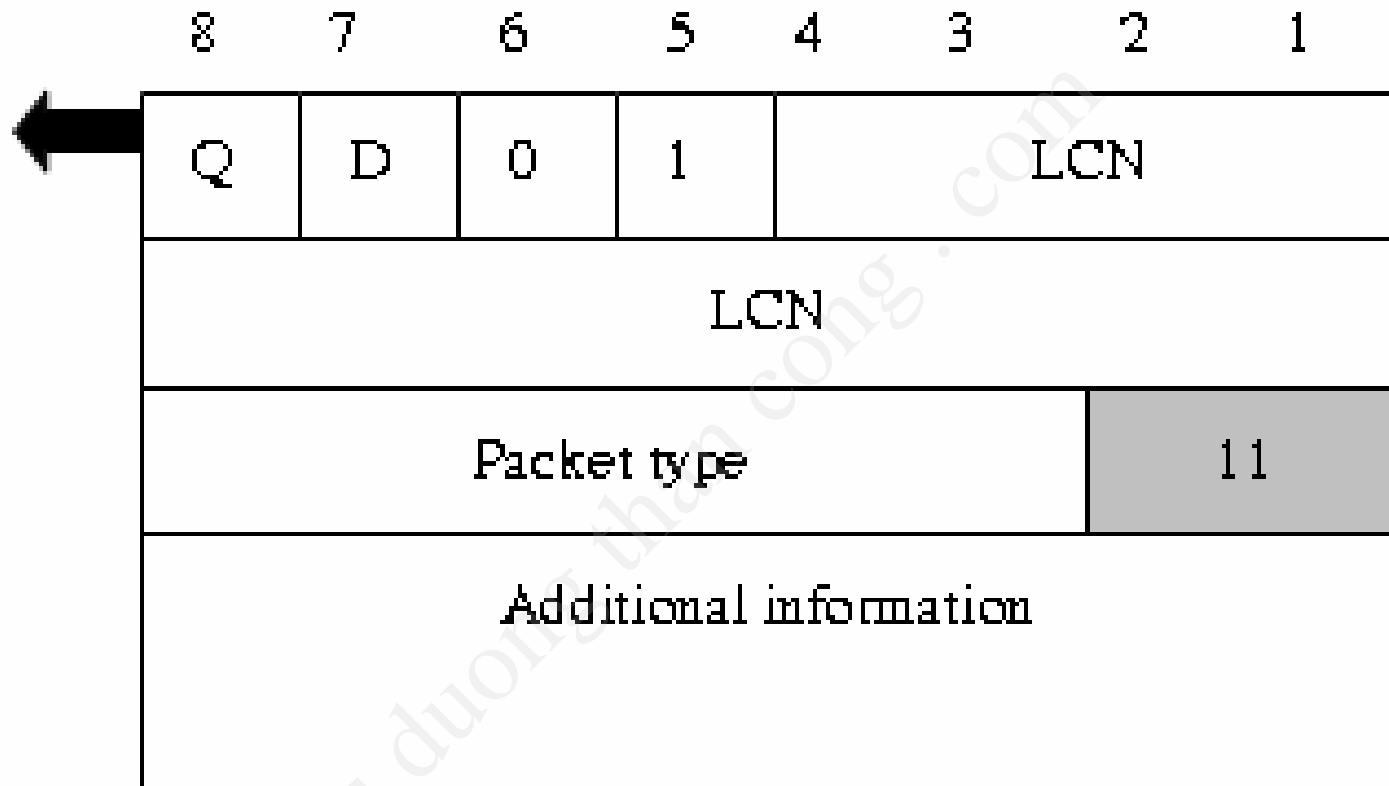
(a) Đánh số 3 bits



(b) Đánh số 7 bits

Cấu trúc các gói RR, RNR và REJ

# CÁC LOẠI GÓI TRÔNG X.25



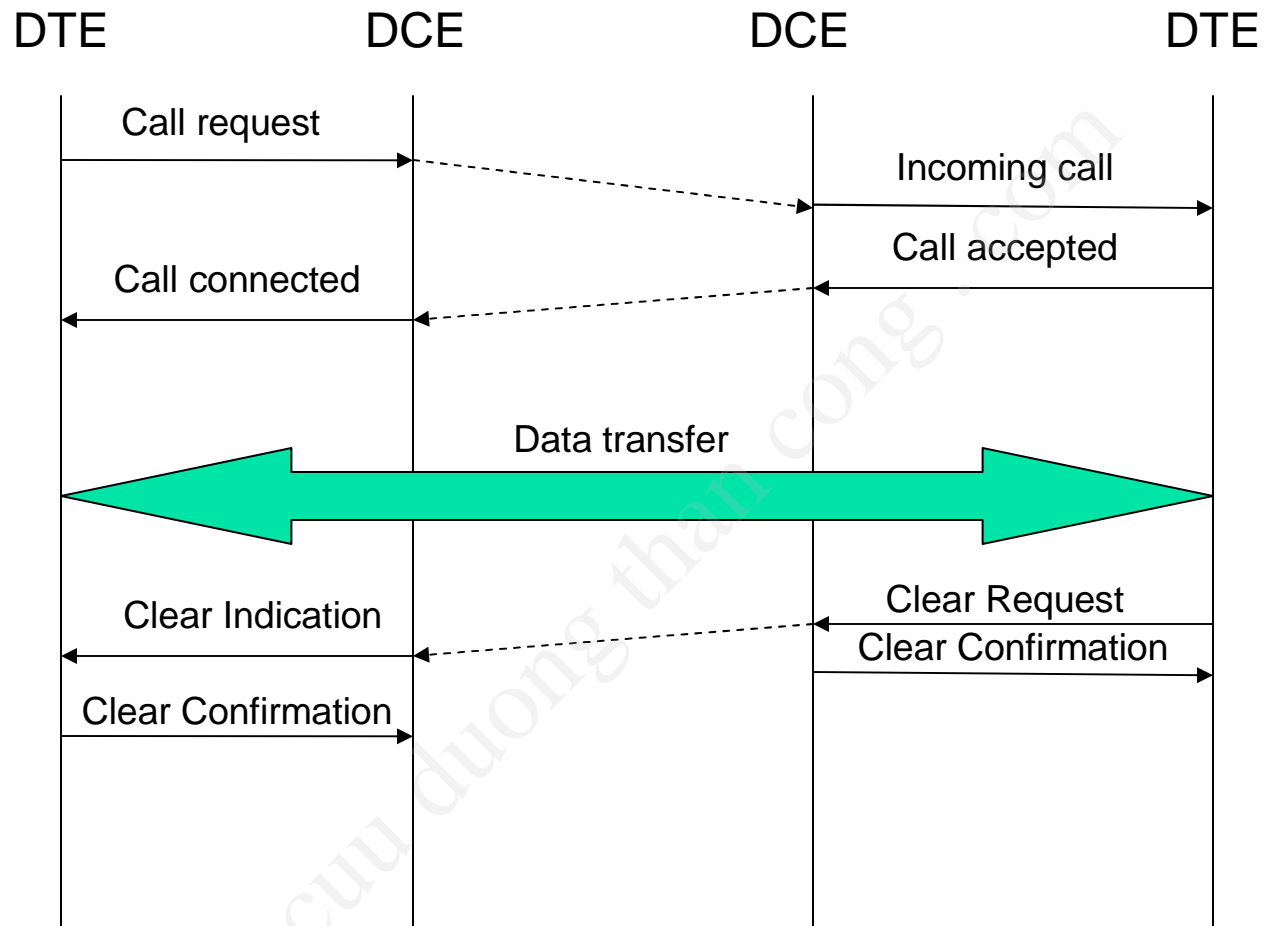
Cấu trúc các gói điều khiển khác RR, RNR và REJ



# CÁC LOẠI GÓI TRÔNG X.25

DTE to DCE	DCE to DTE	Type
Call request	Incoming call	000010
Call accepted	Call connected	000011
Clear request	Clear indication	000100
Clear confirm	Clear confirm	000101
Interrupt	Interrupt	001000
Interrupt confirm	Interrupt confirm	001001
Reset request	Reset indication	000110
Reset confirm	Reset confirm	000111
Restart request	Restart indication	111110
Restart confirm	Restart confirm	111111
Registration request		111100
	Registration confirm	111101

Một số loại gói điều khiển thường dùng trong X.25



# Frame Relay

# Tại sao cần Frame Relay ?

- Sự cải thiện đáng kể của môi trường truyền do sự phát triển của công nghệ
- Xuất hiện yêu cầu một phương thức truyền mới:
  - *Tốc độ cao và rẻ hơn trong việc nối kết các mạng LAN*
  - *Chấp nhận tốc độ bộc phát (burst data)*
  - *Đơn giản, giảm điều khiển*

# Đặc tính của Frame Relay

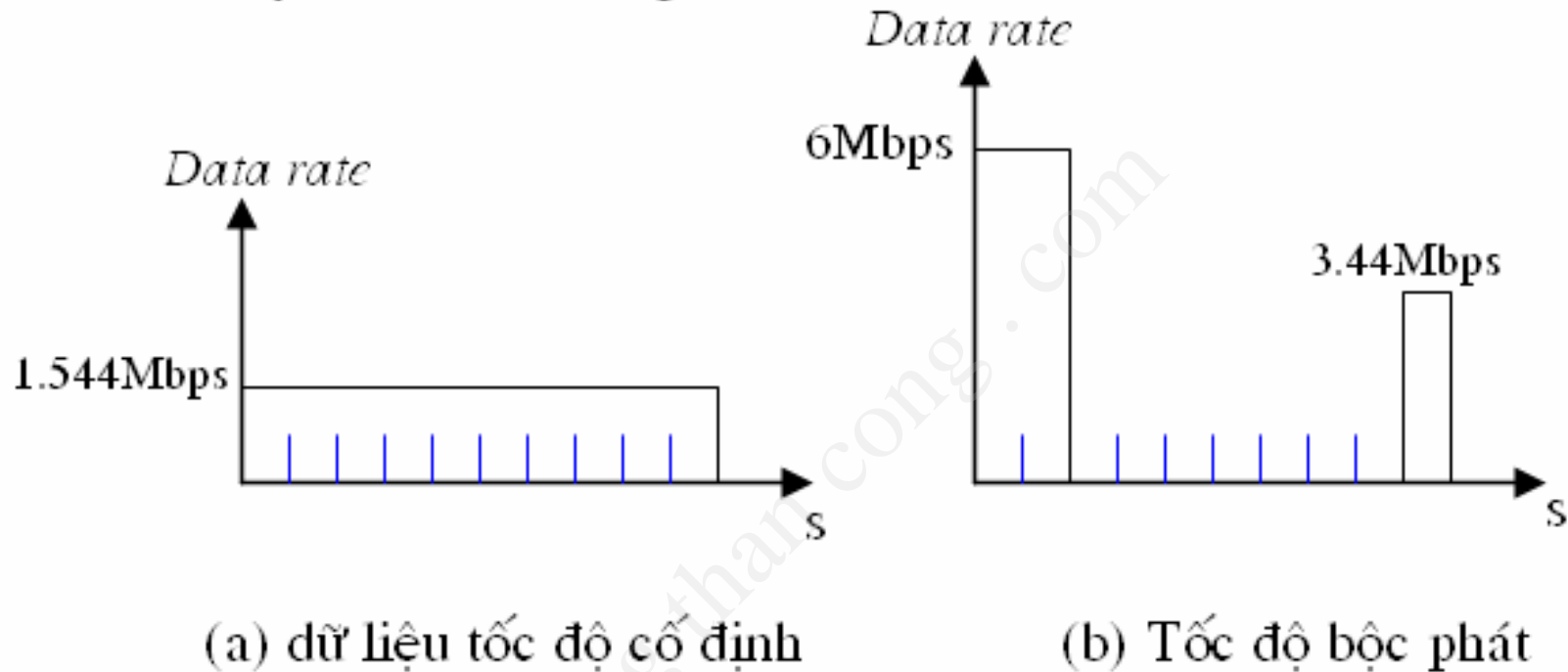
- Không cung cấp kiểm tra lỗi ở lớp 2, giúp cải thiện tốc độ truyền
- Hỗ trợ tốc độ bùng phát: cho phép người dùng có tốc độ thay đổi bùng phát
- Kích thước khung lớn nhất 9000 bytes cho phép chứa các khung của mạng LAN nhưng tạo ra độ trễ khác nhau giữa các khung nên chỉ thích cho truyền dữ liệu hơn là tín hiệu thời gian thực
- Rẻ tiền hơn các mạng LAN truyền thống

Đặc tính	X.25	Frame Relay
Connection	Lớp 2	Không
Kiểm soát luồng	Lớp 2 & 3	---
Kiểm soát lỗi	Lớp 2 & 3	---
Tốc độ	Cố định	Thay đổi
Kiểm soát nghẽn	Không cần thiết	Cần thiết

# Ưu điểm

- Thời gian truyền được rút ngắn do không cần sửa sai lỗi trên mỗi thiết bị chuyển mạch trung gian
- Giảm tải trên các thiết bị trung gian
- Đơn giản trong việc thực hiện các thiết bị
- Các thiết bị gởi “keepalive” cho phép xác định các thiết bị có còn hoạt động đúng hay không

# FRAME RELAY

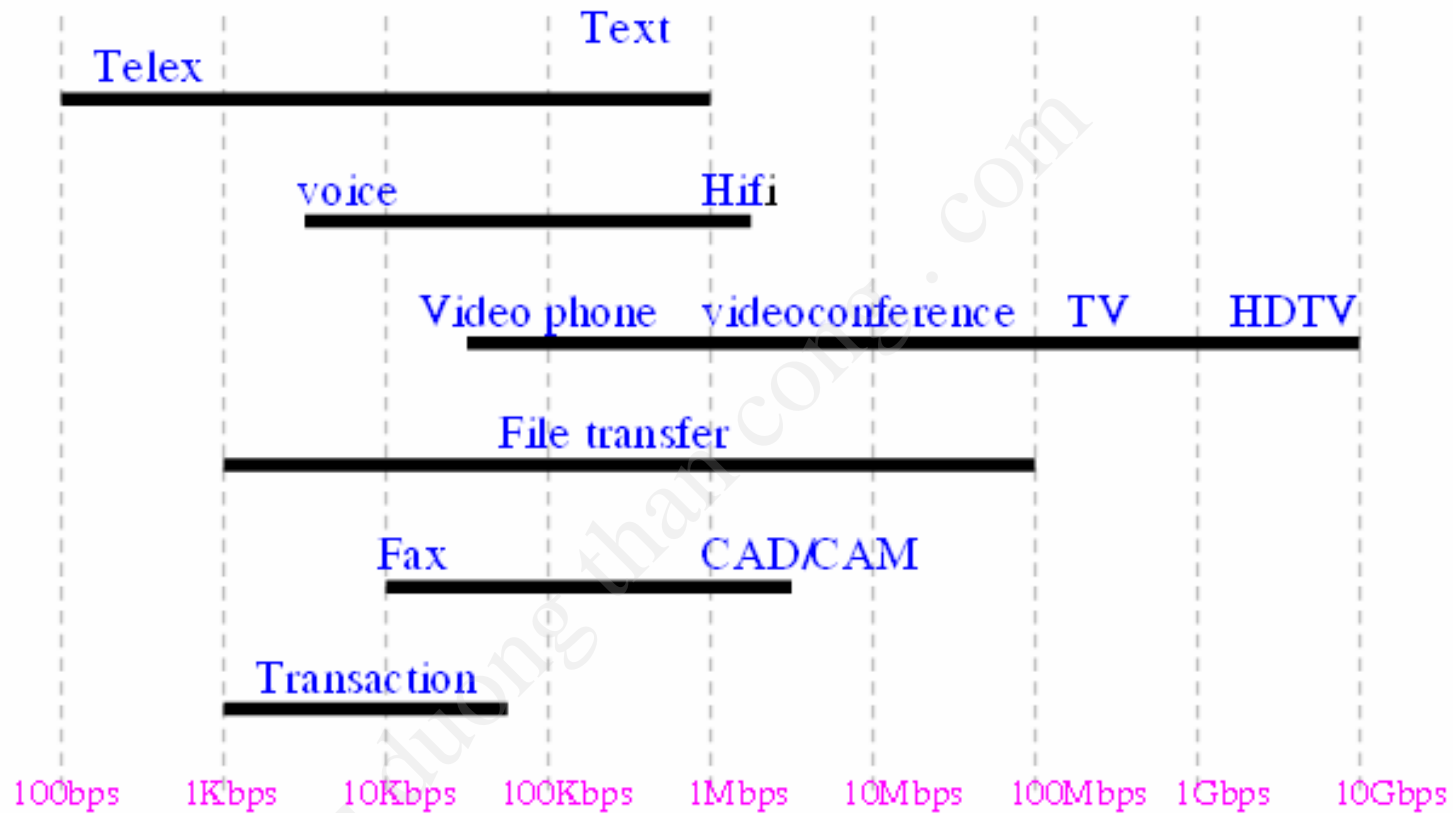


Cơ sở phát triển của FRAME RELAY

- Chất lượng môi trường truyền dẫn được cải thiện (sợi quang)
- Nhu cầu truyền dữ liệu tốc độ cao

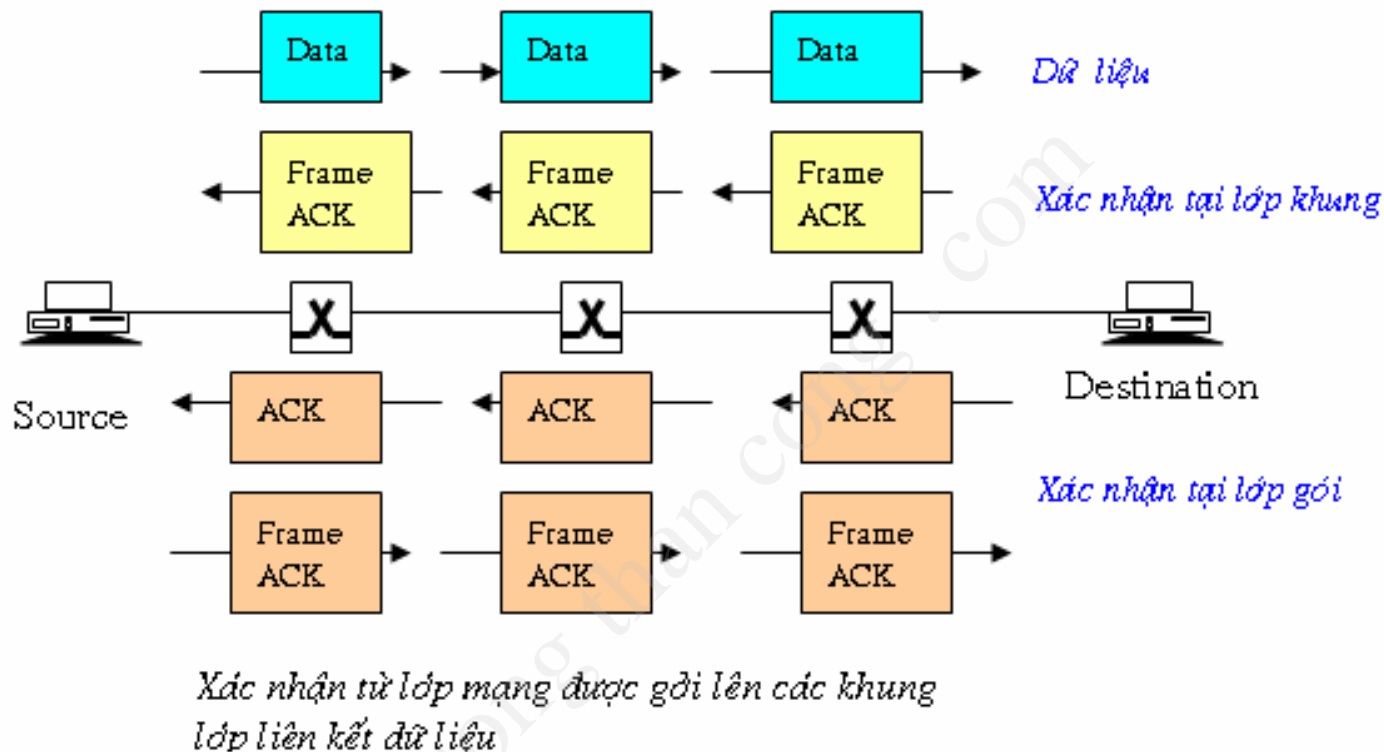


# FRAME RELAY



Hình: Tốc độ bit của một số ứng dụng

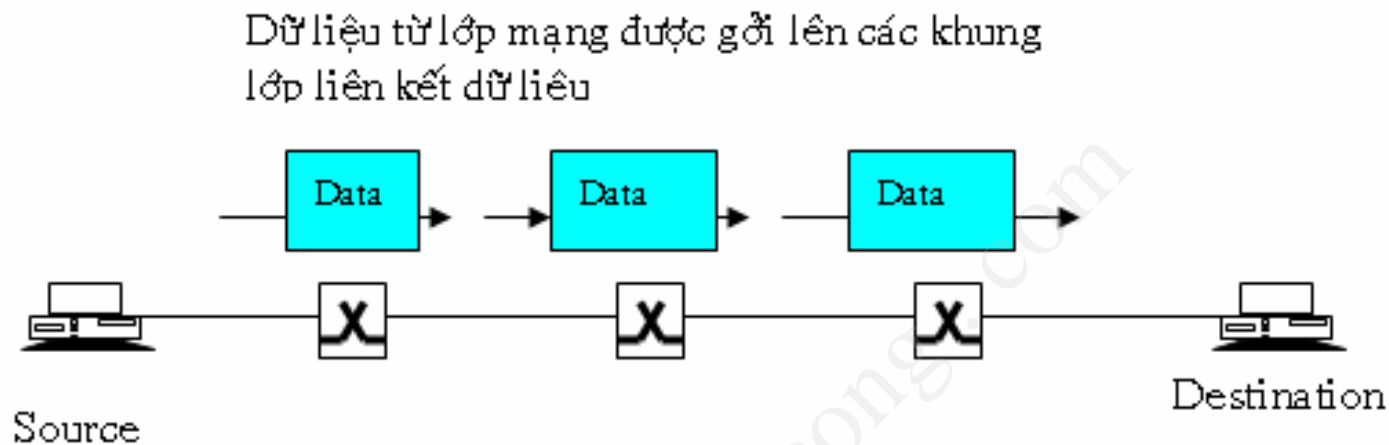
# LƯU LƯỢNG TRONG X.25



Hình Lưu lượng X.25

**Nhận xét:** X.25 thực hiện kiểm tra lỗi tại lớp khung và lớp gói -> hiệu suất truyền dữ liệu kém -> tốc độ thấp

# LƯU LƯỢNG TRONG FRAME RELAY



Hình 5.3.4 Lưu lượng Frame relay

**Nhận xét:** Frame relay không thực hiện kiểm tra lỗi tại lớp liên kết dữ liệu. Việc kiểm soát lỗi được dành cho các lớp cao hơn (lớp mạng hoặc giao vận)

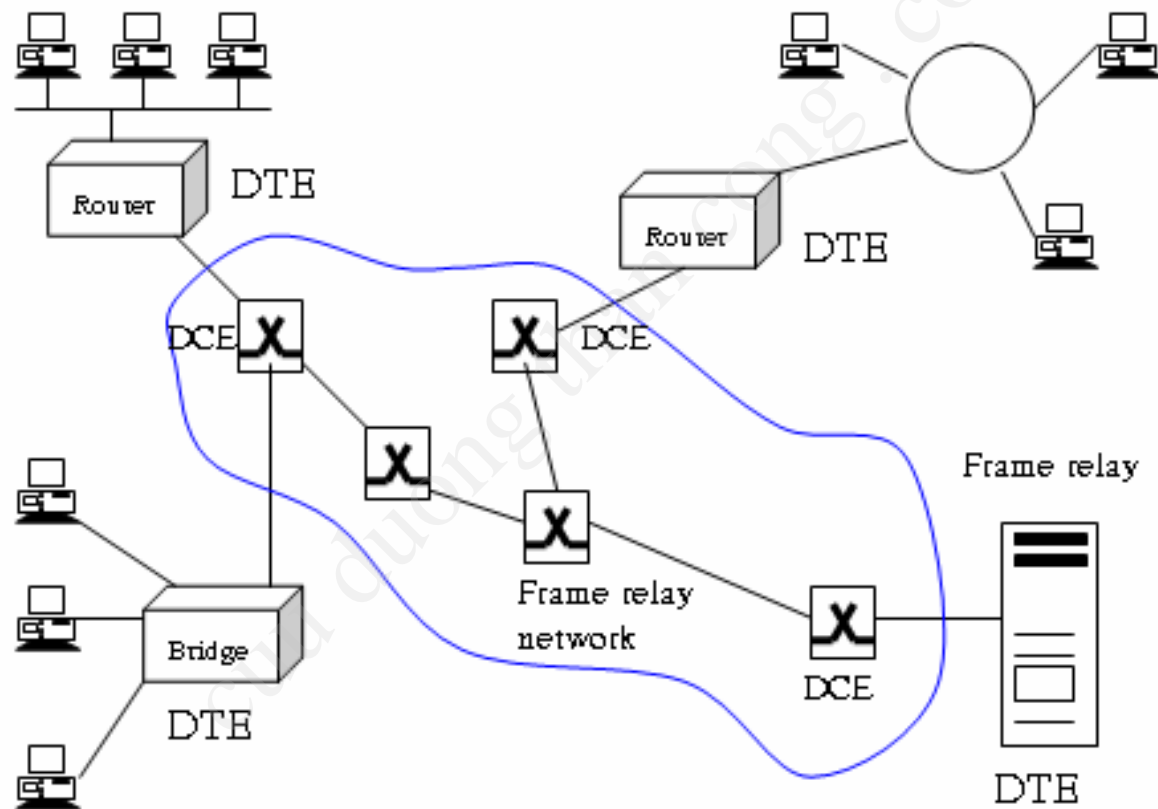
Feature	X.25	Frame Relay
Connection establishment	At the network layer	None
Hop-by-hop flow control and error control	At the data link layer	None
End-to-end flow control and error control	At the network layer	None
Data rate	Fixed	Bursty
Multiplexing	At the network layer	At the data link layer
Congestion control	Not necessary	Necessary

# **ĐẶC ĐIỂM CỦA FRAME RELAY**

- Frame relay hoạt động ở tốc độ cao (44,376Mbps). Có thể thay thế các đường T lines.
- Frame relay chỉ hoạt động ở lớp vật lý và liên kết dữ liệu
- Frame relay có thể truyền dữ liệu với tốc độ bùng phát (khác với X.25 và T lines)
- Kích thước khung dữ liệu trong frame relay có thể thay đổi. Do đó trễ giữa các khung cũng khác nhau. Do đó frame relay không thích hợp với các loại dữ liệu nhạy với trễ như thông tin tiếng nói hay video thời gian thực

# HOẠT ĐỘNG CỦA FRAME RELAY

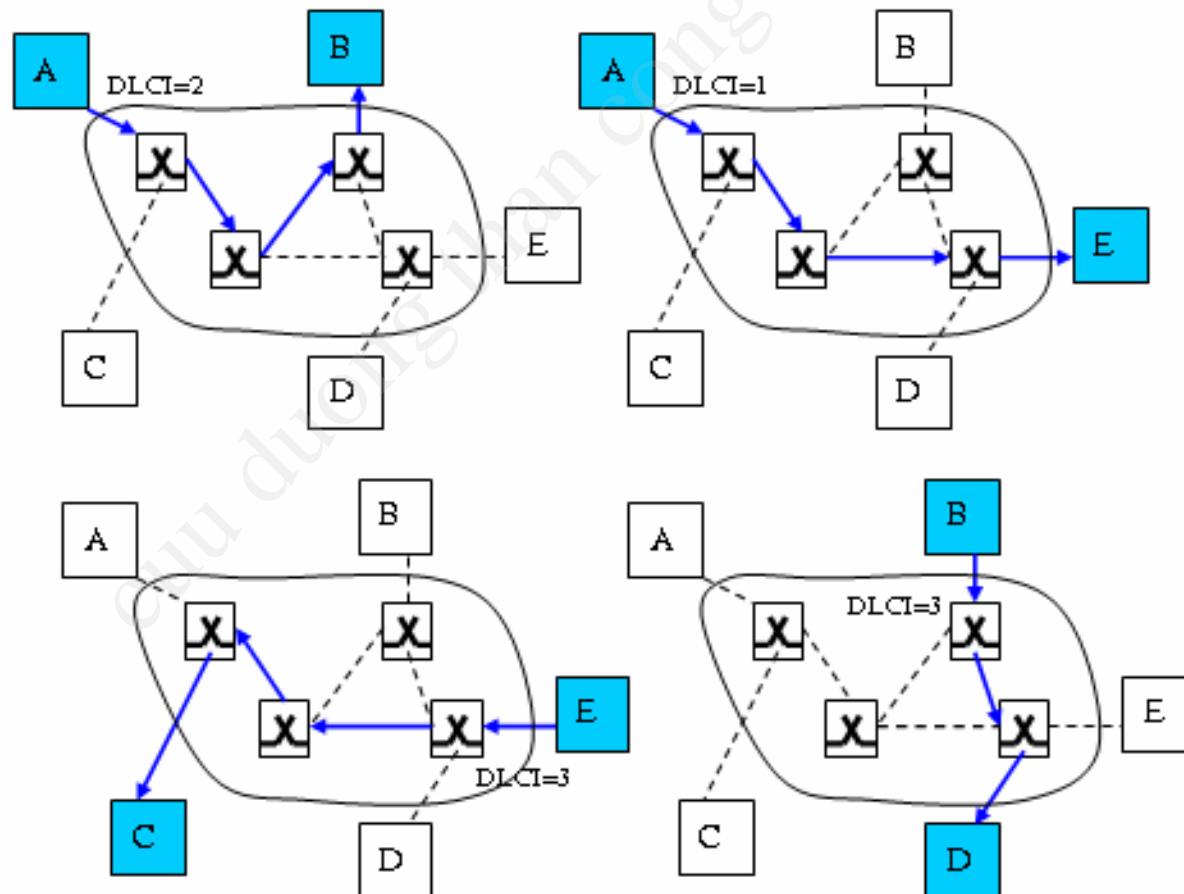
- Frame relay được dùng như WAN (Wide Area Network) để nối các mạng LAN (Local Area Network)



Hình Mạng frame relay

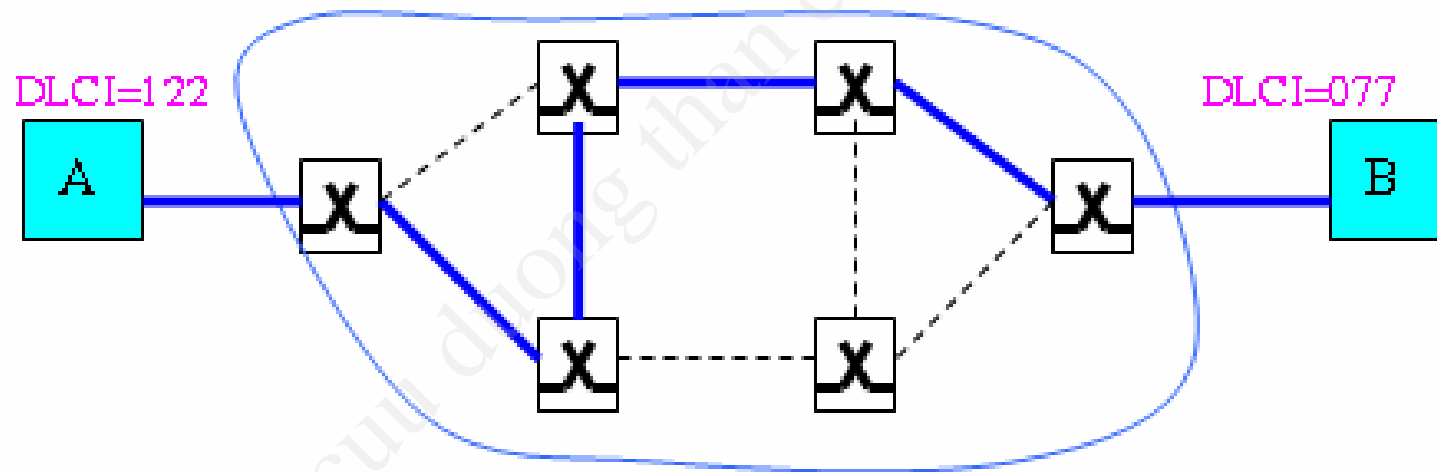
# HOẠT ĐỘNG CỦA FRAME RELAY

- Frame relay hoạt động như một mạch ảo.
- DLCI (Data Link Connection Identifier) là định danh mạch ảo trong frame relay ở trong lớp liên kết dữ liệu.
- Khi mạch ảo được thiết lập bởi mạng, DTE được cấp DLCI để truy cập DTE đầu xa.



# **HOẠT ĐỘNG THEO KIỂU MẠCH ẢO THƯỜNG XUYÊN**

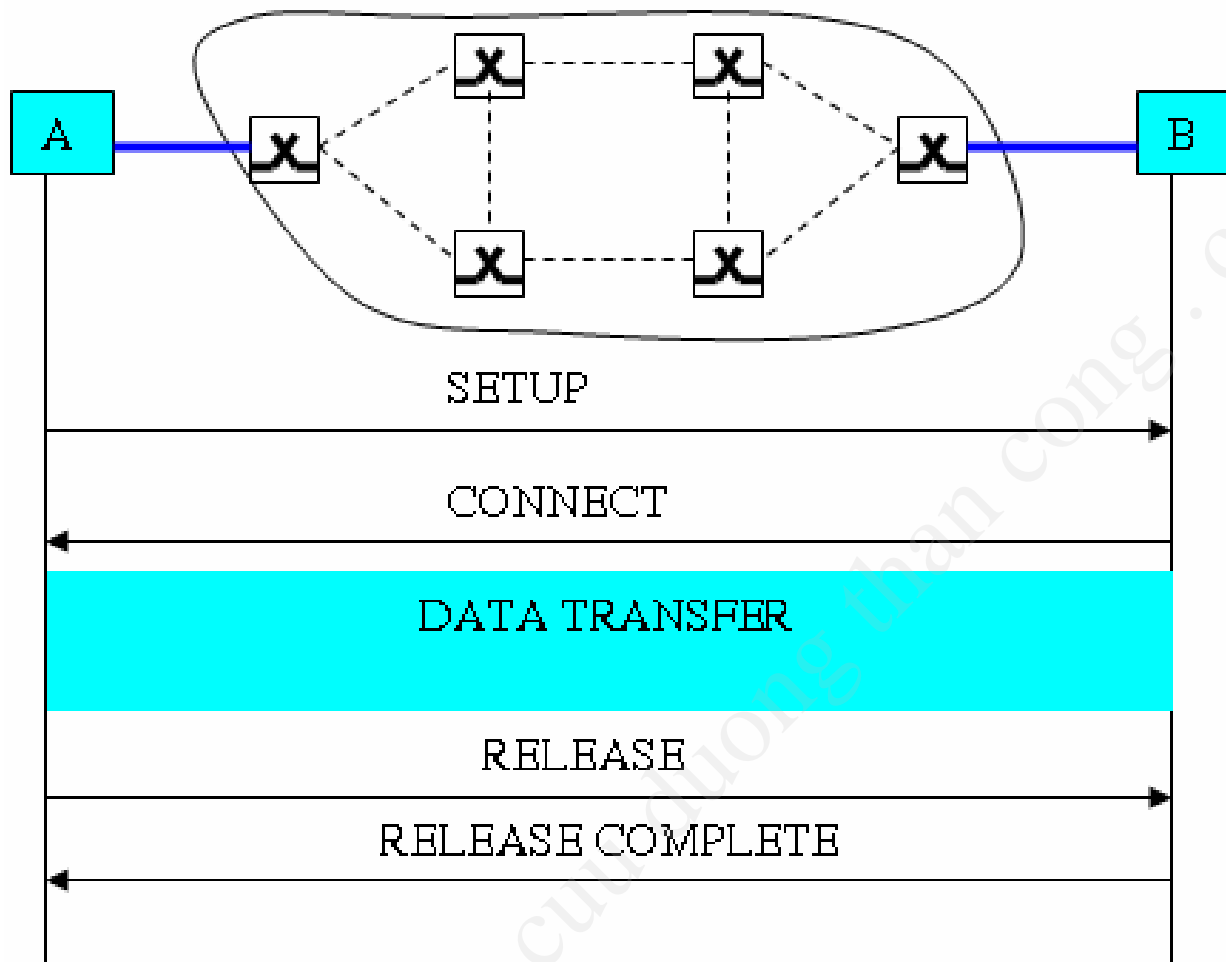
- Hai DTE có kết nối mạch ảo thường xuyên theo 2 định danh DLCI
- Quá trình truyền dữ liệu không phải qua giai đoạn thiết lập và giải tỏa kênh truyền.
- DTE A gửi dữ liệu đến DTE B theo DLCI 122
- DTE B gửi dữ liệu đến DTE A theo DLCI 077



⋮ Hình PVC DLCIs

PVC: Permanent Virtual Circuit: Mạch ảo thường xuyên

# HOẠT ĐỘNG THEO KIỂU MẠCH ẢO CHUYỂN MẠCH



- Mạch ảo chỉ được thiết lập trong quá trình truyền dữ liệu
- Trước khi truyền dữ liệu phải thực hiện thiết lập kênh truyền
- Sau khi truyền dữ liệu phải giải tỏa kênh truyền

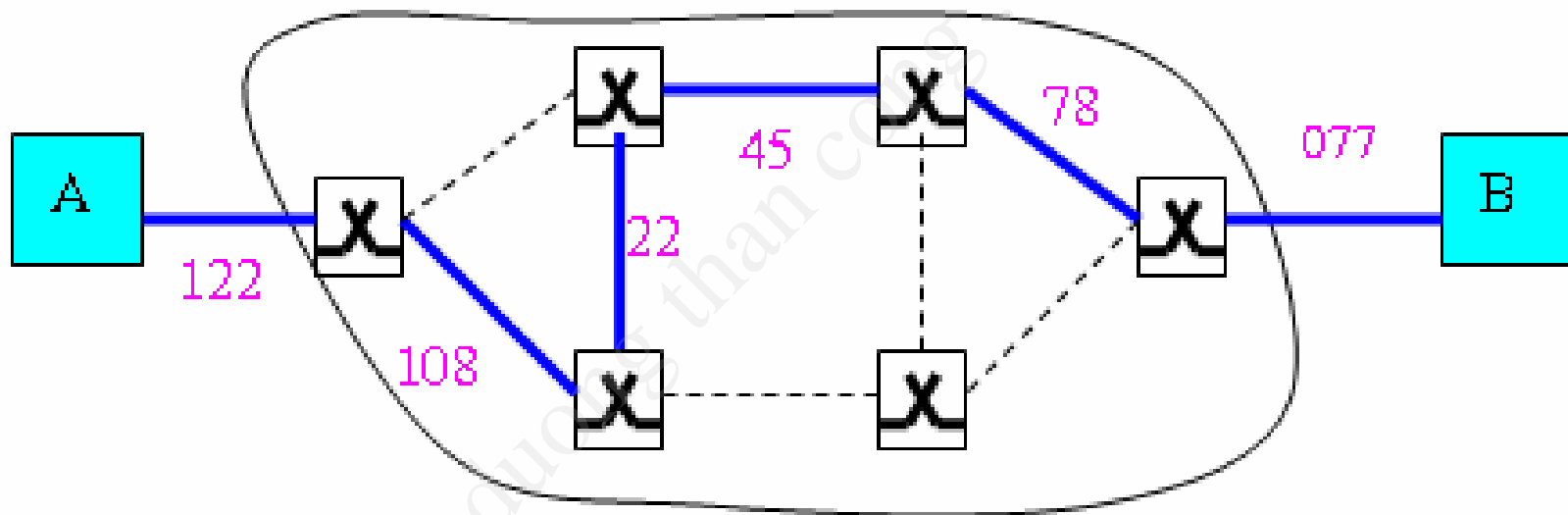
Hình Thiết lập và giải tỏa SVC

SVC: Switched Virtual Circuit



# **DLCI BÊN TRONG MẠNG**

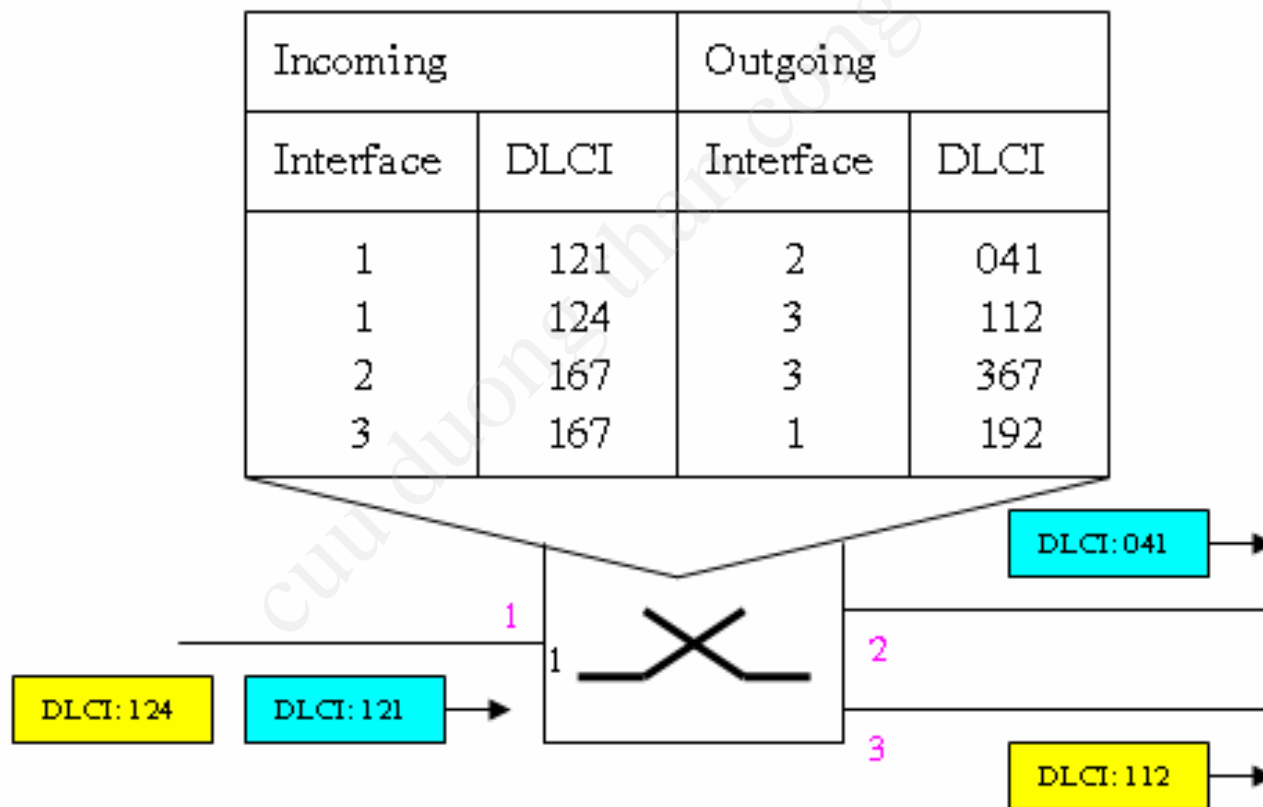
- DLCI được gán cho mạch ảo giữa các DCE bên trong mạng



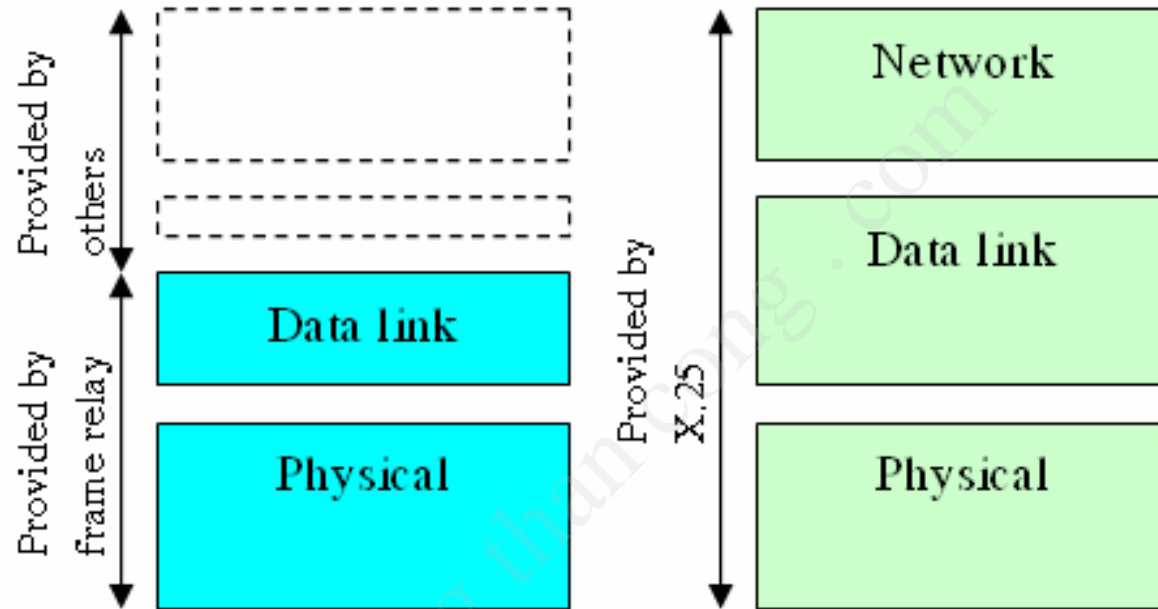
Hình DLCIs bên trong mạng

# HOẠT ĐỘNG CỦA BỘ CHUYỂN MẠCH

- Bộ chuyển mạch trong frame relay hoạt động theo nguyên tắc bảng định tuyến. Mỗi gói dữ liệu ngõ vào sẽ được chuyển đến ngõ ra tương ứng trên cơ sở DLCI



# CÁC LỚP TRONG FRAME RELAY



Lưu ý:

- frame relay chỉ có 1,5 lớp so với X.25 có 3 lớp
- frame relay loại bỏ tất cả chức năng các lớp mạng và một phần lớp liên kết dữ liệu

# LỚP LIÊN KẾT DỮ LIỆU TRONG FRAME RELAY

C/R: common/response

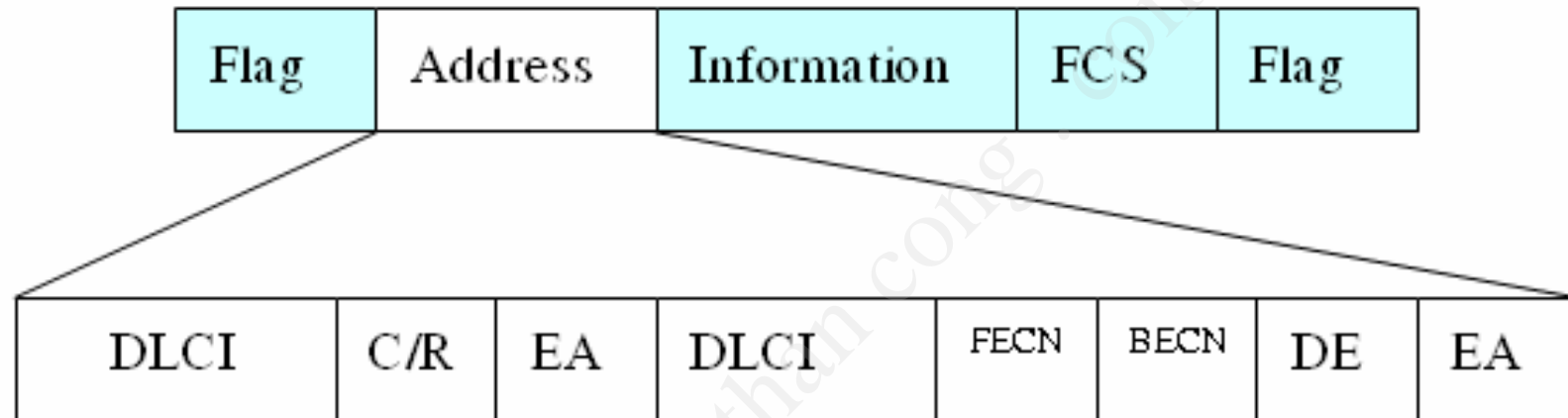
EA: Extended address

FECN: Forward explicit congestion notification

BEcn: Backward explicit congestion notification

DE: Discard eligibility

DLCI: Data link connection identifier



Hình Khung frame relay

Lưu ý:

- Lớp liên kết dữ liệu trong frame relay hoạt động theo nghi thức LAPF, dựa trên cơ sở là nghi thức HDLC
- LAPF: Link Access Protocol for Frame Relay

# Ý NGHĨA CÁC FIELD TRONG FRAME RELAY

Lưu ý:

- **DLCI** : định danh mạch ảo, xác định tuyến gói dữ liệu được gửi đi
- **C/R** xác định khung là Command hay Response
- **EA**: *Extended Address*, cho biết byte hiện hành có phải là byte cuối cùng của địa chỉ hay không. Nếu bằng 0 nghĩa là còn 1 byte tiếp theo.
- **FECN**: (*Forwardward Explicit Congestion Notification*) Được đặt bởi bộ chuyển mạch để chỉ ra lưu lượng đang nghẽn theo hướng khung dữ liệu đang đi.
- **BECN** (*Backward Explicit Congestion Notification*) Được đặt bởi bộ chuyển mạch để chỉ ra lưu lượng đang nghẽn theo hướng ngược lại với hướng mà khung dữ liệu đang đi.
- **DE** (*Discard Eligibility*) chỉ ra mức ưu tiên của khung. Khung có DE=1 có độ ưu tiên cao hơn so với khung có DE=0. DE được đặt bởi trạm phát hay các trạm chuyển mạch trên mạng

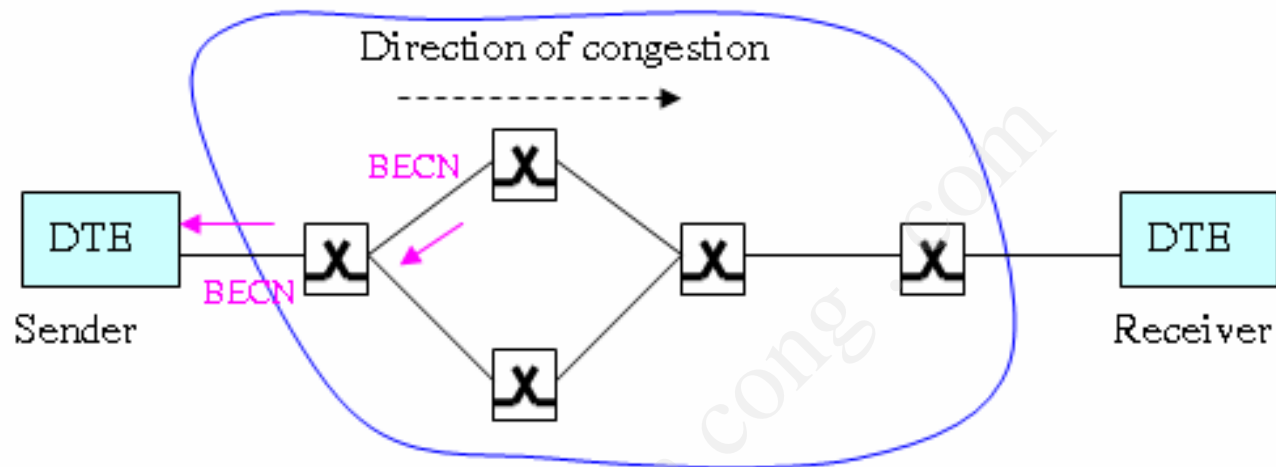
# NGHẼN (CONGESTION)

---

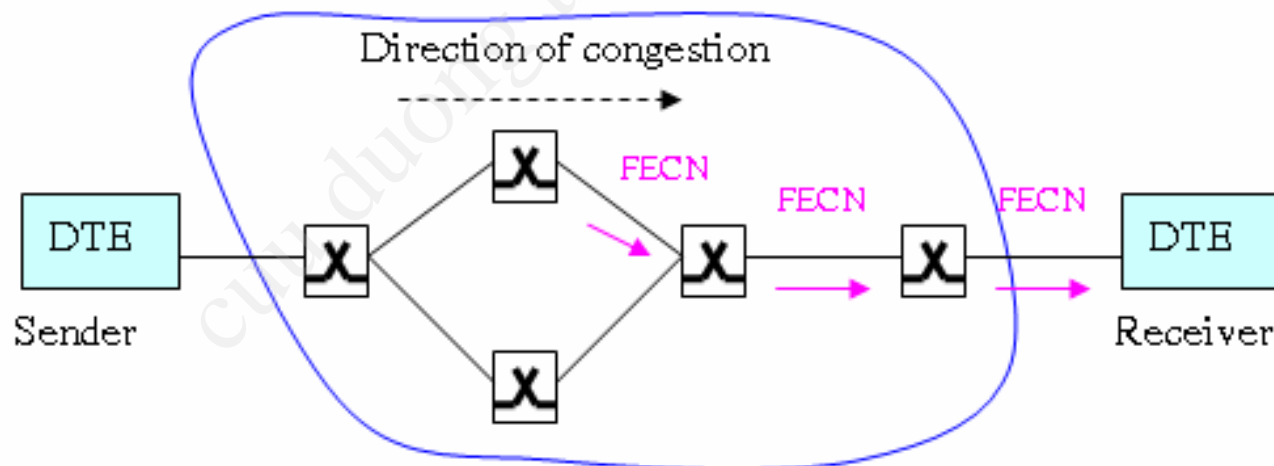
## Lưu ý:

- Hiện tượng nghẽn trong mạng xảy ra khi người sử dụng gửi dữ liệu vào mạng với tốc độ lớn hơn tốc độ cho phép bởi tài nguyên mạng
- X.25 kiểm soát luồng tại 2 lớp, lớp liên kết dữ liệu thực hiện kiểm soát luồng theo cấp nút-nút, kiểm soát luồng tại lớp mạng thực hiện tại cấp đầu cuối - đầu cuối
- Frame relay không có lớp mạng, tại lớp liên kết dữ liệu frame relay cũng không thực hiện kiểm soát luồng. Frame relay cho phép truyền dữ liệu tốc độ bùng phát

# NGHẼN (CONGESTION)

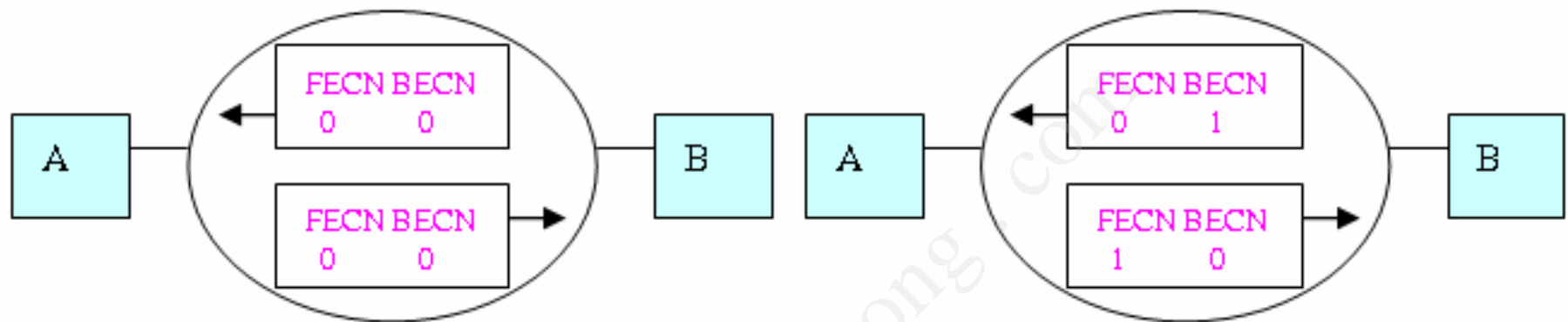


Hình Nguyên lý BECN



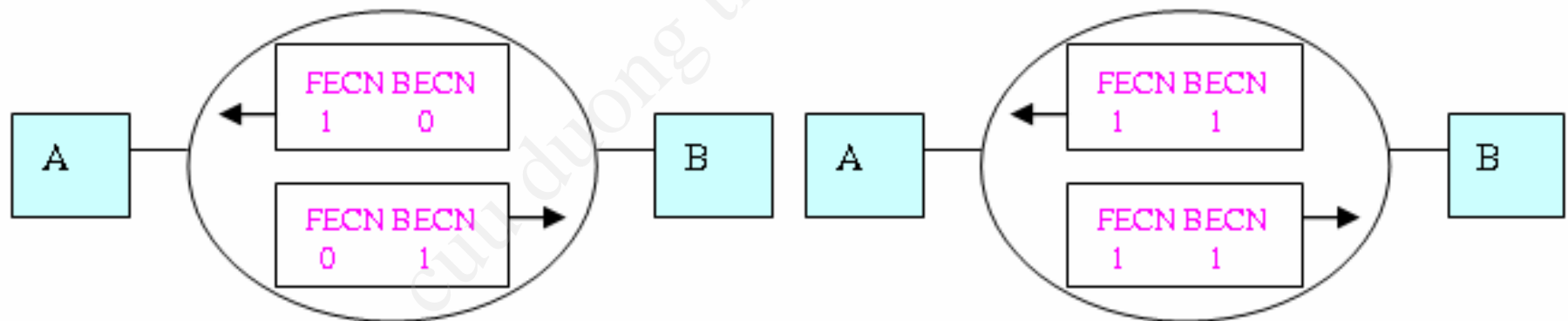
Hình Nguyên lý FECN

# CÁC KHẢ NĂNG NGHẼN



a. No congestion

b. Congestion in the direction A-B



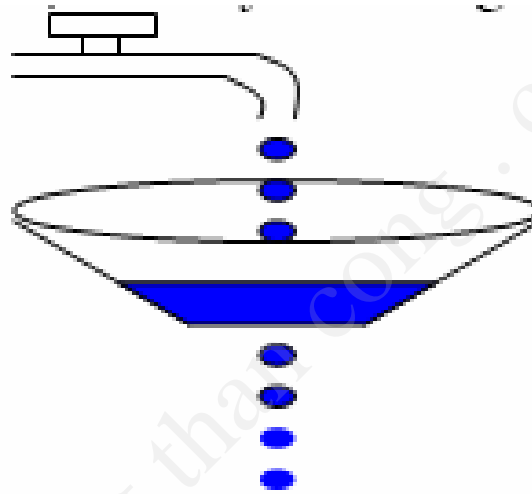
c. Congestion in the direction B-A

d. Congestion in both direction



# GIẢI THUẬT THÙNG RÒ

- Kiểm soát lưu lượng theo giải thuật thùng rò

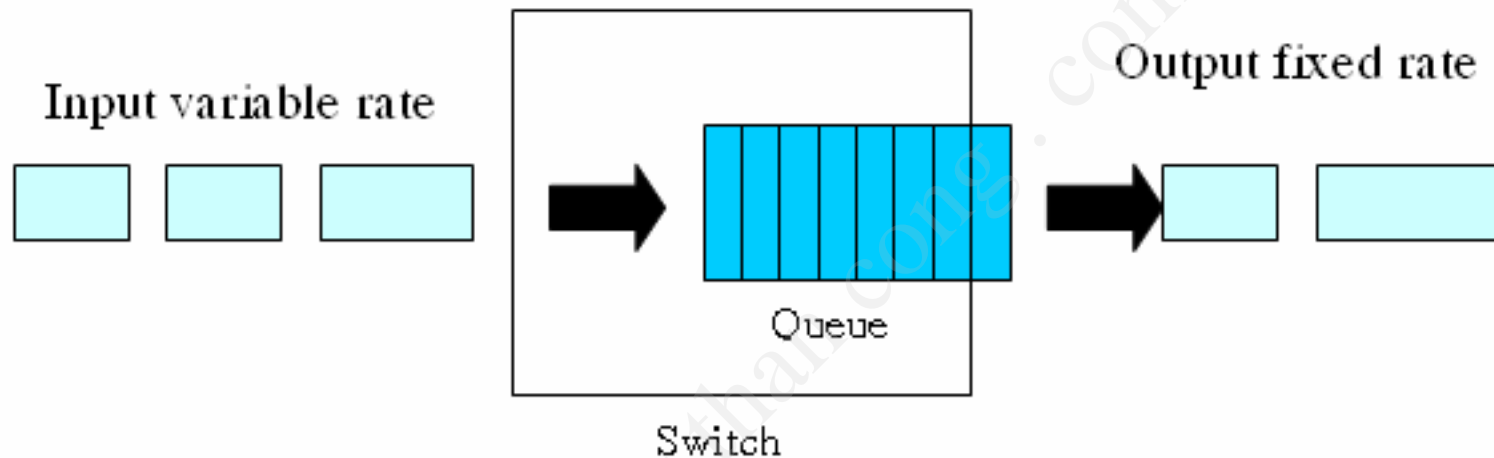


## Đặc điểm

- Tốc độ nước chảy ra cố định
- Tốc độ nước chảy vào có thể bị thay đổi
- Nếu lượng nước chảy vào nhiều hơn so với lượng nước chảy ra thì thùng sẽ bị tràn

# GIẢI THUẬT THÙNG RÒ

- Kiểm soát lưu lượng theo giải thuật thùng rò

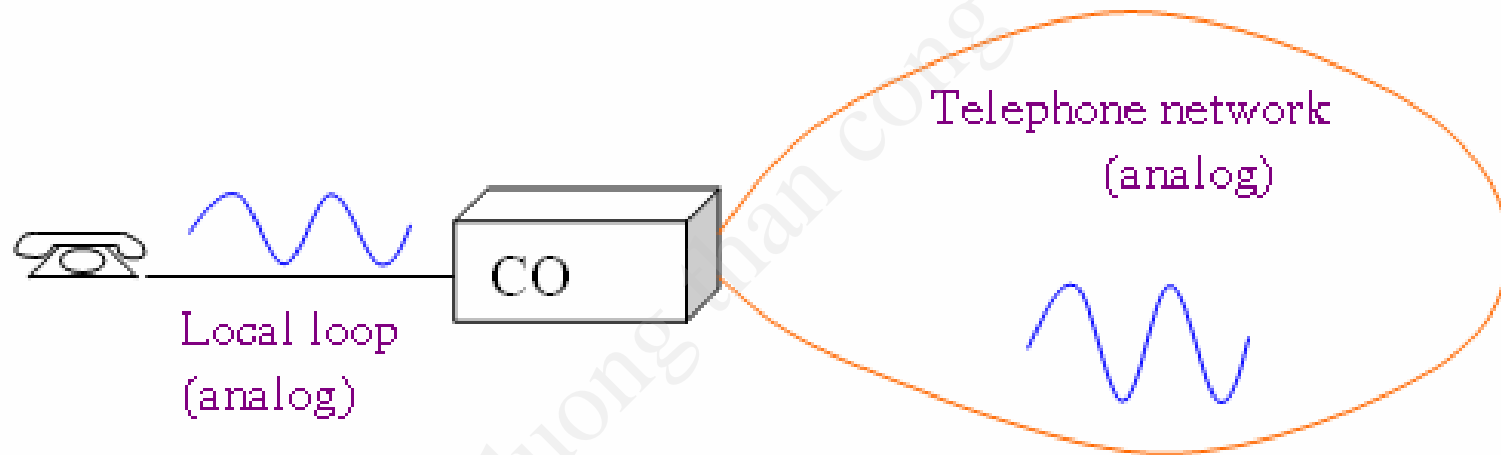


## Đặc điểm

- Tốc độ dữ liệu ra là cố định (ví dụ: 1,544Mbps)
- Tốc độ dữ liệu vào có thể thay đổi bất phát
- Bộ chuyển mạch sử dụng một bộ đệm (hàng đợi) để lưu dữ liệu trước khi phát

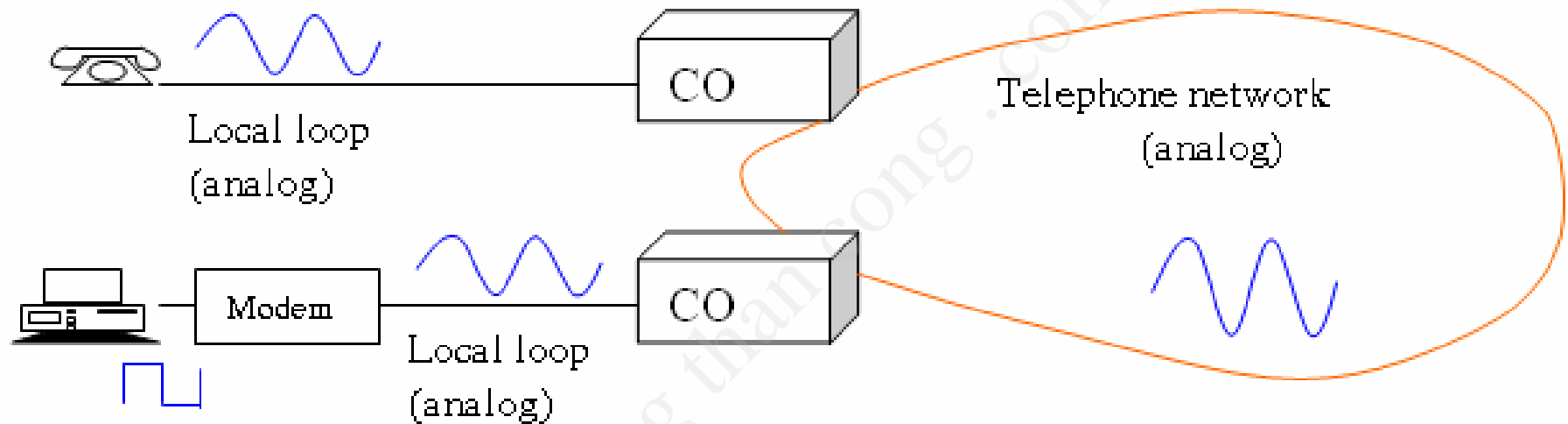
# ISDN

- ISDN: Integrated Services Digital Network, mạng số liên kết dịch vụ
- Mục tiêu của mạng ISDN là cung cấp các dịch vụ số tích hợp cho người sử dụng

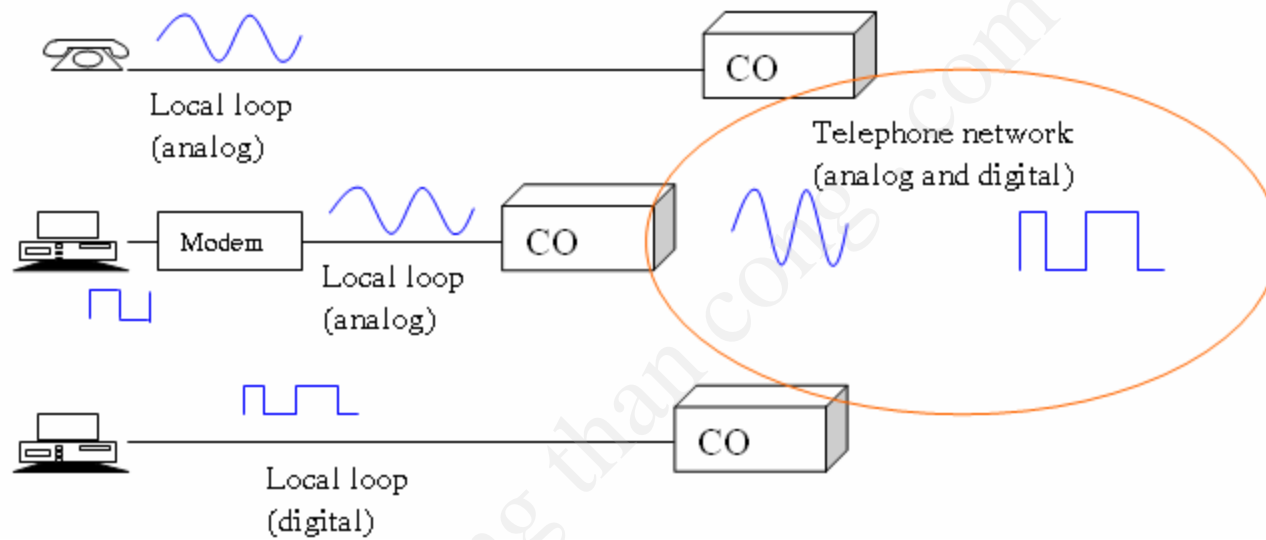


Thông tin thoại trên mạng điện thoại tương tự

# ISDN

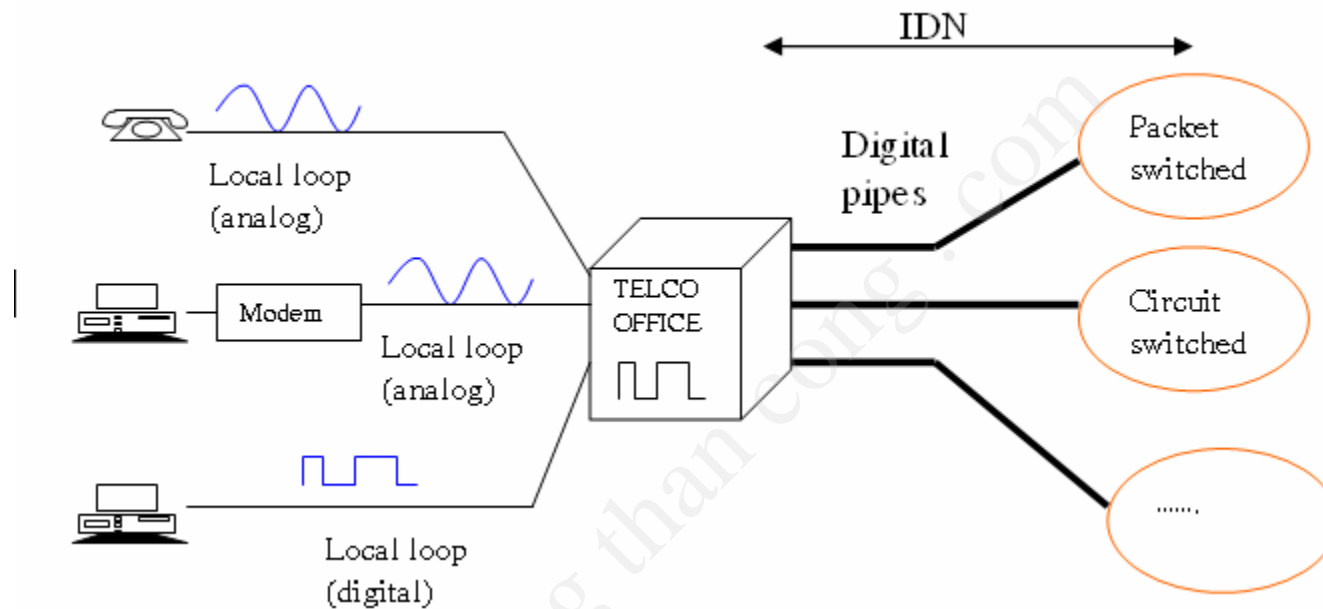


Thông tin thoại (analog) và dữ liệu (digital) trên mạng điện thoại tương tự



Các dịch vụ tương tự và số trên mạng điện thoại tương tự và số

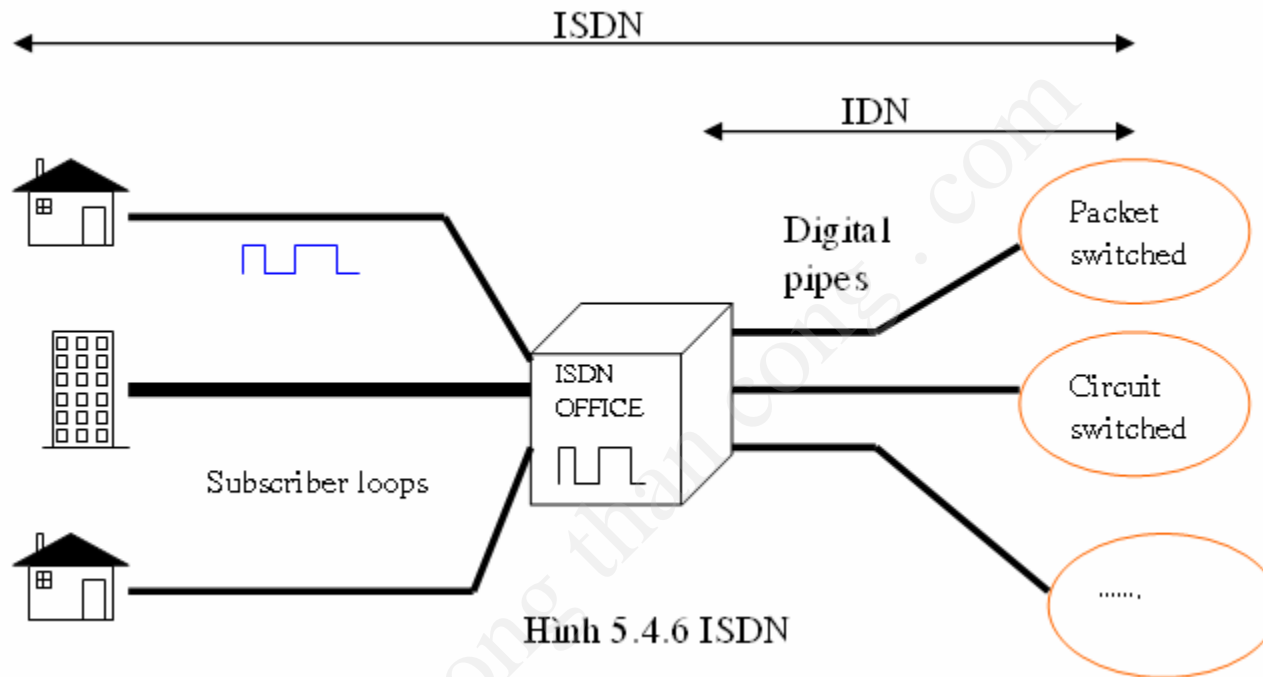
# ISDN



Các dịch vụ tương tự và số trên mạng số liên kết IDN

➤ IDN: Integrated Digital Network, Mạng dữ liệu số tích hợp

# ISDN



## Mạng số liên kết dịch vụ ISDN

- ISDN: Integrated Services Digital Network, mạng số liên kết dịch vụ

# **S'DH, ATM**

---

- Tài liệu tham khảo

cuu duong than cong . com



cuu duong than cong . com