

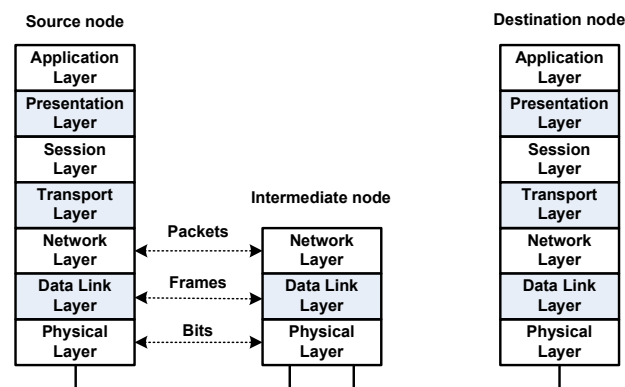
## Tầng liên kết dữ liệu

- Chức năng của tầng liên kết dữ liệu (DLL)
- Địa chỉ MAC và tính chất
- Phương pháp framing
- Điều khiển luồng
- Kiểm soát lỗi
- Các giao thức của tầng liên kết dữ liệu: HDLC và PPP
- Bridge: transparent và sourouting

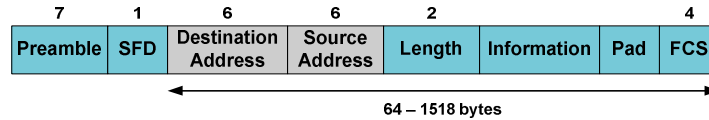


## • Chức năng của tầng liên kết dữ liệu

- ✓ Khả năng truyền khung tin cậy trên đường liên kết vật lý không tin cậy



✓ Địa chỉ MAC và tính chất: IEEE 802.3 MAC Frame



#### o Destination address

- Single
- Group
- Broadcast: 11...11

#### o Addresses

- Local or global

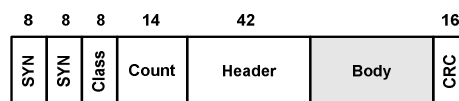
#### o Global addresses ( $2^{46}$ possible)

- First 24 bits assigned to vendors (OUI)
- Next 24 bits assigned by vendors
- CISCO: 00-00-0C
- 3COM: 02-60-8C



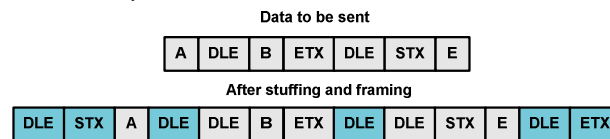
### • Phương pháp Framing

- ✓ DLL tổ chức bit stream thành các frame
- ✓ Bắt đầu và kết thúc của frame được xác định: Character count, control character, flag, ...
- ✓ **Character count**
  - o Số ký tự được chỉ ra trong vị trí xác định của header
  - o Bên nhận đếm số ký tự để xác định nơi kết thúc frame
  - o Ví dụ: DDCMP (Digital data comm. message protocol)



✓ **Control character**

- o Mỗi frame bắt đầu và kết thúc với chuỗi ký tự đặc biệt
  - Ví dụ: bắt đầu với DLE STX (data link escape start of text) và kết thúc với DLE ETX (end of text)
- o Phần dữ liệu phải là số nguyên lần các ký tự
- o **Character stuffing**: chèn DLE trước mỗi ký tự DLE trong dữ liệu, ngăn chặn khả năng xuất hiện các ký tự bắt đầu và kết thúc xuất hiện bên trong frame
  - Ví dụ: BISYNC



- STX: 0x02, ETX: 0x03
- DLE: 0x01 (non-printable)

✓ **Flagging**

- o Mỗi frame bắt đầu và kết thúc bằng chuỗi ký tự đặc biệt – flag
- o Flag byte: 01111110 (0x7E)
- o Phần dữ liệu của frame có thể chứa số bit bất kỳ
- o **Bit stuffing**: chèn bit 0 vào sau mỗi chuỗi 5 bit 1 liên tiếp
- o Ví dụ: SDLC, HDLC, LAPB

01111110111101011110011110111110011111110  
 Flag                      Frame                      Flag



### • Điều khiển luồng (flow control)

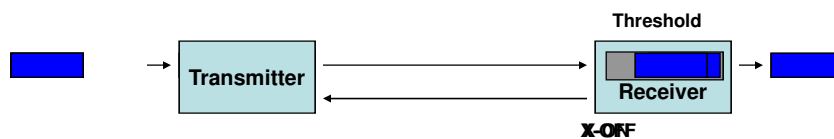
#### ✓ Khái niệm điều khiển luồng

- o Kích thước bộ đệm để lưu giữ các frame nhận được bị giới hạn
- o Tràn bộ đệm có thể xảy ra nếu tốc độ xử lý tại phía thu chậm hơn tốc độ truyền frame
- o Điều khiển luồng ngăn chặn tràn bộ đệm bằng cách điều khiển tốc độ truyền dẫn từ phía phát (Tx) đến phía thu (Rx)
- o Các phương pháp phổ biến:
  - X-ON/X-OFF
  - Stop and wait
  - Sliding window



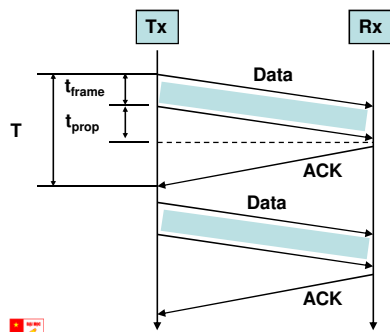
#### ✓ X-ON/X-OFF

- o Rx truyền X-OFF (DC3) nếu bộ đệm đầy
- o Tx dừng truyền khi nhận được X-OFF
- o Rx truyền X-ON (DC1) khi bộ đệm được giải phóng
- o Tx bắt đầu truyền lại khi nhận được X-ON
- o Có thể sử dụng các frame RR (Receive Ready) và RNR (Receive not Ready)
- o Nếu số bit trong bộ đệm vượt quá ngưỡng  $2t_{prop} \cdot R$ , X-OFF được gửi tới Tx
- o Phương pháp: Stop-and-wait, sliding window



### o Stop-and-wait

- Tx ngừng truyền frame tiếp theo cho đến khi nhận được ACK từ Rx
- Rx truyền frame ACK khi đã sẵn sàng nhận frame tiếp theo
- Đơn giản, nhưng không hiệu quả khi trễ đáp ứng đường truyền lớn



#### Utilization:

$$U = \frac{t_{\text{frame}}}{2t_{\text{prop}} + t_{\text{frame}}} = \frac{1}{2a + 1}, \quad a = \frac{t_{\text{prop}}}{t_{\text{frame}}} = \frac{d/v}{L/R}$$

❖ Satellite link:  $t_{\text{prop}} = 270 \text{ ms}$ ,  $L = 500 \text{ byte}$

$R = 56 \text{ kbps} \rightarrow t_{\text{frame}} = 4/56 = 71 \text{ ms}$ ,

$\rightarrow a = t_{\text{prop}} / t_{\text{frame}} = 270/71 = 3.8$ ,

$\rightarrow U = 0.12 = 12\%$  của 56 kbps

❖ Short link:  $t_{\text{prop}} = 5 \mu\text{s}$ ,  $L = 500 \text{ byte}$

$R = 10 \text{ Mbps} \rightarrow t_{\text{frame}} = 4/10 \text{ M} = 400 \mu\text{s}$ ,

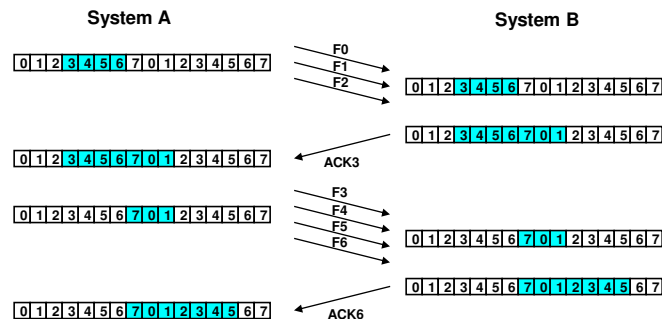
$\rightarrow a = t_{\text{prop}} / t_{\text{frame}} = 5/400 = 0.012$

$\rightarrow U = 0.98 = 98\%$  của 10 Mbps



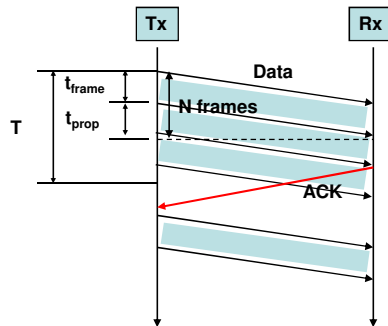
### o Sliding window

- Window  $W_S$  bằng kích thước bộ đệm của bên nhận có thể sử dụng để nhận các frame liên tiếp từ Tx không cần ACK
  - ❖ Có thể tránh hiện tượng tràn bộ đệm của Rx
  - ❖ Phải chọn  $W_S$  lớn hơn  $a$  (delay of bandwidth)



Utilization:

$$U = \frac{N t_{\text{frame}}}{2t_{\text{prop}} + t_{\text{frame}}} = \begin{cases} 1 & N \geq 1 + 2a \\ \frac{N}{1 + 2a} & N < 1 + 2a \end{cases}, \quad a = \frac{t_{\text{prop}}}{t_{\text{frame}}} = \frac{d/v}{L/R}$$



• Kiểm soát lỗi (error control)

✓ Khái niệm

- o Lỗi do môi trường truyền dẫn
- o Quá trình kiểm soát lỗi có 2 chức năng: phát hiện và sửa lỗi
- o 2 loại lỗi phổ biến: mất frame và lỗi frame

✓ Các phương pháp kiểm soát lỗi

- o Forward error control: sửa lỗi tại phía Rx (Forward Error Correction)
- o Backward error control: truyền lại frame (Automatic Repeat Request)
  - ➔ cần backward channel



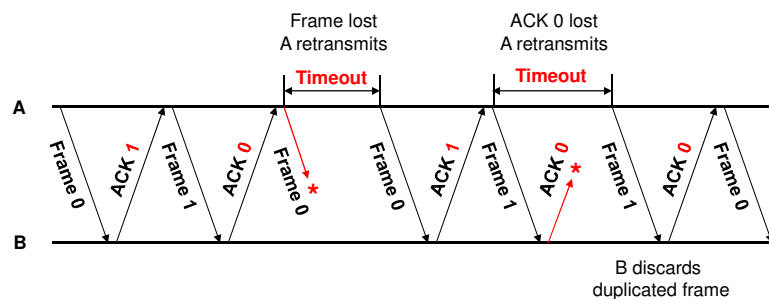
- ✓ **Automatic Repeat reQuest (ARQ)**

- o Cơ chế
  - Phát hiện lỗi
  - Positive acknowledgement (ACK): frame nhận OK tại Rx
  - Truyền lại sau timeout
  - Negative acknowledgement (NACK): truyền lại frame
- o Các phương pháp ARQ
  - Stop and Wait ARQ
  - Continuous ARQ
    - ❖ Go-back-N ARQ
    - ❖ Selective-reject (or repeat) ARQ



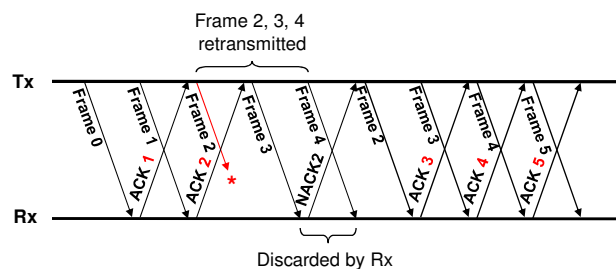
- **Stop and wait ARQ**

- Tx gửi 1 frame và đợi ACK từ Rx trước khi truyền next frame
- Tx phải giữ bản copy của frame đã gửi đến khi nhận được ACK
- Sử dụng timeout cho các frame hoặc ACK bị mất
- Đánh số các frame gửi và nhận để nhận biết khi lặp frame
- Ví dụ: sử dụng 1 bit để đánh số frame



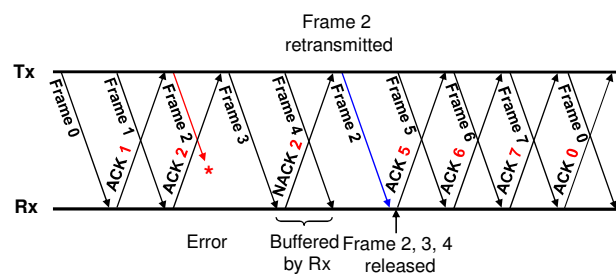
- **Go-back-N ARQ**

- Tx có thể truyền liên tiếp các frame
- Rx gửi negative acknowledgement (REJ) khi phát hiện lỗi
- Tx phải truyền lại tất cả các frame từ frame bị lỗi
- Tx phải giữ bản copy của tất cả các frame đã gửi



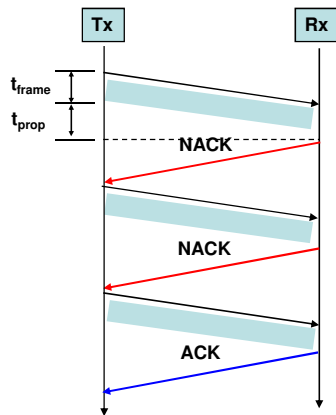
- o **Selective-repeat ARQ**

- Tx có thể truyền liên tiếp các frame
- Rx gửi negative acknowledge (SREJ) khi phát hiện lỗi
- Rx lưu lại tất cả các frame OK sau frame bị lỗi
- Tx chỉ gửi lại duy nhất frame bị lỗi
- Rx phải sắp xếp lại các frame đã lưu sau khi nhận được frame truyền lại





▪ Performance: **Stop-and-wait ARQ**



❖  $P$ : xác suất lỗi khung

❖  $N_f$  = số lần truyền khung trung bình

$$U = \frac{1}{N_f} U_{\text{error-free}}$$

$$N_f = \sum_i i \cdot P^{i-1} (1-P) = \frac{1}{1-P}$$

$$\Rightarrow U = \frac{1}{N_f} \cdot \frac{1}{1+2a} = \frac{1-P}{1+2a}$$



▪ Performance: **Go-back-N ARQ**

❖  $P$ : xác suất lỗi khung

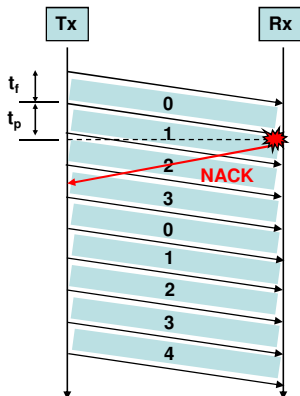
❖  $N_f$  = số lần truyền khung trung bình

$$N_f \approx 1 \cdot (1-P) + (1+K) \cdot P \cdot (1-P) + (1+2K) \cdot P^2 \cdot (1-P) + \dots + \frac{1+P(K-1)}{1-P}$$

•  $N \geq 1+2a$ :  $K = 1+2a$

•  $N < 1+2a$ :  $K = N$

$$\Rightarrow U = \begin{cases} \frac{1-P}{1+P(K-1)} = \frac{1-P}{1+2aP}, & N \geq 1+2a \\ \frac{1-P}{1+P(K-1)} \cdot \frac{N}{1+2a} = \frac{N(1-P)}{(1+2a)(1-P+NP)}, & N < 1+2a \end{cases}$$



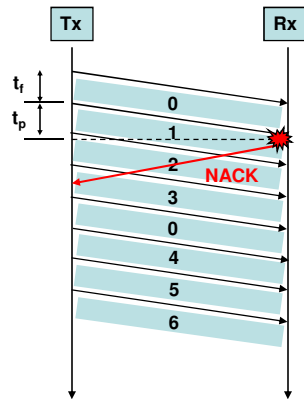
▪ Performance: **Selective-repeat ARQ**

❖ P: xác suất lỗi khung

❖  $N_t$  = số lần truyền khung trung bình

$$N_t = \sum i \cdot P^{i-1} (1-P) = \frac{1}{1-P}$$

$$U = \begin{cases} 1-P, & N \geq 1+2a \\ \frac{N(1-P)}{1+2a}, & N < 1+2a \end{cases}$$



• Các giao thức của tầng liên kết dữ liệu

✓ High Level Data Link Control (HDLC)

- o ISO phát triển từ IBM synchronous DLC (SDLC)
- o Cơ chế chèn bit (bit-stuffing), data transparency
- o Hỗ trợ truyền đồng bộ, HDX, FDX, point-to-point
- o Flow control: X-ON / X-OFF
- o Error control: Go-back-N, Selective-repeat ARQ

✓ Point to point protocol (PPP)

- o Tạo kết nối điểm-điểm
  - Router-router leased line (PPP), dial-up host-router (PPP, SLIP)
- o Chuẩn chính thức của Internet, làm việc ở tầng liên kết dữ liệu
- o Làm việc trên dial-up tel line, SONET, ADSL, X25, ISDN, ...
- o Chức năng: error detection, IP address negotiation, authentication

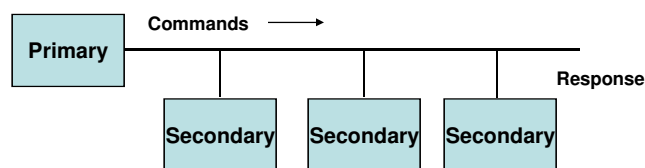


## ✓ HDLC

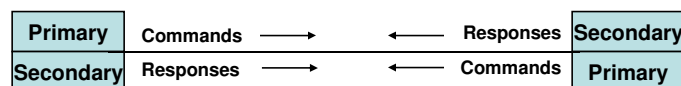
- o Họ giao thức HDLC
  - SDLC: IBM
  - HDLC: ISO
  - Link Access Procedure Balanced (LAPB): X25
  - Link Access Procedure for D-channel (LAPD): ISDN
  - PPP: Internet
  - Logical Link Control (LCC): IEEE
  - Frame relay
- o Chế độ hoạt động của HDLC
  - Configurations: Balanced, unbalanced link
  - Mode: Normal Response Mode (NRM) – unbalanced multipoint  
Asynchronous Response Mode (ARM) – unbalanced PPP  
Asynchronous Balanced Mode (ABM) – balanced PPP



- NRM: polling multidrop lines



- ABM: Asynchronous Balanced Mode

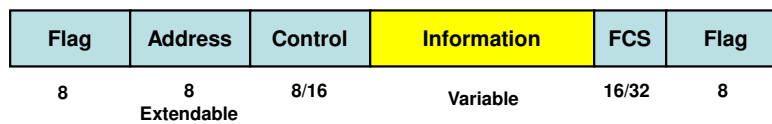


- Mode được chọn khi thiết lập liên kết



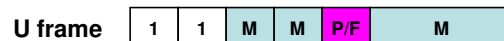
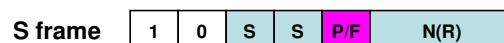
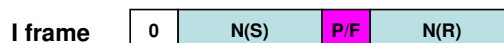
## o HDLC frame format

- Flag (01111110): Bắt đầu và kết thúc frame, sử dụng trong đồng bộ
- Address (có thể mở rộng): xác định secondary trong multidrop link
- Control: Xác định dạng frame
- FCS: CRC-16 hoặc 32



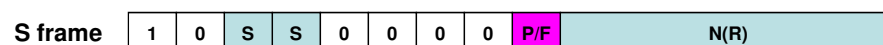
## o HDLC frame types: I-frame, U-frame, S-frame

## 8-bit control field format



N(S): send sequence #  
 N(R): receive sequence #  
 S: super functions bits  
 M: Unnumbered functions bits  
 P/F: Poll / Final bits

## 16-bit control field format



- **I-frame:** User data
  - ❖ Mỗi frame có số thứ tự  $N(S)$
  - ❖ Cơ chế ACK:
    - +  $N(S)$  = số thứ tự của frame đang gửi
    - +  $N(R)$  = số thứ tự của frame đợi nhận tiếp theo, xác nhận đã nhận OK các frame có số thứ tự đến  $N(R)-1$
  - ❖ Sử dụng 3 hoặc 7 đánh số thứ tự các frame: Kích thước cửa sổ cực đại là 7 hoặc 127
  - ❖ P/F: Trong NRM, chế độ hồi vòng của primary ( $P=1$ ), secondary thiết lập  $F=1$  khi gửi đáp ứng với khung I cuối cùng



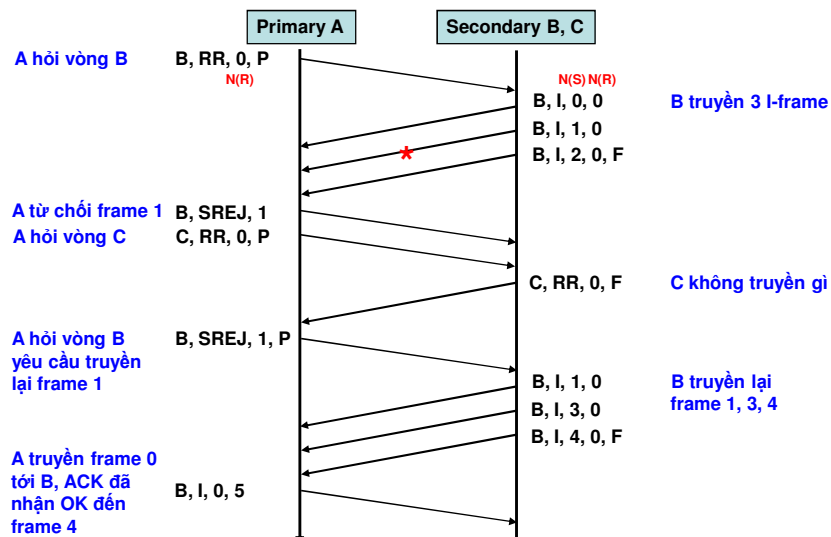
- **S-frame:** error control + flow control
  - ❖ Receive Ready (RR), Receive Not Ready (RNR), Reject (REJ): Go-back-N ARQ, Selective Reject (SREJ): Selective-repeat ARQ
  - ❖ RR: SS = 00, ACK đã nhận OK  $N(R)-1$  frame
  - ❖ REJ: SS = 01, NACK frame  $N(R)$  đầu tiên có lỗi, Tx phải gửi lại frame  $N(R)$  và các frame sau đó
  - ❖ RNR: SS = 10, ACK đã nhận OK frame  $N(R)-1$ , không nhận tiếp các I frame
  - ❖ SREJ: SS = 11, NACK cho frame  $N(R)$  và yêu cầu truyền lại frame này



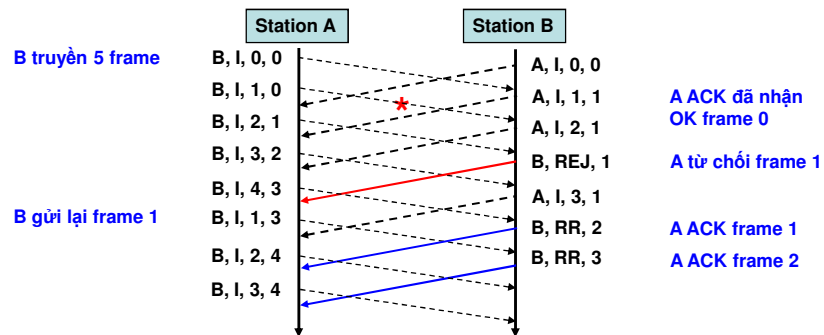
- **U-frame:** Link initialization, maintenance, disconnection
  - ❖ Cung cấp commands + response: mode settings, recovery
  - ❖ Mode settings: thiết lập chế độ liên kết
    - + SABM: Set Asynchronous Balanced Mode
    - + UA: ACK đã chấp nhận các lệnh thiết lập chế độ
    - + DISC: Hủy bỏ liên kết logic đã thiết lập
  - ❖ Truyền tin giữa các trạm sử dụng unnumbered info (UI)
  - ❖ Recovery: khi error/flow control không thực hiện được
    - + FRMR: frame có FCS đúng, nhưng sai cú pháp
    - + RSET: Tx khởi tạo lại số thứ tự các frame được gửi



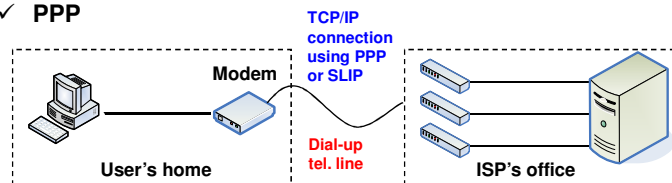
- Ví dụ: HDLC sử dụng NRM (polling)



- Ví dụ: HDLC sử dụng ABM



✓ **PPP**



Flag 01111110	Address 11111111	Control 00000011	Protocol	Information	CRC	Flag 01111110
Integer # of bytes						

- o Giống HDLC, nhưng là giao thức hướng byte
  - Flag 01111110 được coi như là ký tự 0x7E
  - Chèn byte (byte stuffing) sử dụng ký tự đặc biệt: 0x7D
- o Address: luôn luôn là 11111111 → các trạm đều nhận được frame
- o Protocol: cho biết loại packet chứa trong information (IP, IPX, LCP, NCP ...)

- o Link Control Protocol (LCP): Quản lý kết nối, hỗ trợ multilink (kết hợp nhiều kết nối vật lý vào 1 đường kết nối logic), authentication (sử dụng ID và password: Password Authentication Protocol - PAP)
- o Network Control Protocol (NCP): hỗ trợ IP, IPX, Decnet, Apple Talk), dynamic IP address assignment
- o Byte stuffing in PPP
  - Control escape: 0x7D
  - Khi xuất hiện flag hoặc control escape bên trong frame, chèn vào 0x7D (01111101) và dữ liệu tương ứng được XOR với 0x20 (00100000)

Dữ liệu cần truyền

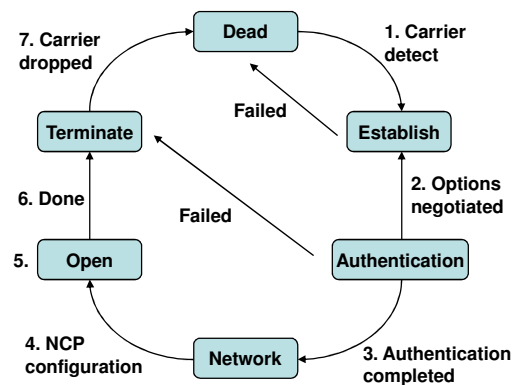
41	7D	42	7E	50	70	46
----	----	----	----	----	----	----

Sau khi stuffing và framing

7E	41	7D	5D	42	7E	5E	50	70	46	7E
----	----	----	----	----	----	----	----	----	----	----



- o PPP phase



## Home PC kết nối với ISP

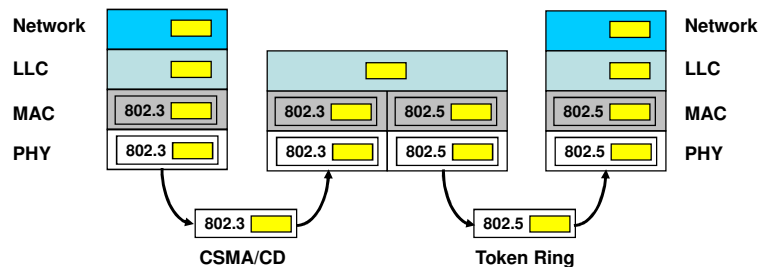
1. PC calls router via modem
2. PC and router exchange LCP packets to negotiate PPP
3. Check on identities
4. NCP packets exchanged to configure the network layer TCP/IP (require IP address assignment)
5. Data transport: send/receive IP packets
6. NCP used to tear down the network layer connection (free up IP address); LCP used to shut down data link layer connection
7. Modem hangs up



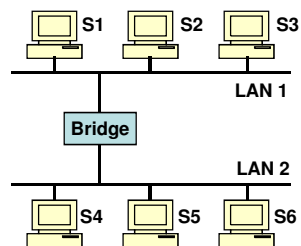


• Bridge: transparent và source routing

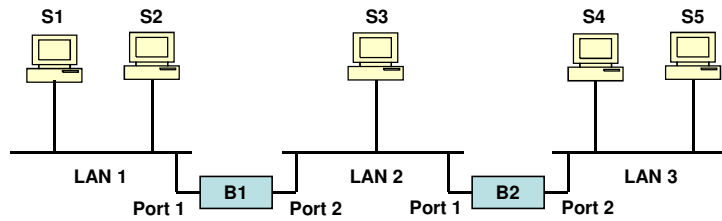
- ✓ Repeater, bridge: làm việc ở tầng liên kết dữ liệu để liên kết nhiều phân lớp mạng với MAC format khác nhau, data rate khác nhau, kích thước frame khác nhau
- ✓ Tổng quát về bridge



- ✓ Transparent bridge
  - o Kết nối IEEE LAN đảm bảo data transparency
  - o Sử dụng bảng tham chiếu (table lookup)
  - o Sử dụng phương pháp học sau để xây dựng bảng tham chiếu
    - Theo dõi địa chỉ nguồn của mạng LAN gửi đến
    - Theo dõi sự thay đổi cấu trúc mạng để cập nhật bảng tham chiếu



o Ví dụ

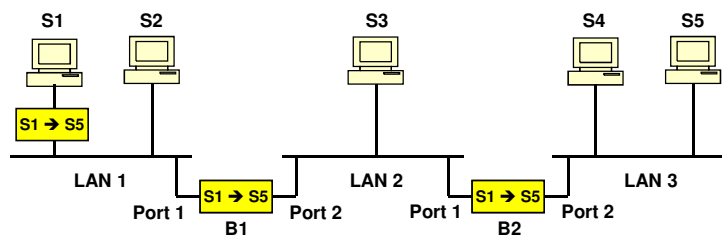


Address	Port

Address	Port



S1 → S5

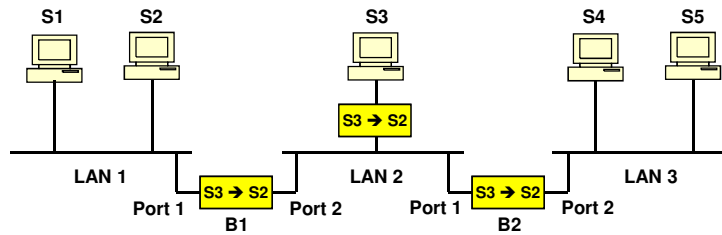


Address	Port
S1	1

Address	Port
S1	1



S3 → S2

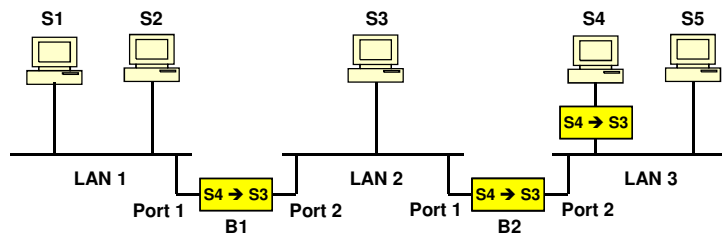


Address	Port
S1	1
S3	2

Address	Port
S1	1
S3	1



S4 → S3

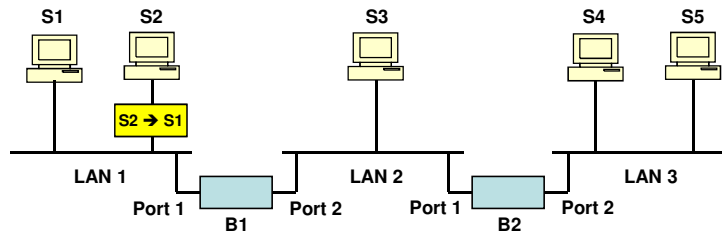


Address	Port
S1	1
S3	2
S4	2

Address	Port
S1	1
S3	1
S4	2



S2 → S1



Address	Port
S1	1
S3	2
S4	2
S2	1

Address	Port
S1	1
S3	1
S4	2

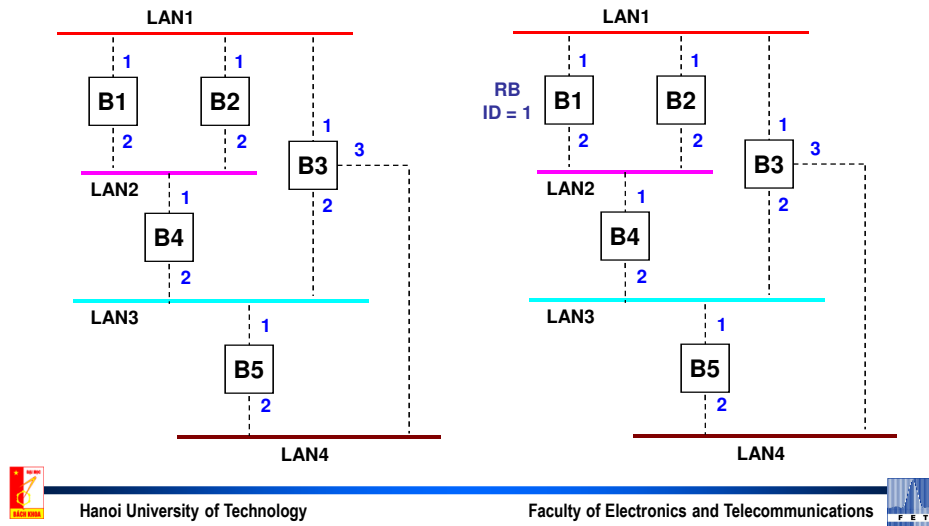


- ✓ Ngăn chặn vòng lặp sử dụng **spanning tree algorithm**
  - o Chọn bridge có **root ID** thấp nhất làm **root bridge (RB)**
  - o Xác định **root port (R)** cho từng bridge, trừ **RB**: port có đường đi ngắn nhất (**least-cost**) tới **RB**
  - o Chọn **designated bridge (D)** cho từng mạng LAN:
    - + bridge có đường đi ngắn nhất từ mạng LAN tới **RB**
    - + **Designated port**: nối mạng LAN với **D**
  - o Tất cả các root port **R** và designated port **D** được thiết lập trạng thái chuyển tiếp (**forwarding**). Chỉ các port này được phép chuyển tiếp frame. Các port còn lại được thiết lập trạng thái khóa (**blocking**)



o Ví dụ: Giả sử cost cho mỗi LAN đều bằng nhau

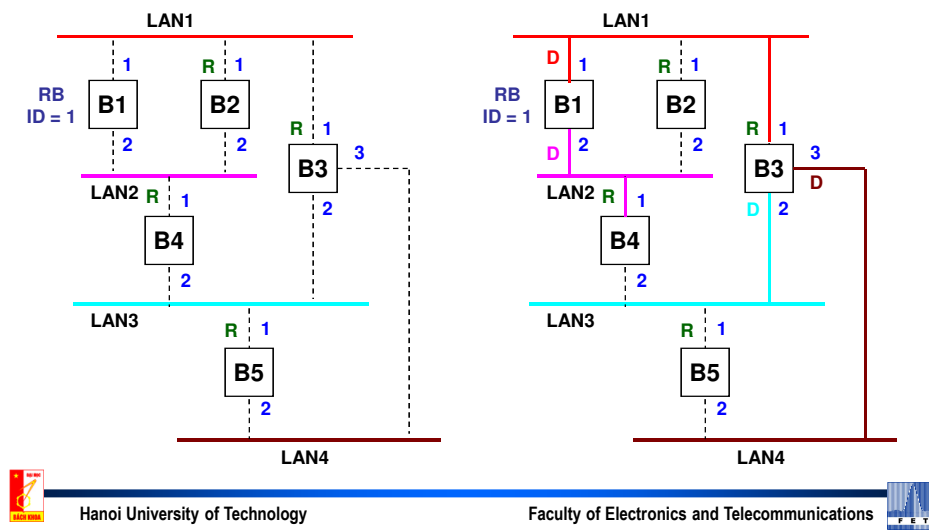
1. B1 được chọn làm root bridge



2. Root port được chọn cho từng bridge trừ RB

3. Chọn designated bridge cho mỗi LAN

4. Thiết lập root port và designated port



## ✓ Network analyzer: Ethereal

- o Do Gerald Combs [gerald@ethereal.com](mailto:gerald@ethereal.com) , [www.ethereal.com](http://www.ethereal.com)
- o Ethereal có thể bắt, hiển thị và phân tích các frame qua Ethernet NIC
- o Chuỗi các frame và nội dung của frame có thể được phân tích chi tiết ở mức byte
- o Rất hữu hiệu khi phân tích lỗi mạng, đồng thời là phần mềm dạy học
- o Chọn website [www.vnexpress.net](http://www.vnexpress.net) làm ví dụ



## ✓ Ethereal window

The screenshot displays the Ethereal network analyzer interface. The main window is titled "(Untitled) - Ethereal" and contains a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Help) and a toolbar. Below the toolbar is a filter bar. The main display area is divided into three panes:

- Top pane:** Frame/packet sequence. It shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, and Info. The selected packet is 1, a DNS Standard query A for www.vnexpress.net.
- Middle pane:** Encapsulation for a given frame. It shows the hierarchical structure of the selected packet, including Ethernet II, Internet Protocol, User Datagram Protocol, and Domain Name System (query).
- Bottom pane:** Giá trị Hexa & text tương ứng. It shows the raw data of the selected packet in hexadecimal and ASCII format.

Annotations with arrows point to these panes: "Top pane Frame/packet sequence", "Middle pane Encapsulation for a given frame", and "Bottom pane Giá trị Hexa & text tương ứng".



## ✓ Top pane: Frame sequence

**DNS query**

**TCP Connection Setup**

**HTTP request & response**

File: (Untitled) 486 KB 00:00:11 Dr P: 961 D: 961 M: 0



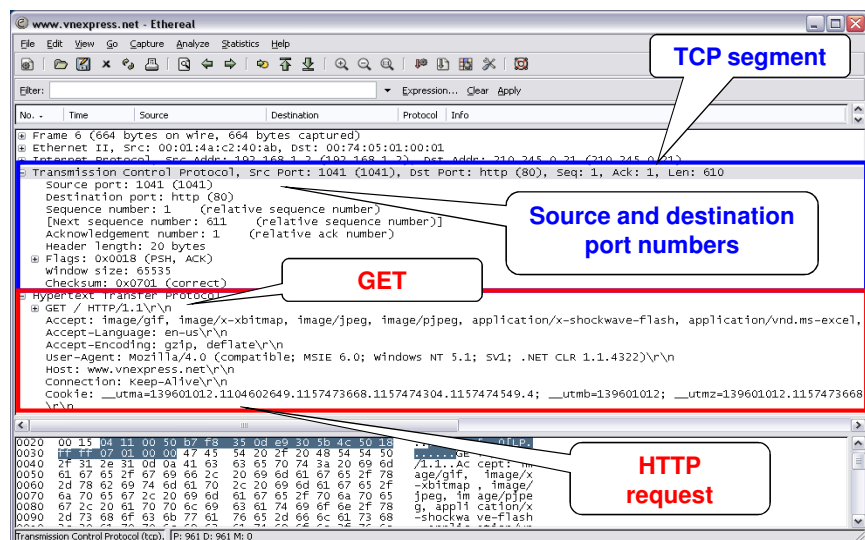
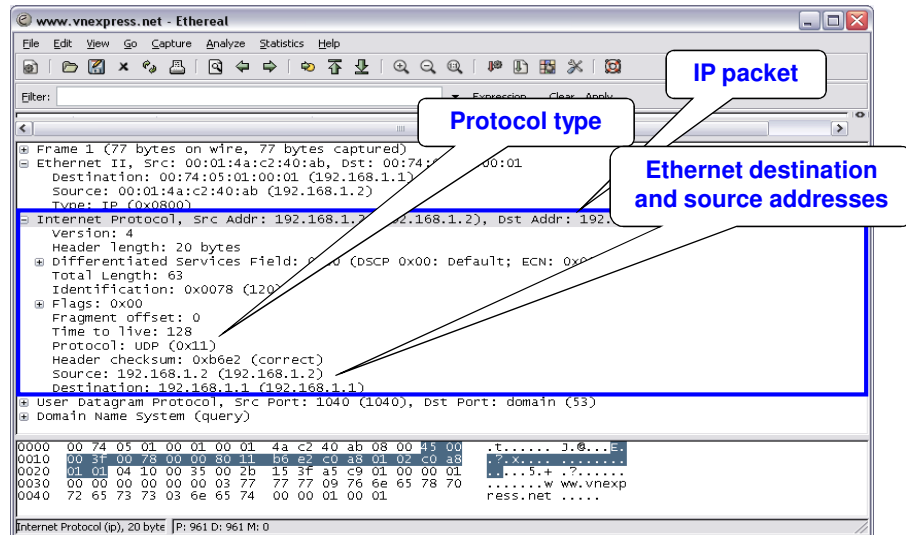
## ✓ Middle pane: Encapsulation

**Protocol type**

**Ethernet destination and source addresses**

Internet Protocol (IP), 20 bytes P: 961 D: 961 M: 0







## Tổng kết

- **Tầng liên kết dữ liệu (DLL)**
  - ✓ Framing
  - ✓ Flow control: Stop-and-wait control, sliding window
  - ✓ Error control: Stop-and-wait ARQ, Go-back-N ARQ, selective-repeat ARQ
- **Giao thức của tầng liên kết dữ liệu**
  - ✓ HDLC: HDLC frame format, operation mode
  - ✓ PPP: PPP frame format
- **Kết nối và mở rộng mạng ở tầng liên kết dữ liệu**
  - ✓ Transparent bridge: self-learning của bảng tham chiếu
  - ✓ Spanning tree algorithm: ngăn chặn vòng lặp trong kết nối mạng

