Chapter 4
Data Link Layer

A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in powerpoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach Featuring the Internet, 2nd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2002.

Chương 4: Tầng liên kết dữ liệu

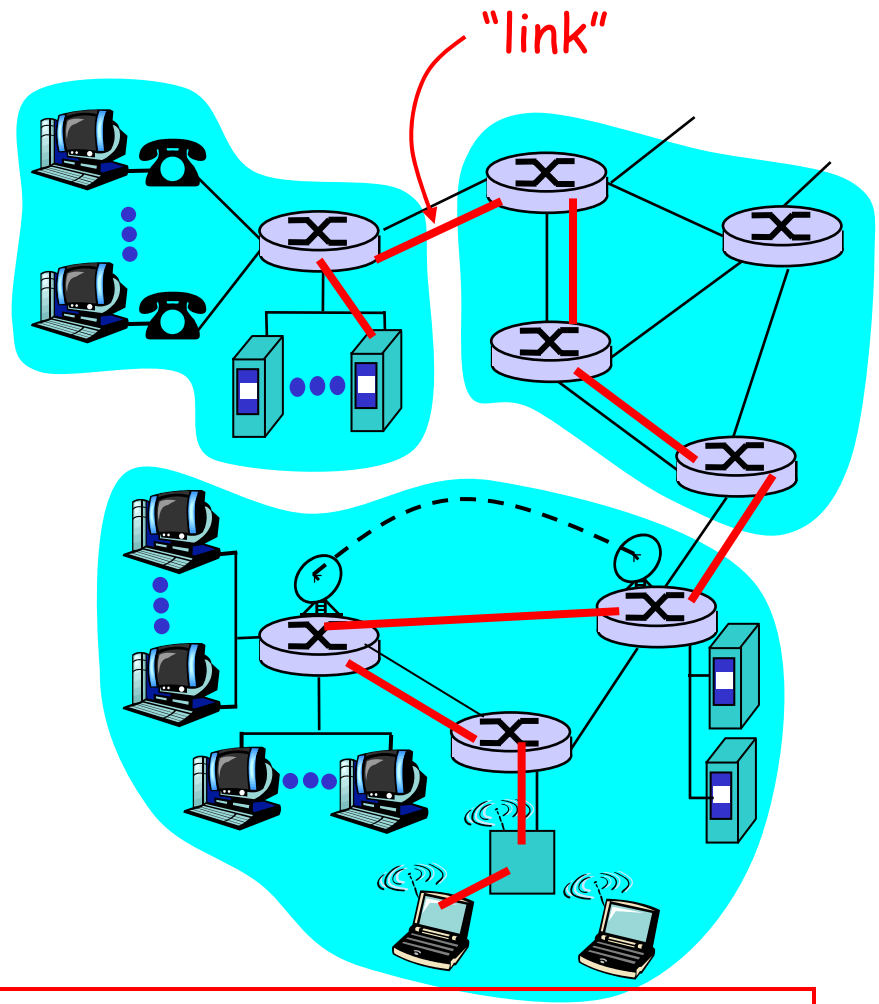
Mục tiêu:

- Nắm được các nguyên lý đằng sau các dịch vụ của tầng liên kết dữ liệu (LKDL) :
  - định khung và đồng bộ hóa
  - dò lỗi và sửa lỗi
  - cách thức chia sẻ một kênh truyền quảng bá: đa truy cập
  - đánh địa chỉ ở tầng LKDL
  - truyền dữ liệu tin cậy, kiểm soát luồng
- Tìm hiểu một số công nghệ/kỹ thuật phổ biến ở tầng LKDL

Giới thiệu

Một số thuật ngữ:

- hosts và routers được gọi là các nút (**nodes**)
  (bridges và switches cũng vậy)
- Các kênh truyền thông kết nối các nút gần kề dọc theo đường dẫn truyền thông được gọi là các liên kết (**links**)
  - các liên kết hữu tuyến
  - các liên kết vô tuyến
  - các mạng cục bộ
- đơn vị dữ liệu giao thức ở tầng 2 là **frame**, đóng gói/bao bọc gam dữ liệu



Tầng liên kết dữ liệu có trách nhiệm truyền gam dữ liệu từ một nút đến nút gần kề qua một liên kết truyền thông

Tầng LKDL: ngữ cảnh

- Gam dữ liệu được chuyên chở bởi các giao thức khác nhau ở tầng LKDL qua các liên kết truyền thông khác nhau:
  - Vd: Ethernet ở liên kết đầu tiên, frame relay ở các liên kết trung gian, 802.11 ở liên kết cuối cùng (xem hình ở slide trước)
- Mỗi giao thức ở tầng LKDL cung cấp các dịch vụ khác nhau
  - Vd: có thể cung cấp dịch vụ chuyển dữ liệu tin cậy hoặc không qua liên kết truyền thông.

Tương tự như sự vận chuyển

- Một chuyến đi từ Princeton đến Lausanne
  - limo: Princeton đến JFK
  - plane: JFK đến Geneva
  - train: Geneva đến Lausanne
- khách du lịch = datagram
- đoạn vận chuyển = liên kết truyền thông
- phương thức vận chuyển = giao thức tầng lkdl
- đại lý du lịch = giải thuật định tuyến

Các dịch vụ của tầng LKDL

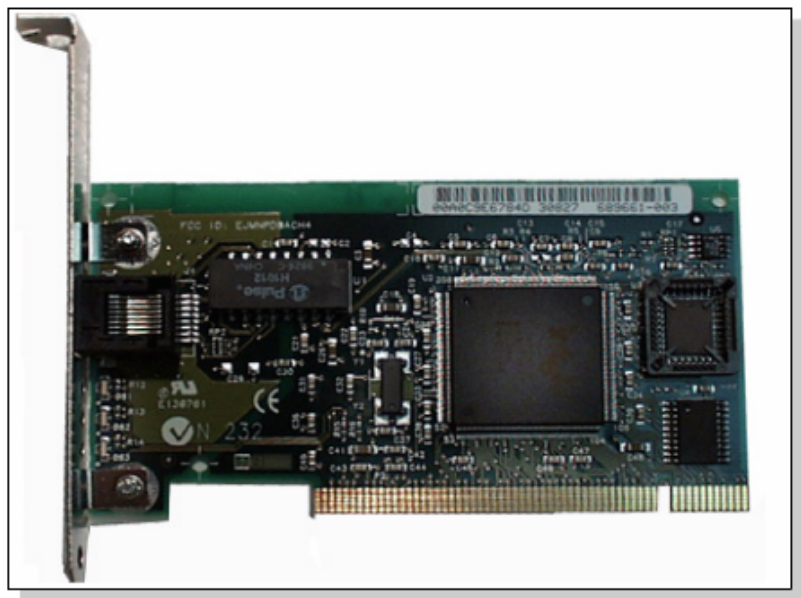
- Định khung, truy cập đường truyền:
  - đóng gói gam dữ liệu vào khung, thêm thông tin điều khiển và kiểm soát lỗi (header, trailer)
  - truy cập kênh truyền nếu phương tiện truyền là chia sẻ
  - địa chỉ vật lý được dùng trong khung để định danh nguồn và đích
    - khác với địa chỉ IP!
- Chuyển dữ liệu tin cậy giữa các nút liền kề
  - hiếm khi được thực hiện ở các liên kết có tỷ lệ lỗi bit thấp (cáp quang, một số loại cáp đôi dây xoắn)
  - được cung cấp ở các đường truyền vô tuyến vì tỷ lệ lỗi cao

Các dịch vụ của tầng LKDL (tiếp theo)

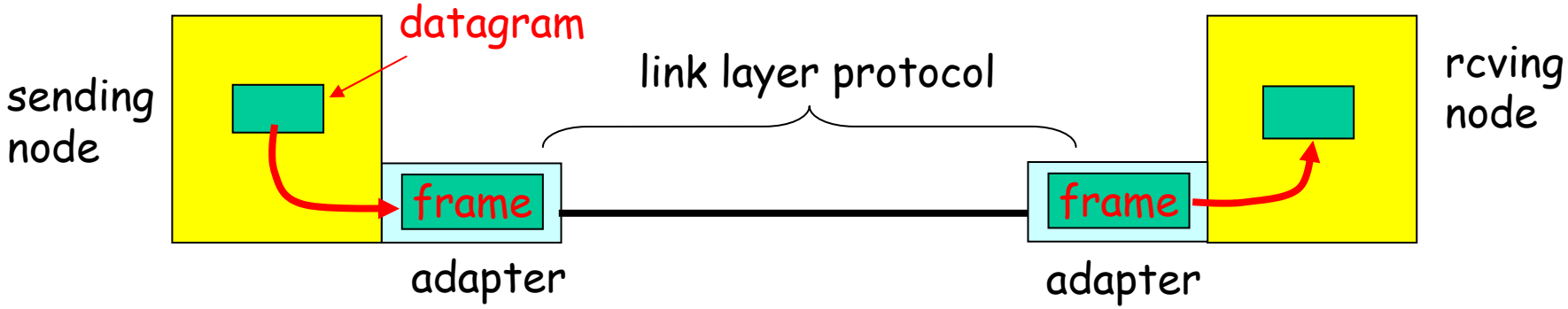
- *Kiểm soát luồng:*
  - điều chỉnh tốc độ giữa các nút gửi và nhận (gần kề)
- *Phát hiện lỗi:*
  - lỗi bị sinh ra bởi sự suy giảm tín hiệu, nhiễu
  - nơi nhận dò tìm sự xuất hiện của lỗi:
    - báo hiệu cho nơi gửi để truyền lại hoặc bỏ frame lỗi đó
- *Sửa lỗi:*
  - nơi nhận xác định và sửa các bit lỗi mà không phải viện đến việc truyền lại
- *Bán song công và song công*
  - với bán song công, các nút tại hai đầu của liên kết có thể truyền dữ liệu nhưng không cùng thời điểm

Kết nối Host – Mạch giao tiếp mạng

- NIC hay card mạng là thiết bị tầng 2, mỗi NIC có một mã duy nhất gọi là địa chỉ MAC.
- Khi lựa chọn một card mạng cần cân nhắc các yếu tố:
  - Kiến trúc mạng mà NIC đó hỗ trợ
  - Hệ điều hành
  - Loại phương tiện truyền
  - Tốc độ truyền dữ liệu
  - Loại bus sẵn có



Truyền thông giữa các bộ thích ứng mạng



- tầng lkdl được thực thi trên bộ thích ứng mạng
  - Ethernet card, PCMCIA card, 802.11 card
- bên gởi:
  - đóng gói gam dữ liệu vào khung
  - thêm thông tin điều khiển và kiểm soát lỗi
- bên nhận
  - tìm các lỗi và thông tin điều khiển
  - trích gam dữ liệu và chuyển lên cho nút nhận
- NICs là thiết bị bán tự trị
- thực hiện các chức năng của tầng vật lý và LKDL

Định khung và đồng bộ hóa

- Vấn đề: dồn dòng bit vào các khung
- Phải xác định các bit đầu tiên và cuối cùng của khung
  - Định khung và đồng bộ hóa có quan hệ chặt chẽ với nhau
- Thường được thực thi bởi card mạng
- Bộ thích ứng mạng lấy/đặt các khung ra từ/vào bộ nhớ host/switch

Các phương pháp định khung

- Dựa trên đồng hồ
  - Một mẫu bit đặc biệt xuất hiện định kỳ để báo hiệu bắt đầu một khung
- Hướng ký tự/byte
  - Đếm số ký tự/byte
    - Vấn đề: khi trường chứa số ký tự/byte của khung bị sai lệch do lỗi truyền thì bên nhận mất đồng bộ.
  - Dùng các ký tự bắt đầu và kết thúc
    - STX (start of text) và ETX (end of text)
    - Vấn đề: khi dữ liệu có chứa những ký tự bắt đầu hay kết thúc?
    - Nhồi ký tự
      - Nhồi thêm vào trước các ký tự đặc biệt một ký tự "thoát" DLE
      - Nếu dữ liệu chứa ký tự "thoát" thì sao?

Các phương pháp định khung (tiếp theo)

- Phương pháp hướng bit
  - Mỗi frame bắt đầu và kết thúc với một chuỗi bit đặc biệt
    - Flag hay preamble 01111110
  - Nhồi bit
    - Bên gởi: khi nào có 5 bits 1 liên tiếp nhau trong phần dữ liệu thì nhồi thêm một bit 0
    - Bên nhận: khi 5 bits 1 liên tiếp đến thì
      - nếu bit tiếp theo là 0 thì bỏ đi bit đó
      - nếu các bit tiếp theo là 10: dấu hiệu kết thúc frame
      - nếu các bit tiếp theo là 11: lỗi

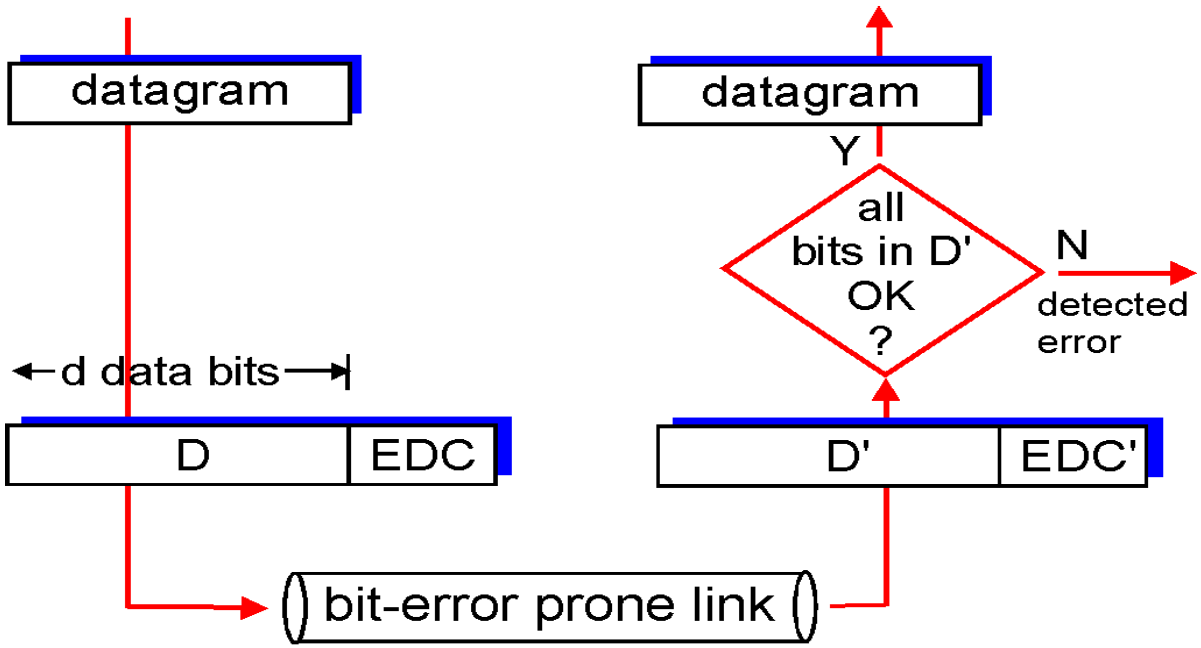
Xử lý lỗi

- Dữ liệu có thể bị sai lệch trong quá trình truyền
  - giá trị bit thay đổi
- Đưa thêm vào khung thông tin để kiểm soát lỗi
  - đặt vào bởi bên gởi
  - được kiểm tra bởi bên nhận
- Dò lỗi so với sửa lỗi
  - Cả hai đều cần thông tin "thừa"
  - Dò: lỗi có xuất hiện hay không
  - Sửa: sửa lỗi nếu xuất hiện lỗi
- Chỉ là sự đảm bảo mang tính thống kê

Sự phát hiện lỗi

EDC= các bít "dư thừa" để dò và sửa lỗi
D = dữ liệu được bảo vệ bằng phương pháp kiểm tra lỗi, có thể bao gồm các trường điều khiển

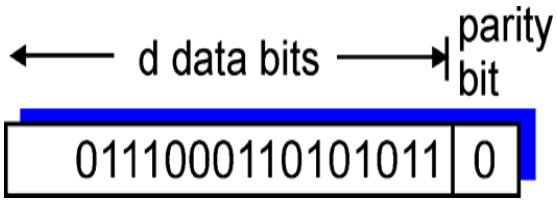
- Phát hiện lỗi không tin cậy 100%!
  - các cơ chế dùng để kiểm soát lỗi có thể bị sót một số lỗi (hiếm);
  - trường EDC càng lớn thì có thể dò và sửa lỗi tốt hơn.



Kiểm tra tính chẵn lẻ (Parity Checking)

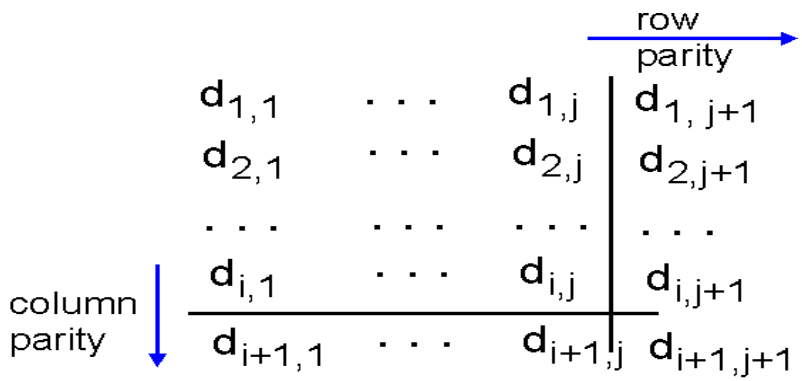
Single Bit Parity:

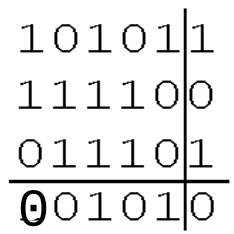
Detect single bit errors



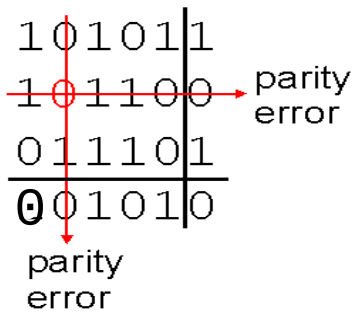
Two Dimensional Bit Parity:

Detect *and correct* single bit errors





no errors



correctable single bit error

Các kỹ thuật phát hiện lỗi khác

- Kiểm tra tổng (Checksum)
  - Xét dữ liệu như là dãy các số nguyên (integers)
  - Tính và gửi số kiểm tra tổng
  - Xử lý được nhiều bit lỗi
  - Không thể xử lý được tất cả các lỗi
- Kiểm dư vòng (Cyclic Redundancy Check)
  - Dùng các hàm toán học để xét dữ liệu
  - Tính toán phức tạp hơn rất nhiều
  - Có thể xử lý được nhiều lỗi hơn

Internet checksum (RFC 1071)

Mục tiêu: phát hiện "các lỗi" (vd: các bit bị lật) trong các segment được truyền (lưu ý: chỉ được dùng ở tầng vận chuyển)

Bên gửi:

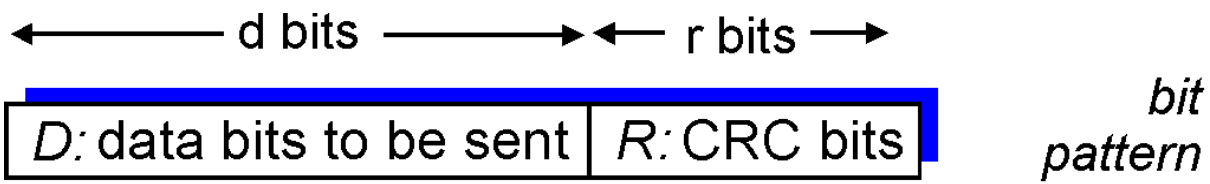
- xem nội dung các segment như là dãy các số nguyên 16 bit
- checksum: thêm vào tổng phần bù 1 của nội dung segment
- Bên gửi đưa giá trị checksum và trường UDP checksum

Bên nhận:

- tính checksum của segment vừa nhận được
- kiểm tra xem checksum vừa được tính có trùng với giá trị trong trường checksum hay không:
  - NO – lỗi được phát hiện
  - YES – không có lỗi bị phát hiện. Tuy nhiên, vẫn có thể tồn tại lỗi?!

Kiểm dư vòng – Cyclic Redundancy Check

- xem các bit dữ liệu, D, như là các số nhị phân
- chọn đa thức sinh, G , mẫu r+1 bit
- mục tiêu: chọn r CRC bits, R, sao cho
  - <D,R> có thể được chia hết hoàn toàn bởi G (theo modulo 2)
  - bên nhận biết G, chia <D,R> bởi G. Nếu phần dư khác 0: lỗi bị phát hiện!
  - có thể phát hiện tất cả các lỗi ít hơn r+1 bits
- được sử dụng rộng rãi trong thực tế (Ethernet, HDLC)



bit pattern

$D * 2^r$ XOR R

mathematical formula

Ví dụ về CRC

Ta muốn:

$D \cdot 2^r$ XOR R = nG

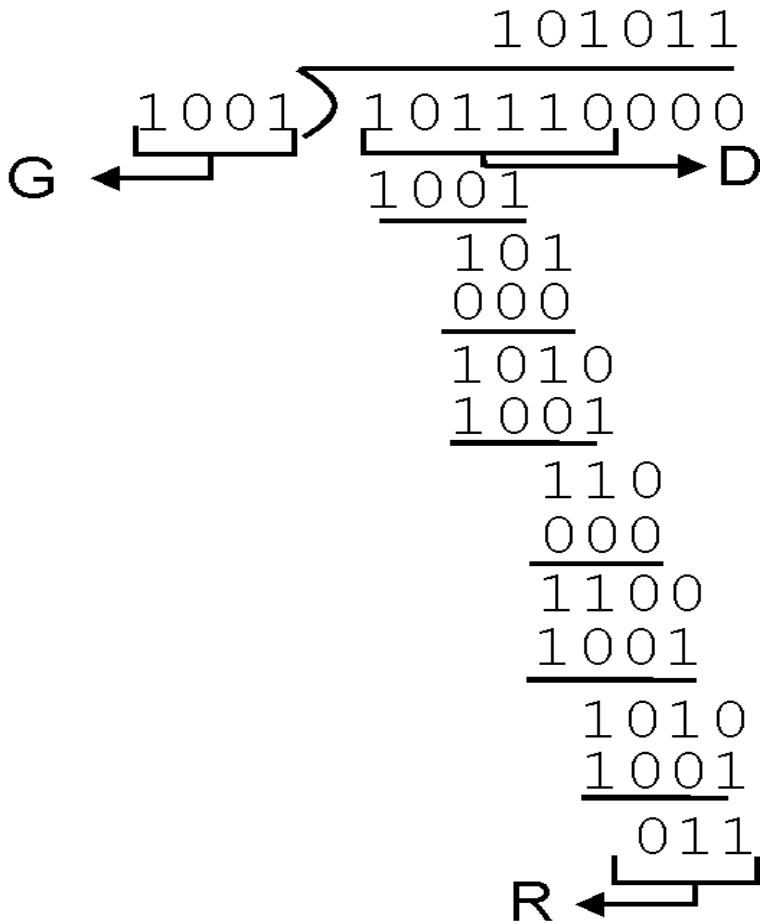
tương đương với:

$D \cdot 2^r$ = nG XOR R

Tương đương với:

nếu ta chia $D \cdot 2^r$ cho G, ta có phần dư R

$$R = \text{phần dư } \left[\frac{D \cdot 2^r}{G}\right]$$



Gởi đi: 101110011

- Ethernet và các mạng token ring sử dụng CRC-32
  - $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$

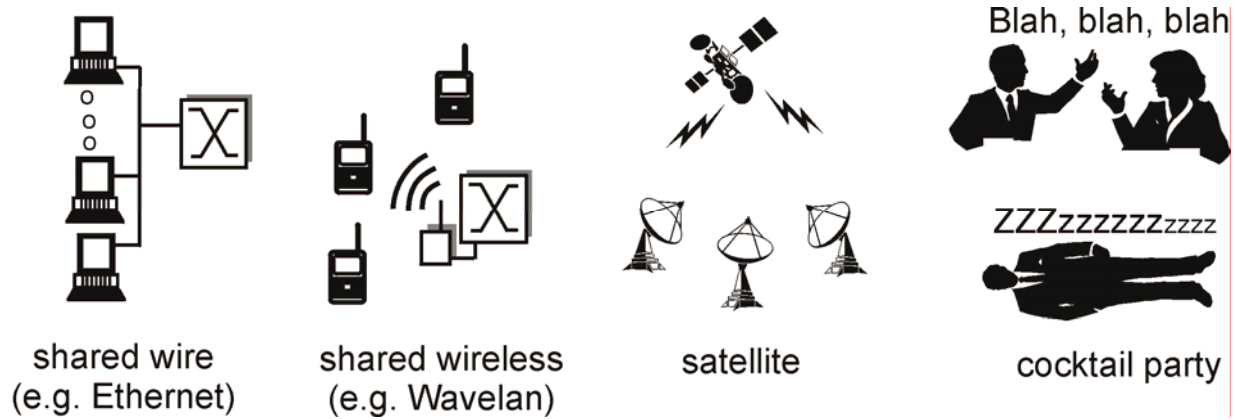
Tóm tắt về sự phát hiện lỗi

- Để phát hiện lỗi có thể xảy ra trong quá trình truyền:
  - Bên gởi thêm vào gói tin một số thông tin
  - Bên nhận dựa vào các thông tin trên để kiểm tra
- Các kỹ thuật dò tìm lỗi phổ biến:
  - Kiểm tra tính chẵn lẻ (Parity bit checking)
  - Kiểm tra tổng (Checksum)
  - Kiểm dư vòng (Cyclic Redundancy Check)
- Chỉ đảm bảo phát hiện được lỗi ở mức thống kê nào đó mà thôi!

Giao thức và liên kết đa truy cập

Hai loại "liên kết" :

- điểm – điểm
  - liên kết truy cập điểm-điểm qua quay số
  - liên kết điểm giữa Ethernet switch và host
- quảng bá (dây dẫn hay phương tiện truyền được chia sẻ)
  - Ethernet truyền thống
  - upstream HFC
  - 802.11 wireless LAN



Giao thức đa truy cập

- kênh truyền quảng bá đơn được chia sẻ
- khi có hai hay nhiều trạm truyền đồng thời thì xảy ra xung đột
  - chỉ một nút truyền thành công tại một thời điểm

Giao thức đa truy cập

- giải thuật phân tán giúp các nút chia sẻ kênh truyền vd: xác định khi nào một nút có thể truyền
- truyền thông về việc điều khiển chia sẻ kênh truyền có thể dùng một kênh riêng
- những gì các giao thức đa truy cập hướng tới (slide tiếp theo) :

Giao thức đa truy cập lý tưởng

Kênh truyền quảng bá tốc độ R bps

1. Khi chỉ một nút muốn truyền, nó có thể truyền với tốc độ R.
2. Khi có M nút muốn truyền, mỗi nút có thể truyền với tốc độ trung bình là R/M
3. Giao thức phải là hoàn toàn phân tán:
   - không một nút đặc biệt nào sắp xếp việc truyền tin
   - không cần đến sự đồng bộ hóa đồng hồ, khe (thời gian)
4. Đơn giản

Một sự phân loại việc kiểm soát truy cập phương tiện truyền

Có 3 loại chính:

- Phân chia kênh truyền
  - chia kênh truyền thành những "mảnh" nhỏ (theo khe thời gian, tần số, mã)
  - cấp phát các mảnh đó cho các nút và chúng được "độc quyền" sử dụng trong khoảng được chia
- Truy cập ngẫu nhiên
  - không chia kênh truyền, chấp nhận xung đột
  - vấn đề chính là "phục hồi" việc truyền khi có xung đột
- "Luân phiên"
  - điều phối chặt chẽ việc truy cập phương tiện truyền để tránh xung đột

Các giao thức truy cập ngẫu nhiên

- khi một nút có gói tin để gởi
  - truyền với tốc độ tối đa của kênh (R).
  - không có sự phối hợp từ trước giữa các nút
- khi có hai hay nhiều nút truyền đồng thời -> "xung đột"
- Giao thức MAC truy cập ngẫu nhiên định rõ:
  - làm thế nào để dò ra xung đột
  - làm thế nào để phục hồi khi có xung đột xảy ra
- Một số giao thức MAC truy cập ngẫu nhiên tiêu biểu:
  - ALOHA
  - slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

CSMA – Đa truy cập cảm nhận sóng mang

CSMA: "nghe trước khi nói"

- Nếu kênh truyền rỗi: truyền toàn bộ frame
- Nếu kênh truyền bận, trì hoãn việc truyền

- Tương tự như con người: không ngắt lời người khác!

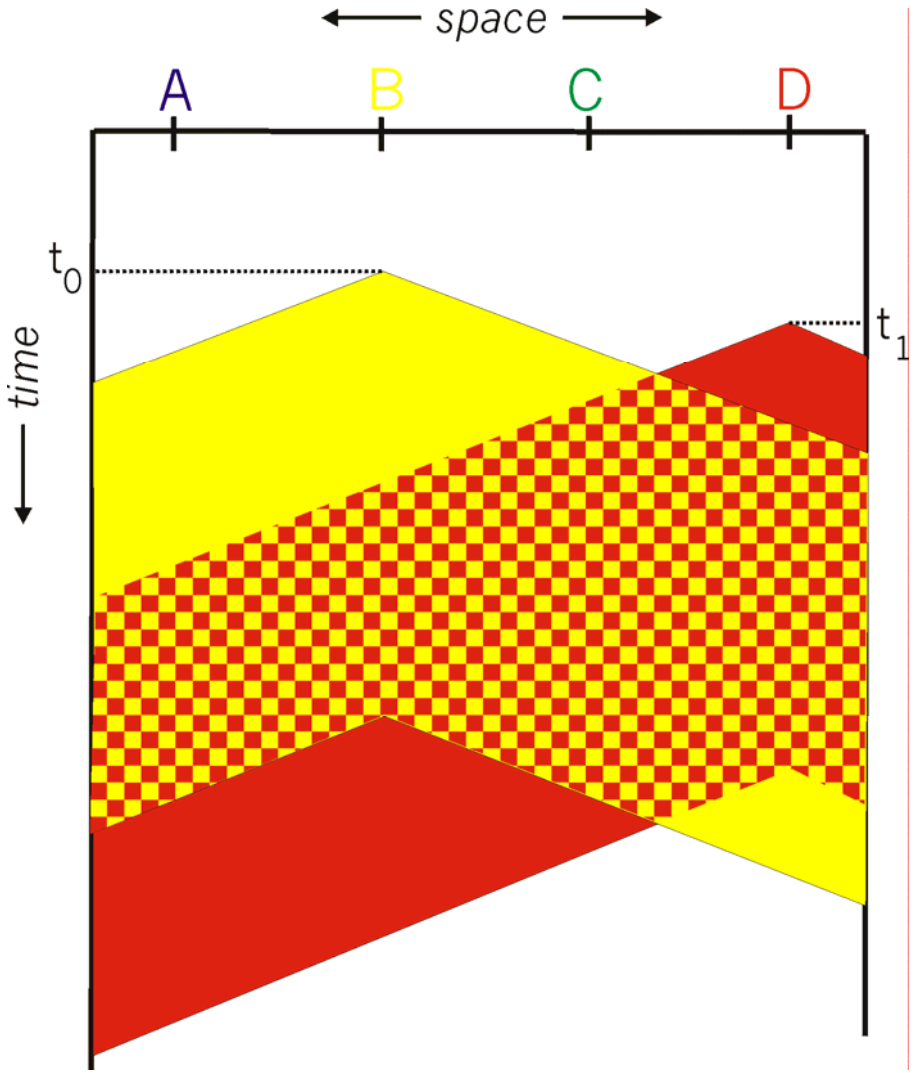
Xung đột trong CSMA

xung đột vẫn có thể xảy ra: hai nút không nghe được việc truyền của nhau do độ trễ truyền tin.

khi có xung đột: toàn bộ thời gian truyền gói tin là lãng phí do nó bị hỏng

lưu ý:
Vai trò của khoảng cách và độ trễ truyền tin là rất quan trọng trong việc xác định xác suất xung đột

Bố trí về mặt không gian của các nút



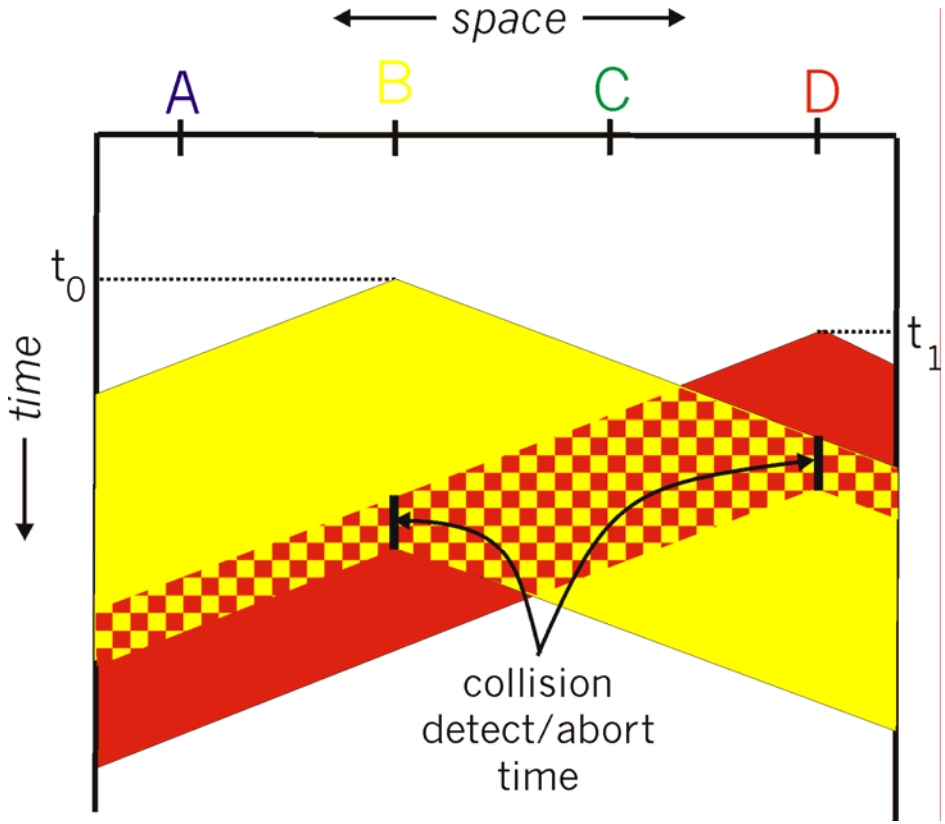
Đa truy cập cảm nhận sóng mang có dò xung đột (CSMA/CD)

CSMA/CD: "nghe trong khi nói"

- xung đột được phát hiện trong khoảng thời gian ngắn
- những truyền thông gây xung đột bị hủy bỏ sớm, giảm sự lãng phí kênh truyền

- phát hiện xung đột:
  - dễ thực hiện trong wired LANs: đo lường cường độ tín hiệu, so sánh tín hiệu truyền và tín hiệu nhận được
  - khó thực hiện trong wireless LANs
- tương tự như con người: người có tài nói chuyện (lịch sự)

Minh họa phát hiện xung đột CSMA/CD



Các giao thức MAC "luân phiên"

Các giao thức MAC phân chia kênh truyền:

- chia sẻ kênh truyền hiệu quả và công bằng khi tải cao
- không hiệu quả khi tải thấp: bị trễ khi truy cập kênh truyền, chỉ có 1/N dải thông được cấp nếu chỉ có một nút hoạt động!

Các giao thức MAC truy cập ngẫu nhiên

- hiệu quả khi tải thấp: một nút có thể tận dụng toàn bộ kênh truyền
- tải cao: gánh nặng do xung đột

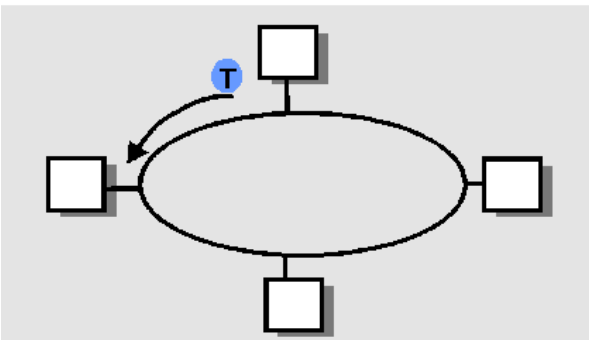
Các giao thức "luân phiên"

- tìm kiếm cơ chế tốt nhất từ hai loại giao thức trên

Các giao thức MAC "luân phiên" (tiếp theo)

Kiểm soát vòng:

- nút chủ "mời" nút tớ truyền dữ liệu theo lượt
- các vấn đề cần quan tâm:
  - gánh nặng kiểm soát vòng
  - độ trễ chờ đợi đến lượt
  - sự hư hỏng của nút chủ sẽ làm gãy vòng



Chuyển thẻ bài:

- thẻ bài điều khiển được chuyển tuần tự từ nút này sang nút khác.
- trạm nào có thẻ bài sẽ được quyền truyền thông điệp
- các vấn đề cần quan tâm:
  - gánh nặng quản lý thẻ bài
  - độ trễ chờ đợi thẻ bài
  - thẻ bài bị mất sẽ làm gãy quá trình truyền

Tóm tắt về các giao thức điều khiển truy cập phương tiện truyền chia sẻ

- Các phương pháp chính để điều khiển việc truy cập phương tiện truyền chia sẻ
  - Phân chia kênh theo thời gian, tần số, mã
    - Time Division, Frequency Division, Code Division
  - Truy cập ngẫu nhiên,
    - ALOHA, S-ALOHA, CSMA, CSMA/CD
    - Cảm nhận sóng mang: dễ thực hiện trong một số công nghệ (wire) nhưng khó ở các công nghệ khác (wireless)
    - CSMA/CD được sử dụng trong Ethernet
  - Luân phiên
    - Kiểm soát vòng bằng một trạm trung tâm, chuyển thẻ bài

Điều khiển liên kết dữ liệu điểm – điểm

- Một người gửi, một người nhận, một kết nối -> dễ điều khiển truy cập hơn các liên kết quảng bá:
  - không kiểm soát truy cập phương tiện truyền
  - không cần sự hiện diện của địa chỉ MAC
  - vd: kết nối quay số, đường truyền ISDN
- Các giao thức kiểm soát LKDL điểm-điểm phổ biến:
  - Giao thức PPP (point-to-point protocol)
  - HDLC: Điều khiển LKDL tầng cao (LKDL từng được xem là "tầng cao" trong chồng giao thức!

Các yêu cầu thiết kế của PPP [RFC 1557]

- định khung gói tin: đóng gói gam dữ liệu của tầng mạng trong khung của tầng LKDL
  - có thể mang *đồng thời* dữ liệu từ tầng mạng của bất kỳ giao thức tầng mạng nào (không chỉ IP)
  - có khả năng tách ngược trở lại ở phía bên nhận
- tính trong suốt của bit: có thể mang bất cứ mẫu bit nào trong trường dữ liệu
- phát hiện lỗi (không sửa lỗi)
- sự "sống" của kết nối: phát hiện, báo hiệu về kết nối hỏng cho tầng mạng
- đàm phán về địa chỉ tầng mạng: các điểm cuối có thể học/cấu hình địa chỉ mạng của nhau

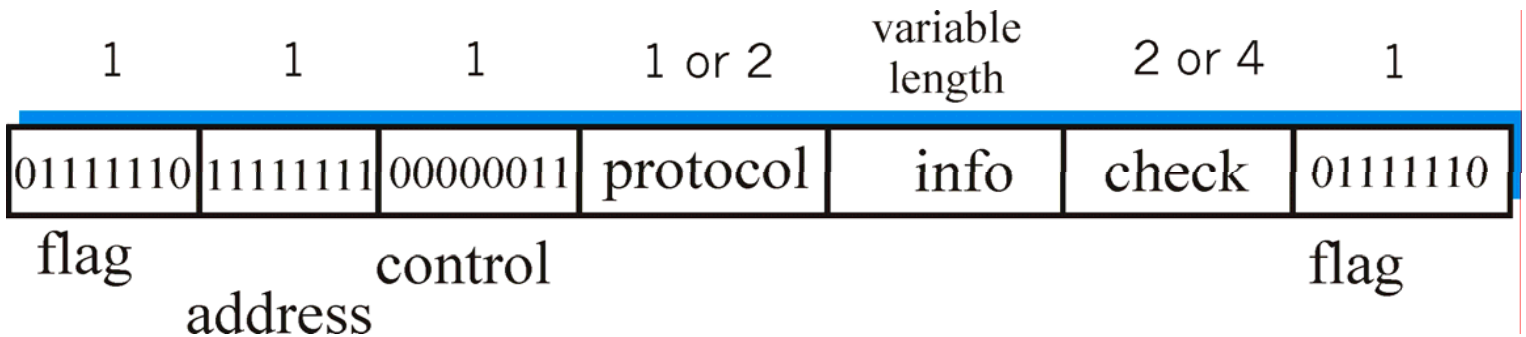
Những điều không yêu cầu đối với PPP

- Không phục hồi/sửa lỗi
- Không kiểm soát luồng dữ liệu
- Phân phát dữ liệu sai thứ tự vẫn okie
- Không cần hỗ trợ các kết nối multipoint

Phục hồi lỗi, kiểm soát luồng, tái sắp đặt dữ liệu đều được "đá" lên các tầng cao hơn!

Cấu trúc khung PPP

- Flag: cờ để phân cách giữa các khung
- Address: không làm gì (chỉ là một tùy chọn)
- Control: không làm gì; có thể là các trường điều khiển trong tương lai
- Protocol: chỉ giao thức ở tầng trên (mạng) mà khung sẽ được phân phát (vd: PPP-LCP, IP, IPCP, …)
- info: dữ liệu được mang của tầng trên
- check: kiểm dư vòng để phát hiện lỗi



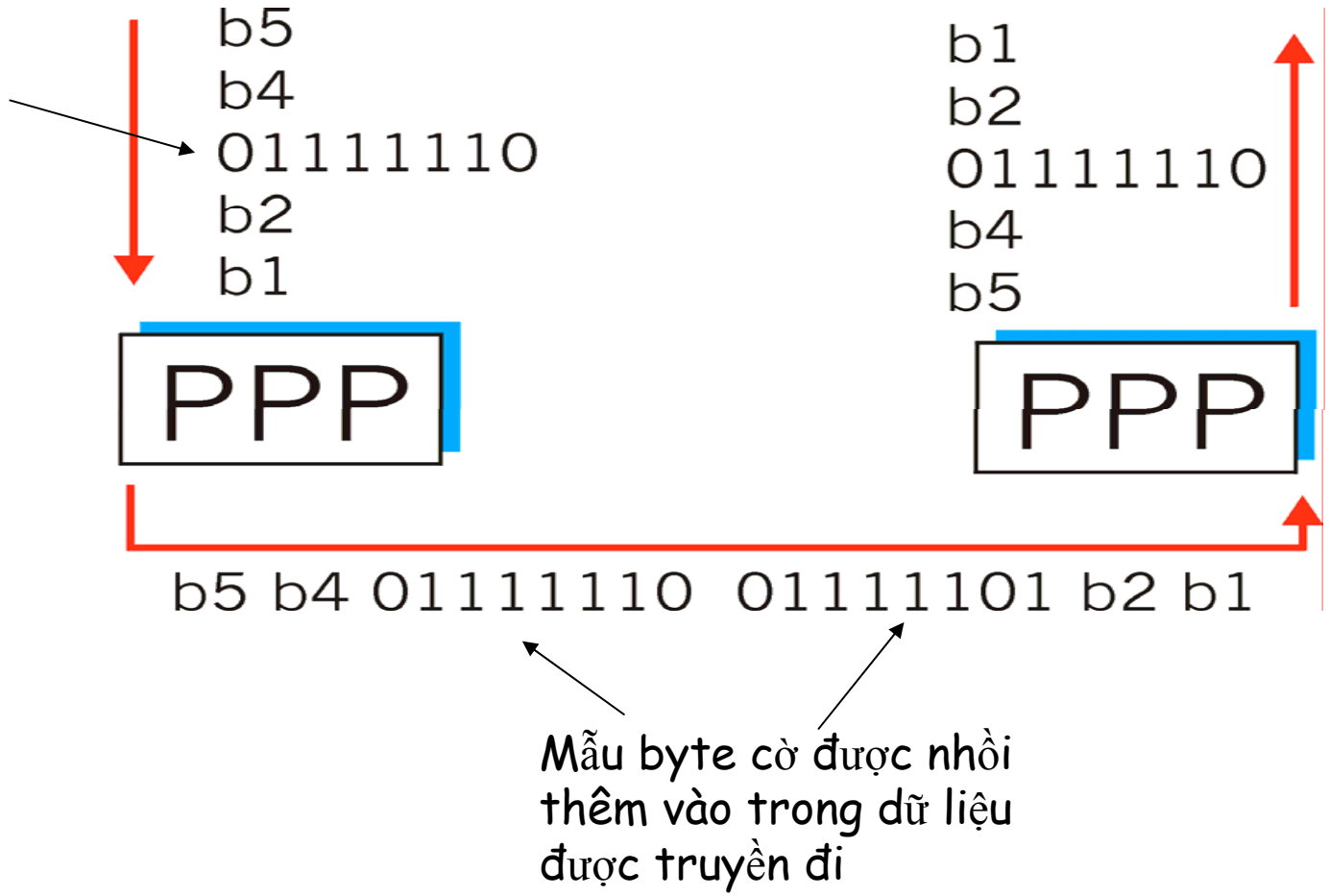
Nhồi Byte

- Yêu cầu của "sự trong suốt dữ liệu" : trong trường dữ liệu có thể chứa mẫu bit cờ <01111110>
  - Câu hỏi: khi nhận được <01111110> thì đó là dữ liệu hay cờ?

- Bên gửi: "nhồi" thêm một byte <01111110> vào sau mỗi byte <01111110> *dữ liệu*
- Bên nhận:
  - nếu nhận được 2 bytes 01111110 liên tục thì bỏ đi byte đầu tiên và tiếp tục thu nhận dữ liệu
  - nếu nhận được một byte 01111110 đơn thì đó là cờ

Minh họa nhồi byte trong PPP

Mẫu byte cờ trong dữ liệu để gửi



Mẫu byte cờ được nhồi thêm vào trong dữ liệu được truyền đi

Sự hoạt động của giao thức PPP

Trước khi trao đổi dữ liệu tầng mạng, các thực thể LKDL ngang hàng phải thực hiện

- cầu hình cho kết nối PPP (độ dài khung tối đa, xác thực)
- học/cấu hình thông tin tầng mạng
  - đối với IP: mang các thông điệp IP Control Protocol (IPCP) (có giá trị trường protocol là 8021) để cấu hình/học địa chỉ IP