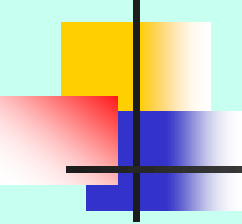


BÀI GIẢNG MÔN HỌC LẬP TRÌNH MẠNG





Chương 3:

Các giao thức cơ bản

- Giới thiệu về giao thức
- IP (Internet Protocol)
- ICMP (Internet Control Message Protocol)
- IGMP (Internet Group Management Protocol)
- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)

Chương 3:

Các giao thức cơ bản

3.1. Giới thiệu về giao thức :

3.1.1. Giới thiệu :

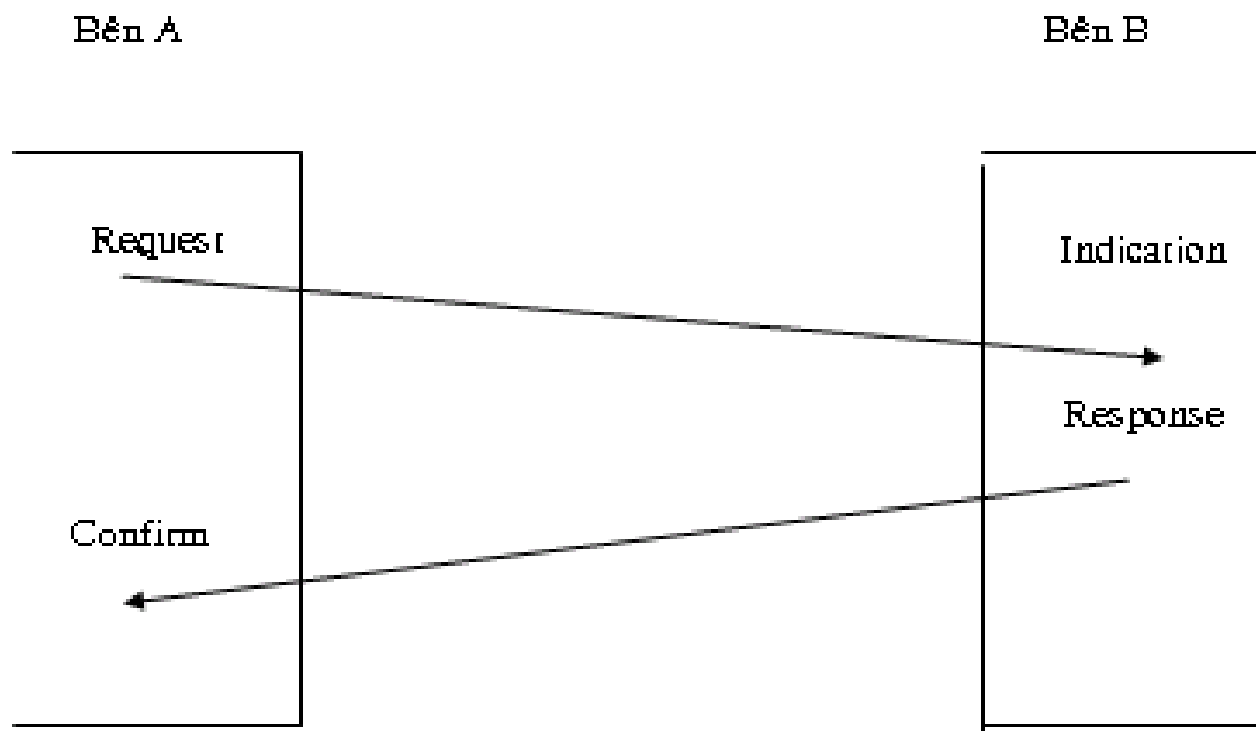
Protocol là một bộ các qui ước ràng buộc về trao đổi thông tin. Khi cài đặt có thể là một driver hoặc một đoạn mã trong ROM, trong chương trình ứng dụng, ...

Protocol có 4 tác vụ cơ bản :

- + **Request** : yêu cầu thực hiện một thao tác
- + **Indication** : Thông báo đã nhận được một sự kiện đang chờ xử lý (thông thường đó là một request của một layer khác)
- + **Response** : Trả lời chấp nhận hoặc không chấp nhận đối với yêu cầu sự kiện đang chờ xử lý.
- + **Confirm** : Báo cáo rằng layer khác đã phúc đáp yêu cầu

Chương 3: Các giao thức cơ bản

3.1.1 Giới thiệu :



Các tác vụ cơ bản của giao thức

Chương 3:

Các giao thức cơ bản



3.1.2 Các kiểu liên lạc của Protocol :

Gồm có liên lạc không kết nối và liên lạc hướng kết nối

+ **Đối với các giao thức không kết nối** : thì chỉ cần chuyển các gói dữ liệu đến layer mà nó quan hệ trực tiếp mà không cần biết gói sẽ đi đường nào để tới nơi nhận

Ví dụ : gửi thư thông qua bưu điện

+ **Đối với các giao thức có kết nối** : việc trao đổi dữ liệu tiến hành cẩn thận hơn. Đầu tiên là gửi yêu cầu kết nối đến bên nhận, kế tiếp là thủ tục hand shaking và sau đó là quá trình trao đổi thông tin. Cuối cùng là thủ tục kết thúc kết nối.

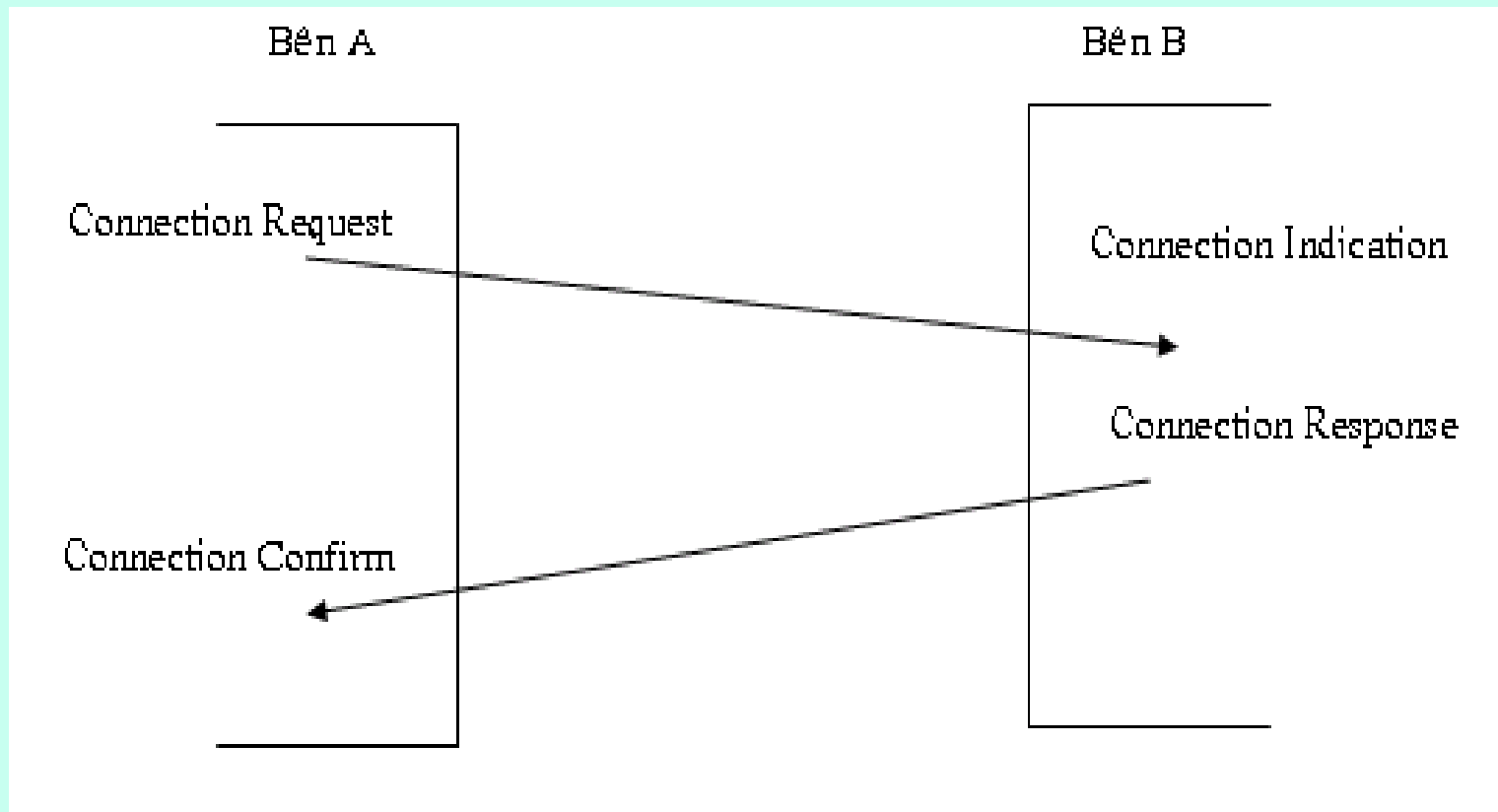
Ví dụ : Tiến trình này tương tự như gọi điện thoại để trao đổi thông tin với một người ở xa.

Chương 3:

Các giao thức cơ bản

3.1.2 Các kiểu liên lạc của Protocol :

Bước 1 : Kết nối và bắt tay

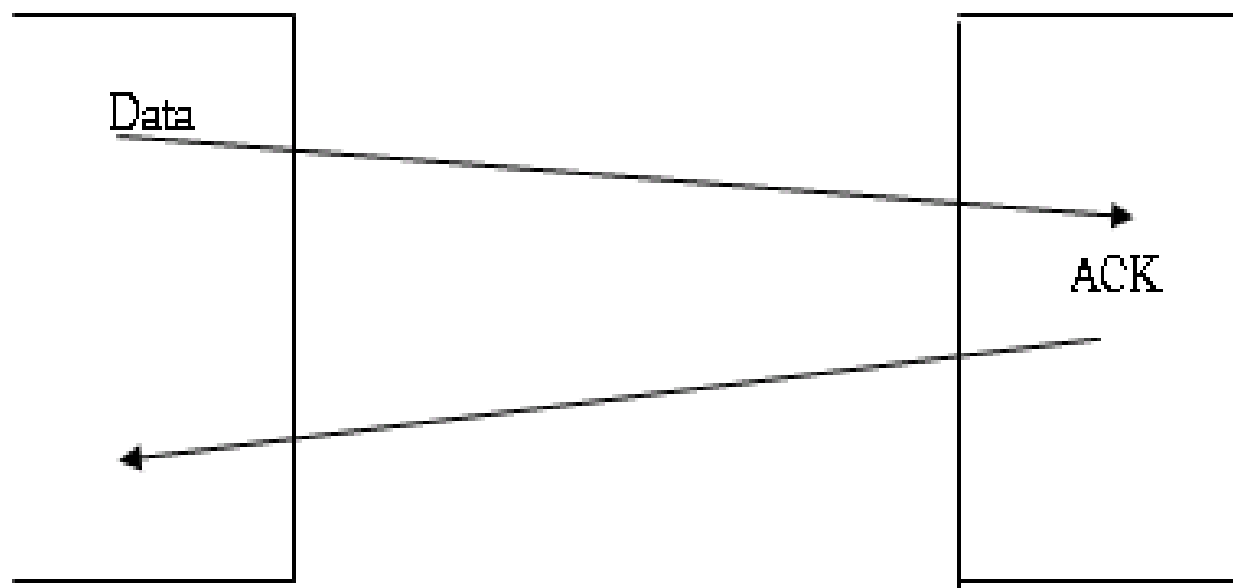


Chương 3:

Các giao thức cơ bản

3.1.2 Các kiểu liên lạc của Protocol :

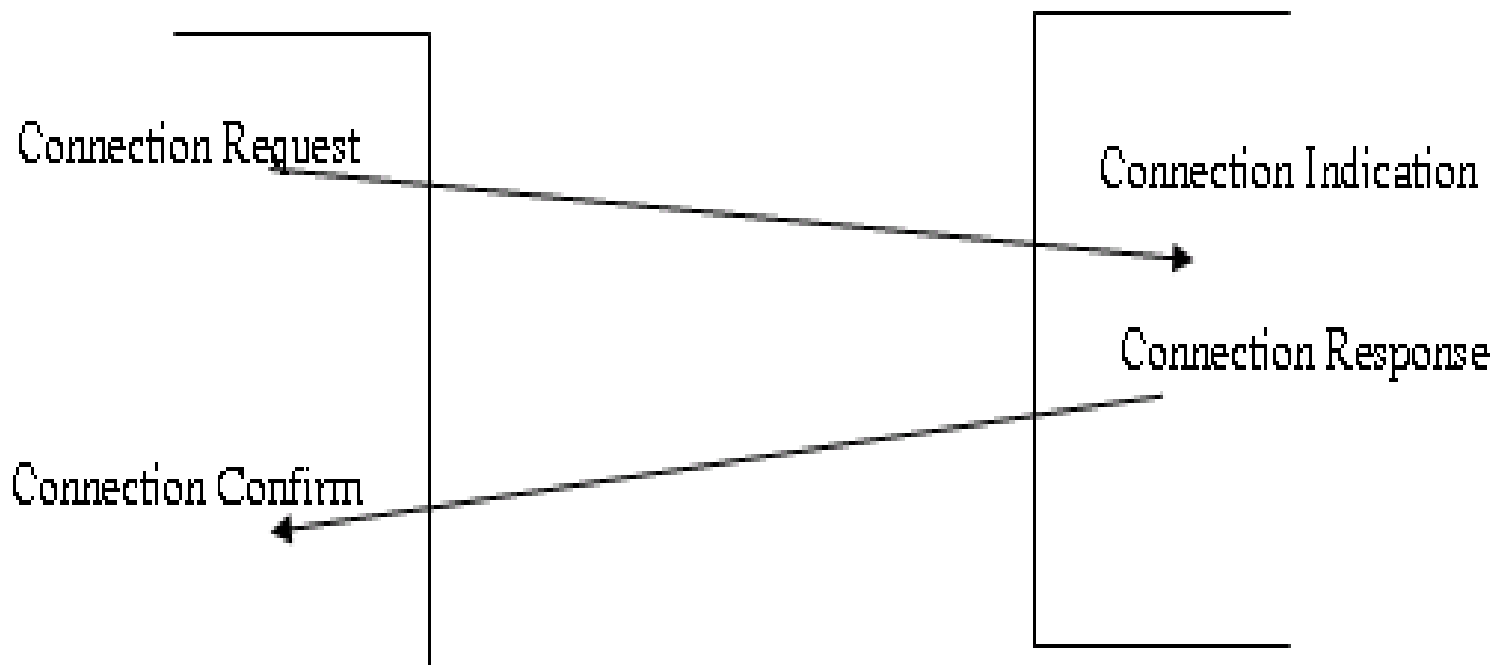
Bước 2 : Truyền dữ liệu



Chương 3: Các giao thức cơ bản

3.1.2 Các kiểu liên lạc của Protocol :

Bước 3 : Chấm dứt kết nối



Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

Tổng hợp lại ta dùng phương pháp yêu cầu phát lại ARQ (Automatic Repeat reQuest). Sau đây chúng ta khảo sát hai phương pháp kiểm soát lỗi quen thuộc là Idle-RQ và Continuous RQ.

a. Phương pháp kiểm lỗi Idle-RQ :

Idle-RQ còn có hai tên gọi khác là Stop and Wait và Send and Wait, làm việc với kiểu kết nối đường truyền half-duplex.

Bên phát gọi là primary và bên nhận secondary. Stop and Wait có hai kiểu hoạt động gọi là kiểu ẩn và kiểu hiện được đề cập sau đây:

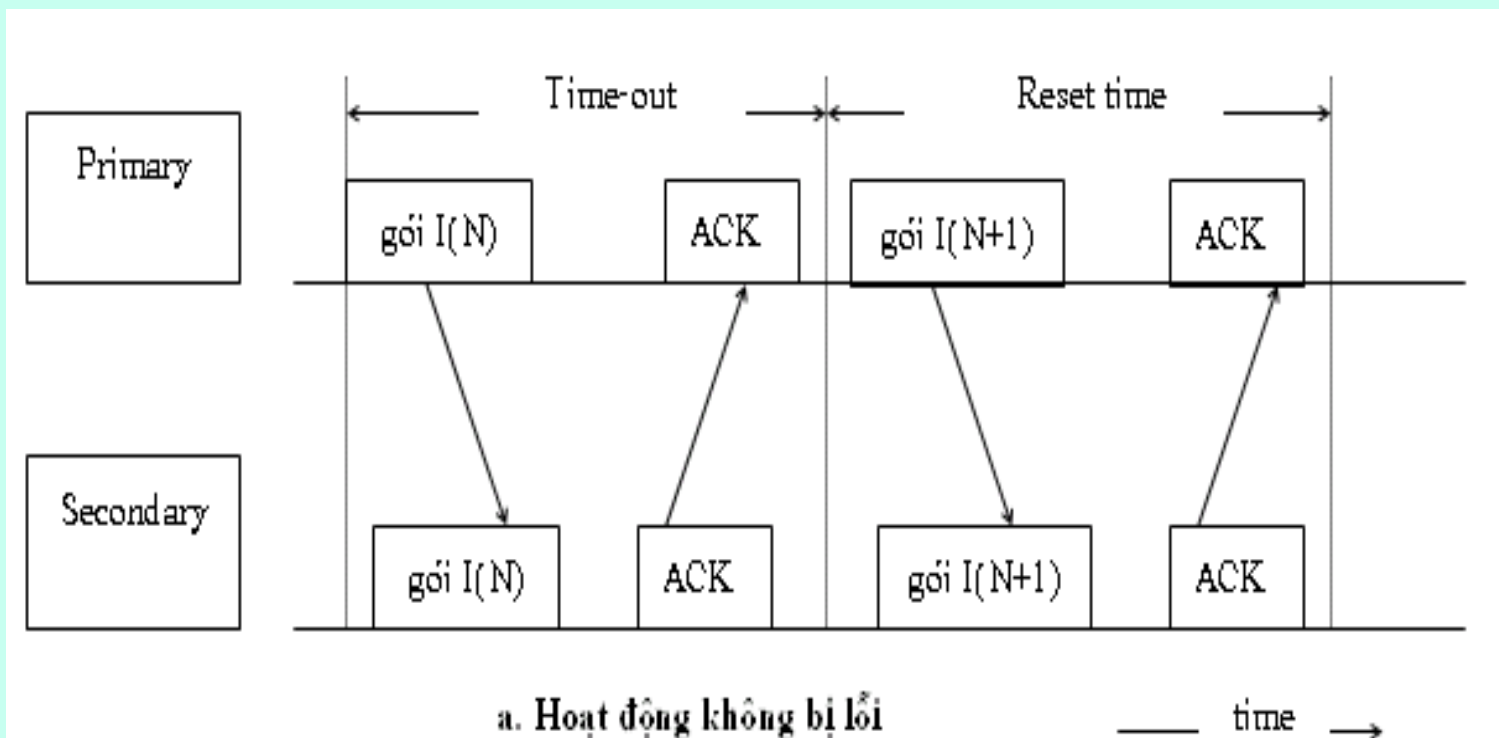
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

* Hoạt động kiểu ẩn :



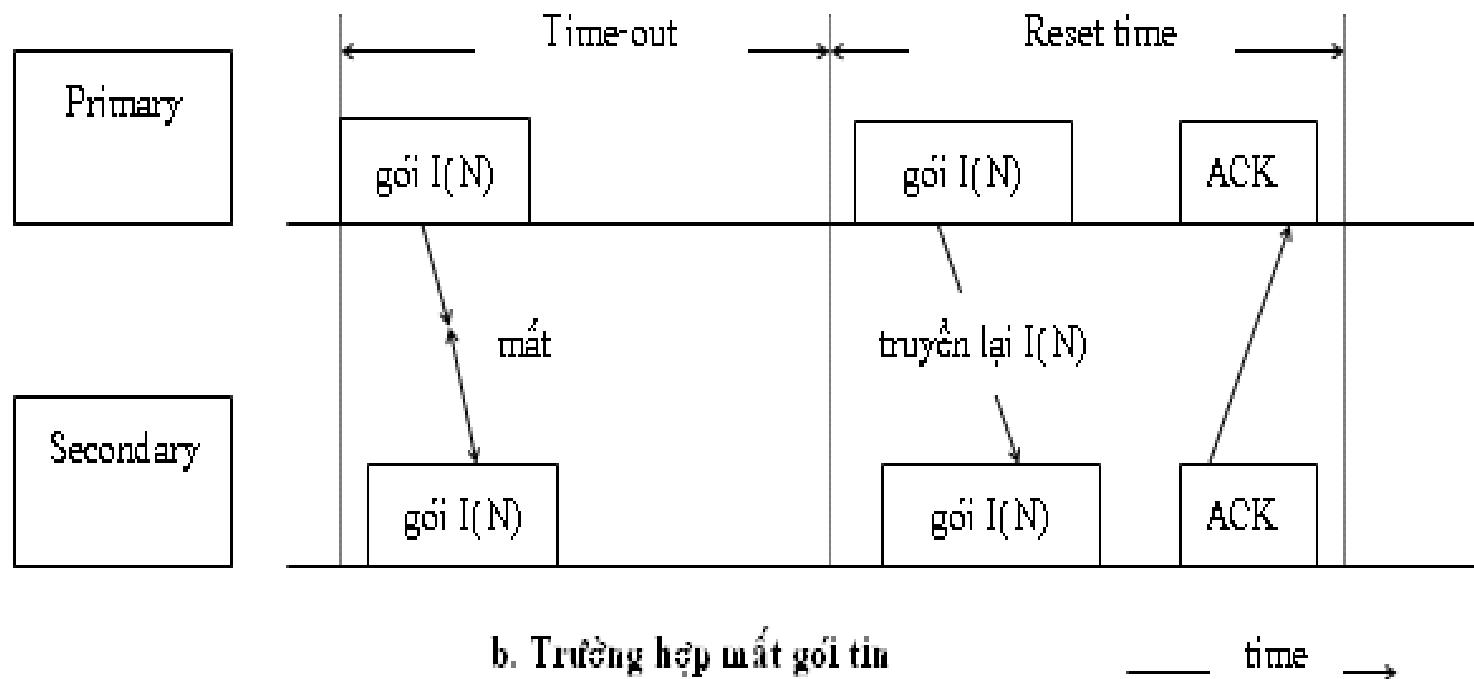
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

* Hoạt động kiểu ẩn :



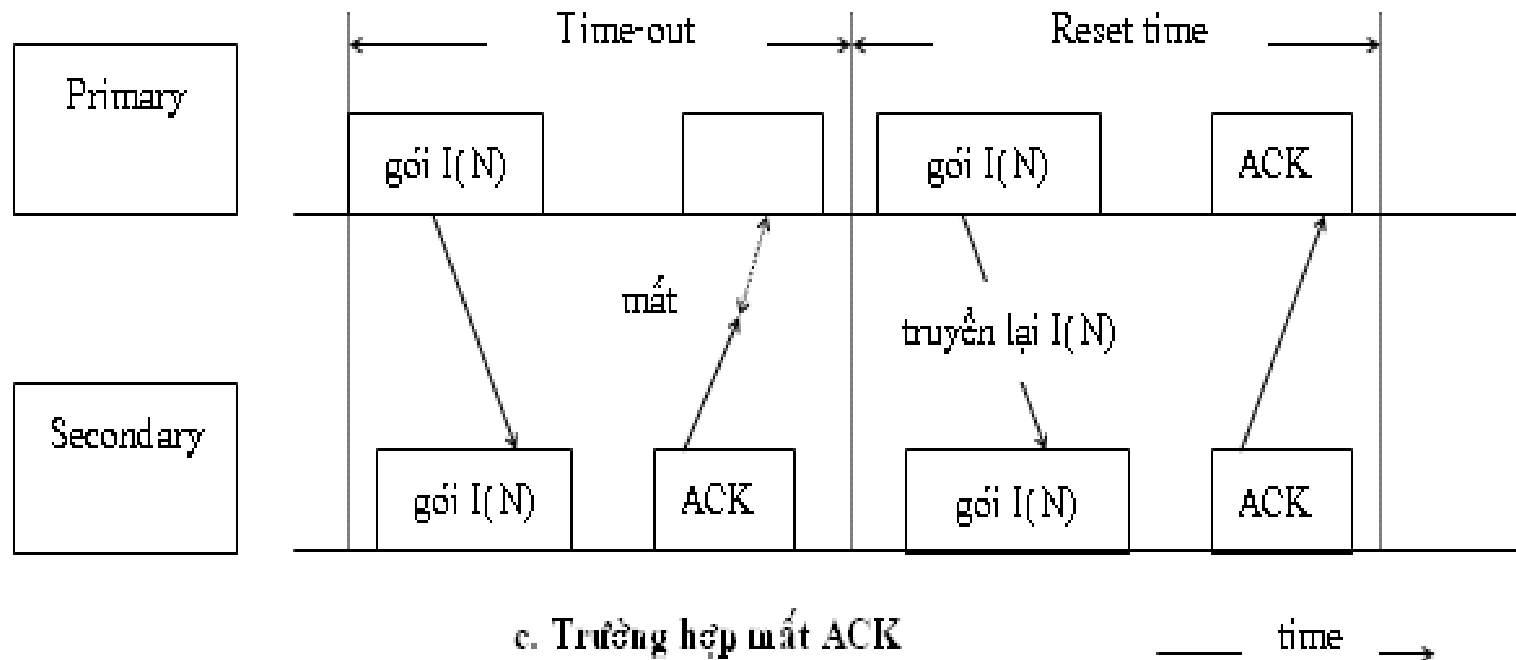
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

* Hoạt động kiểu ẩn :



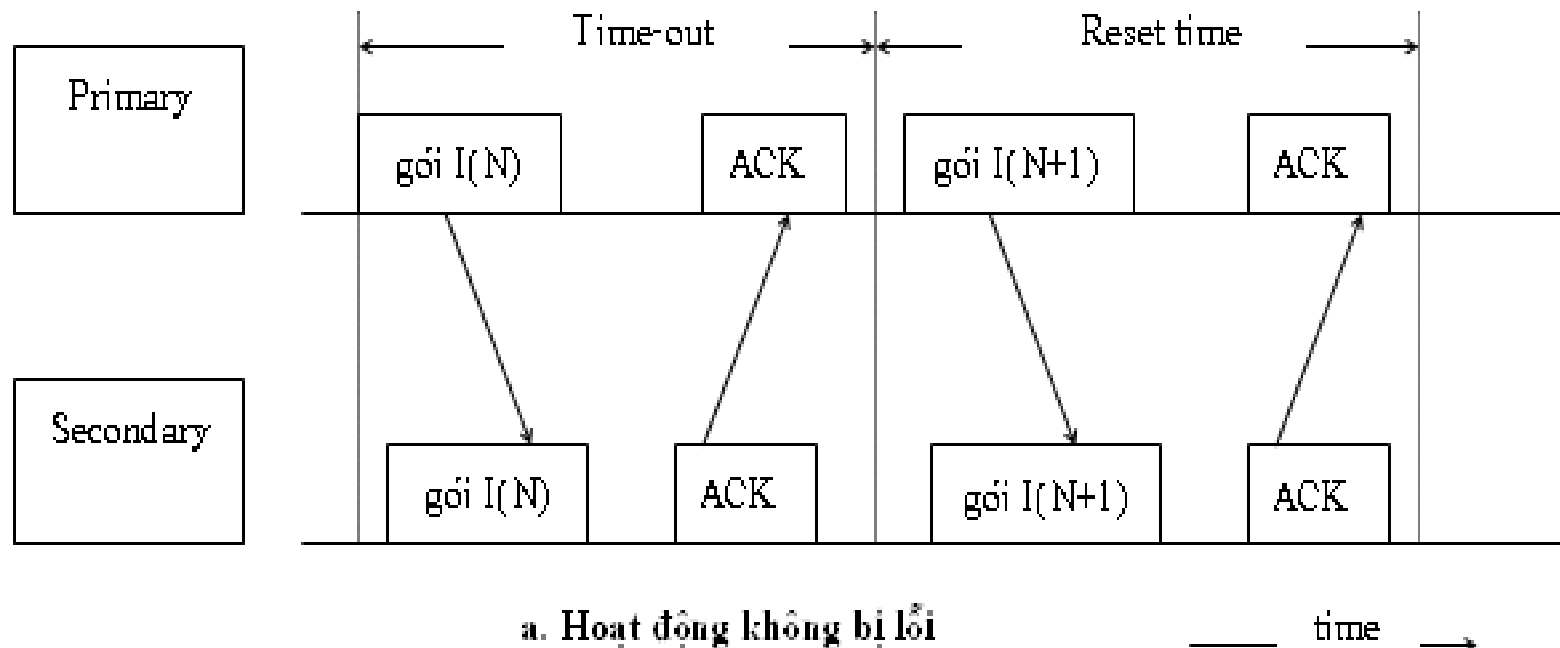
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

- Hoạt động kiểu hiện:



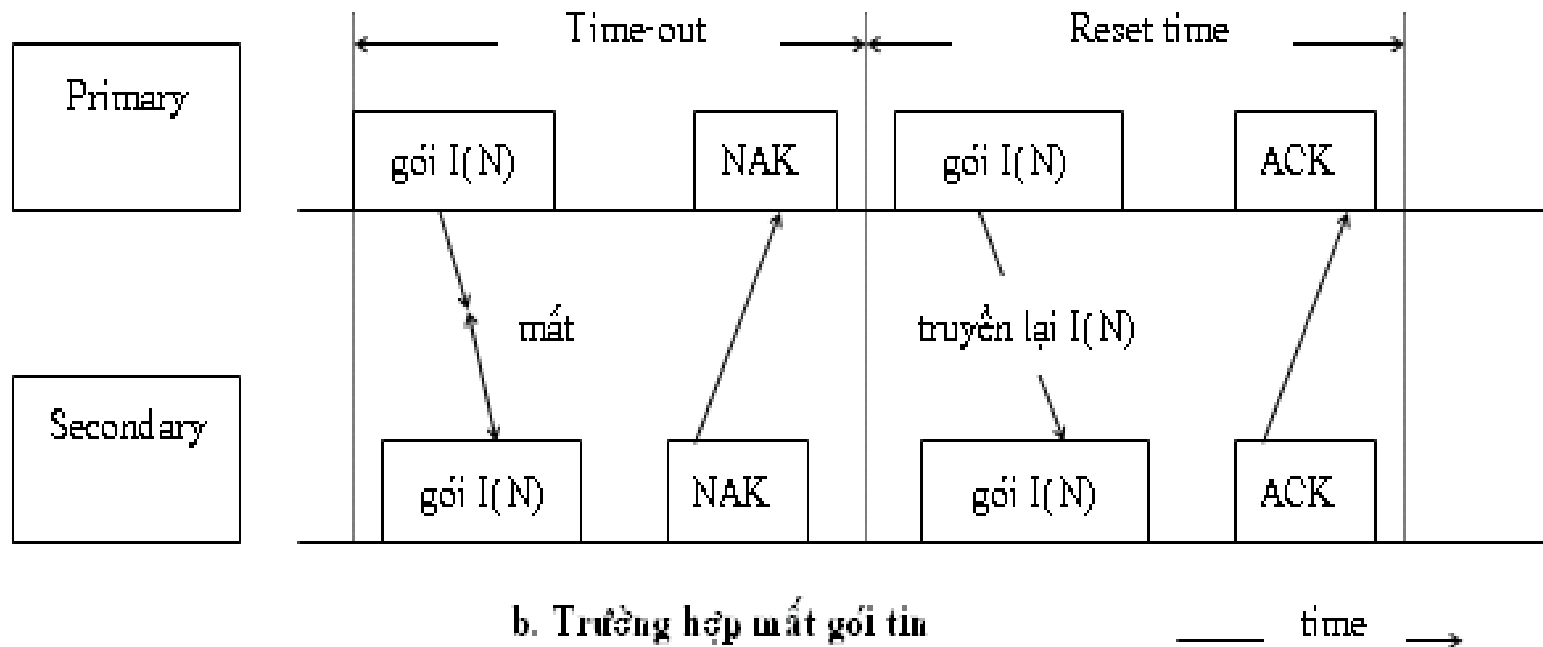
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

* Hoạt động kiểu hiện :



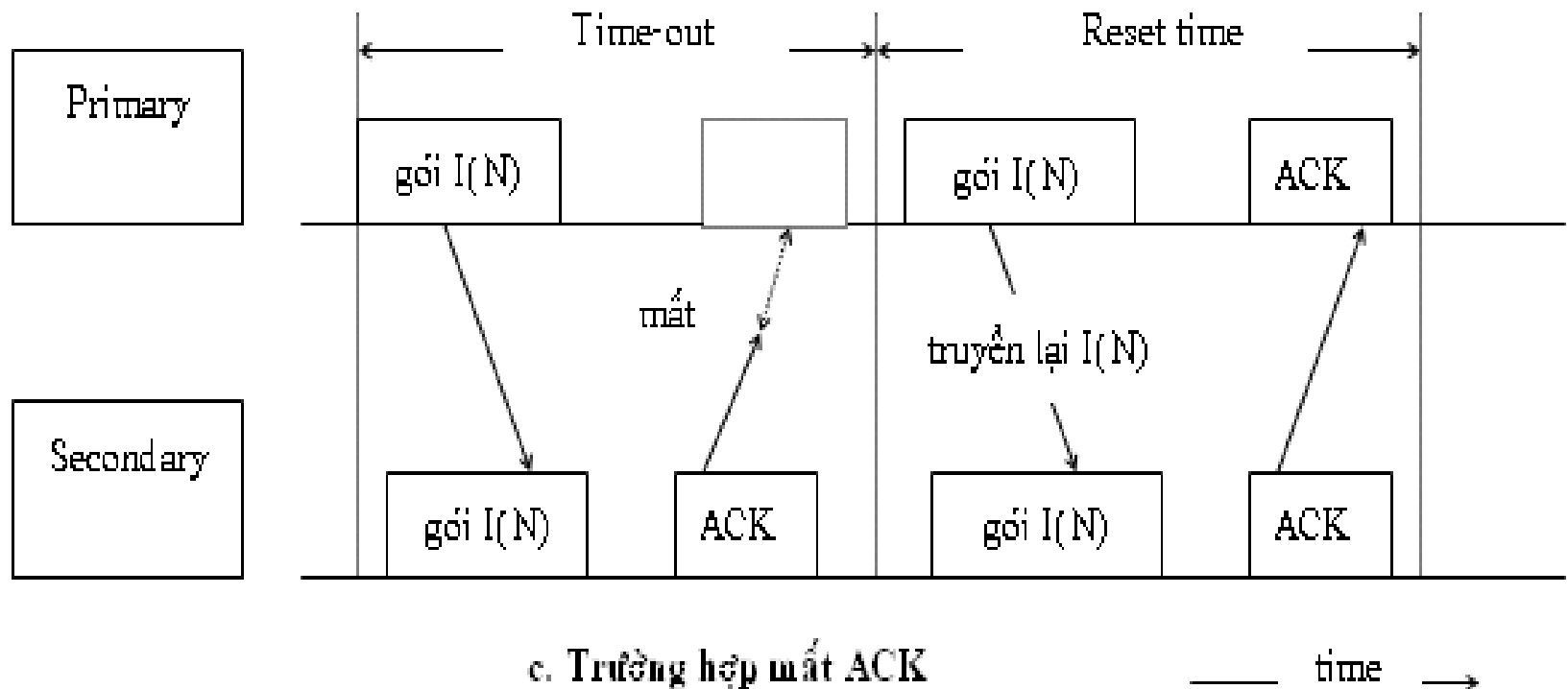
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

* Hoạt động kiểu hiện :



Chương 3:

Các giao thức cơ bản



3.1.3 Cơ chế kiểm soát lỗi của Protocol :

a. Phương pháp kiểm lỗi Idle-RQ :

+ Đối với kiểu ẩn, nếu gói giữ liệu bị mất, bên nhận sẽ không gửi một thông báo nào, phải làm cho bên phát phải đợi đến thời gian time-out, và nó cũng không biết rằng gói dữ liệu bị mất hay ACK bị mất.

+ Ngược lại đối với kiểu hiện , nếu gói dữ liệu mất, bên nhận sẽ gửi thông báo NAK về cho bên phát.

+ Ưu điểm của phương pháp Idle RQ là ít tốn bộ nhớ , nó chỉ cần một vùng RAM đủ chứa một gói thông tin. Nói cách khác Idle RQ điều khiển luồng bằng cửa sổ trượt có cách thức bằng 1.

Phương pháp này được dùng trong các giao thức hướng kí tự (Character Oriented) như giao thức BSC , DDCMP,...

Chương 3:

Các giao thức cơ bản



3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

+ Trong khi Idle RQ là kỹ thuật kiểm lỗi theo nguyên tắc truyền gói nào, chờ kết quả gói đó xong mới tiếp tục truyền gói kế tiếp thì Continuous RQ cho phép truyền hàng loạt.

+ Bên nhận làm việc theo nguyên tắc nhận gói nào thì phúc đáp gói đó. Như vậy bên nhận làm việc theo nguyên tắc FIFO chứa danh sách ghi nhớ tất cả các gói mà nó đã truyền đi.

+ Khi nhận được phúc đáp thành công của gói nào (ACK) từ phía bên nhận, nó sẽ xóa gói đó ra khỏi danh sách ghi nhớ. Bên nhận cũng lập danh sách ghi nhớ tất cả các gói mà nó đã nhận được.

Chương 3:

Các giao thức cơ bản



3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* Selective Repeat:

+ Hồi đáp kiểu ẩn :

Bên nhận chỉ phúc đáp cho những gói mà nó nhận được. Bên phát dựa vào đó để suy đoán là gói nào bị thất lạc trên đường truyền.

Chương 3:

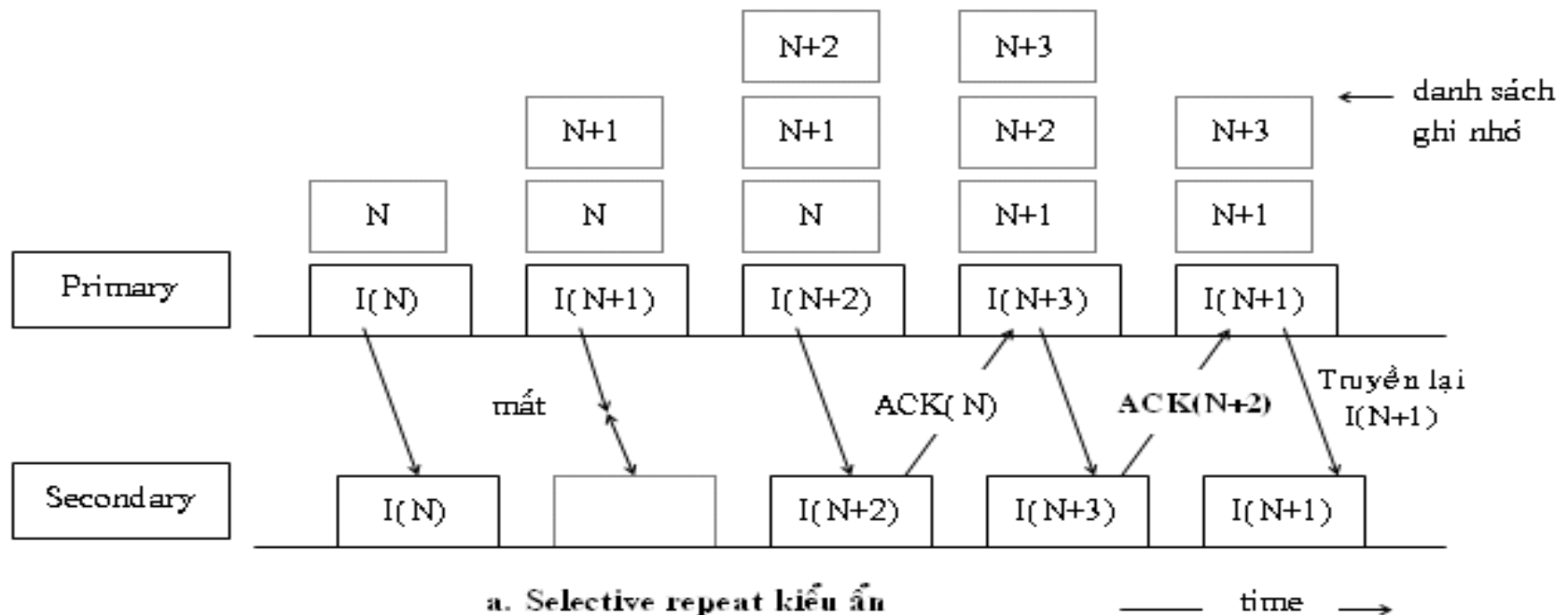
Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* **Selective Repeat:**

+ **Hồi đáp kiểu ẩn :**



Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* Selective Repeat:

+ Hồi đáp kiểu hiện :

Đối với kiểu ẩn việc hồi đáp được thực hiện theo nguyên tắc nhận được gói mới phức đáp mà không quan tâm gói nào bị mất, trong khi kiểu hiện lại quan tâm đến những gói bị mất.

Ví dụ : Nếu bên nhận được hai gói I(1) và I(3) thì nó đoán chắc rằng gói I(2) bị mất và lập tức thông báo về bên truyền rằng NAK(2). Trên đường phản hồi, nếu các gói ACK hoặc NAK bị mất thì bên phát cũng suy đoán được

Chương 3:

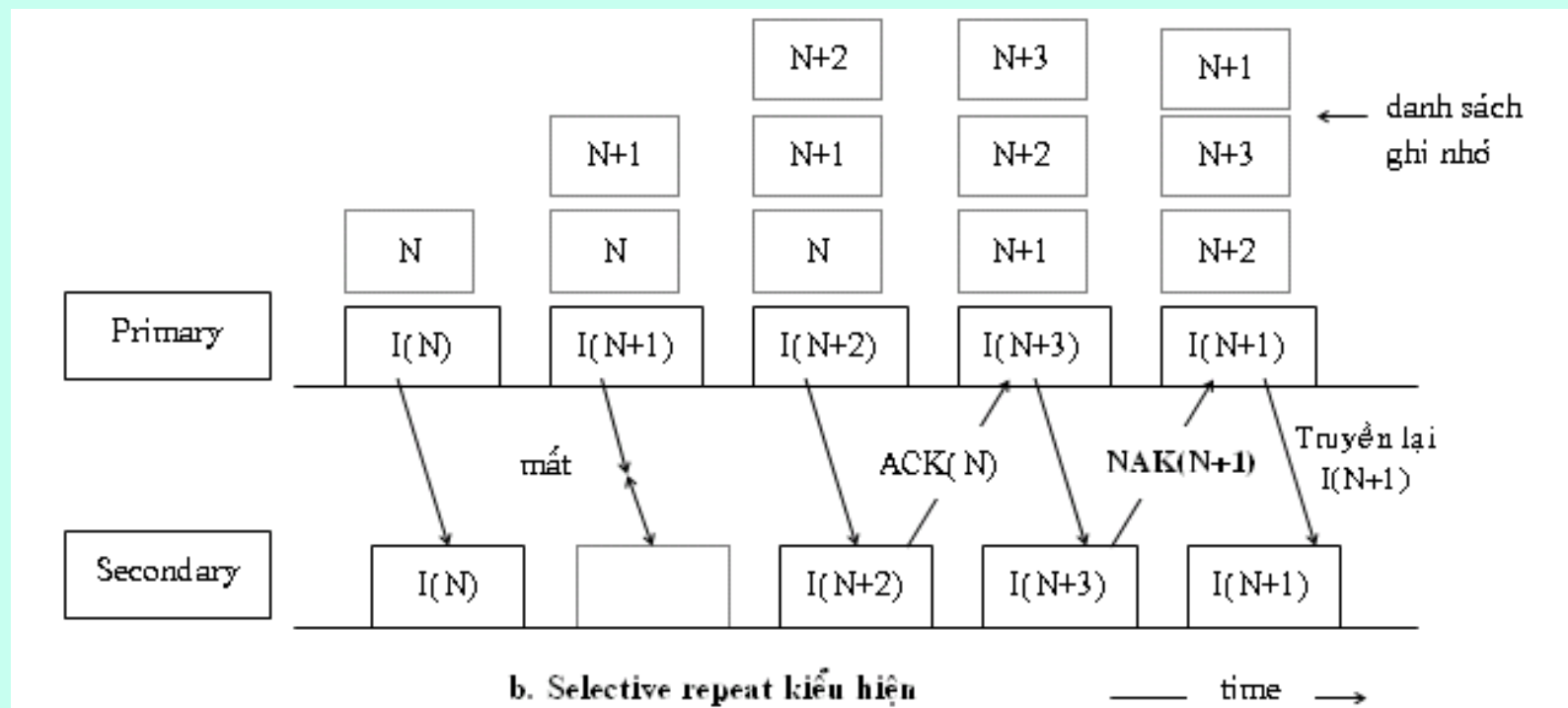
Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* Selective Repeat:

+ Hồi đáp kiểu hiện :



Chương 3:

Các giao thức cơ bản



3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* Selective Repeat:

+ Phương pháp Selective Repeat dùng kỹ thuật điều khiển luồng cửa sổ trượt với kích thước là K (kể cả cửa sổ nhận),

+ Khi truyền sẽ truyền một lượt K gói nhưng không bắt buộc các gói phải có thứ tự liên tục và khi nhận cũng nhận một lượt K gói.

Chương 3:

Các giao thức cơ bản



3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* Phương pháp Goback-N:

+ Phương pháp này dùng kỹ thuật điều khiển luồng cửa sổ trượt với kích thước cửa sổ gửi là K gói và cửa sổ nhận có kích thước là 1, có nghĩa là khi truyền sẽ truyền một lượt K gói có thứ tự liên tục và khi nhận chỉ nhận một gói.

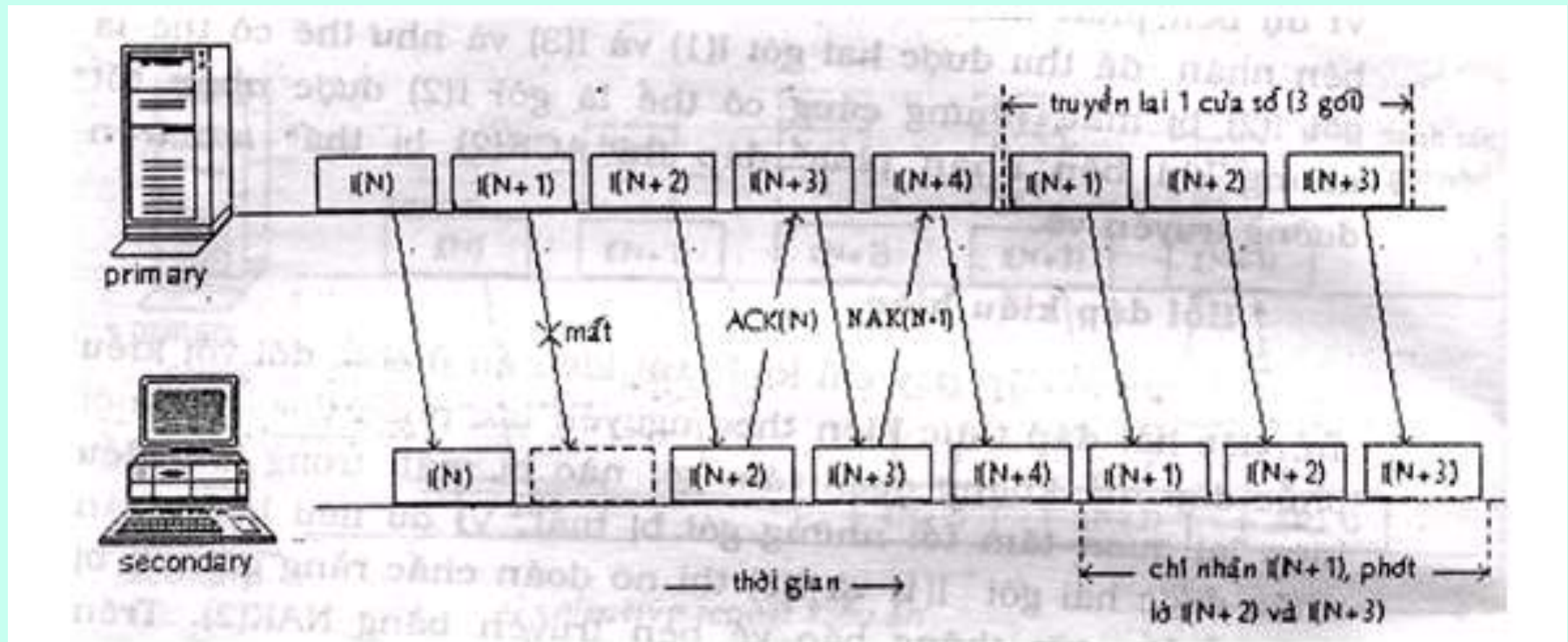
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

b. Phương pháp kiểm lỗi Continuous RQ:

* Phương pháp Goback-N:



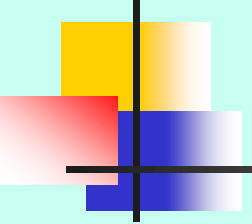
Chương 3:

Các giao thức cơ bản

3.1.3 Cơ chế kiểm soát lỗi của Protocol :

Bảng so sánh 3 phương pháp kiểm soát lỗi :

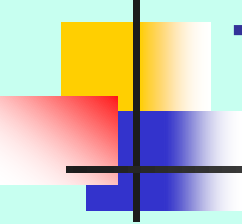
| Protocol | Kích thước của số truyền | Kích thước của số nhận |
|------------------|-----------------------------|---------------------------|
| Idle-RQ | 1 | 1 |
| Selective Repeat | K | K |
| Goback-N | K | 1 |



Chương 3:

Các giao thức cơ bản

- Họ giao thức TCP/IP hiện nay là giao thức được sử dụng rộng rãi nhất để liên kết các máy tính và các mạng.
- Giao thức TCP/IP thực chất là một họ giao thức cho phép các hệ thống mạng cùng làm việc với nhau thông qua việc cung cấp phương tiện truyền thông liên mạng
- TCP (Transmission Control Protocol) là giao thức thuộc tầng vận chuyển và IP (Internet Protocol) là giao thức thuộc tầng mạng của mô hình OSI.



Tầng mạng – Internet Layer

- Giao thức tầng mạng – Internet Protocol
- Cấu trúc gói tin IP
- Địa chỉ IP
- Dịch vụ DHCP
- Giao thức ICMP

Giao thức tầng mạng

– Internet Protocol

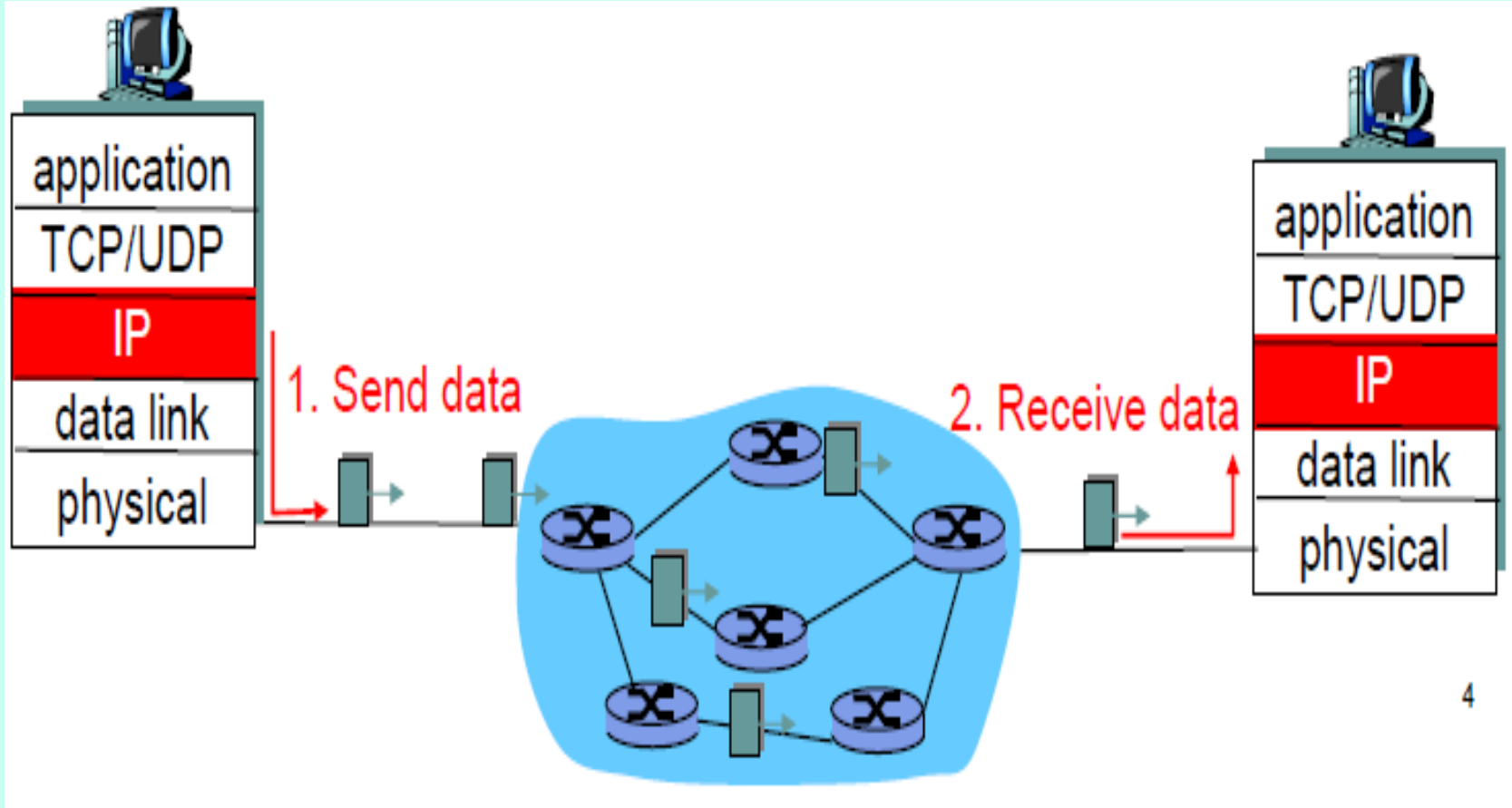
- Khái niệm cơ bản
- Nguyên lý lưu và chuyển tiếp
- Giao thức IP



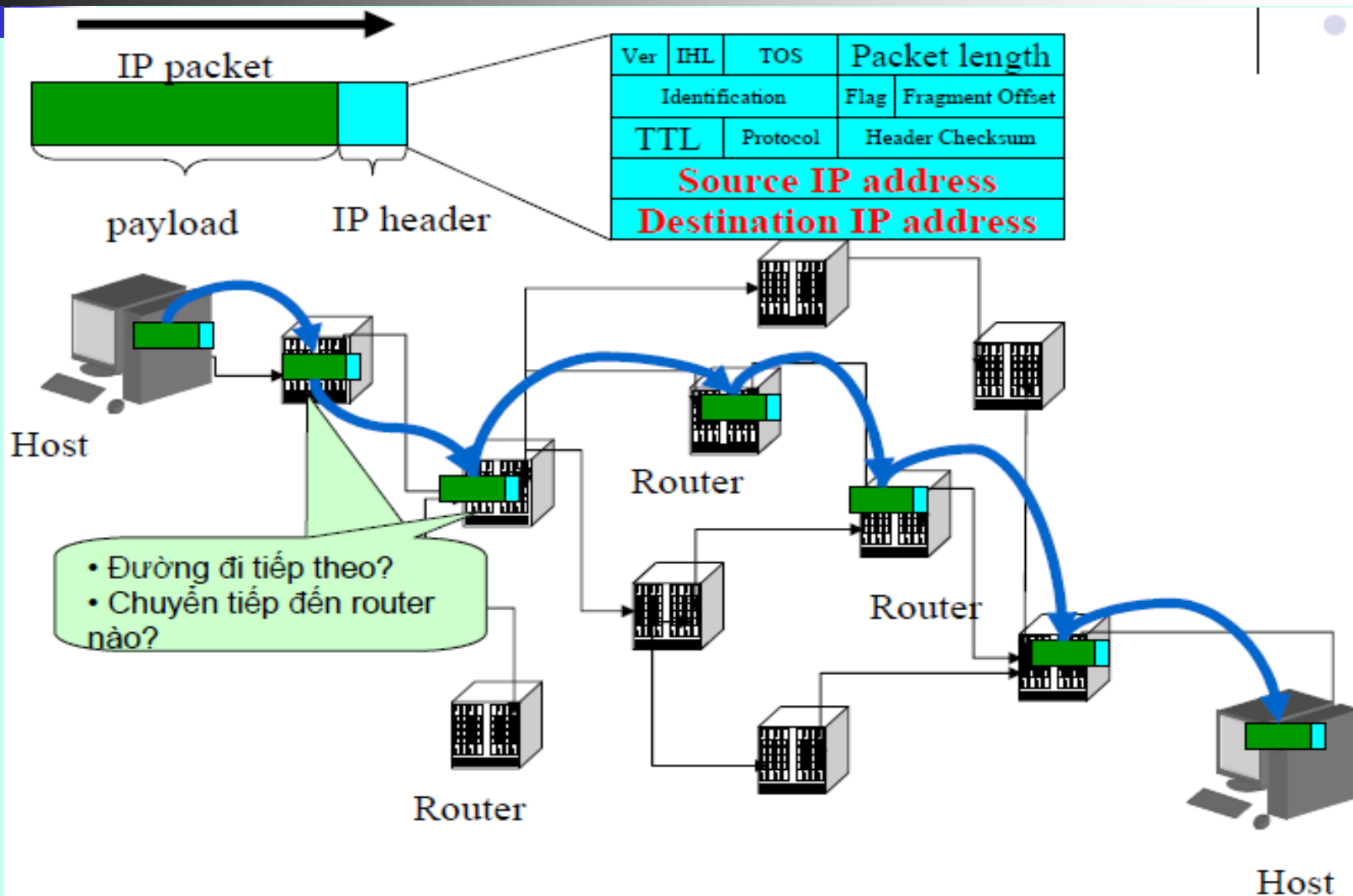
Internet Protocol

- Là một giao thức tầng mạng
- Hai chức năng cơ bản
 - Chọn đường(*Routing*): Xác định đường đi của gói tin từ nguồn đến đích
 - Chuyển tiếp (*Forwarding*): Chuyển dữ liệu từ đầu vào đến đầu ra của bộ định tuyến (router)

Internet Protocol



Chọn đường và chuyển tiếp gói tin





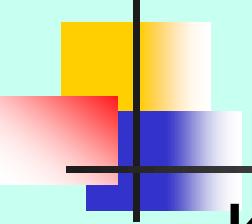
Internet Protocol

- Network Layer: Giữa các máy trạm hoặc các bộ định tuyến (Hosts)
- Transport Layer: Giữa các tiến trình trên máy trạm (Processes)



Đặc điểm giao thức IP

- Không tin cậy/Nhanh
 - Truyền dữ liệu theo phương thức không tin cậy
 - Không có cơ chế phục hồi lỗi
 - Khi cần sẽ sử dụng dịch vụ tầng trên để đảm bảo độ tin cậy
- Giao thức không liên kết
 - Các gói tin được xử lý độc lập



Hoạt động của giao thức IP

- Khi giao thức IP được khởi động nó trở thành một thực thể tồn tại trong máy tính
- Bắt đầu thực hiện những chức năng của mình
- Lúc đó thực thể IP là cấu thành của tầng mạng
- Nhận yêu cầu từ các tầng trên nó và gửi yêu cầu xuống các tầng dưới nó.



Hoạt động của giao thức IP

- Đối với thực thể IP ở máy nguồn, khi nhận được một yêu cầu gửi từ tầng trên, nó thực hiện các bước sau đây:
 - ❖ Tạo một IP datagram dựa trên tham số nhận được.
 - ❖ Tính checksum và ghép vào header của gói tin.
 - ❖ Ra quyết định chọn đường: hoặc là trạm đích nằm trên cùng mạng hoặc một gateway sẽ được chọn cho chặng tiếp theo.
 - ❖ Chuyển gói tin xuống tầng dưới để truyền qua mạng.



Hoạt động của giao thức IP

- ❑ Đối với router, khi nhận được một gói tin đi qua, nó thực hiện các động tác sau:
 - ❖ Tính checksum, nếu sai thì loại bỏ gói tin
 - ❖ Giảm giá trị tham số Time - to Live, nếu thời gian đã hết thì loại bỏ gói tin
 - ❖ Ra quyết định chọn đường
 - ❖ Phân đoạn gói tin, nếu cần
 - ❖ Kiến tạo lại IP header, bao gồm giá trị mới của các vùng Time - to -Live, Fragmentation và Checksum
 - ❖ Chuyển datagram xuống tầng dưới để chuyển qua mạng



Hoạt động của giao thức IP

- Cuối cùng khi một datagram nhận bởi một thực thể IP ở trạm đích, nó sẽ thực hiện bởi các công việc sau:
 - ❖ Tính checksum. Nếu sai thì loại bỏ gói tin
 - ❖ Tập hợp các đoạn của gói tin (nếu có phân đoạn)
 - ❖ Chuyển dữ liệu và các tham số điều khiển lên tầng trên

Cấu trúc gói tin IP

Bit 0 3 4 7 8 15 16 31

| | | | | |
|--------------------|-----|-----------------|----------------|-----------------|
| VER | IHL | Type of service | Total Length | |
| Identification | | | Flags | Fragment offset |
| Time to live | | Protocol | Heder Checksum | |
| Source address | | | | |
| Destintion Address | | | | |
| Option + Padding | | | | |
| Data | | | | |



Cấu trúc gói tin IP

- VER (4 bits): chỉ version hiện hành của giao thức IP hiện được cài đặt
- IHL (4 bits): chỉ độ dài phần đầu (Internet header Length) của gói tin datagram, tính theo đơn vị từ (32 bits).
 - Trường này bắt buộc phải có vì phần đầu IP có thể có độ dài thay đổi tùy ý. Độ dài tối thiểu là 5 từ (20 bytes), độ dài tối đa là 15 từ hay là 60 bytes.
- Type of service (8 bits): đặc tả các tham số về dịch vụ nhằm thông báo cho mạng biết dịch vụ nào mà gói tin muốn được sử dụng
 - Chẳng hạn ưu tiên, thời hạn chậm trễ, năng suất truyền và độ tin cậy.



Cấu trúc gói tin IP

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------|---|---|---|---|---|----------|---|
| Precedence | | | D | T | R | Reserved | |

- Precedence (3 bit): chỉ thị về quyền ưu tiên gửi datagram, nó có giá trị từ 0 (gói tin bình thường) đến 7 (gói tin kiểm soát mạng).
- D (Delay) (1 bit): chỉ độ trễ yêu cầu trong đó
 - D = 0 gói tin có độ trễ bình thường
 - D = 1 gói tin độ trễ thấp
- T (Throughput) (1 bit): chỉ độ thông lượng yêu cầu sử dụng để truyền gói tin với lựa chọn truyền trên đường thông suất thấp hay đường thông suất cao.
 - T = 0 thông lượng bình thường
 - T = 1 thông lượng cao
- R (Reliability) (1 bit): chỉ độ tin cậy yêu cầu
 - R = 0 độ tin cậy bình thường
 - R = 1 độ tin cậy cao



Cấu trúc gói tin IP

- Total Length (16 bits): chỉ độ dài toàn bộ gói tin, kể cả phần đầu tính theo đơn vị byte
- Identification (16 bits): cùng với các tham số khác (như Source Address và Destination Address) tham số này dùng để định danh duy nhất cho một datagram trong khoảng thời gian nó vẫn còn trên liên mạng.



Cấu trúc gói tin IP

- Flags (3 bits): liên quan đến sự phân đoạn (fragment) các datagram
 - Các gói tin khi đi trên đường đi có thể bị phân thành nhiều gói tin nhỏ, trong trường hợp bị phân đoạn thì trường Flags được dùng để điều khiển phân đoạn và tái lắp ghép bộ dữ liệu.
 - Tùy theo giá trị của Flags sẽ có ý nghĩa là gói tin sẽ không phân đoạn, có thể phân đoạn hay là gói tin phân đoạn cuối cùng.
- Fragment Offset : cho biết vị trí dữ liệu thuộc phân đoạn tương ứng với đoạn bắt đầu của gói dữ liệu gốc.



Cấu trúc gói tin IP

- Ý nghĩa cụ thể của trường Flags là:

| | | |
|----------|-----------|-----------|
| <i>0</i> | <i>1</i> | <i>2</i> |
| O | DF | MF |

- bit 0: reserved - chưa sử dụng, luôn lấy giá trị 0.
 - bit 1: (DF) = 0 (May Fragment) = 1 (Don't Fragment)
 - bit 2: (MF) = 0 (Last Fragment) = 1 (More Fragments)
- Fragment Offset (13 bits): chỉ vị trí của đoạn (fragment) ở trong datagram tính theo đơn vị 8 bytes, có nghĩa là phần dữ liệu mỗi gói tin (trừ gói tin cuối cùng) phải chứa một vùng dữ liệu có độ dài là bội số của 8 bytes. Điều này có ý nghĩa là phải nhân giá trị của Fragment offset với 8 để tính ra độ lệch byte.



Cấu trúc gói tin IP

- Time to Live (8 bits): qui định thời gian tồn tại (tính bằng giây) của gói tin trong mạng để tránh tình trạng một gói tin bị quẩn trên mạng. Sau đây là 1 số điều cần lưu ý về trường Time To Live:
 - Nút trung gian của mạng không được gởi 1 gói tin mà trường này có giá trị = 0.
 - Một giao thức có thể ấn định Time To Live để thực hiện cuộc ra tìm tài nguyên trên mạng trong phạm vi mở rộng.
 - Một giá trị cố định tối thiểu phải đủ lớn cho mạng hoạt động tốt.
- Protocol (8 bits): chỉ giao thức tầng trên kế tiếp sẽ nhận vùng dữ liệu ở trạm đích (hiện tại thường là TCP hoặc UDP được cài đặt trên IP).
 - Ví dụ: TCP có giá trị trường Protocol là 6, UDP có giá trị trường Protocol là 17



Cấu trúc gói tin IP

- Header Checksum (16 bits): Mã kiểm soát lỗi của header gói tin IP.
- Source Address (32 bits): Địa chỉ của máy nguồn.
- Destination Address (32 bits): địa chỉ của máy đích
- Options (độ dài thay đổi): khai báo các lựa chọn do người gửi yêu cầu (tuỳ theo từng chương trình).
- Padding (độ dài thay đổi): Vùng đệm, được dùng để đảm bảo cho phần header luôn kết thúc ở một mốc 32 bits.
- Data (độ dài thay đổi): Trên một mạng cục bộ như vậy, hai trạm chỉ có thể liên lạc với nhau nếu chúng biết địa chỉ vật lý của nhau. Như vậy vấn đề đặt ra là phải thực hiện ánh xạ giữa địa chỉ IP (32 bits) và địa chỉ vật lý (48 bits) của một trạm.



Địa chỉ IP

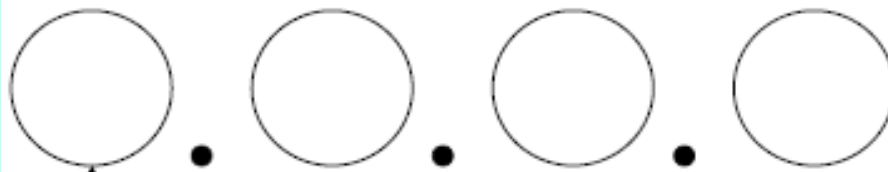
- Để định danh các trạm (host) trong liên mạng người ta dùng địa chỉ IP.
- Mỗi địa chỉ IP có độ dài 32 bits được tách thành 4 vùng (octet), mỗi vùng 1 byte.
- Mục đích của địa chỉ IP là để định danh duy nhất cho một máy tính bất kỳ trên liên mạng.



Địa chỉ IP

- Để địa chỉ không được trùng nhau cần phải có cấu trúc địa chỉ đặc biệt quản lý thống nhất
- Trung tâm thông tin mạng Internet - Network Information Center (NIC) chủ trì phân phối
- NIC chỉ phân địa chỉ mạng (Net ID) còn địa chỉ máy chủ trên mạng đó (Host ID) do các Tổ chức quản lý Internet của từng quốc gia một tự phân phối

Địa chỉ IP



8 bits

0 – 255 integer

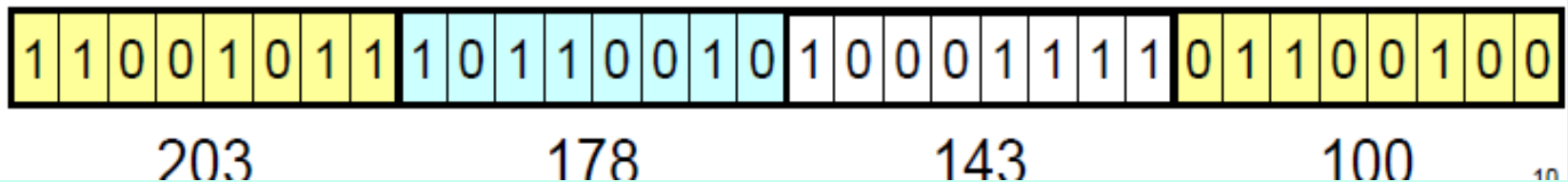
Ví dụ:

203.178.136.63

259.12.49.192

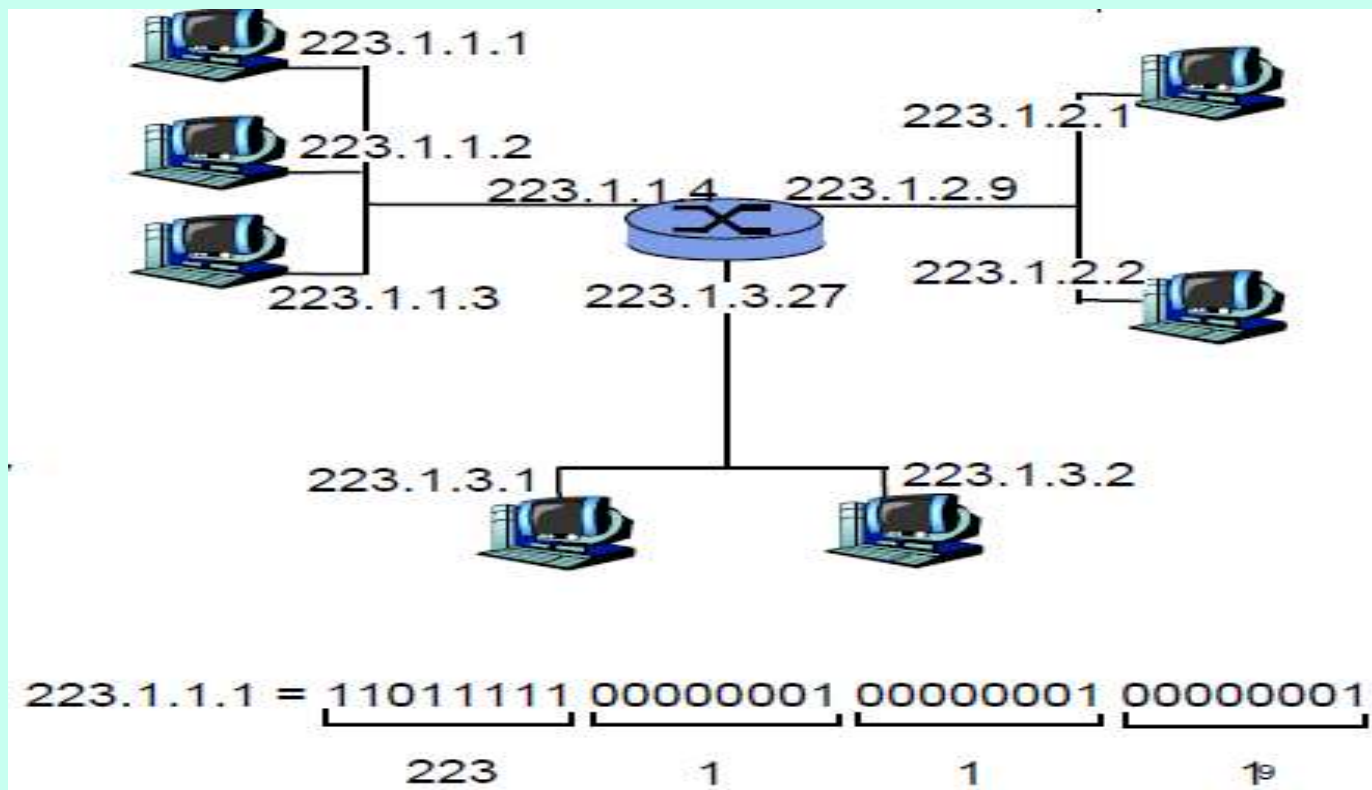
133.27.4.27

Sử dụng 4 phần 8 bits để miêu tả một địa chỉ 32 bits



Địa chỉ IP

- Mỗi địa chỉ IP được gán cho giao diện
- Địa chỉ IP có tính duy nhất





Địa chỉ IP

- Do tổ chức và độ lớn của các mạng con (subnet) của liên mạng có thể khác nhau
- Người ta chia các địa chỉ IP thành 5 lớp :
 - A, B, C, D và E
- Các bit đầu tiên của byte đầu tiên được dùng để định danh lớp địa chỉ

0 - lớp A

10 - lớp B

110 - lớp C

1110 - lớp D

11110 - lớp E

Địa chỉ IP

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------|-------|------|------|------|------------------------|--|--|-------|---|---|--|--|--|--|-------|--|---|---|--|--|--|-------|--|--|--|--|--|--|--|--|
| | 8bits | | | | | | | 8bits | | | | | | | 8bits | | | | | | | 8bits | | | | | | | | |
| Class A | 0 | 7bit | | | | | | | H | H | | | | | | | H | H | | | | | | | | | | | | |
| Class B | 1 | 0 | 6bit | | | | | | N | H | | | | | | | H | H | | | | | | | | | | | | |
| Class C | 1 | 1 | 0 | 5bit | | | | | N | N | | | | | | | H | H | | | | | | | | | | | | |
| Class D | 1 | 1 | 1 | 0 | Multicast | | | | | | | | | | | | | | | | | | | | | | | | | |
| Class E | 1 | 1 | 1 | 1 | Reserve for future use | | | | | | | | | | | | | | | | | | | | | | | | | |

| | # of network | # of hosts |
|---------|--------------|------------|
| Class A | 128 | 2^{24} |
| Class B | 16384 | 65536 |
| Class C | 2^{21} | 256 |



Địa chỉ IP

- ❑ Lớp A cho phép định danh tới 128 mạng, với tối đa 16 triệu host trên mỗi mạng.
 - ❑ Lớp này được dùng cho các mạng có số trạm cực lớn.
- ❑ Lớp B cho phép định danh tới 16384 mạng, với tối đa 65536 host trên mỗi mạng.
- ❑ Lớp C cho phép định danh tới 2 triệu mạng, với tối đa 256 host trên mỗi mạng.
 - ❑ Lớp này được dùng cho các mạng có ít trạm.
- ❑ Lớp D dành riêng cho lớp kỹ thuật multicasting.
- ❑ Lớp E được dành những ứng dụng tương lai.



Địa chỉ IP

- Khoảng IP của các lớp
 - Lớp A : Từ 0.0.0.0 đến 127.0.0.0
 - Lớp B : Từ 128.0.0.0 đến 191.255.0.0
 - Lớp C : Từ 192.0.0.0 đến 223.255.255.0
 - Lớp D : Từ 224.0.0.0 đến 239.255.255.0
Không phân
 - Lớp E : Từ 240.0.0.0 đến 255.0.0.0 Không phân



Các dạng địa chỉ

- Địa chỉ mạng
 - Địa chỉ IP gán cho một mạng
- Địa chỉ máy trạm
 - Địa chỉ IP gán cho một card mạng
- Địa chỉ quảng bá
 - Địa chỉ dùng để gửi cho tất cả máy trạm trong mạng
 - Toàn bit 1 phần ứng với địa chỉ máy trạm



Cấu trúc logic

- Địa chỉ IP có 2 phần :
 - Net ID : địa chỉ mạng
 - Host ID : địa chỉ máy trạm trên NetID

NETID

HOSTID

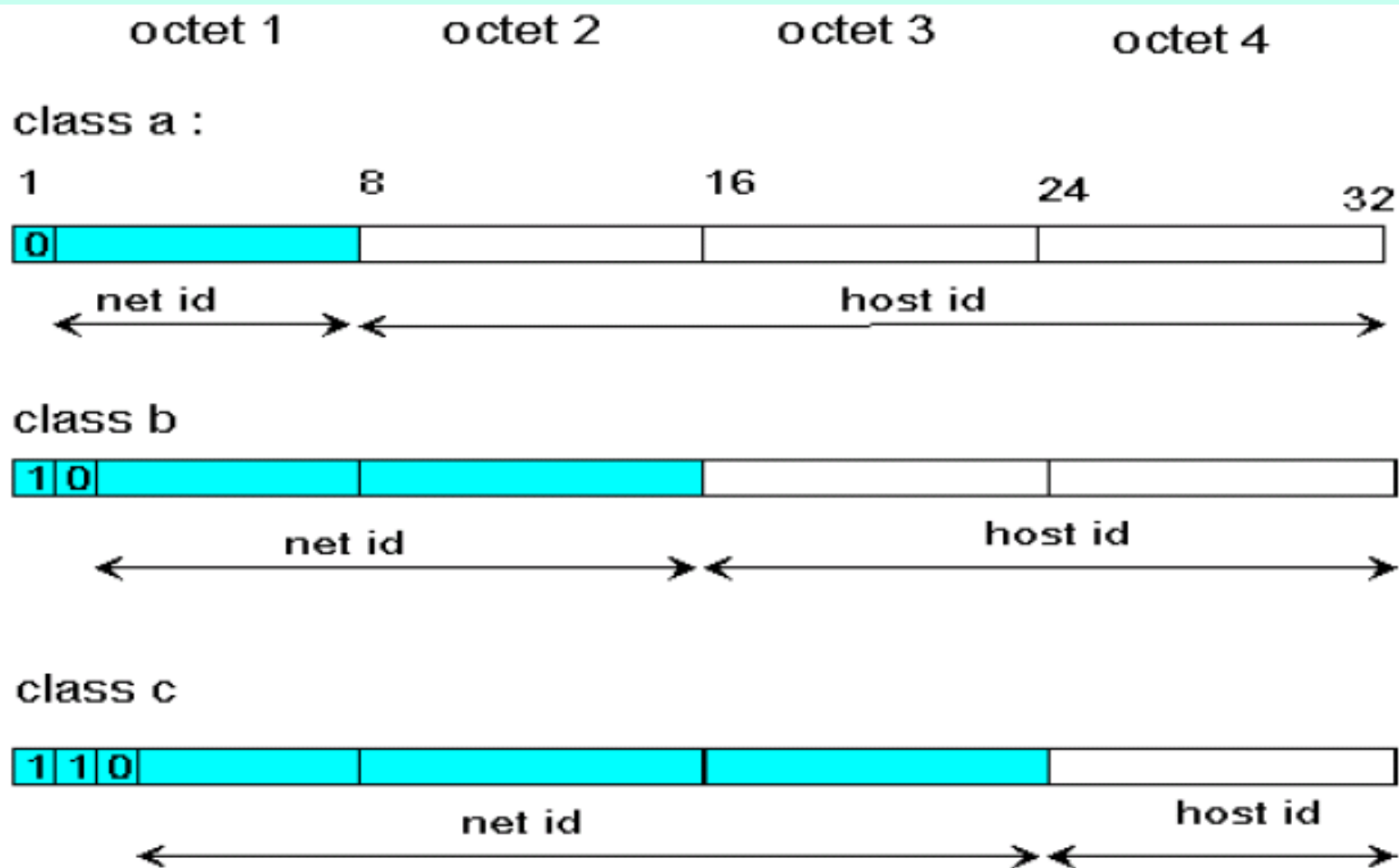


Địa chỉ IP

Cấu trúc của các địa chỉ IP như sau:

- Mạng lớp A: địa chỉ mạng (netid) là 1 Byte và địa chỉ host (hostid) là 3 byte.
- Mạng lớp B: địa chỉ mạng (netid) là 2 Byte và địa chỉ host (hostid) là 2 byte.
- Mạng lớp C: địa chỉ mạng (netid) là 3 Byte và địa chỉ host (hostid) là 1 byte.

Địa chỉ IP





Địa chỉ IP

+ Một địa chỉ có hostid = 0 được dùng để hướng tới mạng định danh bởi vùng netid.

+ Ngược lại, một địa chỉ có vùng hostid gồm toàn số 1 được dùng để hướng tới tất cả các host nối vào mạng netid,

+ Nếu vùng netid cũng gồm toàn số 1 thì nó hướng tới tất cả các host trong liên mạng

Địa chỉ IP

Ví dụ :

- 128.3.2.3
- 192.0.1.255
- 192.168.3.17
- 19.45.2.3
- 86.2.3.5

| | | | |
|----------|----------|----------|----------|
| 10000000 | 00000011 | 00000010 | 00000011 |
|----------|----------|----------|----------|

= 128.3.2.3 (lớp B)

netid = 128.3

hostid = 2.3

| | | | |
|----------|----------|----------|----------|
| 11000000 | 00000000 | 00000001 | 11111111 |
|----------|----------|----------|----------|

= 192.0.1.255 (lớp C)

netid = 192.0.1

hostid = 255 → hướng

tới tất cả các host

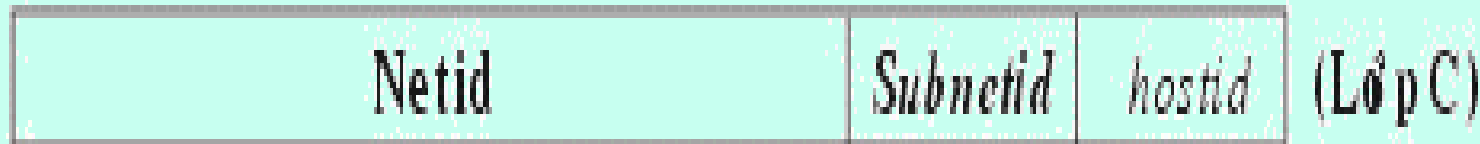
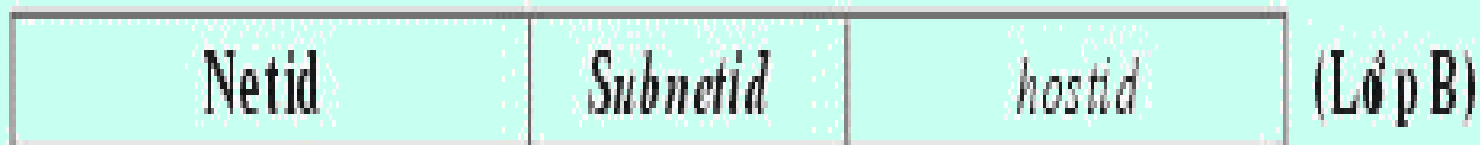
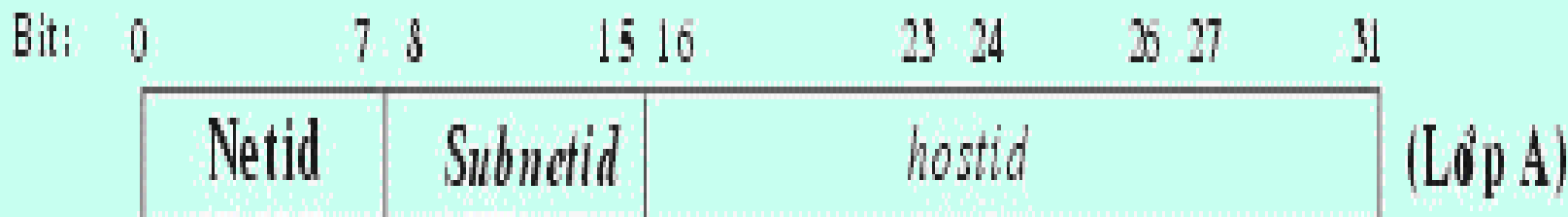


Địa chỉ IP

- Lưu ý rằng các địa chỉ IP dùng để định danh các host và mạng ở tầng mạng của mô hình OSI
 - Chúng không phải là các địa chỉ vật lý (hay địa chỉ MAC) của các trạm trên một mạng cục bộ (Ethernet, Token Ring...).
- Trong nhiều trường hợp, một mạng có thể được chia thành nhiều mạng con (subnet), lúc đó có thể đưa thêm các vùng subnetid để định danh các mạng con.
 - Vùng **subnetid** được lấy từ vùng **hostid**, cụ thể đối với lớp A, B, C như ví dụ sau:

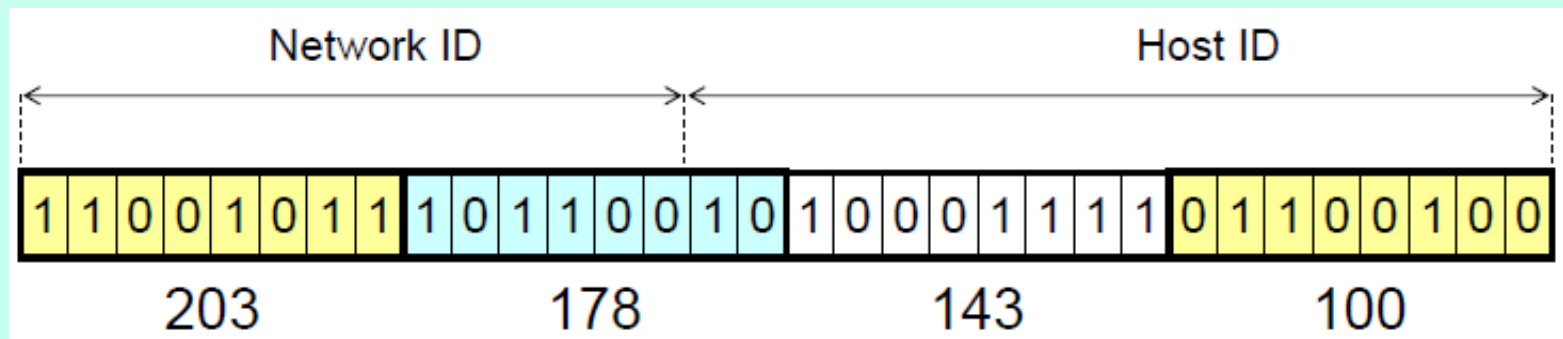


Địa chỉ IP



Địa chỉ IP

- Làm sao để biết phần nào cho máy trạm, phần nào cho mạng
 - Phân lớp địa chỉ
 - Không phân lớp





Địa chỉ IP

- Lãng phí không gian địa chỉ
 - Việc phân chia thành các lớp hạn chế việc sử dụng toàn bộ không gian địa chỉ
- Nhằm hạn chế việc sử dụng lãng phí không gian địa chỉ (CIDR :Classless Inter Domain Routing)
 - Phần địa chỉ mạng sẽ có độ dài bất kỳ
 - Dạng địa chỉ : a.b.c.d/x (x: là mặt nạ mạng, chính là số bit trong phần ứng với địa chỉ mạng)
 - Với cách địa chỉ hóa theo CIDR, địa chỉ IP và mặt nạ mạng luôn phải đi cùng nhau



Mặt nạ mạng

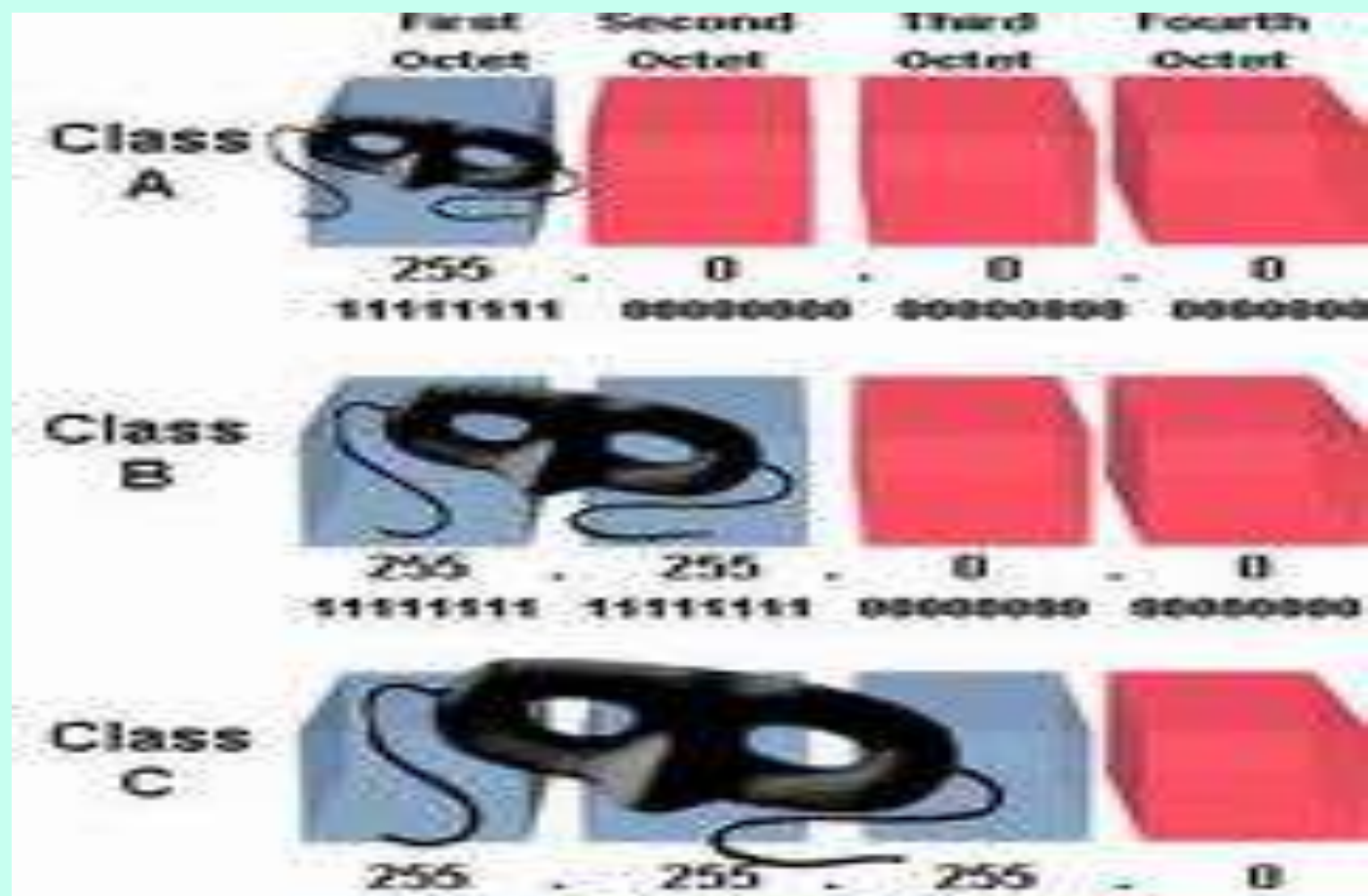
- Mặt nạ mạng chia địa chỉ IP thành 2 phần :
 - Phần ứng với máy trạm
 - Phần ứng với mạng
- Dùng toán tử AND
 - Tính địa chỉ mạng
 - Tính khoảng địa chỉ IP



Mặt nạ mạng

- Là một dãy số 32 bit dùng để nhận diện ra netID
 - $\text{netID} = \text{IP address AND netmask}$
- Chuẩn thì :
 - Lớp A có subnetmask : 255.0.0.0
 - Lớp B có subnetmask 255.255.0.0
 - Lớp C có subnetmask 255.255.255.0

Mặt nạ mạng





Mặt nạ mạng

Ví dụ:

- ▶ IP 203.165.7.123 netmask 255.255.255.0
- ▶ → netID= IP 203.165.7.0
- ▶ IP 20.162.7.123 thì netmask ????
- ▶ → netID= IP 20.0.0.0
- ▶ IP 153.162.7.123 netmask ????
- ▶ → netID= IP 153.162.0.0
- ▶ IP 203.162.7.123 netmask 255.255.0.0
- ▶ → netID= IP ????
- ▶ IP 203.162.0.123 netmask 255.255.255.0
- ▶ → netID= IP ????



Bảng tóm tắt

| Lớp | netID | netmask | Số máy/mạng | Số mạng |
|-----|---------|---------------|---------------|--------------|
| A | x.0.0.0 | 255.0.0.0 | $256*256*256$ | 128 |
| B | x.x.0.0 | 255.255.0.0 | $256*256$ | $64*256$ |
| C | x.x.x.0 | 255.255.255.0 | 256 | $32*256*256$ |



Các địa chỉ đặc biệt

- ▶ Địa chỉ loopback là các địa chỉ có **127.x.x.x**
- ▶ Địa chỉ **broadcast** là địa chỉ mạng sẽ dùng để quảng bá mạng mình cho các mạng khác biết.
 - ▶ Mục đích giúp cho các router cập nhật bảng định tuyến
- ▶ Địa chỉ broadcast được qui định là địa chỉ cuối cùng của một mạng
- ▶ Ví dụ: 167.8.5.0 thì broadcast là 167.8.255.255
 192.45.67.1
 154.63.72.5



Chú ý:

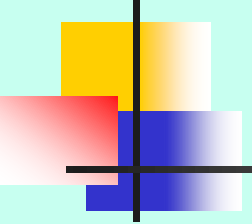
- Các máy cùng mạng sẽ có cùng NETID, subnetmask và khác host ID
- Hai máy khác mạng hoặc khác NET ID hoặc khác subnet mask



Private address

- ▶ Là vùng địa chỉ cho phép cấu hình mạng cục bộ mà không cần phải mua
- ▶ Không được sử dụng trên internet để cấp phát

| Lớp | Khoảng mạng | Số mạng |
|-----|-----------------------|---------|
| A | 10.0.0.0 | 1 |
| B | 172.16.0.0-172.31.0.0 | 16 |
| C | 192.168.x.0 | 256 |

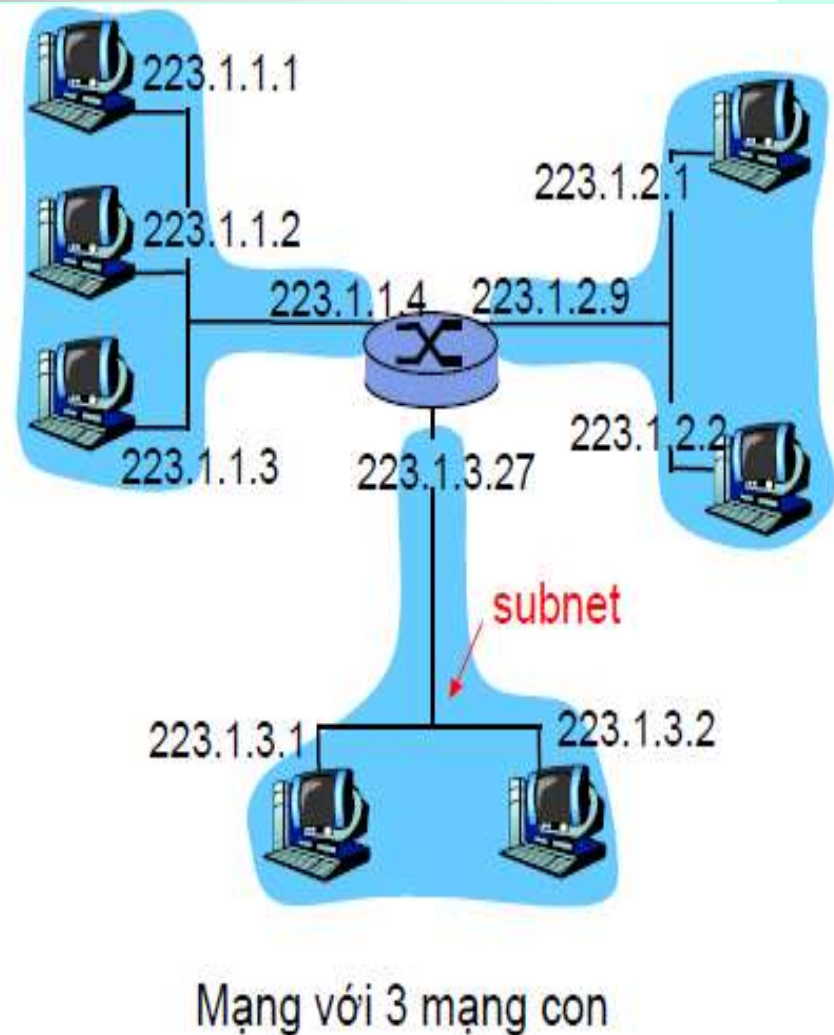


Mạng con- Subnet

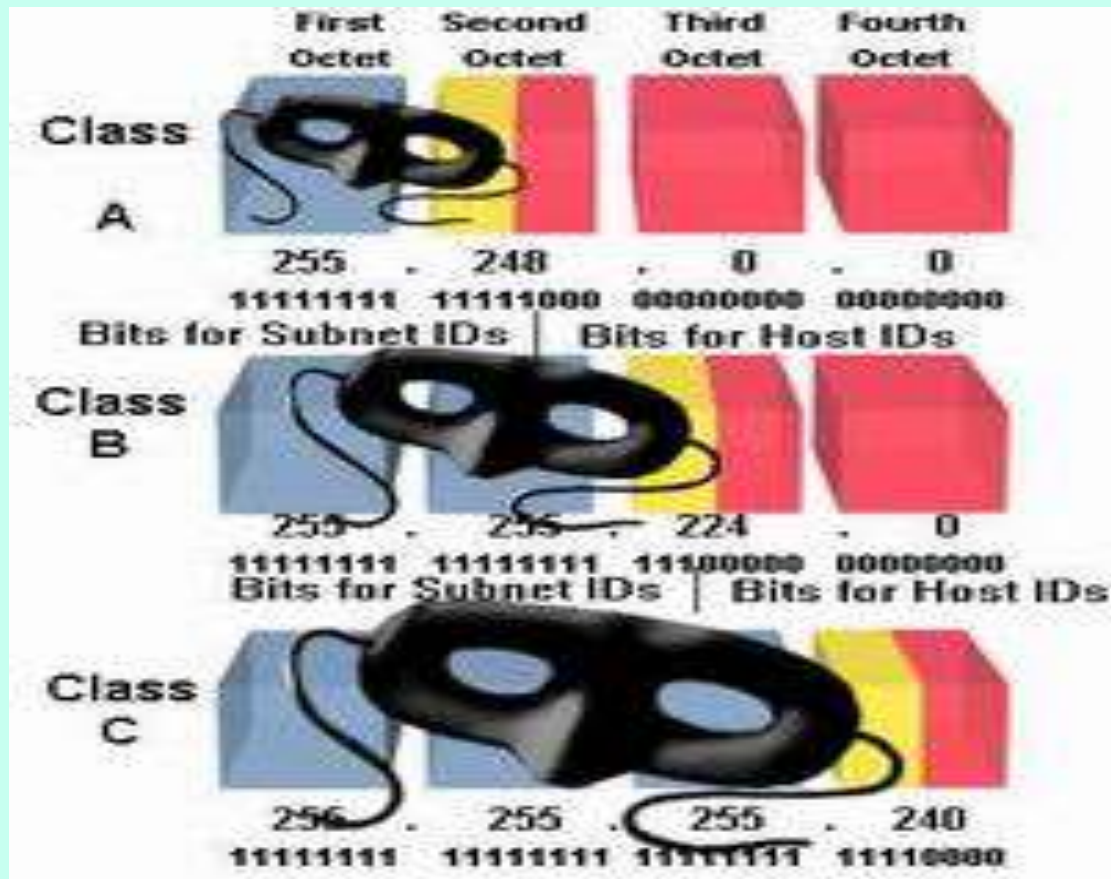
- Một NETID thật trên internet phải mua
- Do nhu cầu chia tách mạng con trong một đơn vị
- Số máy trên một mạng quá nhiều nên cần chia nhỏ mạng → nhiều mạng con

Mạng con - Subnet

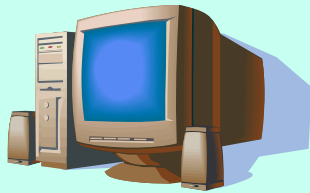
- Là một phần của một mạng nào đó
 - ISP thường được gán một khối địa chỉ IP
 - Một vài mạng con sẽ được tạo ra
- Tạo subnet như thế nào?
 - Sử dụng một mặt nạ mạng dài hơn



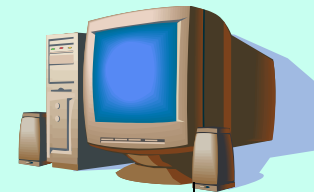
Mạng con - Subnet



Mạng con - Subnet



192.168.1.57



192.168.1.51

255.255.255.0

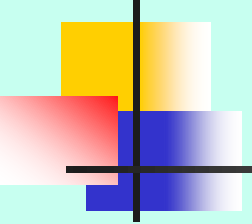


255.255.255.192



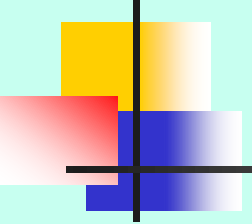
255.255.255.248





Mạng con - Subnet

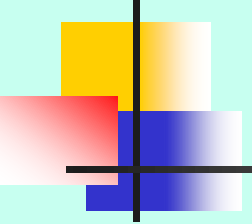
| | | | |
|---------|----------|--------|----------|
| ■ 0... | 10000000 | 000... | |
| ■ 1... | | 001... | |
| ■ 00... | | 010... | |
| ■ 01... | 11000000 | 011... | 11100000 |
| ■ 10... | | 100... | |
| ■ 11... | | 101... | |
| | | 110... | |
| | | 111... | |



Mạng con - Subnet

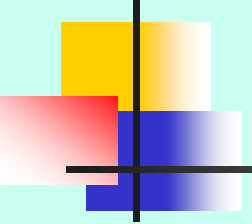
Cách chia :

- Có m bit còn dư
- Mượn n bit
- Vậy có thể có 2^n mạng , $2^n - 2$ mạng được sử dụng
- Số bù là $m < 8 \rightarrow (2^{m-n}) \bmod 256$
- $m > 8 \rightarrow ((2^{m-n}) / 256) \bmod 256$ (***)
- Số trong subnetmask : $(2^m - 2^{m-n}) \bmod 256$
 - Nếu $n < 8 \rightarrow x$
 - Nếu $n > 8 \rightarrow 255.x$
 - Nếu $n > 16 \rightarrow 255.255.x$



Mạng con - Subnet

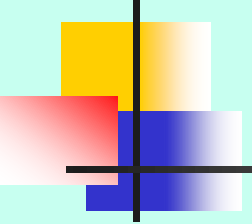
- ▶ Số bù dùng để ghi khoảng mạng
- ▶ Vd : số bù là 32 của mạng 172.16.0.0
- ▶ Mạng 1 : 172.16.0.0
172.16.31.255
- ▶ Mạng 2 : 172.16.32.0
172.16.63.255
- ▶ Mạng 3 : 172.16.64.0
172.16.95.255
- ▶ Mạng 4 : 172.16.96.0
- ▶



Mạng con - Subnet

- Số máy trên một mạng con là:

$$2^{m-n} - 2$$



Mạng con - Subnet

- Mượn 1 bit có 2 mạng \rightarrow số bù là 128 \rightarrow 128
- Mượn 2 bit có 4 mạng \rightarrow số bù là 64 \rightarrow 192
- Mượn 3 bit có 8 mạng \rightarrow số bù là 32 \rightarrow 224
- Mượn 4 bit có 16 mạng \rightarrow số bù là 16 \rightarrow 240
- Mượn 5 bit có 32 mạng \rightarrow số bù là 8 \rightarrow 248
- Mượn 6 bit có 64 mạng \rightarrow số bù là 4 \rightarrow 252
- Mượn 7 bit có 128 mạng \rightarrow số bù là 2 \rightarrow 254
- Mượn 8 bit có 256 mạng \rightarrow số bù là 1 \rightarrow 255



Gán địa chỉ IP

- Do người quản trị gán trực tiếp
 - Windows: Control panel->network->configuration->Tcp/ip->properties
- Sử dụng dịch vụ DHCP (Dynamic Host Configuration Protocol)
 - Plug and Play



Quản lý địa chỉ IP

- Một mạng con lấy địa chỉ IP từ đâu?
 - Chia ra từ không gian địa chỉ ISP
- ISP lấy địa chỉ IP từ đâu?
- ICANN : Internet Corporation for Assigned Names and Numbers
 - Cấp phát địa chỉ
 - Quản DNS,..



DHCP (Dynamic Host Configuration Protocol)

- Mỗi thiết bị trên mạng có dùng bộ giao thức TCP/IP đều phải có một địa chỉ IP hợp lệ, phân biệt.
- Để hỗ trợ cho vấn đề theo dõi và cấp phát các địa chỉ IP được chính xác, đã phát triển ra giao thức DHCP (Dynamic Host Configuration Protocol)
- Dịch vụ DHCP này cho phép chúng ta cấp động các thông số cấu hình mạng cho các máy trạm (client)



DHCP (Dynamic Host Configuration Protocol)

- Cơ chế sử dụng các thông số mạng được cấp phát động có ưu điểm hơn so với cơ chế khai báo tĩnh các thông số mạng như:
 - Khắc phục được tình trạng đụng địa chỉ IP và giảm chi phí quản trị cho hệ thống mạng.
 - Giúp cho các nhà cung cấp dịch vụ (ISP) tiết kiệm được số lượng địa chỉ IP thật (Public IP).
 - Phù hợp cho các máy tính thường xuyên di chuyển qua lại giữa các mạng.
 - Kết hợp với hệ thống mạng không dây (Wireless) cung cấp các điểm Hotspot như: nhà ga, sân bay, trường học,...



HOẠT ĐỘNG CỦA GIAO THỨC DHCP

Giao thức **DHCP** làm việc theo mô hình **client/server**. Quá trình tương tác giữa **DHCP client** và **server** diễn ra theo các bước sau:

- Khi máy **client** khởi động, máy sẽ gửi **broadcast** gói tin **DHCPDISCOVER**, yêu cầu một **server** phục vụ mình.
- Gói tin này cũng chứa địa chỉ **MAC** của máy **client**.



HOẠT ĐỘNG CỦA GIAO THỨC DHCP

- Các máy **Server** trên mạng khi nhận được gói tin yêu cầu đó, nếu còn khả năng cung cấp địa chỉ **IP**, đều gửi lại cho máy **Client** gói tin **DHCPOFFER**
 - Nhằm đề nghị cho thuê một địa chỉ **IP** trong một khoản thời gian nhất định, kèm theo là một **subnet mask** và địa chỉ của **Server**.
 - **Server** sẽ không cấp phát địa chỉ **IP** vừa đề nghị cho những **Client** khác trong suốt quá trình thương thuyết.



Hoạt động của giao thức DHCP

- Máy **Client** sẽ lựa chọn một trong những lời đề nghị (**DHCP OFFER**) và gửi **broadcast** lại gói tin **DHCP REQUEST** chấp nhận lời đề nghị đó.
- Điều này cho phép các lời đề nghị không được chấp nhận sẽ được các **Server** rút lại và dùng để cấp phát cho **Client** khác.



Hoạt động của giao thức DHCP

- Máy **Server** được **Client** chấp nhận sẽ gửi ngược lại một gói tin **DHCPACK** như là một lời xác nhận
- Cho biết là địa chỉ **IP** đó, **subnet mask** đó và thời hạn cho sử dụng đó sẽ chính thức được áp dụng.
 - Ngoài ra **Server** còn gửi kèm theo những thông tin cấu hình bổ sung như địa chỉ của **gateway** mặc định, địa chỉ **DNS Server**, ...



Các giao thức điều khiển

- Để mạng với giao thức IP hoạt động được tốt người ta cần một số giao thức bổ sung
- *Giao thức ICMP (Internet Control Message Protocol):* *Giao thức này thực hiện truyền các thông báo điều khiển (báo cáo về các tình trạng các lỗi trên mạng...) giữa các gateway hoặc một nút của liên mạng.*
 - Tình trạng lỗi có thể là: một gói tin IP không thể tới đích của nó, hoặc một router không đủ bộ nhớ đệm để lưu và chuyển một gói tin IP
 - Một thông báo ICMP được tạo và chuyển cho IP. IP sẽ "bọc" (encapsulate) thông báo đó với một IP header và truyền đến cho router hoặc trạm đích



Các giao thức điều khiển

- *Giao thức ARP (Address Resolution Protocol): Ở đây cần lưu ý rằng các địa chỉ IP được dùng để định danh các host và mạng ở tầng mạng của mô hình OSI, và chúng không phải là các địa chỉ vật lý*
 - Trên một mạng cục bộ hai trạm chỉ có thể liên lạc với nhau nếu chúng biết địa chỉ vật lý của nhau.
 - Vấn đề đặt ra là phải tìm được ánh xạ giữa địa chỉ IP (32 bits) và địa chỉ vật lý của một trạm.
 - *Giao thức ARP đã được xây dựng để tìm địa chỉ vật lý từ địa chỉ IP khi cần thiết.*
- *Giao thức RARP (Reverse Address Resolution Protocol): Là giao thức ngược với giao thức ARP.*
 - *Giao thức RARP được dùng để tìm địa chỉ IP từ địa chỉ vật lý.*

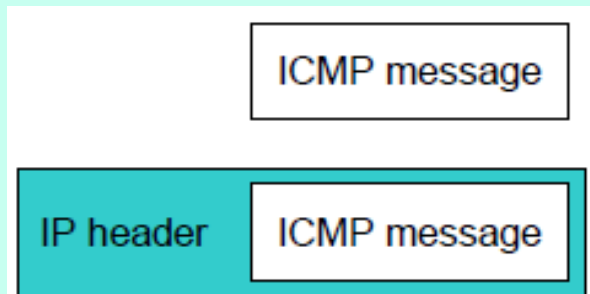


Tổng quan ICMP

- IP là giao thức không tin cậy, không liên kết
 - Thiếu các cơ chế hỗ trợ và kiểm soát lỗi
- ICMP được sử dụng ở tầng mạng để trao đổi thông tin
 - Báo lỗi gói tin không đến được máy trạm, một mạng, một cổng, một giao thức
 - Thông điệp phản hồi

Tổng quan ICMP

- Cũng là giao thức tầng mạng, “phía trên” IP
 - Thông điệp ICMP chứa trong các gói tin IP
- ICMP message : Type, Code cùng với 8 bytes đầu tiên của gói tin IP bị lỗi



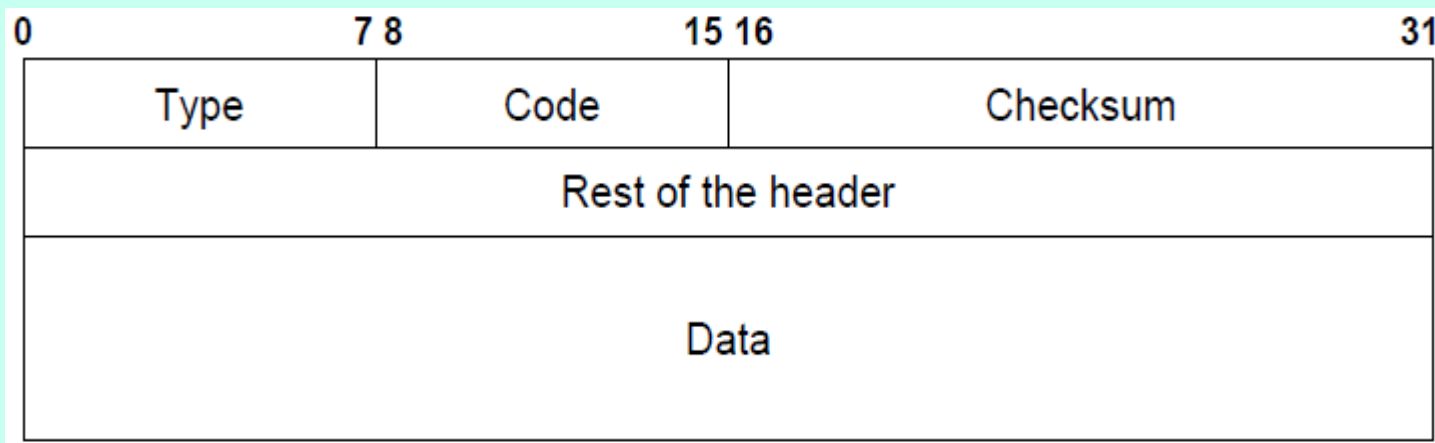
| | | | | |
|------------------------|----------|-----------------|--------------|-------------------------|
| Ver | HLEN | DS | Total Length | |
| Identification | | | Flags | Fragmentation offset |
| TTL | Protocol | Header Checksum | | |
| Source IP address | | | | |
| Destination IP address | | | | |
| Option | | | | |

Protocol:
1: ICMP
2: IGMP
6: TCP
17: UDP
89: OSPF



Khuôn dạng gói tin ICMP

- Type : dạng gói tin ICMP
- Code : Nguyên nhân gây lỗi
- Checksum
- Mỗi dạng có phần còn lại tương ứng





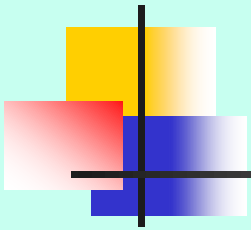
Khuôn dạng gói tin ICMP

- ICMP luôn hoạt động song song suốt với người sử dụng
- Có thể sử dụng ICMP thông qua các công cụ debug
 - ping : sử dụng để kiểm tra kết nối
 - traceroute : Công cụ dò vết đường đi



Tầng giao vận

- Tổng quan tầng giao vận
- Giao thức UDP
- Giao thức TCP

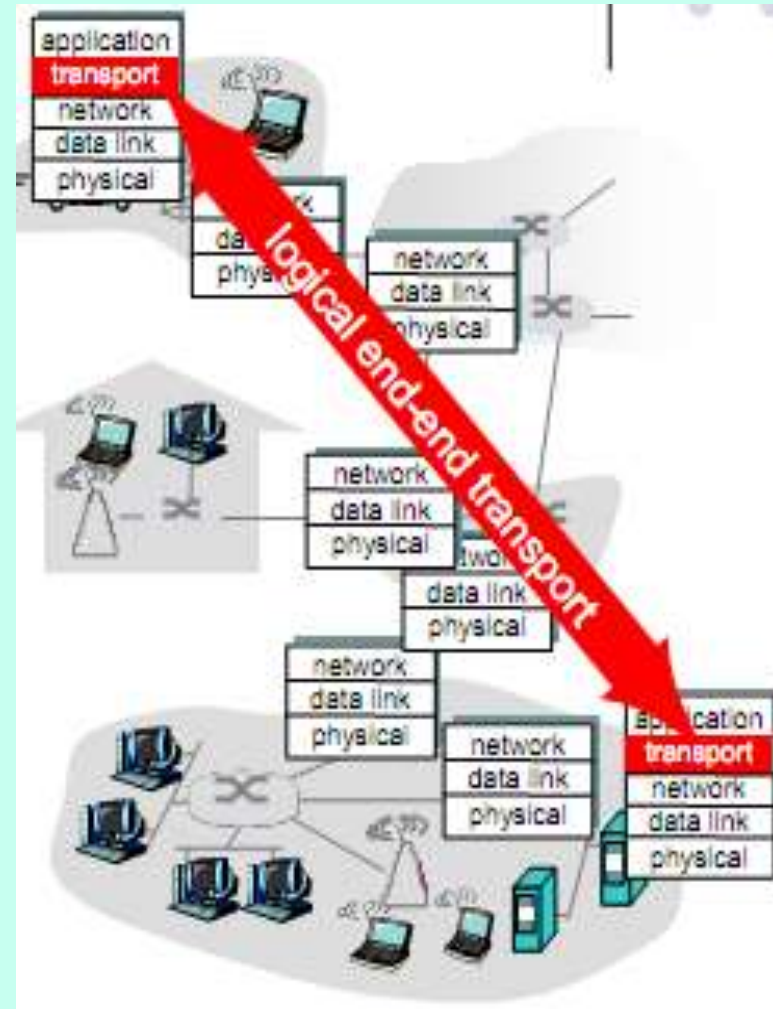


Tổng quan tầng giao vận

- Cung cấp phương tiện truyền giữa các ứng dụng cuối
- Bên gửi :
 - Nhận dữ liệu từ ứng dụng
 - Đặt dữ liệu vào các đoạn tin và chuyển cho tầng mạng
 - Nếu dữ liệu quá lớn thì sẽ được chia thành nhiều phần và sẽ đặt vào các đoạn tin khác nhau.
- Bên nhận :
 - Nhận các đoạn tin từ tầng mạng
 - Tập hợp dữ liệu và chuyển lên cho ứng dụng

Tổng quan tầng giao vận

- Được cài đặt trên các hệ thống cuối
 - Không cài đặt trên các routers, switches,..
- Hai dạng dịch vụ giao vận
 - Không tin cậy, không kết nối : UDP
 - Tin cậy và hướng kết nối : TCP





Sử dụng hai loại dịch vụ

- Các yêu cầu đến từ tầng ứng dụng thì đa dạng
- Các ứng dụng cần dịch vụ với 100% độ tin cậy :
Web, Email, ...
 - Sử dụng dịch vụ của TCP
- Các ứng dụng cần chuyển dữ liệu nhanh, có khả năng chịu lỗi : VoIP, Video Streaming
 - Sử dụng dịch vụ của UDP



Ứng dụng và dịch vụ giao vận

| Ứng dụng | Giao thức ứng dụng | Giao thức giao vận |
|------------------------|--|--------------------|
| e-mail | SMTP | TCP |
| remote terminal access | Telnet | TCP |
| Web | HTTP | TCP |
| file transfer | FTP | TCP |
| streaming multimedia | giao thức riêng (e.g. RealNetworks) | TCP or UDP |
| Internet telephony | giao thức riêng (e.g., Vonage, Dialpad) | thường là UDP |

UDP

(User Datagram Protocol)

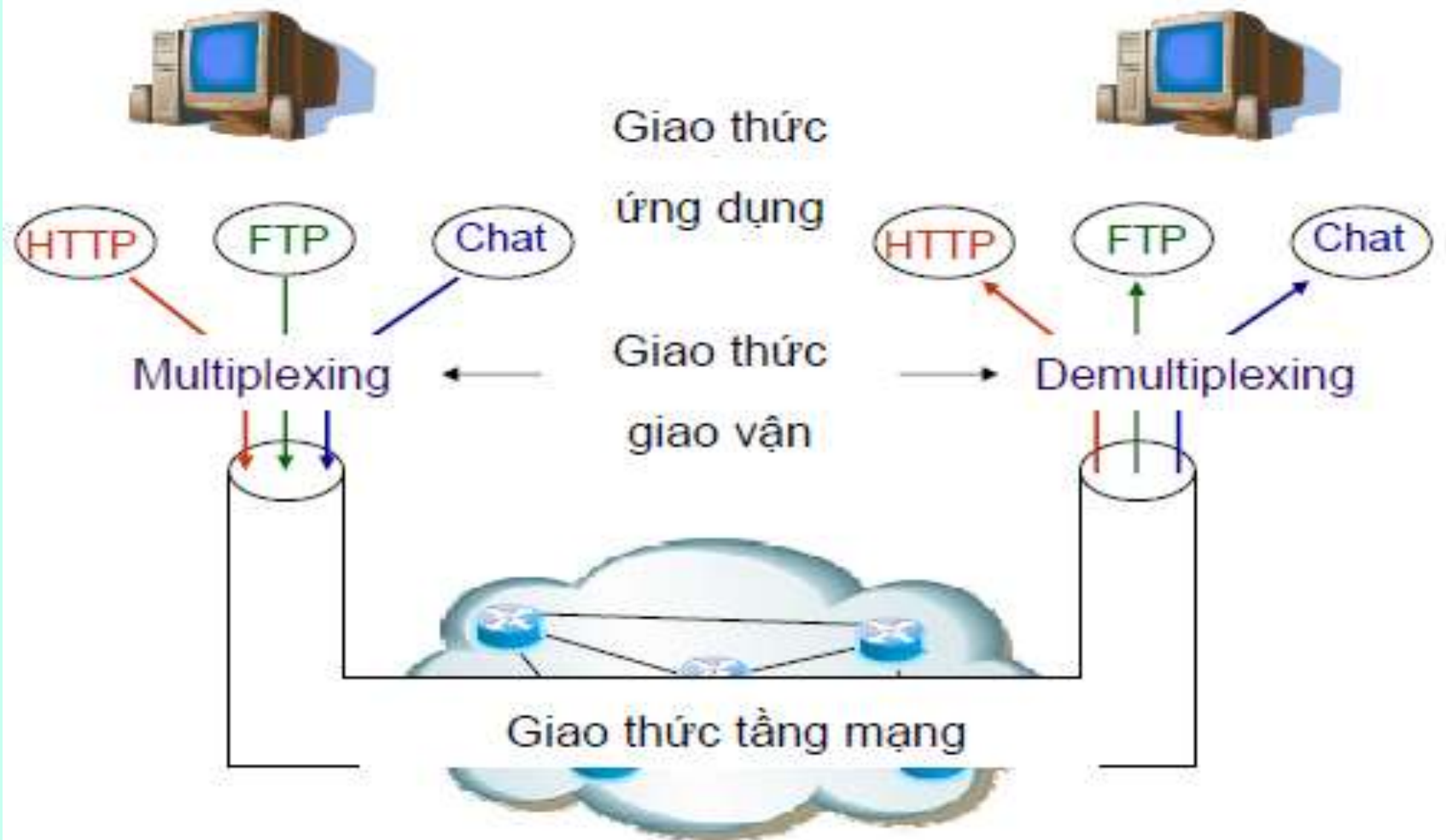
- Tổng quan UDP
- Khuôn dạng gói tin



Tổng quan UDP

- Tại sao sử dụng UDP
 - Không cần phải thiết lập liên kết
 - Phần đầu đoạn tin nhỏ
 - Không cần lưu lại trạng thái liên kết ở bên gửi và bên nhận
 - Không có quản lý tắc nghẽn: UDP gửi dữ liệu nhanh nhất, nhiều nhất nếu có thể
- Chức năng UDP
 - Hợp kênh/phân kênh
 - Phát hiện lỗi bit bằng checksum

Hợp kênh/phân kênh





Hợp kênh/phân kênh

- Tại tầng mạng, gói tin được định danh bởi địa chỉ IP để xác định máy trạm
- Để phân biệt các ứng dụng trên cùng một máy
 - Sử dụng số hiệu cổng
 - Mỗi ứng dụng được gán một cổng
- UDP cũng cung cấp cơ chế gán và quản lý các số hiệu cổng để định danh duy nhất cho các ứng dụng chạy trên một trạm của mạng.
- Socket : Một cặp địa chỉ IP và số hiệu cổng



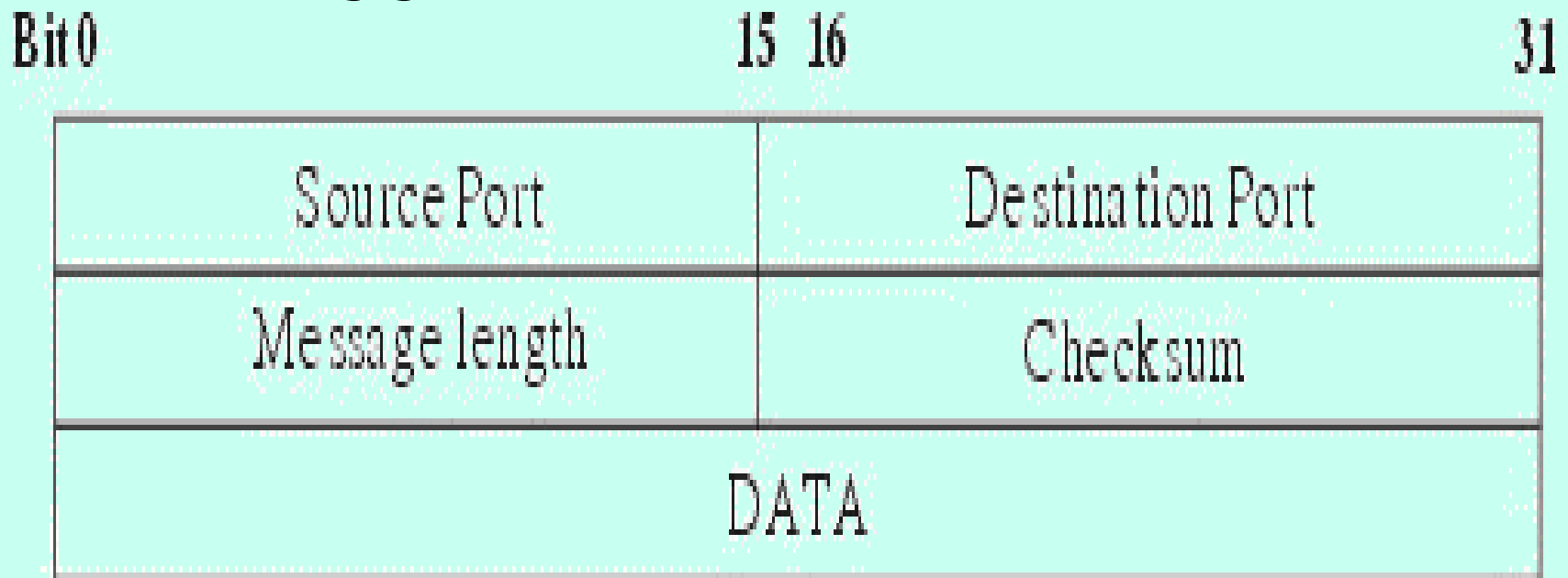
Checksum

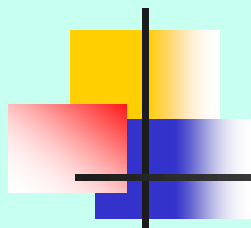
- Phát hiện lỗi bit trong các gói tin/đoạn tin
- Giống checksum của giao thức IP



Khuôn dạng gói tin UDP

- Do ít chức năng phức tạp nên UDP thường có xu thế hoạt động nhanh hơn so với TCP
- Thường được dùng cho các ứng không đòi hỏi độ tin cậy cao trong giao vận





Các vấn đề khác của UDP

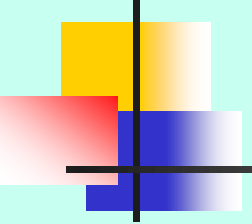
- Không có kiểm soát tắc nghẽn
 - Làm Internet bị quá tải
- Không đảm bảo độ tin cậy
 - Các ứng dụng phải cài đặt cơ chế tự kiểm soát lỗi (ARQ)
- Phát triển ứng dụng sẽ phức tạp hơn



TCP

Transmission Control Protocol

- Tổng quan về TCP
- Khuôn dạng gói tin TCP
- Quản lý liên kết
- Kiểm soát luồng
- Kiểm soát tắc nghẽn



Tổng quan về TCP

- TCP là một giao thức có liên kết
- Cần thiết lập liên kết (logic), giữa một cặp thực thể TCP trước khi chúng trao đổi dữ liệu với nhau.
- TCP cung cấp khả năng truyền dữ liệu một cách an toàn giữa các máy trạm trong hệ thống các mạng.
- Cung cấp thêm các chức năng nhằm kiểm tra tính chính xác của dữ liệu khi đến và bao gồm cả việc gửi lại dữ liệu khi có lỗi xảy ra



Tổng quan về TCP

- TCP cung cấp các chức năng chính sau:
 - Thiết lập, duy trì, kết thúc liên kết giữa hai quá trình.
 - Phân phát gói tin một cách tin cậy.
 - Đánh số thứ tự (sequencing) các gói dữ liệu nhằm truyền dữ liệu một cách tin cậy.
 - Cho phép điều khiển lỗi.
 - Cung cấp khả năng đa kết nối với các quá trình khác nhau giữa trạm nguồn và trạm đích nhất định thông qua việc sử dụng các cổng.
 - Truyền dữ liệu sử dụng cơ chế song công (full-duplex).

| | | | | | | | | |
|-----------------------|----------|------------------|-------------|----------------|-------------|-------------|-------------|--------|
| Source Port | | Destination Port | | | | | | |
| Sequence Number | | | | | | | | |
| Acknowledgment Number | | | | | | | | |
| Data Offset | Reserved | U G R | A C K | P S H | R S T | S I N | F I N | Window |
| Checksum | | | | Urgent Pointer | | | | |
| Options | | | | Padding | | | | |
| TCP data | | | | | | | | |



Khuôn dạng gói tin TCP

- Source Port (16 bits): Số hiệu cổng TCP của trạm nguồn.
- Destination Port (16 bit): Số hiệu cổng TCP của trạm đích.
- Sequence Number (32 bit): số hiệu của byte đầu tiên của segment trừ khi bit SYN được thiết lập. Nếu bit SYN được thiết lập thì Sequence Number là số hiệu tuần tự khởi đầu (ISN) và byte dữ liệu đầu tiên là ISN+1.
- Acknowledgment Number (32 bit): số hiệu của segment tiếp theo mà trạm nguồn đang chờ để nhận. Ngầm ý báo nhận tốt (các) segment mà trạm đích đã gửi cho trạm nguồn.



Khuôn dạng gói tin TCP

- Data offset (4 bit): số lượng bội của 32 bit (32 bit words) trong TCP header (tham số này chỉ ra vị trí bắt đầu của nguồn dữ liệu).
- Reserved (6 bit): dành để dùng trong tương lai
- Control bit (các bit điều khiển):
- URG: Vùng con trỏ khẩn (Urgent Pointer) có hiệu lực.
- ACK: Vùng báo nhận (ACK number) có hiệu lực.
- PSH: Chức năng PUSH.



Khuôn dạng gói tin TCP

- RST: Khởi động lại (reset) liên kết.
- SYN: Đồng bộ hóa số hiệu tuần tự (sequence number).
- FIN: Không còn dữ liệu từ trạm nguồn.
- Window (16 bit): cấp phát credit để kiểm soát nguồn dữ liệu (cơ chế cửa sổ). Đây chính là số lượng các byte dữ liệu, bắt đầu từ byte được chỉ ra trong vùng ACK number, mà trạm nguồn đã sẵn sàng để nhận.
- Checksum (16 bit): mã kiểm soát lỗi cho toàn bộ segment (header + data)



Khuôn dạng gói tin TCP

- Urgent Pointer (16 bit): con trỏ này trỏ tới số hiệu tuần tự của byte đi theo sau dữ liệu khẩn. Vùng này chỉ có hiệu lực khi bit URG được thiết lập.
- Options (độ dài thay đổi): khai báo các option của TCP, trong đó có độ dài tối đa của vùng TCP data trong một segment.
- Padding (độ dài thay đổi): phần chèn thêm vào header để đảm bảo phần header luôn kết thúc ở một mốc 32 bit. Phần thêm này gồm toàn số 0.
- TCP data (độ dài thay đổi): chứa dữ liệu của tầng trên, có độ dài tối đa ngầm định là 536 byte. Giá trị này có thể điều chỉnh bằng cách khai báo trong vùng options.

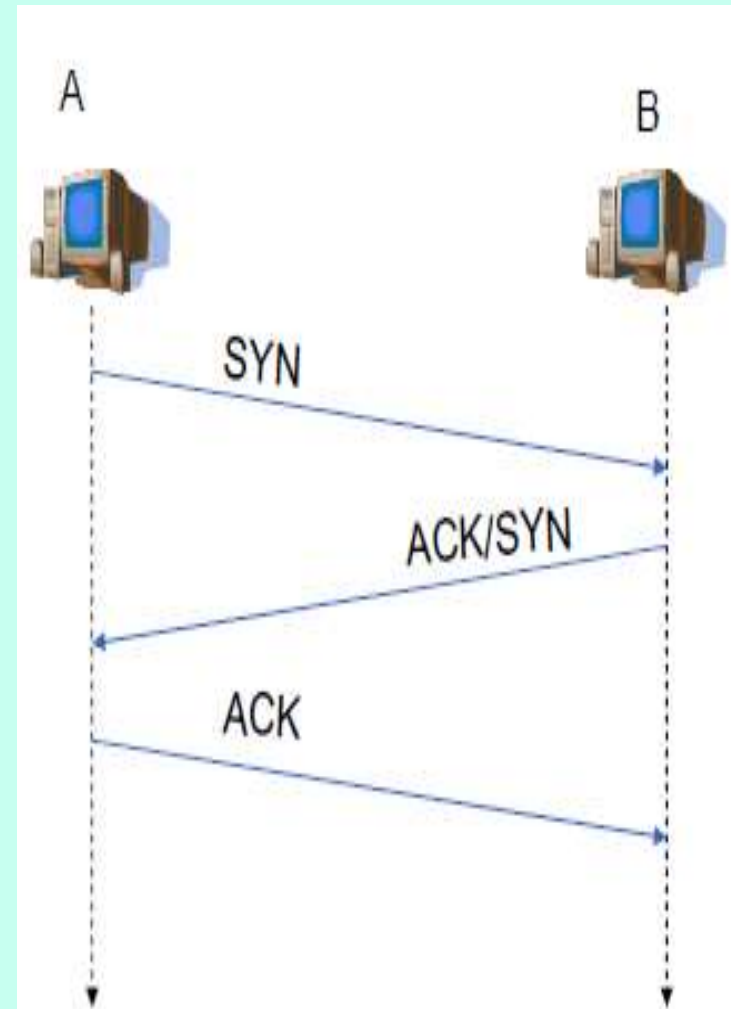


Quản lý liên kết

- Chu trình làm việc của TCP:
 - Thiết lập liên kết
 - Bắt tay ba bước
 - Truyền/nhận dữ liệu
 - Đóng liên kết

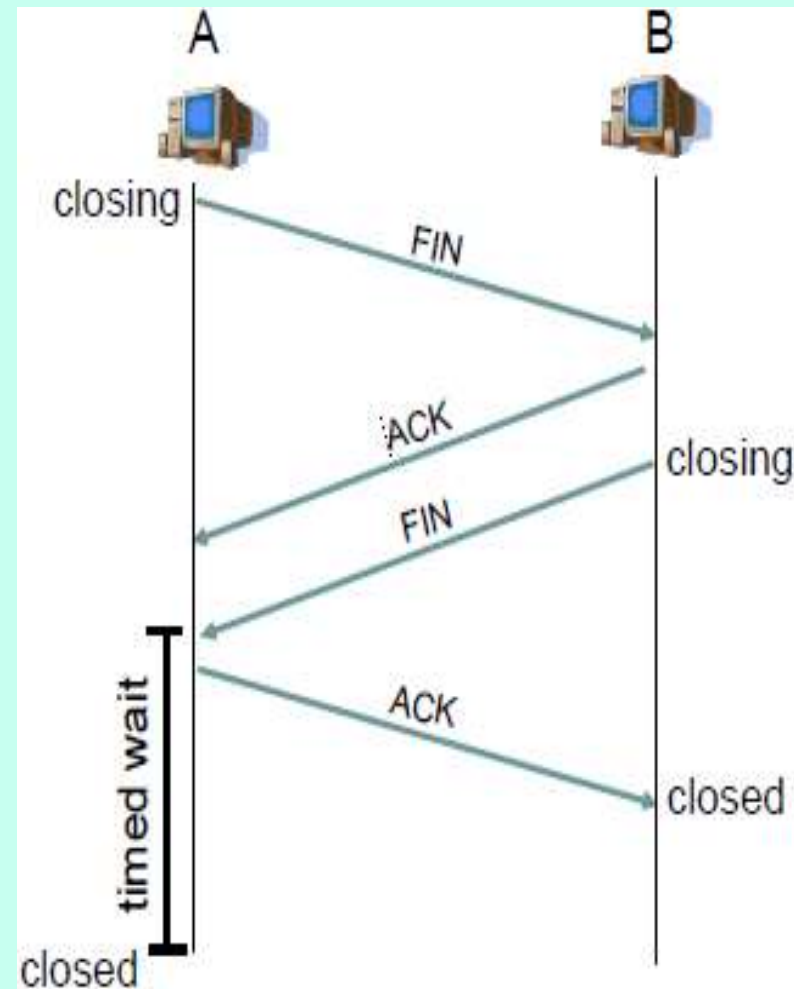
Thiết lập liên kết

- **Bước 1:** A gửi SYN cho B
 - Chỉ ra giá trị khởi tạo seq # của A
 - không có dữ liệu
- **Bước 2:** B nhận SYN, trả lời bằng SYNACK
 - B khởi tạo vùng đệm
 - Chỉ ra giá tr khởi tạo seq. # của B
- **Bước 3:** A nhận SYNACK, trả lời ACK, có thể kèm theo dữ liệu

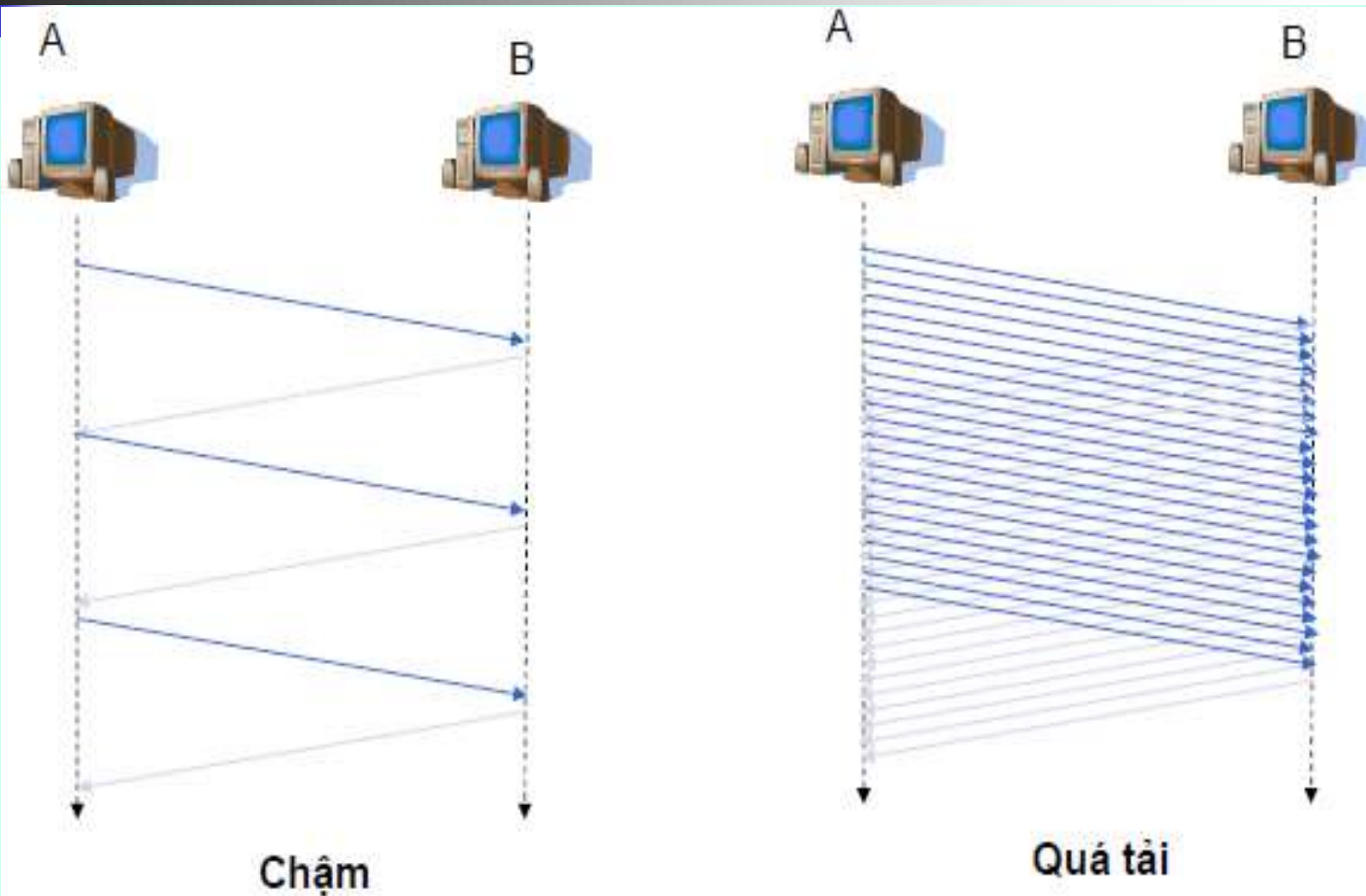


Đóng liên kết

- Bước 1: Gửi FIN cho B
- Bước 2: B nhận được FIN, trả lời ACK, đồng thời đóng liên kết và gửi FIN
- Bước 3: A nhận FIN, trả lời ACK, vào trạng thái chờ
- Bước 4 : B nhận ACK, đóng liên kết



Kiểm soát luồng





Kiểm soát luồng

- Điều khiển lượng dữ liệu được gửi đi
 - Bảo đảm hiệu quả là tốt
 - Không làm quá tải các bên
- Các bên sẽ có cửa sổ kiểm soát
 - Rwnd : cửa sổ nhận
 - Cwnd : cửa sổ kiểm soát tắc nghẽn
- Lượng dữ liệu gửi đi phải nhỏ hơn $\min(\text{rwnd}, \text{cwnd})$



Kiểm soát tắc nghẽn

- Khi nào tắc nghẽn xảy ra :
 - Quá nhiều cặp gửi nhận trên mạng
 - Truyền quá nhiều làm mạng quá tải
- Hậu quả của nghẽn mạng:
 - Mất gói tin
 - Thông lượng giảm, độ trễ tăng
 - Mạng sẽ trở nên tồi tệ hơn



Kiểm soát tắc nghẽn

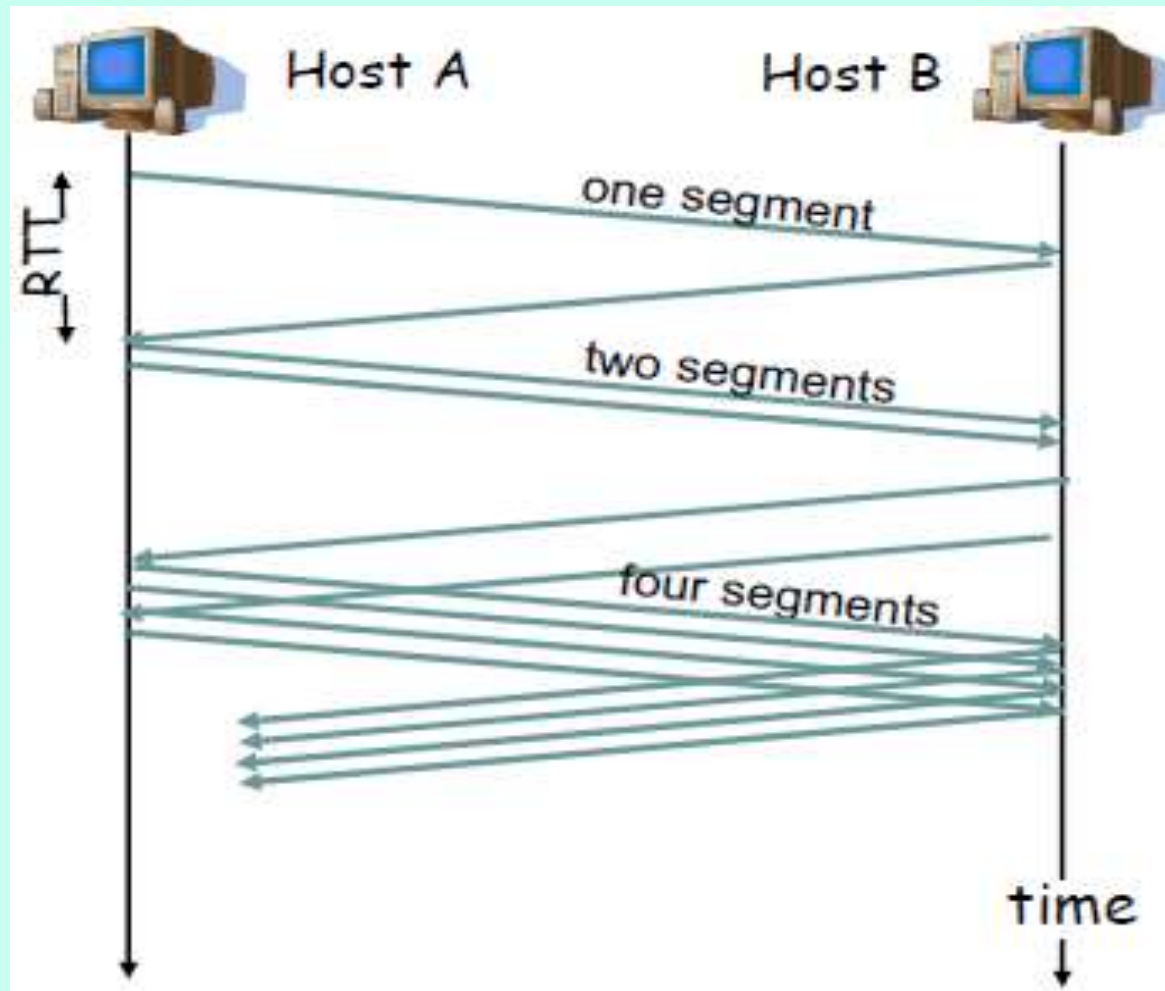
- Nguyên lý kiểm soát tắc nghẽn :
 - Slow-start
 - Tăng tốc độ theo hàm số mũ
 - Tiếp tục tăng đến một ngưỡng nào đó
 - Tránh tắc nghẽn
 - Tăng dần tốc độ theo hàm tuyến tính cho đến khi phát hiện tắc nghẽn



Slow Start

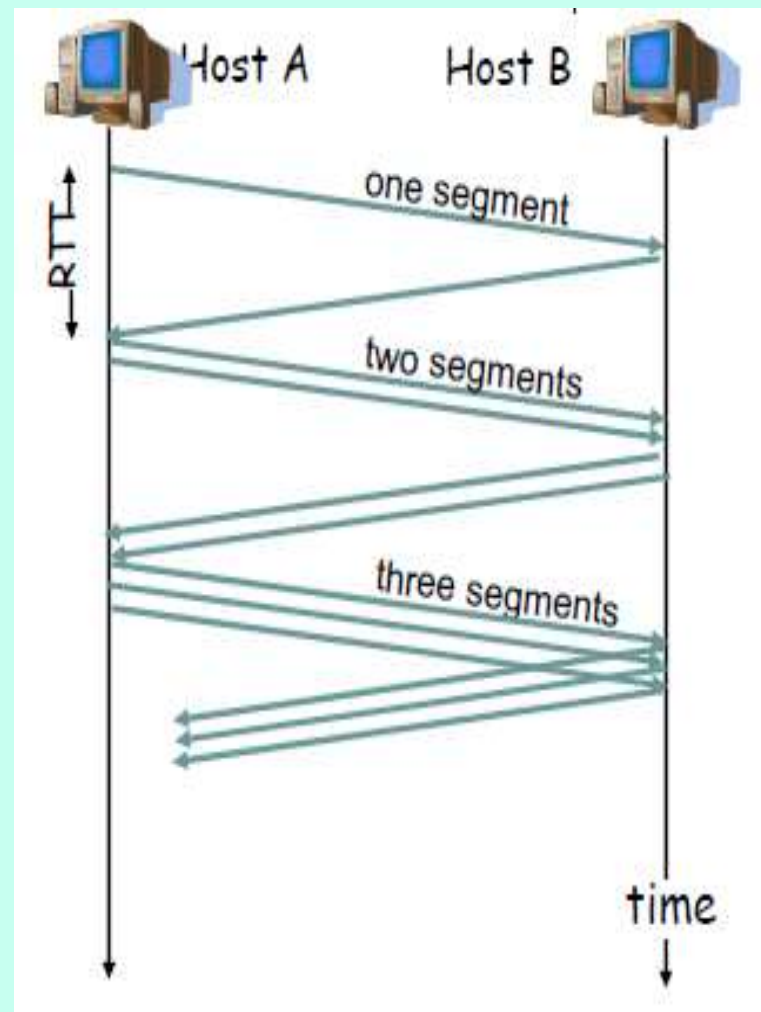
- Đặt cwnd bằng 1 MSS(Maximum segment size)
- Tăng cwnd lên gấp đôi khi nhận được ACK
- Bắt đầu chậm nhưng tăng theo hàm mũ
- Tăng cho đến một ngưỡng : ssthresh
 - Sau đó TCP chuyển sang trạng thái tránh tắc nghẽn

Slow Start



Tránh tắc nghẽn

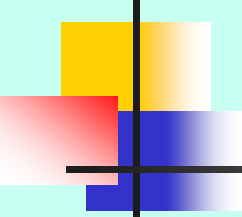
- Tăng cwnd theo cấp số cộng sau khi nó đạt đến ssthresh
- Khi bên gửi nhận được ACK
 - Tăng cwnd thêm 1 MSS





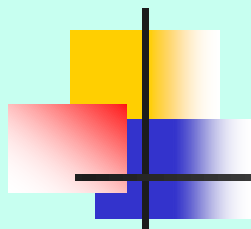
Nhắc lại kiến trúc phân tầng

| | |
|---|--|
| Application (HTTP, Mail, ...) | Hỗ trợ các ứng dụng trên mạng |
| Transport (UDP, TCP ...) | Truyền dữ liệu giữa các ứng dụng |
| Network (IP, ICMP...) | Chọn đường và chuyển tiếp gói tin giữa các máy, các mạng |
| Datalink (Ethernet, ADSL....) | Hỗ trợ việc truyền thông cho các thành phần kế tiếp trên cùng 1 mạng |
| Physical (bits...) | Truyền và nhận dòng bit trên đường truyền vật lý |



Mối quan hệ giữa mô hình OSI và TCP/IP

| OSI | TCP/IP | |
|---|--------------------------|------------------|
| APPLICATION PRESENTATION SESSION | APPLICATION | Telnet, FTP, DNS |
| TRANSPORT | TRANSPORT | TCP, UDP |
| NETWORK | INTERNET | IP |
| DATA LINK PHYSICAL | NETWORK INTERFACE | WANs, TokenRing |



KẾT THÚC BÀI HỌC