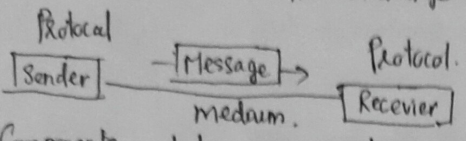
# chapter 1: Introduction.

## 1> Data Communications:

- Telecommunication: communication at a distance
- Data: Information presented in whatever form is agreed upon by the parties creating and using data
- Data communication: the exchange of data between 2 devices via some form of transmission medium



Components of data communication system.

Message — the information (data) to be communicated.
text, numbers, pictures, audio, ...

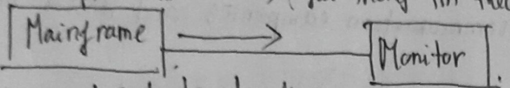
Sender — the devices sends the data message.
computer, workstation, telephone handset, video camera

Recevier — the devices receives the message
computer, workstation, telephone handset, television.

Transmission Medium — the physical path by which message travels from send to rec.
twisted-pair wire, coaxial cable, fiber-optic cable
radio-waves

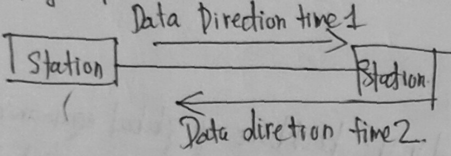
Protocol — A set of rules that govern data communications
Represents an agreement between the communicating devic
Without protocol, 2 devices may be connected but not communicating

- Data flow (simplex, half-duplex, full-duplex)
  + Simplex (Đơn chiều) - gửi thông tin theo 1 hg



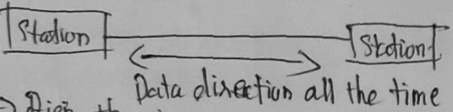
→ TV and radio broadcasting.

+ Half-duplex (2 chiều - mỗi lần theo 1 hướng).



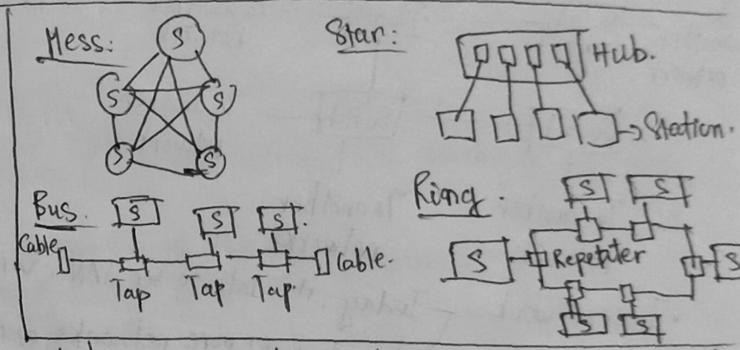
→ Bộ đàm

+ Full-duplex (2 chiều - đồng thời theo 2 hg)



→ Điện thoại

Hiệu suất: simplex < Half-duplex < Full-duplex).



## 2> Networks:

- Network: A set of devices (often referred to as nodes) connected by communication links.
  + Node: computer, printer or any other devices capable of sending and/or receiving data generated by other nodes
  + Link: a cable, air, optical fiber, any medoum can transport a signal carrying data.

- Network Criteria
  - Performance <Hiệu năng> — Depends on Network Elements.
    Measured in terms of Delay <Độ trễ> and Throughput <Thông lg>.
  - Reliability — Ngoài độ chính xác, độ tin cậy của mạng còn đc đo bằng tần suất lỗi thời gian 1 link khôi phục sau lỗi và độ mạnh của mạng
  - Security <Tính bảo mật> — Data protection against corruption / loss of data due to
    - Error
    - Malicious users

Physical Structures.

+ Type of Connection:
  - point to point
  - Multipoint.

+ Physical Topology (Cấu trúc liên kết)
  - Connection of devices
  - Type of transmission

| | Send | receiver |
|---|---|---|
| Unicast | 1-1 | |
| multicast. | 1- ~~đt~~ tập hợp các đ' khác (n k° phải tất cả) | |
| broadcast | 1-all | |

Mesh
↓
n node (devices)
⇒ $\frac{n(n-1)}{2}$ link <cable>
Secure

Star robust
⇒ n cable

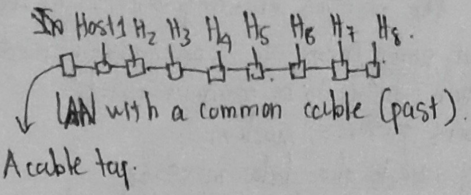
Bus
⇒ n droplines
one backbone
easy installation.

Ring.
⇒ n-1 Cables.
easy falult isolation.

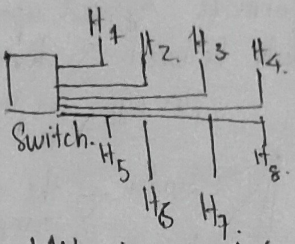
- Categories of Networks.

+ Local Area Networks (LANs).
  - Short Distances
  - Designed to provide local interconnectivity

private owened and connects. some host in a sigle office, building, campus
Each host in LAN has an identifier, an address
LANS today are connecled to each other and to WANs to create communica at a wider level.



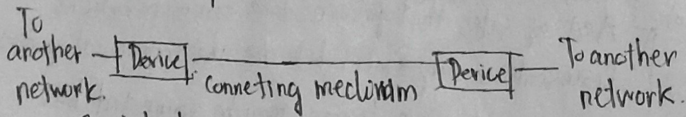
LAN with a common cable (past).



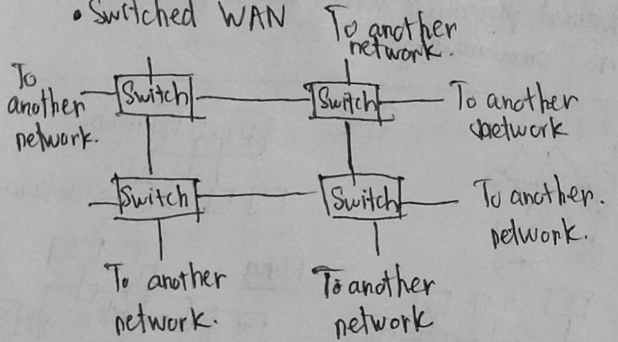
LAN with. a switch (today).

+ Wide Area Network (WANs)
  - has a wider geographical span; town, state, country, even the world.
  - interconnects. connecting devices: switchers, routers, moderms.
  - normally created and run by communicatron campanies and leased by organization use it.

- Point-to-point WAN.

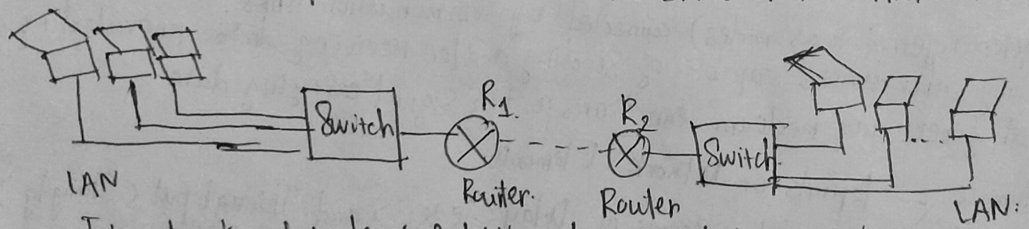


- Switched WAN



A switched WAN is used in the backbone of global communication.
is a combination of many point-to-point WANs. Connected by switchers

- Internetwork.
  - Today, it is rare to see LAN or WAN in isolation, they are connected to one another.
  - when two or more networks are connected ⇒ internetwork.

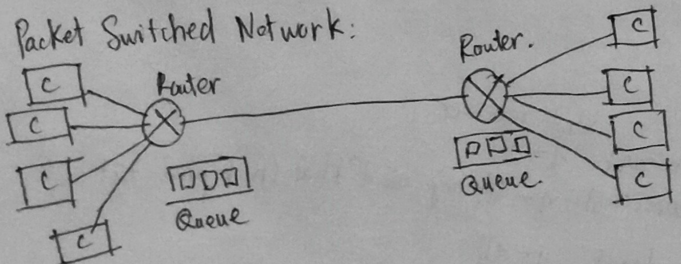


Internetwork ~~not~~ made of 2 LANS and one point to point WAN.

- Switching: An internet is a switched network in which a switch connects at least 2 link together.
A switch needs to forward data from a network to another network when required.

+ ~~8.~~ Circuit Switched Network : always available between the two end system; can only make it active / inactive

+ Packet Switched Network:



3> The Internet.

— Is a communication system that has brought a wealth of information to our fingertips and organized it for our use

4> The protocols

It, Is synonymous with rule.
consists of a set of rules govern data communication.

+ Elements of a protocol
  - Syntax (cú pháp) — Struce or format of data; Indicates how to read bits
  - Semantic (ngữ nghĩa) — Interprets. the meaning of bits. knows the fields. define what action.
  - Timing

PHYSICAL LAYER

Chapter: 2.1 Data and Signal.

To be trans mitted, data must be transformed to electromagentic signals.

1> Analog and Digital:

2> Periodic analog signals.

- If a signal does not change at all, its frequency is zero
- If a signal changes instantaneously, its frequency is infinite.
- Phase describes the position of the waveform relative to time 0.
- Nếu tín hiệu t² tuần hoàn thì tín hiệu miền tần số rời rạc

3> Bandwidth:

The bandwidth of a composite signal is the difference (hiệu) between the highest and the lowest frequency

4> Digital signal.

- The number of bits per level = $\log_2 n$
  n : number of levels.
- Bit Rate:
  - Is used to describe digital signals
  - is the number of bits send in 1s. (bps)
- Bit length = is the distance one bit occupies on the transmission medium.
  = propagation speed × bit duration.

- A digital signal is a composite analog signal with an infinite bandwith.
- Base band transmission.: sending a digital signal over a channel without changing the digital signal to analog signal.
  + Requires a low pass channel (a channel with bandwith that starts from zero)
  + The required bandwidth is proportional to the bit rate
  + If we need send bits faster, need more bandwidth.
  + If the available channel is a BP channel, we cannot send digital signal directly to the channel; we need convert the digital signal to analog signal before transmission.
- Transmission impairment:
  + The imperfection causes signal impairment. => the signal at the beginning of the medium is not same as the end of medium.
  + There are 3 causes of impairment: attenuation (suy giảm); Distortion (Biến dạng); Noise (Nhiễu)
  - Attenuation: to show the loss or gain of energy the unit "decibel"

    $$dB = 10 \log_{10} \frac{P_2}{P_1}$$

    $P_2$: output signal
    $P_1$: input signal
  - Distortion:
  - Noise: impulse noise is a spike (a signal with high energy in a very short time) that comes from power lines, lighting.
- Signal to Noise Ratio (SNR)
  + To measure the quality of a system the SNR is used.
  + Indicates the strength of the signal wrt the noise power in the system.

$$SNR = \frac{\text{average signal power}}{\text{average noise power}}$$

# Chapter 2: Network Models.

1> Layered tasks

We use the concept of layers in our daily life.

2> The OSI Model.

- ISC is organization.
  - OSI is model.
- Seven layers of the OSI model.

Application (7) ← Presentation (6) ← Session (5) ← Transport (4) ← Network (3) ← Data link (2) ← Physical (1).

+ Physical layer:
  - Trao đổi các bit 1 cách riêng rẽ từ thiết bị này → thiết bị bên cạnh
  - Liên quan đến giao diện môi trg — mã hoá tốc độ dữ liệu
  - Để chuyển các bit → tín hiệu điện / quang
  - ⇒ Chuyển mỗi bit từ hop này → hop bên cạnh

+ Data link layer:
  - ⇒ chọn the best links.
  - chuyển frames từ hop (node) này → bên cạnh

+ Network layer:
  - ⇒ chuyển từng packets từ host nguồn tới host đích (đ/chỉ VL k'đổi)

+ Transport layer:
  - ⇒ chuyển 1 message từ 1 quá trình ⇒ quá trình còn lại

+ Session layer
  - ⇒ Dialog control and synchoronization. (đồng bộ).

+ Presention
  - ⇒ Translation, compression and encryption (mã hoá).

+ Application layer
  - ⇒ Cung cấp dịch vụ tới ng dùng.

3> TCP/IP Protocol suite.

- The original TCP/IP protocol suite was defined as having 4 layers ← host-to-host / network / Internet / Application
- When TCP/IP compared to OSI, we can say that TCP/IP have 5 layers – physical; Data link, Network, transport and Application / Application
- Addressing:
  - physical: k° thể thay đổi. — Data link / Physical: layer
  - logical: có thể thay đổi: địa chỉ IP — Network layer
  - Port: phân biệt quá tiến trình nào trên máy tính — Transport layer
  - specific: đặc trưng — Application layer.

VD: Most local-are network use a 48 bit (6 byte) physical address written as 12 hexadecimal digits

07:01:02:01:2C:4B: (a 6 byte physial address).

+ 123.17.149.185 < logical address)

+ D2:35:75:AA:7A (physical)

+ 192.168.1.67:80 (Port address)

+ 763 < A 16-bit port address represented as one sigle number)

- SNR high → the signal is less corrupted by noise.
- SNR low → ——— more ———
- $SNR_{dB} = 10 \log_{10} SNR$

5, Data rate limits ← the bandwidth available.
the level of the signal
the quality of channel.

- Noiseless channel: Nyquist Bit Rate

$BitRate = 2 \times Bandwith \times \boxed{\log_2 L}$ → số bit truyền đi trong 1 lần phát (1s).

L: the number of signal levels use to represent data.

⇒ Increasing the levels of signal may reduce the reliability of system.

- Shannon Capacity.

Determine the theretical highest dater rate for a noisy channal.

Capacity = highest date rate = Bandwidth × $\log_2(1+ SNR)$.

$\underset{\text{(Dung lg)}}{Capacity}$ ⇒ $C = B \log_2 (1+SNR)$.

- Using both limits.

⇒ The shannon capacity gives us the upper limit;
The Nyquist formala tells us how many signals levels we need.

6, Performance:

2 context
- Bandwidth in Hertz ⇒ the range of frequencies in a composite signal
the range of frequencies. that a channel can pass.
- Bandwidth in bits per second: ⇒ the speed of bit. transmission in a channel or link.

- Throughput (thông lg).
- Latency (Delay) = propagation time + transmission time + queuing time + processing delay.
(tg xếp hàng).

$$\text{propagation time} = \frac{\text{Distance}}{\text{Propagation Speed}} + \frac{\text{Message size}}{\text{Bandwidth}} = \text{transmission time}$$

the bandwidth - delay product defines the number of bits that can fill the link.

## Chapter 2.2: Digital Transmission.

22.1 Digital to digital conversion: ← line coding (needed)
Block coding (may or may not be needed).
Scrambing.

- line coding:

→ Coverting digital data to digital signals.

→ Mapping Data symbols onto signal levels:

- A data symbol (or element) can consist of a number of data bits: 1,0 or 11, 10, 01, ...
- A data symbol can be coded into a sigle signal element or multipe signal elements: { 1 → +V ; 0 → -V. / 1 → +V and -V. / 0 → -V and +V.
- The ratio "r" is the number of data elements carried by a signal elements.

→ Relationship between data rate and signal rate.

the number of bits sent per sec/bps.
often referred to the bit rate.

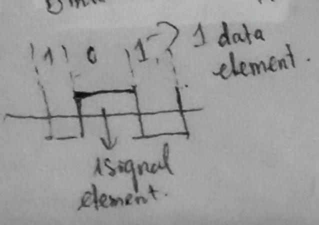
The number of signal elements sent in a second and be measured in bauds
often referred as the modulation rate, pulse rate, baud rate.

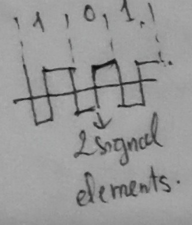
→ Goal is to increase the data rate whilst reducing the $\boxed{\text{baud rate}}$ → $S = \frac{N}{r} = c \times N \times \frac{1}{r}$

{ N: data rate.
c: case factor.
r: ratio data element / sig element.

⚠ the actual bandwidth of a digital signal is infinite.
the effective bandwidth is finite.

$B_{min} = c \times N \times \frac{1}{r}$   $N_{max} = \frac{1}{c} \times B \times r.$





- Considerations for choosing a good signal element referred to as line coding.
+ Baseline wandering: - ~~rece~~
+ line coding o/cs:

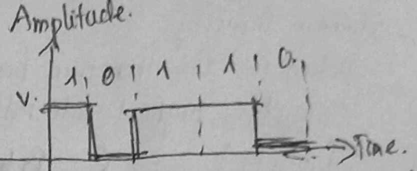
DC components: - when voltagelevel remains constant for along time, the low frequencies of signal is increase.
Most channels are bandpass and may not support the low frequencies.
-> Require the removal of the DC component of a transmission signal.

Self sychoronization: the clocks at the sender and the receiver must have the same bit interval.
if the receiver clock is faster and slower, it will mis interpret the incoming bit stream.

Built-in Error Detection:

- Line Coding Schemes: (lược đồ mã hoá dòng)

(+) Unipolar scheme - All the signal levels are on one side of the time axis, (NRZ)
- NRZ (Non-Return-to Zero).
The positive voltage defines. bit 1 and the zero votlage defines bit 0.
Signal does not return to zero at the middle of the bit.
=> costly.

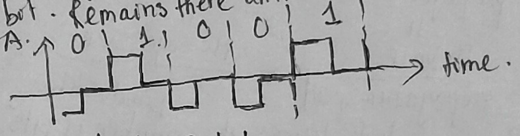


(+) Polar Schemes: NRZ, RZ, Biphase (Manchester, differential Manchester): the voltages are on the both sides of the time axis.
- NRZ: (Non-Return-to Zero)
- Use 2 levels of voltage amplitude.
NRZ-L: the level of the voltage determines. the value of bit.
NRZ-I. the change or lack of change in the level of the voltage determine value of bit.
If no change, bit is 0.
If change, bit is 1.

NRZ-L and NRZ-I both have an average signal rate of $\frac{N}{2}$ Bd.

NRZ-L and NRZ-I both have a DC component problem.

- RZ (Return to Zero):
Use 3 values: (+), (-), (0).
The signal changes not between. bits but during the bit.
The signal go to 0 in the middle. of each bit. Remains there until the beginning of the next bit.



DisAdvantages: greater band width.
complexity.

- Polar biphase:
Manchester encoding: (RZ+ NRZ-L)
the duration of bit is. divied into 2 halves:
the voltage remains at one level during the first half. and move to other level in. the second half.
the transition at the middle of the bit provides synchronization.

Differential Manchester. (RZ+ NRZ-I)
the transition at the middle of the bit.
the bit values are determined at the begining of the bit.
-> No base line wandering, no DC compoent } but signal rate is problem.
=> The minium bandwidth of Man.. and diff. Man is 2 time that of NRZ.

(+) Bipolar schemes: use 3 levels: (+), (0), (-).
the voltage level for one data element in at zero.
other element alternates between (+), (-)

- AMI and Pseudoternary: No DC component
A sequence that creates a constant zero voltage does not have a DC component.
is commonly used for long-distance communication, the problem is synchorozation

(+) Multilevel Schemes.
In mBnL schemes, a pattern of m data elements is encoded as a pattern of n signal elements: $2^m \leq L^n$.
the desire to increase the data rate or decrease the required bandwidth has resulted in creation many shemes
the goal is to increase the number of bits per baud by encoding a pattern of m data elements.
Q1: $2^m$ data patterms.
$L^n$: signal patterts.

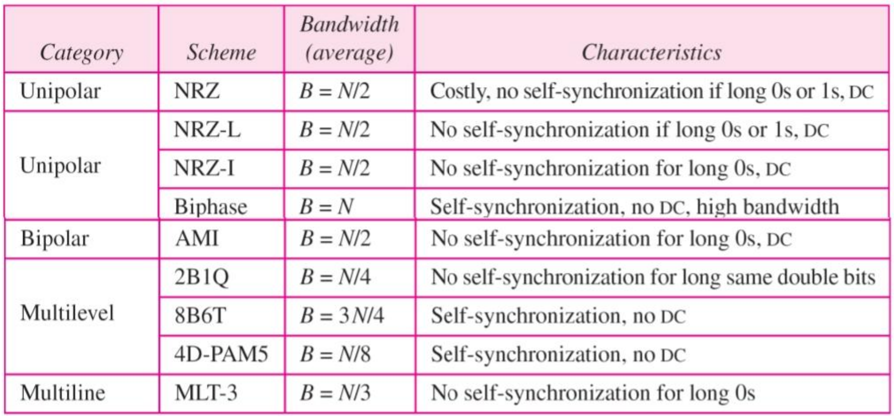
2B1Q. (2binary, one quaternary); use data pattens of size 2
encodes the 2-bit patterns.
m=2; n=1 => L=4. => $S = \frac{N}{4}$

8B6T: (8 binary, 6 tenary):

4D-PAM5: (four-dimensional five-level pulse amptitude modulation

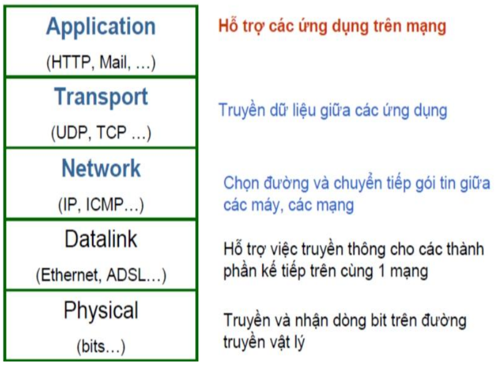
2B1Q; 8B6T; 4D-PAM5; ~~MLT~~-3.

(+) Multitransition: MLT-3. (multiline transmission, three-level)
use 3 levels: +V; 0; -V
If the next bit is 0, there is no trasition.
If ______ is 1, the current level is not 0, the next level: 0.
______, the current is 0 the next level is the opperate of the last nonzero level



| Category | Scheme | Bandwidth (average) | Characteristics |
|---|---|---|---|
| Unipolar | NRZ | $B = N/2$ | Costly, no self-synchronization if long 0s or 1s, DC |
| Unipolar | NRZ-L | $B = N/2$ | No self-synchronization if long 0s or 1s, DC |
| | NRZ-I | $B = N/2$ | No self-synchronization for long 0s, DC |
| | Biphase | $B = N$ | Self-synchronization, no DC, high bandwidth |
| Bipolar | AMI | $B = N/2$ | No self-synchronization for long 0s, DC |
| Multilevel | 2B1Q | $B = N/4$ | No self-synchronization for long same double bits |
| | 8B6T | $B = 3N/4$ | Self-synchronization, no DC |
| | 4D-PAM5 | $B = N/8$ | Self-synchronization, no DC |
| Multiline | MLT-3 | $B = N/3$ | No self-synchronization for long 0s |

❑Hệ giao thức TCP/IP


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

**Block coding** is normally referred to as mB/nB coding; it replaces each m-bit group with an n-bit group

**Figure 2.2.15** *Using block coding 4B/5B with NRZ-I line coding scheme*

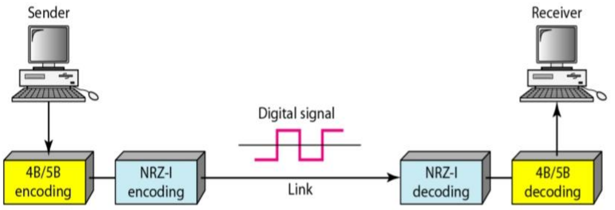
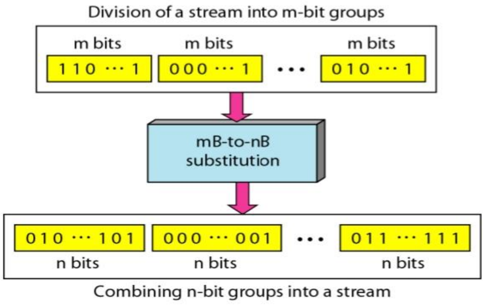


**Figure 2.2.14** *Block coding concept*



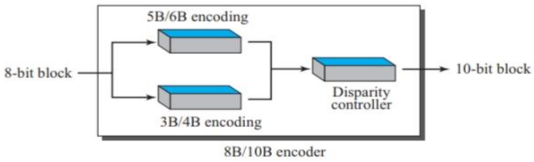
**4B/5B**

The four binary/five binary (4B/5B) coding scheme was designed to be used in combination with NRZ-. In 4B/5B, the 5-bit output that replaces the 4-bit input has no more than one leading zero (left bit) and no more than two trailing zeros (right bits).

**8B/10B**

The eight binary/ten binary (8B/10B) encoding is similar to 4B/5B encoding except that a group of 8 bits of data is now substituted by a 10-bit code. It provides greater error detection capability than 4B/5B. The 8B/10B block coding is actually a combination of 5B/6B and 3B/4B encoding. The five most significant bits of a 10-bit block are fed into the 5B/6B encoder; the three least significant bits are fed into a 3B/4B encoder. The coding has 210 − 28 = 768 redundant groups that can be used for disparity checking and error detection.

Figure 4.17 *8B/10B block encoding*



**B8ZS:** is commonly used in North America. In this technique, eight consecutive zero-level voltages are replaced by the sequence 000VB0V substitutes

**Figure 2.2.20** *Different situations in HDB3 scrambling technique*

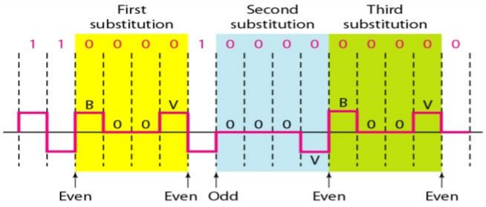
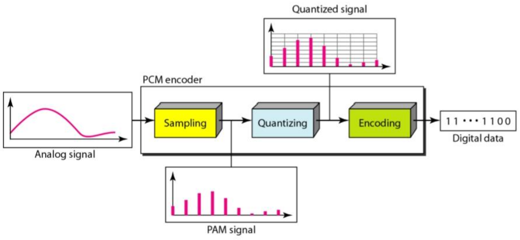


**Figure 2.2.21** *Components of PCM encoder*



1. If the number of nonzero pulses after the last substitution is odd, the substitution pattern will be 000V, which makes the total number of nonzero pulses even.
2. If the number of nonzero pulses after the last substitution is even, the substitution pattern will be B00V, which makes the total number of nonzero pulses even

**HDB3** substitutes four consecutive zeros with 000V or B00V depending on the number of nonzero pulses after the last substitution.

HDB3 is commonly used outside of North America. In this technique, which is more conservative than B8ZS, four consecutive zero-level voltages are replaced with a sequence of 000V or B00V.

### 2.2-2 Analog-to-digital Conversion

### PCM

PCM consists of three steps to digitize an analog signal:

1. Sampling: Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal
2. Quantization: Sampling results in a series of pulses of varying amplitude values ranging between two limits: a min and a max. The amplitude values are infinite between the two limits. We need to map the infinite amplitude values onto a finite set of known values. This is achieved by dividing the distance between min and max into L zones, each of height = (Vmax - Vmin)/L
3. Binary encoding

### 2.2-3 Transmission Modes

The transmission of binary data across a link can be accomplished in either parallel or serial mode. In parallel mode, multiple bits are sent with each clock tick. In serial mode, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous.

**Figure 2.2.32** *Parallel transmission*

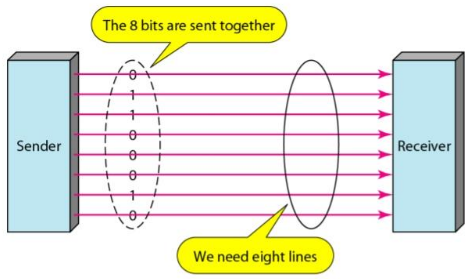
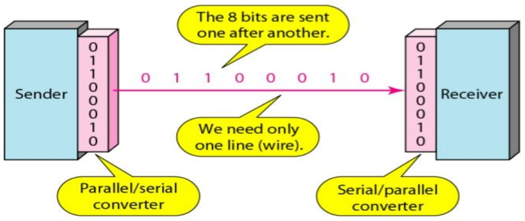


**Figure 2.2.33** *Serial transmission*


The 8 bits are sent one after another.
Sender
0 1 1 0 0 0 1 0
Receiver
We need only one line (wire).
Parallel/serial converter
Serial/parallel converter

- In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1s) at the end of each byte. There may be a gap between each byte.
- Asynchronous here means "asynchronous at the byte level," but the bits are still synchronized; their durations are the same.
- In synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.

**Figure 2.2.34** *Asynchronous transmission*

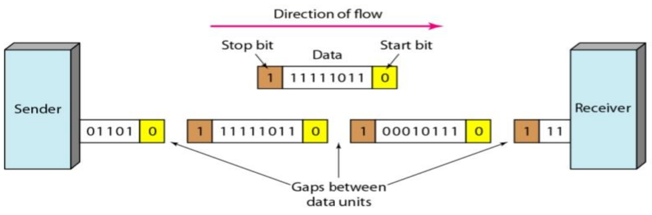
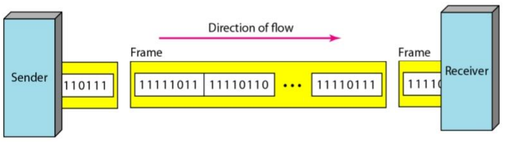


**Figure 2.2.35** *Synchronous transmission*



## Chapter 2.3 Analog Transmission

2.3-1 Digital-to-analog Conversion

Digital data needs to be carried on an analog signal. A carrier signal (frequency fc) performs the function of transporting the digital

data in an analog waveform. The analog carrier signal is manipulated to uniquely identify the digital data being carried.

**Figure 2.3.1** *Digital-to-analog conversion*

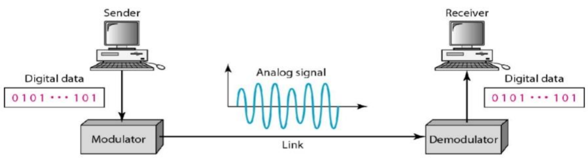
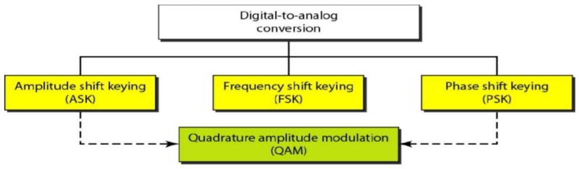


**Figure 5.2** *Types of digital-to-analog conversion*


Digital-to-analog conversion
Amplitude shift keying (ASK)
Frequency shift keying (FSK)
Phase shift keying (PSK)
Quadrature amplitude modulation (QAM)

**Data Rate Versus Signal Rate**

S=N x 1/r band

Bit rate is the number of bits per second. Baud rate is the number of signal elements per second. In the analog transmission of digital data, the baud rate is less than or equal to the bit rate. S=Nx1/r bauds Where r is the number of data bits per signal element

**Amplitude Shift Keying**

In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes

*Binary ASK (BASK)*

ASK is normally implemented using only two levels. The peak amplitude of one signal level is 0; the other is the same as the amplitude of the carrier frequency.

**Figure 5.3** *Binary amplitude shift keying*

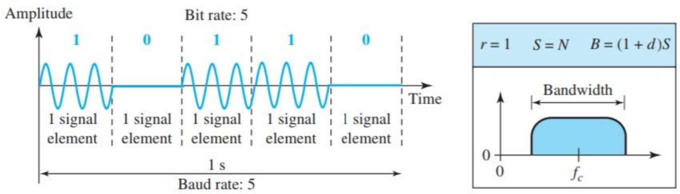
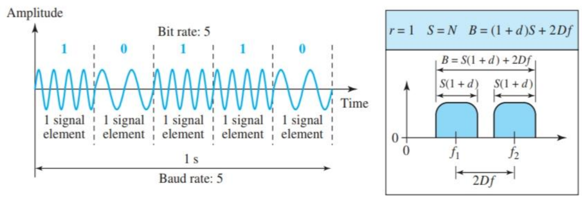


**Figure 5.6** *Binary frequency shift keying*



*Bandwidth for BFSK* **B =(1+d) x S** (S is the signal rate and the B is the bandwidth.) the required bandwidth has a minimum value of S and maximum value of 2S.

**Frequency Shift Keying**

The digital data stream changes the frequency of the carrier signal, fc

For example, a “1” could be represented by f1=fc +Δf; and a “0” could be represented by f2=fc-Δf.

Both *peak amplitude and phase remain constant for all signal elements.*

*Binary FSK (BFSK)*

Binary FSK consider two carrier frequencies. Normally the carrier frequencies are very high, and the difference between them is very small.

Bandwidth for BFSK: If the difference between the two frequencies (f1 and f2) is 2Δf, then the required BW B will be: **B = (1+d)xS +2Δf**

Coherent and Non Coherent

In a non-coherent FSK scheme, when we change from one frequency to the other, we do not adhere to the current phase of the signal. In coherent FSK, the switch from one frequency signal to the other only occurs at the same phase in the signal.

Multi level FSK

Similarly to ASK, FSK can use multiple bits per signal element. That means we need to provision for multiple frequencies, each one to represent a group of data bits. The bandwidth for FSK can be higher B = (1+d)xS + (L-1)/2Δf = LxS

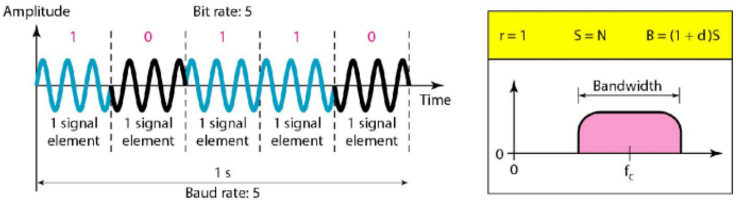
**Phase Shift Keyeing**

We vary the phase shift of the carrier signal to represent digital data.

The bandwidth requirement, B is: B = (1+d)xS

PSK is much more robust than ASK as it is not that vulnerable to noise

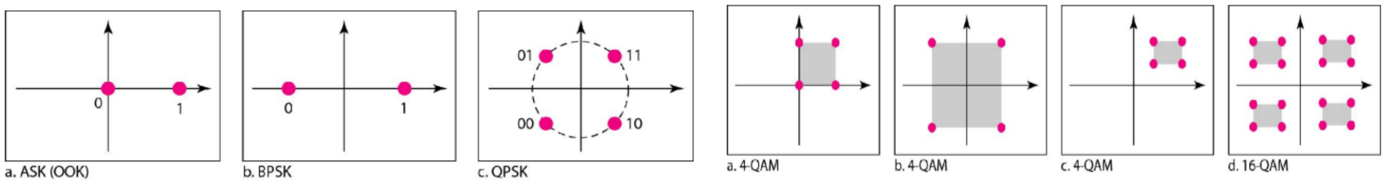
**Figure 2.3.9** ***Binary phase shift keying***



*Quadrature PSK*

To increase the bit rate, we can code 2 or more bits onto one signal element. In QPSK, we parallelize the bit stream so that every two incoming bits are split up and PSK a carrier frequency. One carrier frequency is phase shifted 90o from the other - in quadrature. The two PSKed signals are then added to produce one of 4 signal elements. L = 4 here.

Constellation Diagrams



a. ASK (OOK)

b. BPSK

c. QPSK

a. 4-QAM

b. 4-QAM

c. 4-QAM

d. 16-QAM

**Quadrature Amplitude Modulation(QAM)=ASK+PSK**

Bandwidth for QAM

The minimum bandwidth required for QAM transmission is the same as that required for ASK and PSK transmission. QAM has the same advantages as PSK over ASK

**2.3-2 Analog To Analog Conversion.**

Analog-to-analog conversion is the representation of analog information by an analog signal.

**Amplitude Modulation**

A carrier signal is modulated only in amplitude value. The modulating signal is the envelope of the carrier

*Bandwidth is 2B,* where B is the bandwidth of the modulating signal

Since on both sides of the carrier fc , the spectrum is identical, we can discard one half, thus requiring a smaller bandwidth for transmission.

**Figure 2.3.16** ***Amplitude modulation***

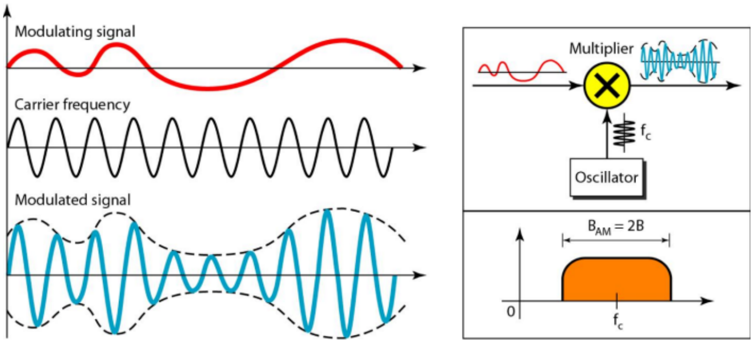
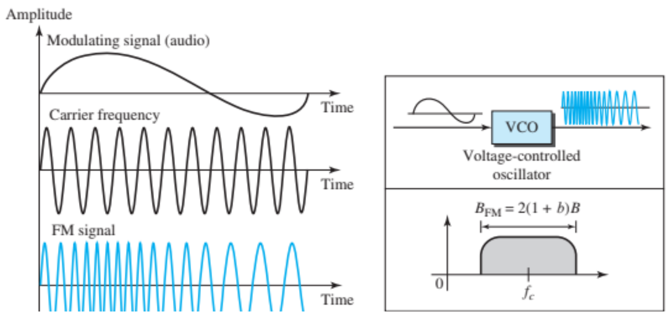


**Figure 5.18** *Frequency modulation*



**Frequency Modulation**

The modulating signal changes the f c of the carrier signal

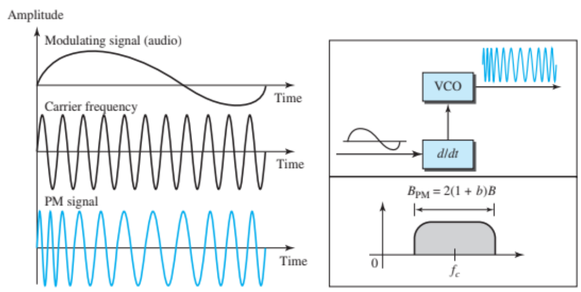
The total bandwidth required for FM can be determined from the bandwidth of the audio signal:

B_FM = 2(1 + β)B. Where  is usually 4.

**Phase Modulation (PM)**

The modulating signal only changes the phase of the carrier signal. The phase change manifests itself as a frequency change but the instantaneous frequency change is proportional to the derivative of the amplitude.  The bandwidth is higher than for AM.

Figure 5.20 *Phase modulation*



The total bandwidth required for PM can be determined from the bandwidth and maximum amplitude of the modulating signal: B_PM = 2(1 + β)B. Where β = 2 most often.

## Chapter 2.4 Bandwidth Utilization: Multiplexing and Spreading

Bandwidth utilization is the wise use of available bandwidth to achieve specific goals. Efficiency can be achieved by multiplexing;  sharing of the bandwidth between multiple users.

### 2.4-1 Multiplexing

Ghép kênh (Multiplexing) là truyền nhiều tín hiệu trên một kênh truyền.

Frequency-Division Multiplexing(FDM)

FDM is an analog multiplexing technique that combines analog signals.

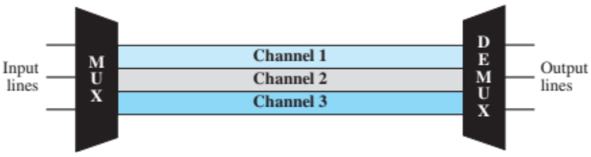
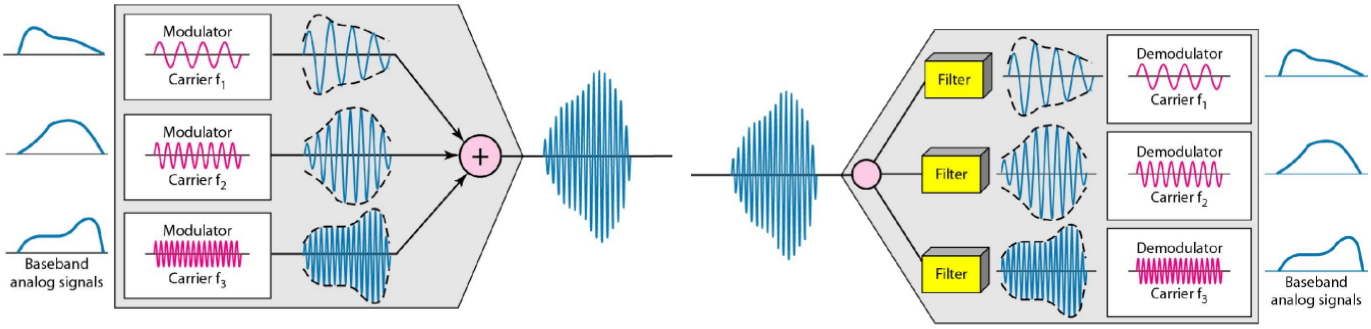


Figure 2.4.4 *FDM process*

Figure 2.4.5 *FDM demultiplexing example*



Example: Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

*Solution* For five channels, we need at least four guard bands. This means that the required bandwidth is at least 5 × 100 + 4 × 10 = 540 kHz

Example 6.3

Four data channels (digital), each transmitting at 1 Mbps, use a satellite channel of 1 MHz. Design an appropriate configuration, using FDM.

**Figure 6.8** *Example 6.3*

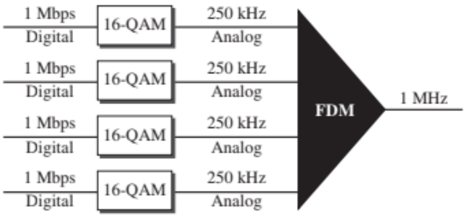
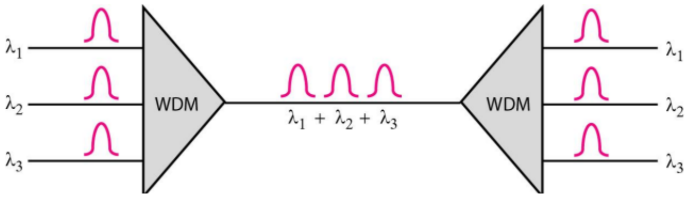


**Figure 2.4.10** ***Wavelength-division multiplexing (WDM)***



Wavelength-division multiplexing (WDM)

WDM is an analog multiplexing technique to combine optical signals. WDM technology is very complex, the basic idea is very simple.

Time-division multiplexing (TDM)

TDM is a digital process that allows several connections to share the high bandwidth of a link. Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link.

Note that the same link is used as in FDM;but, the link is shown sectioned by time rather than by frequency.We can divide TDM into two different schemes: synchronous or statistical.

In synchronous TDM, each input connection has an allotment in the output even if it is not sending data. In statistical TDM, slots are dynamically allocated to improve bandwidth efficiency.

**Figure 2.4.12 Time Division Multiplexing (*TDM*)**

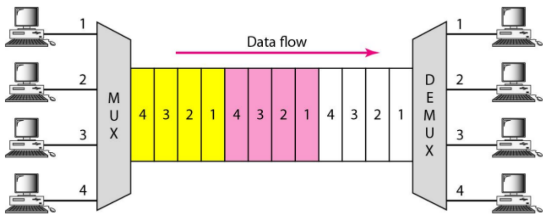
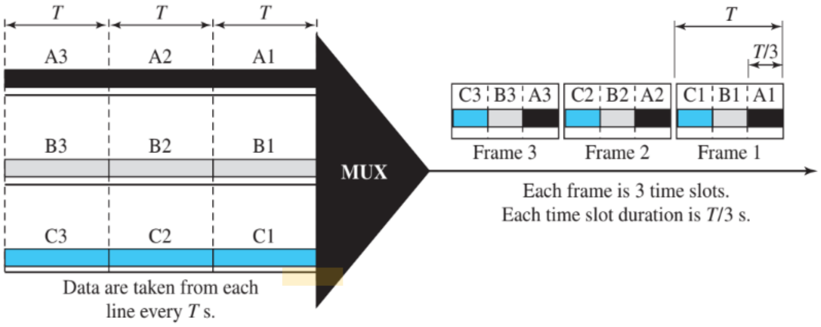


**Figure 6.13** *Synchronous time-division multiplexing*



Each frame is 3 time slots.
Each time slot duration is *T*/3 s.

Data are taken from each
line every *T* s.

TDM is a digital multiplexing technique for combining several low-rate digital channels into one high-rate one.

*Synchronous TDM*

The data rate of the link is n times faster, and the unit duration is n times shorter.

Ex: The data rate for each input connection is 1 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of

1. each input slot,
2. each output slot, and
3. each frame?

Solution

1. The data rate of each input connection is 1 kbps. This means that the bit duration is 1/1000 sor 1 ms. The duration of the input time slot is 1 ms (same as bit duration).
2. The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is 1/3 ms.
3. Each frame carries three output time slots. So the duration of a frame is 3 × 1/3 ms, or 1 ms. The duration of a frame is the same as the duration of an input unit.

*Interleaving*

The process of taking a group of bits from each input line for multiplexing is called interleaving. We interleave bits (1 - n) from each input onto one output.

*Data Rate Management*

Not all input links maybe have the same data rate. Some links maybe slower. There maybe several different input link speeds There are three strategies that can be used to overcome the data rate mismatch: multilevel, multislot and pulse stuffing

*Data rate matching*

Multilevel: used when the data rate of theinput links are multiples of each other.

Multislot: used when there is a GCD betweenthe data rates. The higher bit rate channelsare allocated more slots per frame, and the output frame rate is a multiple of each input link.

Pulse Stuffing: used when there is no GCD between the links. The slowest speed link will be brought up to the speed of the other links by bit insertion, this is called pulse stuffing.

Synchronization

To ensure that the receiver correctly reads the incoming bits,, knows the incoming bit boundaries to interpret a “1” and a “0”, a known bit pattern is used between the frames. The receiver looks for the anticipated bit and starts counting bits till the end of the frame. Then it starts over again with the reception of another known bit. These bits (or bit patterns) are called synchronization bit(s). They are part of the overhead of transmission.

**2.4-2 Spread Spectrum**

In spread spectrum (SS), we combine signals from different sources to fit into a larger bandwidth, but our goals are to prevent eavesdropping and jamming. To achieve these goals, spread spectrum techniques add redundancy.

Spread spectrum is designed to be used in wireless applications in which stations must be able to share the medium without interception by an eavesdrop- per and without being subject to jamming from a malicious intruder. The frequency hopping spread spectrum (FHSS) technique uses M different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency. The direct sequence spread spectrum (DSSS) technique expands the bandwidth of a signal by replacing each data bit with n bits using a spreading code. In other words, each bit is assigned a code of n bits, called chips.

## Chapter 3.1 Error Detection and Correction

### Overview of Data Link layer

Role: transform the physical layer (a raw transmission facility) to a link responsible for node-to-node communications.

Responsibilities:

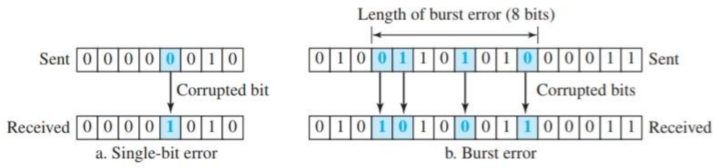
- Framing: divides the bit stream received from the network layer to manageable data units (frames).
- Addressing: adds a header to a frame to define the address of sender and receiver.
- Flow control: to avoid overwhelming (choáng ngợp) receiver when receiving rate is smaller than sending rate.
- Error control: adds mechanisms to detect and retransmit damaged, duplicate and lost frames. Media access control: determines which devices get access to a shared link at a given time the when more than 2 devices are connected with the same link

Data can be corrupted during transmission. Some applications require that errors be detected and corrected.

### Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal. The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1. The term burst error means
that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

**Figure 10.1** *Single-bit and burst error*



To detect or correct errors, we need to send extra (redundant) bits with data.The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver.

**Detection Versus Correction:** The correction of errors is more difficult than the detection.

**Coding**

o Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.

o The receiver checks the relationships between the two sets of bits to detect errors. The ratio of redundant bits to data bits and the robustness of the process are important factors in any coding scheme.

**Block Coding**

In block coding, we divide our message into blocks, each of k bits, called datawords. We add r redundant bits to each block to make the length n = k + r. The resulting n-bit blocks are called codewords.

The 4B/5B block coding is example of this type of coding. In this coding scheme, k = 4 and n = 5. As we saw, we have 2^k = 16 datawords and 2^n = 32 codewords. We saw that 16 out of 32 codewords are used for message transfer and the rest are either used for other purposes or unused.

**Error Detection**

If The following two conditions are met, the receiver can detect a change in the original codeword.

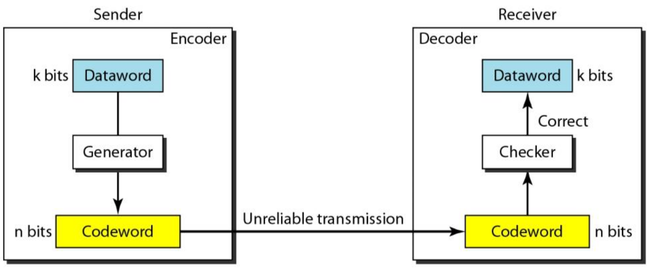
1. The receiver has a list of valid codewords.

2. The original codeword has changed to an invalid one

- Enough redundancy is added to detect an error.
- The receiver knows an error occurred but does not know which bit(s) is(are) in error.
- Has less overhead than error correction.

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

**Figure 10.7** *Structure of encoder and decoder in error correction*



**Hamming Distance**

One of the central concepts in coding for error control is the idea of the Hamming distance. The Hamming distance between two words (of the same size) is the number of differences between the corresponding bit

**The Hamming distance between two words is the number of differences between corresponding bits.**

Let us find the Hamming distance between two pairs of words.

1. The Hamming distance d(000, 011) is 2 because (000 ⊕ 011) is 011 (two 1s).

2. The Hamming distance d(10101, 11110) is 3 because (10101 ⊕ 11110) is 01011 (three 1s)The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

**To guarantee the detection** of up to s errors in all cases, the minimum Hamming distance in a block code must be d_min=s+1

**To guarantee correction** of up to t errors in all cases, the minimum Hamming distance in a block code must be d_min = 2t + 1.

**Linear Block Codes**

Almost all block codes used today belong to a subset called linear block codes. A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.

In a linear block code, the exclusive OR (XOR) of any two valid codewords creates another valid codeword.

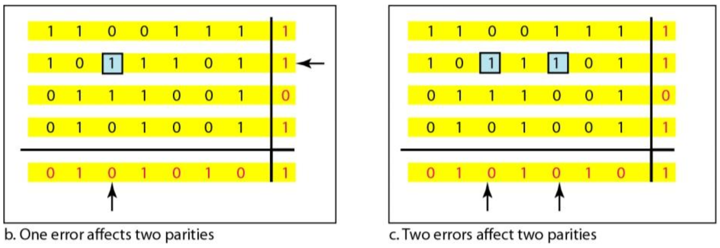
A simple parity-check code is a single-bit error-detecting code in which n = k + 1 with dmin = 2. Even parity (ensures that a codeword has an even number of 1's) and odd parity (ensures that there are an odd number of 1's in the codewor

**Parity-Check Code**

This code is a linear block code. In this code, a k-bit dataword is changed to an n-bit codeword where n = k + 1. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.

A parity-check code can detect an odd number of errors.

**Figure 10.11** *Two-dimensional parity-check code*



b. One error affects two parities

c. Two errors affect two parities

**Figure 10.11** *Two-dimensional parity-check code*

Create r:

r0=a2⊕a1 ⊕a0

r1=a3⊕a2 ⊕a1

r2=a1⊕a0 ⊕a3

Calculate s:

s0=b2⊕b1 ⊕b0 ⊕q0

s1=b3⊕b2 ⊕b1 ⊕q1

s2=b1⊕b0 ⊕b3 ⊕q2

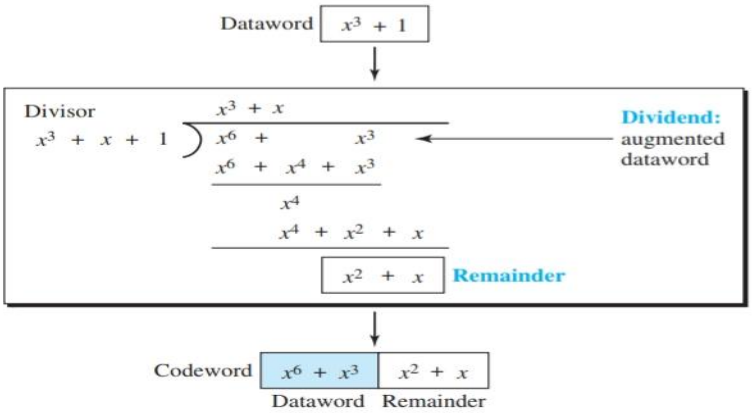
| Sydrome | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| Error | None | q0 | q1 | b2 | q2 | b0 | b3 | b1 |

All Hamming codes have dmin = 3 (2 bit error detection and single bit error correction). A codeword consists of n bits of which k are data bits and r are check bits.

Let m = r, then we have: n = 2^m -1 and k = n-m

**Cyclic codes** are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

**Figure 10.9** *CRC division using polynomials*



**Check sum**

Sender site:

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

Receiver site:

1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one's complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

## Chapter 3.2 Data Link Control

### Framing

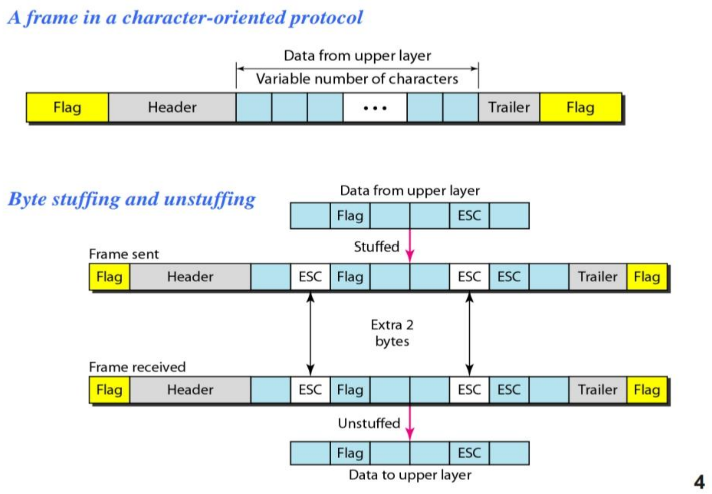
•The data link layer needs to pack bits into frames.

•Each frame is distinguishable from another.

•Framing types:

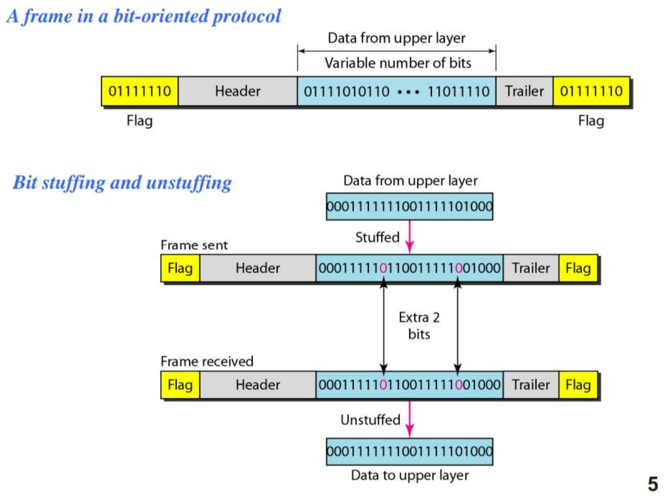
• Fixed-Size Framing: Boundary between frames is not necessary e.g. ATM networks.

• Variable-Size Framing: need methods to define the end and the beginning of frames. We will discuss Character-oriented and Bit-oriented methods.



*A frame in a character-oriented protocol*

*Byte stuffing and unstuffing*



*A frame in a bit-oriented protocol*

*Bit stuffing and unstuffing*

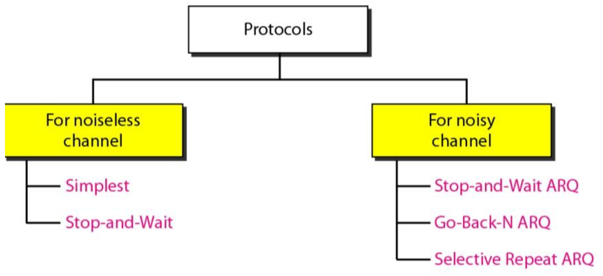
### Flow And Error Control

The most important responsibilities of the data link layer are flow control and error control. These functions are known as data link control. Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

Flow control refers to a set of procedures used to restrict (hạn chế) the amount of data that the sender can send before waiting for acknowledgment.

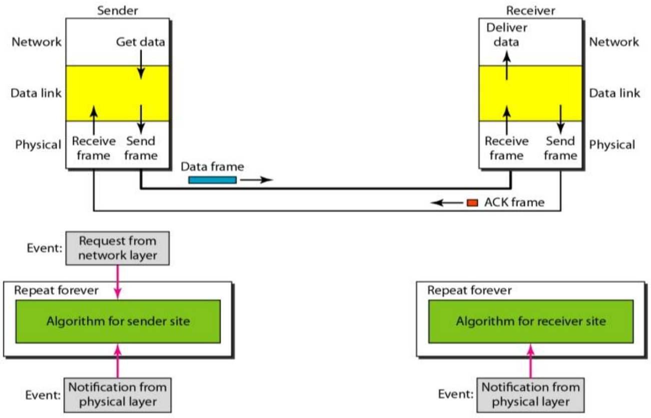
### Protocols

- How the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another?
- The protocols are normally implemented in software.



### Noiseless Channels

Stop-and-Wait Protocol: uses both flow and error control.



**Sender States**

The sender is initially in the ready state, but it can move between the ready and blocking state.

❑Ready State. When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state.

❑ Blocking State. When the sender is in this state, three events can occur:

a. If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.

b. If a corrupted ACK arrives, it is discarded.

c. If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

**Receiver**

The receiver is always in the ready state. Two events may occur:

a. If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.

b. If a corrupted frame arrives, the frame is discarded.

**Noisy Channels**

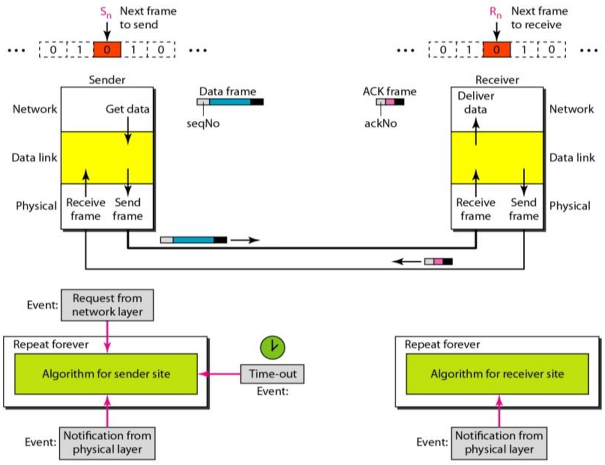
Three protocols in this section that use error control.

- Stop-and-Wait Automatic Repeat Request
- Go-Back-N Automatic Repeat Request
- Selective Repeat Automatic Repeat Request

Stop-and-Wait ARQ

• Keep a copy of the sent frame and retransmit the frame when the timer expires.
• Use sequence numbers to number the frames.
• The acknowledgment number always announces the sequence number of the next frame expected.
• Sequence numbers are based on modulo-2 arithmetic

*Design of the Stop-and-Wait ARQ Protocol*



**Link utilization**

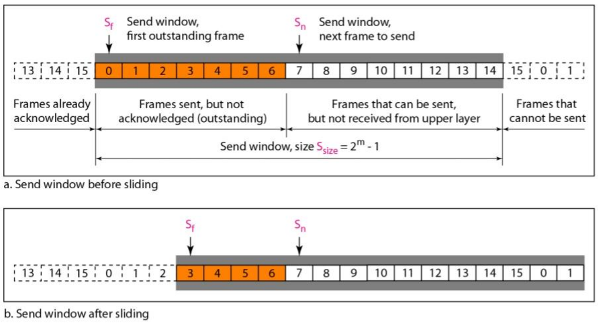
- The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again.
- However, the system sends only 1000 bits i.e. the link utilization is only 1000/20,000, or 5%.
- For a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

**Go-Back-N Automatic Repeat Request**

Selective Repeat Automatic Repeat Request

- Send several frames before receiving acknowledgments
- Keep a copy of sent frames before acknowledgements arrive
- In the Go-Back-N Protocol, the sequence numbers are modulo 2m, where m is the size of the sequence number field in bits.

**Figure 3.2.12** *Send window for Go-Back-N ARQ*



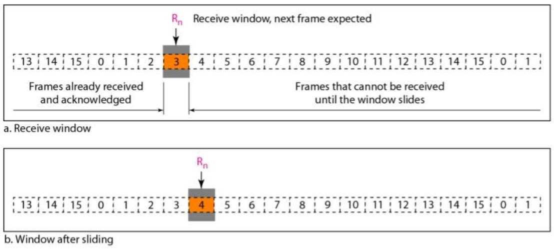
$S_f$ Send window, first outstanding frame
$S_n$ Send window, next frame to send
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already acknowledged
Frames sent, but not acknowledged (outstanding)
Frames that can be sent, but not received from upper layer
Frames that cannot be sent
Send window, size $S_{size} = 2^m - 1$
$S_f$
$S_n$
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Send window before sliding

b. Send window after sliding

The send window is an abstract concept defining an imaginary box of size 2^m − 1 with three variables: Sf, Sn, and S_size. The send window can slide one or more slots when a valid acknowledgment arrives

**Figure 3.2.13** *Receive window for Go-Back-N ARQ*



$R_n$ Receive window, next frame expected
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1
Frames already received and acknowledged
Frames that cannot be received until the window slides
$R_n$
13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1

a. Receive window

b. Window after sliding

The receive window is an abstract concept defining an imaginary box of size 1 with one single variable Rn. The window slides when a correct frame has arrived; sliding occurs one slot at a time.

**Figure 3.2.14** *Design of Go-Back-N ARQ*

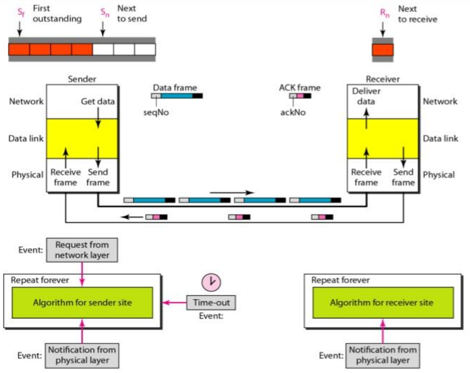
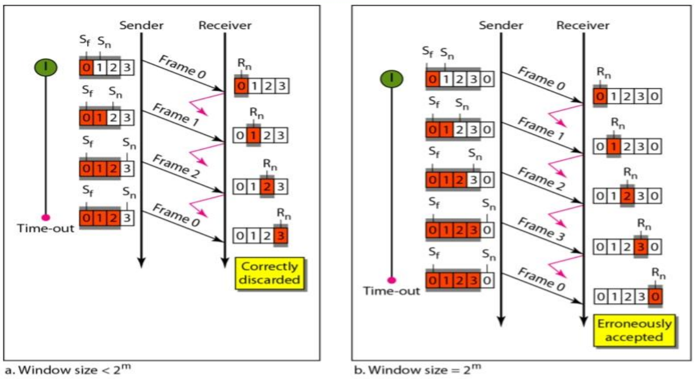


**Figure 3.2.15** *Window size for Go-Back-N ARQ*



a. Window size < $2^m$

b. Window size = $2^m$

In Go-Back-N ARQ, the size of the send window must be less than 2^m; the size of the receiver window is always 1. Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

**Send window for Selective Repeat ARQ**

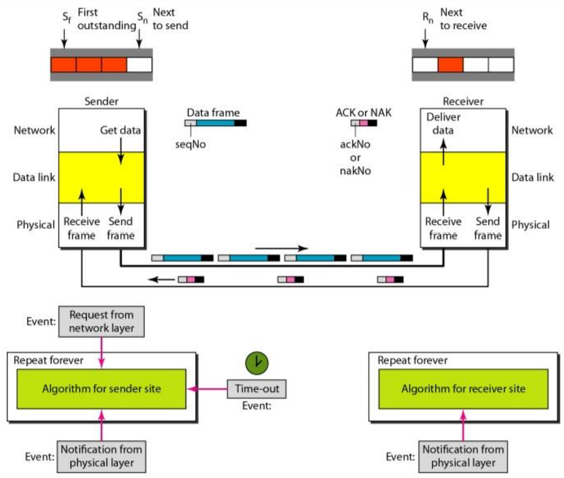
ARQ go-back-N đơn giản hóa phía thu, bộ thu chỉ cần 1 biến Rn, do đó không cần bộ đệm cho các khung không đúng thứ tự, các khung này bị loại bỏ, nên không hiệu quả

• Chỉ gửi lại 1 khung bị lỗi thay vì phải gửi lại toàn bộ N khung tính từ khung bắt đầu bị lỗi-> gọi là lặp lại tự động có lựa chọn. Kích thước cửa sổ gửi và nhận cùng bằng 2^(m-1)

**Receive window for Selective Repeat ARQ**

Giao thức này cho phép bên nhận nhận nhiều khung không đúng thứ tự và giữ cho đến khi có đủ các khung theo đến theo đúng thứ tự

**Design of Selective Repeat ARQ**



**Selective Repeat ARQ, window size**

Trong Selective Repeat ARQ, kích thước của cửa sổ gửi và nhận nhiều nhất là 2^(m-1)

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m

**Figure 3.2.22** *Delivery of data in Selective Repeat ARQ*

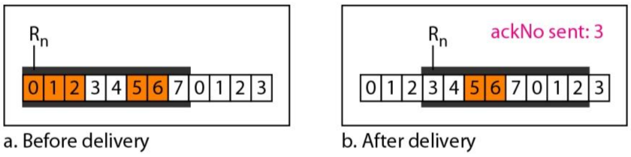
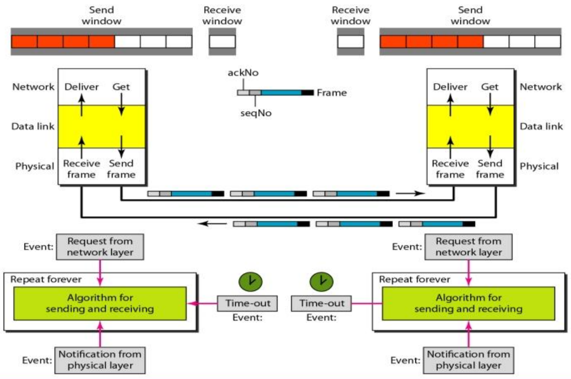


**Figure 3.2.24** *Design of piggybacking in Go-Back-N ARQ*



**HDLC**

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms we discussed in this chapter.

**Figure 3.2.27** *HDLC frames*

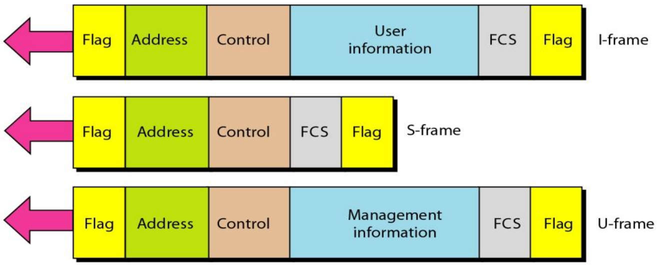
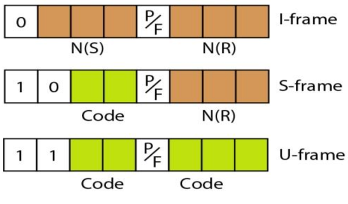


**Figure 3.2.28** *Control field format for the different frame types*



**P: Poll; F: Final (frame I)**

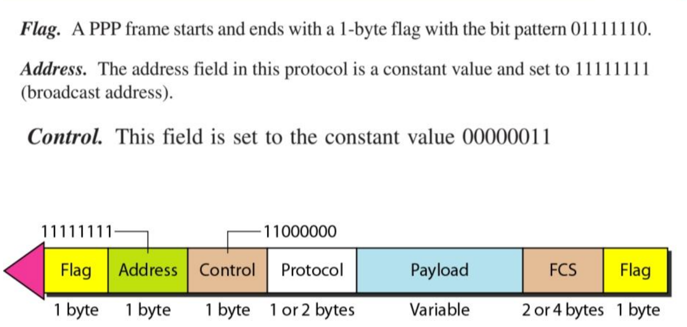
It means ***poll*** when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver). It means ***final*** when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

**Point-to-point Protocol**

Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). PPP is a byte-oriented protocol.

PPP frame format

**Figure 11.32** *PPP frame format*

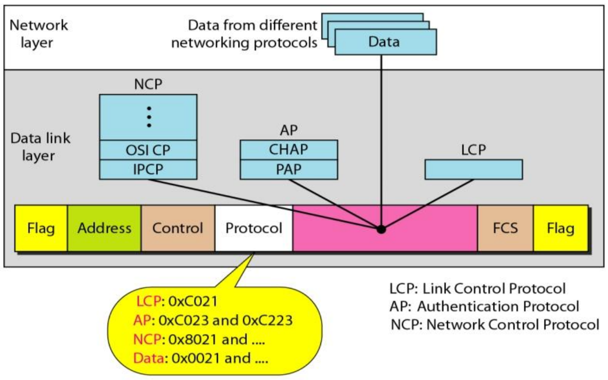


*Flag.* A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110.

*Address.* The address field in this protocol is a constant value and set to 11111111 (broadcast address).

*Control.* This field is set to the constant value 00000011

**Figure 3.2.34** *Multiplexing in PPP*



PPP is a byte-oriented protocol using byte stuffing with the escape byte 01111101.

**Figure 3.2.35** *LCP packet encapsulated in a frame*

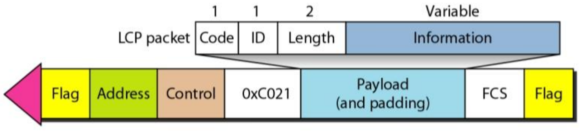
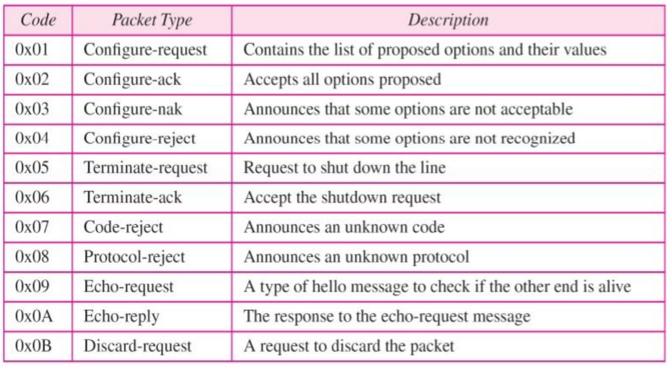


**Table 3.2.2** *LCP packets*



| *Code* | *Packet Type* | *Description* |
|---|---|---|
| 0x01 | Configure-request | Contains the list of proposed options and their values |
| 0x02 | Configure-ack | Accepts all options proposed |
| 0x03 | Configure-nak | Announces that some options are not acceptable |
| 0x04 | Configure-reject | Announces that some options are not recognized |
| 0x05 | Terminate-request | Request to shut down the line |
| 0x06 | Terminate-ack | Accept the shutdown request |
| 0x07 | Code-reject | Announces an unknown code |
| 0x08 | Protocol-reject | Announces an unknown protocol |
| 0x09 | Echo-request | A type of hello message to check if the other end is alive |
| 0x0A | Echo-reply | The response to the echo-request message |
| 0x0B | Discard-request | A request to discard the packet |

**Figure 3.2.36** *PAP packets encapsulated in a PPP frame*

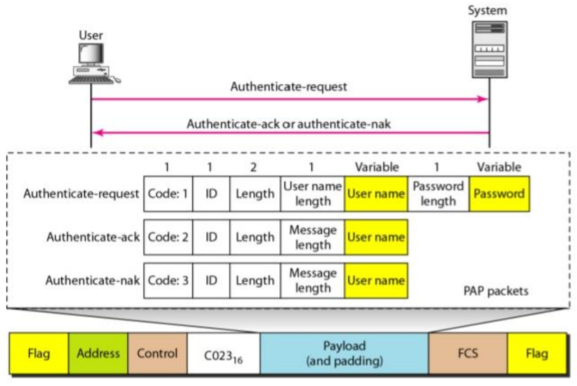
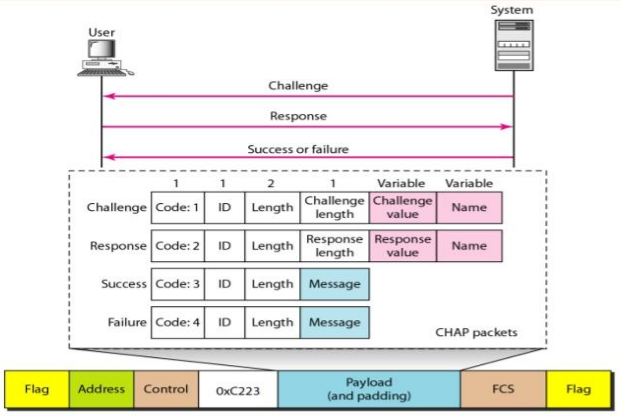


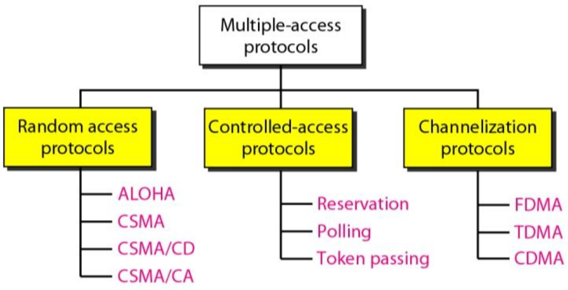
**Figure 3.2.37** *CHAP packets encapsulated in a PPP frame*



## Chapter 3.3: Multiple Access

Data link layer divided into two functionality-oriented sublayers: Data-link control and multiple-access resolution

Taxonomy of multiple-access protocols discussed in this chapter



**Random access protocol**

No station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.

**ALOHA**, the earliest random access method

**Figure 3.3.3** *Frames in a pure ALOHA network*

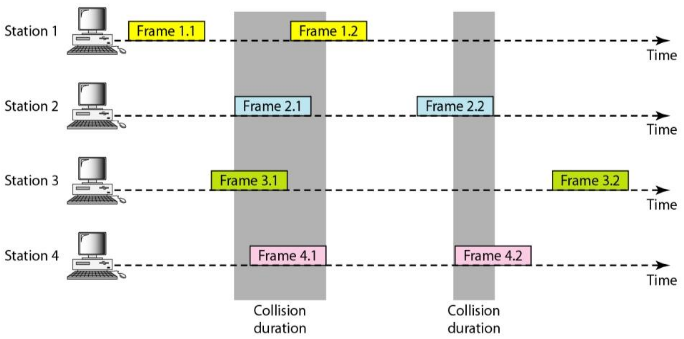
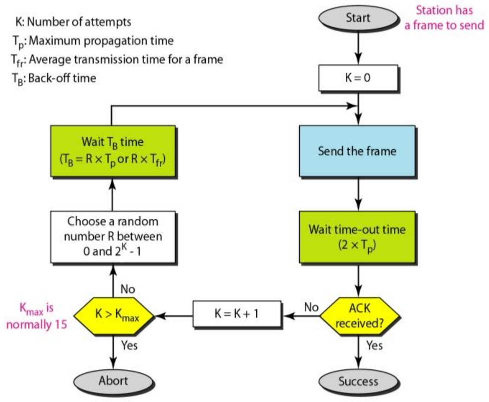


**Figure 3.3.4** *Procedure for pure ALOHA protocol*


K: Number of attempts
$T_p$: Maximum propagation time
$T_{fr}$: Average transmission time for a frame
$T_B$: Back-off time
Start
Station has a frame to send
K = 0
Wait $T_B$ time ($T_B = R \times T_p$ or $R \times T_{fr}$)
Send the frame
Choose a random number R between 0 and $2^K - 1$
Wait time-out time ($2 \times T_p$)
No
$K_{max}$ is normally 15
$K > K_{max}$
K = K + 1
No
ACK received?
Yes
Yes
Abort
Success

The data from the two stations collide and become garbled. The idea is that each station sends a frame whenever it has a frame to sendsince there is only one channel to share, there is the possibility of collision between frames from different stations. Four stations (unrealistic assumption) that contend with one another for access to the shared channel.

Pure ALOHA vulnerable time = 2 x T_fr.

The throughput for pure ALOHA is S = G × e −2G .The maximum throughput Smax = 0.184 when G= (1/2).

G là số khung trung bình tạo ra bởi hệ thống trong thời gian truyền dẫn 1 khung.

The throughput for slotted ALOHA is S = G × e^ −G. The maximum throughput Smax = 0.368 when G = 1.

**CSMA/CD**

The CSMA method does not specify the procedure following a collision. Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

**CSMA/CA** Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks. Collisions are avoided through the use of CSMA/CA's three strategies: the interframe space, the contention window, and acknowledgments. In CSMA/CA, the IFS can also be used to define the priority of a station or a frame. In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

**Controlled Access**

In controlled access, the stations consult (hỏi) one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three popular controlled-access methods.

**Figure 3.3.18** *Reservation access method*

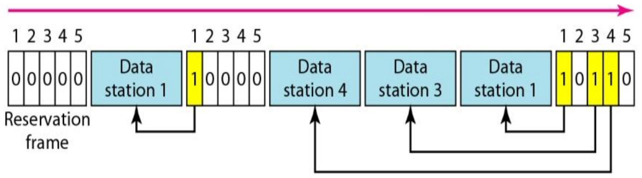
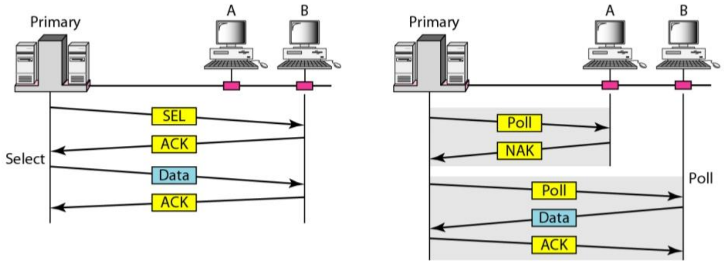


**Figure 3.3.19** *Select and poll functions in polling access method*



**Channelization**

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. In this section, we discuss three channelization protocols.

**Figure 3.3.21** *Frequency-division multiple access (FDMA)*

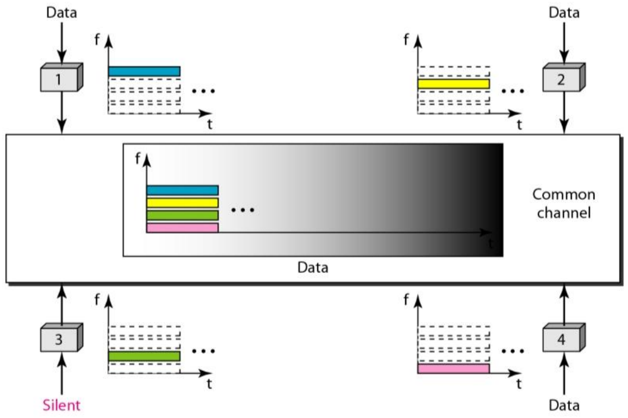
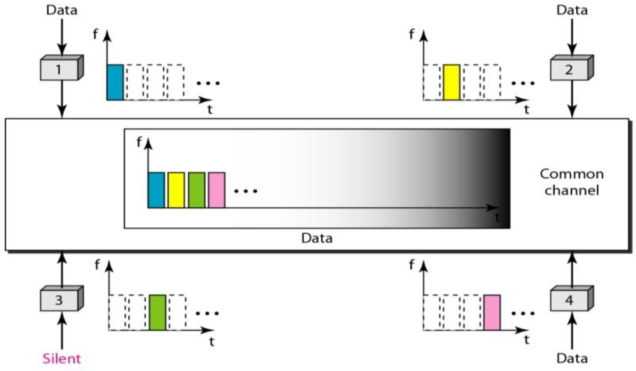


**Figure 3.3.22** *Time-division multiple access (TDMA)*



In FDMA, the available bandwidth of the common channel is divided into bands that are separated by guard bands.

In TDMA, the bandwidth is just one channel that is timeshared between different stations.

In CDMA, one channel carries all transmissions simultaneously

**Idea** Let us assume we have four stations, 1, 2, 3, and 4, connected to the same channel. The data from station 1 are d1, from station 2 are d2, and so on. The code assigned to the first station is c1, to the second is c2, and so on. We assume that the assigned codes have two properties.

1. If we multiply each code by another, we get 0.
2. If we multiply each code by itself, we get 4 (the number of stations).

***The number of sequences in a Walsh table needs to be N = 2^m***

**Figure 3.3.23** *Simple idea of communication with code*

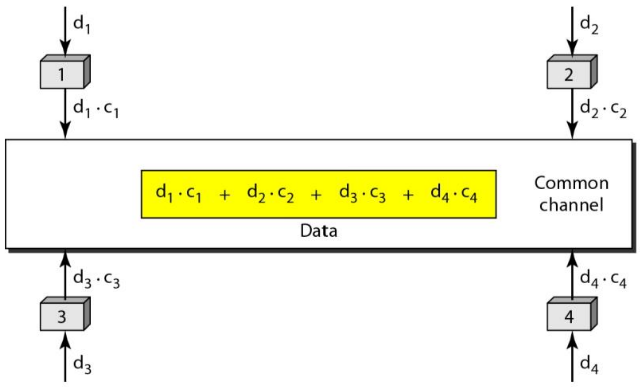
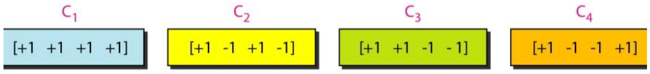


**Figure 3.3.24** *Chip sequences*



## Chapter 4.1 Logical Addressing

### Ipv4 Addresses

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device to the Internet
The address space of IPv4 is 2^32 or 4,294,967,296. Each number needs to be less than or equal to 255.

**Figure 4.1** *Dotted-decimal notation and binary notation for an IPv4 address*

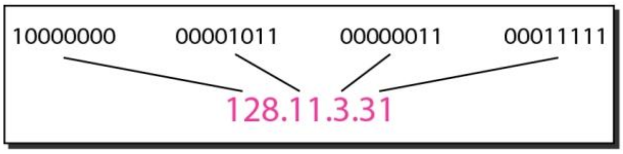
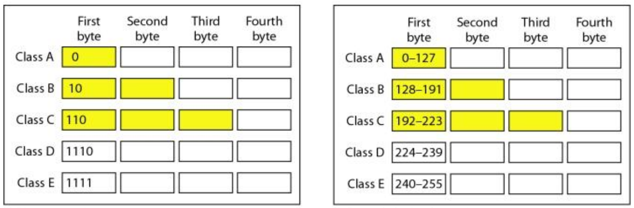


**Figure 4.1.2** *Finding the classes in binary and dotted-decimal notation*



| | First byte | Second byte | Third byte | Fourth byte |
|---|---|---|---|---|
| Class A | 0 | | | |
| Class B | 10 | | | |
| Class C | 110 | | | |
| Class D | 1110 | | | |
| Class E | 1111 | | | |

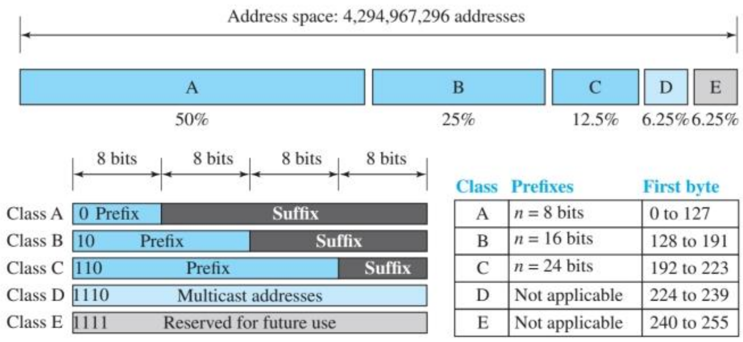
a. Binary notation

| | First byte | Second byte | Third byte | Fourth byte |
|---|---|---|---|---|
| Class A | 0–127 | | | |
| Class B | 128–191 | | | |
| Class C | 192–223 | | | |
| Class D | 224–239 | | | |
| Class E | 240–255 | | | |

b. Dotted-decimal notation

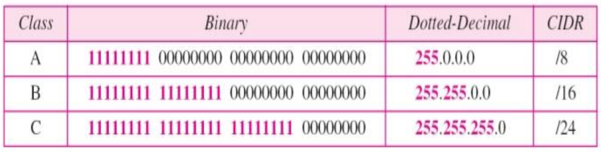
**Classful addressing,** the address space is divided into five classes: A, B, C, D, and E.
In classful addressing, a large part of the available addresses were wasted.
Advs: Given an address, we can easily find the class of the address and, since the prefix length for each class is fixed, we can find the prefix length immediately. In other words, the prefix length in classful addressing is inherent in the address; no extra information is needed to extract the prefix and the suffix.

**Table 4.1** *Number of blocks and block size in classful IPv4 addressing*



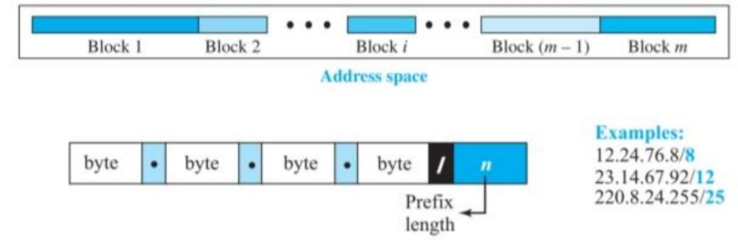
| Class | Prefixes | First byte |
|---|---|---|
| A | $n = 8$ bits | 0 to 127 |
| B | $n = 16$ bits | 128 to 191 |
| C | $n = 24$ bits | 192 to 223 |
| D | Not applicable | 224 to 239 |
| E | Not applicable | 240 to 255 |

**Table 4.1.2** *Default masks for classful addressing*

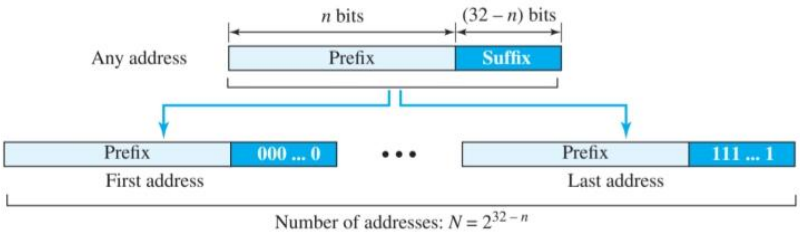


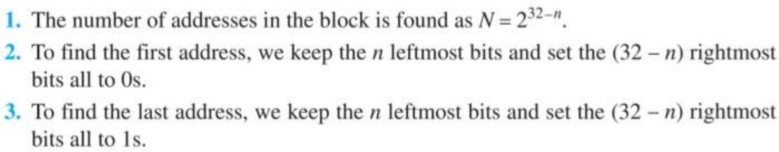
| *Class* | *Binary* | *Dotted-Decimal* | *CIDR* |
|---|---|---|---|
| A | 11111111 00000000 00000000 00000000 | 255.0.0.0 | /8 |
| B | 11111111 11111111 00000000 00000000 | 255.255.0.0 | /16 |
| C | 11111111 11111111 11111111 00000000 | 255.255.255.0 | /24 |

### Classless Addressing



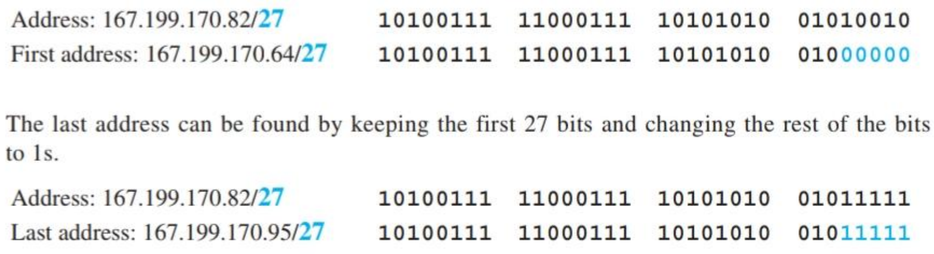
CIDR = Classless Inter-Domain Routing
Classful addressing, which is almost obsolete, is replaced with classless addressing





1. The number of addresses in the block is found as $N = 2^{32-n}$.
2. To find the first address, we keep the $n$ leftmost bits and set the $(32 - n)$ rightmost bits all to 0s.
3. To find the last address, we keep the $n$ leftmost bits and set the $(32 - n)$ rightmost bits all to 1s.

Ex1: A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows. The number of addresses in the network is 2^(32 − n )= 2^5 = 32 addresses.



```
Address: 167.199.170.82/27        10100111  11000111  10101010  01010010
First address: 167.199.170.64/27  10100111  11000111  10101010  01000000
```

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

```
Address: 167.199.170.82/27        10100111  11000111  10101010  01011111
Last address: 167.199.170.95/27   10100111  11000111  10101010  01011111
```

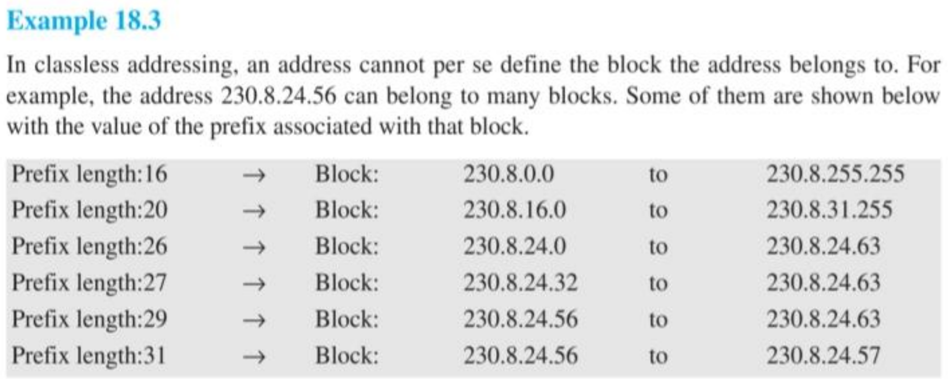
**Address Mask**

Another way to find the first and last addresses in the block is to use the address mask. The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits (32 − n) are set to 0s. A computer can easily find the address mask because it is the complement of (232 − n − 1). The reason for defining a mask in this way is that it can be used by a computer program to extract the information in a block, using the three bit-wise operations NOT, AND, and OR.

1. The number of addresses in the block N = NOT (mask) + 1.
2. The first address in the block = (Any address in the block) AND (mask).
3. The last address in the block = (Any address in the block) OR [(NOT (mask)].

Ex2: Repeat Ex1 using the mask. The mask in dotted-decimal notation is 256.256.256.224. The AND, OR, and NOT operations can be applied to individual bytes using calculators and applets at the book website.

Number of addresses in the block: N = NOT (mask) + 1= 0.0.0.31 + 1 = 32 addresses
First address: First = (address) AND (mask) = 167.199.170.82
Last address: Last = (address) OR (NOT mask) = 167.199.170.255

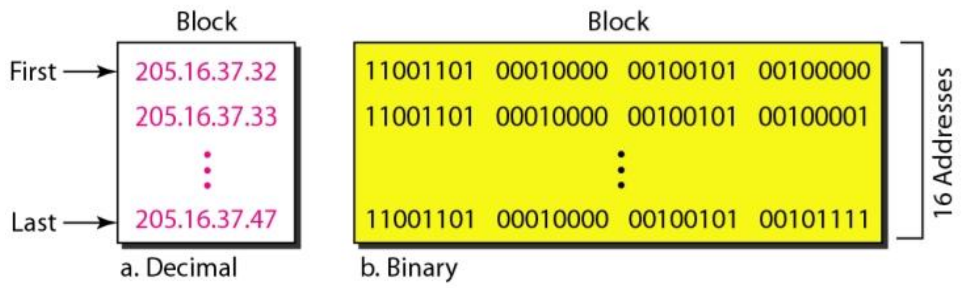


### Example 18.3

In classless addressing, an address cannot per se define the block the address belongs to. For example, the address 230.8.24.56 can belong to many blocks. Some of them are shown below with the value of the prefix associated with that block.

| | | | | | |
|---|---|---|---|---|---|
| Prefix length:16 | → | Block: | 230.8.0.0 | to | 230.8.255.255 |
| Prefix length:20 | → | Block: | 230.8.16.0 | to | 230.8.31.255 |
| Prefix length:26 | → | Block: | 230.8.24.0 | to | 230.8.24.63 |
| Prefix length:27 | → | Block: | 230.8.24.32 | to | 230.8.24.63 |
| Prefix length:29 | → | Block: | 230.8.24.56 | to | 230.8.24.63 |
| Prefix length:31 | → | Block: | 230.8.24.56 | to | 230.8.24.57 |

**Figure 4.1.4** *A network configuration for the block 205.16.37.32/28*



The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

**Figure 4.1.6** *A frame in a character-oriented protocol*

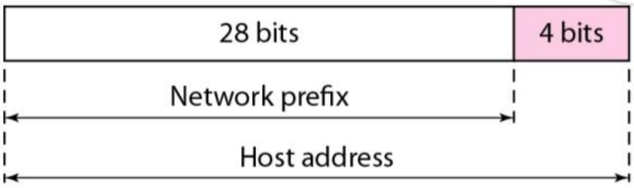
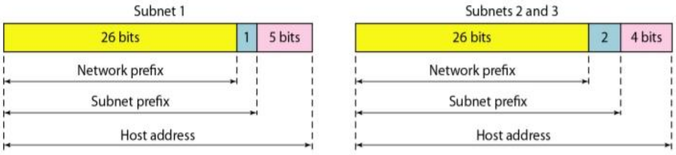


**Figure 4.1.8** *Three-level hierarchy in an IPv4 address*



Each address in the block can be considered as a two-level hierarchical structure: the leftmost n bits (prefix) define the network; the rightmost 32 − n bits define the host.

**Subnetting**

An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet). Note that nothing stops the organization from creating more levels. A subnetwork can be divided into several sub-subnetworks.

A sub-subnetwork can be divided into several sub-sub-subnetworks.

Designing Subnets

The subnetworks in a network should be carefully designed to enable the routing of packets. We assume the total number of addresses granted to the organization is N, the prefix length is n, the assigned number of addresses to each subnetwork is N_sub, and the prefix length for each subnetwork is nsub. Then the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

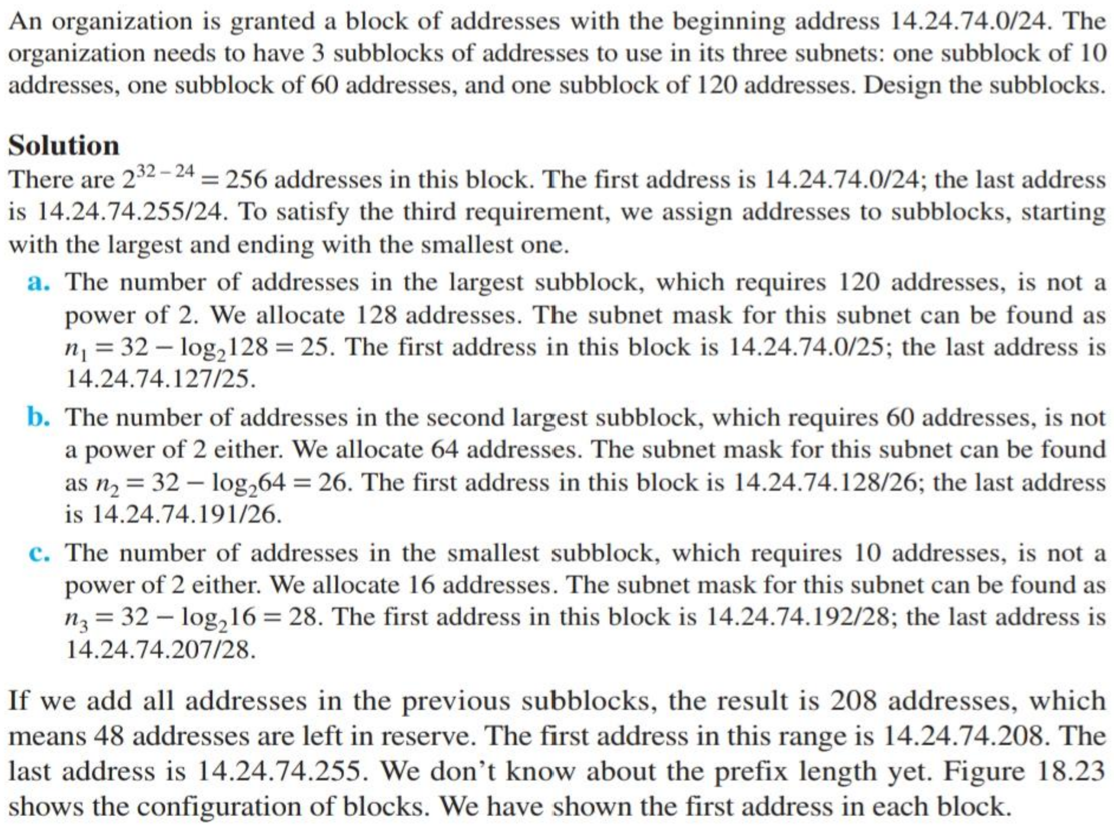
❑ The number of addresses in each subnetwork should be a power of 2.

❑ The prefix length for each subnetwork should be found using the following formula: n_sub = 32 − log2N_sub

❑ The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger subnetworks.

.After designing the subnetworks, the information about each subnetwork, such as first and last address, can be found using the process we described to find the information about each network in the Internet.

Example



An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets: one subblock of 10 addresses, one subblock of 60 addresses, and one subblock of 120 addresses. Design the subblocks.

**Solution**

There are $2^{32-24} = 256$ addresses in this block. The first address is 14.24.74.0/24; the last address is 14.24.74.255/24. To satisfy the third requirement, we assign addresses to subblocks, starting with the largest and ending with the smallest one.

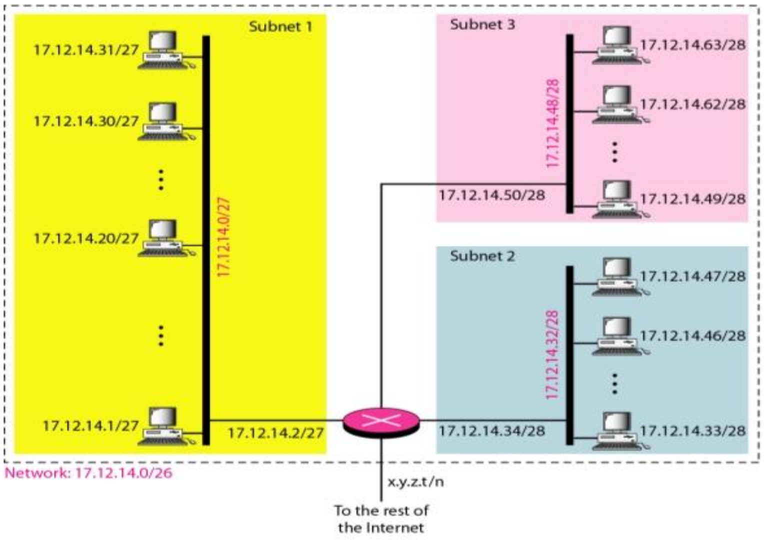
a. The number of addresses in the largest subblock, which requires 120 addresses, is not a power of 2. We allocate 128 addresses. The subnet mask for this subnet can be found as $n_1 = 32 - \log_2 128 = 25$. The first address in this block is 14.24.74.0/25; the last address is 14.24.74.127/25.

b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as $n_2 = 32 - \log_2 64 = 26$. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

c. The number of addresses in the smallest subblock, which requires 10 addresses, is not a power of 2 either. We allocate 16 addresses. The subnet mask for this subnet can be found as $n_3 = 32 - \log_2 16 = 28$. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.

If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.208. The last address is 14.24.74.255. We don't know about the prefix length yet. Figure 18.23 shows the configuration of blocks. We have shown the first address in each block.

gure 4.1.7 *Configuration and addresses in a subnetted network*



Ex: An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and find out how many addresses are still available after these allocations.

**Figure 4.1.9** *An example of address allocation and distribution by an ISP*

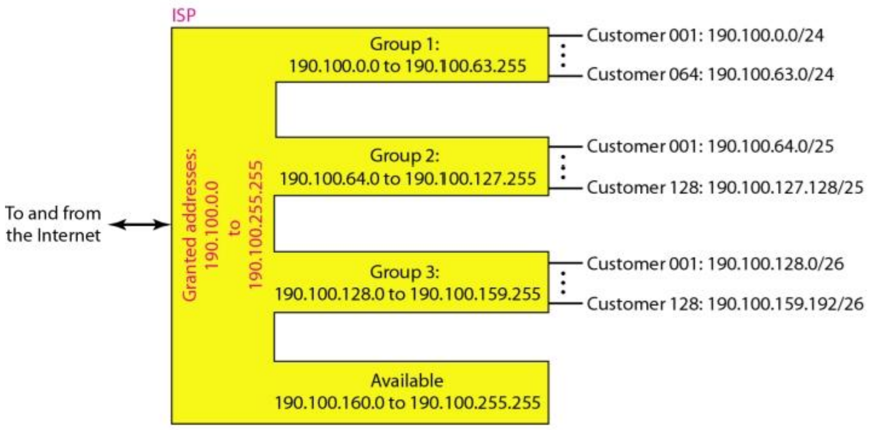
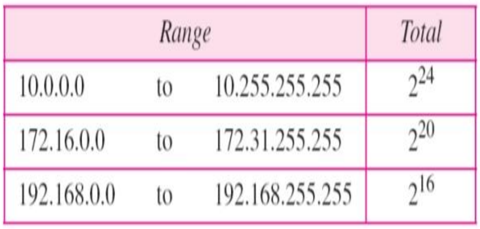

ISP
Group 1:
190.100.0.0 to 190.100.63.255
Customer 001: 190.100.0.0/24
Customer 064: 190.100.63.0/24
Group 2:
190.100.64.0 to 190.100.127.255
Customer 001: 190.100.64.0/25
Customer 128: 190.100.127.128/25
Group 3:
190.100.128.0 to 190.100.159.255
Customer 001: 190.100.128.0/26
Customer 128: 190.100.159.192/26
Available
190.100.160.0 to 190.100.255.255
Granted addresses:
190.100.0.0
to
190.100.255.255
To and from the Internet

**Table 4.1.3** *Addresses for private networks*



| *Range* | | | *Total* |
|---|---|---|---|
| 10.0.0.0 | to | 10.255.255.255 | $2^{24}$ |
| 172.16.0.0 | to | 172.31.255.255 | $2^{20}$ |
| 192.168.0.0 | to | 192.168.255.255 | $2^{16}$ |

## Chapter 4.2. - Internet Protocol

**Internetworking**: connecting networks together to make an internetwork or an Internet.

**Figure 4.2.1** *Links between two hosts*

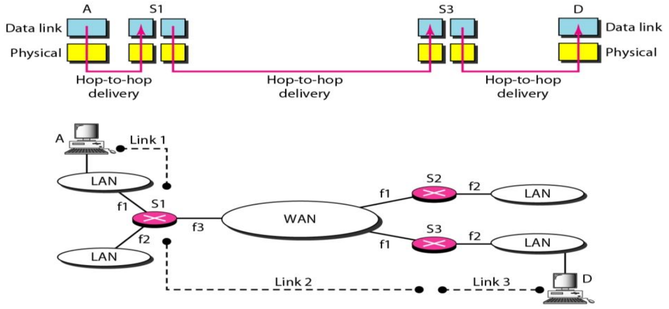
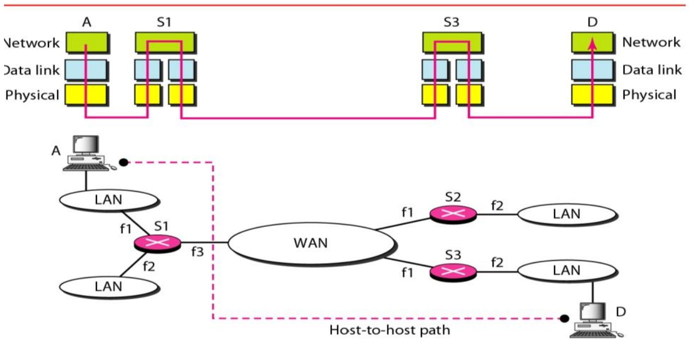
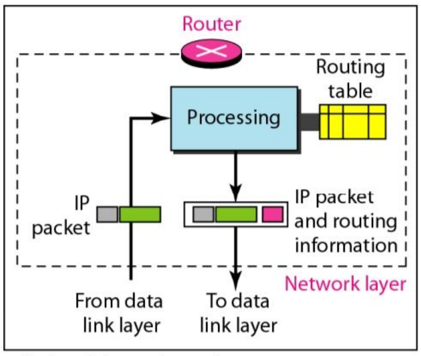


**Figure 4.2.2** *Network layer in an internetwork*

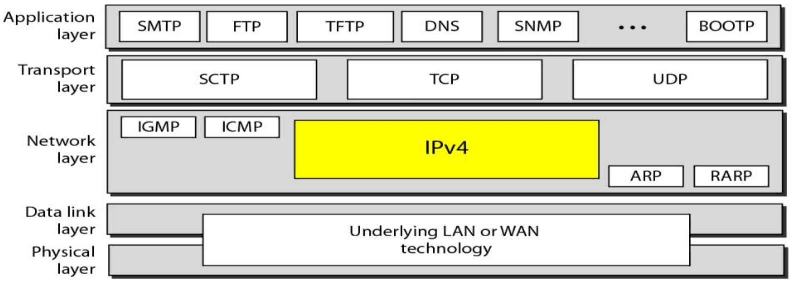


Communication at the network layer in  the Internet is connectionless. Switching at the network layer in the Internet uses the datagram approach to packet switching.



c. Network layer at a router

**Figure 4.2.4** *Position of IPv4 in TCP/IP protocol suite*



### IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

**Figure 4.2.5** *IPv4 datagram format*

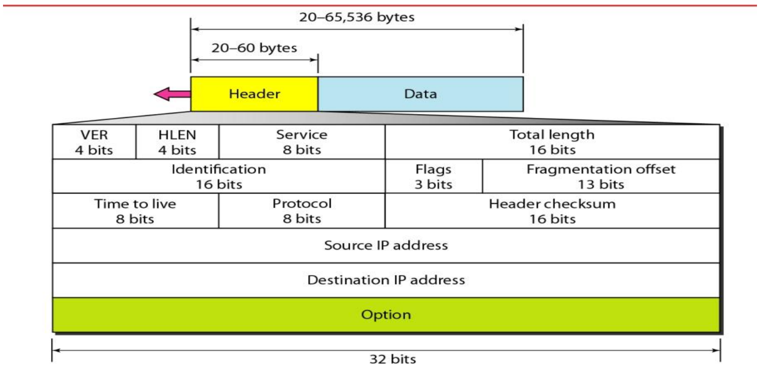


**Figure 4.2.6** **Service:** *Service type or differentiated services*

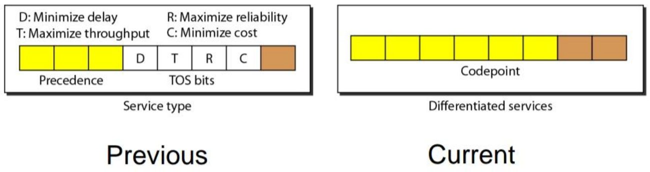
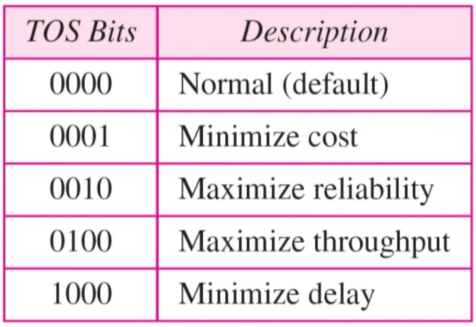
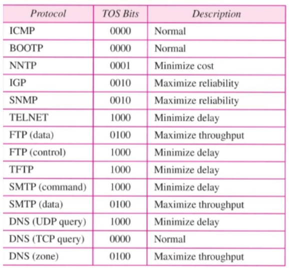


**Table 4.2.1** *Types of service*



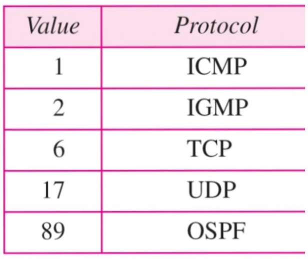
| *TOS Bits* | *Description* |
|---|---|
| 0000 | Normal (default) |
| 0001 | Minimize cost |
| 0010 | Maximize reliability |
| 0100 | Maximize throughput |
| 1000 | Minimize delay |

**Table 4.2.2** *Default types of service*



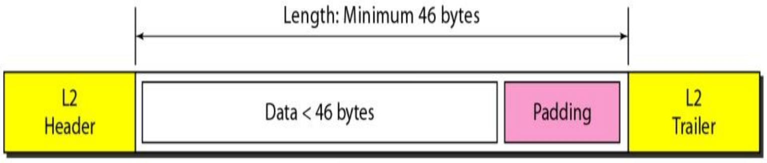
| *Protocol* | *TOS Bits* | *Description* |
|---|---|---|
| ICMP | 0000 | Normal |
| BOOTP | 0000 | Normal |
| NNTP | 0001 | Minimize cost |
| IGP | 0010 | Maximize reliability |
| SNMP | 0010 | Maximize reliability |
| TELNET | 1000 | Minimize delay |
| FTP (data) | 0100 | Maximize throughput |
| FTP (control) | 1000 | Minimize delay |
| TFTP | 1000 | Minimize delay |
| SMTP (command) | 1000 | Minimize delay |
| SMTP (data) | 0100 | Maximize throughput |
| DNS (UDP query) | 1000 | Minimize delay |
| DNS (TCP query) | 0000 | Normal |
| DNS (zone) | 0100 | Maximize throughput |

**Table 4.2.3** *Values for codepoints*



| *Value* | *Protocol* |
|---|---|
| 1 | ICMP |
| 2 | IGMP |
| 6 | TCP |
| 17 | UDP |
| 89 | OSPF |

**Figure 4.2.7** *Encapsulation of a small datagram in an Ethernet frame*



Ex1:: An IPv4 packet has arrived with the first 8 bits as shown: 01000010 The receiver discards the packet. Why?

=> There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct（4= IPv4, 6= IPv6). The next 4 bits (0010) show an invalid header length (2 × 4 = 8). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Ex2 In an IPv4 packet, the value of HLEN is 1000 in binary.How many bytes of options are being carried by this packet?
=> The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Ex3: In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?
The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20)

Ex4: An IPv4 packet has arrived with the first few hexadecimal digits as shown. 0x45000028000100000102 . . .How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?
=>To find the time-to-live field, we skip 8 bytes. The time-to- live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP

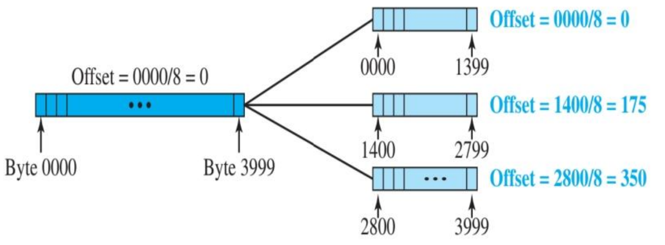
**Phân mảnh gói tin:**

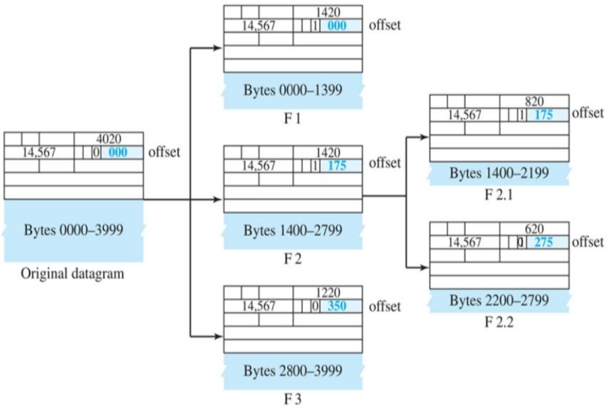
o Trường nhận dạng: 16 bit, nhận dạng gói tin được tạo ra từ nguồn, là duy nhất được copy vào tất cả các mảnh, giúp đích tổng hợp lại gói tin

o Cờ: 3 bit, Bit đầu tiên không sử dụng, Bit thứ 2 (D): không phân mảnh, nếu nó bằng 1 thì nút không phải phân mảnh gói tin, nếu nó bằng 0 thì có thể phân mảnh nếu cần. Bit cuối cùng (M): 1 tức là không phải mảnh cuối, 0 có nghĩa là mảnh cuối hoặc chỉ có 1 mảnh.

o Offset: 13 bit cho biết vị trí tương đối của mảnh so với toàn bộ gói tin

o Được đo theo đơn vị 8 byte





**IPv6**

**Transition From Ipv4 To Ipv6**

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems

*3 strategy:* Dual Stack; Tunneling; Header Translation

Figure 4.2.19 *Dual stack*

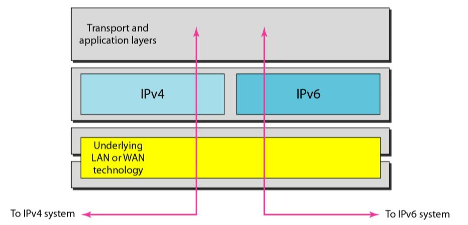

Transport and application layers
IPv4
IPv6
Underlying LAN or WAN technology
To IPv4 system
To IPv6 system

Figure 4.2.20 *Tunneling strategy*

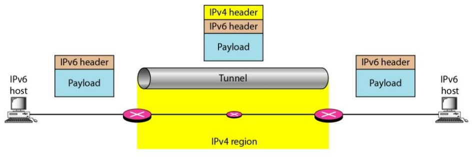
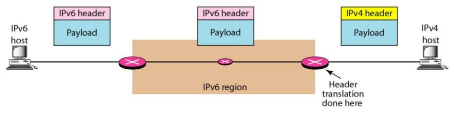


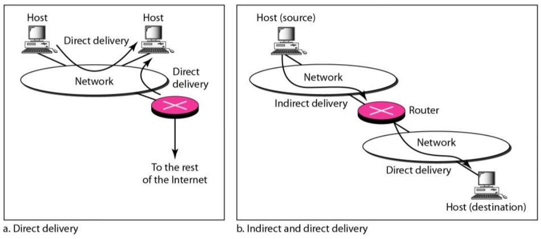
Figure 4.2.21 *Header translation strategy*



## Chapter 4.3: Routing-Phân phối, chuyển tiếp và định tuyến

### Phân Phối

**Hình 4.4.1** *Phân phối trực tiếp và gián tiếp*



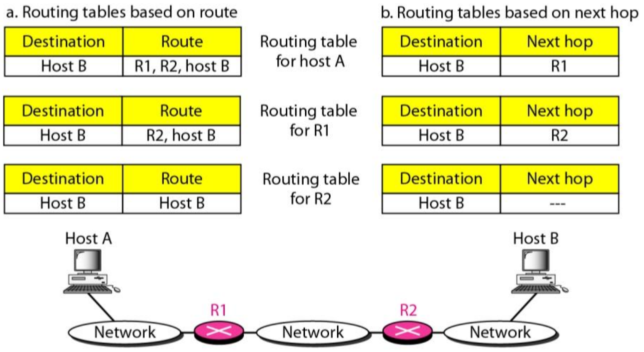
a. Direct delivery

b. Indirect and direct delivery

### Chuyển Tiếp - Forwarding

- Chuyển tiếp có nghĩa là đạt gói tin trên đường đi để đến đích
- Chuyển tiếp yêu cầu một host hay một router có bảng định tuyến.
- Khi một host có một gói tin gửi đi hay khi một router nhận được một gói tin cần được chuyển tiếp, thì sẽ dựa vào bảng định tuyến để tìm đường đi đến đích cuối cùng.

**Hình 4.4.2** *Phương pháp tìm đường so với phương pháp hop – tiếp theo*



a. Routing tables based on route

| Destination | Route |
|---|---|
| Host B | R1, R2, host B |

| Destination | Route |
|---|---|
| Host B | R2, host B |

| Destination | Route |
|---|---|
| Host B | Host B |

Routing table for host A

Routing table for R1

Routing table for R2

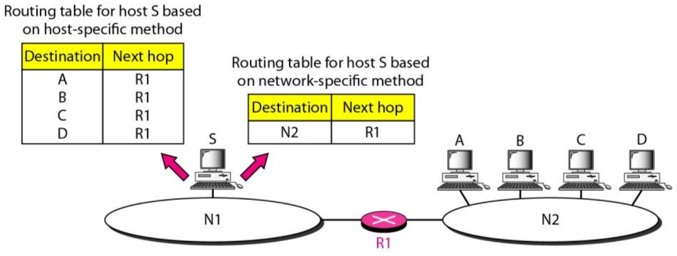
b. Routing tables based on next hop

| Destination | Next hop |
|---|---|
| Host B | R1 |

| Destination | Next hop |
|---|---|
| Host B | R2 |

| Destination | Next hop |
|---|---|
| Host B | --- |

**Hình 4.4.3** *Xác định host cụ thể so với phương pháp xác định mạng*



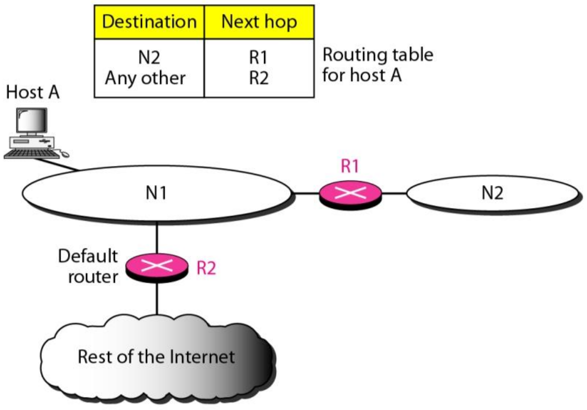
Routing table for host S based on host-specific method

| Destination | Next hop |
|---|---|
| A<br>B<br>C<br>D | R1<br>R1<br>R1<br>R1 |

Routing table for host S based on network-specific method

| Destination | Next hop |
|---|---|
| N2 | R1 |

**Hình 4.4.4** *Phương pháp mặc định*

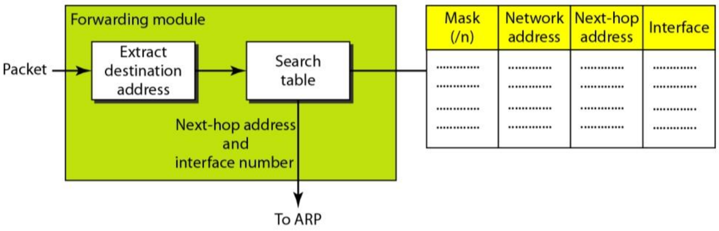


| Destination | Next hop |
|---|---|
| N2<br>Any other | R1<br>R2 |

Routing table for host A

**Hình 4.4.5** *Khối chuyển tiếp đơn giản hóa trong địa chỉ không phân lớp*

7-8 / 44

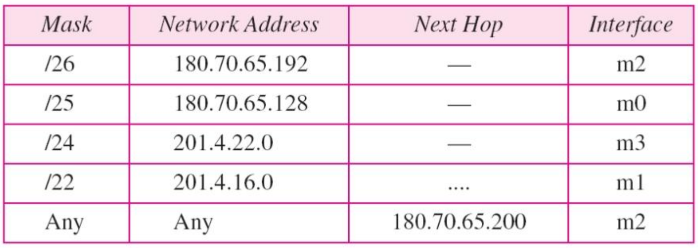

Forwarding module
Packet
Extract destination address
Search table
Next-hop address and interface number
To ARP

| Mask (/n) | Network address | Next-hop address | Interface |
|---|---|---|---|
| ............ | ............ | ............ | ............ |
| ............ | ............ | ............ | ............ |
| ............ | ............ | ............ | ............ |
| ............ | ............ | ............ | ............ |

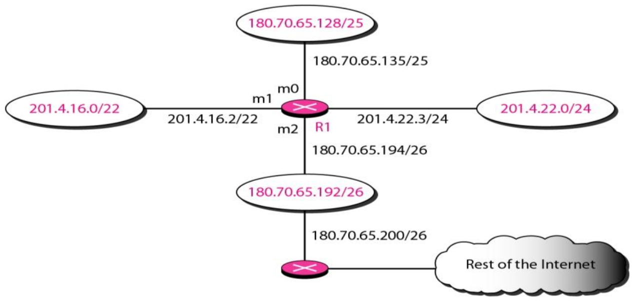
Ví dụ: Xây dựng một bảng định tuyến cho router R1, sử dụng định tuyến trong Hình 4.4.6.

**Hình 4.4.6** *Cấu hình cho ví dụ 4.4.1*

**Bảng 4.4.1** *Bảng định tuyến cho router R1 trong Hình 4.4.6*



| Mask | Network Address | Next Hop | Interface |
|---|---|---|---|
| /26 | 180.70.65.192 | — | m2 |
| /25 | 180.70.65.128 | — | m0 |
| /24 | 201.4.22.0 | — | m3 |
| /22 | 201.4.16.0 | .... | m1 |
| Any | Any | 180.70.65.200 | m2 |

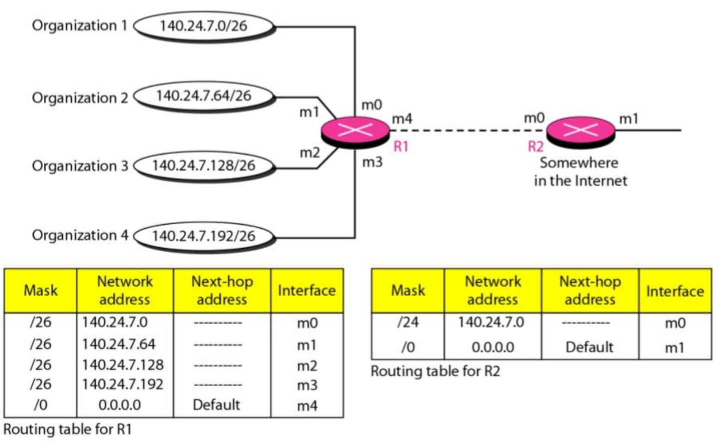


Ví dụ 2: Biểu diễn quá trình chuyển tiếp nếu một gói tin đến tại R1 trong Hình 4.4.6 với địa chỉ đích là 180.70.65.140.
=> Router thực hiện các bước sau:
1. Mặt nạ đầu tiên (/26) được áp dụng cho địa chỉ đích. Kết quả là 180.70.65.128, không tương thích với địa chỉ mạng tương ứng.
2. Mặt nạ thứ hai (/25) được áp dụng cho địa chỉ đích. Kết quả là 180.70.65.128, phù hợp với địa chỉ mạng tương ứng. Địa chỉ hop tiếp theo và giao diện m0 được chuyển cho ARP để thực hiện quá trình tiếp theo.

Ví dụ 3: Biểu diễn quá trình chuyển tiếp nếu một gói tin đến tại R1 trong Hình 4.4.6 với địa chỉ đích là 201.4.22.35.
=> Router thực hiện các bước sau:
1. Mặt nạ đầu tiên (/26) được áp dụng cho địa chỉ đích. Kết quả là 201.4.22.0, không phù hợp với địa chỉ mạng tương ứng.
2. mặt nạ thứ hai (/25) được áp dụng cho địa chỉ đích. Kết quả là 201.4.22.0, không tương thích với địa chỉ mạng tương ứng (row 2).
3. Mặt nạ thứ ba (/24) được áp dụng cho địa chỉ đích. Kết quả là 201.4.22.0, phù hợp với địa chỉ mạng tương ứng. Địa chỉ đích của gói tin và giao diện m3 được chuyển tới ARP.

Ví dụ 4: Biểu diễn quá trình chuyển tiếp nếu một gói tin đến tại R1 trong Hình 4.4.6 với địa chỉ đích 18.24.32.78.
=> tất cả các mặt nạ đều được lần lượt áp dụng cho địa chỉ đích, nhưng không tìm thấy địa chỉ mạng phù hợp. Khi kết thúc bảng, khối mô-đun đia địa chỉ của hop tiếp theo 180.70.65.200 và giao diện m2 đến ARP. Điều này có thể là một gói tin đi ra ngoài cần được gửi đi, thông qua một router mặc định, để đến một nơi nào đó trong Internet.

**Hình 4.4.7** ***Tập hợp địa chỉ***



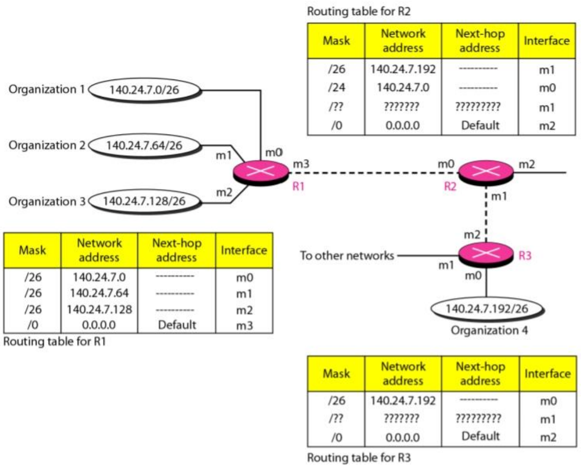
| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /26 | 140.24.7.0 | ---------- | m0 |
| /26 | 140.24.7.64 | ---------- | m1 |
| /26 | 140.24.7.128 | ---------- | m2 |
| /26 | 140.24.7.192 | ---------- | m3 |
| /0 | 0.0.0.0 | Default | m4 |

Routing table for R1

| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /24 | 140.24.7.0 | ---------- | m0 |
| /0 | 0.0.0.0 | Default | m1 |

Routing table for R2

**Hình 4.4.8** ***Tương thích mặt nạ dài nhất***



Routing table for R2

| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /26 | 140.24.7.192 | ---------- | m1 |
| /24 | 140.24.7.0 | ---------- | m0 |
| /?? | ??????? | ????????? | m1 |
| /0 | 0.0.0.0 | Default | m2 |

| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /26 | 140.24.7.0 | ---------- | m0 |
| /26 | 140.24.7.64 | ---------- | m1 |
| /26 | 140.24.7.128 | ---------- | m2 |
| /0 | 0.0.0.0 | Default | m3 |

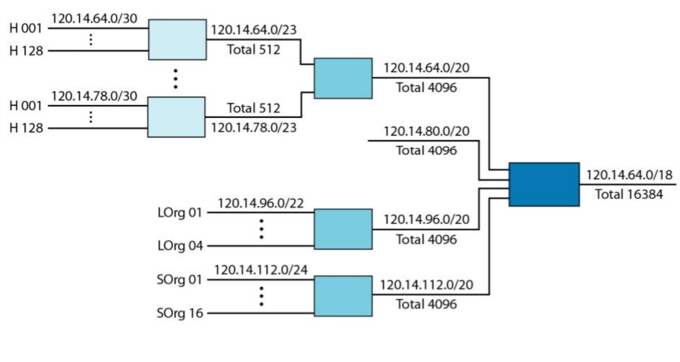
Routing table for R1

| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /26 | 140.24.7.192 | ---------- | m0 |
| /?? | ??????? | ????????? | m1 |
| /0 | 0.0.0.0 | Default | m2 |

Routing table for R3

Example 1:
As an example of hierarchical routing, let us consider Figure 4.4.9. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into four subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs; the second subblock is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.
The first local ISP has divided its assigned subblock into 8 smaller blocks and assigned each to a small ISP. Each small ISP provides services to 128 households, each using four addresses.
The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations. There is a sense of hierarchy in this configuration. All routers in the Internet send a packet with destination address 120.14.64.0 to 120.14.127.255 to the regional ISP.
The third local ISP has divided its block into 16 blocks and assigned each block to a small organization. Each small organization has 256 addresses, and the mask is /24.

**Figure 4.4.9** *Hierarchical routing with ISPs*



## Unicast Routing Protocols

- A routing table can be either static or dynamic.
- A static table: manual entries.
- A dynamic table: is updated automatically when there is a change somewhere in the Internet.
- A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.

**Figure 4.4.12** *Autonomous systems*

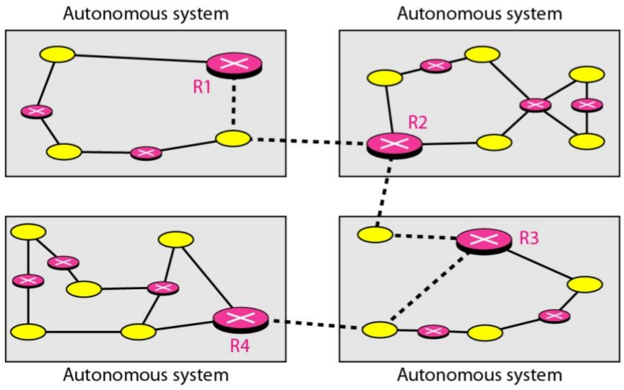
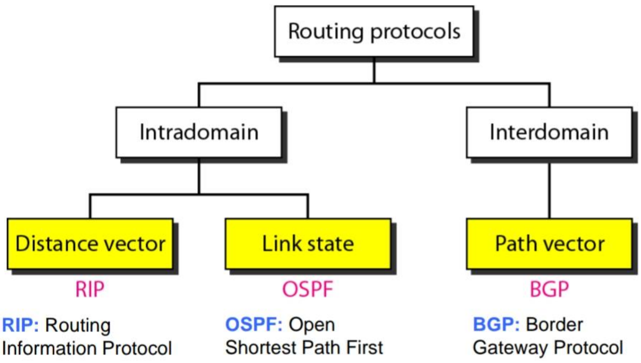


**Figure 4.4.13** *Popular routing protocols*



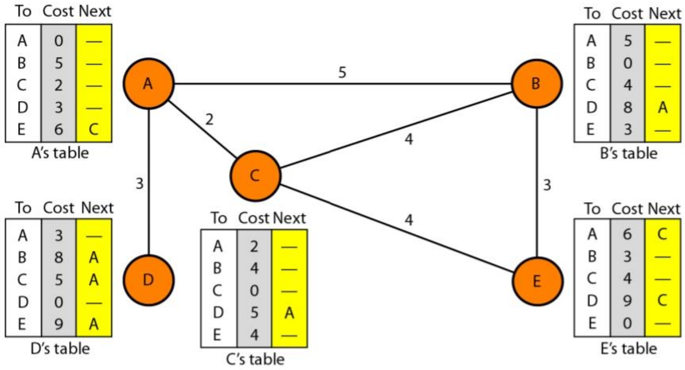
**RIP:** Routing Information Protocol

**OSPF:** Open Shortest Path First

**BGP:** Border Gateway Protocol

**Distance Vector Routing: In** distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

**Figure 4.4.14** *Distance vector routing tables*



| To | Cost | Next |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | 6 | C |

A's table

| To | Cost | Next |
|---|---|---|
| A | 5 | — |
| B | 0 | — |
| C | 4 | — |
| D | 8 | A |
| E | 3 | — |

B's table

| To | Cost | Next |
|---|---|---|
| A | 3 | — |
| B | 8 | A |
| C | 5 | A |
| D | 0 | — |
| E | 9 | A |

D's table

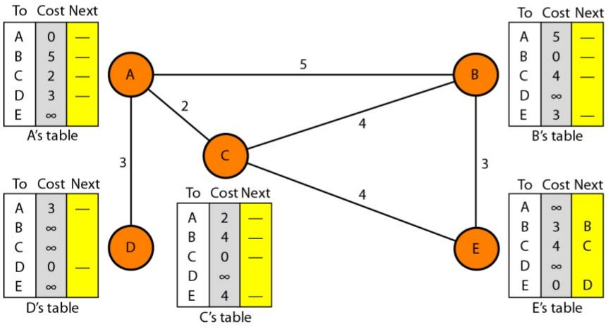
| To | Cost | Next |
|---|---|---|
| A | 2 | — |
| B | 4 | — |
| C | 0 | — |
| D | 5 | A |
| E | 4 | — |

C's table

| To | Cost | Next |
|---|---|---|
| A | 6 | C |
| B | 3 | — |
| C | 4 | — |
| D | 9 | C |
| E | 0 | — |

E's table

**Figure 4.4.15** *Initialization of tables in distance vector routing*



| To | Cost | Next |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | ∞ | |

A's table

| To | Cost | Next |
|---|---|---|
| A | 5 | — |
| B | 0 | — |
| C | 4 | — |
| D | ∞ | |
| E | 3 | — |

B's table

| To | Cost | Next |
|---|---|---|
| A | 3 | — |
| B | ∞ | |
| C | ∞ | |
| D | 0 | — |
| E | ∞ | |

D's table

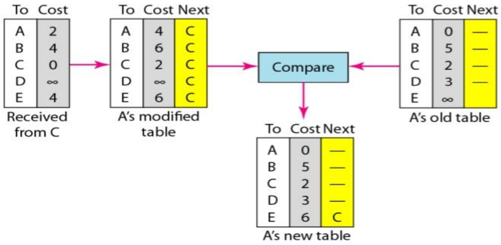
| To | Cost | Next |
|---|---|---|
| A | 2 | — |
| B | 4 | — |
| C | 0 | — |
| D | ∞ | |
| E | 4 | — |

C's table

| To | Cost | Next |
|---|---|---|
| A | ∞ | |
| B | 3 | B |
| C | 4 | C |
| D | ∞ | |
| E | 0 | D |

E's table

**Figure 4.4.16** *Updating in distance vector routing*



Received from C

| To | Cost |
|---|---|
| A | 2 |
| B | 4 |
| C | 0 |
| D | ∞ |
| E | 4 |

A's modified table

| To | Cost | Next |
|---|---|---|
| A | 4 | C |
| B | 6 | C |
| C | 2 | C |
| D | ∞ | C |
| E | 6 | C |

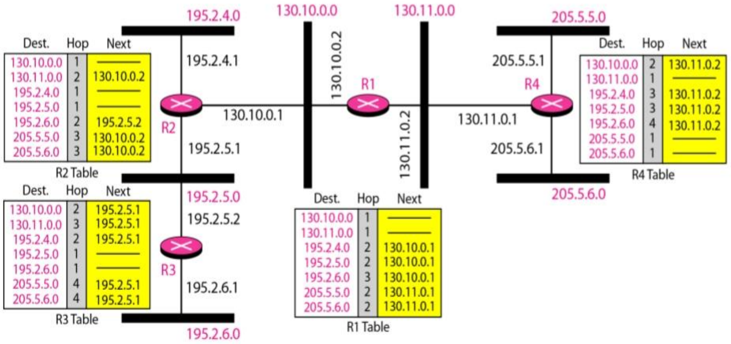
A's old table

| To | Cost | Next |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | ∞ | |

A's new table

| To | Cost | Next |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | 6 | C |

**Figure 4.4.19** *Example of a domain using RIP*



R2 Table

| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 1 | ——— |
| 130.11.0.0 | 2 | 130.10.0.2 |
| 195.2.4.0 | 1 | ——— |
| 195.2.5.0 | 1 | ——— |
| 195.2.6.0 | 2 | 195.2.5.2 |
| 205.5.5.0 | 3 | 130.10.0.2 |
| 205.5.6.0 | 3 | 130.10.0.2 |

R3 Table

| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 2 | 195.2.5.1 |
| 130.11.0.0 | 3 | 195.2.5.1 |
| 195.2.4.0 | 2 | 195.2.5.1 |
| 195.2.5.0 | 1 | ——— |
| 195.2.6.0 | 1 | ——— |
| 205.5.5.0 | 4 | 195.2.5.1 |
| 205.5.6.0 | 4 | 195.2.5.1 |

R1 Table

| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 1 | ——— |
| 130.11.0.0 | 1 | ——— |
| 195.2.4.0 | 2 | 130.10.0.1 |
| 195.2.5.0 | 2 | 130.10.0.1 |
| 195.2.6.0 | 3 | 130.10.0.1 |
| 205.5.5.0 | 2 | 130.11.0.1 |
| 205.5.6.0 | 2 | 130.11.0.1 |

R4 Table

| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 2 | 130.11.0.2 |
| 130.11.0.0 | 1 | ——— |
| 195.2.4.0 | 3 | 130.11.0.2 |
| 195.2.5.0 | 3 | 130.11.0.2 |
| 195.2.6.0 | 4 | 130.11.0.2 |
| 205.5.5.0 | 1 | ——— |
| 205.5.6.0 | 1 | ——— |

## Link State Routing

**Figure 4.4.20** *Concept of link state routing*

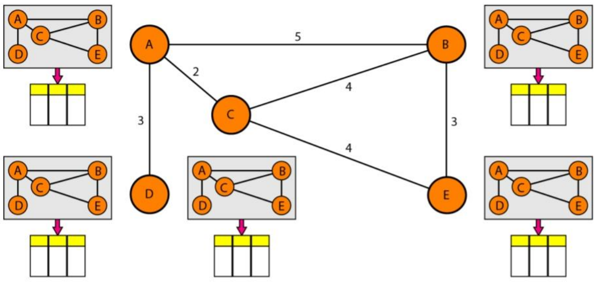


**Figure 4.4.21** *Link state knowledge*

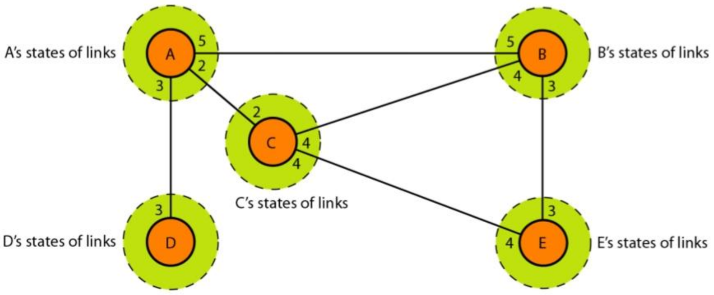
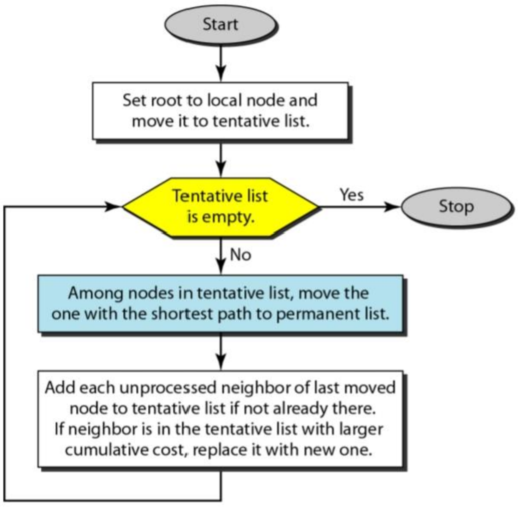
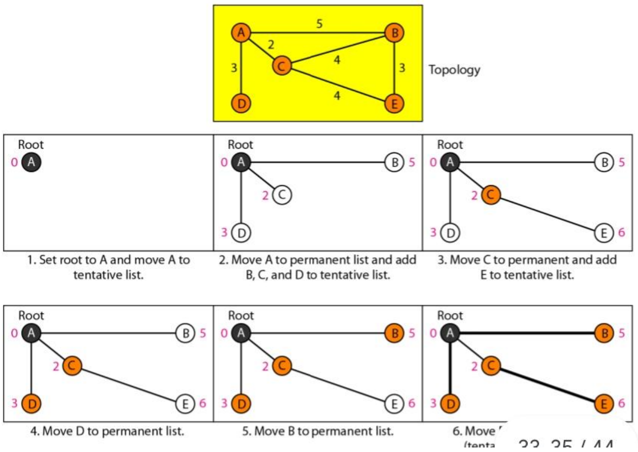


**Figure 4.4.22** *Dijkstra algorithm*



Start
Set root to local node and move it to tentative list.
Tentative list is empty.
Yes
Stop
No
Among nodes in tentative list, move the one with the shortest path to permanent list.
Add each unprocessed neighbor of last moved node to tentative list if not already there. If neighbor is in the tentative list with larger cumulative cost, replace it with new one.

**Figure 4.4.23** *Example of formation of shortest path tree*



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

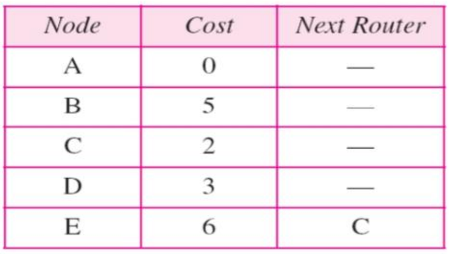
3. Move C to permanent and add E to tentative list.

4. Move D to permanent list.

5. Move B to permanent list.

6. Move
(tenta

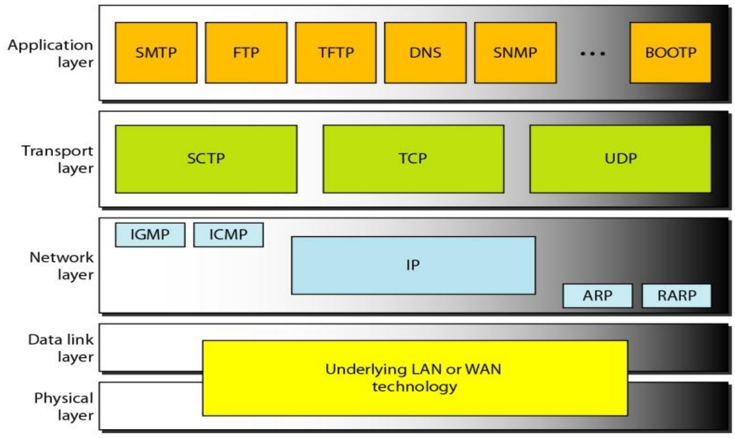
**Table 4.4.2** *Routing table for node A*



| *Node* | *Cost* | *Next Router* |
|---|---|---|
| A | 0 | — |
| B | 5 | — |
| C | 2 | — |
| D | 3 | — |
| E | 6 | C |

## Chapter 5.1: Protocols: UDP, TCP, SCTP

**Figure 5.1.8** *Position of UDP, TCP, and SCTP in TCP/IP suite*



### Process-to-process Delivery

•A process is an application program running on a host.

•The transport layer is responsible for process-to- process delivery—the delivery of a packet, part of a message, from one process to another.

•Two processes communicate in a client/server relationship.

**Figure 5.1.1** *Types of data deliveries*

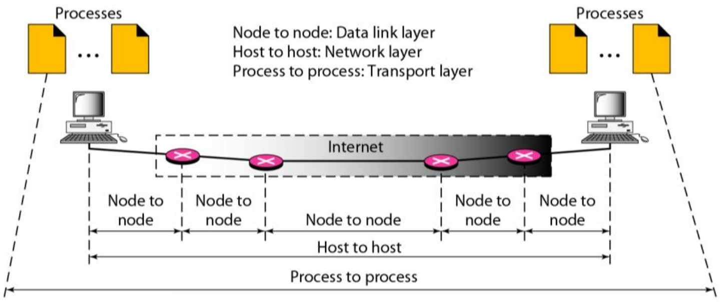


**Figure 5.1.2** *Port numbers*

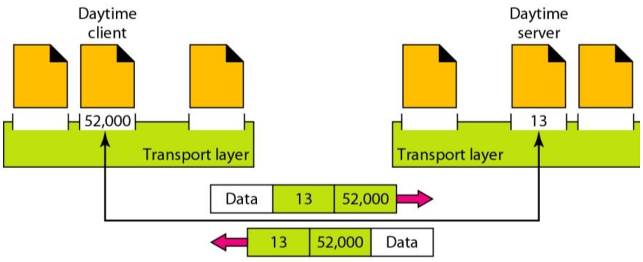


**Figure 5.1.3** *IP addresses versus port numbers*

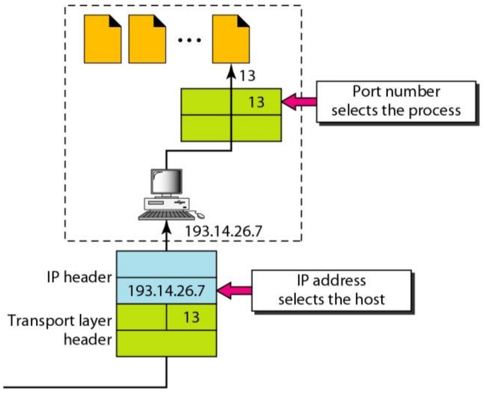


**Figure 5.1.5** *Socket address*

7-8 / 2

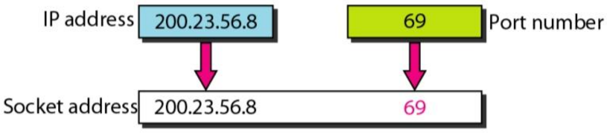


**Figure 5.1.6** *Multiplexing and demultiplexing*

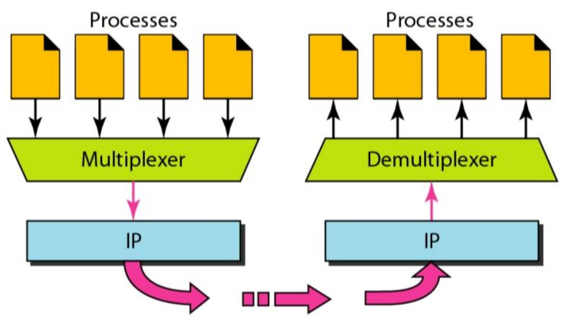
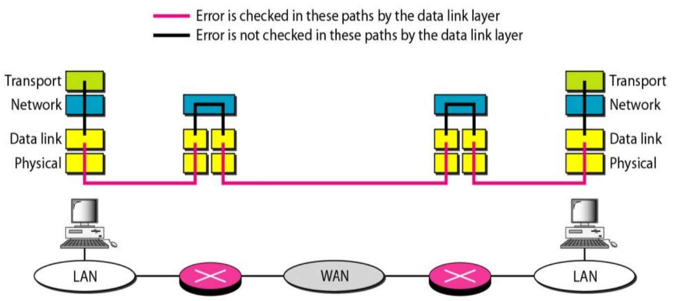


**Figure 5.1.7** *Error control*

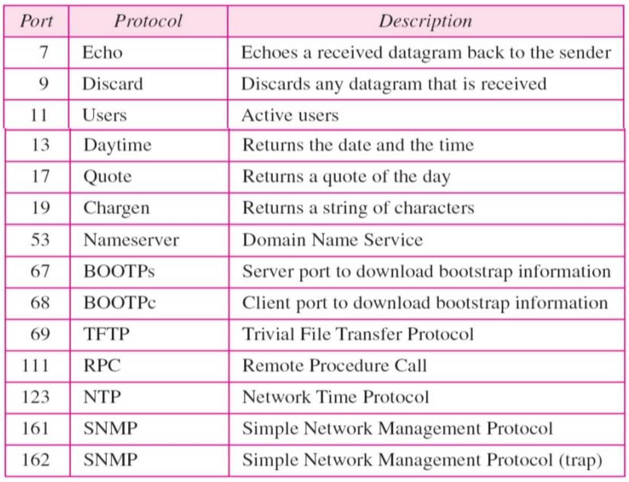

Error is checked in these paths by the data link layer
Error is not checked in these paths by the data link layer
Transport
Network
Data link
Physical
Transport
Network
Data link
Physical
LAN
WAN
LAN

**User Datagram Protocol (Udp)**

•The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol.

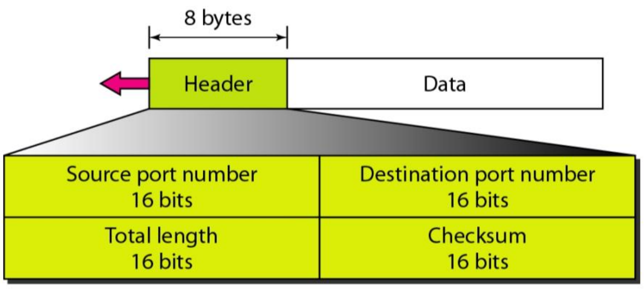
•It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

**Table 5.1.1** *Well-known ports used with UDP*



| *Port* | *Protocol* | *Description* |
|---|---|---|
| 7 | Echo | Echoes a received datagram back to the sender |
| 9 | Discard | Discards any datagram that is received |
| 11 | Users | Active users |
| 13 | Daytime | Returns the date and the time |
| 17 | Quote | Returns a quote of the day |
| 19 | Chargen | Returns a string of characters |
| 53 | Nameserver | Domain Name Service |
| 67 | BOOTPs | Server port to download bootstrap information |
| 68 | BOOTPc | Client port to download bootstrap information |
| 69 | TFTP | Trivial File Transfer Protocol |
| 111 | RPC | Remote Procedure Call |
| 123 | NTP | Network Time Protocol |
| 161 | SNMP | Simple Network Management Protocol |
| 162 | SNMP | Simple Network Management Protocol (trap) |

**Figure 5.1.9** *User datagram format*



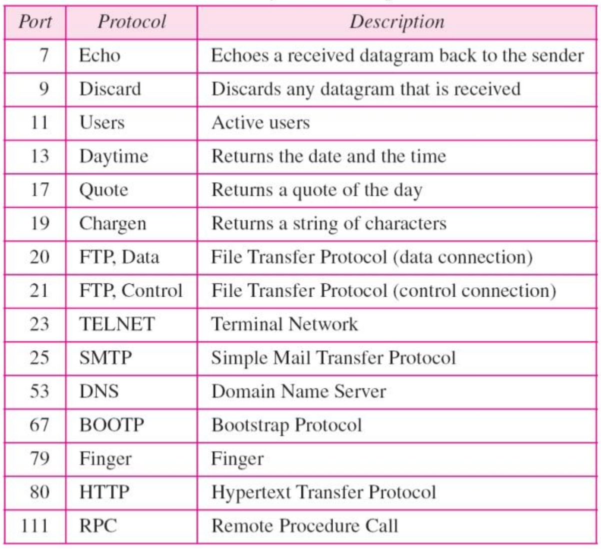
UDP length = IP length – IP header's length

**TCP**

TCP is a connection-oriented protocol. It creates a virtual connection between two TCPs to send data.

In addition, TCP uses flow and error control mechanisms at the transport level.

**Table 5.1.2** *Well-known ports used by TCP*



| *Port* | *Protocol* | *Description* |
|---|---|---|
| 7 | Echo | Echoes a received datagram back to the sender |
| 9 | Discard | Discards any datagram that is received |
| 11 | Users | Active users |
| 13 | Daytime | Returns the date and the time |
| 17 | Quote | Returns a quote of the day |
| 19 | Chargen | Returns a string of characters |
| 20 | FTP, Data | File Transfer Protocol (data connection) |
| 21 | FTP, Control | File Transfer Protocol (control connection) |
| 23 | TELNET | Terminal Network |
| 25 | SMTP | Simple Mail Transfer Protocol |
| 53 | DNS | Domain Name Server |
| 67 | BOOTP | Bootstrap Protocol |
| 79 | Finger | Finger |
| 80 | HTTP | Hypertext Transfer Protocol |
| 111 | RPC | Remote Procedure Call |

**Figure 5.1.13** *Stream delivery*

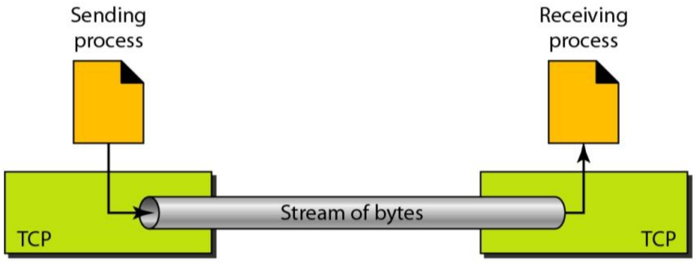


**Figure 5.1.14** *Sending and receiving buffers*

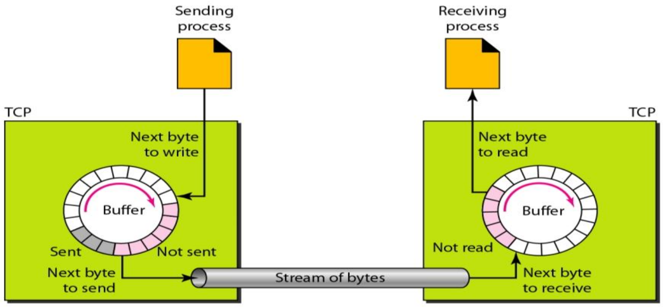
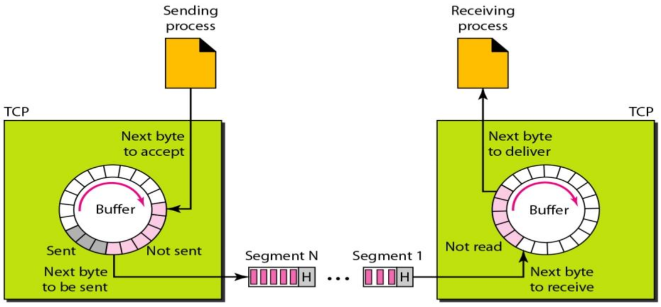
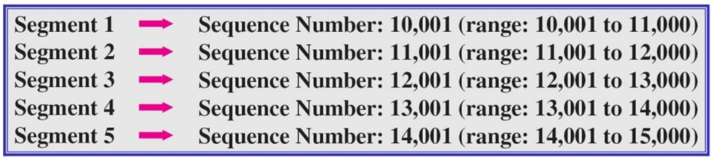


**Figure 5.1.15** *TCP segments*



***The following shows the sequence number for each segment:***



| | | |
|---|---|---|
| Segment 1 | ➡ | Sequence Number: 10,001 (range: 10,001 to 11,000) |
| Segment 2 | ➡ | Sequence Number: 11,001 (range: 11,001 to 12,000) |
| Segment 3 | ➡ | Sequence Number: 12,001 (range: 12,001 to 13,000) |
| Segment 4 | ➡ | Sequence Number: 13,001 (range: 13,001 to 14,000) |
| Segment 5 | ➡ | Sequence Number: 14,001 (range: 14,001 to 15,000) |

**Figure 5.1.16** *TCP segment format*

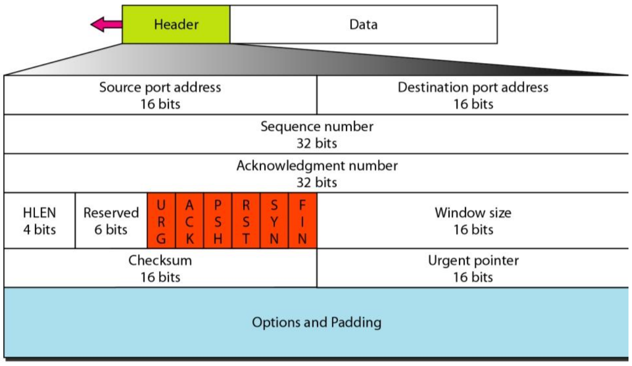


**Figure 5.1.17** *Control field*

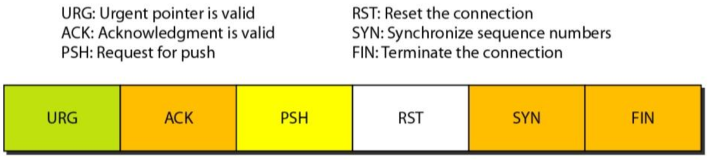
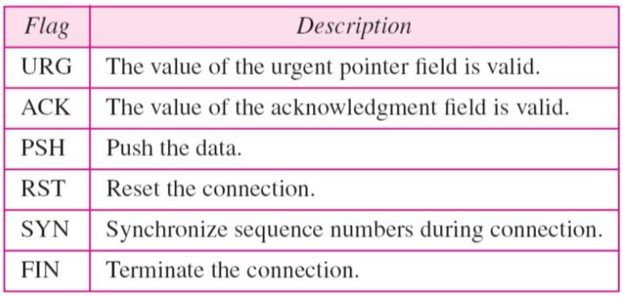

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection
URG
ACK
PSH
RST
SYN
FIN

**Table 5.1.3** *Description of flags in the control field*



| *Flag* | *Description* |
|---|---|
| URG | The value of the urgent pointer field is valid. |
| ACK | The value of the acknowledgment field is valid. |
| PSH | Push the data. |
| RST | Reset the connection. |
| SYN | Synchronize sequence numbers during connection. |
| FIN | Terminate the connection. |

## Chapter 6: Application Layer

**Dịch vụ truyền File FTP (**File Transfer Protocol) là một trong những dịch vụ sớm nhất ứng dụng giao thức TCP/ IP. FTP cho phép người dùng thực hiện các chức năng: Sao chép, Đổi tên, Xóa file, Tạo thư mục …..ở một hệ thống ở xa.

Hệ thống FTP ở xa thường yêu cầu người dùng cung cấp định danh ID và mật khẩu trước khi truy nhập hệ thống. Các máy chủ thường cung cấp hai dạng dịch vụ truy nhập.

* Truy nhập vào các file công cộng dùng chung qua tài khoản ẩn danh (Anonymous).

* Truy nhập vào các file riêng chỉ dành cho những người sử dụng với quyền truy nhập ở mức hệ thống.

**Network File System (NFS)**

Hệ tập tin mạng (Network File System-NFS) cung cấp việc truy xuất trực tuyến các tập tin dùng chung. Người sử dụng có thể thực hiện một chưng trình ứng dụng bất kỳ và sử dụng bất kỳ một tập tin nào trong việc xuất nhập. Bn thân tên các tập tin không cho biết chúng cục bộ hay ở xa. NFS là một RPC (Remote Procedure Call )

**Domain Name Service (DNS)**

Đối với những người truy nhập Internet, việc nhớ nhiều địa chỉ IP cùng một lúc là rất khó. Do đó, các nhà thiết kế tạo nên những tên dễ nhớ. Người dùng muốn truy nhập đến địa chỉ nào thì chỉ việc gõ bàn phím những tên đó vào. Tuy nhiên, giao thức lớp mạng IP chỉ có thể hiệu và làm việc được với địa chỉ IP. Do vậy cần có sự chuyển đổi qua lại giữa tên và địa chỉ IP. Việc chuyển đổi tên thành địa chỉ được thực hiện qua hệ thống tên miền (Domain Name System – DNS). Hệ thống DNS thực chất là những CSDL (DNS database) chứa tên và địa chỉ tưng ứng cùng với các thông tin khác đi kèm.

**Dịch vụ Mail:** là dịch vụ thư điện tử. Để dịch vụ Mail hoạt động được thì phải đảm bảo 2 thành phần: Mail Server, Mail Client.