



IF2230 Jaringan Komputer

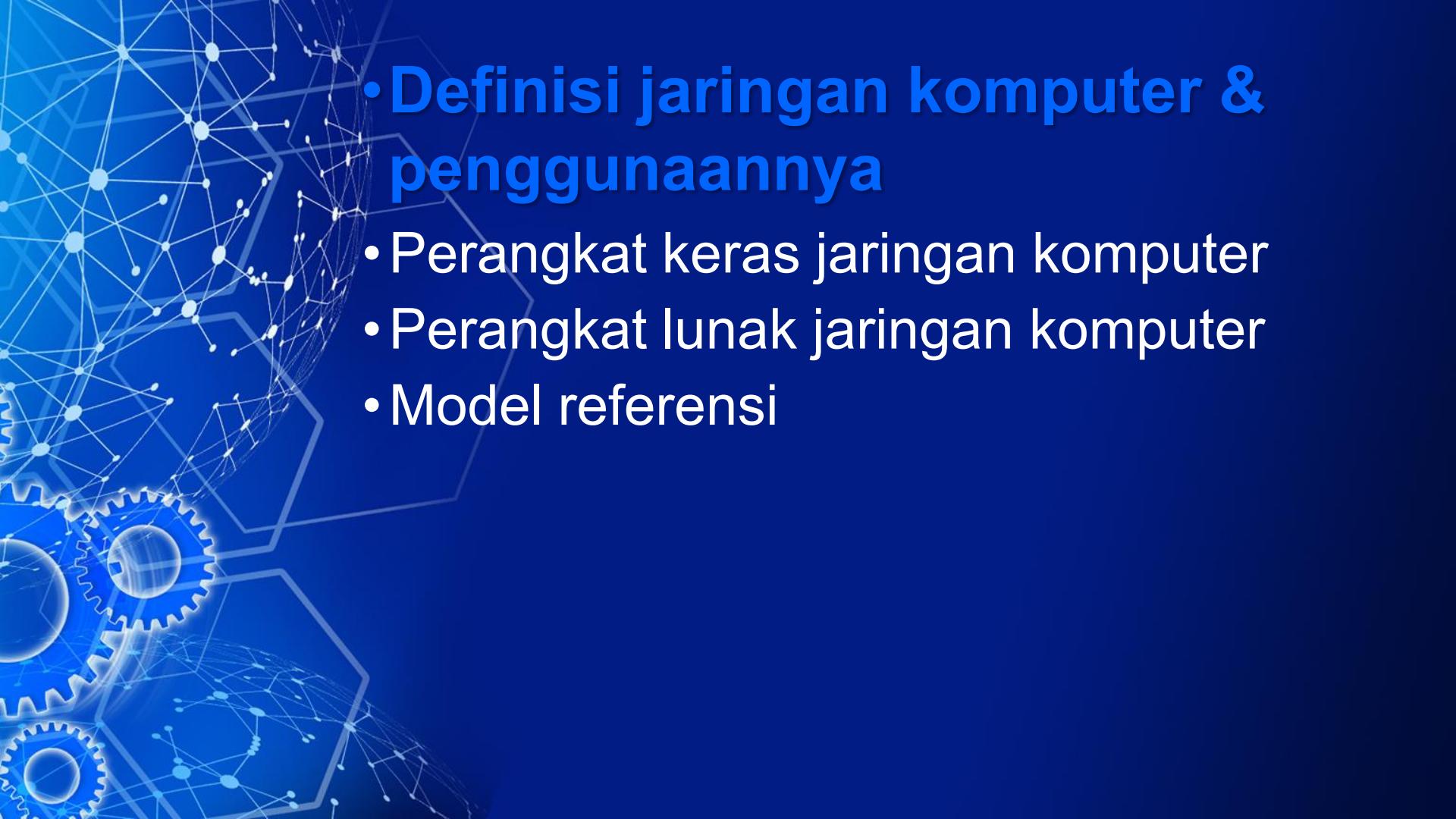
Introduction

Robithoh Annur
Andreas Bara Timur
Monterico Adrian



Pengantar jaringan komputer

- Definisi jaringan komputer & penggunaannya
- Perangkat keras jaringan komputer
- Perangkat lunak jaringan komputer
- Model referensi

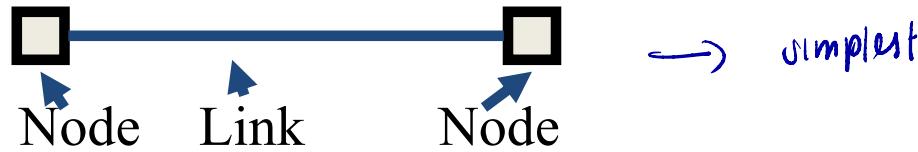
- 
- Definisi jaringan komputer & penggunaannya
 - Perangkat keras jaringan komputer
 - Perangkat lunak jaringan komputer
 - Model referensi

- 
- Collection of nodes and links that connect them
 - This is vague. Why? Consider different networks:
 - Internet
 - Andrew
 - Telephone
 - Your house
 - Others – sensor nets, cell phones, ...
 - Jaringan komputer: sekumpulan komputer yang terhubung dengan saluran komunikasi

How to Draw a Network



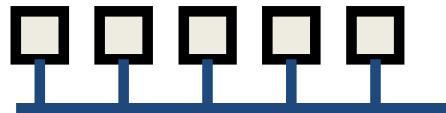
Basic Building Block: Links



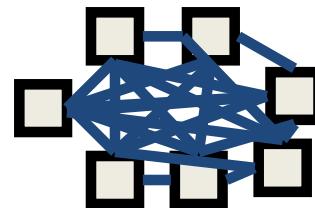
- Electrical questions
 - Voltage, frequency, ...
 - Wired or wireless?
- Link-layer issues: How to send data?
 - When to talk – can either side talk at once?
 - What to say – low-level format?
- Okay... what about more nodes?

Basic Building Block: Links

- ... But what if we want more hosts?



One wire



Wires for everybody!

how to connect?
if we only have
2 nodes,
ga bisa byk
linknya (?)

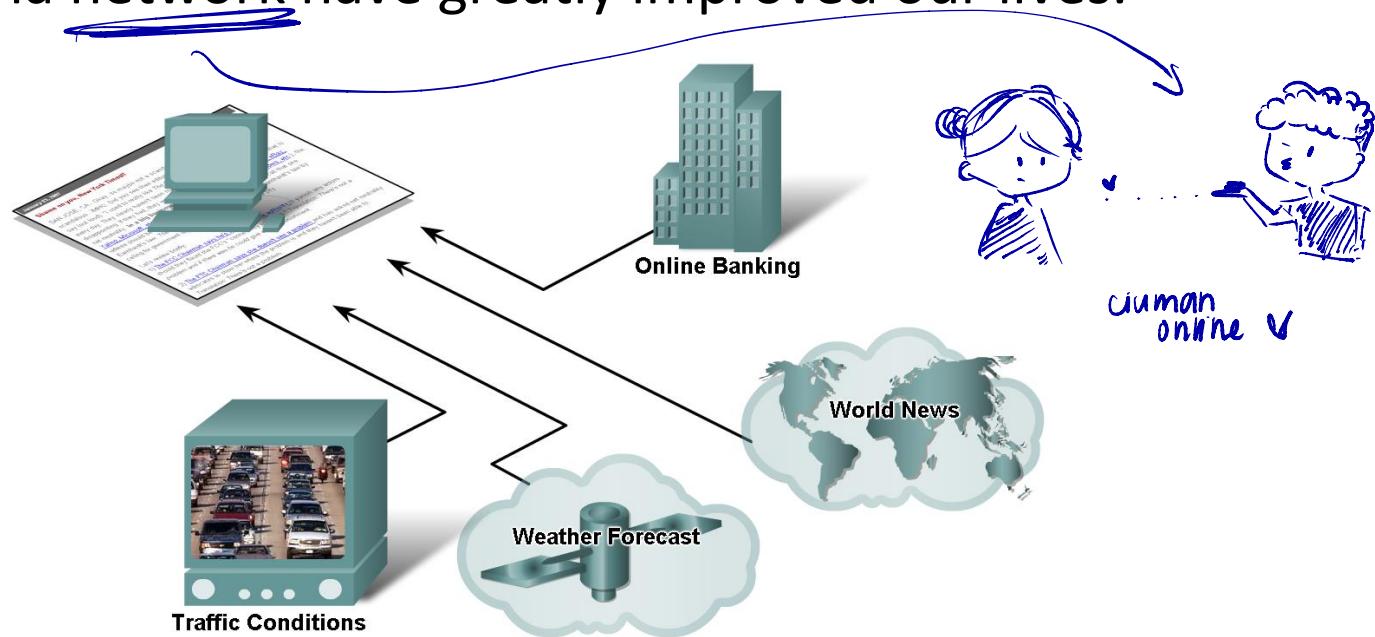
→ bertumbuh + berkembang

- Scalability?!

tapi
kl sgt banyak jd tp bisa collapse ^

How Networks Impact Daily Life

- The benefits of instantaneous communication via network have greatly improved our lives.



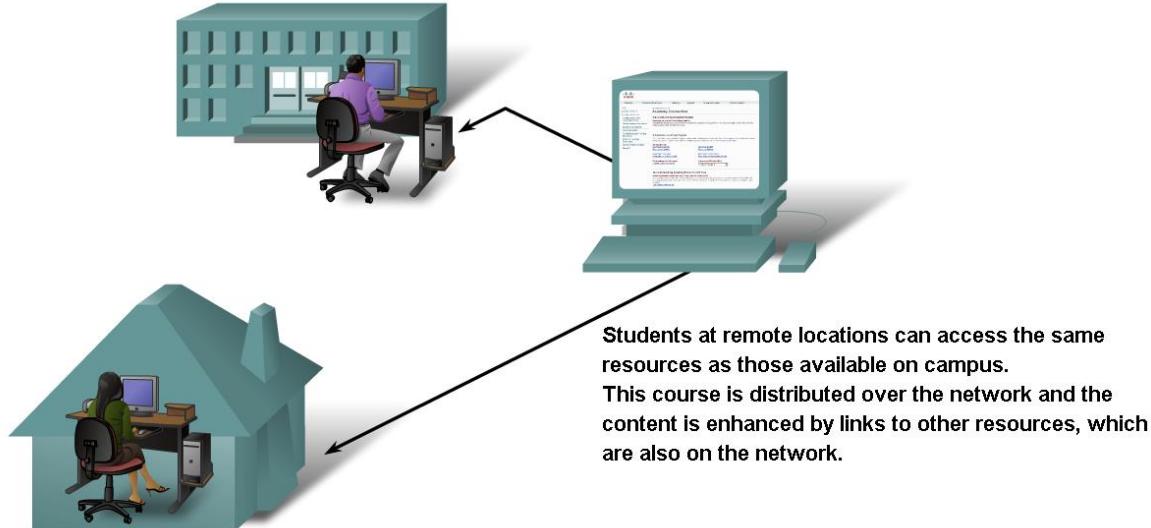
How Networks Impact Daily Life

- Communication over a network has also greatly changed the way we work.



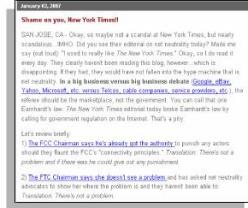
How Networks Impact Daily Life

- and also changed the way we study....

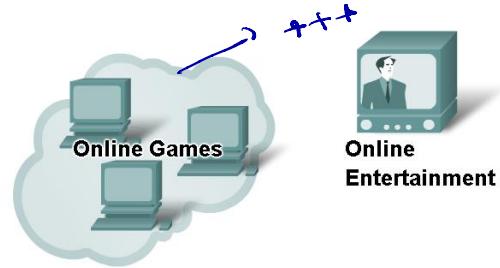


How Networks Impact Daily Life

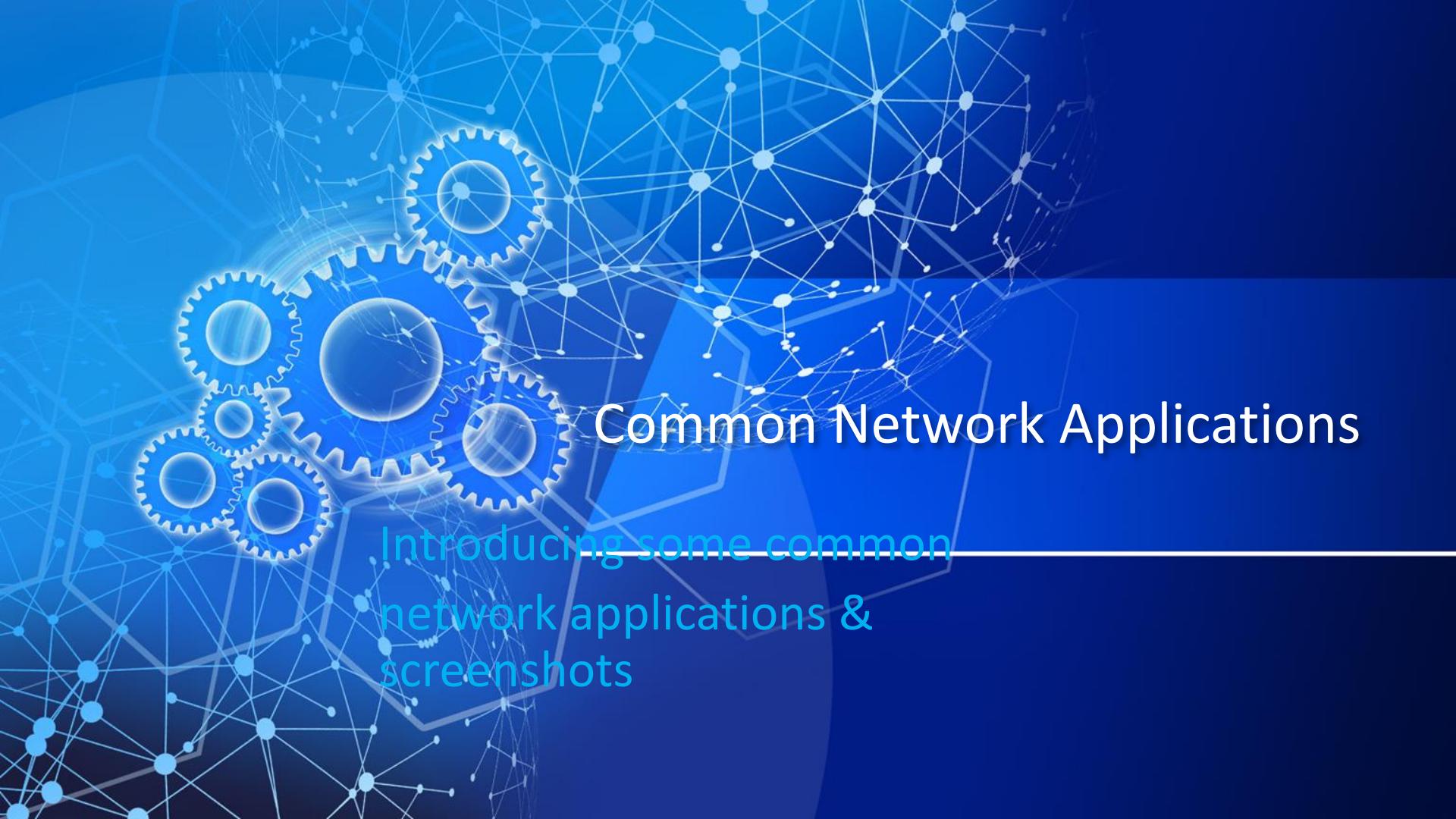
- the way we entertain ourselves....



Online Interest Groups



Instant Messaging

The background features a complex network of interconnected nodes and lines in shades of blue, resembling a digital or industrial gear system. A large, semi-transparent hexagonal shape is centered on the right side of the slide.

Common Network Applications

Introducing some common
network applications &
screenshots



Internet & Daily Life

- When you access the Internet, do you use the following applications?
 - Internet Explorer, Firefox
 - Google Mail, Hotmail, Yahoo Mail,
 - Yahoo Messenger, Skype, Whatsapp
 - Facebook, Google+, LinkedIn
 - Online Games (The Secret World, Final Fantasy XIV)
 - and others....
- Since these applications require the access to the Internet, we refer these applications as network applications.

SP yg
m/n Pake
dk

Web Browser

- By definition:
 - A web browser is a network application for retrieving, presenting, and traversing information resources on the World Wide Web.
- The following network applications (web browsers) do the same thing as Internet Explorer:
 - Google Chrome
 - Mozilla Firefox
 - Netscape
 - Opera
 - Safari
 - Camino
 - Konqueror
 - and others....



Netscape 2.0
is The browser that
make the Internet
or World Wide Web
into a global
phenomena back in
1996.

Other Web Browsers' Screenshots

Safari



Google Chrome



Other Web Browsers' Screenshots

Firefox



Opera

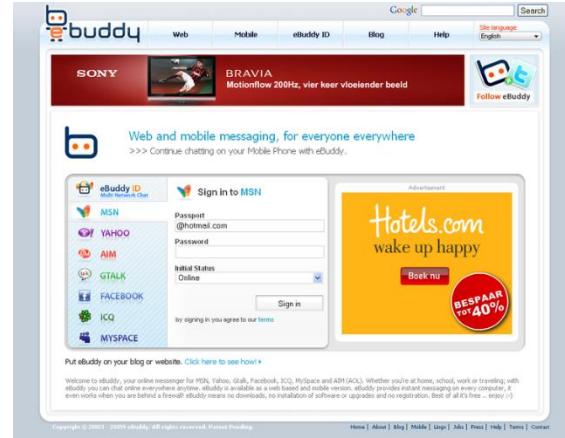
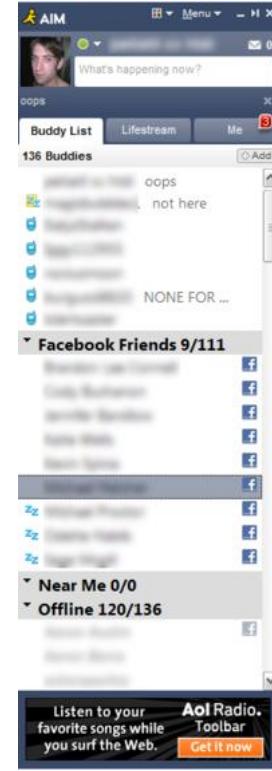
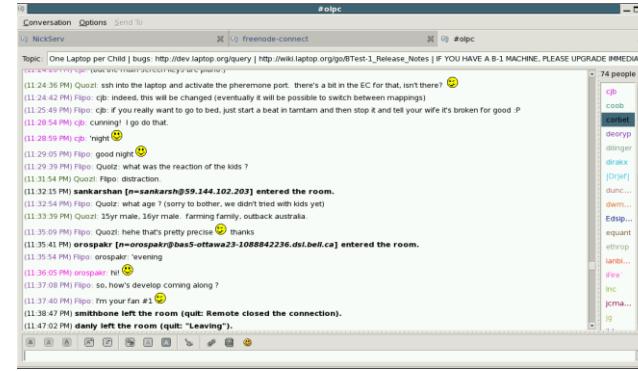
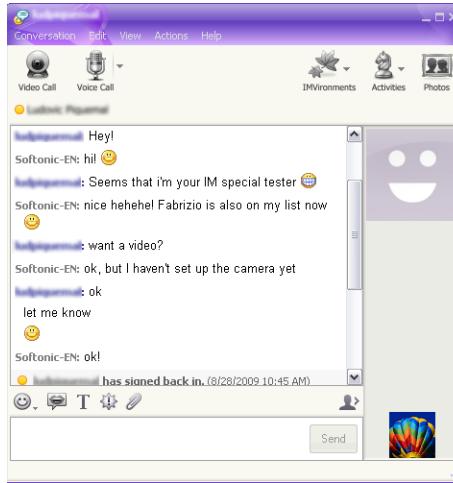


Instant Messaging

- Another popular network application beside the web browsers is Instant Messaging.
- Instant messaging (IM) is a form of real-time direct text-based communication between two or more people using personal computers (PCs) or other devices.
 - More advanced instant messaging software also allow live voice or video calling.
- Some of the popular instant messaging software:
 - ICQ
 - Skype
 - Yahoo! Messenger
 - Facebook Messenger
 - Ebuddy
 - Line
 - WhatsApps



Screenshots of Other Instant Massagers



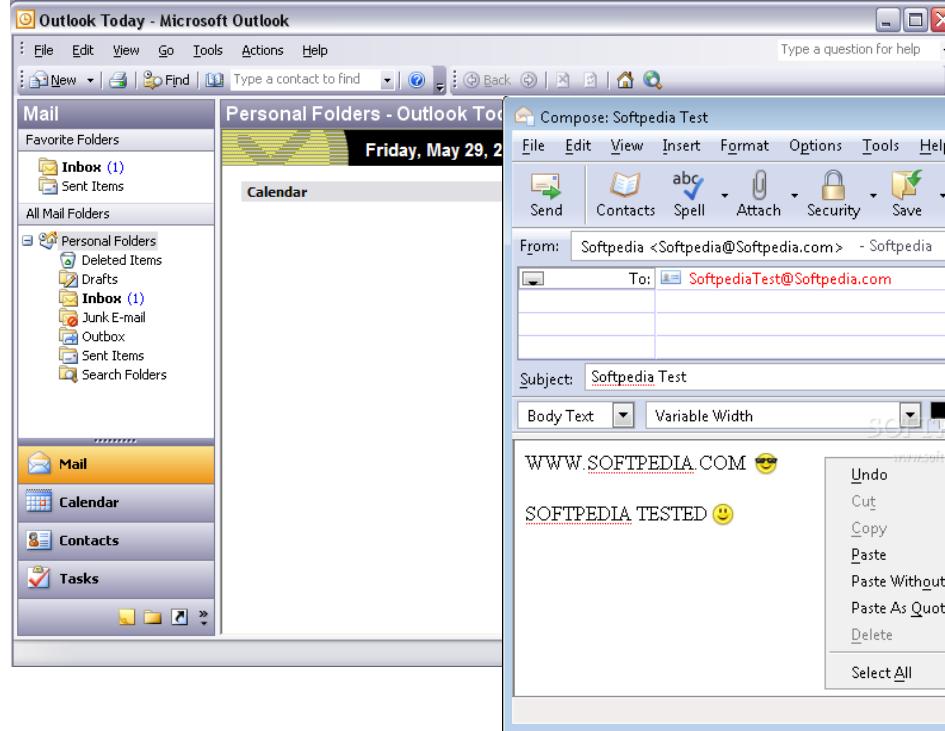


Email

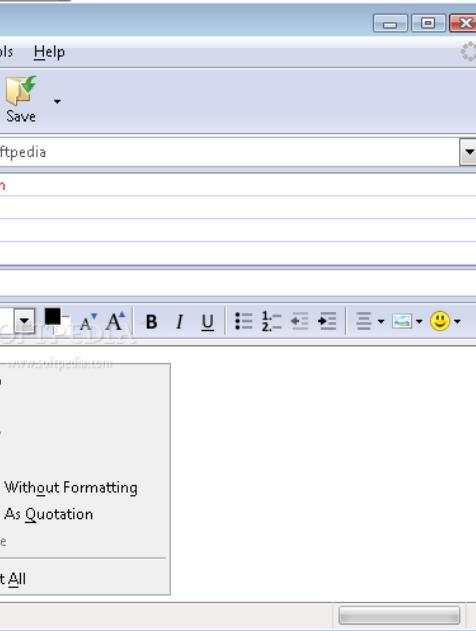
- For all business activities via the Internet, any network manager will tell you that the Email is the most important network application.
- Email remains as the oldest and still the most frequently used network communications in the Internet.

Screenshots of Email Software

Microsoft's Outlook



Mozilla Thunderbird





Other Network Applications

- Besides Web Browser, Instant Messaging & Email, there are other network applications that do:
 - File transfer
 - (Ws_FTP, Free Download Manager)
 - Remote login
 - (PuTTY, Terra Term)
 - Accessing remote database
 - (SQuirreL SQL Client)
 - Internet Relay Chat
 - (mIRC)
 - and other ...



Topik

Definisi jaringan komputer & penggunaannya
Perangkat keras jaringan komputer
Perangkat lunak jaringan komputer
Model referensi



Pengelompokan Jaringan

- Pengelompokan berdasarkan jenis transmisi:
 - Broadcast links
 - Point-to-point links
- Pengelompokan berdasarkan skala

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	
1000 km	Continent	
10,000 km	Planet	The Internet



Personal area networks

- A PAN is a computer network that used for communication among computer devices, including telephones and personal digital assistants, in proximity to an individual's body
- Bluetooth, ZigBee

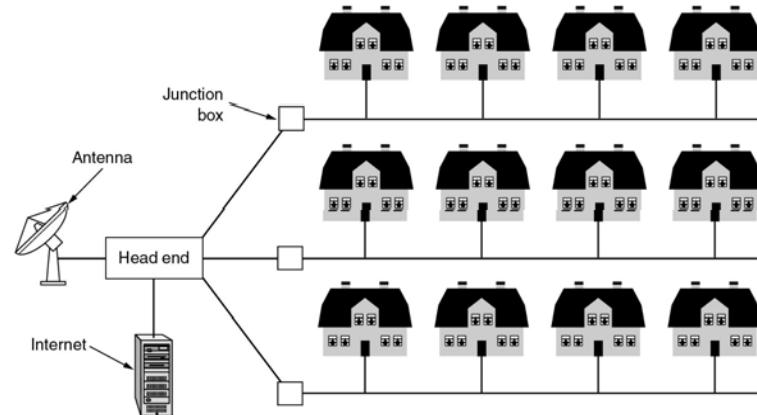
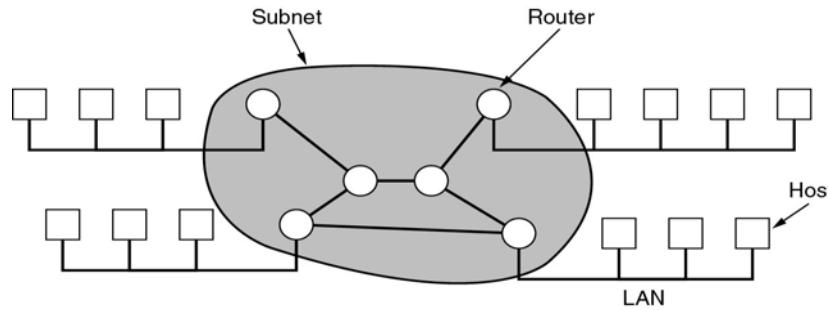


Local area networks

- A LAN is a computer network covering a small physical area, like a home, office, or small groups of buildings, such as a school, or an airport.
- Menghubungkan komputer2 dengan peralatan lain (printer, data, files) untuk resource sharing
- Medium broadcast kabel:
 - Kecepatan awal 10 Mbps atau 100 Mbps. Teknologi terakhir mencapai 100 Gbps.
- Medium broadcast wireless (WLAN)
 - Kecepatan awal 11 Mbps, 54 Mbps. Teknologi terakhir mencapai 1.3Gbs.
- Standards:
 - <http://standards.ieee.org/getieee802/portfolio.html>

Metropolitan Area Network

- A MAN is a large computer network that usually spans a city or a large area

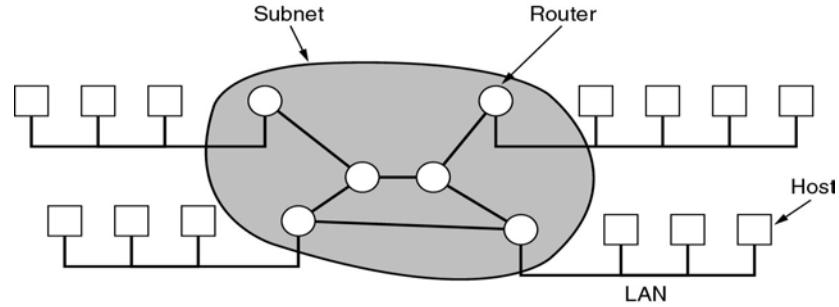


- Router: node/device/komputer yang menyediakan layanan komunikasi
- Subnet: kumpulan router (definisi umum). Subnet dapat pula berarti sekelompok node jaringan yang memiliki alamat IP awal sama.
- Relation between hosts on LANs and the subnet.

- A metropolitan area network based on cable TV.

Wide Area Networks

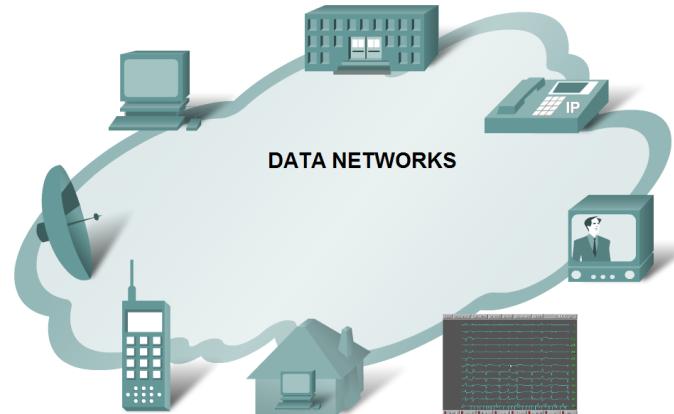
- A WAN is a computer network that covers a broad area (across metropolitan, regional, or national boundaries)
- Mencakup area geografis yang luas
- Umumnya terdiri atas banyak koneksi point-to-point
- Komputer/LAN terhubung dengan WAN melalui subnet
- Subnet dapat berbasis paket maupun circuit
 - Paket: pengguna berebut bandwidth yang ada
 - Circuit: pengguna memiliki jatah bandwidth tetap



- Router: node/device/komputer yang menyediakan layanan komunikasi
- Subnet: kumpulan router (definisi umum). Subnet dapat pula berarti sekelompok node jaringan yang memiliki alamat IP awal sama.
- Relation between hosts on LANs and the subnet.

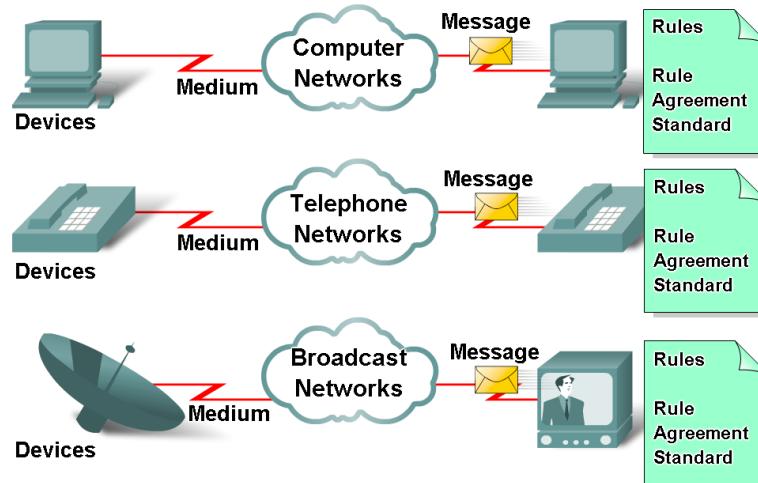
Network as a Platform

- The function of a network is to serve as a platform for communications between end users.
- End users can be in the form of:
 - Servers & clients
 - Smart phones & other mobile devices
 - PCs and webcam



Elements of a Network

- All networks have four basic elements in two categories:
 - Hardware: (i) Devices, (ii) Medium,
 - Software: (iii) Message, (iv) Rules/Agreement.



Hardware: Devices & Medium

- Devices
 - will be explained in the later slides
- Medium
 - this is the channel over which a message travels
 - the following diagram shows some examples of a network medium

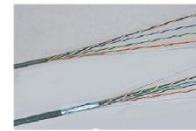
Network Media



Copper



Fiber Optics



Wireless





Software: Message & Rules

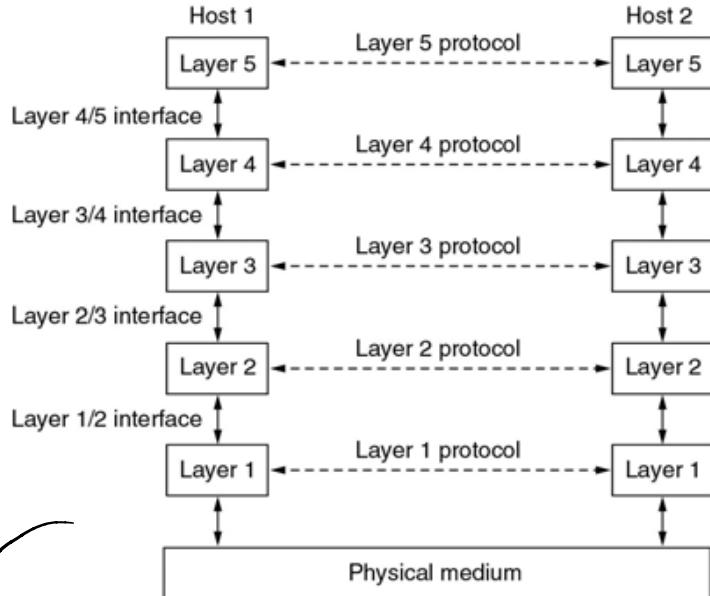
- Message:
 - Generic term that encompasses web pages, emails, instant messages, telephone calls, video, multimedia streaming, etc.
- Rules:
 - Addressing schemes (IP, MAC address, port numbers)
 - Protocols



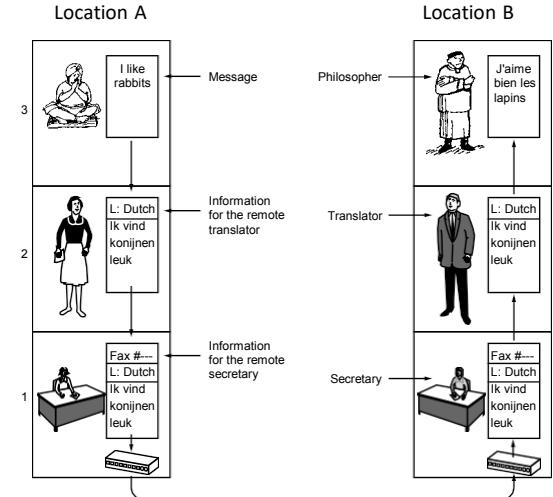
Network Protocol

- Networks diorganisasikan ke dalam sederetan layer-layer
- Setiap layer memberikan layanan kepada layer di atasnya, dan menggunakan layanan dari layer di bawahnya.
- Layer N pada satu mesin berkomunikasi dengan layer N pada mesin lainnya (disebut sebagai peer), menggunakan aturan/protokol tertentu. Data dikirimkan melalui layer di bawahnya.
- Antarmuka antar layer yang bersebelahan mendefinisikan operasi primitif dan layanan yang disediakan

Protocol Hierarchies



setiap layer punya fungsi nya masing²,
tapi dia tetap harus serve layer
diatasnya.



- The philosopher-translator-secretary architecture.

- Definisi jaringan komputer & penggunaannya
- Perangkat keras jaringan komputer
- Perangkat lunak jaringan komputer
- **Model referensi**



Reference Models

- The OSI Reference Model
- The TCP/IP Reference Model
- A Comparison of OSI and TCP/IP
- A Critique of the OSI Model and Protocols
- A Critique of the TCP/IP Reference Model

OSI Layered Model

- OSI was developed back in 1977, by International Organization for Standardization (ISO).
- OSI architecture has begun with two major components:
 - an abstract model of networking, with specific functions at each layer,
 - a set of specific protocols associated with a particular layer.

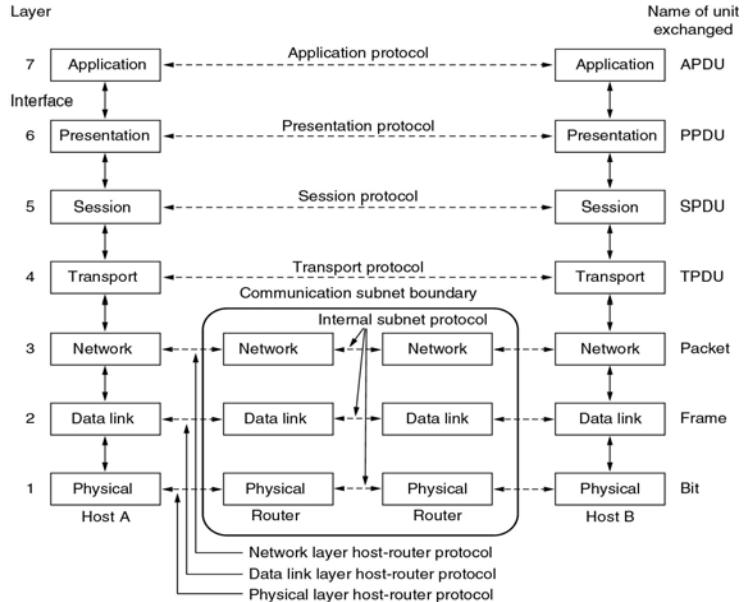
		OSI Model	
	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation and encryption
		5. Session	Interhost communication
Media layers	Segment	4. Transport	End-to-end connections and reliability
	Packet	3. Network	Path determination and logical addressing
	Frame	2. Data Link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

(fokus ke
basically
the nodes
)

masing ada yg
speednya ... bits/s
gitu berarti dan di
physical

The OSI Reference Model

- ISO menetapkan model Open Systems Interconnection untuk membantu pembangunan implementasi jaringan yang dapat berinteroperasi
- Aspek/masalah komunikasi dibagi menjadi 7 bagian yang lebih kecil, sehingga dapat lebih mudah dikelola, dengan membuat layer-layer





OSI layer 1 – physical layer

- Layer 1 mengatur spesifikasi electrical, mechanical, procedural dan functional untuk:
 - Aktivasi sambungan fisik antar end systems
 - Pengelolaan sambungan fisik antar end systems
 - Deaktivasi sambungan fisik antar end systems
- Contoh:
 - Mengatur level tegangan
 - Timing sinyal
 - Data rate fisik
 - Jarak maksimum transmisi
 - konektor



OSI layer 2 – data link layer

- Layer 2 menyediakan layanan transmisi data yang bebas dari error antar 2 node yang tersambung melalui physical layer
- Layer ini memecah data dari layer network menjadi frame-frame, dan mengirimkannya node lainnya yang kemudian menggabungkannya kembali. Layer ini menangani:
 - Frame acknowledgements
 - Error detection & correction
 - Flow control
 - Medium access



OSI layer 3 – network layer

- Layer 3 mengontrol bagaimana sebuah paket dapat diteruskan dari komputer asal ke tujuan dalam sebuah jaringan. Layer ini mengatur:
 - Penentuan rute paket
 - Congestion control/pengendalian kemacetan
 - Informasi untuk accounting
 - Menangani masalah interkoneksi antara subnet yang heterogen (antar LAN & WAN yang menggunakan protokol yang beragam)



OSI layer 4 – transport layer

- Layer 4 adalah layer end-to-end yang paling bawah antara aplikasi sumber dan tujuan
- Layer ini menyediakan end-to-end flow control, end-to-end error detection & correction, dan mungkin juga menyediakan congestion control tambahan



OSI layer 5 – session layer

- Layer ini menyediakan
 - dialogue control:
 - Siapa giliran berbicara/mengirim data
 - Token management
 - Siapa yang memiliki akses ke resource bersama
 - Sinkronisasi data
 - Apa status terakhir sebelum link putus



OSI layer 6 – presentation layer

- Berkaitan dengan sintaks dan semantik data yang dikirimkan (bukan lagi masalah transmisi data)
- Menyediakan abstraksi data yang seragam sehingga dapat digunakan untuk komunikasi data antar komputer yang heterogen

yg divent voice, text, images, etc



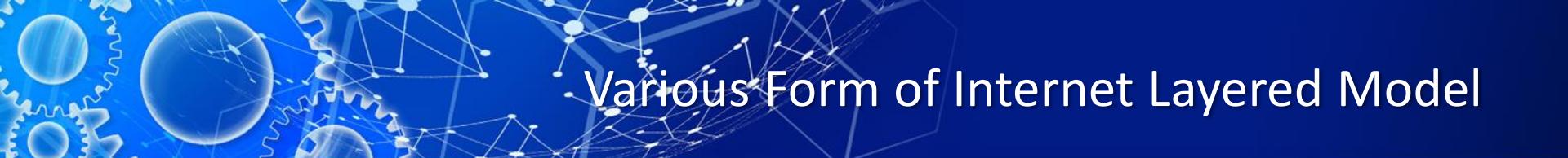
OSI layer 7 – application layer

- Aplikasi yang menggunakan jaringan:
 - Network terminal/telnet
 - File transfer
 - E-mail
 - Newsgroup
 - Web
 - Directory lookup
 - Information retrieval/searching



Terminologi OSI

- Elemen yang aktif dalam setiap layer disebut sebagai **entities** (dapat berupa hardware maupun software)
- Entities yang berada dalam layer yang sama pada mesin yang berbeda disebut sebagai peer entities. Data dikirim antar entities dalam satuan yang disebut sebagai **Protocol Data Units (PDU)**
- Entities pada layer N mengimplementasikan layanan yang digunakan layer N+1. Layer N disebut sebagai **service provider**, layer N+1 disebut sebagai **service user**.



Various Form of Internet Layered Model

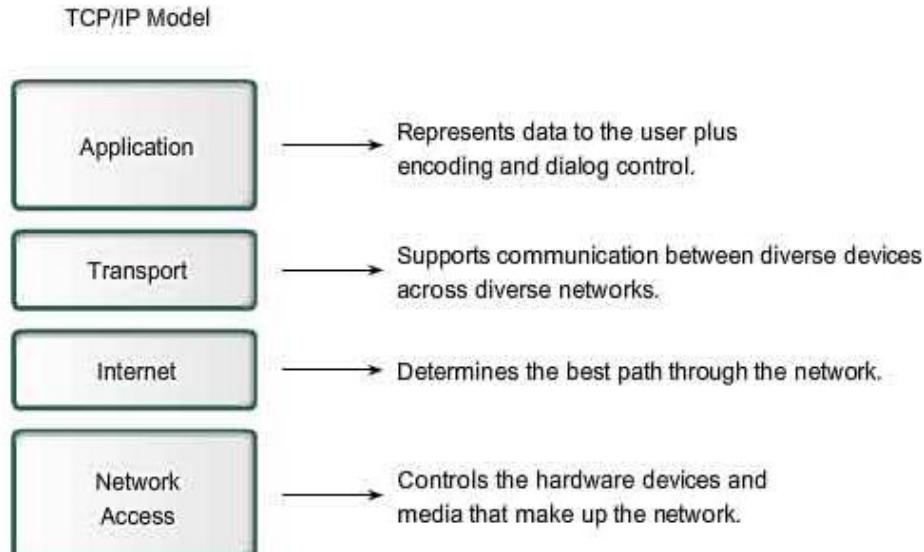
- The following table shows the layer model names and the number of layers of Internet model (or TCP/IP model) presented in the textbooks in today's university computer networking courses.

Kurose ^[7] , Forouzan ^[8]	Comer ^[9] , Kozierok ^[10]	Stallings ^[11]	Tanenbaum ^[12]	RFC 1122 ^[13]	Cisco Academy ^[13]
Five layers	Four+one layers	Five layers	Four layers	Four layers	Four layers
"Five-layer Internet model" or "TCP/IP protocol suite"	"TCP/IP 5-layer reference model"	"TCP/IP model"	"TCP/IP reference model"	"Internet model"	"Internet model"
Application	Application	Application	Application	Application	Application
Transport	Transport	Host-to-host or transport	Transport	Transport	Transport
Network	Internet	Internet	Internet	Internet	Internetwork
Data link	Data link (Network interface)	Network access	Host-to-network	Link	Network interface
Physical	(Hardware)	Physical			



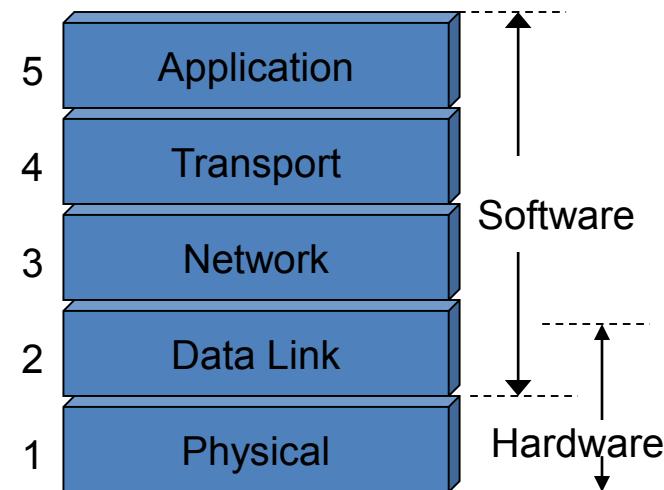
TCP/IP Model

- Typical functions for each layer of TCP/IP model.



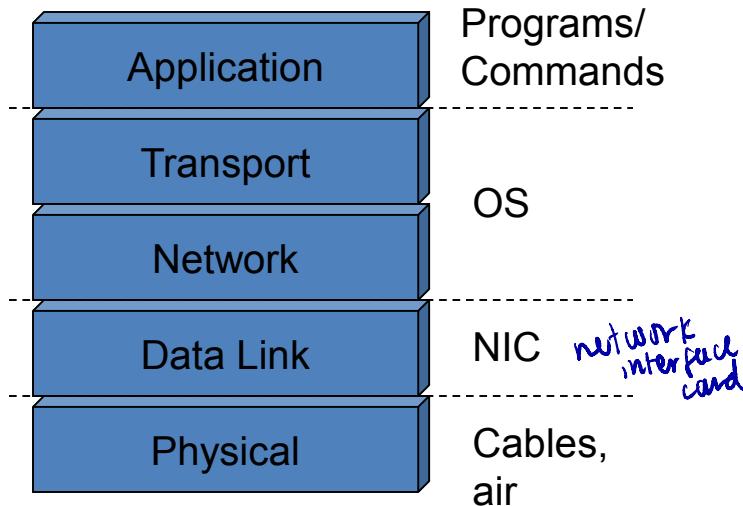
Internet Layered Model

- There are 5 layer in Internet model.
 - Memorize this!!!
- Physical layer is called layer 1
- Application layer is called layer 5.
- First four layers deals mainly with software
- The physical layer (and data link layer) deals with hardware.



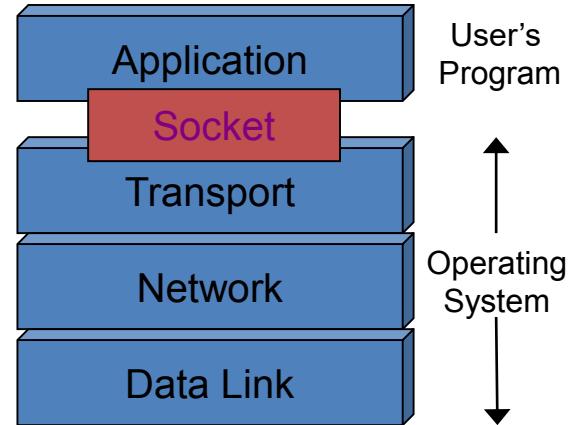
Corresponding Layer in a Host

- For easy visualization of layered model in PC
- Layer 1 is made up of
 - Cables, transmission and reception of NIC
- Layer 2
 - Processing part of NIC
- Layer 3, 4, 5
 - CPU, RAM and hard disk



TCP/IP Suite

- The TCP/IP protocol suite is the protocol architecture of the **Internet**
- The TCP/IP suite has four layers: **Application, Transport, Network, and Data Link Layer**
- End systems (hosts) implement all four layers. Gateways (Routers) only have the bottom two layers (Network and Data Link)





Layered Architecture and Protocol

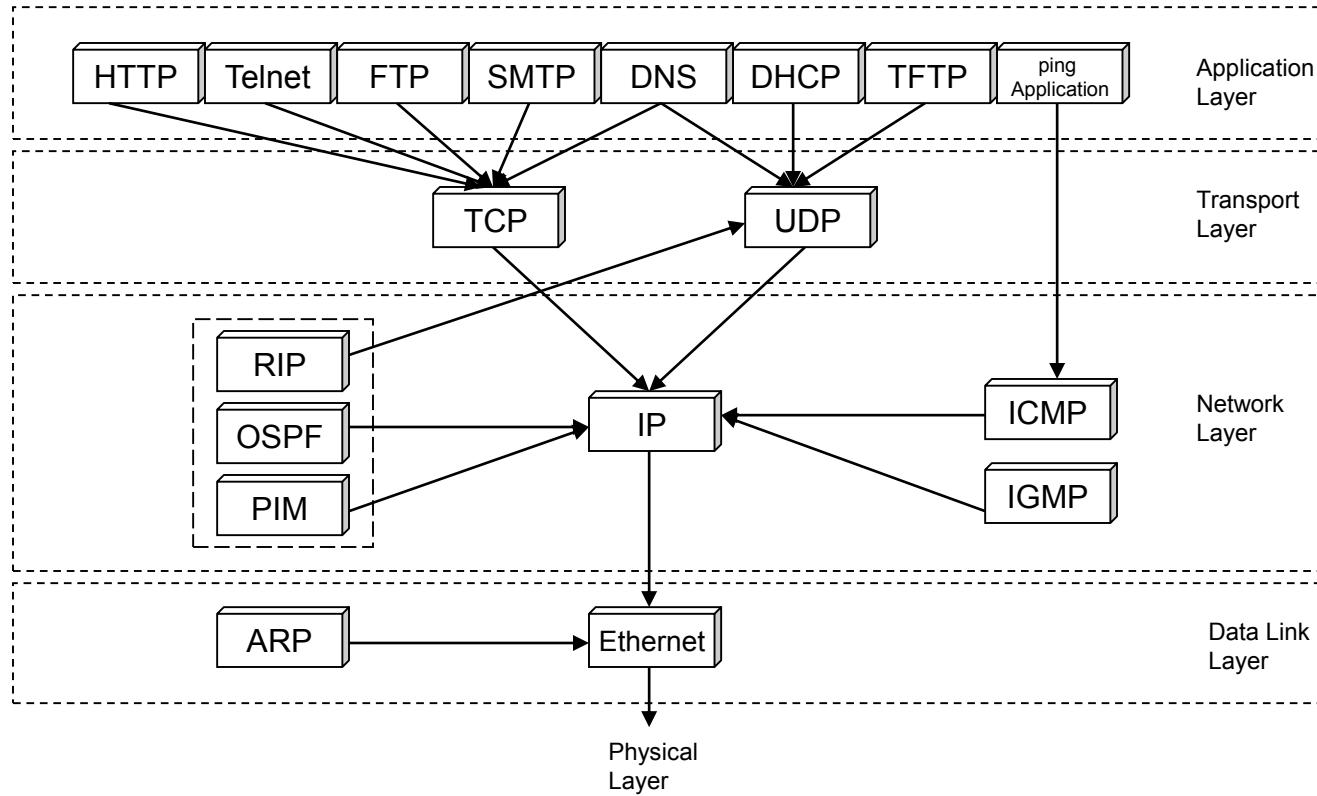
- Each layer has its own protocol
 - Application layer protocol: HTTP, FTP, DNS, DHCP...
 - Transport layer protocol: TCP, UDP
 - Network layer protocol: IP, ICMP, RIP....
 - Data Link layer protocol: ARP, *Ethernet*
 - The complexity of the communication task is reduced by using **multiple protocol layers**:
 - Each protocol is implemented independently
 - Each protocol is responsible for a specific subtask
 - Protocols are grouped in a hierarchy
 - A structured set of protocols is called a **communications architecture** or **protocol suite**.
- protocol for LAN (local)*



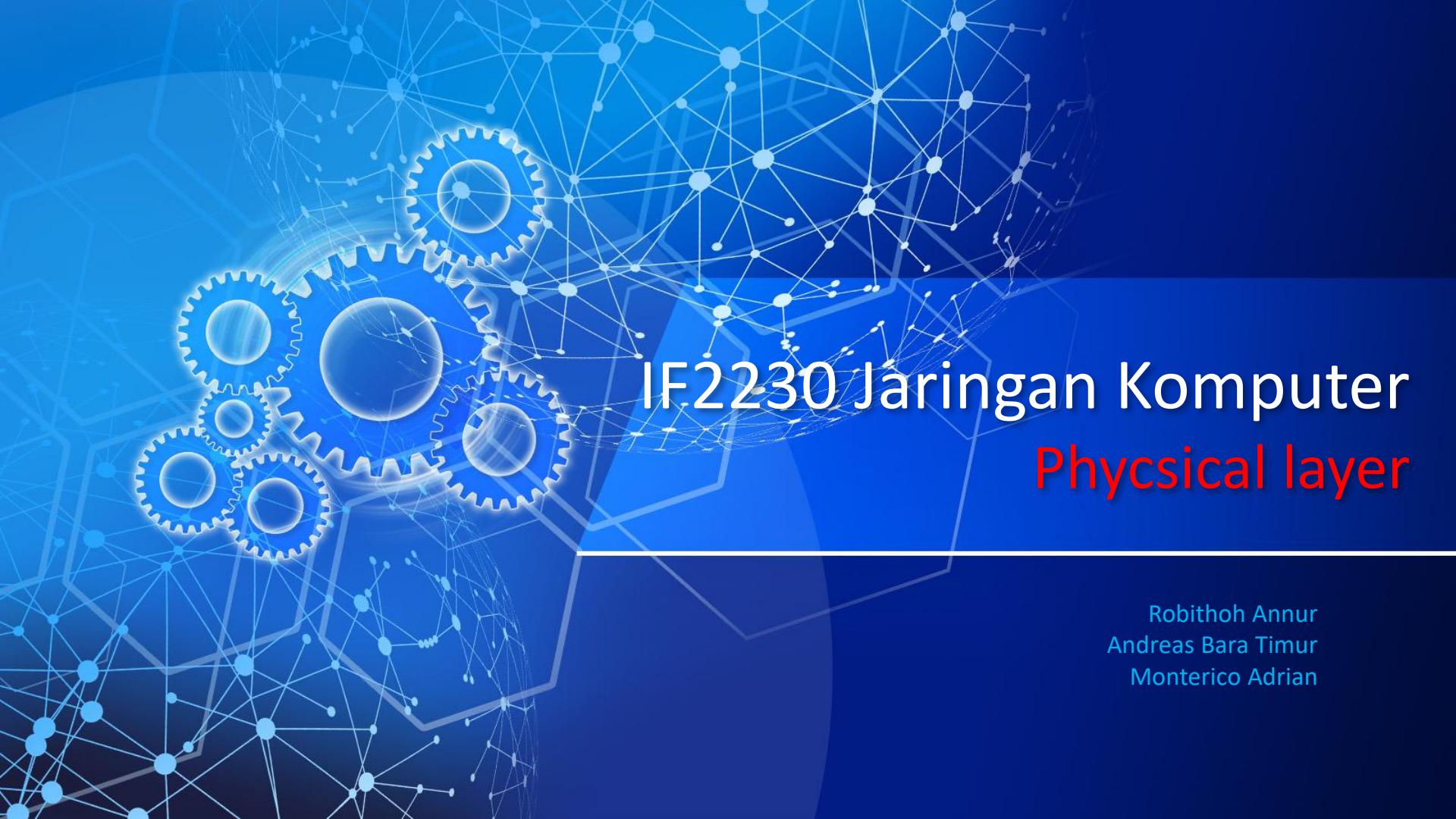
IMPORTANT!!

- If you still don't understand protocol or layered architecture, at least remember the following:
 - Transport Layer gives port numbers
 - Network Layer gives IP address
 - Data Link Layer gives MAC address
- And, the “chunk” of data in
 - Transport Layer is called “segment”
 - Network Layer is called “packet”
 - Data Link Layer is called “frame”
 - Physical Layer is called “bits” or “bit streams”

IMPORTANT: Protocols in Various layer





The background features a complex, abstract design. It consists of a dark blue gradient with a central white hexagonal frame. Inside this frame, there are several glowing, semi-transparent 3D-style gears of varying sizes. These gears are illuminated with a bright blue light, creating a sense of motion and connectivity. The background also contains a dense network of small, glowing blue dots connected by thin lines, forming a mesh-like structure that suggests a digital or physical network.

IF2230 Jaringan Komputer

Physical layer

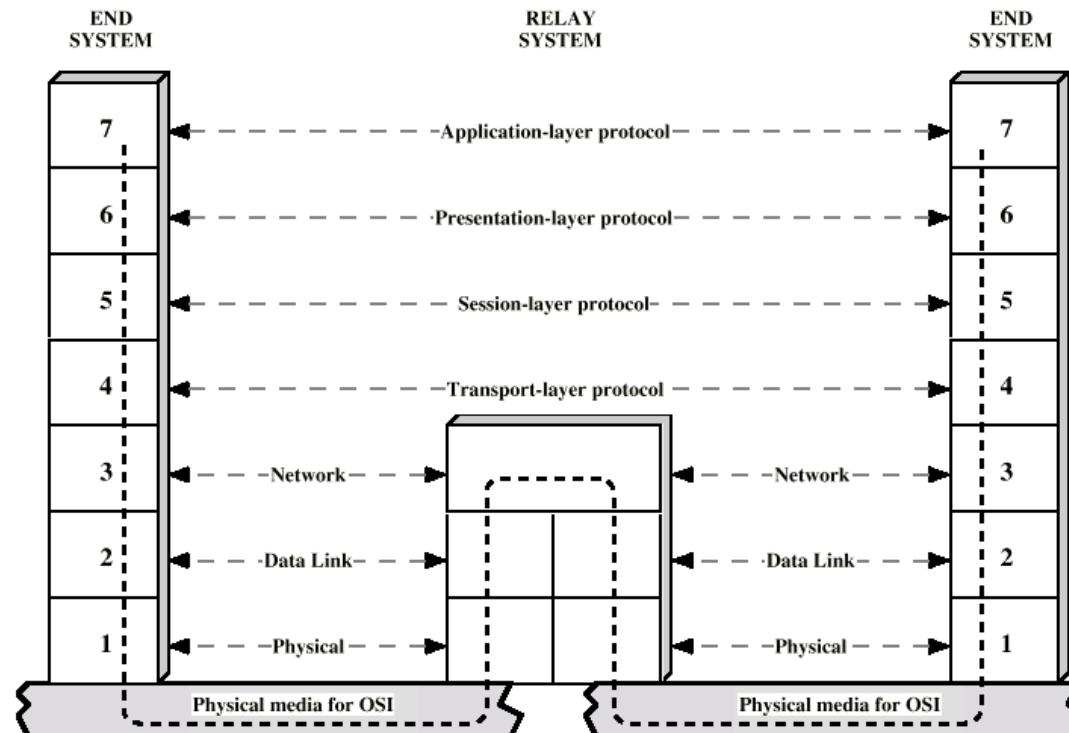
Robithoh Annur
Andreas Bara Timur
Monterico Adrian



Today's Lecture

- Dasar teori komunikasi data
- Jenis media transmisi
 - Media transmisi kabel
 - Media transmisi tanpa kabel
- Komunikasi satelit
- Contoh sistem komunikasi:
 - PSTN
 - Sistem telpon mobile
 - TV Kabel

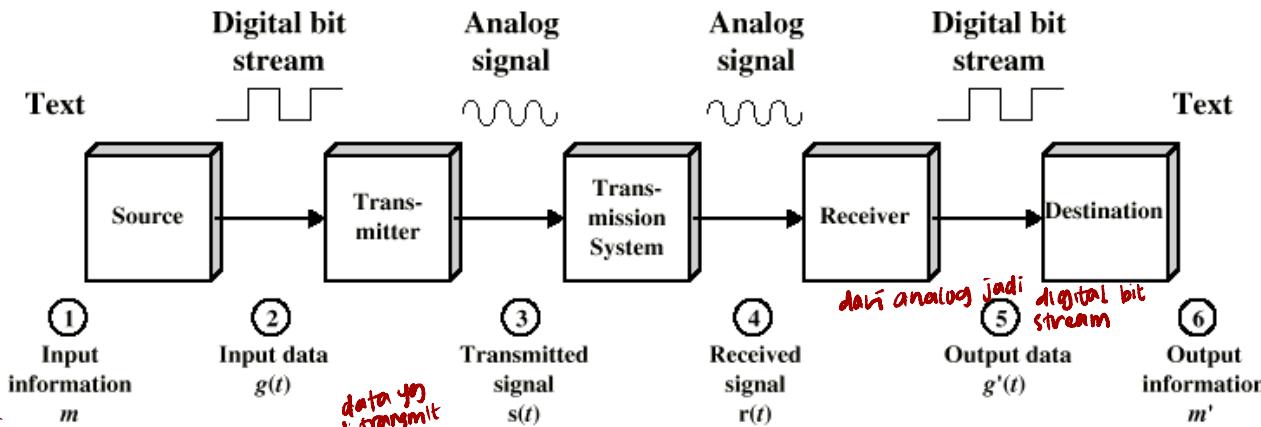
Introduction



Physical layer

L, Layer paling bawah.

don't forget!
($\overline{Q}, \overline{D}$)





Physical Components of a Transmission System

- Transmitter
- Receiver
- Transmission media
 - Guided: cable, twisted pair, fiber
 - Unguided: wireless (radio, infrared)

A Transmission System



Transmitter

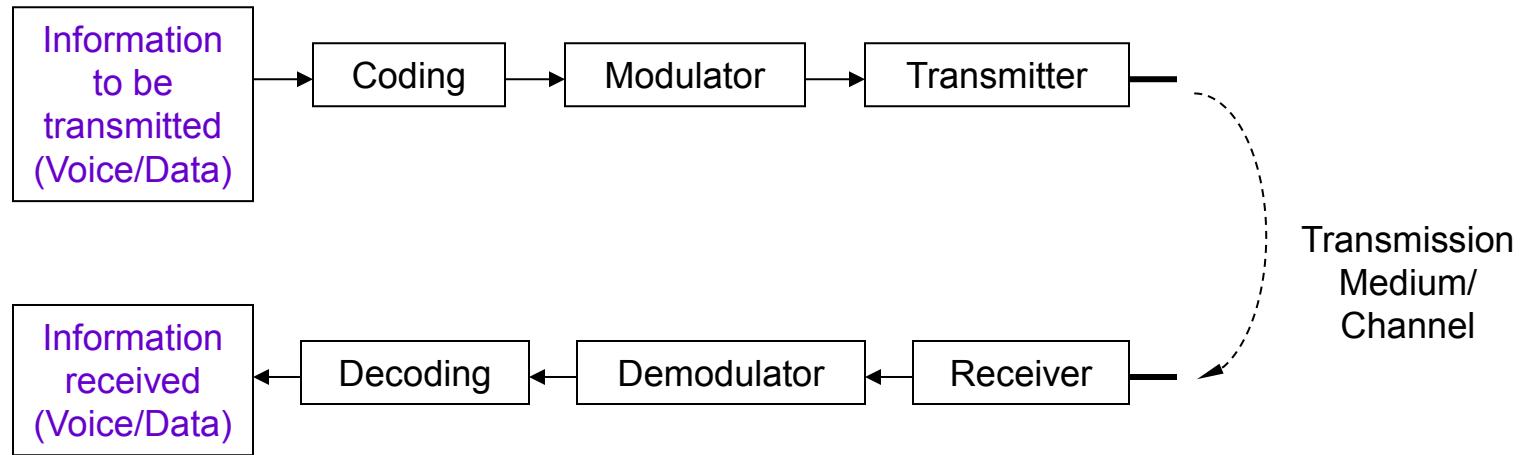
- Converts information into *signal* suitable for transmission
- Injects energy into communication medium or channel
 - Telephone converts voice into electric current
 - Modem converts bits into tones

Receiver

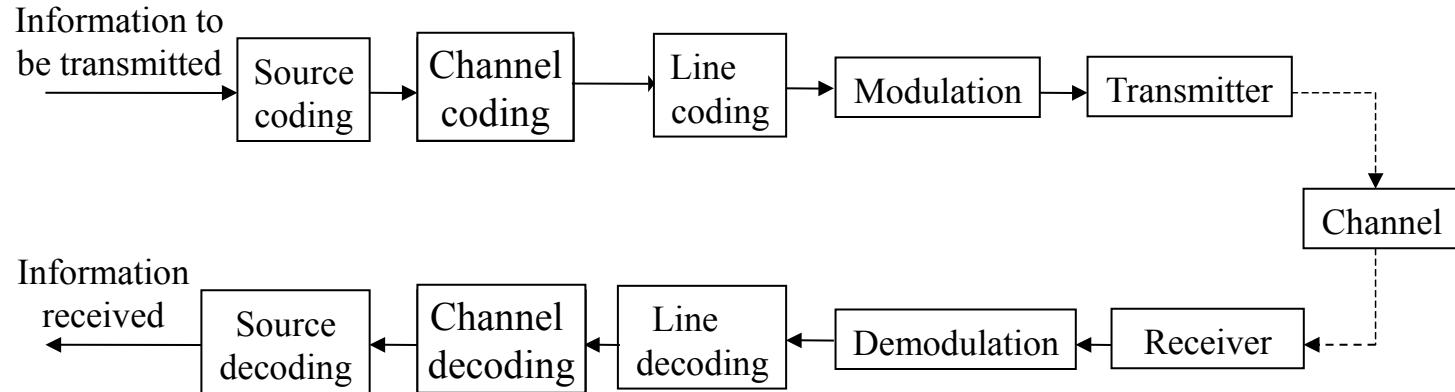
- Receives energy from medium
- Converts received signal into a form suitable for delivery to users
 - Telephone converts current into voice
 - Modem converts tones into bits



A Simplified Communication System



What You Need for Better Understanding





Information



Data vs Signal

- Data:
 - Information units that can be stored in storage devices;
- Signals:
 - Transmission units that can be transmitted over transmission media.
- Data is converted to signal in physical layer.

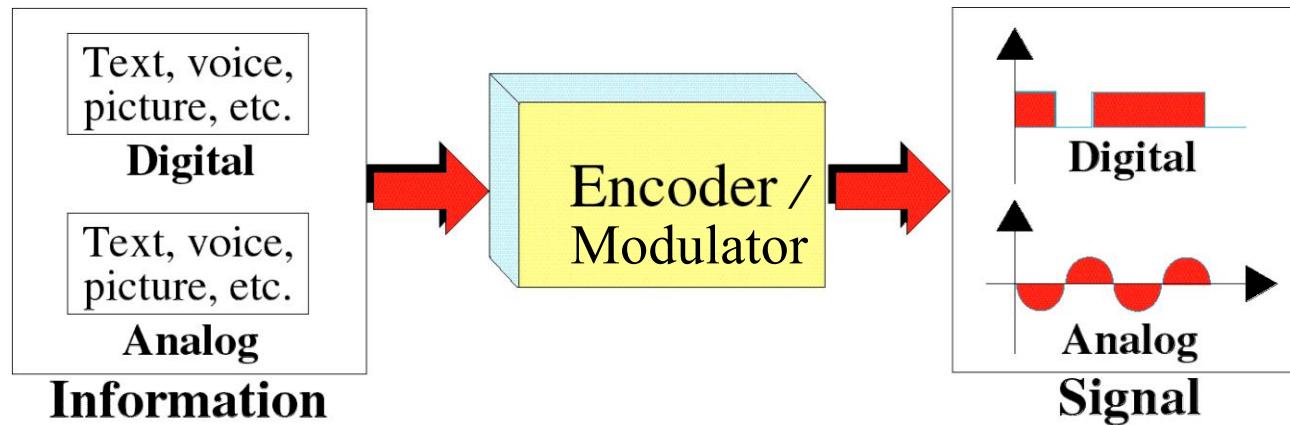


Analog and Digital

- Data can be analog or digital.
Signals can also be analog or digital
- Analog is something that is continuous, can have **infinite values in a range.**
- Digital is something that is discrete, can have only a **limited number of values.** (*depends on the bits*)
- Hence, there are
 - Analog and Digital Data
 - Analog and Digital Signals
 - Periodic and Nonperiodic Signals

Data to Signal

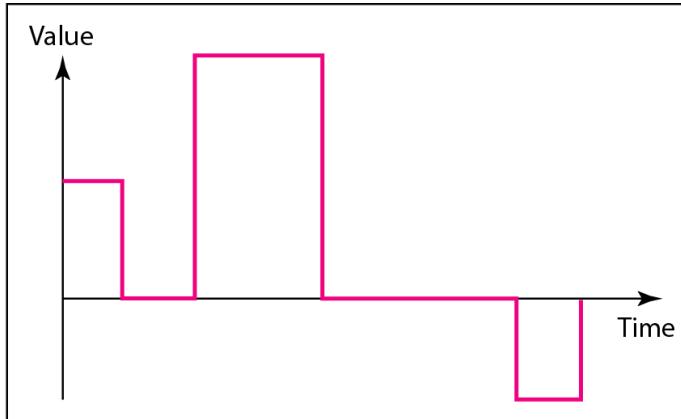
conversion from data to signal.



Analog Signal and Digital Signal

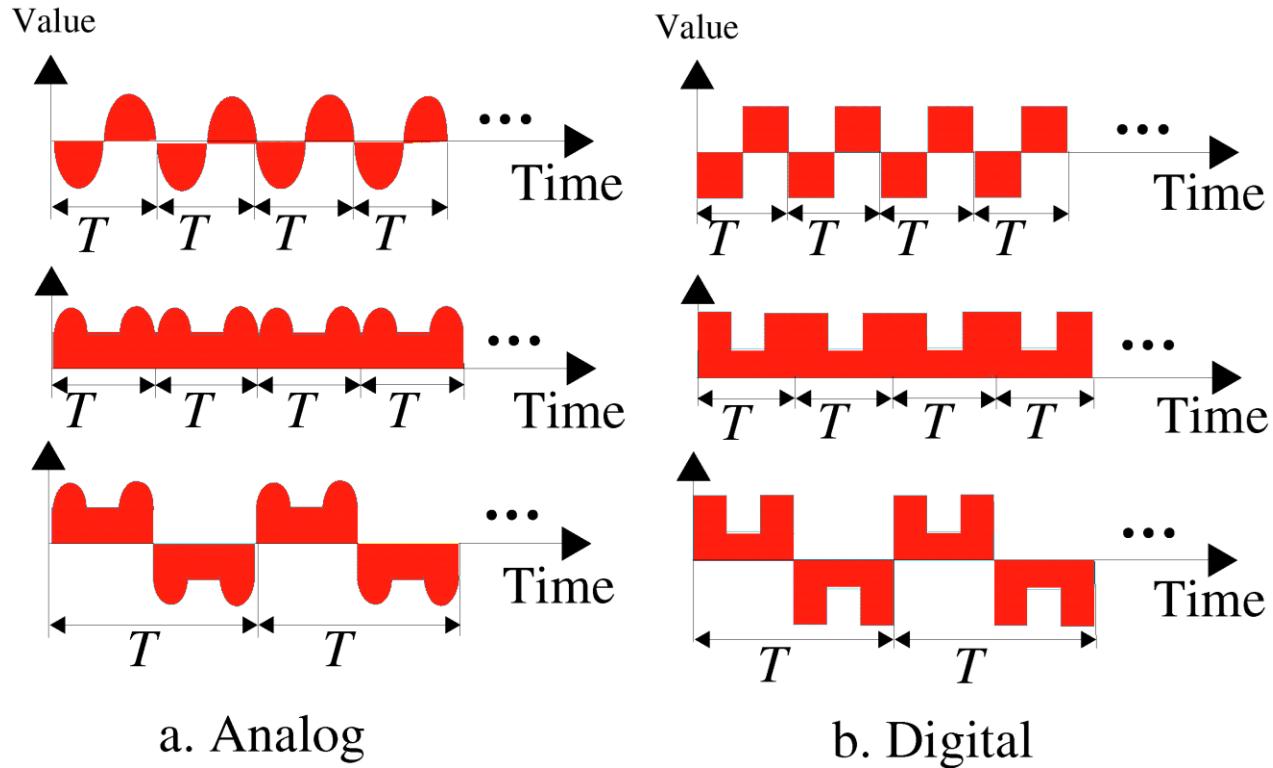


a. Analog signal



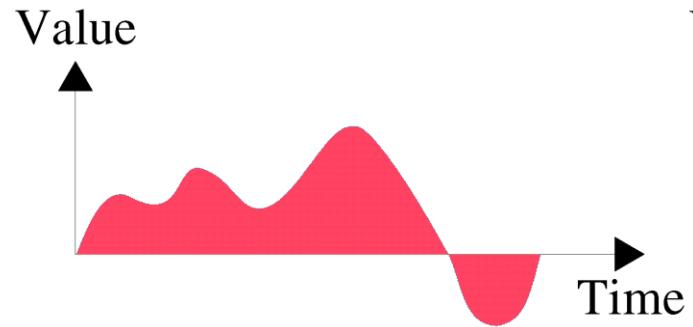
b. Digital signal

Periodic Signal

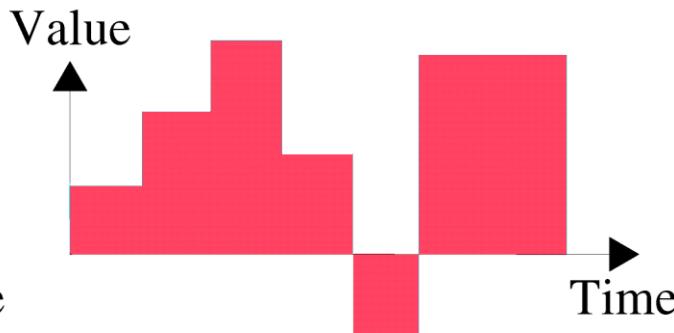


Aperiodic Signal

↳ the frequency of the upcoming signal is not certain



a. Analog



b. Digital

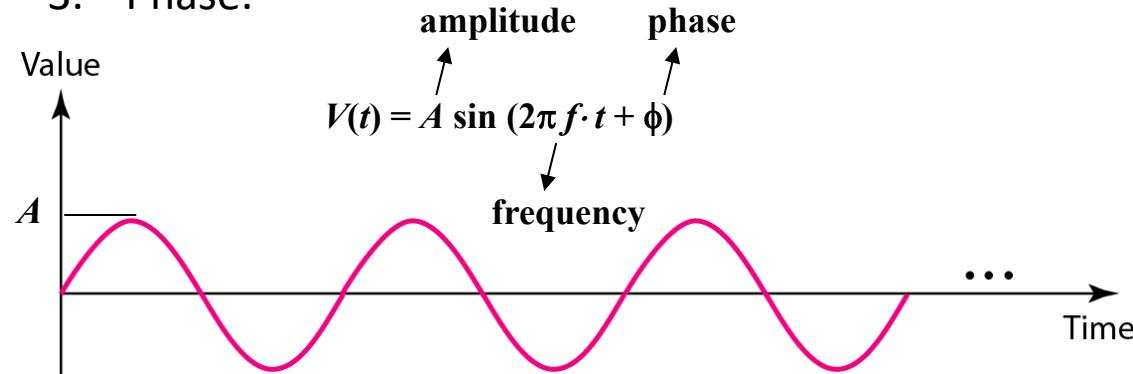


Periodic Analog Signals

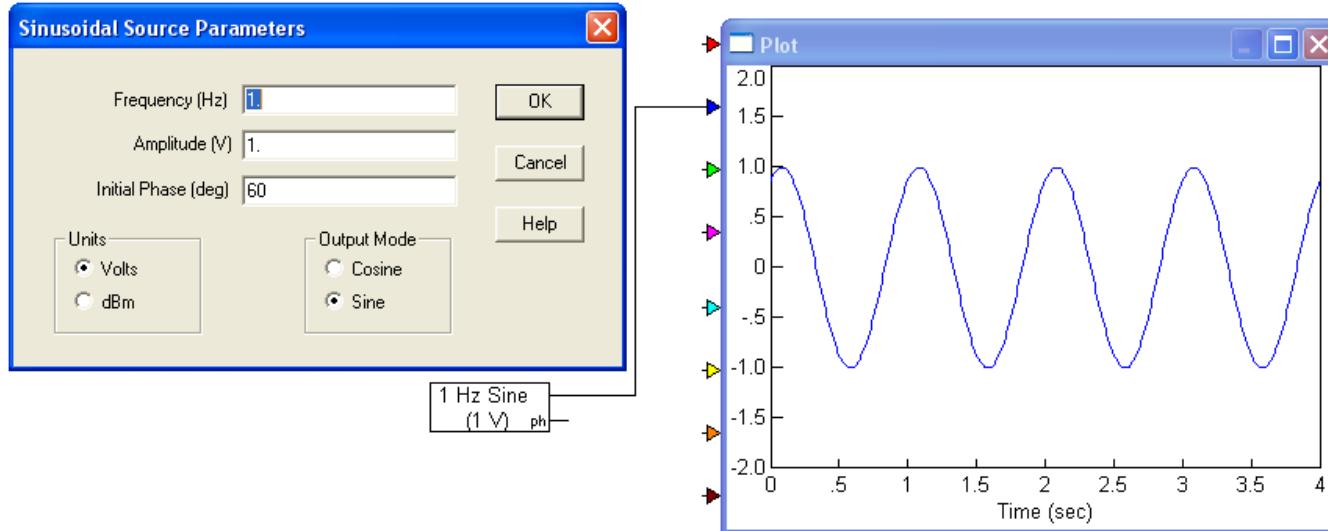
- Analog signals can be classified as simple or composite.
- A simple periodic analog signal, a sine wave, cannot be decomposed into simpler signals.
- A composite periodic analog signal is composed of multiple sine waves.
 - Square wave, Triangular wave, Sawtooth wave
 - Fourier series, still remember

Analog Signals

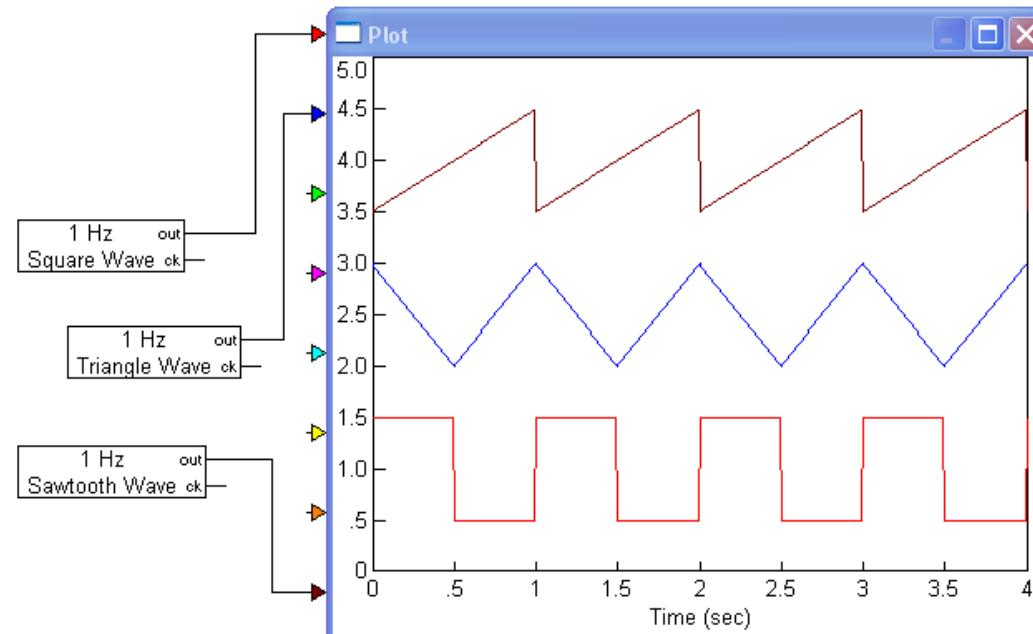
- The sine wave is the most fundamental form of a periodic analog signal.
- Sine waves can be fully described by three characteristics:
 1. Amplitude
 2. Frequency (which is equal to 1/Period)
 3. Phase.



Commsim – Sine Wave

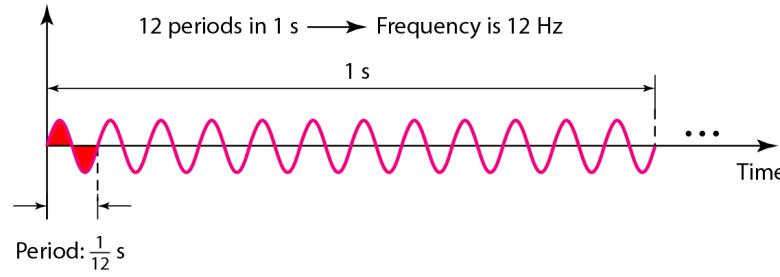


Composite Signals



Frequency and Period

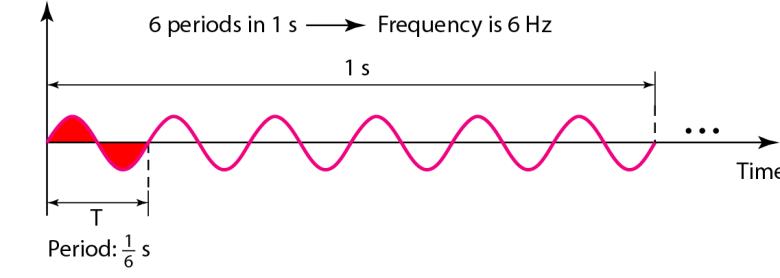
Amplitude



a. A signal with a frequency of 12 Hz

$$f = \frac{1}{T} \quad \text{and} \quad T = \frac{1}{f}$$

Amplitude



b. A signal with a frequency of 6 Hz



Example

*The power we use at home has a frequency of 60 Hz.
The period of this sine wave can be determined as follows:*

$$T = \frac{1}{f} = \frac{1}{60} = 0.0166 \text{ s} = 0.0166 \times 10^3 \text{ ms} = 16.6 \text{ ms}$$



Example

The period of a signal is 100 ms. What is its frequency in kilohertz?

bagi 1000

Solution

First we change 100 ms to seconds, and then we calculate the frequency from the period (1 Hz = 10⁻³ kHz).

$$100 \text{ ms} = 100 \times 10^{-3} \text{ s} = 10^{-1} \text{ s}$$

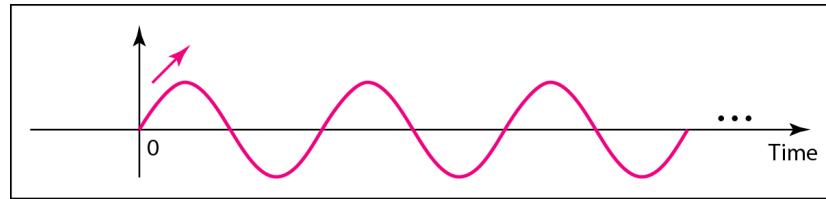
$$f = \frac{1}{T} = \frac{1}{10^{-1}} \text{ Hz} = 10 \text{ Hz} = 10 \times 10^{-3} \text{ kHz} = 10^{-2} \text{ kHz}$$



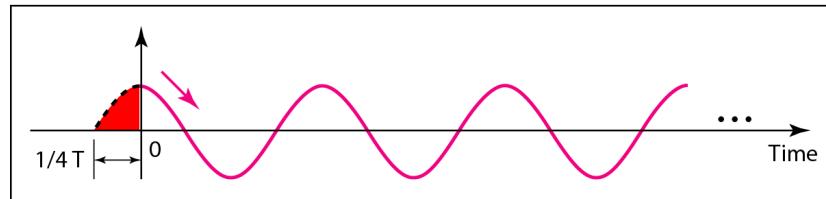
Notes on Frequency

- Frequency is the rate of change with respect to time.
- Change in a short span of time means high frequency.
- Change over a long span of time means low frequency
- If a signal does not change at all, its frequency is zero. *→ basically no signal :)*
- If a signal changes instantaneously, its frequency is infinite.

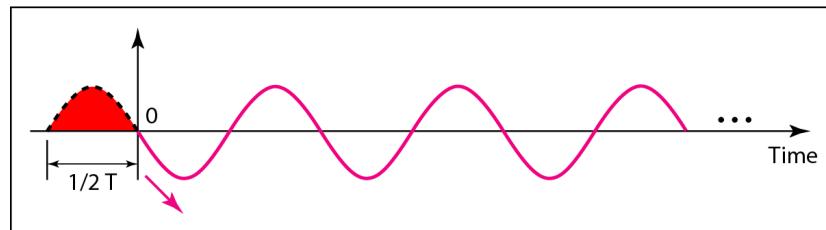
Phase



a. 0 degrees



b. 90 degrees



c. 180 degrees
shift waves by



Example

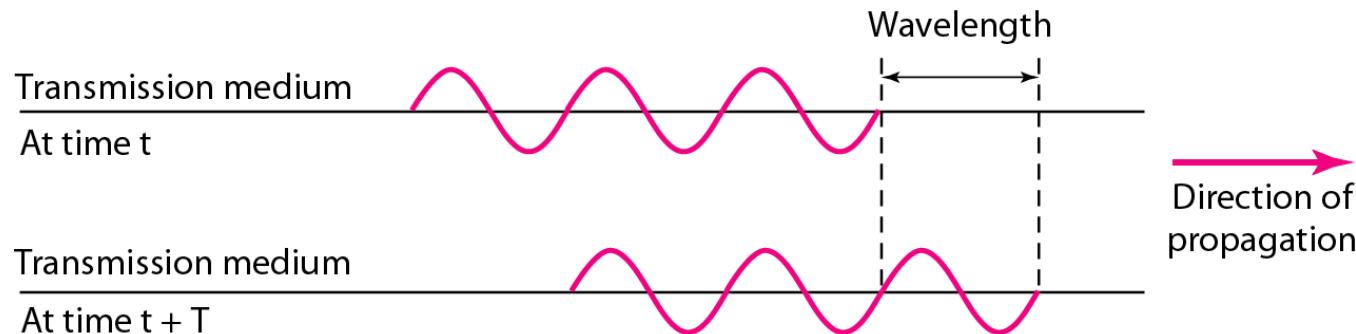
*A sine wave is offset 1/6 cycle with respect to time 0.
What is its phase in degrees and radians?*

Solution

We know that 1 complete cycle is 360° . Therefore, 1/6 cycle is

$$\frac{1}{6} \times 360 = 60^\circ = 60 \times \frac{2\pi}{360} \text{ rad} = \frac{\pi}{3} \text{ rad} = 1.046 \text{ rad}$$

Wavelength and Period



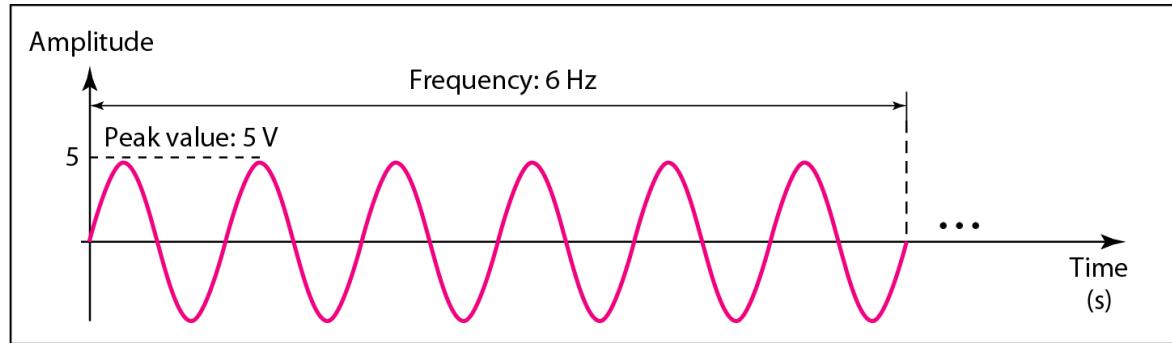
$$c \cdot \lambda \cdot f$$

wavelength (λ)
= $\frac{c}{f}$ speed of
light
($3 \cdot 10^8$)

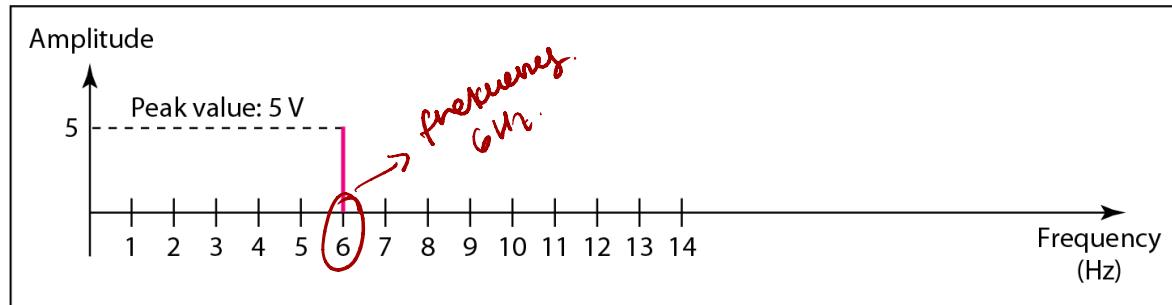
Direction of
propagation

anahans

Time & Frequency Domain



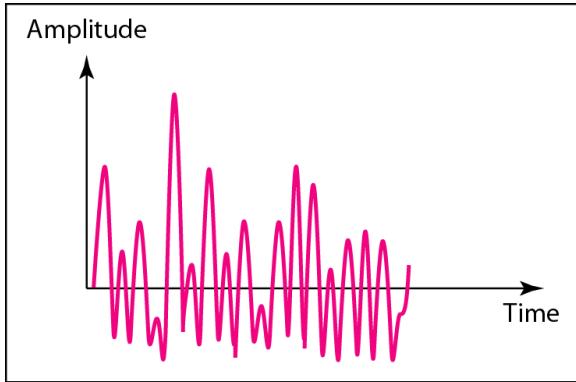
a. A sine wave in the time domain (peak value: 5 V, frequency: 6 Hz)



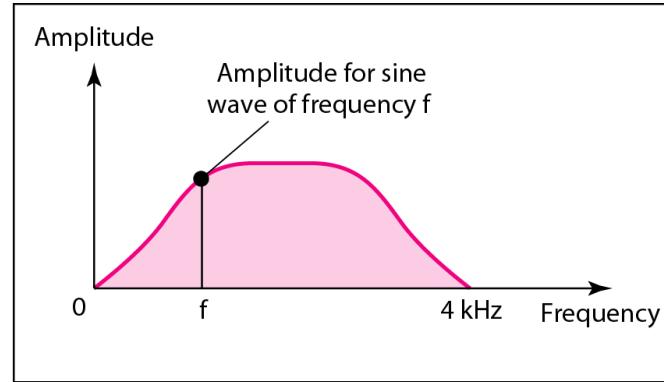
b. The same sine wave in the frequency domain (peak value: 5 V, frequency: 6 Hz)

Non-Periodic Signal

↳ irregular frequencies.



a. Time domain



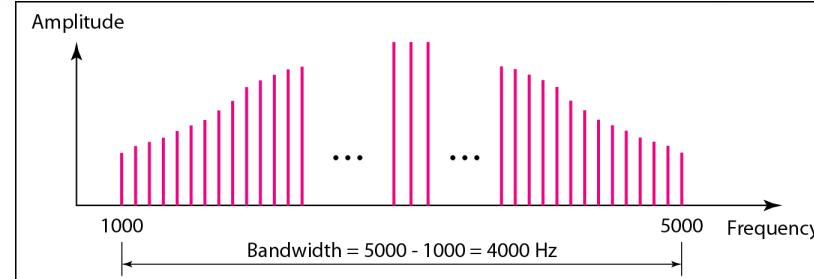
b. Frequency domain

a nonperiodic composite signal. It can be the signal created by a microphone or a telephone set when a word or two is pronounced. In this case, the composite signal cannot be periodic, because that implies that we are repeating the same word or words with exactly the same tone.

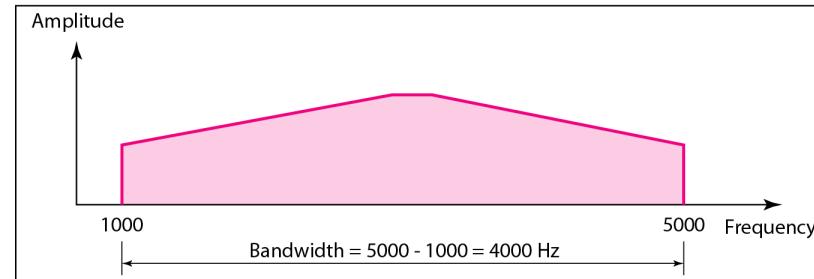
Bandwidth

- The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.

$$B = f_h - f_l$$



a. Bandwidth of a periodic signal



b. Bandwidth of a nonperiodic signal

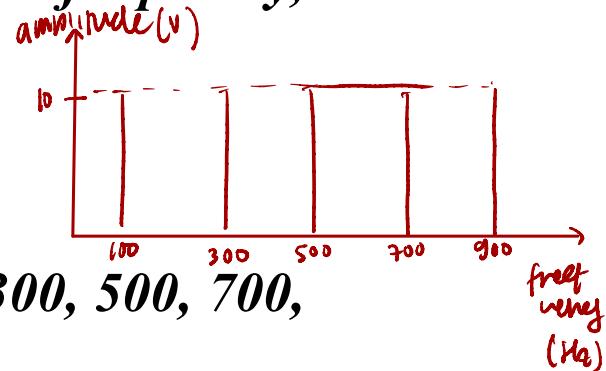
Example

If a periodic signal is decomposed into five sine waves with frequencies of 100, 300, 500, 700, and 900 Hz, what is its bandwidth? Draw the spectrum, assuming all components have a maximum amplitude of 10 V.

Solution

Let f_h be the highest frequency, f_l the lowest frequency, and B the bandwidth. Then

$$B = f_h - f_l = 900 - 100 = 800 \text{ Hz}$$



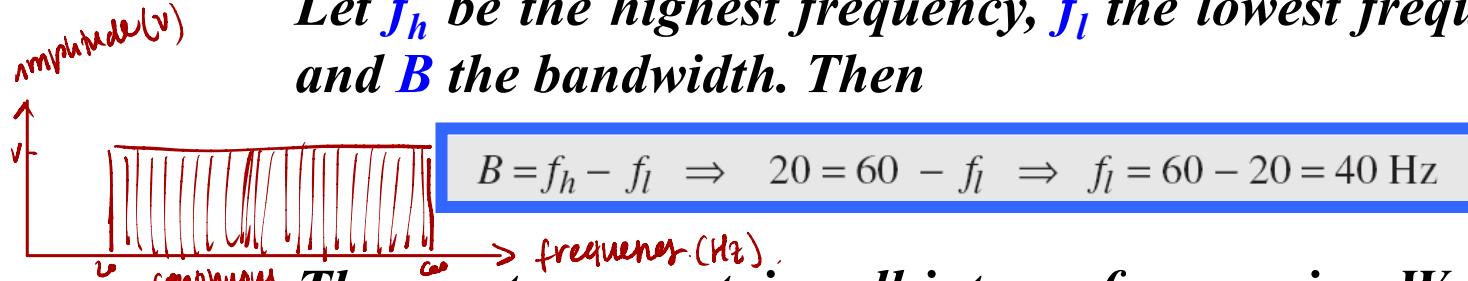
The spectrum has only five spikes, at 100, 300, 500, 700, and 900 Hz

Example

A periodic signal has a bandwidth of 20 Hz. The highest frequency is 60 Hz. What is the lowest frequency? Draw the spectrum if the signal contains all frequencies of the same amplitude.

Solution

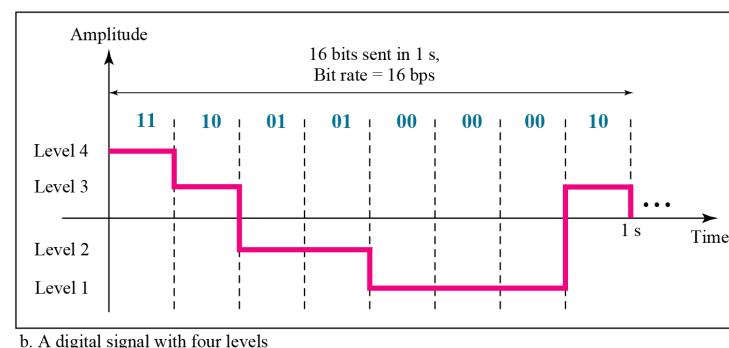
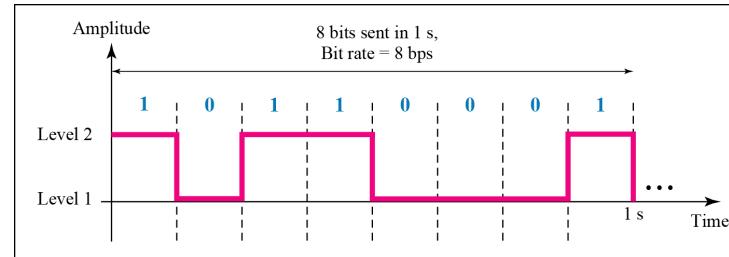
Let f_h be the highest frequency, f_l the lowest frequency, and B the bandwidth. Then



The spectrum contains all integer frequencies. We show this by a series of spikes.

Digital Signal

- In digital signal, a 1 can be encoded as a positive voltage and a 0 as zero voltage.
- A digital signal can have more than two levels. In this case, we can send more than 1 bit for each level.
- Bit rate is used to describe digital signals. The bit rate is the number of bits sent in one seconds, expressed in bits per second (bps).





Example

A digital signal has eight levels. How many bits are needed per level? We calculate the number of bits from the formula

$$\text{Number of bits per level} = \log_2 8 = 3$$

Each signal level is represented by 3 bits.

Example

Assume we need to download text documents at the rate of 100 pages **per minute**. What is the required bit rate of the channel? Given that a page is an average of 24 lines with 80 characters in each line, and **one character** requires 8 bits.

Solution

$$\frac{100 \times 24 \times 80 \times 8}{60} = 25600 \text{ bps} = 25.6 \text{ kbps}$$

total char / page

total char

bits required utk 1 char

| minute



Example

A digitized voice channel, as we will see in Chapter 4, is made by digitizing an analog voice signal with spectrum from 0 Hz to 4-kHz. We need to sample the signal at twice the highest frequency (two samples per hertz). We assume that each sample requires 8 bits. What is the required bit rate?

Solution

The bit rate can be calculated as

$$2 \times 4000 \times 8 = 64,000 \text{ bps} = 64 \text{ kbps}$$



Example

What is the bit rate for high-definition TV (HDTV)?

Solution

HDTV uses digital signals to broadcast high quality video signals. The HDTV screen is normally a ratio of 16 : 9. There are 1920 by 1080 pixels per screen, and the screen is renewed 30 times per second. Twenty-four bits represents one color pixel.



$$1920 \times 1080 \times 30 \times 24 = 1,492,992,000 \text{ or } 1.5 \text{ Gbps}$$

The TV stations reduce this rate to 20 to 40 Mbps through compression.



Other Definition

**bit duration = time period for a bit
= $1 / \text{bit rate}$**

Example: A digitized voice channel has a bit rate of 64kbps. Then the duration of each bit is $1/64\text{kbps} = 0.000015625 \text{ s} = 15.625 \mu\text{s}$

**bit length = distance a bit occupies the medium
= propagation speed \times bit duration**

Example: In the previous example, suppose that the signal propagation speed in the medium is $2 \times 10^8 \text{ m/s}$, Then the bit length is $2 \times 10^8 \text{ m/s} \times 0.000015625 \text{ s} = 3125 \text{ m} = 3.125 \text{ km}$



Data Rate Limits

- A very important consideration in data communications is how fast we can send data, in bits per second, over a channel. Data rate depends on three factors:
 - The bandwidth available
 - The level of the signals we use
 - The quality of the channel (the level of noise)



Nyquist Rate

- Nyquist bit rate defines the theoretical maximum bit rate:
 - Bit rate = $2 \times \text{bandwidth} \times \log_2 L$,
 - bandwidth is the analog bandwidth, and, L is the signal level.
- Theoretically, if we can have infinitely many levels, then we can achieve infinite bit rate.
- But in reality, it can never be achieved because increasing the levels of a signal may reduce the reliability of the system as the noise may more easily corrupt the signals.



Example

Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. The maximum bit rate can be calculated as

$$\text{BitRate} = 2 \times 3000 \times \log_2 2 = 6000 \text{ bps}$$

Consider the same noiseless channel transmitting a signal with four signal levels (for each level, we send 2 bits). The maximum bit rate can be calculated as

$$\text{BitRate} = 2 \times 3000 \times \log_2 4 = 12,000 \text{ bps}$$



Example

We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels do we need?

Solution

We can use the Nyquist formula as shown:

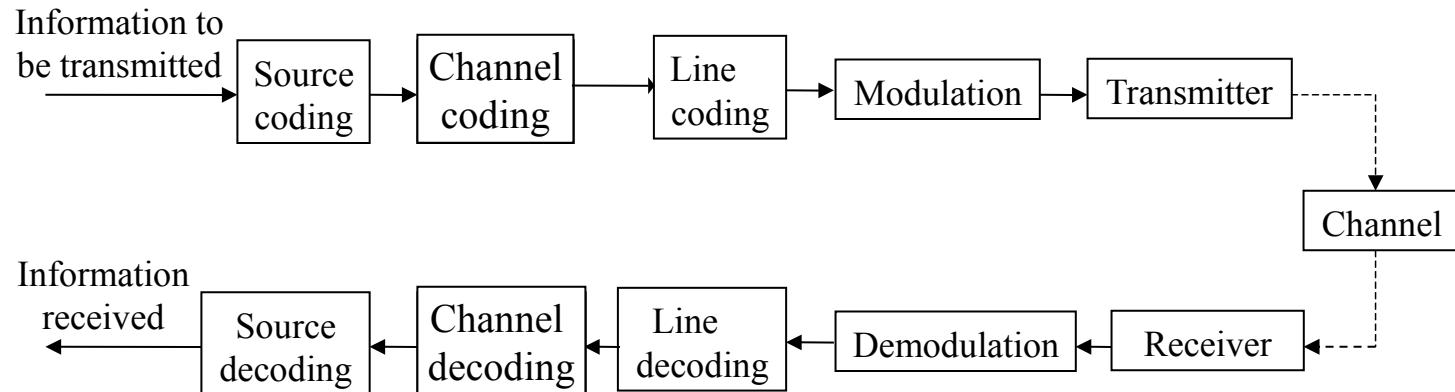
$$265,000 = 2 \times 20,000 \times \log_2 L$$
$$\log_2 L = 6.625 \quad L = 2^{6.625} = 98.7 \text{ levels}$$

Since this result is not a power of 2, we need to either increase the number of levels or reduce the bit rate. If we have 128 levels, the bit rate is 280 kbps. If we have 64 levels, the bit rate is 240 kbps.



Encoding

What You Need for Better Understanding





Encoding

- The first step in turning nodes and links into usable building blocks is to
- understand how to connect them in such a way that bits can be transmitted from one node to the other.
- **Signals propagate over physical links.**
- The task, therefore, is to encode the binary data that the source node wants to send into the signals that the links are able to carry
- At the receiving node, it needs to decode the signal back into the corresponding binary data



Source Coding

- Networks are handling streams of 0's and 1'
- **Source Encoding**: compression, according to statistics of 0's and 1's, map blocks of bits to more regular "shorter" blocks! Get rid of redundancy
- **Source Decoding**: inverse of source encoding



Channel Coding

*physical trans-
mission link*

- **Channel Encoding:** According to channel conditions, add redundancy for more reliable transmission
- **Channel decoding:** the inverse
- **Observation:** source encoding attempts to eliminate “useless information”, while channel encoding add “useful information”; both deal with redundancies!



Modulation/Demodulation

- **Modulation:** maps blocks of bits to well-defined waveforms or symbols (a set of signals for better transmission), then shifts transmission to the carrier frequency band (the band you have right to transmit)
- **Demodulation:** the inverse of modulation
- **Demodulation vs. Detection:** Detection is to recover the modulated signal from the “distorted noisy” received signals

BPSK → *google
for more!*



Source vs. Channel vs. Line Coding

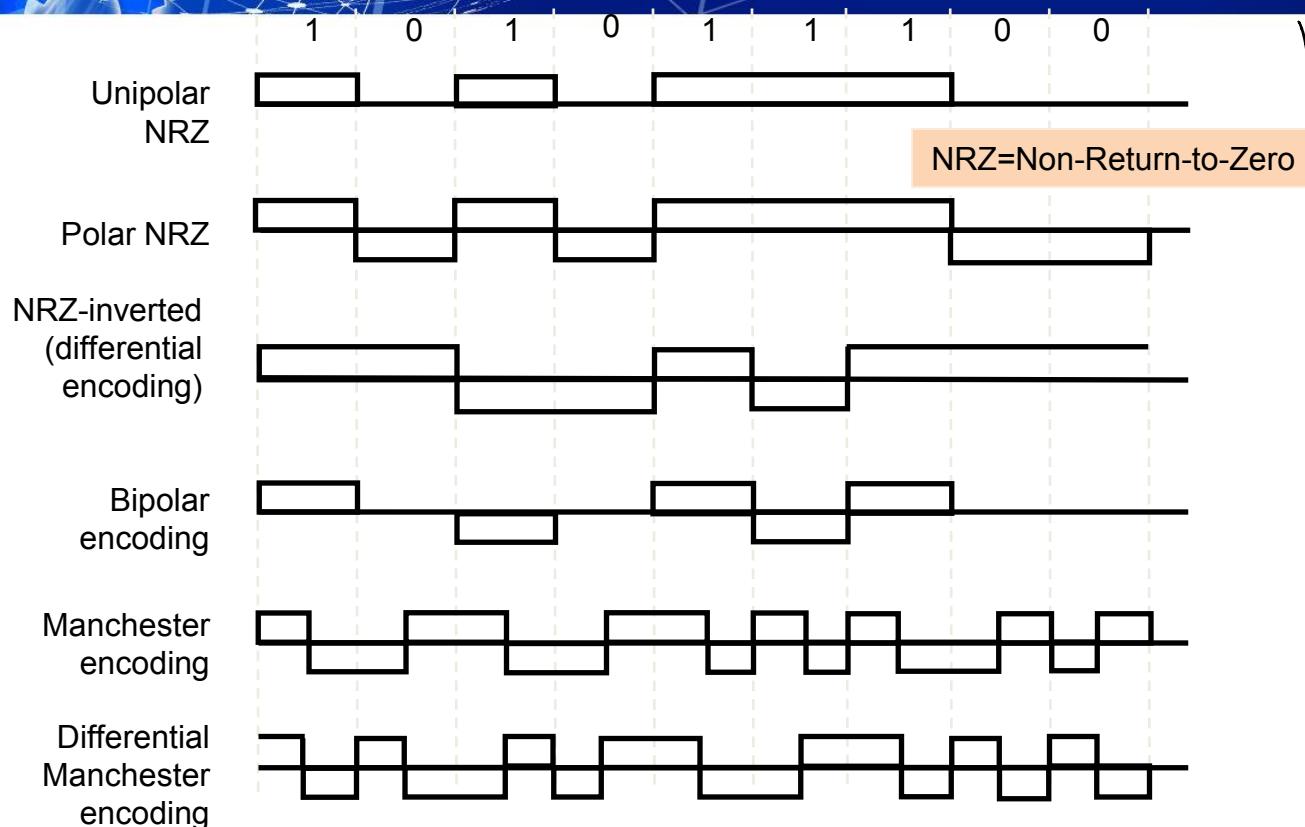
- **Source coding:** eliminating redundancy in order to make efficient use of storage space and/or transmission channels
 - Huffman coding/ Morse code
- **Channel coding:** a pre-transmission mapping applied to a digital signal or file, usually designed to make error-correction possible
 - Parity check / Hamming code / Reed-Soloman code
- **Line coding:** performed to adapt the transmitted signal to the (electrical) characteristics of a transmission channel
- Order: source coding -> channel coding -> line coding



What is Line Coding?

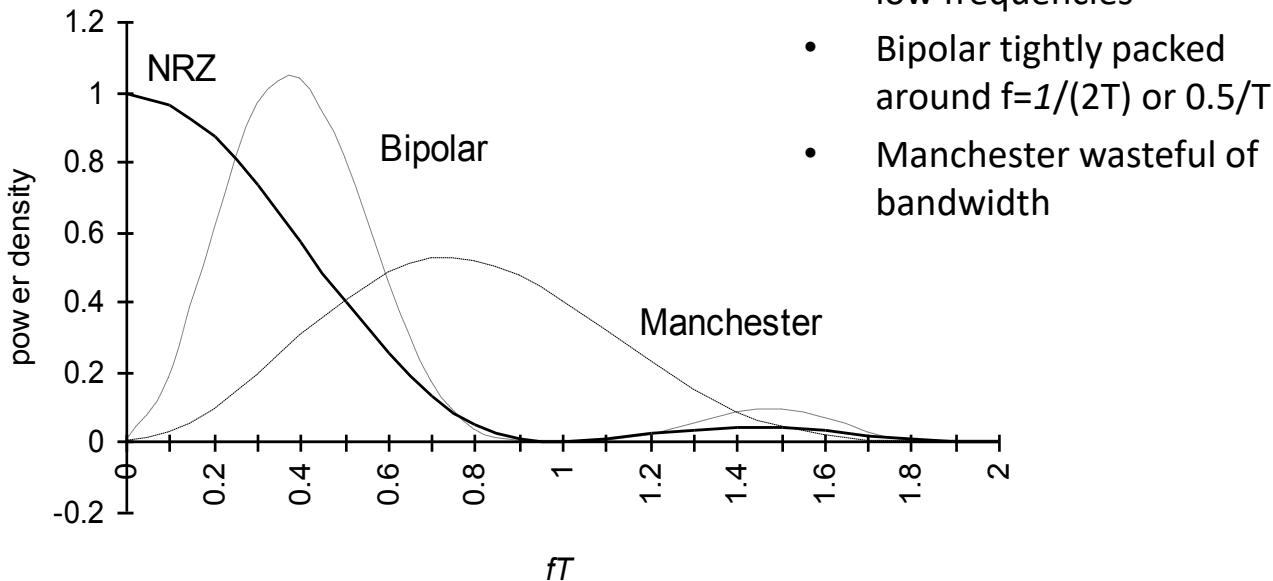
- Mapping of binary information sequence into the digital signal that enters the channel
 - Ex. “1” maps to +A square pulse; “0” to –A pulse
- Line code selected to meet system requirements:
 - *Transmitted power*: Power consumption = \$
 - *Bit timing*: Transitions in signal help timing recovery
 - *Bandwidth efficiency*: Excessive transitions wastes bw
 - *Low frequency content*: Some channels block low frequencies
 - long periods of +A or of –A causes signal to “droop”
 - Waveform should not have low-frequency content
 - *Error detection*: Ability to detect errors helps
 - *Complexity/cost*: Is code implementable in chip at high speed?

Line coding examples

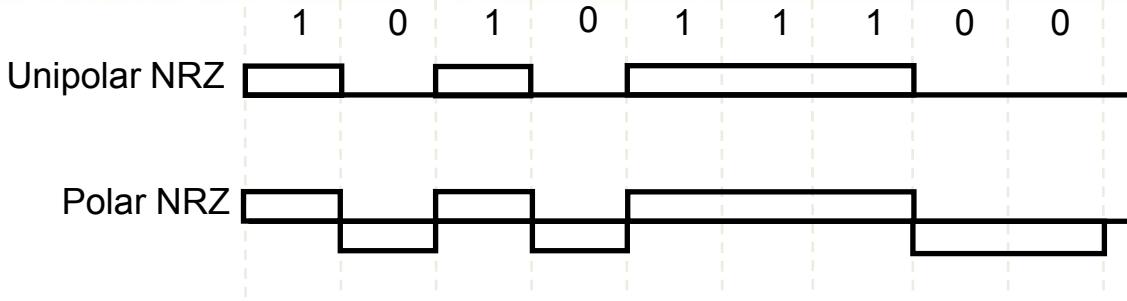


Spectrum of Line codes

- Assume 1s & 0s independent & equiprobable



Unipolar & Polar Non-Return-to-Zero (NRZ)



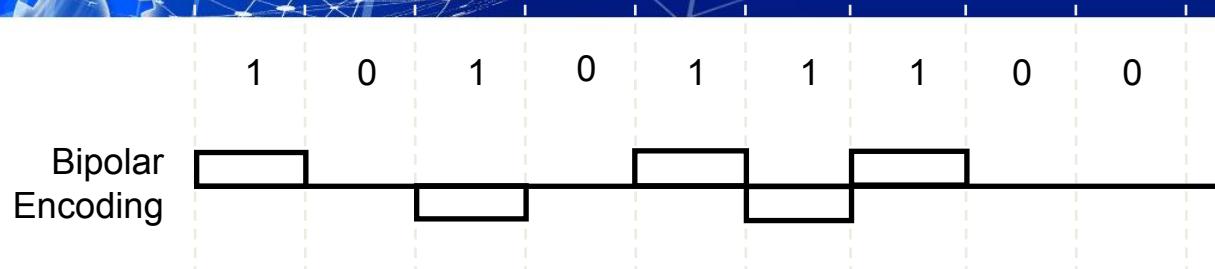
Unipolar NRZ

- “1” maps to +A pulse
- “0” maps to no pulse
- High Average Power
$$0.5*A^2 + 0.5*0^2 = A^2/2$$
- Long strings of A or 0
 - Poor timing
 - Low-frequency content
- Simple

Polar NRZ

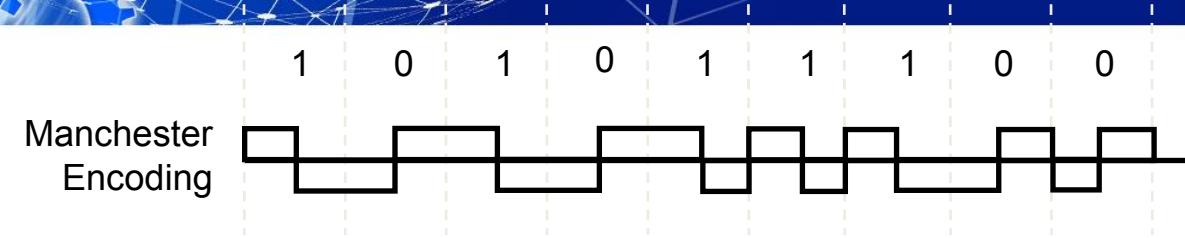
- “1” maps to +A/2 pulse
- “0” maps to -A/2 pulse
- Better Average Power
$$0.5*(A/2)^2 + 0.5*(-A/2)^2 = A^2/4$$
- Long strings of +A/2 or -A/2
 - Poor timing
 - Low-frequency content
- Simple

Bipolar Code



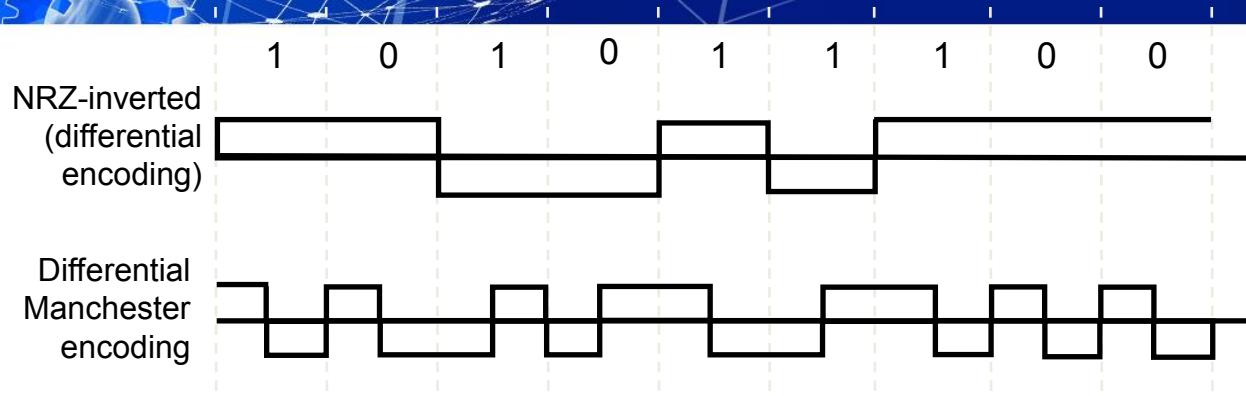
- Three signal levels: $\{-A, 0, +A\}$
- “1” maps to $+A$ or $-A$ in alternation
- “0” maps to no pulse
 - Every $+A$ pulse matched by $-A$ pulse so little content at low frequencies
- String of 1s produces a square wave
 - Spectrum centered at around $f=1/(2T)$ or $0.5/T$
- Long string of 0s causes receiver to lose synchronization
- Zero-substitution codes are needed

Manchester code & *mBnB* codes



- “1” maps into A/2 first $T/2$, -A/2 last $T/2$
- “0” maps into -A/2 first $T/2$, A/2 last $T/2$
- Every interval has transition in middle
 - Timing recovery easy
 - Uses double the minimum bandwidth
- Simple to implement
- Used in 10-Mbps Ethernet & other LAN standards
- *mBnB* line code
- Maps block of m bits into n bits
- Manchester code is 1B2B code
- 4B5B code used in FDDI LAN
- 8B10b code used in Gigabit Ethernet
- 64B66B code used in 10G Ethernet

Differential Coding

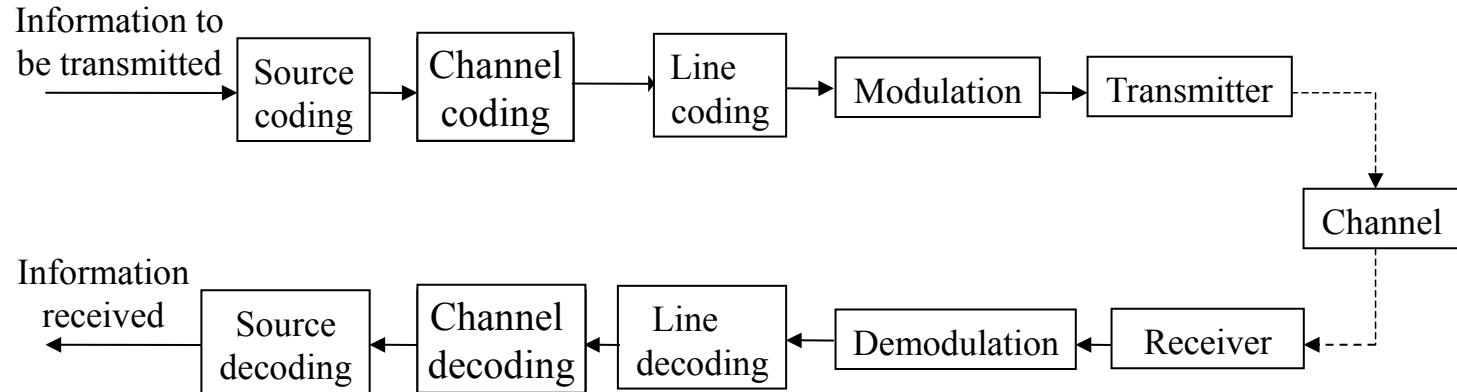


- Errors in some systems cause transposition in polarity, +A become -A and vice versa
 - All subsequent bits in Polar NRZ coding would be in error
- Differential line coding provides robustness to this type of error
- “1” mapped into transition in signal level
- “0” mapped into no transition in signal level
- Also used along with Manchester coding



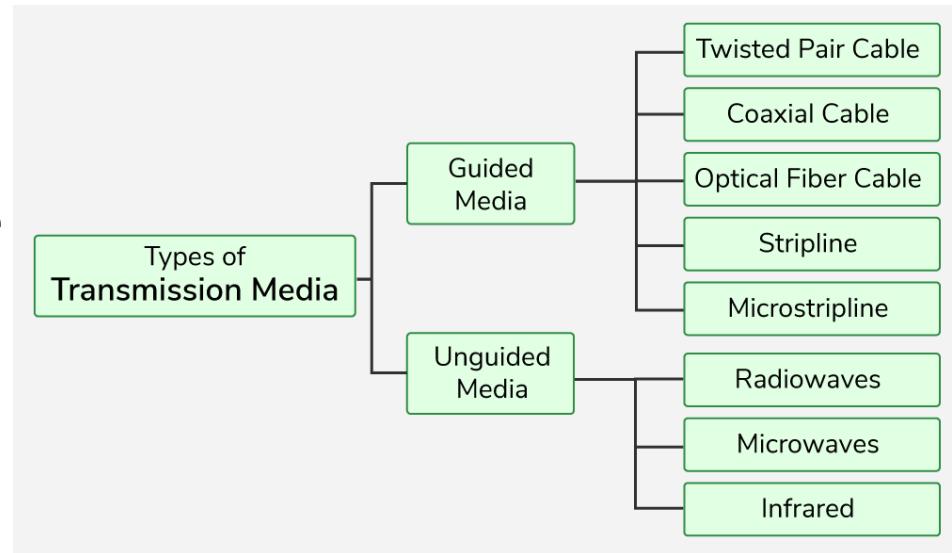
Transmission Medium

What You Need for Better Understanding



Transmission Medium

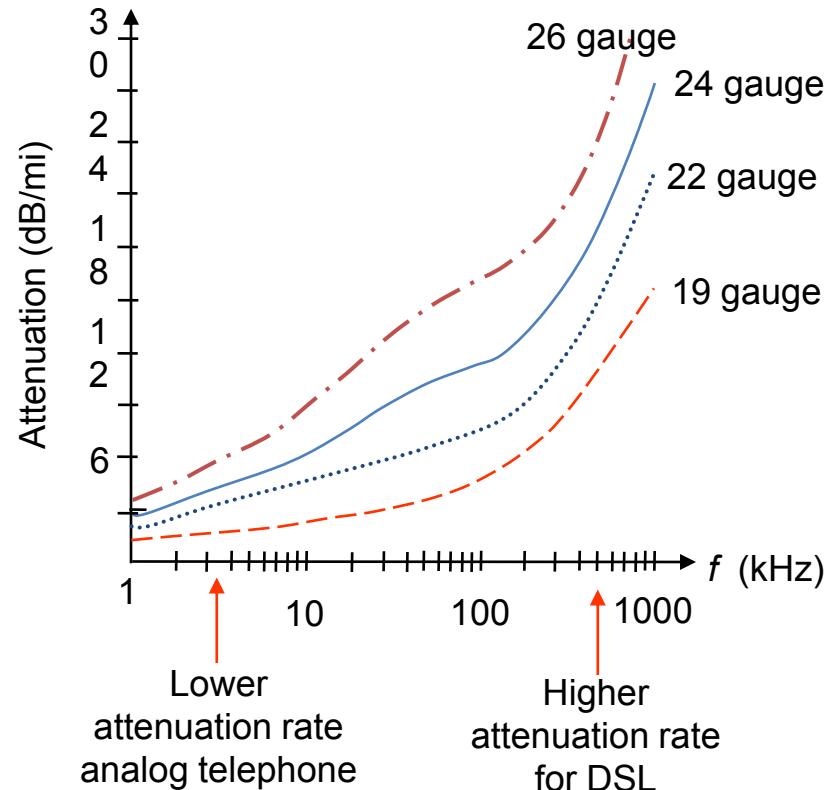
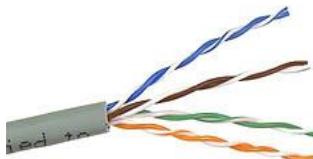
- A transmission medium is a physical path or a **communication channel** that carries the information from the sender to the receiver



Twisted Pair

Twisted pair

- Two insulated copper wires arranged in a regular spiral pattern to minimize interference
- Various thicknesses, e.g. 0.016 inch (24 gauge)
- Low cost
- Telephone subscriber loop from customer to CO
- Old trunk plant connecting telephone COs
- Intra-building telephone from wiring closet to desktop
- In old installations, loading coils added to improve quality in 3 kHz band, but more attenuation at higher frequencies





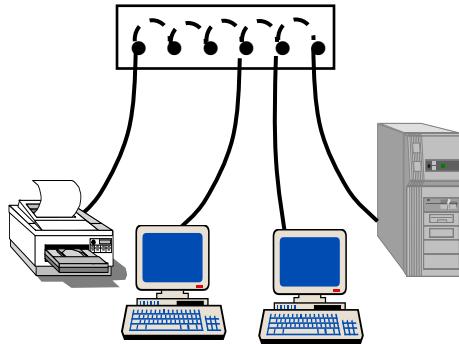
Twisted Pair Bit Rates

Data rates of 24-gauge twisted pair

Standard	Data Rate	Distance
T-1	1.544 Mbps	18,000 feet, 5.5 km
DS2	6.312 Mbps	12,000 feet, 3.7 km
1/4 STS-1	12.960 Mbps	4500 feet, 1.4 km
1/2 STS-1	25.920 Mbps	3000 feet, 0.9 km
STS-1	51.840 Mbps	1000 feet, 300 m

- Twisted pairs can provide high bit rates at short distances
- Asymmetric Digital Subscriber Loop (ADSL)
 - High-speed Internet Access
 - Lower 3 kHz for voice
 - Upper band for data
 - 64 kbps inbound
 - 640 kbps outbound
- Much higher rates possible at shorter distances
 - Strategy for telephone companies is to bring fiber close to home & then twisted pair
 - Higher-speed access + video

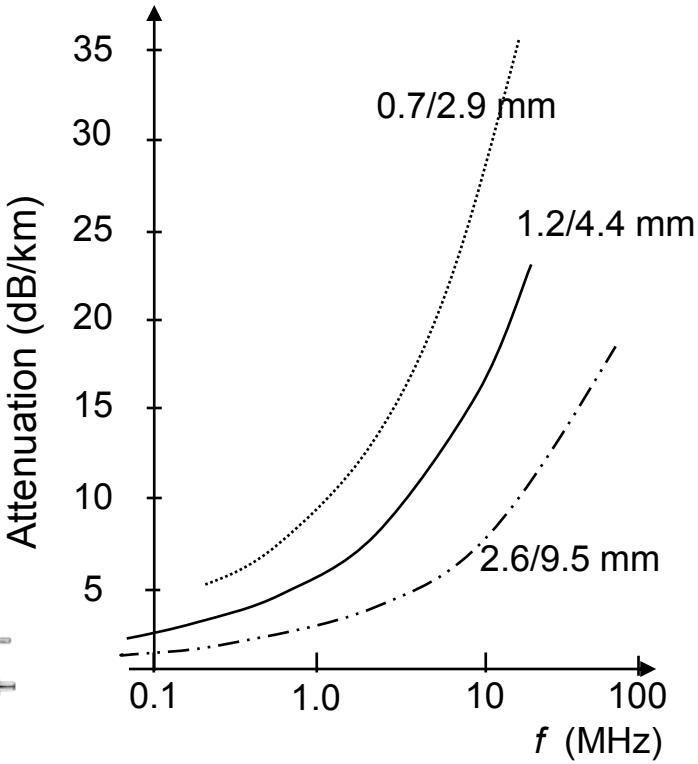
Ethernet LANs



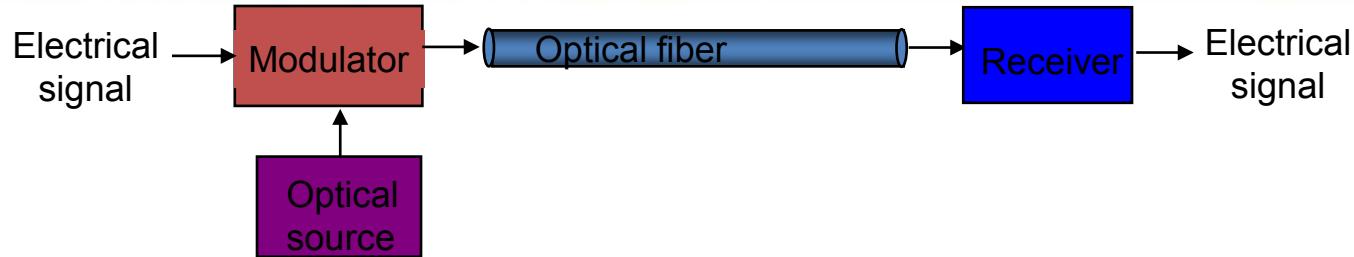
- Category 3 unshielded twisted pair (UTP): ordinary telephone wires
- Category 5 UTP: tighter twisting to improve signal quality
- Shielded twisted pair (STP): to minimize interference; costly
- 10BASE-T Ethernet
 - 10 Mbps, Baseband, Twisted pair
 - Two Category 3 UTPs
 - Manchester coding, 100 meters
- 100BASE-T4 *Fast* Ethernet
 - 100 Mbps, Baseband, Twisted pair
 - Four Category 3 UTPs
 - Three pairs for one direction at-a-time
 - 100/3 Mbps per pair;
 - Limited to 100 meters
- Category 5 & STP provide other options

Coaxial Cable

- Cylindrical braided outer conductor surrounds insulated inner wire conductor
- High interference immunity
- Higher bandwidth than twisted pair
- Hundreds of MHz
- Cable TV distribution
- Long distance telephone transmission
- Original Ethernet LAN medium



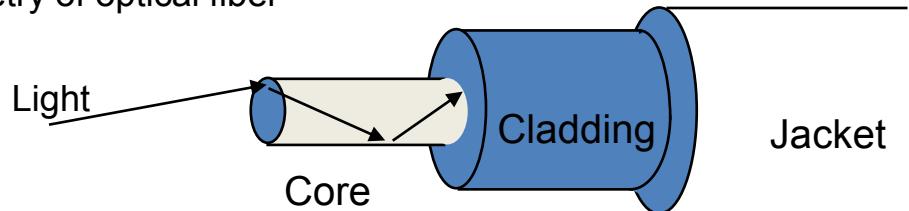
Optical Fiber



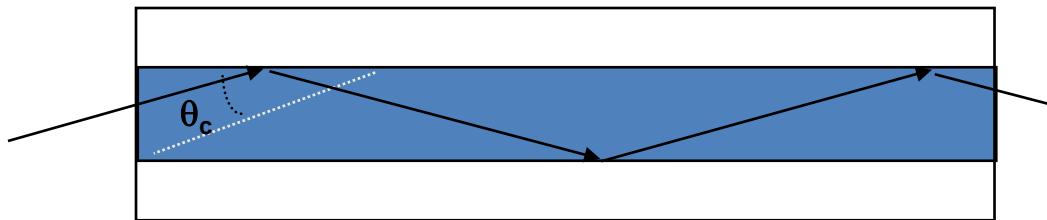
- Light sources (lasers, LEDs) generate pulses of light that are transmitted on optical fiber
 - Very long distances (>1000 km)
 - Very high speeds (>40 Gbps/wavelength)
 - Nearly error-free (BER of 10^{-15})
- Profound influence on network architecture
 - Dominates long distance transmission
 - Distance less of a cost factor in communications
 - Plentiful bandwidth for new services

Transmission in Optical Fiber

Geometry of optical fiber



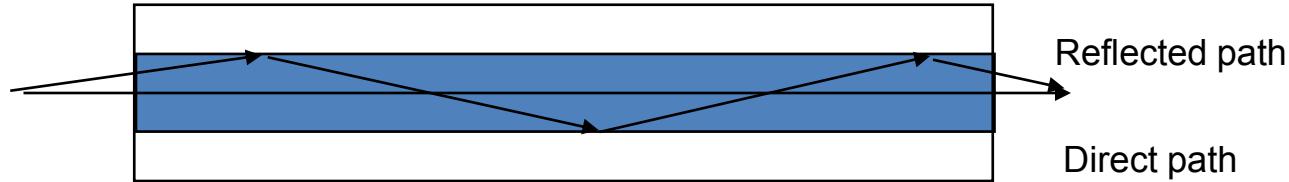
Total Internal Reflection in optical fiber



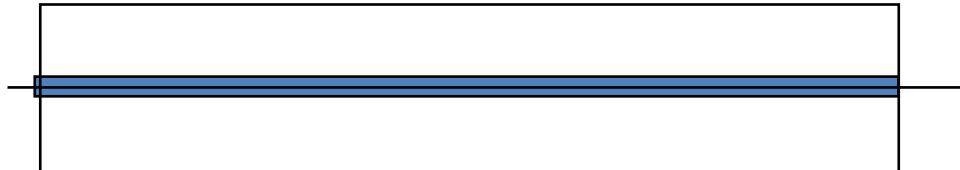
- Very fine glass cylindrical core surrounded by concentric layer of glass (cladding)
- Core has higher index of refraction than cladding
- Light rays incident at less than critical angle θ_c is completely reflected back into the core

Multimode & Single-mode Fiber

Multimode fiber: multiple rays follow different paths



Single-mode fiber: only direct path propagates in fiber



- Multimode: Thicker core, shorter reach
 - Rays on different paths interfere causing dispersion & limiting bit rate
- Single mode: Very thin core supports only one mode (path)
 - More expensive lasers, but achieves very high speeds



Optical Fiber Properties

Advantages

- ***Very low attenuation***
- ***Noise immunity***
- ***Extremely high bandwidth***
- Security: Very difficult to tap without breaking
- No corrosion
- More compact & lighter than copper wire

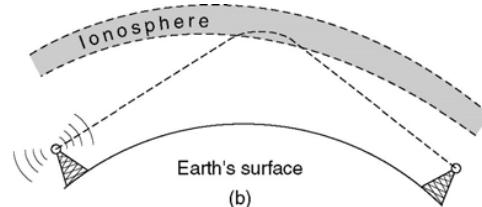
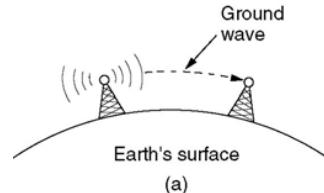
Disadvantages

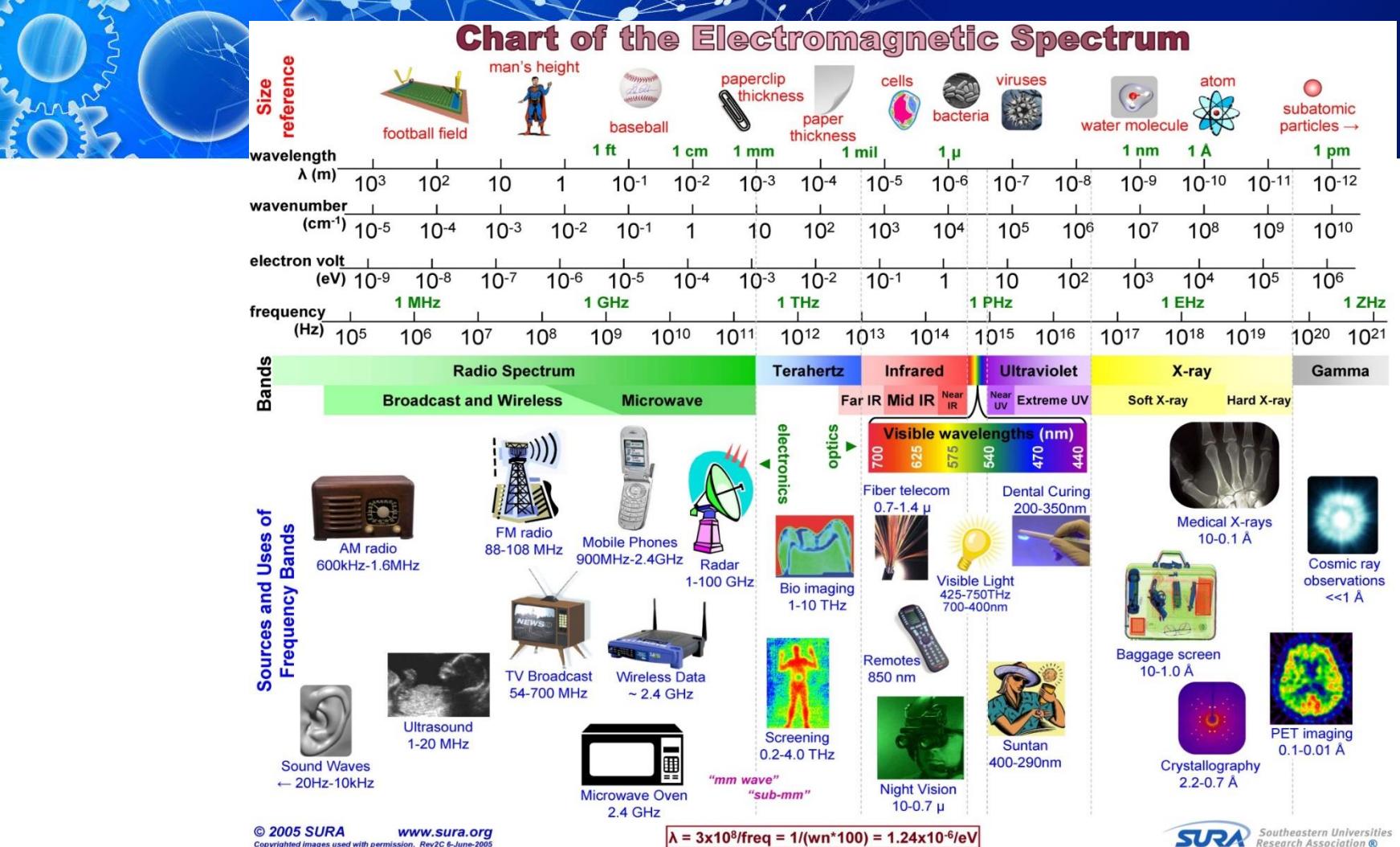
- New types of optical signal impairments & dispersion
 - Polarization dependence
 - Wavelength dependence
- Limited bend radius
 - If physical arc of cable too high, light lost or won't reflect
 - Will break
- Difficult to splice
- Mechanical vibration becomes signal noise

Radio Transmission

Radio transmission sends information from one place to another through the air. The information is sent on radio waves

- (a) In the VLF, LF, and MF bands, radio waves follow the curvature of the earth.
- (b) In the HF band, they bounce off the ionosphere







Performance



Performance

- Important metrics in networking
 - Bandwidth
 - Throughput
 - Latency (Delay)



Bandwidth (again)

- In networking, we use the term bandwidth in two contexts.
- The first, bandwidth in hertz (or, analog bandwidth), refers to the range of frequencies in a composite signal or the range of frequencies that a channel can pass.
- The second, bandwidth in bps (or, digital bandwidth), refers to the speed of bit transmission in a channel or link.
- The bandwidth of a subscriber line is 4 kHz for voice or data. The bandwidth of this line for data transmission can be up to 56,000 bps using a sophisticated modem to change the digital signal to analog.



Throughput

- The throughput is a measure of how fast we can actually send data through a network. A link may have a bandwidth of B bps, but we can only send T bps, where $T \leq B$.
- For example, we may have a link with a bandwidth of 1Mbps, but the devices connected to the end of the link may handle only 200kbps.



Example

A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

Solution

We can calculate the throughput as

$$\text{Throughput} = \frac{12,000 \times 10,000}{60} = 2 \text{ Mbps}$$

The throughput is almost one-fifth of the bandwidth in this case.



Delay

- The latency or delay defines how long it takes for an entire message to arrive at the destination from the first bit is sent out from the source.
- It is basically made of four components.
- Latency = propagation time +
transmission time +
queueing time +
processing delay



IF2230 Jaringan Komputer

Data Link Layer

Robithoh Annur
Andreas Bara Timur
Monterico Andrian

- 
- Prinsip dasar
 - Peran data link layer
 - Error handling
 - Flow control
 - Ethernet and Medium Access



Prinsip dasar (1)

- Masalah utama dalam komunikasi data: reliability. Sinyal yang dikirim melalui medium tertentu dapat mengalami pelemahan, distorsi, keterbatasan bandwidth
- Data yang dikirim dapat menjadi rusak, hilang, berubah, terduplicasi
- Tugas data link layer adalah menangani kerusakan dan hilangnya data antar 2 titik komunikasi yang terhubung oleh satu medium transmisi fisik

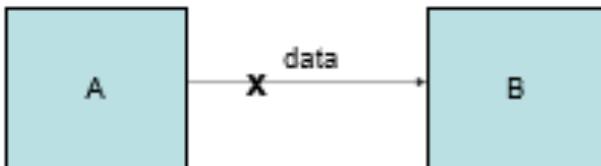


Basic Principle (1)

- The main problem in data communication: reliability. Signals sent through a certain medium can experience attenuation, distortion, bandwidth limitations.
- The data sent can be damaged, lost, changed, duplicated
- The task of the data link layer is to handle damage and loss of data between 2 communication points connected by one physical transmission medium

Prinsip dasar (2)

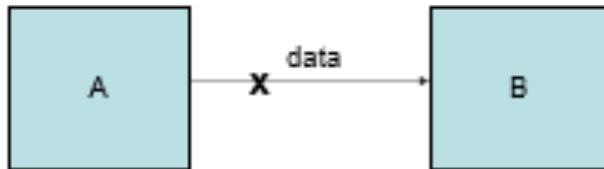
- A mengirim data ke B. Jalur antara A dan B tidak reliable, sehingga mungkin ada data yang rusak/hilang.
- Bagaimana menjamin transmisi data A ke B tetap reliable?
- A mengirim data yang panjang ke B. Data dibagi menjadi **frame**, sehingga kerusakan sebuah frame tidak merusak keseluruhan data.
- Bagaimana B dapat mendeteksi bahwa frame yang dikirim A mengalami kerusakan?
 - A menambahkan error check bits ke frame, sehingga B dapat memeriksa frame dan menentukan apakah telah terjadi perubahan



Basic Principle (2)

- A sends data to B. The path between A and B is unreliable, so there may be data corruption/loss.
- How to ensure that data transmission from A to B remains reliable?

di message yg si n kirim
ada error check bits ma

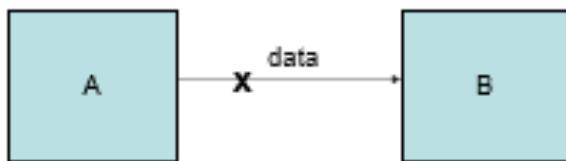


- A sends a long data to B. The data is divided into frames, so that the corruption of one **frame** does not corrupt the entire data.
- How can B detect that the frame sent by A is corrupted?
 - A adds **error check bits** to the frame, so that B can examine the frame and determine if any changes have occurred.

meski yg error cuma 1 bit,
the whole data is considered
corrupted.

Prinsip dasar (3)

- Bagaimana A mengetahui data yang dikirimnya telah diterima B?
 - B dapat mengirimkan ACK/pemberitahuan jika data diterima dengan benar, dan NAK/pemberitahuan data salah jika data rusak
 - A dapat mengirimkan ulang frame yang rusak

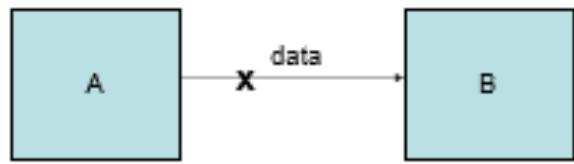


- Mengapa frame dapat hilang?
 - Bagian alamat/id/header mengalami kerusakan, sehingga frame tidak dikenali
 - Temporer disconnection
 - Apa yg terjadi jika frame dapat hilang?
 - B tidak mengetahui ada pengiriman dari A, A menunggu ACK dari B
 - B mengirimkan ACK namun hilang di jalan. A menunggu ACK dari B

Basic Principle (3)

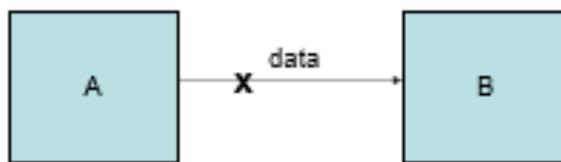
- How does A know that the data he sent has been received by B?
 - B can send ACK/acknowledgement if the data is received correctly, and NAK/error data
 - acknowledgement if the data is corrupted.
 - A can retransmit the corrupted frame.
- kalau corrupted,
A harus mengirim
kembali*

- Why can frames be lost?
 - The part of the frame containing the address/id/header is damaged, so the frame is not recognized
 - Temporary disconnection
 - What happens if a frame can be lost?
 - B is unaware of any transmission from A, A waits for an ACK from B *nunggu² terus nanti si A*.
 - B sends an ACK but it gets lost on the way. A waits for an ACK from B



Prinsip dasar (4)

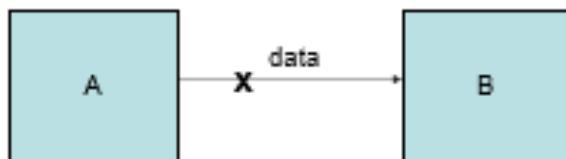
- A harus memiliki timer, yang akan mengirim ulang jika tidak menerima kabar dari B
- Jangka waktu timeout harus diatur.
 - Jika timeout terlalu cepat, A akan mengirimkan ulang sebelum ACK dari B tiba.
 - Jika timeout terlalu lama, A akan menunggu terlalu lama jika ada frame yg hilang
- A mengirim frame 1
- A mengalami timeout, dan mengirimkan ulang frame 1
- A menerima ACK, melanjutkan mengirim frame 2
- A menerima ACK kedua untuk frame 1, namun dianggap sebagai ACK untuk frame 2 (error)
 - A harus memberikan frame number, sehingga B dapat memberikan ACK spesifik untuk frame number tertentu



Basic Principle (4)

→ ada timeout! . Jadi ga nunggu² forever

- A must have a timer, which will resend if it does not receive news from B.
- The timeout period must be set.
 - If the timeout is too fast, A will retransmit before the ACK from B arrives.
 - If the timeout is too long, A will wait too long if there is a lost



- A sends frame 1
- A times out, and resends frame 1
- A receives ACK, continues sending frame 2
- A receives second ACK for frame 1, but treats it as ACK for frame 2 (error)
 - A must provide frame number, so B can provide specific ACK for a particular frame number

→ jadinya ACK nya harus dikasih nomor yg untuk tanda dia untuk frame berapa

timeoutnya
kelewat

- 
- Prinsip dasar
 - Peran data link layer
 - Framing
 - Error handling
 - Ethernet and Medium Access



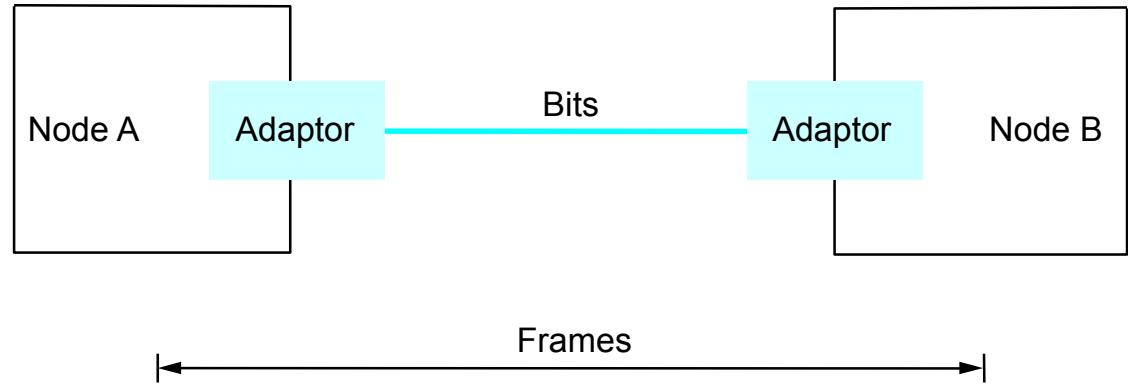
Role of Data link layer

- Physical layer determines how bits are encoded to signals sent through the transmission medium
- DL handles transmission errors, and provides services to the network layer including:
 - Error control, error detection
 - Flow control
 - Link management
 - Medium access

- Prinsip dasar
- Peran data link layer
- Error handling
- Flow control
- Ethernet and Medium Access

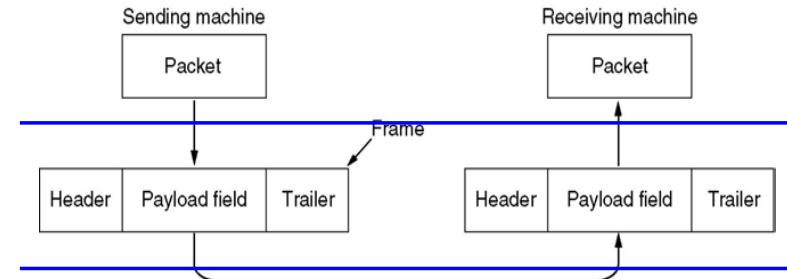
Framing

- Break sequence of bits into a frame – why?
determine where a frame starts and ends
- Typically implemented by network adaptor



Framing

- Data-link layer takes the packets from the Network Layer and encapsulates them into frames with adding control information like headers and trailers for transmission over the physical medium.
- DL performs framing:
 - Reduce the possibility of errors
 - Adjust to the physical layer
 - Limited receiver buffer capacity
 - ↳ reduce the length of buffer capacity.



how to perform framing?

- Approaches
 - Byte oriented protocols
 - Binary Synchronous Communication (**BISYNC**)
 - Digital Data Communication Message Protocol (**DDCMP**)
 - Point-to-Point Protocol (**PPP**)
 - Bit oriented protocols
 - High-Level Data Link Control (**HDLC**)
 - Clock based protocols
 - Synchronous Optical Network (**SONET**)

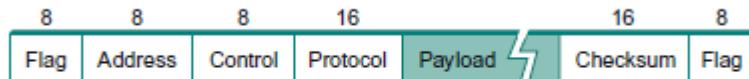
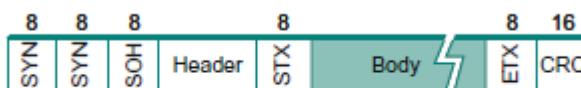
tip user
gabut tau
device varai
approach apa.
yg ditulis
cuman transmission
success / no

Byte-Oriented: Sentinel Approach



by default,
devices² tu
pakai ini
approach ini

- Add **START** and **END** sentinels to the data
- Problem: what if **END** appears in the data?
 - Add a special **DLE** (Data Link Escape) character before **END**
 - What if **DLE** appears in the data? Add **DLE** before it.
 - Similar to escape sequences in C
 - `printf("You must \"escape\" quotes in strings");`
 - `printf("You must \\\"escape\\\" forward slashes as well");`
- Used by Point-to-Point protocol, e.g. modem, DSL, cellular



■ FIGURE 2.7 BISYNC frame format.

■ FIGURE 2.8 PPP frame format.

Byte Oriented: Byte Counting

awal
→ measure
the length
of the data.



- Sender: insert length of the data in bytes at the beginning of each frame ♀: "basically datanya nanti 132"
- Receiver: extract the length and read that many bytes ♀: "ok"
- What happens if there is an error transmitting the count field?



■ FIGURE 2.9 DDCMP frame format.

Bit Oriented: Bit Stuffing

01111110

Data

01111110

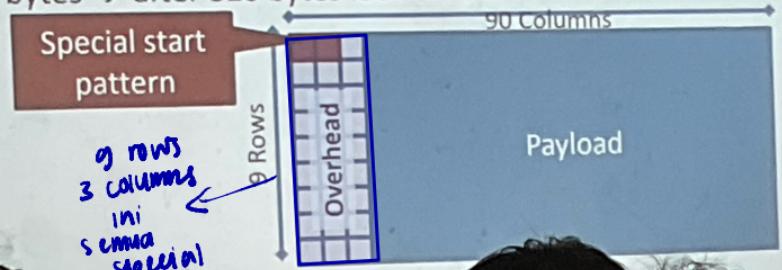
- Add sentinels to the start and end of data
 - Both sentinels are the same
 - Example: 01111110 in High-level Data Link Protocol (HDLC)
- Sender: insert a 0 after each 11111 in data
 - Known as “bit stuffing”
- Receiver: after seeing 11111 in the data...
 - 111110 → remove the 0 (it was stuffed)
 - 111111 → look at one more bit
 - 1111110 → end of frame
 - 1111111 → error! Discard the frame
- Disadvantage: 20% overhead at worst
- What happens if error in sentinel transmission?



■ FIGURE 2.10 HDLC frame format.

Clock-based Framing: SONET

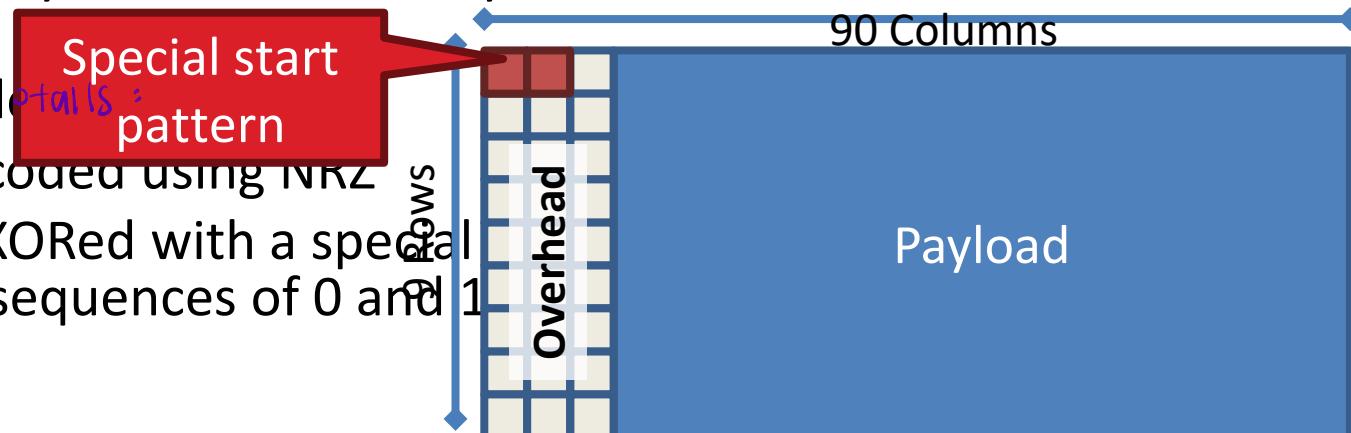
- Synchronous Optical Network
 - Transmission over very fast optical links
 - STS- n , e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
 - $9 \times 90 = 810$ bytes → after 810 bytes look for start pattern



9 rows
3 columns
ini
semia
special
pattern.

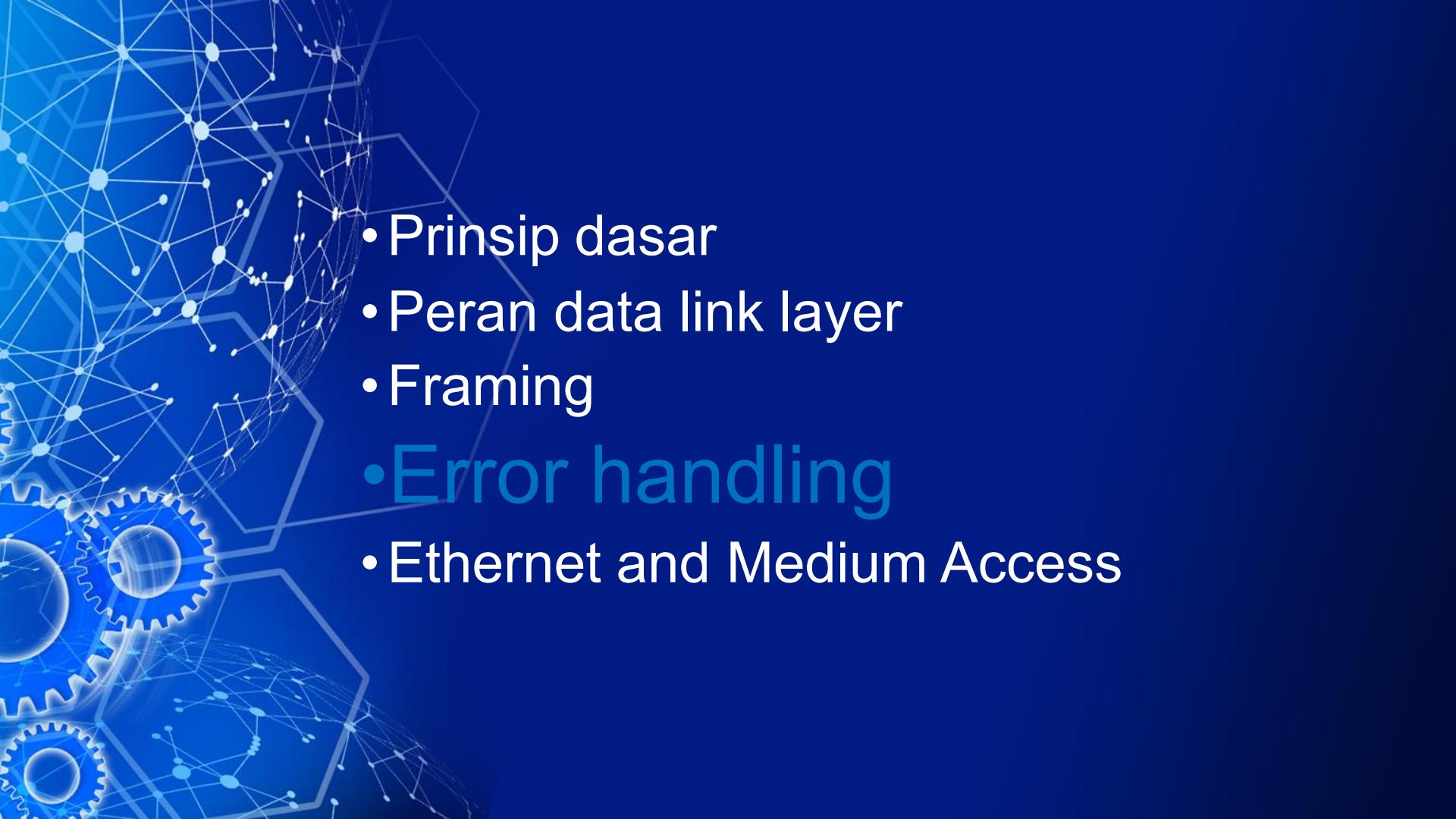
Clock-based Framing: SONET

- Synchronous Optical Network
 - Transmission over very fast optical links
 - STS- n , e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
 - $9 \times 90 = 810$ bytes → after 810 bytes look for start pattern
- Physical layer details:
 - Bits are encoded using NRZ
 - Payload is XORed with a special pattern to avoid long sequences of 0 and 1



Clock-based Framing: SONET

- Synchronous Optical Network
 - Transmission over very fast optical links
 - STS- n , e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
 - $9 \times 90 = 810$ bytes → after 810 bytes look for start pattern
- Physical layer details
 - Bits are encoded using NRZ
 - Payload is XORed with a special 127-bit pattern to avoid long sequences of 0 and 1

- 
- Prinsip dasar
 - Peran data link layer
 - Framing
 - **Error handling**
 - Ethernet and Medium Access

The background features a complex, abstract design on a dark blue gradient. It includes a large, semi-transparent sphere composed of numerous small white dots connected by thin lines, resembling a molecular or network structure. Overlaid on this are several interlocking blue and white gears of varying sizes, suggesting mechanical or data processing elements.

Error Detection and Correction



Dealing with Noise

- The physical world is inherently noisy
 - Interference from electrical cables
 - Cross-talk from radio transmissions, microwave ovens
 - Solar storms
- How to detect bit-errors in transmissions?
- How to recover from errors?



Error Detection and Correction

- Umumnya penanganan error transmisi dilakukan pada data link layer. Error control dapat pula dilakukan pada layer lain (physical/higher layer)
- Error dapat diperbaiki dan dideteksi dengan menggunakan data redundant/tambahan pada setiap pengiriman data
- Jenis error:
 - Single bit error: hanya sebuah bit yang berubah. Disebabkan oleh white noise
 - Burst error: sederetan bit-bit mengalami error. Disebabkan oleh impulse noise
- Makin tinggi data rate, makin besar efeknya



Naïve Error Detection

- Idea: send two copies of each frame
 - if (memcmp(frame1, frame2) != 0) { OH NOES, AN ERROR! }
- Why is this a bad idea?
 - Extremely high overhead
 - Poor protection against errors
 - Twice the data means twice the chance for bit errors



Channel Coding

- Channel coding is most often applied to communications links in order to improve the reliability of the information being transferred.
- By adding **additional bits** to the transmitted data stream, it is possible to **detect and even correct** for errors in the receiver.
- The added coding bits lower the raw data transmission rate through the channel (Coding **expands** the occupied **bandwidth** for a particular message data rate).
- Channel codes that are used to detect errors are called **error detection codes**, while codes that can detect and correct errors are called **error correction codes**. There are three general types of channel codes:-
Block codes, Convolutional codes and Concatenated Codes

- Error Detection
 - In its most elementary form this involves recognizing which part of the received information is in error.
- Error Detection and Correction
 - With added complexity, it is possible not only to detect errors, but also to build in the ability to correct errors without recourse to retransmission.
 - This is particularly useful where there is no feedback path to the transmission source with which to request a resend. This process is known as FEC (Forward Error Correction).



Error Detection Schemes – Parity

- One of the simplest yet most frequently used techniques for detecting errors is the **parity check bit**.
- The parity check bit is usually a **single bit** (1 or 0) appended to the end of a data word such that the number of 1s in the new data word is even for **even parity**, or odd for **odd parity**.
- On **reception**, each data word, with appended parity bit, is **checked** to see how many **1s** are present. For an even parity design, the number must be even. If it is found to be odd, it can be concluded that at least one error has occurred during transmission and the **Automatic Repeat Request (ARQ) process** can begin. Of course, if two bits are in error, the parity check will pass, and the errors will go undetected.

7 bits of data (count of 1 bits)	8 bits including parity	
	even	odd
0000000 (0)	00000000 (0)	10000000 (1)
1010001 (3)	11010001 (4)	01010001 (3)
1101001 (4)	01101001 (4)	11101001 (5)
1111111 (7)	11111111 (8)	01111111 (7)



Exercise 1

- Data given: “AZ15” (Use Parity Check Bit – even parity)
- A -> 0x41 -> 1000001 -> 010000010
- Z -> 0x5A -> 1011010 -> 010110100
- 1 -> 0x31 -> 0110001 -> 001100011
- 5 -> 0x35 -> 0110101 -> 001101010



Checksums

- Idea:
 - Add up the bytes in the data
 - Include the sum in the frame



- Use ones-complement arithmetic
- Lower overhead than parity: 16 bits per frame
- But, not resilient to errors
 - Why? $1\ 101001 + 0\ 101001 = 10010010$
- Used in UDP, TCP, and IP



Error Detection Schemes – Checksum

The Sender follows these steps:

- data is divided into k segments, each of n bits.
- All segments are added using one's complement toget the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data

The Receiver follows these steps:

- data is divided into k segments, each of n bits.
- All sections are added using one's complement toget the sum.
- The sum is complemented.
- If the result is zero, the data are accepted:otherwise, rejected.

Error Detection Schemes – Checksum

- Suppose the following block of 16 bits is to be sent using a checksum of 8 bits. 10101001 00111001. The numbers are added using one's complement:

10101001
00111001
00000000

Sum 11100010
Checksum 00011101

- The pattern sent is 10101001 00111001 00011101
- Now suppose the receiver receives the pattern with no error.
10101001 00111001 00011101
- When the receiver adds the three blocks, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.

10101001
00111001
00011101
Sum 11111111
Complement 00000000 means that the pattern is OK.



Error Detection Schemes – CRC

- One of the most common, and one of the most powerful, error-detecting codes is the **Cyclic Redundancy Check (CRC)**. CRC is well suited for detecting **burst errors** and is particularly **easy to implement** in hardware, and is therefore commonly used in digital networks and storage devices such as hard disk drives.
- Implementation of CRC can be described as follow:
 - Given a **k** bit block of bits, or message, the transmitter generates an $(n - k)$ -bit sequence, known as a **Frame Check Sequence (FCS)**, such that the resulting frame, consisting of n bits, is exactly divisible by some predetermined number.
 - The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.



Error Detection Schemes – CRC

- CRC is a **block code** which uses a **shift register** to perform encoding and decoding.
- The code word with n bits is expressed as

$$c(x) = c_1x^{n-1} + c_2x^{n-2} + \dots + c_nx^0$$

where each c_i is either a 1 or 0.

$$c(x) = m(x)x^{n-k} + c_p(x)$$

where $c_p(x)$ = remainder from dividing $m(x)x^{n-k}$ by generator $g(x)$

- If the received signal is $c(x) + e(x)$, where $e(x)$ is the error,
 - To check if received signal is error free, the remainder (syndrome) from dividing $c(x) + e(x)$ by $g(x)$ is obtained.
 - If this is 0 then the received signal is considered error free else error pattern is detected from known error syndromes.



CRC – Code Creation & Detection

- The **CRC creation** process is defined as follows:
 - Get the block of raw message (k bits).
 - Left shift the raw message by $(n-k)$ bits and then divide it by G (r bits).
 - Get the remainder R as FCS (number of bits, $r=n-k$ bits)
 - Append the R to the raw message . The result (n bits) is the frame to be transmitted.
- **CRC** is **checked** using the following process:
 - Receive the frame (n bits)
 - Divide it by G (r bits)
 - Check the remainder. If the remainder is not zero, then there is an error in the frame.
- These procedures can be further clarified in three ways: **modulo 2 arithmetic**, **polynomials**, and **digital logic**.



CRC – Modulo 2 Arithmetic

- Modulo-2 arithmetic is in fact just the exclusive-OR (XOR) operation.

$$\begin{array}{r} 1111 \\ + \underline{1010} \\ \hline 0101 \end{array} \quad \begin{array}{r} 1111 \\ - \underline{0101} \\ \hline 1010 \end{array}$$

- Now define
 - $T = n$ -bit frame to be transmitted
 - $D = k$ -bit block of data, or message, the first k bits of T
 - $F = (n - k)$ -bit FCS, the last $(n - k)$ bits of T
 - P = pattern of $n - k + 1$ bits; this is the predetermined divisor

- $T_{(n \text{ bits})} = D_{(k \text{ bits})} | F_{(n-k \text{ bits})}$

$$\Rightarrow T = 2^{(n-k)}D + F$$

By multiplying D by $2^{(n-k)}$, we have in effect shifted it to the left by $(n - k)$ bits and padded out the result with zeroes. Adding F yields the concatenation of D and F , which is T .

CRC – Modulo 2 Arithmetic Example

1. Given

Message $D = 1010001101$ (10 bits)

Pattern $P = 110101$ (6 bits)

FCS R = to be calculated (5 bits)

Thus, $n = 15$, $k = 10$, and $(n - k) = 5$.

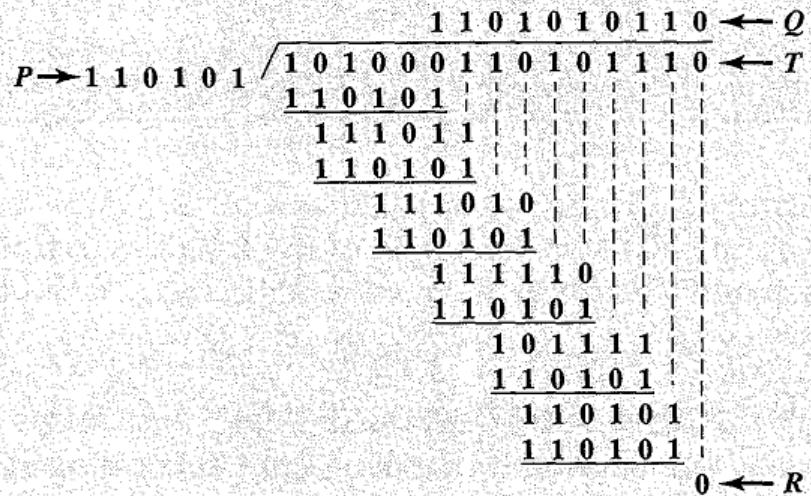
2. The message is multiplied by 2^5 , yielding 101000110100000.

3. This product is divided by P :

$$\begin{array}{r}
 P \rightarrow 110101 / \quad \begin{array}{r}
 11010101010110 \\
 1010001101000000
 \end{array} \leftarrow Q \\
 \hline
 110101 \\
 111011 \\
 \hline
 110101 \\
 111010 \\
 \hline
 110101 \\
 111110 \\
 \hline
 110101 \\
 111110 \\
 \hline
 110101 \\
 1011100 \\
 \hline
 110101 \\
 110101 \\
 \hline
 110101 \\
 110010 \\
 \hline
 110101 \\
 011110 \leftarrow R
 \end{array}$$

CRC – Modulo 2 Arithmetic Example (cont)

4. The remainder is added to 2^5D to give $T = 101000110101110$, which is transmitted.
5. If there are no errors, the receiver receives T intact. The received frame is divided by P :



Because there is no remainder, it is assumed that there have been no errors.



CRC – Polynomials

- In polynomials approach, the CRC process can be viewed by expressing all values as **polynomials in a dummy variable X**, with **binary coefficients**. The coefficients correspond to the bits in the binary number.
 - A **generator polynomial** with **constant coefficients**, is used as the divisor in a **polynomial long division** over a finite field.
 - Taking the **input data** (blocks of input bits called **message polynomial**) as the dividend.
 - Where the **remainder** (the **FCS**) becomes the result.
- **Even parity** is a special case of a **single-bit CRC**, where the **single-bit FCS** is generated by the generator polynomial (divisor) $x+1$.



CRC - Polynomials

- Message polynomial is divided by Generator polynomial giving quotient and remainder, the coefficients of the remainder form the bits of final CRC.
- Define:
 - M – The original frame (k bits) to be transmitted before adding the Frame Check Sequence (FCS).
 - F – The resulting FCS of r bits to be added to M (usually r=8, 16, 32).
 - T – The cascading of M and F.
 - G – The predefined CRC generating polynomial with pattern of r+1 bits..
- The main idea in CRC algorithm is that the FCS is generated so that the remainder of T/G is zero.

CRC – Polynomials Example

0 1 1 1

1 0 1 1

Example : Find the codewords $c(x)$ if $m(x)=x^2+x+1$ and $g(x)=x^3+x+1$ for (7,4) CRC.

We have n = Total number of bits = 7, k = Number of information bits = 4,
 r = Number of parity bits = $n - k = 3$.

$$\begin{aligned}\therefore c_p(x) &= \text{rem} \left[\frac{m(x)x^{n-k}}{g(x)} \right] = \text{rem} \left[\frac{(x^2 + x + 1)x^3}{x^3 + x + 1} \right] \\ &= \text{rem} \left[\frac{x^5 + x^4 + x^3}{x^3 + x + 1} \right] = x\end{aligned}$$

m(x) has been
shifted left for r bits
=> 0 1 1 1 0 0 0

Then,

$$c(x) = m(x)x^{n-k} + c_p(x) = x^5 + x^4 + x^3 + x$$

0 1 1 1 0 1 0

If the received bits = transmitted bits = 0 1 1 1 0 1 0,

$$\frac{T}{G} = \text{rem} \left[\frac{x^5 + x^4 + x^3 + x}{x^3 + x + 1} \right] = 0 \quad \Rightarrow \text{No Error !}$$



Error Detection Schemes – CRC Capabilities

- An error $E(X)$ will only be undetectable if it is divisible by $G(X)$. It can be shown that all of the following errors are not divisible by a suitably chosen $G(X)$ and hence are **detectable**:
 - All single-bit errors, if $G(X)$ has more than one non-zero term.
 - All double-bit errors, as long as $G(X)$ has a factor with at least three terms.
 - Any odd number of errors, as long as $P(X)$ contains a factor $(X + 1)$
 - Any burst error for which the length of the burst is less than or equal to $(n - k)$; that is, less than or equal to the length of the FCS
 - A fraction of error bursts of length $n - k + 1$; the fraction equals $(1 - 2^{-(n-k-1)})$
 - A fraction of error bursts of length greater than $n - k + 1$; the fraction equals $(1 - 2^{-(n-k)})$



Error Detection Schemes – CRC

- Common CRC Codes

Code	Generator polynomial $g(x)$	Parity check bits
CRC-12	$1+x+x^2+x^3+x^{11}+x^{12}$	12
CRC-16	$1+x^2+x^{15}+x^{16}$	16
CRC-CCITT	$1+x^5+x^{15}+x^{16}$	16

- The CRC-12 system is used for transmission of streams of 6-bit characters and generates a 12-bit FCS.
- Both CRC-16 and CRC-CCITI are popular for 8-bit characters, in the United States and Europe, respectively, and both result in a 16-bit FCS.



Exercise 3

- Data given: “AZ15” (Use CRC-16)
- A -> 0x41 -> 01000001
- Z -> 0x5A -> 01011010
- 1 -> 0x31 -> 00110001
- 5 -> 0x35 -> 00110101

x^{31}	x^{30}	x^{29}	x^{28}	x^{27}	x^{26}	x^{25}	x^{24}		x^0
0	1	0	0	0	0	0	1	...	1

$$\begin{aligned} R(x) &= rem\left[\frac{m(x)x^{n-k}}{g(x)}\right] = rem\left[\frac{(x^{30} + x^{24} + \dots + x^2 + 1)x^{16}}{x^{16} + x^{15} + x^2 + 1}\right] \\ &= rem\left[\frac{x^{46} + x^{40} + \dots + x^{18} + x^{16}}{x^{16} + x^{15} + x^2 + 1}\right] = x^{13} + x^{12} + x^{10} + x^9 + x^6 + x^5 + x^3 + x + 1 \end{aligned}$$



$$R(x) = x^{13} + x^{12} + x^{10} + x^9 + x^6 + x^5 + x^3 + x + 1$$

x^{15}	x^{14}	x^{13}	x^{12}	x^{11}	x^{10}	x^9	x^8	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
0	0	1	1	0	1	1	0	0	1	1	0	1	0	1	1

- CRC-16: 0x366B
- Codeword (in HEX): 0x415A3135**366B**

- Kode Hamming merupakan kode non-trivial untuk koreksi kesalahan yang pertama kali diperkenalkan.
- Kode ini dan variansinya telah lama digunakan untuk kontrol kesalahan pada sistem komunikasi digital.
- Kode Hamming biner dapat direpresentasikan dalam bentuk persamaan:
$$(n,k) = (2^m - 1, 2^m - 1 - m)$$



- Contoh:

jika $m = \text{jumlah paritas} = 3$

$k = \text{jumlah data} = 4$

$n = \text{jumlah bit informasi yang membentuk n sandi} = 7$

maka kode Hamming nya adalah C (7,4) dengan $d_{\min} = 3$

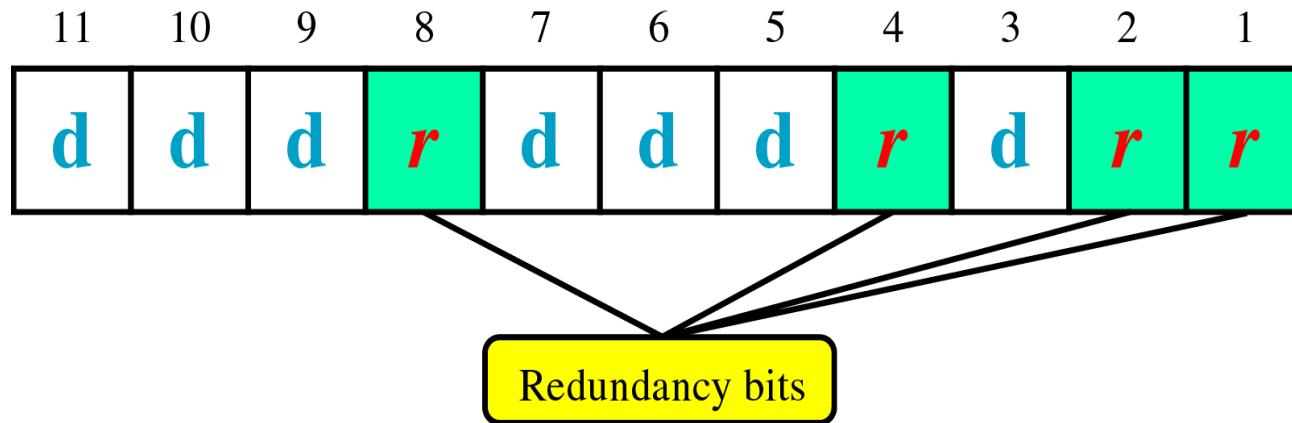


Forward Error Correction

- Error Correcting codes dinyatakan sebagai penerusan koreksi kesalahan untuk mengindikasikan bahwa pesawat penerima sedang mengoreksi kesalahan.
- Kode pendekripsi yang paling banyak digunakan merupakan kode Hamming.
- Posisi bit-bit Hamming dinyatakan dalam 2^n dengan n bilangan bulat sehingga bit-bit Hamming akan berada dalam posisi 1, 2, 4, 8, 16, dst..

Error Correction(cont'd)

- Hamming Code
 - ~ developed by R.W.Hamming
- positions of redundancy bits in





The key to the Hamming Code is the use of extra parity bits to allow the identification of a single error. Create the code word as follows:

- Mark all bit positions that are powers of two as parity bits. (positions 1, 2, 4, 8, 16, 32, 64, etc.)
- All other bit positions are for the data to be encoded. (positions 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)
- Each parity bit calculates the parity for some of the bits in the code word. The position of the parity bit determines the sequence of bits that it alternately checks and skips.

Position 1: check 1 bit, skip 1 bit, check 1 bit, skip 1 bit, etc. (1,3,5,7,9,11,13,15,...)

Position 2: check 2 bits, skip 2 bits, check 2 bits, skip 2 bits, etc. (2,3,6,7,10,11,14,15,...)

Position 4: check 4 bits, skip 4 bits, check 4 bits, skip 4 bits, etc.

(4,5,6,7,12,13,14,15,20,21,22,23,...)

Position 8: check 8 bits, skip 8 bits, check 8 bits, skip 8 bits, etc. (8-15,24-31,40-47,...)

Position 16: check 16 bits, skip 16 bits, check 16 bits, skip 16 bits, etc. (16-31,48-63,80-95,...)

Position 32: check 32 bits, skip 32 bits, check 32 bits, skip 32 bits, etc. (32-63,96-127,160-191,...)

etc.

- Set a parity bit to 1 if the total number of ones in the positions it checks is odd. Set a parity bit to 0 if the total number of ones in the positions it checks is even.



Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Encoded data bits	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15
Parity bit coverage	p1	X	X	X		X	X		X	X	X	X	X	X	X	X	X	X		
p2		X	X			X	X		X	X			X	X			X	X		
p3				X	X	X	X					X	X	X	X					X
p4								X	X	X	X	X	X	X	X					
p5																X	X	X	X	X



Example:

A byte of data: **10011010**

Create the data word, leaving spaces for the parity bits: **_ _ 1 _ 0 0 1 _ 1 0 1 0**

Calculate the parity for each parity bit (a ? represents the bit position being set):

Position 1 checks bits 1,3,5,7,9,11:

? _ 1 _ 0 0 1 _ 1 0 1 0. Even parity so set position 1 to a 0: **0 _ 1 _ 0 0 1 _ 1 0 1 0**

Position 2 checks bits 2,3,6,7,10,11:

0 ? 1 _ 0 0 1 _ 1 0 1 0. Odd parity so set position 2 to a 1: **0 1 1 _ 0 0 1 _ 1 0 1 0**

Position 4 checks bits 4,5,6,7,12:

0 1 1 ? 0 0 1 _ 1 0 1 0. Odd parity so set position 4 to a 1: **0 1 1 1 0 0 1 _ 1 0 1 0**

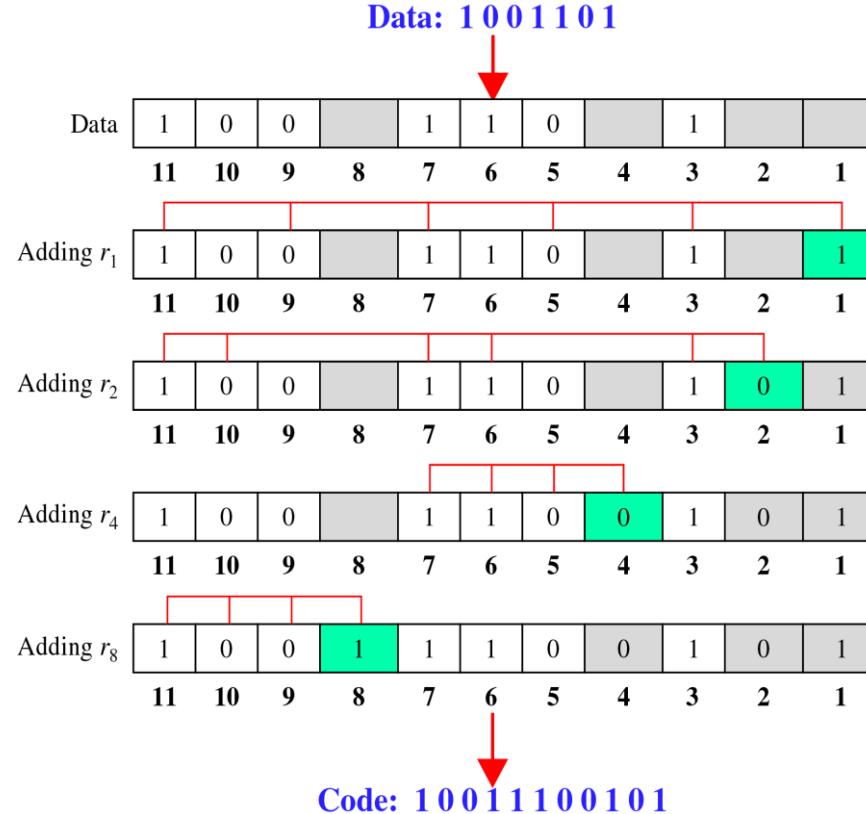
Position 8 checks bits 8,9,10,11,12:

0 1 1 1 0 0 1 ? 1 0 1 0. Even parity so set position 8 to a 0: **0 1 1 1 0 0 1 0 1 0 1 0**

Code word: 011100101010.

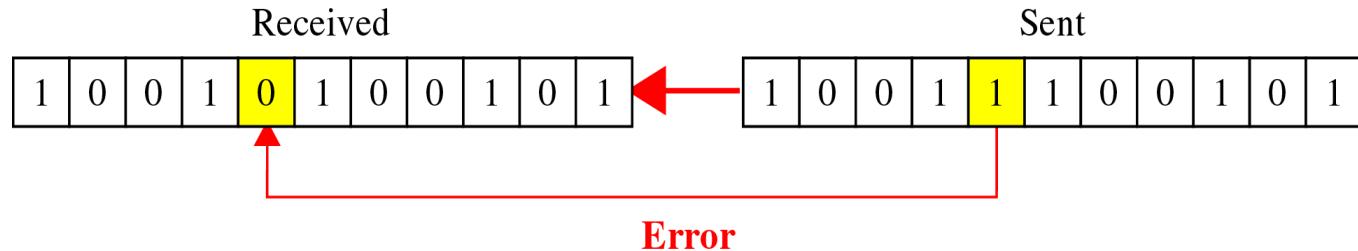
Error Correction(cont'd)

- Calculating the r values



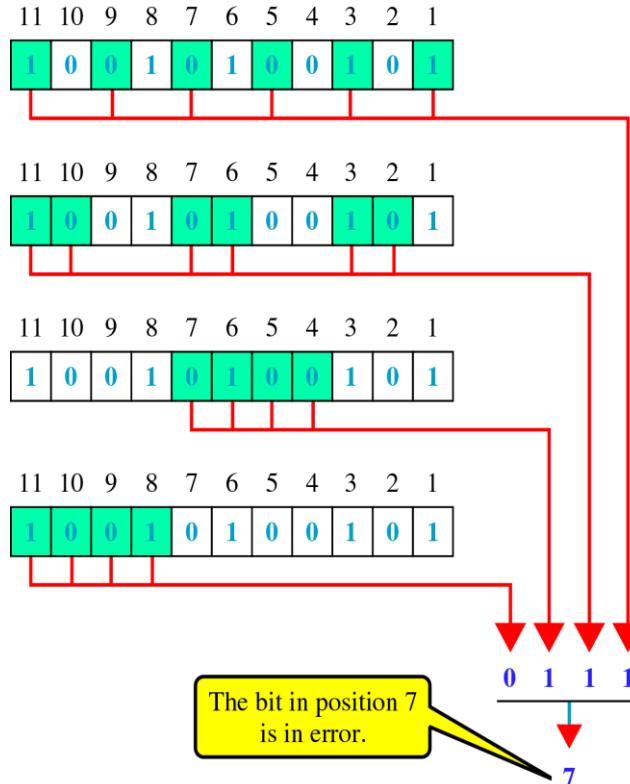
Contoh Error Correction(cont'd)

- Error Detection and Correction



Error Correction(cont'd)

- Error detection using Hamming Code





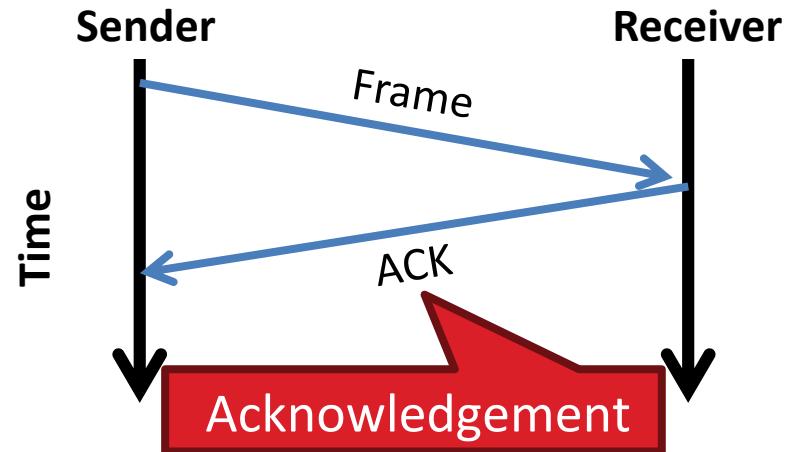
Error Correction(cont'd)

- Multiple-Bit Error Correction
 - redundancy bits calculated on overlapping sets of data units can also be used to correct multiple-bit errors.

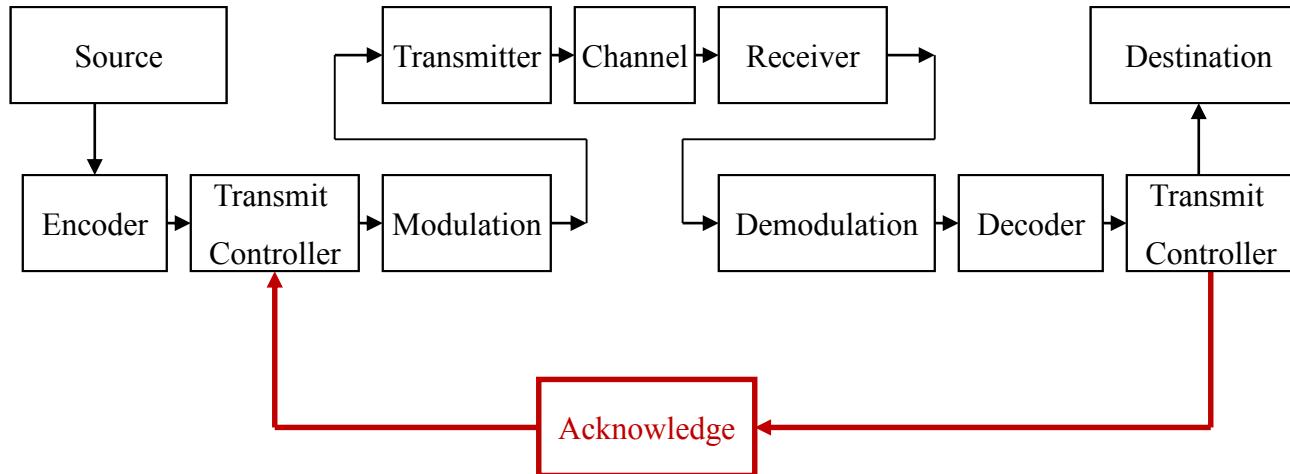
Ex) to correct double-bit errors, we must take into consideration that two bits can be a combination of any two bits in the entire sequence

What About Reliability?

- How does a sender know that a frame was received?
 - What if it has errors?
 - What if it never arrives at all?



Channel Coding – Automatic Repeat Request Systems (ARQ)



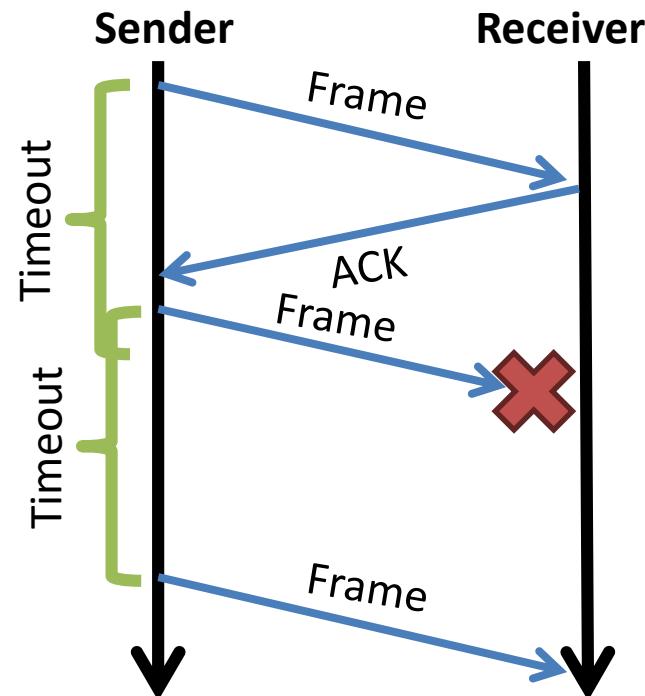
- Stop and Wait ARQ
- Sliding Window

Stop and Wait

- Simplest form of reliability
- Example: Bluetooth
- Problems?
 - Utilization
 - Can only have one frame in flight at any time
- 10Gbps link and 10ms delay
 - Assume packets are 1500B

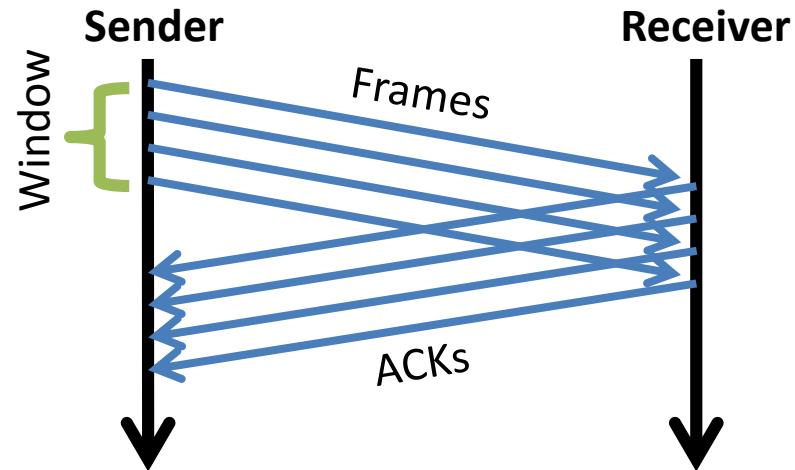
$$1500B * 8\text{bit} / (2 * 10\text{ms}) = 600\text{Kbps}$$

Utilization is 0.006%



Sliding Window

- Allow multiple outstanding, un-ACKed frames
- Number of un-ACKed frames is called the **window**



- Made famous by TCP
 - We'll look at this in more detail later



Should We Error Check in the Data Link?

- Recall the End-to-End Argument
- Cons:
 - Error free transmission cannot be guaranteed
 - Not all applications want this functionality
 - Error checking adds CPU and packet size overhead
 - Error recovery requires buffering
- Pros:
 - Potentially better performance than app-level error checking
- Data link error checking in practice
 - Most useful over lossy links
 - Wifi, cellular, satellite

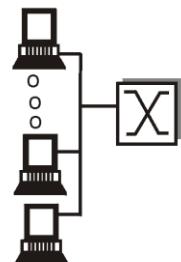
- Prinsip dasar
 - Peran data link layer
 - Framing
 - Error handling
- Ethernet and Medium Access



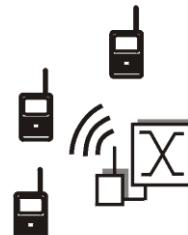
Multiple Access Links and LANs

Two types of “links”:

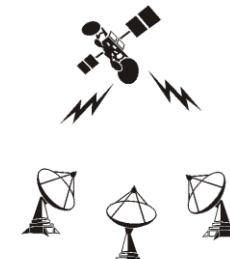
- point-to-point, e.g.,
 - PPP for dial-up access, or over optical fibers
- broadcast (shared wire or medium), e.g.
 - traditional Ethernet
 - 802.11 wireless LAN



shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)



satellite



cocktail party

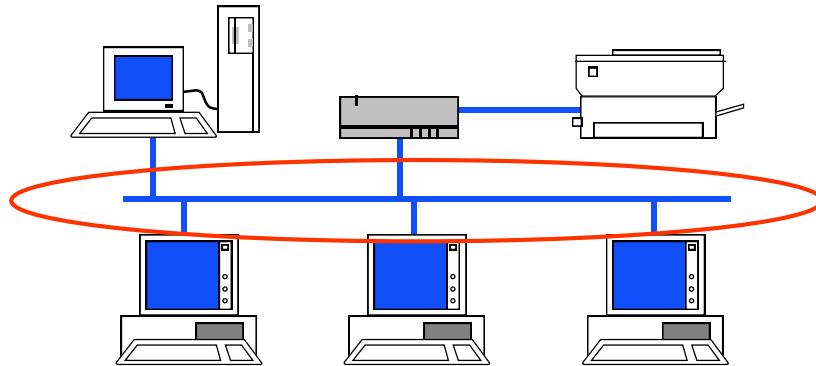


Broadcast Links: Multiple Access

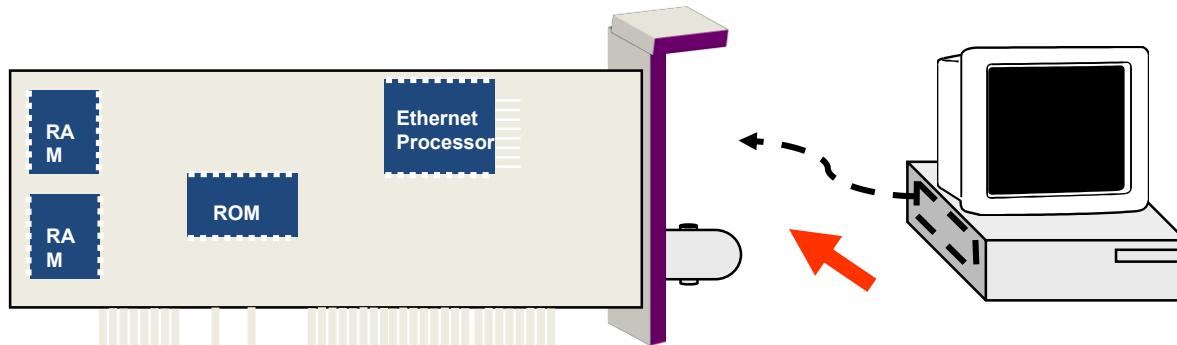
Single shared communication channel

- Only one can send successfully at a time
- Two or more simultaneous transmissions
 - interference!
- How to share a broadcast channel
 - media access control uses same shared media
- Humans use multi-access protocols all the time

Typical LAN Structure

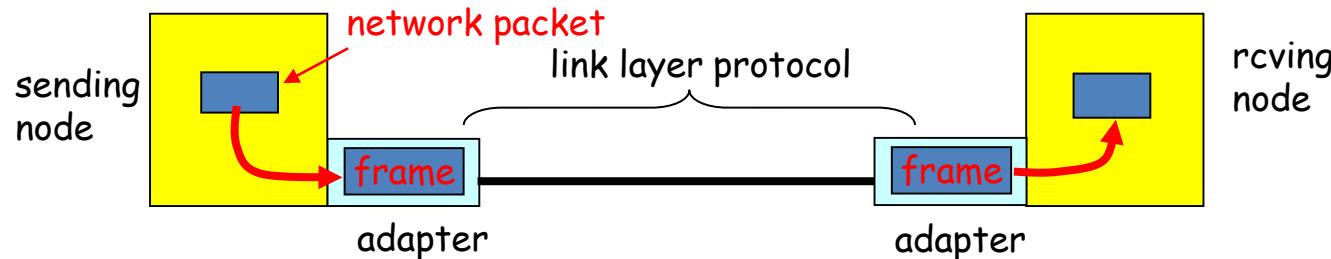


- Transmission Medium
- Network Interface Card (NIC)
- Unique MAC "physical" address



Adaptors Communicating

- link layer implemented in “adaptor” (aka NIC), with “transceiver” in it
 - Ethernet card, dial-up modem, 802.11 wireless card
- sending side:
 - encapsulates packet in frame
 - adds error checking bits, flow control, reliable data transmission, etc.
- receiving side
 - looks for errors, flow control, reliable data transmission, etc
 - extracts packet, passes to receiving node
- data link & physical layers are closely coupled!





MAC (Physical) Addresses

- Addressing needed in shared media
 - MAC (media access control) or physical addresses
 - To identify source and destination interfaces and get frames delivered from one interface to another physically-connected interface (i.e., on same physical local area network!)
- 48 bit MAC address (for most LANs)
 - fixed for each adaptor, burned in the adapter ROM
 - MAC address allocation administered by IEEE
 - 1st bit: 0 unicast, 1 multicast.
 - all 1's : broadcast
- MAC flat address -> portability
 - can move LAN card from one LAN to another



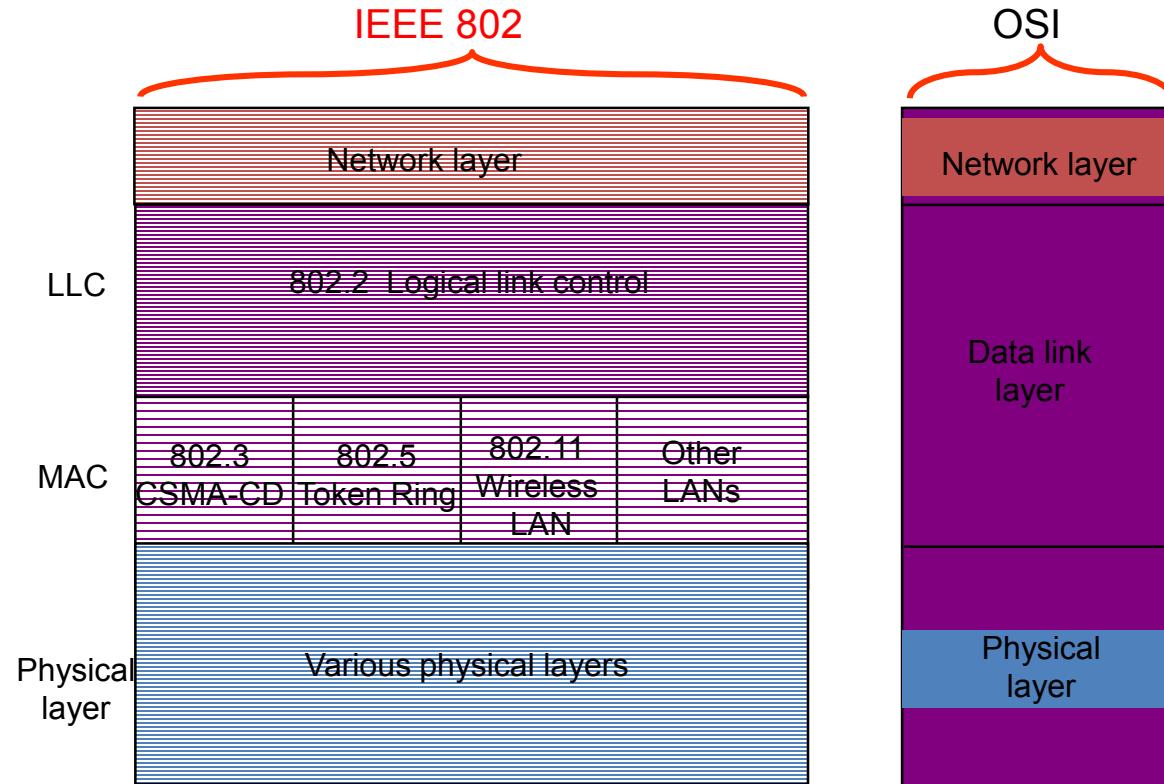
MAC (Physical, or LAN) Addresses

MAC addressing operations on a LAN:

- each adaptor on the LAN “sees” all frames
- accept a frame **only if dest. (unicast) MAC address matches its own MAC address**
- accept all **broadcast** (MAC= all 1’s) frames
- accept all frames if set in “**promiscuous**” mode
- can configure to accept certain **multicast addresses** (first bit = 1)



MAC Sub-layer



Ethernet Overview

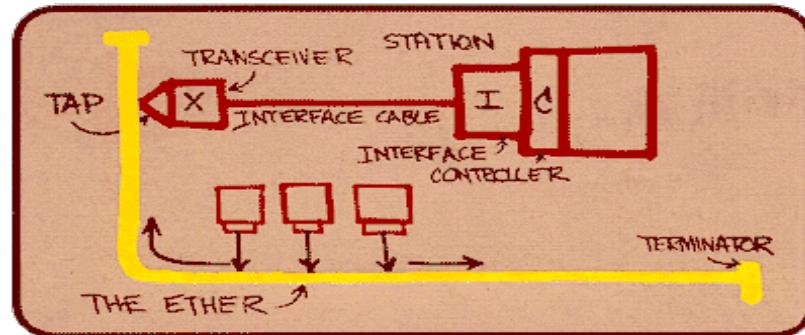
- History
 - developed by Xerox PARC in mid-1970s
 - roots in Aloha packet-radio network
 - standardized by Xerox, DEC, and Intel in 1978
 - similar to **IEEE 802.3** standard
- CSMA/CD
 - carrier sense
 - multiple access
 - collision detection
- Frame format
 - 64
 - 48
 - 48
 - 16
 - 32



Ethernet

“Dominant” LAN technology:

- cheap \$20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token ring LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps



Metcalfe's Ethernet
sketch

Ethernet Frame Format

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**

DIX frame format

8 bytes	6	6	2	0-1500	4
Preamble	Dest addr	Src addr	Type	Data	CRC

IEEE 802.3 format

8 bytes	6	6	2	0-1500	4
Preamble	Dest addr	Src addr	Length	Data	CRC

- Ethernet has a maximum frame size: data portion ≤ 1500 bytes
 - It imposes a minimum frame size: 64 bytes (excluding preamble)
If data portion < 46 bytes, pad with "junk" to make it 46 bytes
- Q: Why minimum frame size in Ethernet?**



Fields in Ethernet Frame Format

- **Preamble:**
 - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 (SoF: start-of-frame)
 - used to synchronize receiver, sender clock rates, and identify beginning of a frame
- **Addresses:** 6 bytes
 - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to network layer protocol (specified by TYPE field)
 - otherwise, adapter discards frame
- **Type:** indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk
 - 802.3: Length gives data size; “protocol type” included in data
- **CRC:** checked at receiver, if error is detected, the frame is simply dropped



IEEE 802.3 MAC: Ethernet

MAC Protocol:

- CSMA/CD
- Truncated binary exponential backoff
 - for retransmission n: $0 < r < 2^k * \text{ASlotTime}$, where $k=\min(n,10)$
 - give up after 16 retransmissions
- *Slot Time* is the critical system parameter
 - upper bound on time to detect collision
 - upper bound on time to acquire channel
 - upper bound on length of frame segment generated by collision
 - quantum for retransmission scheduling
 - $\max\{\text{round-trip propagation}, \text{MAC jam time}\}$



IEEE 802.3 Parameters

- 1 bit time = time to transmit one bit
 - 10 Mbps → 1 bit time = $0.1 \mu\text{s}$
- Maximum network diameter $\leq 2.5\text{km}$
 - Maximum 4 repeaters
- “Collision Domain”
 - Distance within which collision can occur and be detected
 - IEEE 802.3 specifies:
worst case collision detection time: $51.2 \mu\text{s}$
- Slot time
 - $51.2 \mu\text{s} = 512 \text{ bits} = 64 \text{ bytes}$
- Why minimum frame size?
 - $51.2 \mu\text{s} \Rightarrow$ minimum # of bits can be transited at 10Mbps is 512 bits $\Rightarrow 64 \text{ bytes}$ is required for collision detection



Random Access

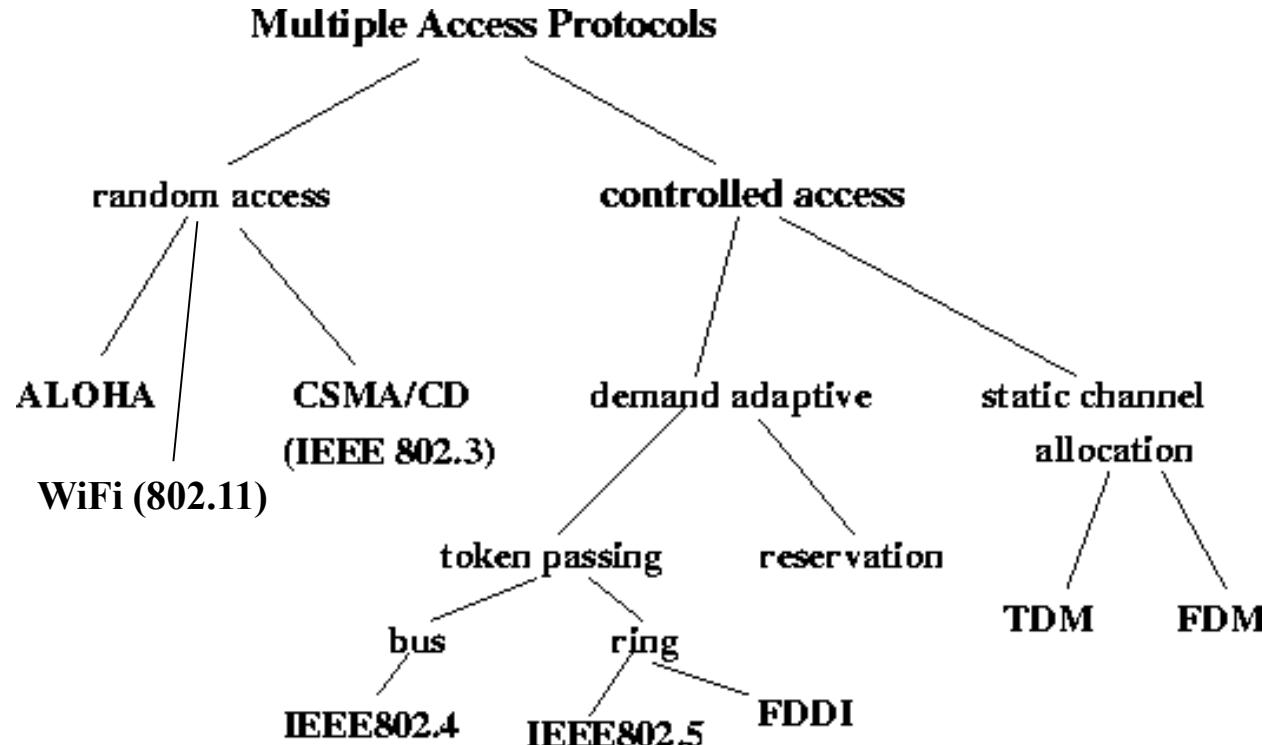
- Stations contend for channels
- Overlapping transmissions (**collisions**) can occur
 - Carrier sensing?
 - Collision detection?
- Protocols
 - Aloha (**not covered**)
 - Slotted Aloha (**not covered**)
 - Carrier Sense Multiple Access: Ethernet



Controlled Access

- Stations reserve or are allocated channel
 - No collisions
 - Allocation: static or dynamic
- Protocols
 - Static channel allocation
 - Time division multiple access
 - Demand adaptive channel allocation
 - Reservation protocols
 - Token passing (token bus, token ring)

Taxonomy of MAC Protocols





Carrier Sense Multiple Access

- CSMA: Listen before transmit
 - If channel idle, transmit entire packet
 - If busy, defer transmission
 - How long should we wait?
 - Human analogy: don't interrupt others
- Can carrier sense avoid collisions completely?



Persistent and Non-persistent CSMA

- Non-persistent
 - If idle, transmit
 - If busy, wait random amount of time
 - If collision, wait random amount of time
- p-persistent
 - If idle, transmit with probability p
 - If busy, wait till it becomes idle
 - If collision, wait random amount of time

CSMA with collision detection (CD)

- Listen while talking
- Stop transmitting when collision detected
 - Compare transmitted and received signals
 - Save time and bandwidth
 - Improvement over persistent and nonpersistent protocols
- Human analogy
 - Polite conversationalist
- Worst case time to detect a collision?



Ethernet MAC Protocol: Basic Ideas

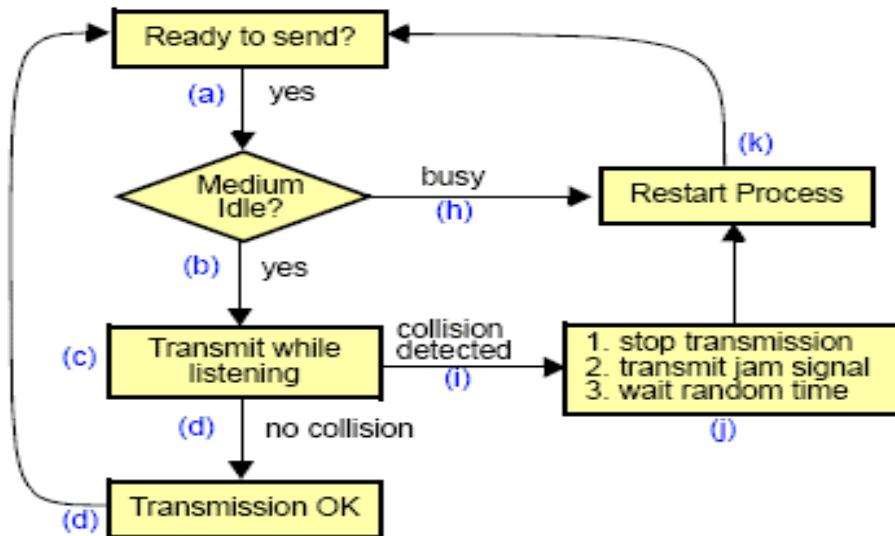
- Ethernet uses CSMA/CD – listens to line before/during sending
- If line is idle (no carrier sensed)
 - send packet immediately
 - upper bound message size of 1500 bytes
 - must wait 9.6us between back-to-back frames
- If line is busy (carrier sensed)
 - wait until idle and transmit packet immediately
 - called *1-persistent* sending
- If collision detected
 - Stop sending and jam signal
 - Try again later



Ethernet CSMA/CD Algorithm

1. Adaptor gets datagram and creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame ! Signal to network layer “transmit OK”
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the m^{th} collision, adapter chooses a K at random from $\{0,1,2,\dots,2^m-1\}$. Adapter waits $K \times 512$ bit times and returns to Step 2
6. Quit after 16 attempts, signal to network layer “transmit error”

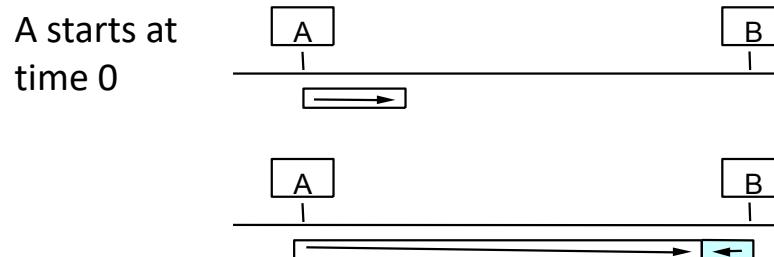
Ethernet CSMA/CD Alg. Flow Chart



Collisions

Collisions are caused when two adaptors transmit at the same time (adaptors sense collision based on voltage differences)

- Both found line to be idle
- Both had been waiting to for a busy line to become idle



Message almost
there at time T when
B starts – collision!

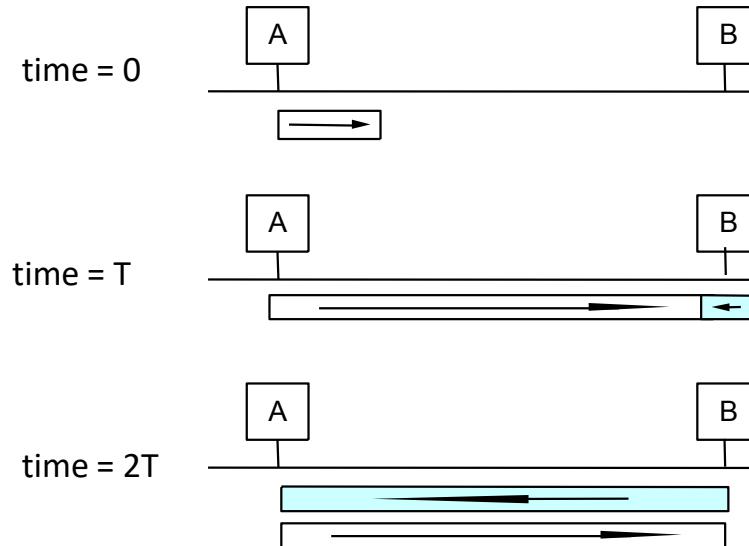
How can we be sure A knows about the collision?



Collision Detection

- How can A know that a collision has taken place?
 - There must be a mechanism to insure retransmission on collision
 - A's message reaches B at time T
 - B's message reaches A at time 2T
 - So, A must still be transmitting at 2T
- IEEE 802.3 specifies max value of 2T to be 51.2us
 - This relates to maximum distance of 2500m between hosts
 - At 10Mbps it takes 0.1us to transmit one bit so 512 bits (64B) take 51.2us to send
 - So, Ethernet frames must be at least 64B long
 - 14B header, 46B data, 4B CRC
 - Padding is used if data is less than 46B
- Send jamming signal after collision is detected to insure all hosts see collision
 - 48 bit signal

Collision Detection contd.





Exponential Backoff

- If a collision is detected, delay and try again
- Delay time is selected using binary exponential backoff
 - 1st time: choose K from {0,1} then delay = $K * 51.2\mu s$
 - 2nd time: choose K from {0,1,2,3} then delay = $K * 51.2\mu s$
 - n th time: delay = $K \times 51.2\mu s$, for $K=0..2^n - 1$
 - Note max value for k = 1023
 - give up after several tries (usually 16)
 - Report transmit error to host
- If delay were not random, then there is a chance that sources would retransmit in lock step
- Why not just choose from small set for K
 - This works fine for a small number of hosts
 - Large number of nodes would result in more collisions



MAC Algorithm from the Receiver Side

- Senders handle all access control
- Receivers simply read frames with acceptable address
 - Address to host
 - Address to broadcast
 - Address to multicast to which host belongs
 - All frames if host is in promiscuous mode



Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 32 bits;

Bit time: .1 microsec for 10 Mbps Ethernet ; for K=1023, wait time is about 50 msec

Exponential Backoff:

- *Goal:* adapt retransmission attempts to estimated current load
 - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K x 512 bit transmission times
- after second collision: choose K from {0,1,2,3}...
- after ten collisions, choose K from {0,1,2,3,4,...,1023}



CSMA/CD Efficiency

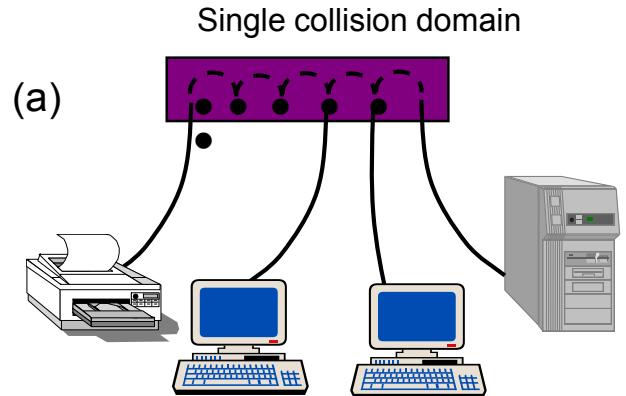
Relevant parameters

- cable length, signal speed, frame size, bandwidth
- t_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

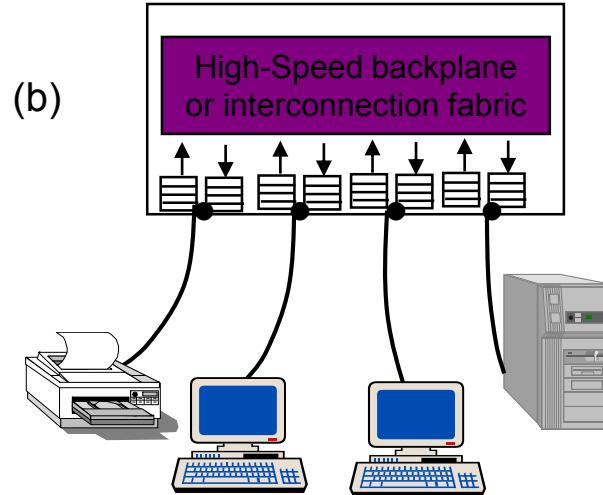
$$\text{efficiency} \approx \frac{1}{1 + 5t_{prop} / t_{trans}}$$

- Efficiency goes to 1 as t_{prop} goes to 0
- Goes to 1 as t_{trans} goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

Ethernet Hubs & Switches



Twisted Pair Cheap
Easy to work with
Reliable
Star-topology CSMA-
CD



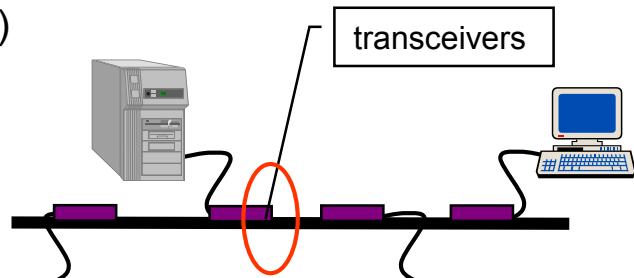
Twisted Pair Cheap
Bridging increases
scalability
Separate collision domains
Full duplex operation

IEEE 802.3 Physical Layer

IEEE 802.3 10 Mbps medium alternatives

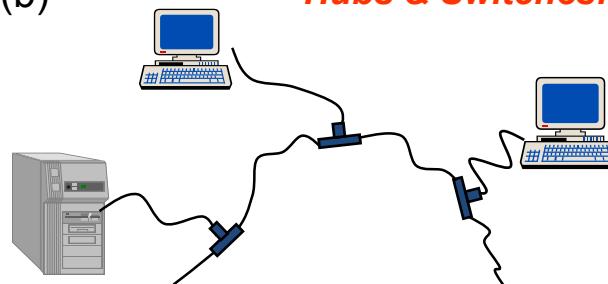
	10base5	10base2	10baseT	10baseFX
Medium	Thick coax	Thin coax	Twisted pair	Optical fiber
Max. Segment Length	500 m	200 m	100 m	2 km
Topology	Bus	Bus	Star	Point-to-point link

(a)



Thick Coax: Stiff, hard to work with

(b)



T connectors flaky



Evolution of Ethernet

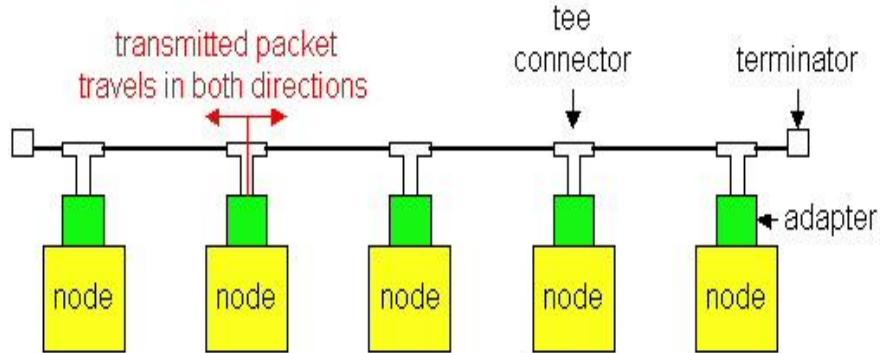
From early 80's 10Base Ethernet to 90's 100Base (Fast) Ethernet
to today's Gigabit Ethernet to 10 Gigabit Ethernet,

IEEE 802.3 Original Parameters

- transmission Rate: 10 Mbps
- Min Frame: 512 bits = 64 bytes
- slot time: 512 bits/10 Mbps = 51.2 μ sec
 - $51.2 \mu\text{sec} \times 2 \times 10^5 \text{ km/sec} = 10.24 \text{ km}$
 - 5.12 km round trip distance
- max Length: 2500 meters + 4 repeaters
- For compatibility, desire to maintain same frame format!
 - *Each x10 increase in bit rate, must be accompanied by x10 decrease in distance ?!*

Ethernet Technologies: 10Base2

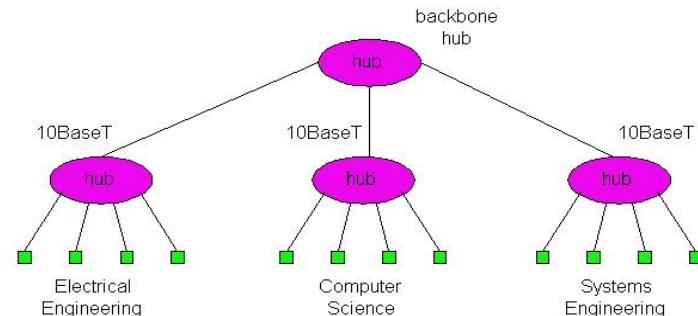
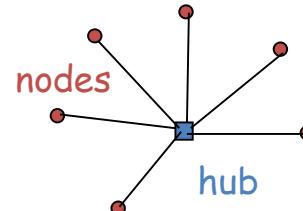
- 10: 10Mbps; 2: under 200 meters max cable length
- thin coaxial cable in a bus topology



- repeaters used to connect up to multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- has become a legacy technology

10BaseT

- 10 Mbps rate
- T stands for Twisted Pair
- Nodes connect to a hub: “star topology”; 100 m max distance between nodes and hub
- Hubs are essentially physical-layer repeaters:
 - bits coming in one link go out all other links
 - no frame buffering
 - no CSMA/CD at hub: adapters detect collisions
 - provides net management functionality





Fast (100Mbps) Ethernet

IEEE 802.3 100 Mbps Ethernet medium alternatives

- Fast Ethernet (100Mbps) has technology very similar to 10Mbps Ethernet
 - Uses different physical layer encoding (4B5B)
 - Many NIC's are 10/100 capable
 - Can be used at either speed

	100baseT4	100baseT	100baseFX
Medium	Twisted pair category 3 UTP 4 pairs	Twisted pair category 5 UTP two pairs	Optical fiber multimode Two strands
Max. Segment Length	100 m	100 m	2 km
Topology	Star	Star	Star

To preserve compatibility with 10 Mbps Ethernet:

- Same frame format, same interfaces, same protocols
- Hub topology only with twisted pair & fiber
- Bus topology & coaxial cable abandoned
- Category 3 twisted pair (ordinary telephone grade) requires 4 pairs
- Category 5 twisted pair requires 2 pairs (most popular)
- Most prevalent LAN today



Gigabit Ethernet

Gigabit Ethernet Physical Layer Specification (IEEE 802.3 1 Gigabit Ethernet medium alternatives)

	1000baseSX	1000baseLX	1000baseCX	1000baseT
Medium	Optical fiber multimode Two strands	Optical fiber single mode Two strands	Shielded copper cable	Twisted pair category 5 UTP
Max. Segment Length	550 m	5 km	25 m	100 m
Topology	Star	Star	Star	Star



Gigabit Ethernet

- use standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- Compatible with lower speeds
- Uses standard framing and CSMA/CD algorithm
- Distances are severely limited
- Typically used for backbones and inter-router connectivity
- Becoming cost competitive
- Commonly used today: Gigabit switches!
 - Full-Duplex at 1 Gbps for point-to-point links
 - Frame structure preserved but CSMA-CD essentially abandoned
 - Extensive deployment in backbone of enterprise data networks and in server farms



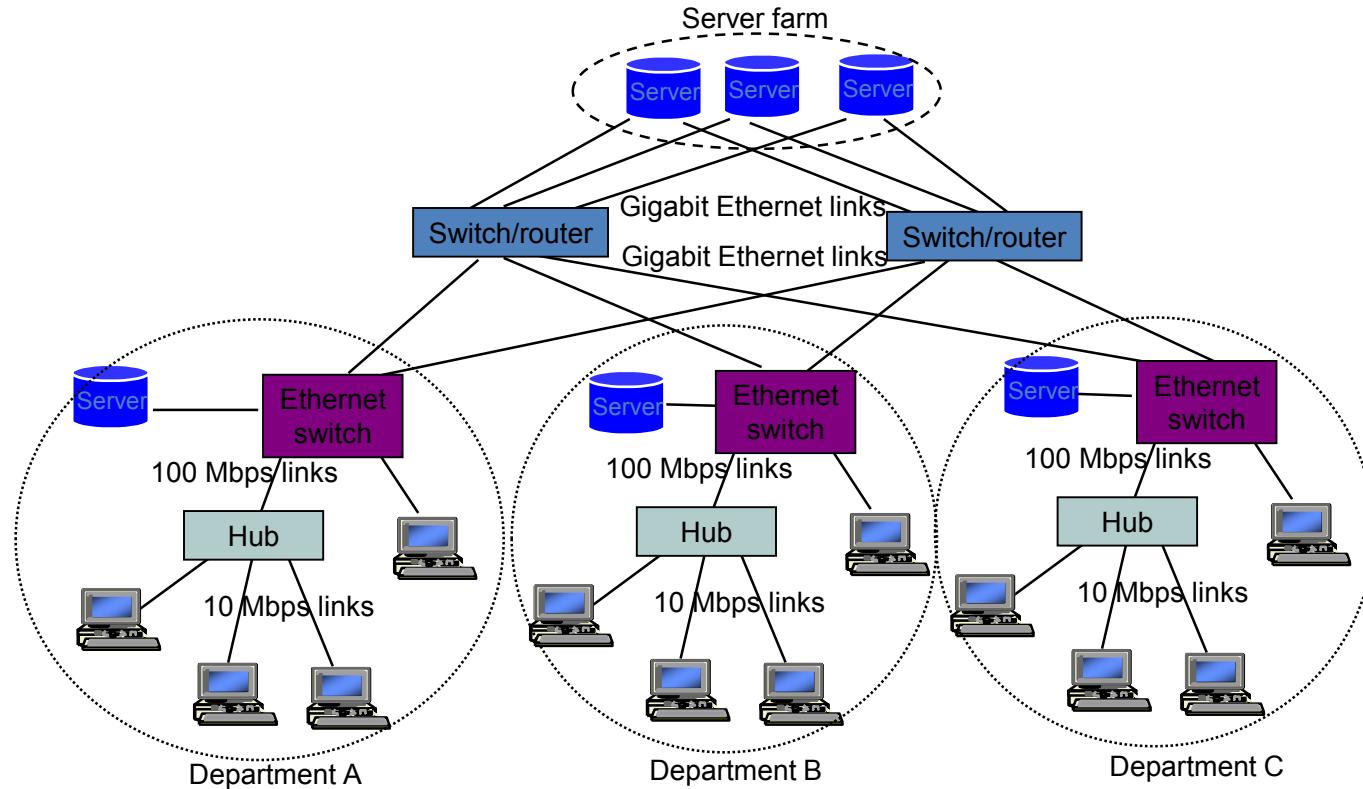
10 Gigabit Ethernet

IEEE 802.3 10 Gbps Ethernet medium alternatives

	10GbaseSR	10GBaseLR	10GbaseEW	10GbaseLX4
Medium	Two optical fibers Multimode at 850 nm 64B66B code	Two optical fibers Single-mode at 1310 nm 64B66B	Two optical fibers Single-mode at 1550 nm SONET compatibility	Two optical fibers multimode/single-mode with four wavelengths at 1310 nm band 8B10B code
Max. Segment Length	300 m	10 km	40 km	300 m - 10 km

- Frame structure preserved
- CSMA-CD protocol officially abandoned
- LAN PHY for local network applications
- WAN PHY for wide area interconnection using SONET OC-192c
- Extensive deployment in metro networks anticipated

Typical Ethernet Deployment





Experiences with Ethernet

- Ethernets work best under light loads
 - Utilization over 30% is considered heavy
 - Network capacity is wasted by collisions
- Most networks are limited to about 200 hosts
 - Specification allows for up to 1024
- Most networks are much shorter
 - 5 to 10 microsecond RTT
- Transport level flow control helps reduce load (number of back to back packets)
- Ethernet is inexpensive, fast and easy to administer!



Ethernet Problems

- Ethernet's peak utilization is pretty low
- Peak throughput worst with
 - More hosts
 - More collisions needed to identify single sender
 - Smaller packet sizes
 - More frequent arbitration
 - Longer links
 - Collisions take longer to observe, more wasted bandwidth
 - Efficiency is improved by avoiding these conditions

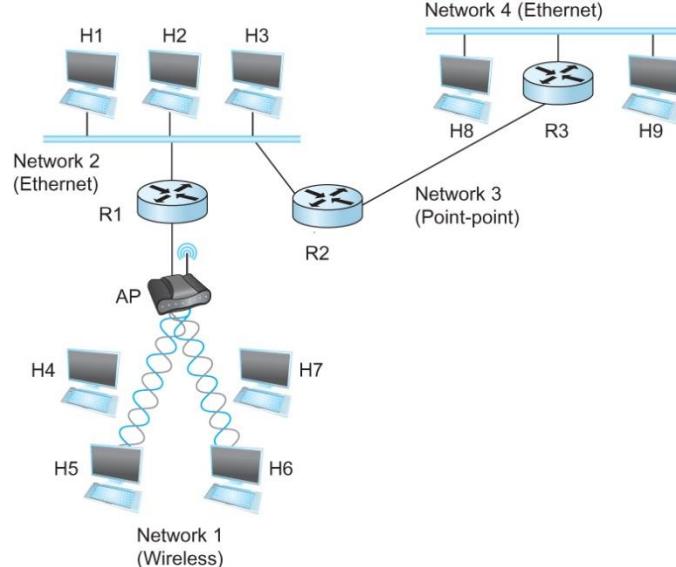


IF2230 Jaringan Komputer Internetworking **Bridging and Switching**

Robithoh Annur
Andreas Bara Timur
Monterico Andrian

Introduction

- What is internetwork
 - An arbitrary collection of networks interconnected to provide some sort of host-host to packet delivery service



A simple internetwork where H represents hosts and R represents routers



Introduction

- It is necessary to connect a LAN to another LAN or to a WAN.
 - Computers within a LAN are often connected using a hub
 - LAN to LAN connections are often performed with a bridge.
 - Segments of a LAN are usually connected using a switch.
 - LAN to WAN connections are usually performed with a router (next lecture).



Topics

- Interconnecting LAN segments
 - HUB (Physical Layer)
 - Bridge (Link layer)
 - Layer 2 Switch (multi-port bridge, link layer)
- Interconnecting networks
 - Layer 3 Switch (network layer)
 - Router (network layer)



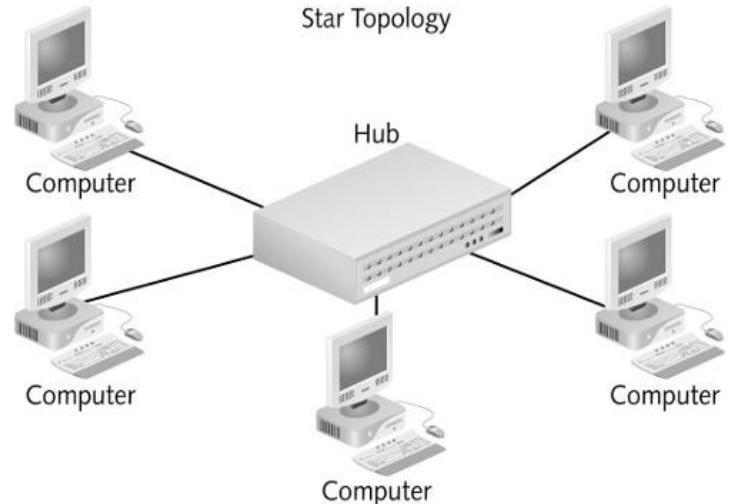
Interconnecting LAN Segments

Three Major Devices

- Hubs (layer 1 devices)
- Bridges (layer 2 devices)
 - Basic Functions
 - Self learning and bridge forwarding table
 - Forwarding/filtering algorithm
 - Bridge looping problem and spanning tree algorithm
- Ethernet Switches
 - Remark: switches are essentially multi-port bridges.
 - What we say about bridges also holds for switches!

Interconnecting with Hubs

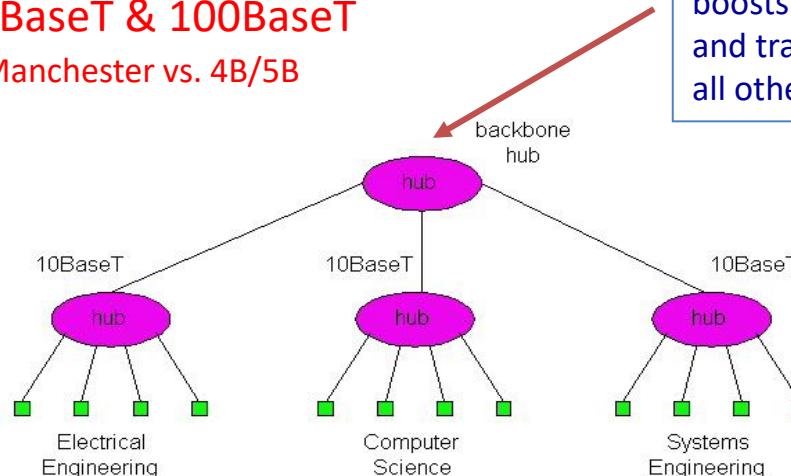
- A hub interconnects two or more nodes into a local area network.
- A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations.
- Hubs cannot filter data, so data packets are sent to all connected devices.
- Hubs do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.



Interconnecting with Hubs

- Backbone hub interconnects LAN segments
- Hubs expand one Ethernet connection into many and extends max distance between nodes
- But individual segment collision domains become one large collision domain
 - if a node in CS and a node EE transmit at same time: collision
- **Can't interconnect 10BaseT & 100BaseT**
 - Encoding is different: Manchester vs. 4B/5B

Recreates each bit,
boosts its energy strength,
and transmits the bit to
all other interfaces



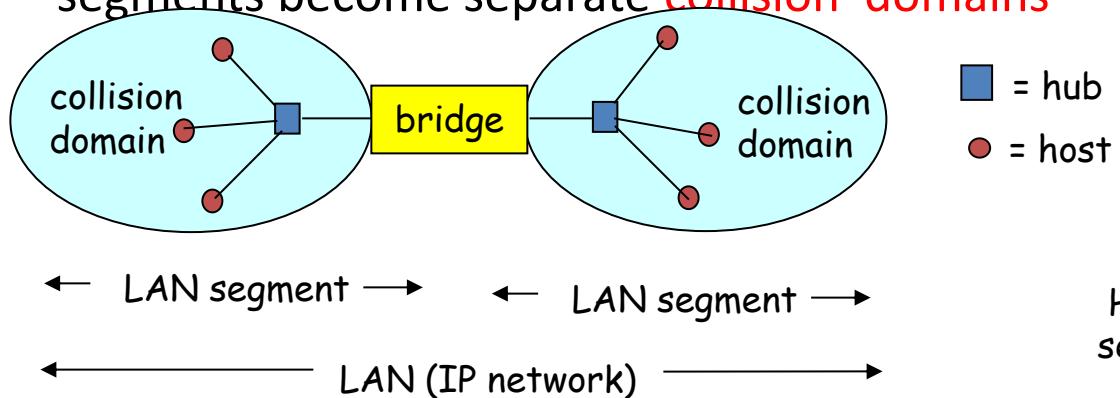


Bridges

- Link layer device
 - stores and forwards Ethernet frames
 - examines frame header and selectively forwards frame based on MAC destination address -- filtering
 - when frame is to be forwarded on a LAN segment, uses CSMA/CD to access the LAN segment
- transparent
 - hosts are unaware of the presence of bridges, it appears to them as a single whole network
- plug-and-play, self-learning
 - bridges do not need to be configured

Bridges: Traffic Isolation

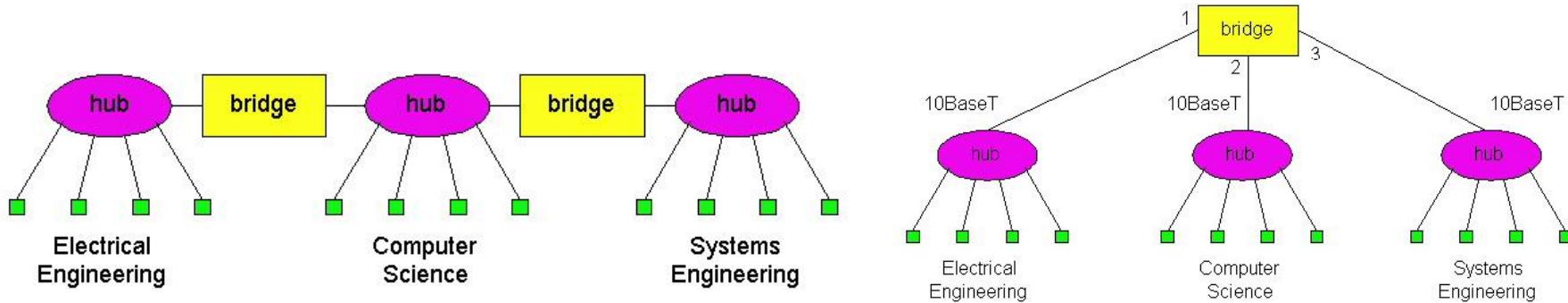
- Bridge installation breaks LAN into LAN segments
- Bridges **filter** packets:
 - same-LAN-segment frames not usually forwarded onto other LAN segments
 - segments become separate **collision domains**



How to determine to which LAN segment to forward frame?

Interconnection without Backbone?

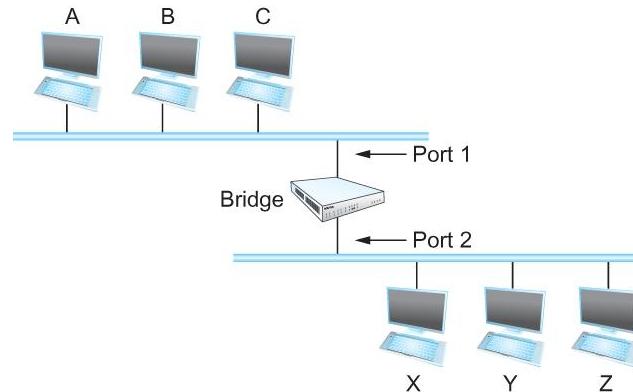
- Not recommended for two reasons:
 - single point of failure at Computer Science hub
 - all traffic between EE and SE must path over CS segment



Recommended !

Bridges

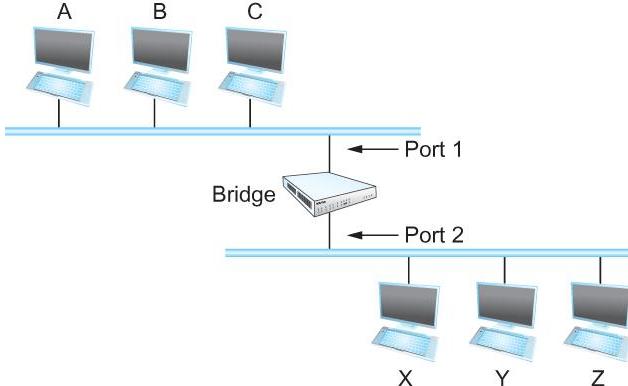
- Consider the following figure
 - When a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.



- How does a bridge come to learn on which port the various hosts reside?

Bridges

- Solution
 - Download a table into the bridge



Host	Port

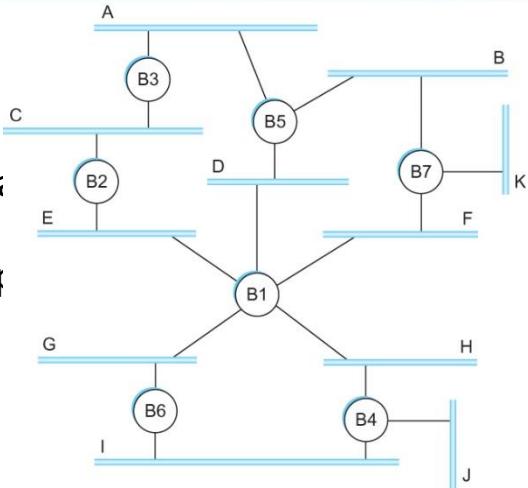
A	1
B	1
C	1
X	2
Y	2
Z	2

- Who does the download?
 - Human
 - Too much work for maintenance

- A bridge has a **bridge (forwarding) table**
- How to perform self learning?
 - Each bridge inspects the source address in all the frames it receives
 - Record the information at the bridge and build the table
 - When a bridge first boots, this table is empty
 - Entries are added over time
 - A timeout is associated with each entry
 - The bridge discards the entry after a specified period of time
 - To protect against the situation in which a host is moved from one network to another
- If the bridge receives a frame that is addressed to host not currently in the table
 - Forward the frame out on all other ports

Bridges

- Strategy works fine if the extended LAN does not have a loop in it
- Looping
 - Pros= for increased reliability, desirable to have redundant, i.e. multiple paths from source to destination
 - Cons= with multiple paths, **cycles** result - bridges may multiplex forever
- How does an extended LAN come to have a loop in it?
 - Network is managed by more than one administrator
 - For example, it spans multiple departments in an organization
 - It is possible that no single person knows the entire configuration of the network
 - A bridge that closes a loop might be added without anyone knowing
 - Loops are built into the network to provide redundancy in case of failures (by designed)
- Solution
 - Distributed Spanning Tree Algorithm → disabling subset of interfaces



Bridges B1, B4, and B6 form a loop (by designed)

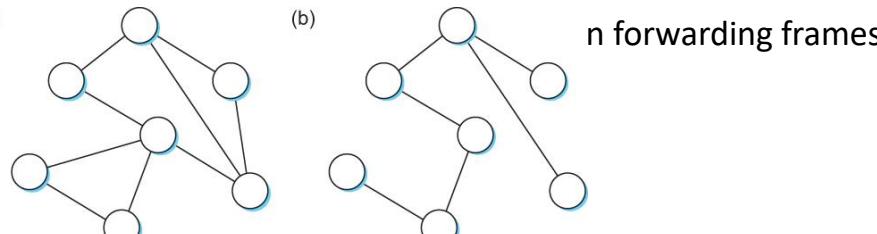


Bridges

- How does an extended LAN come to have a loop in it?
 - Network is managed by more than one administrator
 - For example, it spans multiple departments in an organization
 - It is possible that no single person knows the entire configuration of the network
 - A bridge that closes a loop might be added without anyone knowing
 - Loops are built into the network to provide redundancy in case of failures
- Solution
 - Distributed Spanning Tree Algorithm

Spanning Tree Algorithm

- Think of the extended LAN as being represented by a graph that possibly has loops (cycles)
- A spanning tree is a sub-graph of this graph that covers all the vertices but contains no cycles
 - Spanning tree keeps all the vertices of the original graph but throws out some of the edges
- Developed by Radia Perlman at Digital
 - A protocol used by a set of bridges to agree upon a spanning tree for a particular extended LAN
 - IEEE 802.1 specification for LAN bridges is based on this algorithm
 - Each bridge decides the ports over which it is and is not willing to forward frames
 - In a sense, it is by removing ports from the topology that the extended LAN is reduced to an acyclic tree
 - It is even possible^(a)



Example of (a) a cyclic graph; (b) a corresponding spanning tree.



Spanning Tree Algorithm

- Algorithm is dynamic
 - The bridges are always prepared to reconfigure themselves into a new spanning tree if some bridges fail
- Main idea
 - Each bridge selects the ports over which they will forward the frames
- Algorithm selects ports as follows:
 - Each bridge has a unique identifier
 - B1, B2, B3,...and so on.
 - Elect the bridge with the smallest id as the root of the spanning tree
 - The root bridge always forwards frames out over all of its ports
 - Each bridge computes the shortest path to the root and notes which of its ports is on this path
 - This port is selected as the bridge's preferred path to the root
 - Finally, all the bridges connected to a given LAN elect a single *designated bridge* that will be responsible for forwarding frames toward the root bridge

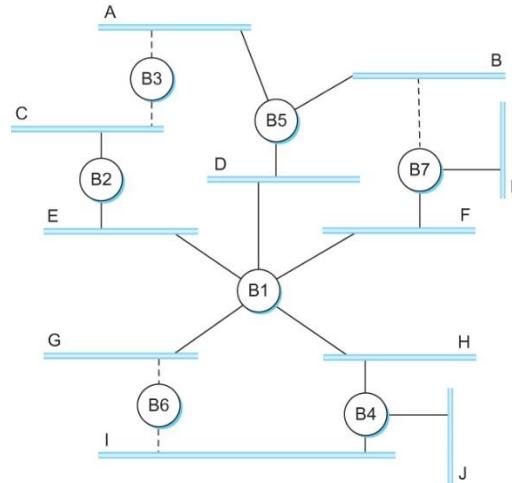
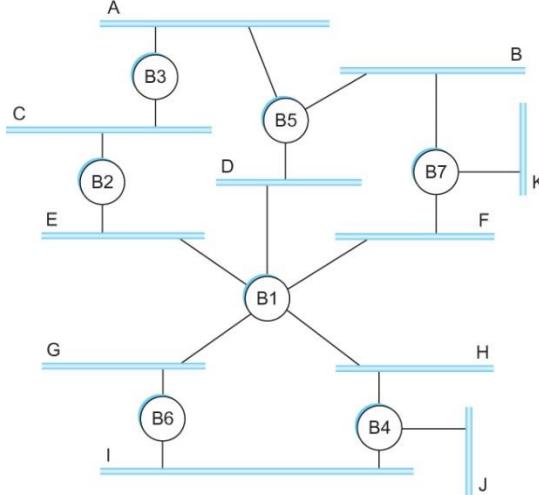


Spanning Tree Algorithm

- Each LAN's designated bridge is the one that is closest to the root
- If two or more bridges are equally close to the root,
 - Then select bridge with the smallest id
- Each bridge is connected to more than one LAN
 - So it participates in the election of a designated bridge for each LAN it is connected to.
 - Each bridge decides if it is the designated bridge relative to each of its ports
 - The bridge forwards frames over those ports for which it is the designated bridge

Spanning Tree Algorithm

- B1 is the root bridge
- B3 and B5 are connected to LAN A, but B5 is the designated bridge
- B5 and B7 are connected to LAN B, but B5 is the designated bridge





Spanning Tree Algorithm

- Initially each bridge thinks it is the root, so it sends a configuration message on each of its ports identifying itself as the root and giving a distance to the root of 0
- Upon receiving a configuration message over a particular port, the bridge checks to see if the new message is *better* than the current best configuration message recorded for that port
- The new configuration is better than the currently recorded information if
 - It identifies a root with a smaller id or
 - It identifies a root with an equal id but with a shorter distance or
 - The root id and distance are equal, but the sending bridge has a smaller id

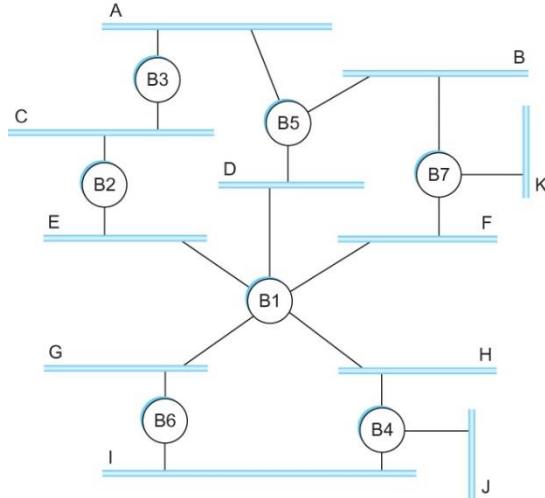


Spanning Tree Algorithm

- If the new message is better than the currently recorded one,
 - The bridge discards the old information and saves the new information
 - It first adds 1 to the distance-to-root field
- When a bridge receives a configuration message indicating that it is not the root bridge (that is, a message from a bridge with smaller id)
 - The bridge stops generating configuration messages on its own
 - Only forwards configuration messages from other bridges after 1 adding to the distance field
- When a bridge receives a configuration message that indicates it is not the designated bridge for that port
=> a message from a bridge that is closer to the root or equally far from the root but with a smaller id
 - The bridge stops sending configuration messages over that port
- When the system stabilizes,
 - Only the root bridge is still generating configuration messages.
 - Other bridges are forwarding these messages only over ports for which they are the designated bridge

Spanning Tree Algorithm

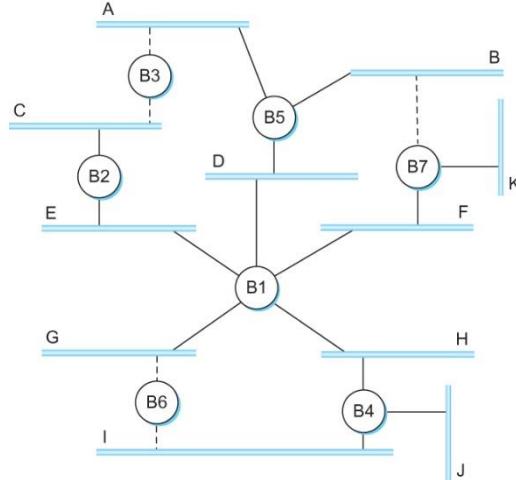
- Consider the situation when the power had just been restored to the building housing the following network



- All bridges would start off by claiming to be the root

Spanning Tree Algorithm

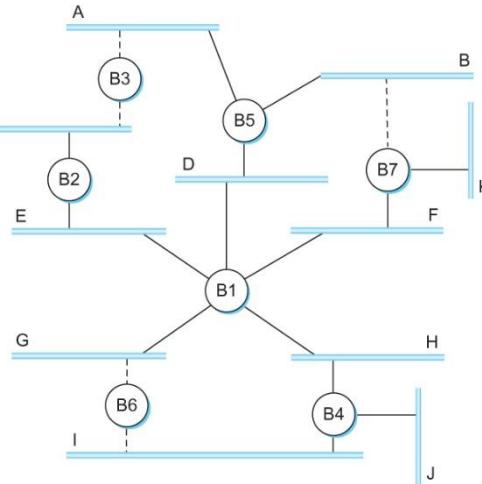
- Denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



- Consider the activity at node B3

Spanning Tree Algorithm

- B3 receives (B2, 0, B2)
- Since $2 < 3$, B3 accepts B2 as root
- B3 adds 1 to the distance advertised by B2 and sends (B2, 1, B3) to B5
- Meanwhile B2 accepts B1 as root because it has the lower id and $i_1 < i_2$
- B5 accepts B1 as root and sends (B1, 1, B5) to B3
- B3 accepts B1 as root and it notes that both B2 and B5 are closer to it
 - Thus B3 stops forwarding messages on both its interfaces
 - This leaves B3 with both ports not selected





Spanning Tree Algorithm

- Even after the system has stabilized, the root bridge continues to send configuration messages periodically
 - Other bridges continue to forward these messages
- When a bridge fails, the downstream bridges will not receive the configuration messages
- After waiting a specified period of time, they will once again claim to be the root and the algorithm starts again
- Note
 - Although the algorithm is able to reconfigure the spanning tree whenever a bridge fails, it is not able to forward frames over alternative paths for the sake of routing around a congested bridge



Spanning Tree Algorithm

- Broadcast and Multicast
 - Forward all broadcast/multicast frames
 - Current practice
 - Learn when no group members downstream
 - Accomplished by having each member of group G send a frame to bridge multicast address with G in source field



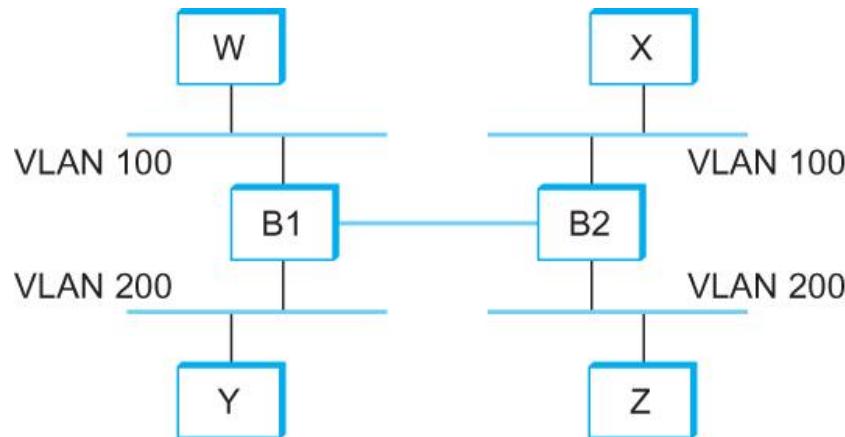
Spanning Tree Algorithm

- Limitation of Bridges
- Do not scale → it is not realistic to connect more than a few LANs by means of bridges
 - Spanning tree algorithm does not scale
 - Broadcast does not scale
- Do not accommodate heterogeneity



Virtual LAN

- One approach to increasing the scalability of extended LANs is the *virtual LAN* (VLAN).
- VLANs allow a single extended LAN to be partitioned into several seemingly separate LANs



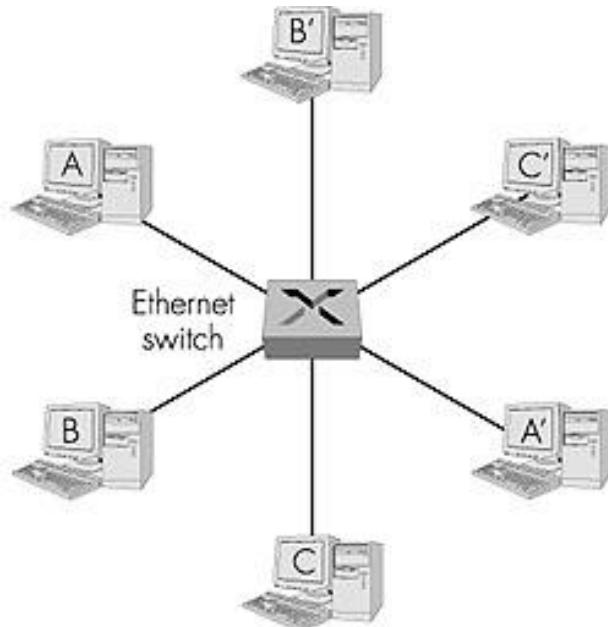


More Words about VLAN

- **Virtual LAN (VLAN) – defined in IEEE 802.1q**
 - Partition a physical LAN into several “logically separate” LANs
 - reduce broadcast traffic on physical LAN!
 - provide administrative isolation
 - Extend over a WAN (wide area network), e.g., via layer 2 tunnels (e.g., L2TP, MPLS) over IP-based WANs!
- Two types: port-based or MAC address-based
 - each port optionally configured with a VLAN id
 - inbound packets tagged with this “VLAN” id
 - require change of data frames, carry “VLAN id” tags
 - tagged and untagged frames can co-exist
 - “VLAN-aware” switches forward on ports part of same VLAN
- More complex ! - require administrative configuration
 - static (“manual”) configuration
 - **Find more in the practical**

Ethernet Switches

- Switches A switch is a combination of a hub and a bridge.
- It can interconnect two or more workstations, but like a bridge, it observes traffic flow and learns.
- When a frame arrives at a switch, the switch examines the source address and forwards the frame out the one necessary connection.
- Essentially a multi-interface bridge
- layer 2 (frame) forwarding, filtering using LAN addresses
- **Switching:** A-to-A' and B-to-B' simultaneously, no collisions
- large number of interfaces
- often: individual hosts, star-connected into switch
 - Ethernet, but no collisions!





- Major role: isolating traffic patterns and providing multiple access.
- This design is usually done by the network manager. Switches are easy to install and have components that are hotswappable.
- The backplane of a switch is fast enough to support multiple data transfers at one time.
- Multiple workstations connected to a switch use dedicated segments.
- This is a very efficient way to isolate heavy users from the network.



Ethernet Switches

- **cut-through switching:** frame forwarded from input to output port without awaiting for assembly of entire frame
 - slight reduction in latency
 - Cut-through vs. store and forward
- combinations of shared/dedicated, 10/100/1000 Mbps interfaces



Switching and Forwarding Network Layer

- **Switching and Forwarding**
 - Generic Switch Architecture
 - Forwarding Tables:
 - Bridges/Layer 2 Switches; VLAN
 - Routers and Layer 3 Switches
- **Forwarding in Layer 3 (Network Layer)**
 - Network Layer Functions
 - Network Service Models: VC vs. Datagram
 - ATM and IP Datagram Forwarding



Hubs vs. Bridges vs. Routers

- **Hubs (aka Repeaters): Layer 1 devices**
 - repeat (i.e., regenerate) physical signals
 - don't understand MAC protocols!
 - LANs connected by hubs belong to same collision domain
- **Bridges (and Layer-2 Switches): Layer 2 devices**
 - store and forward layer-2 frames based on MAC addresses
 - speak and obey MAC protocols
 - bridges segregate LANs into different collision domains
- **Routers (and Layer 3 Switches): Layer 3 devices**
 - store and forward layer-3 packets based on network layer addresses (e.g., IP addresses)
 - rely on data link layer to deliver packets to (directly connected) next hop
 - network layer addresses are logical (i.e. virtual), need to map to MAC addresses for packet delivery



Forwarding in Layer 3

Putting in context

- What does layer-3 (network layer) do?
 - deliver packets “hop-by-hop” across a network
 - rely on layer-2 to deliver between neighboring hops
- Key Network Layer Functions
 - Addressing: need a global (logical) addressing scheme
 - Routing: build “map” of network, find routes, ...
 - Forwarding: actual delivery of packets!
- Two basic network layer service models
 - datagram: “connectionless”
 - virtual circuit (VC): connection-oriented



Network Layer Functions

- Addressing
 - Globally unique address for each routable device
 - Logical address, unlike MAC address (as you've seen earlier)
 - Assigned by network operator
 - Need to map to MAC address (as you'll see later)
- Routing: building a “map” of network
 - Which path to use to forward packets from src to dest
- Forwarding: delivery of packets hop by hop
 - From input port to appropriate output port in a router

Routing and forwarding depend on network service models: *datagram* vs. *virtual circuit*



Virtual Circuit vs. Datagram

- Objective of both: move packets through routers from source to destination
- **Datagram Model:**
 - *Routing*: determine next hop to each destination a priori
 - *Forwarding*: **destination address in packet header**, used at each hop to look up for next hop
 - routes may change during “session”
 - analogy: driving, asking directions at every corner gas station, or based on the road signs at every turn
- **Virtual Circuit Model:**
 - *Routing*: determine a path from source to each destination
 - “*Call*” Set-up: fixed path (“virtual circuit”) set up at “*call setup time*”, remains fixed thru “*call*”
 - *Data Forwarding*: each packet carries “tag” or “label” (**virtual circuit id, VCI**), which determines next hop
 - **“routers maintain “per-call” state”**

Datagram

- No connection setup phase
- Each packet forwarded independently
- Sometimes called *connectionless* model
- The idea behind datagrams is incredibly simple
 - just include in every packet enough information i.e. a complete destination address
- Each switch maintains a forwarding (routing) table, to decide how to forward the packet to its destination.

Analogy: postal system

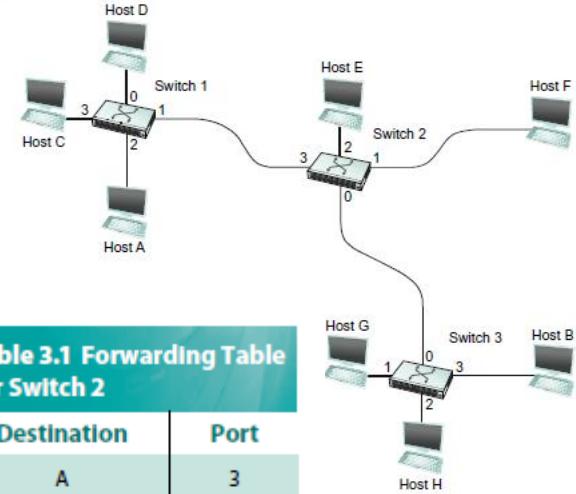


Table 3.1 Forwarding Table for Switch 2

Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0



Datagram

Characteristics of Connectionless (Datagram) Network

- A host can send a packet anywhere at any time, since any packet that turns up at the switch can be immediately forwarded (assuming a correctly populated forwarding table)
- When a host sends a packet, it has no way of knowing if the network is capable of delivering it or if the destination host is even up and running
- Each packet is forwarded independently of previous packets that might have been sent to the same destination.
 - Thus two successive packets from host A to host B may follow completely different paths
- A switch or link failure might not have any serious effect on communication if it is possible to find an alternate route around the failure and update the forwarding table accordingly

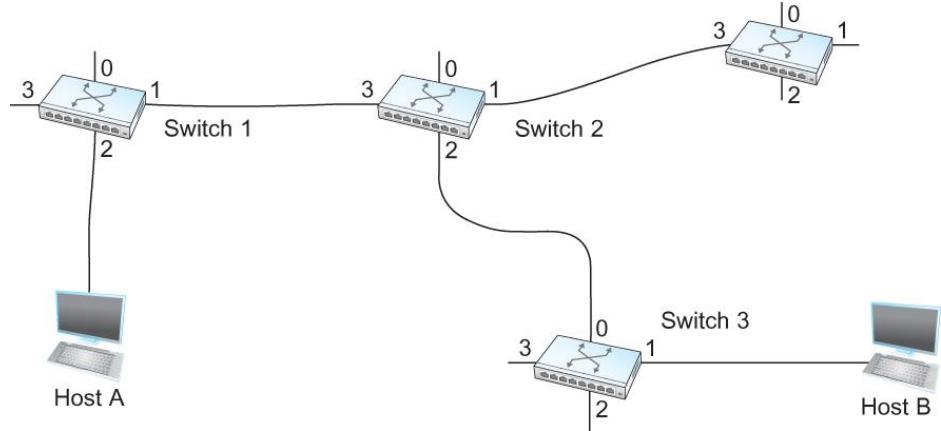


Datagram (Summary)

- There is no round trip delay waiting for connection setup; a host can send data as soon as it is ready.
- Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
- Since packets are treated **independently**, it is possible to route around link and node failures.
- Since every packet must carry the **full address** of the destination, the overhead per packet is higher than for the connection-oriented model.

Virtual Circuit Switching

- Widely used technique for packet switching
 - Uses the concept of *virtual circuit* (VC) → Subsequence packets follow same circuit
 - Explicit connection setup (and tear-down) phase
 - Sometimes called *connection-oriented* model
- still packet switching, not circuit switching!
- Each switch maintains a VC table



Host A wants to send packets to host B. How?



Virtual Circuit Switching

Two-stage process

- Connection setup
- Data Transfer
- Connection setup
 - Establish “connection state” in each of the switches between the source and destination hosts
 - The connection state for a single connection consists of an entry in the “VC table” in each switch through which the connection passes



Virtual Circuit Switching

Two broad classes of approach to establishing connection state

- Network Administrator will configure the state
 - The virtual circuit is **permanent** (PVC)
 - The network administrator can delete this
 - Can be thought of as a long-lived or administratively configured VC
- A host can send messages into the network to cause the state to be established
 - This is referred as **signalling** and the resulting virtual circuit is said to be **switched** (SVC)
 - A host may set up and delete such a VC dynamically without the involvement of a network administrator



Virtual Circuit Switching

One entry in the VC table on a single switch contains

- A virtual circuit identifier (VCI) that uniquely identifies the connection at this switch and that will be carried inside the header of the packets that belong to this connection
 - An incoming interface on which packets for this VC arrive at the switch
 - An outgoing interface in which packets for this VC leave the switch
 - A potentially different VCI that will be used for outgoing packets
-
- The semantics for one such entry is
 - If a packet arrives on the designated incoming interface and that packet contains the designated VCI value in its header, then the packet should be sent out the specified outgoing interface with the specified outgoing VCI value first having been placed in its header

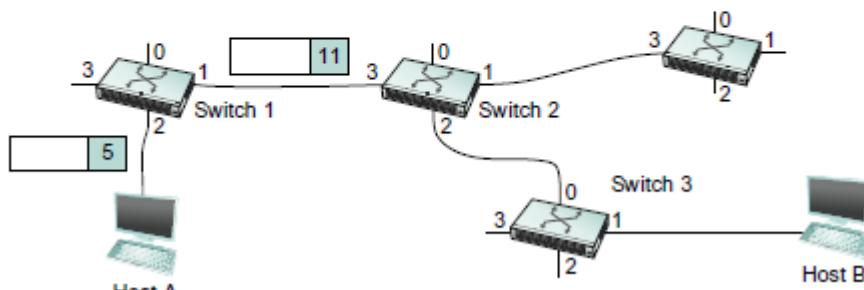
Virtual Circuit Switching

Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B

- First the administrator identifies a path through the network from A to B
- The administrator then picks a VCI value that is currently unused on each link for the connection
- For our example,
 - Suppose the VCI value 5 is chosen for the link from host A to switch 1
 - 11 is chosen for the link from switch 1 to switch 2
 - So the switch 1 will have an entry in the VC table

Table 3.2 Virtual Circuit Table Entry for Switch 1

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11



Virtual Circuit Switching

Similarly, suppose

- VCI of 7 is chosen to identify this connection on the link from switch 2 to switch 3
- VCI of 4 is chosen for the link from switch 3 to host B
- Switches 2 and 3 are configured with the following VC table

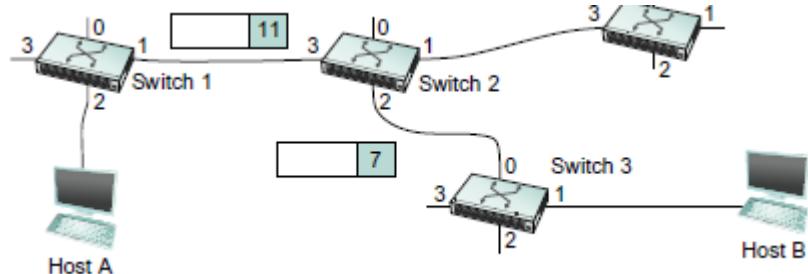
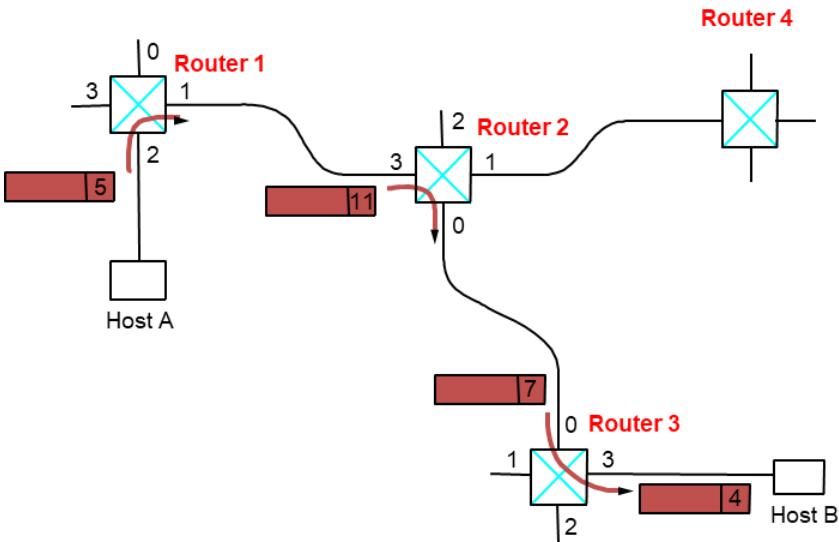


Table 3.3 Virtual Circuit Table Entries for Switches 2 and 3

VC Table Entry at Switch 2			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	2	7
VC Table Entry at Switch 3			
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	1	4

Virtual Circuit Switching

- For any packet that A wants to send to B, A puts the VCI value 5 in the header of the packet and sends it to switch 1
- Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry.
- The table entry on switch 1 tells the switch to forward the packet out of interface 1 and to put the VCI value 11 in the header
- Packet will arrive at switch 2 on interface 3 bearing VCI 11
- Switch 2 looks up interface 3 and VCI 11 in its VC table and sends the packet on to switch 3 after updating the VCI value appropriately
- This process continues until it arrives at host B with the VCI value of 4 in the packet
- To host B, this identifies the packet as having come from host A



"call" from host A to host B along path:
host A → router 1 → router 2 → router 3 → host B



Virtual Circuit Switching

- In real networks of reasonable size, the burden of configuring VC tables correctly in a large number of switches would quickly become excessive
 - Thus, some sort of signalling is almost always used, even when setting up “permanent” VCs
 - In case of PVCs, signalling is initiated by the network administrator
 - SVCs are usually set up using signalling by one of the hosts
- How does the signalling work
 - To start the signalling process, host A sends a setup message into the network (i.e. to switch 1)
 - The setup message contains (among other things) the complete destination address of B.
 - The setup message needs to get all the way to B to create the necessary connection state in every switch along the way
 - It is like sending a datagram to B where every switch knows which output to send the setup message so that it eventually reaches B
 - Assume that every switch knows the topology to figure out how to do that
 - When switch 1 receives the connection request, in addition to sending it on to switch 2, it creates a new entry in its VC table for this new connection
 - The entry is exactly the same shown in the previous table
 - Switch 1 picks the value 5 for this connection



Virtual Circuit Switching

- Now to complete the connection, everyone needs to be told what their downstream neighbor is using as the VCI for this connection
 - Host B sends an acknowledgement of the connection setup to switch 3 and includes in that message the VCI value that it chose (4)
 - Switch 3 completes the VC table entry for this connection and sends the acknowledgement on to switch 2 specifying the VCI of 7
 - Switch 2 completes the VC table entry for this connection and sends acknowledgement on to switch 1 specifying the VCI of 11
 - Finally switch 1 passes the acknowledgement on to host A telling it to use the VCI value of 5 for this connection
- When host A no longer wants to send data to host B, it tears down the connection by sending a teardown message to switch 1
- The switch 1 removes the relevant entry from its table and forwards the message on to the other switches in the path which similarly delete the appropriate table entries
- At this point, if host A were to send a packet with a VCI of 5 to switch 1, it would be dropped as if the connection had never existed



Virtual Circuit Switching

- Characteristics of VC
 - Since host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet, there is at least one RTT of delay before data is sent
 - While the connection request contains the full address for host B (which might be quite large, being a global identifier on the network), each data packet contains only a small identifier, which is only unique on one link.
 - Thus the per-packet overhead caused by the header is reduced relative to the datagram model
 - If a switch or a link in a connection fails, the connection is broken and a new one will need to be established.
 - Also the old one needs to be torn down to free up table storage space in the switches
 - The issue of how a switch decides which link to forward the connection request on has similarities with the function of a routing algorithm
- Good Properties of VC
 - By the time the host gets the go-ahead to send data, it knows quite a lot about the network-
 - For example, that there is really a route to the receiver and that the receiver is willing to receive data
 - It is also possible to allocate resources to the virtual circuit at the time it is established



Virtual Circuit Switching

- In VC, we could imagine providing each circuit with a different quality of service (QoS)
 - The network gives the user some kind of performance related guarantee
 - Switches set aside the resources they need to meet this guarantee
 - For example, a percentage of each outgoing link's bandwidth
 - Delay tolerance on each switch
- Most popular examples of VC technologies are X.25, Frame Relay and ATM
 - One of the applications of Frame Relay is the construction of VPN
- X.25 is an old standard protocol for connection-oriented packet-switched network (used in old telecommunication companies and Automated Teller Machines (ATM's). This employs the following three-part strategy:
 - Buffers are allocated to each virtual circuit when the circuit is initialized
 - The sliding window protocol is run between each pair of nodes along the virtual circuit, and this protocol is augmented with the flow control to keep the sending node from overrunning the buffers allocated at the receiving node
 - The circuit is rejected by a given node if not enough buffers are available at that node when the connection request message is processed

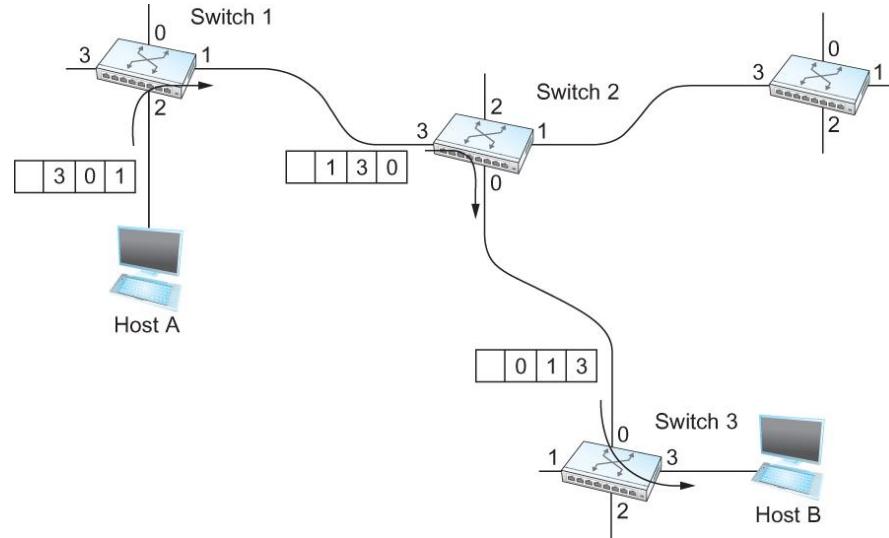
Virtual Circuit Switching

- ATM (Asynchronous Transfer Mode)
 - Connection-oriented packet-switched network
 - Packets are called cells
 - 5 byte header + 48 byte payload
 - Fixed length packets are easier to switch in hardware
 - Simpler to design
 - Enables parallelism
-
- | | | | | | | |
|-----|-----|-----|------|-----|-------------|----------------|
| 4 | 8 | 16 | 3 | 1 | 8 | 384 (48 bytes) |
| GFC | VPI | VCI | Type | CLP | HEC (CRC-8) | Payload |
- Host-to-switch format
 - GFC: Generic Flow Control
 - VCI: Virtual Circuit Identifier
 - Type: management, congestion control
 - CLP: Cell Loss Priority
 - HEC: Header Error Check (CRC-8)

Switching and Forwarding

Source Routing

- Source Routing
 - Is the third approach to switching that uses neither virtual circuits nor conventional datagrams
 - All the information about network topology that is required to switch a packet across the network is provided by the source host



Switching and Forwarding

- Other approaches in Source Routing

Header entering switch



Header leaving switch



(a)



(b)



(c)

(a) rotation; (b) stripping; (c) pointer.

The labels are read right to left.

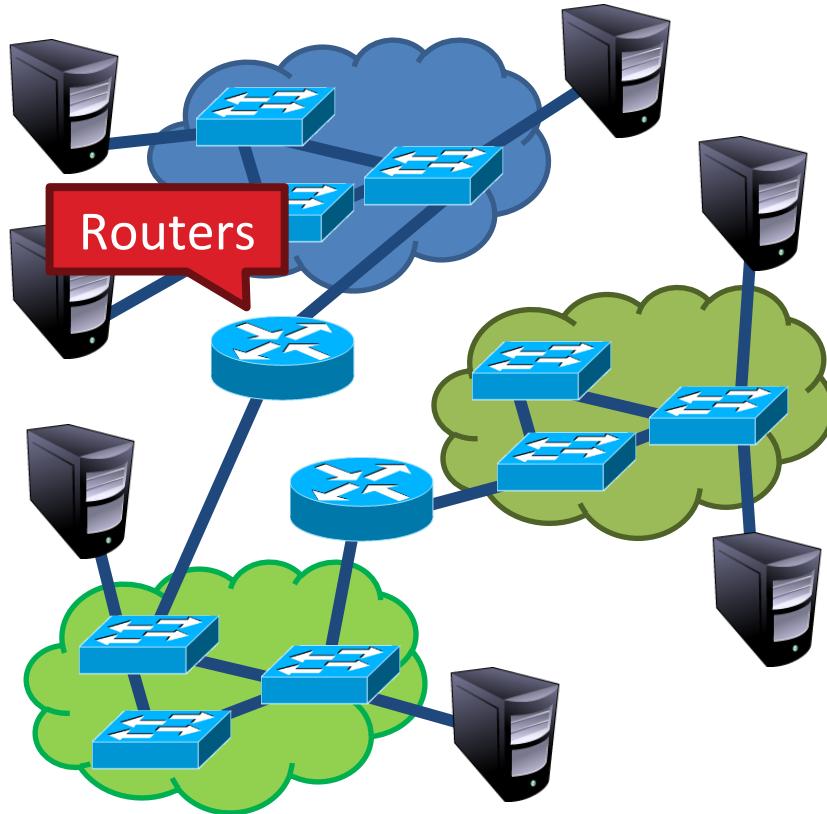


IF2230 Jaringan Komputer Network (IP) Layer **Internetworking**

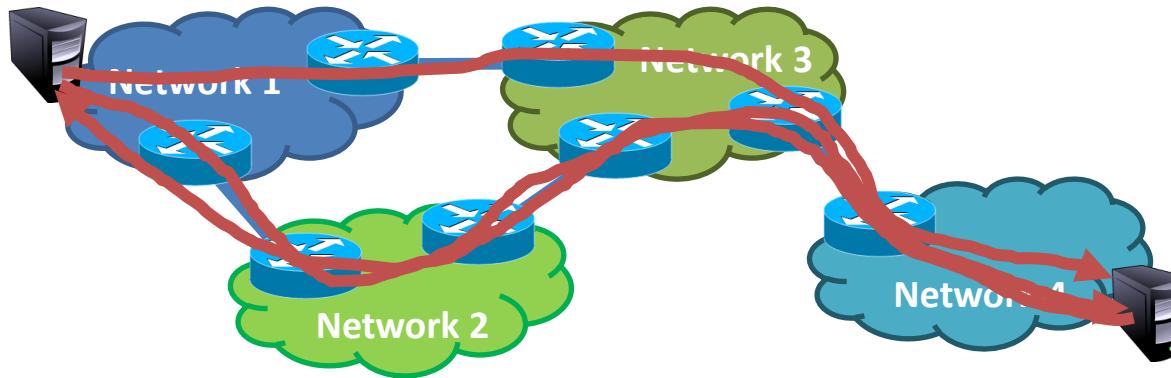
Robithoh Annur
Andreas Bara Timur
Monterico Andrian

Introductions

- How to connect multiple LANs?
- LANs may be incompatible
 - Ethernet, Wifi, etc...
- Connected networks form an **internetwork**
 - The Internet is the best known example



Structure of the Internet



- Ad-hoc interconnection of networks
 - No organized topology
 - Vastly different technologies, link capacities
- Packets travel end-to-end by hopping through networks
 - Routers “peer” (connect) different networks
 - Different packets may take different routes

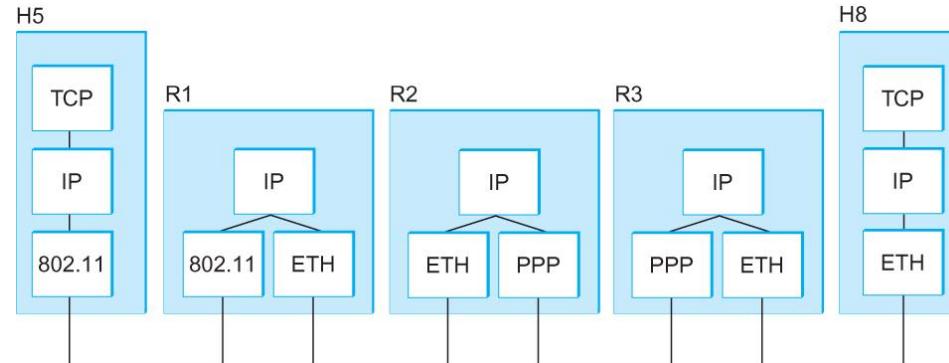


Outline

- ❑ Addressing
 - ❑ Class-based
 - ❑ CIDR
 - ❑ IP forwarding
 - ❑ NAT
- ❑ IPv4 Protocol Details
 - ❑ Packed Header
 - ❑ Fragmentation
- ❑ IPv6

Internet Protocol

- What is IP
 - IP stands for Internet Protocol
 - Key tool used today to build scalable, heterogeneous internetworks
 - It runs on all the nodes in a collection of networks and defines the infrastructure that allows these nodes and networks to function as a single logical internetwork



A simple internetwork showing the protocol layers



IP Service Model

- Packet Delivery Model
 - Connectionless model for data delivery
 - Best-effort delivery (unreliable service)
 - packets are lost
 - packets are delivered out of order
 - duplicate copies of a packet are delivered
 - packets can be delayed for a long time
- Global Addressing Scheme
 - Provides a way to identify all hosts in the network

IP Addressing

- Globally unique (for “public” IP addresses)
- **IP address:** IPv4 32-bit identifier for host, router *interface*
- **Interface:** connection between host/router and physical link
 - router’s typically have multiple interfaces
 - host may have multiple interfaces
 - IP addresses associated with each interface
- Usually written in dotted notation, e.g. 192.168.21.76
- Each number is a byte
- Stored in Big Endian order

	0	8	16	24	31
Decimal	192	168	21	76	
Hex	C0	A8	15	4C	
Binary	11000000	10101000	00010101	01001100	

IP Addressing: Network vs. Host

- **Two-level hierarchy** → Separate the address into a network and a host
 - network part (high order bits)
 - host part (low order bits)
- **What's a network ?**
(from IP address perspective)
 - device interfaces with same network part of IP address
 - can physically reach each other without intervening router

0

Pfx

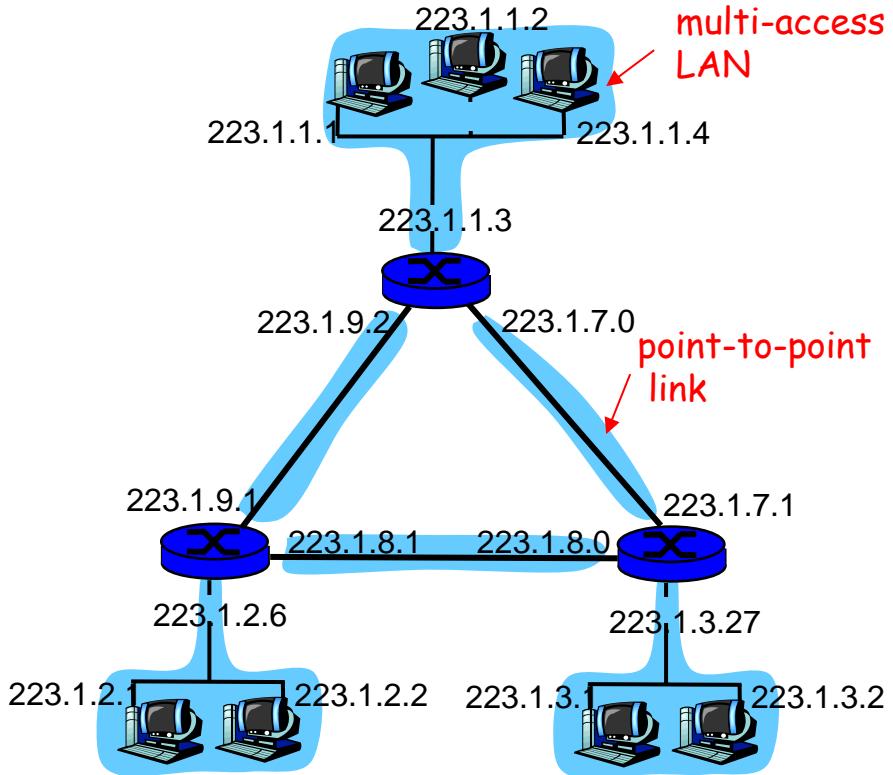
Network

Host

31

Known by all routers

Known by edge (LAN) routers



“Classful” IP Addressing

class	7	15	23	31		
A	Onetwork		host		1.0.0.0 to 127.255.255.255	$2^{24} - 2 = 16,777,214$ (All 0 and all 1 are reserved)
B	10	network		host	128.0.0.0 to 191.255.255.255	$2^{16} - 2 = 65,534$ (All 0 and all 1 are reserved)
C	110	network		host	192.0.0.0 to 223.255.255.255	$2^8 - 2 = 254$ (All 0 and all 1 are reserved)
D	1110		multicast address		224.0.0.0 to 239.255.255.255	
↔ 32 bits ↔						

- Disadvantage: inefficient use of address space; address space exhaustion
- e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network

CIDR: Classless InterDomain Routing

- A technique that addresses two scaling concerns in the Internet
 - The growth of backbone routing table as more and more network numbers need to be stored in them
 - Potential exhaustion of the 32-bit address space
- Address assignment efficiency
 - Arises because of the IP address structure with class A, B, and C addresses
 - Forces us to hand out network address space in fixed-size chunks of three very different sizes
 - A network with two hosts needs a class C address
 - » Address assignment efficiency = $2/255 = 0.78$
 - A network with 256 hosts needs a class B address
 - » Address assignment efficiency = $256/65535 = 0.39$

Classless Addressing: CIDR

CIDR: Classless InterDomain Routing

- Network portion of address is of **arbitrary length**
- Addresses allocated in contiguous blocks
 - Number of addresses assigned always power of 2
- Address format: **a.b.c.d/x**
 - x is number of bits in network portion of address



200.23.16.0/23



Classless Addressing

- CIDR tries to balance the desire to minimize the number of routes that a router needs to know against the need to hand out addresses efficiently.
- CIDR uses aggregate routes
 - Uses a single entry in the forwarding table to tell the router how to reach a lot of different networks
 - Breaks the rigid boundaries between address classes



Representation of Address Blocks

- “Human Readable” address format: **a.b.c.d/x**
 - x is number of bits in network portion of address, the network portion is also called the **network prefix**
- machine representation of a network (addr block):
using a combination of
 - first IP of address blocks of the network
 - network mask (x “1”’s followed by 32-x “0”’s

network w/ address block: 200.23.16.0/23

first IP address of address block:

11001000 00010111 00010000 00000000

network mask:

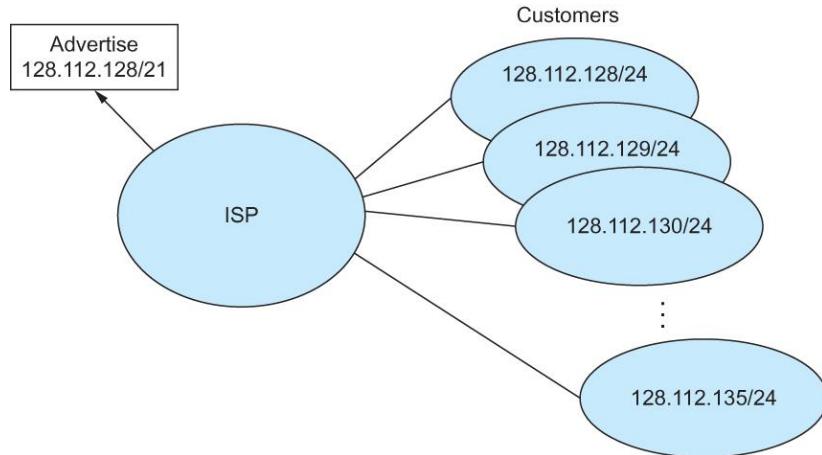
11111111 11111111 11111110 00000000



Classless Addressing

- Consider an Autonomous System (AS) with 16 class C network numbers.
- Instead of handing out 16 addresses at random, hand out a block of contiguous class C addresses
- Suppose we assign the class C network numbers from 192.4.16 through 192.4.31
- Observe that top 20 bits of all the addresses in this range are the same (11000000 00000100 0001)
 - We have created a 20-bit network number (which is in between class B network number and class C number)
- Requires to hand out blocks of class C addresses that share a common prefix
- The convention is to place a /X after the prefix where X is the prefix length in bits
- For example, the 20-bit prefix for all the networks 192.4.16 through 192.4.31 is represented as 192.4.16/20
- By contrast, if we wanted to represent a single class C network number, which is 24 bits long, we would write it 192.4.16/24

Classless Addressing



Route aggregation with CIDR



IP Addresses: How to Get One? ...

Q: How does a *network* get network part of IP addr?

A: gets an allocated portion of its provider ISP's address space

ISP's block	<u>11001000</u> <u>00010111</u> <u>00010000</u> <u>00000000</u>	200.23.16.0/20
Organization 0	<u>11001000</u> <u>00010111</u> <u>00010000</u> <u>00000000</u>	200.23.16.0/23
Organization 1	<u>11001000</u> <u>00010111</u> <u>00010010</u> <u>00000000</u>	200.23.18.0/23
Organization 2	<u>11001000</u> <u>00010111</u> <u>00010100</u> <u>00000000</u>	200.23.20.0/23
...
Organization 7	<u>11001000</u> <u>00010111</u> <u>00011110</u> <u>00000000</u>	200.23.30.0/23



Host Configurations

- Notes
 - Ethernet addresses are configured into network by manufacturer and they are unique
 - IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork
 - Most host Operating Systems provide a way to manually configure the IP information for the host
 - Drawbacks of manual configuration
 - A lot of work to configure all the hosts in a large network
 - Configuration process is error-prune
 - Automated Configuration Process is required



Dynamic Host Configuration Protocol (DHCP)

Goal: allow host to *dynamically* obtain its IP address from network DHCP server when it joins network

- Can renew its lease on address in use

- Allows reuse of addresses (only hold address while connected as “on”)

- Support for mobile users who want to join network (more shortly)

- DHCP server is responsible for providing configuration information to hosts
- There is at least one DHCP server for an administrative domain
- DHCP server maintains a pool of available addresses



Outline

- ❑ Addressing
 - ❑ Class-based
 - ❑ CIDR
 - ❑ IP forwarding
 - ❑ NAT
- ❑ IPv4 Protocol Details
 - ❑ Packed Header
 - ❑ Fragmentation
- ❑ IPv6

IP Forwarding: Destination in Same Net

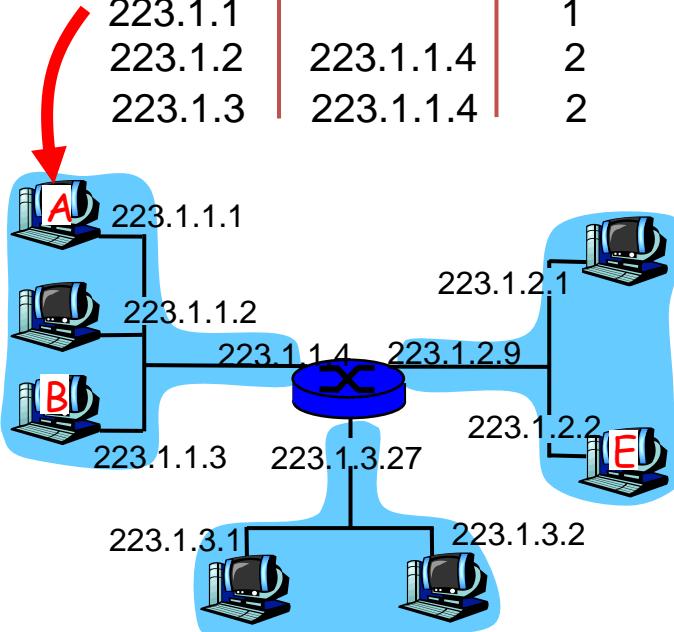
misc fields	223.1.1.1	223.1.1.3	data
-------------	-----------	-----------	------

Starting at A, send IP datagram addressed to B:

- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
 - B and A are directly connected

forwarding table in A

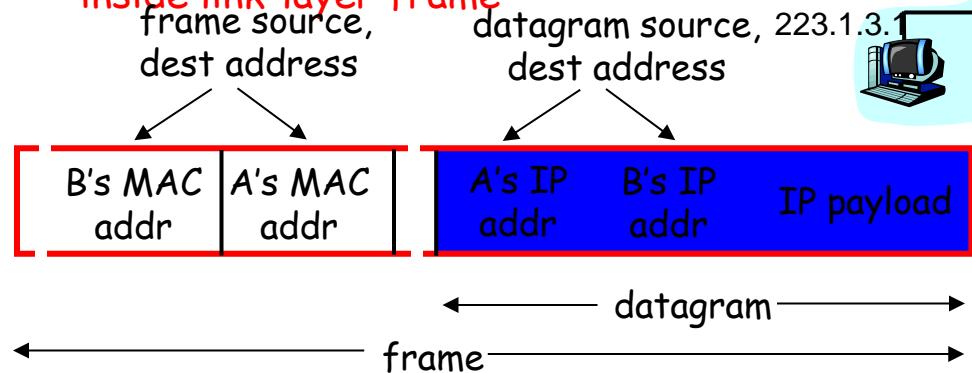
Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



IP Datagram Forwarding on Same LAN: Interaction of IP and data link layers

Starting at A, given IP datagram addressed to B:

- look up net. address of B, find B on same net. as A
- **link layer send datagram to B inside link-layer frame**





MAC (Physical) Addresses -- Revisited

- used to get frames from one interface to another physically-connected interface (same physical network, i.e., p2p or LAN)
- 48 bit MAC address (for most LANs)
 - fixed for each adaptor, burned in the adapter ROM
 - MAC address allocation administered by IEEE
 - 1st bit: 0 unicast, 1 multicast.
 - all 1's : broadcast
- MAC flat address -> portability
 - can move LAN card from one LAN to another
- MAC addressing operations on a LAN:
 - each adaptor on the LAN “sees” all frames
 - accept a frame if dest. MAC address matches its own MAC address
 - accept all broadcast (MAC= all1's) frames
 - accept all frames if set in “promiscuous” mode
 - can configure to accept certain multicast addresses (first bit = 1)



MAC vs. IP Addresses

32-bit IP address:

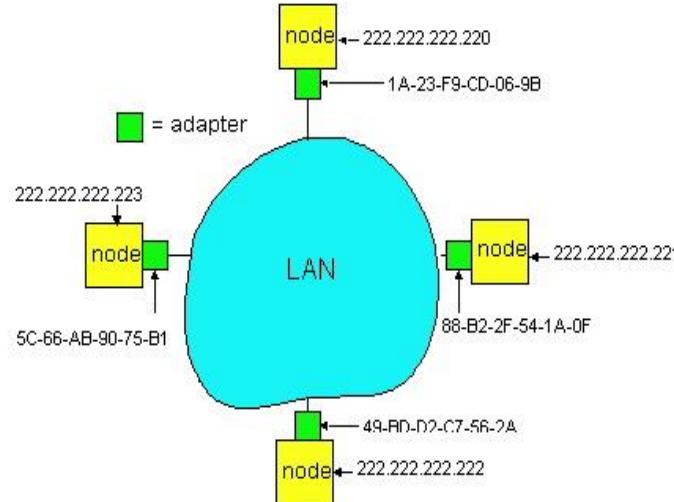
- *network-layer* address, logical
 - i.e., not bound to any physical device, can be re-assigned
- IP hierarchical address NOT portable
 - depends on IP network to which an interface is attached
 - when move to another IP network, IP address re-assigned
- used to get IP packets to destination IP network
 - Recall how IP datagram forwarding is performed
- **IP network is “virtual,” actually packet delivery done by the underlying physical networks**
 - from source host to destination host, hop-by-hop via IP routers
 - over each link, different link layer protocol used, with its own frame headers, and source and destination MAC addresses
 - Underlying physical networks do not understand IP protocol and datagram format!

ARP: Address Resolution Protocol

- Each IP node (host, router) on LAN has ARP table
 - ARP Table: IP/MAC address mappings for some LAN nodes
- < IP address; MAC address; timer>

— timer: time after which address mapping will be forgotten (typically 15 min)

Question: how to determine MAC address of B knowing B's IP address?





What does ARP do?

- The main functions of ARP
 - Obtaining the MAC address of an destination IP.
 - Forming the ARP table with lookup entry of “destination IP to MAC address”
- Issued by a host OS that tries to obtain the MAC address of an destination IP (automatically).
- After obtaining the MAC address of the “desired destination IP” thru ARP, the host will use the information to form an entry in the ARP table (or ARP cache)
 - Remember that the Frame MUST have the destination MAC address before sending out thru the wire.
- There are two parts of the ARP
 - ARP request (issued by the source host)
 - ARP reply (issued by the destination host)

```
D:\>Documents and Settings\Administrator>arp -a
Interface: 172.16.10.16 --- 0x4
  Internet Address      Physical Address          Type
  172.16.10.1           00-15-f9-04-57-93        dynamic
  172.16.10.3           00-15-fa-a4-99-4a        dynamic
  172.16.10.129         00-13-46-32-af-45        dynamic
```

ARP Table



ARP Protocol

- A wants to send datagram to B, and A knows B's IP address.
- A looks up B's MAC address in its ARP table
- Suppose B's MAC address is not in A's ARP table.
- A **broadcasts (why?)** ARP query packet, containing B's IP address
 - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
 - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
 - soft state: information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:
 - nodes create their ARP tables without intervention from net administrator



ARP Messages

0	8	16	24	31			
HARDWARE ADDRESS TYPE		PROTOCOL ADDRESS TYPE					
HADDR LEN	PADDR LEN	OPERATION					
SENDER HADDR (first 4 octets)							
SENDER HADDR (last 2 octets)		SENDER PADDR (first 2 octets)					
SENDER PADDR (last 2 octets)		TARGET HADDR (first 2 octets)					
TARGET HADDR (last 4 octets)							
TARGET PADDR (all 4 octets)							

Hardware Address Type: e.g., Ethernet

Protocol address Type: e.g., IP

Operation: ARP request or ARP response

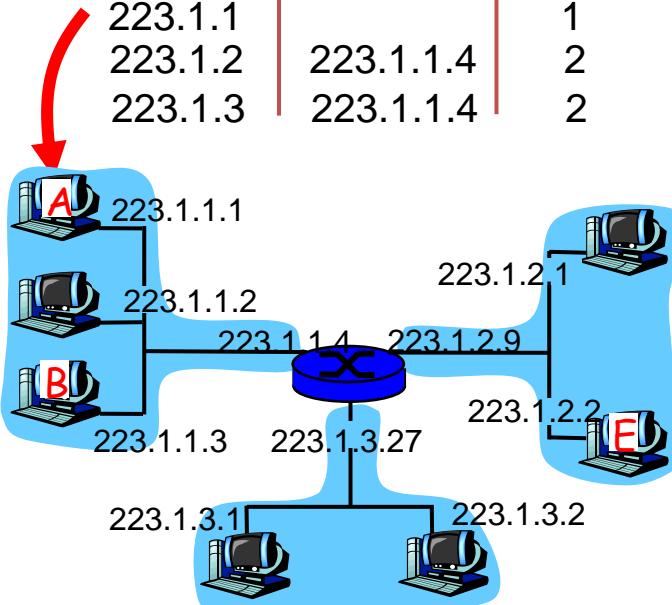
IP Forwarding: Destination in Different Network

misc fields	223.1.1.1	223.1.2.3	data
-------------	-----------	-----------	------

- Starting at A, dest. E:
- look up network address of E in forwarding table
- E on different network
 - A, E not directly attached
- routing table: next hop router to E is 223.1.1.4
- link layer sends datagram to router 223.1.1.4 inside link-layer frame
- datagram arrives at 223.1.1.4
- continued.....

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



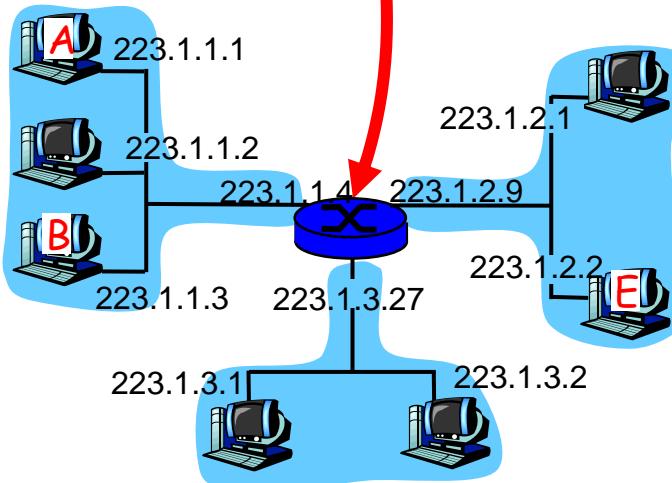
IP Forwarding: Destination in Diff. Net ...

misc fields	223.1.1.1	223.1.2.3	data
-------------	-----------	-----------	------

- Arriving at 223.1.4, destined for 223.1.2.2
- look up network address of E in router's forwarding table
- E on same network as router's interface 223.1.2.9
 - router, E directly attached
- link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!! (hooray!)

forwarding table in router

Dest. Net	router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27



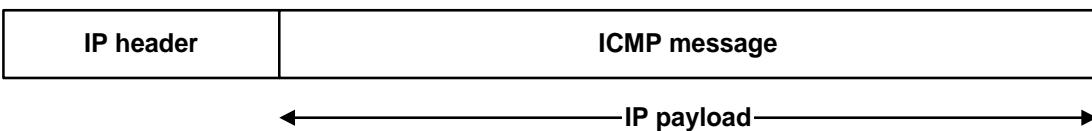
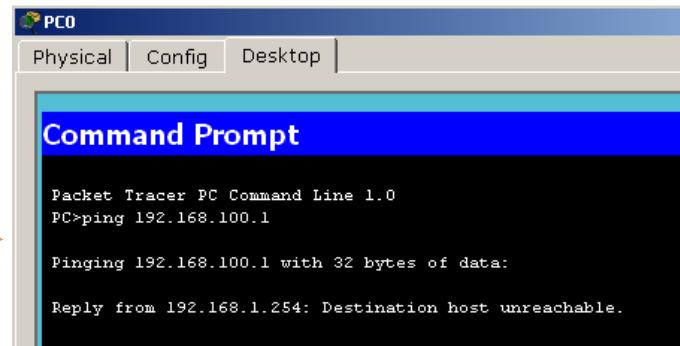
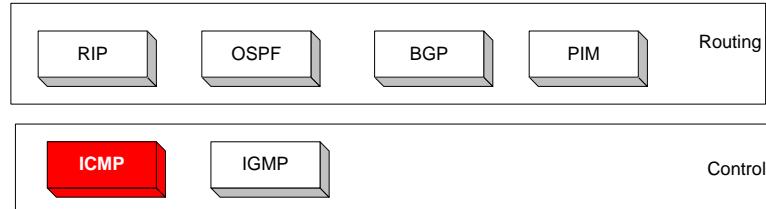


IP Forwarding Revisited

- IP forwarding mechanism assumes that it can find the network number in a packet and then look up that number in the forwarding table
- We need to change this assumption in case of CIDR
- CIDR means that prefixes may be of any length, from 2 to 32 bits
- It is also possible to have prefixes in the forwarding tables that overlap
 - Some addresses may match more than one prefix
- For example, we might find both 171.69 (a 16 bit prefix) and 171.69.10 (a 24 bit prefix) in the forwarding table of a single router
- A packet destined to 171.69.10.5 clearly matches both prefixes.
 - The rule is based on the principle of “longest match”
 - 171.69.10 in this case
- A packet destined to 171.69.20.5 would match 171.69 and not 171.69.10

Internet Control Message Protocol (ICMP)

- The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:
 - Control functions (ICMP)
 - Multicast signaling (IGMP)
 - Setting up routing tables (RIP, OSPF, BGP, PIM, ...)
- The **Internet Control Message Protocol (ICMP)** is a helper protocol that supports IP with facility for
 - Error reporting
 - Simple queries
- ICMP messages are encapsulated as IP datagrams:





Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
 - Destination host unreachable due to link /node failure
 - Reassembly process failed
 - TTL had reached 0 (so datagrams don't cycle forever)
 - IP header checksum failed

Frequent ICMP Error message

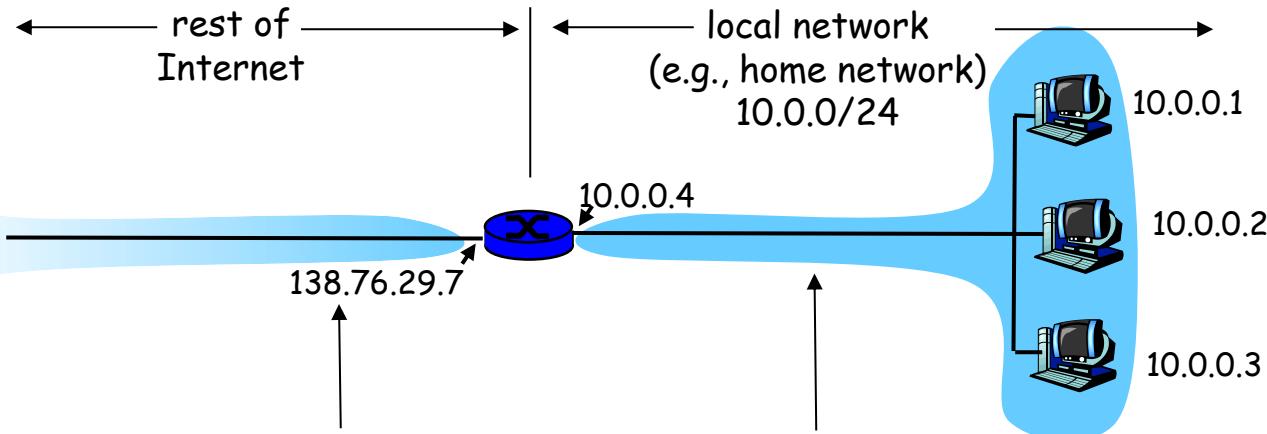
Type	Code	Description	
3	0–15	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.
11	0, 1	Time exceeded	Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)
12	0, 1	Parameter problem	Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)



Outline

- ❑ Addressing
 - ❑ Class-based
 - ❑ CIDR
 - ❑ IP forwarding
 - ❑ NAT
- ❑ IPv4 Protocol Details
 - ❑ Packed Header
 - ❑ Fragmentation
- ❑ IPv6

NAT: Network Address Translation



All datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

10.0.0.0/8 has been reserved for private networks!



Why do we need this “NAT”?

- NAT is used for three major reasons:
 - IPv4 address exhaustion
 - Masquerading for security purpose
 - TCP load sharing
- NAT for alleviating the consequences of IPv4 address exhaustion.
 - It has become a standard, indispensable feature in routers for home and small-office Internet connections.
 - One public IP can be used by thousands of private network computers.
- NAT as IP masquerading
 - Obscures an internal network's structure,
 - All network traffic appears to outside network as if it is originated from the one IP address of a router.
- NAT for TCP load sharing
 - Useful for server farm
 - A few servers with similar functions represented by one single IP address.



NAT: Network Address Translation

- **Then:** local network uses just one IP address as far as outside world is concerned:
 - no need to be allocated range of addresses from ISP: - just one IP address is used for all devices
 - can change addresses of devices in local network without notifying outside world
 - can change ISP without changing addresses of devices in local network
 - devices inside local net not explicitly addressable, visible by outside world (a security plus).



NAT: Network Address Translation

Implementation: NAT router must:

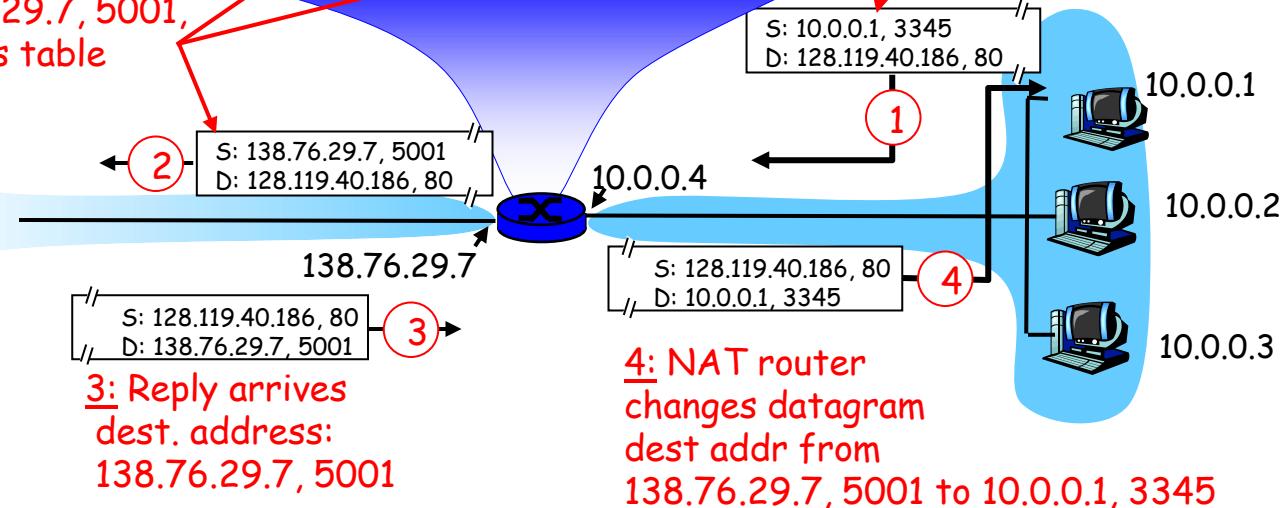
- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
 - ... remote clients/servers will respond using (NAT IP address, new port #) as destination addr.
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
.....

1: host 10.0.0.1 sends datagram to 128.119.40, 80





Four Types of NAT

- Static Network Address Translation (static NAT)
 - 1 private IP to 1 global IP address translation
- Dynamic Network Address Translation (dynamic NAT)
 - Many private IP to many global IP address translation (a pool of IP)
- Port Address Translation (PAT)
 - Many private IP to 1 global IP address translation.
 - Is also called NAT overloading.
 - 2 sub-mode: Interface mode & pool mode
- Port Forwarding (Type of static NAT)
 - Accessing a inside local network service from outside global host.

More Details in in the practical labs

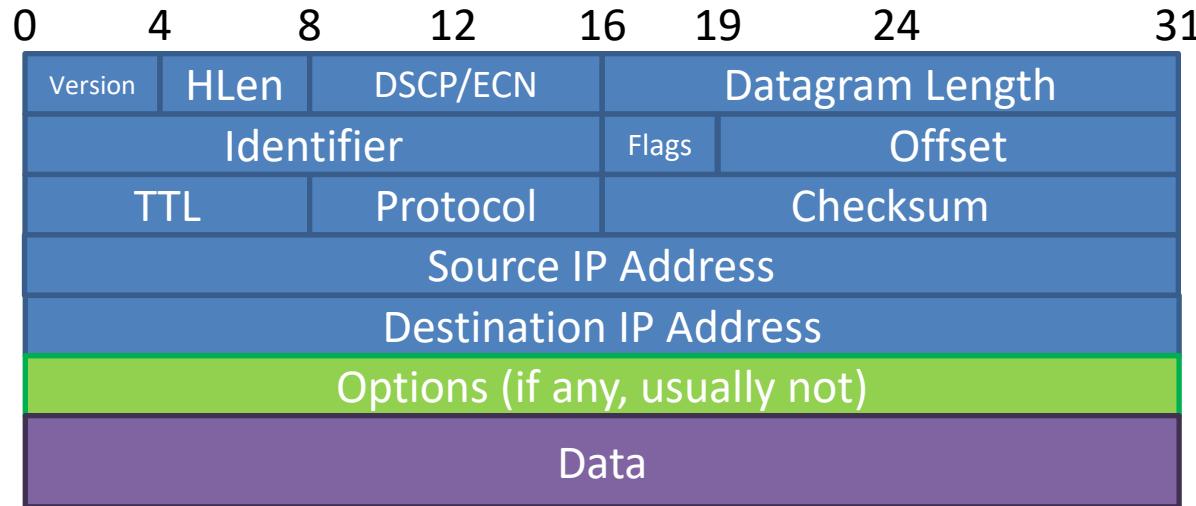


Outline

- ❑ Addressing
 - ❑ Class-based
 - ❑ CIDR
 - ❑ IP forwarding
 - ❑ NAT
- ❑ IPv4 Protocol Details
 - ❑ Packed Header
 - ❑ Fragmentation
- ❑ IPv6

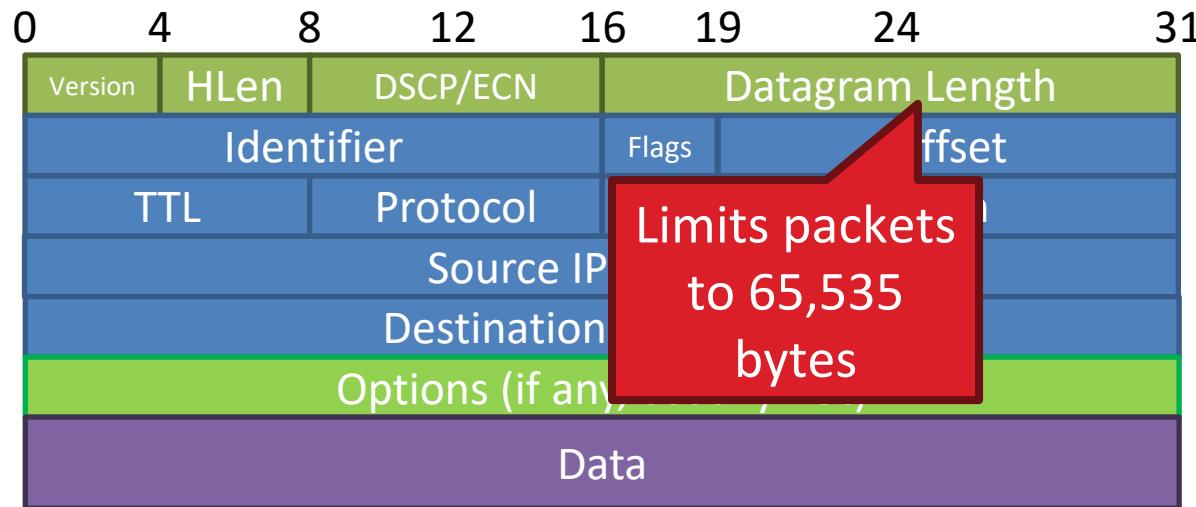
IP Datagrams

- IP Datagrams are like a letter
 - Totally self-contained
 - Include all necessary addressing information
 - No advanced setup of connections or circuits



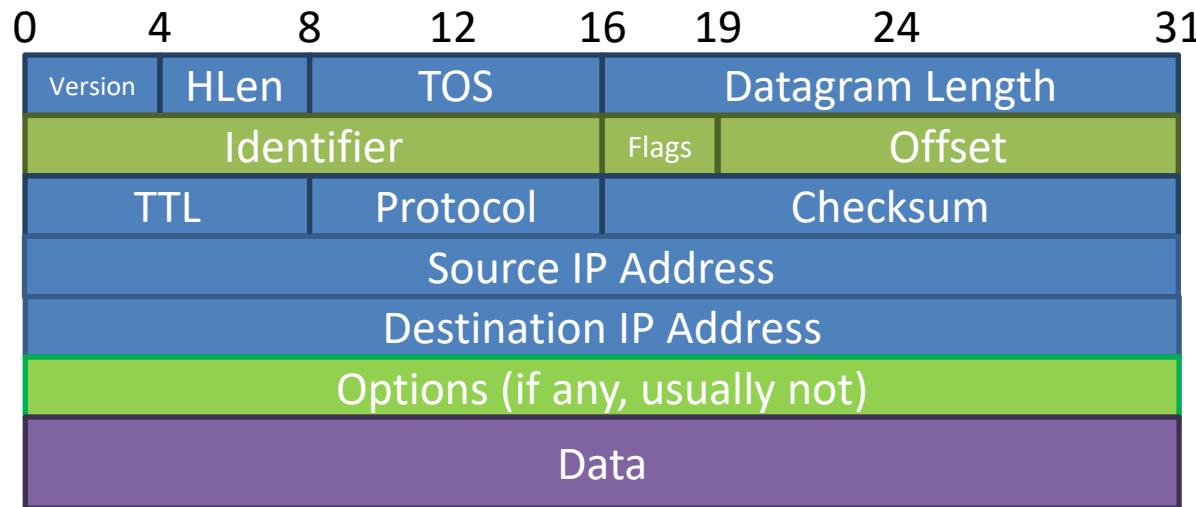
IP Header Fields: Word 1

- Version: 4 for IPv4
- Header Length: Number of 32-bit words (usually 5)
- Type of Service: Priority information (unused)
- Datagram Length: Length of header + data in bytes



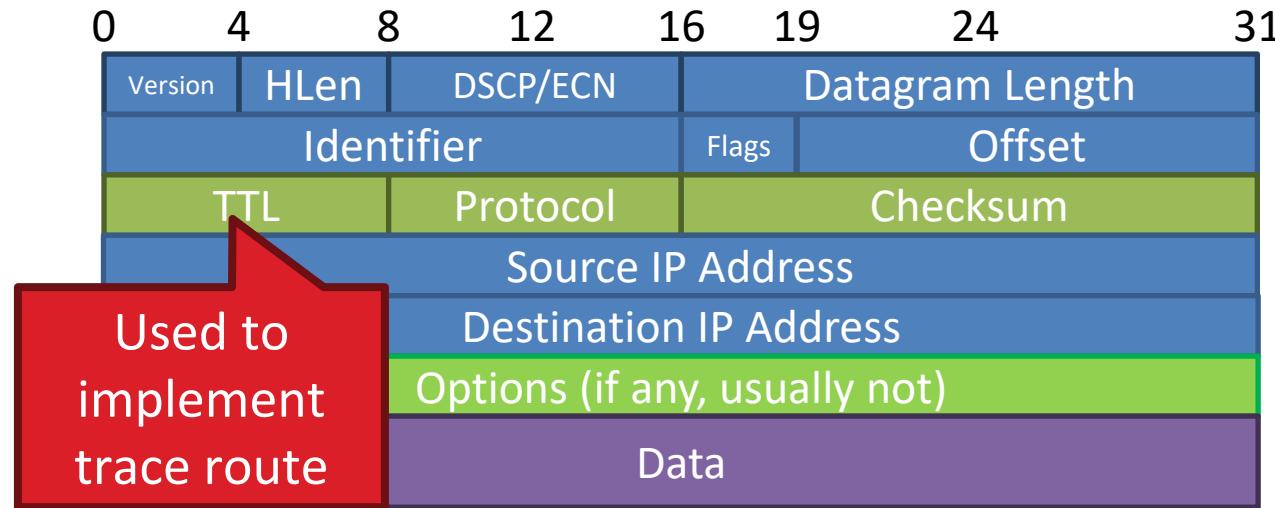
IP Header Fields: Word 2

- Identifier: a unique number for the original datagram
- Flags: M flag, i.e. this is the last fragment
- Offset: byte position of the first byte in the fragment
 - Divided by 8



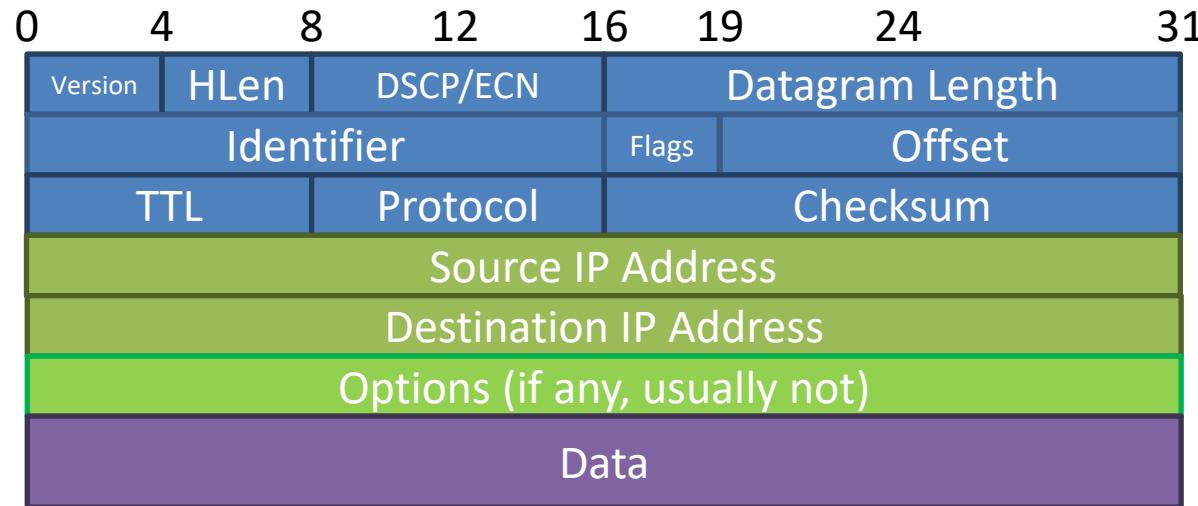
IP Header Fields: Word 3

- Time to Live: decremented by each router
 - Used to kill looping packets
- Protocol: ID of encapsulated protocol
 - 6 = TCP, 17 = UDP
- Checksum



IP Header Fields: Word 4 and 5

- Source and destination address
 - In theory, must be globally unique
 - In practice, this is often violated

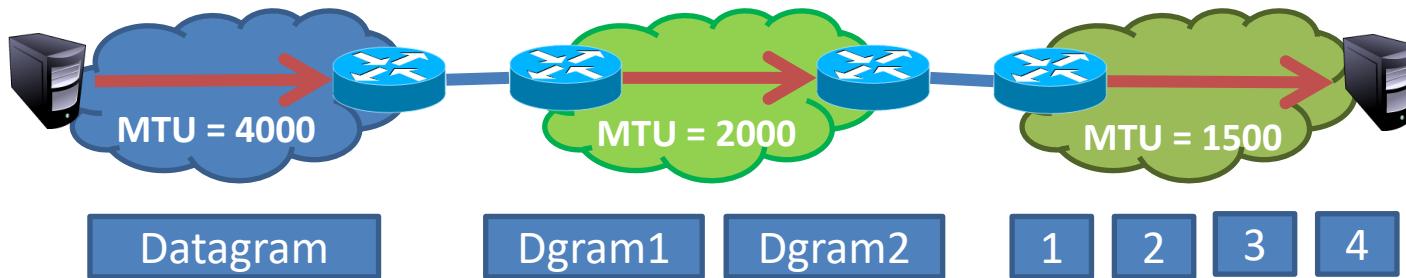




Outline

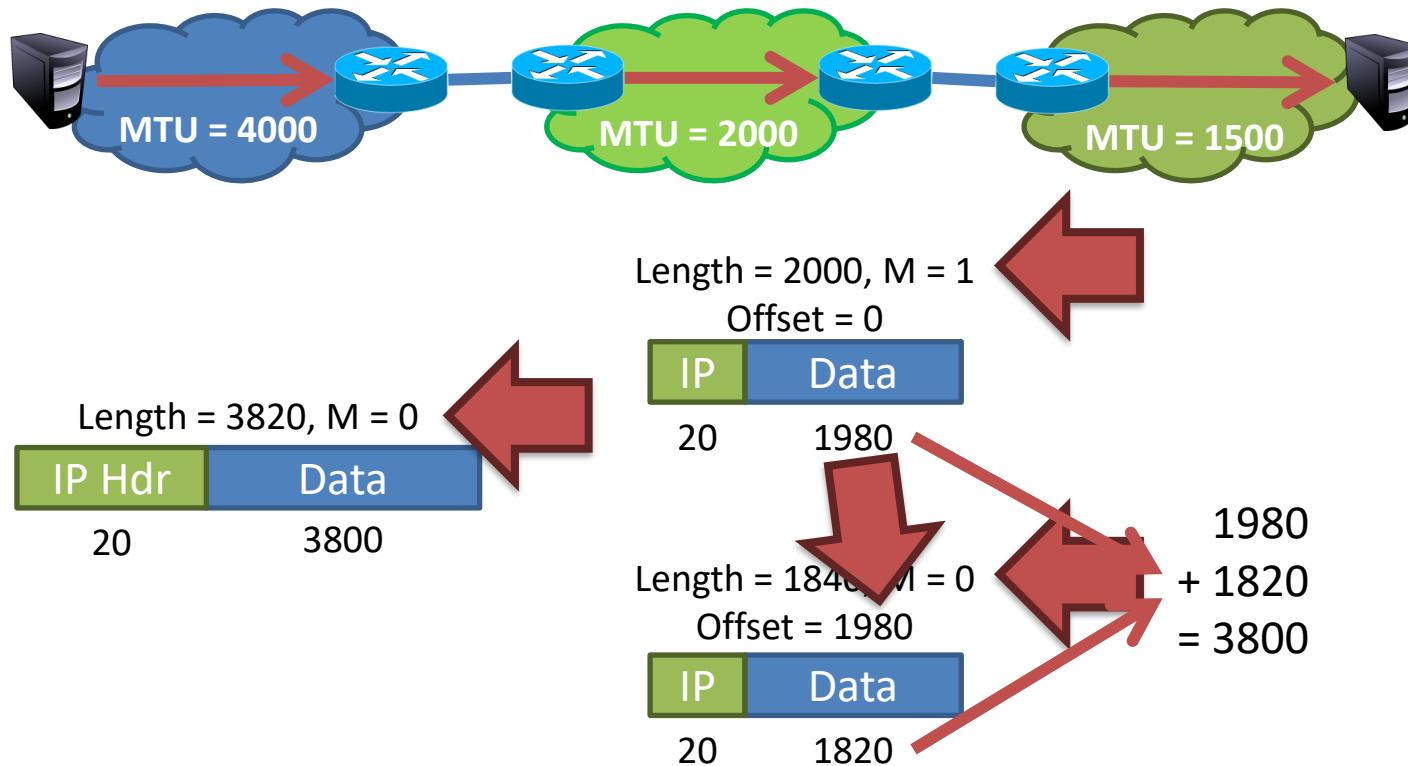
- ❑ Addressing
 - ❑ Class-based
 - ❑ CIDR
 - ❑ IP forwarding
 - ❑ NAT
- ❑ IPv4 Protocol Details
 - ❑ Packed Header
 - ❑ Fragmentation
- ❑ IPv6

Problem: Fragmentation

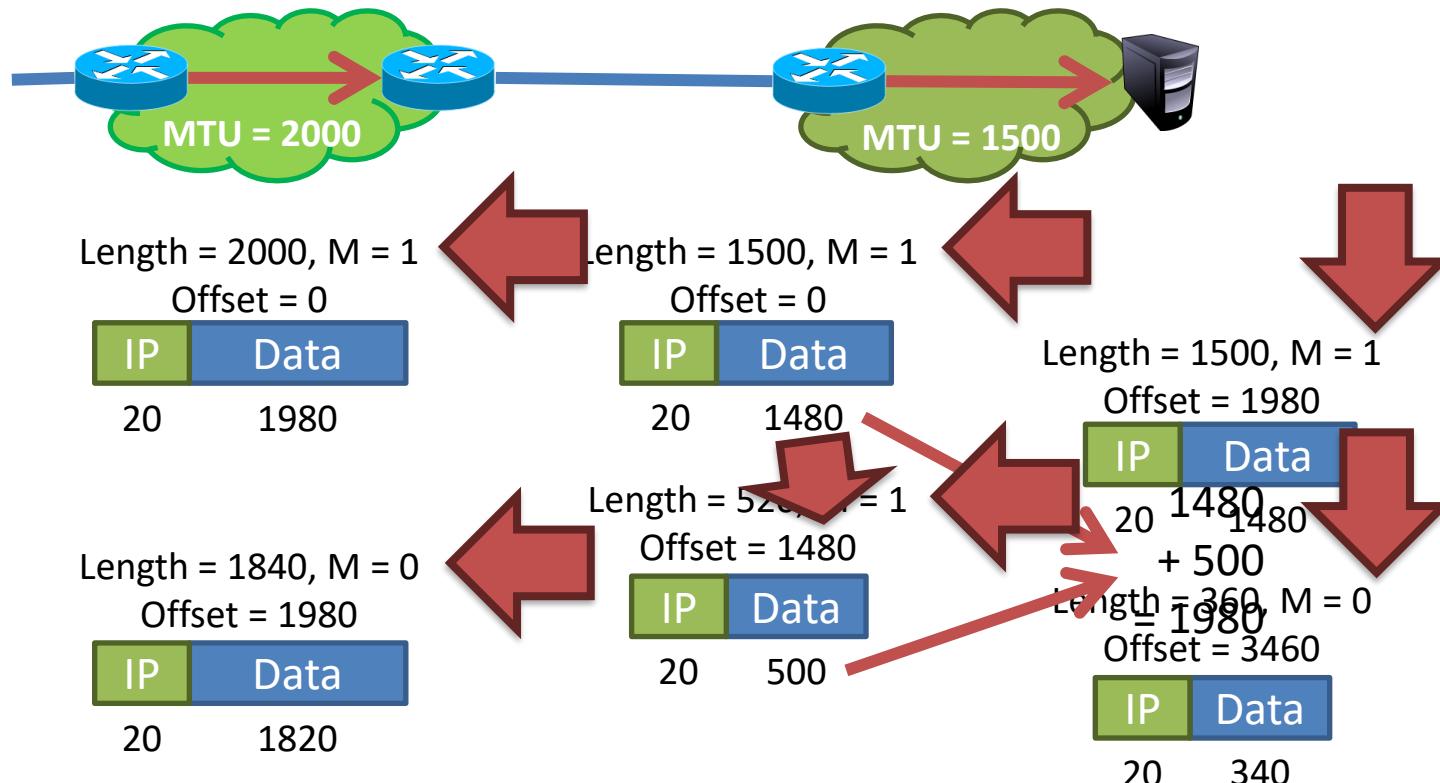


- Problem: each network has its own MTU
 - DARPA principles: networks allowed to be heterogeneous
 - Minimum MTU may not be known for a given path
- IP Solution: fragmentation
 - Split datagrams into pieces when MTU is reduced
 - Reassemble original datagram at the receiver

Fragmentation Example



Fragmentation Example





IP Fragment Reassembly

Length = 1500, M = 1, Offset = 0

IP	Data
20	1480

Length = 520, M = 1, Offset = 1480

IP	Data
20	500

Length = 1500, M = 1, Offset = 1980

IP	Data
20	1480

Length = 360, M = 0, Offset = 3460

IP	Data
20	340

- Performed at destination
- M = 0 fragment gives us total data size
 - $360 - 20 + 3460 = 3800$
- Challenges:
 - Out-of-order fragments
 - Duplicate fragments
 - Missing fragments
- Basically, memory management nightmare



Fragmentation Concepts

- Highlights many key Internet characteristics
 - Decentralized and heterogeneous
 - Each network may choose its own MTU
 - Connectionless datagram protocol
 - Each fragment contains full routing information
 - Fragments can travel independently, on different paths
 - Best effort network
 - Routers/receiver may silently drop fragments
 - No requirement to alert the sender
 - Most work is done at the endpoints
 - i.e. reassembly



Outline

- ❑ Addressing
 - ❑ Class-based
 - ❑ CIDR
 - ❑ IP forwarding
 - ❑ NAT
- ❑ IPv4 Protocol Details
 - ❑ Packed Header
 - ❑ Fragmentation
- ❑ IPv6



The IPv4 Address Space Crisis

- Problem: the IPv4 address space is too small
 - $2^{32} = 4,294,967,296$ possible addresses
 - Less than one IP per person
- Parts of the world have already run out of addresses
 - IANA assigned the last /8 block of addresses in 2011

Region	Regional Internet Registry (RIR)	Exhaustion Date
Asia/Pacific	APNIC	April 19, 2011
Europe/Middle East	RIPE	September 14, 2012
North America	ARIN	13 Jan 2015 (Projected)
South America	LACNIC	13 Jan 2015 (Projected)
Africa	AFRINIC	17 Jan 2022(Projected)

- IPv6, first introduced in 1998(!)
 - 128-bit addresses
 - $4.8 * 10^{28}$ addresses per person
- Address format
 - 8 groups of 16-bit values, separated by ‘:’
 - Leading zeroes in each group may be omitted
 - Groups of zeroes can be omitted using ‘::’

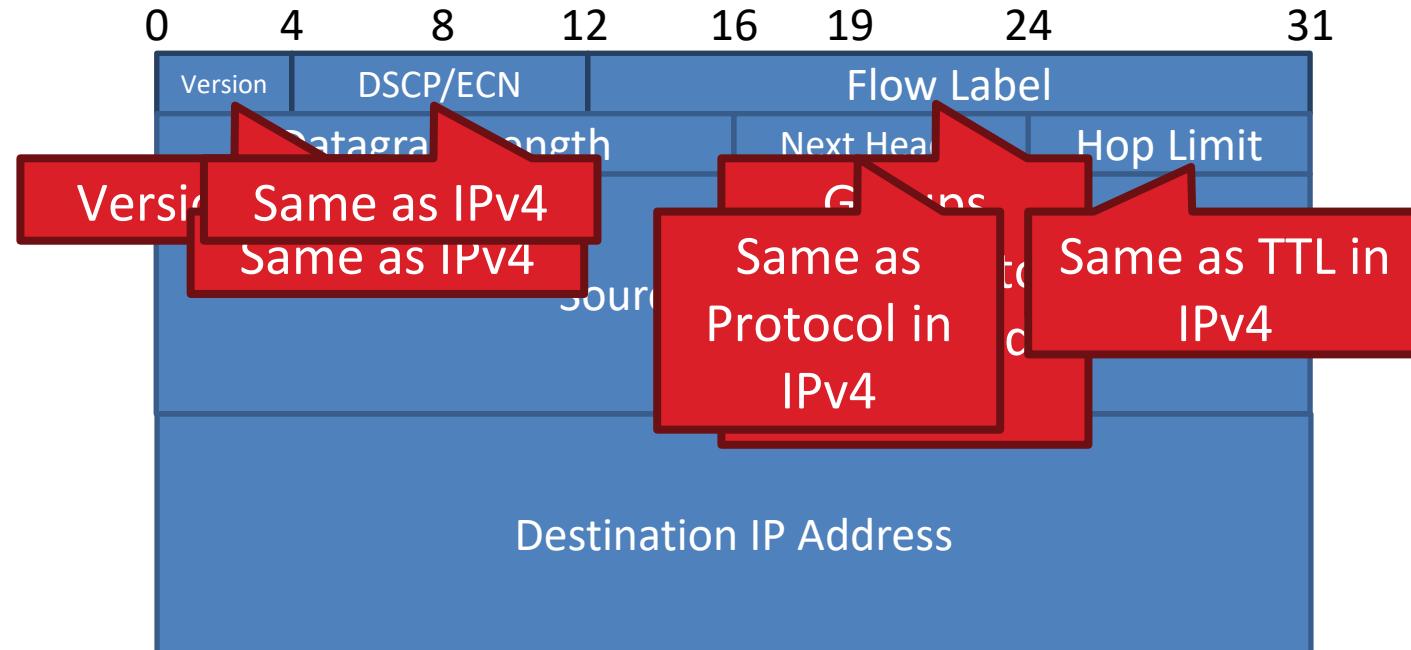
2001:0db8:0000:0000:0000:ff00:0042:8329

2001:~~0~~db8:0:0:0:~~0~~ff00:42:~~8~~329

2001:0db~~8~~::~~ff~~00:42:8329

IPv6 Header

- Double the size of IPv4 (320 bits vs. 160 bits)





Differences from IPv4 Header

- Several header fields are missing in IPv6
 - Header length – rolled into Next Header field
 - Checksum – was useless, so why keep it
 - Identifier, Flags, Offset
 - IPv6 routers do not support fragmentation
 - Hosts are expected to use path MTU discovery
- Reflects changing Internet priorities
 - Today's networks are more homogeneous
 - Instead, routing cost and complexity dominate



Performance Improvements

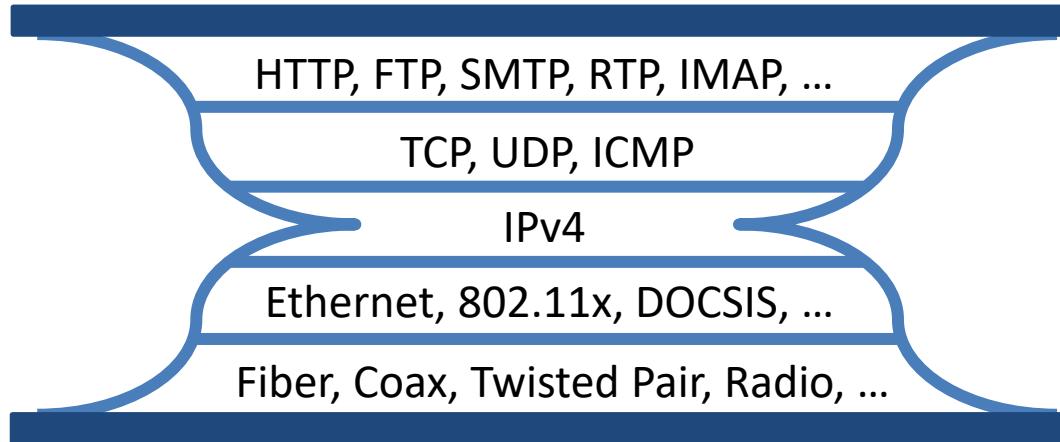
- No checksums to verify
- No need for routers to handle fragmentation
- Simplified routing table design
 - Address space is huge
 - No need for CIDR (but need for aggregation)
 - Standard subnet size is 2^{64} addresses
- Simplified auto-configuration
 - Neighbor Discovery Protocol
 - Used by hosts to determine network ID
 - Host ID can be random!



Additional IPv6 Features

- Source Routing
 - Host specifies the route the packet wants to take
- Mobile IP
 - Hosts can take their IP with them to other networks
 - Use source routing to direct packets
- Privacy Extensions
 - Randomly generate host identifiers
 - Make it difficult to associate one IP to a host
- Jumbograms
 - Support for 4Gb datagrams

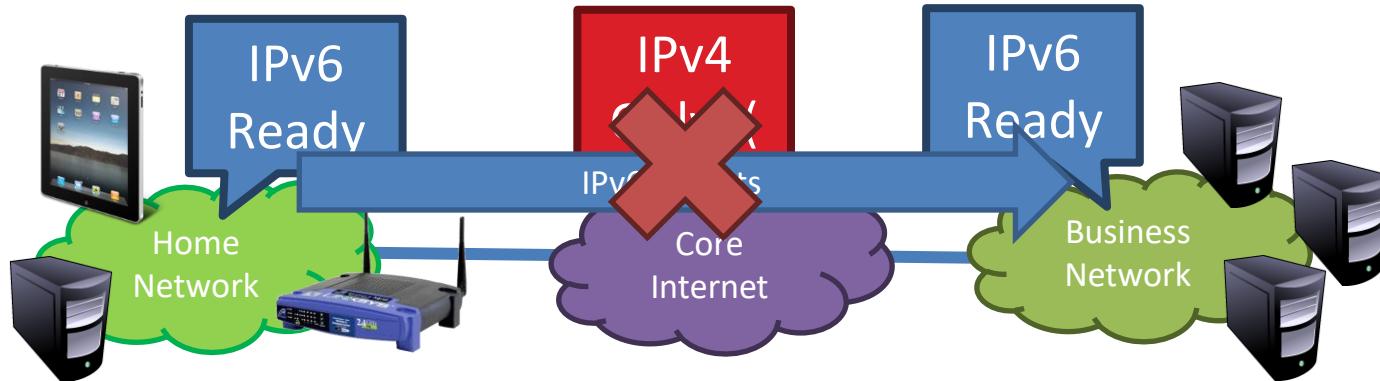
Deployment Challenges



- Switching to IPv6 is a whole-Internet upgrade
 - All routers, all hosts
 - ICMPv6, DHCPv6, DNSv6
- 2013: 0.94% of Google traffic was IPv6, 2.5% today

Transitioning to IPv6

- How do we ease the transition from IPv4 to IPv6?
 - Today, most network edges are IPv6 ready
 - Windows/OSX/iOS/Android all support IPv6
 - Your wireless access point probably supports IPv6
 - The Internet core is hard to upgrade
 - ... but a IPv4 core cannot route IPv6 traffic





Transition Technologies

- How do you route IPv6 packets over an IPv4 Internet?
- Transition Technologies
 - Use **tunnels** to **encapsulate** and route IPv6 packets over the IPv4 Internet
 - Several different implementations
 - 6to4
 - IPv6 Rapid Deployment (6rd)
 - Teredo
 - ... etc.

6to4 Basics

- Problem: you've been assigned an IPv4 address, but you want an IPv6 address
 - Your ISP can't or won't give you an IPv6 address
 - You can't just arbitrarily choose an IPv6 address
- Solution: construct a 6to4 address
 - 6to4 addresses always start with 2002::
 - Embed the 32-bit IPv4 inside the 128-bit IPv6 address





Problems with 6to4

- Uniformity
 - Not all ISPs have deployed 6to4 relays
- Quality of service
 - Third-party 6to4 relays are available
 - ...but, they may be overloaded or unreliable
- Reachability
 - 6to4 doesn't work if you are behind a NAT
- Possible solutions
 - IPv6 Rapid Deployment (6rd)
 - Each ISP sets up relays for its customers
 - Does not leverage the 2002:: address space
 - Teredo
 - Tunnels IPv6 packets through UDP/IPv4 tunnels
 - Can tunnel through NATs, but requires special relays

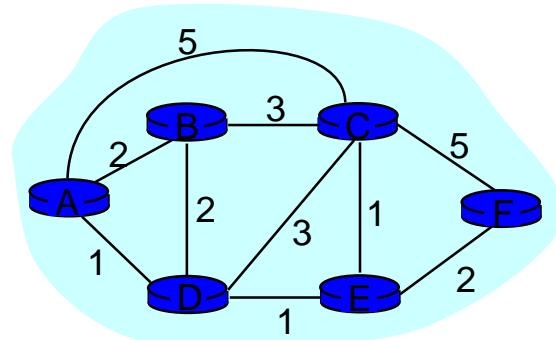


IF2230 Jaringan Komputer Routing

Robithoh Annur
Andreas Bara Timur
Monterico Andrian

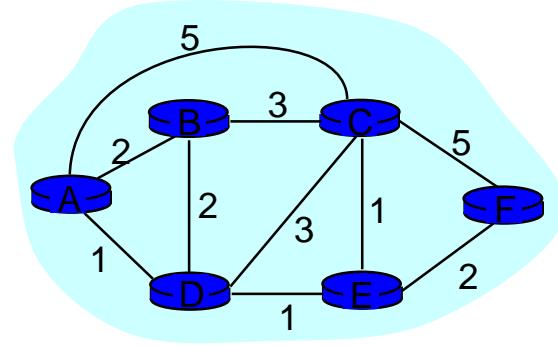
Routing

- Goal: determine a “good” path through the network from source to destination
 - Good means usually the shortest path
- Network modeled as a graph
 - Routers → nodes
 - Link → edges
 - Edge cost: delay, congestion level,...



Basic Routing Problem

- Assume
 - A network with N nodes, where each edge is associated a cost
 - A node knows **only** its neighbors and the cost to reach them
- How does each node learn how to reach every other node along the shortest path?





Routing: Issues

- How are routing tables determined?
- Who determines table entries?
- What info is used in determining table entries?
- When do routing table entries change?
- Where is routing info stored?
- How to control routing table size?

Answer these questions, we are done!

- Hop-by-hop Routing
 - Each packet contains destination address
 - Each router chooses next-hop to destination
 - routing decision made at each (intermediate) hop!
 - packets to same destination may take different paths!
 - Example: IP's default datagram routing
- Source Routing
 - Sender selects the path to destination precisely
 - Routers forward packet to next-hop as specified
 - Problem: if specified path no longer valid due to link failure!
 - Example:
 - IP's loose/strict source route option
 - virtual circuit setup phase in ATM (or MPLS)



Routing Algorithms/Protocols

Issues Need to Be Addressed:

- Route selection may depend on different criteria
 - Performance: choose route with the smallest delay
 - Policy: choose a route that doesn't cross .gov network
- Adapt to changes in network topology or condition
 - Self-healing: little or no human intervention
- Scalability
 - Must be able to support a large number of hosts, routers

Centralized vs. Distributed Routing Algorithms

Centralized:

- A centralized route server collects routing information and network topology, makes route selection decisions, then distributes them to routers

Distributed:

- Routers **cooperate** using a distributed protocol
 - to create **mutually consistent** routing tables
- Two standard **distributed** routing algorithms
 - Distance Vector (DV) routing
 - Link State (LS) routing



Link State vs Distance Vector

- Both assume that
 - The address of each neighbor is known
 - The **cost** of reaching each neighbor is known
- Both find **global** information
 - By exchanging routing info among neighbors
- Differ in the information exchanged and route computation
 - LS: tells **every other node** its **distances to neighbors**
 - DV: tells **neighbors** its **distance to every other node**

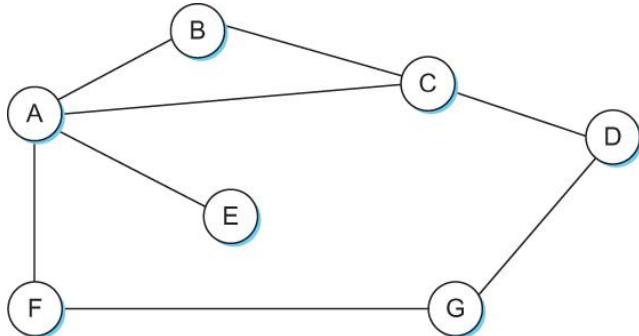


Distance Vector

- The distance vector routing algorithm is sometimes called as Bellman-Ford algorithm
- Every T seconds each router sends its table to its neighbor each each router then updates its table based on the new information
- Problems include fast response to good new and slow response to bad news. Also too many messages to update

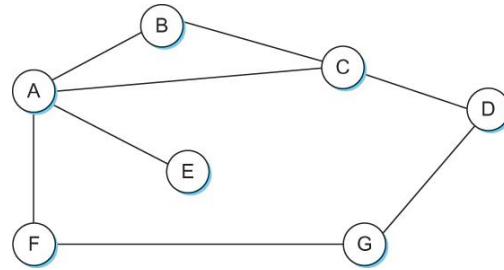
Distance Vector

- Each node constructs a one dimensional array (a vector) containing the “distances” (costs) to all other nodes and distributes that vector to its immediate neighbors
- Starting assumption is that each node knows the cost of the link to each of its directly connected neighbors



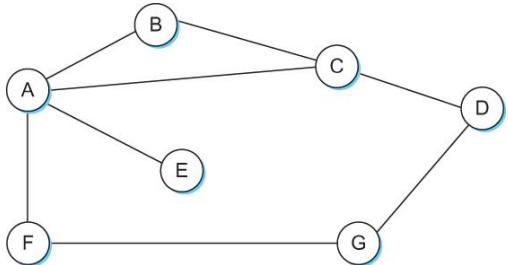
Distance Vector

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0



Initial distances stored at each node (global view)

Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

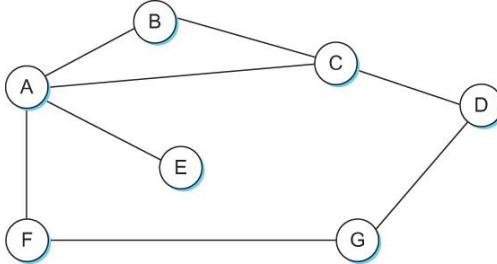
Initial routing table at node A

Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Final routing table at node A

Distance Vector

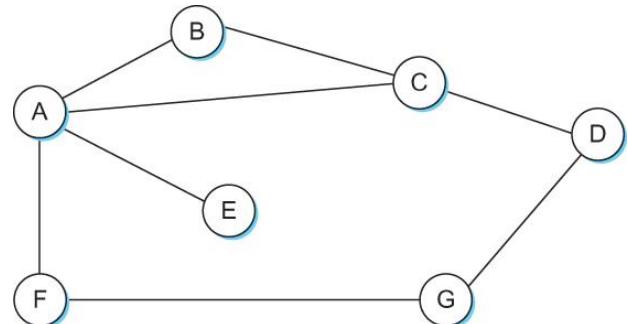
Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0



Final distances stored at each node (global view)

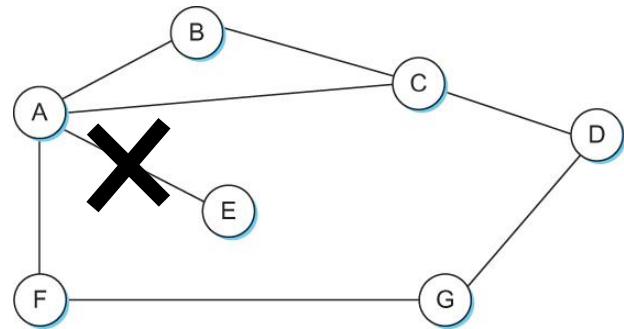
Distance Vector

- When a node detects a link failure
 - F detects that link to G has failed
 - F sets distance to G to infinity and sends update to A
 - A sets distance to G to infinity since it uses F to reach G
 - A receives periodic update from C with 2-hop path to G
 - A sets distance to G to 3 and sends update to F
 - F decides it can reach G in 4 hops via A



Distance Vector

- Slightly different circumstances can prevent the network from stabilizing
 - Suppose the link from A to E goes down
 - In the next round of updates, A advertises a distance of infinity to E, but B and C advertise a distance of 2 to E
 - Depending on the exact timing of events, the following might happen
 - Node B, upon hearing that E can be reached in 2 hops from C, concludes that it can reach E in 3 hops and advertises this to A
 - Node A concludes that it can reach E in 4 hops and advertises this to C
 - Node C concludes that it can reach E in 5 hops; and so on.
 - This cycle stops only when the distances reach some number that is large enough to be considered infinite
 - **Count-to-infinity problem**





Count-to-infinity Problem

- Use some relatively small number as an approximation of infinity
- For example, the maximum number of hops to get across a certain network is never going to be more than 16
- One technique to improve the time to stabilize routing is called *split horizon*
 - When a node sends a routing update to its neighbors, it does not send those routes it learned from each neighbor back to that neighbor
 - For example, if B has the route $(E, 2, A)$ in its table, then it knows it must have learned this route from A, and so whenever B sends a routing update to A, it does not include the route $(E, 2)$ in that update
- In a stronger version of split horizon, called *split horizon with poison reverse*
 - B actually sends that back route to A, but it puts negative information in the route to ensure that A will not eventually use B to get to E
 - For example, B sends the route (E, ∞) to A



Link State Algorithm

- Basic idea: Distribute link state packet to all routers
 - Topology of the network
 - Cost of each link in the network
- Each router **independently** computes **optimal** paths
 - From itself to every destination
 - Routes are guaranteed to be **loop free** if
 - Each router sees the same cost for each link
 - Uses the same algorithm to compute the best path



Link State Routing

Strategy: Send to all nodes (not just neighbors) information about directly connected links (not entire routing table).

- **Link State Packet (LSP)**
 - id of the node that created the LSP
 - cost of link to each directly connected neighbor
 - sequence number (SEQNO)
 - time-to-live (TTL) for this packet
- **Reliable Flooding**
 - store most recent LSP from each node
 - forward LSP to all nodes but one that sent it
 - generate new LSP periodically; increment SEQNO
 - start SEQNO at 0 when reboot
 - decrement TTL of each stored LSP; discard when TTL=0



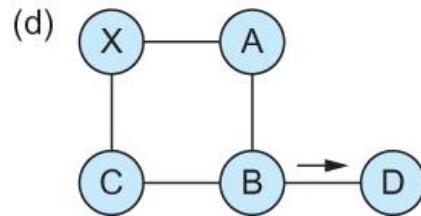
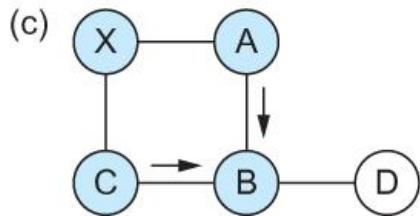
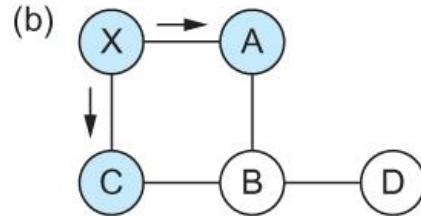
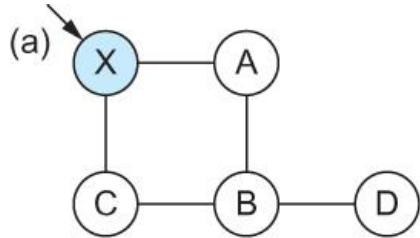
Topology Dissemination

- Each router creates a set of **link state packets** (LSPs)
 - Describing its links to neighbors
 - LSP contains
 - Router id, neighbor's id, and cost to its neighbor
- Copies of LSPs are distributed to all routers
 - Using **controlled flooding**
- Each router maintains a topology database
 - Database containing all LSPs



Link State

Reliable Flooding



Flooding of link-state packets. (a) LSP arrives at node X; (b) X floods LSP to A and C; (c) A and C flood LSP to B (but not X); (d) flooding is complete



Shortest Path Routing

- Dijkstra's Algorithm - Assume non-negative link weights
 - N : set of nodes in the graph
 - $l(i, j)$: the non-negative cost associated with the edge between nodes $i, j \in N$ and $l(i, j) = \infty$ if no edge connects i and j
 - Let $s \in N$ be the starting node which executes the algorithm to find shortest paths to all other nodes in N
 - Two variables used by the algorithm
 - M : set of nodes incorporated so far by the algorithm
 - $C(n)$: the cost of the path from s to each node n
 - The algorithm

```
M = {s}
For each n in N - {s}
    C(n) = l(s, n)
while ( N ≠ M)
    M = M ∪ {w} such that C(w) is the minimum
                                for all w in (N-M)
    For each n in (N-M)
        C(n) = MIN (C(n), C(w) + l(w, n))
```

Shortest Path Routing

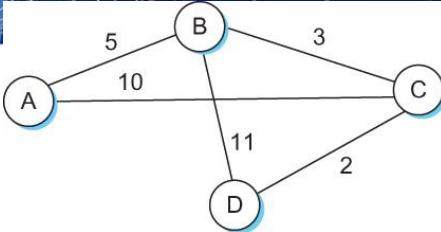
- In practice, each switch computes its routing table directly from the LSP's it has collected using a realization of Dijkstra's algorithm called the *forward search algorithm*
- Specifically each switch maintains two lists, known as **Tentative** and **Confirmed**
- Each of these lists contains a set of entries of the form (Destination, Cost, NextHop)



Shortest Path Routing

- The algorithm
 - Initialize the **Confirmed** list with an entry for myself; this entry has a cost of 0
 - For the node just added to the **Confirmed** list in the previous step, call it node **Next**, select its LSP
 - For each neighbor (Neighbor) of **Next**, calculate the cost (Cost) to reach this Neighbor as the sum of the cost from myself to Next and from Next to Neighbor
 - If Neighbor is currently on neither the **Confirmed** nor the **Tentative** list, then add (Neighbor, Cost, Nexthop) to the **Tentative** list, where Nexthop is the direction I go to reach Next
 - If Neighbor is currently on the **Tentative** list, and the Cost is less than the currently listed cost for the Neighbor, then replace the current entry with (Neighbor, Cost, Nexthop) where Nexthop is the direction I go to reach Next
 - If the **Tentative** list is empty, stop. Otherwise, pick the entry from the **Tentative** list with the lowest cost, move it to the **Confirmed** list, and return to Step 2.

Shortest Path Routing



Step	Confirmed	Tentative	Comments
1	(D,0,-)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,-)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,-) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,-) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,-) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,-) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,-) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.



Link State vs Distance Vector

- Tells everyone about neighbors
- Controlled flooding to exchange link state
- Dijkstra's algorithm
- Each router computes its own table
- May have oscillations
- Open Shortest Path First (OSPF)
- Tells neighbors about everyone
- Exchanges distance vectors with neighbors
- Bellman-Ford algorithm
- Each router's table is used by others
- May have routing loops
- Routing Information Protocol (RIP)



Link State vs. Distance Vector (cont'd)

Message complexity

- LS: $O(n^2 \cdot e)$ messages
 - n: number of nodes
 - e: number of edges
- DV: $O(d \cdot n \cdot k)$ messages
 - d: node's degree
 - k: number of rounds

Time complexity

- LS: $O(n \cdot \log n)$
- DV: $O(n)$

Convergence time

- LS: $O(1)$
- DV: $O(k)$

Robustness: what happens if router malfunctions?

- LS:
 - node can advertise incorrect *link* cost
 - each node computes only its *own* table
- DV:
 - node can advertise incorrect *path* cost
 - each node's table used by others; error propagate through network



Routing in the Real World

Our routing study thus far - idealization

- all routers identical
- network "flat"

How to do routing in the Internet

- scalability and policy issues

scale: with 200 million
destinations:

- can't store all dest's in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

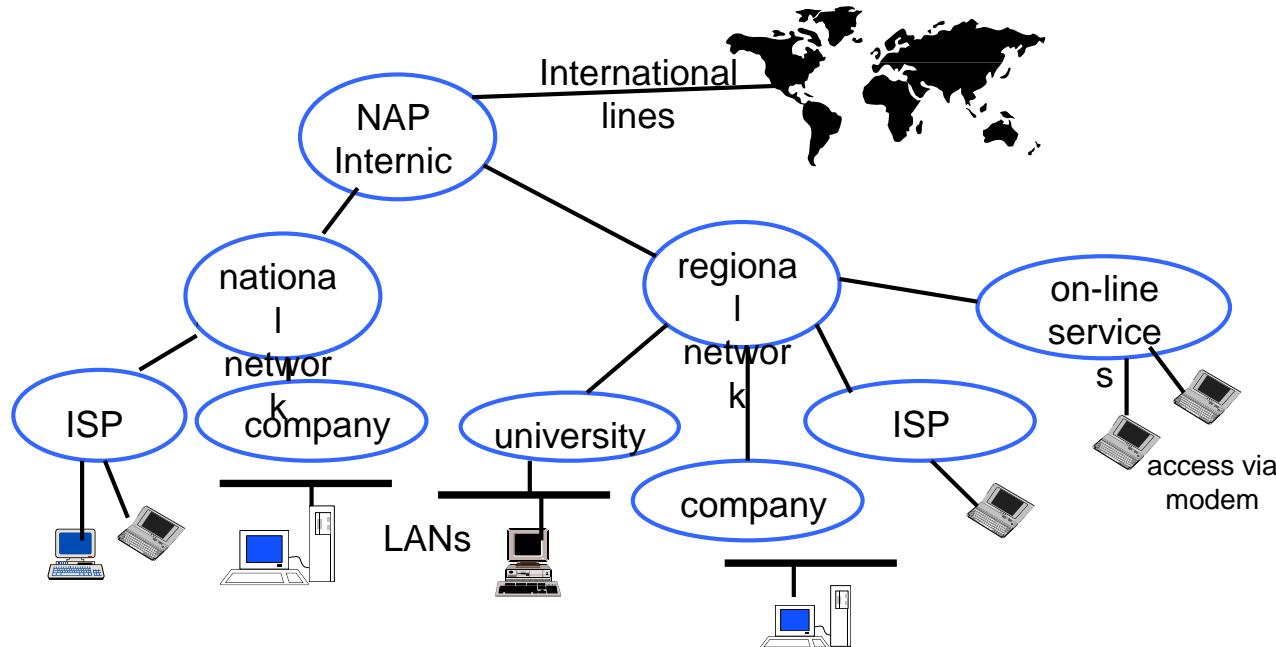


Routing in the Internet

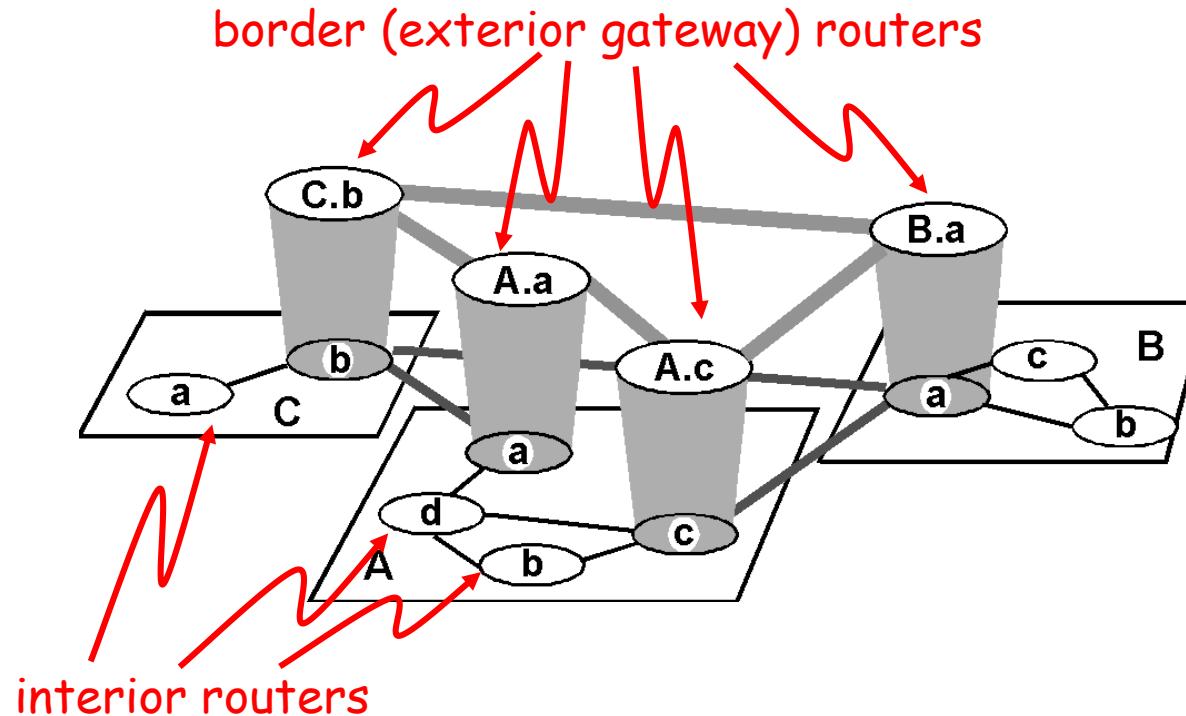
- The Global Internet consists of **Autonomous Systems (AS)** interconnected with each other hierarchically:
 - Stub AS: small corporation: one connection to other AS's
 - Multihomed AS: large corporation (no transit): multiple connections to other AS's
 - Transit AS: provider, hooking many AS's together
- Two-level routing:
 - Intra-AS: administrator responsible for choice of routing algorithm within network
 - Inter-AS: unique standard for inter-AS routing: BGP

Internet Architecture

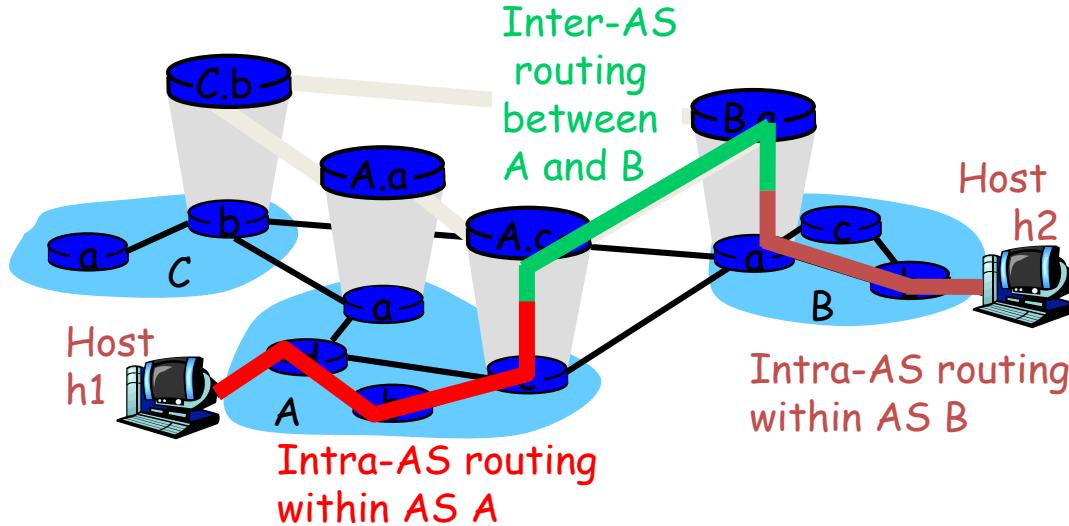
Internet: “networks of networks”!



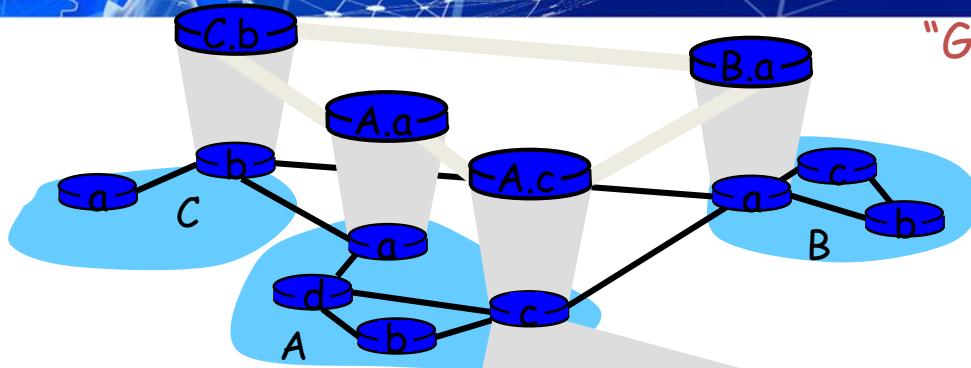
Internet autonomous system AS Hierarchy



Intra-AS vs. Inter-AS Routing



Intra-AS and Inter-AS Routing



"Gateways":

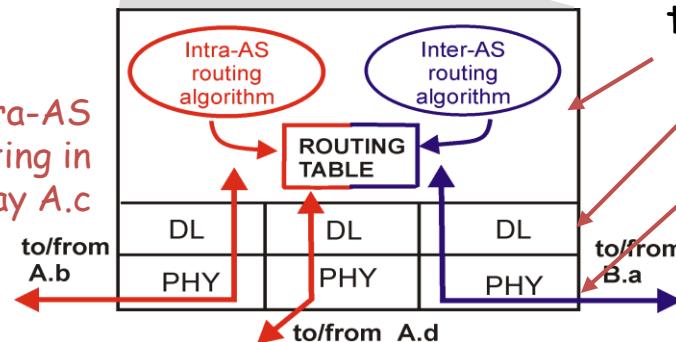
- perform inter-AS routing amongst themselves
- perform intra-AS routing with other routers in their AS

network layer

link layer

physical layer

inter-AS, intra-AS
routing in
gateway A.c





Intra-AS Routing

- Also known as **Interior Gateway Protocols (IGP)**
- Most common Intra-AS routing protocols:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IS-IS: Intermediate System to Intermediate System
(OSI Standard)
 - EIGRP: Extended Interior Gateway Routing Protocol
(Cisco proprietary)



Why Different Intra- and Inter-AS Routing?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, update traffic

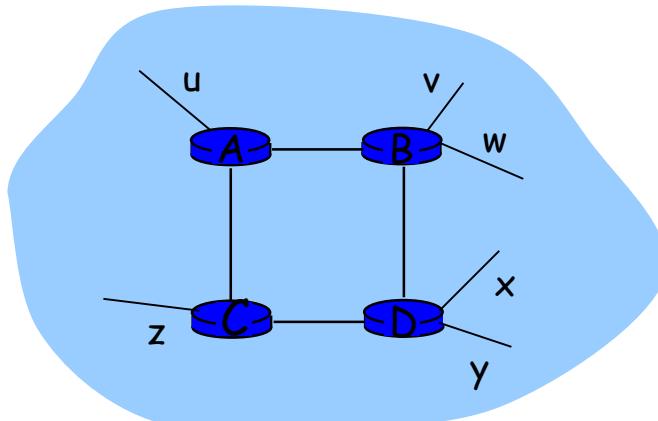
Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)

Number of hops from source router A to various subnets:



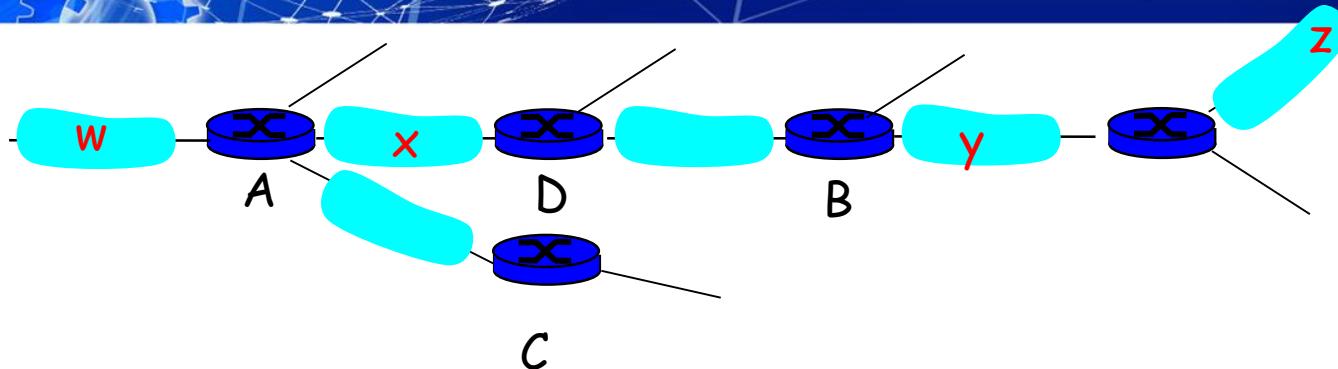
<u>destination</u>	<u>hops</u>
u	1
v	2
w	2
x	3
y	3
z	2



RIP advertisements

- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called **advertisement**)
- Each advertisement: list of up to 25 destination nets within AS

RIP: Example



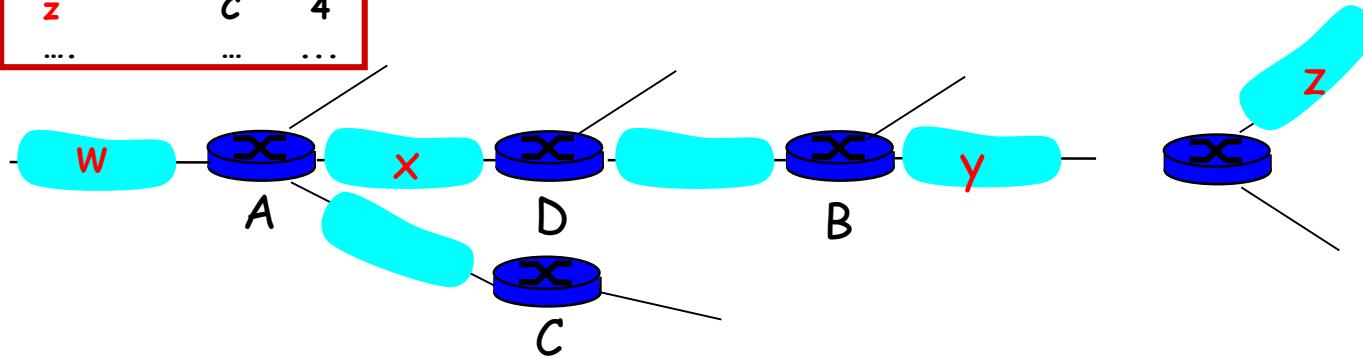
Destination Network	Next Router	Num. of hops to dest.
W	A	2
Y	B	2
Z	B	7
X	--	1
...

Routing table in D

RIP: Example

Dest	Next hops	
w	-	-
x	-	-
z	c	4
...

Advertisement
from A to D



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B A	5
x	--	1
...

Routing table in D



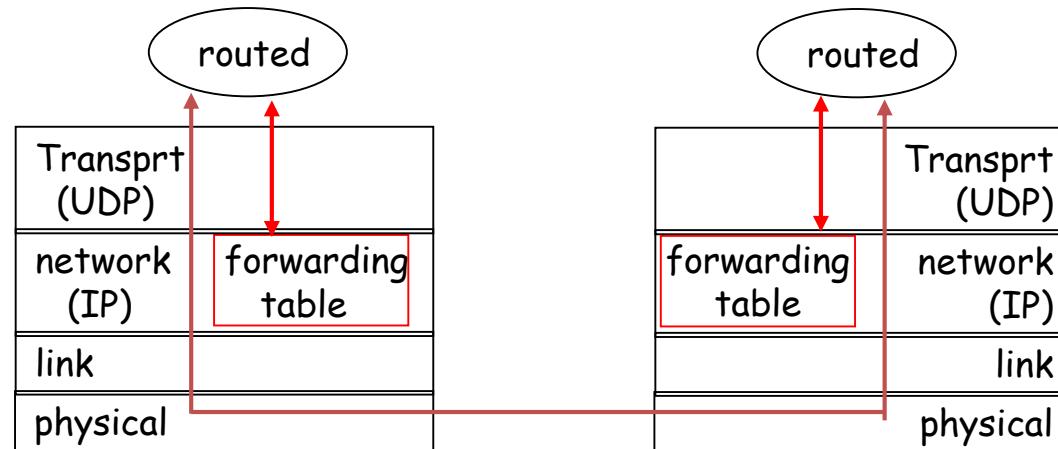
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec --> neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly propagates to entire net
- poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

RIP Table processing

- RIP routing tables managed by **application-level** process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated





OSPF (Open Shortest Path First)

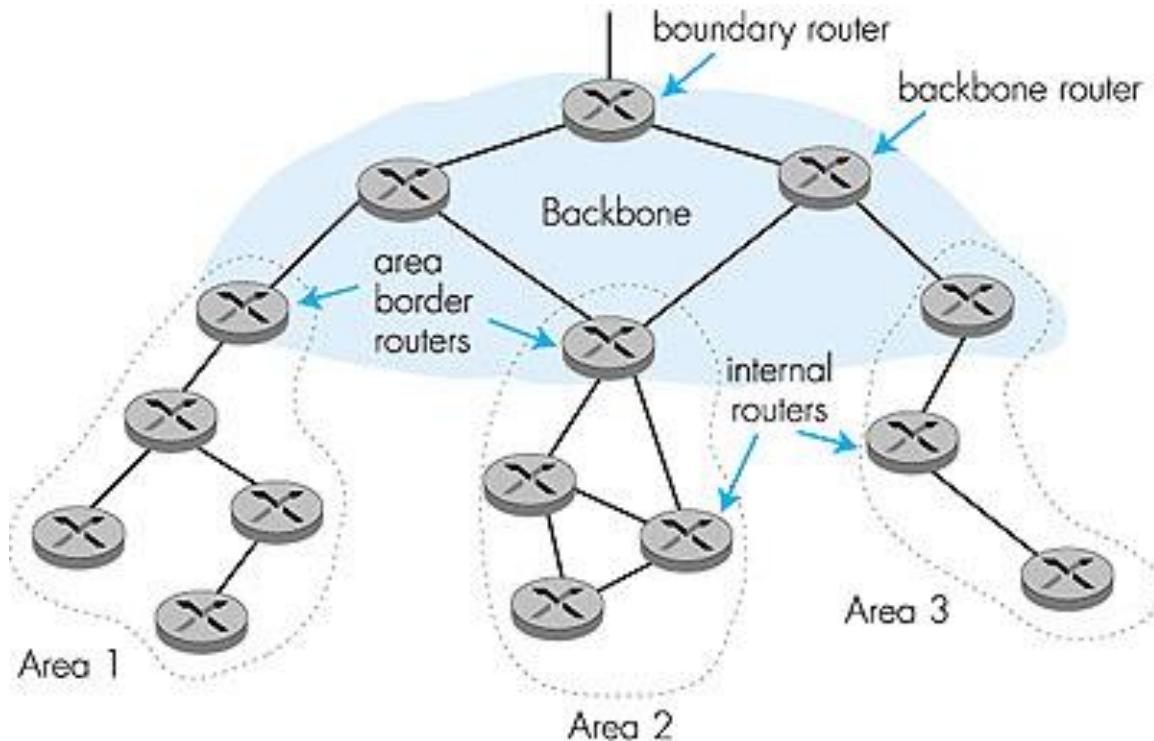
- “open”: publicly available
- Uses Link State algorithm
 - LS packet dissemination
 - Topology map at each node
 - Route computation using Dijkstra’s algorithm
- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to **entire AS** (via flooding)
 - Carried in OSPF messages directly over IP (rather than TCP or UDP)



OSPF “advanced” features (not in RIP)

- **Security:** all OSPF messages authenticated (to prevent malicious intrusion)
- **Multiple same-cost paths** allowed (only one path in RIP)
- For each link, multiple cost metrics for different **TOS** (e.g., satellite link cost set “low” for best effort; high for real time)
- Integrated uni- and **multicast** support:
 - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- **Hierarchical** OSPF in large domains.

Hierarchical OSPF





Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
 - Link-state advertisements only in area
 - each node has detailed area topology; only know direction (shortest path) to nets in other areas.
 - Communications between areas via backbone
- **Area border routers:** “summarize” distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers:** run OSPF routing limited to backbone.
- **Boundary routers:** connect to other AS's.

Inter-AS Routing in the Internet: BGP

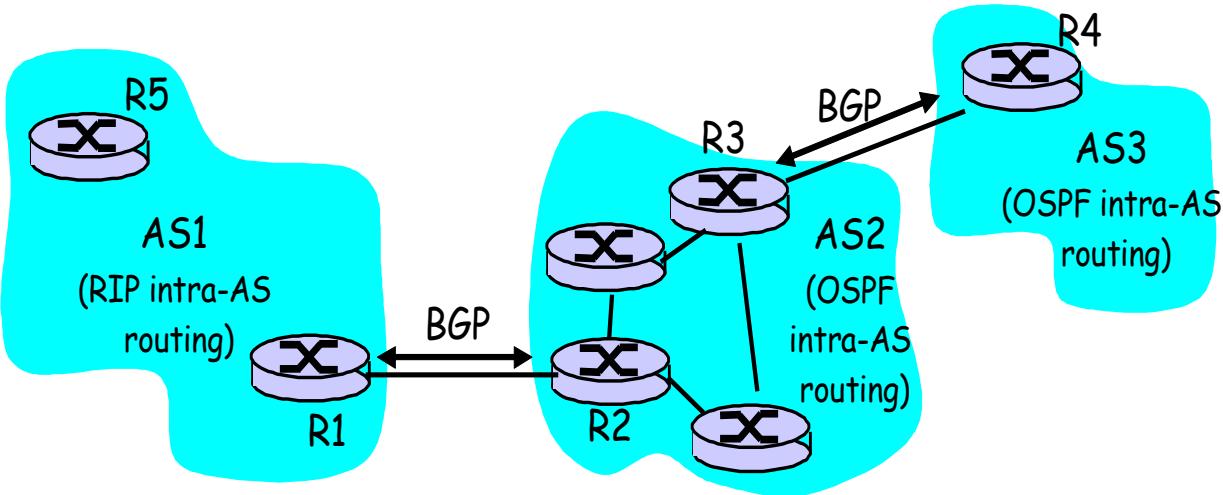


Figure 4.5.2-new2: BGP use for inter-domain routing

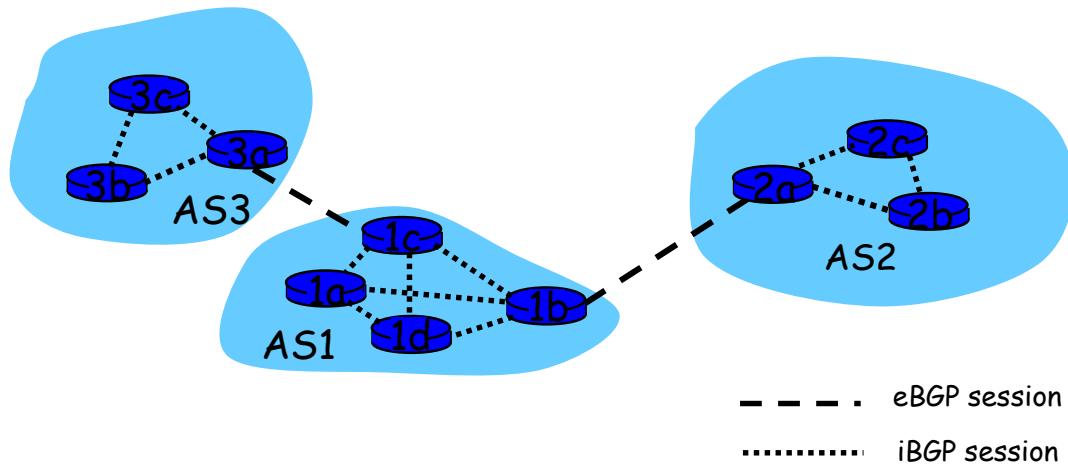


Internet inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** *the de facto standard*
- BGP provides each AS a means to:
 1. Obtain subnet reachability information from neighboring ASs.
 2. Propagate the reachability information to all routers internal to the AS.
 3. Determine “good” routes to subnets based on reachability information and policy.
- Allows a subnet to advertise its existence to rest of the Internet: *“I am here”*

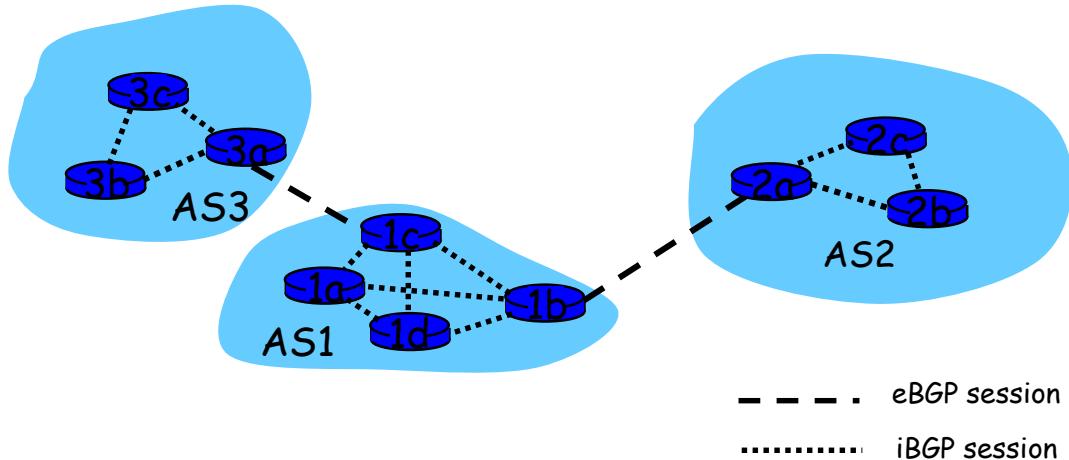
BGP basics

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: **BGP sessions**
- Note that BGP sessions do not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
 - AS2 can aggregate prefixes in its advertisement



Distributing reachability info

- With eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
- 1c can then use iBGP to distribute this new prefix reach info to all routers in AS1
- 1b can then re-advertise the new reach info to AS2 over the 1b-to-2a eBGP session
- When router learns about a new prefix, it creates an entry for the prefix in its forwarding table.





Path attributes & BGP routes

- When advertising a prefix, advert includes BGP attributes.
 - prefix + attributes = “route”
- Two important attributes:
 - **AS-PATH:** contains the ASs through which the advert for the prefix passed: AS 67 AS 17
 - **NEXT-HOP:** Indicates the specific internal-AS router to next-hop AS. (There may be multiple links from current AS to next-hop-AS.)
- When gateway router receives route advert, uses **import policy** to accept/decline.



BGP route selection

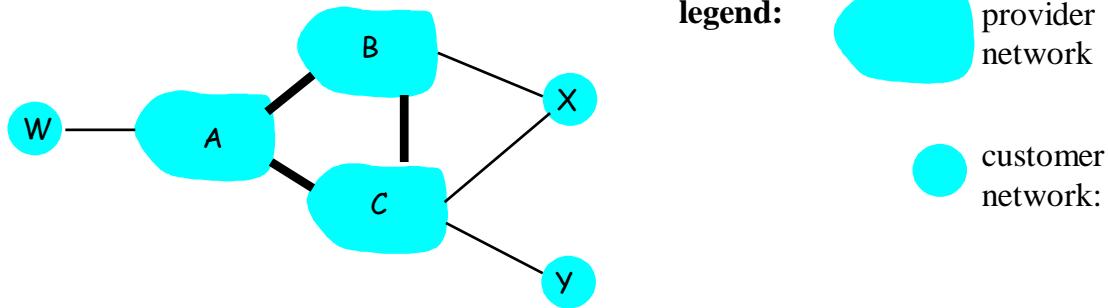
- Router may learn about more than 1 route to some prefix. Router must select route.
- Elimination rules:
 1. Local preference value attribute: policy decision
 2. Shortest AS-PATH
 3. Closest NEXT-HOP router: hot potato routing
 4. Additional criteria



BGP messages

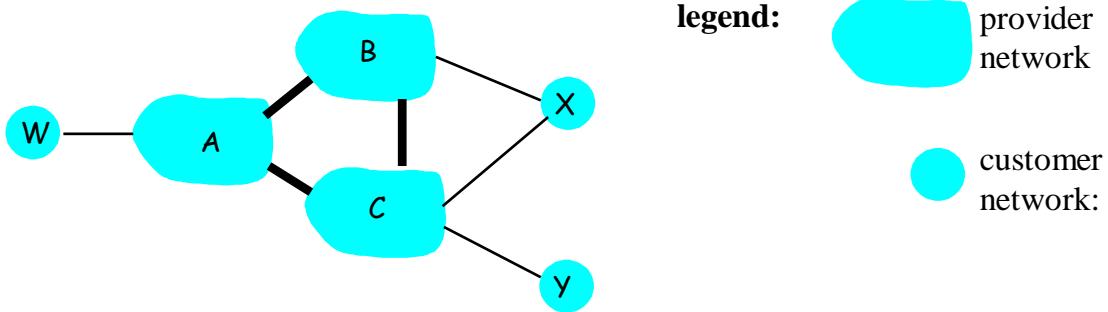
- BGP messages exchanged using TCP.
- BGP messages:
 - **OPEN**: opens TCP connection to peer and authenticates sender
 - **UPDATE**: advertises new path (or withdraws old)
 - **KEEPALIVE** keeps connection alive in absence of UPDATES; also ACKs OPEN request
 - **NOTIFICATION**: reports errors in previous msg; also used to close connection

BGP routing policy



- A,B,C are **provider networks**
- X,W,Y are customer (of provider networks)
- X is **dual-homed**: attached to two networks
 - X does not want to route from B via X to C
 - .. so X will not advertise to B a route to C

BGP routing policy (2)



- A advertises to B the path AW
- B advertises to X the path BAW
- Should B advertise to C the path BAW?
 - No way! B gets no "revenue" for routing CBAW since neither W nor C are B's customers
 - B wants to force C to route to w via A
 - B wants to route **only** to/from its customers!



Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

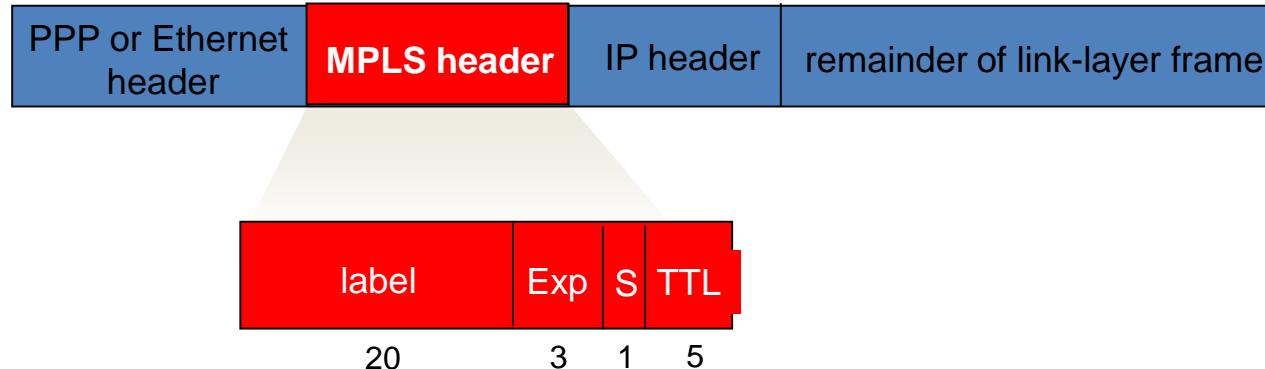
- hierarchical routing saves table size, reduced update traffic

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

Multi-Protocol Label Switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
 - borrowing ideas from Virtual Circuit (VC) approach
 - but IP datagram still keeps IP address!

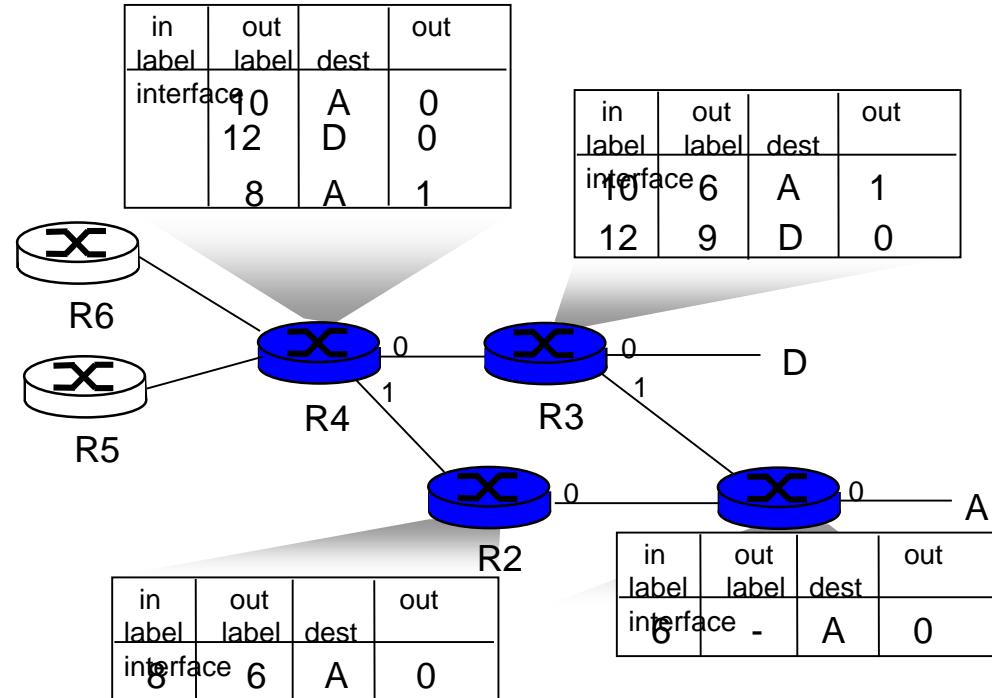




MPLS Capable Routers

- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don't inspect IP address)
 - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
 - RSVP-TE, LDP
 - forwarding possible along paths that IP alone would not allow (e.g., least cost path routing)
!!
 - use MPLS for traffic engineering
- must co-exist with IP-only routers

MPLS Forwarding Tables





Why Mobile IP?

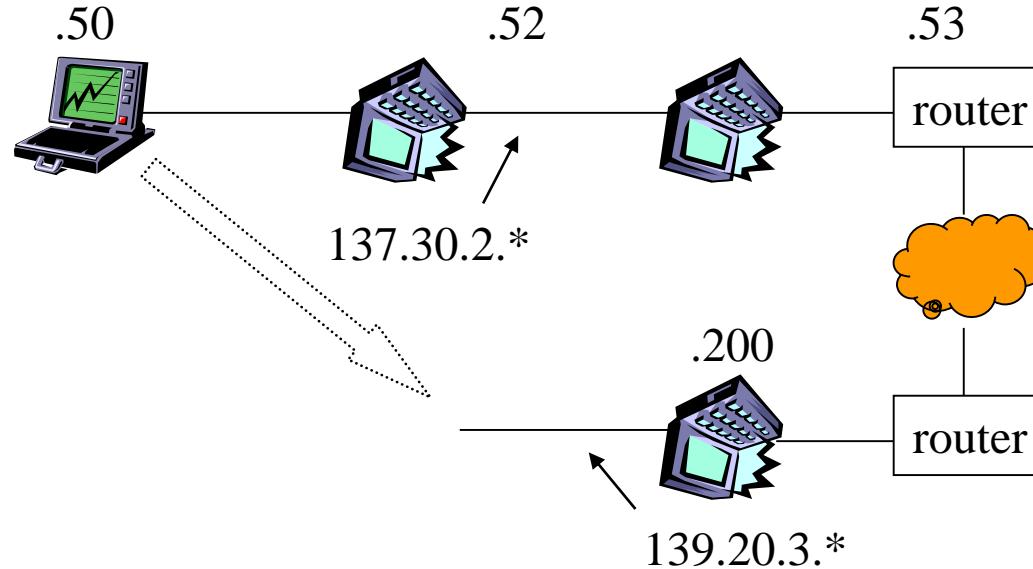
- Need a protocol which allows **network connectivity** across host movement
- Protocol to enable mobility must not require massive changes to router software, etc.
- Must be compatible with large installed base of IPv4 networks/hosts
- Confine changes to mobile hosts and a few support hosts which enable mobility



Internet Protocol (IP)

- Network layer, "best-effort" packet delivery
- Supports UDP and TCP (transport layer protocols)
- IP host addresses consist of two parts
 - **network id + host id**
- By design, IP host address is tied to home network address
 - Hosts are assumed to be wired, immobile
 - Intermediate routers look only at network address
 - Mobility without a change in IP address results in un-route-able packets

IP Routing Breaks Under Mobility



Why this hierarchical approach? Answer: **Scalability!**
Millions of network addresses, billions of hosts!



Mobile IP: Basics

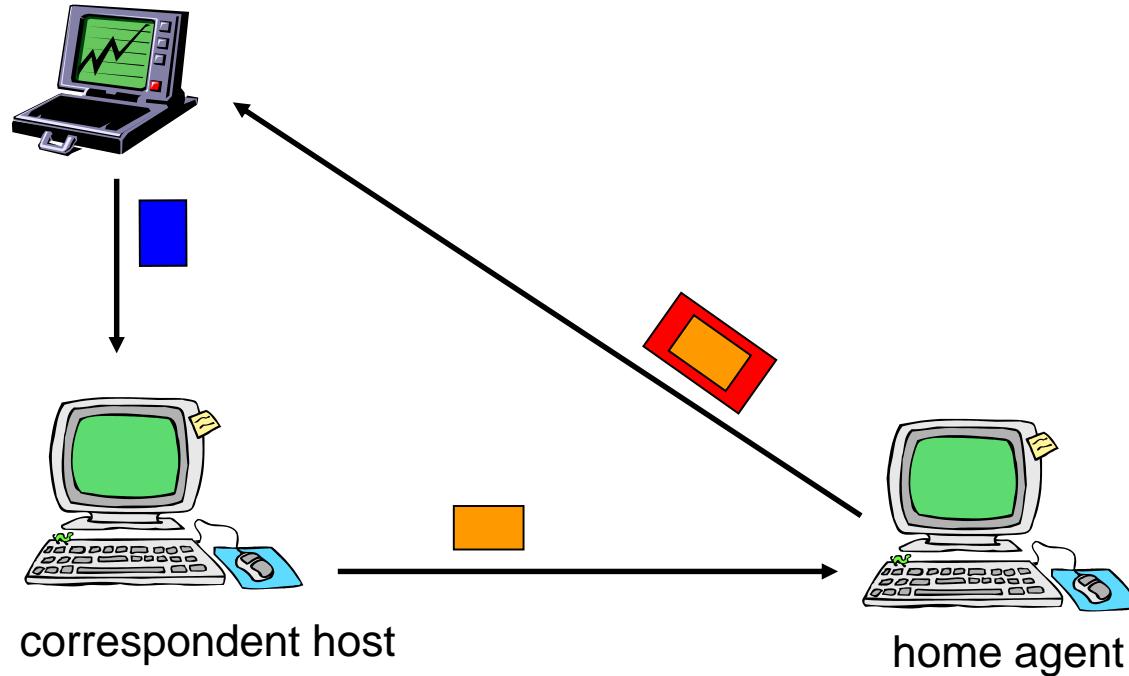
- Proposed by IETF (Internet Engineering Task Force)
 - Standards development body for the Internet
- Mobile IP allows a mobile host to move about without changing its ***permanent*** IP address
- Each mobile host has a ***home agent*** on its ***home network***
- Mobile host establishes a ***care-of*** address when it's away from home



Mobile IP: Basics, Cont.

- **Correspondent host** is a host that wants to send packets to the mobile host
- Correspondent host sends packets to the mobile host's IP permanent address
- These packets are routed to the mobile host's home network
- Home agent forwards IP packets for mobile host to current care-of address
- Mobile host sends packets directly to correspondent, using permanent home IP as source IP

Mobile IP: Basics, Cont.



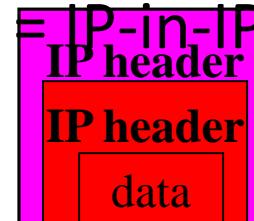


Mobile IP: Care-of Addresses

- Whenever a mobile host connects to a remote network, two choices:
 - care-of can be the address of a *foreign agent* on the remote network
 - foreign agent delivers packets forwarded from home agent to mobile host
 - care-of can be a temporary, foreign IP address obtained through, e.g., DHCP
 - home agent *tunnels* packets directly to the temporary IP address
- Regardless, care-of address must be *registered* with home agent

IP-in-IP Tunneling

- Packet to be forwarded is encapsulated in a new IP packet
- In the new header:
 - Destination = care-of-address
 - Source = address of home agent
 - Protocol number = IP-in-IP

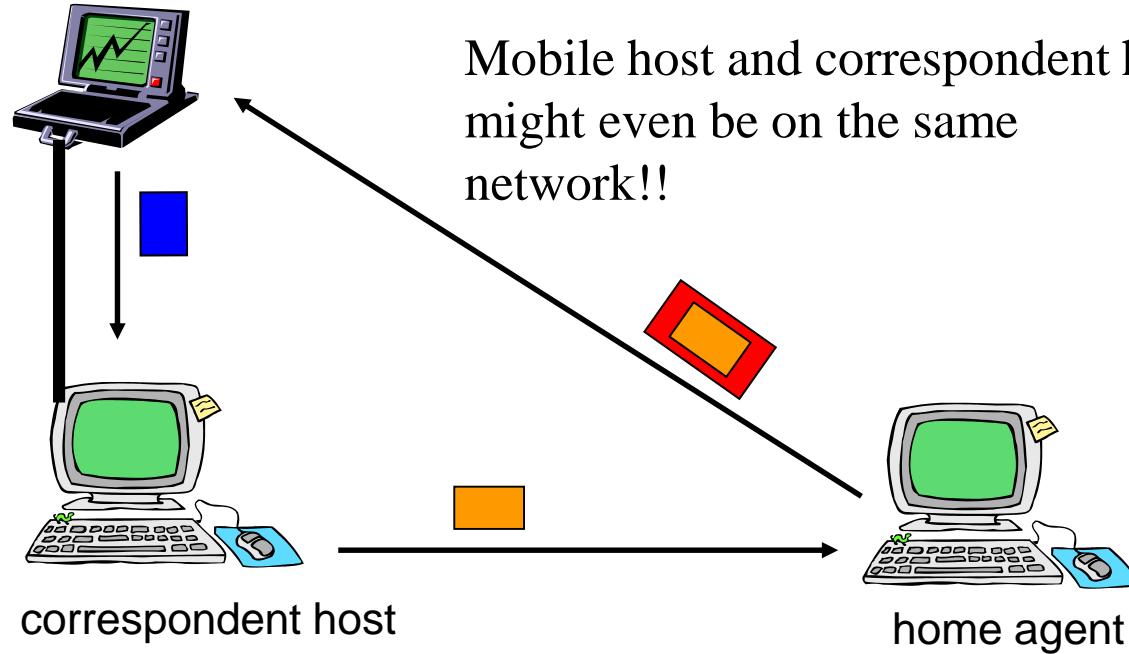




At the Other End...

- Depending on type of care-of address:
 - Foreign agent or
 - Mobile host
- ... strips outer IP header of tunneled packet, which is then fed to the mobile host
- Aside: Any thoughts on advantages of foreign agent vs. co-located (foreign IP) address?

Routing Inefficiency

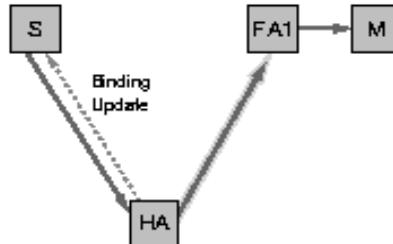




Route Optimizations

- Possible Solution:
 - Home agent sends current care-of address to correspondent host
 - Correspondent host caches care-of address
 - Future packets tunneled directly to care-of address
- But!
 - An instance of the cache consistency problem arises...
 - Cached care-of address becomes stale when the mobile host moves
 - Potential security issues with providing care-of address to correspondent

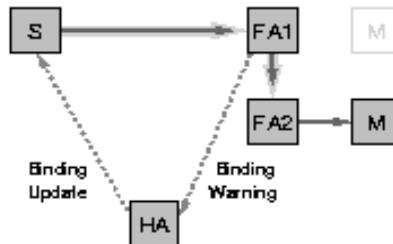
Possible Route Optimization



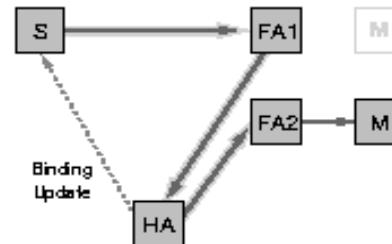
(a) Sending the first packet
to a mobile host



(b) Sending subsequent packets
to a mobile host



(c) Sending the first packet after
a mobile host moves



(d) Tunneling the packet in case the
cache entry has been dropped