Kolokvijum-2

MREŽNO RAČUNARSTVO

3- GODINA

William A. Shay

UVOD

William A. Shay

- This and the other PowerPoint files represent an outline only. Each student is responsible for additional details specified in the book or in class

Introduction: Chapters 1 and 2

- First Data Networks:
  - France in the late 1700s (Holzmann and Pehrson, "The First Data Networks", *Scientific American*, January 1994, pp. 124-129)

Network Applications

- Email
- Web
- Business
- video conferencing
- file sharing
- Gaming
- streaming (Internet radio/videos)
- TV
- Phones
- Just about everything!!

How do users communicate with a Computer?

- Text-based.
  - Typing
  - Mouse use
  - Card scanners
- Probably still the most common.
- All require some physical actions.

Voice communication

- Well documented
- Voice recognition software:
  - Can enter data through voice commands
- Becoming much more common.

Can a computer read your mind?

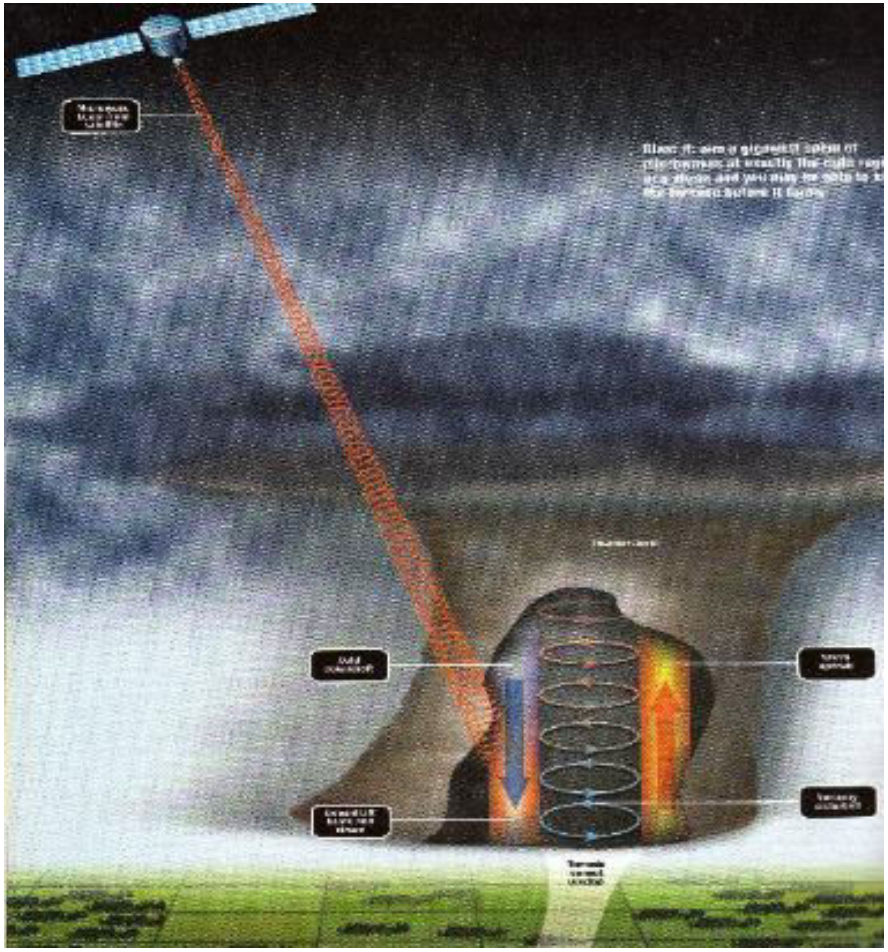
- [http://www.joystiq.com/2006/03/10/mind-controlled-pong-featured-at-cebit/]
- [http://www.cbsnews.com/video/watch/?id=4707958n]
- [http://science.discovery.com/videos/kapow-superhero-science-mind-over-matter.html]

Basic Principle

- EEG (electroencephalogram)
- Measure and store electrical activity of the brain.
- Computer simply learns what patterns correspond to what thoughts (on a very primitive level, of course)

Can a computer network do something about the weather?

- http://www.nssl.noaa.gov/primer/tornado/tor_predicting.html
- http://ascelibrary.org/proceedings/resource/2/ascecp/204/40479/129_1?isAuthorized=no



- Telepresence
  - [http://whatis.techtarget.com/definition/0,,sid9_gci1150556,00.html]
  - [https://www.cisco.com/en/US/netsol/ns669/networking_solutions_solution_segment_home.html],

- Physical communications: (Quantum Teleportation)
- [http://www.research.ibm.com/quantuminfo/teleportation/]
- [http://news.nationalgeographic.com/news/2004/08/0818_040818_teleportation.html]

Decoding the jargon

- Technical Specification: (for example) D-Link router.
- What do all those acronyms mean?

Standards and organizations:

- What is a standard?
- agreed upon way of doing something.

American National Standards Institute (ANSI)

- [http://www.ansi.org/membership/overview/overview.aspx?menuid=2]
- [http://www.ansi.org/standards_activities/overview/overview.aspx?menuid=3].
- ASCII American Standard Code for Information Interchange.
- Standards Store.

International Electrotechnical Commission (IEC)

- nongovernmental agency devising standards for data processing and interconnections and safety in office equipment [http://www.iec.ch/]:
- Was involved in the development of the Joint Photographic Experts Group (JPEG) [http://www.jpeg.org/]

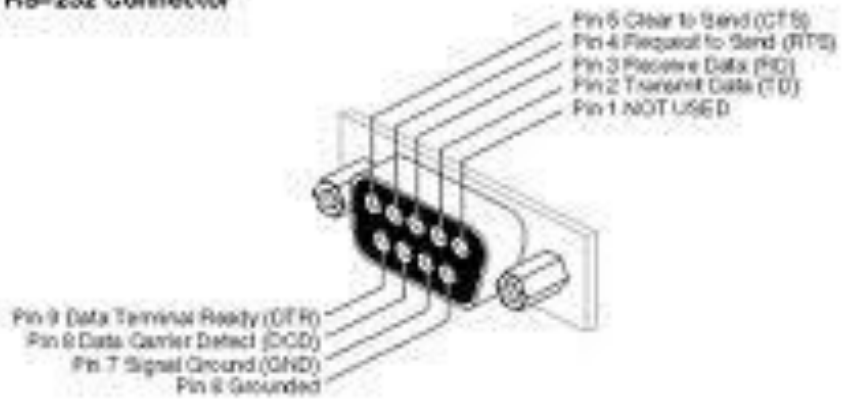
International Telecommunications Union (ITU)

- formerly called Comité Consultatif International de Télégraphique et Téléphonique (CCITT): agency of the United Nation.
- V series (modems) and X series (data networks)
- email, directory services, security certificates, packet switched interfaces.
- [http://www.itu.int/ITU-/publications/recs.html]

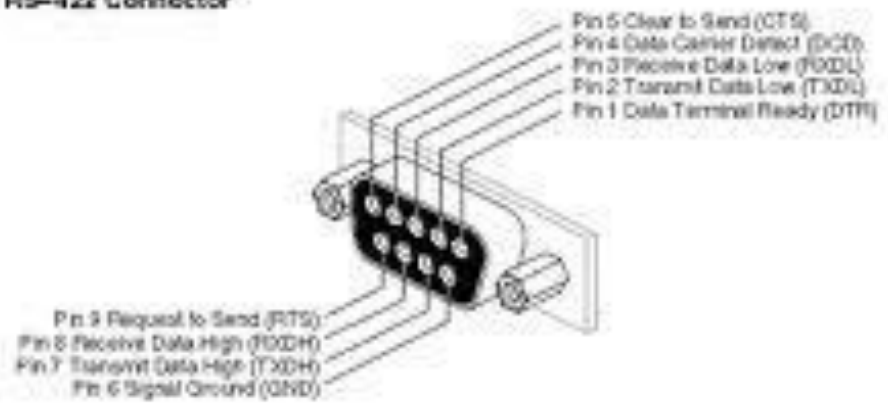
Electronic Industries Alliance (EIA):

- electronics firms and manufacturers of telecommunications equipment.
- RS-232 (also called EIA-232)
- Once a common way to connect devices to a computer.

RS–232 Connector



RS–422 Connector



Telecommunications Industry Association (TIA):

- providers of communications and information technology products and services for the global marketplace.
- optical cable, wires, and connectors used in local area networks

Internet Engineering Task Force (IETF):

- international community whose members include network designers, vendors, and researchers.
- Internet standards
- RFCs

Institute of Electrical and Electronic Engineers (IEEE):

- consists of computing and engineering professional.
- Project 802 LAN standards.

International Organization for Standardization (ISO):

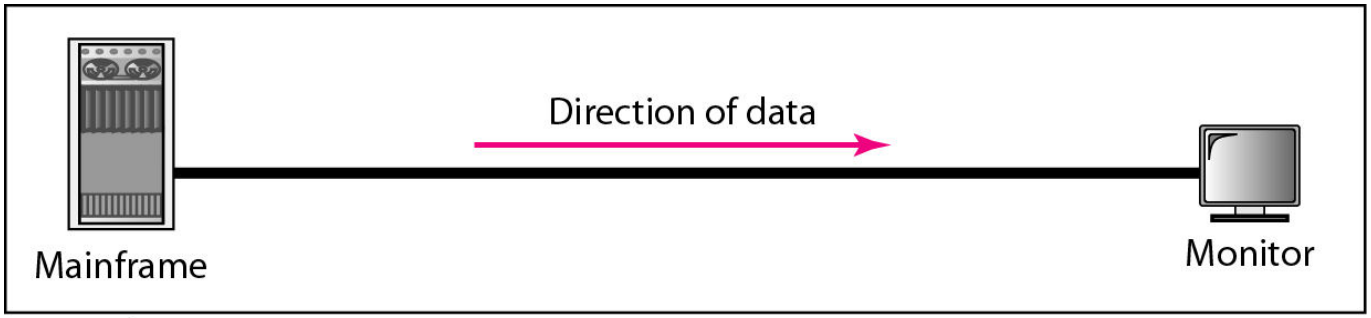
- worldwide organization consisting of standards bodies from many countries, such as ANSI from the United State.
- Open Systems Interconnect (OSI), a seven-layer organization of communication protocols.
- Designed to allow any two computers to communicate regardless of differences as long as there is an communications infrastructure.
- Predates the Internet.

National Institute of Standards and Technology (NIST) (formerly NBS):

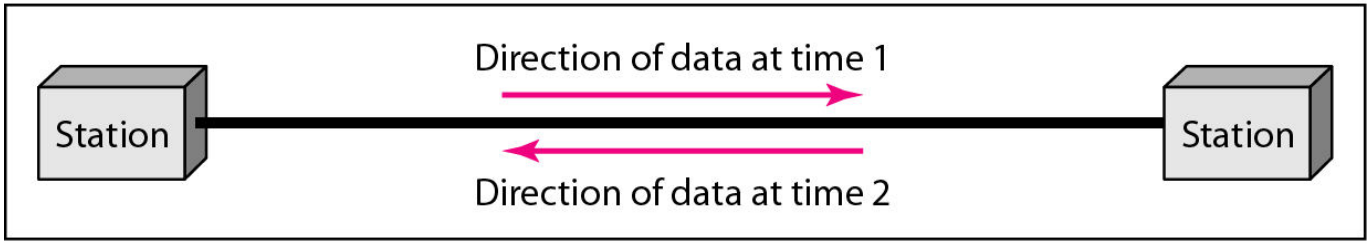
- Agency of the United States Department of Commerce.
- length, temperature, radioactivity, and radio frequencies,
- Data Encryption Standard (DES).
- Also AES (Advanced Encryption Standard).

Some terms:

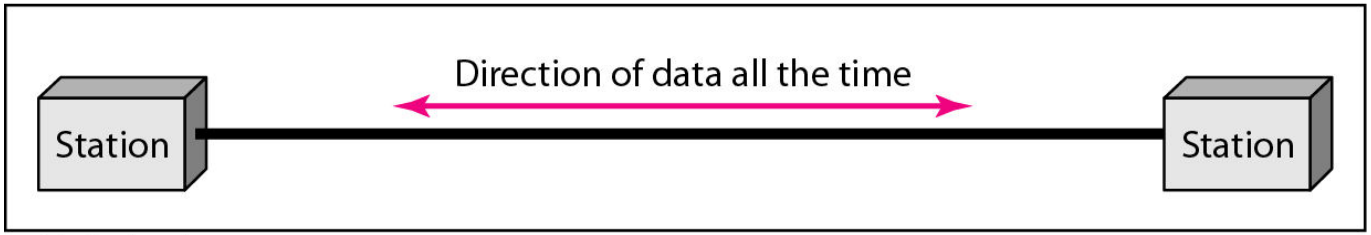
(Fig 1.2 on page 6)



a. Simplex



b. Half-duplex



c. Full-duplex

Figure 1.6 ***A star topology connecting four stations***

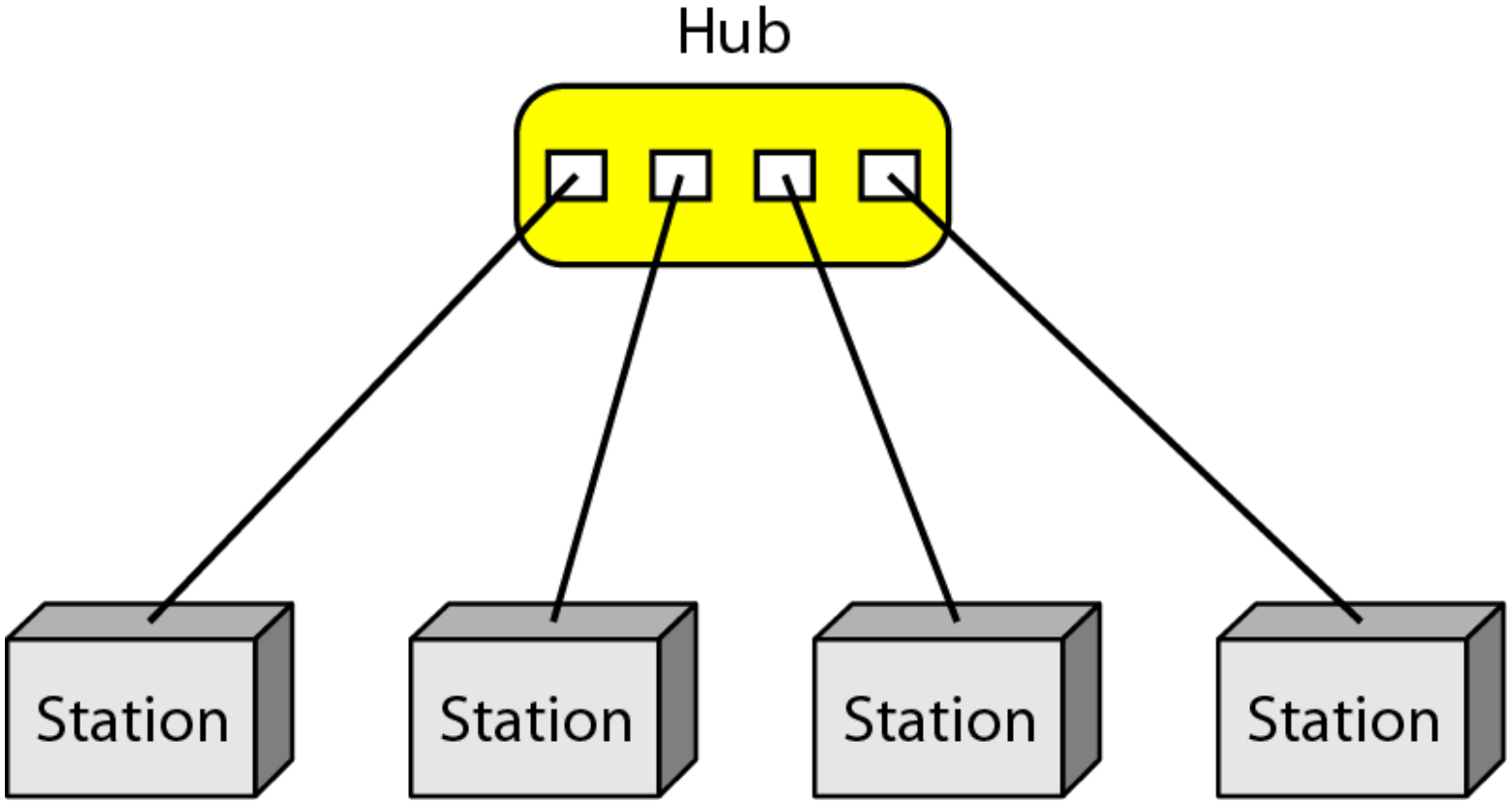


Figure 1.7 ***A bus topology connecting three stations***

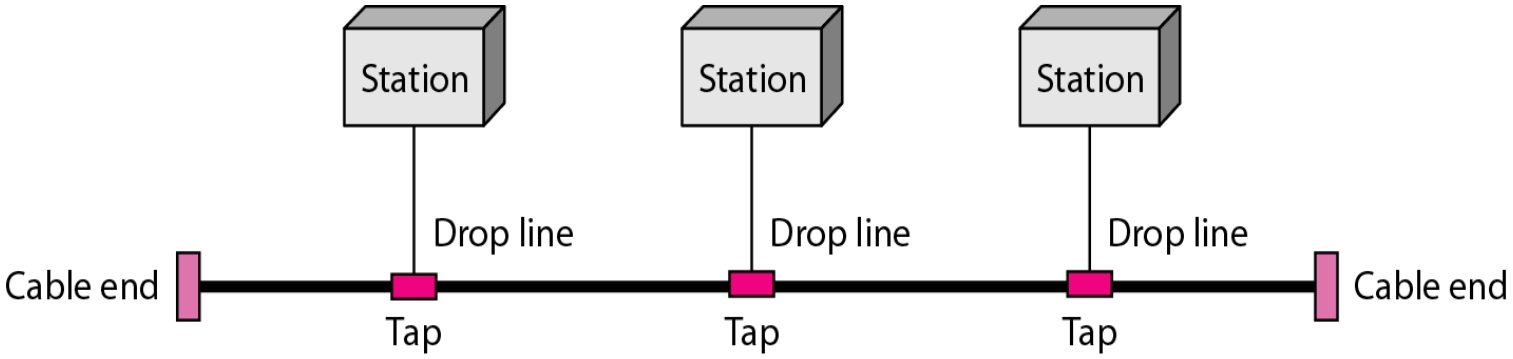


Figure 1.8 *A ring topology connecting six stations*

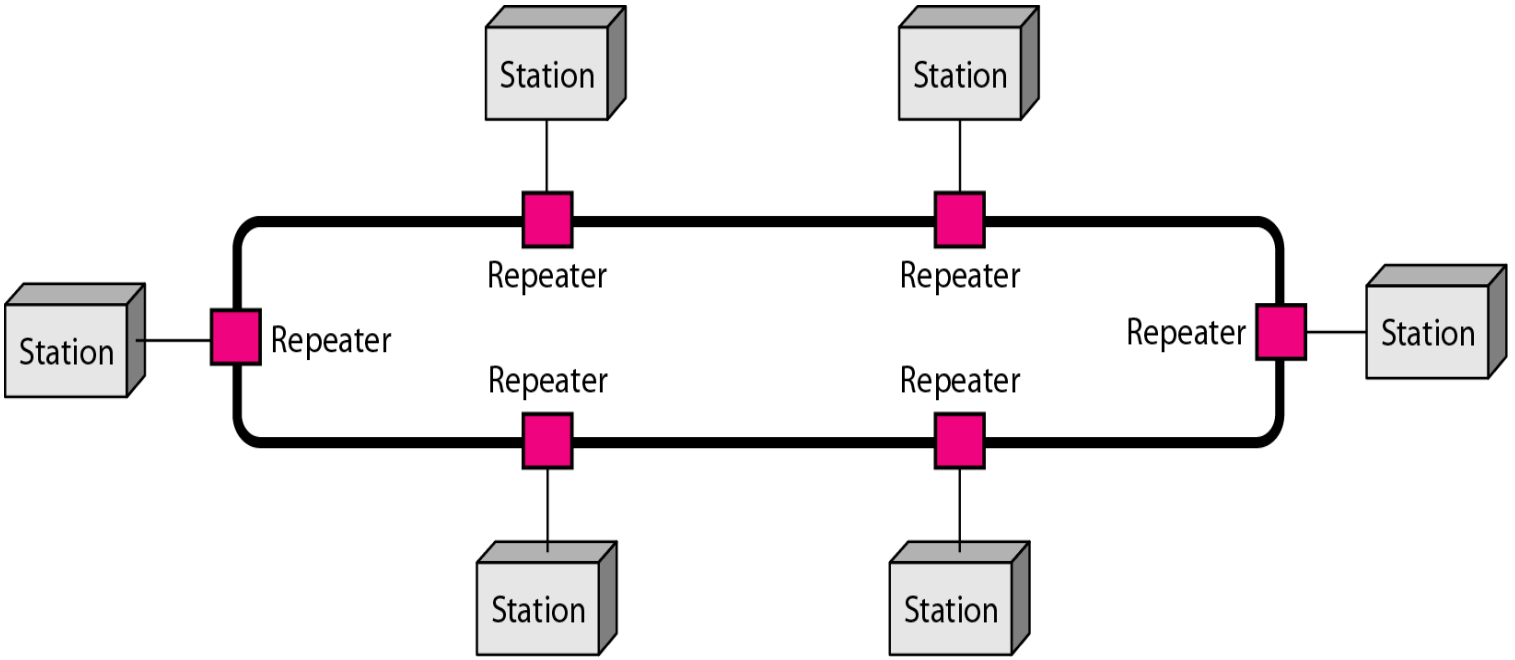
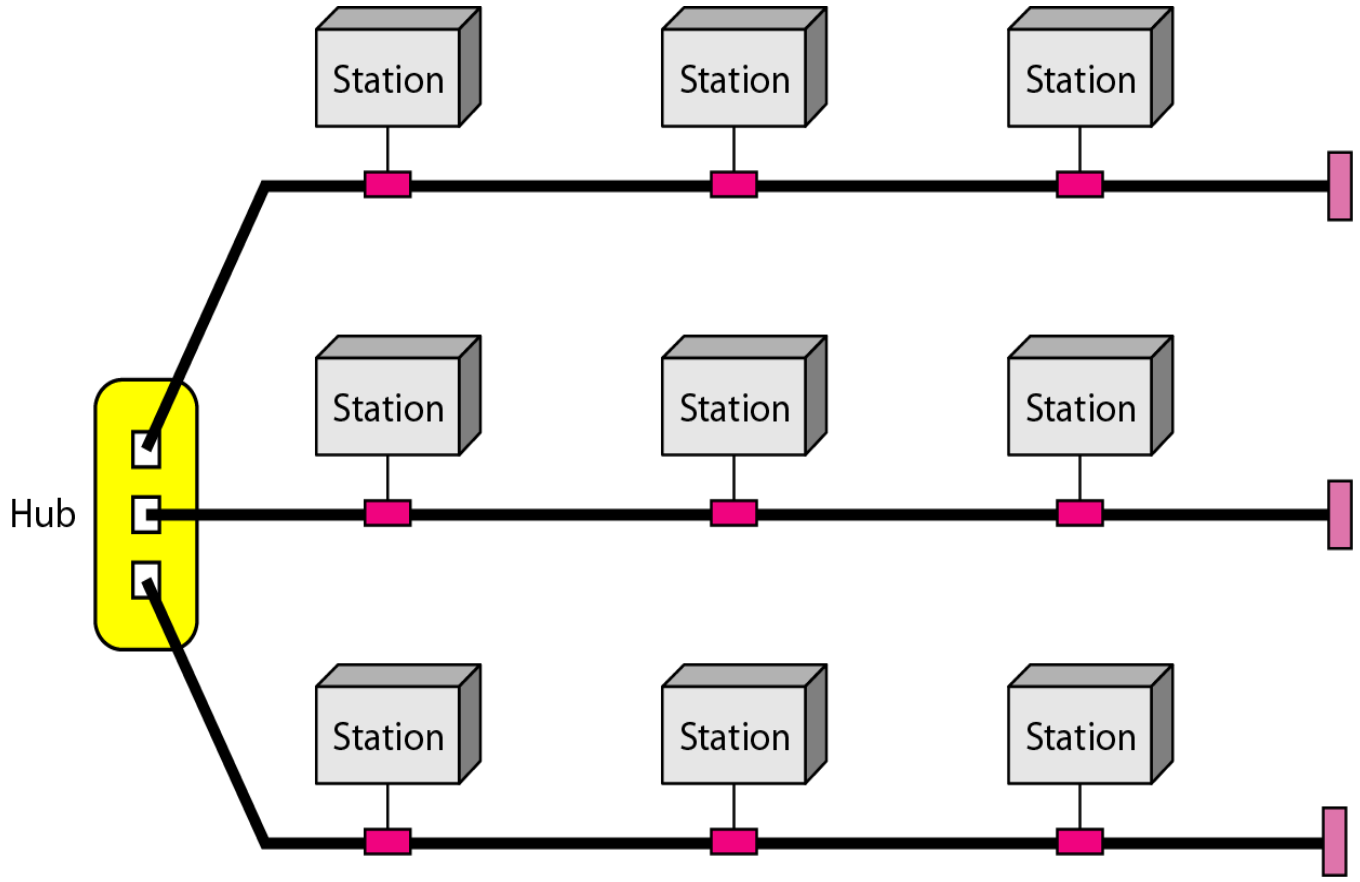


Figure 1.9 ***A hybrid topology: a star backbone with three bus networks***



More in the book

Layering protocols

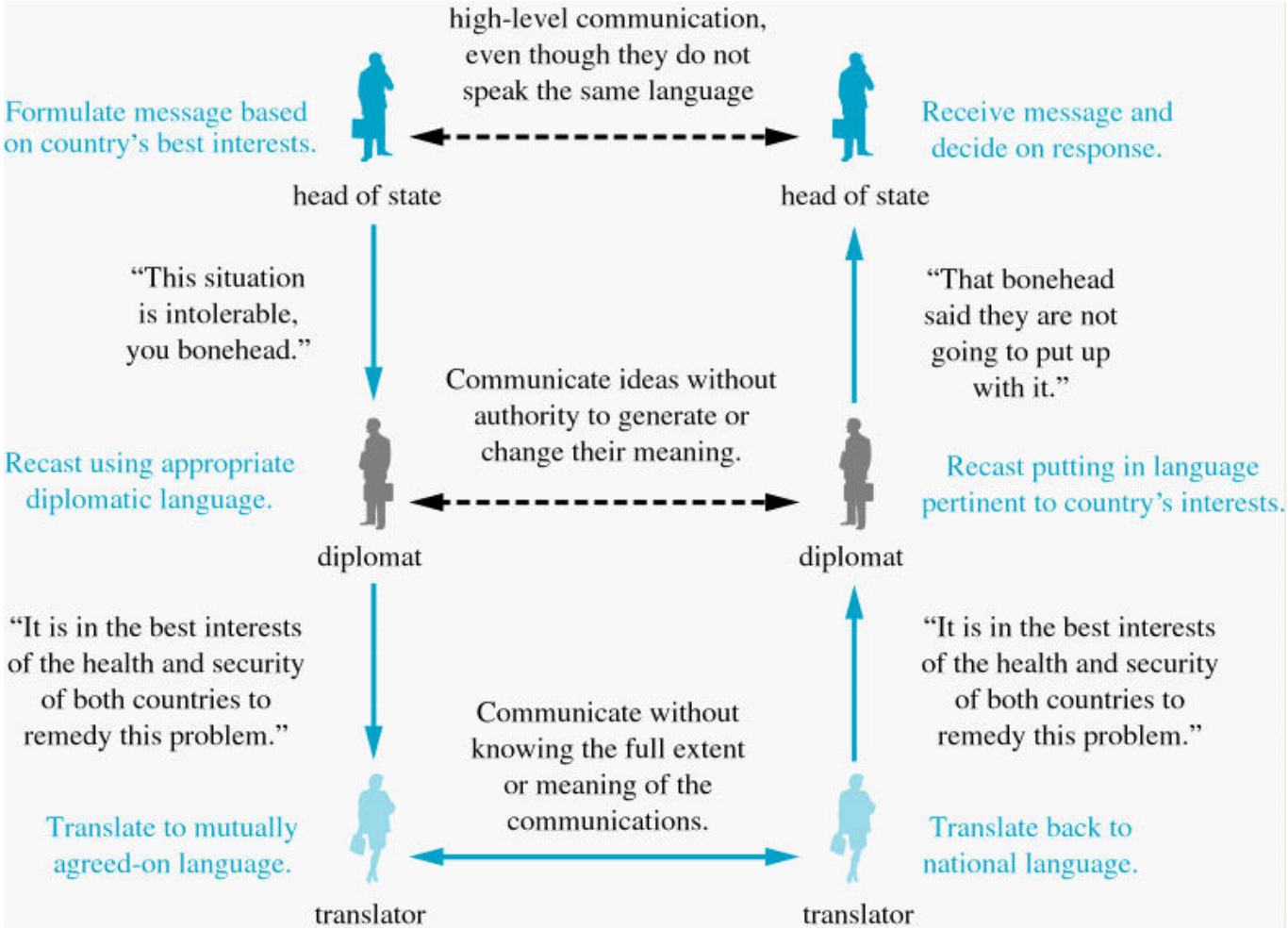


Fig 2.3: 7 layer OSI (Open Systems Interconnect) model. ISO standard.

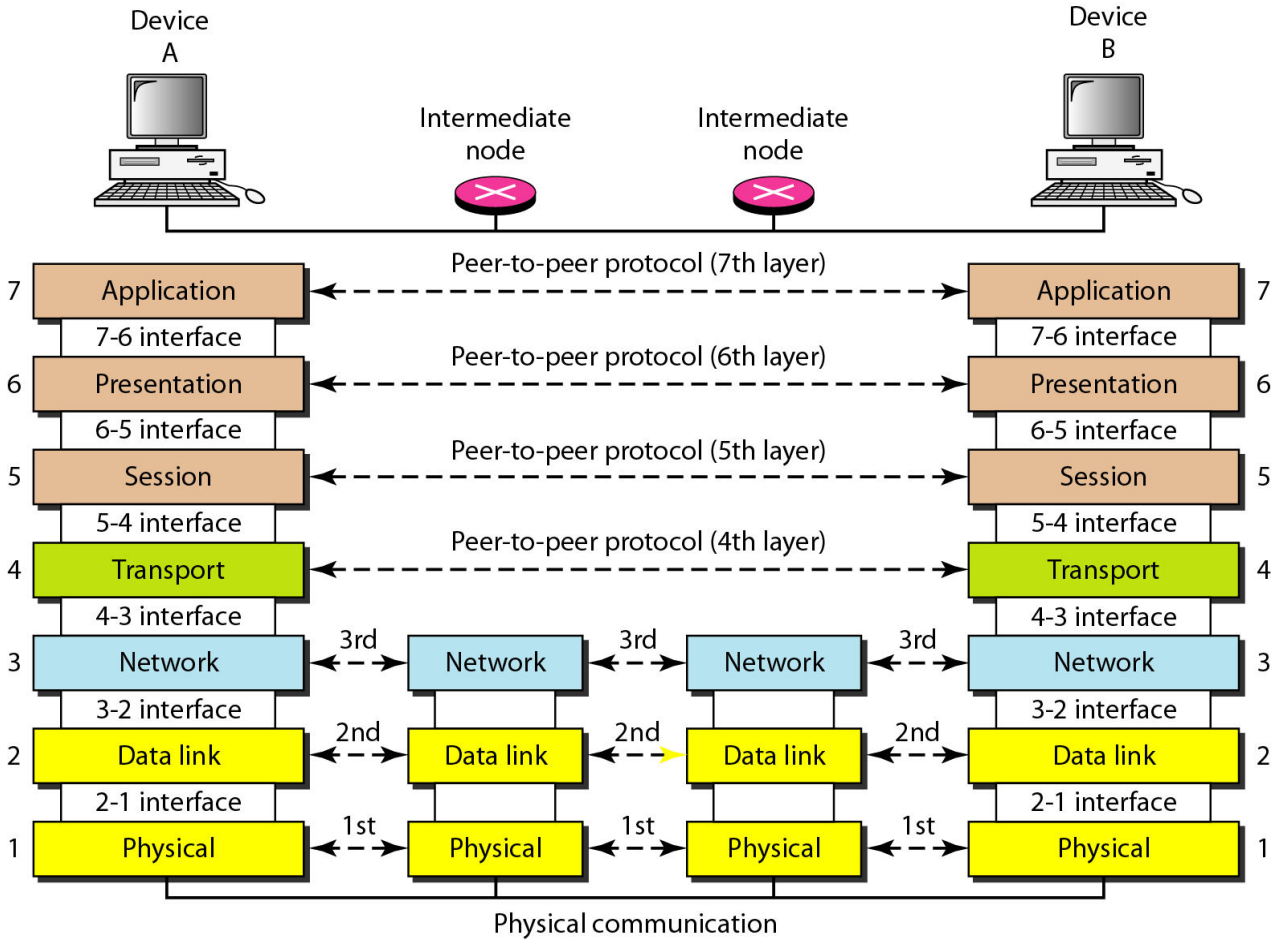
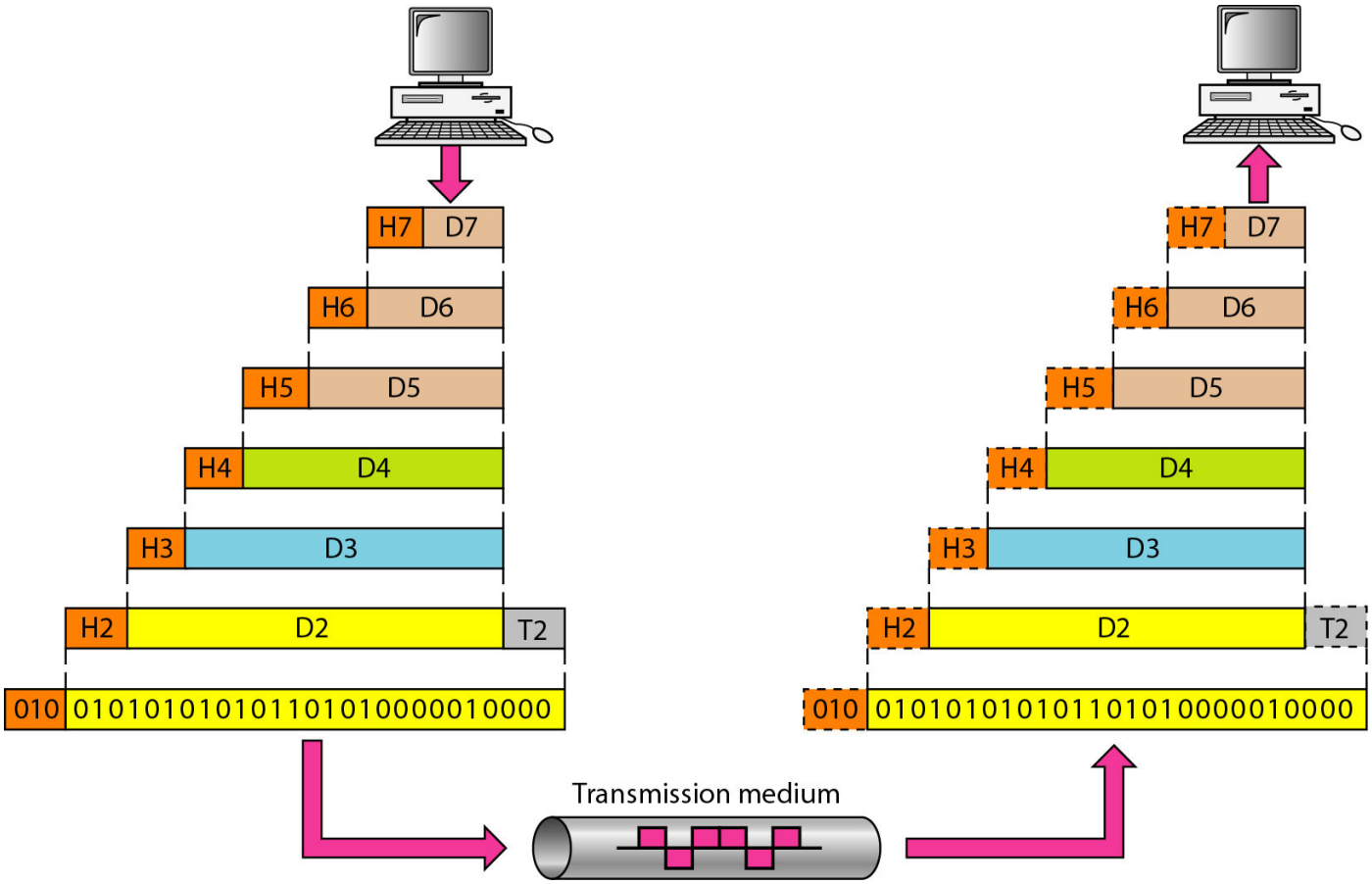


Figure 2.4 *An exchange using the OSI model*



Physical Layer:

- Transmit bits via electrical, optical, or electromagnetic signals
- electrical, optical, and transmission characteristics.
- Digital & analog signals, characteristics, wavelengths, voltage levels, bit rates, etc
- Need a degree in physics and/or mathematics to understand all of this stuff

Data Link Layer:

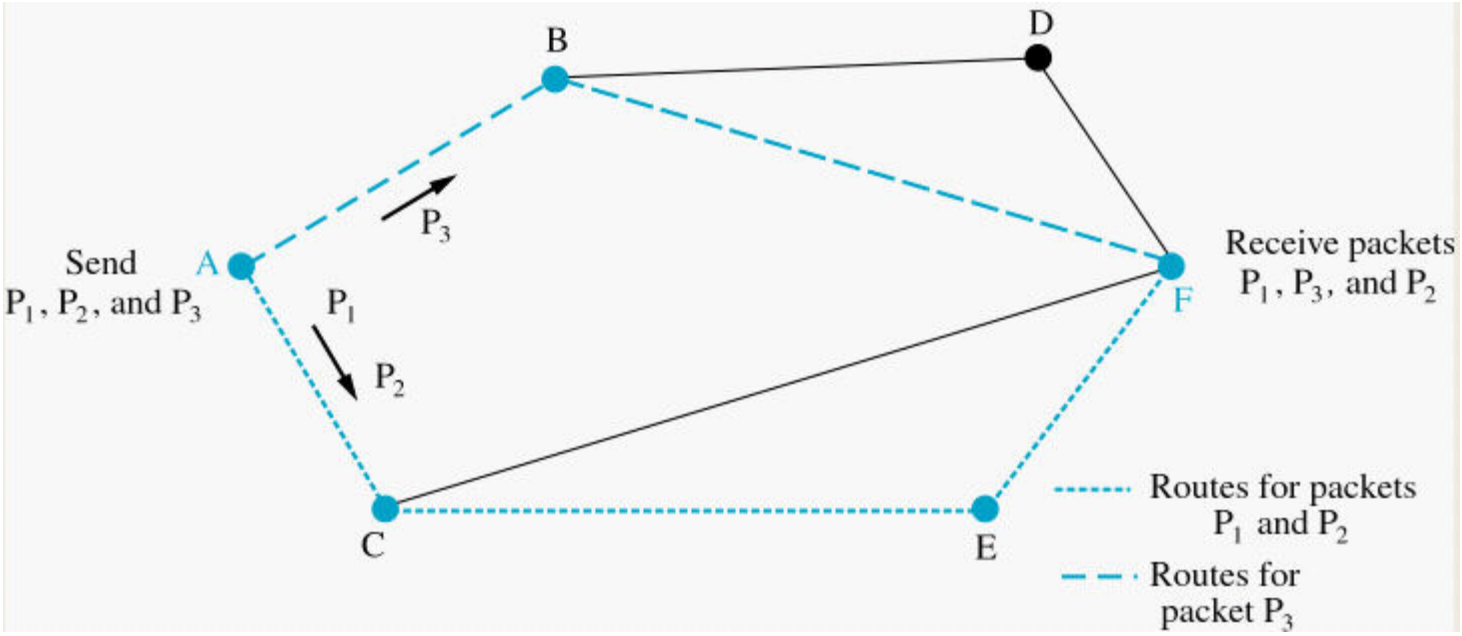
- Transmit a frame
  - unit of layer 2 information) from one device to the next (usually within a local area network or LAN)
- Contention
  - how the devices access the media (tokens, CSMA-Carrier Sense Multiple Access)
- error detection
- flow control
  - controls how much information is sent at a time

Network Layer:

- Delivery of packets
  - units of layer 3 information across a wide area network (Internet)
- routing
- network operations,
- IP (Internet Protocol) of TCP/IP.

Packet switched network

- General term to describe operation of most networks.
- Divide data into one or more *packets*
- Send each packet to its intended destination
- Each packet may routed separately

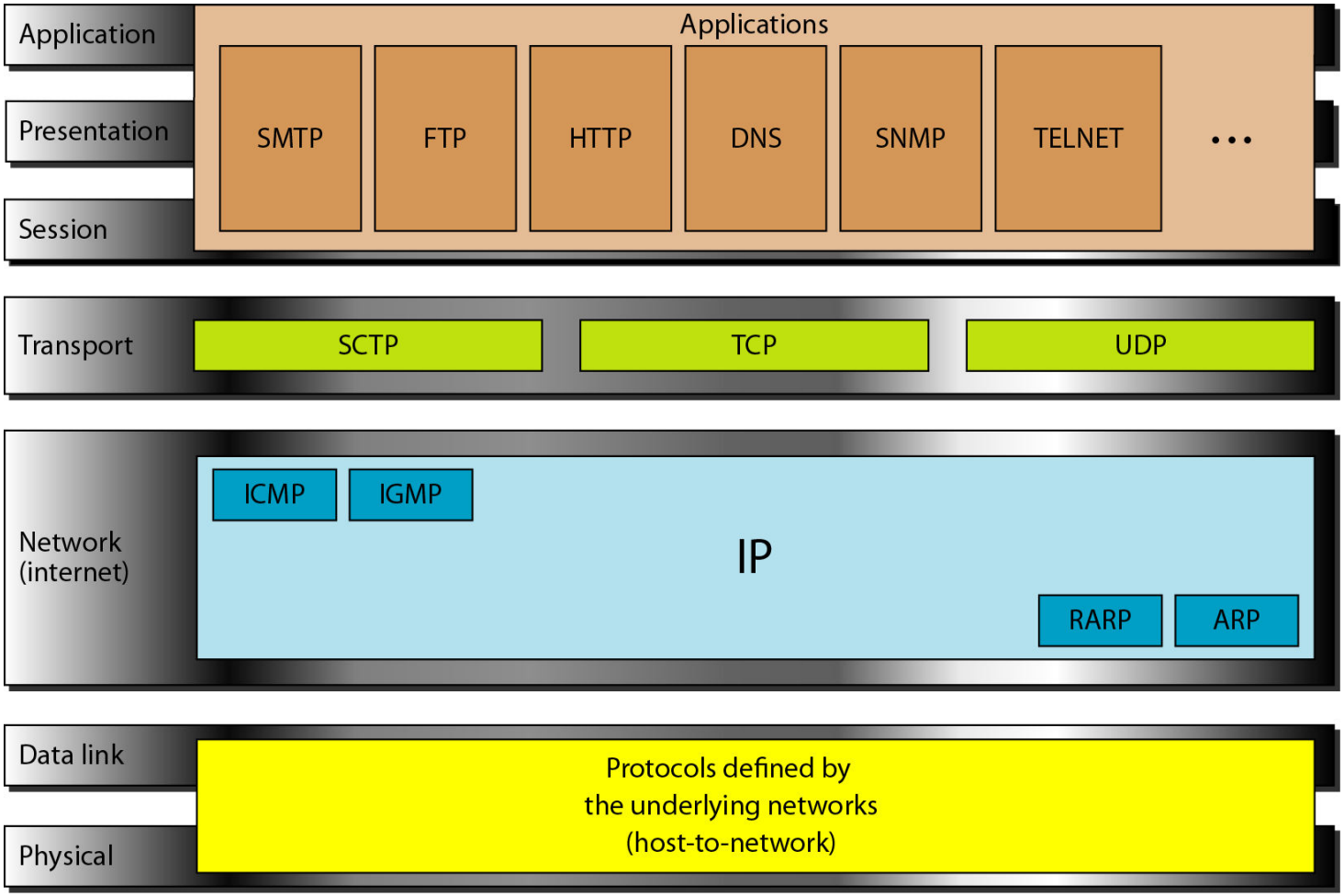

B
D
Send
P_1, P_2, and P_3
A
P_3
P_1
P_2
F
Receive packets
P_1, P_3, and P_2
C
E
Routes for packets
P_1 and P_2
Routes for
packet P_3

- It’s possible to set up a route in advance and have each packet follow the same route. That’s a *virtual circuit* or *virtual route* and is a different protocol.
- Often used in streaming applications where it’s important that packets arrive in order and at a predictable rate.

Transport Layer:

- lowest end-to-end layer
- Does not care what goes on in between
- Opening and closing connections
- Flow control
- Reassembly of IP packets
- Error control
- TCP-Transport Control Protocol (for example) of TCP/IP.
- Exchanges segments (units of layer 4 information)

Fig 2.16 TCP/IP and OSI Model



Addressing

- Physical address – also MAC (Media Access Control) address
- Each network card has a unique one
- 48 bits – used in layer 2 (LAN layer)
- Go to command prompt and enter the command *ipconfig /all*

- Logical address – also IP address
- Again *ipconfig /all*
- 32-bit value (IPv4)
- May also show a 128-bit value (IPv6)
- Needed for IP routing (layer 3)

- Port address
- Machine address not enough-there may be multiple apps
- Port number (16-bit number) defines the app.
- Well known port numbers

Review (book resource)

- http://highered.mcgraw-hill.com/sites/0072967757/student_view0/chapter1/flashcards.html
- http://highered.mcgraw-hill.com/sites/0072967757/student_view0/chapter1/multiple_choice_quiz.html
- http://highered.mcgraw-hill.com/sites/0072967757/student_view0/chapter2/flashcards.html
- http://highered.mcgraw-hill.com/sites/0072967757/student_view0/chapter2/multiple_choice_quiz.html

Zadaci-1:

1. Definišite OSI model.
2. Uporedite OSI model sa ostalim modelima. Definišite razlike i sličnosti.

SIGNALI

RAČUNARSKE MREŽE I WEB PROGRAMIRANJE

IV- GODINA

prof. dr Zlatko Langović

William A. Shay

- Transmission Media (Chapter 7)
- Many of the figures are from the textbook.

- Most of the diagrams in this powerpoint presentation (and future ones as well) come from the powerpoint file available via the book's web site. They reflect diagrams contained in the textbook.

Twisted pair

Figure 7.3 ***Twisted-pair cable***

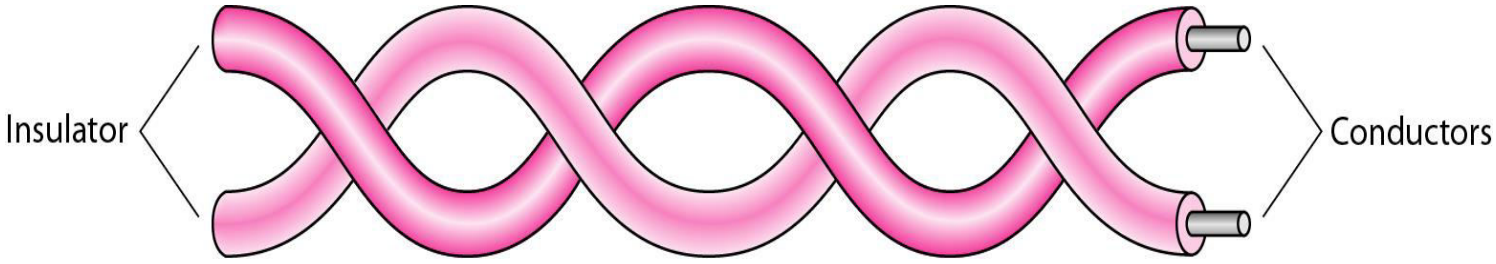
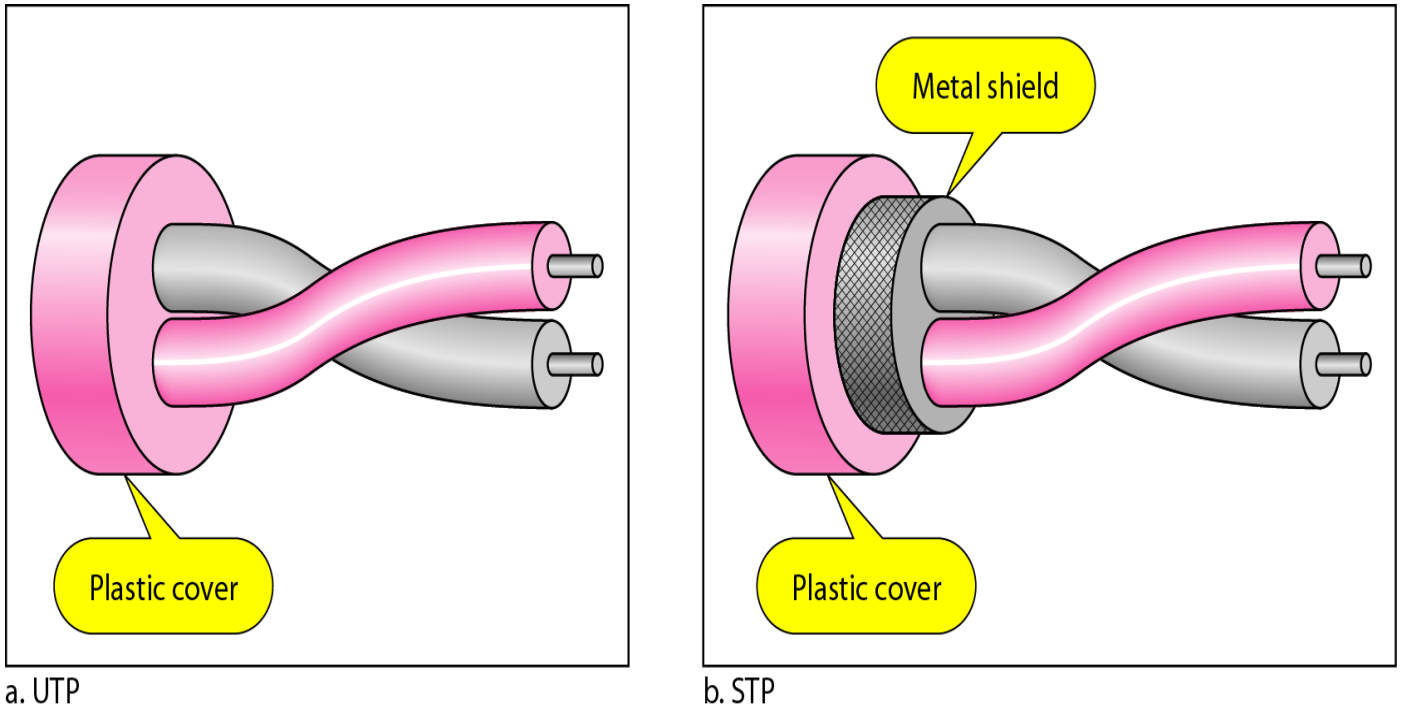


Figure 7.4 *UTP and STP cables*



a. UTP

b. STP

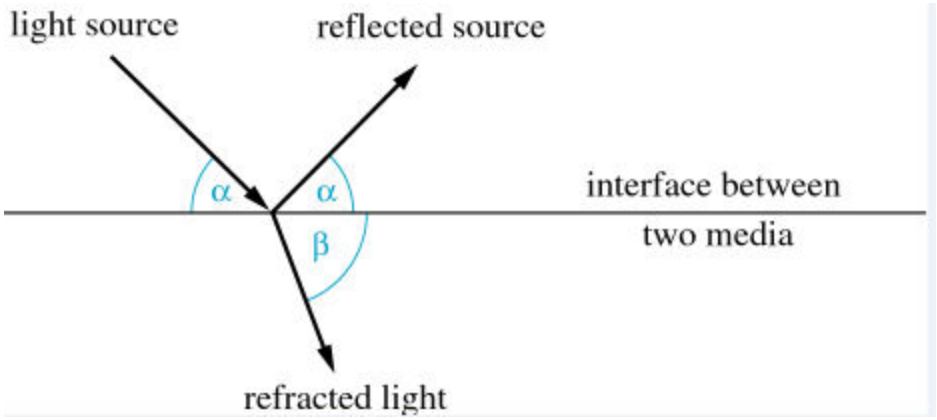
- twisted to avoid crosstalk [http://en.wikipedia.org/wiki/Inductance]
- [http://en.wikipedia.org/wiki/Crosstalk]
- [http://en.wikipedia.org/wiki/Twisted_pair]

- CAT 1 through CAT 6
- [http://en.wikipedia.org/wiki/Category_5_cable]
- [http://www.duxcw.com/digest/Howto/network/cable/cable2.htm]

- Coax cable: [http://en.wikipedia.org/wiki/Coaxial_cable]

Optical fiber:

- Light
- Refraction and reflection



- Optical signals are not electrical in nature and not subject to same interference as are electrical signals
- less attenuation
- [http://www.arcelect.com/fibercable.htm]

- LED/laser, multimode, graded index multimode, single mode
- modal dispersion (spreading of light in multimode fibers)
- infrared range of E/M waves.
- Dark Fiber: Optical fiber that's in place but currently not being used

- [http://electronics.howstuffworks.com/fiber-optic5.htm]
- [http://electronics.howstuffworks.com/fiber-optic2.htm]

Wireless:

- Electromagnetic spectrum [http://www.lbl.gov/MicroWorlds/ALSTool/EMSpec/EMSpec2.html]
- [http://www.purchon.com/physics/electromagnetic.htm]

Figure 7.18 *Propagation methods*

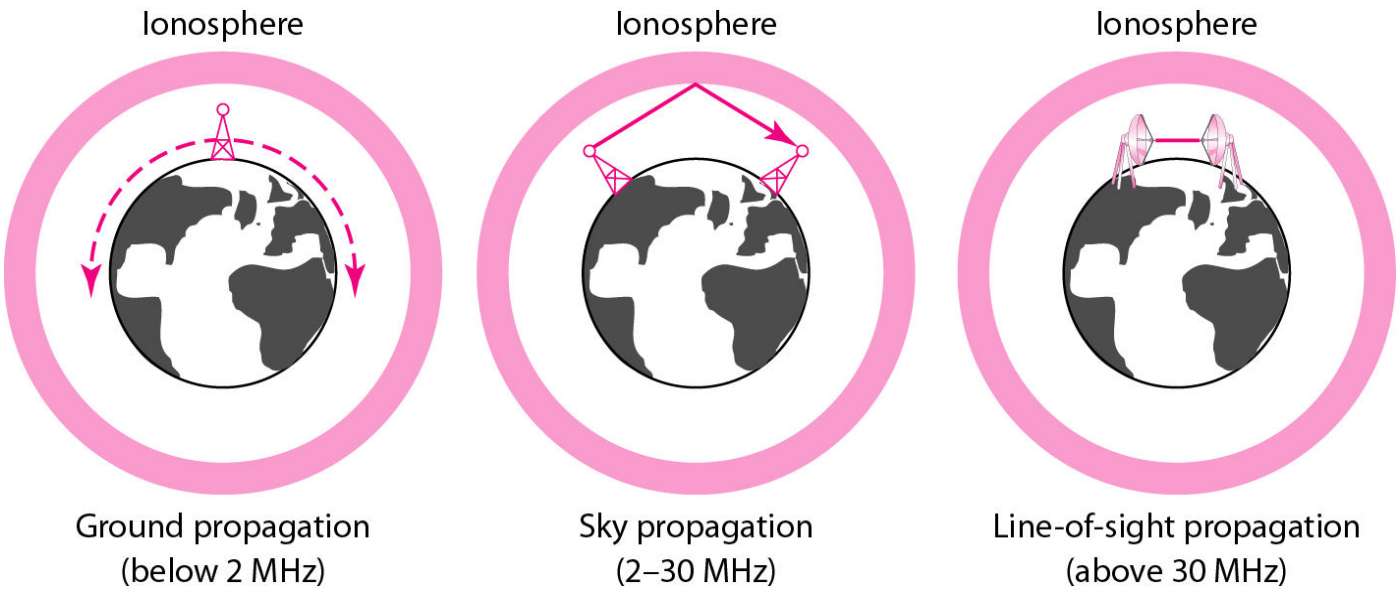


Table 7.4 *Bands*

Band	*Range*	*Propagation*	*Application*
VLF (very low frequency)	3–30 kHz	Ground	Long-range radio navigation
LF (low frequency)	30–300 kHz	Ground	Radio beacons and navigational locators
MF (middle frequency)	300 kHz–3 MHz	Sky	AM radio
HF (high frequency)	3–30 MHz	Sky	Citizens band (CB), ship/aircraft communication
VHF (very high frequency)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
UHF (ultrahigh frequency)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
SHF (superhigh frequency)	3–30 GHz	Line-of-sight	Satellite communication
EHF (extremely high frequency)	30–300 GHz	Line-of-sight	Radar, satellite

Radio waves

- Generally between 3 KHz and 1 GHz
- Regulated by the FCC
- Low data rates and not particularly suited for digital communications
- Length of antenna proportional to the wavelength (inversely proportional to the frequency)

Project ELF

- [http://www.fas.org/nuke/guide/usa/c3i/elf.htm]
- [http://enterprise.spawar.navy.mil/UploadedFiles/fs_clam_lake_elf2003.pdf]
- [http://en.wikipedia.org/wiki/Extremely_low_frequency]
- No longer active [http://www.senate.gov/~feingold/statements/03/01/2003310D02.html].

Danish King Harald Blatand:

- (Danish: Harald Blåtand, Old Norse: Haraldr blátönn, Norwegian: Harald Blåtann)
- born around 935, the son of King Gorm the Old, king of Jutland (that is, peninsular Denmark) and of Thyra (also known as Thyre Danebod) a supposed daughter of Harald Klak, Jarl of Jutland, or daughter of a noblemen of Sønderjylland who is supposed to have been kindly disposed towards Christianity.
- He died in 986 having ruled as King of Denmark from around 958 and king of Norway for a few years probably around 970.
- Some sources state that he was forcefully deposed by his son Sweyn as king [Wikipedia]

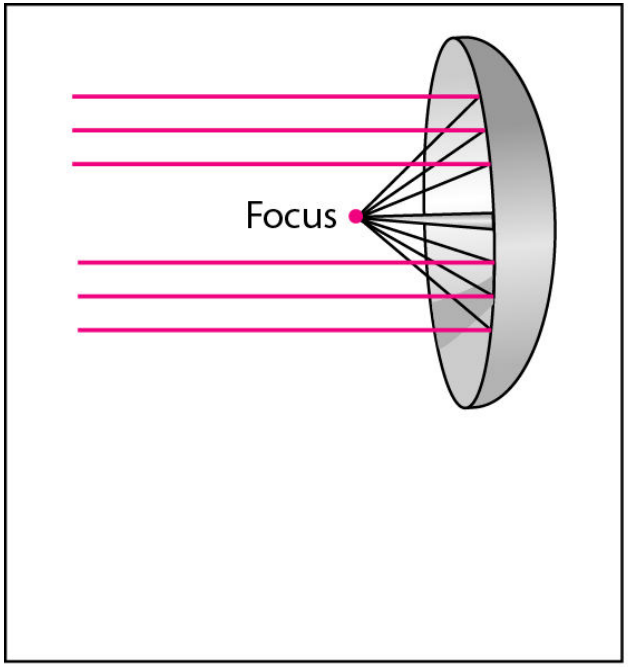
- You might know Blåtand by its other name
  - Bluetooth
- [http://en.wiktionary.org/wiki/Bl%C3%A5tand]
- [http://www.bluetooth.com]
- http://www.bluetooth.com/Pages/Fast-Facts.aspx

- Wireless LANs:
- infrared vs radio waves
- covered later

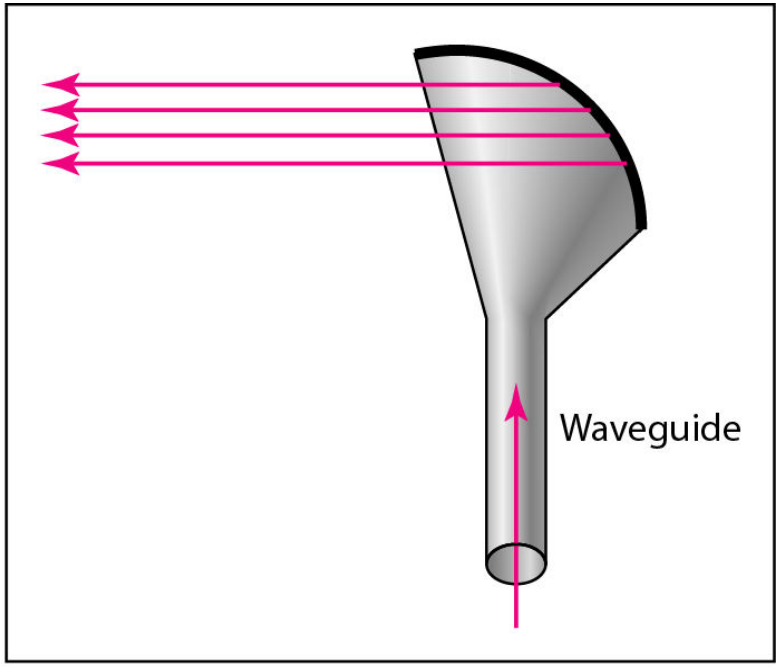
Free space Optics:

- [http://www.freespaceoptics.org/]
- optical technology without the fiber
- transmits using narrowly focused laser beams (10^{12} Hz range, which is unregulated by law)
- line of sight
- buildings sway and can lose tracking (although can use auto tracking receivers)
- after about 1-2 km signal becomes less focused
- fog can disrupt
- public perception of birds flying into the path of a laser, getting fried, and falling to the ground.

- Microwaves: parabolic and horn antenna, requires line-of-sight.



a. Dish antenna



b. Horn antenna

Satellites:

- Arthur C. Clarke wrote about them in 1940
- Sputnik
- Interestingly part of the sputnik crashed in Manitowoc
- [http://www.manitowoc.org/index.aspx?NID=1026]
- [http://www.roadsideamerica.com/sights/sightstory.php?tip_AttrId=%3D12959]

Planetary Orbits:

- too slow and object falls to earth; too fast and it speeds into space
- Keplers laws of planetary motion defines the speed as a function of, in part, the distance from the planet
- Geosynchronous (22,300 miles) orbital speed matches earth's rotation.
- By comparison, the space shuttle may orbit between 200 and 400 miles.
- TV satellite technology uses geosynchronous orbits

LEO (Low Earth orbit) Satellites

- Satellites move relative to ground position
- Useful for surveillance

LEO systems

- Iridium
- Orbcomm
- GlobalStar
- Other references

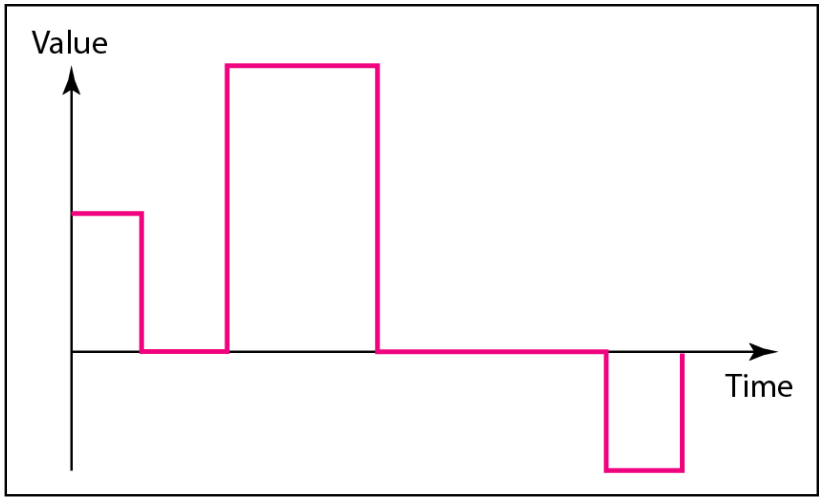
- GPS Systems
- Applications to Einstein's theory of relativity

- Cell phones, grids

Digital and Analog signals (Chapter 3 and 4)

- Digital signal vs analog signal
- Sound and images are naturally analog
- Computer data is digital

a. Analog signal



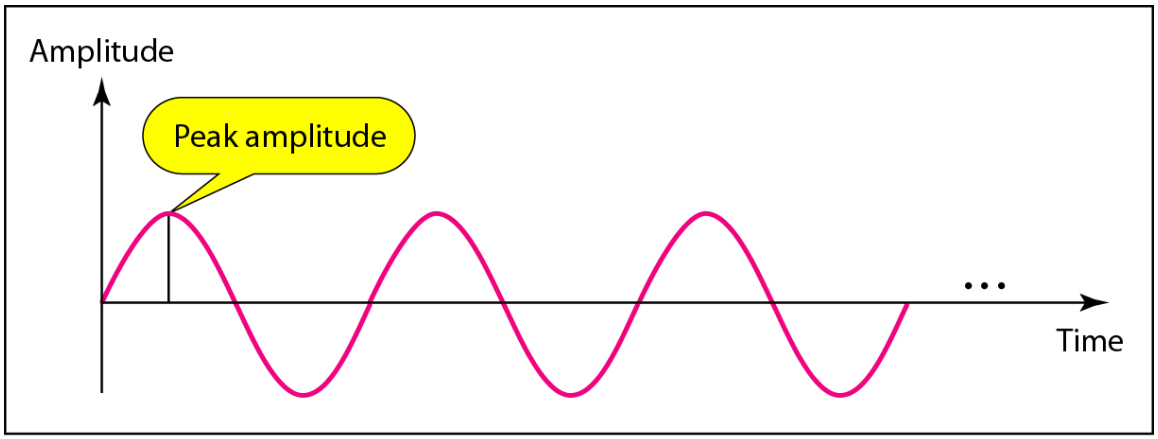
b. Digital signal

- periodic signal: repeating signal
- Cycle: part that repeats
- Period (p): length of a cycle
- Frequency (f): 1/period (1 Hertz (Hz) = 1 cycle per second (cps))
- f = 1/p and p = 1/f
- KHz = 10^3 Hz; MHz = 10^6 Hz; GHz = 10^9 Hz
- Bandwidth: (frequency range)
- bit rate: # bits per second (bps or Kbps or Mpbs or Gbps)

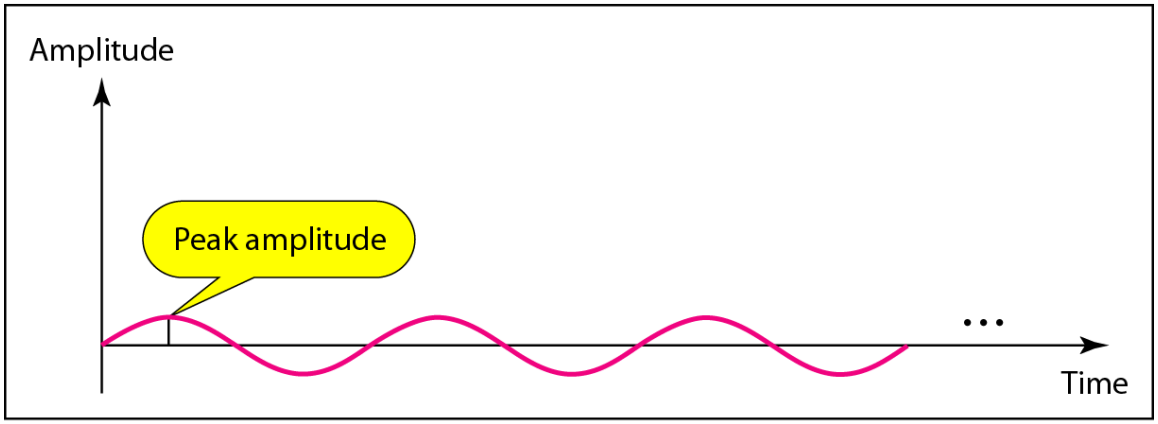
Periodic Analog signals (Chapter 3)

- An analog signal is defined by its frequency, amplitude, and phase.

Figure 3.3 *Two signals with the same phase and frequency, but different amplitudes*

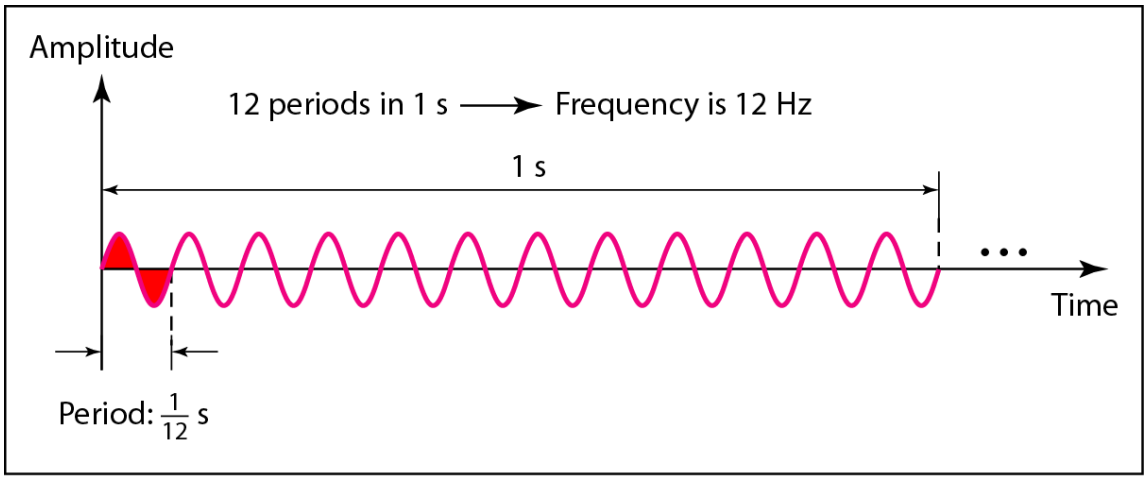


a. A signal with high peak amplitude

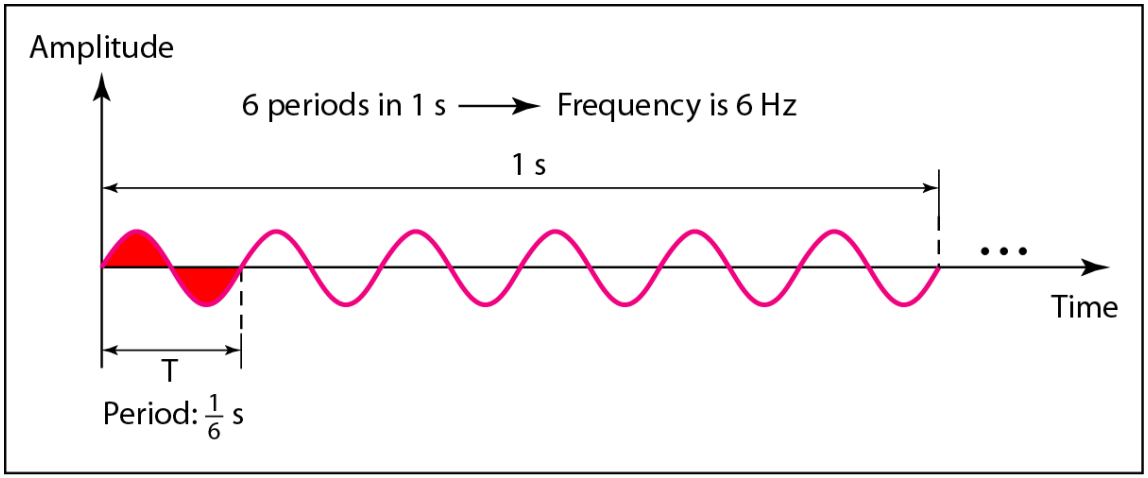


b. A signal with low peak amplitude

Figure 3.4 *Two signals with the same amplitude and phase, but different frequencies*

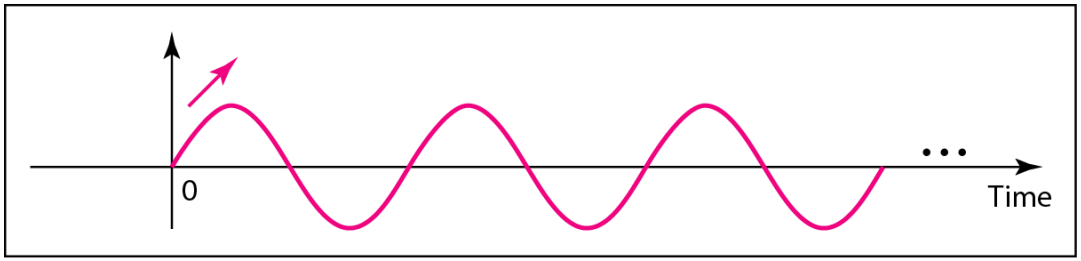


a. A signal with a frequency of 12 Hz

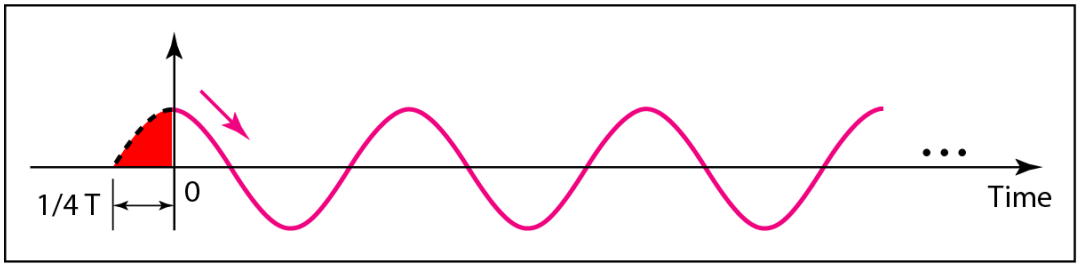


b. A signal with a frequency of 6 Hz

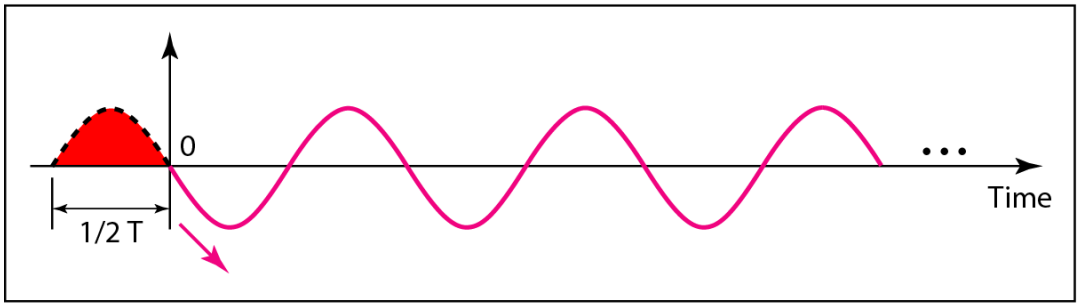
Figure 3.5 *Three sine waves with the same amplitude and frequency, but different phases*



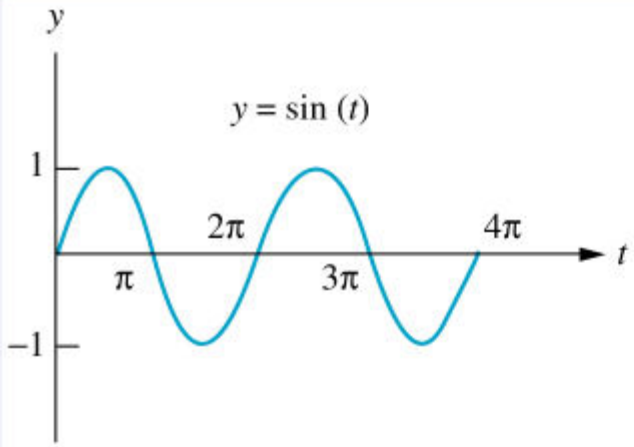
a. 0 degrees



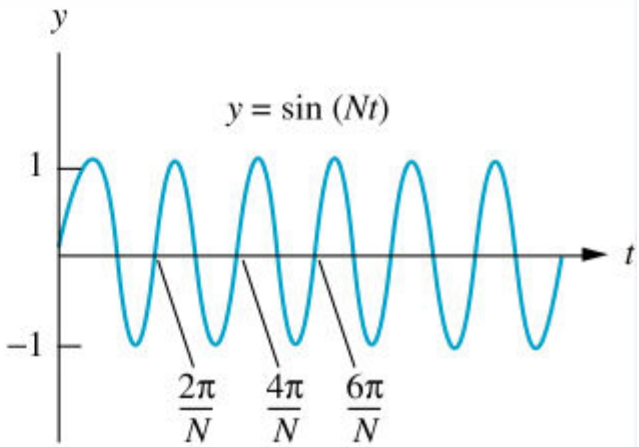
b. 90 degrees



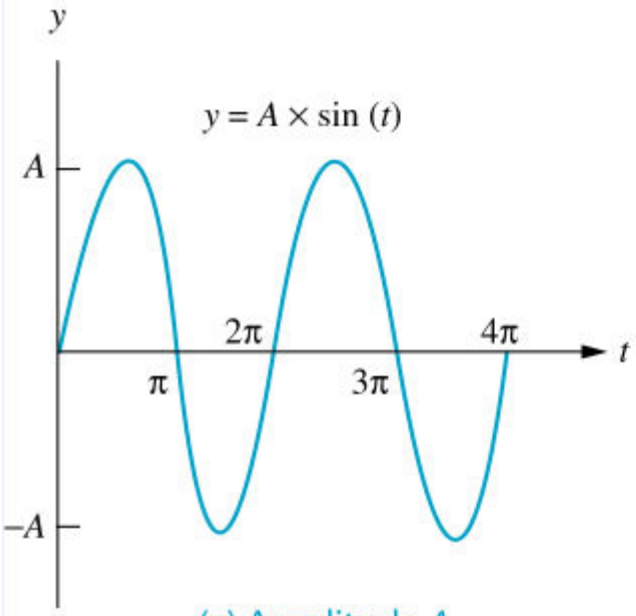
c. 180 degrees



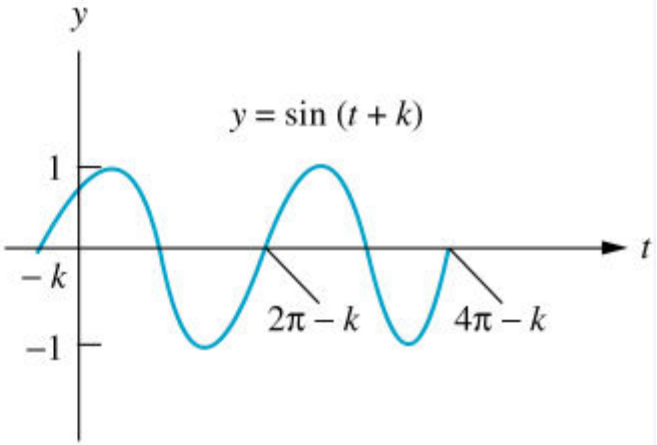
(a) Period 2π



(b) Period $\frac{2\pi}{N}$



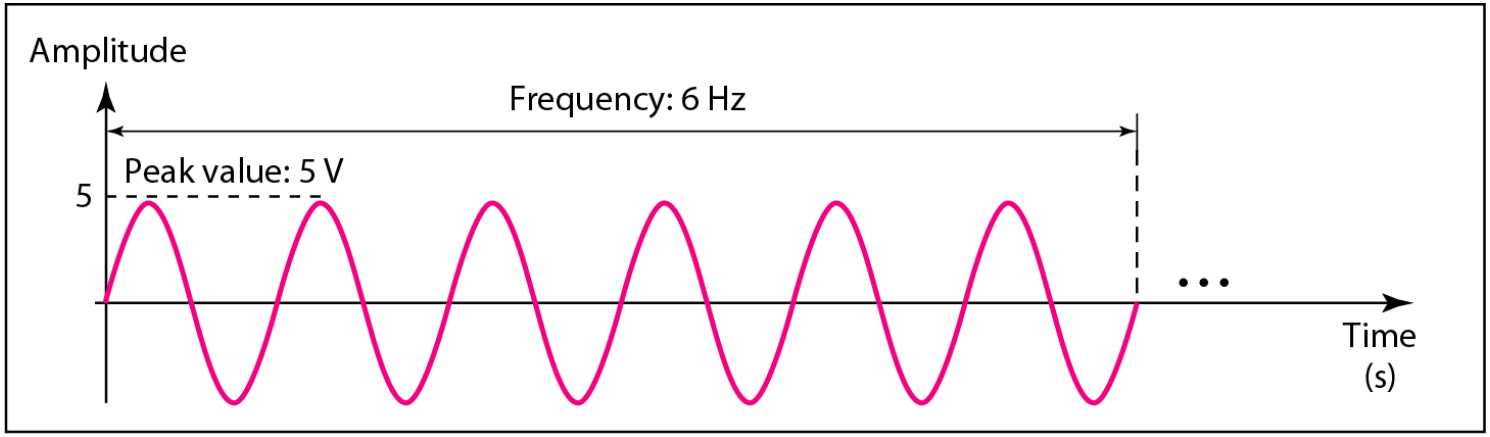
(c) Amplitude A



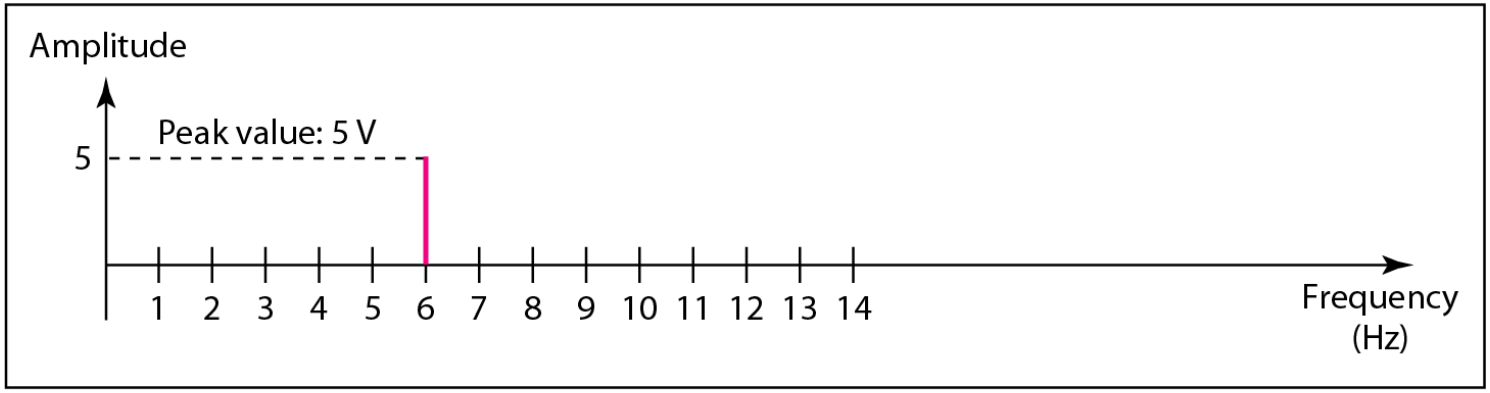
(d) Phase Shift k

- Maple worksheet has some examples

Figure 3.7 The time-domain and frequency-domain plots of a sine wave



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)

- Fourier results: a composite signal is a combination of simple sine and cosine wave with different frequencies and amplitudes.
- Applications to equalizers, filters, CATV.
- Ref: Maple worksheet
- [http://www.falstad.com/fourier/]
- Skip sections 3.3 and 3.4

Bit rate limits: section 3.5

- An analog signal with a fixed amplitude, frequency, and phase can represent a number of bits.
- Changing the characteristics at regular intervals can be used to transmit a bit stream
- bit rate: number of bits per second
- baud rate: frequency with which a signal's characteristics change
- Higher frequency signals can have a higher baud rate.

- Let n = #bits per baud and L = #different signals
- $L=2^n$ Equivalently, $n=\log_2(L)$
- eg. L=8 amplitudes => n=3 bits per baud (since $2^3 = 8$)
- Bit rate = baud rate·n = baud rate·$\log_2(L)$

Nyquist result:

- bit rate = $2 \cdot F \cdot n = 2 \cdot F \cdot \log_2(L)$ where F is the bandwidth (highest frequency minus lowest frequency) of the signal.
- This implies no theoretical limit on bit rates.

- Does not consider
  - Noise that affects a signal's characteristics
  - Limitations on devices to measure small differences between signals

Problem:

- noisy channels
- Large L (# different signals) means more subtle differences and more difficulty distinguishing them.
- Eg. With just a few amplitudes the amplitudes need not be close together.
- More amplitudes means the values are all closer together.

Shannon's result:

- S and N are signal and noise power
- S/N ratio: 1 Bel = $\log_{10}(S/N)$; 1 decibel (db) = 0.1 Bels
- Bit rate = bandwidth$\cdot\log_{10}(1 + S/N)$ bps
- If little noise, S/N is large and the bit rate is larger.
- If lots of noise, bit rate is smaller.

Phone system:

- Maximum bit rates for dial-up modems
- bandwidth ~ 3000 Hz
- S/N ratio ~ 35 db;
- 3.5 Bels = $\log_{10}(S/N) \rightarrow S/N \sim 10^{3.5} \rightarrow S/N \sim 3162$
- bit rate ~ $3000 \cdot \log_{10}(1 + 3162) \sim 35{,}000$ bps max.
- This applies to old cases where there was a modem on each end.

- 56Kbps possible because remote end usually connected to an ISP and there's no analog component at that end.
- Thus downloads do NOT encounter an analog-digital conversion at the remote end, which (using PCM-covered later) is susceptible to quantization noise.

- Can skip section 3.6

Digital data (Chapter 4). Can skip section 4.1 except for the schemes below.

- NRZ schemes: generally 0=high or low and 1=low or high (the opposite of 0)
- Ref: [http://www.frontiernet.net/~prof_tcarr/Encodings/applet.html#APPLET]
- NRZ subject to baseline wandering or loss of synchronization.
- A very long string of 0s or 1s may be difficult to interpret correctly.

- Manchester 1=Low-Hi, 0 = Hi-Low;
- Differential Manchester 1=no change at start, 0=change at start; transition in the middle.
- Manchester codes are self synchronizing
- frequency = 2·bit rate (an issue).
- Section 4.1 has many other schemes but we'll skip most and defer a couple others until later to put them into a context.

Analog to digital (section 4.2) Can skip material not related to PAM or PCM

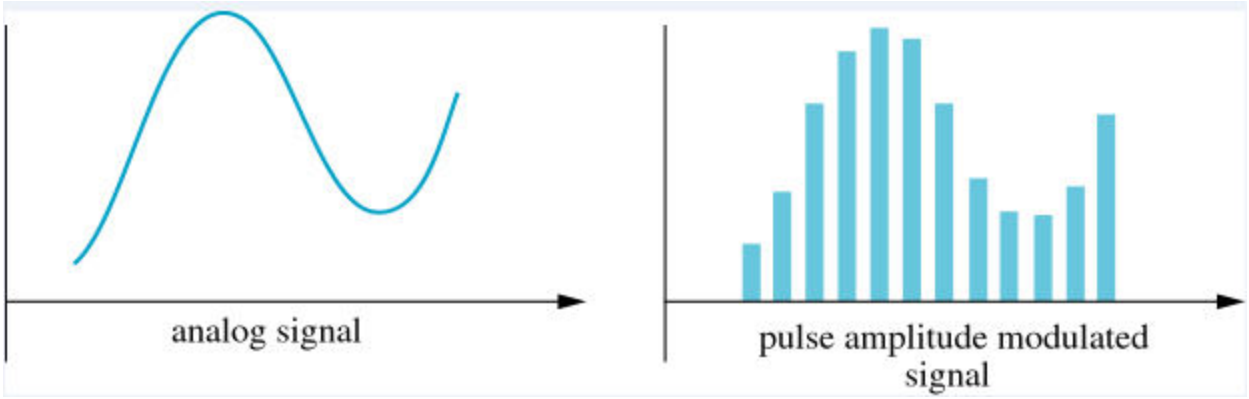
- Here we deal with analog data, not simple sine/cosine waves with fixed characteristics.

Sampling theorem

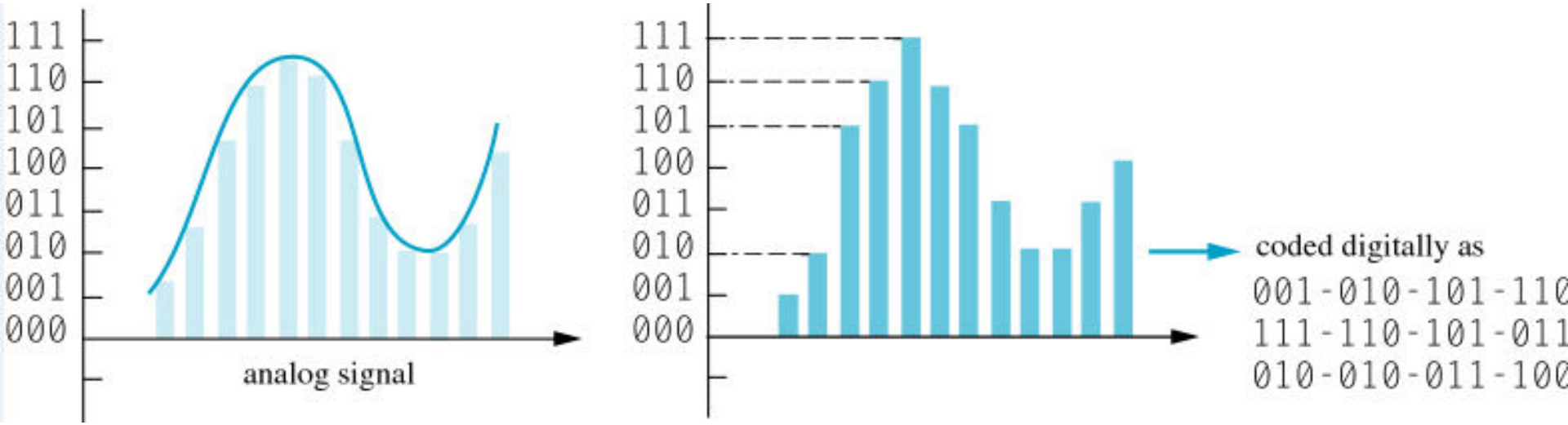
- F = max frequency
- receiver can reconstruct a signal by sampling it at least 2F times per second.
- Sample obtained by generating a sampling signal consisting of pulses at specified intervals.

- [http://www.cs.cf.ac.uk/Dave/Multimedia/node149.html#sine]
- [http://www2.egr.uh.edu/~glover/applets/Sampling/Sampling.html]

- Pulse Amplitude modulation
- Each pulse has analog characteristics in that it can be any real value



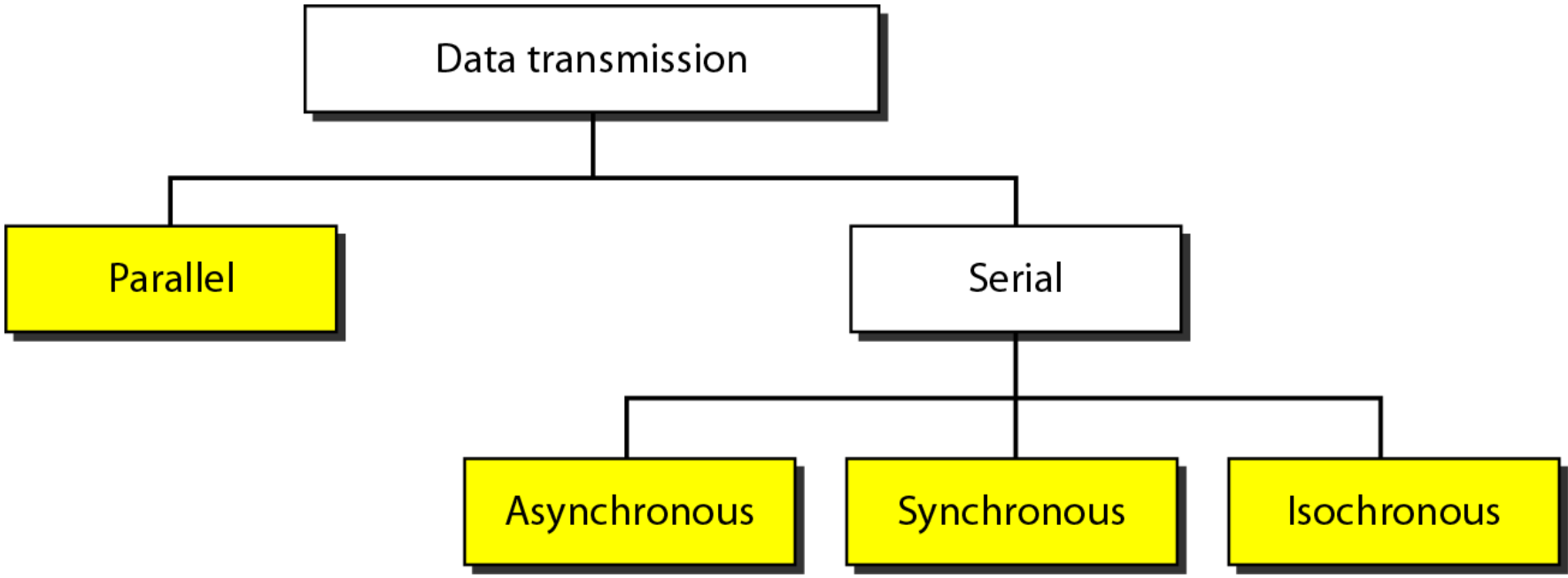
- PCM (uncompressed video and audio)
- Divide vertical into bit groups
- Take samples and round to the nearest bit group. This rounding is called **quantization noise** and results in some signal loss.



- [http://en.wikipedia.org/wiki/PCM]
  - Digitized voice (telephone) → 8000 samples (4000 Hz max frequency)·8 bits per sample → 64kbps
  - CD player tech specs: Sampling frequency 44.1 KHz, D-A conversion = 16-bit linear → frequency response up to ~20,000 Hz and 64,000 signal amplitudes.

Transmission Modes: Section 4.3

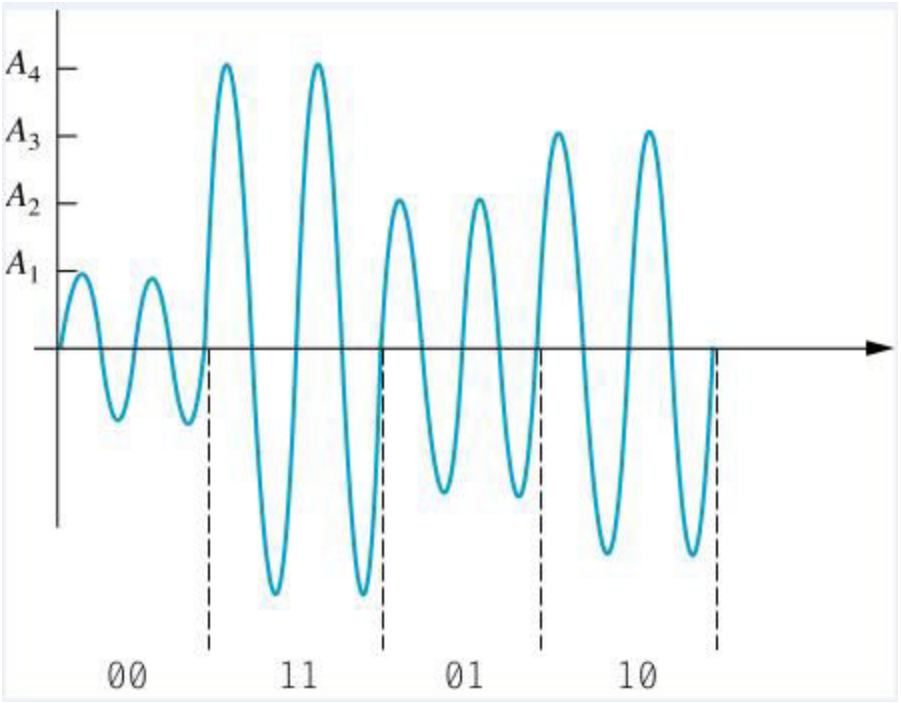
Figure 4.31 *Data transmission and modes*



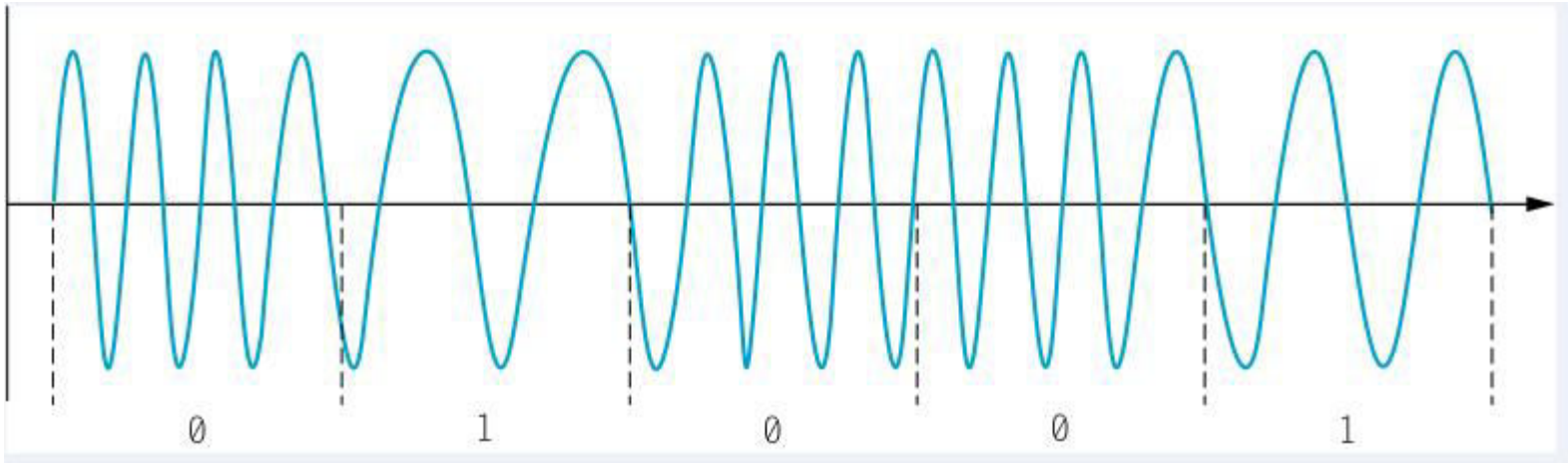
Digital to analog (Section 5.1)

- Modulation:
  - Amplitude Shift Keying (ASK)
  - Frequency Shift keying (FSK)
  - Phase shift keying (PSK)
  - Quadrature modulation (QAM)

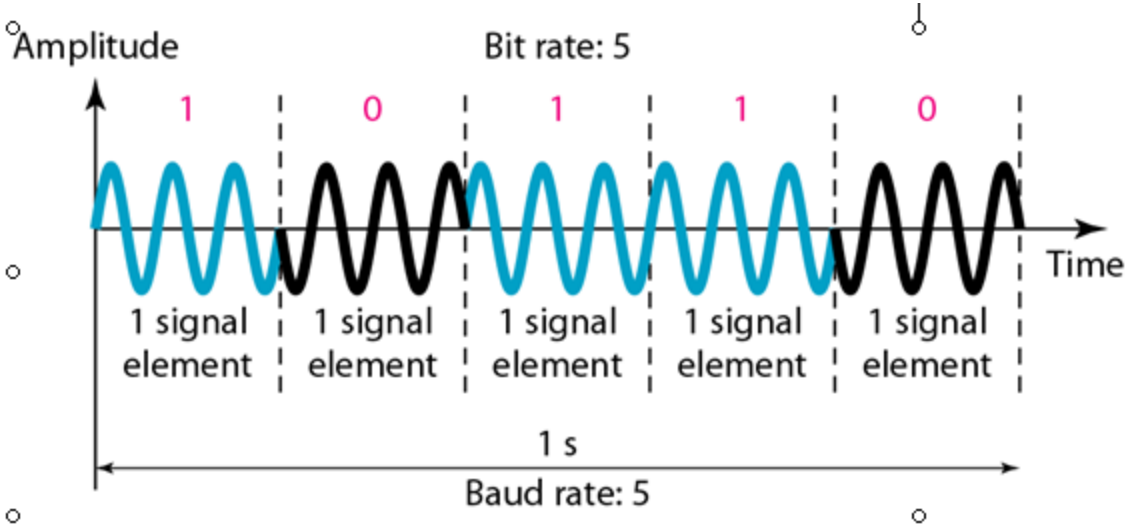
ASK



FSK



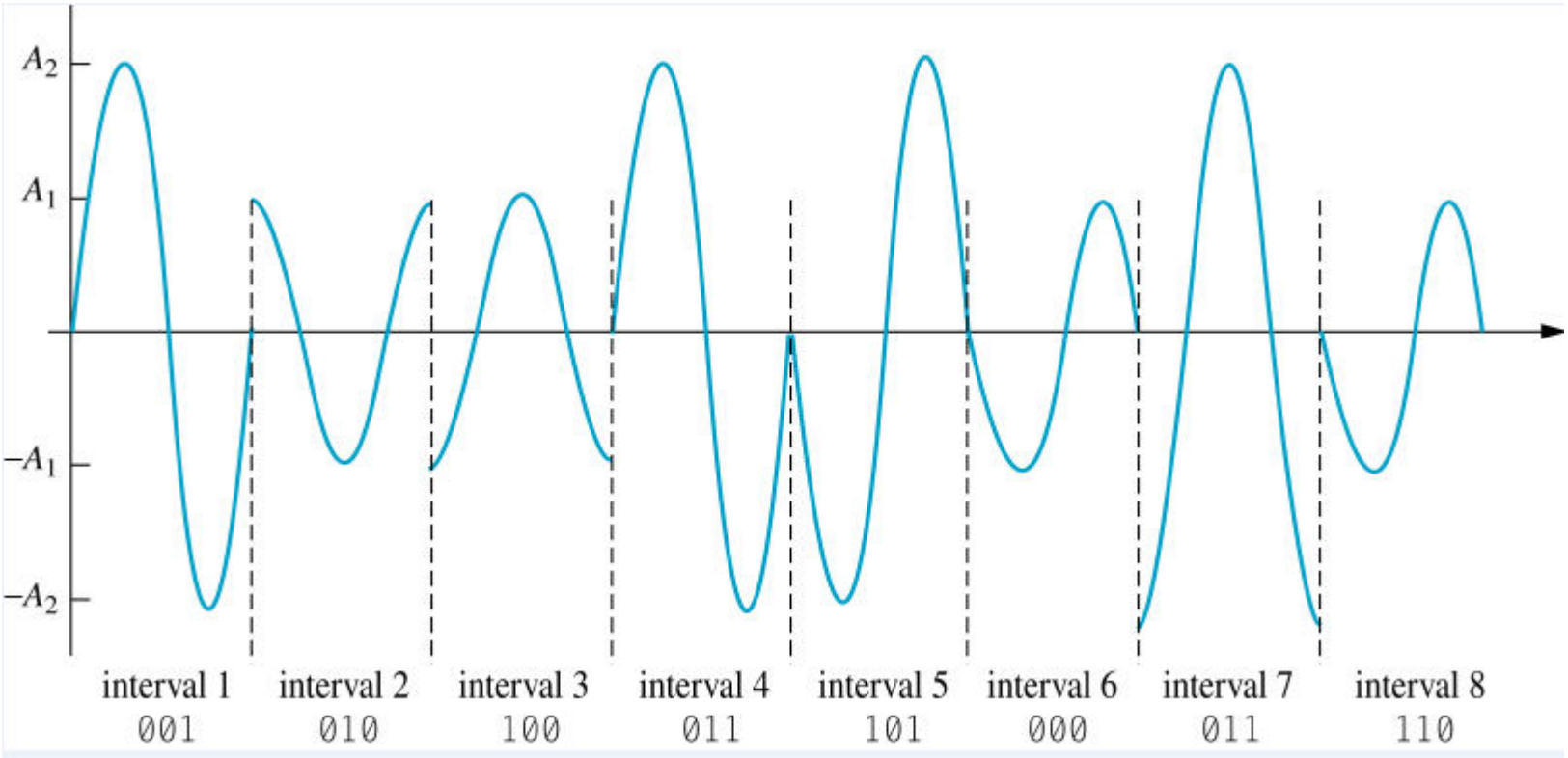
PSK


Amplitude
Bit rate: 5
1
0
1
1
0
Time
1 signal element
1 signal element
1 signal element
1 signal element
1 signal element
1 s
Baud rate: 5

Example Signal Association for Quadrature Amplitude Modulation

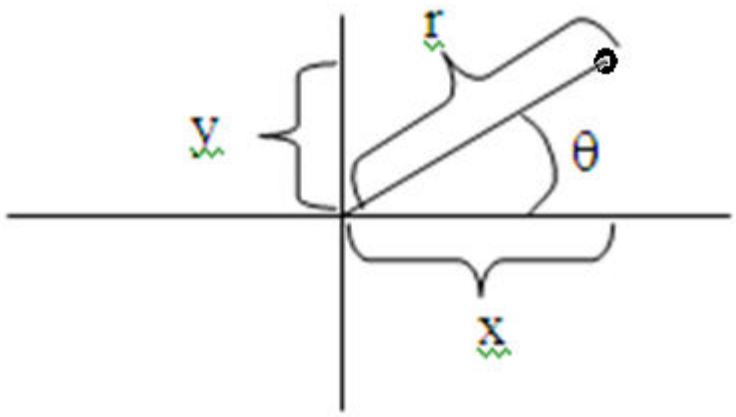
Bit Values	Amplitude of Generated Signal	Phase Shift of Generated Signal
000	A_1	0
001	A_2	0
010	A_1	$1/(4f)$
011	A_2	$1/(4f)$
100	A_1	$2/(4f)$
101	A_2	$2/(4f)$
110	A_1	$3/(4f)$
111	A_2	$3/(4f)$

- Phase is relative to the signal in the previous interval

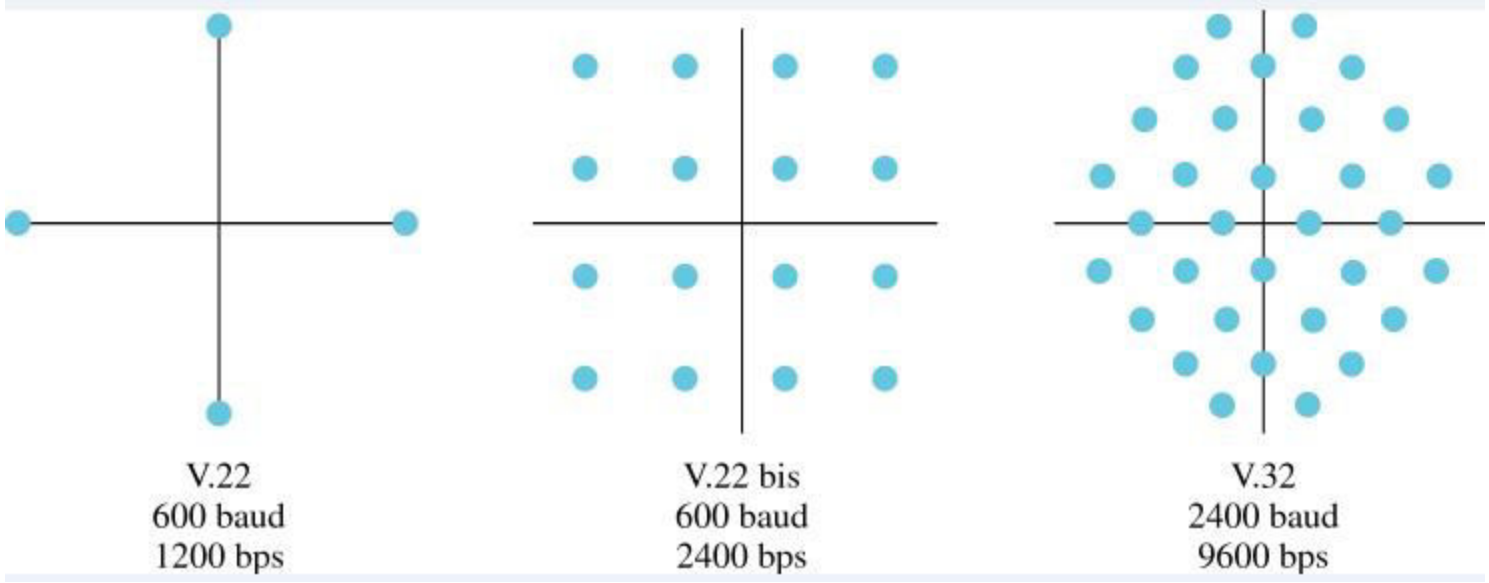


- There is a simulation at [http://williams.comp.ncat.edu/Networks/modulate.htm] Also look at [http://www.mathsnet.net/graphs/cuoc3.html]

- Signal constellation: Each signal is represented by a point as shown below
- θ is the phase shift
- r is the amplitude



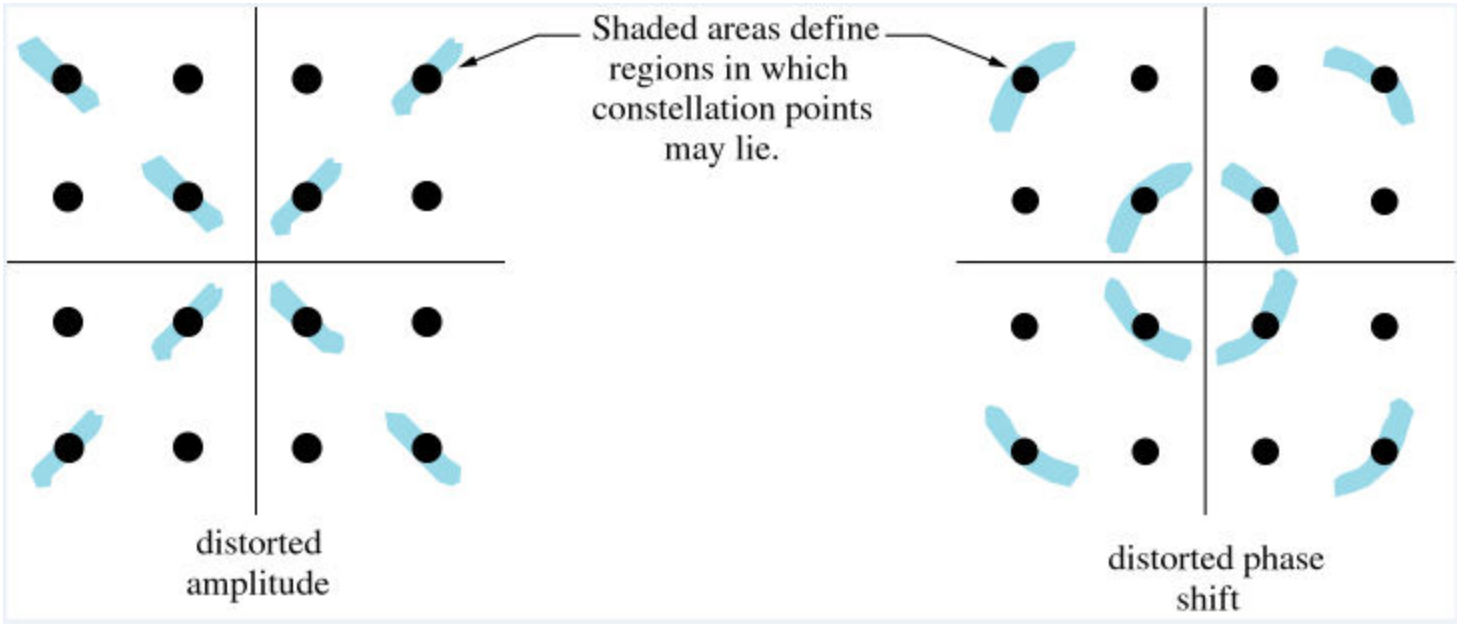
$$r = \sqrt{x^2 + y^2}\,; \quad \sin\theta = \frac{y}{\sqrt{x^2 + y^2}} \;\text{->}\; \theta = \arcsin\frac{y}{\sqrt{x^2 + y^2}}$$



V.22
600 baud
1200 bps

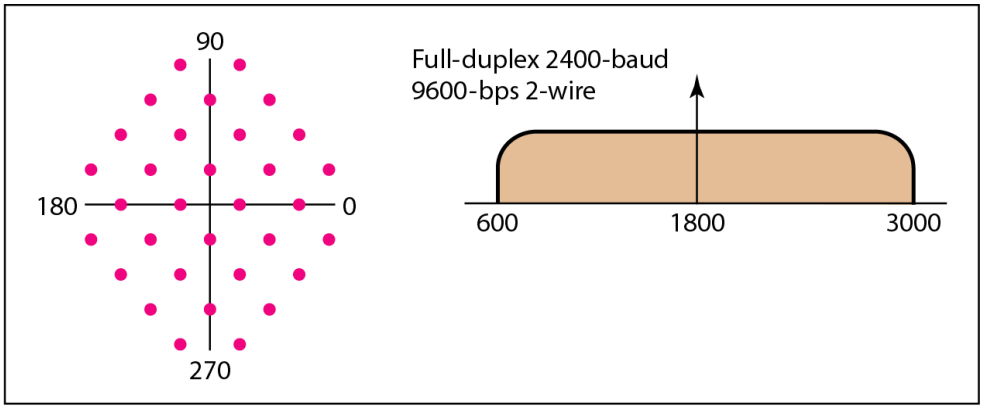
V.22 bis
600 baud
2400 bps

V.32
2400 baud
9600 bps

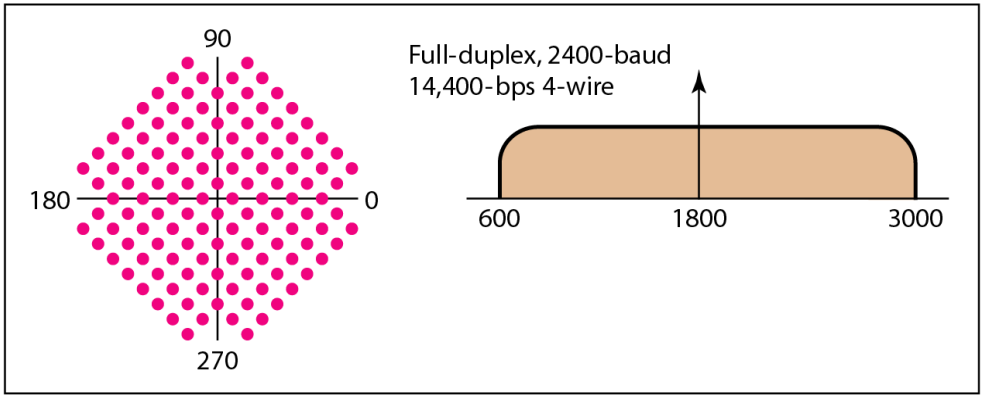

Shaded areas define regions in which constellation points may lie.
distorted amplitude
distorted phase shift

- Modems (standards defined by CCITT (now ITU) denoted by V.xx
- See also Section 9.2

Figure 9.8 *The V.32 and V.32bis constellation and bandwidth*

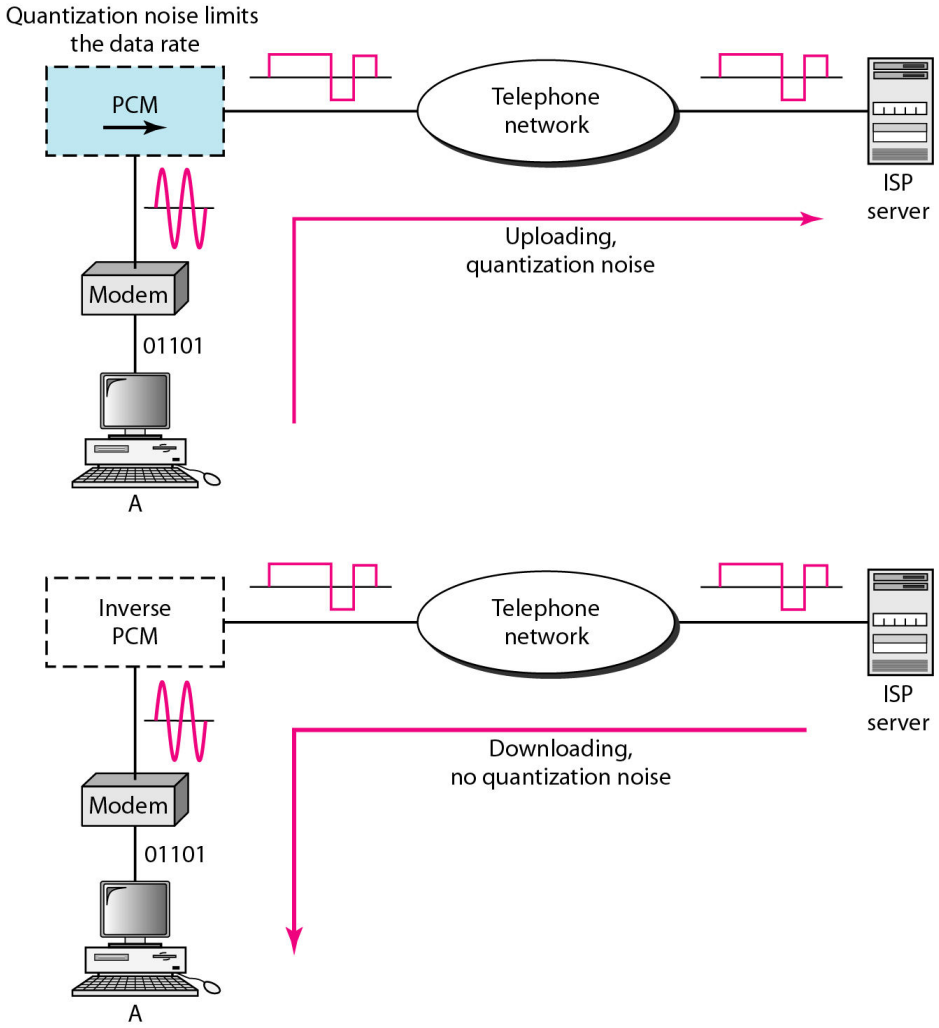


a. Constellation and bandwidth for V.32



b. Constellation and bandwidth for V.32bis

Figure 9.9 *Uploading and downloading in 56K modems*



- Cable modems
- See also Section 9.5

DSL – Section 9.3

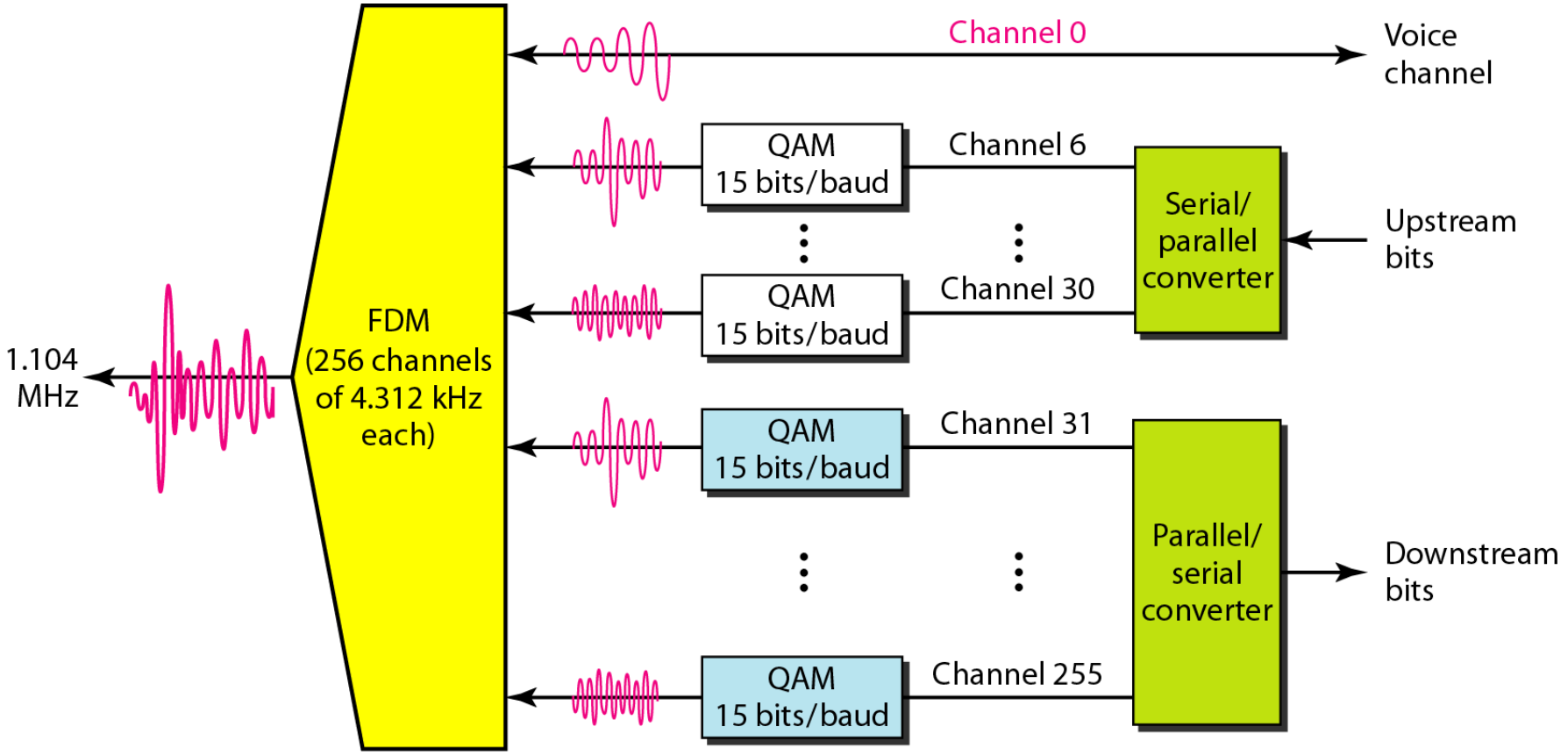
- DSL (Digital Subscriber Line):
- local loop (last mile)
- POTS (Plain Old Telephone system)
- not high quality like CAT5 and not likely to be ripped out and new wires installed (costly)
- It is capable of transferring higher frequencies than telephone produces.

- Discrete Multitone
  - Divide the frequency range from 0 Hz to 1.104 MHz into 256 separate channels
  - Use the five lowest channels for POTS
  - Use the remaining channels for upstream and/or downstream transmission with more channels reserved for downstream

- some channels to be used by both upstream and downstream transfers
- To transmit data, divide a bit stream into smaller groups of bits, one group for each channel
- Apply a QAM technique to the bits in each channel

- Combine the QAM-generated signals and subject the result to an Inverse Fast Fourier Transform. (Mathematical function that is able to determine frequency components of a complex signal)
- Downstream (in theory) up to 6 Mbps. Upstream is less.

Figure 9.10 *Discrete multitone technique*



Multiplexing: Section 6.1

- Multiplexing: combining multiple data signals onto a single data link.

Figure 6.4 *FDM process*

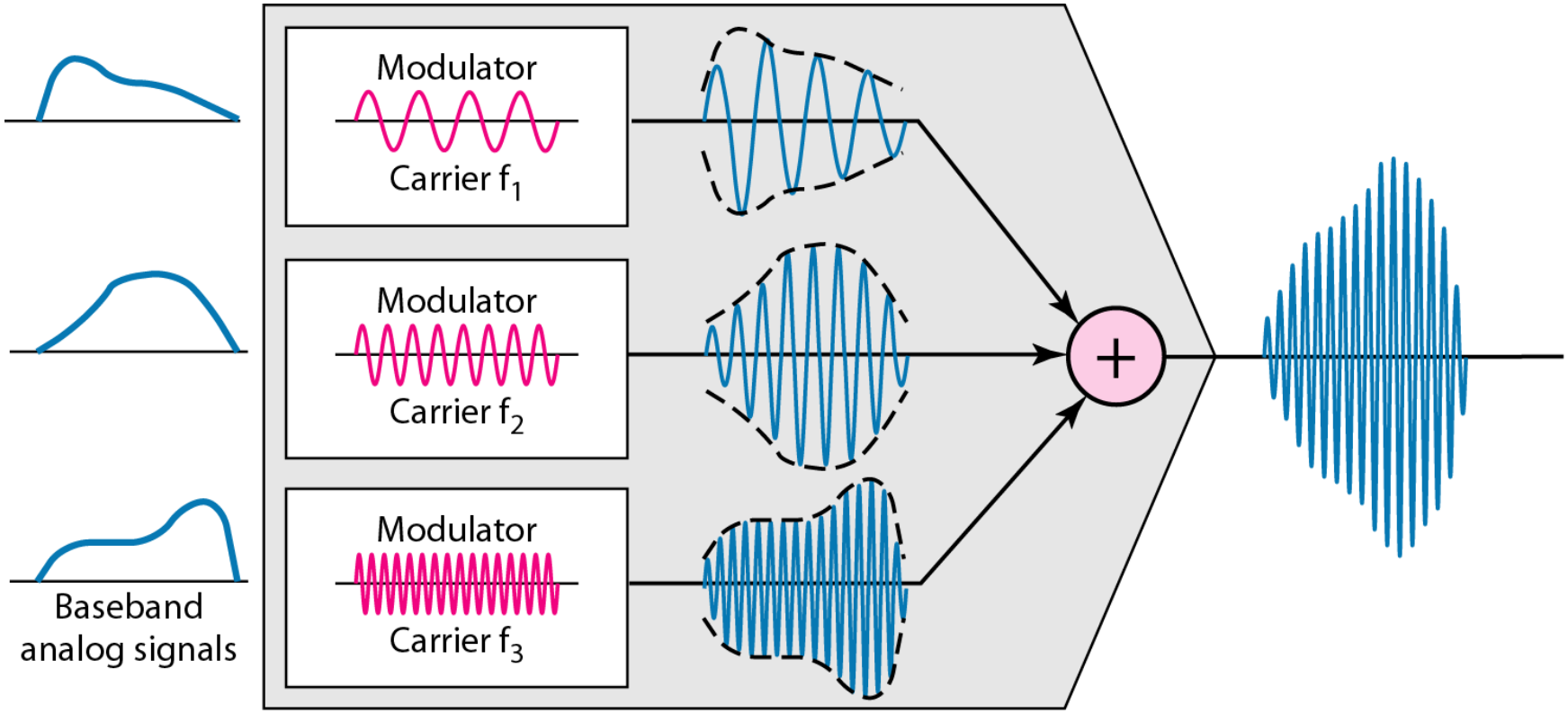


Figure 6.5 *FDM demultiplexing example*

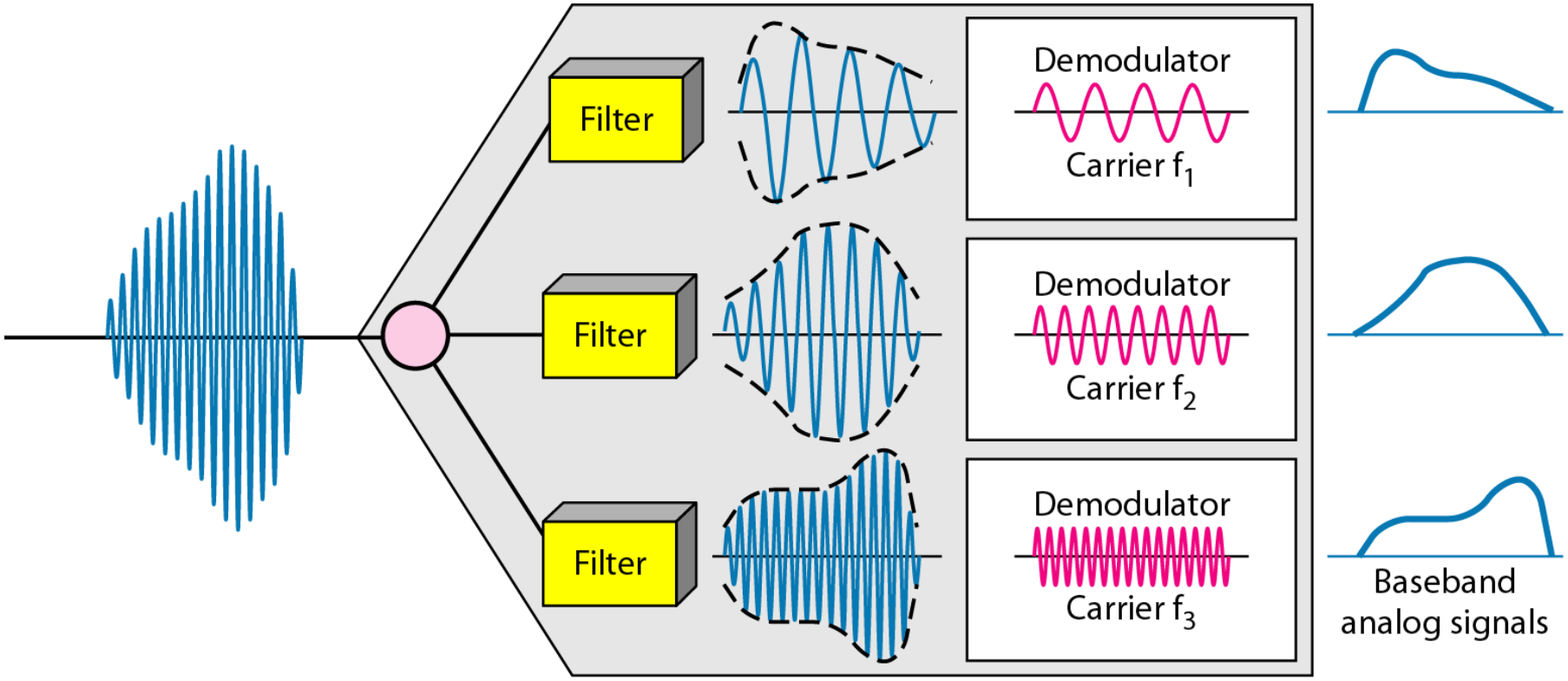
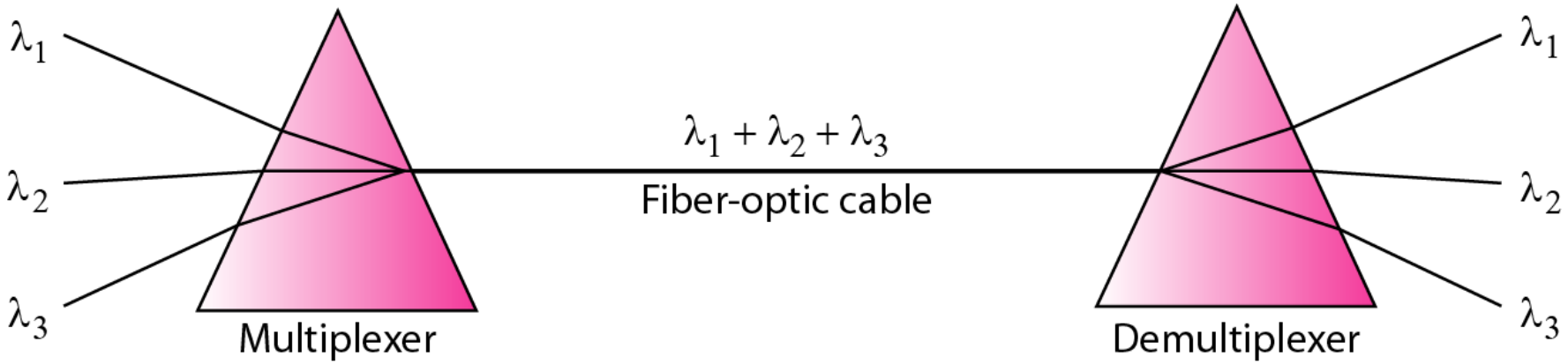


Figure 6.11 *Prisms in wavelength-division multiplexing and demultiplexing*



Time Division Multiplexing

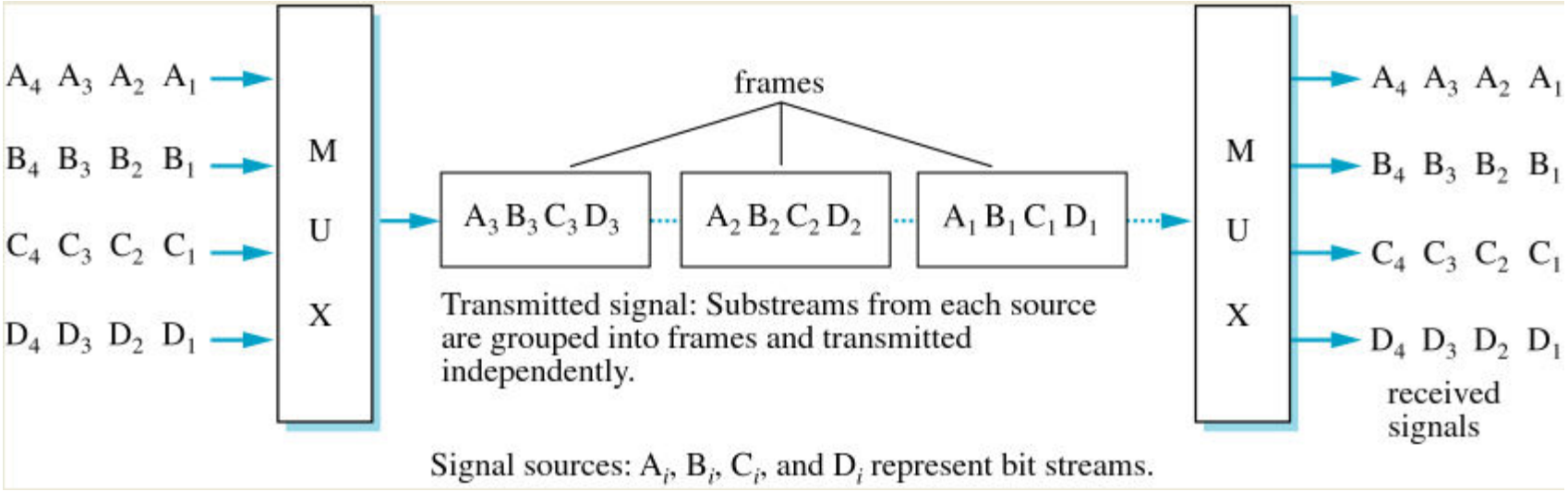


Figure 6.23 *Digital hierarchy*

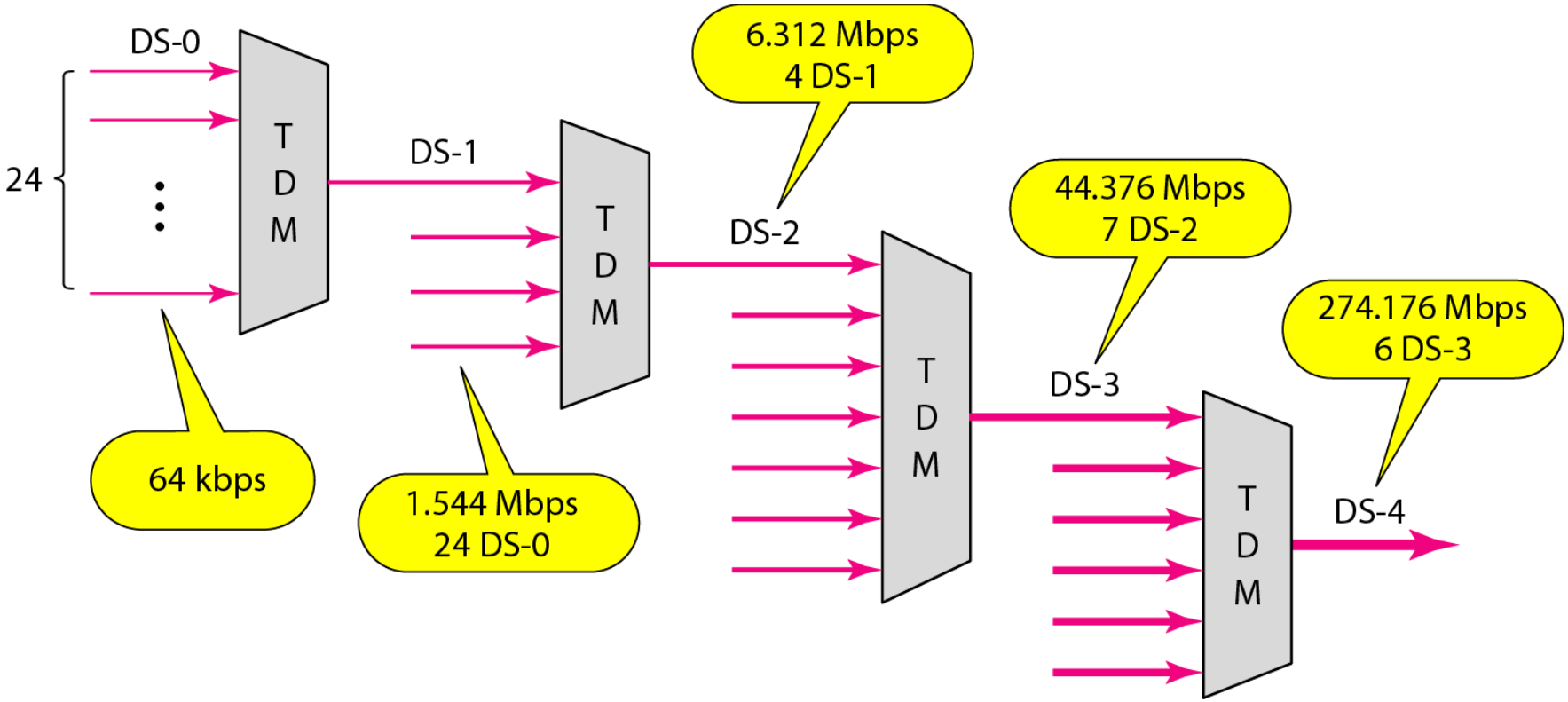


Figure 6.24 *T-1 line for multiplexing telephone lines*

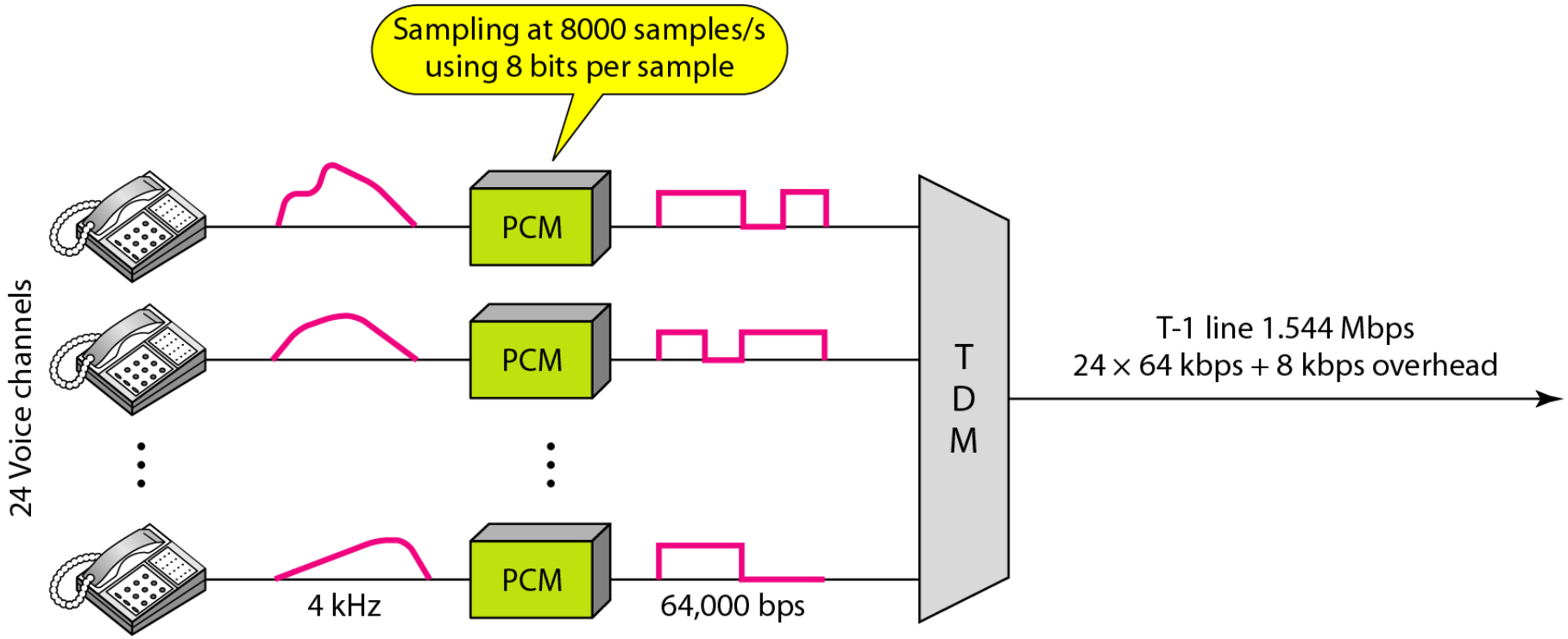


Figure 6.25 *T-1 frame structure*

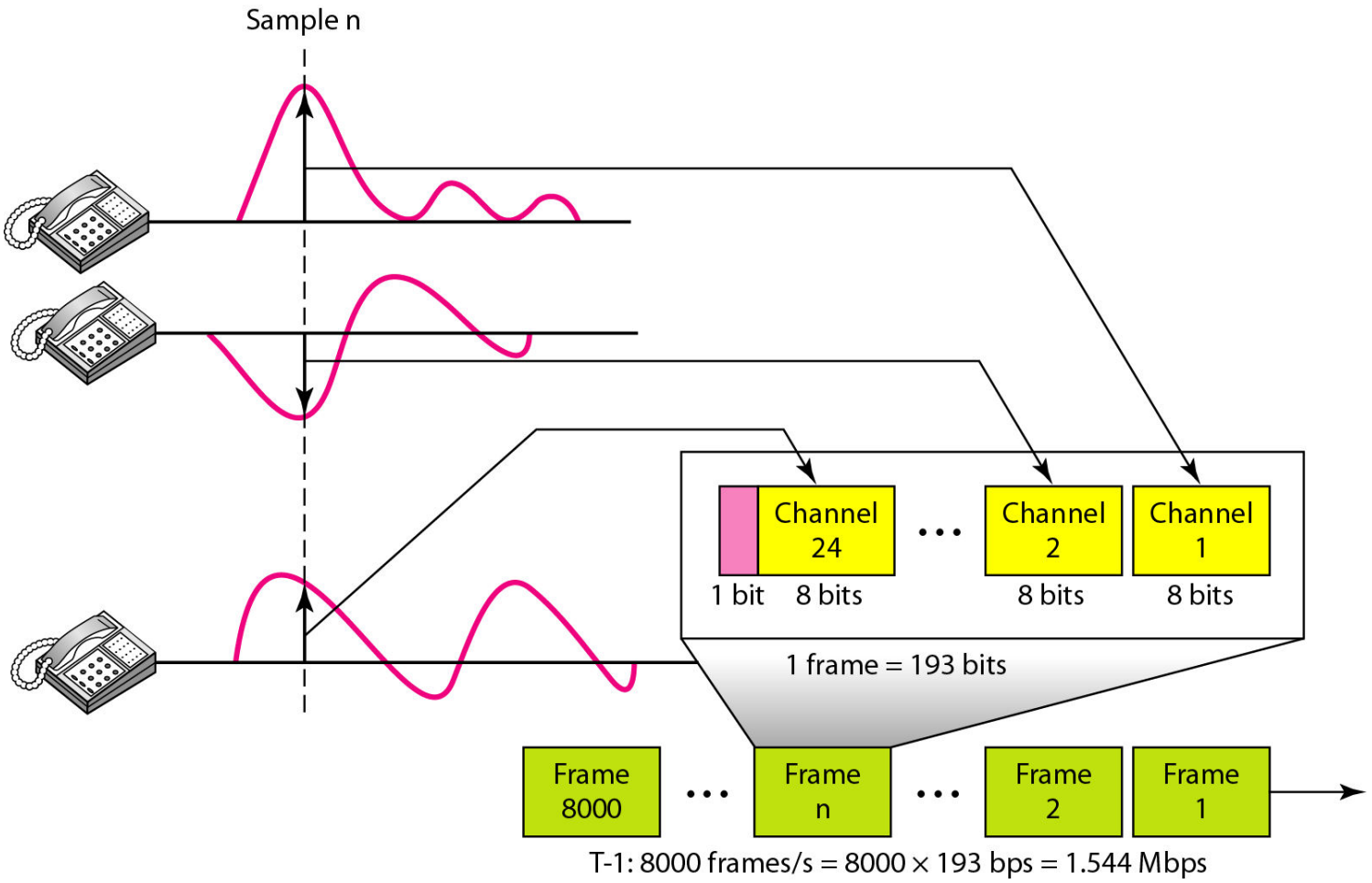


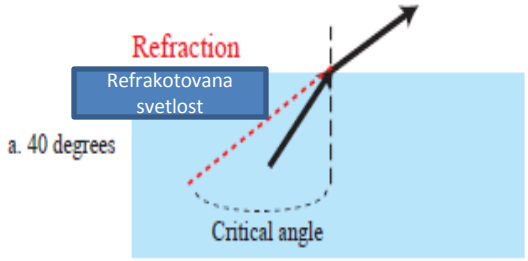
Table 6.1 *DS and T line rates*

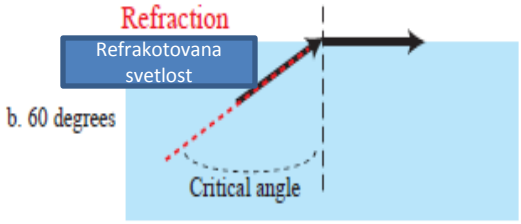
Service	*Line*	*Rate (Mbps)*	*Voice Channels*
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

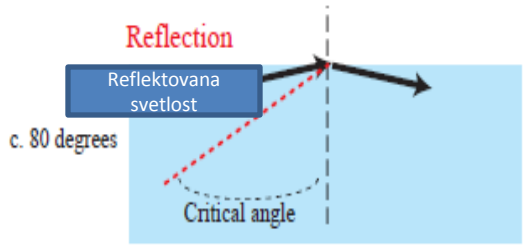
Zadaci-2:

2. Neka α (upadni ugao) bude ugao pod kojim svetlost seče datu graničnu oblast. Jedan deo svetlosti se reflektuje nazad pod uglom α u odnosu na ravan, a drugi deo prolazi kroz granicu do drugog medijuma pod uglom β (prelomni ugao), to je refrakcije (prelamanje). Kada je α manje od određenog kritičkog ugla, nema refraktovane svetlosti. Drugim rečima sva svetlost se reflektuje. Zahvaljujući ovom fenomenu, omogućeno je funkcionisanje optičkog fibera. Optička tehnologija nalazi sve veću primenu u poslovnom okruženju. U zadatki je izvor svetlosti u sredini, koja je obojena plavom bojom.

   1. Definišite, da li se, "drugi" delovi svetlosti reflektuju ili/i refraktuju, ako je za posmatrane medijume kritična vrednost ugla 60 stepeni (crvena isprekidana linija na grafiku).
   2. U kom odnosu uglova, nema refraktovane svetlosti tj. koja sličica prikazuje fenomen na kome se zasniva funkcionisanje opitčkog fibera?


Refraction
Refrakotovana svetlost
a. 40 degrees
Critical angle


Refraction
Refrakotovana svetlost
b. 60 degrees
Critical angle


Reflection
Reflektovana svetlost
c. 80 degrees
Critical angle

3. Neka je α, ugao pod kojim svetlsot “seče” datu graničnu oblast. Jedan deo svetlosti se reflektuje nazad pod uglom α u odnosu na ravan, a drugi deo prolazi kroz granicu do drugog medijuma. Između datih uglova postoji relacija koja se definiše Šnelovim pravilom, koji se može definisati na sledeći način: cosα /cosβ. Ako je dati indeks manji od 1, svetlost putuje u medijum sa manjom optičkom gustinom, i obrnuto ako je veći od 1 putuje u medijum sa većom optičkom gustinom. (Ova karakteristika je važna za optičku tehnologiju koja nalazi sve veću primenu u poslovnom okruženju). Ako je ugao β = 45°, nađi ugao α tako da dati koeficijent 1.
   - Rešenje:
     - Uglovi β i α su uglovi čije se vrednosti nalaze u intervalu od 0° do 90°.
     - Možemo napisati sledeću nejednakost, cos α > cos β. Sledi da je jedna od vrednosti βi=30°, ugla β.

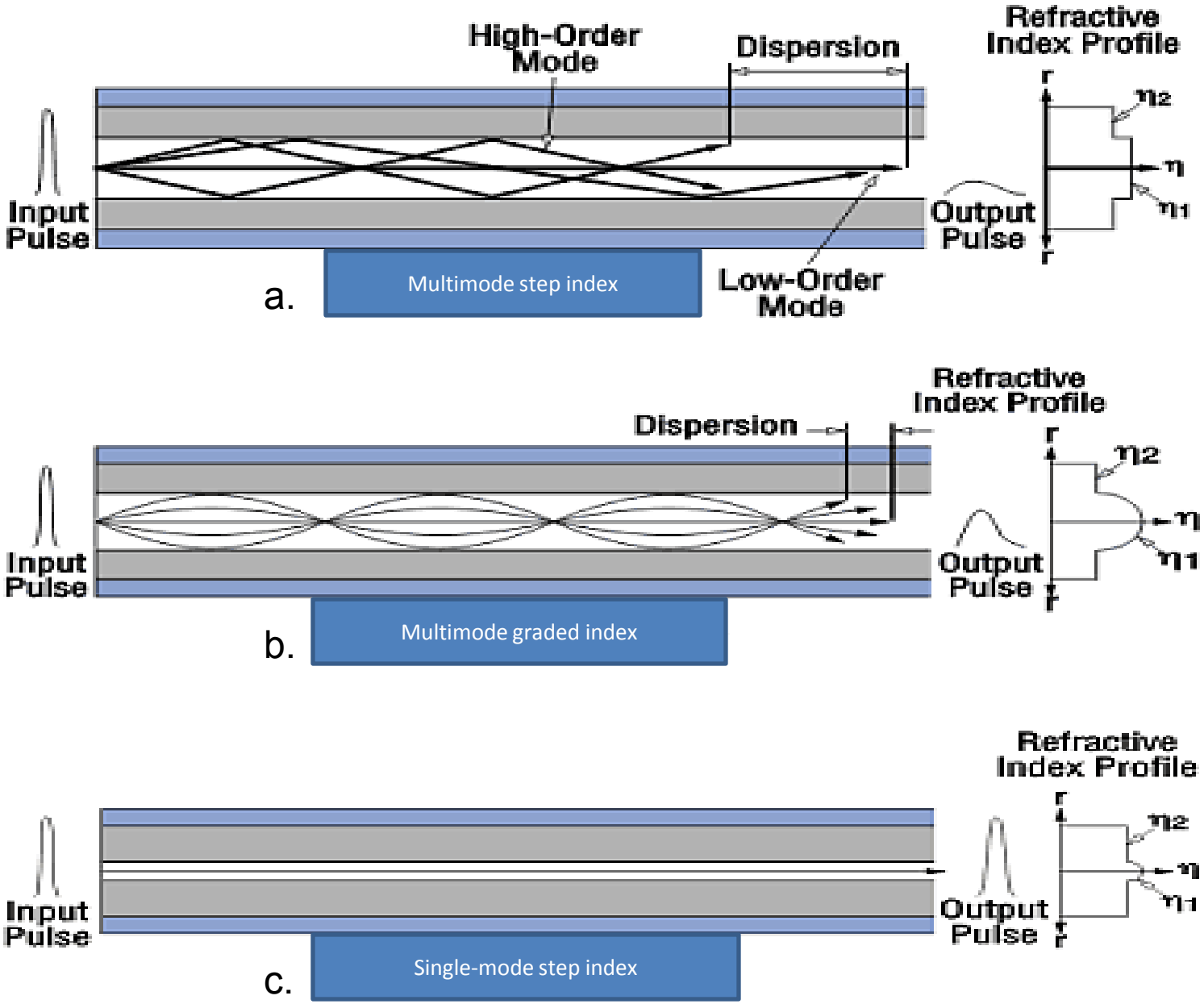
- Tri glavne komponente fiber opitičkog kabla su jezgro, obloga i zaštitni omotač. Jezgro je sastavljeno od čistog stakla ili plastičnog materijala. Obloga okružuje jezgro. I ona je od stakla, ili plastike, ali sa manjom optičkom gustinom od jezgra.
- Na jedan kraj fibera postavlja se svetlosni izvor, kao što su LED dioda ili laser. U pitanju je uredjaj koji reaguje na električni naboj stvarajući svetlosni impuls, koji se obično nalazi u intervalu od 10^{14}Hz. Svetlosni izvor generiše kratke, ali brze svetlosne impulse koji ulaze u jezgro pod različitim uglovima. Svetlost pogađa granicu jezgro/obloga pod uglom koji je manji od kritičnog ugla i potpunosti se reflektuje nazad u jezgro, gde uglavnom pogađa granicu na njegovoj drugoj strani. Kretanje svetlosti predstavlja skup odbijanja od granice do granice u procesu prostiranja kroz jezgro.
- Ako svetlost pogađa granicu pod uglom većim od kritičnog ugla deo svetlosti se refraktuje u oblozi te je apsorbuje zaštitni omotač, te se na taj način sprečava mogućnosti apsorbcije u susednim fiberima. Ako je jezgro prilično debelo u odnosu na talasnu dužinu svetlosti pogađa različita mesta pod različitim uglovima . Proučavanjem Maksvelovih jednačina ukazuje da dolazi do ometanja interference između nekih reflektovanih svetlosnih talasa. Zbog toga postoji određeni broj uglova pod kojim se zraci reflektuju i prostiru duž fibera. Svaki ugao definiše putanju odnosno mod.
    a. Fiber koji prenosi svetlost na ovaj način naziva se step-index mulitimode fiber. Svetlosti koje se reflektuje pod većim uglovima u odnosu na horizontalu reflektuje se češće i prelazi veća rastojanja od svetlosti koja se reflektuje pod manjim uglom, te joj je potrebno više vremena od jednog do drugog kraja fibera. Ovaj fenomen se naziva modalna disperzija.

Dati fenomen se naziva modalna disperzija. Svetlost koja se emituje iz jednog impulsa a koja se refektuje pod manjim uglovima može da zahvati svetlost iz prethodnih impulsa koja se reflektuje pod većim uglovima čime se eliminišu razmaci koji se stvaraju između njih. Ovaj fenomen definiše interferencu odnosno međusobno ometanje.

b. Jedno od načina rešenja datog problema je korišćenje graded-index multimode fibera. Ova tehnologija omogućava činjenica da svetlost brže putuje kroz medijum sa manjom optičkom gustinom. Graded index mulitimode fiber nema jasno definisanu granicu između jezgra i obloge. Dok se svetlost radijalno udaljava od jezgra materijalu se postepeno smanjuje optička gustina te se brzine prostiranja svetlosti izjednačavaju. Drugim rečim svetlost koja prolazi veća rastojanja se ubrzava, te na taj način izjednačavaju brzine propagacije svetlosnih zraka, odnosno minimizira se efekat modalne disperzije.

c. Postoji konačan broj modova za prostiranje svetlosnih zraka. Tačan broj zavisi od prečnika jezgra i talasne dužine svetlosnog zraka. Smanjivanjem prečinika jezgra smanjuje se broj uglova pod kojim svetlost "pogađa" granice jezgra tj. smanjuje se broj modova. Ako se smanji prečnik u dovoljnoj meri fiber će imati samo jedan mod. U tom slučaju imamo fiber samo sa jednim modom (single-mode fiber).

– U tekst boksevima na slikama odnosno sledećem slajdu, upisati naziv vrste (tip) optičkog fiber kabla.

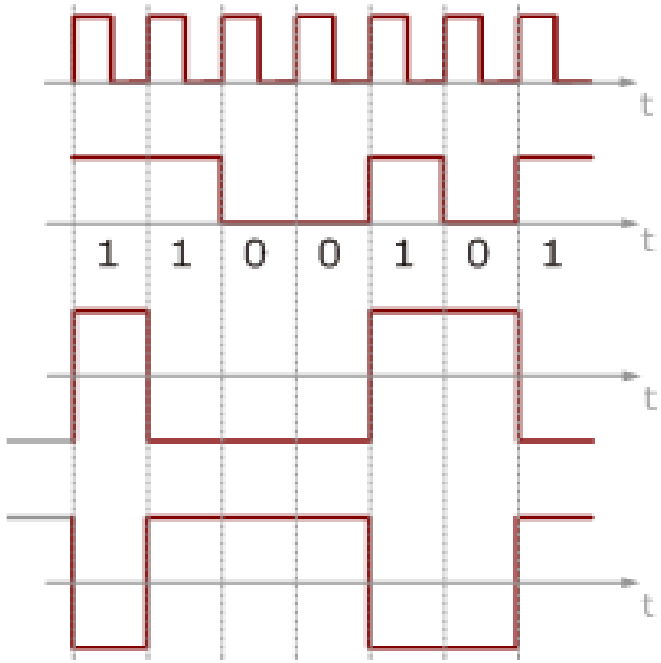

High-Order Mode
Dispersion
Refractive Index Profile
Input Pulse
Output Pulse
Low-Order Mode
Multimode step index
a.
Dispersion
Refractive Index Profile
Input Pulse
Output Pulse
Multimode graded index
b.
Refractive Index Profile
Input Pulse
Output Pulse
Single-mode step index
c.

4. Za date vrednosti periode T izračunati frekvenciju.

   - $f = 1 / T = 1 / (5\ s) = 0.2\ Hz$,
   - $f = 1 / T = 1 / (12 \times 10^{-6} s) = 83333\ Hz = 83.333 \times 10^{3} Hz = 83.333\ KHz$,
   - $f = 1 / T = 1 / (220 \times 10^{-9}\ s) = 4550000\ Hz = 4.55 \times 10^{6}\ Hz = 4.55\ MHz$.

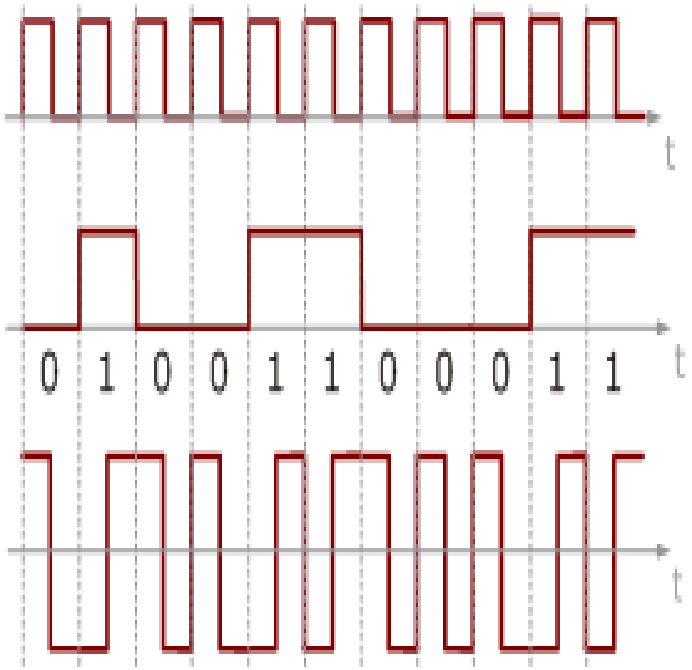
3. Najprostija šema za kodiranje je NRZ kodiranje. NRZI ne vraća se na 0 ako se bit prenosi . Nacrtati grafik za vrednost stringa u binarnom obliku: 1100101.

1. Sistemski sat
2. **Podatak**
3. **NRZI**
4. NRZI (opcija)



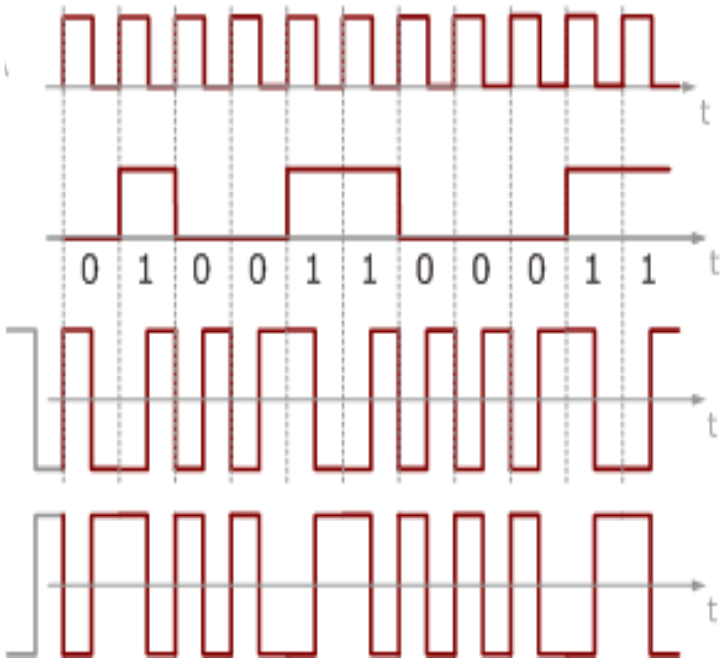
4. Mančester kodiranje definiše prenosom 0 pri promeni napona sa visoke na nisku vrednost, a prenos 1 vrši se obrnuto. Promena signala na polovini svakog intervala. Nacrtati grafik za vrednost stringa u binarnom obliku: 010011000011.

1. Sistemski sat
2. **Podatak**
3. **Mančester kodiranje**



5. Diferencijalno mančester kodiranje vrši promenu signala isto kao u prethodnom slučaju na polovini svakog signala, ali sa tom razlikom, što 1 zadržava signal na istom nivou na kome je bio na kraju prethodnog intervala. Nacrtati grafik za vrednost stringa u binarnom obliku: 010011000011.

1. Sistemski sat
2. **Podatak**
3. **Diferencijalno Mančester kodiranje**
4. Diferencijalno Mančester kodiranje (opcija)



6. Telefonski sistem ima opseg signala približno 3000Hz, a koeficijent signal-šum od oko 35dB, ili 3,5bela. Izračunajte bitsku brzinu.
   - Rešenje:

     3 ,5 = log10(S/N) odnosno S=1035 *N≈3612N sledi da je S/N jednako 3612.

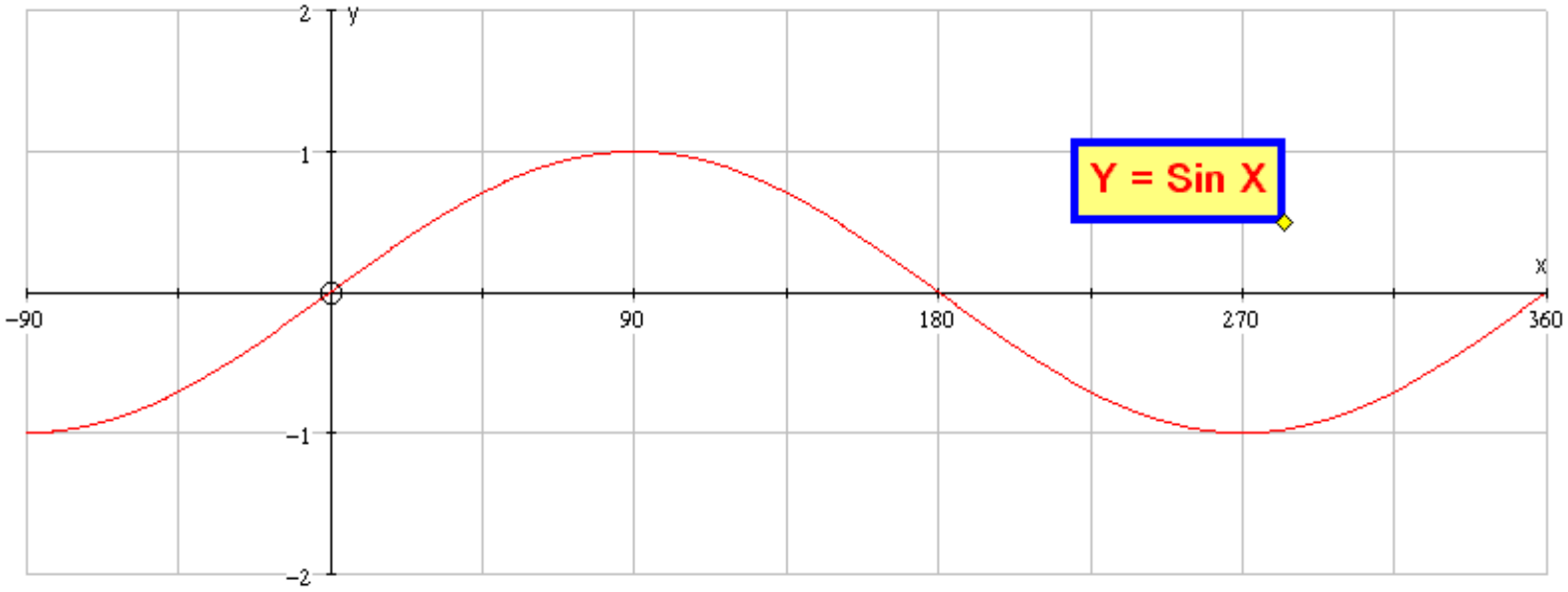
     Koristeći Šenonovu formulu, dobijamo:

     bitskabrzina = opseg signala x log(1+S/N)=

     = 3000*log2(1+3162)≈3000*11,63≈34880b/s

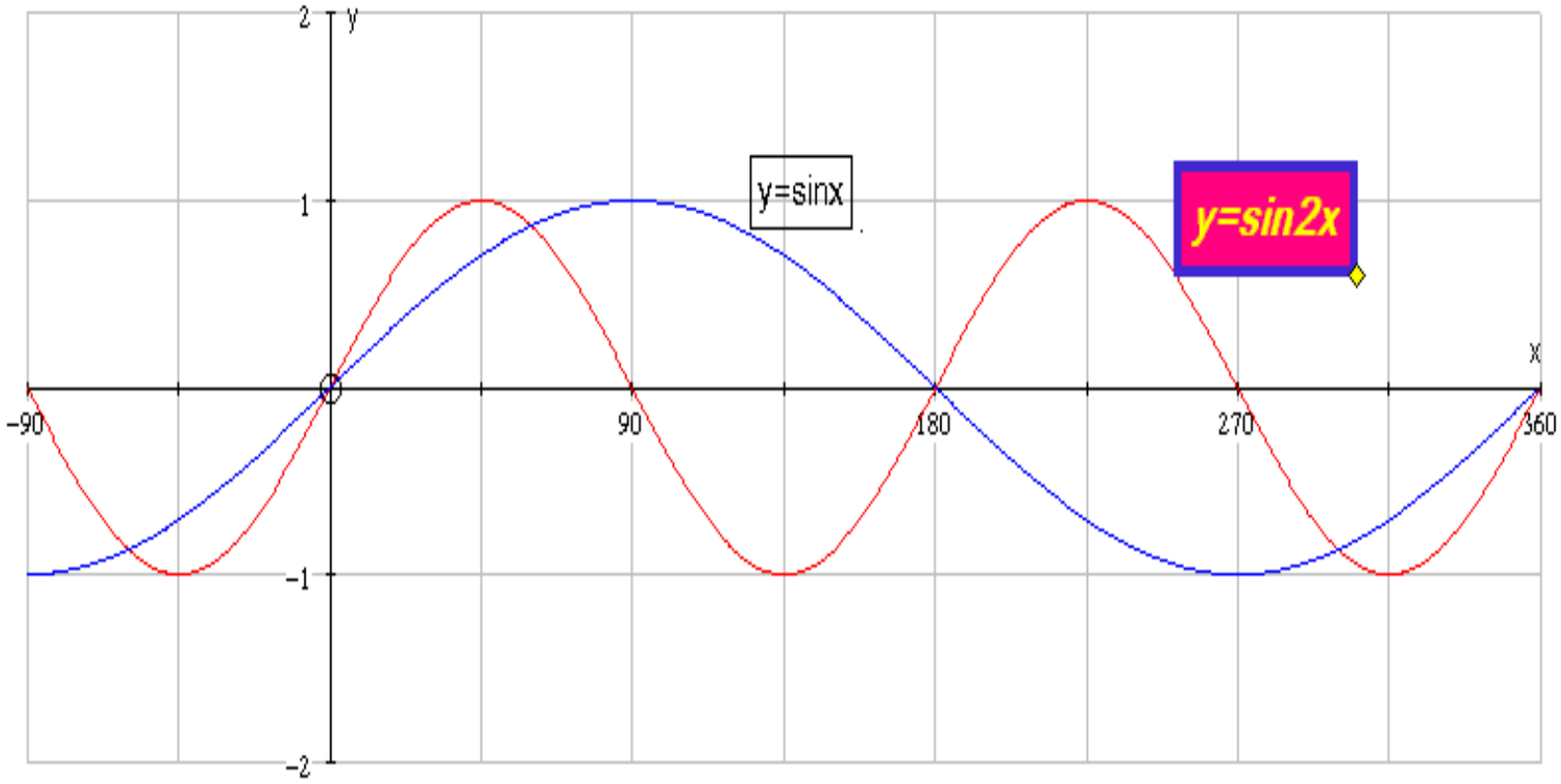
7. Nacrtajte sledeće grafike sinusnih funkcija. Koje predstavljaju matematičke modele signala u određenim uređajima mrežnih sistema.
   - $y = \sin x$,
   - $Y = 2\sin x$,
   - $Y = 5\sin 2x$,
   - $Y = \sin x + 1$,
   - $Y = -\sin x$,
   - $Y = 4\sin x$,
   - $Y = 4\sin 2x + 3$,
   - $Y = \sin(x - 60°)$.

Y = SIN X

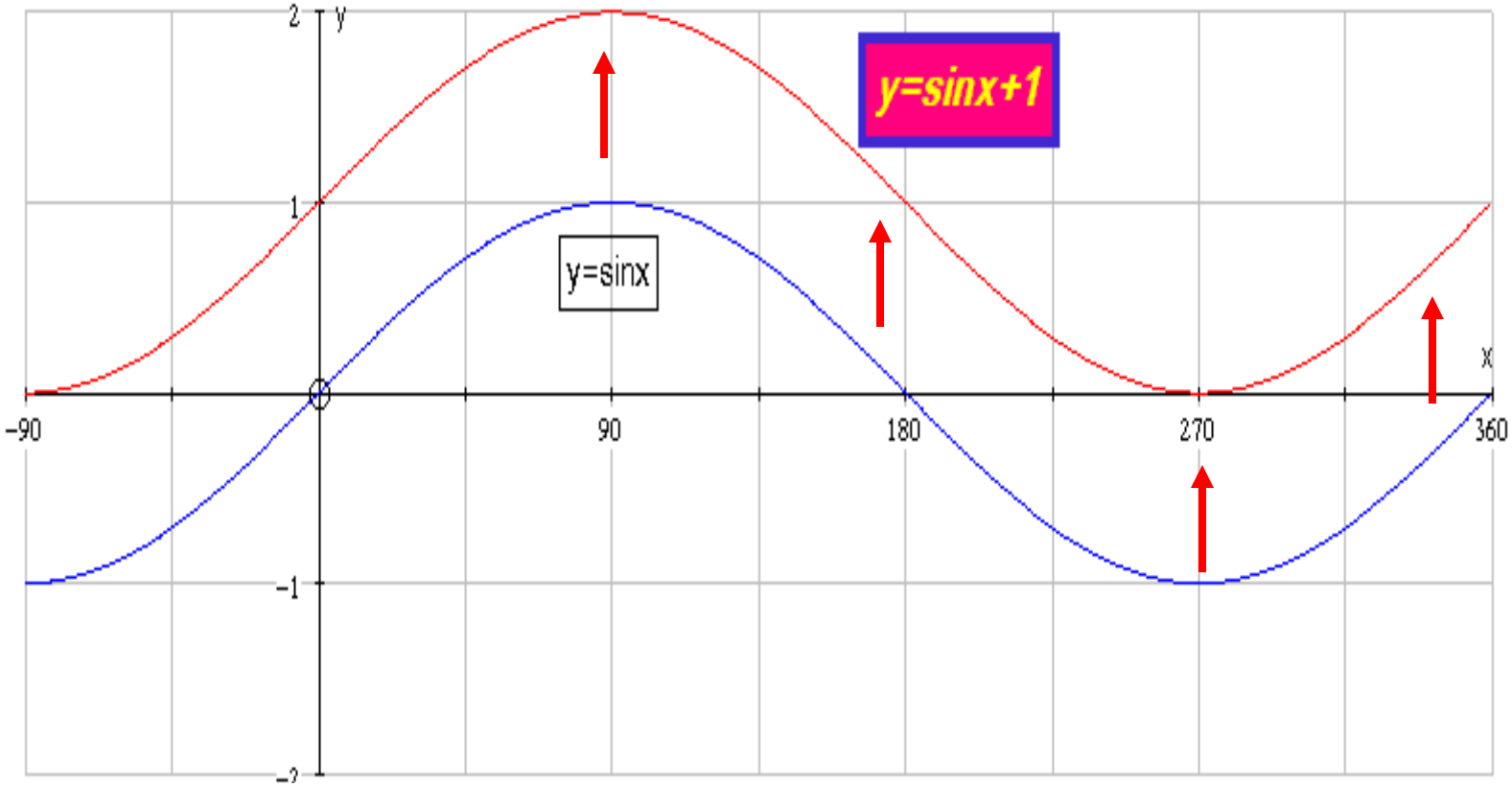


Y= 2 SIN X

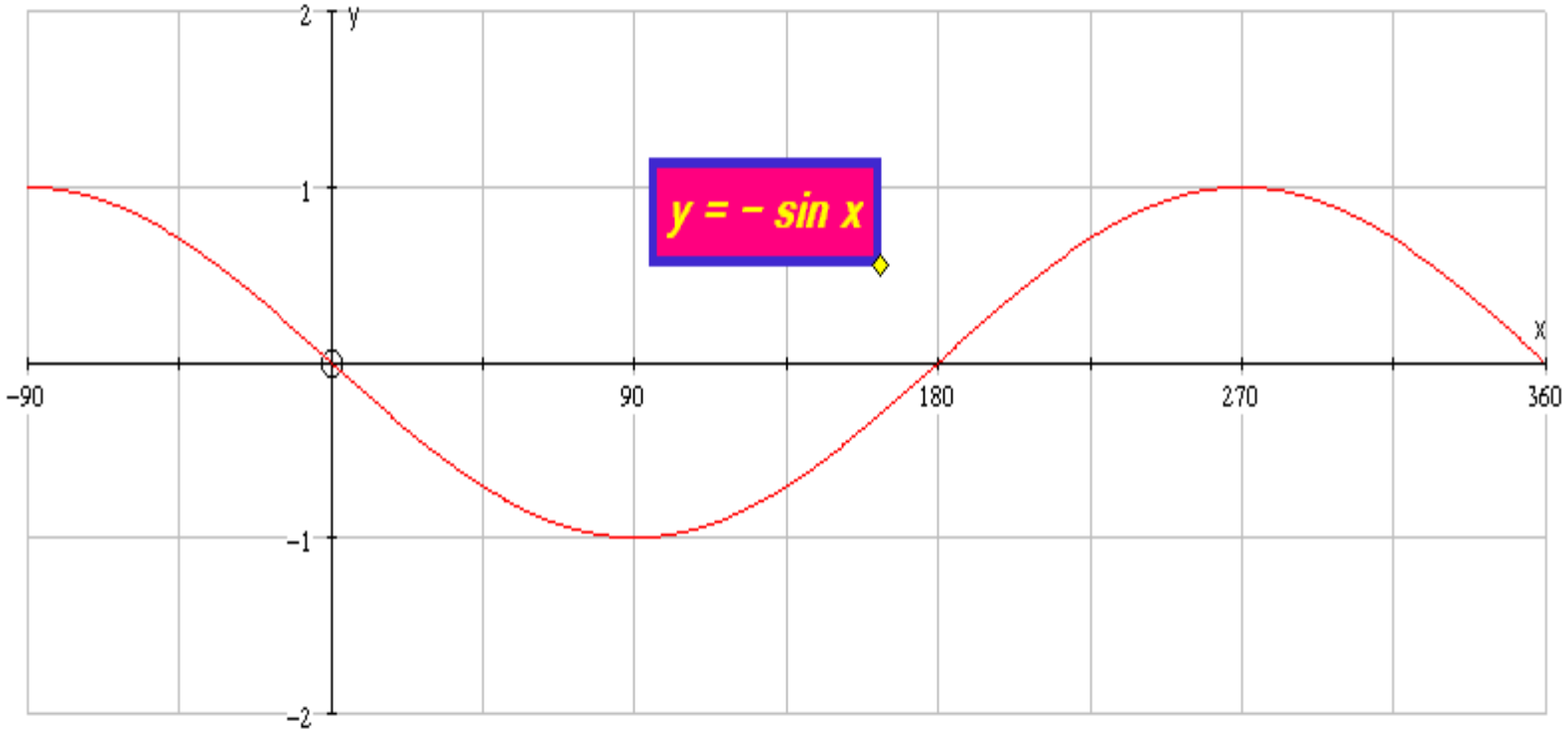
Y = SIN 2X

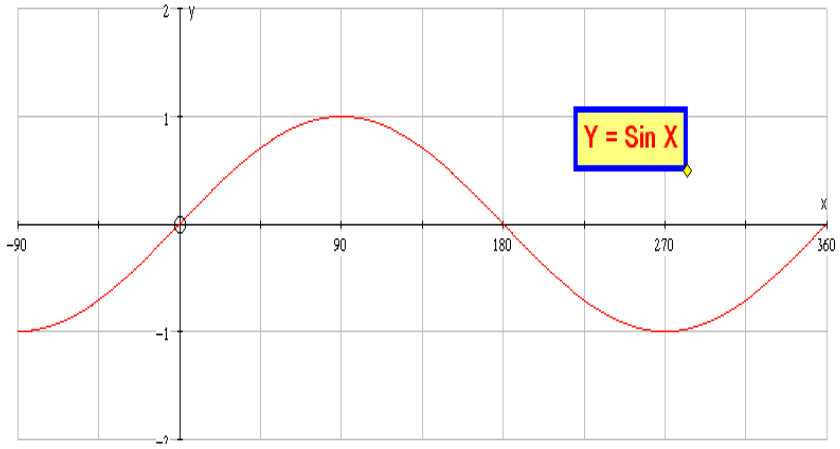


Y= SIN X + 1



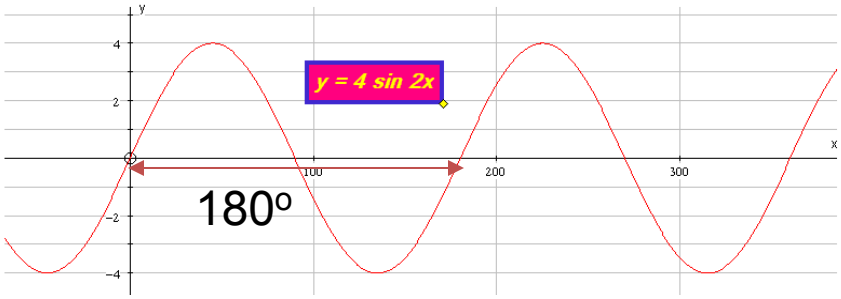
Y= – SIN X



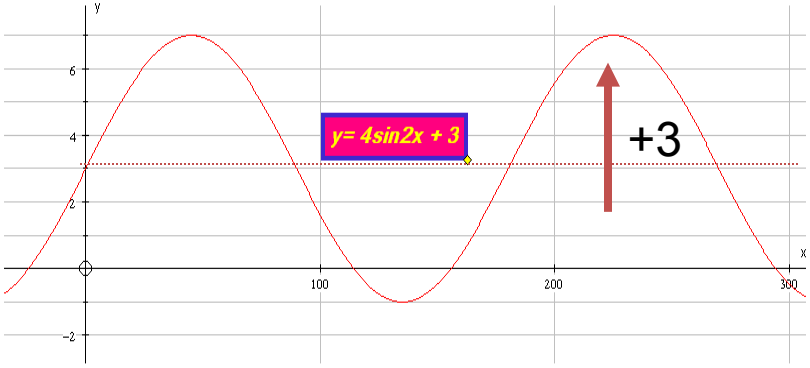


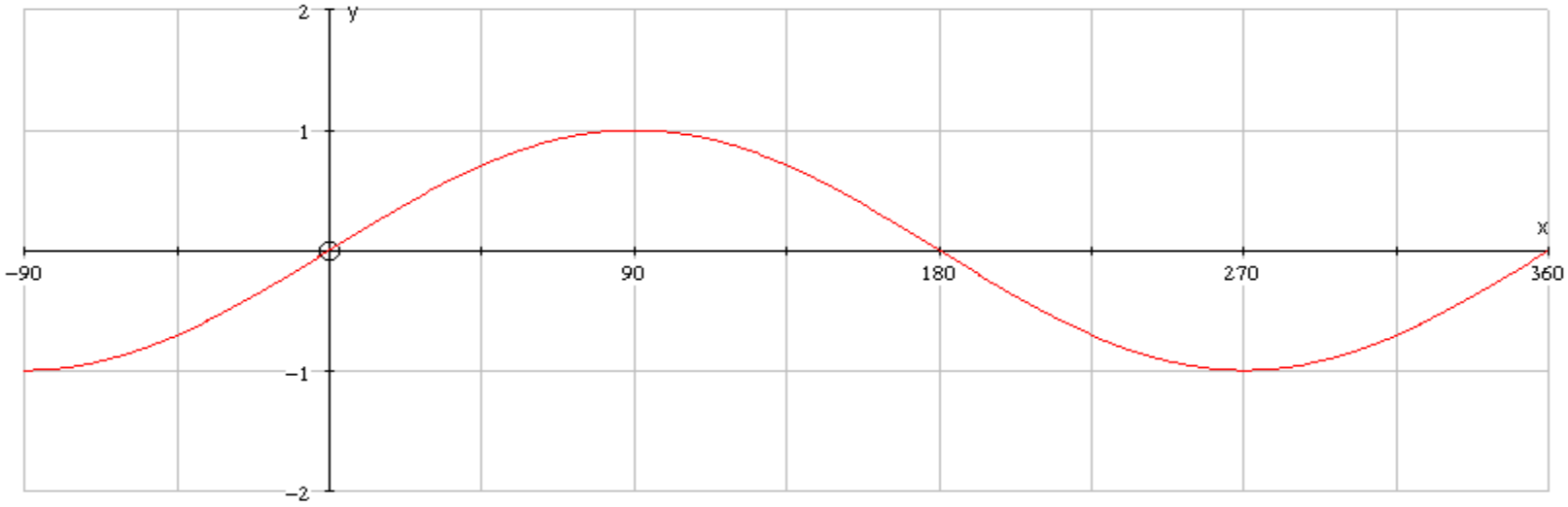
y = 4 sin x

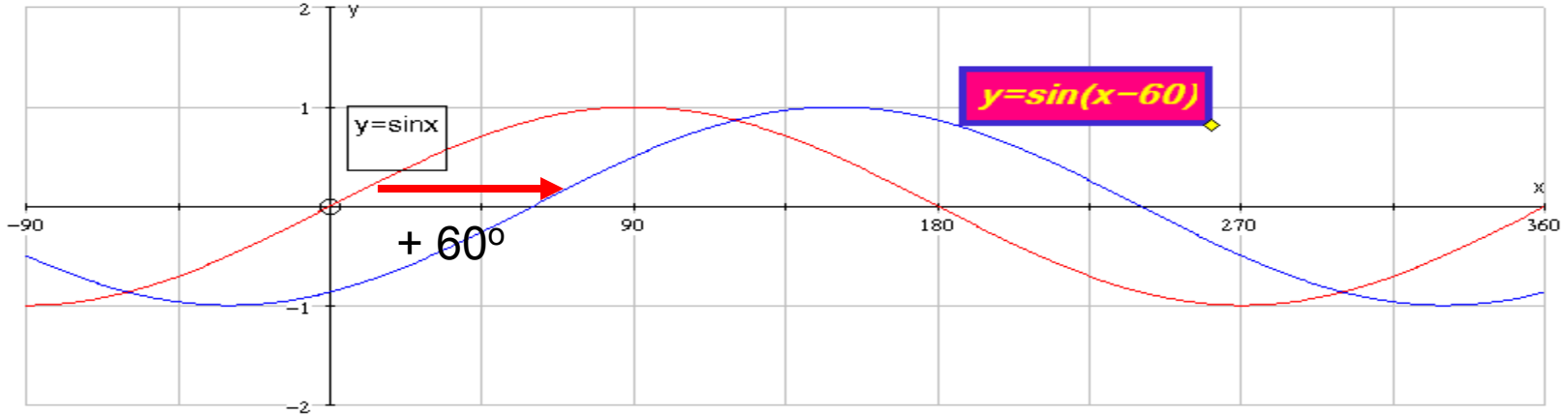
y = 4 sin 2x



y = 4sin 2x + 3

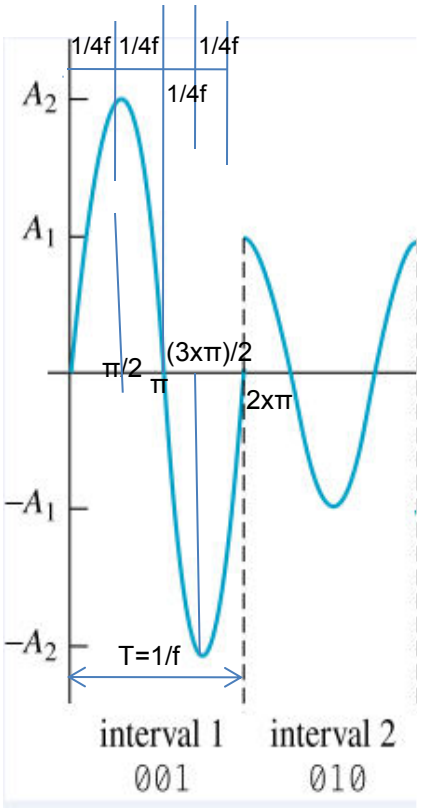



y
x
2
1
−1
−2
−90
90
180
270
360

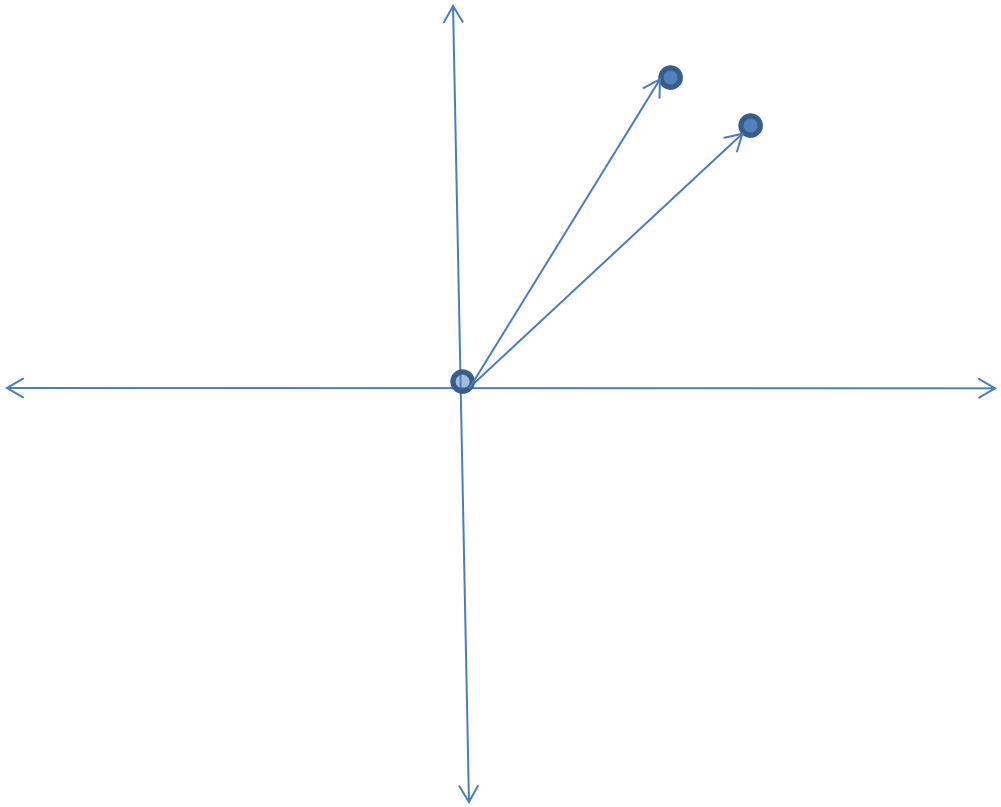

y
x
2
1
−1
−2
−90
90
180
270
360
y=sinx
+ 60°
y=sin(x−60)

8. Kvadratna amplitudska modulacija definiše grupe bitova dodeljuju signalu definisanim sa amplitudom i faznim pomakom. U tabeli na sledećem slajdu data je relacija između vrednosti stringova sa tri bita i signala. Date su amplitude A1 i A2 i fazni pomak 1/4f. (Gde je f=1/T sledi T=1/f). Nacrtajte grafik koji definiše stringove odnosno bitove na osnovu amplitude i faznog pomaka koriteći datu tabelu.

Bit Values	Amplitude of Generated Signal	Phase Shift of Generated Signal
000	A_1	0
001	A_2	0
010	A_1	$1/(4f)$


1/4f
1/4f
1/4f
1/4f
A_2
A_1
π/2
π
(3xπ)/2
2xπ
$-A_1$
$-A_2$
T=1/f
interval 1
001
interval 2
010

9. QAM kvadraturna amplitudska modulacija, mogu da se opišu vizuelno pomoću konstelacije signala, odnosno jedna tačka u koordinatonom sistemu je definisana sa udaljenošću (duž) u odnosu na koordinatni početak koji definiše amplitudu dok fazni pomak odgovara uglu kojeg zaklapa rastojanje odnonso data duž u odnosu na horizontalnu osu. Nacrtajte dati grafik konstalaciju sa dve tačke koje poseduju isto rastojanje u odnosu na koordinatni početak i sledeće uglove iz intervala koji definisan sa sledećom nejednakošću: $0°<\alpha<90°$.



10. Pretpostavimo da pet uređaja povezano na statistički multiplekser sa podelom vremena (generiše izlaze slično situaiciji u knjizi 4.31, stranica 184). Konstruišite okvire osnuvu izlaza koji multiplekser generiše odnosno šalje, (dati izlazi definisani su na slici na sledećem slejdu).

Ø

- UREDJAJ1: ...A3 / A2 A1 /
- UREDJAJ2: ...B4 B3 / B2 B1
- UREDJAJ3: ...C2 / / C1 /
- UREDJAJ4: ...D5 D4 D3 D2 D1
- UREDJAJ5: ... / E2 / E1 /

- Rešenje:

11. Noseći signal je definisan formulom g(t)=f1(t) i signal koji se prenosi definisan je funkcijom s(t)=f2(t). Date funkcije su sinusne funkcije. Definisati modulisani signal.
    - Rešenje:
    - Množenjem posmatranih sinusnih funkcija dobijamo rezultujući modulisani signal r(t)=g(t) x s(t).

11. Pretpostavimo da prenos čekaju dva uređaja koja koriste 0.5 – perzistentni protokol. Kada je medijum neaktivanj, svaki inicira prenos sa verovatnoćom od 0.5 odnos no 50%. Definisati moguće događaje. (Perzistentni protokol CSMA inicira prenos nakon provere da li ima saobraćaja u meži sa verovatnoćom p(0<=p<=1). Neperzistenti čeka jedan vremenski slot te onda proverava ako nema saobraćaja šalje.)

    Rešenje:

    – Sva četiri događaja umaju jednaku verotnoću dešavanja.
        1. Oba uređaja odmah iniciraju prenos.
        2. Oba uređaja čekaju.
        3. Prvi uređaj šalje okvir a drugi uređaj čeka.
        4. Drugi uređaj šalje okvir a prvi uređaj čeka.

KOMPRESIJA

RAČUNARSKE MREŽE I WEB PROGRAMIRANJE

IV- GODINA

prof. dr Zlatko Langović

William A. Shay

- Textbook does not really deal with compression.
- This pptx file outlines the material from the book *Understanding Data Communications and Networks*, which is on reserve.

- Compression
  - will make a large file into a smaller file.
  - It typically works by reducing redundancy or repetition in the data
  - Need to also be able to change the smaller file back into the larger one (or to at least something very close)

- Lossless compression
  - No information is lost.
  - Decompressing the smaller file results in an exact copy of the file that was compressed
- lossy compression:
  - Some information is lost.
  - Decompressing the smaller file creates an approximation to the file that was compressed

- When is it OK to lose information in a file?
- When is it not OK?

Compression

- Different categories of repetition.
  - Common characters
  - common strings
  - Runs (sequences of 0s, 1s, or a single character)
  - temporal redundancy (similar data over time)
  - Combinations of these

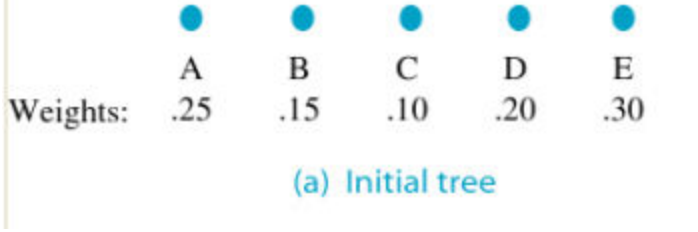
Huffman code

- frequency dependent
- Short binary codes to common characters; longer binary codes to less common ones.
- Used, in part, in image files and fax machines.
- No-Prefix property (no code exists as a prefix of another)

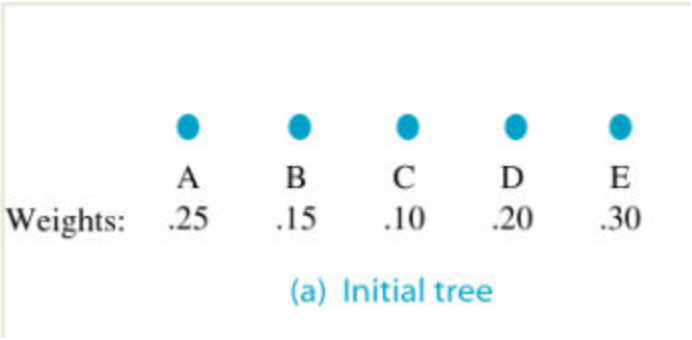
Huffman algorithm outline

- For each character, create a binary tree consisting of just one node.
- For each tree define the tree's *weight as* the character's frequency of appearance in the text.
- Look for the two lightest-weight trees. If there are more than two, choose any two.
- Merge the two into a single tree with a new root node whose left and right subtrees are the two we chose.

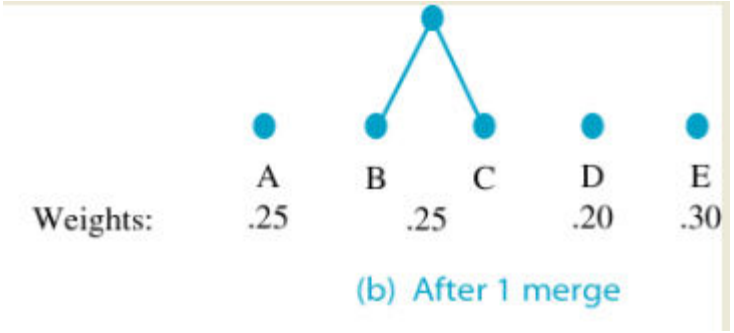
- Define the weight of the new tree to be the sum of weights of the merged trees.
- Repeat the previous steps until just one tree is left.
- A character’s code is defined by the path from the root to the node, which is always a leaf node (has no subtrees).



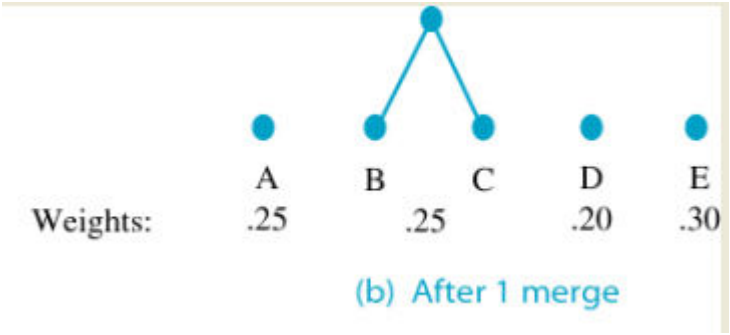
(a) Initial tree



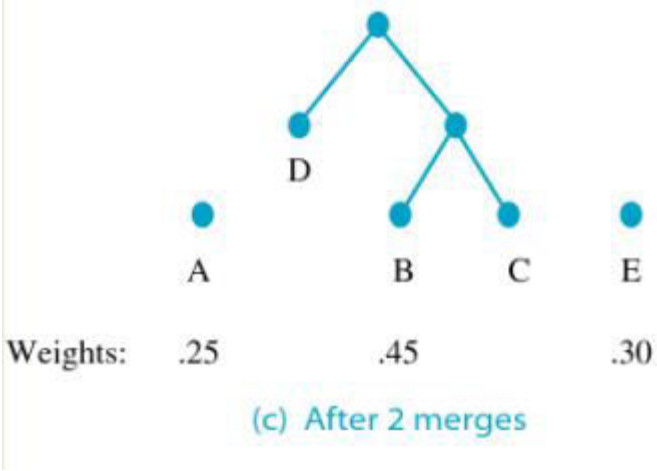
(a) Initial tree



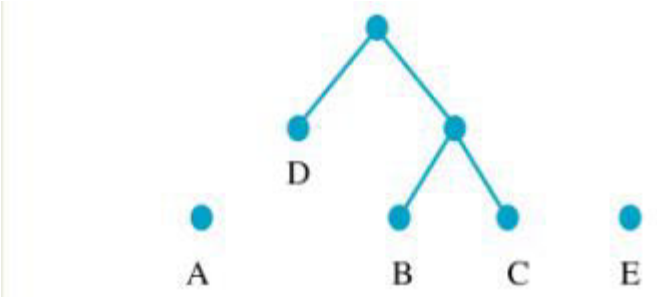
(b) After 1 merge



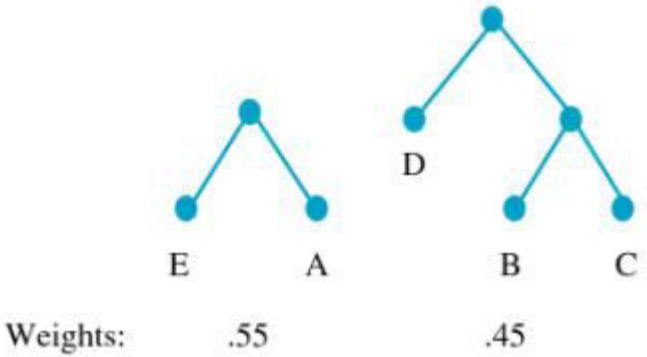
(b) After 1 merge



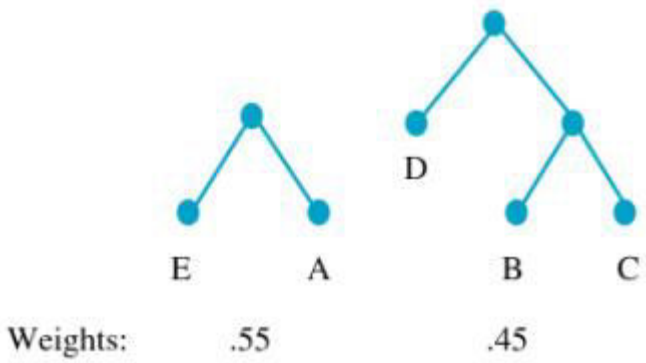
(c) After 2 merges



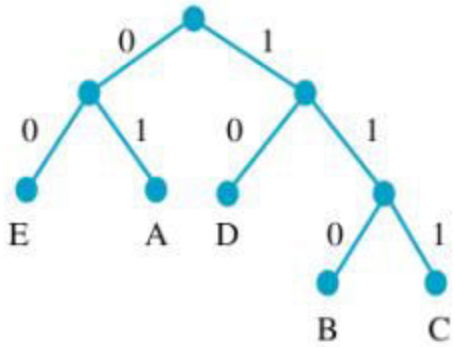
(c) After 2 merges



(d) After 3 merges

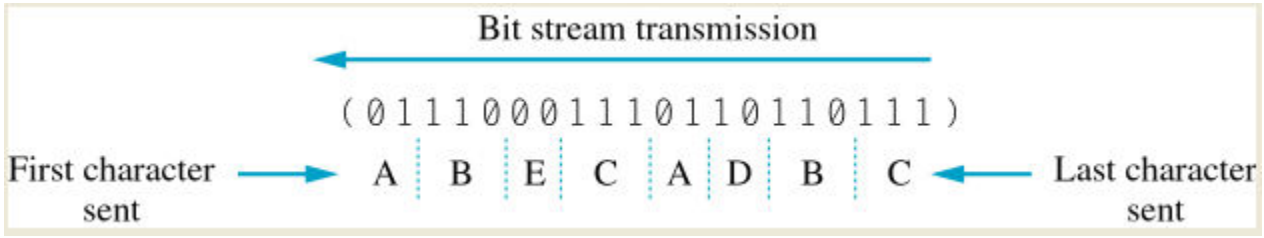


(d) After 3 merges



(e) After 4 merges

- A:01
- B:110
- C:111
- D:10
- E: 00



- A:01
- B:110
- C:111
- D:10
- E: 00

- Why is the no-prefix property important?

- Applet:

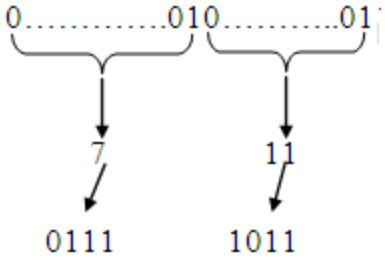
- Arithmetic encoding: Also frequency based. Type of Redundancy: some characters appear more frequently.

Letter	Frequency	subinterval [p, q]
A	25%	[0, 0.25]
B	15%	[0.25, 0.40]
C	10%	[0.40, 0.50]
D	20%	[0.50, 0.70]
E	30%	[0.70, 1.00]

- Start with the interval [x, y] = [0, 1].
- Look at the first character and determine the appropriate subinterval of [x, y] based on that character's probability.
- Redefine the interval [x, y] to be that subinterval. That is, shrink [x, y].
- Examine the next character and again determine the appropriate subinterval of [x, y] depending on that character's probability. This is exactly what we did in step 2, except now we are working with an arbitrary interval [x, y], instead of [0, 1].
- Repeat steps 3 and 4 for each character in the string.

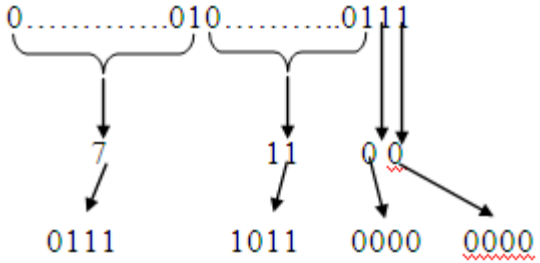
- Run Length
  - usedful in fax compression
  - Used in windows bitmap compression
- can get compression ratios of 20:1 or 30:1.
- Type of Redundancy:
  - string consisting of a single repeated character or bit (we will assume for this example that we are compressing runs of 0s, each of which is separated by a 1).
- EX. 4 bits used to represent # of 0-bits in a run (obviously can use more but keeps the example manageable)

- Store run lengths as opposed to the actual runs



- Above string containing 20 bits is replaced by 8 bits: 0111 1011

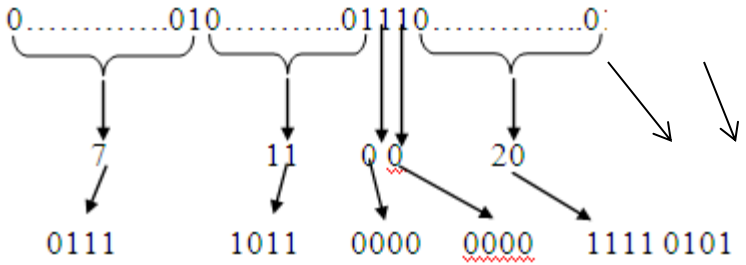
- What if you have consecutive 1s?



- Compression is less effective.
- Compression algorithms must be used on files that contain the type of redundancy the algorithm is designed to exploit.
- If not, compression is less effective, and the algorithm could increase the file size.

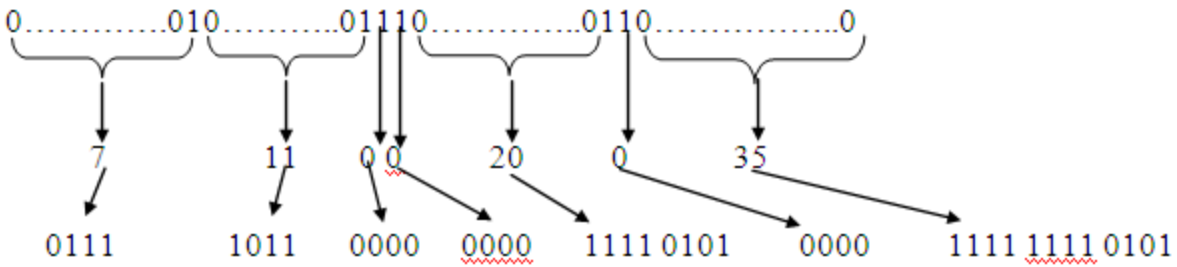
What if you have more than fifteen 0s in a run?

- Consider a run of twenty 0s
- 20 = 15 + 5 (in binary: 1111 followed by 0101)



20 = 15 + 5

- How about a run of thirty five 0s?
- 35 = 15 + 15 + 5



- How do I differentiate two consecutive runs of 15 from one run of 30?

Fax compression

- sometimes called modified Huffman: frequencies of run lengths [http://en.wikipedia.org/wiki/fax]
  - Each line consists of alternating runs of white and black pixels.
  - Each line begins with a run of white pixels. Even if the page to be faxed has a black border, a process call overscanning adds one white pixel to the beginning and end of each line.
  - Calculate codes for the alternating white and black pixel runs and transmit the coded bits: See table in next slide.

Partial fax compression table

	Number of Pixels in Run	Code: White Pixel Run	Code: Black Pixel Run
Terminating Codes	0	00110101	0000110111
	1	000111	010
	2	0111	11
	3	1000	10
	10	00111	0000100
	20	0001000	00001101000
	30	00000011	000001101000
	40	00101001	000001101100
	50	01010011	000001010010
	60	01001011	000000101100
Makeup Codes	64	11011	0000001111
	128	10010	000011001000
	256	0110111	000001011011
	512	01100101	0000001101100
	768	011001101	000000101100
	1024	011010101	0000001110100
	1280	011011001	0000001010010
	1536	010011001	0000001011010

Example

- run length of 50 white pixels is coded as 01010011.
- run length of 572 white pixels - interpreted as a run of length 512 pixels followed by another run of 60 pixels.
- associated code is 01100101–01001011

Relative encoding:

- send differences between consecutive frames
- Useful in Video (mpeg) where differences in consecutive images are not large
- Can also be used in fax (often, successive lines are similar)
- Type of Redundancy: similarities in successive frames.

Lempel-Ziv:

- replace strings with short code words
- Unix compress and gzip commands
- winzip program.
- Type of Redundancy: frequently appearing strings.
- I will illustrate with text but it need not be strictly text files as the algorithm looks for any repeating sequences, text or not.

Basic logic

- Assume basic code for individual characters (bytes)
- Read through string
- Look for and track (remember) repeated sequences.
- Assign codes to those sequences and store in a dictionary.
- Use those codes when sequence reappears.

COMPRESSION

```
1    void compress (FILE * fileid)
2    {
3      initialize(code table);
4      buffer=string consisting
        of first character from the file.
5      while ( (c=getc(fileid)) != EOF)
6      {
7        tempstring=concat(buffer, c);
8        search for tempstring in the
          code table;
9        if found
10       buffer = tempstring;
11       else
12       {

13         send the code associated
           with buffer;
14         assign a code to tempstring;
            store both in the code table;
15         buffer=string consisting of
            one character c;
16       }
17     } //while loop
18     send the code associated
        with buffer;
19   } //compress
```

Compress the string: ABABABCBABABABCBABAB

	Buffer	c	What is stored	What is Stored in dictionary	New Buffer Value
1	A	B	0 (code for A)	AB (code=3)	B
2	B	A	1 (code for B)	BA (code=4)	A
3	A	B	—	—	AB
4	AB	A	3 (code for AB)	ABA(code=5)	A
5	A	B	—	—	AB
6	AB	C	3 (code for AB)	ABC (code=6)	C
7	C	B	2 (code for C)	CB (code=7)	B
8	B	A	—	—	BA
9	BA	B	4 (code for BA)	BAB (code=8)	B
10	B	A	—	—	BA
11	BA	B	—	—	BAB
12	BAB	A	8 (code for BAB)	BABA (code=9)	A
13	A	B	—	—	AB
14	AB	C	—	—	ABC
15	ABC	B	6 (code for ABC)	ABCB (code=10)	B
16	B	A	—	—	BA
17	BA	B	—	—	BAB
18	BAB	A	—	—	BABA
19	BABA	B	9 (code for BABA)	BABAB (code=11)	B

The dictionary

String	A	B	C	AB	BA	ABA	ABC	CB	BAB	BABA	ABCB	BABAB	BABC	CBA
Code	0	1	2	3	4	5	6	7	8	9	10	11	12	13

```
1     void decompress
2     {
3       initialize(code table);
4       receive first code, call it prior;

5       print the string associated with prior;
6       while (true)
7       {
8         receive code, call it current; if no
           code then break;
9         search for current in the code table;
10        if not found
11        {
12           c=1st character of string
              associated with prior;
13           tempstring=concat(string
              associated with prior, c);
14           assign a code to tempstring;
              store both in the code table;
;
15           print tempstring;

16        }
17        else
18      {

19           c=1st character of string
              associated with current;
20           tempstring=concat(string
              associated with prior, c);
21           assign a code to tempstring;
              store both in the code table
22           print string associated with
              current
23           }
24      prior=current
```

Decompress the code: 0 1 3 3 2 4 8 6 9 8 7 0

	Prior (String)	Current (String)	Is Current Code in Table?	C	Tempstring/Code Pair	What is restored (current or tempstring)
1	0 (A)	1 (B)	Yes	B	AB/3	B (current)
2	1 (B)	3 (AB)	Yes	A	BA/4	AB (current)
3	3 (AB)	3 (AB)	Yes	A	ABA/5	AB (current)
4	3 (AB)	2 (C)	Yes	C	ABC/6	C (current)
5	2 (C)	4 (BA)	Yes	B	CB/7	BA (current)
6	4 (BA)	8	No	B	BAB/8	BAB (tempstring)
7	8 (BAB)	6 (ABC)	Yes	A	BABA/9	ABC (current)
8	6 (ABC)	9 (BABA)	Yes	B	ABCB/10	BABA (current)
9	9 (BABA)	8 (BAB)	Yes	B	BABAB/11	BAB (current)
10	8 (BAB)	7 (CB)	Yes	C	BABC/12	CB (current)
11	7 (CB)	0 (A)	Yes	A	CBA/13	A (current)

- NOTE on Lempel-Ziv Decompression algorithm:
- Algorithm may receive a code not in its table.
- Happens if compression algorithm sends it immediately after storing in it its table (see example).
- Receiver does not get a chance to reconstruct this code. How can this occur?

- NOTE: Every new code’s string begins with the last character of a previously stored code’s string.
- The string for the code just stored appears again immediately. In the book example, 8 corresponds to BAB and BAB occurs again after storing the previous BAB.
- Thus, the code‘s string starts and ends with the same character.

- Conclusion: the code's string was created from prior code's string with its first character concatenated to the end.
- If some other character was at the end, the string could not immediately appear again since the next string would start differently than the previous string.
- Applet

- Issue: How table is created and accessed – need quick lookup (i.e. hashing).
- Also don’t want to store all strings redundantly in the table. If ABABA is stored at 9, might store ABABAB as 9B.

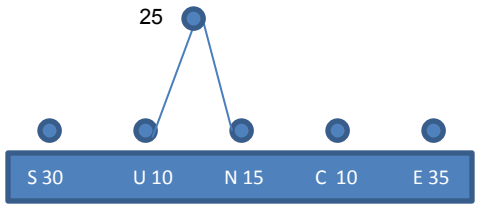
- JPEG. Type of Redundancy: small variance in colors.
- MPEG. Type of Redundancy: similarities in successive frames.
- MP3. Type of Redundancy: Signal components that are masked by other, more powerful signals or sounds that are nearly indistinguishable

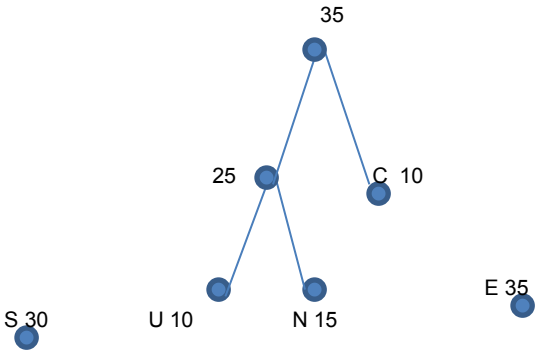
Compression Technique	Type of redundancy exploited	How It Compresses
Huffman code	Certain characters appear more frequently than others.	Uses short bit patterns for more frequently used letters and longer ones for less frequently used letters.
Run-length encoding	Data contains long strings of the same character or bit.	Looks for long runs of a particular bit or character.
Facsimile Compression	Looks for both long strings of the same bit and the frequency with which specific strings appear.	Divides a line of pixels into alternating runs of white and black pixels. Each run is encoded using a modified Huffman algorithm.
Relative encoding	Two consecutive pieces of data differ by very little.	Looks for small differences between consecutive frames.
Lempel-Ziv encoding	Certain character strings appear more frequently than others.	Looks for repeated occurrences of strings without assuming what those strings are.
JPEG	Small subsets of pictures often contain little detail.	Compresses still images by applying discrete cosine transforms to 8×8 blocks of pixels, quantizing the results, and encoding the quantized frequency coefficients.
MPEG	Consecutive frames often contain nearly identical scenes.	Uses methods similar to JPEG compression but also takes advantage of redundancy between successive frames to use interframe compression by calculating differences between successive frames and using motion prediction techniques.
MP3 (alternatively MPEG layer 3 audio compression)	Signal components that are masked by other, more powerful signals.	Uses complex psychoacoustic models and filter banks to determine which parts of an audio signal will be inaudible and seeks to remove those parts.

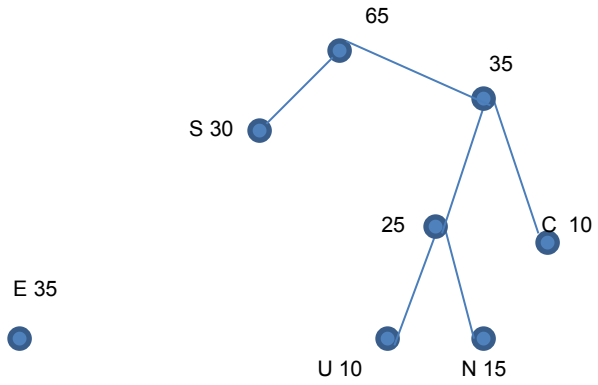
Zadaci-2:

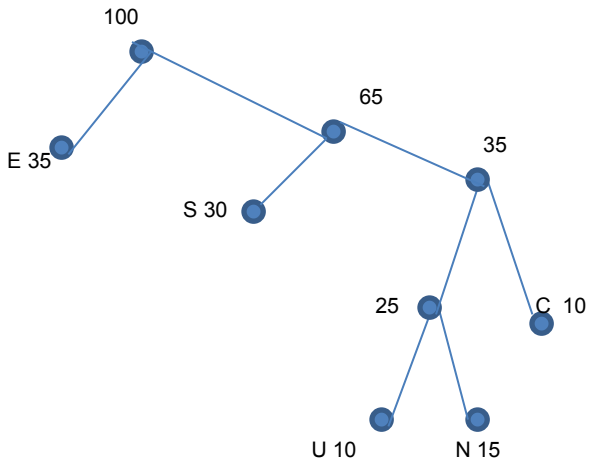
12. Koristeći Hafmanov kod kodirajte poruku.

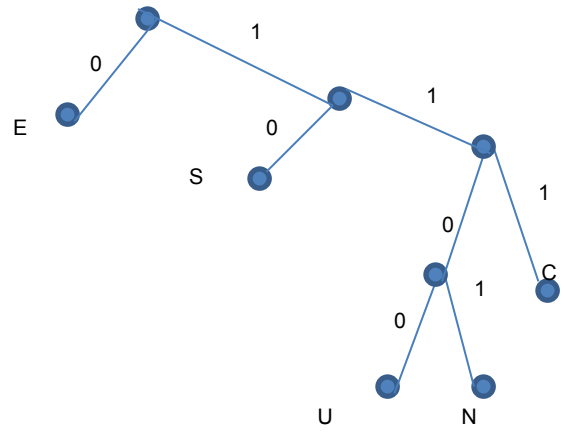
- Kod jednog karaktera ne može nikad da se koristi kao prefiks drugog koda dato prvilo je poznato kao no-prefix svojstvo.
- Svako karakteru dodeljuje se binarno stablo koje se sastoji iz jednog čvora. Svakom stablu je dodeljena učestalost pojavljivanja karaktera, koju nazivamo težina stabla (tree's weight)
- Potražite dva najlakša stabla. Ako ih ima više od dva izaberite bilo koja od data dva stabla. Spojite ih u jedno sa novim korenom, čije levo i desno podstablo odgovara ranijim stablima. Dodelite sumu težina spojenih stabala kao sumu novog stabla.
- Ponovite sledeći korak sve dok ne budete imali samo jedno stablo.
- Koristeći Hafmanov kod kodirajte istu poruku, sa drugačijim rasporedom karaktera u datom grafu.


25
S 30
U 10
N 15
C 10
E 35


35
25
C 10
S 30
U 10
N 15
E 35


65
35
S 30
25
C 10
E 35
U 10
N 15


100
65
E 35
35
S 30
25
C 10
U 10
N 15



Na osnovu date grafičke interpretacije imamo sledeću strukturu Hafmanovog koda.
S-10
U-01
N-1100
C-111
E-0
S | U | N | C | E | S | U | N; Gde je S prvi poslati karater a N je poslednji poslati karkter.
10|01|1100|111|0|10|01|1100|1101

13. Primenite aritmetičku kompresiju koja interpretira string, kao jedan realni broj, na string SUNCESUN.
    - Na osnovu tabele na sledećem slajdu imamo da slovo A ima učestalost pojavljivanja 25% i dodeljujemo mu interval [0, 0.3] odnosno predstavalj 30% intervala od 0 do 1 te je dužina 0,3 definisana intervalom [0, 0.3]. Zatim slovo B ima učestalost 10% dodeljujemo mu sledećeih 10% intervala nakon podintervala [0, 0.3] te dobijamo [0.3, 0.4]. pošto C ima učestalost pojavljivanja 15% dodeljuje mu se 15% od intervala od 0 do 1. Datih 15% posmatranog intervala što odgovara podintervalu [0.4, 0.55] dužine 10 odsto. Prateći dati šablon dodeljujemo slovu D 10% [0.55,0.65]a preostalih 35% slovu E i na kraju [0.65,1]. Ukupna suma podintervala je 100 odsto tj. 1.
    - Startujte od intervala [x, y] = [0, 1]
    - Pogledajte prvi karakter i utvrdite odgovarajući podinterval od [x, y], koji se zasniva na učestalosti karakatera.
    - Redifinišite interval [x, y] koji će predstavljati taj podinterval, tj. smanjite [x, y]
    - Ispitajte sledeći karakter i ponovite utvrdite odgovarajući podinterval [x, y], u zavisnosti od učestalosti karaktera. Ovo je identično onome što ste uradili u koraku 2. Osim što sada radite sa manjim intervalom [x, y], umesto [0, 1]
    - Ponavljajte korake 3 i 4 za svaki karakter u stringu,
    - Uradite dekompresiju (dodatna objašnjenja-aritmetička kompresija, strana: 220).

Slovo	Učestalost (%)	Podinterval [p, q]
S	30	[0,0.3]
U	10	[0.3,0.4]
N	15	[0.4,0.55]
C	10	[0.55,0.65]
E	35	[0.65,1]

KORAK	STRING	SLEDEĆI KARAKTER	TEKUĆI INTERVAL [x, y]	[p,q]	ŠIRINA INTERVALA (w=y-x)	IZRAČINAVANJE ZA NOVO x (x = x + w * p)	IZRAČUNVANJE ZA NOVO y (y = x + w * q)
1	-	S	[0,1]	[0, 0.3]	1	0+0.3x0=0	0+1x0.3=0.3
2	S	U	[0, 0.3]	[0.3,0.4]	0.3	0 + 0.3x0.3=0.09	0+0.3x0.4=0.12
3	SU	N	[0.09, 0.12]	[0.4,0.55]	0.03	0.09+0.03x0.4= 0.102	0.09+0.03x0.55 =0.105
4	SUN	C	[0.102, 0.105]	[0.55,0.65]	0.003	0.102+0.003x0. 55=0.10365	0.102+0.003x0. 65=0.10395
5	SUNC	E	[0.10365, 0.10395]	[0.65, 1]	0.0003	0.10365+0.0003 x0.65=0.103845	0.10365+0.0003 x1=0.10395
6	SUNCE	S	[0.103845, 0.10395]	[0,0.3]	0.000105	0.103845+0.000 105x0=0.10384 5	0.103845+0.000 105x0.3=0.1038 765
7	SUNCES	U	[0.103845, 0.1038765]	[0.3,0.4]	0.0000315	0.103845+0.000 0315x0.3=0.103 85445	0.103845+0.000 0315x0.4=0.103 8576

14. Česta je primena relativnog kodiranja je u prenosu video signala. Prosečno video signal šalje 30 slika u sekundi. Svaka slika u opštem slučaju ima samo manje varijacije u poređenju sa prethodnom. Princip je jednostavan. Šalje se prvi kadar koji se smešta u bafer prijemnika. Nakon toga pošiljalac poredi prvi kadar sa drugim, kodira razlike i šalje ih u formatu okvira. Prijemnik dobija ovkir i primenjuje razlike na kadar koji ima i tako se kreira sledeći kadar. Drugi kadar se smešta u bafer, pa se proces nastavlja za svaki sledeći kadar. Date su dve matrice koje definišu dva okvira. Nađite razlike i date razlike definišite sa 0 (nema promena) i na pimer 1 (nastale su promene).

- 5 7 2 8 6 6 3 5 6
- 6 5 7 4 5 6 3 2 1
- 8 4 8 9 6 4 8 8 5
- 5 1 3 8 6 5 5 6 7
- 5 5 2 9 9 6 8 9 1

Prvi okvir

- 5 7 2 8 6 6 3 5 6
- 6 5 **8** 4 5 **5** 3 2 1
- 8 4 8 9 6 4 8 8 5
- 5 **2** 3 8 6 5 5 **5** 7
- 5 5 2 9 9 6 8 9 1

Drugi okvir

- Preneti okvir sadrži kodirane razlike između prvog i drugog okvira

- 0 0 0 0 0 0 0 0
- 0 0 **1** 0 0 **1** 0 0
- 0 0 0 0 0 0 0 0
- 0 **1** 0 0 0 0 **1** 0
- 0 0 0 0 0 0 0 0

15. Kompresujte tekst koji se sastoji od tri karakterra A, C i R, primenom Lemp Ziv kompresije. Dati tekst kojeg je neophodno kompresovati: ACAACARACACARCACARAAARACR.

    U datoj tabeli imamo samo tri ulaza A, C i R, kojima su pridruženi kodovi 0,1 i 2. Data je kompresija u tabeli na sledećem slajdu dovršite proces kompresije. (Knjiga 229 strana) . Definišite preneti kod na osnovu polja pod nazivom “šta je poslato”. I na kraju definišite proces dekompresije koji je započet u tabeli na slajdu (broj: 229.).

Prolazak kroz petlju	Bafer	C	Šta je poslato	Šta je smešteno u tabeli	Nova vrednost u baferu
1	A	C	0 (kod za A)	AC (kod = 3)	C
2	C	A	1 (kod za C)	CA (kod = 4)	A
3	A	A	0 (kod za A)	AA (kod = 5)	A
4	A	C	-	-	AC
5	AC	A	3 (kod za AC)	ACA (kod = 6)	A
6	A	R	0 (kod za A)	AR (kod = 7)	R
8	R	A	2 (kod za R)	RA (kod = 8)	RA
9	A	C	-	-	AC
10	AC	A	-	-	ACA
11	ACA	C	ACA (kod = 6)-	ACAC (kod = 9)	C
12	C	A	-	-	CA
13	CA	R	4 (kod za CA)	CAR (kod = 11)	R

PROLAZAK KROZ PETLJU	PRIOR (STRING)	CURRENT (STRING)	DA LI JE TRENUTNO U TABELI	C	PAR TEMPSTRING/KOD	TA SE ŠTAMPA (CURRENT ILI TEMPSTRING)
1	0 - A	1 -C	DA	C	AC/3	UPISATI ?
2	1 - C	0 -A	DA	A	CA/4	UPISATI ?
3	0-A	3-AC	DA	A	AA/5	UPISATI ?
4	3-AC	0-A	DA	A	ACA/6	UPISATI ?
5	0-A	2-R	DA	R	AR/7	UPISATI ?

16. JPEG kompresija je akronim za Joint Photographic Expert Group, grupu koja je formirana zajedničkim naporima ISO, ITU i IEC – njen standrard za kompresija, opšte popznat kao JPEG kompresija. Postoje tri faze JPEG kompresije prva faza je DCT tj. faza diskretne kosinusne transformacije, kvantizacije i faze kodiranja. Treća faza odnosno faza kvantizacije je karakteristično zbog ignorisanja malih promena na slici koje velika verovatnoća neće biti primetne. Dat je niz T. Generištie niz Q deljenjem datog niza skalarom 10 i zaokružiti na najbliži ceo broj. Komentarište dobijeni razultat.

$$T = \begin{bmatrix} 152 & 0 & -47 & 0 & -7 & 0 & -6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -47 & 0 & 37 & 0 & -3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5 & 0 & -3 & 0 & 11 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 3 & -1 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q=\begin{bmatrix} 15 & 0 & -5 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Matrica Q ima više nula te će se bolje kompresovati u odnosu na matricu T. Međutim, u ovom procesu izgubili smo orginalne podatke. Ako pokušamo da pomnožimo matricu Q sa 10 nećemo dobiti “staru” matricu.

DETEKCIJA GREŠKI

RAČUNARSKE MREŽE I WEB PROGRAMIRANJE

IV- GODINA

prof. dr Zlatko Langović

William A. Shay

Error Detection (Chapter 10)

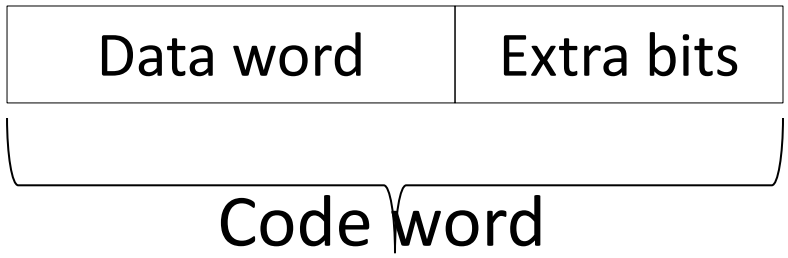
- A transmits bit to B
- B retrieves bits
- Are they the same as what A sent?
- Errors can be caused by
  - Noise in lines
  - Errors at intermediate sites that corrupt data
    - One reason why error detection is done at different layers (e.g. data link and transport layers)

- Detection
  - Did error occur?
  - Retransmit the information?
  - Ignore the error?
  - Depends on Quality of Service (QOS)
  - A file download protocol will probably respond to an error
    - Thus, a more complex protocol.
  - Errors in streaming applications are usually ignored.
    - Thus, a simpler protocol.

- Correction
  - Fix the error in the transmitted message
  - Need know not only that error occurred but which bits were affected
  - May be done if the overhead to retransmit data is very high or is not likely to result in an improvement.
  - Time sensitive applications
  - Deep space probes
  - Not common for most network applications.

- Bit error
  - one bit is damaged
  - Less likely
  - With Gbps speeds, one bit requires about 1 nanosecond. Most interference lasts longer than a nanosecond
- Burst error
  - Multiple bits in a transmission damaged

- Usual approach
  - Data word: group of *k* bits
  - Code word: data word followed by more bits calculated from the data.



  - How many more depends on the approach

- Can skip the stuff in 10.2 and 10.3 related to Hamming codes and distances. Could be included in a paper on error correction.

- Parity
  - Even parity:
    - add one bit to the end of a string to make the total number of 1s even
  - Odd parity:
    - similar but total is odd
  - We'll assume even parity from this point and beyond

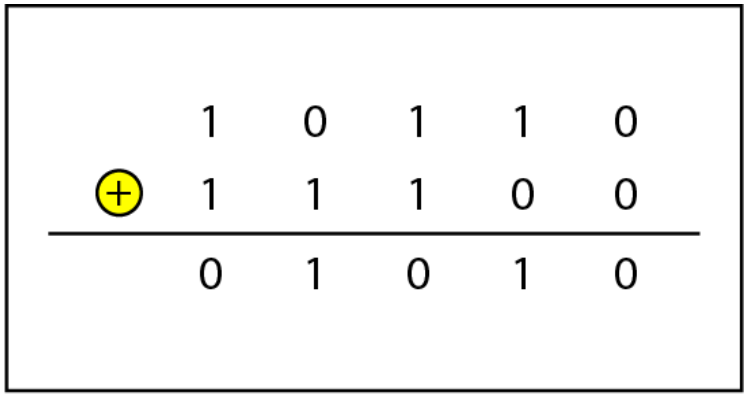
- Ex: data is 0101101100101100
- Add a 0 for parity.
- Transmitted message is 0101101100101100 0

- Ex: data is 0101101110101100
- Add a 1 for even parity.
- Transmitted message is 0101101110101100 1

Figure 10.4 *XORing of two single bits or two words*

a. Two bits are the same, the result is 0.

b. Two bits are different, the result is 1.

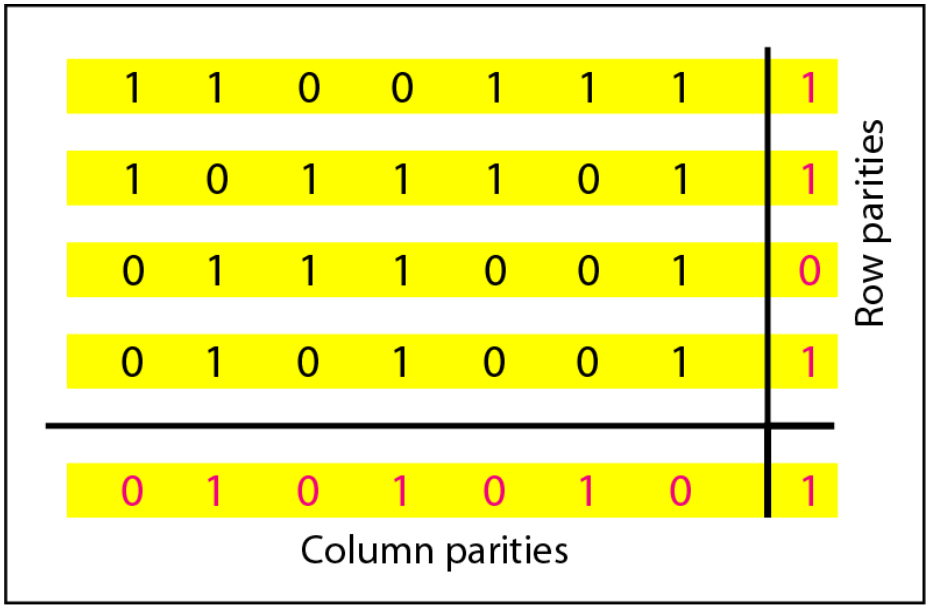


c. Result of XORing two patterns

- Bit string is $b_1b_2b_3b_4.....b_n$
- P (parity bit) = $b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus \ldots \oplus b_n$
- Where $\oplus$ is the exclusive-OR operation
- Receiver performs an ex-or operation among all bits in the code word
  - Result is 1 $\Rightarrow$ an error
  - Result is 0 $\Rightarrow$ no error detected
  - This is <u>not</u> the same as no error.

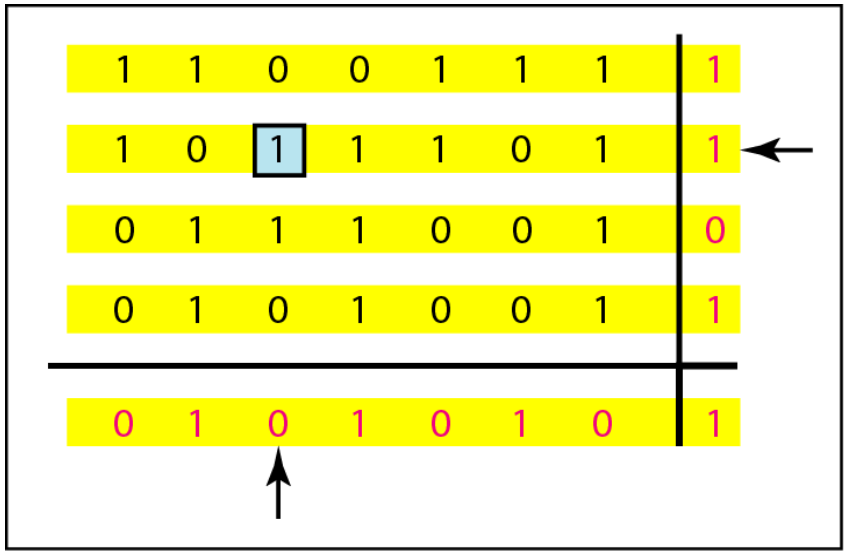
- Parity detects any errors affecting an odd number of bits.
- Assuming random noise, this is about 50% of all errors.

Figure 10.11 *Two-dimensional parity-check code*

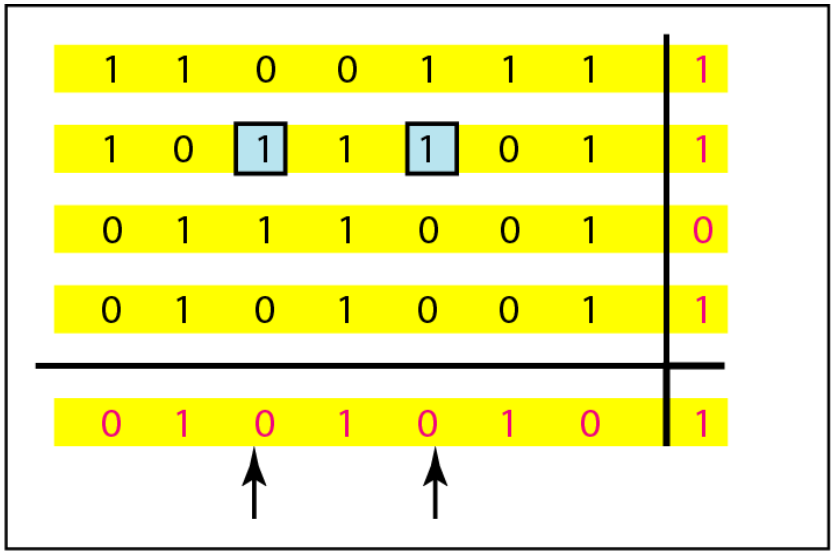


a. Design of row and column parities

Figure 10.11 *Two-dimensional parity-check code*

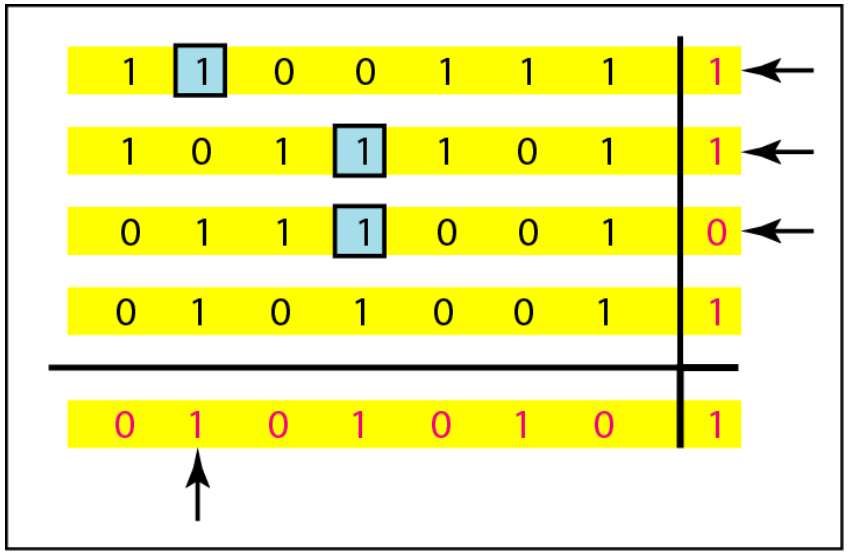


b. One error affects two parities

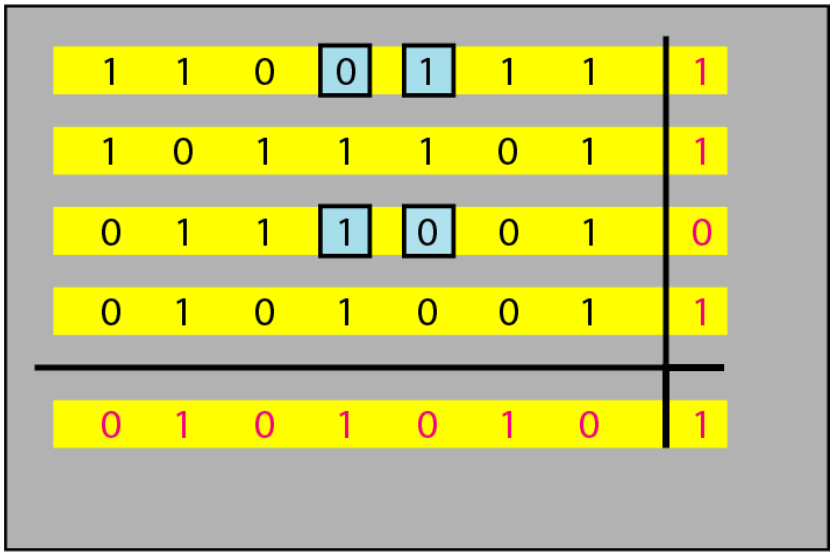


c. Two errors affect two parities

Figure 10.11 *Two-dimensional parity-check code*



d. Three errors affect four parities



e. Four errors cannot be detected

Checksums (Section 10.5)

- Interpret a byte stream as a sequence of 8, 16, or 32-bit ints.
- Sum the ints and store the sum (mod 2^8, 2^{16}, or 2^{32}) at the end of a packet.
- If the byte stream is damaged, the ints change and the checksum value changes.
  - At least most of the time.

8-bit ints

- Data is

01011010 01101010 11001101 11000011

- 90 + 106 + 205 + 195 = 84

(mod 256)

- Code word is

01011010 01101010 11001101 11000011 01010100

84

- Assume the bits in red below represent altered bits

01011111 01101010 11001101 10111110 01010100

95+ 106+ + 205 +190

- Same checksum
- Undetected error

Cyclic redundancy check (CRC) (Section 10.4)

- [http://docs.oracle.com/javase/1.4.2/docs/api/java/util/zip/CRC32.html]
- http://introcs.cs.princeton.edu/java/51data/CRC32.java.html

- Mod 2 arithmetic (0 and 1 are the only elements)

0+0=0	0+1=1	1+0=1	1+1=0
0-0=0	0-1=1	1-0=1	1-1=0

Example of multiplying polynomials (modulo 2)

$x^a \cdot x^b = x^{a+b}$ $\quad\quad$ $x^a / x^b = x^{a-b}$

$(x^5+x^3+x^1)\cdot(x^4+x^2+1) =$

$x^9+x^7+x^5 + x^7+x^5+x^3 + x^5+x^3+x^1 =$

$x^9+x^7+x^7+x^5+ x^5+x^5+x^3 +x^3+x^1 =$

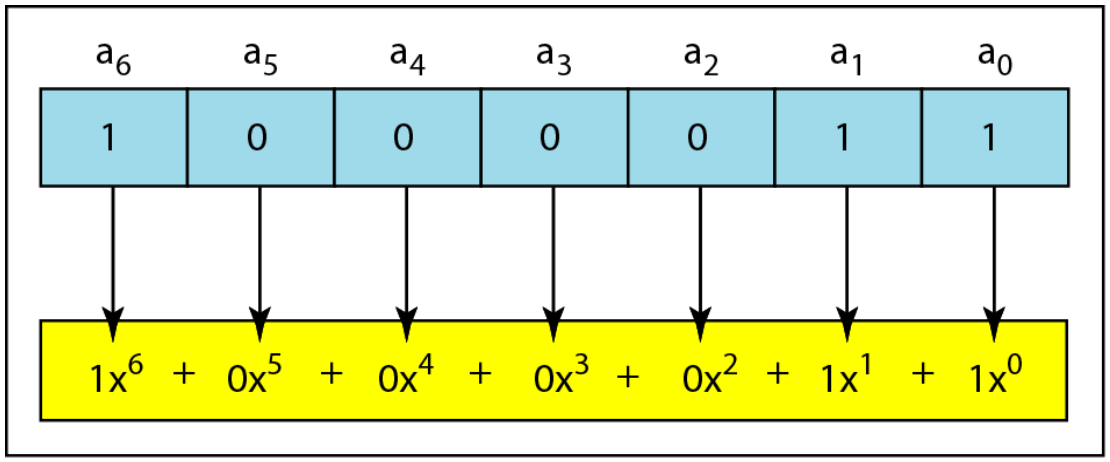
$x^9 + x^5 + x^1$

Example of dividing polynomials (mod 2).

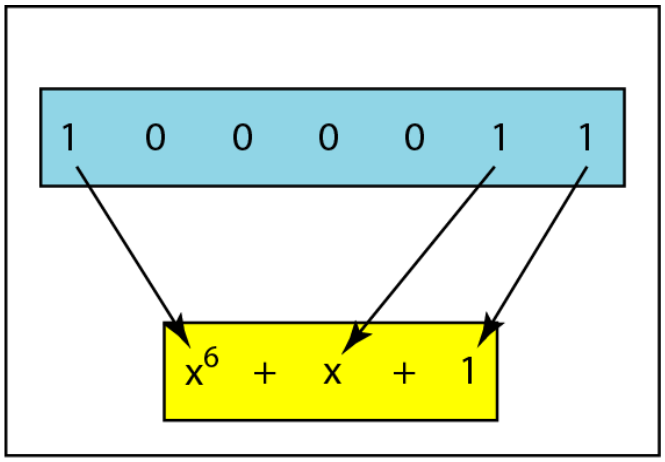
- Only interested in remainder

$$
\begin{array}{r}
 & & & & & x^6 & & + x^3 & & + x \\
\cline{2-10}
x^4 + x^3 + 1 \;\big)\; & x^{10} & + x^9 & & + x^7 & & + x^5 & + x^4 & & \\
 & x^{10} & + x^9 & & & + x^6 & & & & \\
\cline{2-10}
 & & & x^7 & + x^6 & + x^5 & + x^4 & & & \\
 & & & x^7 & + x^6 & & & + x^3 & & \\
\cline{3-10}
 & & & & & x^5 & + x^4 & + x^3 & & \\
 & & & & & x^5 & + x^4 & & + x & \\
\cline{5-10}
 & & & & & & & x^3 & + x & \text{remainder}
\end{array}
$$

Figure 10.21 *A polynomial to represent a binary word*



a. Binary pattern and polynomial



b. Short form

CRC error detection

- d…….bit string (data)
- append some 0's to d
- d(x)…corresponding polynomial
- Divide d(x) by g(x) (generator polynomial) and determine r(x), the remainder (book calls it a syndrome, s(x))
- Calculate c(x) = d(x) – s(x)
- Transmit c (codeword, bits corresponding to c(x))
- Receive c's bits. Divide c(x) by g(x). If s(x) != 0 then error.

Shortcuts to dividing

$$
\begin{array}{r}
\qquad\qquad x^6 \qquad + x^3 \qquad + x \\
x^4 + x^3 + 1 \;\overline{)}\; x^{10} + x^9 \qquad + x^7 \qquad + x^5 + x^4 \\
\underline{x^{10} + x^9 \qquad\quad + x^6} \\
\qquad x^7 + x^6 + x^5 + x^4 \\
\qquad \underline{x^7 + x^6 \qquad\quad + x^3} \\
\qquad\qquad x^5 + x^4 + x^3 \\
\qquad\qquad \underline{x^5 + x^4 \qquad\quad + x} \\
\qquad\qquad\qquad x^3 \qquad + x \qquad \text{remainder}
\end{array}
$$

```
                1 0 0 1 0 1 0
          ┌─────────────────────
1 1 0 0 1 ) 1 1 0 1 0 1 1 0 0 0 0
            1 1 0 0 1
            ---------
              0 0 1 1 1
              0 0 0 0 0
              ---------
                0 1 1 1 1
                0 0 0 0 0
                ---------
                  1 1 1 1 0
                  1 1 0 0 1
                  ---------
                    0 1 1 1 0
                    0 0 0 0 0
                    ---------
                      1 1 1 0 0
                      1 1 0 0 1
                      ---------
                        0 1 0 1 0
                        0 0 0 0 0
                        ---------
                          1 0 1 0   remainder
```

- Can you see the similarity between these two diagrams?

Figure 10.14 *CRC encoder and decoder*

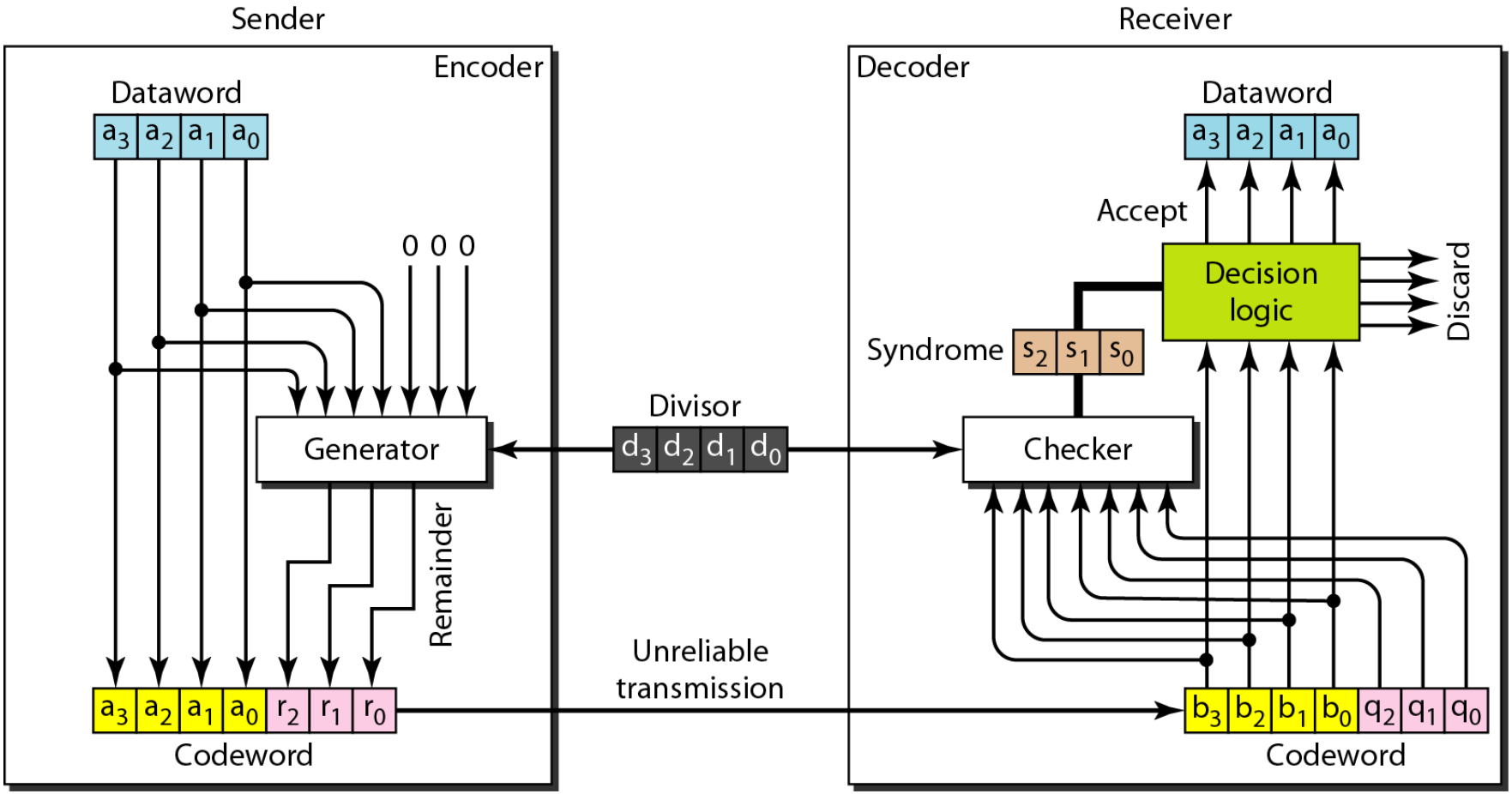


Figure 10.15 *Division in CRC encoder*

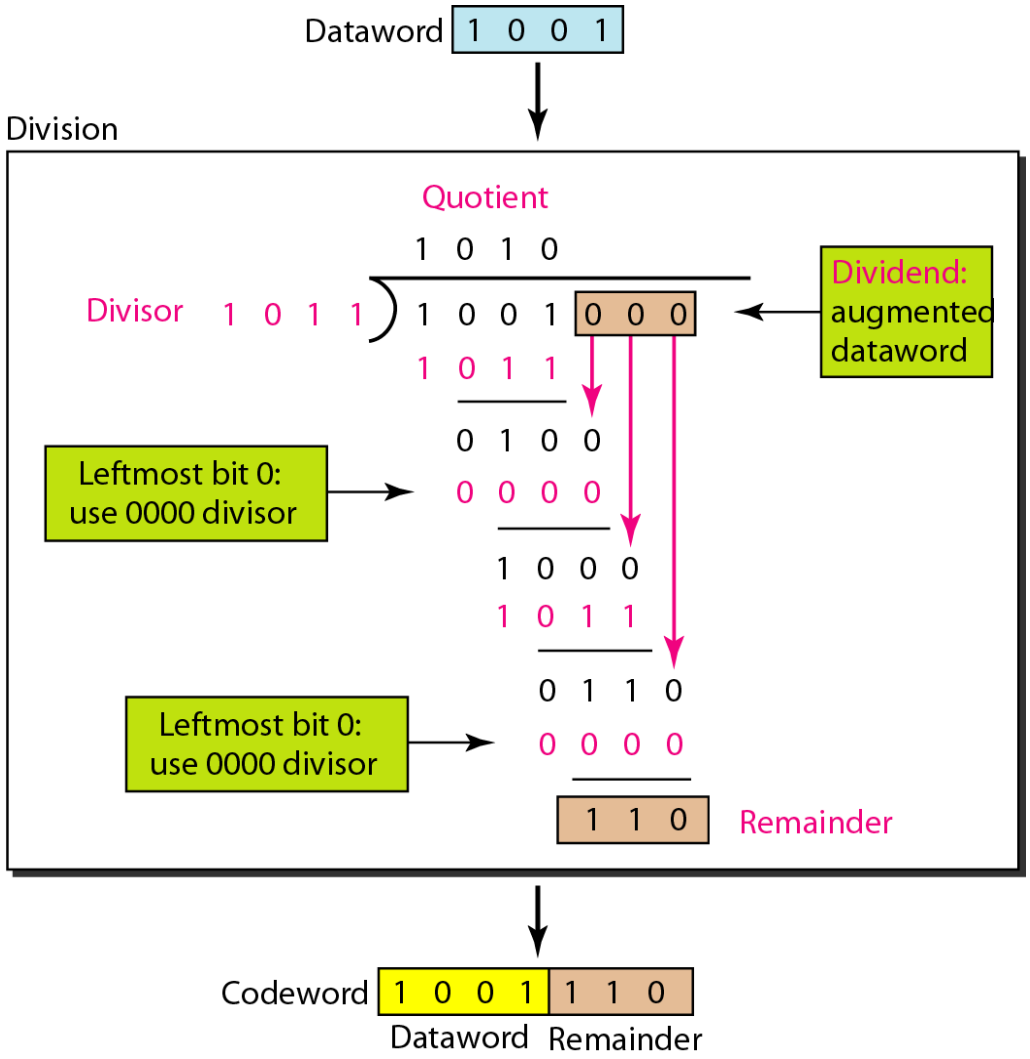
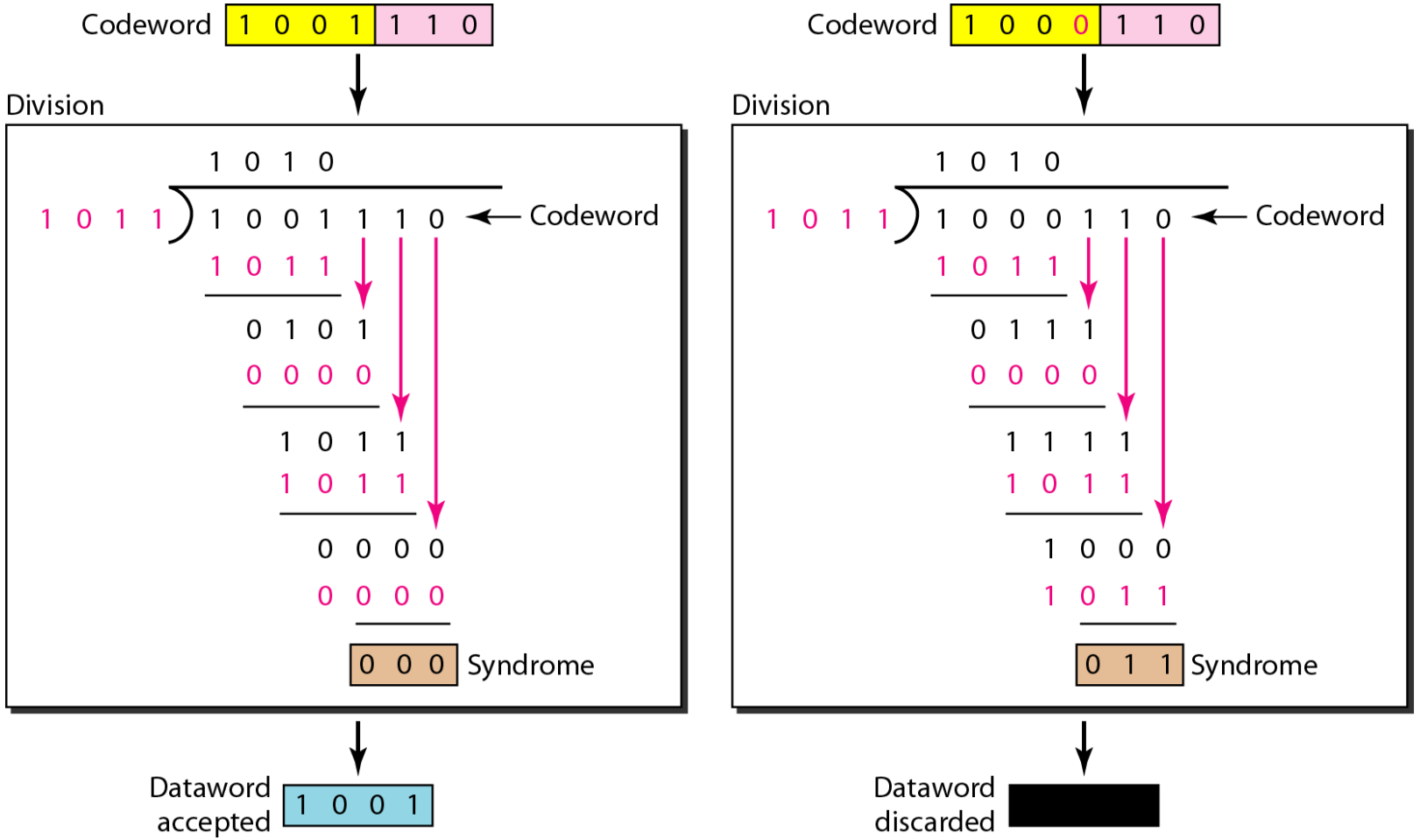


Figure 10.16 *Division in the CRC decoder for two cases*



Analysis

- c(x) is sent
- c(x) + e(x) is received (e(x) defines altered bits)
- has same remainder as

- So, if e(x) is not 0, when can this remainder be 0?
- ANS: when g(x) is a factor of e(x).
- Alternatively e(x) = g(x)·some polynomial
- When can that happen?

$$\frac{e(x)}{g(x)}$$

- Consider burst error of size k <= degree g(x);
- $e(x) = x^{i+k-1} + \ldots\ldots + x^{i}$
- $= x^{i}(x^{k-1} + \ldots. + 1)$

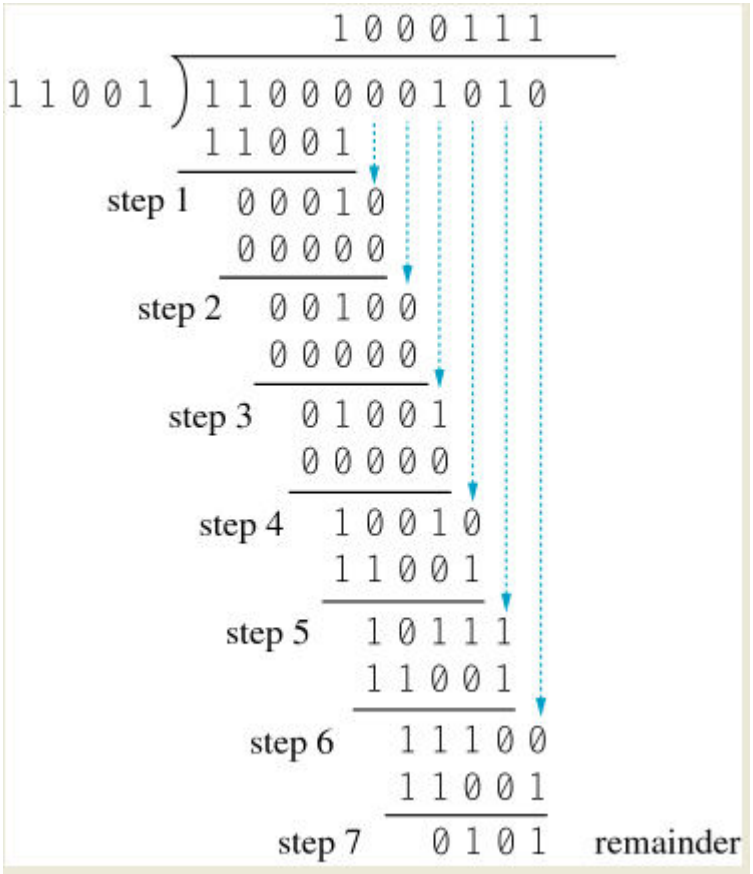
- $\frac{e(x)}{g(x)} = \frac{x^{i}(x^{k-1}+\ldots+1)}{g(x)}$

- Assume: x not a factor of g(x).
- Then no remainder $\Rightarrow$ g(x) is a factor of $(x^{k-1} + \ldots + 1)$.
- Impossible since the degree of g(x) is larger than that of $(x^{k-1} + \ldots + 1)$

- Consider an odd number of bits in the error.
- e(x) has an odd number of terms. Therefore e(1) = 1.
- Assume x+1 is a factor of g(x).
- Then $g(x) = (x+1) \cdot h(x) \Rightarrow g(1) = 0$
- Undetected error $\Rightarrow \frac{e(x)}{g(x)} = k(x) \Rightarrow e(x) = g(x) \cdot k(x) \Rightarrow e(1) = 0$
- Contradiction: so the assumption that there is an undetected error is wrong.
- Proof by contradiction from CS241

- Detects: all burst errors < degree g(x)
- All burst errors affecting an odd # of bits
- All burst error of length > r+1 with probability of $\frac{2^r - 1}{2^r}$
- If r=32, probability of detection is $\frac{2^{32} - 1}{2^{32}} \sim$ 99.99999998%

How to do this efficiently: Shift register for x^4+x^3+1



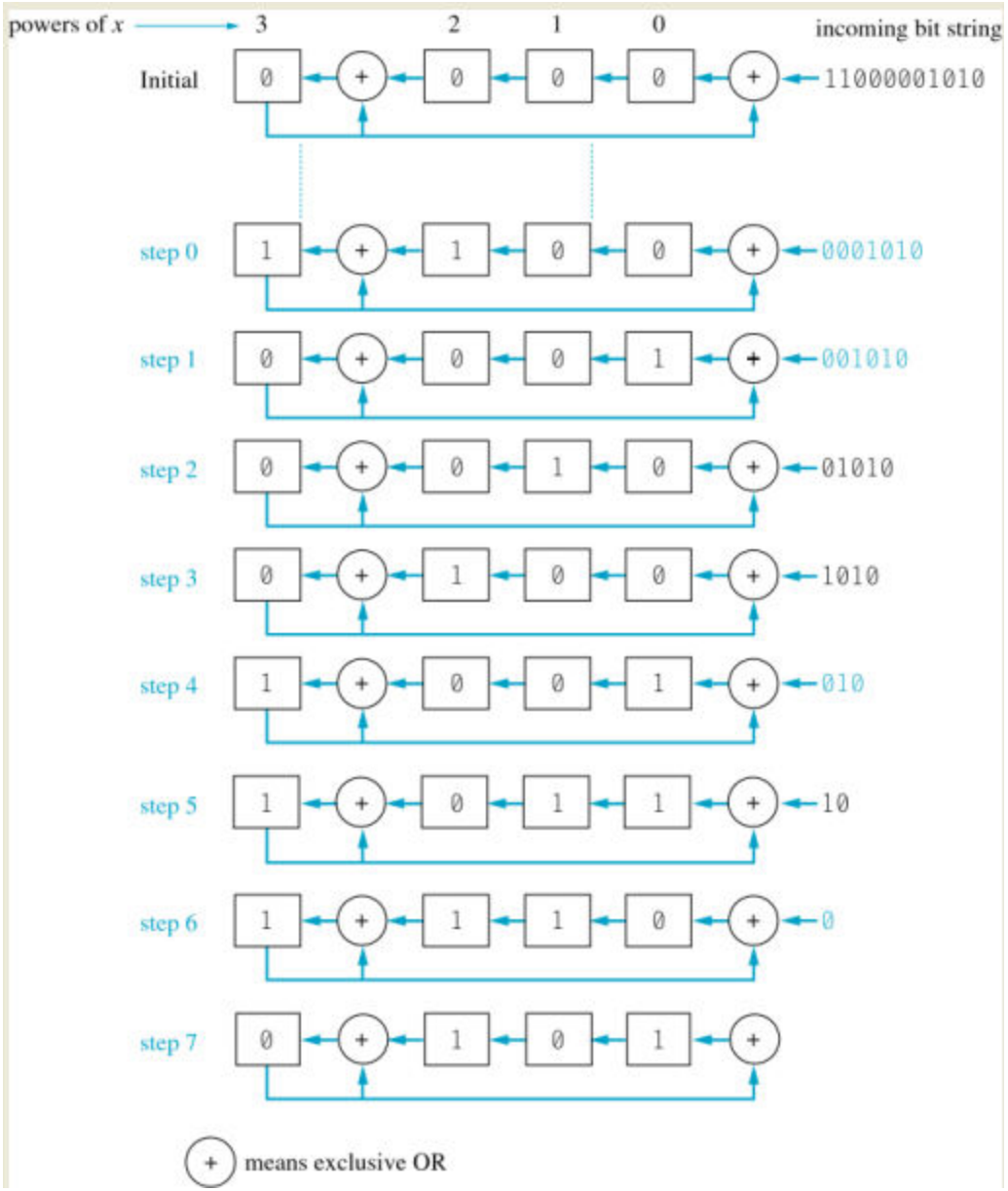


Figure 10.18 *Simulation of division in CRC encoder*

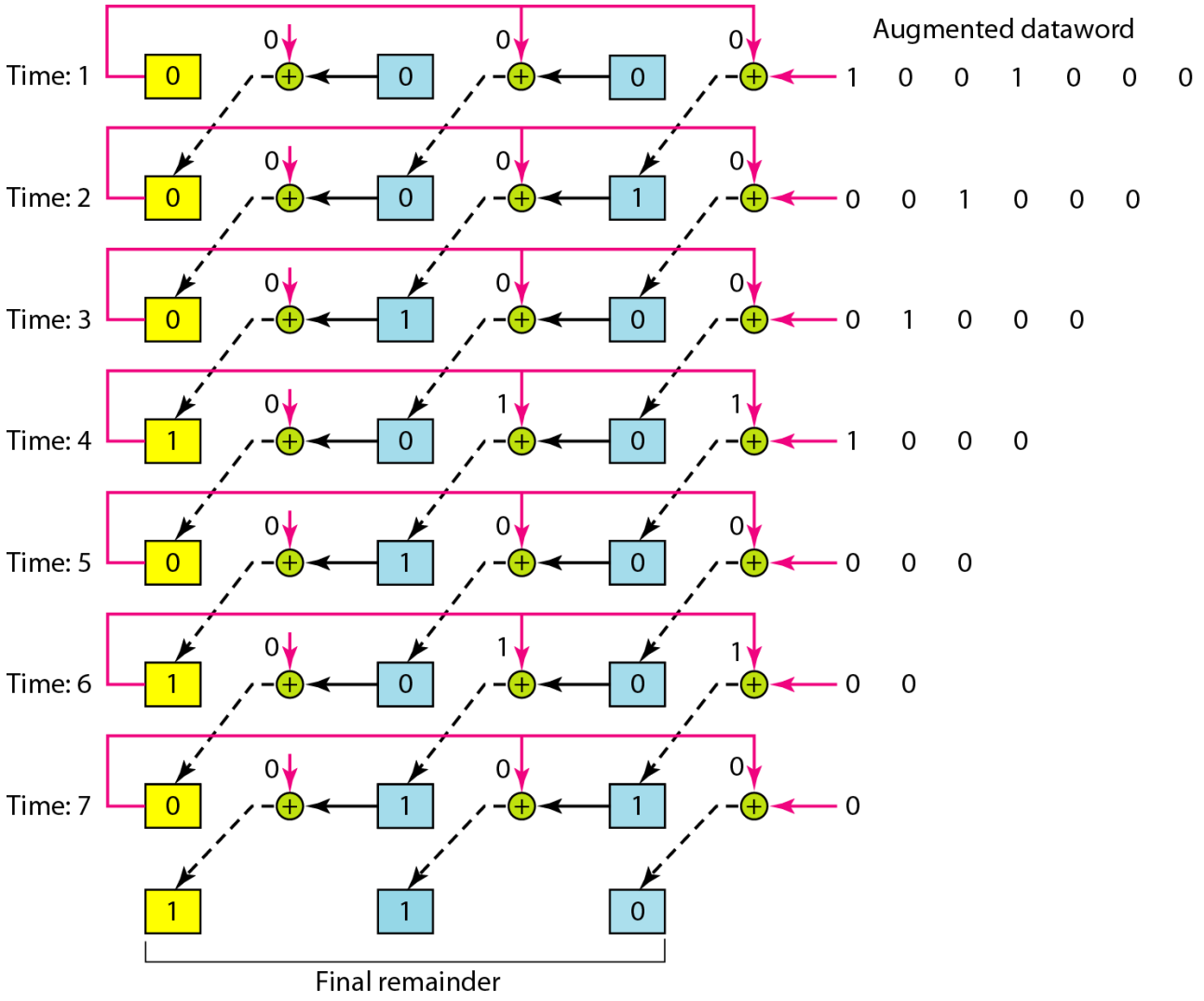


Figure 10.19 *The CRC encoder design using shift registers*

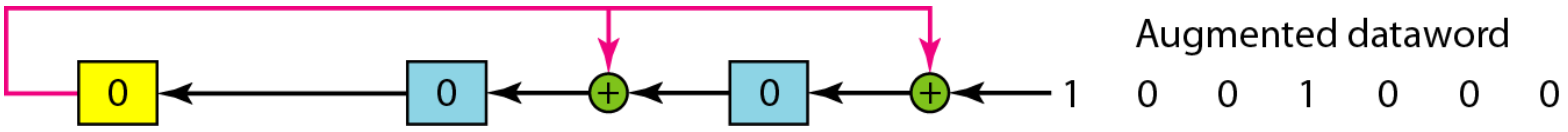


Figure 10.20 *General design of encoder and decoder of a CRC code*

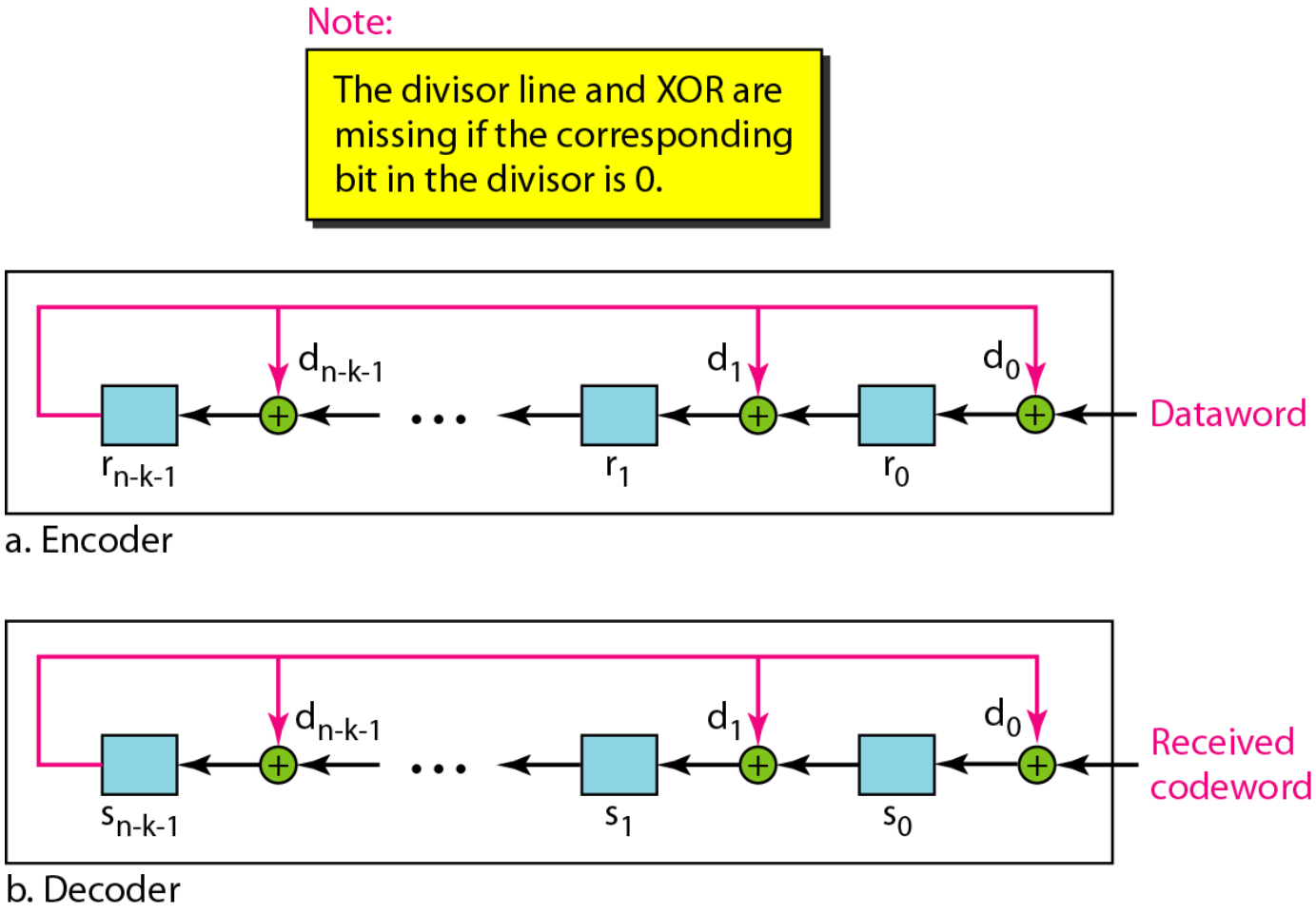


Table 10.7 *Standard polynomials*

Name	*Polynomial*	*Application*
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

Zadaci-3:

17. Detekcija grešaka pomoću ciklične provere redudantnosti (CRC). Dati metod koji proverava tačnosti odnosno pronalazi greške izvodi deljenje polinoma. Podeli date polinome.

$T(x) = x^{10}+x^8+x^6+x^5+x^3$

$= x^{10}+0*x^9+x^8+0*x^7+x^6+x^5+0*x^4+x^3+0*x^2+0*x^1+0*x^0$

$G(x) = x^4+x^2+x = x^4+0*x^3+x^2+x^1+0*x^0$

$(x^{10} / x^4) = x^6$

$(x^9 / x^4) = x^5$

$(x^8 / x^4) = x^4$

$(x^7 / x^4) = x^3$

$(x^5 / x^4) = x$

$$
\begin{array}{l|l}
 & x^6+x^5+x^4+x^3+0+x \\
\hline
x^4+0*x^3+x^2+x^1+0 & x^{10}+x^9+0*x^8+x^7+0*x^6+x^5+x^4+0*x^3+0*x^2+0*x^1+0*x^0 \\
 & x^{10}+0*x^9+x^8+x^7+0*x^6 \\
\hline
 & x^9+x^8+0+0+x^5 \\
 & x^9+0+x^7+x^6+0 \\
\hline
 & x^8+x^7+x^6+x^5+x^4 \\
 & x^8+0+x^6+x^5+0 \\
\hline
 & x^7+0+0+x^4+0 \\
 & x^7+0+x^5+x^4+0 \\
\hline
 & x^5+0+0+0+0+0 \\
 & x^5+0+x^3+x^2+0+0 \\
\hline
 & x^3+x^2+0+0 \quad \text{Ostatak}
\end{array}
$$

- 111101
- 10110 | 11010110000
- 10110
- 11001
- 10110
- 11111
- 10110
- 10010
- 10110
- 01000
- 00000
- 10000
- 10110
- 01100
- 00000
- 1100 Ostatak

KRIPTOGRAFIJA

RAČUNARSKE MREŽE I WEB PROGRAMIRANJE

IV- GODINA

prof. dr Zlatko Langović

William A. Shay

Cryptography: Chapter 30

- Plaintext
  - some easily recognized data
- Encrypt
  - change the plaintext into something difficult to comprehend (ciphertext)
- Decrypt
  - change the ciphertext back into plaintext.
- To encrypt
  - need an algorithm (cipher) and a key.
  - The key affects the result of the cipher.
- To decrypt
  - need an algorithm and a key.

- Two main cipher classes
- Symmetric key (secret key)
  - same key use to encrypt and decrypt.
- Must be carefully guarded.
- Public key
  - used to encrypt and you don’t care who knows the key.
- Private key used to decrypt.

Figure 30.3 *Symmetric-key cryptography*

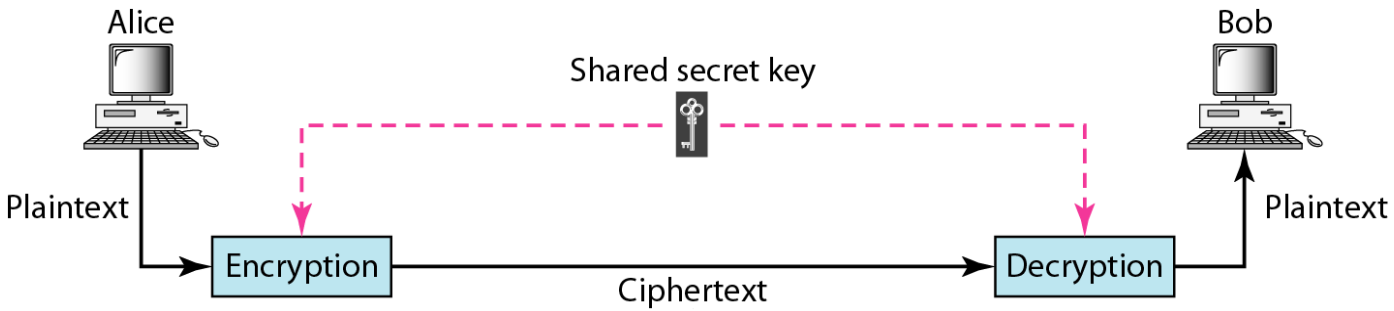
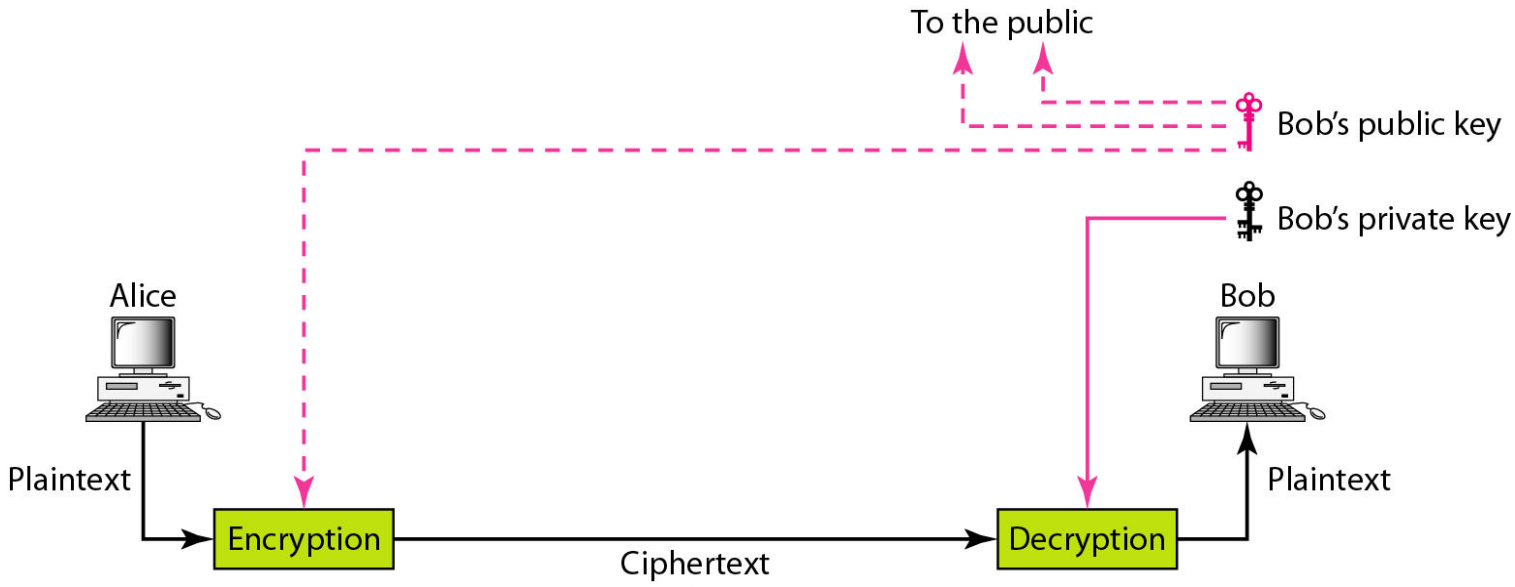


Figure 30.4 *Asymmetric-key cryptography*



Notation

- P…plaintext
- C…ciphertext
- E…encryption algorithm
- D…decryption algorithm
- k and k'…keys
- $C=E_k(P)$
- $P= D_{k'}(C)$
- $P=D_{k'}(E_k(P))$
- Maybe k=k' and maybe not

- Substitution cipher
  - replace one bit group with another (e.g. one character with another)
- Monoalphabetic cipher
  - one character always encrypted the same way
- Caesar Cipher
  - classic example of a substitution cipher
- Example
  - $E_k(c)=c+k$ (add a number, k, to the code for c)

The following shows a plaintext and its corresponding ciphertext. Is the cipher monoalphabetic?

Plaintext: HELLO
Ciphertext: KHOOR

Solution

The cipher is probably monoalphabetic because both occurrences of L's are encrypted as O's.

- This is lousy encryption
- Preserves frequency of characters.
  - If ‘E’ occurs 25% of the time in the plaintext and ‘E’ is encrypted to ‘X’ then ‘X’ occurs 25% of the time in the ciphertext.
  - This is a pattern (clue) to the method.
  - The previous example had KHOOR as the ciphertext.
  - What letter got encrypted as an ‘O’?
  - Probably not a ‘H’, ‘J’, ‘Q’, or ‘Y’ since they rarely appear consecutively in normal text.
  - This narrows the possibilities a bit.
  - More patterns narrow them even more.

- This method also preserves sequences:
  - If $E_k(c)=c+2$ then there might be many occurrences of "VJG".
  - Again, this is a clue that could eventually reveal "VJG" as the encrypted form of "THE".

- These are important clues to someone who has the ciphertext but does not know how to decrypt.
- They use these clues to deduce how the ciphertext may be been created.
- This, in turn, allows them to determine ways in which the ciphertext can be converted back to plaintext, in effect, “cracking the code”.
- Patterns in a ciphertext are a bad thing.
- A good method will disrupt all patterns as much as possible.
- We need to know what patterns can occur.

Polyalphabetic cipher.

- Same character may be encrypted different ways
- Example
- P…array of plaintext characters
- C…array of ciphertext characters

for i=1 to n do

$C[i] = P[i] + k_{i \bmod 3}$

- suppose $k_0=1$; $k_1=2$; $k_2=3$.
- Then “ABC” is not always encrypted the same way.
- That pattern is disrupted.

- But there is still a pattern-it's just a little harder to see.
- "ABC" would always be encrypted as "BDF", "CED", or "DCE".
- Sum of differences of characters from the same relative position in any two of these ciphertexts is always 0.
- For example, look at the first two
  - B-C is -1; D-E is -1; F-D is 2; the numbers sum to 0.
- Other pairs are similar.

- Can you find the pattern if $k_0=1$; $k_1=5$; $k_2=4$?

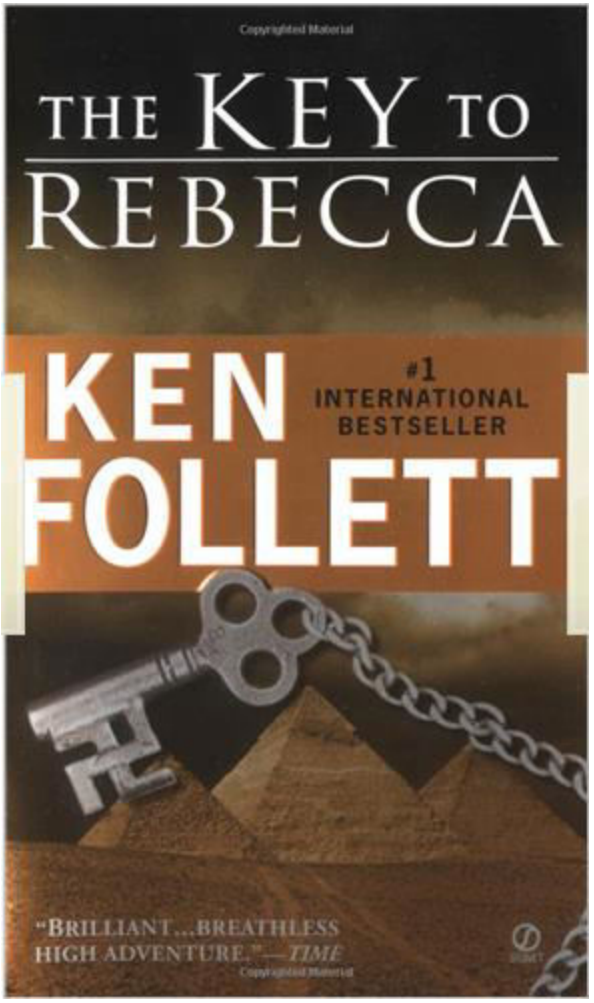
- Bit level (ex-or)
- $C=E_k(P)=P \oplus k$
- P=10101010101010
- k=11001100110011
- C=01100110011001
- NOTE: if two plaintext messages differ by 1 bit, so do the encrypted versions.
- That's a pattern

key length

- If P is longer (has more bits) than k then P is divided into blocks, each with the same length as k.
- Each block is encrypted with k.
- This is a *block cipher*.
- If k has a small number of bits, more patterns exist in C
- If k is longer, there are usually fewer patterns in C
- For a given algorithm, a longer key is usually more secure than a shorter key.

- In the extreme case, k and P have the same number of bits. If k's bit are random, the encryption has no patterns.
- One time pad
- k (random bits) used only once and key length = length of message
- Leaves NO statistical base for analyzing encrypted message.
- About the only method considered truly unbreakable, short of brute force (trying all possible keys).
- Managing and sharing the key is an issue.


LOOK INSIDE!
REBECCA
DAPHNE DU MAURIER


Copyrighted Material
THE KEY TO REBECCA
KEN FOLLETT
#1 INTERNATIONAL BESTSELLER
"BRILLIANT...BREATHLESS HIGH ADVENTURE." —TIME
Copyrighted Material

- Key to Rebecca by Ken Follett
  - a reference to a Novel “Rebecca” by Daphne Du Maurier
- Each day a different page from the novel contains the key used to encrypt messages sent to a Nazi superspy
- [http://www.math.okstate.edu/~wrightd/crypt/crypt-intro/node12.html]

- 13-3-2-21-1-1-8-5
- O, Draconian devil
- Oh, lame saint
- A clue to a murder from “The Da Vinci Code”

- Anagrams for:
- Fibonacci numbers (1,1,2,3,5,8,13,21
- Leonardo Da Vinci
- The Mona Lisa

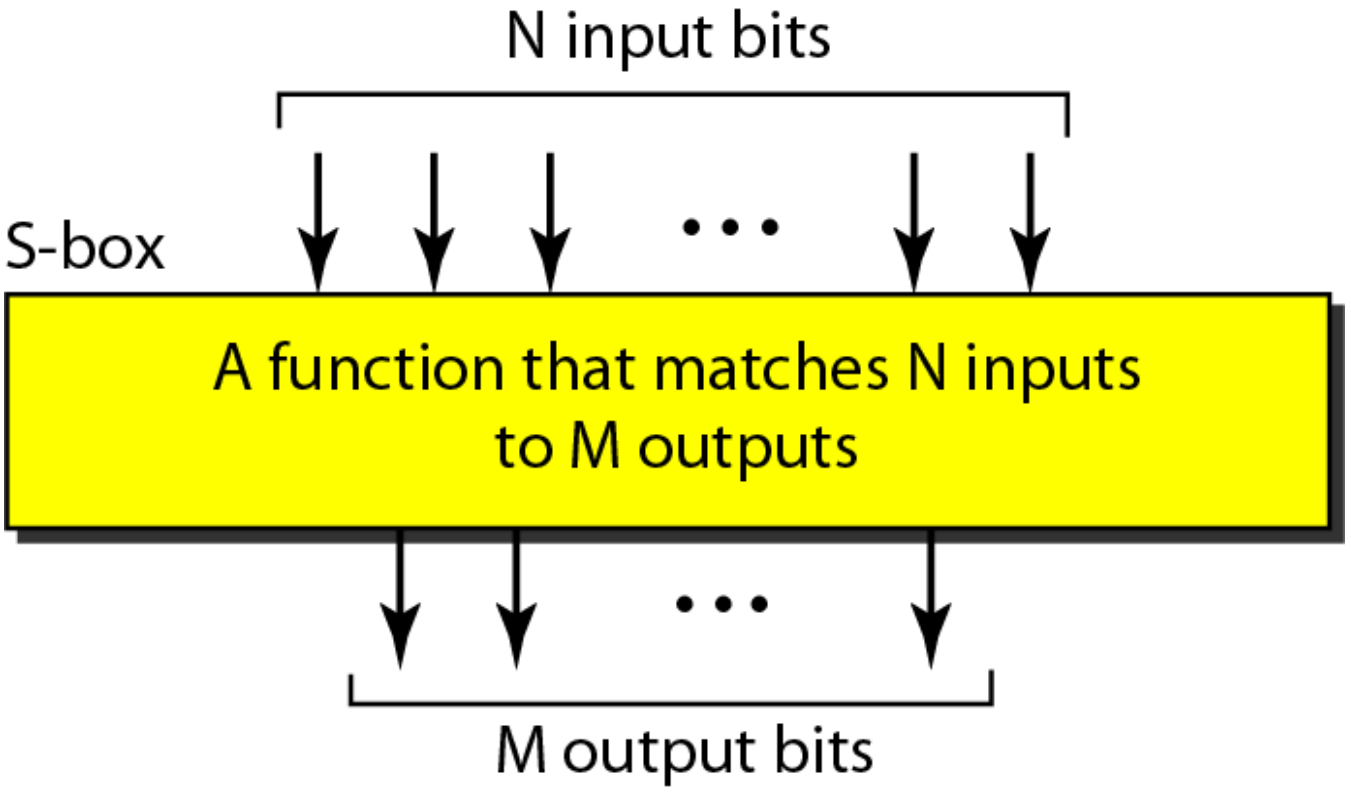
Transposition cipher

- Rearrangement of blocks in P
- Key is a permutation of integers that define the rearrangement
- Not particularly secure
- Characters are preserved
- Most useful as part of a more complex scheme.

S-Box

- A type of substitution that replaces an N-bit input with an M-bit result
- Often an intermediate activity in a cipher

Figure 30.11 *S-box*



Data Encryption Standard (DES): Developed by IBM and NSA (No Such Agency)

- confusion and diffusion.
- Multiple rounds
- Substitutions
- Transpositions

Figure 30.13 *DES*

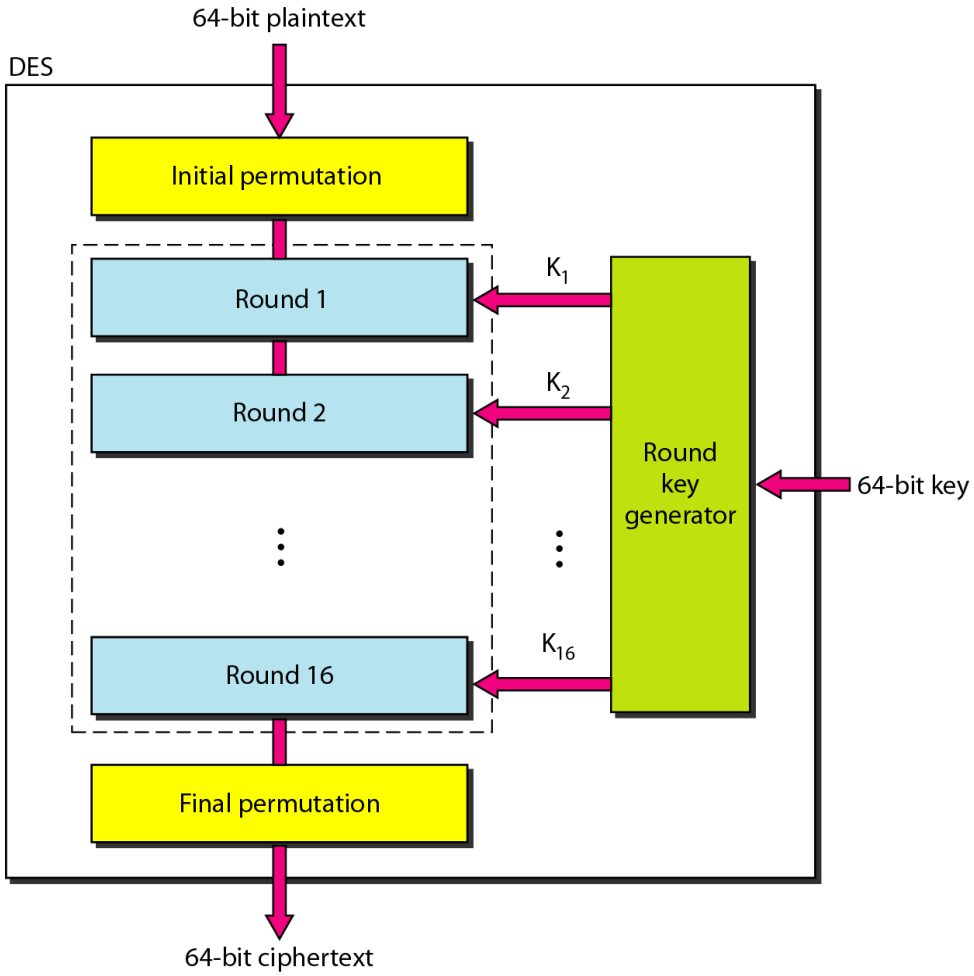


Figure 30.14 *One round in DES ciphers*

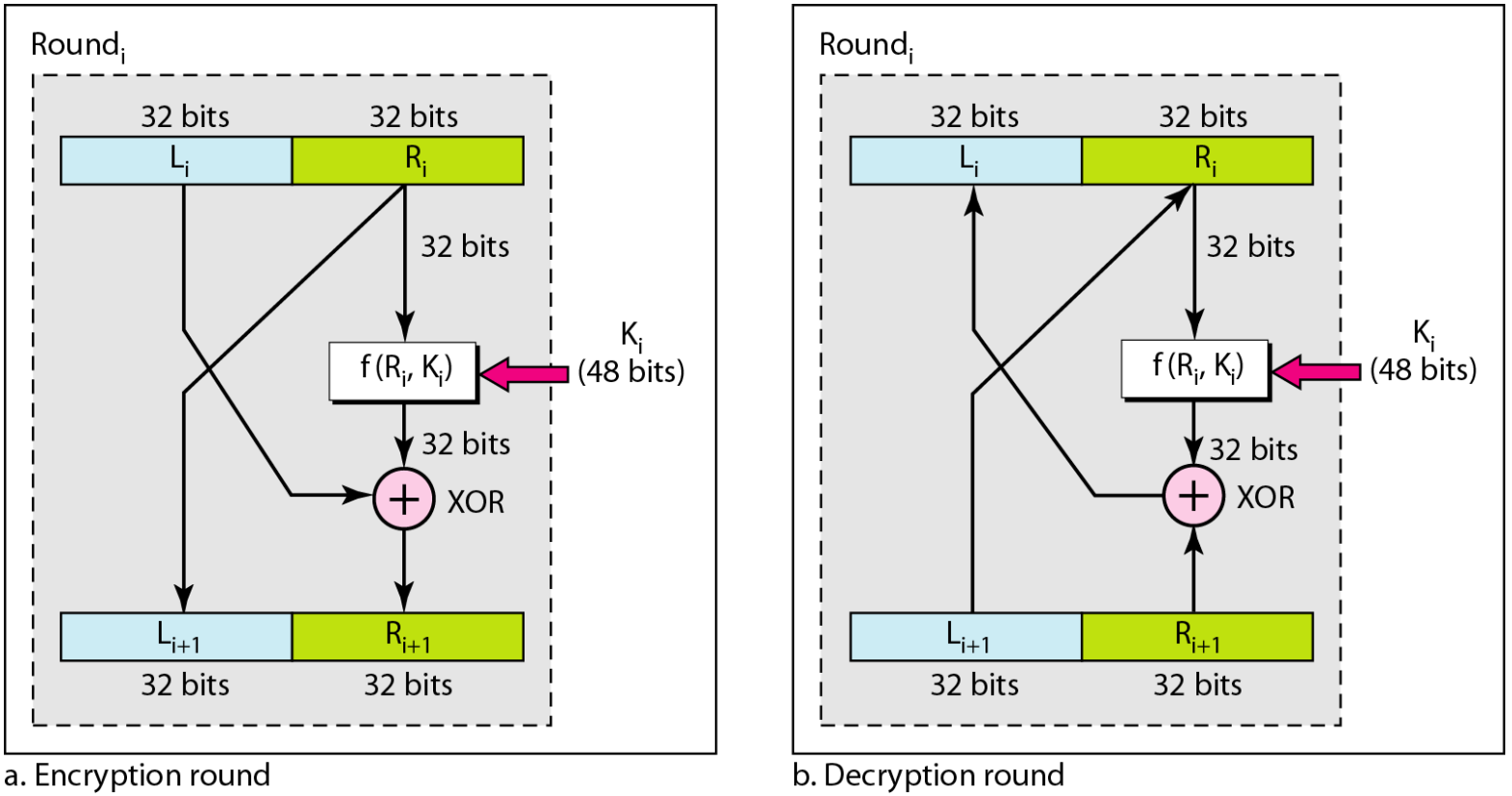
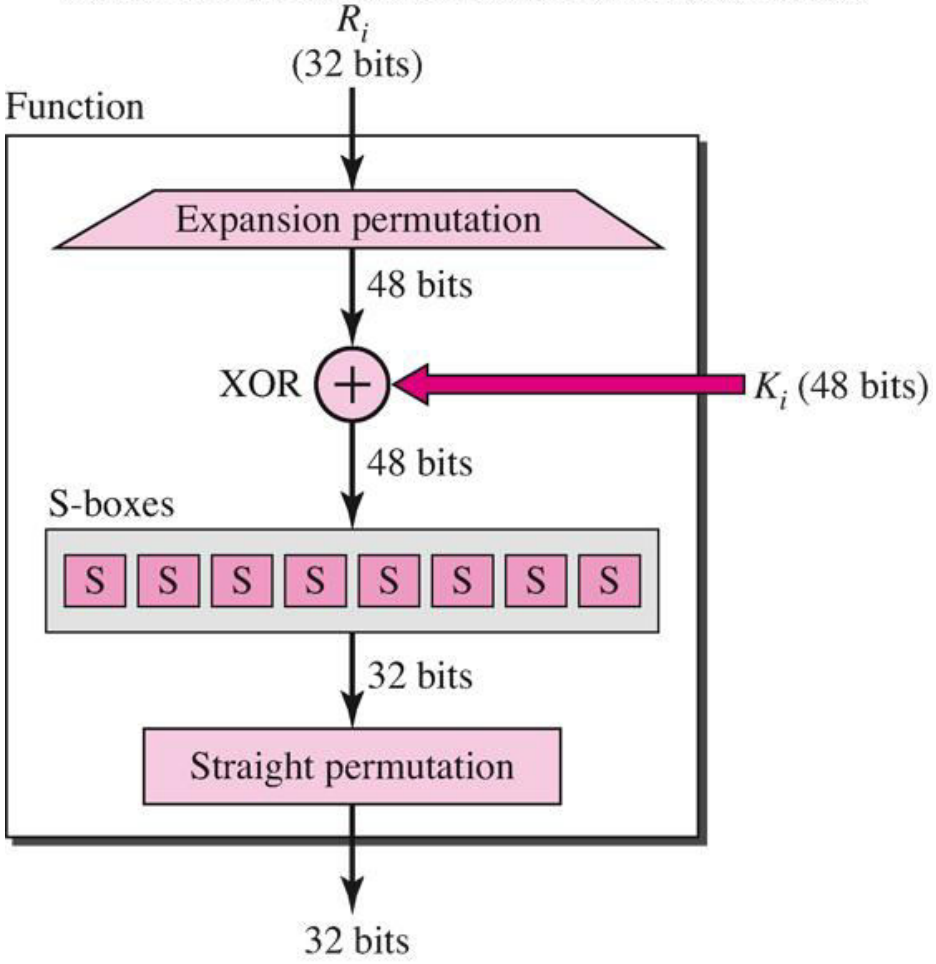


Figure 30.15 *DES function: $f(R_i, K_i)$*



DES controversy

- 2nd page of [http://nvl.nist.gov/pub/nistpubs/sp958-lide/250-253.pdf]

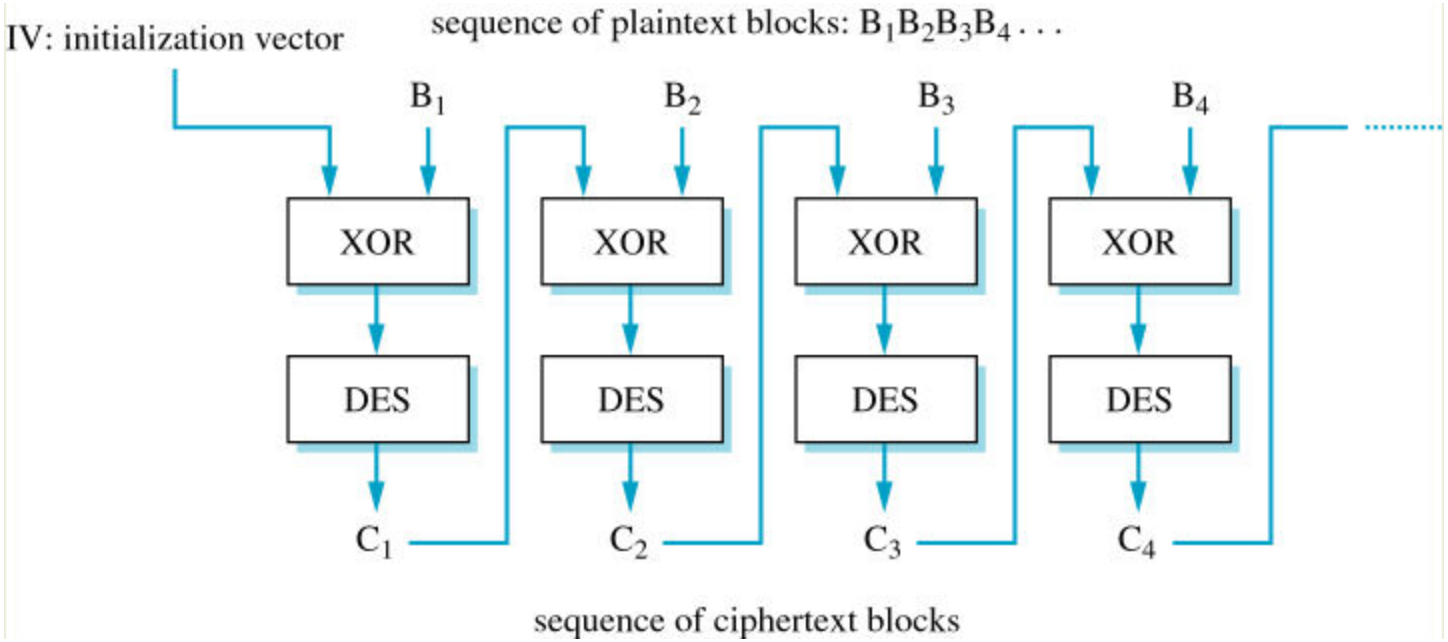
- Electronic Frontier Foundation [http://www.eff.org/] – sort of an ACLU of cyberspace
- built "DES Cracker" (1/4 million $) in 1998. In conjunction with worldwide network of PCs, could try billions of keys per second and cracked DES in hours
- [http://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/19980716_eff_des.faq]
- [http://en.wikipedia.org/wiki/Deep_Crack]

Electronic codebook mode (ECB):

- Each block encrypted independently.
- Same blocks always encrypted the same way.
- Another pattern – more prevalent if the blocks are small.
- Less so – if the blocks are large.

Cipher block chaining mode (CBC):

- Step i feeds into step i+1
- First step requires an IV (Initialization Vector)



Triple DES (3DES):

- ANSI standard X9.52.
- $E_k(P)$ represents DES$\Rightarrow$Triple DES defined by $E_{k3}(D_{k2}(E_{k1}(P)))$.
- Allows all keys to be equal for backward compatibility with DES.
- Question:
  - Does applying the same cipher in sequence really yield anything different?
- ANS:
  - Maybe/maybe not

Examples

- $E_k(P)=P+k$
  - Caesar cipher
- Then $E_{k1}(E_{k2}(P))=E_{k1+k2}(P)$
  - no substantial difference!!!
- The collection of Caesar ciphers form a mathematical group and the composition of two ciphers is equivalent to another.
- This is NOT true for DES because DES does NOT form a mathematical group (Proven in 1993).

Advanced Encryption Standard (AES)

- Rijndael algorithm
  - Named after Belgian cryptographers: Vincent Rijmen and Joan Daemen)
- NIST sent out a call for algorithms to be new standards in 1997.
- Finalists included MARS, RC6, Rijndael, Serpent, Twofish (look them up on Internet).
- Rijndael adopted in 2000.
- Key sizes of 128, 192, or 256 bits

- At least as strong as 3DES but quicker.
- Block cipher:
  - Organizes 128 bits into 16 bytes & puts them into a 4x4 matrix
- Goes through a series of steps: xor, multiple rounds each with a separate key derived from the initial key, permutations, substitutions, shift operations. [http://en.wikipedia.org/wiki/Advanced_Encryption_Standard]

- Each round has a separate key defined by a key generator:
  - Each key is a sequence of 4-byte words w_i i = 0…43
  - First key is initial key, w_0 through w_3.
  - If i%4 !=0: $w_i = w_{i-1} \oplus w_{i-4}$
  - If i%4==0: Permute the bytes in w_i; substitute each byte with another using an S-box; do another ex-or with a round constant defined by Rijndael.

- 4 steps in each round. They are
  - Replace each byte using an S-box
  - Circular shift bytes in each row (except 1st).
  - (Most complex) interpret each column as a polynomial, multiply it by another defined by Rijndael (modulo yet a 3rd polynomial). Polynomial's coefficients are interpreted as 8-bit nos. from a Galois field, a mathematical structure. Rules for multiplying are complex and requires knowledge of groups and rings found in an abstract algebra course.
  - Do ex-or with round key.

– Entire books written on Rijndael!!!

Section 30.3: Public key encryption (Asymmetric-key cryptography)

- Previous discussion involved private (symmetric) key methods
- decryption key easily derivable (or was equal to) from the encryption key and method and the encryption key had to be kept secret (private).

- With public key encryption, the key and algorithm are known to anyone!!!
- The decryption key is NOT easily derivable, even if encryption key and algorithm are known.
- Many clients encrypt using the same key but none can decrypt another's message.
- Cannot even decrypt your own.
- Useful when sending encrypted information to a central site (servers, banks, secure sites, etc).

RSA challenge.

- Until the end of 2007 you could have won $200,000 simply by factoring this number:
- 25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748808428401797429100642458691817195118746121515172654632282216869987549182422433637259085141865462043576798423387184774447920739934236584823824281198163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386356373289912154831438167899885040445364023527381951378636564391212010397122822120720357
- previous winner:

What is RSA?

- It's actually who?
- Ronald Rivest, Adi Shamir, Leonard Adleman
- [http://www.rsa.com/node.aspx?id=1003]
- [http://www.rsa.com/rsalabs/]

RSA Algorithm

- public key encryption
- based on the works of number theorists Leonhard Euler and Pierre de Fermat circa 1700s and products of large prime numbers
- A simple (small numbers) illustration of the RSA algorithm follows.

- Assume characters A through Z
- Assign codes 1-26 to A through Z
- Choose $n = p \cdot q$, where p and q are prime.
- E.g. p=7, q=11, n=77.
  - In practice, maybe a couple of hundred digits long.
- Find e where e is relatively prime to $\phi = (p\text{-}1)\cdot(q\text{-}1)$. This means the two numbers have no common factors.
  - And, yes, such an e can always be found.
- E.g. e=7 is relatively prime to $6\cdot10$=60.

- Divide message into blocks
- Interpret the binary representation of 1 block as an integer x.
- For each x, compute x^e mod n
- E.g.: H E L L O = 8 5 12 12 15
- Compute:
  - 8^7 mod 77 = 57
  - 5^7 mod 77 = 47
  - 12^7 mod 77 = 12 (this is just coincidence)
  - 12^7 mod 77 =12
  - 15^7 mod 77 =71

- To decrypt find d where $d \cdot e$-1 is divisible by

  $\phi = (p\text{-}1)\cdot(q\text{-}1)$

  - Yes, such a *d* can always be found according to number theory.
- Another way to say this is $d \cdot e = 1 \bmod \phi$
- In our example, $7 \cdot d$-1 must be divisible by 60. Can use d=43
- For each y in the encrypted message compute $y^d \bmod n$. You will always get back to x because these two operations are inverse operations (proof left to number theorists)

- See also examples on pages 950-951
- Numbers could be huge but there is a shortcut since all calculations are modular.

Calculate $71^{43} \bmod 77$

- $71^{43} \bmod 77 = 71^{32+8+2+1} \bmod 77$ (exponent is sum of powers of 2

$= 71^{32} \cdot 71^{8} \cdot 71^{2} \cdot 71 \bmod 77$

$= (71^{2})^{16} \cdot (71^{2})^{4} \cdot (71^{2}) \cdot 71 \bmod 77$

Since $71^{2} \bmod 77$ is 36 we get

$= 36^{16} \cdot 36^{4} \cdot 36 \cdot 71 \bmod 77$ (all the nos are smaller

Since $36^{2} \bmod 77$ is 64 we get

$= 64^{8} \cdot 64^{2} \cdot 36 \cdot 71 \bmod 77$ (smaller yet)

Since $64^{2} \bmod 77$ is 15 we get

$$= 15^4 \cdot 15 \cdot 36 \cdot 71 \bmod 77$$

$$= 71^2 \cdot 15 \cdot 36 \cdot 71 \bmod 77$$

$$= 36 \cdot 15 \cdot 36 \cdot 71 \bmod 77$$

$$= 15 \bmod 77$$

Analysis

- Suppose you know n and e – the necessary parameters to the encryption calculation.
- Can you derive d from this?
- All you need to do is find d where $d \cdot e\text{-}1$ is divisible by $\phi = (p\text{-}1)\cdot(q\text{-}1)$.
- Finding d requires knowing p and q, the factors of n
- If n has hundreds of digits, this is hard.
- really hard!

- See the course web site for a java program that implements rsa using its BigInteger class.

Key exchange/distribution for symmetric key algorithms.

- Some protocols require keys to be communicated/shared among two or more users secretly (ssh or secure shell and IPSec are examples)
- If A and B both want to exchange data securely how can they agree on a key?
- RSA is good for one direction – to the site with the private key.
- How do you share keys securely?

Shamir's method

- Need any k people to decrypt
- Create polynomial $a_0 + \ldots + a_{k-1}x^{k-1}$ where a_0 is the encryption key
- Assign 1 point on graph to each person
- Any k people $\rightarrow$ k points $\rightarrow$ can construct polynomial and get key

Confidentiality

- Both sides need to exchange information confidentially.
- Session key:
  - encryption key used only for the duration of a session.
- New one negotiated with each session

- If A and B both have a public key, each can encrypt using the other's public key.
  - Each can decrypt using their private key.
- If ONLY B has a public key,
  - A can choose a session key
  - A can encrypt it using B's public key
  - B can decrypt it (using its private key) to get the key
- If neither has a public key
  - Diffie-Hellman

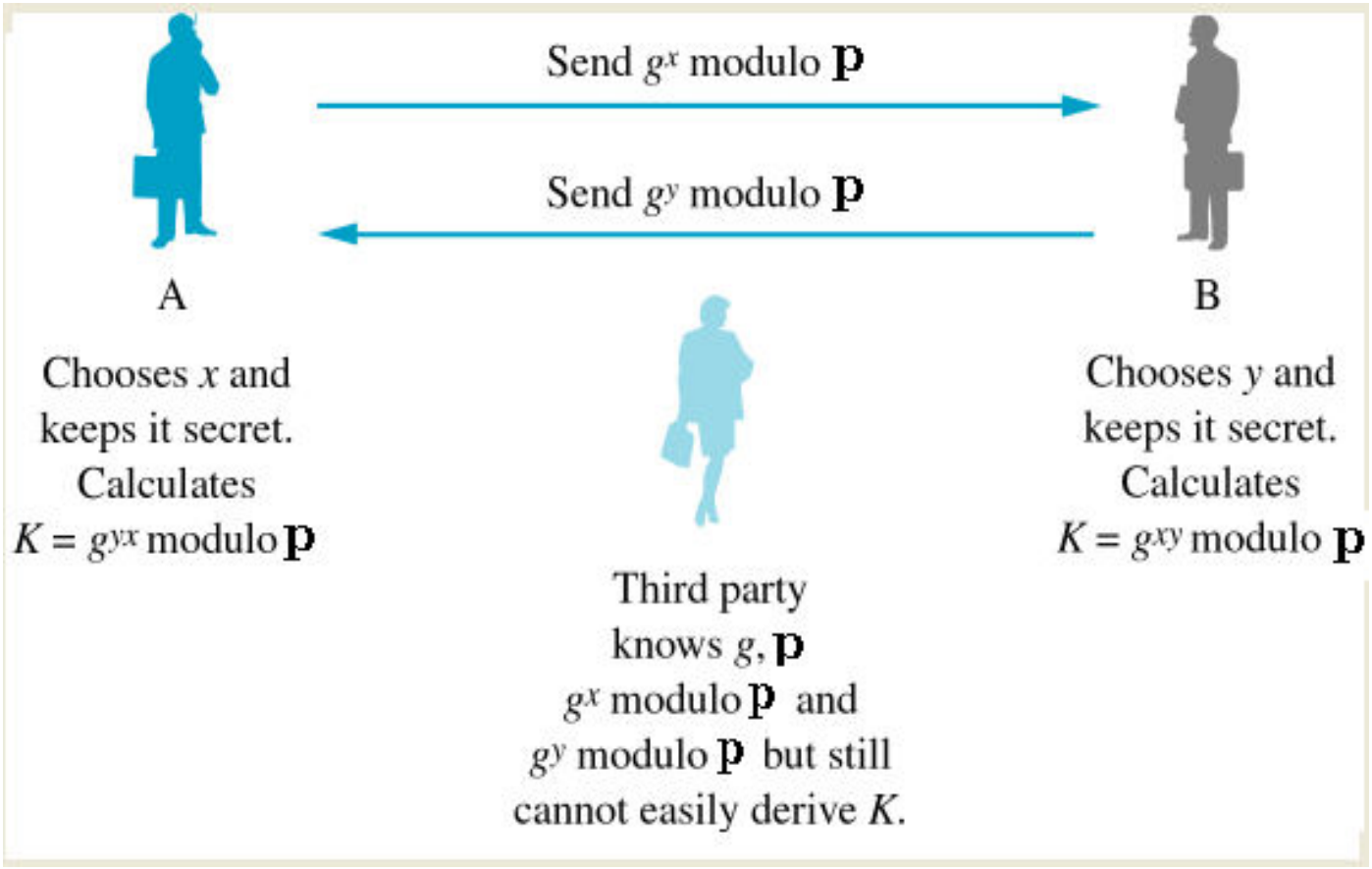
Diffie-Hellman Key exchange

- Look at *javax.crypto.interface at*

http://docs.oracle.com/javase/6/docs/api/javax/crypto/interfaces/package-summary.html

- Suppose *A* and *B* agree to use two ints (*g* and *p*) in the calculation of an encryption key.
  - These do not need to be kept secret.
- A chooses x privately and calculates $g^x \bmod p$ and sends this to B.
- B chooses y privately and calculates $g^y \bmod p$ and sends this to A
- B calculates $(g^x \bmod p)^y$ which is the same as $g^{xy} \bmod p$
- *A* calculates $(g^y \bmod p)^x$ which is the same as $g^{xy} \bmod p$

- *A* and *B* both have $K = g^{xy} \bmod p$ which they can use as the encryption key.
- Suppose a 3rd eavesdropping party knows *g*, *p*, $g^x \bmod n$, $g^y \bmod n$
- Still cannot derive $g^{xy} \bmod p$ (if nos are large).

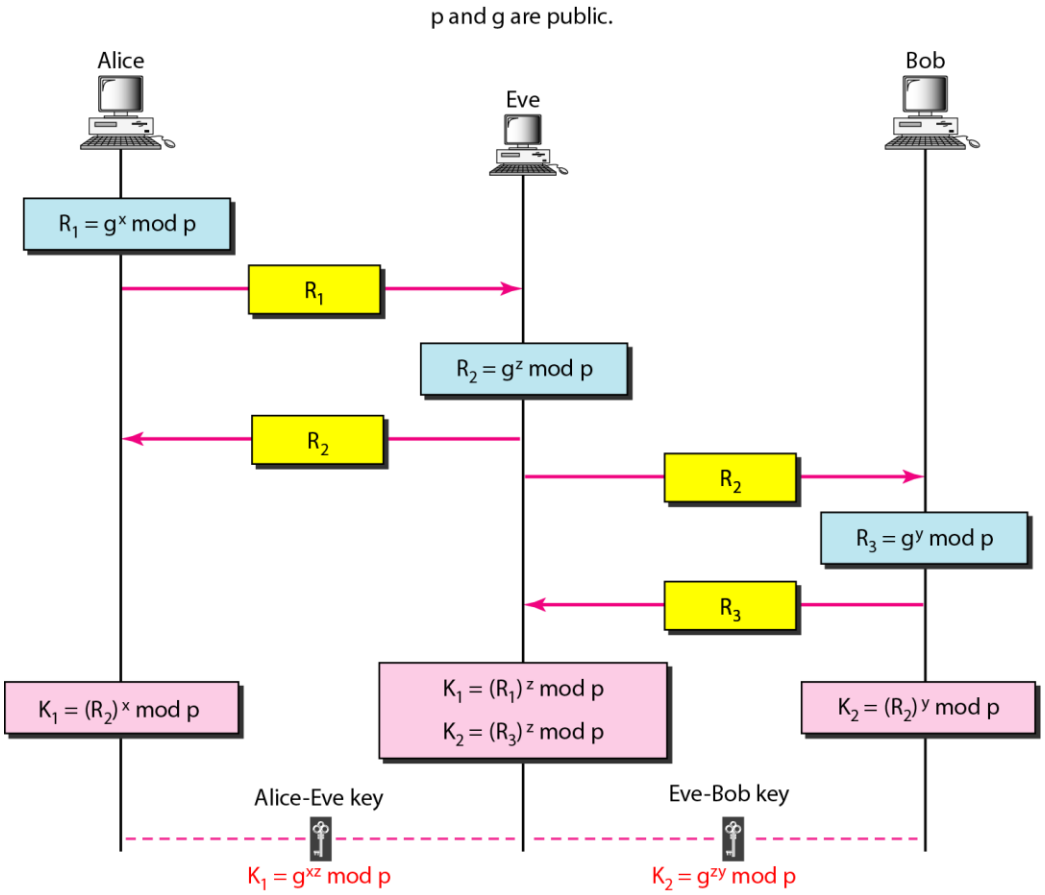

Send g^x modulo $\mathbf{p}$
Send g^y modulo $\mathbf{p}$
A
B
Chooses x and keeps it secret. Calculates $K = g^{yx}$ modulo $\mathbf{p}$
Chooses y and keeps it secret. Calculates $K = g^{xy}$ modulo $\mathbf{p}$
Third party knows g, $\mathbf{p}$ g^x modulo $\mathbf{p}$ and g^y modulo $\mathbf{p}$ but still cannot easily derive K.

- EX: g=7, p=17, A chooses x=5, and B chooses y=9
- A sends $7^5 \bmod 17 = 11$ to B
- B sends $7^9 \bmod 17 = 10$ to A
- A calculates $10^5 \bmod 17 = 6$
- B calculates $11^9 \bmod 17 = 6$
- Encryption key is 6

- Can find references to Diffie-Hellman as part of JCE (Java Cryptographic Extension) and in references to X.509 certificates to authenticate web sites that do business. [http://java.sun.com/j2se/1.4.2/docs/guide/security/jce/JCERefGuide.html]

- Diffie –Hellman does have a weakness – the *man-in-the-middle attack*.
- Person in between *A and B* intercepts the transmitted calculations and fools each one into thinking each is communicating with the other.

Figure 30.28 *Man-in-the-middle attack*



- Also called bucket brigade attack.
- Alice needed some way to authenticate to whom she was sending.
- Bob needed some way to authenticate to whom he was sending.
- This is called *Authenticaion.*
- That’s coming up in the next set of slides.

Zadaci-4:

18. Polialafabetsko šifrovanje, predstavlja način da se promeni frekvencija pojavljivanja slova i uobičajene sekvence je da se koristi polialfabetsko šifrovanje. Za razliku od monoalfabetkskog šifrovanja, svaki karakter se menja uvek različitim šifrovanim karakterom. Algoritm možemo definisati na sledeći način.

- Rešenje:

```
for ( int i  = 0; i < lenght of P; i++ )
    C[i]=P[i]+K+(i mod 3)
```

Uz datu pretpostavku da je K=1. Zatim se dodaje 1 na ASCII kodove na pozicijama 0, 3, 6. itd. Zatim se dodaje 2 na pozicijama 1, 4, 7, i tako dalje. Tri se dodaje na pozicijama 2, 5, 8 i tako redom. Na primer T ako zauzima poziciju 0 u nizu odnosn u reči, uvećava se za 1 te prelazi odnosno šifruje se u slovo U. Slovo H koje se nalazi na poziciji 1, uvećava se za 2 i prelazi u slovo J, i tako redom. Šifrujte ovom metod sledeći tekst: ŠIFROVANI TEKST.

19. Iako mnogo algoritama i ključeva rešavaju pitanja bezbednosti , algoritam RSA (nazvan po svojim pronalazačima, Ronu Rivestu, Adiju Shamiru i Leonardu Adlemenu) postao je skoro sisnonim za kriptografiju javnim ključem. Korišćenjem RSA izaberite p=3 i q=11 i šifrujte reč „kod“. Primenite algoritam za šifrovanje odnosno dešifrovanje. Da bi rekonstruisali prvobitnu poruku otvorenog teksta.

$p = 3$ i $q = 11$ naosnovu datih vrednosti dobijamo $n = p * q = 33$ i $z = (p-1)*(q-1) = 20$ biramo broj $e = 9$, manji od n, koji nema zajedničkih činilaca sa brojem z. Zatim, biramo $d = 9$ tako da je, e*d -1 tačno deljivo sa z. Sledi $e * d = 81$ odnosno $e * d - 1 = 80$ gde je dobijena vrednost tačno deljivasa 24. Sada možemo koristiti RSA šifrovanje i dešifrovanje, korišćenjem dobijenih vrednosti: $n = 33$, $e = 9$ i $d = 9.$ javni ključ je K_B^+, je par brojeva (n,e); njegov privatni ključ, K_B^-, je par brojeva (n,d). Dve vrednosti su javne n i e. Dok je vrednost d tajna. (Celobrojni ostatak kada se ceobroj x podeli celim brojem n, označava se sa: x mod n).

karakter	m	m^e	$c = m^e \bmod n$ (šifrovani tekst)
k	8	13421778	29
o	5	1953125	20
d	12	5159780352	12

Šifrovani tekst	c^d	$m = c^d \bmod n$	Slovo otvorenog teksta
29	14507145975869	8	k
20	512000000000	5	o
12	5159780352	12	d