Modern Wireless Networks

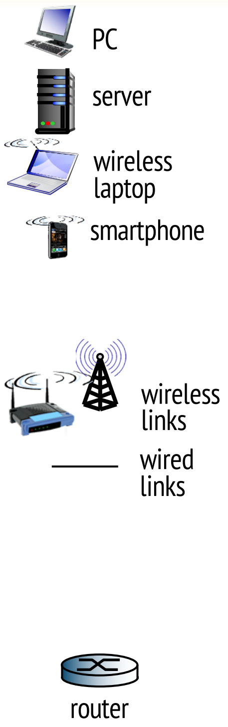
Wireless Fundamentals

UNIVERSITY AT ALBANY
State University of New York

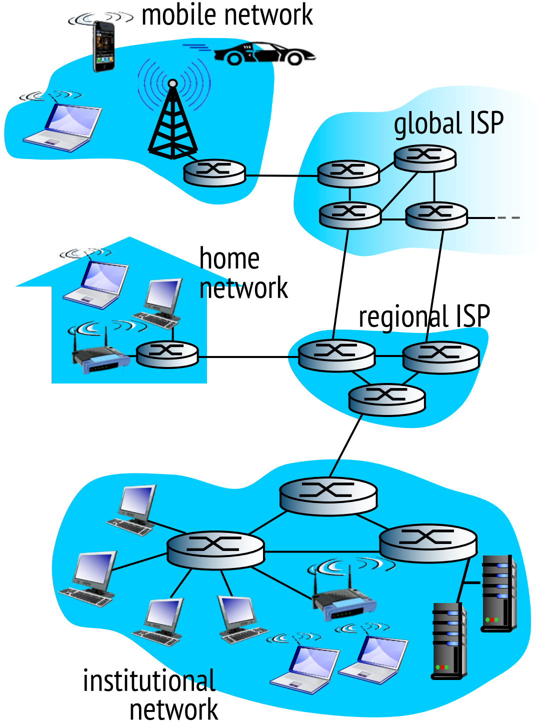
IECE 574– Spring 2021

Prof. Dola Saha

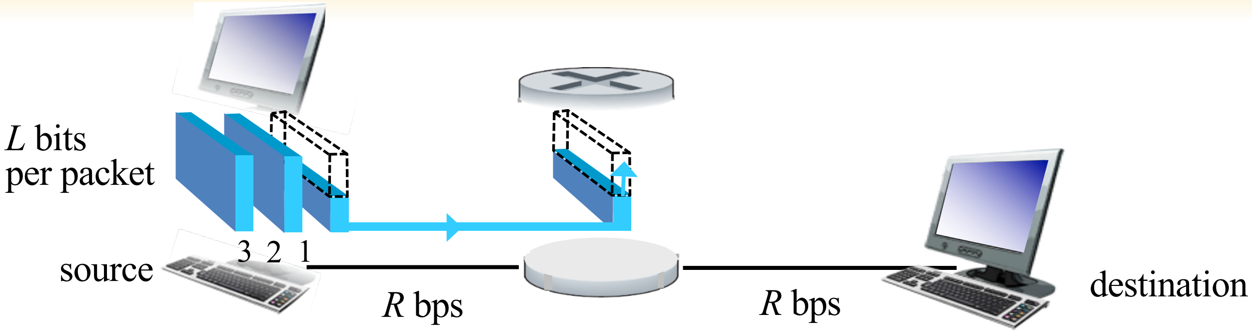
What's the Internet: nuts and bolts



- Packet switches: forward packets (chunks of data)
  - routers and switches
- Millions of connected computing devices:
  - hosts = end systems
  - running network apps
- Communication links
  - Fiber, copper, radio, satellite
  - Transmission rate: bandwidth



Packet-switching: store-and-forward



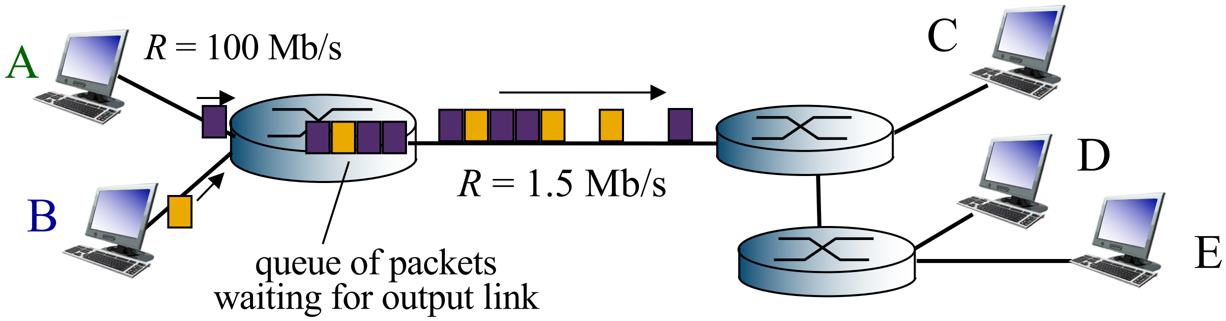
- takes L/R seconds to transmit (push out) L-bit packet into link at R bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
  - end-end delay = $2L/R$ (assuming zero propagation delay)

} more on delay shortly …

one-hop numerical example:

- L = 7.5 Mbits
- R = 1.5 Mbps
- one-hop transmission delay = 5 sec

Packet Switching: queueing delay, loss

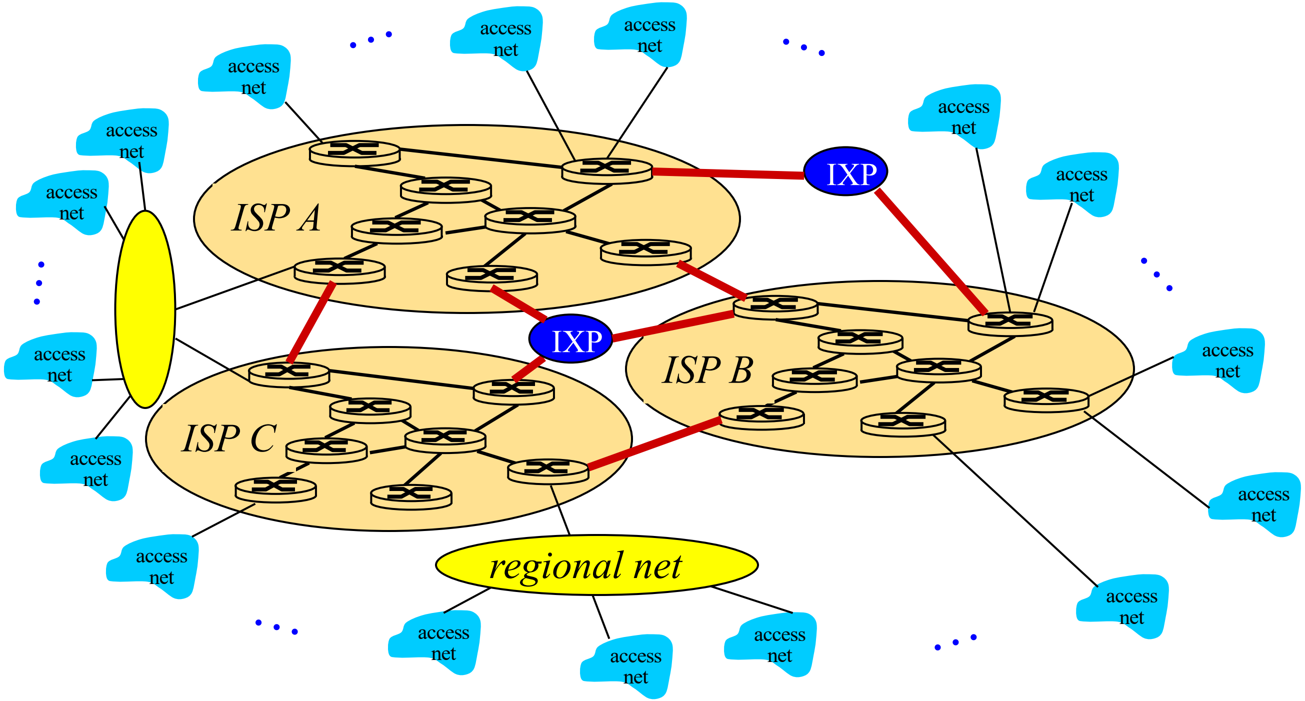


queuing and loss:

- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

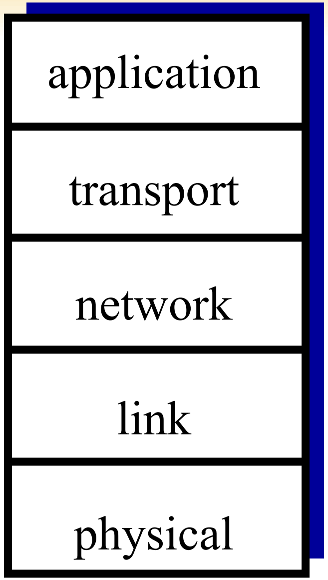
Internet structure: network of networks

… and regional networks may arise to connect access nets to ISPS



Internet Protocol Stack (RFC 1122, 1989)

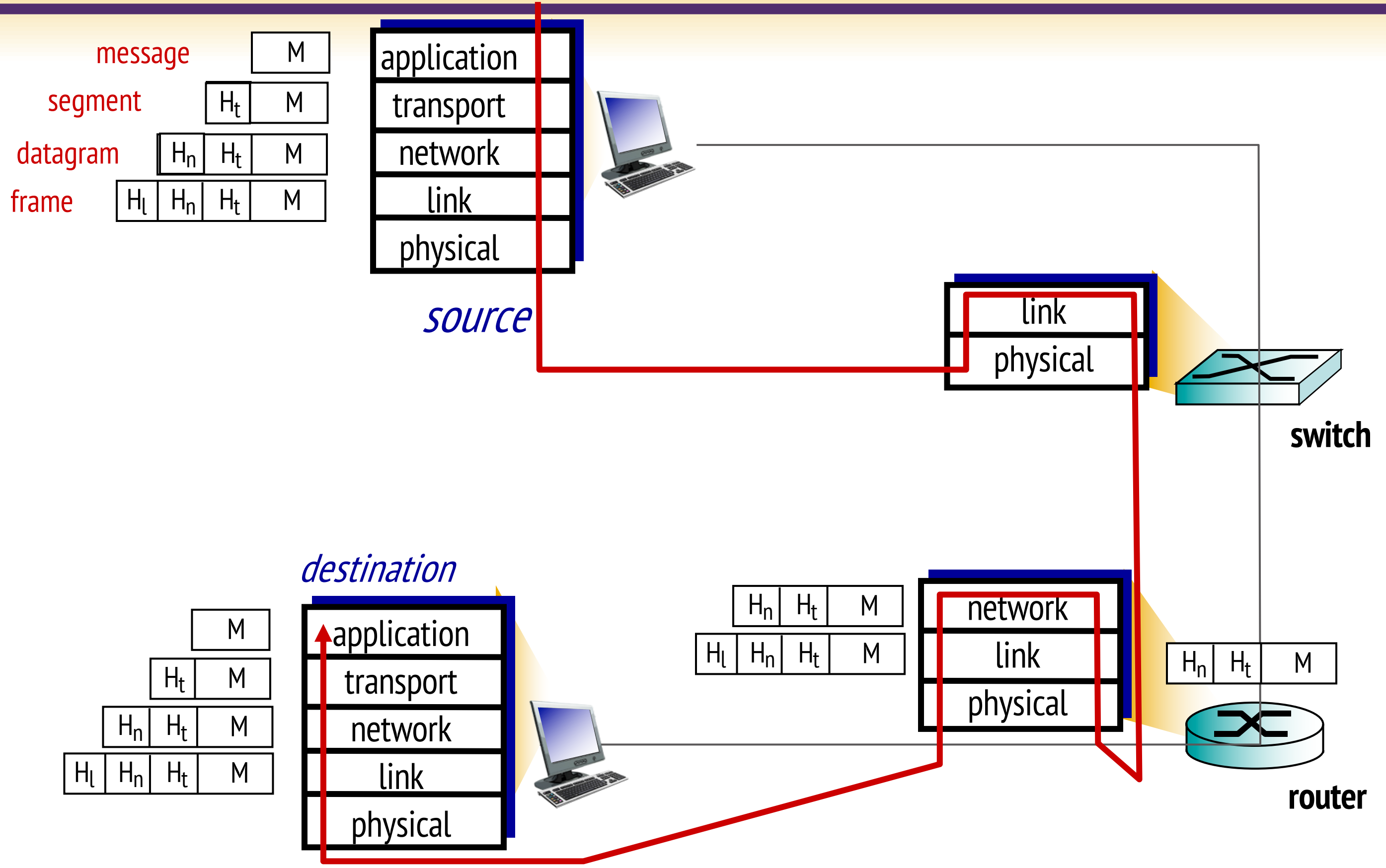
- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: process-process data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi)
- physical: bits “on the wire” / “over the air”



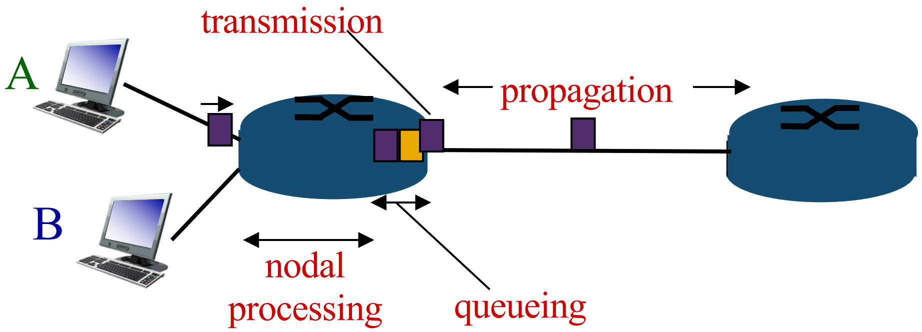
Why layering?

- dealing with complex systems:
  - explicit structure allows identification, relationship of complex system's pieces
    - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

Encapsulation



Four sources of packet delay



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

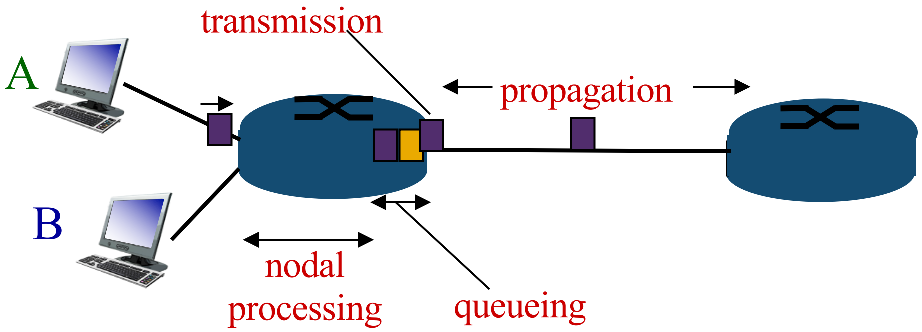
d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Four Sources of Packet Delay



$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{trans} and d_{prop} *very* different

d_{prop}: propagation delay:

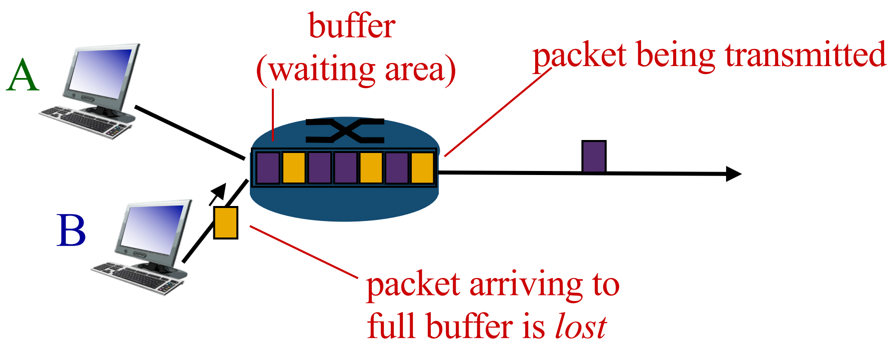
- d: length of physical link
- s: propagation speed in medium (~3×10^8 m/sec)
- $d_{prop} = d/s$

Synthesis: a day in the life of a web request

- journey down protocol stack complete!
  - application, transport, network, link
- putting-it-all-together: synthesis!
  - *goal:* identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

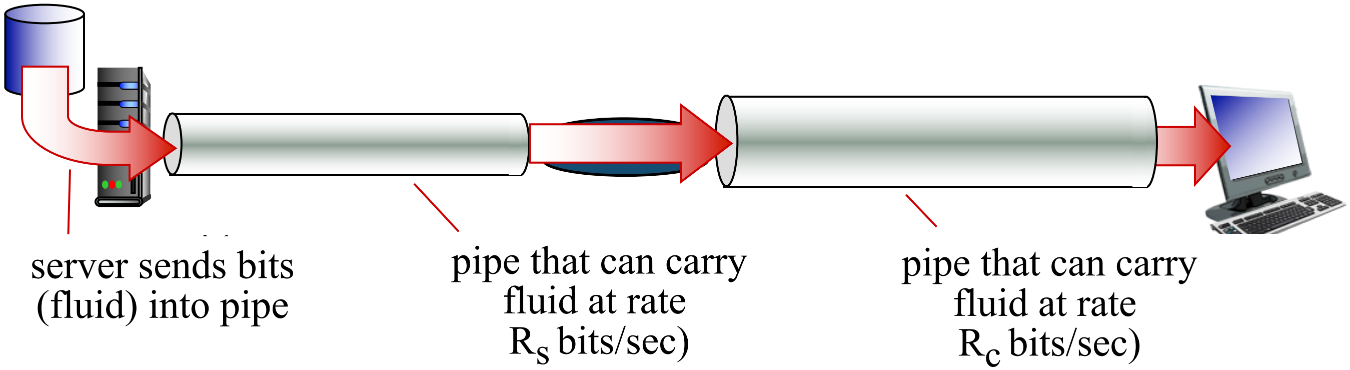
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



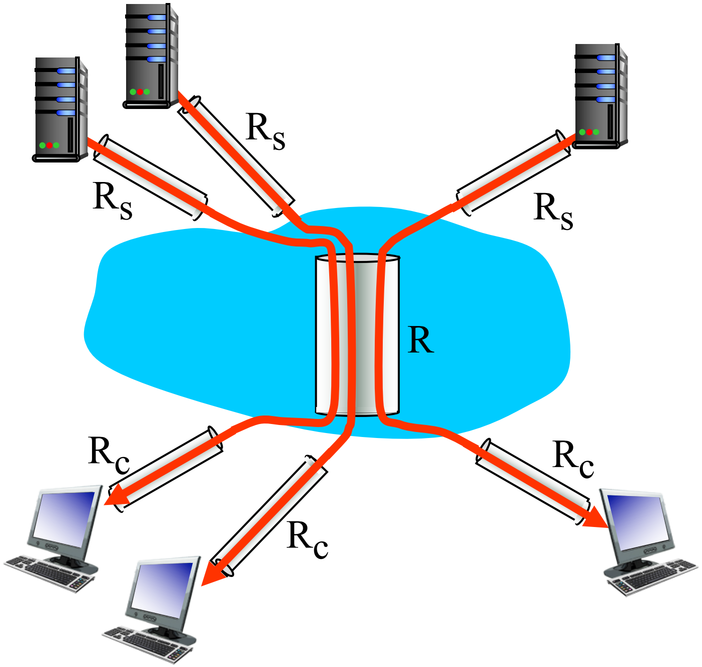
Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



Throughput: Internet scenario

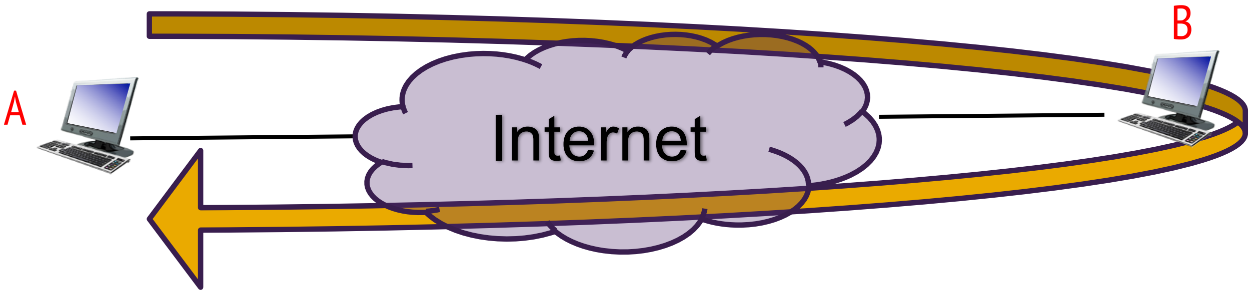
- per-connection end-end throughput: $\min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck



10 connections (fairly) share backbone bottleneck link R bits/sec

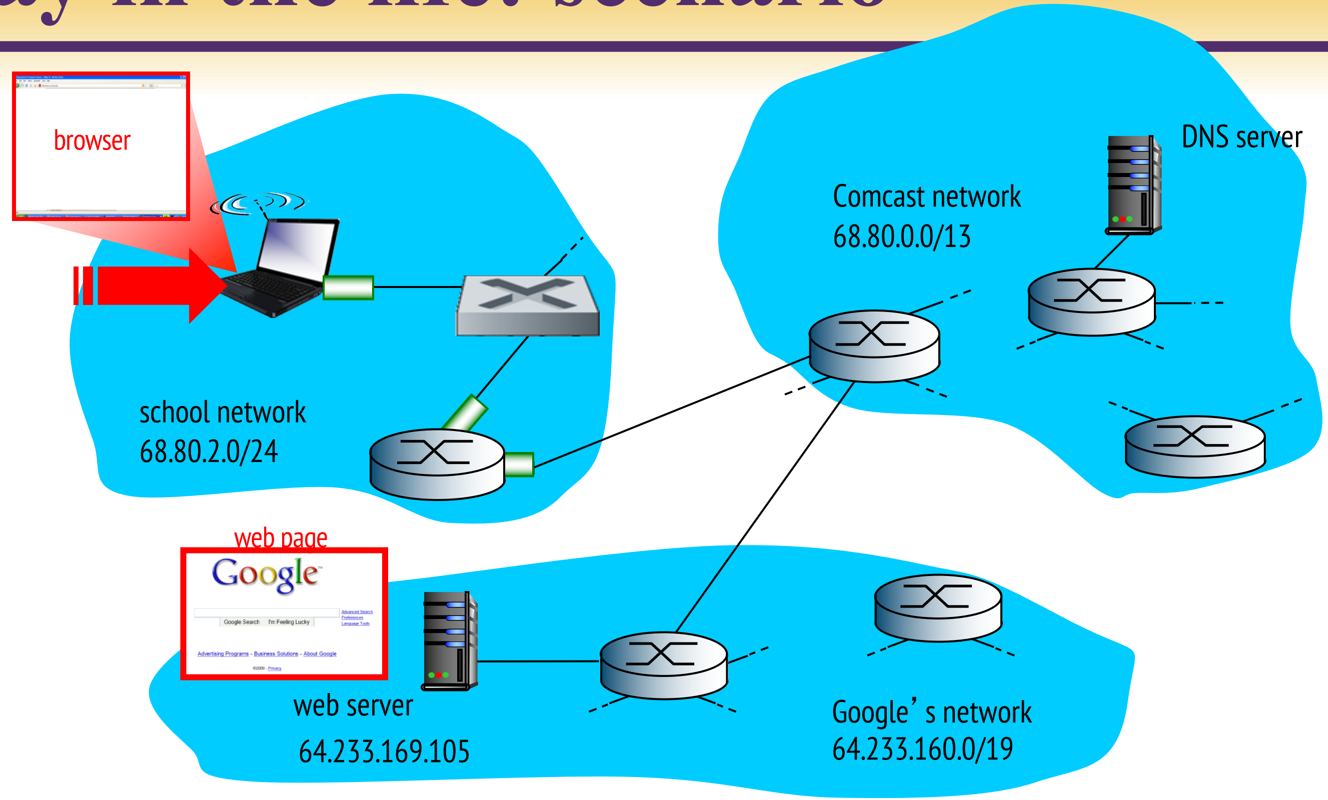
Performance

- Latency = Propagation + processing + transmit + queue
- Propagation = distance/speed of light
- Transmit = size/bandwidth
- Processing = depends on the node (hardware + software)
- RTT => Round Trip Time

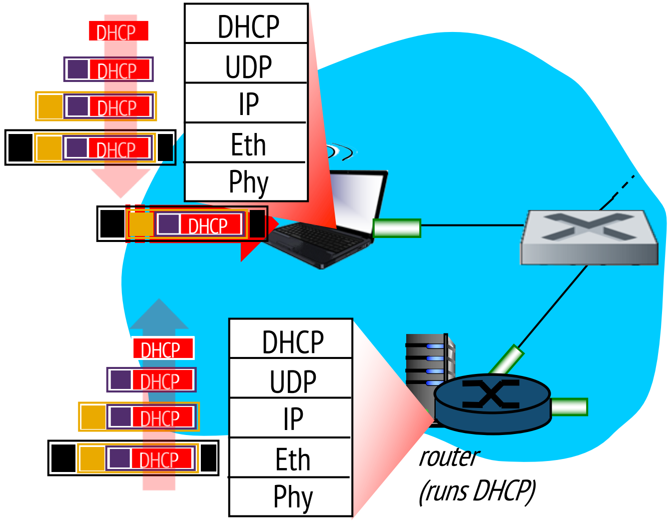


- One bit transmission => propagation is important
- Large bytes transmission => bandwidth is important

A day in the life: scenario

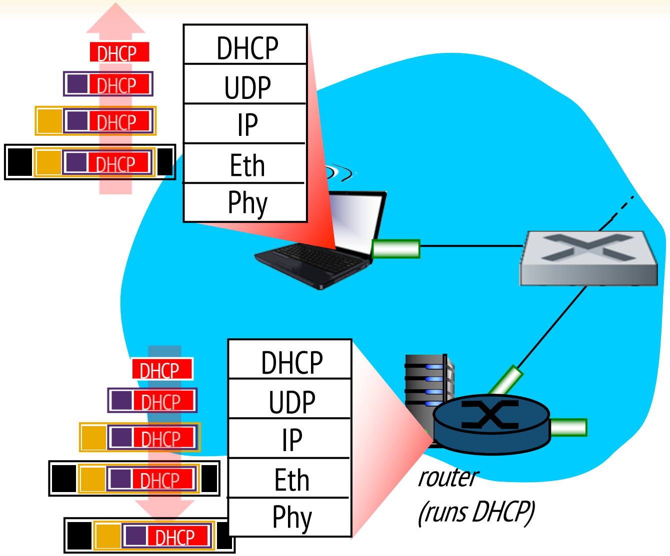


A day in the life… connecting to the Internet



- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*
  - DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet
  - Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
  - Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

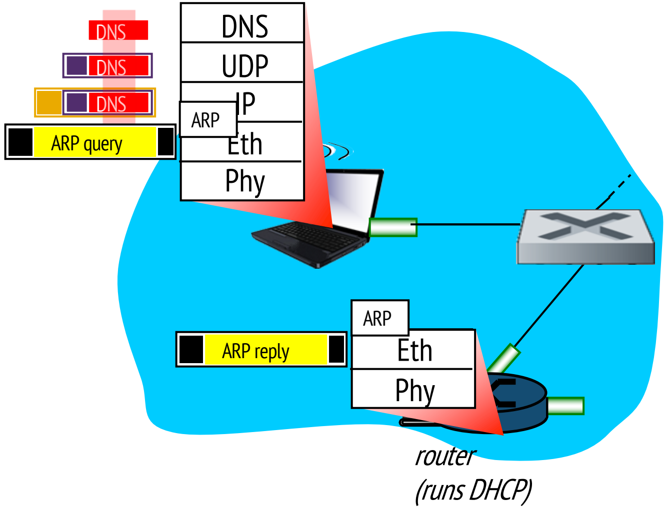
A day in the life… connecting to the Internet



- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
  - encapsulation at DHCP server, frame forwarded (switch learning) through LAN, demultiplexing at client
  - DHCP client receives DHCP ACK reply

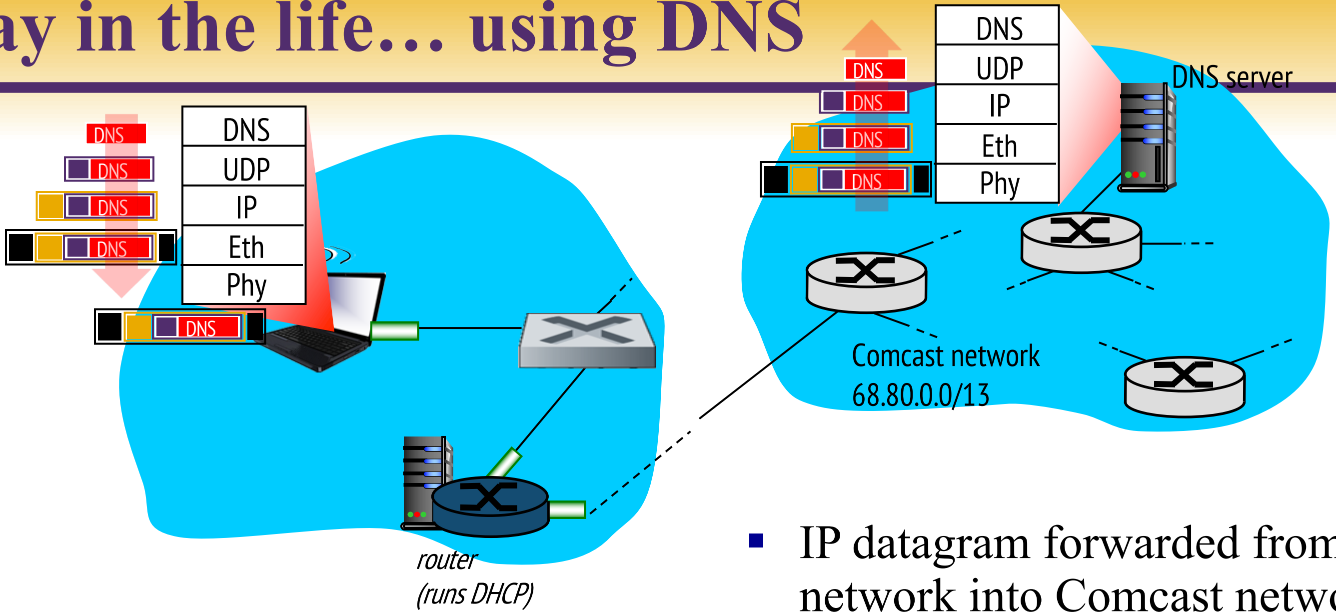
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

A day in the life… ARP (before DNS, before HTTP)



- before sending *HTTP* request, need IP address of www.google.com: *DNS*
  - DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. To send frame to router, need MAC address of router interface: ARP
  - ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface
  - client now knows MAC address of first hop router, so can now send frame containing DNS query

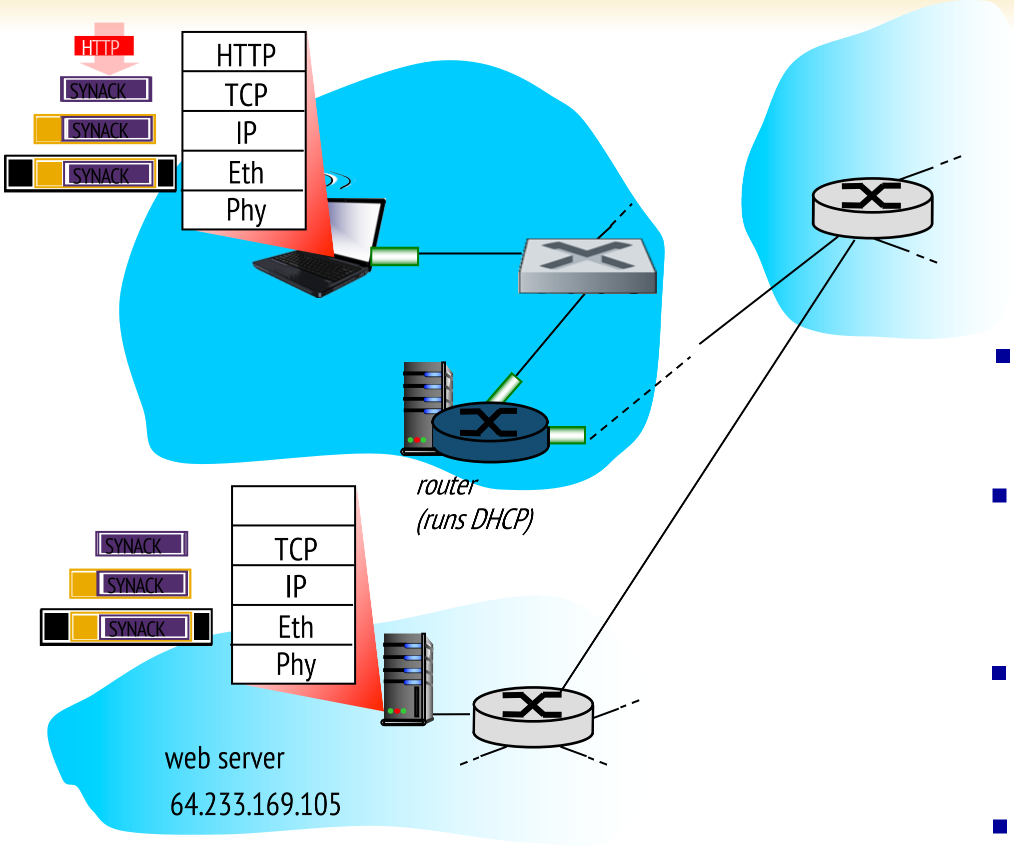
A day in the life… using DNS



- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

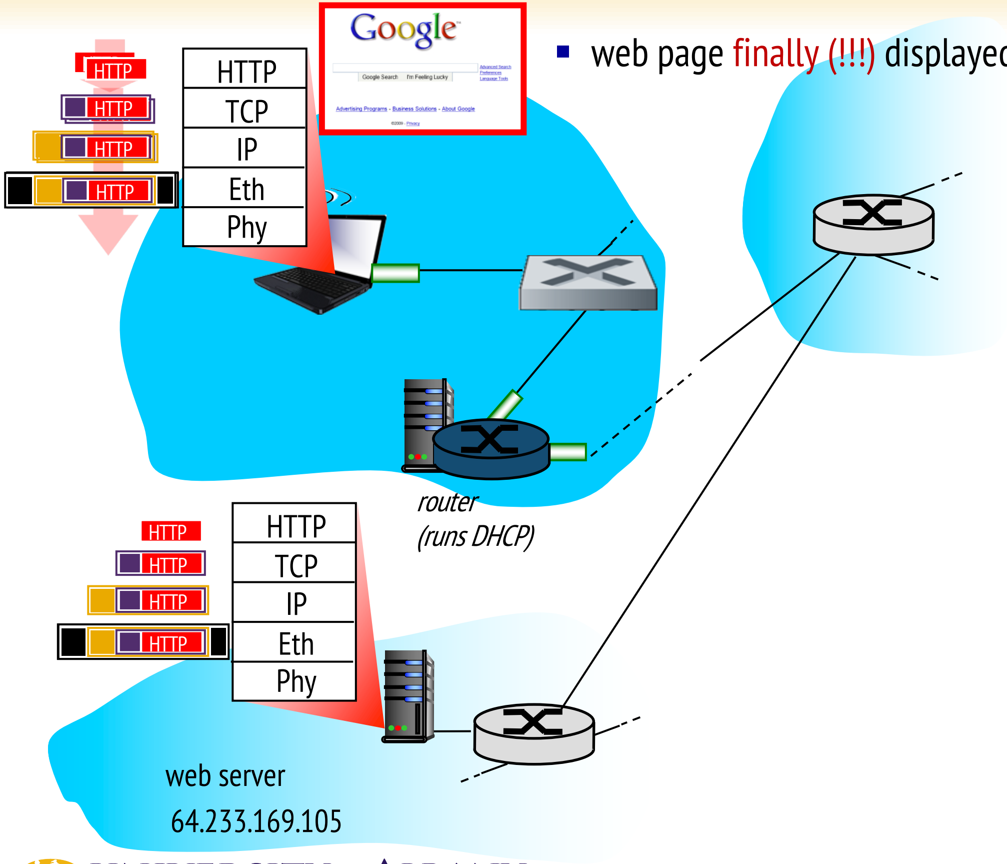
- IP datagram forwarded from campus network into Comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server
- demuxed to DNS server
- DNS server replies to client with IP address of www.google.com

A day in the life…TCP connection carrying HTTP



- to send HTTP request, client first opens TCP socket to web server
- TCP SYN segment (step 1 in 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK (step 2 in 3-way handshake)
- TCP connection established!

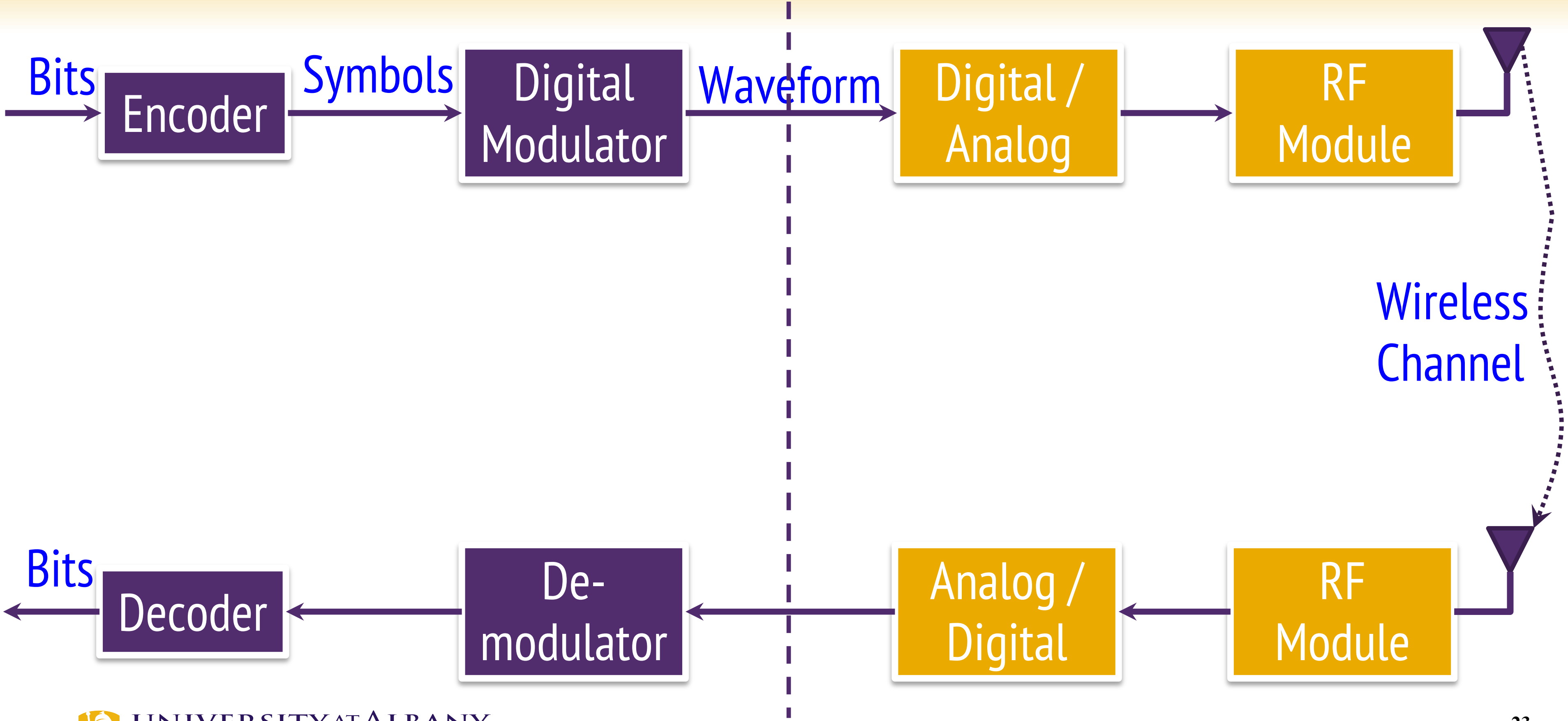
A day in the life… HTTP request/reply



- web page finally (!!!) displayed

- HTTP request sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with HTTP reply (containing web page)
- IP datagram containing HTTP reply routed back to client

Wireless Digital Communication System

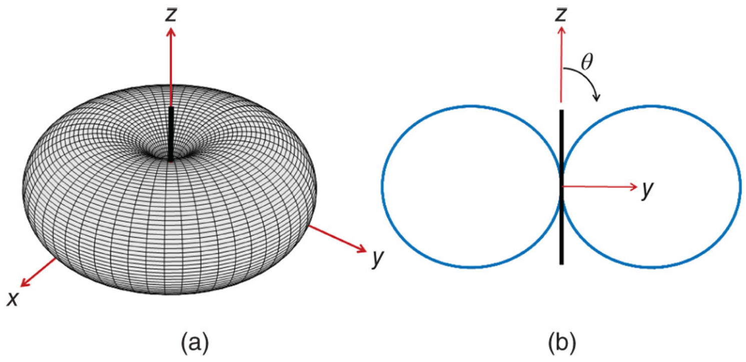


Antenna

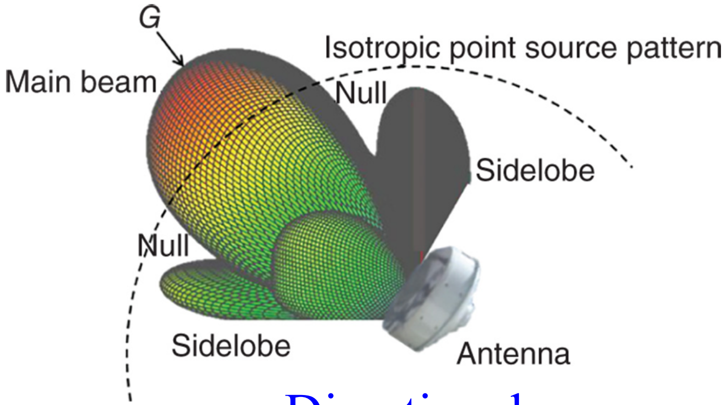
➢ The effective (or equivalent) isotropic radiated power (EIRP) of a transmitter $EIRP = P_t G_t$

➢ Antenna Gain $G = \frac{4\pi A_e}{\lambda^2} = \frac{4\pi f^2 A_e}{c^2}$

- A_e = Effective Area (area where the signal energy can be received)
- f = carrier frequency, c = speed of light, λ = wavelength



Omni-directional

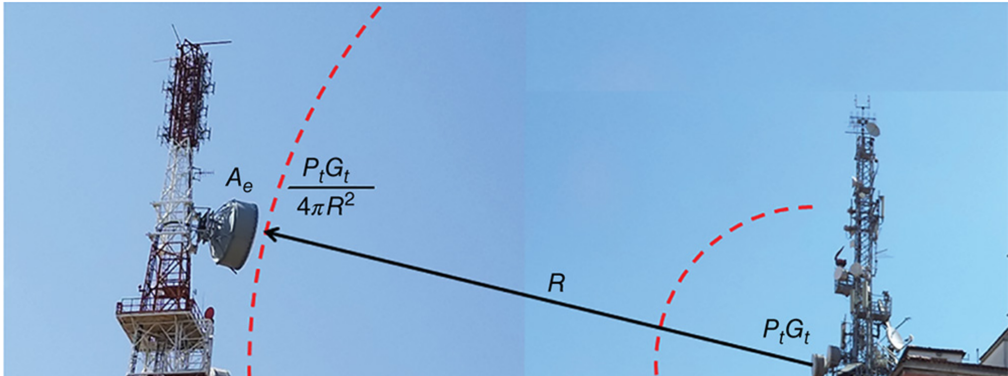


Directional

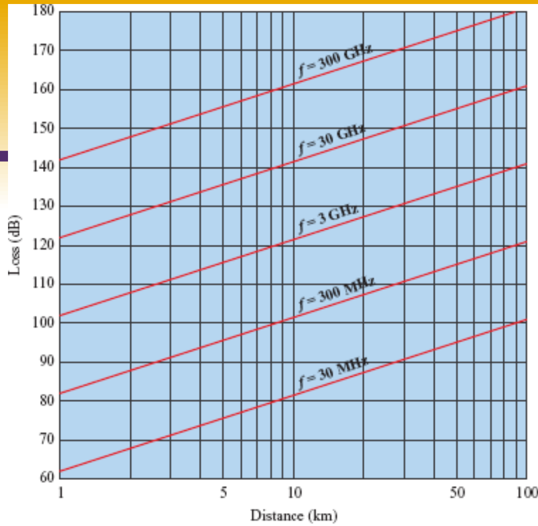
Free Space Propagation

- Free space propagation assumes a signal propagates in a vacuum with no obstacles
- Transmitter generates P_t , antenna magnifies with gain G_t
- Assume that transmitted signal travels in all directions (isotropic antenna)
- Power density at receive antenna is $\frac{P_t G_t}{4\pi R^2}$
- If receiver has an effective aperture A_e
- Received power is $P_r = \frac{P_t G_t A_e}{4\pi R^2}$

Friis transmission formula



Free Space Pathloss



- Free Space Pathloss $\frac{P_t}{P_r} = \frac{(4\pi d)^2}{\lambda^2} = \frac{(4\pi f d)^2}{c^2}$
- In dB

$$L_{dB} = 10\ \log \frac{P_t}{P_r} = 20\ \log\left(\frac{4\pi d}{\lambda}\right) = -20\ \log(\lambda) + 20\ \log(d) + 21.98\ \text{dB}$$

$$= 20\ \log\left(\frac{4\pi f d}{c}\right) = 20\ \log(f) + 20\ \log(d) - 147.56\ \text{dB}$$

as the frequency increases, the free space loss also increases

- Accounting for gain of antennas:

$$\frac{P_t}{P_r} = \frac{(4\pi)^2 (d)^2}{G_r G_t \lambda^2} = \frac{(\lambda d)^2}{A_r A_t} = \frac{(cd)^2}{f^2\ A_r A_t}$$

- In dB $L_{dB} = 20\ \log\ (\lambda) + 20\ \log\ (d) - 10\ \log\ (A_t A_r)$

$$= -20\ \log\ (f) + 20\ \log\ (d) - 10\ \log\ (A_t A_r) + 169.54\ \text{dB}$$

with antenna gain, the change in loss is measured by −20log(f)

Pathloss in Practical Systems

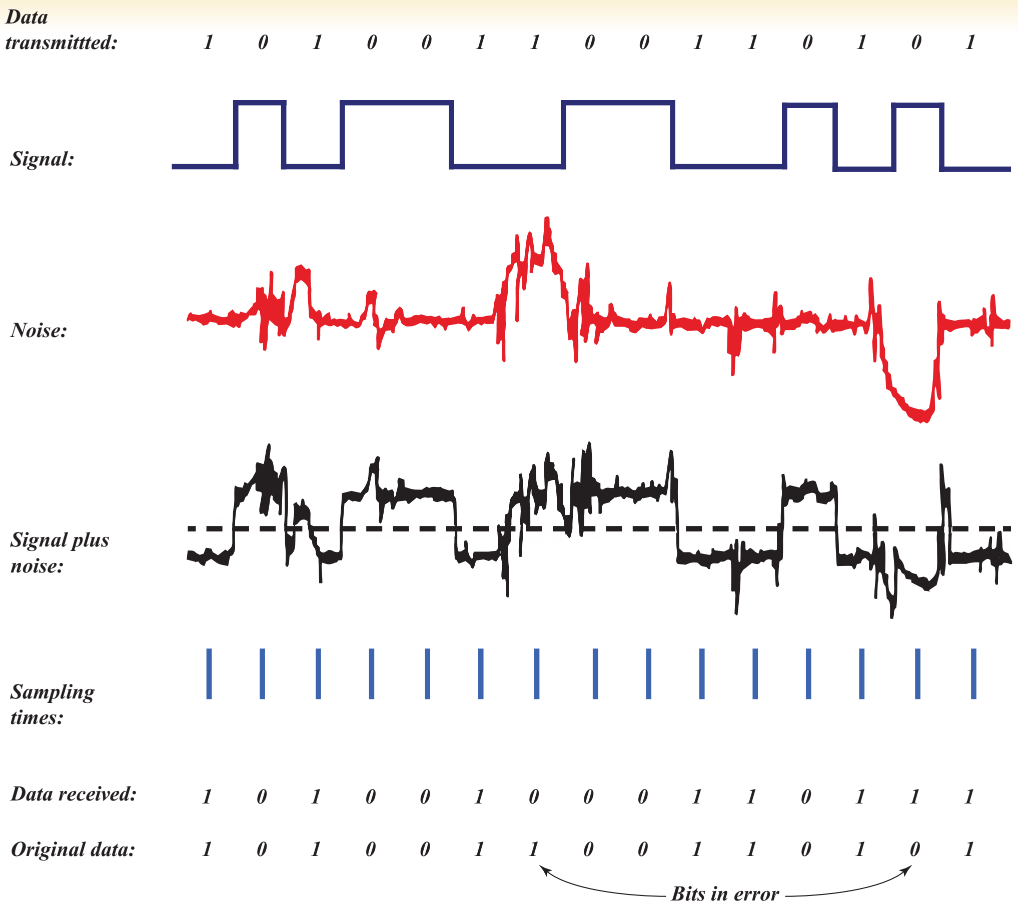
$$\frac{P_t}{P_r} = \left(\frac{4\pi}{\lambda}\right)^2 d^n = \left(\frac{4\pi f}{c}\right)^2 d^n$$

$$L_{dB} = 10\,\log\frac{P_t}{P_r} = 10\,\log\left(\left(\frac{4\pi}{\lambda}\right)^2 d^n\right) = -20\,\log(\lambda) + 10n\,\log(d) + 21.98\,\mathrm{dB}$$

$$= 10\,\log\left(\left(\frac{4\pi f}{\mathrm{c}}\right)^2 d^n\right) = 20\,\log(f) + 10n\,\log(d) - 147.56\,\mathrm{dB}$$

Environment	Path Loss Exponent, n
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Channel


Data transmittted: 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1
Signal:
Noise:
Signal plus noise:
Sampling times:
Data received: 1 0 1 0 0 1 0 0 0 1 1 0 1 1 1
Original data: 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1
Bits in error

Signal to Noise Ratio

- Ratio of the power in a signal to the power contained in the noise that's present at a particular point in the transmission
- Typically measured at a receiver
- Signal-to-noise ratio (SNR, or S/N)

$$(SNR)_{\text{dB}} = 10\log_{10}\frac{\text{signal power}}{\text{noise power}}$$

- A high SNR means a high-quality signal
- SNR sets upper bound on achievable data rate

Maximum Achievable Channel Capacity

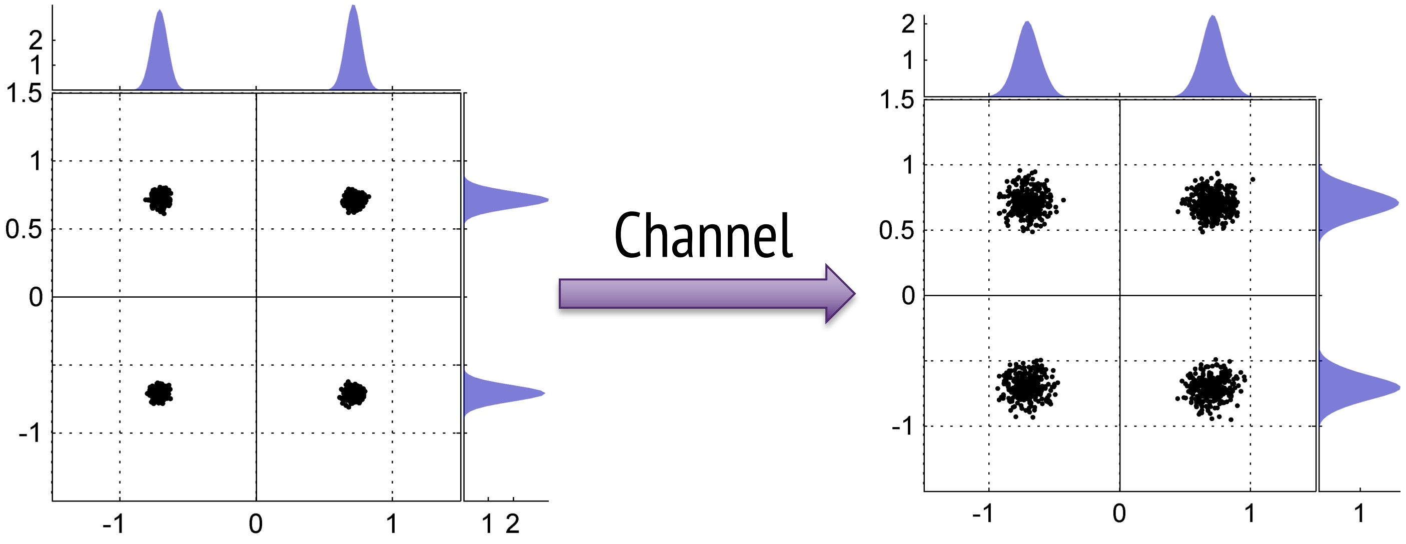
- Shannon's Capacity

$$C = B \log_2 (1 + \text{SNR})$$

  - C is the capacity of the channel in bits per second
  - B is the bandwidth of the channel in Hertz

Channel Effects

- Transmitted and Received QPSK Signal

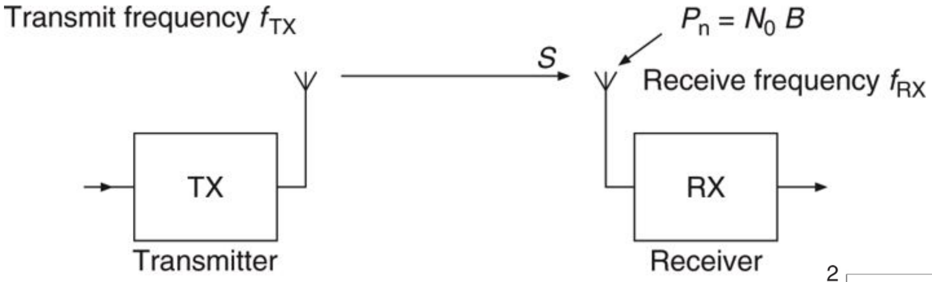


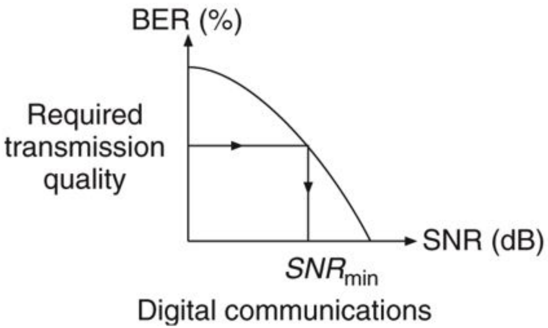
Transmitted

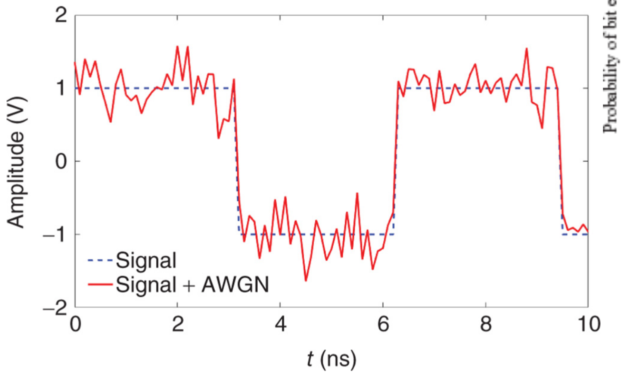
Received

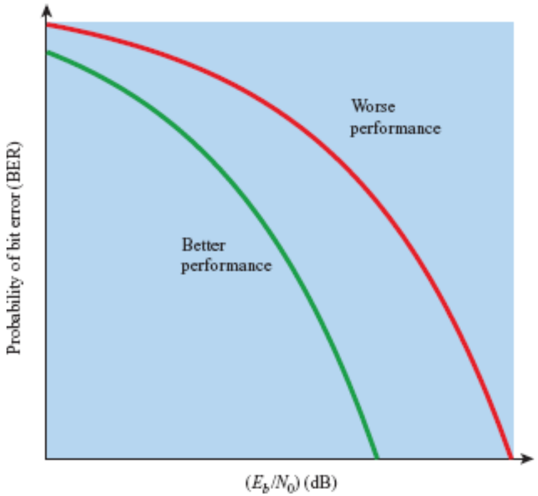
Signal to Noise Ratio

- How to calculate SNR from a received signal trace?









bit error rate (BER) = number of bits in error divided by the total number of bits over a long interval

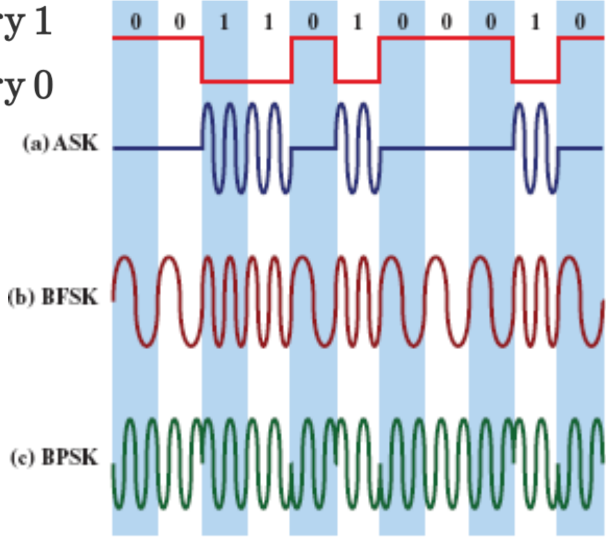
Modulation

- Amplitude Shift Keying
- Frequency Shift Keying
- Phase Shift Keying

ASK $$s(t) = \begin{cases} A\cos(2\pi f_c t) & \text{binary } 1 \\ 0 & \text{binary } 0 \end{cases}$$

BFSK $$Bs(t) = \begin{cases} A\cos(2\pi f_1 t) & \text{binary } 1 \\ A\cos(2\pi f_2 t) & \text{binary } 0 \end{cases}$$

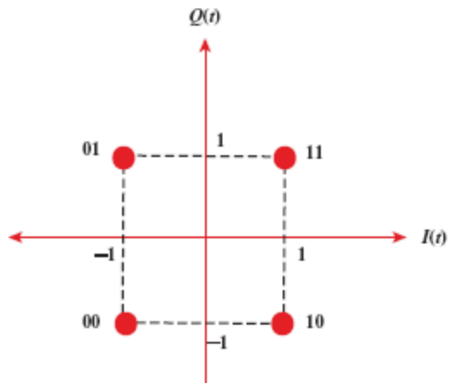
BPSK $$s(t) = \begin{cases} A\cos(2\pi f_c t) \\ A\cos(2\pi f_c t + \pi) \end{cases} = \begin{cases} A\cos(2\pi f_c t) & \text{binary } 1 \\ A\cos(2\pi f_c t) & \text{binary } 0 \end{cases}$$



Higher Order PSK

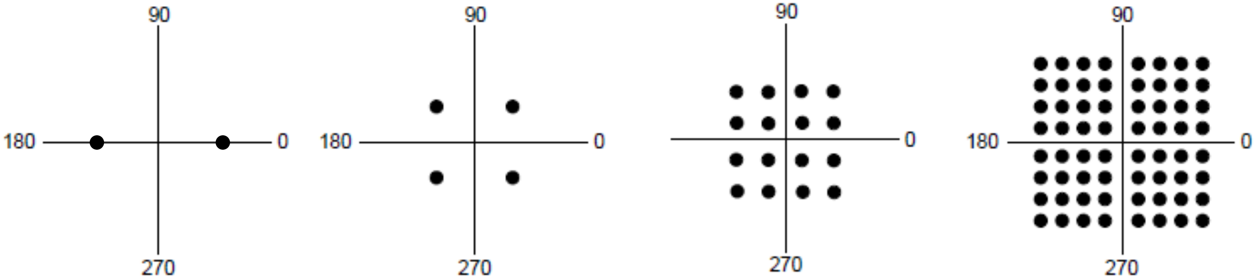
QPSK

$$s(t) = \begin{cases} A\cos\left(2\pi f_c t + \frac{\pi}{4}\right) & 11 \\ A\cos\left(2\pi f_c t + \frac{3\pi}{4}\right) & 01 \\ A\cos\left(2\pi f_c t - \frac{3\pi}{4}\right) & 00 \\ A\cos\left(2\pi f_c t - \frac{\pi}{4}\right) & 10 \end{cases}$$



Quadrature Amplitude Modulation

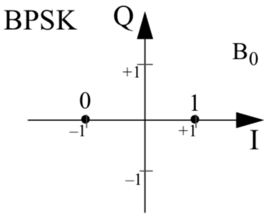
- Commonly used in Wireless Standards
- Combination of ASK & PSK

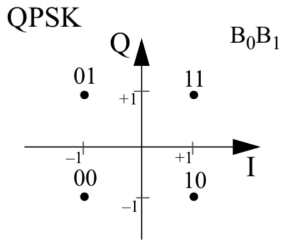


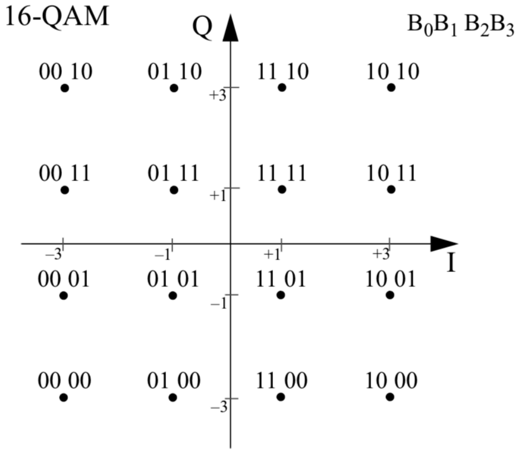
BPSK	QPSK	QAM-16	QAM-64
2 symbols	4 symbols	16 symbols	64 symbols
1 bit/symbol	2 bits/symbol	4 bits/symbol	6 bits/symbol

Energy Normalization

- Goal: Average Power of Modulated Signal remains unity



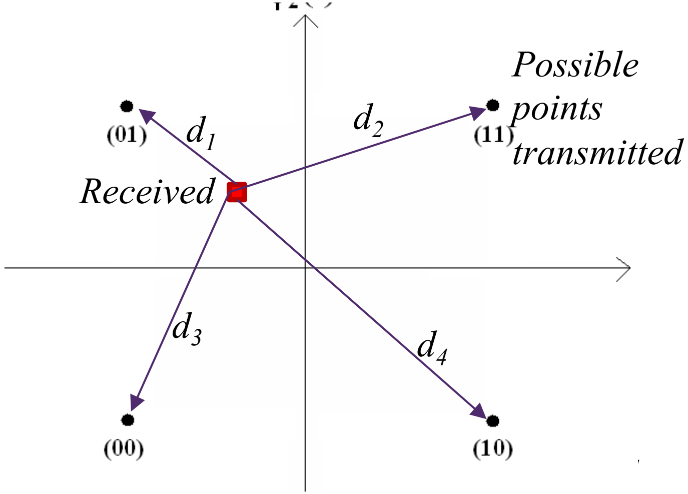




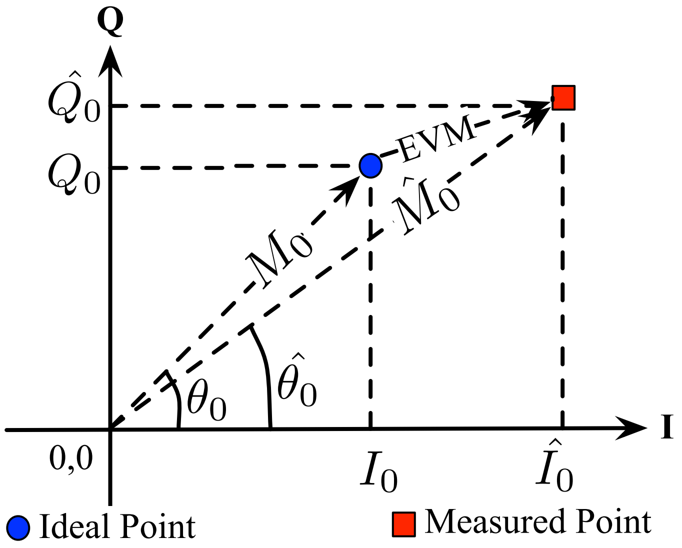
Modulation	K_{MOD}
BPSK	1
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

Demodulation

- Compute Euclidean distance between received value and all possible transmitted values
- Map to the value, which is minimum
- Decision Boundary
  - Half the distance between two nearest signal constellations



Error Vector Magnitude (EVM)



$\{I_0, Q_0, M_0, \theta_0\}$ = Ideal I, Q, Magnitude, Phase

$\{\hat{I}_0, \hat{Q}_0, \hat{M}_0, \hat{\theta}_0\}$ = Measured I, Q, Magnitude, Phase

$$EVM = \sqrt{\left(I_0 - \hat{I}_0\right)^2 + \left(Q_0 - \hat{Q}_0\right)^2}$$

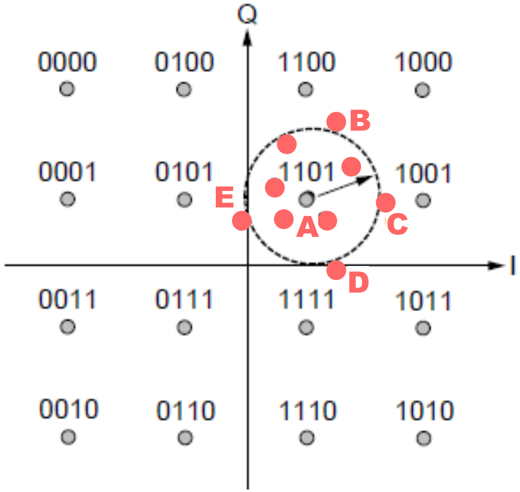
I_{Disp} = Dispersion in I $= I_0 - \hat{I}_0$

Q_{Disp} = Dispersion in Q $= Q_0 - \hat{Q}_0$

M_{Disp} = Dispersion in Magnitude $= M_0 - \hat{M}_0$

Gray Coding

- Gray-coding assigns bits to symbols so that small symbol errors cause few bit errors:



When 1101 is sent:

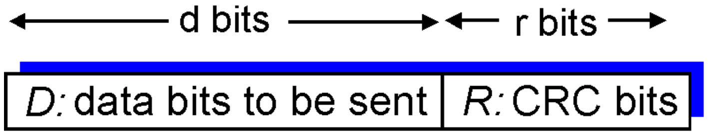
Point	Decodes as	Bit errors
A	1101	0
B	1100	1
C	1001	1
D	1111	1
E	0101	1

Channel Coding

- Wireless channel introduces errors in burst
- Coping with transmission error
- Error Detection Codes
  - Detect error and retransmit
  - Example: Cyclic Redundancy Check (CRC)
- Error Correcting Codes (Forward Error Correction)
  - transmit enough redundant data to allow receiver to recover from errors
  - No sender retransmission required.
  - Examples: Block codes, Cyclic codes, Reed-Solomon codes, Convolutional codes, Turbo codes, etc.

Cyclic Redundancy Check (CRC)

- powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)



bit pattern

$D * 2^r$ XOR R

mathematical formula

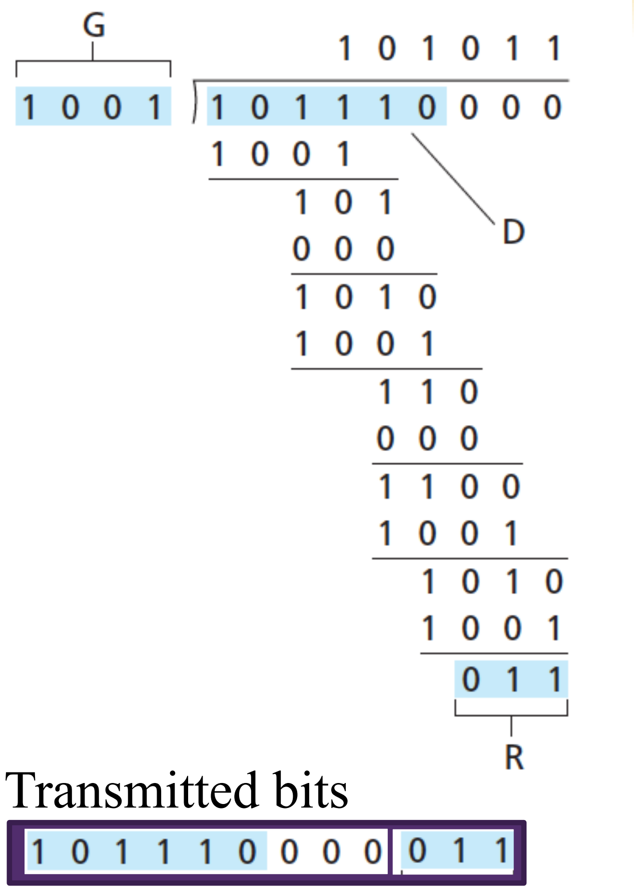
Modulo 2 Arithmetic

- CRC Calculations are done in modulo-2 arithmetic.
  - Without carries and borrows in addition and subtraction
- Addition & Subtraction are identical and equivalent to bitwise XOR.
- Multiplication and division are same as in base-2 arithmetic.

CRC Example

- want:
  - $D.2^r$ XOR R = nG
- equivalently:
  - $D.2^r$ = nG XOR R
- equivalently:
  - if we divide $D.2^r$ by G, we want remainder R to satisfy:

$$R = remainder\frac{D.2^r}{G}$$



Homework

- Consider the 5-bit generator, G=10011. Suppose D has a value of 1010101010. What is the value of R?

Cyclic Redundancy Check (CRC)

- Six generator polynomials that have become international standards are:
  - CRC-8 = x^8+x^2+x+1
  - CRC-10 = $x^{10}+x^9+x^5+x^4+x+1$
  - CRC-12 = $x^{12}+x^{11}+x^3+x^2+x+1$
  - CRC-16 = $x^{16}+x^{15}+x^2+1$
  - CRC-CCITT = $x^{16}+x^{12}+x^5+1$
  - CRC-32 = $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

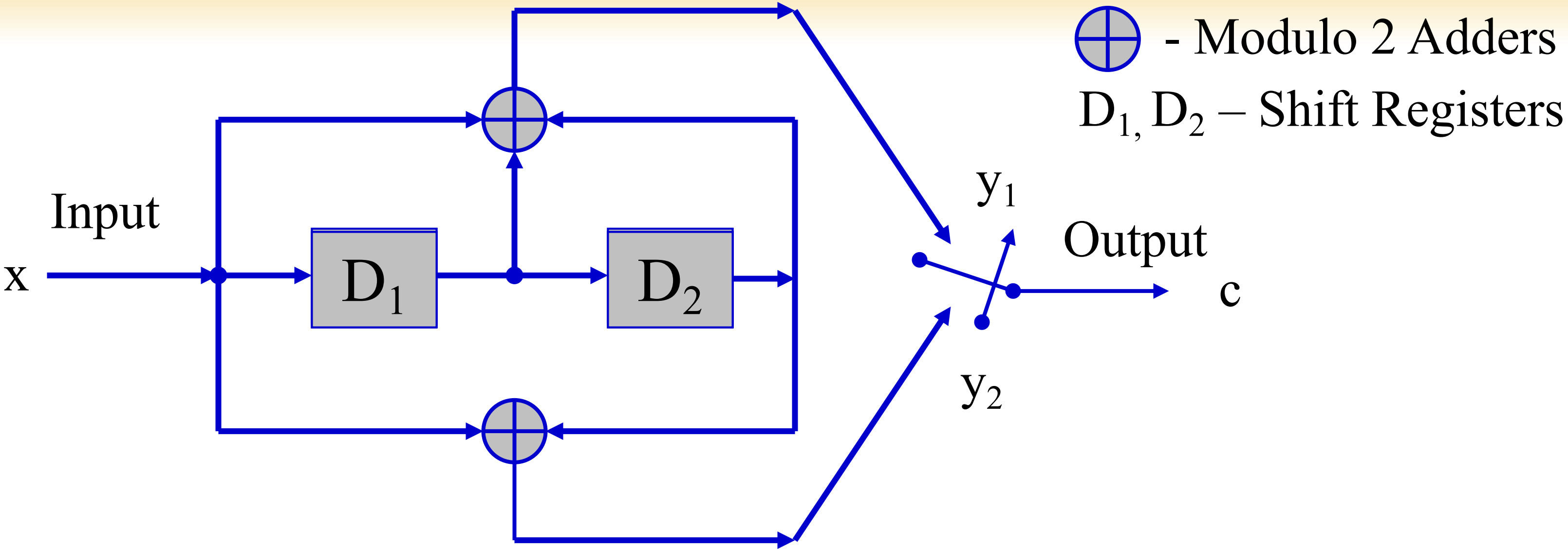
Convolution Codes

- Most widely used channel code
- encoded bits depend not only on the current input data bits but also on past input bits
- Decoding is mostly performed by the Viterbi Algorithm
- Constraint length K for a convolution code is defined as K=M+1
  - M is the maximum number of stages in any shift register

Convolution Codes

- Shift registers store the state information of the convolutional encoder
- Constraint length relates the number of bits on which the output depends
- The code rate r is defined as r = k/n
  - k is the number of parallel information bits
  - n is the the number of parallel output encoded bits at one time interval
- Initial values of shift registers is 0

Convolution Encoder (n=2, k=1, r=1/2)

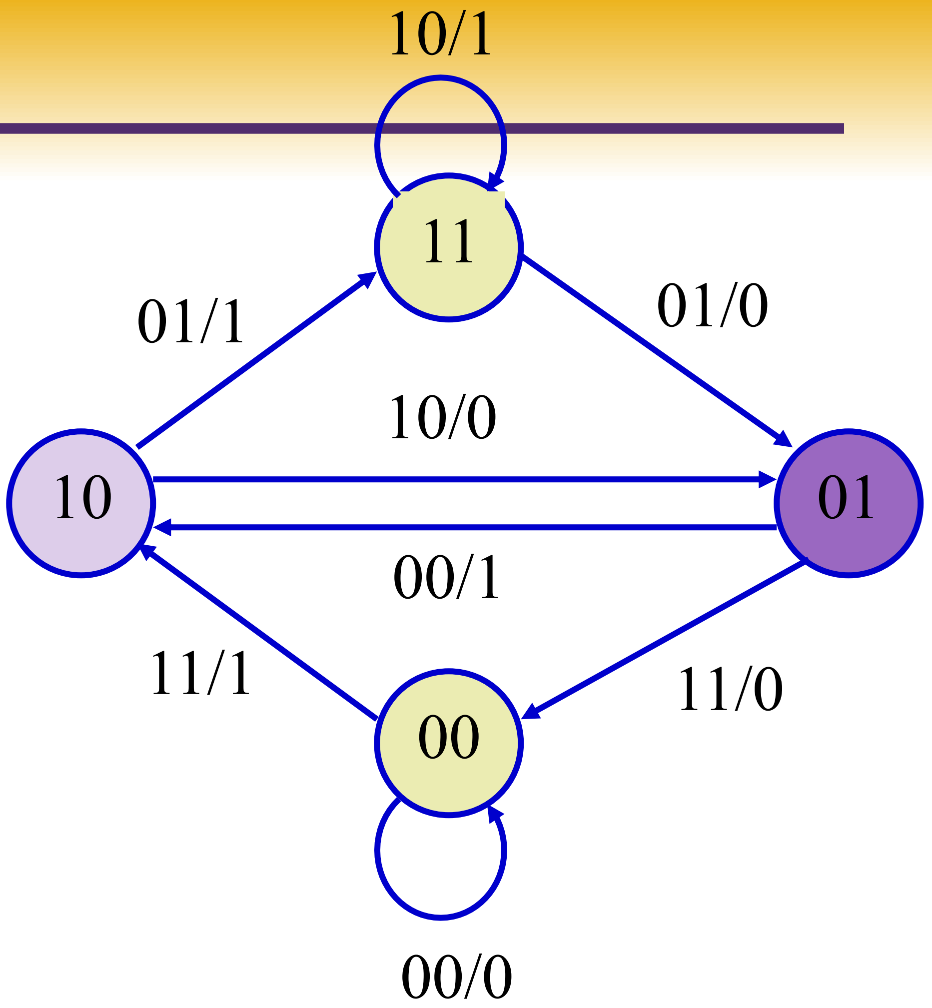


Input x:	1	1	1	0	0	0	…
Output y_1y_2:	11	01	10	01	11	00	…

Input x:	1	0	1	0	0	0	…
Output y_1y_2 :	11	10	00	10	11	00	…

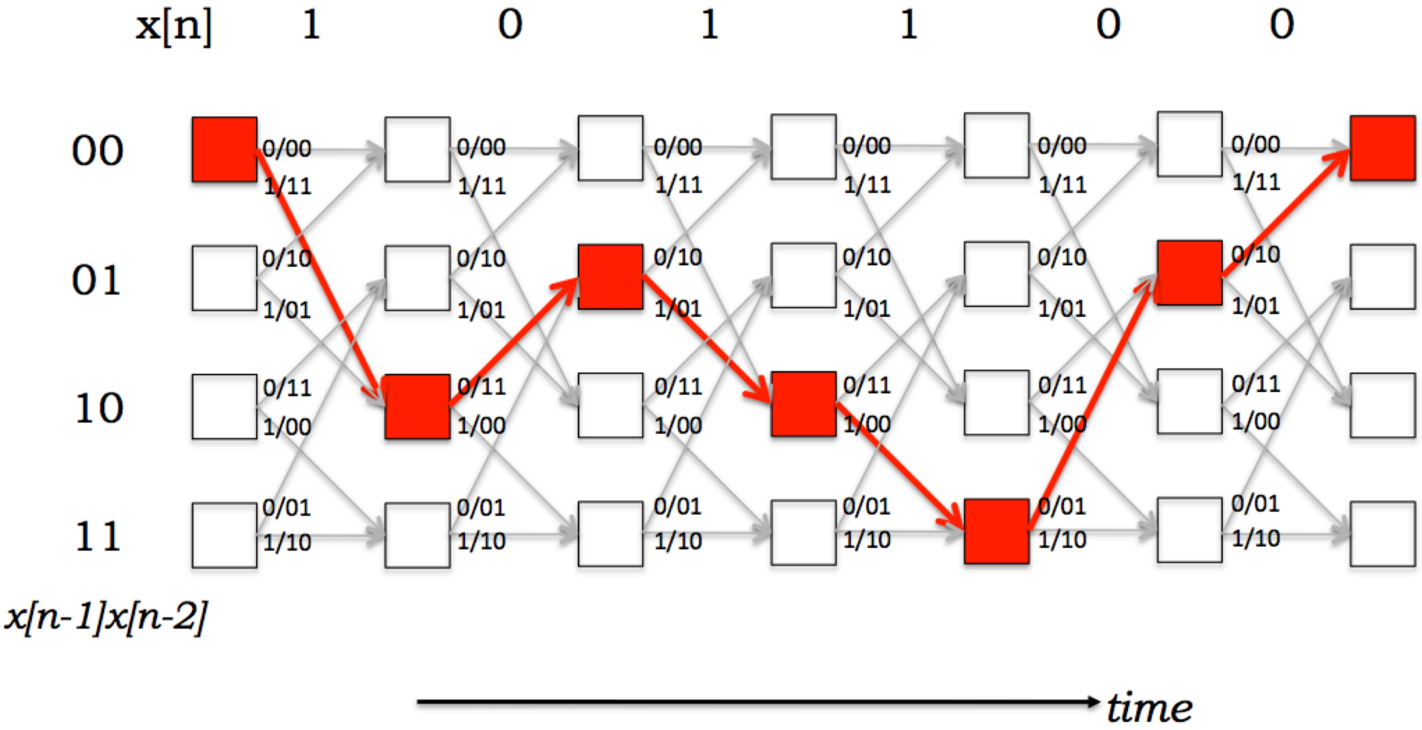
State Diagram

- Encoder changes state
- Each State (D_1D_2)
- Output (y_1y_2)
- Input (x_1)



Trellis

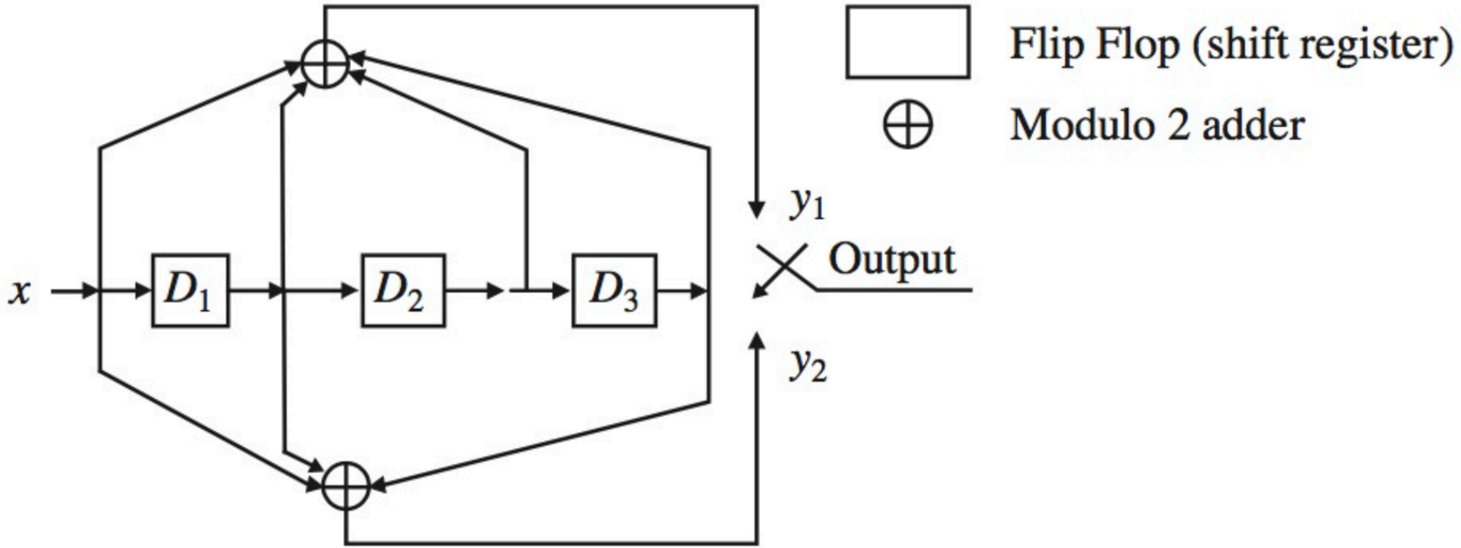
- The Viterbi decoder finds a maximum likelihood path through the Trellis



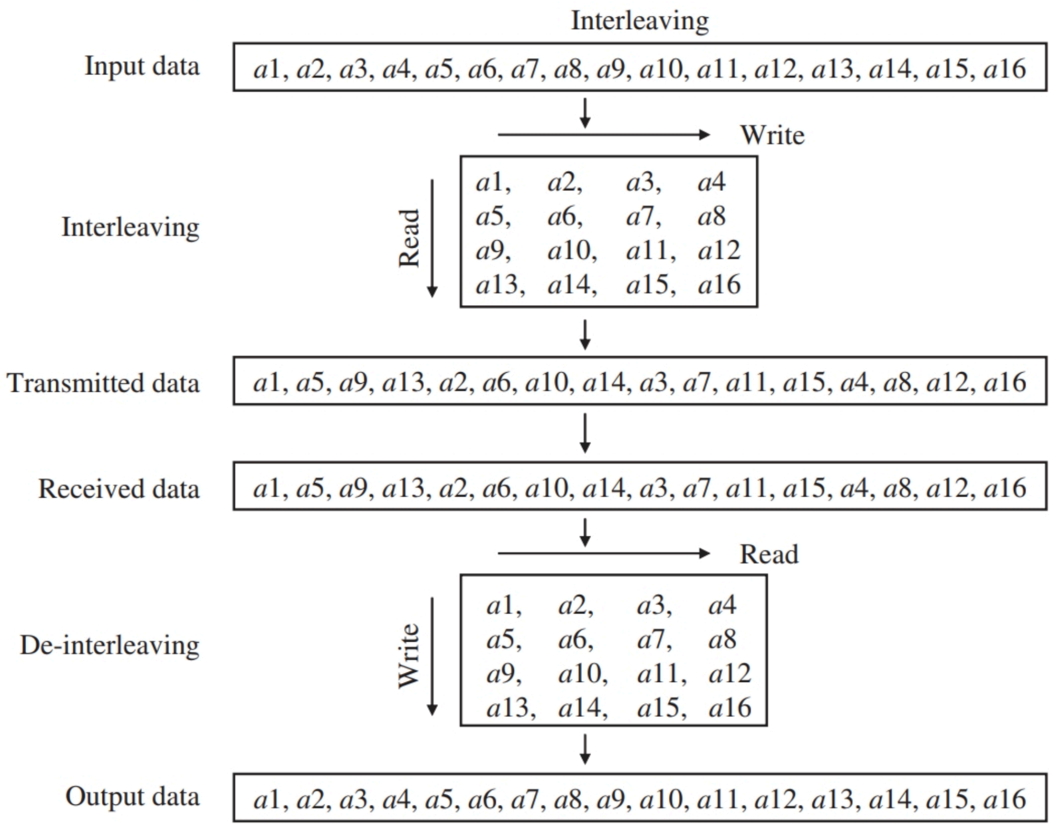
UNI
State Uni

Homework

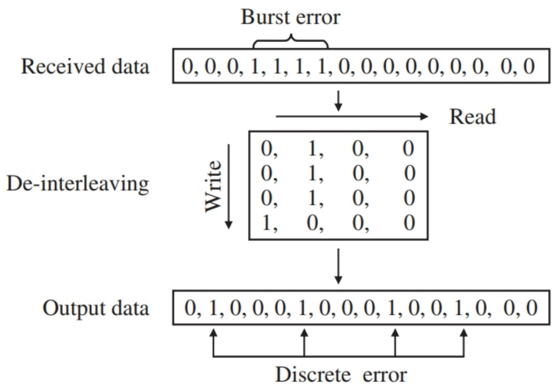
- Use the encoder to determine output produced by input sequence of 1011



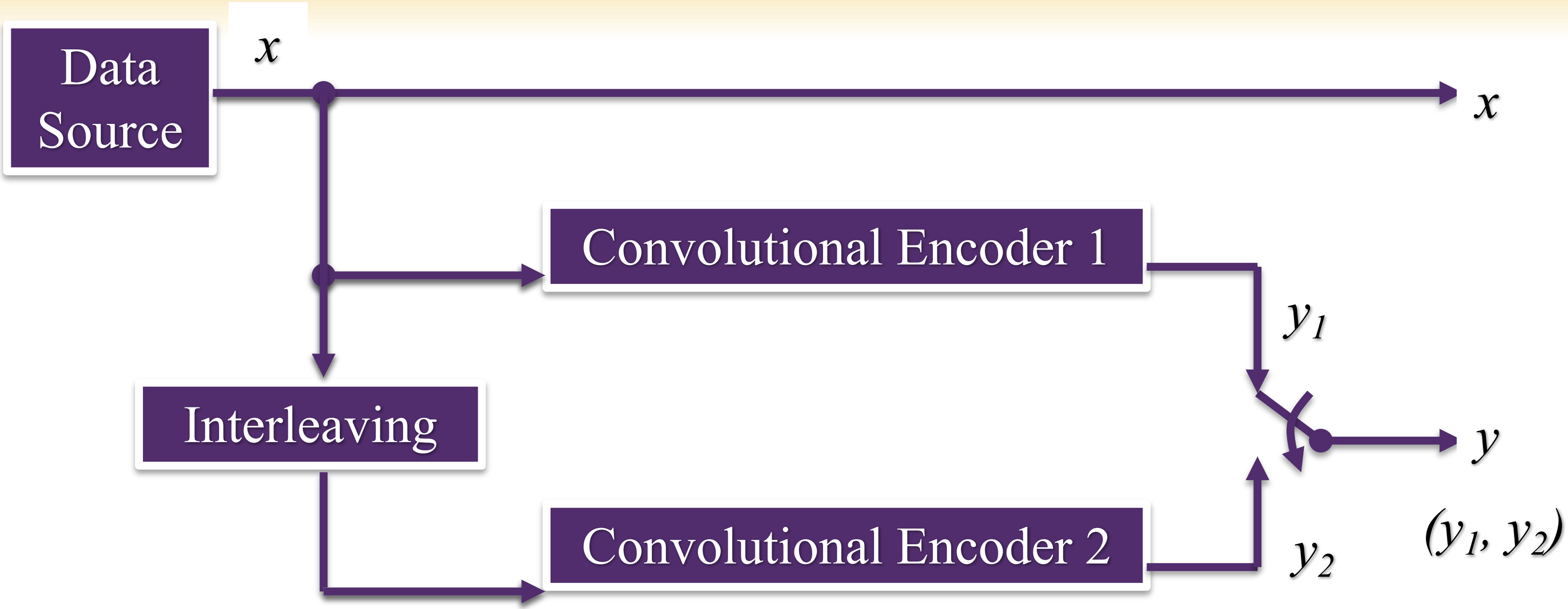
Interleaving



- Protect transmitted data from *burst errors*
  - no redundancy
  - does not have error-correcting capability
  - error-correcting codes are capable of correcting individual errors, but not a burst error



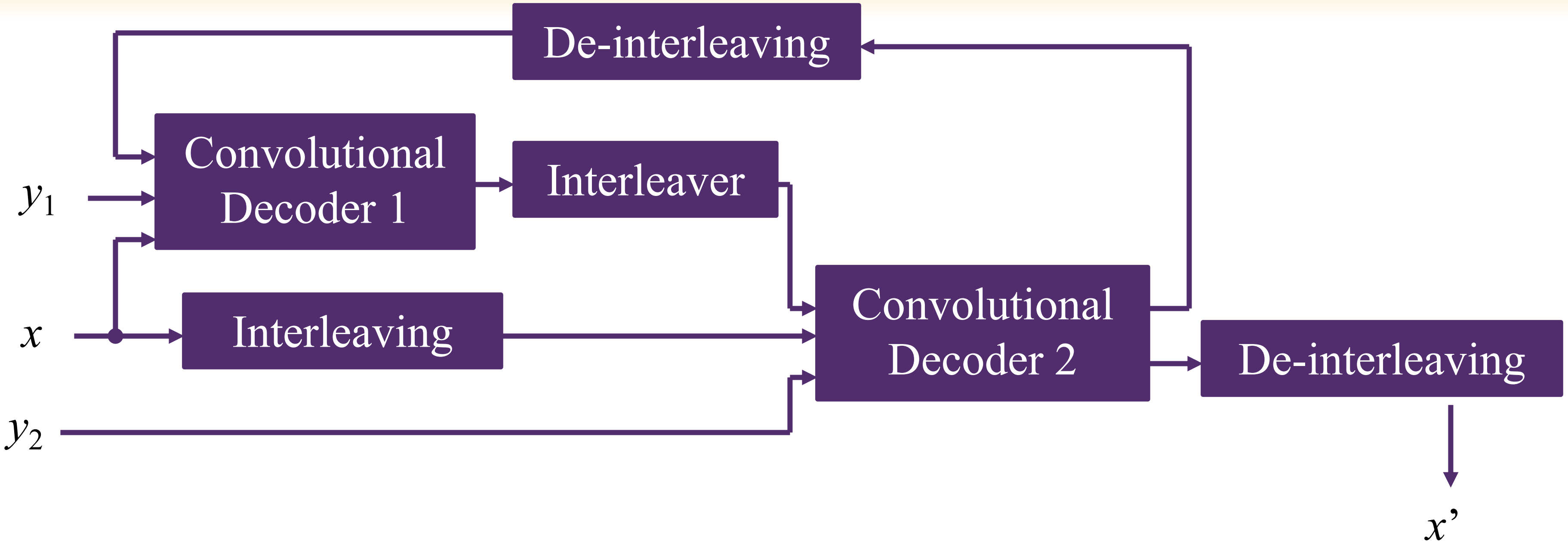
Turbo Code: Encoder



x: Information

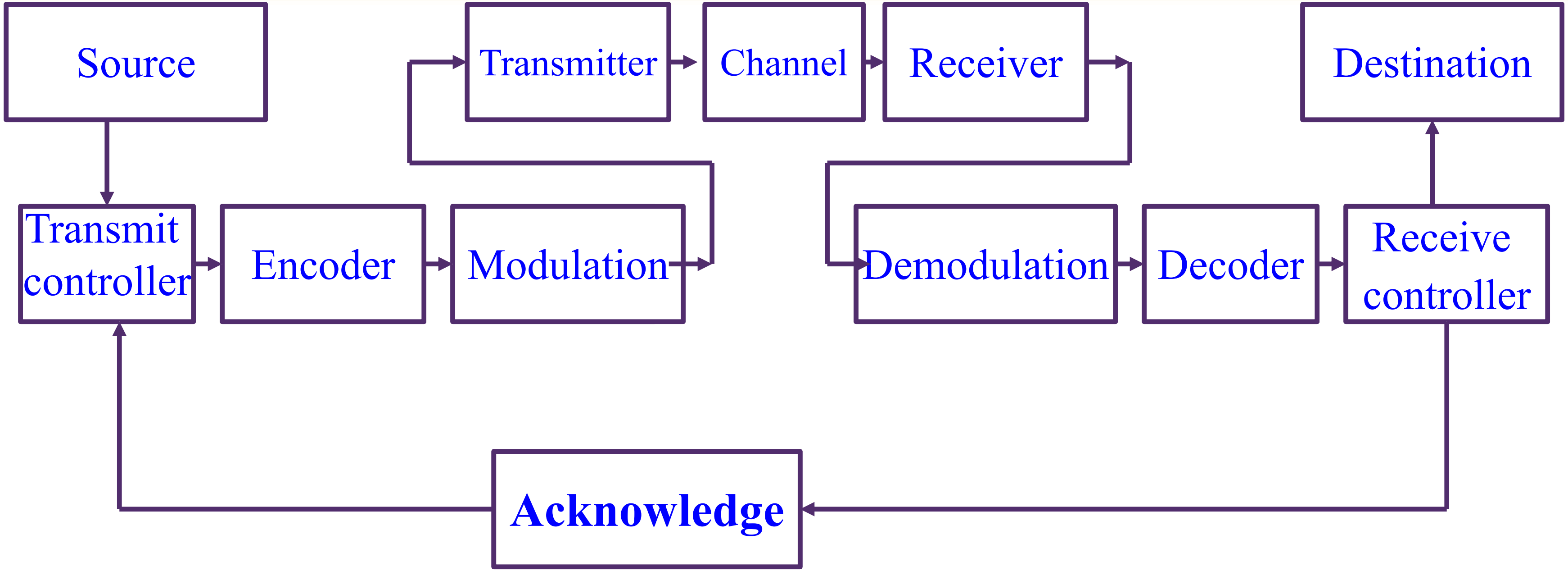
y_i: Redundancy Information

Turbo Code: Decoder

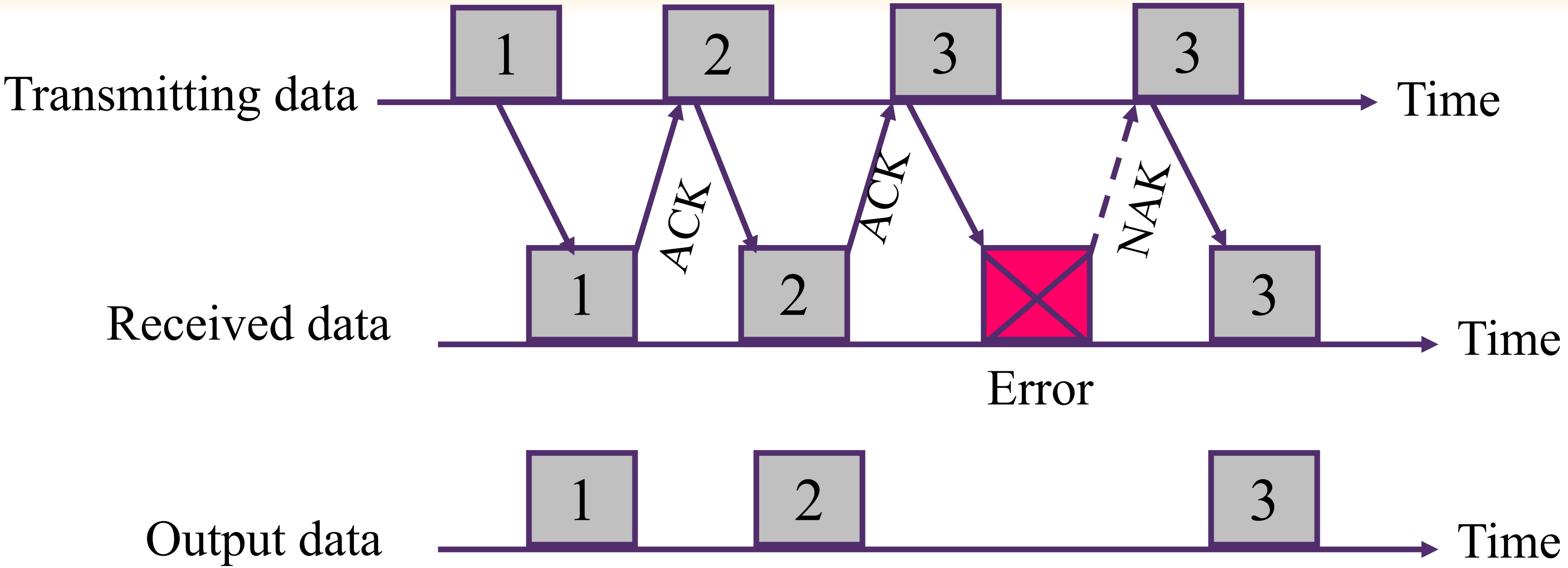


x': Decoded Information

Automatic Repeat Request (ARQ)



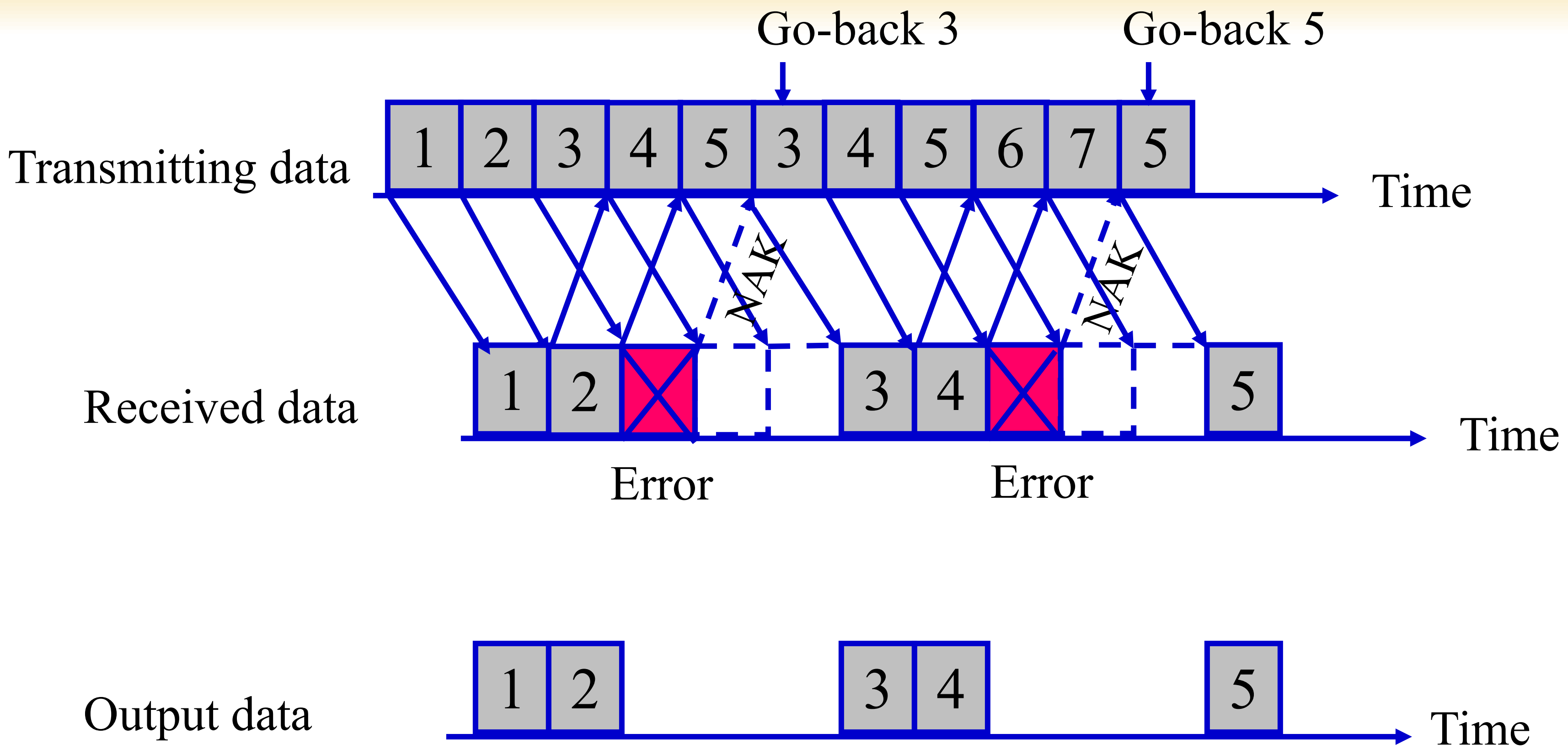
Stop and Wait



ACK: Acknowledge

NAK: Negative ACK

Go Back N



Go-back 3
Go-back 5
Transmitting data
1 2 3 4 5 3 4 5 6 7 5
Time
NAK
NAK
Received data
1 2
3 4
5
Time
Error
Error
Output data
1 2
3 4
5
Time

Selective Repeat