COMP2001J Computer Networks

Lecture 2 – Physical Layer

Dr. Shen WANG (王燊)

shen.wang@ucd.ie

北京工业大学北京－都柏林国际学院

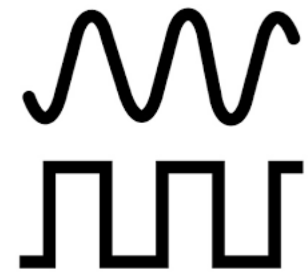
BEIJING–DUBLIN INTERNATIONAL COLLEGE AT BJUT

Announcement

- Enroll yourself on CS Moodle:
  - Key: **CN19**
- The lab session **might be** changed:
  - Clashes with TAs' timetable

  - New time **could be** 9:55-11:30am Tuesday week 3-15
  - I will put the new time and room on "Announcement-2019" on our CS Moodle when confirmed.
  - Otherwise, please still follow the time and room as of now.

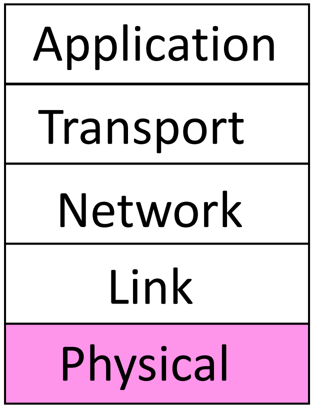
Outline



- Objective
- Media Types
- Digital Modulation
- Nyquist & Shannon Theorem
- Switching & Multiplexing
- Network Performance

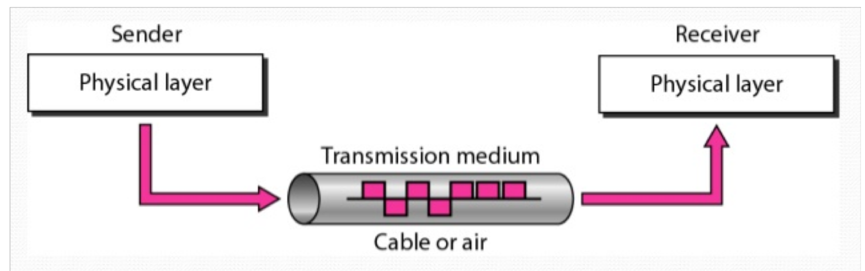
Objective

- The key problem addressed by the physical layer is to send (digital) bits using only (analog) signals
  - This process is called **modulation**
- Foundation on which other layers build
  - Properties of wires, fiber, wireless limit what the network can do



Media Types

- Wires
  - Twisted Pair
  - Coaxial Cable
  - Optical Fiber
- Wireless
  - Satellite
  - Terrestrial Microwave / Radio

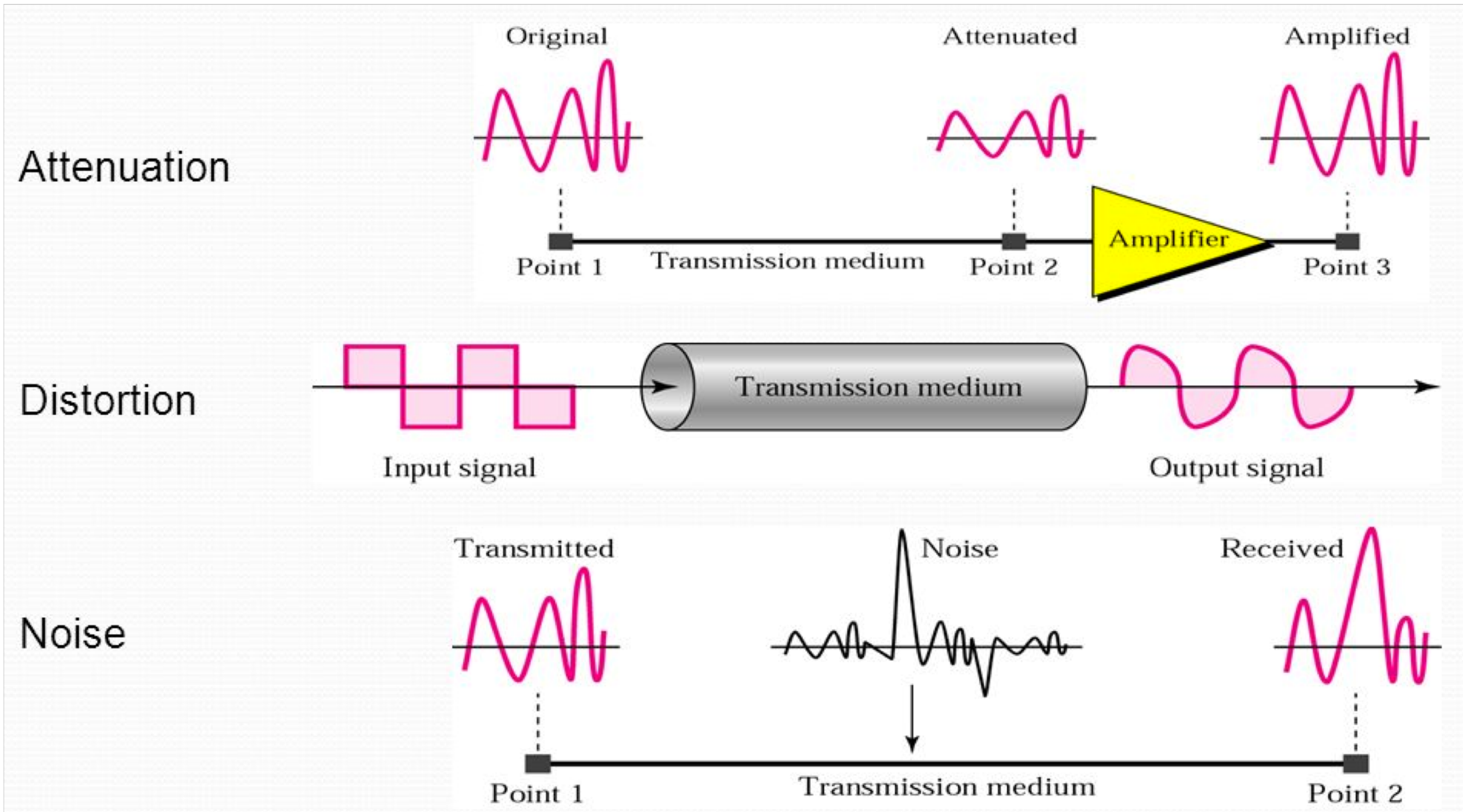


- Note: transmission medium is NOT a part of physical layer!!!

Errors and Media Types

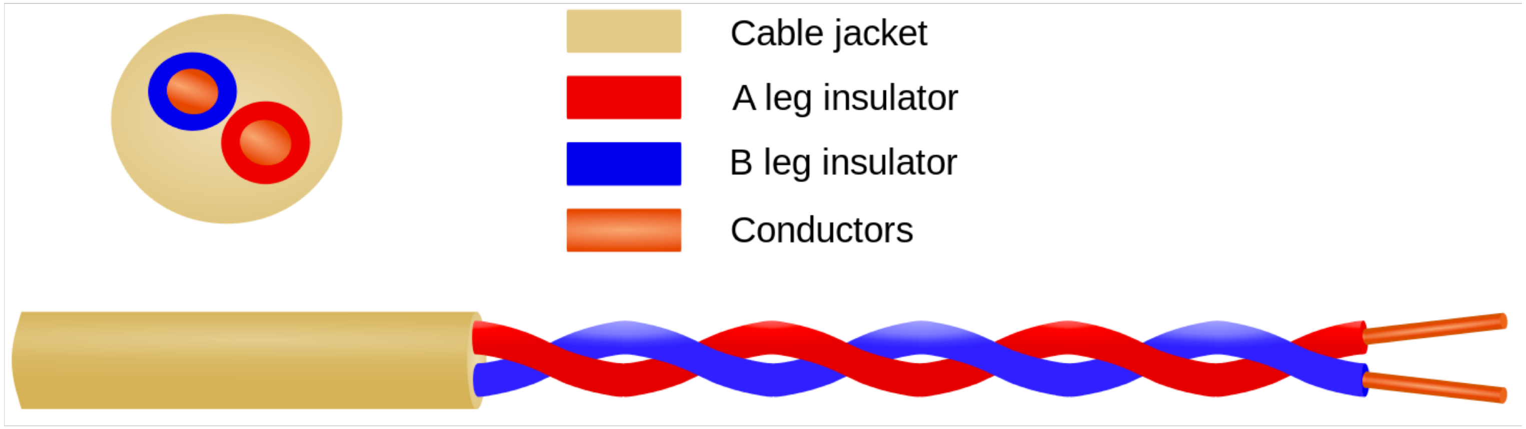
- As signals propagate along a transmission line they suffer from **attenuation** and **distortion**, they might be incorrectly interpreted by the receiver, resulting in a **bit error**
  - Attenuation is the loss of signal strength
  - Distortion is where the signal is changed
  - Bit Error Rate (BER) is the main performance metric
- The effect of attenuation and distortion varies by different *transmission media types*, and normally can be reduced by lower *transmission rate* and shorter *communication distance*.
- The choice of medium types depends on:
  - Distance to be covered
  - Desired bit rate (in bits per second, b/s)
  - Cost considerations

Comparison


Attenuation
Original
Attenuated
Amplified
Amplifier
Point 1
Transmission medium
Point 2
Point 3
Distortion
Transmission medium
Input signal
Output signal
Noise
Transmitted
Noise
Received
Point 1
Transmission medium
Point 2

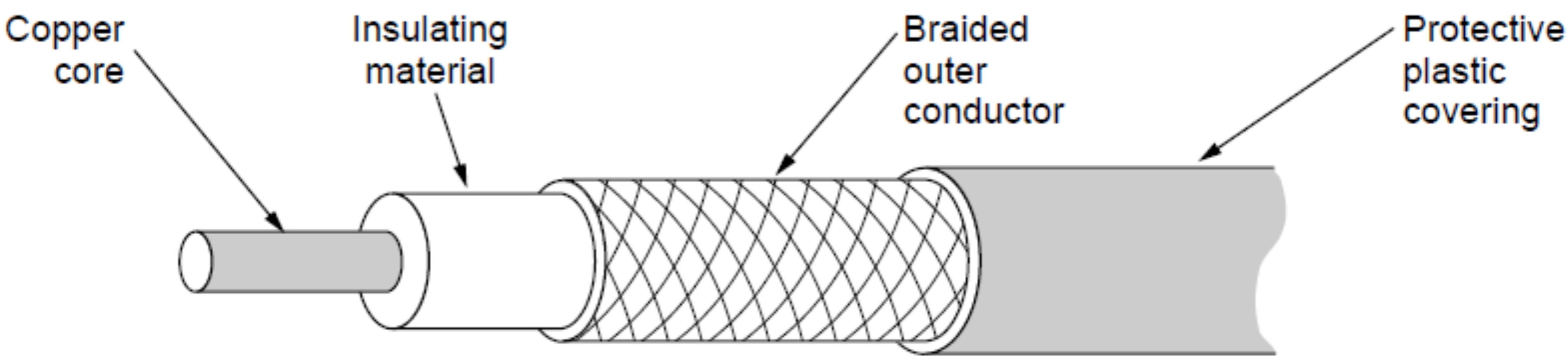
Twisted Pair

- Very common; used in LANs, telephone lines
- Signals get distorted due to:
  - Cross-talk between the two signals
  - Susceptibility to noise signals
- Twists reduce radiated signal (interference)
- It can offer higher bit rates over 10 Mb/s, over short distances (<100 m) or lower rates over longer distances.

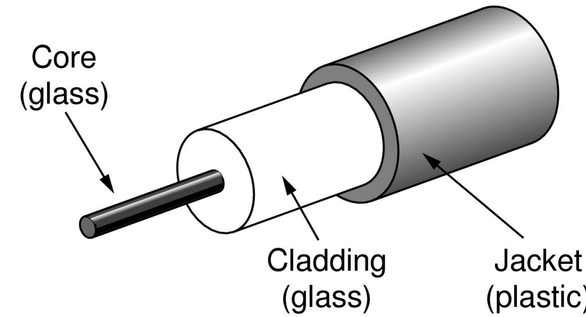


Coaxial Cable

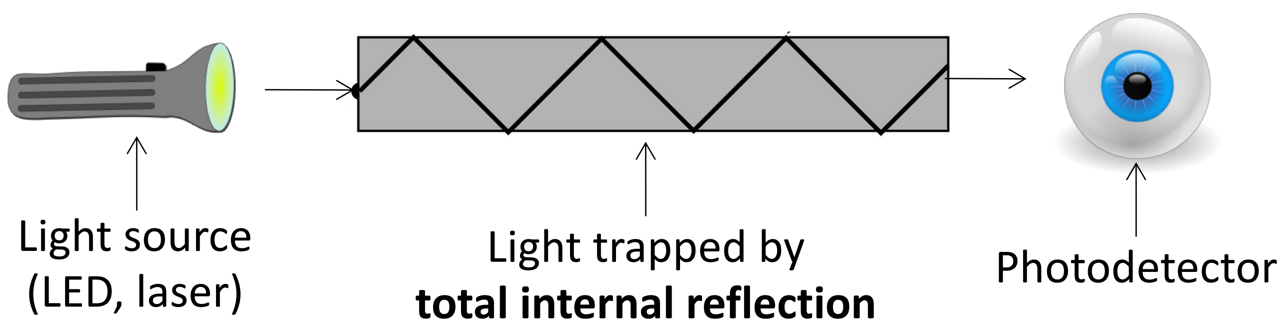
- This is made up of an inner conductor surrounded by an insulator and that surrounded again by the outer conductor
- Use of the dielectric material and outer conductor effectively isolate the core conductor from external noise interference
- Can be used at rates from over 100 Mbps, over distances of several hundred meters, up to higher bit rates of Gb/s over shorter distances ~100 meters



Optical Fiber



- It transmits light (not electrical) signals down a thin piece of glass.
  - has significantly higher bit rates, maybe from 100’s of Mb/s to many Gb/s or even Tb/s can be achieved
  - immune to electromagnetic interference and cross-talk



Satellite

- The first satellites were launched in the 1970’s and now the most common are the geo-stationary ones at 35,000 km above the earth

- Satellites are using direct line of sight with the transmitters and receivers

- Data transmitted using electromagnetic waves propagating through the atmosphere at > 4GHz

- Typically many signals will be multiplexed onto a single satellite channel utilizing a high bit rate

Terrestrial Microwave/Radio

- Again these provide direct line of sight between the transmitter and receiver where possible
- Provide communication links when it is impractical or too costly to install a physical link
- Suffer from factors such as bad weather conditions and obstruction by man-made objects
- Use radio waves (150kHz - 1GHz) for long distances and microwaves (>1GHz) for shorter distances but limited by the curvature of the earth

Digital Modulation

- Baseband
  - NRZ(Non-Return-to-Zero) / NRZI
  - Manchester Encoding
  - 4B/5B
- Passband
  - ASK
  - FSK
  - PSK

Digital Modulation

- Wires and wireless channels carry analog signals such as continuously varying voltage, light intensity, or sound intensity

- To send digital information, we must devise analog signals to represent a bit

- The process of converting between bits and signals that represent them is called **digital modulation**

Digital Modulation

- There are a number of different schemes for converting bits into a signal
- Schemes that directly convert bits into a signal result in what is called **baseband transmission**
  - In this scheme the signal uses the entire bandwidth of the signal **from 0 HZ**
- Scheme that regulate the amplitude, phase, or frequency of a carrier signal to convert bits into a signal result in what is called **passband transmission**
  - In these schemes the signal occupies a band of frequencies around the frequency of the carrier signal

About “bandwidth”

- To electrical engineers:
  - The width of the frequency range transmitted without being strongly attenuated is called the (analog) **bandwidth** (Hz).
  - It is a physical property of the transmission medium that depends on, for example, the construction, thickness, and length of a wire or fiber.
  - It is the difference between the upper and lower frequencies in a continuous set of frequencies.
  - Any band of a given width can carry the same amount of information, **regardless of where that band is located** in the frequency spectrum
    - It doesn’t matter if we use the frequencies 0 – 3 Khz or 12 – 15 Khz

About “bandwidth”

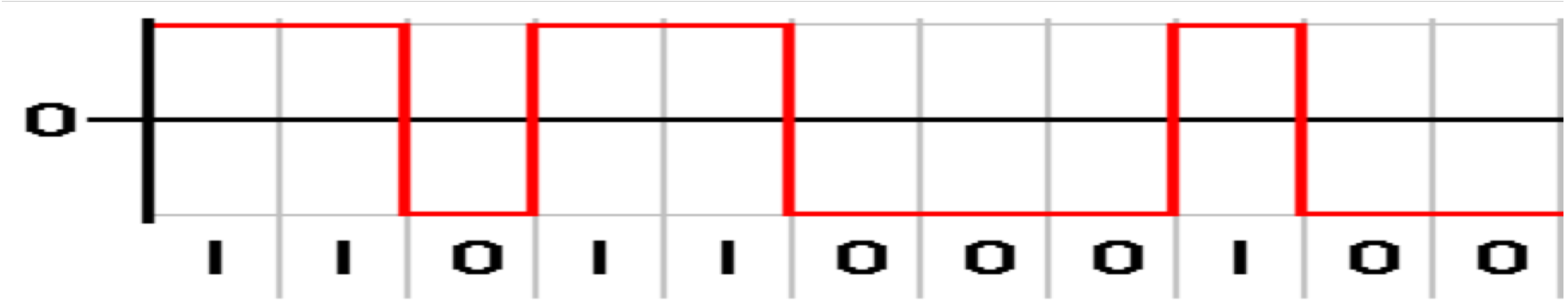
- To computer scientists:
  - (digital) **bandwidth** is the maximum data rate of a channel, a quantity measured in bits/sec.
  - That data rate is the end result of using the analog bandwidth of a physical channel for digital transmission, and the two are related.

Baseband Transmission

- The simplest digital modulation is to use a positive voltage to represent a 1 and a negative voltage to represent a 0
- For an optical fiber, the presence of light might represent a 1 and the absence of light might represent a 0
- This scheme is called **NRZ (Non-Return-to-Zero)**.

Non Return to Zero

- Once sent, the NRZ signal travels down the wire
- At the other end, the receiver converts it into bits by sampling the signal at regular intervals of time



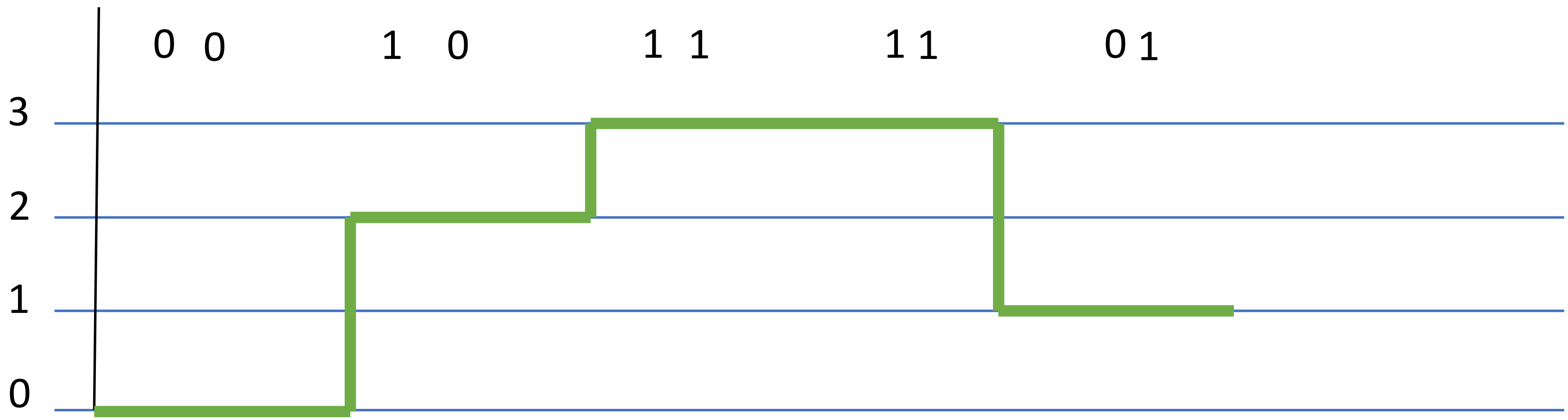
Non Return to Zero

- This signal will not look exactly like the signal that was sent
- It will be attenuated and distorted by the channel and noise at the receiver
- To decode the bits, the receiver maps the signal samples to the closest symbols
- For NRZ, a positive voltage will be taken to indicate that a 1 was sent and a negative voltage will be taken to indicate that a 0 was sent.

Other Schemes

- It is possible to send more than one bit at a time

- This requires more **signal levels**
  - To send 2 bits at a time requires 4 levels



NRZ Limitation

- For schemes that encode bits into symbols, the receiver must know when one symbol ends and the next symbol begins to correctly decode the bits.
- With NRZ, in which the symbols are simply voltage levels, *a long run of 0s or 1s* leaves the signal unchanged.
- After a while it is hard to tell the bits apart. For example, 15 zeros look much like 16 zeros unless you have a very accurate clock.

Clock Recovery

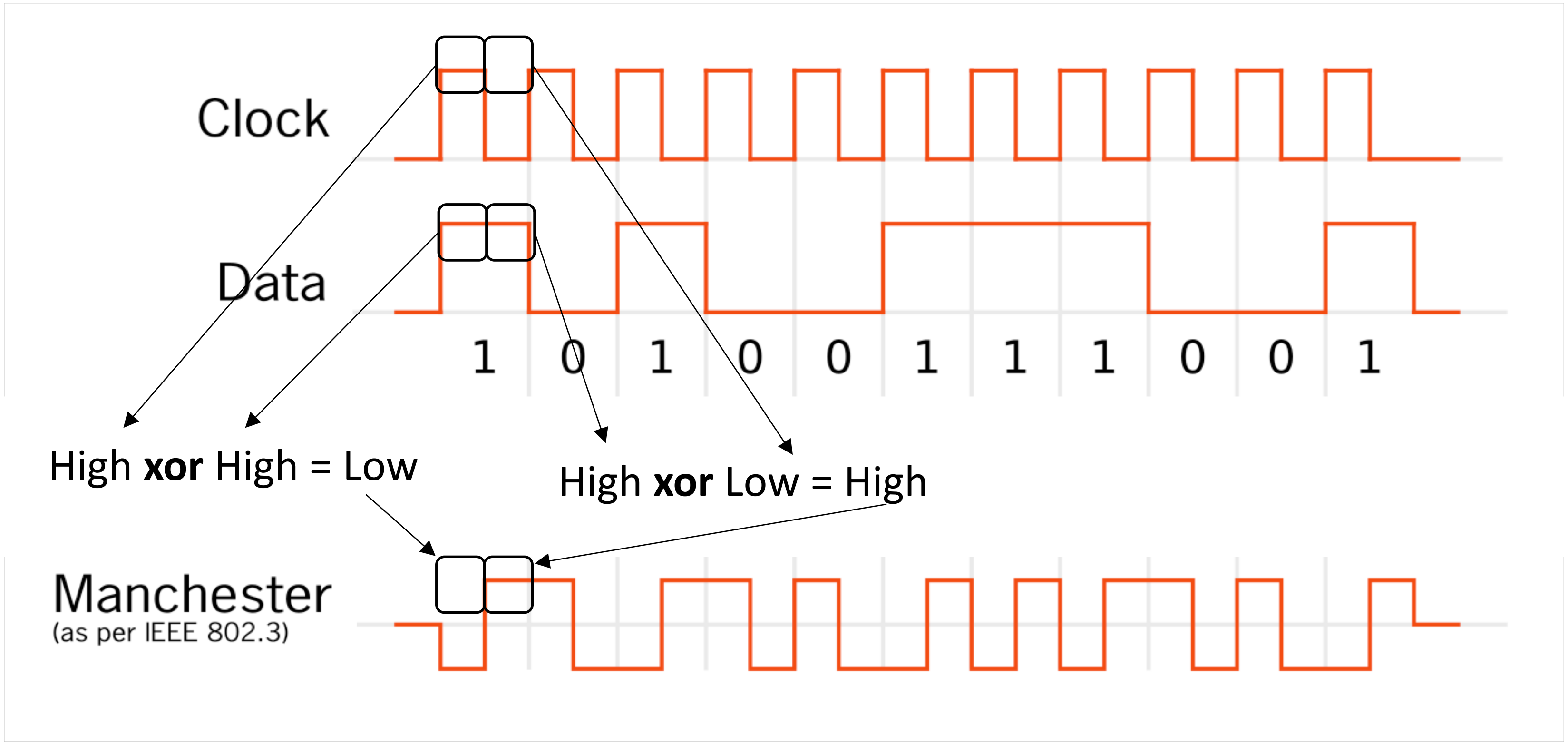
- Accurate clocks would help with this problem, but they are an **expensive** solution for equipment that will be very common
- A possible solution is to send a **separate clock signal**
  - The receiver then only has to keep time with the clock signal
- The clock signal changes the level of the signal at regular intervals
  - These changes are called **transitions**
- This however requires an *extra cable* which could be used to carry data

Manchester Encoding

- A clever trick here is to mix the clock signal with the data signal by **XORing** them together so that no extra line is needed.
- The clock makes a clock transition in every bit time, so it runs at **twice** the bit rate
- This scheme is called **Manchester encoding** (used in **Ethernet**, a family of computer networking technologies commonly used in local area networks (LAN), IEEE 802.3)

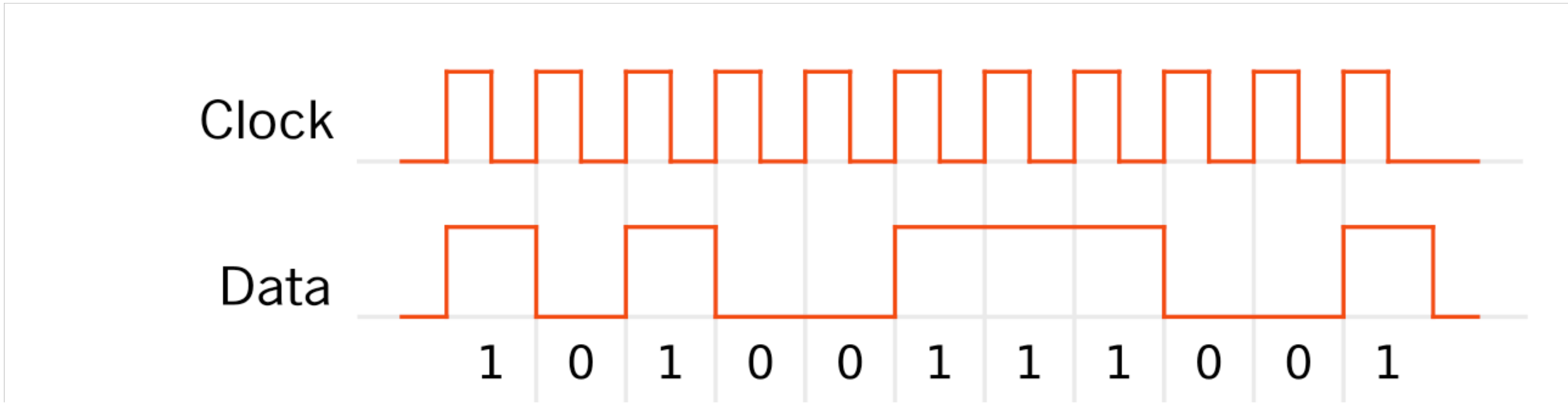
Manchester Encoding

- The downside of Manchester encoding is that it requires **twice** as much bandwidth as NRZ because

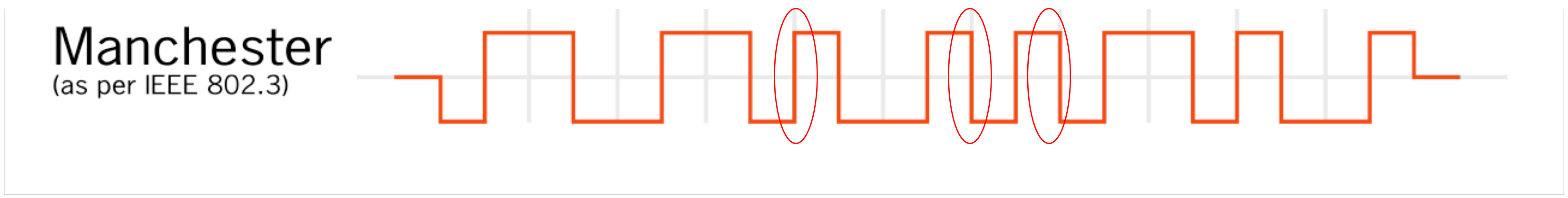


Manchester Encoding

- The downside of Manchester encoding is that it requires twice as much bandwidth as NRZ because



These transitions do not signify data and are considered overhead

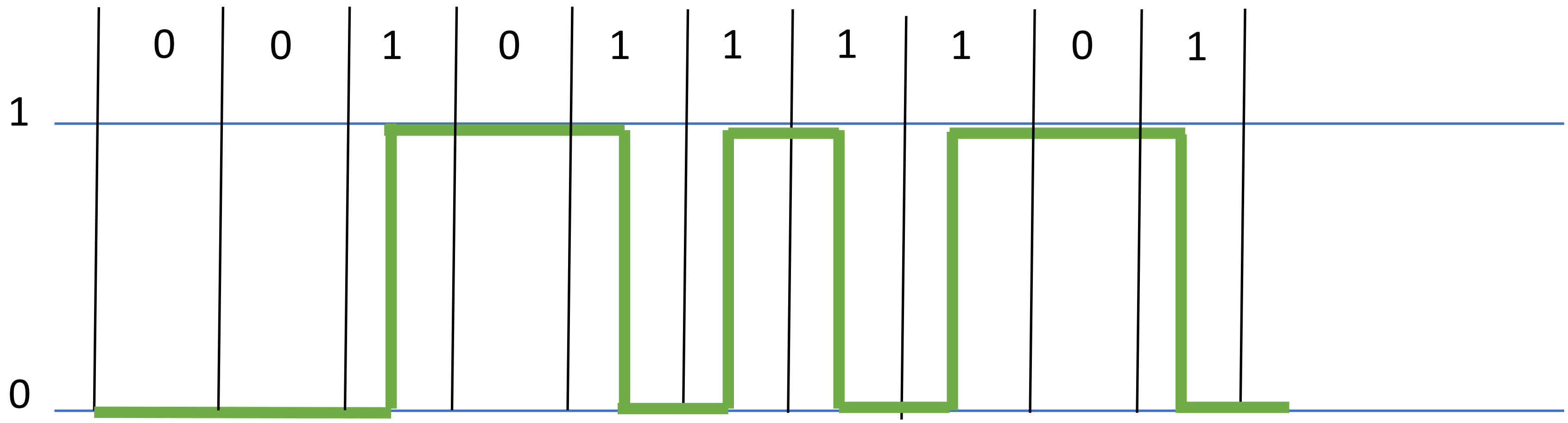


Clock Recovery

- A different strategy is based on the idea that we should code the data to ensure that there are enough transitions in the signal
- Consider that NRZ will have clock recovery problems only for long runs of 0s and 1s
- If there are frequent transitions, it will be easy for the receiver to stay synchronized with the incoming stream of symbols
- As a step in the right direction, we can simplify the situation by coding a 1 as a transition and a 0 as no transition, or vice versa

Non Return to Zero Invert

- Non return to zero invert (NRZI) is a special case of NRZ where changes to the level of the signal called **transitions** are used to represent a 1 and no change to represent a 0



Non Return to Zero Invert

- This encoding scheme removes problems with long sequences of 1s
- But long sequences of 0s still cause a problem
- To really fix the problem we can break up runs of 0s by mapping small groups of bits to be transmitted so that groups with successive 0s are mapped to slightly longer patterns that do not have too many consecutive 0s

4B/5B

- A well known code to breaking up sequences of 0s is called 4B/5B
- Every possible combination of 4 bits is mapped to a sequence of 5 bits
- The 5 bit sequences are chosen so there will never be more than 3 consecutive 0s
- At the receiver the codes are converted back to their original values

4B/5B Mappings

Data 4B	Codeword 5B	Data 4B	Codeword 5B
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

4B/5B

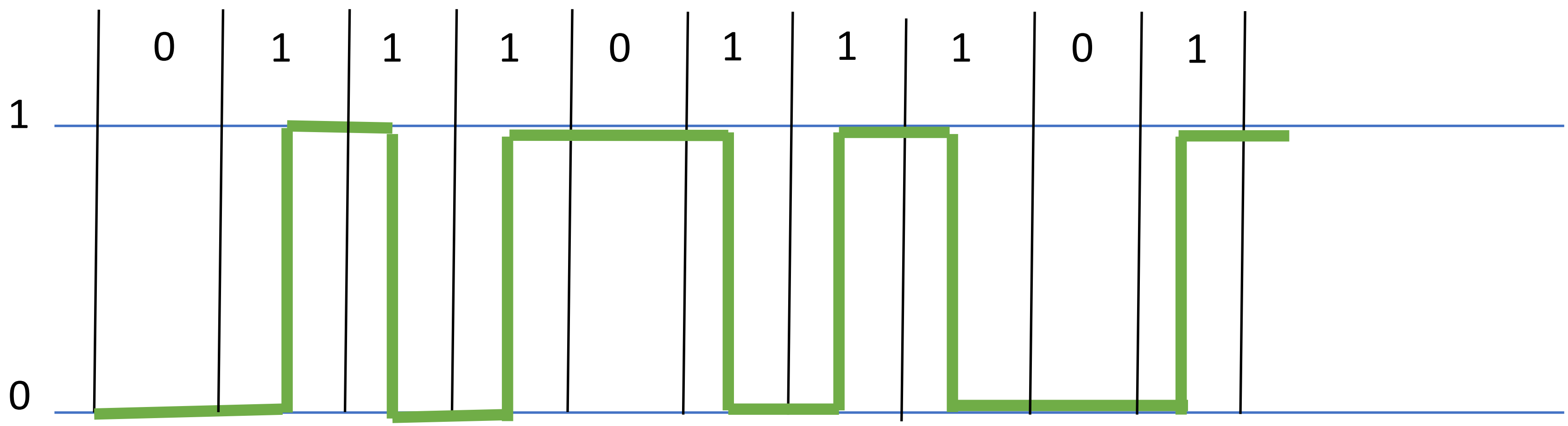
- Since there are 16 input combinations and 32 output combinations, some of the output combinations are not used

- Some are not used because there are **too many (3 or more) successive 0s**

- Others non-data codes are used to represent physical layer control signals

4B/5B

- If we wish to send the number 0110 1111
- First we convert it to codewords
  - 01110 11101



Passband Transmission

- Often, we want to use a range of frequencies that **does not start at zero** to send information across a channel
- For wireless channels, it is not practical to send very low frequency signals because the size of the antenna needs to be a fraction of the signal wavelength, which becomes large
- Even for wires, placing a signal in a given frequency band is useful to let different kinds of signals coexist on the channel
- This kind of transmission is called **passband transmission**

Passband Transmission

- Digital modulation is accomplished with passband transmission by regulating or modulating a carrier signal that sits in the passband
- We can modulate the amplitude, frequency, or phase of the carrier signal
- Each of these methods has a corresponding name
  - In **ASK (Amplitude Shift Keying)**, two different amplitudes are used to represent 0 and 1
  - With **FSK (Frequency Shift Keying)**, two or more different tones are used
  - In the simplest form of **PSK (Phase Shift Keying)**, the carrier wave is systematically shifted 0 or 180 degrees at each symbol period

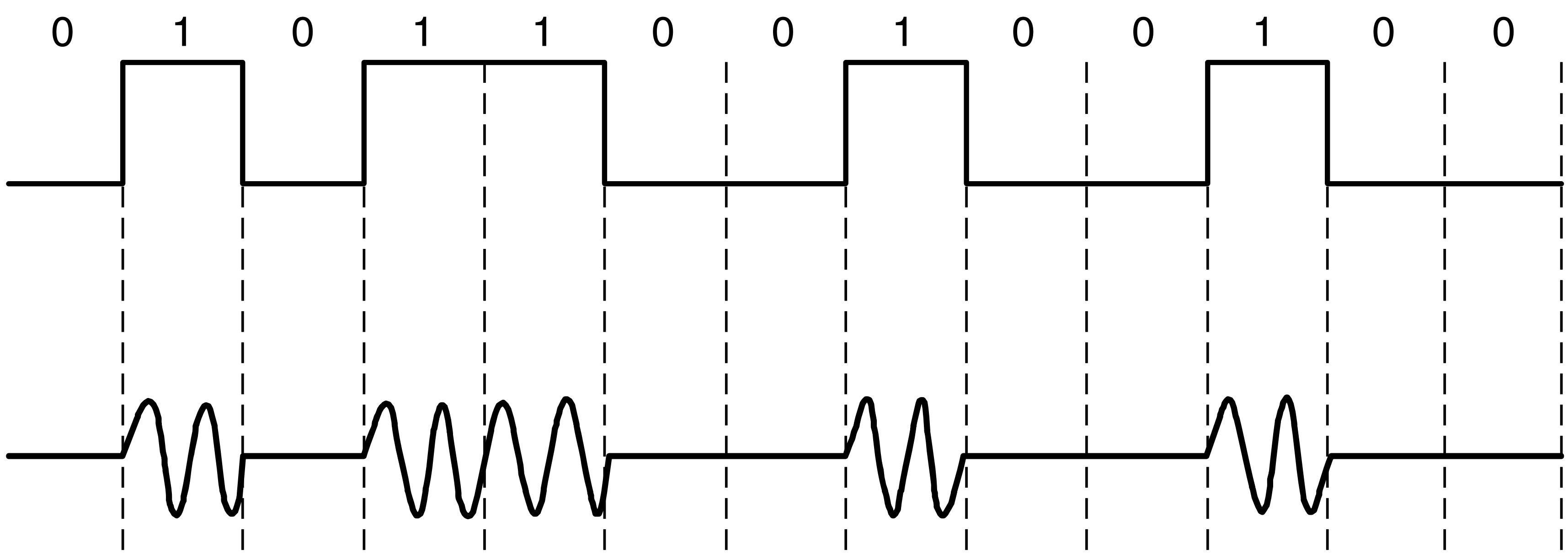
Carrier Signals

- A carrier is simply a signal oscillating at a desired frequency

- We can modulate it by changing the **amplitude** (size of the signal), **frequency** (speed of oscillation) or **phase** (position of the cycle)

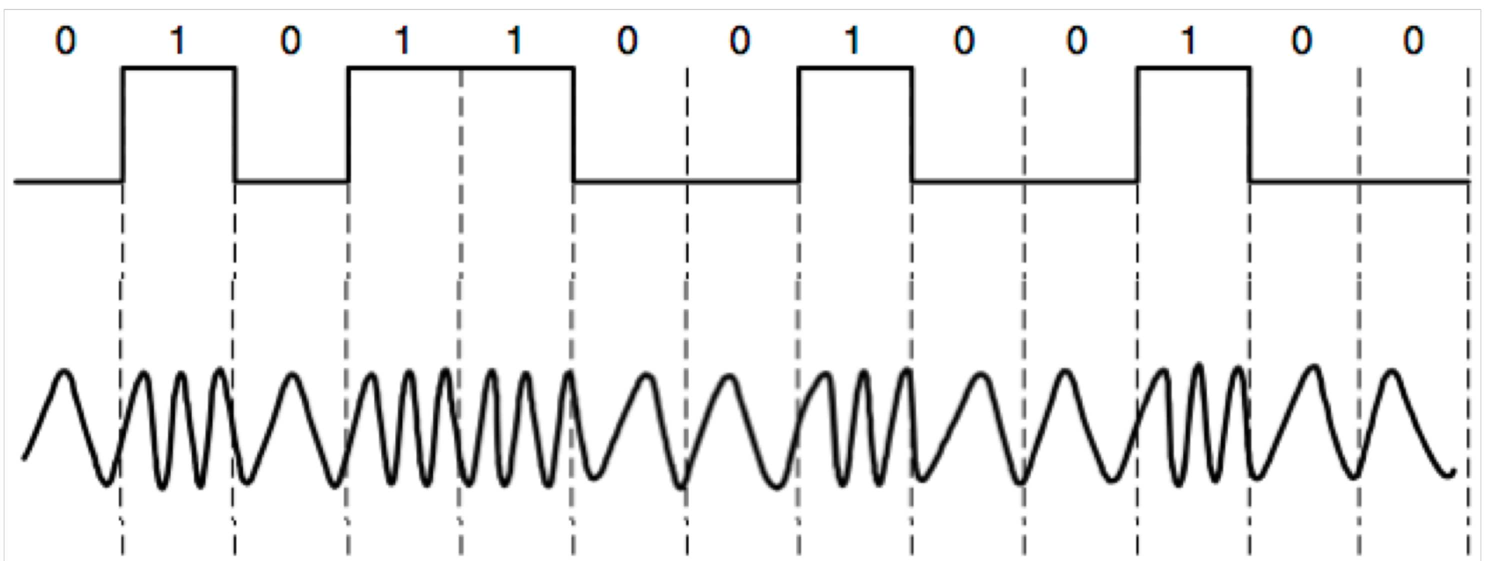
Amplitude Shift Keying

- In **ASK** two different amplitudes are used to represent 0 and 1
- In this example one of the amplitudes used is 0



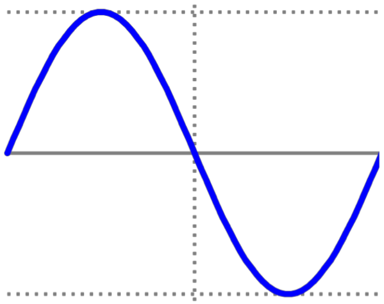
Frequency Shift Keying

- In **FSK** two different frequencies are used to represent 0 and 1
- In this example a lower frequency is used for 0 and a higher frequency for 1

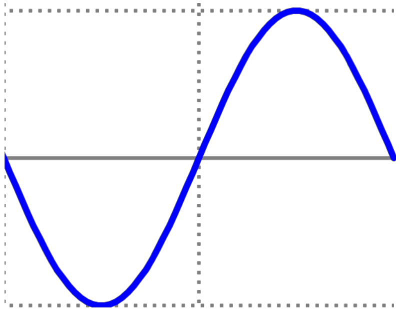


Phase

- Phase in a carrier signal refers to the position of the cycle within the time period
- If we think of the original signal as

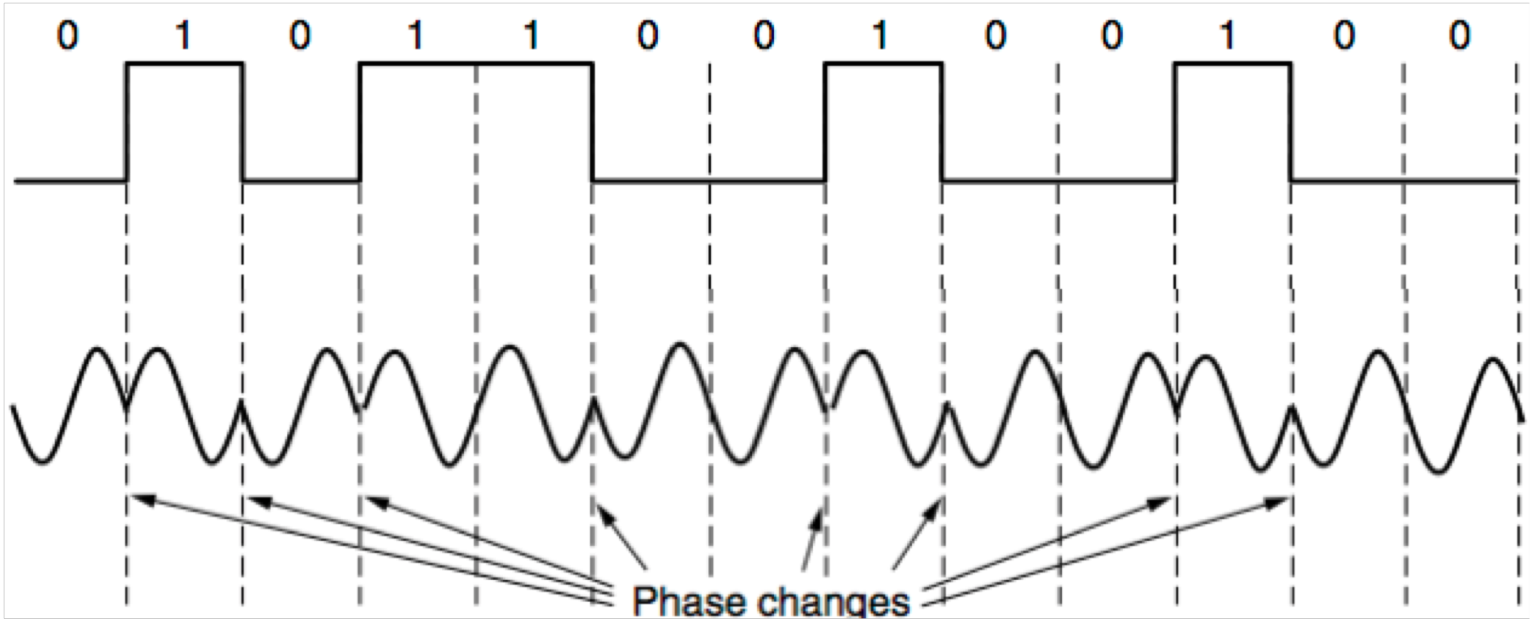


- Then one with a the phase shifted by 180 degrees would be



Phase Shift Keying

- In the simplest form of **PSK (Phase Shift Keying)**, the carrier wave is systematically shifted 0 or 180 degrees at each symbol period



Passband Transmission

- We can combine these schemes and use more levels to transmit more bits per symbol
- **Only one of frequency and phase** can be modulated at a time because they are related, with frequency being the rate of change of phase over time
- Usually, **amplitude and phase** are modulated in combination

Fundamental Limits

- How fast can we send information over a link?

- There are two limits which have been calculated that give us an idea
  - The Nyquist limit
    - Discovered in 1924
  - The Shannon Capacity
    - Discovered in 1948

- Both of these limits determine how much information it is possible to send over a link

Key Channel Properties

- There are three key properties of communication channels that are represented in both calculations

- These are
  - Bandwidth (B)
  - Signal Strength (S)
  - Noise (N)

Noise

- On any line, even in the absence of a data signal, random perturbations of the line voltage and current will occur

- This effect is known as line noise level, or simply **background noise**.

- There are three main causes of this noise:
    - Cross-talk
    - Impulse noise
    - Thermal noise

Cross-talk & Impulse Noise

- Cross-talk - Occurs when a signal on one line is picked up by adjacent lines as a small noise signal
- Particularly troublesome is Near-End Cross-talk (NEXT) caused when a strong transmitter output signal interferes with a much weaker incoming receiver signal.
- Impulse Noise - Caused by external activity or equipment. Generally takes the form of electrical impulses on the line which cause large signal distortion for their duration

Thermal Noise

- Thermal noise is always present because it is caused by the thermal agitation of electrons associated with each atom in the device or transmission line material.
- It consists of random frequency components of continuously varying amplitude
- For this reason it is also known as White Noise
- One of the main aims of transmission is to contain the effects of the different type of noise

Signal Strength and Noise

- Signal strength is the strength of the signal at the receiver
- Noise is also the amount of noise in a signal as measured at the receiver
- Signal strength and noise determine how many signal levels we can distinguish

Nyquist Limit

- The Nyquist limit determines for a communication link **without noise** the maximum *symbol rate*
- A symbol is a wave that can represent a number of bits
- If we have a bandwidth of **B** the maximum symbol rate is **2B** (Baud rate)
- If we have **V** signal levels (see the slide on page 20) the maximum bit rate R is

$$R = 2B\log_2 V \quad bits/\sec ond$$

Shannon Capacity

- Shannon capacity tells us the maximum information carrying rate of a channel.

- It is calculated based on the noise and signal strength
  - We calculate what is called the signal-to-noise ratio (SNR)

- SNR is given on a log-scale in deciBels:
  - $SNR_{dB} = 10\log_{10}(S/N)$
  - E.g.: if $S/N = 100$, $SNR_{dB}=20$

Shannon Capacity

- The Shannon limit for capacity (C)
  - This is the maximum information carrying rate of the channel
- $C = B \log_2(1 + S/N)$ bits/second
- Information cannot be transferred any faster than this rate
- If the results are different using Shannon and Nyquist, then choose the **minimum** one between them as the maximum bit rate in practice.

Sharing

- To save cost, network resources must be ***shared*** between the users, while still allowing senders to transmit data to their receivers
- The two basic techniques that permit connectivity while sharing resources are **switching** and **multiplexing**
- Switching means sharing ***network resources*** among multiple transmissions
- Multiplexing means sharing a ***single link*** among multiple transmissions

Switching

- There are 2 types of common switching techniques!

- Circuit Switching
  - Making a full connection, through a number of shared resources, between the sender and receiver.
  - Maintaining the connection (using the resource) while the transfer is taking place.
  - Releasing the connection when the transfer is finished.
  - Data flows like a stream through the connection!

- Packet Switching
  - Data is broken into pieces.
  - Pieces of data are sent into the network without making a connection.
  - The pieces of data are inspected at each resource (taking time) and forwarded to the next location on the way to the receiver.

Circuit Switching

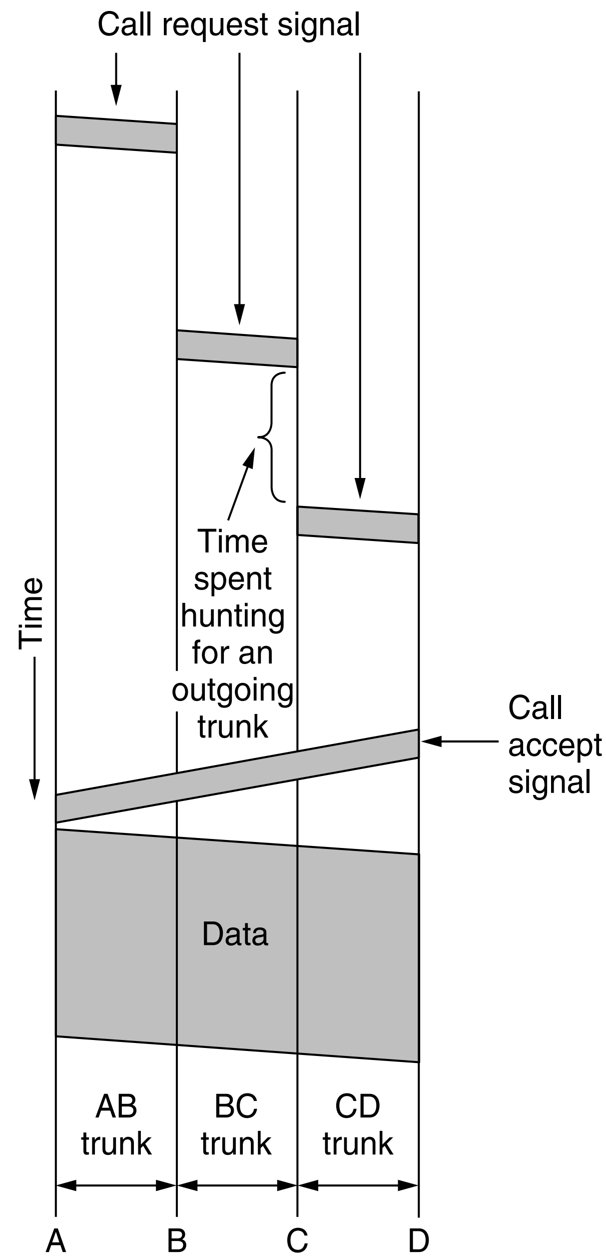
- A path through the switches is made.
- the necessary network resources are **reserved** for the connection prior to any data transfer; if this is not possible, the connection request is **blocked**
  - While the resources are blocked no one else may use them
- These reserved resources are then **held** for the duration of the connection, regardless of actual usage
  - Even when the connection is idle another computer cannot transmit
- The connection set-up delay can be significant (>1 second)

Circuit Switching

A = Sender

B,C = Intermediary nodes

D = Receiver



Circuit Switching

- Circuit switching is ideal for “smooth” network traffic
  - e.g. telephone network
- what if the traffic from sender to receiver is “bursty”?
  - Bursty traffic is when the connection is only used for short periods of time and idle for the rest
- Computer-to-computer traffic can be very bursty. We have two options, we
  - could set up a **new circuit** for each burst
  - could **hold** original circuit for duration of data transfer
- both of these solutions are **wasteful** of network resources

Packet Switching

- A packet is a string of bits
  - Usually up to 10000

- Packet switching is based on the **store-and-forward** operation

- Each router within the network receives packets stores them and forwards them onto the next destination

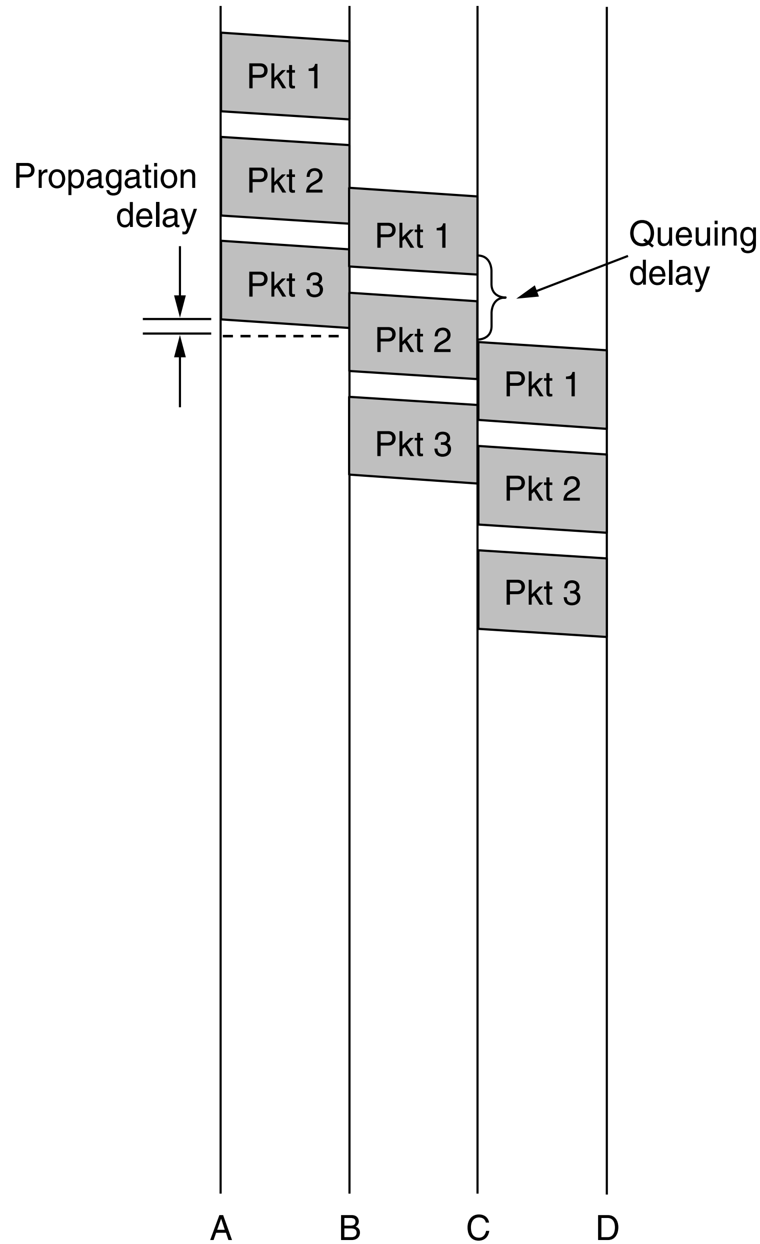
- There is **no need** to reserve bandwidth along the entire route of the data

Packet Switching

A = Sender

B,C = Intermediary nodes

D = Receiver



Packet Switching

- The size of packets within networks is strictly limited
- This way no user can monopolize any transmission line for very long
  - Each packet should only take a small number of milliseconds to transmit
- It also **reduces delay** because the first packet of a long message can be forwarded before the second one has fully arrived
- There are 2 basic types of packet switching: **datagram** and **virtual circuit**

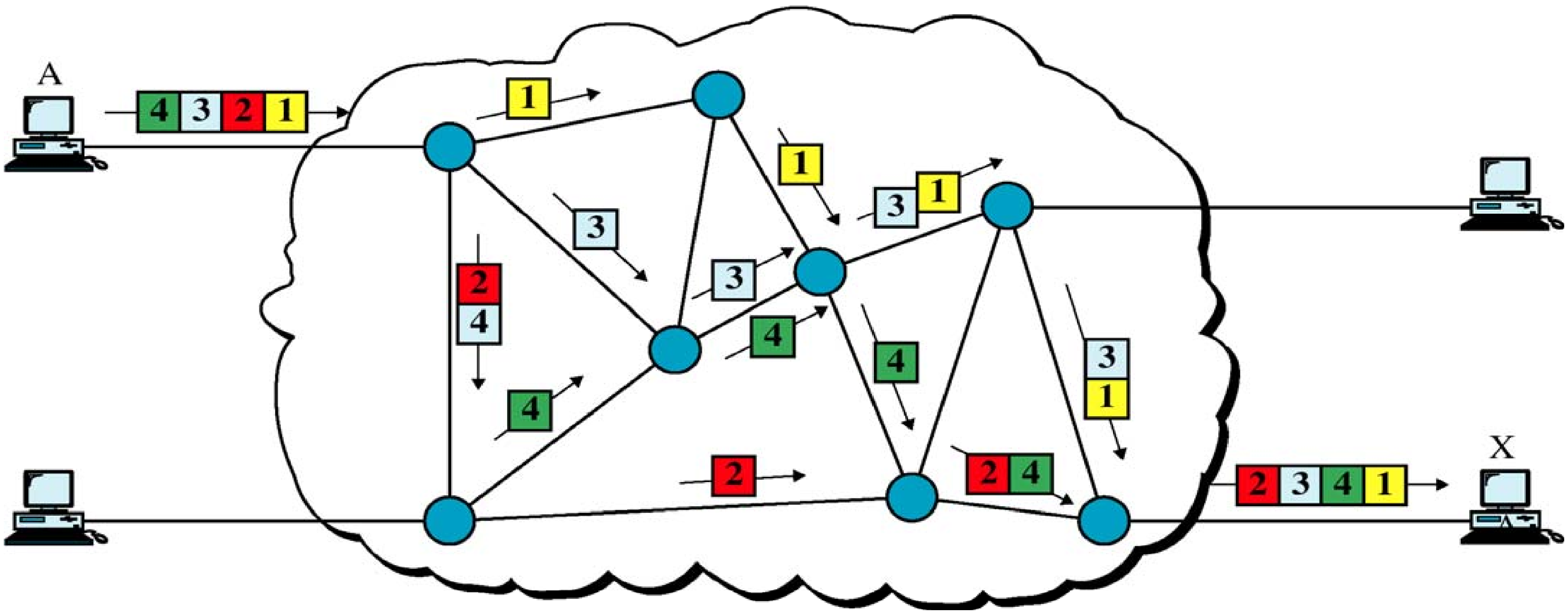
Datagram Packet Switching

- Each packet is treated **individually** within the network, so successive packets may follow **different routes** through the network

- Each packet contains the receiver's **address** and a **sequence number** (so that receiver can put them into correct order)

- Network nodes are **routers**, which have routing tables telling them which output link to use for each possible destination

Datagram Packet Switching



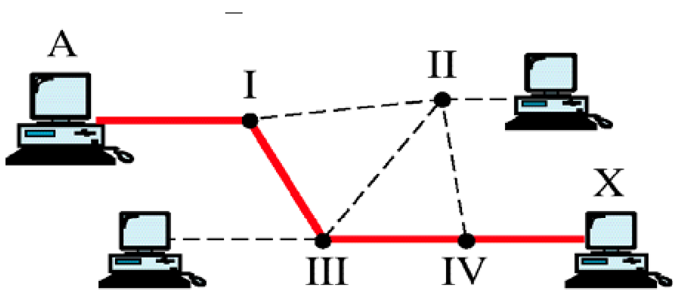
Datagram Packet Switching

- There is no connection set-up needed
- **Flexible routing** is possible if a router crashes
- Network resources are *not shared at the same time*
  - Each packet monopolizes a link during its transmission, after which the link is available for other packet transmissions
- Datagram packet switching is ideal for *short-lived* bursty traffic
- Datagram packet switching is less suitable for *long-lived* &/or *interactive* bursty traffic

Virtual Circuit Packet Switching

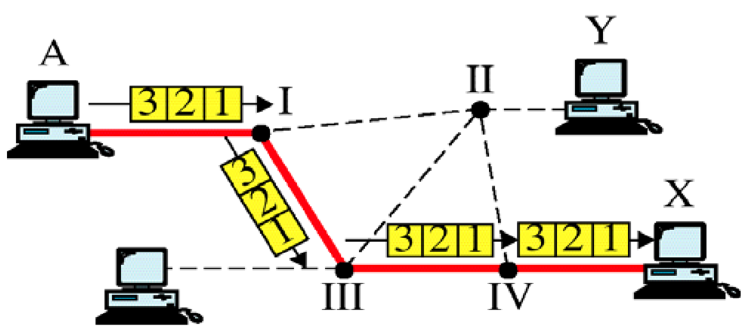
- A **route** is set up in the network between sender and receiver by making appropriate entries in the routing tables
- Resources may or may not be reserved for this route. If resources need to be reserved and are not available, the connection request is **blocked**
- Each packet contains its **virtual circuit identifier**

Virtual Circuit Packet Switching

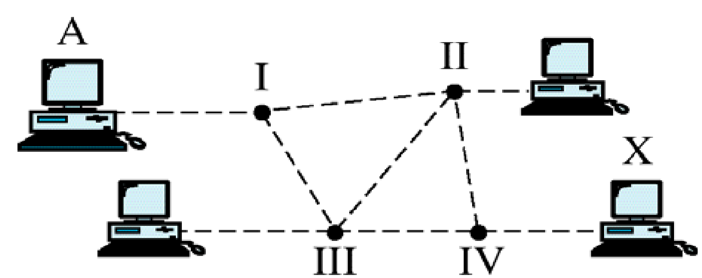


Connection establishment

Routers have <u>routing tables</u> telling them which output link to use for each established virtual circuit



Data transfer



Connection release

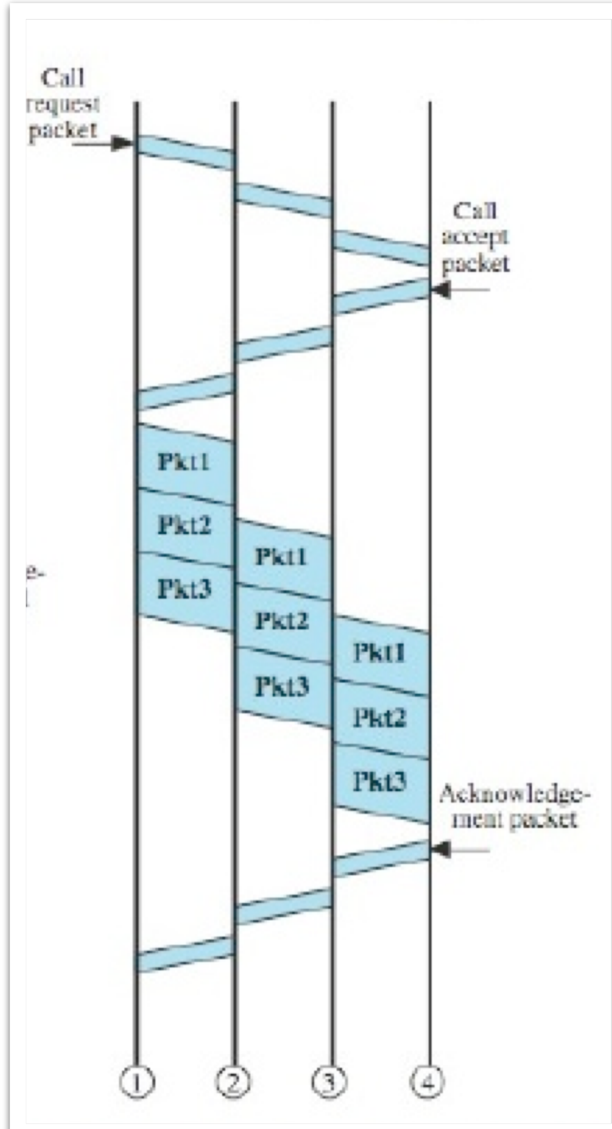
Virtual Circuit Packet Switching

- Connection set-up is required, which may cause a significant **delay**
- Network resources are ***not shared*** *at the same time*
  - Each packet monopolizes a link during its transmission, after which the link is available for other packet transmissions
- There is less work required at intermediate routers than for datagram packet switching
  - Given a packet's input link and virtual circuit identifier, the router can look up its routing table to find the output link

Virtual Circuit Packet Switching

- Virtual circuits not as robust to network problems as datagram packet switching
- Virtual circuit packet switching is a **compromise** between circuit switching and datagram packet switching
  - Circuit switching creates a **path** in the network; virtual circuit packet switching creates a **route** which exists only in software; datagram packet switching doesn't have routes
  - In circuit switching, the links in the path cannot be shared during the connection; in virtual circuit and datagram packet switching they can

Virtual Packet Switching



Multiplexing

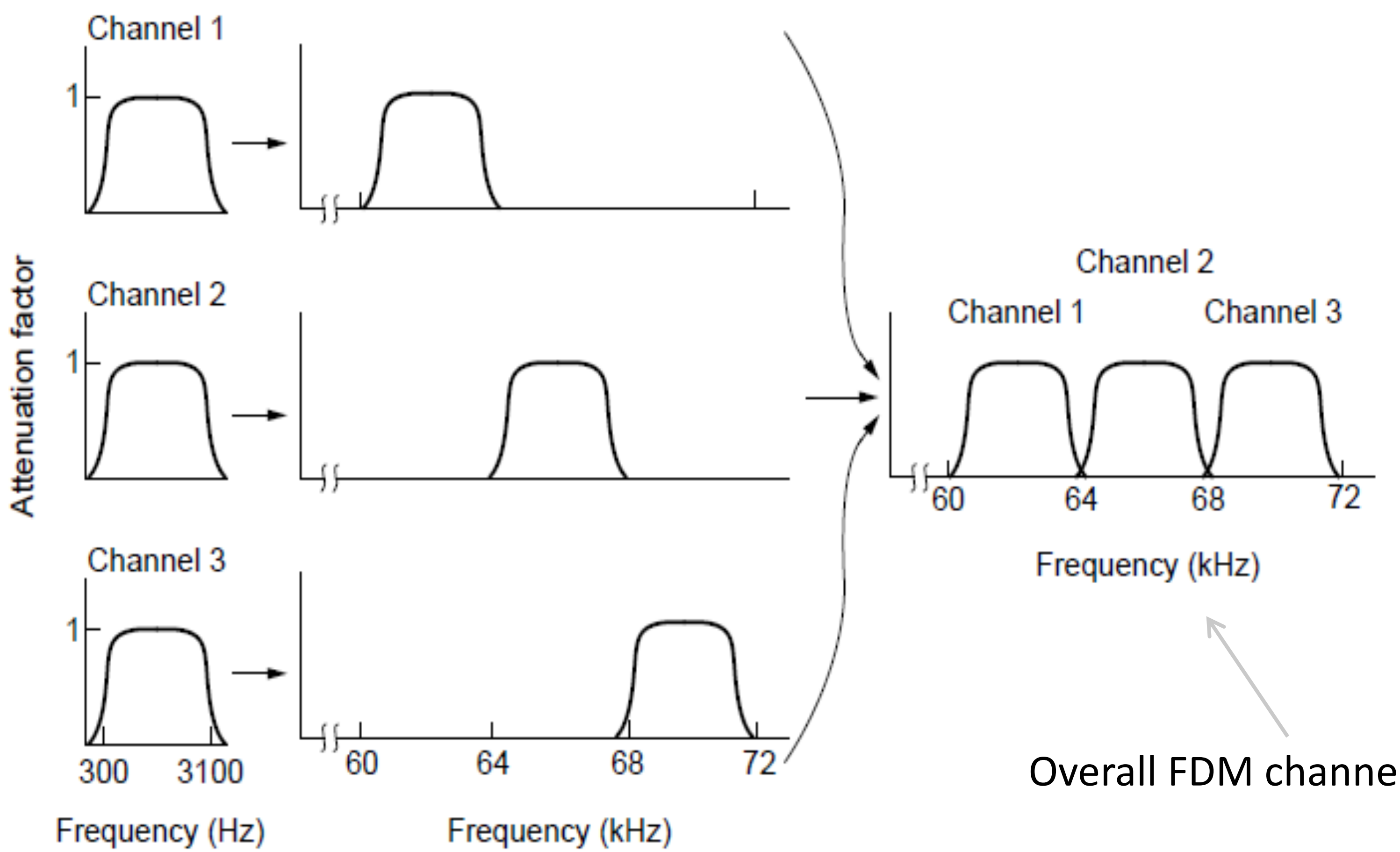
- Multiplexing is sharing a single link among multiple transmissions

- There are 3 basic possibilities for achieving this
  - **Frequency division multiplexing (FDM)**
  - **Time division multiplexing (TDM)**
  - **Statistical multiplexing**

Frequency Division Multiplexing

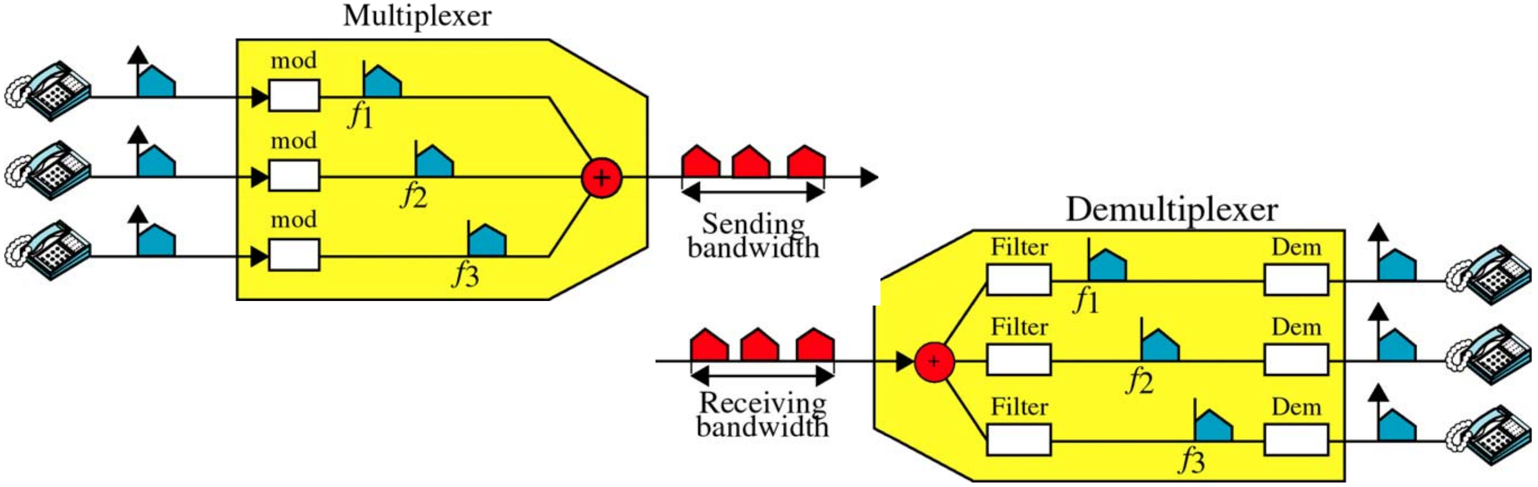
- Frequency division multiplexing is a technique by which the total **bandwidth** available in a communication link is divided into a series of **non-overlapping** frequency **sub-bands**
- Each sub-band is used to carry a **separate** signal
- This allows a single transmission medium such as the radio spectrum, a cable or optical fiber to be shared by multiple separate signals.

Frequency


Channel 1
Channel 2
Channel 3
Attenuation factor
1
300
3100
Frequency (Hz)
60
64
68
72
Frequency (kHz)
Channel 2
Channel 1
Channel 3
60
64
68
72
Frequency (kHz)
Overall FDM channel

Frequency Division Multiplexing

- Frequency division multiplexing requires a **multiplexer** to combine several sub-bands together and a **demultiplexer** to separate the sub-bands later

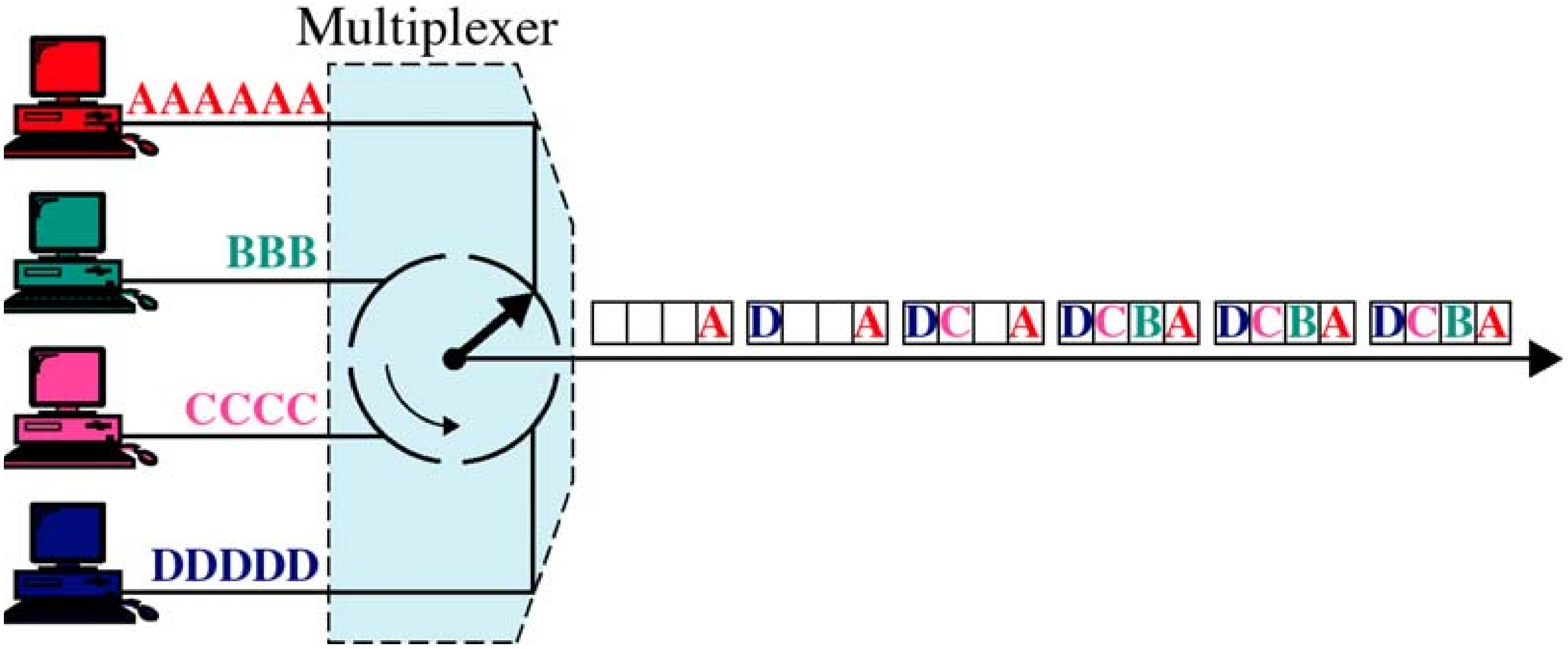


Time Division Multiplexing

- Time division multiplexing is a technique by which users take turns using the total **bandwidth** available in a communication link
- Bits from each input stream are taken in a fixed **time slot** and output to the aggregate stream.
- For this to work, the streams must be synchronized in time.
  - Small gaps of time can be added between the time slots to prevent messages getting confused

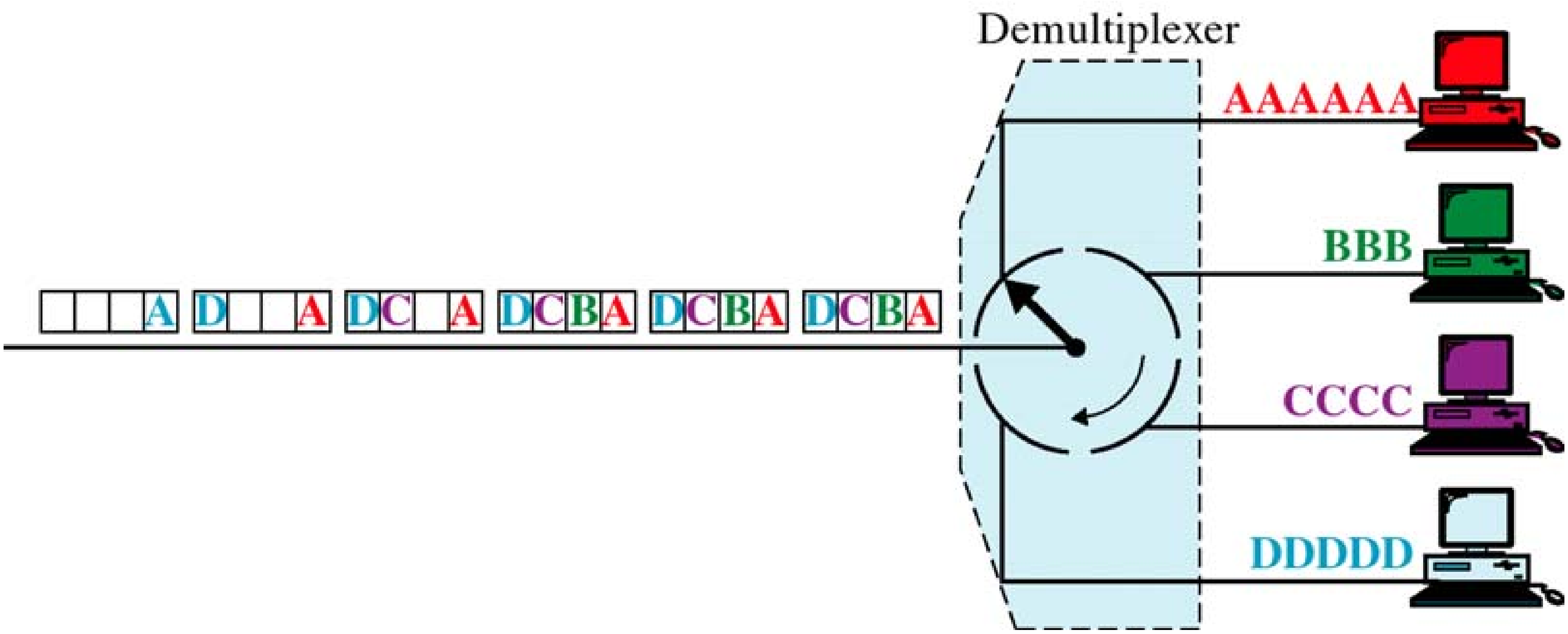
Time Division Multiplexing

- A multiplexer is required to combine the transmission into frames



Time Division Multiplexing

- A demultiplexer is required to split the frames back into the original transmissions



Statistical Multiplexing

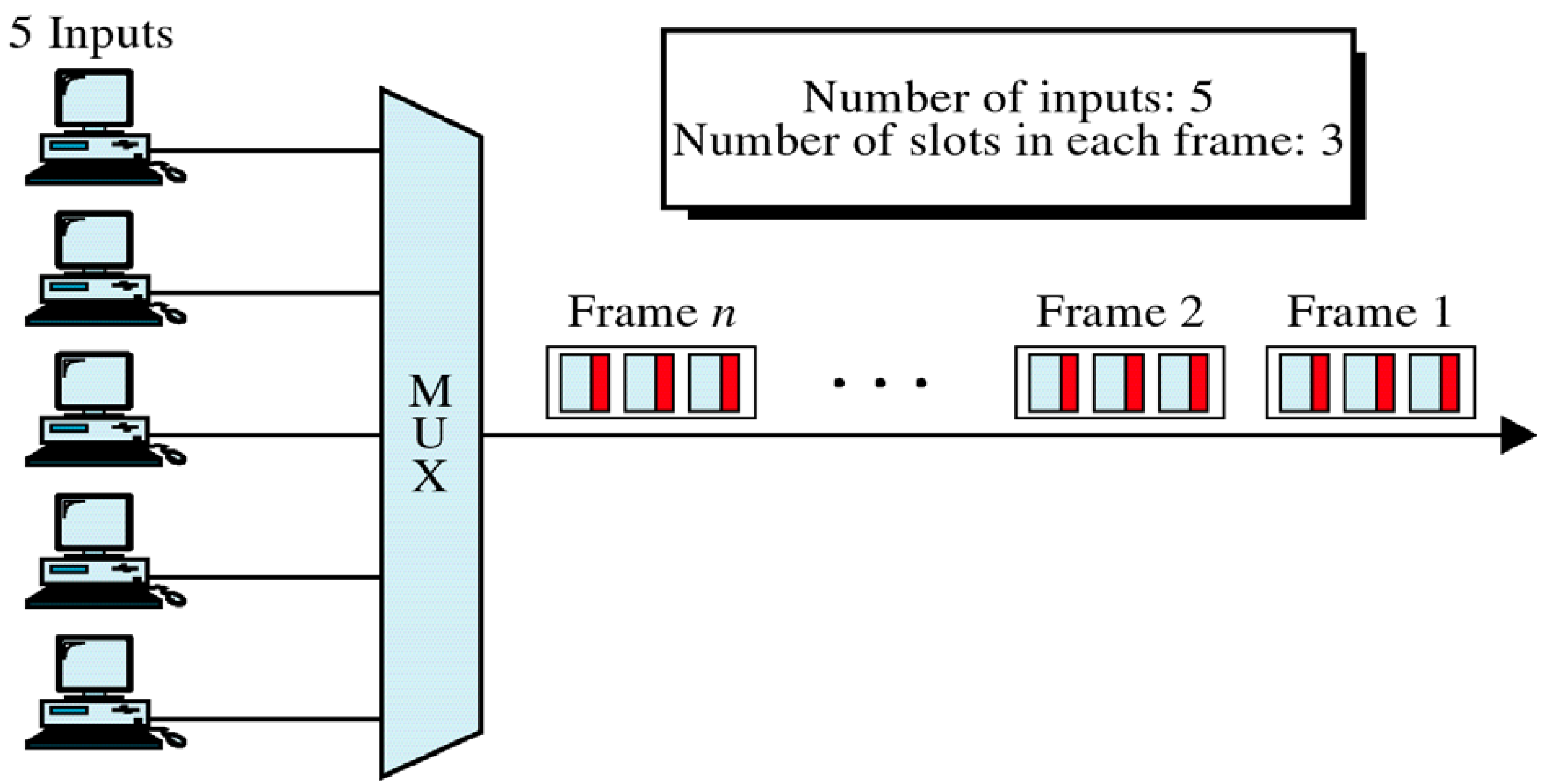
- Both frequency domain multiplexing and time domain multiplexing divide the link into **independent** channels

- Because there is no sharing allowed the system is **inefficient** if traffic is bursty

- In statistical multiplexing, the idea is that the link should **never be idle** when there is data to be transferred
- sharing based on the statistics of demand, rather than by giving each host a fixed fraction of the bandwidth that it may or may not use

- Sometimes this is called Statistical Time Division Multiplexing

Statistical Multiplexing



Transmission Modes

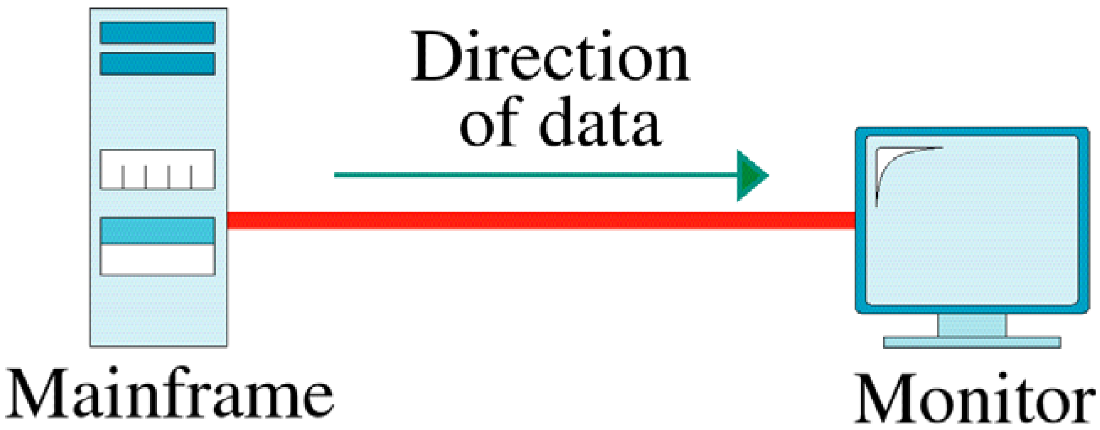
- There are three different modes of transmission that classify how data is transmitted

- These are
  - Simplex
  - Half-Duplex
  - Full-Duplex

Transmission Modes: Simplex

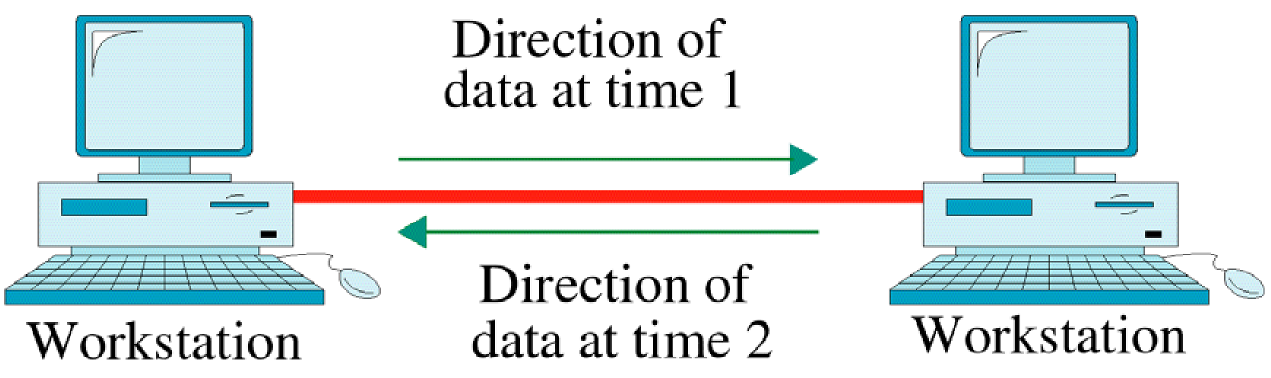
- In the simplex transmission mode data can only ever travel in **one direction**

- This is how television and radio are transmitted



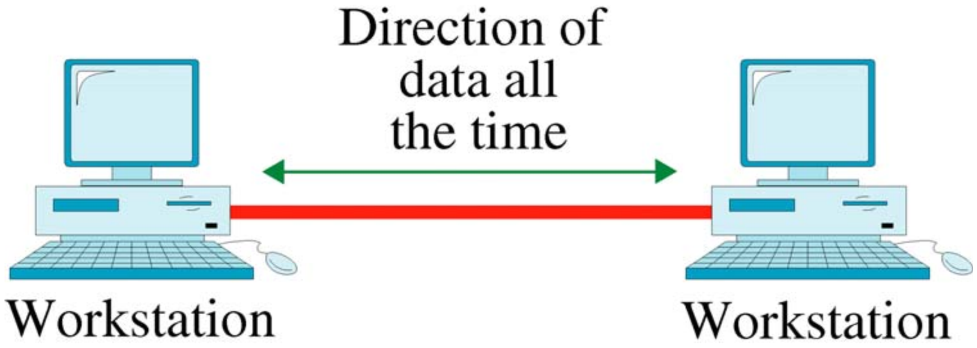
Transmission Modes: Half-Duplex

- In the half-duplex transmission mode data can be transmitted in both directions, but **never at the same time**
- A two-way radio is an example of a half-duplex transmission



Transmission Modes: Full-Duplex

- In the full-duplex transmission mode data can be transmitted in both directions **at the same time**
- A telephone call is an example of a full-duplex transmission



Network Performance

- Packet Delay
  - Nodal Processing delay
  - Queueing delay
  - Transmission delay
  - Propagation delay
- Bandwidth
- Throughput
- Efficiency

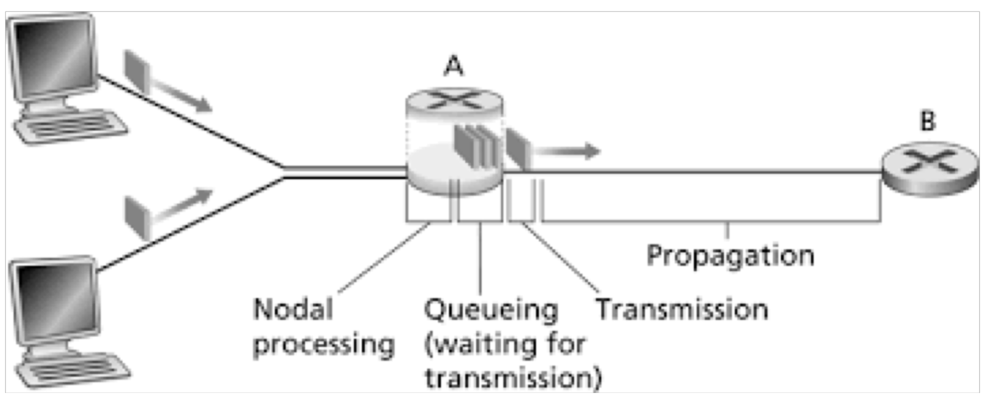
Metric Units

- In computer network, we normally use "bit" not "byte", that is, we use **"b"** not "B"
  - Use powers of 10 for **rates**, powers of 2 for storage
  - E.g., 1 Mbps = 1,000,000 bps, 1 KB = 1024 bytes
- The lowercase k in kbps is for historical reasons. Thus, **k**bps = **K**bps

Prefix	Exp.	prefix	exp.
K(ilo)	10^3	m(illi)	10^{-3}
M(ega)	10^6	μ(micro)	10^{-6}
G(iga)	10^9	n(ano)	10^{-9}

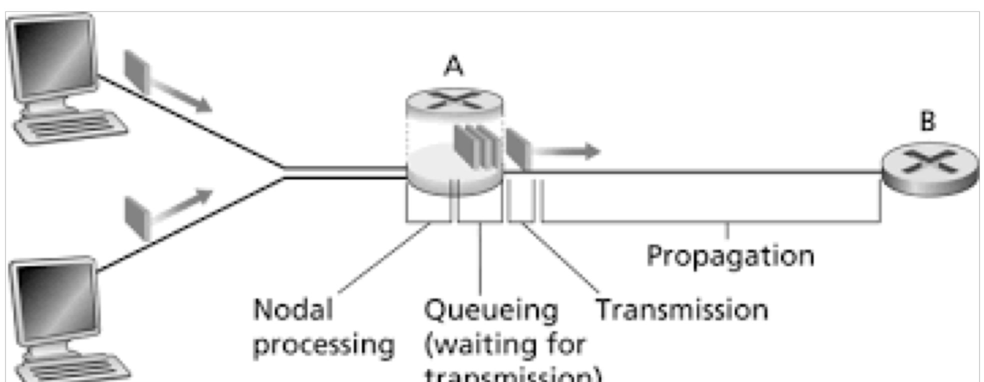
Packet Delay (d_{packet})

- how long it takes from sending the first bit of a packet to receive the last bit of such packet.
- $d_{packet} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$
- Nodal processing delay (d_{proc}): The time it takes routers to process the packet header.
  - During processing of a packet, routers may check for bit-level errors in the packet that occurred during transmission as well as determining where the packet's next destination is.



Packet Delay (d_{packet})

- Queueing delay (d_{queue}): time the packet spends in routing queues.
- Transmission delay (d_{trans}): time it takes to push the packet's bits onto the link.
- Propagation delay (d_{prop}): time for a signal to reach its destination over a communication link.



Computing Delays

- $d_{prop} = {}^{L}/_{C}$
  - Propagation delay is calculated by dividing the length ($\boldsymbol{L}$, measured in **meters**) of the signal path in the communication link by the signal propagation speed ($\boldsymbol{C}$, measured in **meters per second**)
- $d_{trans} = {}^{l_p}/_{V}$
  - Transmission time is calculated by dividing the average length ($\boldsymbol{l_p}$, measured in **bits**) of the a packet by the transmission rate ($\boldsymbol{V}$, measured in **bits per second**)
- d_{proc}: depends on the specifications of the router
- d_{queue}: depends on the network condition

Example 1

Consider an optical fiber 3000 km long with a transmitter transmitting at 1.5 Gbps (1 Gbps = 1 000 000 000 bps). The signal propagation speed in optical fiber is approximately 200 000 km/sec. Suppose packet switching is being used with a packet length of 2000 bits.

- Q1. What is the propagation delay along the fiber?
  - $d_{prop} = {}^{L}/_{C} = {}^{3*10^6}/_{2*10^8} = 1.5 * 10^{-2} second = 15ms$
- Q2. What is the packet transmission time?
  - $d_{trans} = {}^{l_p}/_{V} = {}^{2*10^3}/_{1.5*10^9} = 1.33 * 10^{-6} second = 1.33\mu s$

Example 1

Consider an optical fiber 3000 km long with a transmitter transmitting at 1.5 Gbps (1 Gbps = 1 000 000 000 bps). The signal propagation speed in optical fiber is approximately 200 000 km/sec. Suppose packet switching is being used with a packet length of 2000 bits.

- Q3. How many packets have been transmitted and are propagating over the fiber when the first bit reaches the destination ?
  - The number of packets = d_{prop} / d_{trans}=$1.5 * 10^{-2}/1.33 * 10^{-6}$ = 11250 **packets** (not bits!)

Example 2

Consider a route in a store-and-forward network going through 8 intermediate nodes. The packets contain 1000 bits and are transmitted at 64 kbps. Assume processing and propagation delays are 0. As a packet travels along the route, it encounters an average of 5 packets when it arrives at each node.

- How long does it take for the packet to get to the receiver if the nodes transmit on a “first come first served” basis ?

Example 2

- The transmission time for one packet at every node is
  - $d_{trans} = {}^{l_p}/_{V} = {}^{1*10^3}/_{6.4*10^4} = 1.5625 * 10^{-2} second$
- At each intermediate node, there are 5 packets ahead of ours. Due to the "first come first served" policy, it takes $(5 + 1) * d_{trans}$ to get "our" packet transmitted.
- The total travel time for our packet through the network (8 intermediate node) is $d_{trans} + 8 * 6 * d_{trans}$= $7.65625 * 10^{-3} second$

Example 2

- Note that the “pure” transmission delays for our packet only account for $d_{trans} + 8 * d_{trans}$= $0.14\ second$

- The rest are all queueing delays. This represents $(1 - {}^{9*d_{trans}}/_{49*d_{trans}})$ 81.63% of the total travel time

Queuing Delay

- Queuing delay can account for a **substantial** "extra" delay experienced by packets in the network.
- One way to reduce this queuing delay form some packets would be to use a different policy in the intermediate nodes, rather than first-come-first-served.
- This could result in some "higher-priority" packets getting to their receivers much quicker, while "lower-priority" packets would experience longer delays.

Other Measures

- Throughput
  - the rate of *successful* message delivery over a communication channel (*per-node throughput*) or a network (*network throughput*, which is a summation of all nodes' per-node throughput in this network).
  - measured in **bits per second** or packets per second
- Bandwidth
  - the maximum rate of data transfer across a given path
- Efficiency
  - $\frac{throughput}{bandwidth} * 100\%$
  - This is how well the communication link is being used.

Example 3

- 150 nodes are connected to a 1000 meter length of coaxial cable. Using some protocol, each node can transmit 70 packets/second, where each packet is 1000 bits long. The transmission rate at each node is 100 Mbps.
- Q1. What is the per-node throughput?
  - $thru_{node} = 70 * 1000 = 7 * 10^4 bps$
- Q2. What is the total throughput (of the 150 nodes)?
  - $thru_{total} = 150 * thru_{node} = 1.05 * 10^7 bps = 10.5Mbps$
- Q3. What is the efficiency of this protocol?
  - $\frac{thru\_total}{bandwidth} * 100\% = \frac{1.05*10^7}{100*10^6} * 100\% = 10.5\%$

Network Efficiency Discussion

- What would give us 100% efficiency, and why is the efficiency so far below 100% in our example?
- To reach 100% efficiency there would always have to be someone **transmitting "data" all the time**
- However the protocol used to decide who's turn it is to transmit introduces some delays
  - This is either because it permits collisions to occur, which must be recovered from; or because some "permission to transmit" token must be passed
- These delays result in a drop in efficiency