# FUNDAMENTAL : Communication Transmission

Week 3 – day 1

# Addressing Channel Impairment

- Adaptive Modulation
  - Can be performed adaptively according to channel condition
- Data Encoding :
  - Error Control Coding
  - Adding extra bit into the data so that error can be detected or corrected
- Equalization
  - Counteract the multipath effect of the channel
- Multiple input – multiple output (MIMO)
  - The use of multiple antenna either for transmitting and receiving
- Direct sequence spread spectrum
  - Signal occupies large bandwidth, so that problems in parts of it are overcome

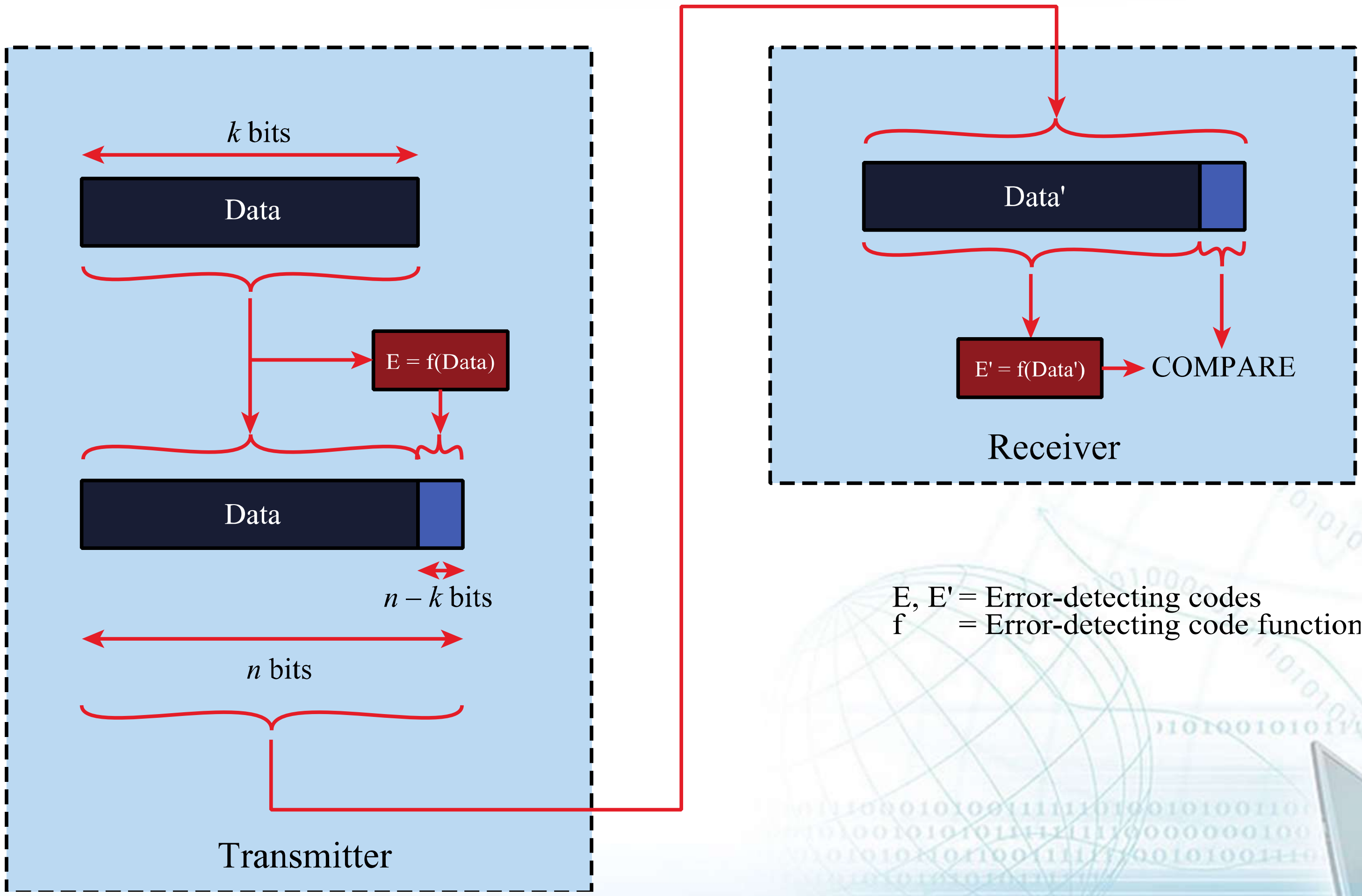
# Error Control Coding

- Coding and Error Control
  - Applied to digital data
    - Data would be a sequence of bits with certain length, called as frame
  - There are two approaches:
    - Error Detection Codes
    - Error Correction Codes
- Error Detection Codes
  - Simply detect the presence of error bit
  - Transmitter
    - For a given frame, an error-detecting code (check bits) is calculated from data bits
    - Check bits are appended to data bits
  - Receiver
    - Separates incoming frame into data bits and check bits
    - Calculates check bits from received data bits
    - Compares calculated check bits against received check bits
    - Detected error occurs if mismatch
      - Automatic repeat request (ARQ) protocols
        - Block of data with error is discarded
        - Transmitter retransmits that block of data

# Error Control Coding

- Forward error correction
  - Transmitter adds error-correcting code to data block
    - Block code
    - Convolutional Code
    - Turbo codes
  - Code is a function of the data bits
  - Receiver calculates error-correcting code from incoming data bits
  - If calculated code matches incoming code, no error occurred
  - If error-correcting codes don't match, receiver attempts to determine bits in error and correct

# Error Detection Techniques



# Error Control Coding

- Parity Check
  - Parity bit appended to a block of data
  - Even parity
    - Added bit ensures an even number of 1s
  - Odd parity
    - Added bit ensures an odd number of 1s
  - Example, 7-bit character [1110001]
    - Even parity [11100010]
    - Odd parity [11100011]
- Cyclic Redundancy Check
  - Transmitter
    - For a *k*-bit block, transmitter generates an (*n*-*k*)-bit frame check sequence (FCS)
    - Resulting frame of *n* bits is exactly divisible by predetermined number
  - Receiver
    - Divides incoming frame by predetermined number
    - If no remainder, assumes no error

# Cyclic Redundancy Code (CRC)

- Modulo 2 arithmetic
  - Since it is a digital data, where only two values are involved in each bit (‘0’ or ‘1’)
  - Use operation of addition, division, multiplication with no carrier bit
    - In computer logic, implementation of these operations uses XOR gates and shift register
      - Example : 1111

        0110 +

        1001 (NOT 10101)

The idea is following: A result of division operation between two integer numbers can be expressed as them summation of quotient and remainder. In general, it can be written as follows

$\frac{T}{P} = Q + R$, where T and P are integer numbers, Q is quotient and R is remainder.

Any change on T would change the remainder

# Generating CRC

- Append CRC bit into data bit
  - Suppose following bit sequence represent data, whose value is denoted by D, and pattern (or polynomial generator in CRC context), whose value is denoted by P respectively
    - D= 1010001101 (width equals to k=10 bits) and P=110101 (width equals to p=6 bits)
    - Number of bits to be added : (n-k)=p-1, thus total number of bits is n=k+p-1
  - Find remainder R as follows
    - Multiply D with $2^{(n-k)}$, yielding 101000110100000
    - Divided the product with P, which then yielding R (illustration is given in the next slide)
  - Append R to D (thus total frame length, n, equals to n=k+p-1=15)

# Modulo 2 Division Illustrated

```
                 1101010110  ──→ Q
               ______________________
P = 110101   / 1010001101 00000       ──→
               110101
               ------
                 111011
                 110101
                 ------
                   11101
                       0
                   -----
                   111010
                   110101
                   ------
                     11111
                         0
                     -----
                     11111 0
                     11010 1
                     -------
                       101 10
                            0
                       ------
                       101 100
                       110 101
                       -------
                        11 0010
                        11 0101
                        -------
                           1110  ──→ R
```

$T = 2^5 . D = (2^5). 1010001101$

$= 1010001101\ 00000$

$$\frac{T}{P} = Q + R$$

Append (n-k) bits of R to D. If number of bits of R is less than (n-k), then add prefix ‘0’s as necessary to it. Thus the frame to be transmitted would be **1010001101 01110** (add prefix ‘0’ to R)

# Modulo 2 Division Illustrated

```
                 11010101100  ──→ Q
                ___________________
P = 110101    / 1010001101 01110
                110101
                ------
                  111011
                  110101
                  ------
                     11101
                         0
                     -----
                     111010
                     110101
                     ------
                         11111
                             0
                         -----
                         11111 0
                         11010 1
                         -------
                             101 11
                                  0
                             ------
                             101 111
                             110 101
                             -------
                               11 0101
                               11 0101
                               -------
                                    00  ──→ R
```

If received frame is T'= **1010001101 01110** (which means no error occurs), then find

$$\frac{T'}{P} = Q' + R$$

If R = 0 means no error occurs

Check R again if any of one bit is flipped !

# CRC : Polynomial Format

- The data bits, pattern are written in polynomial format
  - D= 1010001101 $\equiv D(x) = x^9 + x^7 + x^3 + x^2 + 1$
  - T = $x^5 . D \equiv T(x) = x^{14} + x^{12} + x^8 + x^7 + x^5$
  - P=110101 $\equiv P(x) = x^5 + x^4 + x^2 + 1$

- Widely used versions of *P(x)*
  - 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
    - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

# CRC : Application Notes

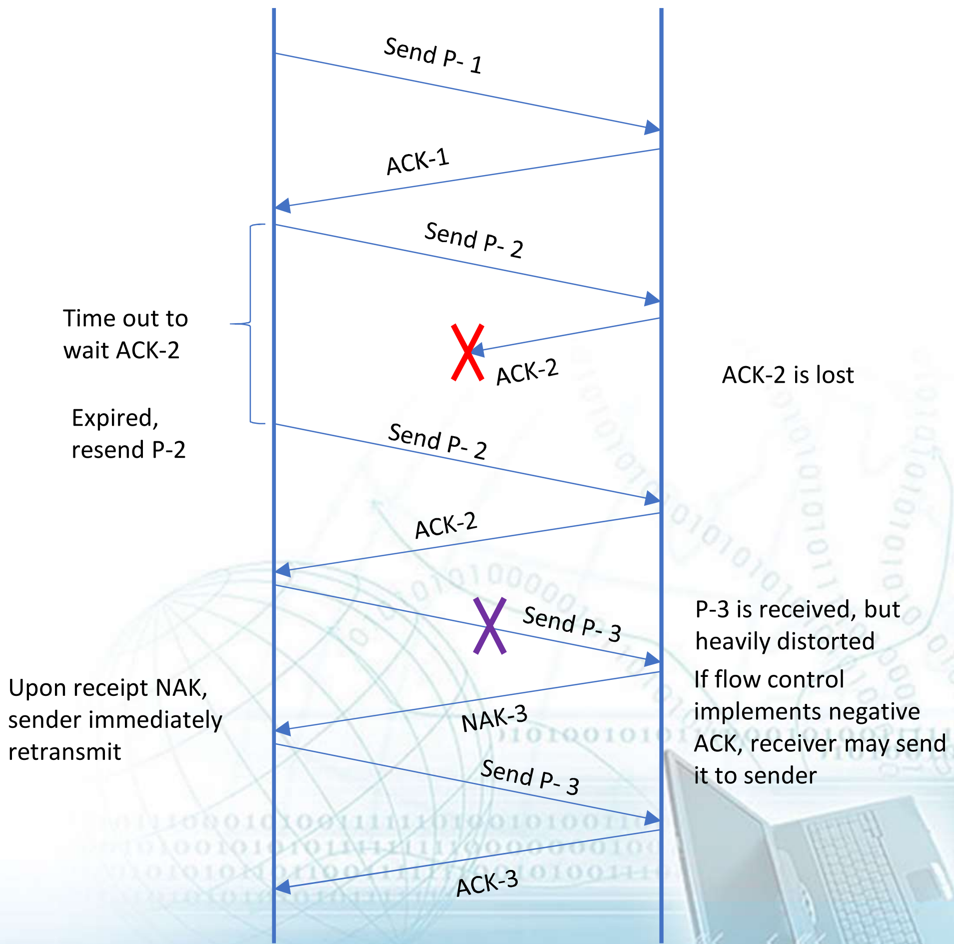
- With suitably chosen of P(x), following pattern of error is detectable
  - All single bit errors, if P(x) has more than one nonzero term
  - All double bit errors, as long as P(x) has a factor with at least three terms
  - Any odd number of errors, as long as P(x) contains a factor (x+1)
  - Any burst error for which the length of the burst is less than or equal to (n-k)
- If an error were detected, then data needs to be retransmitted
  - Channel with low error probability and short propagation delay may benefit from this technique.
  - However, a channel with high error rate or having a long propagation delay, such as wireless link and satellite link. Frequent retransmission may cause inefficient system
  - Forward Error Correction code is exploited

# Automatic Repeat Request

- Applied with conjunction with error detection schemes
  - Sender needs to retransmit the data when the data being received by received suffers from error ➔ ARQ error control
  - Although it works with error detection scheme, ARQ schemes can be viewed from data link layer OR transport layer point of view
    - In multi-hop links, retransmission can be done on per hop basis (which is relevant to data link layer) or end to end retransmission (which is relevant to transport layer)
    - Regardless of these two layers, basic mechanism of ARQ schemes remain the same
    - Work closely with flow control
      - Technique for assuring that a transmitting entity does not overwhelm a receiving entity with data.

# Stop & Wait Protocol

- Stop and Wait protocol is a simple example used for ARQ
  - Sender sends one packet and waits for acknowledgement from receiver. It would not transmit subsequent packet unless getting acknowledgement packet from receiver
  - If sender did not get acknowledgement or experience time out, sender may retransmit the packet

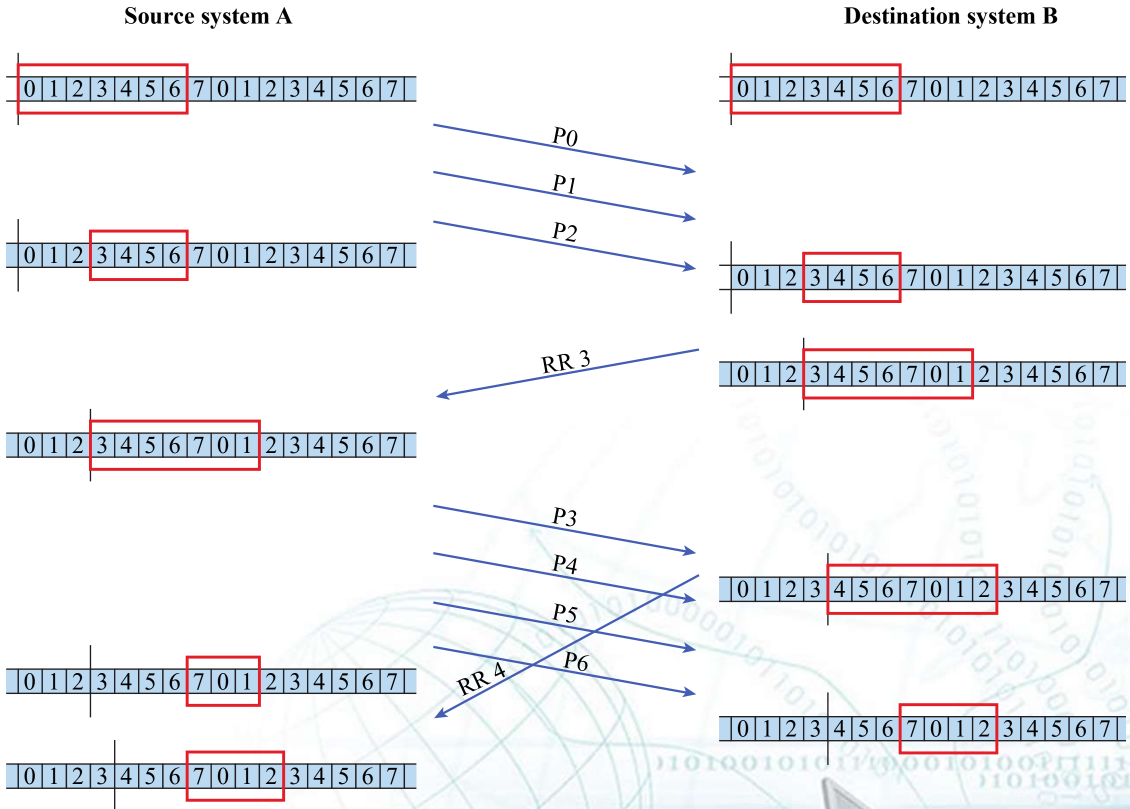


# Flow Control

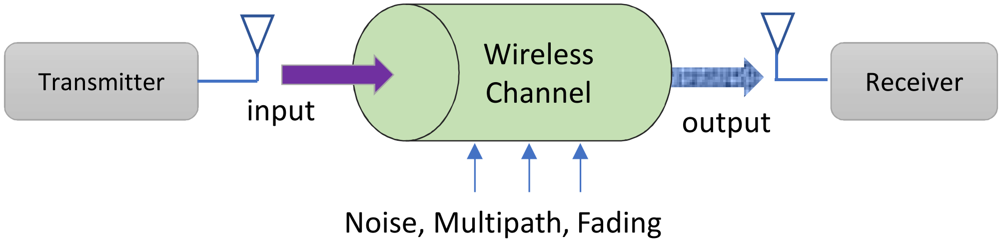
- Assures that transmitting entity does not overwhelm a receiving entity with data
- Protocols with flow control mechanism allow multiple information packet in transit at the same time
  - Generic name for block of bits that represents data to exchanged at any protocol or communication layer
- Packets arrive in same order they're sent
- Sliding-window flow control
  - Transmitter maintains list (window) of sequence numbers allowed to send
  - Receiver maintains list allowed to receive

# Sliding Window Illustrated

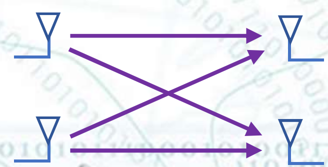
- Reasons for breaking up a block of data before transmitting:
  - Limited buffer size of receiver
  - Retransmission of packets due to error requires smaller amounts of data to be retransmitted
  - On shared medium, larger packets occupy medium for extended period, causing delays at other sending stations



# Space Diversity




- Space diversity – techniques involving physical transmission path, spacing antennas
- Based on the illustration above, we may have SISO (Single Input Single Output), MISO(Multiple Input Single Output), SIMO(Single Input Multiple Output), and MIMO(Multiple Input Multiple Output)
- Multiple input multiple output (MIMO)
  - If being treated properly, signal diversity due to multipath can be advantageous.
  - MIMO can be also used to increase the reliability of the link
    - Each antenna can be seen as independent channel. Each transmitter's antenna emits different stream of data

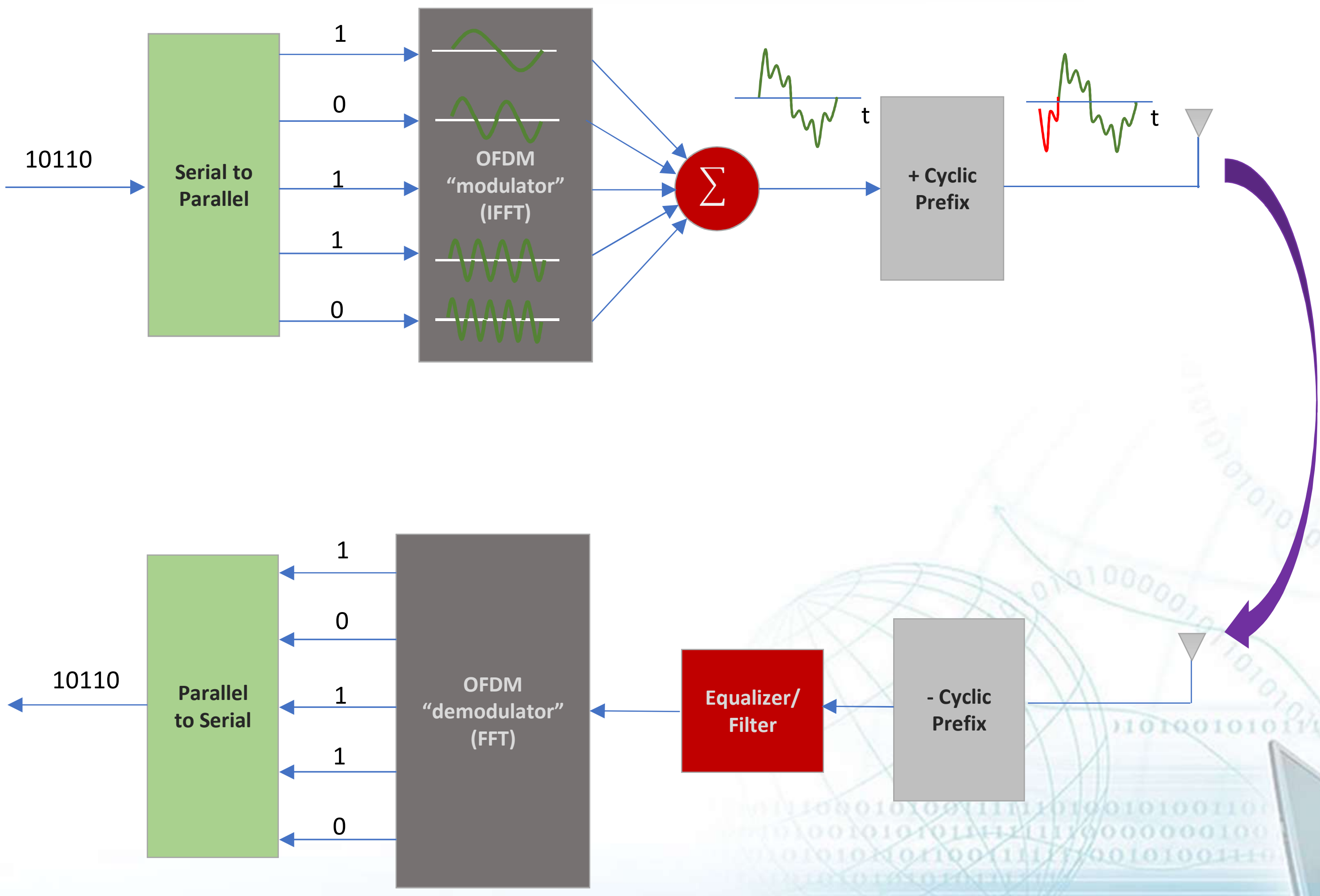


2x2 MIMO

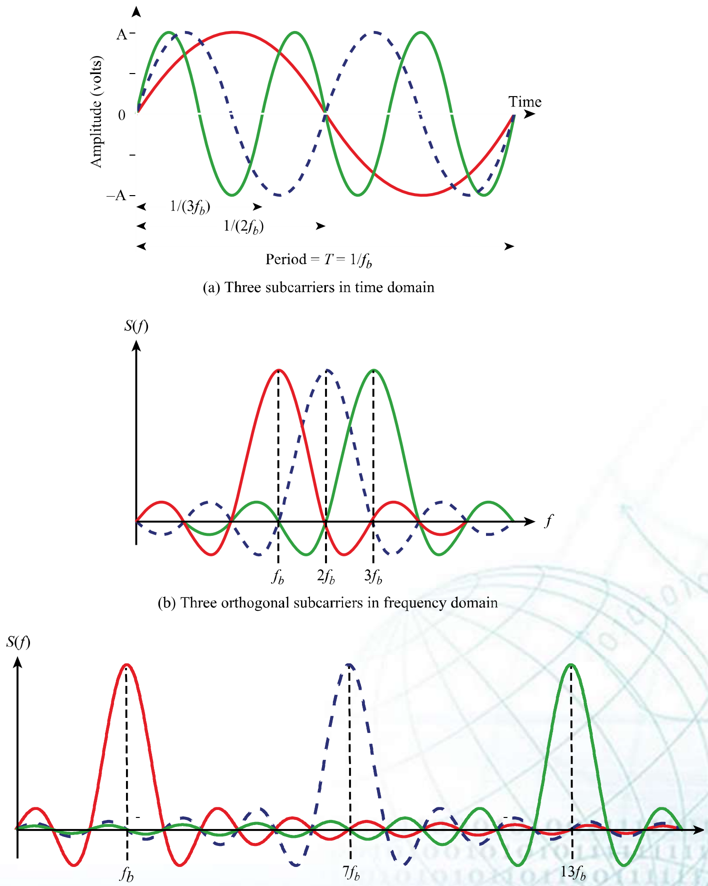
# Frequency Diversity

- Frequency diversity – techniques where the signal is spread out over a larger frequency bandwidth or carried on multiple frequency carriers (Actually, we have talked about this in previous slide!)
  - Frequency Division Multiplexing (FDM), orthogonal FDM (OFDM)
  - OFDM
    - Also called multicarrier modulation
    - Start with a data stream of $R$ bps
      - Could be sent with bandwidth $Nf_b$
      - With bit duration $1/R$
    - OFDM splits into $N$ parallel data streams
      - Called *subcarriers*
      - Each with center frequency $f_b$
      - And data rate $R/N$ (bit time $N/R$)

# OFDM : Diagram Block



# Orthogonality on OFDM




(a) Three subcarriers in time domain

(b) Three orthogonal subcarriers in frequency domain

(c) Three carriers using traditional FDM

# OFDM

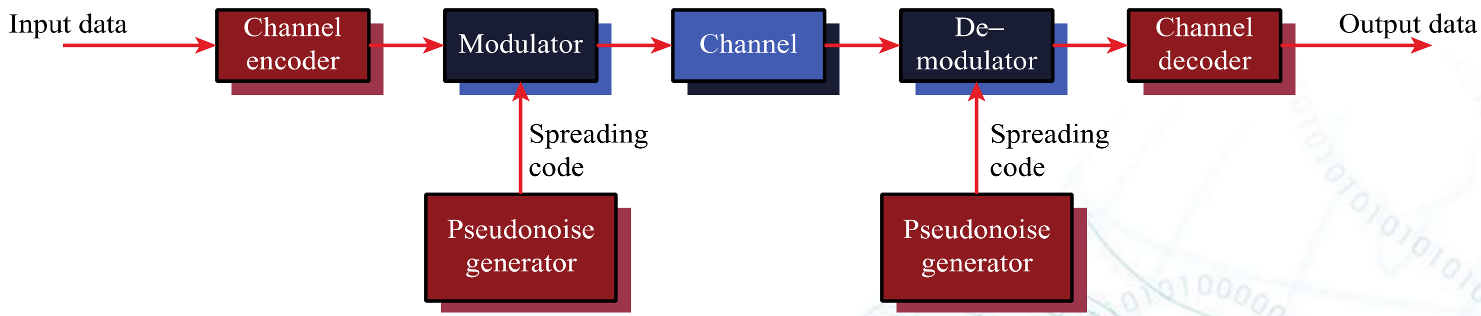
- The spacing of the $f_b$ frequencies allows tight packing of signals
  - Actually with overlap between the signals
  - Signals at spacing of $f_b$ ,2$f_b$, 3$f_b$ ,etc.
- The choice of $f_b$ is related to the bit rate to make the signals *orthogonal*
- Traditional FDM makes signals completely avoid frequency overlap
  - Because of tight packing of signals, overlap in OFDM greatly increases capacity.
  - Although these frequencies overlap, orthogonal properties of the frequencies prevents interference

# OFDM : Pros and Cons

- Frequency selective fading only affects some subcarriers
- More importantly, OFDM overcomes inter-symbol interference (ISI)
  - ISI is a caused by multipath signals arriving in later bits
  - OFDM bit times are much, much longer (by a factor of *N*)
    - ISI is dramatically reduced
  - OFDM's long bit times eliminate most of the ISI
  - OFDM also uses a *cyclic prefix* (CP) to overcome the residual ISI
    - Adds additional time to the OFDM symbol before the real data is sent
    - Acted like *guard interval* . Instead of an empty guard interval, a copy of the last OFDM symbol is put in front of the symbol
    - ISI diminishes before the data starts
- Cons
  - Poor performance under time selective fading and doppler spread
  - Sensitive to non-linear distortion

# Spread Spectrum

- Spread spectrum technique spreads the information signal over a wider bandwidth
  - This technique makes jamming and interception more difficult, since the receiver will see the modulated signal like a noise
  - This technique was initially developed for military and intelligence requirement

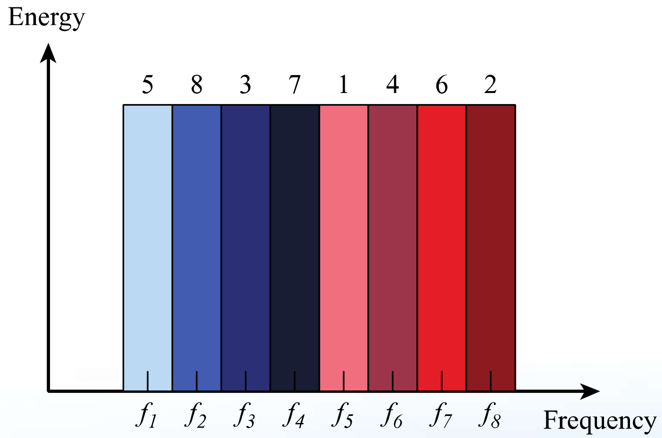



General model of spread spectrum in Digital Communication System

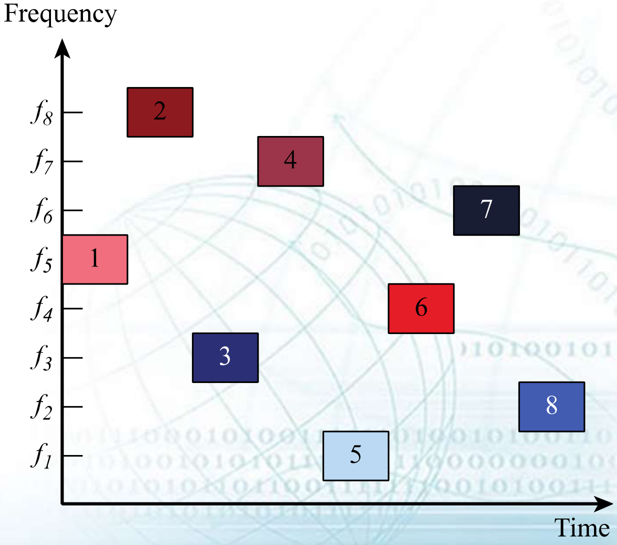
- Analogy : speaking with different language within a crowd that also speak different language
- Two important known spectrum techniques : Frequency hopping and direct sequence

# Frequency Hopping Spread Spectrum

- Signal is broadcast over a seemingly random series of radio frequencies, hopping from frequency to frequency at fixed interval
  - Receiver should be synchronized with transmitter
  - Eaves dropping only hears a random blips, jamming in one frequency only knocks out a few of bits
  - Bluetooth (IEEE 802.15.1) uses FHSS for its signal transmission, number of frequencies  C=80



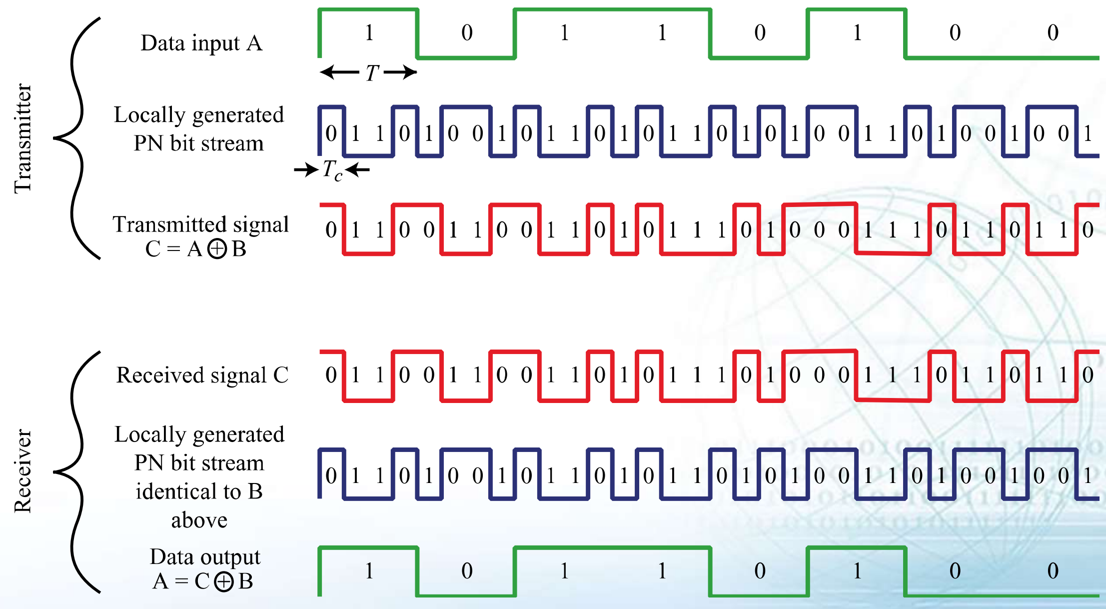

(a) Channel assignment



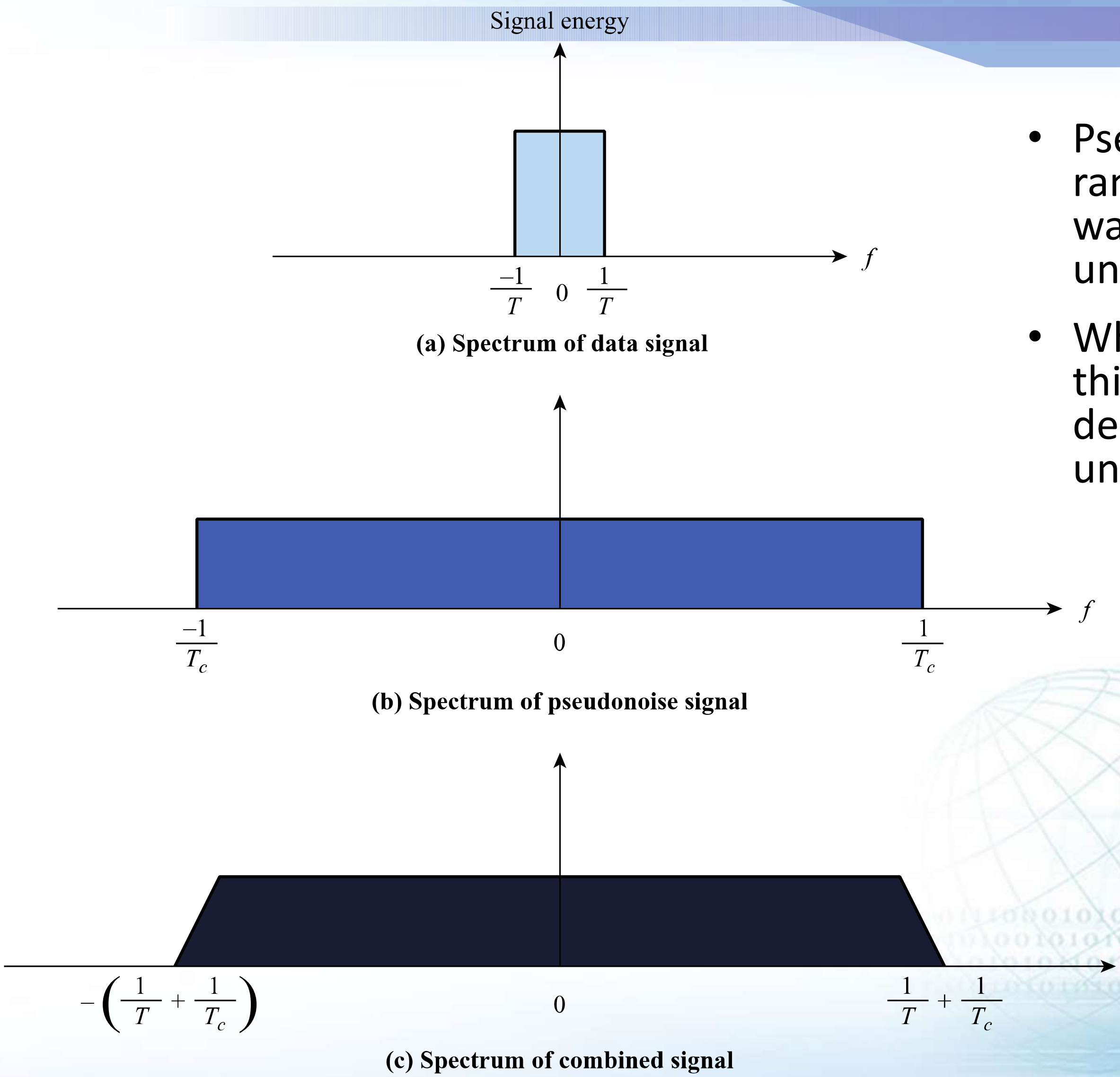

(b) Channel use

# Direct Sequence Spread Spectrum (DSSS)

- Each bit in original signal is represented by multiple bits in the transmitted signal
- Spreading code spreads signal across a wider frequency band
  - Spread is in direct proportion to number of bits used
- One technique combines digital information stream with the spreading code (called as Pseudonoise (PN) bit stream using exclusive-OR



# Spectrum of DSSS




- Pseudonoise code : It is not truly random, it is generated in such way that the bit in this code are uncorrelated stochastically
- When AWGN noise is applied to this code, the power spectral density of the noise remain unchanged