Digital Data Communication Techniques

Dr Steve Gordon

ICT, SIIT

Contents

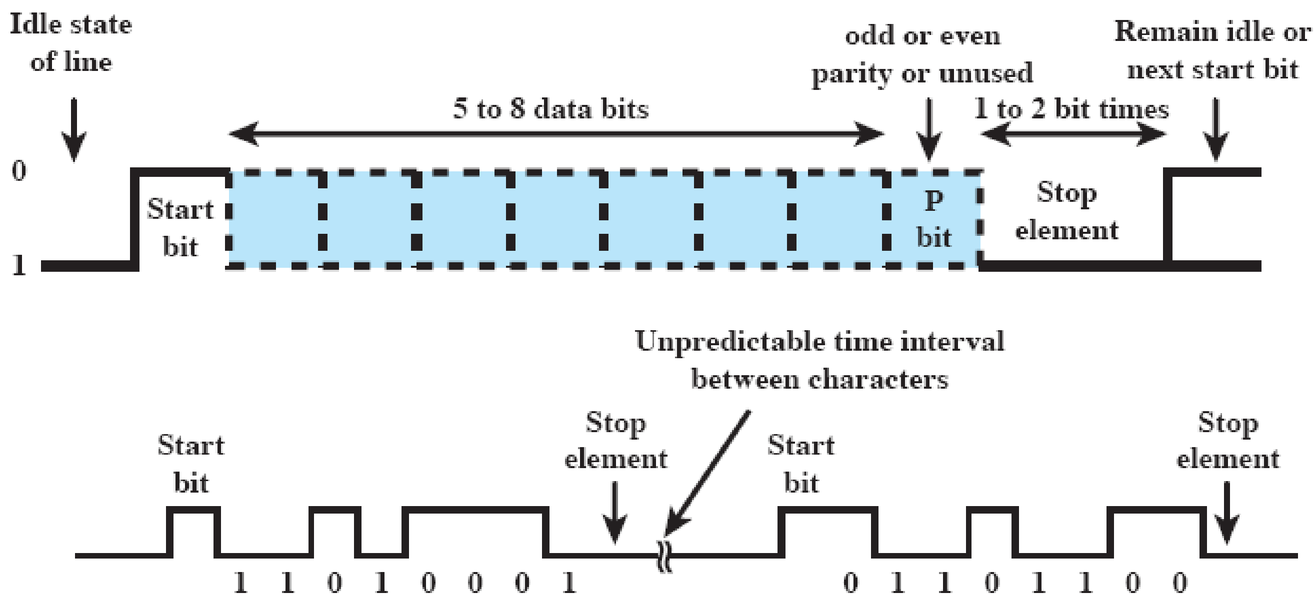
- Asynchronous and Synchronous Transmission
- Handling Errors
  - Types of Errors
  - Error Detection
  - Error Correction
- Line Configurations
  - Topology
  - Duplex

Asynchronous and Synchronous Transmission

- Problem:
  - Transmitter sends a sequence of bits
  - Receiver must know when a bit starts and finishes
  - Sender and receiver must be synchronized
    - If not, then receiver may sample at wrong time and get bit errors
- Two solutions to synchronizing clocks
  - Asynchronous transmission
  - Synchronous transmission

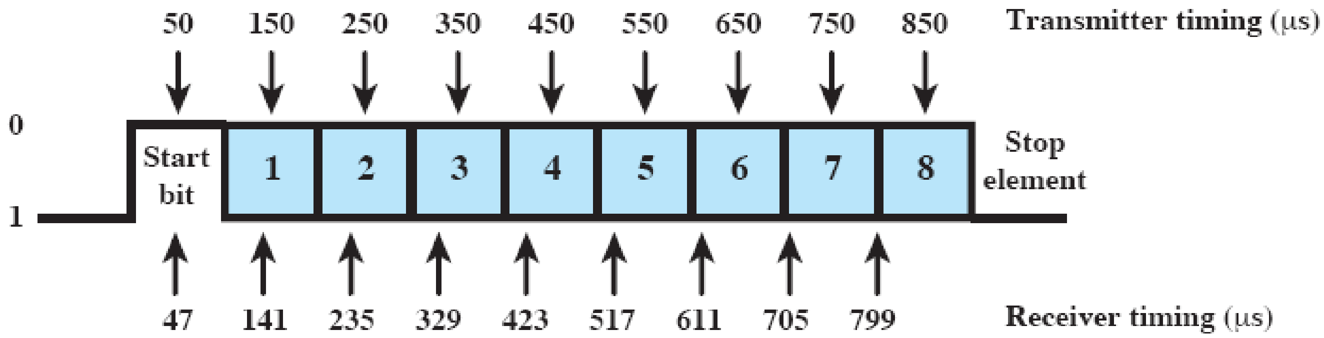
Asynchronous Transmission

- Send a character at a time (characters is 5 to 8 bits)
  - A special bit is used to indicate the start of character and end of character
  - Example:
    - When no character being transmitted, binary 1 is transmitted
    - Start bit is binary 0; then character bits are sent (5 or 8 bits); then stop bit (binary 1)



Asynchronous Transmission

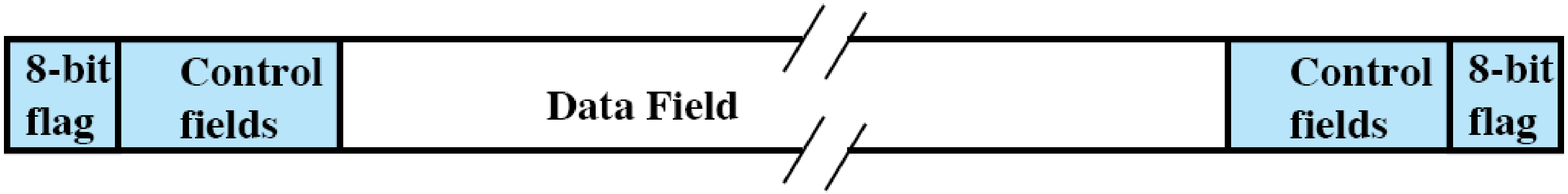
- Can tolerate modest timing errors
  - Example: 10kb/s data rate; receiver is fast by 5% can still correctly sample data
    - If receiver fast by 6% (figure below), then error



- Asynchronous transmission is simple and cheap
  - Requires overhead of two to three bits per character
  - Good for data with large gaps, e.g. keyboard data

Synchronous Transmission

- Block of data transmitted sent as a frame
- Clocks at transmitter and receiver must be synchronized
  - Can use separate clock line
  - Or embed clock signal in data
- Need to indicate start and end of block
  - use pre-amble and post-amble bit patterns
- More efficient (lower overhead) than asynchronous transmission



- Example: HDLC (covered later) contains 48-bits of pre/post-amble. With 1000 bytes of data, overhead is 0.6% (compared to about 20% with asynchronous)

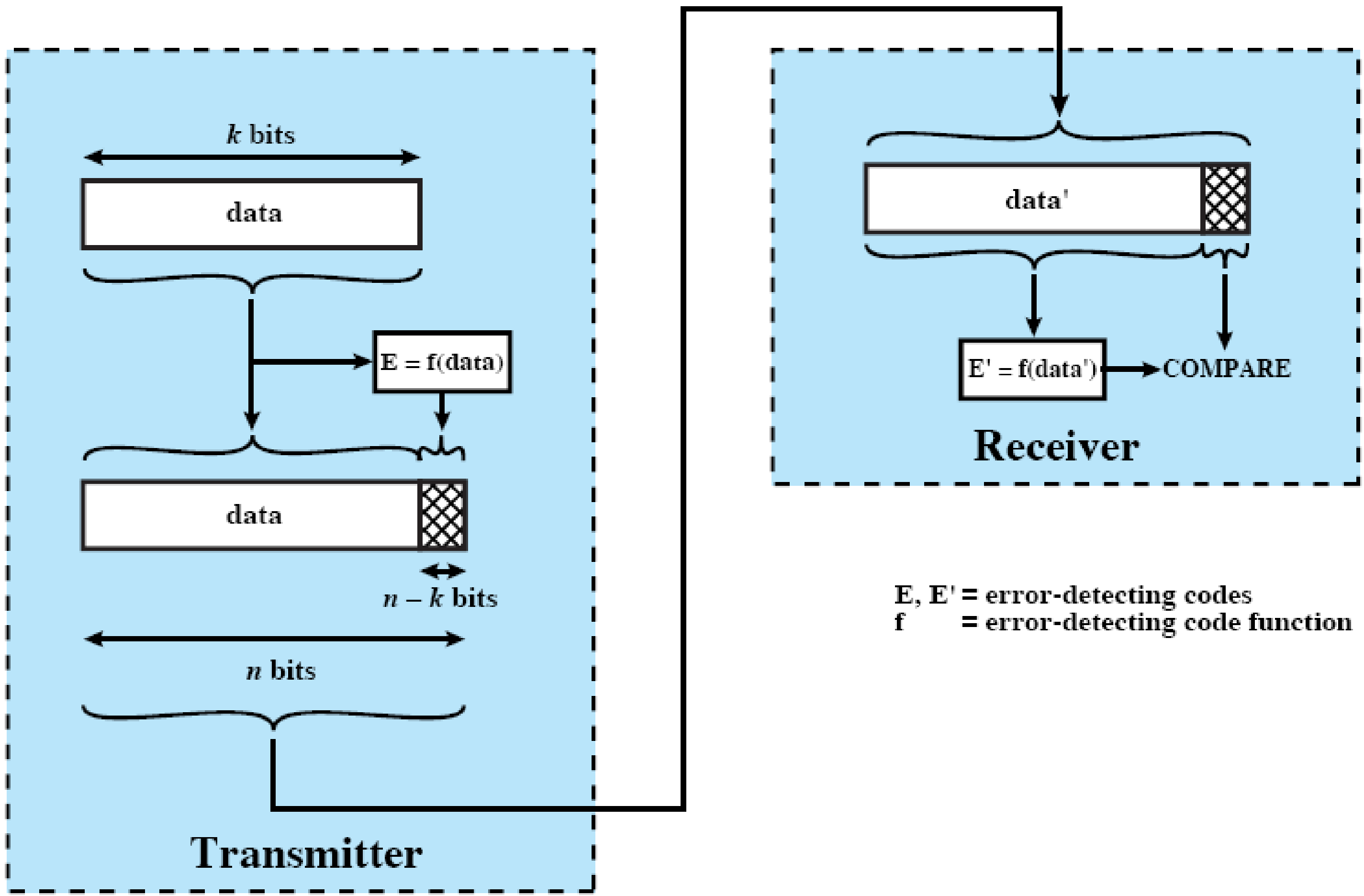
Types of Error

- An error occurs when a bit is altered between transmission and reception
  - E.g. transmitter sends a 1 but receiver thinks it is a 0
- Single bit errors
  - only one bit altered
  - caused by white noise, the SNR is too low for receiver to determine the correct bit
- Burst errors
  - Contiguous sequence of *B* bits in which first last and any number of intermediate bits in error
  - caused by impulse noise or by fading in wireless
  - effect greater at higher data rates

Error Detection

- Will always have errors
- Detect the errors (so can retry or inform higher layer)
  - Transmitter adds extra information to transmitted data, i.e. an error-detecting code
  - Receiver recalculates the error-detecting code from received data, and compares to received error-detecting code
  - If the same, good. If not, then error (in data or code)
    - Still a chance that an error is not detected
- Simple Error Detecting: Parity Check
  - Single parity bit added to character to make the number of 1's even (if using even parity) or odd (if using odd parity)
    - E.g. assume odd parity is used: a 7-bit IRA character 1110001 is sent with an eighth parity bit set to 1. Transmitted: 11110001
    - If 1 bit is in error, receiver will detect it: Receive 11100001
    - If 2 (or even number) bits in error, then not detected

Error Detection Process



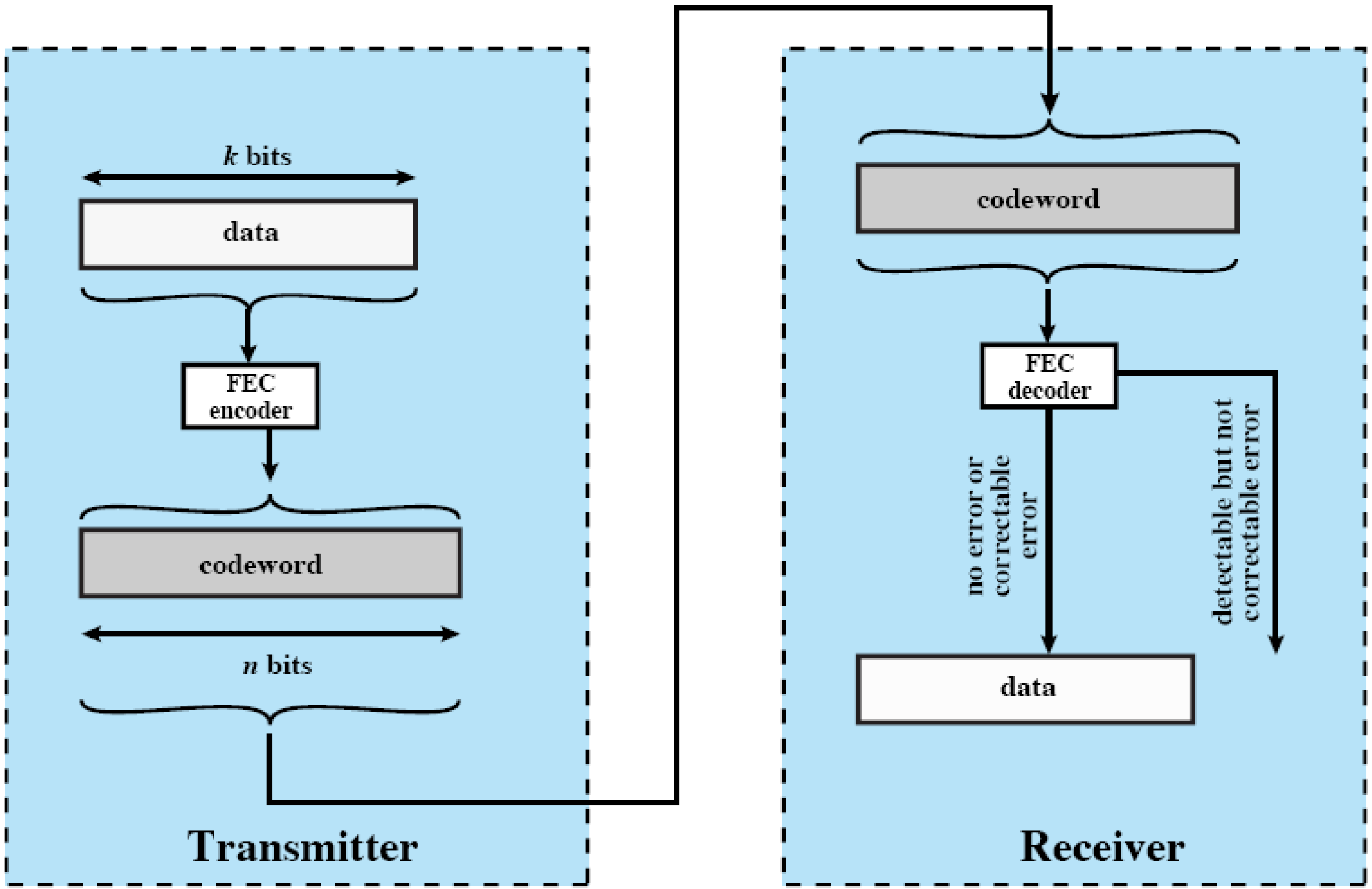
Cyclic Redundancy Check (CRC)

- One of most common and powerful checks
- For block of *k* bits, transmitter generates an *n* bit frame check sequence (FCS)
- Transmits *k*+*n* bits which is exactly divisible by some number
- Receiver divides frame by that number
  - if no remainder, assume no error
  - for math, see Stallings chapter 6

Error Correction

- Correction of detected errors usually requires data block to be retransmitted
- Not appropriate for wireless applications
  - Bit error rate is high causing lots of retransmissions
  - When propagation delay long (satellite) compared with frame transmission time, resulting in retransmission of frame in error plus many subsequent frames
- Instead need to correct errors on basis of bits received
- Error correction provides this

Error Correction Process



How Error Correction Works

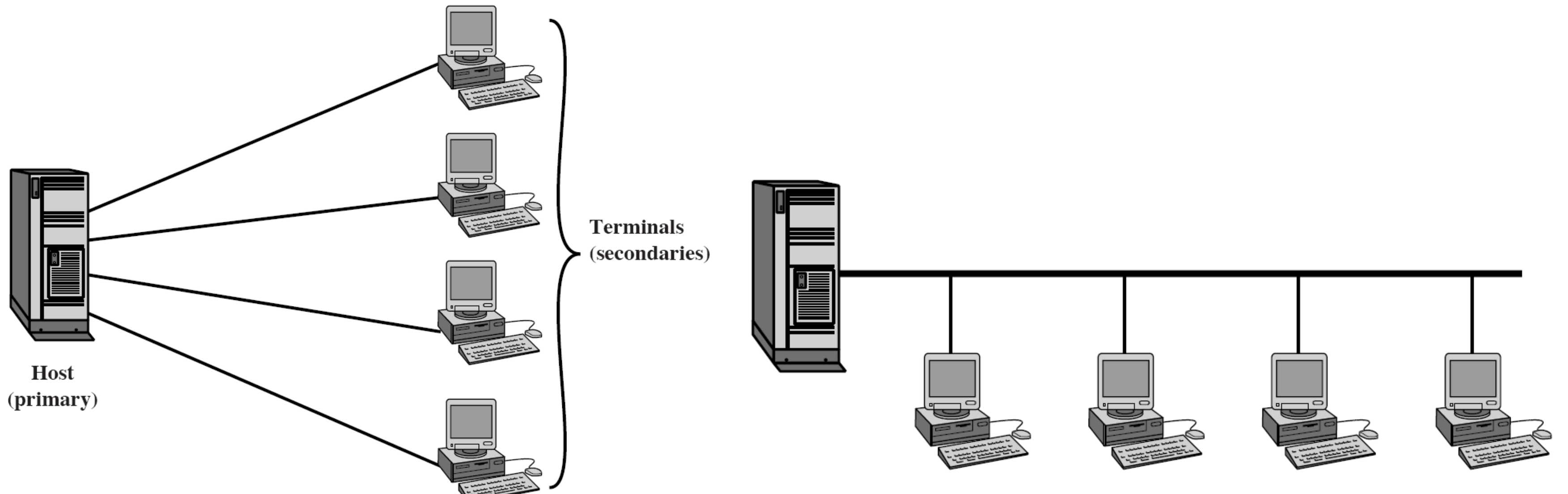
- Transmitter adds redundancy to transmitted message
- Receiver applies FEC decoder:
  - If no bit errors, input to decoder is same as original codeword, and original data is output
  - Certain error patterns, decoder will detect and correct errors (decoder outputs the original data)
  - Certain error patterns, decoder will detect (but not correct) errors
  - Certain (often rare) error patterns, decoder will not detect nor correct errors (decoder outputs data which is in error)
- Example: block error correction code
  - Map k bit input onto an n bit codeword
  - Each codeword is distinctly different
  - If get error assume codeword sent was closest to that received
- Results in reduced effective data rate
  - $(n-k)/n$ is the redundancy of the code; k/n is the code rate
  - ½ rate code uses double capacity of uncoded system

Example Error Correcting Code

- Hamming Distance
  - Number of bits of two *n*-bit sequences that disagree
    - v_1 = 011011 v_2 = 110001
    - $d(v_1,v_2) = 3$
- Our ECC maps two bits into 5 bit codeword (k=2, n=5)
  - 00 00000
  - 01 00111
  - 10 11001
  - 11 11110
- If receiver receives an invalid codeword (C_r), then assumes the codeword which is minimum Hamming distance from C_r is the transmitted codeword (C_t)
  - Only works if only 1 unique codeword with minimum distance

Line Configuration

- Topology
  - Physical arrangement of stations on medium
    - Point to point - two stations
      - such as between two routers / computers
    - Multi point - multiple stations
      - traditionally mainframe computer and terminals
      - now typically a local area network (LAN)



(a) Point-to-point (b) Multipoint

Line Configuration

- Duplex
  - Classify data exchange as half or full duplex
    - half duplex (two-way alternate)
      - only one station may transmit at a time
      - requires one data path
      - Wireless transmission systems usually half-duplex
        - » Since transmitter and receiver usually use same components/antenna
    - full duplex (two-way simultaneous)
      - simultaneous transmission and reception between two stations
      - requires two data paths
        - » separate media or frequencies used for each direction