ONLY
GOD

2021

Chapter 3

DATA TRANSMISSION

TRANSMISSION MODES

Data Link

- Data Communications: exchange of digitally encoded information between 2 DTEs
  - (i) Information: user data
  - (ii) Error control: detect and respond to corrupted data
    - Corruption from physical layer ---> bit erroneously recovered
    - Error detection ---> probability of accepting undetected errors must be low
  - (iii) Flow control: detect and respond to lost data
    - Buffer overflow
    - Intermediate storage problems
    - …

Data Link: link between 2 directly connected DTEs (no inter-network)

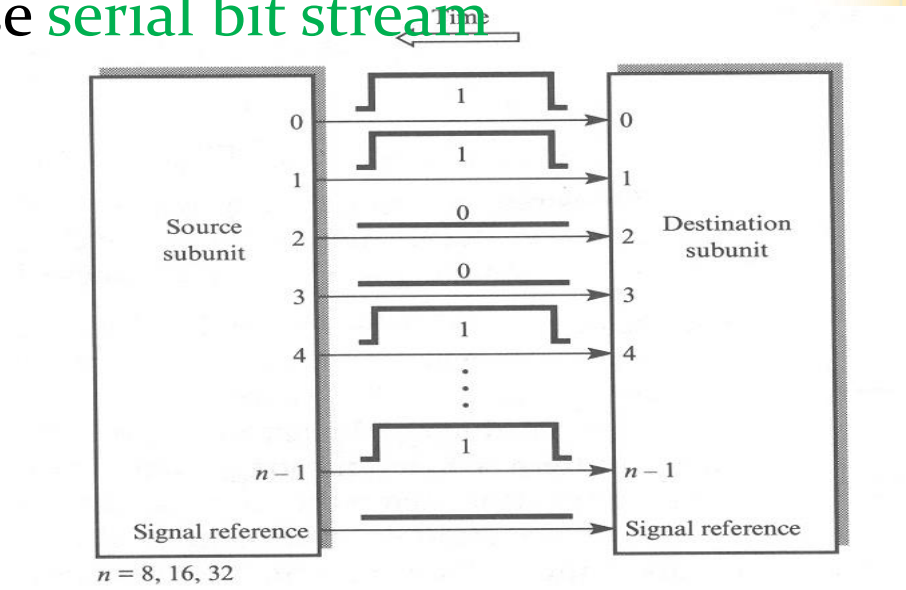
Data Transmission: techniques to achieve reliable transfer of information across bit-serial links between 2 DTEs

Data Link

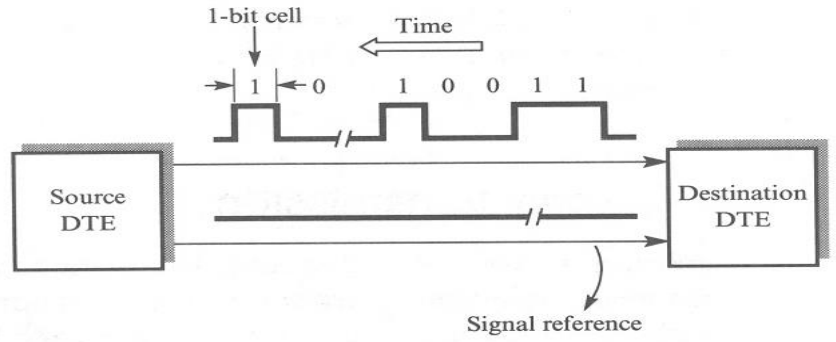
- Data Transmission Techniques
  - Encode: representing characters by bits
  - Decode: extracting characters from bit representations
  - Codeword: bit representation of characters
- EBCDIC: 8 bit code (used with IBM peripherals)
- ASCII: 7 bit code
- Types of characters
  - Printable characters (‘q’, ‘w’, …)
  - Control characters
    - Format control (LF, CR, DEL, ESC, …)
    - Information separators (FS, RS, file & record separator)
    - Transmission control character (SOH, STX, ETX, ACK, NAK, SYN)
- Internal representation of data: bits, bytes, words

Data Transmission

- Bit serial transmission
  - Inter-IC communications: usually parallel, short distance, low power
  - Inter-device communications: remote devices use serial bit stream
    - Cabling, cost, signal power, noise & interference
  - Bit representation:
    - Bipolar (serial)
    - Unipolar (parallel)



- Communications modes
  - Simplex: 1-way (pager)
  - Half duplex: 2 way alternate (push to talk)
  - Full duplex: 2 way simultaneous



Transmission Modes

- Data often transmitted between 2 DTEs in 8 bit units
- Receiving DTE receives signal levels which vary according to bit pattern
- Type of synchronization depends on clocking
  - Asynchronous transmission: local independent clocks
  - Synchronous transmission: global clock or synchronized local clocks
- Synchronization tasks for receiver to interpret signal
  1. Bit synchronization
     - Start of each bit cell period (clock) (to sample at mid-point)
  2. Byte synchronization
     - Start & end of each word (byte)
  3. Frame synchronization
     - Start & end of each block (frame)

Asynchronous Transmission

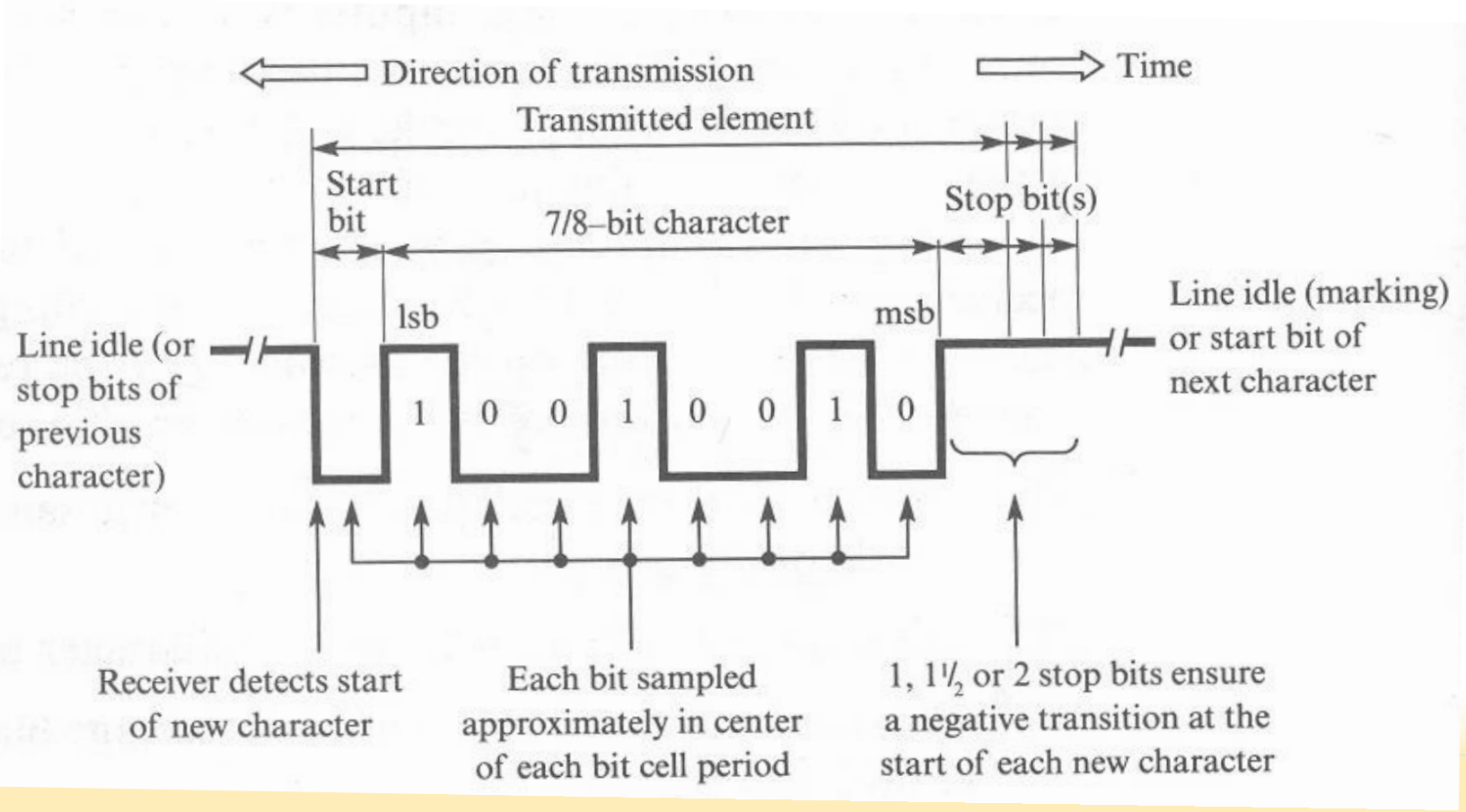
- Applications
  - Most useful for data with irregular arrival times (human actions)
  - Transmission line with long idle states (marking)

- Examples:
  - Terminal & keyboard I/O
  - Block character transfers between 2 computers

- Characterized by:
  - No direct clock information between receiver and transmitter
  - Low bit rates, long idle times
  - Relatively coarse and inefficient method
  - Coarse bit synchronization

Asynchronous Transmission

- Each byte treated independently for clock synchronization purposes
- Receiver synchronizes at the start of each byte
- Uses start bit & stop bit for each byte

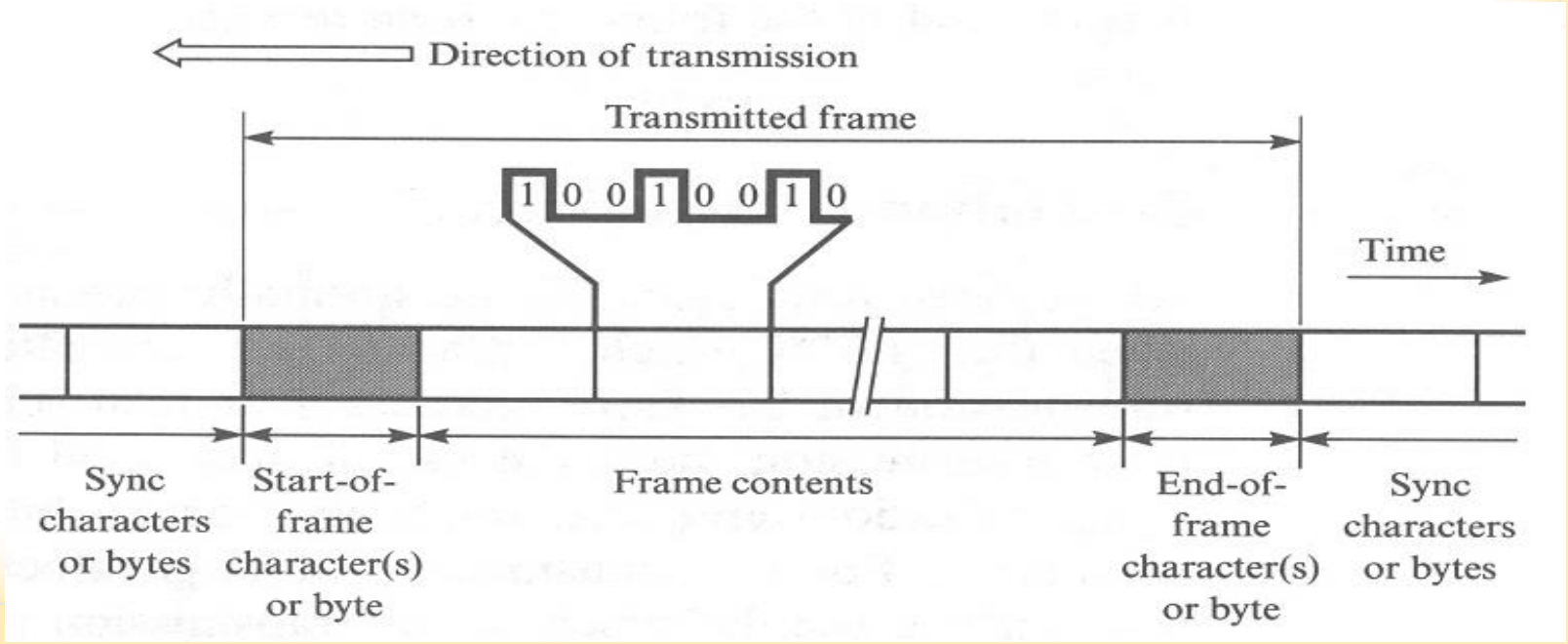


Asynchronous Transmission

- Bit synchronization
  - Stop & start bits have opposite polarity
    - At least 1 transition between each byte
    - Start bit: 1st 1 -> 0 transition after idle period
    - 1, 1.5 or 2 stop bits used to ensure end of byte
- Byte synchronization
  - Bytes are encapsulated in start bit & stop bit
  - Receiver has to re-synchronize at each byte
- Frame synchronization
  - Blocks are encapsulated by transmission control characters, SOF & EOF (start & end of frame)
  - Ensures receiver can distinguish new frame or end of frame after idle period

Synchronous Transmission

- Higher bit rates
- Unit of transmission = frame
- Continuous transmissions (frame transmitted as a contiguous bit stream)
- Receiver synchronizes each bit for duration of frame
- Contains overhead control bits and payload data



Synchronous Transmission

- Receiver synchronization
  - Bit synchronization
    - Bits are explicitly encoded in signal
  - Byte synchronization
    - Reserved bytes precedes each frame
    - Ensure receiver interprets bit stream on correct byte boundary
  - Frame synchronization
    - Frame is encapsulated between reserved bytes
    - Unique SOF & EOF identifiers to guarantee frame doesn't contain embedded SOF and EOF
  - Inter-frame synchronization (with inter-frame idle time)
    1. Each frame is preceded by synch bytes, so receiver can regain synchronization
    2. Synchronous idle bytes continuously transmitted to allow receiver to retain bit & byte synchronization

Ex. of Asynchronous vs. Synchronous Transmission

- How much overhead with 100 byte message?
  - Asynchronous transmission with 1 start bit, 2 stop bits
    - 100 data bytes + SOF byte + EOF byte
    - Efficiency $= \frac{100\times8}{(100+2)\times(8+1+2)} = \frac{800}{1122} \cong 71.30\ \%$
    - Overhead $= 1122 - 800 = 322$ bits
  - Synchronous transmission
    - 100 data bytes + 2 synch characters + SOF byte + EOF byte
    - Efficiency $= \frac{100\times8}{(100+2+2)\times8} = \frac{800}{832} \cong 96.15\ \%$
    - Overhead $= 832 - 800 = 32$ bits

Error Control

- Detect transmission errors (bit corruptions)
- Manage/Respond to detected errors
- Bit corruption: Signal level representing a bit is altered ---> incorrectly interpreted by receiver
- Error detection: depends on transmission techniques
  - Asynchronous transmission: detect errors in bytes
    - Parity bit attached to each byte
    - Each byte treated as separate entity
  - Synchronous transmission: detect errors over entire frame
    - Frame is basic unit of transmission
    - Probability of bit errors increases with frame size
    - Check sequence attached to frames ---> much more effective than parity bits
      - Transmitter: compute check bits, append to frame data, then send frame
      - Receiver: recompute check bits, compare with transmitted check bits

Flow Control

- How to manage lost data
- Received data
  - Must be temporarily stored (input buffer) at destination
  - On a network, may be intermediate storage on switches, routers, …
- Overflow
  - Amount of received data exceeds available storage space ---> data is lost
  - If DTEs operate at different rates, faster device can overflow slower device
- Lost data: data that is sent but never received by intended recipient
  - Can be from overflow
  - In a network, corruption of control information can result in data being lost

Data Link Protocols

- Protocol:
  - Set of rules adhered to by all communicating entities to ensure successful exchange of information across serial data link
- Components of data link protocol
  - Flow control
  - Error control
  - Data format: bits/element, signal encoding
  - Type & order of message transmissions for reliable information transfer
- Example: connection set-up messages that ensure both DTEs are ready