Computer Networks
COL 334/672

Bits, Frames, and Fixes

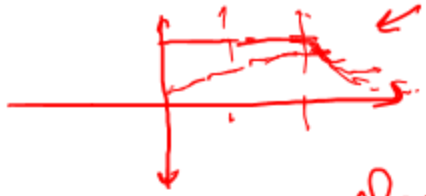
Slides adapted from KR

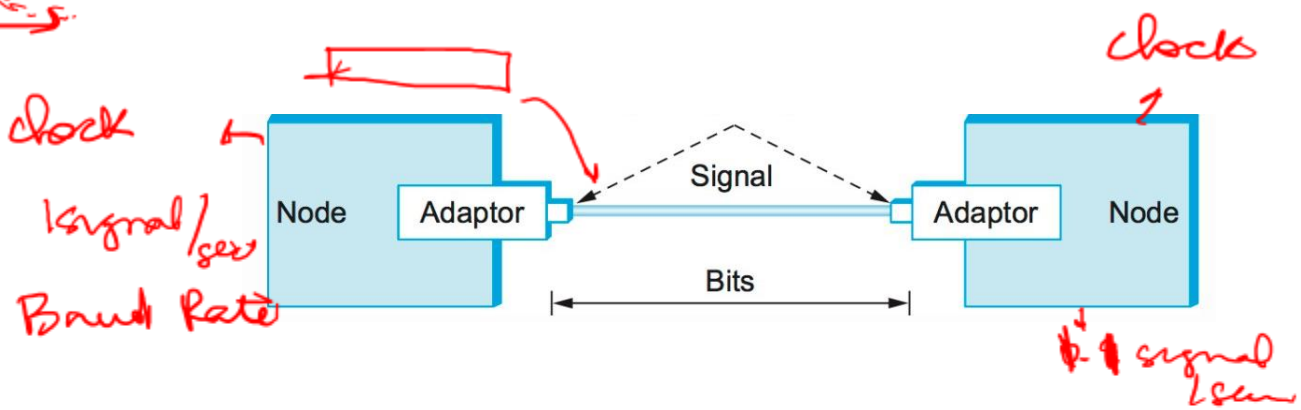
Sem 1, 2024-25

Link Layer: Services

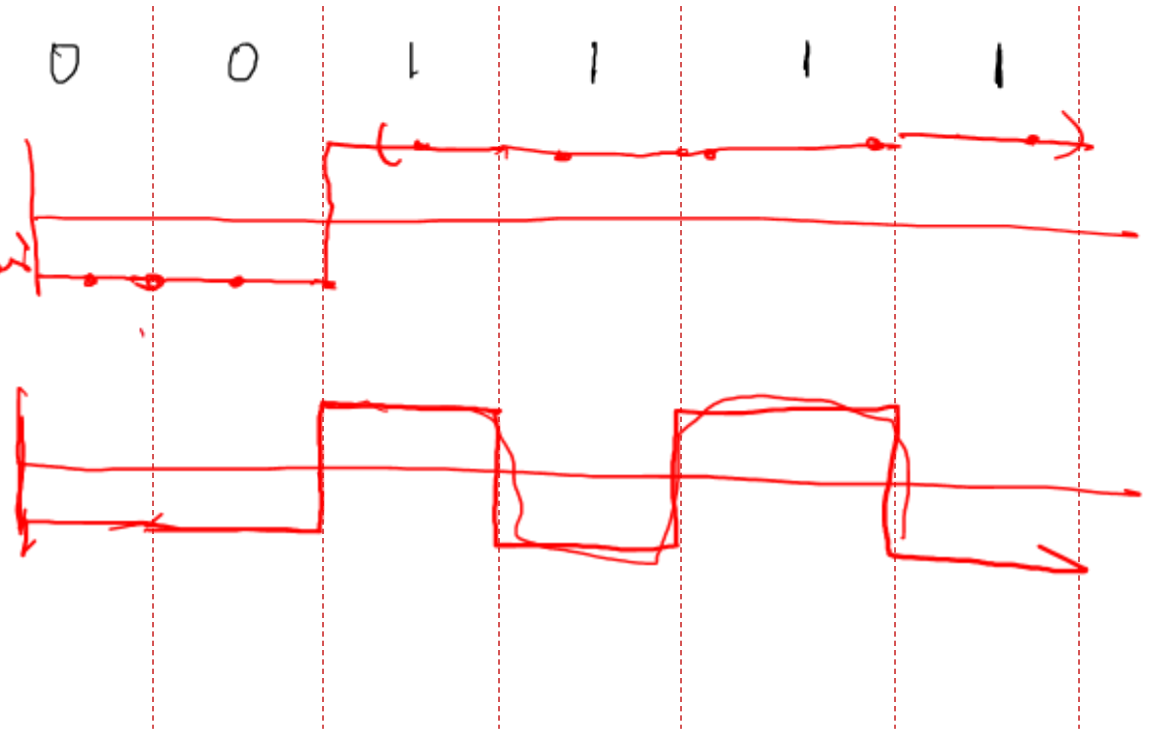
- **Encoding**
- Framing
- Error detection
- Addressing
- Link access (Multiple Access control)

Encoding





- Convert bits to signals
- Various encoding schemes
- Non-return-to-zero (NRZ): 1-> High voltage, 0 -> Low voltage
  - baseline wander,
  - clock recovery
- NRZ-Inverted (NRZI): 1 -> Transition from current signal, 0 -> stay at the same signal
  - consecutive 0s is an issue



Encoding Scheme

Modulation coding Scheme (MCS) → 64QAM
16QAM

Manchester coding

:

XOR the signals with clock cycle

0 → low to high

1 → high to low

Bitrate is ½ baud rate

0 → 01

1 → 10

1 bits in 2 signals : 50%

n bits in n+k signals

↳ 4B/5B (Manchester + NRZI)

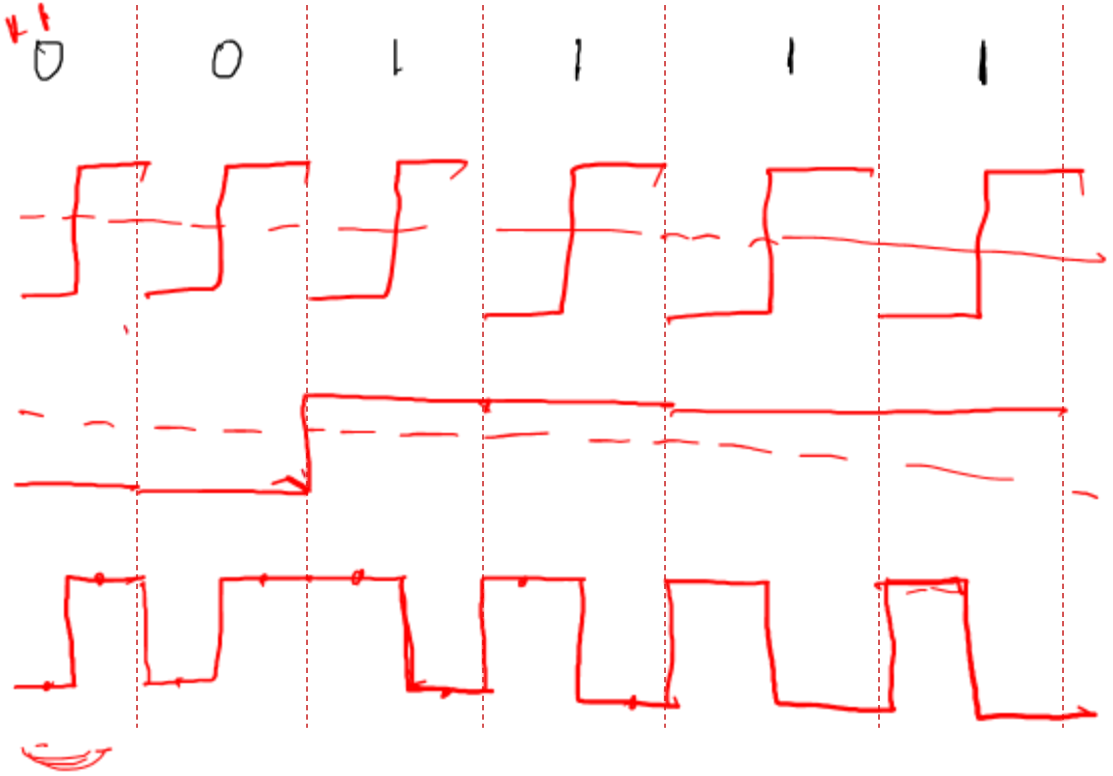
① 4 levels → [low, mild low, mild high, high]

00 → low

01 → mlow

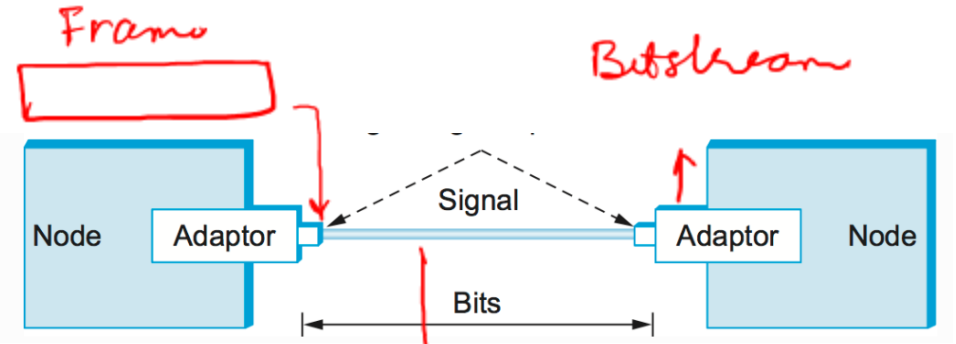
10 → mhigh

11



Framing

- Sender: Encapsulate datagram into frame
- Receiver: Assemble bitstream into frames
- ***Challenge: How to detect frame boundaries?***



Frame Boundary Detection

- Including number of bytes in the header
  - Can lead to framing errors in case of bit errors

HW: Find out the framing mechanisms used by Ethernet

- Sentinel approach
  - Use special token to denote start and end of frame or sent (e.g., 01111110)
  - What happens when the token appears in the payload?
  - Use esc character or bit stuffing → Bit-oriented protocols
  - Used in High-level data link control (HDLC) protocol

Byte-oriented protocols

Bit stuffing algorithm

- **Sender side**
  - If see 5 consecutive 1s then insert a zero after them

Flag: 01111110

Data bits: ..111110...

..1111101...

- **Receiver side**
  - If see 5 consecutive 1s then remove the stuffed bit 0 following them

Receiver: ..1111101... → ..111111..

...111110 → ..111110...

If bit errors, similar to byte-oriented protocol