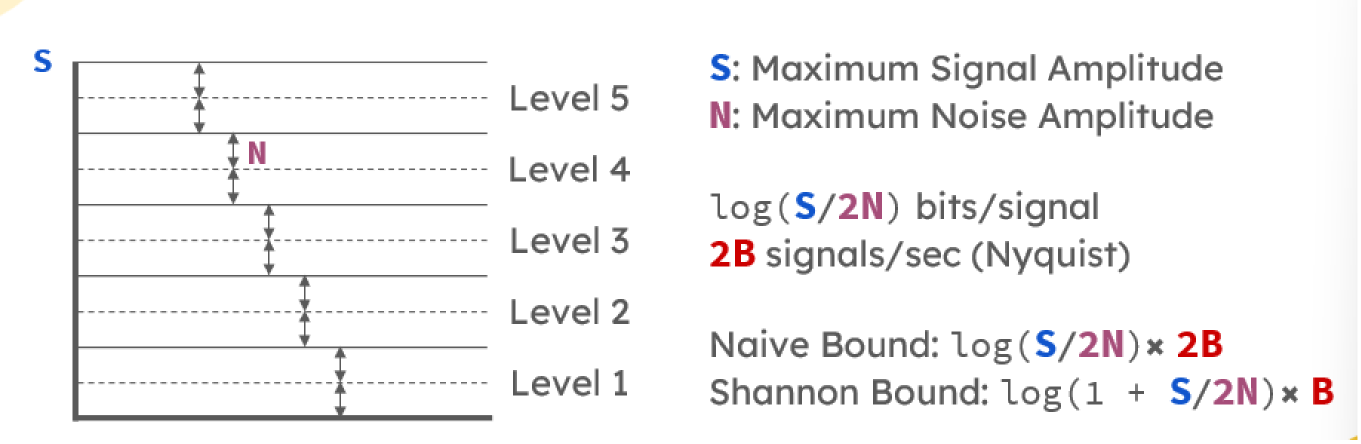
# CS 118 Midterm Cheat Sheet

## Physical Layer

**Layer Violation** - layers may only access/view their own headers/layer content
easy way to identify: if a header of a lower layer is changed, it should not impact current layer service - if it does there is a violation
**ECN** - Explicit Congestion Notification - added to IP & TCP to inform source abt congestion and to decrease sent packet rate
**Bandwidth/Frequency (B)** = $F = 1/T$ hz
where $T$ is period an $F$ is frequency
**Intersymbol Interference (ISI)** - interference between the lag of the previous symbol and the next symbol
**Nyquist Limit** = $2B$ bits/s
you can bypass Nyquist limit by sending on different phases or frequency
**Baud Rate** = $\log_2 L \times 2B$ bits/s
where $L$ is the number of signal amplitudes
**Shannon Bound** = $B \log_2(1 + S/2N)$



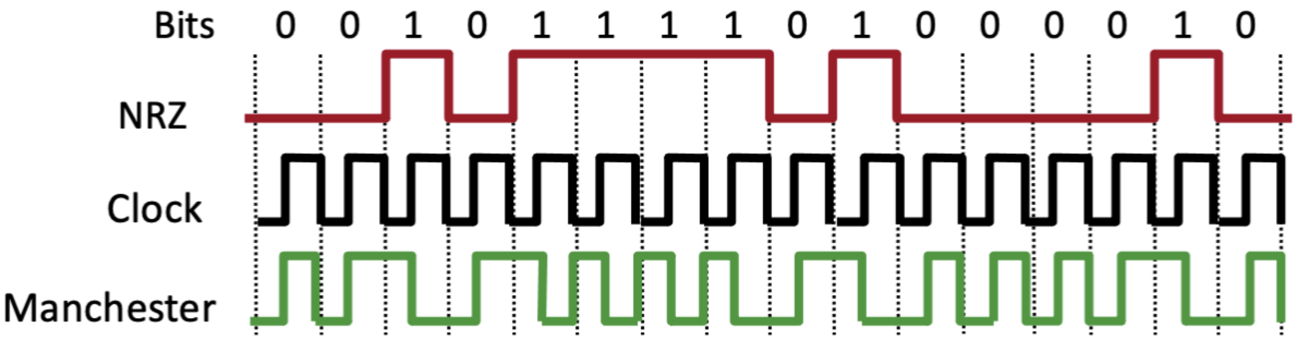
S: Maximum Signal Amplitude
N: Maximum Noise Amplitude

log(S/2N) bits/signal
2B signals/sec (Nyquist)

Naive Bound: log(S/2N) × 2B
Shannon Bound: log(1 + S/2N) × B

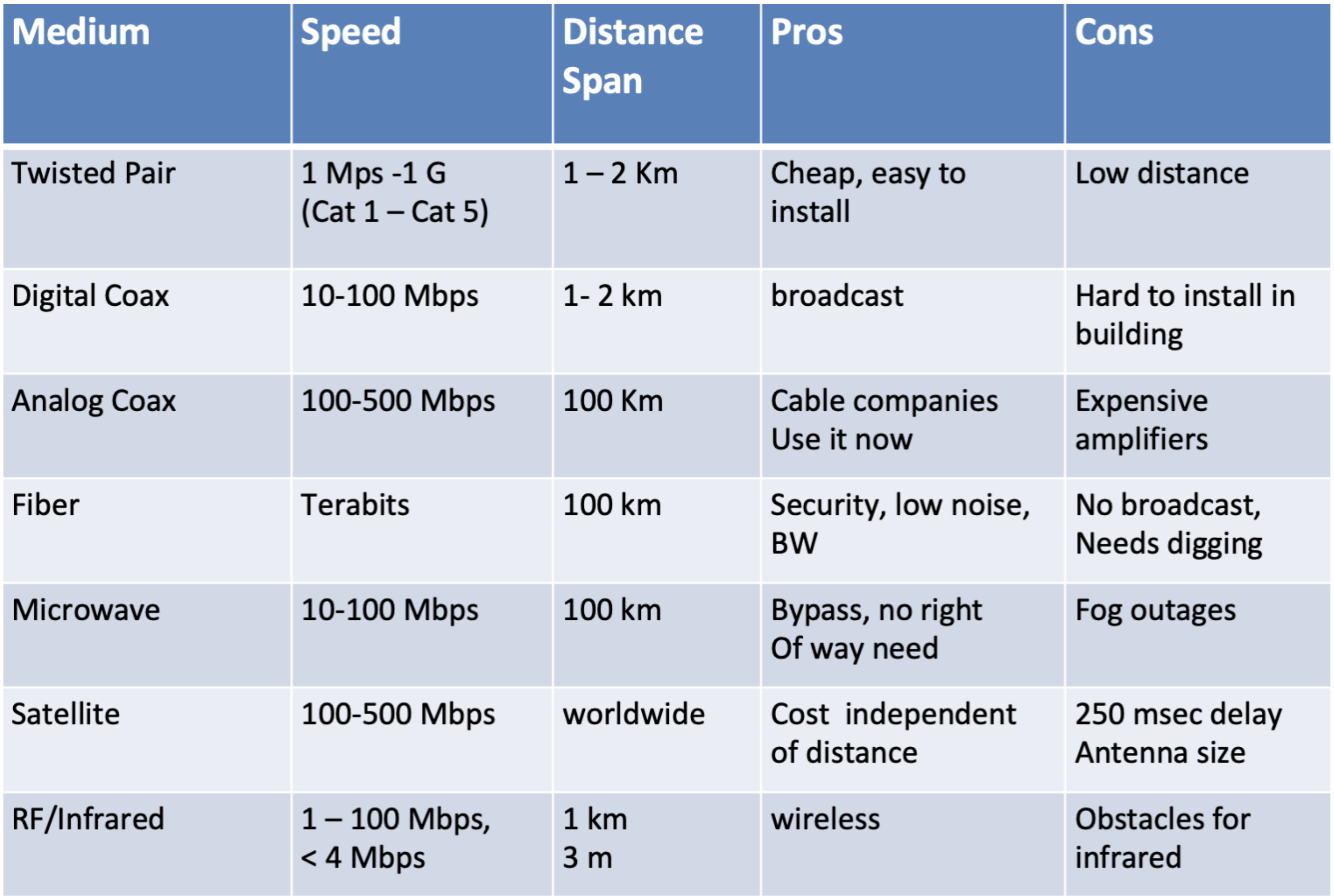
**Nyquist-Shannon Sampling Thm** - Anti-aliasing iff $f_s > 2f_{\max}$
where $f_s$ is sampling freq and $f_{\max}$ is og max freq
**Synch. Clock Recovery** - signals require preamble w/ transitions to reduce receiver clock overhead when sampling synchronously

**Manchester Encoding** - encodes bits to transitions at mid bit width: 1:hi→lo, 0:lo→hi
con: 50% efficient - encodes only half bit per transition



**Alternate Mark Inversion (AMI) Encoding** - encodes bits to alternating voltage levels: 0:0V, 1:±V. Each bit alternates positive and negative voltage.
e.g. `11100111` → `+-+00+-+`
con: issues with long seq of 0s
**4-5 Encoding** - encodes 4 bit seq to 5 bit seq w/ transisition
e.g., `1111` → `00001`, mitigates long preamble
con: introduces new overhead for every 4 bit pattern
**Broadband Encoding** - Frequency Shift Keying (FSK), Amplitude Shift Keying (ASK), Phase Shift Keying (PSK)
not limited to energy levels like baseband encoding above
**Signal Demux** - Time/Freq/Phase Division Mux (T/F/PDM)
e.g., tv channels - signals muxed by frequency of signal
**Twisted Pair Cable** - low bandwidth, cheap → Cat 5 twisted pair higher quality
**Coax Cable** - high bandwidth, og ether, too clunky replaced by Cat5
**Fiber Optic Cable** - huge bandwidth, unidirectional, but chromatic and modal (bounce) dispersion, expensive, multichannel via multicolor but expensive with prism to demux color channels
**Wireless 802.11b** - broadband, requires spectrum allocation,

possibly satellite, radio large passes through objects



| Medium | Speed | Distance Span | Pros | Cons |
|---|---|---|---|---|
| Twisted Pair | 1 Mps -1 G<br>(Cat 1 – Cat 5) | 1 – 2 Km | Cheap, easy to install | Low distance |
| Digital Coax | 10-100 Mbps | 1- 2 km | broadcast | Hard to install in building |
| Analog Coax | 100-500 Mbps | 100 Km | Cable companies<br>Use it now | Expensive amplifiers |
| Fiber | Terabits | 100 km | Security, low noise, BW | No broadcast,<br>Needs digging |
| Microwave | 10-100 Mbps | 100 km | Bypass, no right<br>Of way need | Fog outages |
| Satellite | 100-500 Mbps | worldwide | Cost independent of distance | 250 msec delay<br>Antenna size |
| RF/Infrared | 1 – 100 Mbps,<br>< 4 Mbps | 1 km<br>3 m | wireless | Obstacles for infrared |

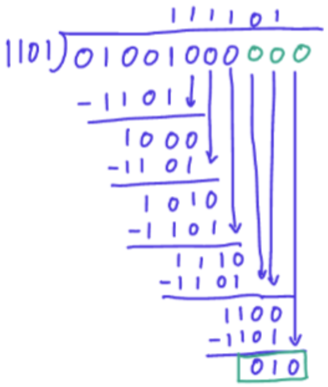
## Data Link Layer

**Flags** - wrap datagrams to fragment into frames, signify start and end
**HDLC** - bit stuffing for false flags, no escapes
**PPP (Ethernet)** - byte stuffing, with escapes
**Stuffing Overhead** - #stuffed bits / #og bits
**Stuffing Efficiency** - Probability of stuff = #flags / #bit combs/patterns
**CRC32 Mod2 Div** - shift left by `len(gen)-1` then long divide generator, xor only for leading 1s, if leading 0 → move right until leading 1

Sender:

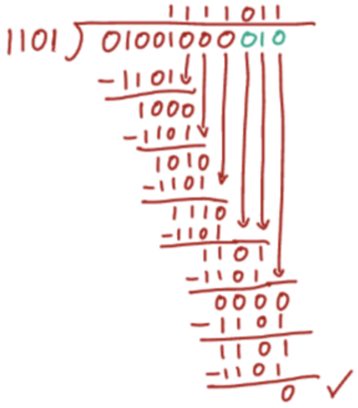
'H' → 01001000     Gen → 1101

raw. data: 01001000000



Receiver

received → 01001000010   Gen → 1101



CRC-16: $X^{16} + X^{15} + X^{2} + 1 = 11000000000000101$

We skip proofs of these properties this quarter but they are in your notes, Not required for HWs and tests.

Odd bit errors: can handle but not a big deal as parity can handle with using just 1 bit. 1

Two bit errors specially designed CRCs can do this. Beats parity!

Burst errors: CRC-32 can catch any 32 bit burst error for sure. Further it can catch larger burst errors with very high probability: (1 - 1/ $2^{32}$ )

Summary: So the big deal is that it can for sure catch up to 3 bit errors. and can detect **any** error with very high probability. Like a hash function with with some deterministic guarantees

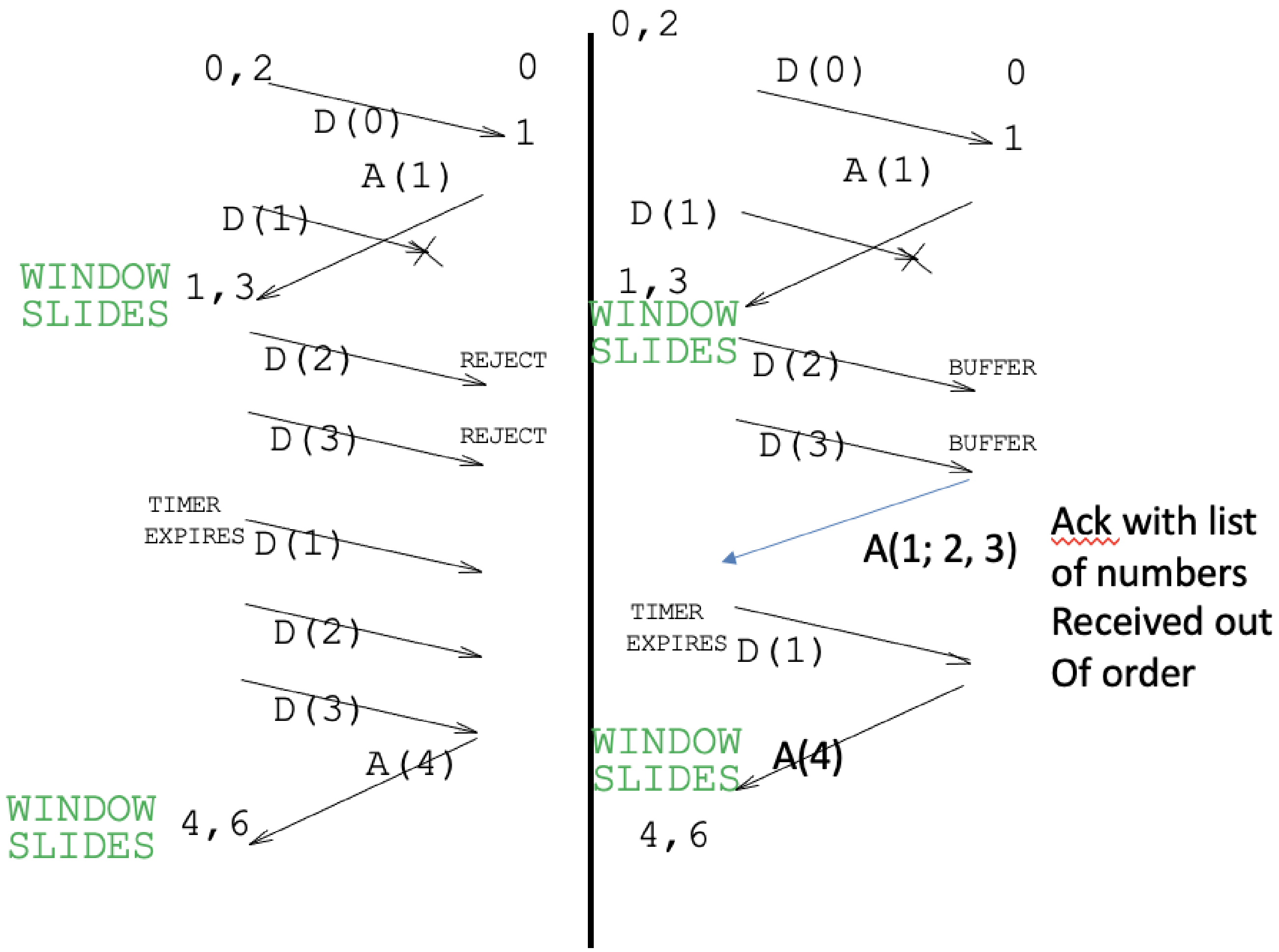
**Band Invariance** - sender and receiver will alway be within $x + 1$ packet ids of each other.
Receiver state is id of packet waiting to receive, ack is id of receiver state
Sender state updates with ack

```
 ----------        5 4 3 2 1 0         ---------
 |  s = 3 | ---------------------------------|  r = 3 |
 |        |----------------------------------|        |
  ---------                                    --------
```

**Throughput (bits/s)** - jobs/s (usually round trip)
**Latency (s)** - worst case time to complete 1 job
**1-way propagation delay (s)** - time for transmitted bit to cross link
**Transmission Rate (bps)** - bit rate over link bits/s
**Pipe Size aka Bandwidth Delay Product (bits)** = Transmission Rate × Round-trip propagation delay
**Sliding Window Ack Protocol**
only fifo packets with ordering, fails on UDP
mod for packet ids depends on window size W (max number of packets sent in sequence)
alternating bit - mod 2
go back W - mod W+1
selective reject - mod 2W

## GO BACK 3

## SELECTIVE REJECT

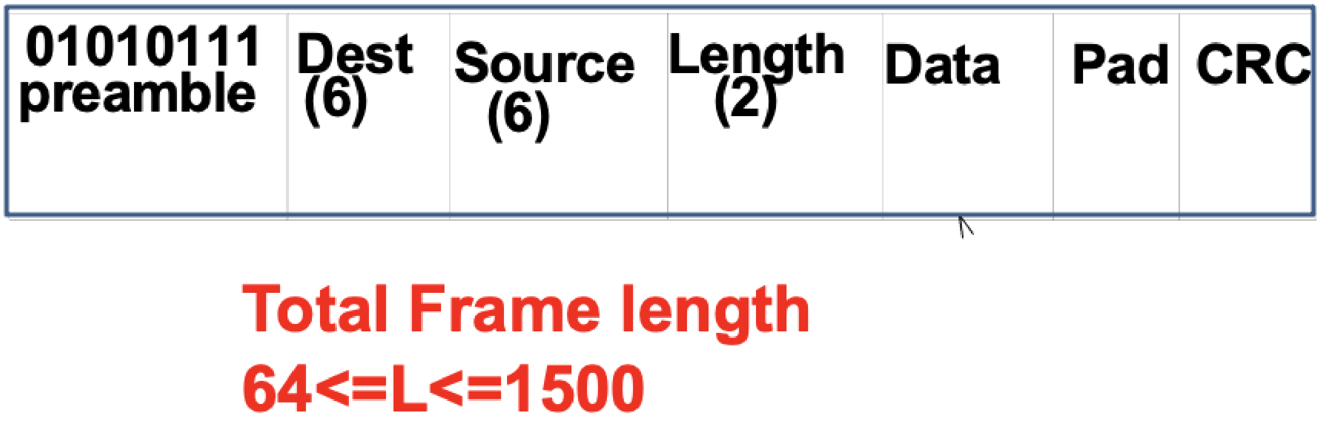


Ack with list of numbers Received out Of order

**Restart Signal** - requires ids to mitigate multi restart usually uses max frame time timer then restart - issue is must wait for timer so longer time to reboot

## LAN

**Ethernet** - multi-access many-to-many, demux via MAC addr



| 01010111<br>preamble | Dest<br>(6) | Source<br>(6) | Length<br>(2) | Data | Pad | CRC |
|---|---|---|---|---|---|---|

Total Frame length
64<=L<=1500

**Strict Multiplexing (B/N)** - allocate static bandwidth via TDM/FDM

**Stat. Mux.(B/x s.t. x < N)** - allocate bandwidth based on traffic

allows clients to use others' bandwidth when low traffic

**CSMA/CD** - Collision detection via carrier sense - stations must listen and detect collisions occurring at their station and propagate info to all stations

Ethernet uses min frame size of 64 bytes = Pipe size = Trans. Rate (10 Mbps) $\times$ RTPD (51.2$\mu s$) = 512 bits

Limits cable length if link has higher transmission rate

Collision detection via voltage, if high is 1, avg volt for 0 or 1 is 0.5V, collision would cause avg volt of 1V

Jam bits during collision to extend collision to be detected by other stations

**Binary Exponential Backoff** - wait longer time for more collisions

Choose wait time after $k$ collisions from $2^k - 1$ time sots

e.g., after 1st, choose 0 or 1 wait, then 2nd choose 0 to 3 wait

**Hubs** - Single point of connection for all nodes on ether, requires CD

## Wireless 802.11b

**Multichannel** - 12 allocated channels, 3 orthogonal channels at a time

Stations can be on orthogonal channels so CD wont be detected if on diff channel

**RTS/CTS (MACA)** - before node A transmits, send couple bytes called Request-to-Send on all channels, node B hears and calls Clear-to-Send broadcast → node C hears and defers

## Bridges/Switches

**MAC Addr** - 6 bytes (48 bits)

Unique to device, unique to terminal (rec/send)

first 3 bytes for vendor, last 3 host

MSB leading 1 → multicast

**IPv4** - 4 bytes (32 bits)

allocated per network via DHCP

Accessible IPs mapped via DNS

leading 1 (or `1110`) signifies multicast

**Switches** - 1-to-many, point to point, buffer frames if link is busy

Entries in DB by looking at Src addr (up)

Flood down if dest not known

No loops → tree topology

Timer for buffering, timer expires → flood all buffered frames

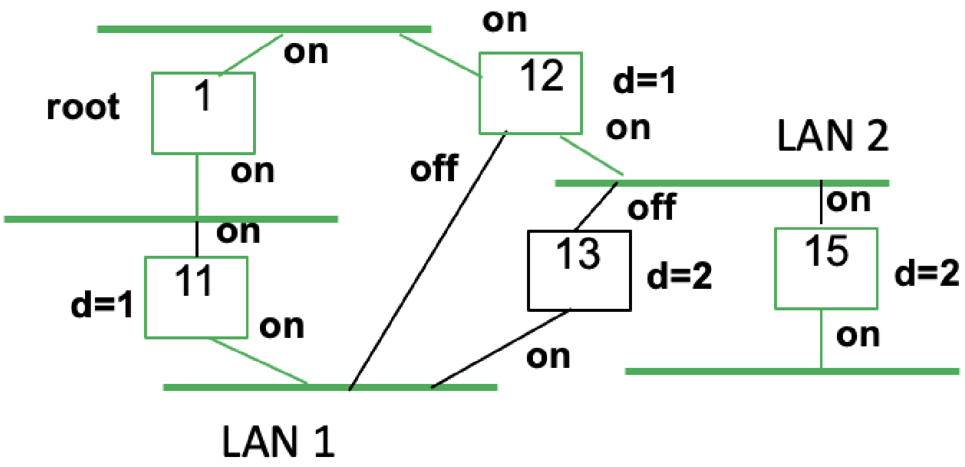
**Switch Transparency** - bridges must be transparent to nodes, must appear as simple ether/cable

**Promiscuous Receive** - Switches buffer from all stations regardless of src

**Flood** - Forward to all stations on line, picked up by correct MAC

**Filtering** - filter packets by ether header for forward or buffer

**Bridge Spanning Tree Algo** - bridge ids, drop longer links to same LANs



- Root is Min ID node (in this case Bridge 1)
- Other bridges finds Min port, port through which it has shortest path to root (parent), For 11 it is upper port.
- Each bridge also finds the ports for which this bridge is on the shortest path between root and corresponding LAN: Designated Ports. For example, 11 and 12 have d= 2 for LAN 1, so we pick shorter ID as tiebreaker, Bridge 11 is designated bridge for LAN 1, 12 for LAN 2
- Each bridge turns ON Min port and all Designated Ports. ON,OFF are software states: always receive hello and management messages on all ports. Drop data packets to/from OFF port.