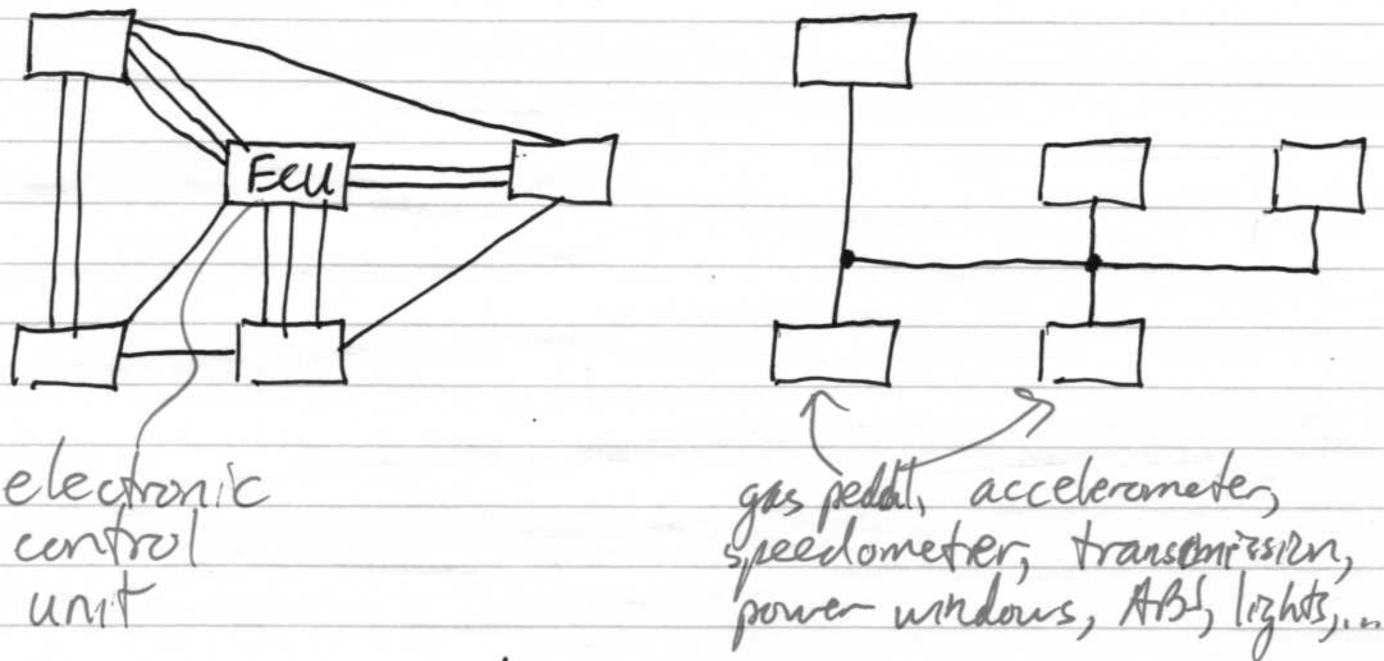
Networks

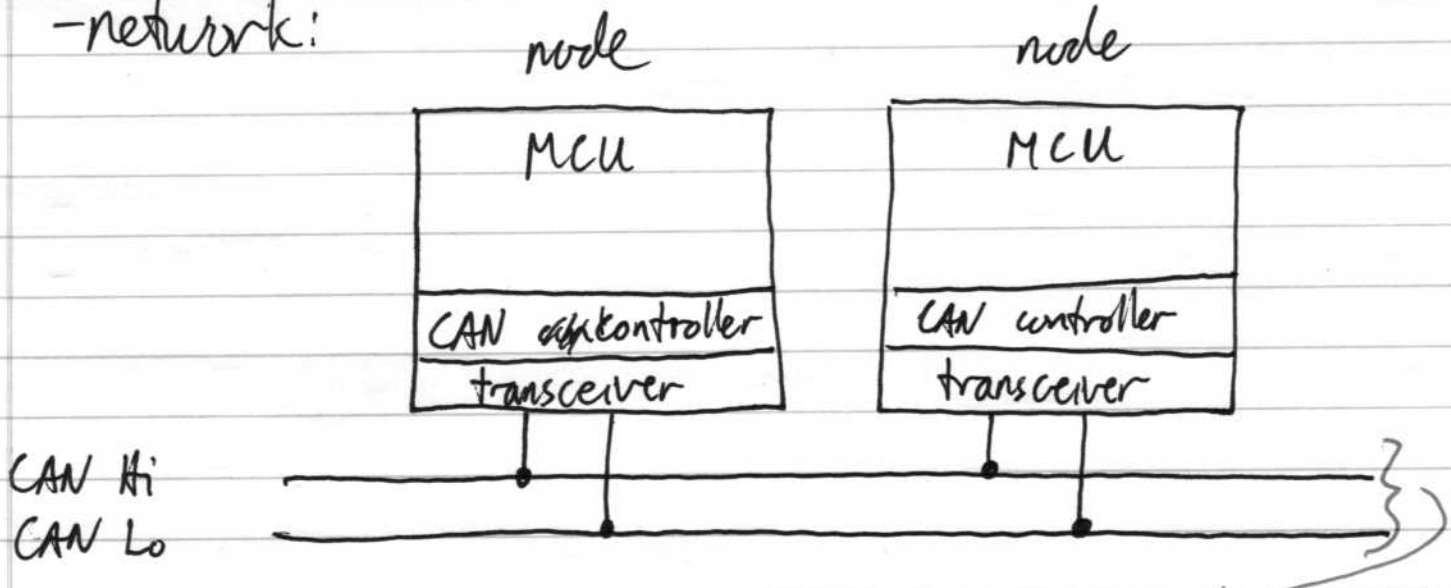
1) Controller Area Network (CAN) [12.6.2]

- developed by Bosch in 1985
- started as an in-vehicle network to replace numerous point-to-point connections

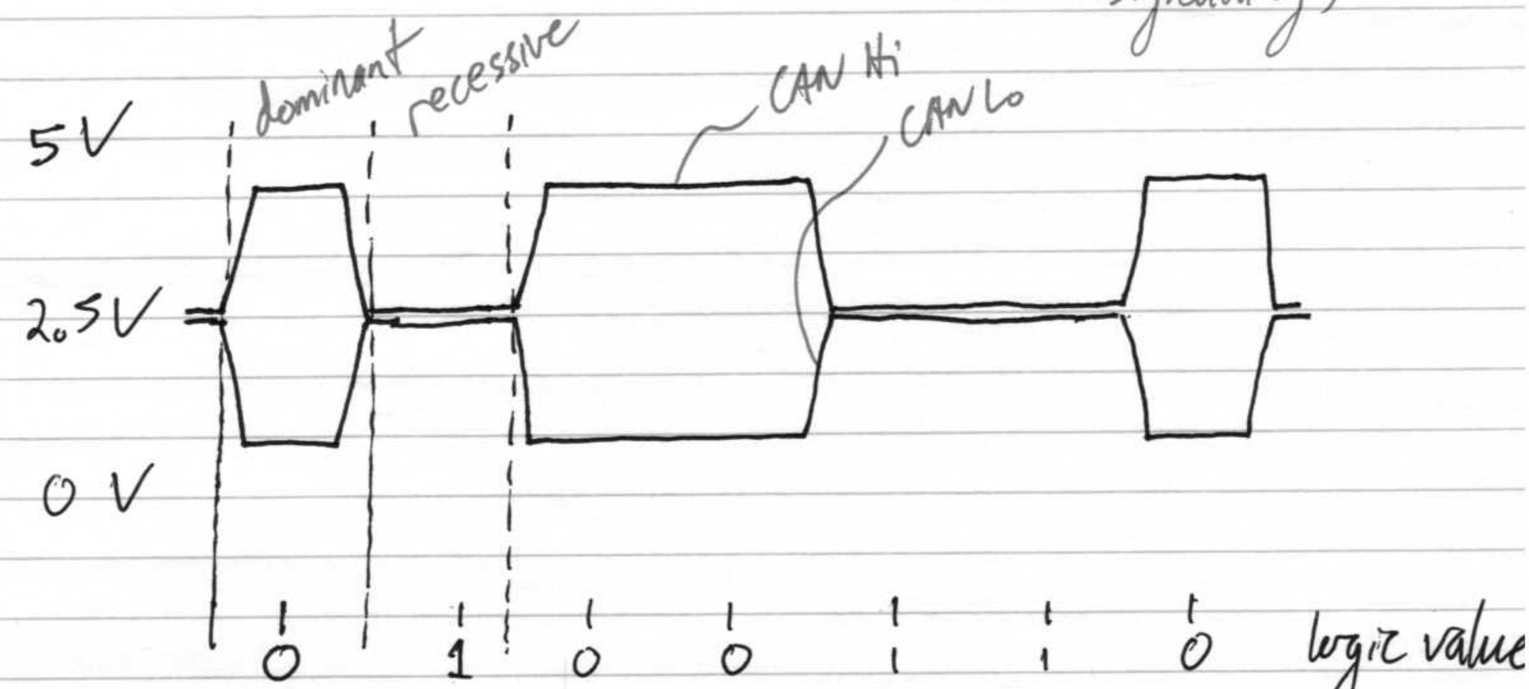


- has been adopted in industrial automation, medical equipment, aerospace, ...
- it is a high-integrity serial broadcast bus with speeds up to 1 Mbps (Megabits per second)

-network:



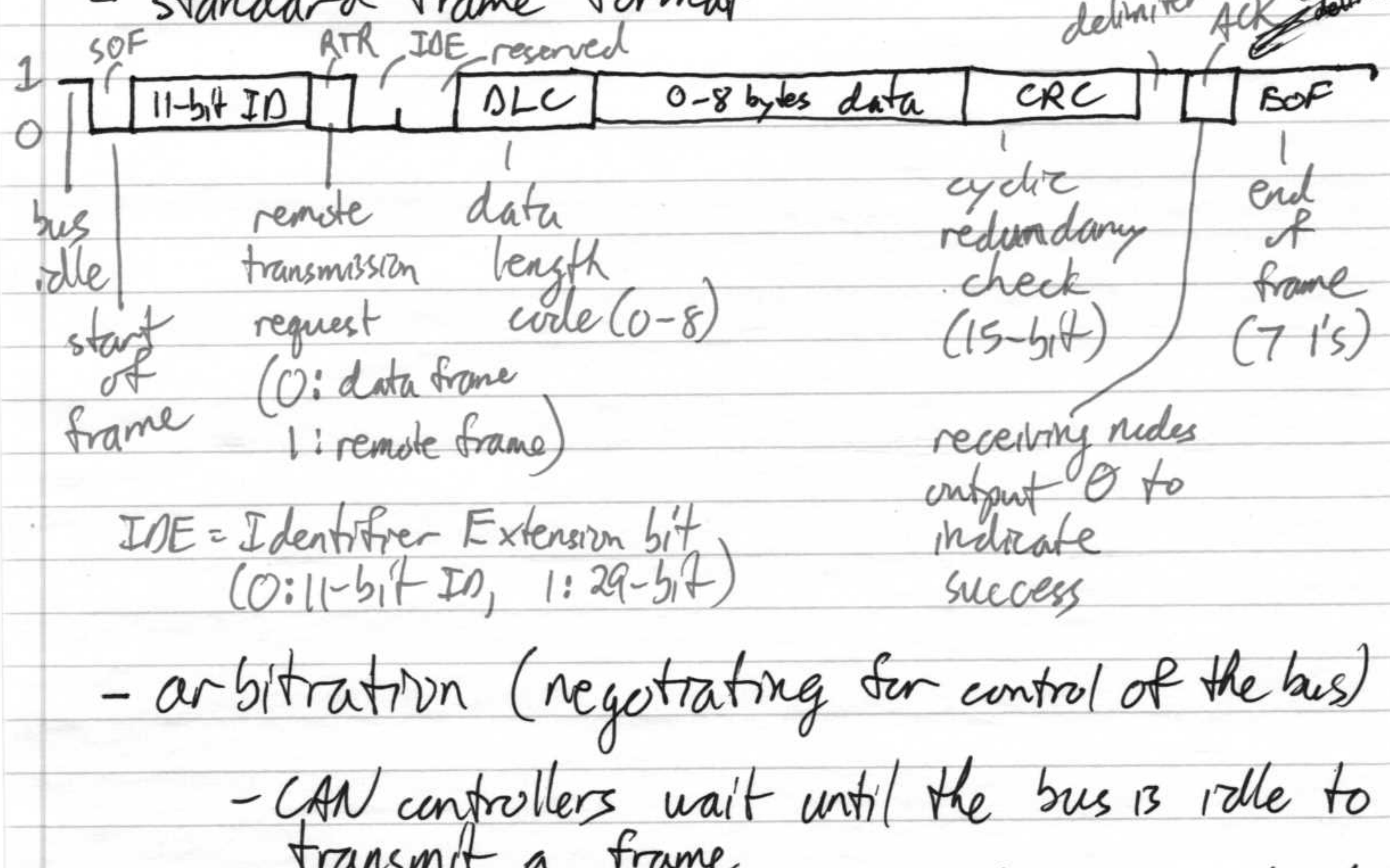
uses two wires for one bit (differential signalling)



- messages are sent in frames that include the message identifier (msgID)
  - there are no addresses
  - the msgID defines the priority (lower ID ⇒ higher priority)

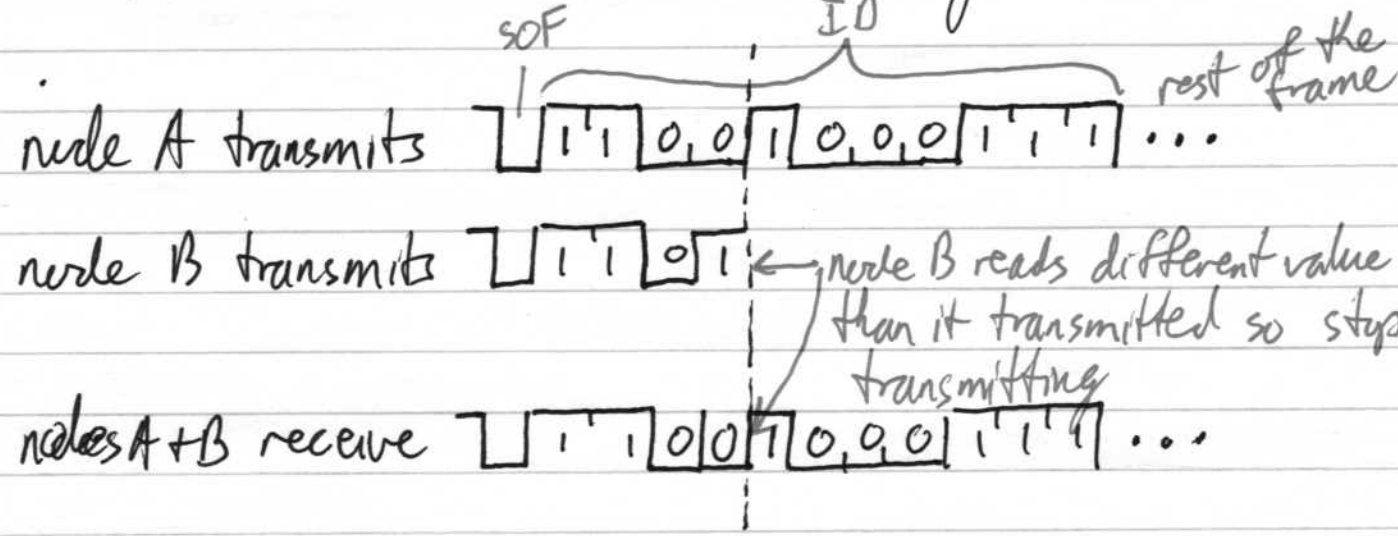
- 4 frame types: data, remote (request for data), error, overload

- standard frame format



- arbitration (negotiating for control of the bus)
  - CAN controllers wait until the bus is idle to transmit a frame
  - in the case that multiple controllers start at the same time, the value read from the bus is the logical AND of the values transmitted

e.g. node A sends message with ID = 11001000111
node B sends message with ID = 11011000111
node A and node B start transmitting at the same time



- node A has the lower message ID and hence the higher priority, so it won the arbitration

- CAN is suitable for real-time because:
  1. in the case of arbitration, the high-priority message experiences zero delay
  2. if a message becomes ready during transmission of another frame, it will have bounded delay

- schedule analysis similar to RM has been developed for CAN messages

- bit stuffing
  - nodes synchronize their CAN clocks on the falling edge of the SOF bit
  - nodes can get out of sync if there is a long sequence with no transitions in bit value (polarity)
  - therefore a bit of opposing polarity is inserted after every sequence of 5 bits of like polarity

e.g. 1 0 0 0 0 0 ^ 0 1 1 1 1 1 ^ 0 1

becomes

1 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1

- when a node detects an error it outputs the error flag = 0 0 0 0 0 0
  - in this case the transmitter will attempt to resend the frame

- CAN error sources:
  - transmitter reads different polarity bit than it transmitted
  - bit stuffing (6 bits of same polarity)
  - frame check (missing deliminator bits, bad EOF)
  - ACK check (no ACK received)
  - CRC check (checksum error)

- the EOF is 7 recessive ("1") bits so an out-of-sync node can detect 6 recessive bits as a bit-stuffing error and start the error flag on the 7th bit of the EOF

1.1) Cyclic Redundancy Check [13.2.3]

- an n-bit CRC can detect any single error burst of ≤n bits (and some longer bursts)
- it is based on polynomial division in a Galois Field (GF)
- easily implemented in hardware with XOR logic

e.g. 3-bit CRC on an 8-bit message
- "divisor" = 1011 (represents a polynomial of degree 3 e.g. $x^3 + x + 1$)
- message = 10011001

```
                 message            padding
            1 0 0 1 1 0 0 1      0 0 0
divisor   ⊕ 1 0 1 1
            0 0 1 0 1 0
              ⊕ 1 0 1 1
                0 0 0 1 0 1 0
                    ⊕ 1 0 1 1
                      0 0 0 1 0 0
                            check value
```

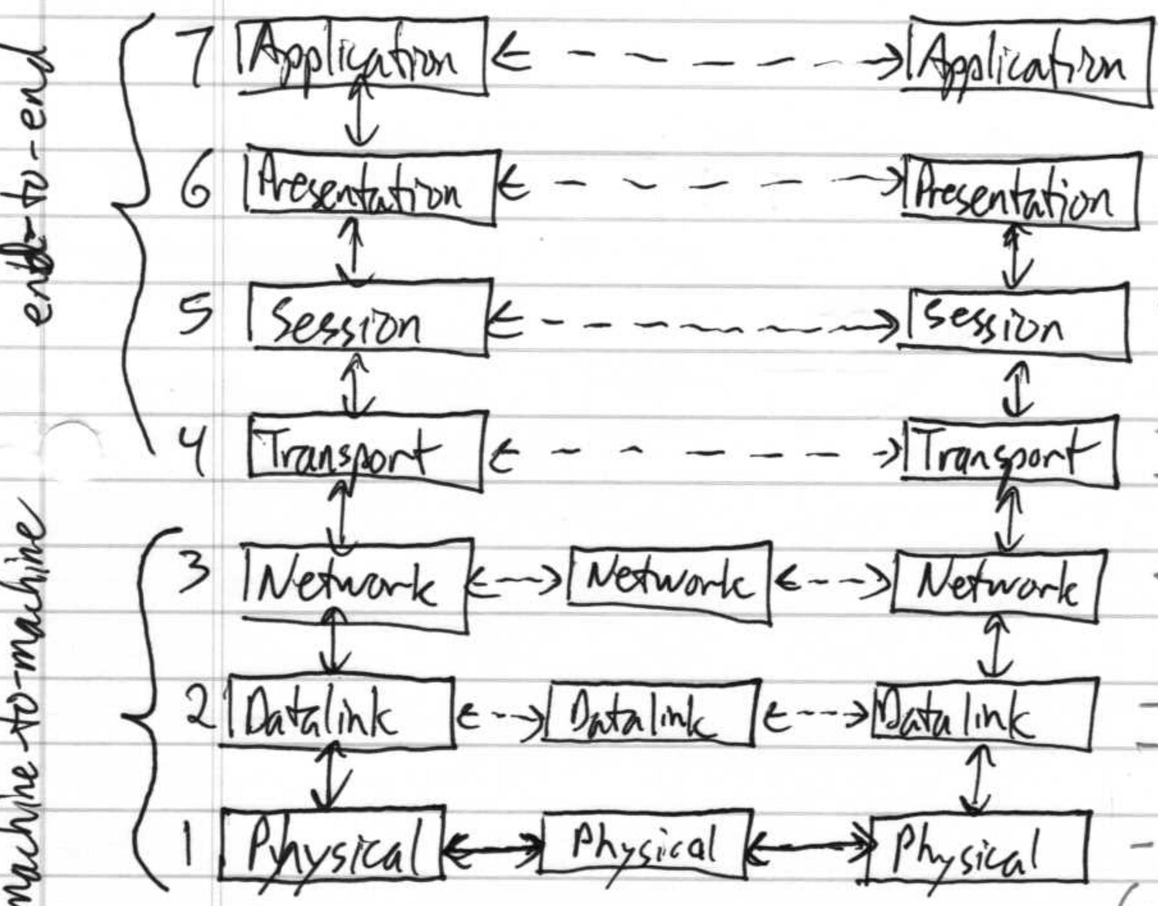
- the transmitter sends the message plus the CRC check value
- the CRC calculation is two-way

- the receiver computes the CRC to verify that it gets "000" back

```
   message    check value
  10011001     100
⊕1011 xxxx     xxx
  001010
  ⊕1011
    000101 1
     ⊕1011
      000000
```

2) OSI Model [Tanenbaum]

- published in 1984
- Open Systems Interconnection (Reference) Model
- describes 7 abstraction layers of networks



- protocol (ssh, http, ftp, ...)
- encoding (UTF-8)
- encryption (SSL)
- dialogue control (sync. checkpoints)
- create connections
- orders messages
- routing
- framing
- error detection
- how bits are sent (elec., mech. specs)

eg.

physical	datalink layer
RS232 (serial comm.)	PPP (point-to-point protocol)
CAN transceiver	CAN controller
100 BASE-T (twisted pair)	ethernet

Advanced Research Projects Agency

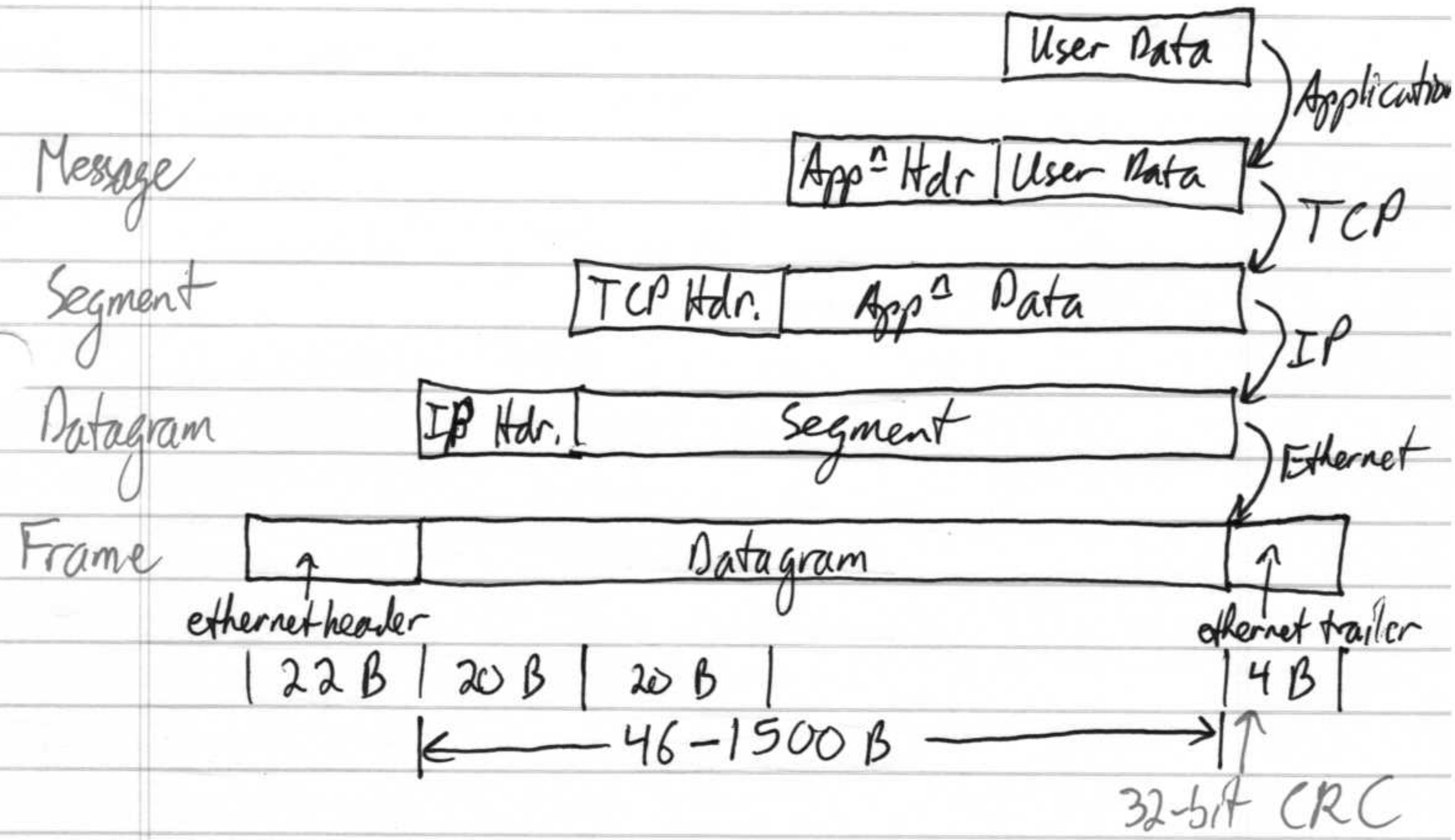
3) Internet [12.2.9]

- grew from the ARPAnet which connected research networks in industry and universities
- ARPAnet predates the OSI model by ~10 years
- uses TCP/IP (Transmission Control Protocol/ Internet Protocol)

TCP/IP

Application Presentation	Application	http, smtp, ssh, ...
Session Transport	Transport	TCP/UDP
Network	Internet	IP
Datalink Physical	Medium Access Control / Physical Layer	

- each layer adds a header to the data being sent and strips it off of data received
- some layers subdivide data into smaller packets to pass to lower layers and reassembles packets received from lower layers



- IP header includes a 32-bit (IPv4) or 128-bit (IPv6) address
  - Domain Name Server (DNS) hosts translate domain names to IP addresses
  - Internet Corporation for Assigned Names and Numbers (ICANN) oversees global IP address allocation and manages the top-level DNS hosts
  - Internel layer routes datagrams hop by hop
  - unreliable: there can be loss, duplication, delay, out-of-order

- TCP
  - is an end-to-end protocol
  - creates a virtual "circuit" between hosts (e.g. like establishing a telephone connection)
  - uses handshaking (via ACK messages) and sequence numbers to ensure <u>all data</u> gets to the receiving host <u>in order</u>
  - data is treated as a continuous stream of bytes delivered in segments of at most 64 KiB (often 1460 B to fit in an Ethernet frame with IP header (20 B) and TCP header (20 B)