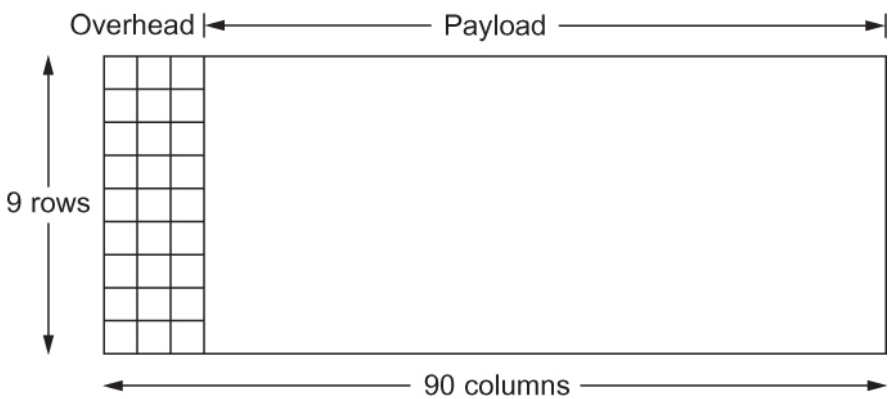
Framing and error detection



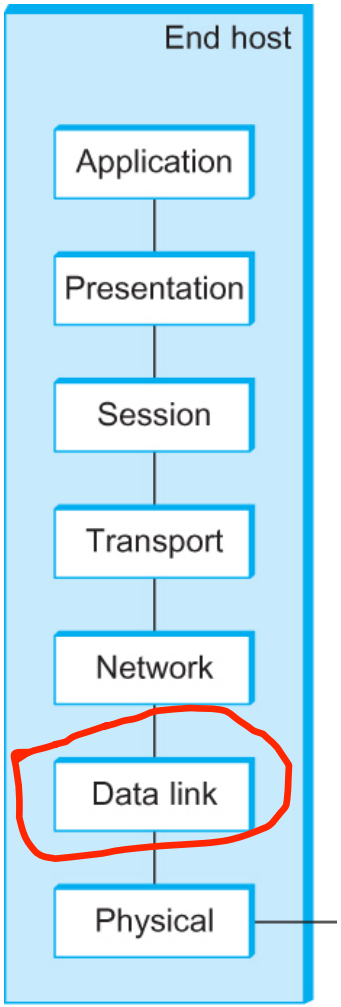

```
1 1 0 1 | 1 0 0 1 1 0 1 0 1 0 1
          1 1 0 1
            1 0 0 1
            1 1 0 1
              1 0 0 0
              1 1 0 1
                1 0 1 1
                1 1 0 1
                  1 1 0 0
                  1 1 0 1
                      1 1 0 1
                      1 1 0 1
                            0
```

Overview

- Chapter 2:
  1. How do we transmit bits from one place to another?
  2. How do we aggregate bits into frames?
  3. How do we detect errors?
  4. How do we make links appear reliable?
  5. How do we share links between multiple hosts?

Overview

1. How do we aggregate bits into frames?
   - Determine bits that constitutes a frame
     - Byte-oriented protocols
     - Bit-oriented protocols
     - Clock-based protocols
2. How do we detect errors?
   - Parity
   - Checksums
   - Cyclic redundancy check

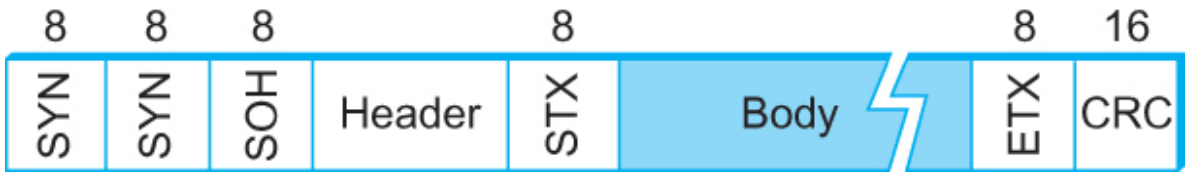


Byte-oriented protocols

- View frame as collection of bytes
  - Sentinel-based approaches
    - BISYNC
    - Point-to-point protocol
  - Byte-counting approaches
    - DDCMP

Sentinel-based

- BISYNC, Binary Synchronous Communication
  - Developed by IBM in the 1967 for System/360



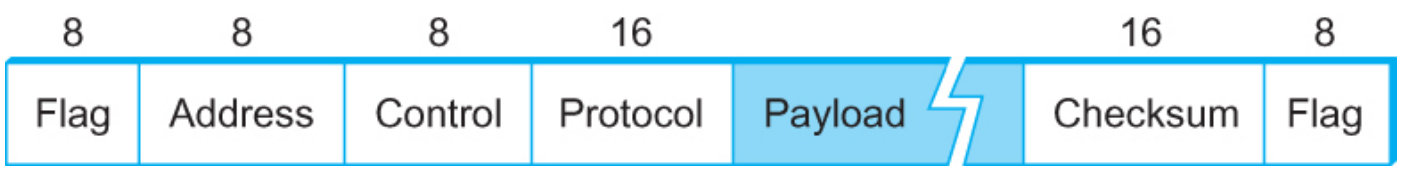
  - Special SYN, SOH, STX, ETX flag byte values
  - Escape occurrences of ETX in body
  - CRC → error detection

"If somehow all of the bisync interconnected machines in the world were to stop all at once, the results would be catastrophic. Many banks would cease to function. Some air traffic control systems would collapse. Many of the point-of-sale systems in retail stores would fail. Many credit and debit cards would become useless. EDI (electronic data interchange) networks that manage much of the business-to-business commerce would crash, We don't mean to suggest that there would be a total collapse of all, or even most, of these systems but nonetheless, bisync is still a vital link in the chain of the world's computer infrastructure."

-Serengeti Systems

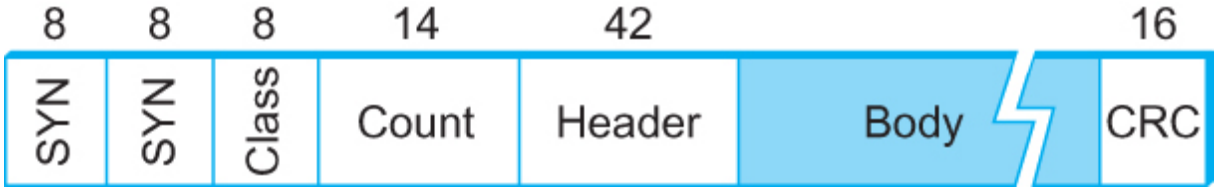
Sentinel-based

- PPP, Point-to-Point Protocol
  - Common on Internet links, e.g. dialup & DSL
    - PPPoE (PPP over Ethernet), PPPoA (PPP over ATM)
  - Special flag value, 0111 1110
  - Address, control → uninteresting default values
  - Protocol code, e.g. IP/IPX/LCP
  - Payload negotiated via LCP (link control protocol)
  - Checksum → error detection



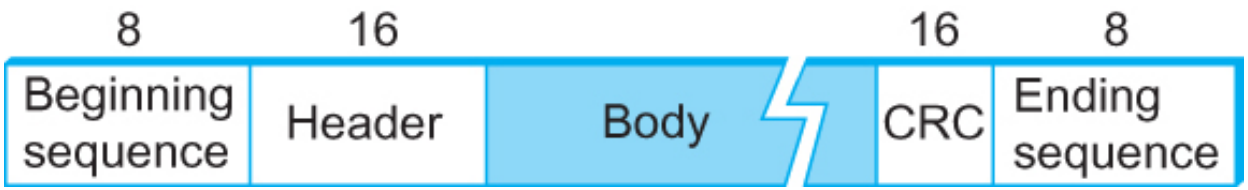
Byte-counting

- Byte-counting approach
  - Instead of sentinels, include count of items
  - DDCMP (Digital Data Communications Message Protocol)
    - Created by DEC in 1974
  - If count corrupted, causes framing error
    - May result in incorrect back-to-back frames
    - Sentinel-based approaches have same problem



Bit-oriented protocols

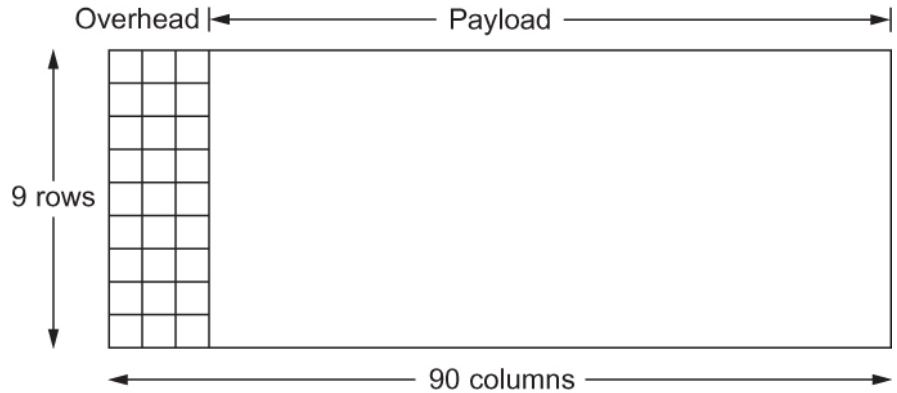
- Frame is a collection of bits
  - HDLC (high-level data link control)
  - Begin and end frame with 0111 1110



  - Sender: except for begin/end, after sending five consecutive 1's, stuff a 0
  - Receiver: after receiving five consecutive 1's, if next bit is 0 discard, else:
    - If next bit 0, end-of-frame else frame error

Clock-based framing

- SONET - Synchronous Optical Network
  - Dominant standard for long haul data
  - No bit stuffing, fixed frame size, 125 μs
  - First two bytes of frame contain special bit pattern
  - Look for special pattern every 810 bytes
  - Payload XOR scrambled to ensure bit transitions



SONET

- Fixed set of data rates
- Multiple low-speed links combined
  - Time division multiplexing (TDM)

Optical carrier level	Frame format	Rate
OC-1	STS-1	51.84 Mbps
OC-3	STS-3	155.520 Mbps
OC-12	STS-12	622.080 Mbps
OC-24	STS-24	1.255 Gbps
OC-48	STS-48	2.488 Gbps
OC-192	STS-192	9.953 Gbps
OC-768	STS-768	39.813 Gbps

- http://www.youtube.com/watch?v=DM-pLPy8Md0

- http://www.youtube.com/watch?v=dOyKdJWPlZY

Error detection

- Error detection
  - Parity checking
  - Checksum
  - Cyclic Redundancy Check
- Error correction
  - Retransmission
  - Forward error correction (ECC)
    - Hamming codes, Reed-Solomon codes, low-density parity check code (LDPC)
    - Examples: DVDs, WiMax, 802.11n

Error detection

- Basic idea: add redundant data
  - Simple scheme:
    - Send two copies of data
    - Compare copies, any differences implies error
    - High overhead, 2n bits to send n bits data
  - More complex schemes:
    - Strong error detection with k redundant bits
    - k << n
    - e.g. Ethernet frame with 12K bits, 32-bit CRC

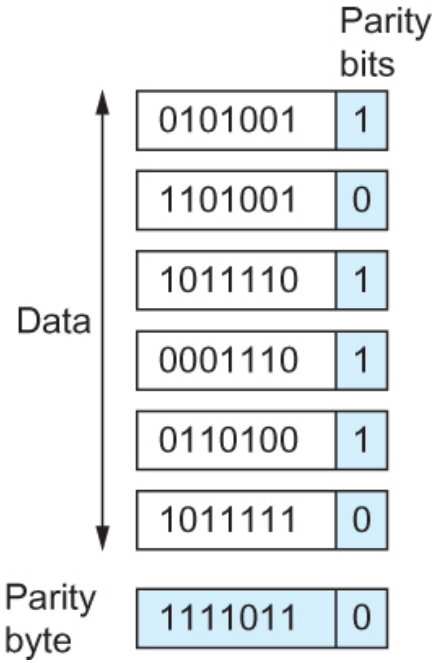
Parity checking

- One dimensional parity
  - Set parity bit so number of 1s odd or even
  - Detects all single bit errors
  - Example (7 bits data, 1 bit data):

data	even parity	odd parity
0010 101	0010 1011	0010 1010
1100 110	1100 1100	1100 1101
0000 000	0000 0000	0000 0001

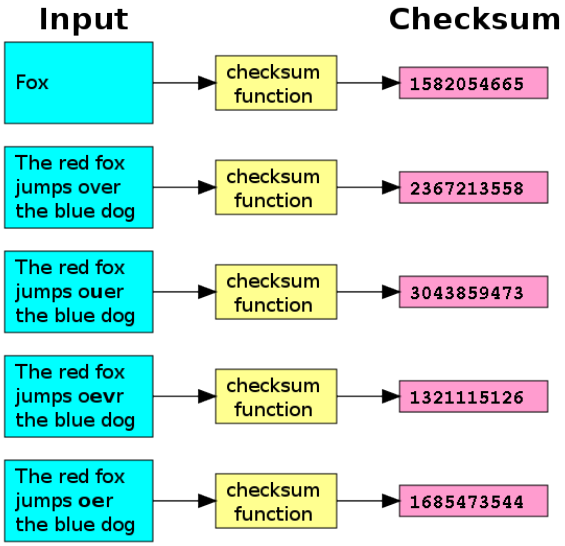
Parity checking

- Two-dimensional parity
  - Arrange bytes in a table
  - Parity over rows and over columns
  - Catches all 1-3 bit errors
  - Catches most 4 bit errors



Checksum

- Internet checksum algorithm
  - Add up 16-bit words and transmit result
  - Not used in link-layer
    - Used in higher layers like TCP and UDP
  - Advantages:
    - Small number of redundant bits
    - Easy to implement
  - Disadvantages:
    - Weak protection



Checksum

- Algorithm:
  - Add data using one's complement
  - Checksum is complement of summation
  - One's complement:
    - Negative numbers are bit complement of positive
    - Carry out from most significant bit, increment by 1

1	1	1	1	1	1	1		
	0	0	0	0	1	1	1	1
+	1	1	1	1	1	0	1	0
	0	0	0	0	1	0	0	1
+								1
	0	0	0	0	1	0	1	0

	15
+	-5
	10

Checksum algorithm

```
u_short cksum(u_short *buf, int count)
{
        register u_long sum = 0;
        while (count--)
        {
                sum += *buf++;
                if (sum & 0xFFFF0000)
                {
                        /* carry occurred so wrap around */
                        sum &= 0xFFFF;
                        sum++;
                }
        }
        return ~(sum & 0xFFFF);
}
```

Cyclic redundancy check

- CRC (cyclic redundancy check)
  - Maximize detection while minimizing redundancy
  - (n+1) bit message = n degree polynomial

Message (x)	polynomial M(x)
1001 1010	$1x^7 + 0x^6 + 0x^5 + 1x^4 + 1x^3 + 0x^2 + 1x^1 + 0x^0$ $= x^7 + x^4 + x^3 + x^1$

  - Sender/receiver agree on divisor polynomial C(x)
    - C(x) is polynomial of degree k (k << n)
  - Extend message M(x) to include extra bits that make it evenly divisible by C(x)

Common CRC polynomials

Name	Used in	C(x)	Generator
CRC-8	ATM	$x^8 + x^2 + x^1 + 1$	1 0000 0111
CRC-10	ATM	$x^{10} + x^9 + x^5 + x^4 + x^1 + 1$	110 0011 0011
CRC-12	Telecom systems	$x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$	1 1000 0000 1111
CRC-16	USB, Bisync	$x^{16} + x^{15} + x^2 + 1$	1 1000 0000 0000 0011
CRC-CCITT	Bluetooth, X.25, SD, HDLC	$x^{16} + x^{12} + x^5 + 1$	1 0001 0000 0010 0001
CRC-32	Ethernet, SATA, MPEG-2, Gzip, PKZIP, PNG, ATM	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$	1 0000 0100 1100 0001 0001 1101 1011 0111

- CRC will detect:
  - All single-bit errors, if x^k and x^0 are nonzero
  - All double-bit errors, if C(x) has a factor with 3 or more terms
  - Any odd number of errors, if C(x) contains the factor (x+1)
  - Any burst error, if burst is less than k bits

Generating a CRC

- Adding a CRC to a message:
  - Assume:
    - Message M(x) of (n+1) bits
    - Generator polynomial C(x) of degree k
  - Add k zeros to right side of M(x)
  - Find remainder by dividing by C(x)
  - Replace k zeros on right of M(X) with remainder

Example CRC generation

Message	1001 1010	$M(x) = x^7 + x^4 + x^3 + x^1$
Generator	1101	$C(x) = x^3 + x^2 + 1$

```
1 1 0 1 | 1 0 0 1 1 0 1 0 0 0 0
          1 1 0 1
            1 0 0 1
            1 1 0 1
              1 0 0 0
              1 1 0 1
                1 0 1 1
                1 1 0 1
                  1 1 0 0
                  1 1 0 1
                      1 0 0 0
                      1 1 0 1
                        1 0 1
```

- Repeatedly XOR generator with bits from augmented M(x)
- Final remainder is CRC
- Final message:
  1001 1010 101

Checking a CRC

- Checking a received message:
  - Receiver gets M(x) + calculated CRC
  - Divides by agreed C(x)
  - If remainder = 0, no error detected

Example uncorrupted message

Message	1001 1010	$M(x) = x^7 + x^4 + x^3 + x^1$
Generator	1101	$C(x) = x^3 + x^2 + 1$
CRC	101	

```
1 1 0 1 | 1 0 0 1 1 0 1 0 1 0 1
          1 1 0 1
          -------
            1 0 0 1
            1 1 0 1
            -------
              1 0 0 0
              1 1 0 1
              -------
                1 0 1 1
                1 1 0 1
                -------
                  1 1 0 0
                  1 1 0 1
                  -------
                        1 1 0 1
                        1 1 0 1
                        -------
                              0
```

- Repeatedly XOR generator with bits from message + CRC
- Final remainder should be zero

Example corrupted message

Message	1001 1010	$M(x) = x^7 + x^4 + x^3 + x^1$
Generator	1101	$C(x) = x^3 + x^2 + 1$
CRC	101	

```
1 1 0 1 | 1 0 0 1 1 0 0 0 1 0 1
          1 1 0 1
          -------
            1 0 0 1
            1 1 0 1
            -------
              1 0 0 0
              1 1 0 1
              -------
                1 0 1 0
                1 1 0 1
                -------
                  1 1 1 0
                  1 1 0 1
                  -------
                    1 1 1 0
                    1 1 0 1
                    -------
                      1 1 1
```

- Repeatedly XOR generator with bits from message + CRC
- Final remainder not zero so must be an error in transmitted message or CRC

Error detection rate

Type	Length	Error detection
checksum	8-bit	99.6094%
checksum	16-bit	99.9985%
CRC	32-bit	99.9999%

Summary

- Framing
  - Numerous ways to separate blocks of data:
    - Sentinel sequences, count fields, clock synchronization
- Error detection
  - Small amount of redundancy provides strong error detection
    - Checksum (used in transport layer)
    - CRC (used in link layer)