

Data Link Layer Overview



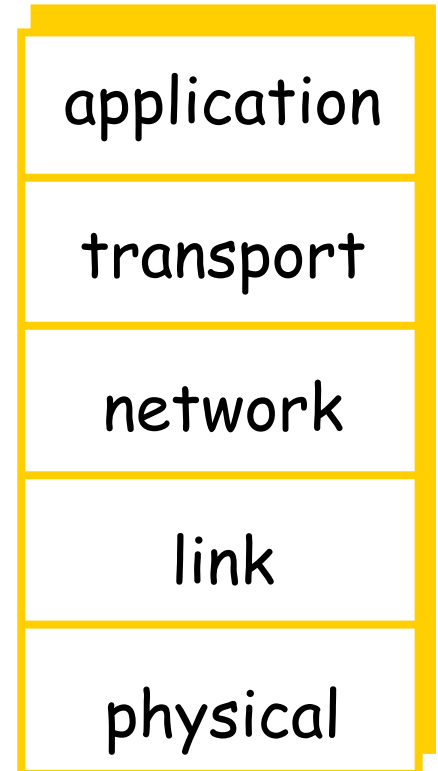
Outline:

- Introduction and services of the data link layer
- Encoding
- Framing
- Error detection and correction

Internet Architecture

Bottom-up:

- **physical:** electromagnetic signals "on the wire"
- **link:** data transfer between neighboring network elements
- **network:** host-host multi-hop connectivity
 - routing, addressing
- **transport:** host-host data transport
 - reliable data transport, congestion control, flow control
- **application:** anything you want to do on computer networks





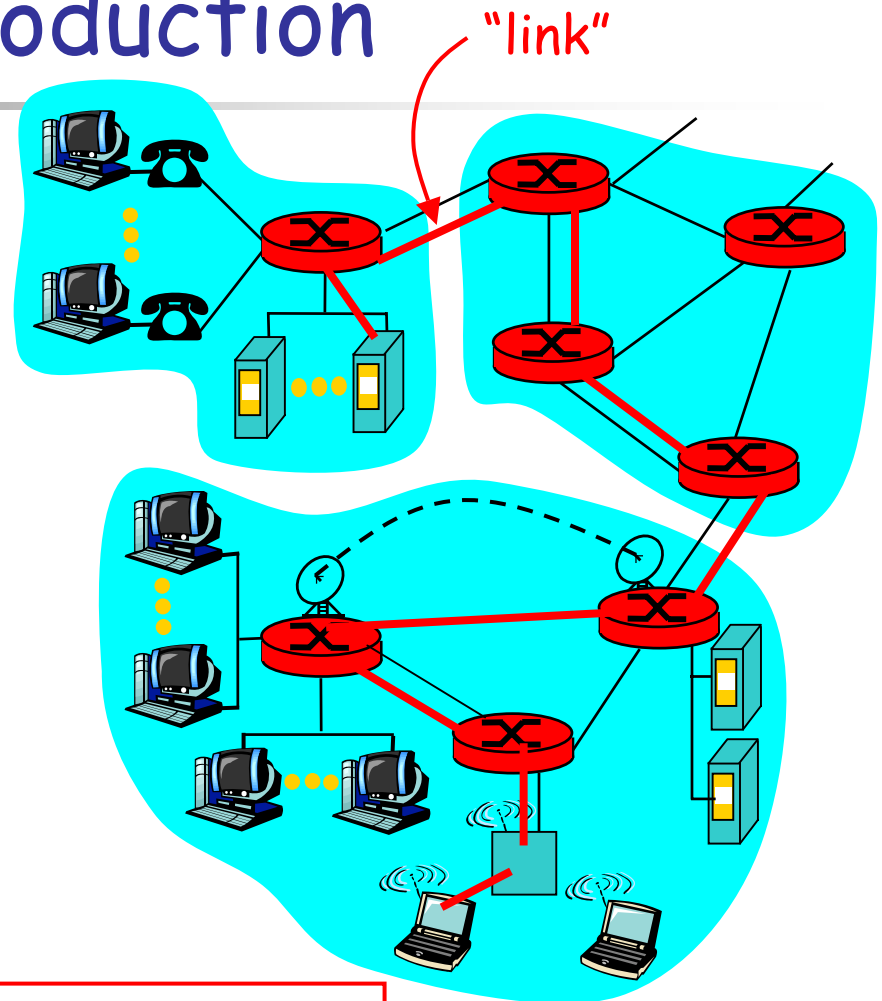
Outline

- Introduction and services of the data link layer
- Encoding
- Framing
- Error detection and correction

Link Layer: Introduction

Some terminology:

- nodes in the network edge are **hosts**; nodes in the network core are **routers**
- communication channels that connect adjacent nodes along communication path are **links**
 - LANs, modem links, satellite links, wireless radio links



data-link layer has responsibility of transferring data from one node to adjacent node over a link.



Link Layer Services

- **Encoding**
 - encode binary data into electromagnetic signals
- **Framing**
 - encapsulate data into frame, adding header, trailer
 - 'physical addresses' used in frame headers to identify source, dest (different from IP address!)
- **Link access control**
 - coordinate access for shared link
- **Flow control**
 - pacing between sending and receiving nodes such that the sender does not overwhelm the receiver

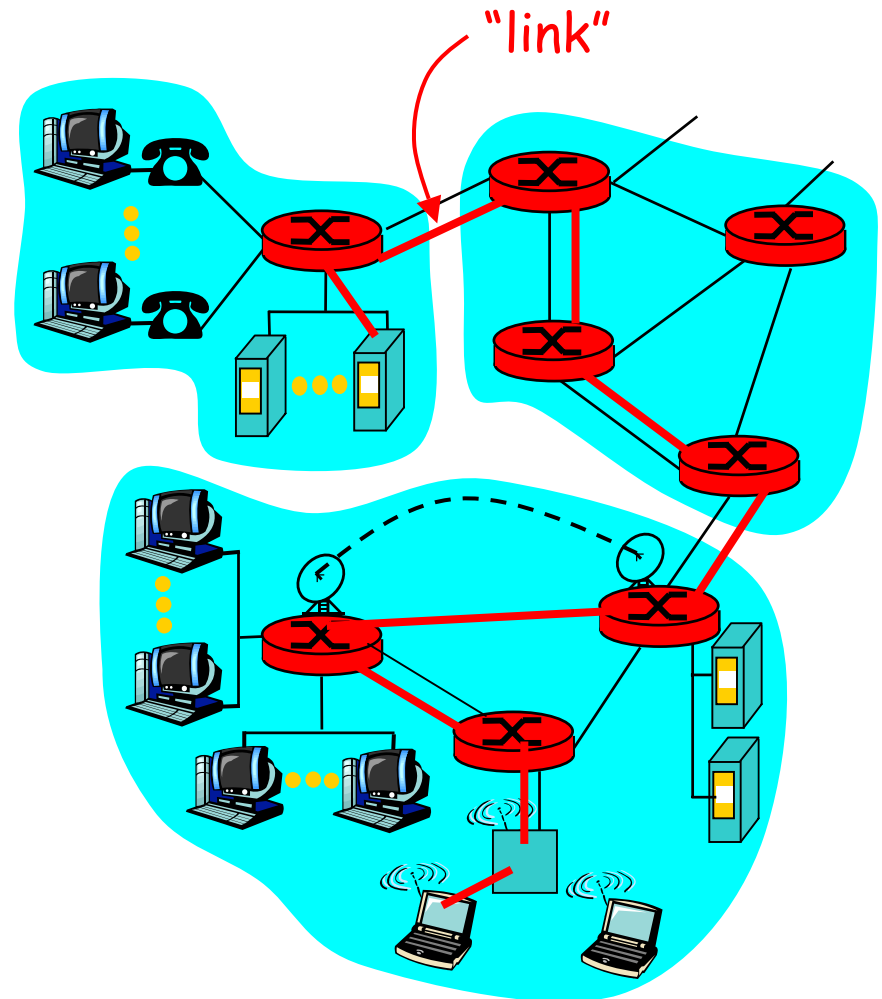


More Link Layer Services

- **Error detection**
 - errors caused by signal attenuation, electromagnetic noise.
 - receiver detects presence of errors:
 - drops frame or signals sender for retransmission
- **Error correction**
 - receiver identifies **and corrects** error(s) without resorting to retransmission
- **Reliable delivery over the link**
 - typically mean no data loss and/or in-order delivery

Variety of Link Types

- Data transfer over different links:
 - e.g., first link (local-area cable)
 - intermediate links (long-haul fiber)
 - last link (wireless)
- Different links may demand different services
 - encoding
 - link access control
 - reliable delivery
 -



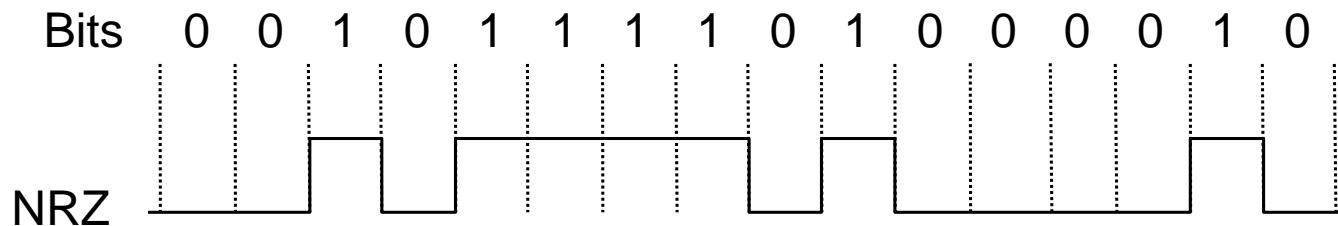


Outline

- Introduction and services of the data link layer
- Encoding
- Framing
- Error detection and correction

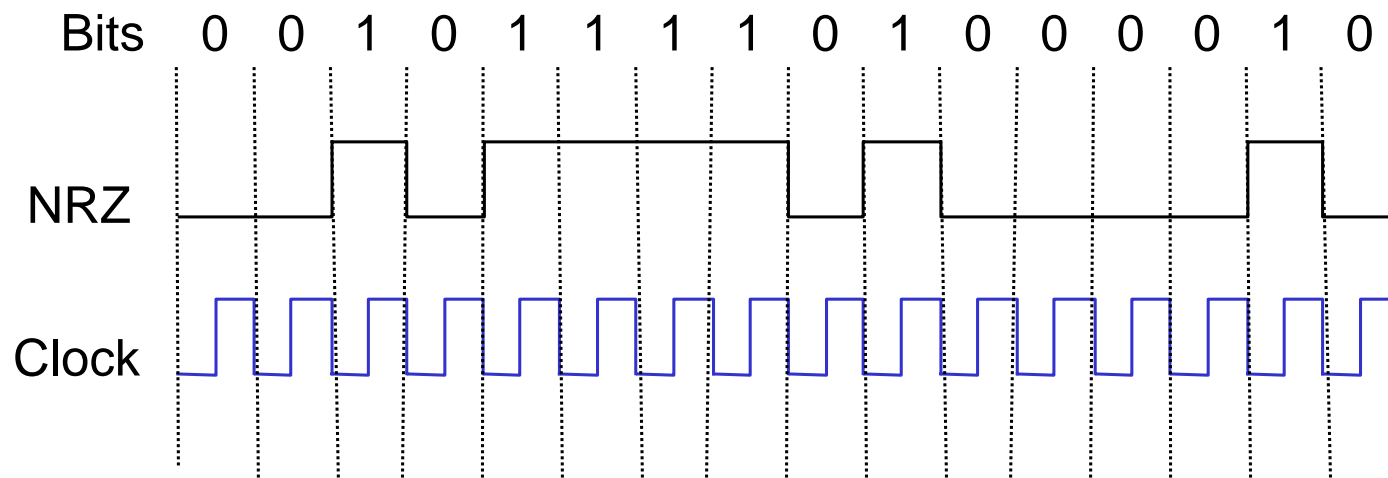
Encoding

- Signals propagate over a physical medium
 - modulate electromagnetic waves
 - e.g., vary voltage
- Encode binary data onto signals
 - e.g., low signal as 0 and high signal as 1
 - known as Non-Return to Zero (NRZ)



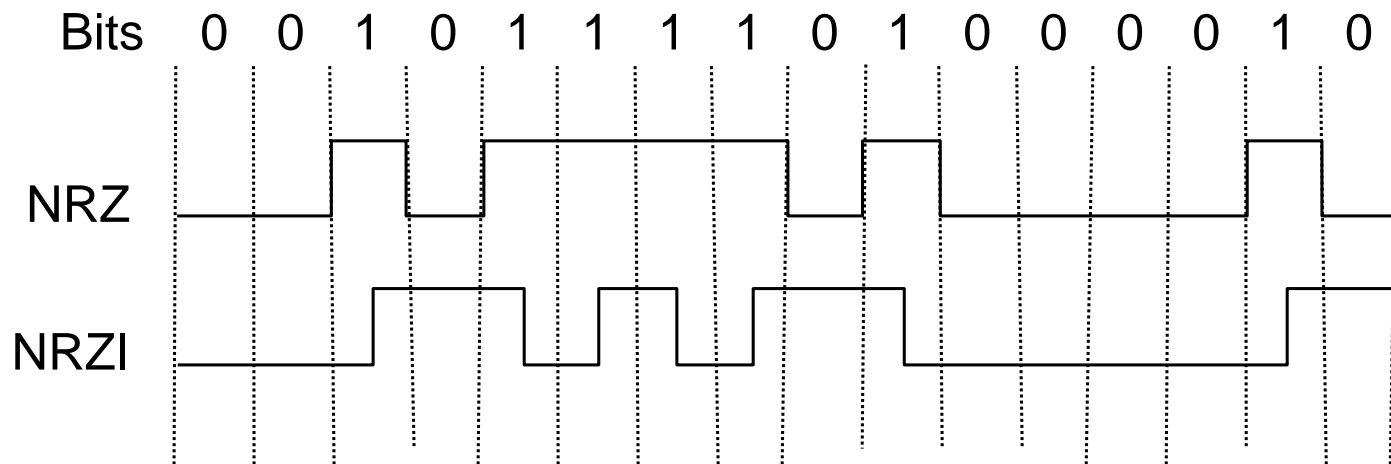
Problem with Consecutive 1s or 0s

- Clock synchronization
 - both the sender and the receiver maintains a clock
 - clock synchronized at signal change



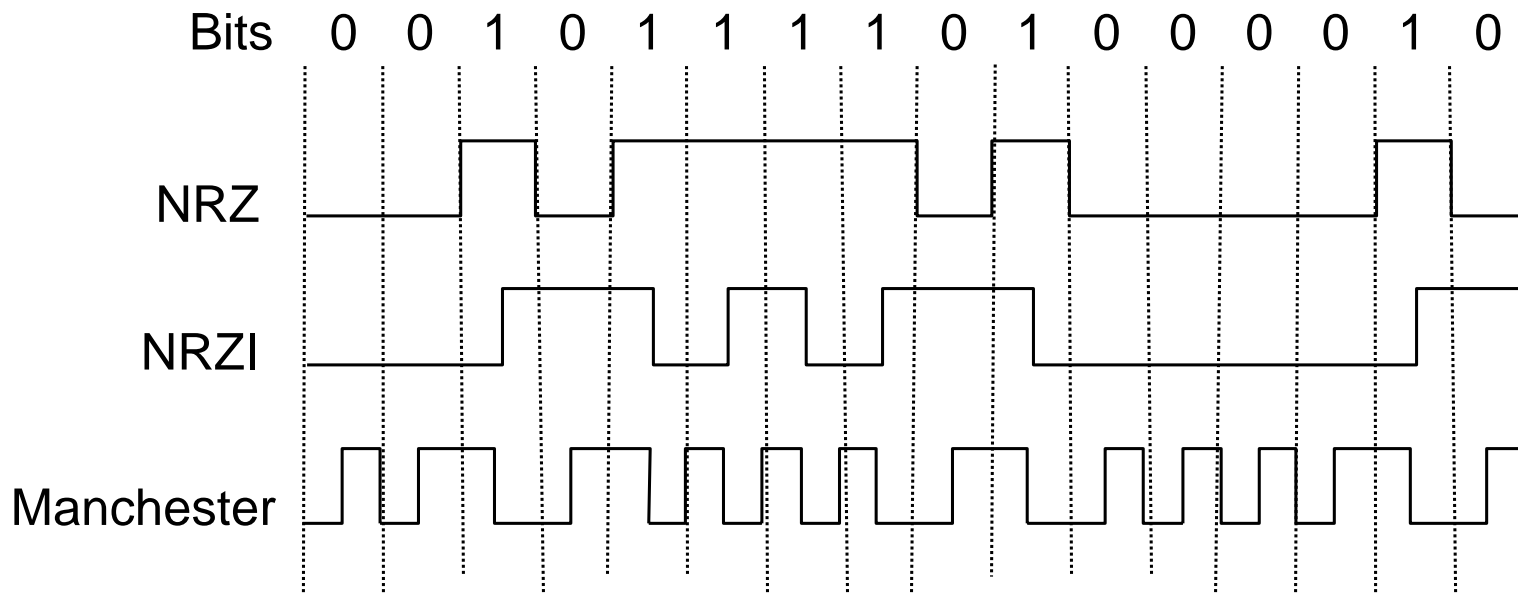
An Alternative Encoding

- Non-return to Zero Inverted (NRZI)
 - make a transition from current signal to encode a one; stay at current signal to encode a zero
 - only solves the problem of consecutive ones



Manchester Encoding

- Manchester
 - encode 0 into low-to-high transition; encode 1 into high-to-low transition.
 - only 50% efficient.





4B/5B Encoding

- every 4 bits of data encoded in a 5-bit code
- since there are more 5-bit codes than 4-bit data; some codes are not used; we can design a mapping/encoding such that no data is encoded into 5-bit code with more than one leading 0 or more than two trailing 0s
- thus, never get more than three consecutive 0s
- resulting 5-bit codes are transmitted using NRZI
- efficiency?
 - 80%



Outline

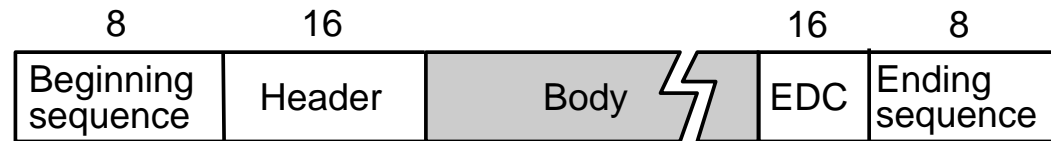
- Introduction and services of the data link layer
- Encoding
- Framing
 - break sequence of bits into frames, add protocol headers/trailers
- Error detection and correction

Where each frame begins and ends?

- Sentinel-based

- delineate frame with special pattern: 01111110

- e.g.,



- problem: special pattern appears in the payload
- solution: **bit stuffing**
 - sender: insert 0 after five consecutive 1s
 - receiver: delete 0 that follows five consecutive 1s
- similar to escape characters in program languages

- Counter-based

- include payload length in header

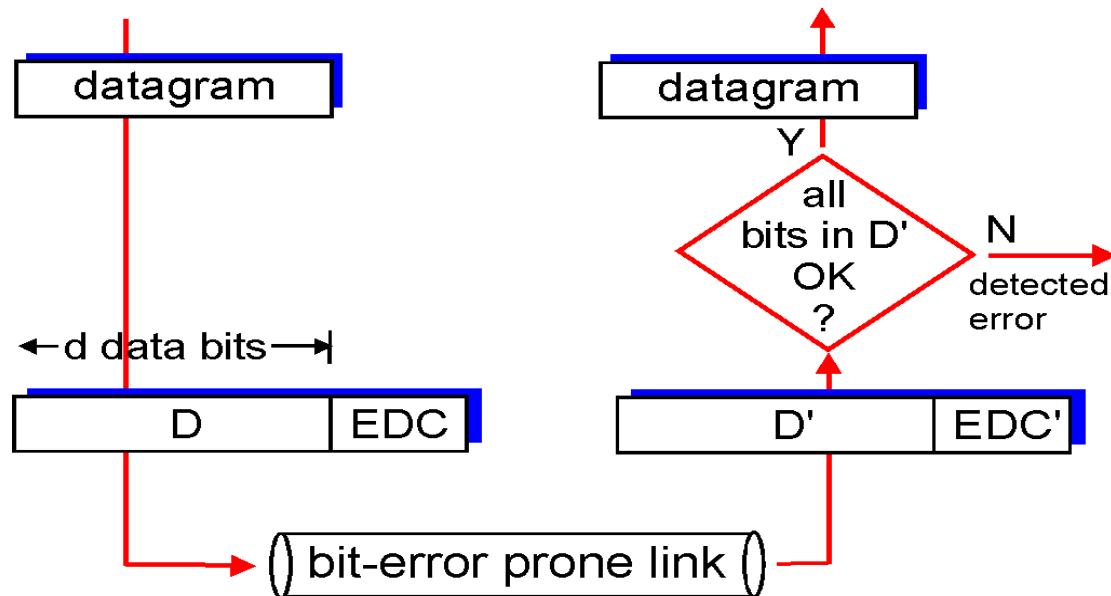


Outline

- Introduction and services of the data link layer
- Encoding
- Framing
- Error detection and correction

Error Detection and Correction

D - Data protected by error checking, may include header fields
EDC - Error Detection and Correction bits (redundancy)



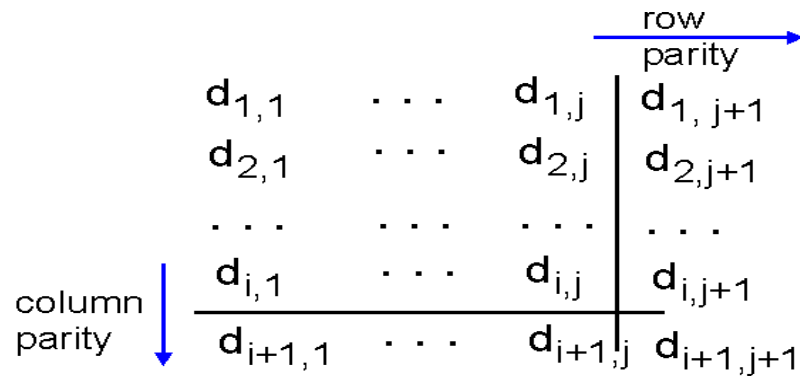
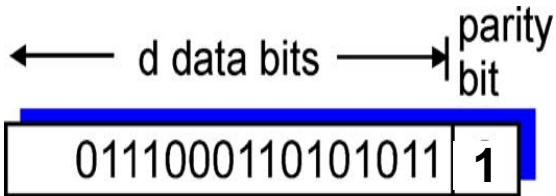
Parity Checking

Two Dimensional Bit Parity:

Detect and correct single bit errors

Single Bit Parity:

Detect single bit errors



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

no errors

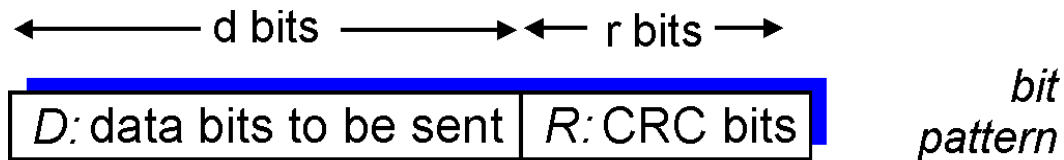
1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

parity
error

*correctable
single bit error*

Cyclic Redundancy Check

- view data bits, D , as a binary number
- choose $r+1$ bit pattern (generator), G
- for each D : choose r CRC bits, R , such that
 - $\langle D, R \rangle$ exactly divisible by G (modulo 2)



$$D * 2^r + R$$

mathematical formula

- receiver knows G , divides $\langle D', R' \rangle$ by G . If non-zero remainder: error detected!
- can detect all burst errors less than $r+1$ bits

CRC Example

There are no carries and borrows in modulo-2 arithmetic. Addition, subtraction, and bit-wise XOR are all identical.

Want:

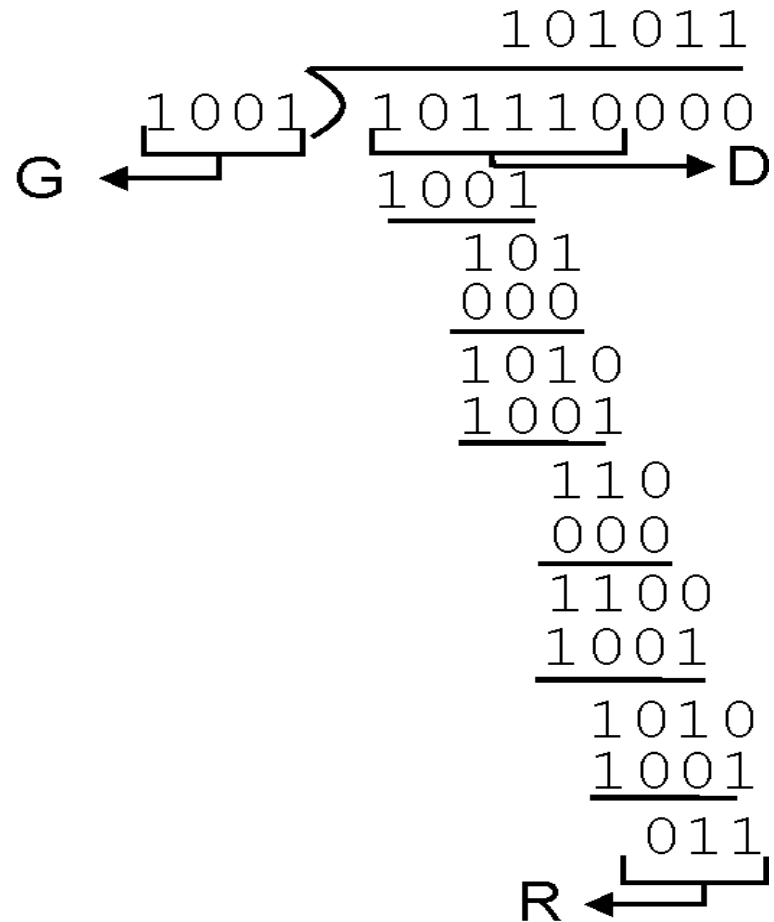
$$D \cdot 2^r + R = nG$$

equivalent to (modulo-2):

$$D \cdot 2^r = nG + R$$

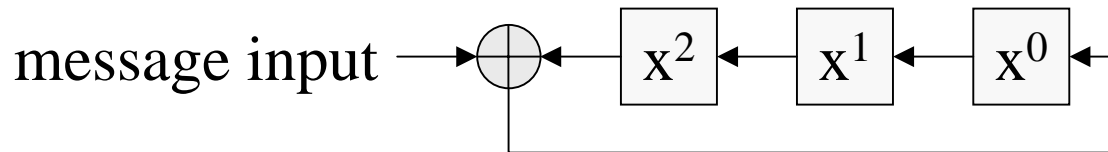
equivalent to:

$$R = \text{remainder} \left[\frac{D \cdot 2^r}{G} \right]$$



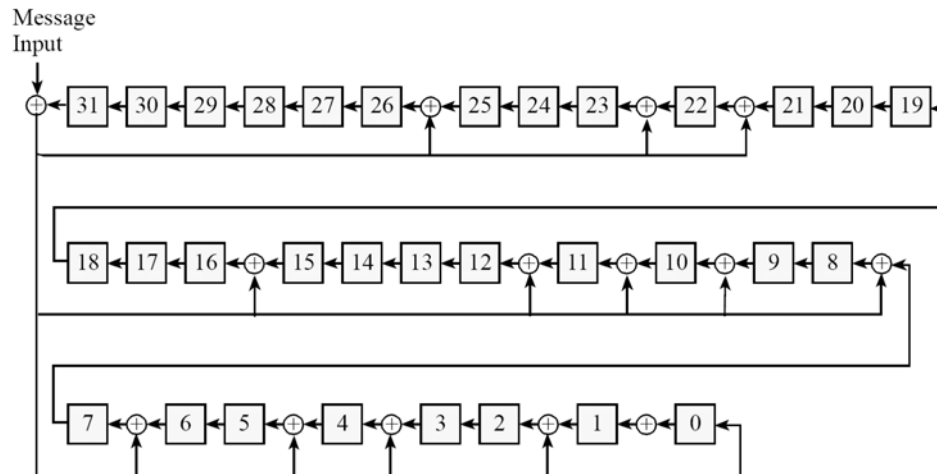
CRC Representations

- Bit strings, processed with modulo-2 arithmetic.
 - Example: $G = 1001$
- Polynomials over $GF(2)$ [Galois field with two elements], bits are polynomial coefficients.
 - Example: $G(x) = x^3 + x^0$
- Hardware feedback shift register, generator determines which register taps receive feedback (with XOR) from the input and the shift out from highest-order bit.
 - Example:



CRC-32

- Bit strings
 - $G = 1\ 0000\ 0100\ 1100\ 0001\ 0001\ 1101\ 1011\ 0111$
- Polynomials over $GF(2)$
 - $G(x) =$
 $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$
- Hardware feedback shift register
 - (diagram by permission of Henry S Warren):





Internet Checksum

Sender:

- treat data as sequence of 16-bit integers
- add using 16-bit 1's complement arithmetic; take 1's complement of the result as the checksum
- sender puts checksum value into TCP/UDP checksum field

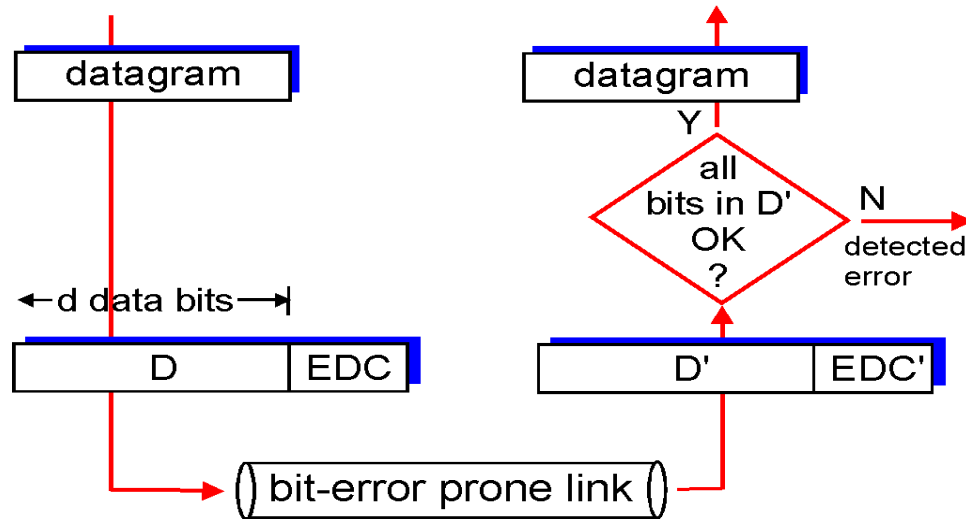
Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
 - NO - error detected
 - YES - no error detected

1's complement addition - carryout from the most-significant bit adds to the least-significant bit: $1010 + 1100 = 0111$

Note: used at network layer (IP header) and transport layer (TCP/UDP).
Easy to recalculate incrementally for small changes (e.g. TTL field).

Error Detection Reliability



- Error detection is not 100% reliable!
 - it is possible for D and D' to share the same EDC
 - what is the chance?
 - larger EDC field yields better detection and correction
 - it is also possible for EDC has errors in a way that hides the data error



Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, Keith Ross, and Henry Warren. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).