

Data Link Layer Overview

First of four classes on the data link layer

9/8/2014

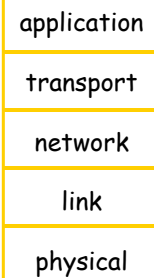
CSC 257/457 - Fall 2014

1

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



9/8/2014

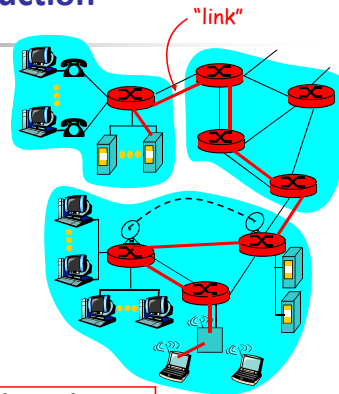
CSC 257/457 - Fall 2014

2

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.

9/8/2014

CSC 257/457 - Fall 2014

3

Link Layer Functions

- **Encoding**
 - encode binary data to/from 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

9/8/2014

CSC 257/457 - Fall 2014

4

More Link Layer Functions

- **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

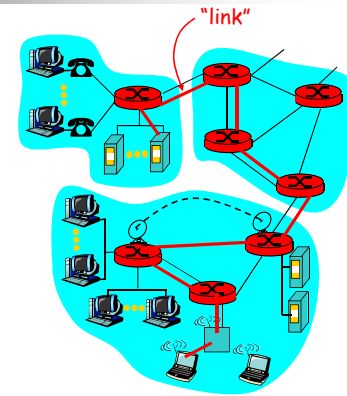
9/8/2014

CSC 257/457 - Fall 2014

5

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 functions
 - encoding
 - link access control
 - reliable delivery
 -



9/8/2014

CSC 257/457 - Fall 2014

6

Outline

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

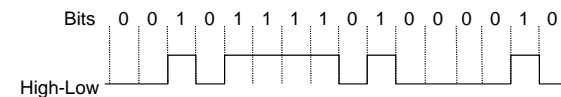
9/8/2014

CSC 257/457 - Fall 2014

7

Encoding

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



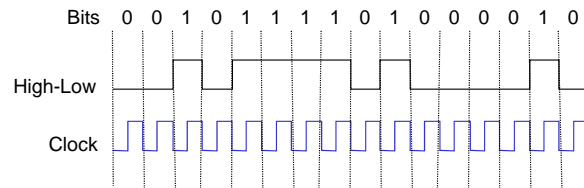
9/8/2014

CSC 257/457 - Fall 2014

8

Problem with Consecutive 1s or 0s

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



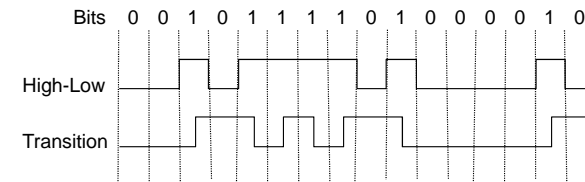
9/8/2014

CSC 257/457 - Fall 2014

9

An Alternative Encoding

- Transition encoding
 - 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



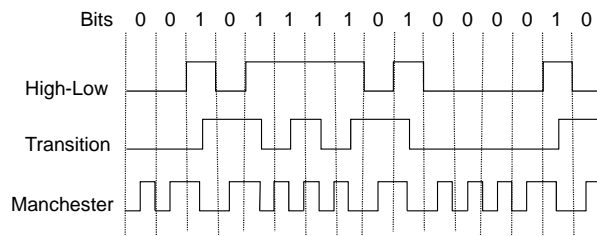
9/8/2014

CSC 257/457 - Fall 2014

10

Manchester Encoding

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



9/8/2014

CSC 257/457 - Fall 2014

11

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 the transition encoding
- efficiency?
 - 80%

9/8/2014

CSC 257/457 - Fall 2014

12

Outline

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

9/8/2014

CSC 257/457 - Fall 2014

13

Where each frame begins and ends?

- Sentinel-based framing
 - delineate frame with special begin/end patterns
 - e.g.,

8	16	16	8
Beginning sequence	Header	Body	EDC Ending sequence
- problem: end pattern appears in the payload
 - solution: **bit stuffing**
 - for pattern 01111110:
 - Sender – insert 0 after five consecutive 1s
 - Receiver – delete 0 that follows five consecutive 1s
 - similar to escape characters in program languages

9/8/2014

CSC 257/457 - Fall 2014

14

Where each frame begins and ends?

- Counter-based framing
 - include payload length in header

9/8/2014

CSC 257/457 - Fall 2014

15

Outline

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

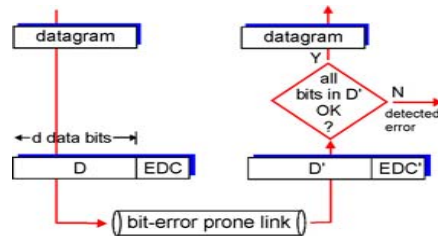
9/8/2014

CSC 257/457 - Fall 2014

16

Error Detection and Correction

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



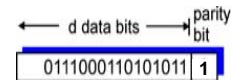
9/8/2014

CSC 257/457 - Fall 2014

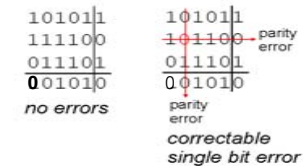
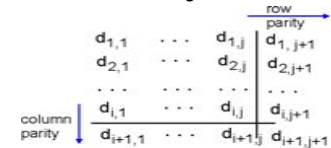
17

Parity Checking

Single Bit Parity:
Detect single bit errors



Two Dimensional Bit Parity:
Detect and correct single bit errors



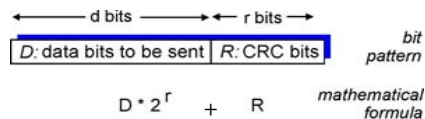
9/8/2014

CSC 257/457 - Fall 2014

18

Cyclic Redundancy Check

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



- Receiver:
 - divides $\langle D', R' \rangle$ by G . Non-zero remainder: error detected!
 - can detect all burst errors fewer than $r+1$ bits

9/8/2014

CSC 257/457 - Fall 2014

19

CRC Example

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

Want:

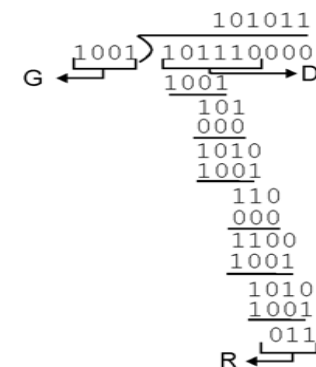
$$D \cdot 2^r + R = nG$$

equivalent to (modulo-2):

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

equivalent to:

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



9/8/2014

CSC 257/457 - Fall 2014

20

Internet Checksum

Note: used at network/transport layers (IP, TCP/UDP).

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

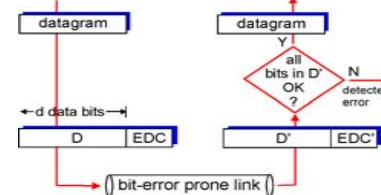
1's complement addition – carryout from the most-significant bit adds to the least-significant bit: $1010 + 1100 = 0111$

9/8/2014

CSC 257/457 - Fall 2014

21

Error Detection Reliability



- Error detection is not 100% reliable!
- ⇒ It is possible for corrupted D' to match EDC'
 - What is the chance that a random error escapes detection? Assume that the EDC field has r bits and that the EDC value for an arbitrary data block is uniformly distributed in the space of possible EDC values.
 - Larger EDC field yields better detection and correction.

9/8/2014

CSC 257/457 - Fall 2014

22

Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. 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).

9/8/2014

CSC 257/457 - Fall 2014

23