# Communication Networks (0368-3030) / Spring 2011

The Blavatnik School of Computer Science,
Tel-Aviv University

Allon Wagner

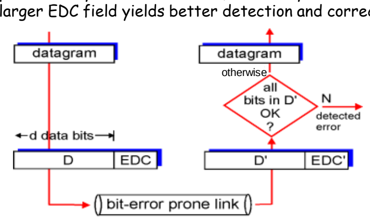
# Error Detection and Correction

Kurose & Ross, Chapter 5.2 ($5^{th}$ ed.)

## Error Detection

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

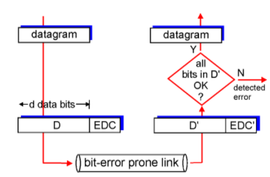
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction




Data Link Layer 5-3

## Example – Parity bit

- Assume D has *d* bits.
- The EDC is one bit s.t. the number of 1's in the *d+1* bits (D and the EDC) is even.
- Receiver can detect an error inverting an odd number of bits
- Example:
  - D = 11101
  - Sender sends 111010
  - Receiver gets 101010
  - Illegal parity – an error has occurred
  - Receiver cannot correct the error




Error detection vs.
Error correction

## Some theory

- Assume all messages are of size *d*
  - We have $2^d$ possible messages, all of them valid
  - When some bits flip, the receiver still gets a valid message
  - It cannot know there was an error
- The proposed solution:
  - Add *r* bits of redundancy .
  - Now, we have $2^{d+r}$ possible messages, but only $2^d$ of them are valid (these are called *codewords*).
  - "Small errors" are likely to transform the valid message into an invalid one, so that the receiver knows an error has occurred.

## Some theory (cont.)

- Definition: The Hamming-distance of two strings x and y is the number of bits in which they differ, denoted dH(x,y).
  - For instance: x = 110010 y = 111000. dH(x,y) = 2
- Definition: The Hamming-distance of an error-correction scheme ( = code) is the minimal Hamming-distance between two valid messages ( = codewords).
  dH(C) = min { dH(x,y) : x, y ϵ C }

## Parity bit revisited

- Assume all messages have *d* bits.
- The valid messages ( = codewords) are all the *d+1* messages s.t their total number of 1's is even.
- The Hamming-distance of this scheme is 2.
  - No two valid codewords x,y s.t. dH(x,y) = 1
    - If dH(x,y) = 1 then either x or y has an odd number of 1's.
  - There are two valid codewords with distance 2:
    - For instance, for d = 6
      x = 111100 y = 111111 → dH(x,y) = 2

## Why is Hamming-distance important?

- Theorem 1: If a code *C* has *dH(C) = k+1*, then it can detect all errors of *k* bits or less.
  - Such errors necessarily produce an invalid codeword
- Theorem 2 : If a code *C* has *dH(C) = 2k+1*, then it can correct all errors of *k* bits or less.
  - Think why
- And indeed: parity bit can detect all single-bit errors, but cannot correct any.

## CRC – Cyclic Redundancy Check

- Bits represent polynomials over GF(2)
  - Example: 100101 = $x^5 + x^2 + 1$
  - Addition and subtraction are actually XOR (no carry)
  - Example: 1101 + 0111 = 1010
    $(x^3 + x^2 + 1) + (x^2 + x + 1) = x^3 + x$
- Sender & receiver agree in advance on a generating polynomial *G* of degree *r*
- When sender wish to send *D,* it calculates *R* s.t. DR is divisible by *G*.
- When the receiver gets *D'R'* it divides it by *G*. If the remainder is not 0 – an error has occurred.

## Calculating *R*

- DR = $x^r \cdot D + R$
- We want: DR = $x^r \cdot D + R = n \cdot G$
  - But addition and subtraction are just XOR – they are interchangeable
- Equivalently, then, we want: $x^r \cdot D = n \cdot G + R$
- Namely, R is the remainder of $x^r \cdot D$ divided by G!
- Observation: *R*'s degree is at most *r-1*.
  - It is the remainder of dividing by *G*, and *deg(G) = r*

## CRC Example

- D = 101110 = $x^5 + x^3 + x^2 + x$
- G = 1001 = $x^3 + 1$
  - *r* = deg(G) = 3
- Shift *D r* bits to the left:
  - $x^r \cdot D = x^8 + x^6 + x^5 + x^4$
  - $x^r \cdot D$ = 101110000
- Now we can divide $x^r \cdot D$ by G:
  - on board

## CRC Example

Want:

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

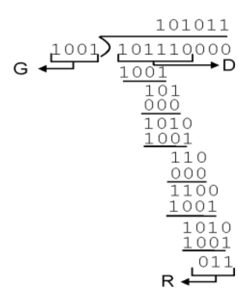
*equivalently:*

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

*equivalently:*

if we divide $D \cdot 2^r$ by G, want remainder R

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




Data Link Layer 5-12

## CRC Example (cont.)

- D = 101110 = $x^5 + x^3 + x^2 + x$
- G = 1001 = $x^3 + 1$
  - *r* = deg(G) = 3
- Now we can divide $x^r \cdot D$ by G:
  - we get: R = x + 1 = 011
- Sender sends:
  - DR = $x^r \cdot D + R$ = 101110011

## Ethernet's CSMA/CD

Kurose & Ross, Chapter 5.5.2 (5th ed.)

## The algorithm

When a the network layer generates a new frame:

1. If the adapter senses the channel to idle (that is no signal detected for 96 bit times) – start transmitting the frame
2. Otherwise (channel is busy) – wait until you sense no signal energy (plus 96 bit times) and then start transmitting
3. While transmitting – listen for signal coming from other adapters. If the adapter transmitted the entire frame without detecting signal energy – it is done with the frame.
4. If signal energy is detected while transmitting – stop transmitting the frame.
   I. Transmit a 48 bit jam-signal
   II. Exponential backoff:
   after experiencing the *n*-th collision is a row for the current frame choose *K* randomly from {0, 1, …, $2^m - 1$} with $m = min\{ n, 10 \}$.
   Wait K · 512 bit times and return to step 1.

## The jam signal

- The jam-signal makes sure all other transmitting adapters are aware of the collision
  - Suppose that A starts to transmit
  - Just before A's signal reaches B, B begins to transmit
  - B senses A's signal and aborts.
  - B transmitted just a few bits before aborting. These bits propogate to A but might not constitute enough energy for A to detect the collision!
  - To make sure A detects the collision, B transmits the 48-bit jam signal

## Exponential backoff

- When an adapter first detects collision, it cannot know how many adapters are involved in the collision,
- Exponential backoff dynamically adapts the waiting-time before reattempting transmission to the number of adapters involved in the collision
  - Few adapters involved: Choose K from a small set, so that no one waits unnecessarily
  - Many adapters involved: Choose K from a large set, so everyone is likely to choose a different time to transmit, and the collision will be resolved.

## Exponential backoff example

- Assume A and B both have a new frame to transmit. They both begin to transmit exactly on the same time and collide.
- They both choose *K* from {0, 1}
- The possible outcomes:

## Exponential backoff example (cont.)

| Case | A chooses | B chooses | Probability | Outcome |
|---|---|---|---|---|
| (a) | 0 | 0 | 0.25 | another collision on round 2 |
| (b) | 0 | 1 | 0.25 | A successful on round 2,<br>B successful on round 3 |
| (c) | 1 | 0 | 0.25 | B successful on round 2,<br>A successful on round 3 |
| (d) | 1 | 1 | 0.25 | another collision on round 3 |