Communication Networks
(0368-3030) / Fall 2013
The Blavatnik School of Computer Science,
Tel-Aviv University

Allon Wagner
Error Detection and Correction

Kurose & Ross, Chapter 5.2 (5th 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

\( d \) = data bits

\( D \quad EDC \) in datagram

(\( \) bit-error prone link (\)

\( D' \quad EDC' \) in datagram

all bits in \( D' \) OK?

N detected error
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 $d_H(x,y)$.
  - For instance: $x = 110010$ $y = 111000$. $d_H(x,y) = 2$

- **Definition:** The Hamming-distance of an error-correction scheme ($= \text{code}$) is the minimal Hamming-distance between two valid messages ($= \text{codewords}$).
  - $d_H(C) = \min \{ d_H(x,y) : x, y \in C \}$
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 $d = 6$
      $x = 111100$ $y = 111111 \Rightarrow dH(x,y) = 2$
Why is Hamming-distance important?

- Theorem 1: If a code $C$ has $d_H(C) = k+1$, 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 $d_H(C) = 2k+1$, 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:

\[ D \cdot 2^r \text{ XOR } R = nG \]

equivalently:

\[ D \cdot 2^r = nG \text{ XOR } R \]

equivalently:

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

\[ R = \text{remainder}\left[ \frac{D \cdot 2^r}{G} \right] \]
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 $\{0, 1, \ldots, 2^m - 1\}$ with $m = \min\{n, 10\}$.
   Wait $K \cdot 512$ 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 propagate 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.)

<table>
<thead>
<tr>
<th>Case</th>
<th>A chooses</th>
<th>B chooses</th>
<th>Probability</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>another collision on round 2</td>
</tr>
<tr>
<td>(b)</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
<td>A successful on round 2, B successful on round 3</td>
</tr>
<tr>
<td>(c)</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
<td>B successful on round 2, A successful on round 3</td>
</tr>
<tr>
<td>(d)</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
<td>another collision on round 3</td>
</tr>
</tbody>
</table>
CSMA/CD – Length of minimum frame

A network CSMA/CD with the following properties:

- Coaxial cable 250 meters long.
- Transmission rate 100 Mbit/sec.
- Signal propagation speed 200,000 km/second.

What is the minimal frame length?

\[ L : \text{minimal frame length} \]
\[ T : \text{the time it takes to transmit a frame of length} \ L \]
\[ \text{Require:} \ T \geq RTT \]
\[ T \geq 2 \cdot \frac{250}{200,000 \cdot 10^3} = 2.5 \cdot 10^{-6} \text{ s} \]
\[ L \geq (100 \cdot 10^6) \cdot (2.5 \cdot 10^{-6}) = 250 \text{ bit} \]
Example: Ethernet’s minimal frame length

- Assume an Ethernet network with the following properties (this is quite an old network):
  - Transmission rate 10Mbps
  - Built of coaxial cables of length up to 500m
  - Up to 4 repeaters allowed
  - Signal speed on coaxial cable 200,000 km/sec
  - Every repeater adds a delay of 3μs
- We show why the minimal frame length in this network is 64 bytes.
Example: Ethernet’s minimal frame length

- 4 repeaters = 5 network segments
- \( RTT = 2 \left( \frac{5 \cdot 500}{200,000 \cdot 10^3} + 4 \cdot 3 \cdot 10^{-6} \right) = 4.9 \cdot 10^{-5} \text{ s} \)
  \( = 49 \mu s \)
- \( L \geq (10 \cdot 10^6) \cdot (4.9 \cdot 10^{-5}) = 490 \text{ bits} \)
- Take the next 8-power (which is both convenient and adds a margin of safety):
  \[ L = 512 \text{ bits} = 64 \text{ bytes} \]