CSC458 - Lecture 2

Bits and Bandwidth
Error Detection and Correction

Stefan Saroiu
http://www.cs.toronto.edu/syslab/courses/csc458

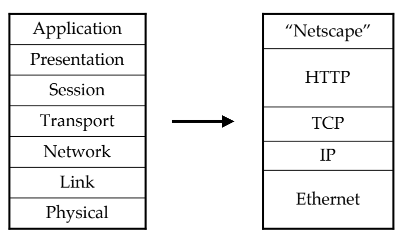
University of Toronto at Mississauga

Administrivia

- Project 1 is due next week
  - No other tutorials on project 1
- Homework 1 is due the following week
  - Next week's tutorial on material relevant to homework #1
  - No tutorial today
- Make sure you can post on Google groups!
  - Takes 1-2 days to join our group!!! Don't wait to the last min!!!
- I got a lot of question about the project
  - Joe might give you a better answer….

Last Time …

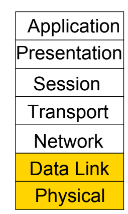
- Protocols, layering and reference models



The OSI Model   Sample Protocol Stack

Part 1

- Focus: *<u>How do we send a message across a wire?</u>*
- The physical / link layers:
  1. Different kinds of media
  2. Encoding bits, messages
  3. Model of a link

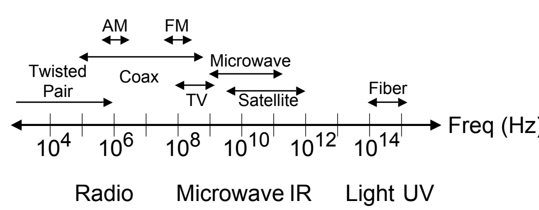


1. Different kinds of media

- Wire
  - Twisted pair, e.g., CAT5 UTP, 10 → 100Mbps, 100m
  - Coaxial cable, e.g, thin-net, 10 → 100Mbps, 200m
- Fiber
  - Multi-mode, 100Mbps, 2km
  - Single mode, 100 → 2400 Mbps, 40km
- Wireless
  - Infra-red, e.g., IRDA, ~1Mbps
  - RF, e.g., 802.11 wireless LANs, Bluetooth (2.4GHz)
  - Microwave, satellite, cell phones, …

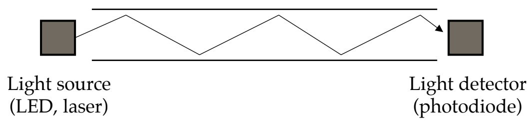
Wireless

- Different frequencies have different properties
- Signals subject to atmospheric/environmental effects



Fiber

- Long, thin, pure strand of glass
  - light propagated with total internal reflection
  - enormous bandwidth available (terabits)



- Multi-mode allows many different paths, limited by dispersion
- Chromatic dispersion if multiple frequencies

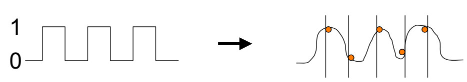
Aside: bandwidth of a channel

- EE: bandwidth (B, in Hz) is the width of the pass-band in the frequency domain
- CS: bandwidth (bps) is the information carrying capacity (C) of the channel

- Shannon showed how they are related by noise
  - noise limits how many signal levels we can safely distinguish
  - geekspeak: "cannot distinguish the signal from the noise"

2. Encoding Bits with Signals

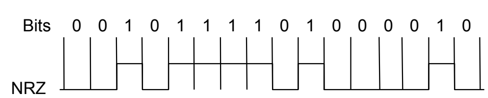
- Generate analog waveform (e.g., voltage) from digital data at transmitter and sample to recover at receiver



- We send/recover symbols that are mapped to bits
  - Signal transition rate = baud rate, versus bit rate
- This is baseband transmission …

NRZ and NRZI

- Simplest encoding, NRZ (Non-return to zero)
  - Use high/low voltages, e.g., high = 1, low = 0
- Variation, NRZI (NRZ, invert on 1)
  - Use transition for 1s, no transition for 0s



Clock Recovery

- Problem: How do we distinguish consecutive 0s or 1s?
- If we sample at the wrong time we get garbage …
- If sender and receiver have exact clocks no problem
  - But in practice they drift slowly
- This is the problem of clock recovery

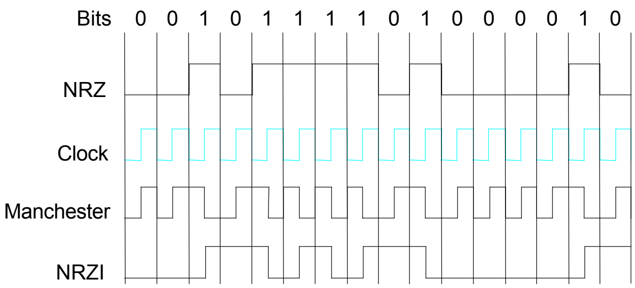
- Possible solutions:
  - Send separate clock signal → expensive
  - Keep messages short → limits data rate
  - Embed clock signal in data signal → other codes

Manchester Coding

- Make transition in the middle of every bit period
  - Low-to-high is 0; high-to-low is 1
  - Signal rate is twice the bit rate
  - Used on 10 Mbps Ethernet

- Advantage: self-clocking
  - clock is embedded in signal, and we re-sync with a phase-locked loop every bit
- Disadvantage: 50% efficiency

Coding Examples



4B/5B Codes

- We want transitions *and* efficiency …
- Solution: map data bits (which may lack transitions) into code bits (which are guaranteed to have them)

- 4B/5B code:
  - 0000 → 11110, 0001 → 01001, … 1111 → 11101
  - Never more than three consecutive 0s back-to-back
  - 80% efficiency

- This code is in LANs such as FDDI, 100Mbps Ethernet

3. Framing

- Need to send message, not just bits
  - Requires that we synchronize on the start of message reception at the far end of the link
  - Complete Link layer messages are called <u>frames</u>

- Common approach: Sentinels
  - Look for special control code that marks start of frame
  - And escape or "stuff" this code within the data region

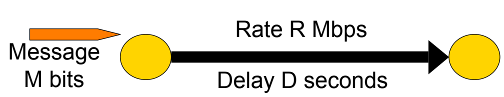
Example: Point-to-Point Protocol (PPP)

- IETF standard, used for dialup and leased lines

Flag 0x7E	(header)	Payload (variable)	(trailer)	Flag 0x7E

- Flag is special and indicates start/end of frame
- Occurrences of flag inside payload must be "stuffed"
  - Replace 0x7E with 0x7D, 0x5E
  - Replace 0x7D with 0x7D, 0x5D

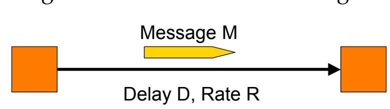
4. Model of a Link



- Abstract model is typically all we will need
  - What goes in comes out altered by the model
- Other parameters that are important:
  - The kind and frequency of errors
  - Whether the media is broadcast or not

Message Latency

- How long does it take to send a message?



- Two terms:
  - Propagation delay = distance / speed of light in media
    - How quickly a message travels over the wire
  - Transmission delay = message (bits) / rate (bps)
    - How quickly you can inject the message onto the wire
- Later we will see queuing delay …

Relationships

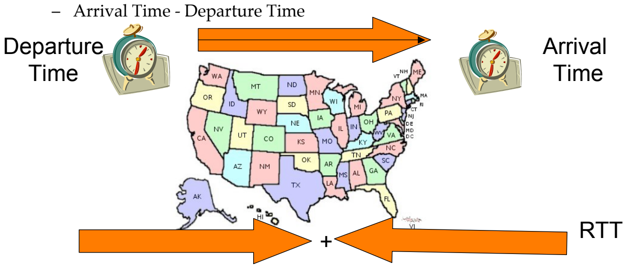
- Latency = Propagation + Transmit + Queue
- Propagation Delay = Distance/SpeedOfLight
- Transmit Time = MessageSize/Bandwidth

One-way Latency

- Dialup with a modem:
  - D = 10ms, R = 56Kbps, M = 1000 bytes
  - Latency = 10ms + (1024 x 8)/(56 x 1024) sec = 153ms!
- Cross-country with T3 (45Mbps) line:
  - D = 50ms, R = 45Mbps, M = 1000 bytes
  - Latency = 50ms + (1024 x 8) / (45 x 1000000) sec = 50ms!
- Either a slow link or long wire makes for large latency

Latency and RTT

- Latency is typically the one way delay over a link
  - Arrival Time - Departure Time



- The round trip time (RTT) is twice the one way delay
  - Measure of how long to signal and get a response

Throughput

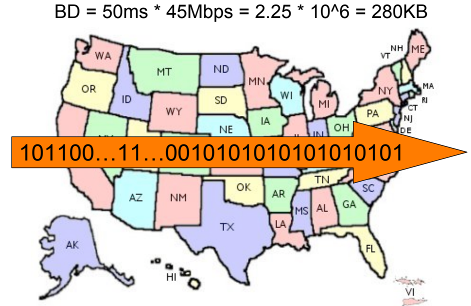
- Measure of system's ability to "pump out" data
  - NOT the same as bandwidth
- Throughput = Transfer Size / Transfer Time
  - Eg, "I transferred 1000 bytes in 1 second on a 100Mb/s link"
    - BW?
    - Throughput?
- Transfer Time = SUM OF
  - Time to get started shipping the bits
  - Time to ship the bits
  - Time to get stopped shipping the bits

Messages Occupy "Space" On the Wire

- Consider a 1b/s network.
  - How much space does 1 byte take?
- Suppose latency is 16 seconds.
  - How many bits can the network "store"
  - This is the BANDWIDTH-DELAY product
  - Measure of "data in flight."
  - 1b/s * 16s = 16b
- Tells us how much data can be sent before a receiver sees any of it.
  - Twice B.D. tells us how much data we could send before hearing back from the receiver something related to the first bit sent.
  - Implications?

A More Realistic Example

BD = 50ms * 45Mbps = 2.25 * 10^6 = 280KB



Part 1: Key Concepts

- We typically model links in terms of bandwidth and delay, from which we can calculate message latency
- Different media have different properties that affect their performance as links
- We need to encode bits into signals so that we can recover them at the other end of the channel.
- Framing allows complete messages to be recovered at the far end of the link

Part 2

- Error detection and correction
- Focus: How do we detect and correct messages that are garbled during transmission?
- The responsibility for doing this cuts across the different layers

Application
Presentation
Session
Transport
Network
Data Link
Physical

Errors and Redundancy

- Noise can flip some of the bits we receive
  - We must be able to detect when this occurs!
  - Why?
  - Who needs to detect it? (links, routers, OSs, or apps?)
- Basic approach: add redundant data
  - Error detection codes allow errors to be recognized
  - Error correction codes allow errors to be repaired too

Motivating Example

- A simple error detection scheme:
  - Just send two copies. Differences imply errors.
- Question: Can we do any better?
  - With less overhead
  - Catch more kinds of errors
- Answer: Yes – stronger protection with fewer bits
  - But we can't catch all inadvertent errors, nor malicious ones
- We will look at basic block codes
  - K bits in, N bits out is a (N,K) code
  - Simple, memoryless mapping

Detection vs. Correction

- Two strategies to correct errors:
  - Detect and retransmit, or Automatic Repeat reQuest. (ARQ)
  - Error correcting codes, or Forward Error Correction (FEC)
- Satellites, real-time media tend to use error correction
- Retransmissions typically at higher levels (Network+)
- Question: Which should we choose?

Retransmissions vs. FEC

- The better option depends on the kind of errors and the cost of recovery
- Example: Message with 1000 bits, Prob(bit error) 0.001
  - Case 1: random errors
  - Case 2: bursts of 1000 errors
  - Case 3: real-time application (teleconference)

The Hamming Distance

- Errors must not turn one valid codeword into another valid codeword, or we cannot detect/correct them.
- Hamming distance of a code is the smallest number of bit differences that turn any one codeword into another
  - e.g, code 000 for 0, 111 for 1, Hamming distance is 3
- For code with distance d+1:
  - d errors can be detected, e.g, 001, 010, 110, 101, 011
- For code with distance 2d+1:
  - d errors can be corrected, e.g., 001 → 000

Parity

- Start with n bits and add another so that the total number of 1s is even (even parity)
  - e.g. 0110010 → 01100101
  - Easy to compute as XOR of all input bits
- Will detect an odd number of bit errors
  - But not an even number
- Does not correct any errors

2D Parity

- Add parity row/column to array of bits
- Detects all 1, 2, 3 bit errors, and many errors with >3 bits.
- Corrects all 1 bit errors

0101001	1
1101001	0
1011110	1
0001110	1
0110100	1
1011111	0
1111011	0

Checksums

- Used in Internet protocols (IP, ICMP, TCP, UDP)
- Basic Idea: Add up the data and send it along with sum
- Algorithm:
  - checksum is the 1s complement of the 1s complement sum of the data interpreted 16 bits at a time (for 16-bit TCP/UDP checksum)
- 1s complement: flip all bits to make number negative
  - Consequence: adding requires carryout to be added back

CRCs (Cyclic Redundancy Check)

- Stronger protection than checksums
  - Used widely in practice, e.g., Ethernet CRC-32
  - Implemented in hardware (XORs and shifts)
- Algorithm: Given n bits of data, generate a k bit check sequence that gives a combined n + k bits that are divisible by a chosen divisor C(x)
- Based on mathematics of finite fields
  - "numbers" correspond to polynomials, use modulo arithmetic
  - e.g, interpret 10011010 as $x^7 + x^4 + x^3 + x^1$

How is C(x) Chosen?

- Mathematical properties:
  - All 1-bit errors if non-zero x^k and x^0 terms
  - All 2-bit errors if C(x) has a factor with at least three terms
  - Any odd number of errors if C(x) has (x + 1) as a factor
  - Any burst error < k bits
- There are standardized polynomials of different degrees that are known to catch many errors
  - Ethernet CRC-32: 100000100110000010001110110110111

Reed-Solomon / BCH Codes

- Developed to protect data on magnetic disks
- Used for CDs and cable modems too
- Property: 2t redundant bits can correct <= t errors
- Mathematics somewhat more involved …

Part 2: Key Concepts

- Redundant bits are added to messages to protect against transmission errors.
- Two recovery strategies are retransmissions (ARQ) and error correcting codes (FEC)
- The Hamming distance tells us how much error can safely be tolerated.