Lecture 3: Modulation & Layering

CSE 123: Computer Networks
Alex C. Snoeren

HW 1 out Today, due 10/09
Lecture 3 Overview

- Encoding schemes
  - Shannon’s Law and Nyquist Limit
  - Clock recovery
  - Manchester, NRZ, NRZI, etc.

- Layering
  - Services provided
Measures of success

- **How fast?**
  - **Bandwidth** measured in bits per second
  - Often talk about KBps or Mbps – Bytes vs bits

- **How long was the wait?**
  - **Delay** (one-way or round trip) measured in seconds

- **How efficiently?**
  - **Overhead** measured in bits or seconds or cycles or…

- **Any mistakes?**
  - **Error rate** measured in terms of probability of flipped bit
How long to send a message?

- Transmit time $T = \frac{M}{R} + D$
  - 10 Mbps Ethernet LAN ($M=1$KB)
    - $\frac{M}{R}=1$ms, $D \approx 5$us
  - 155 Mbps cross country ATM link ($M=1$KB)
    - $\frac{M}{R} = 50$us, $D \approx 40$-100ms

- Where are the bits in the mean time?
  - In transit inside the network

- $R*D$ is called the **bandwidth delay product**
  - How many bits can be “stored” be stored in transit
  - Colloquially, we say “fill the pipe”
Inter-symbol Interference

- Band-limited channels cannot respond faster than some maximum frequency $f$
  - Channel takes some time to settle

- Attempting to signal too fast will mix symbols
  - Previous symbol still “settling in”
  - Mix (add/subtract) adjacent symbols
  - Leads to inter-symbol interference (ISI)

- OK, so just how fast can we send symbols?
In a channel band-limited to $f$, we can send at maximum symbol (baud) rate of $2f$ without ISI.
Multiple Bits per Symbol

- OK, but why not send multiple bits per symbol
  - E.g., multiple voltage levels instead of just high/low
  - Four levels gets you two bits, $\log L$ in general
  - Could we define an infinite number of levels?

- Channel noise limits bit density
  - Intuitively, need level separation
  - Only get $\log(S/2N)$ bits per symbol

- Can combine this observation with Nyquist
  - $C < 2B \log(S/2N)$ in a perfect channel, but…
Noise Matters: Shannon’s Law

- Shannon considered noisy channels and derived

\[ C = B \log (1 + S/N) \]

- Gives us an upper bound on any channel’s performance regardless of signaling scheme

- Old school modems approached this limit
  - B = 3000Hz, S/N = 30dB = 1000
  - C = 3000 x log(1001) =~ 30kbps
  - 28.8Kbps, anyone?
Sampling at the Receiver

- Need to determine correct sampling frequency
  - Signal could have multiple interpretations

Which of these is correct?

CSE 123 – Lecture 3: Modulation & Layering
Nyquist Revisited

- Sampling at the correct rate \((2f)\) yields actual signal
  - Always assume lowest-frequency wave that fits samples

- Sampling too slowly yields aliases
The Importance of Phase

- Need to determine when to START sampling, too

![Diagram showing input and output signals with ideal sampling points at the receiver.](image)
Clock Recovery

- Using a training sequence to get receiver lined up
  - Send a few, known initial training bits
  - Adds inefficiency: only $m$ data bits out of $n$ transmitted

- Need to combat clock drift as signal proceeds
  - Use transitions to keep clocks synched up

- Question is, how often do we do this?
  - Quick and dirty every time: asynchronous coding
  - Spend a lot of effort to get it right, but amortize over lots of data: synchronous coding
Asynchronous Coding

- Encode several bits (e.g. 7) together with a leading “start bit” and trailing “stop bit”
- Data can be sent at any time
- Start bit transition kicks of sampling intervals
  - Can only run for a short while before drifting
Example: RS232 serial lines

- Uses two voltage levels (+15V, -15V), to encode single bit binary symbols
- Needs long idle time – limited transmit rate
Synchronous Coding

- Asynchronous receiver phase locks each symbol
  - Takes time, limiting transmission rates

- So, start symbols need to be extra slow
  - Need to fire up the clock, which takes time

- Instead, let’s do this training once, then just keep sync
  - Need to continually adjust clock as signal arrives
  - Ever hear of Phase Lock Loops (PLLs) ?

- Basic idea is to use transitions to lock in
Non-Return to Zero (NRZ)

- **Signal to Data**
  - High $\Rightarrow$ 1
  - Low $\Rightarrow$ 0

- **Comments**
  - Transitions maintain clock synchronization
  - Long strings of 0s confused with no signal
  - Long strings of 1s causes *baseline wander*
    - We use average signal level to infer high vs low
  - Both inhibit clock recovery

![NRZ Signal Example](image)

**Bits:** 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

**NRZ Signal:**

*Courtesy Robin Kravets*
Non-Return to Zero Inverted (NRZI)

- **Signal to Data**
  - Transition $\Rightarrow 1$
  - Maintain $\Rightarrow 0$

- **Comments**
  - Solves series of 1s, but not 0s

Bits: 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

NRZ

NRZI

CSE 123 – Lecture 3: Modulation & Layering

Courtesy Robin Kravets
Manchester Encoding

- **Signal to Data**
  - XOR NRZ data with sender's clock signal
  - High to low transition ⇒ 1
  - Low to high transition ⇒ 0

- **Comments**
  - Solves clock recovery problem
  - Only 50% efficient (½ bit per transition)
  - Still need preamble (typically 0101010101… trailing 11 in Ethernet)

![Manchester Encoding Diagram](image)
4B/5B (100Mbps Ethernet)

- Goal: address inefficiency of Manchester encoding, while avoiding long periods of low signals
- Solution:
  - Use five bits to encode every sequence of four bits
  - No 5 bit code has more than one leading 0 and two trailing 0’s
  - Use NRZI to encode the 5 bit codes
  - Efficiency is 80%

<table>
<thead>
<tr>
<th>4-bit</th>
<th>5-bit</th>
<th>4-bit</th>
<th>5-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
<td>1000</td>
<td>10010</td>
</tr>
<tr>
<td>0001</td>
<td>01001</td>
<td>1001</td>
<td>10011</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
<td>1010</td>
<td>10110</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
<td>1011</td>
<td>10111</td>
</tr>
<tr>
<td>0100</td>
<td>01010</td>
<td>1100</td>
<td>11010</td>
</tr>
<tr>
<td>0101</td>
<td>01011</td>
<td>1101</td>
<td>11011</td>
</tr>
<tr>
<td>0110</td>
<td>01110</td>
<td>1110</td>
<td>11100</td>
</tr>
<tr>
<td>0111</td>
<td>01111</td>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>

CSE 123 – Lecture 3: Modulation & Layering
Layering: A Modular Approach

- Sub-divide the problem
  - Each layer relies on services from layer below
  - Each layer exports services to layer above

- Interface between layers defines interaction
  - Hides implementation details
  - Layers can change without disturbing other layers

- Interface among peers in a layer is a protocol
  - If peers speak same protocol, they can interoperate
Protocol Standardization

- Communicating hosts speaking the same protocol
  - Standardization to enable multiple implementations
  - Or, the same folks have to write all the software

- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
  - Produces “Request For Comments” (RFCs)
    - Rough consensus and running code
    - After enough time passes, promoted to Internet Standards

- Other standards bodies exist
  - ISO, ITU, IEEE, etc.
TCP/IP Protocol Stack

Application Layer
- HTTP

Transport Layer
- TCP

Network Layer
- IP

Link Layer
- Ethernet interface
- SONET interface

CSE 123 – Lecture 3: Modulation & Layering
Encapsulation

HTTP

TCP

IP

Ethernet interface

Payload

Headers

HTTP

TCP

IP

Ethernet interface

CSE 123 – Lecture 3: Modulation & Layering
Internet Protocol Suite

The Hourglass Model

Applications
Transport
Data Link
Physical

“Thin Waist”

FTP
HTTP
NV
TFTP
TCP
UDP

IP

NET_1
NET_2
...
NET_n

CSE 123 – Lecture 3: Modulation & Layering
Physical layer

2.4Ghz Radio
DS/FH Radio
(1-11Mbps)

802.11b Wireless Access Point

Cat5 Cable (4 wires)
100Base TX Ethernet
100Mbps

Ethernet switch/router

To campus backbone

62.5/125um 850nm MMF
1000BaseSX Ethernet
1000Mbps

CSE 123 – Lecture 3: Modulation & Layering
Link Layer (e.g. Ethernet)

- Break message into frames
- Media Access Control (MAC)
  - Can I send now? Can I send now?
- Send frame
Connecting links

- **Routers/Switches**: moves bits between links
  - *Circuit switching*: guaranteed channel for a session (Telephone system)
  - *Packet switching*: statistical multiplexing of independent pieces of data (Internet)
For Next Class

- Read 2.3
- Get started on Homework 1!