CS434/534: Topics in Networked (Networking) Systems

Wireless Foundation:
Inter-Symbol Interference (ISI):
Wave Shaping, Equalization, OFDM

Yang (Richard) Yang
Computer Science Department
Yale University
208A Watson
Email: yry@cs.yale.edu

http://zoo.cs.yale.edu/classes/cs434/
Outline

- Admin. and recap
- Handling ISI
  - symbol wave shaping
  - equalization
  - OFDM
Feedback on coverage direction next week
- Continue to MAC layer (multi-hop wireless networking)
- Switch to App layer (basic Android), location services and then back to wireless networking

PS2 on wireless to be posted by tomorrow

Please see potential projects
Recap: Main Story of Flat Fading

- Communication over a wireless channel has poor performance due to significant probability that channel is in a deep fade, or has interference.

- Reliability is increased by using diversity: more resolvable signal paths that fade independently.
  - Time diversity: send same info (or coded version) at different instances of time.
  - Space diversity: send/receive same info at different locations.
  - Frequency diversity: send info at different frequencies.
Recap: Direct Sequence Spread Spectrum (DSSS)

- Basic idea: increase signaling function alternating rate to expand frequency spectrum (explores frequency in parallel)

- $f_c$: carrier freq. $R_b$: freq. of data
- $10\text{dB} = 10$; $20\text{dB} = 100$

$B_s$: num. of bits in the chip * $B_b$
Outline

- Admin. and recap
- Inter-Symbol Interference (ISI)
ISI

- ISI happens when the signaling for one symbol leaks into that of another symbol.

Why does ISI happen?
- Band limit produced ISI
- Multipath produced ISI
Outline

- Admin. and recap
- Inter-Symbol Interference (ISI)
  - Bandlimit produced ISI
Spectrum of BPSK Rectangular Pulse

- BPSK: \( g(t) = x(t) \cos(2\pi f_c t) \), for \( t \) in \([0, T]\)
- Since multiplying \( \cos() \) is just to shift frequency, consider the baseband, which is a rectangular pulse
- Fourier transform of a rectangular pulse spans the whole frequency range
  - This may not be acceptable in real deployment

http://mysite.du.edu/~etuttle/electron/elect6.htm
Band Limit Signaling

- Use band limited signaling functions
- Design 1: sinc

\[ W = \frac{1}{2T} \]
'sinc'-like Signaling

Effect: signals spread across symbols.
Received Signal Representation

\[ x(t) = a_0 g(t) + a_1 g(t - T) + a_2 g(t - 2T) + \ldots + w(t) \]
\[ x(t) = a_0 g(t) + a_1 g(t - T) + a_2 g(t - 2T) + \ldots + w(t) \]

**Matched Filter Decoding**

- Suppose matched filter \( h(t) \)

\[
y(t) = [a_0 g(t) + a_1 g(t - T) + a_2 g(t - 2T) + \ldots] * h(t)
\]

\[
= a_0 g(t) * h(t) + a_1 g(t - T) * h(t) + a_2 g(t - 2T) * h(t) + \ldots
\]

where \( g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau) h(t - \tau) d\tau = \int_{-\infty}^{\infty} g(t - \tau) h(\tau) d\tau \)
Impact on Matched Filter Decoding

- Define

\[ p(t) = g(t) \ast h(t) = \int_{-\infty}^{\infty} g(\tau) h(t-\tau) d\tau = \int_{-\infty}^{\infty} g(t-\tau) h(\tau) d\tau \]

\[ y(t) = a_0 p(t-0T) + a_1 p(t-T) + a_2 p(t-2T) + ... \]

- Recall that we make decisions at iT instances of time:

\[ y_i = y(iT) = a_i p(0) + \sum_{n \neq i} a_n p((i-n)T) + w_i \]

- Define \( p_k = p(kT) \)

\[ y_i = y(iT) = a_i p_0 + \sum_{n \neq i} a_n p_{i-n} + w_i \]

Question: what is a condition for no ISI?
Folded Spectrum Design

- Define no ISI as only $p_0$ is non-zero.
- Objective: how to design $p(t) = g(t) \ast h(t)$ to satisfy no ISI?

A design of $p(t)$ has no ISI iff folded spectrum is flat:

$$P_\Sigma(f) \equiv \frac{1}{T} \sum_{n=-\infty}^{\infty} P(f + n/T) = p_0$$
Folded Spectrum Design

\[ P_\Sigma(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} P(f + n/T) \]

\[ P_k = \int_{-\infty}^{\infty} P(f)e^{j2\pi fkT} df \]

\[ = \sum_{(2n+1)/2T}^{\infty} \int_{(2n-1)/2T}^{\infty} P(f)e^{j2\pi fkT} df \quad f' = f - n/T \]

\[ = \sum_{-1/2T}^{1/2T} \int_{-1/2T}^{1/2T} P(f' + n/T)e^{j2\pi k(f' + n/T)T} df' \]

\[ = \int_{-1/2T}^{1/2T} e^{j2\pi kf'T} \sum_{-\infty}^{\infty} P(f' + n/T) df' \quad (1) \]

\[ = T \int_{-1/2T}^{1/2T} e^{j2\pi kf'T} P_\Sigma(f') df' \]
Folded Spectrum Design: Sufficiency

\[ p_k = T \int_{-1/2T}^{1/2T} e^{j2\pi kf'T} P_\Sigma(f') df' \]

\[ = p_0 T \int_{-1/2T}^{1/2T} e^{j2\pi kf'T} df' = \frac{\sin \pi k}{\pi k} p_0 \]
Folded Spectrum Design: Necessary

\[ p_k = T \int_{-1/2T}^{1/2T} e^{j2\pi k f'T} P_{\Sigma}(f') df' \]

\[ \Rightarrow \]

\[ p_k \text{ is Fourier series of } P_{\Sigma}(f') \]

\[ P_{\Sigma}(f') = \sum_{-\infty}^{\infty} p_k e^{j2\pi f k T} = p_0 \]
Flat Folded Spectrum

\[ P_{\Sigma}(f) \text{ flat} \]
Flat Folded Spectrum: Raised Cosine

[Graph showing the frequency domain and time domain representations of raised cosine filters with different beta values]

- beta called rolloff factor or excessive bw
- smaller beta, lower excessive bw, but slower time-domain dying down

From \( P() \) to \( G() \)

- Recall

\[
p(t) = g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau) h(t - \tau) \, d\tau = \int_{-\infty}^{\infty} g(t - \tau) h(\tau) \, d\tau
\]

- Assume \( g(t) \) symmetric, \( h(t) = g(-t) = g(t) \)

\[
P(f) = G^2(f)
\]

\[
G(f) = \sqrt{P(f)}
\]
Root Raised Cosine

functional blocks in their PHY components. These functional blocks are pipelined with one another. Data are streamed through these blocks sequentially, but with different data types and sizes. As illustrated in Figure 1, different blocks may consume or produce different types of data in different rates arranged in small data blocks. For example, in 802.11b, the scrambler may consume and produce one bit, while DQPSK modulation maps each two-bit data block onto a complex symbol which uses two 16-bit numbers to represent the in-phase and quadrature (I/Q) components.

Each PHY block performs a fixed amount of computation on every transmitted or received bit. When the data rate is high, e.g., 11Mbps for 802.11b and 54Mbps for 802.11a/g, PHY processing blocks consume a significant amount of computational power. Based on the model in [19], we estimate that a direct implementation of 802.11b may require 10Gops while 802.11a/g needs at least 40Gops. These requirements are very demanding for software processing in GPPs.

PHY processing blocks directly operate on the digital waveforms after modulation on the transmitter side and before demodulation on the receiver side. Therefore, high-throughput interfaces are needed to connect these processing blocks as well as to connect the PHY and radio front-end. The required throughput linearly scales with the bandwidth of the baseband signal. For example, the channel bandwidth is 20MHz in 802.11a. It requires a data rate of at least 20M complex samples per second to represent the waveform [14]. These complex samples normally require 16-bit quantization for both I and Q components to provide sufficient fidelity, translating into 32 bits per sample, or 640Mbps for the full 20MHz channel. Over-sampling, a technique widely used for better performance [12], doubles the requirement to 1.28Gbps to move data between the RF front-end and PHY blocks for one 802.11a channel.

2.2 Wireless MAC

The wireless channel is a resource shared by all transceivers operating on the same spectrum. As simultaneously transmitting neighbors may interfere with each other, various MAC protocols have been developed to coordinate their transmissions in wireless networks to avoid collisions.

Most modern MAC protocols, such as 802.11, require timely responses to critical events. For example, 802.11 adopts a CSMA (Carrier-Sense Multiple Access) MAC protocol to coordinate transmissions [7]. Transmitters are required to sense the channel before starting their transmission, and channel access is only allowed when no energy is sensed, i.e., the channel is free. The latency between sense and access should be as small as possible. Otherwise, the sensing result could be outdated and inaccurate. Another example is the link-layer retransmission mechanisms in wireless protocols, which may require an immediate acknowledgement (ACK) to be returned in a limited time window.

Commercial standards like IEEE 802.11 mandate a response latency within tens of microseconds, which is challenging to achieve in software on a general purpose PC with a general purpose OS.

2.3 Software Radio Requirements

Given the above discussion, we summarize the requirements for implementing a software radio system on a general PC platform:

- High system throughput. The interfaces between the radio front-end and PHY as well as between some PHY processing blocks must possess sufficiently high...
Outline

- Admin. and recap
- Inter-Symbol Interference (ISI)
  - Bandlimit ISI
  - Multipath ISI
Multipath ISI
Multipath ISI

\[ h(0) \times (3) \]

\[ y_3 \]
Multipath ISI
Multipath ISI
Recall: Representation of Wireless Channels

In the general case, received signal at time $m$ is $y[m]$, $h_l[m]$ is the strength of the $l$-th tap, $w[m]$ is the background noise:

$$y[m] = \sum_{\ell} h_{\ell}[m] x[m - \ell] + w[m]$$
Visualizing ISI

\[ y[m] = \sum_{\ell} h_\ell[m] x[m - \ell] + w[m] \]
ISI Problem Formulation

The problem: given received $y[m]$, $m = 1, \ldots, L+2$, where $L$ is frame size and assume 3 delay taps (it is easy to generalize to $D$ taps):


$\ldots$

$y[L] = x[L] h_0 + x[L-1] h_1 + x[L-2] h_2 + w[L]$

$y[L+1] = x[L] h_1 + x[L-1] h_2 + w[L+1]$

$y[L+2] = x[L] h_2 + w[L+2]$

determine $x[1]$, $x[2]$, $\ldots$ $x[L]$
ISI Equalization:
Given y, what is x?

\[
\begin{align*}
&\vdots \\
y[L+2] &= x[L] h2 + w[L+2]
\end{align*}
\]
Solution Technique

- **Maximum likelihood detection:**
  - if the transmitted sequence is \( x[1], \ldots, x[L] \), then there is a likelihood we observe \( y[1], y[2], \ldots, y[L+2] \)
  - we choose the \( x \) sequence such that the likelihood of observing \( y \) is the largest

\[
\begin{align*}
&\quad\vdots \\
y[L] &= x[L] h_0 + x[L-1] h_1 + x[L-2] h_2 + w[L] \\
y[L+1] &= x[L] h_1 + x[L-1] h_2 + w[L+1] \\
y[L+2] &= x[L] h_2 + w[L+2]
\end{align*}
\]
Likelihood

- For given sequence $x[1], x[2], \ldots, x[L]$.
- Assume white noise, i.e., prob. $w = z$ is

$$f(z) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{z^2}{2\sigma^2}}$$

- What is the likelihood (prob.) of observing $y[1]$?
  - it is the prob. of noise being $w[1] = y[1] - x[1] h_0$

$$\frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{w[1]^2}{2\sigma^2}}$$

$$\ldots$$
The likelihood of observing $y[2]$
- it is the prob. of noise being $w[2] = y[2] - x[2]h0 - x[1]h1$, which is

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

- The overall likelihood of observing the whole $y$ sequence ($y[1], \ldots, y[L+2]$) is the product of the preceding probabilities
One Technique: Enumeration

foreach sequence \((x[1], \ldots, x[L])\)

compute the likelihood of observing the \(y\) sequence

pick the \(x\) sequence with the highest likelihood

\[
\begin{align*}
\end{align*}
\]

Question: what is the computational complexity?
Viterbi Algorithm

- Objective: avoid the enumeration of the $x$ sequences

- Key observation: the memory (state) of the wireless channel is only 3 (or generally $D$ for $D$ taps)

- Let $s[0], s[1], \ldots$ be the states of the channel as symbols are transmitted
  - $s[0]$: initial state---empty
  - $s[1]$: $x[1]$ is transmitted, two possibilities: 0, or 1
  - $s[2]$: $x[2]$ is transmitted, four possibilities: 00, 01, 10, 11
  - $s[3]$: $x[3]$ is transmitted, eight possibilities: 000, 001, \ldots, 111
  - $s[4]$: $x[4]$ is transmitted, eight possibilities: 000, 001, \ldots, 111

- We can construct a state transition diagram

- If we know the $x$ sequence we can construct $s$, and vice versa


$x[1]=0$
- $x[2]=0$  $x[3]=0$  $000$
- $x[1]=1$

$x[1]=1$
- $x[2]=0$  $x[3]=0$  $010$

$x[2]=0$
- $x[2]=0$  $x[3]=0$  $100$

$x[2]=1$
- $x[3]=0$  $110$
- $x[3]=1$  $111$

Viterbi Algorithm

- Each path on the state-transition diagram corresponds to a \( x \) sequence
  - each edge has a probability
  - the product of the probabilities on the edges of a path corresponds to the likelihood that we observe \( y \) if \( x \) is the sequence sent

- Then the problem becomes identifying the path with the largest product of probabilities

- If we take \(-\log\) of the probability of each edge, the problem becomes identifying the shortest path problem!
Viterbi Algorithm: Summary

- Invented in 1967
- Utilized in CDMA, GSM, 802.11, Dial-up modem, and deep space communications

- Also commonly used in
  - speech recognition,
  - computational linguistics, and
  - bioinformatics

Original paper: Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, April 1967
http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1054010
Problems of Viterbi for Multipath ISI

- Its complexity grows exponentially with $D$, where $D$ is the number of multipaths taps relative to the symbol time.

- If we have a high symbol rate, then $D$ can be large, and we need complex receivers.
## CPU Load

<table>
<thead>
<tr>
<th>In (bit)</th>
<th>Out (bit)</th>
<th>Compute Mcyles/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scramble</td>
<td>11Mbps</td>
<td>8</td>
</tr>
<tr>
<td>Descramble</td>
<td>11Mbps</td>
<td>8</td>
</tr>
<tr>
<td>Mapping and Spreading</td>
<td>2Mbps, DQPSK</td>
<td>8</td>
</tr>
<tr>
<td>CCK modulator</td>
<td>5Mbps, CCK</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>11Mbps, CCK</td>
<td>8</td>
</tr>
<tr>
<td>FIR Filter</td>
<td>16-bit I/Q, 37 taps, 22MSps</td>
<td>16<em>2</em>4</td>
</tr>
<tr>
<td>Decimation</td>
<td>16-bit I/Q, 4x Oversample</td>
<td>16<em>2</em>4*4</td>
</tr>
</tbody>
</table>

### IEEE 802.11a

<table>
<thead>
<tr>
<th>In (bit)</th>
<th>Out (bit)</th>
<th>Compute Mcyles/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT/IFFT</td>
<td>64 points</td>
<td>64<em>16</em>2</td>
</tr>
<tr>
<td>Conv. Encoder</td>
<td>24Mbps, 1/2 rate</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>48Mbps, 2/3 rate</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>54Mbps, 3/4 rate</td>
<td>24</td>
</tr>
<tr>
<td>Viterbi</td>
<td>24Mbps, 1/2 rate</td>
<td>8*16</td>
</tr>
<tr>
<td></td>
<td>48Mbps, 2/3 rate</td>
<td>8*24</td>
</tr>
<tr>
<td></td>
<td>54Mbps, 3/4 rate</td>
<td>8*32</td>
</tr>
<tr>
<td>Soft demapper</td>
<td>24Mbps, QAM 16</td>
<td>16*2</td>
</tr>
<tr>
<td></td>
<td>54Mbps, QAM 64</td>
<td>16*2</td>
</tr>
<tr>
<td>Scramble &amp; Descramble</td>
<td>54Mbps</td>
<td>8</td>
</tr>
</tbody>
</table>

Recap

- Inter-Symbol Interference (ISI)
  - Handle band limit ISI
  - Handle multipath ISI
    - Viterbi
      - problems: Its complexity grows exponentially with $D$ (the number of multipaths taps relative to the symbol time)
      - Q: how to reduce $D$?
OFDM: Basic Idea

- Uses multiple carriers modulation (MCM)
  - each carrier (called a subcarrier) uses a low symbol rate
    - for N parallel subcarriers, the symbol time can be N times longer
  - spread symbols across multiple subcarriers
    - also gains frequency diversity
Benefit of Symbol Rate on ISI
Multiple Carrier Modulation
Multiple Carrier Modulation (MCM): Problem

- Despite wave shaping, there can be leak from one subcarrier to another subcarrier

```
  i
  |
  v

  j
```

- Conventional design: guard bands to avoid interference among subcarriers

```
Band 1 | Guard band | Band 2 | Guard band | Band 3
```

- Guard band wastes spectrum
**Objective: Avoid subcarrier interference**

- Interference of subcarrier i on subcarrier j

![Diagram of subcarrier interference](image)

- Assume no pulse wave shaping, matched filter

\[
\int_0^T \sin(2\pi f_i t + \phi_i) \sin(2\pi f_j t + \phi_j) dt
\]

\[
= \frac{1}{2} \int_0^T \cos[2\pi(f_i - f_j)t + \phi_i - \phi_j] + \cos[2\pi(f_i + f_j)t + \phi_i + \phi_j] dt
\]

**Condition for the interference to be always 0?**
Objective: Avoid subcarrier interference

if integer number of cycles in $[0, T]$

$$\int_{0}^{T} \cos[2\pi ft + \phi] dt = 0$$

# cycles in $T$ is $T * f \Rightarrow T * f = \text{integer}$

Symbol period
OFDM Key Idea: Orthogonal Subcarriers

- Each subcarrier frequency is chosen so that an integral number of cycles in a symbol period, i.e.,
  - subcarrier freq = k \( \frac{1}{T} \)

They do not need to have the same phase, so long integral number of cycles in symbol time T!