CS434/534: Topics in Networked (Networking) Systems

Network OS Abstraction: From Data to Function Store; Wireless Foundation: Frequency-Domain Analysis

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
- High-level datapath programming
- Network OS supporting programmable networks
  - overview
  - OpenDaylight
  - distributed network OS (Paxos, Raft)
  - from data store to function store
- Programmable wireless communications and networking
  - Background
Admin

- Hardcopies of PS1 can be turned into my mail box (208A) on the first floor
- A talk on mobile systems
  - Time: Tuesday 10:30 am
  - Location: AKW
Recap

- Distributed data store using Raft
- Programming complexities of data store
  - Programmers need to manually record data access and register subscriptions
  - Programmers need to handle cleanup of any changes made in previous execution
  - Programmers need to make sure that no data-change loops are triggered, to avoid instability
- From data store to function store
  - Function store API
    - `add(func, [meta]), remove(handler)`
  - Data access API
    - `read(xpath), update(xpath, op, val), test(xpath, booleanExpr, ??)`
Assume exec’d the seq f1-f6. Then a changes. How to recover? Which functions must re-execute? Which do not need re-exec?
UpdateAlg1(a)

\[ F_{\text{update}} = a.\text{DEPENDENCY\_CHANGE} \]
\[ F_{\text{fixed}} = \text{executed before } F_{\text{update}} \text{ in total order} \]

restore2( F_{\text{fixed}} )

while ( ! F_{\text{update}}.empty() ) {
    pick f as a root in F_{\text{update}}
    \[ F_{\text{update}} -= f; F_{\text{fixed}} += f \]
    execute f
    add functions: not in F_{\text{fixed}} \wedge \text{depends on f updates}
}

From API to Realization: Design 1
Is UpdateAlg1 correct?
Function Store

- There are multiple directions where one can explore:
  - How to best utilize test()? 
  - How to best utilize past computation to learn dependency (e.g., learning)? 
  - How to utilize multi-cores in recovery? 
  - Can we distribute the functions at multiple servers (e.g., running Raft)? 

- A good topic as a course project.
Roadmap

- High-level networking system programming
- Networking operating system
  - overview
  - OpenDaylight
  - distributed networking OS (Paxos, Raft)
  - from data store to function store
- Programmable wireless communications and networking
  - background
  - software defined wireless networking systems
- Network function virtualization
Recap: Wireless and Mobile Computing

- Driven by infrastructure and device technology
  - global infrastructures
  - device miniaturization and capabilities
  - software development platforms

- Challenges:
  - wireless channel: unreliable, open access
  - mobility
  - portability
  - changing environment
  - heterogeneity
Avoid a full course on wireless communications, but cover enough to understand the main issues and main emerging techniques.
Implementing Wireless Systems: From Hardware to Software

Radio architecture

Antenna

RF/IF

Baseband Processing

High Layer Processing

Hardware (ASIC)

Software (GPP)

ADC/DAC

Programmable Hardware

Network interface

Virtual network interface

Software Defined Radio

Hardware (ASIC)

Programmable Hardware

Software (GPP)

Hardware (ASIC)

Software (GPP)

Software Radio

Hardware (ASIC)

Software (GPP)

Software (GPP)
Overview of Wireless Communications/Networking

Sender:
- Bit stream
- Source coding
- Channel coding
- Modulation

Receiver:
- Bit stream
- Source decoding
- Channel decoding
- Demodulation

Analog signal
Wireless Example: Wireless: 802.11

![Diagram](image_url)

(a) IEEE 802.11b 2Mbps

(b) IEEE 802.11a/g 24Mbps

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 frond-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...
Basic Question: Why Not Send Digital Signal in Wireless Communications?

digital signal
Outline

- Admin and recap
- Programmable wireless communications and networking
  - background
    - Frequency domain analysis (Fourier series)
A periodic real function $g(t)$ on $[-\pi, \pi]$ can be decomposed as a set of harmonics (cos, sin):

$$g(t) = \sum_{k=0}^{\infty} \left[ a_k \cos(kt) + b_k \sin(kt) \right]$$

set $b_k = 0$
Fourier Series

\[ g(t) = \sum_{k=0}^{\infty} \left[ a_k \cos(kt) + b_k \sin(kt) \right] \]

\[
\int_{-\pi}^{\pi} g(t) \cos(kt) \, dt \\
= \int \sum_{m=0}^{\infty} \left[ a_m \cos(mt) + b_m \sin(mt) \right] \cos(kt) \, dt \\
= \int \sum_{m=0}^{\infty} \left[ a_m \cos(mt) \cos(kt) + b_m \sin(mt) \cos(kt) \right] \, dt
\]
**Fourier Series**

\[ \int_{-\pi}^{\pi} g(t) \cos(kt)\, dt \]

\[ = \int \sum_{m=-\infty}^{\infty} \left[ a_m \cos(mt) \cos(kt) + b_m \sin(mt) \cos(kt) \right] dt \]

\[ = \int \sum_{m=-\infty}^{\infty} \left[ \frac{1}{2} a_m \left( \cos(m-k)t + \cos(m+k)t \right) \right] dt \]

\[ = \int \sum_{m=-\infty}^{\infty} \left[ \frac{1}{2} b_n \left( \sin(m+k)t + \sin(m-k)t \right) \right] dt \]

\[ = \int \frac{1}{2} a_k \, dt = \pi a_k \]

\[ a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} g(t) \cos(kt)\, dt \quad k \geq 1 \]

\[ b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} g(t) \sin(kt)\, dt \quad k \geq 1 \]
Fourier Series: Example

\[ a_k = 0 \quad k \geq 1 \]

\[ b_k = 2 \frac{(-1)^{k+1}}{k} \quad k \geq 1 \]

A problem of the expression

\[ g(t) = \sum_{k=0}^{\infty} \left[ a_k \cos(kt) + b_k \sin(kt) \right] \]

It contains both \( \cos() \) and \( \sin() \) functions, and hence is somehow complex to manipulate.
Applying Euler’s formula

\[ e^{jkt} = \cos(kt) + j \sin(kt) \]

We have

\[ \cos(kt) = \frac{1}{2} [ e^{jkt} + e^{-jkt} ] \]

\[ \sin(kt) = \frac{j}{2} [ -e^{jkt} + e^{-jkt} ] \]
Fourier Series: Using Euler’s formula

\[ g(t) = \sum_{k=0}^{\infty} \left[ a_k \cos(kt) + b_k \sin(kt) \right] \]

\[ = \sum_{k=0}^{\infty} \frac{1}{2} \left[ a_k (e^{jkt} + e^{-jkt}) + jb_k (-e^{jkt} + e^{-jkt}) \right] \]

\[ = \sum_{k=0}^{\infty} \left[ \frac{1}{2} (a_k - jb_k) e^{jkt} + \frac{1}{2} (a_k + jb_k) e^{-jkt} \right] \]

\[ = \sum_{k=0}^{\infty} \left[ c_k e^{jkt} + c_{-k} e^{-jkt} \right] \]

\[ c_k = \frac{a_k - jb_k}{2} \]

\[ c_{-k} = \frac{a_k + jb_k}{2} \]
Making Sense of Complex Numbers

\[ Me^{j\phi} \]

\[ M[\cos(\phi) + j\sin(\phi)] \]

What is the effect of multiplying \( c \) by \( e^{j\pi/2} \)?

What is the effect of multiplying \( c \) by \( j \)?
Making Sense of Complex Numbers
Making Sense of Complex Numbers: Conjugate

\[ c^* = a - jb \]
Summary of Progress: Fourier Series of Real Function on \([-\pi, \pi]\)

\[ g(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk t} \]

\[ c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(t)e^{-jk t} \, dt \]

\[ C_k = C_{-k}^* \]
A periodic function $g(t)$ with period $T$ on $[a, a+T]$ can be decomposed as:

$$g(t) = \sum_{k=-\infty}^{\infty} G[k] e^{j2\pi \frac{k}{T} t}$$

$$G[k] = \frac{1}{T} \int_{a}^{a+T} g(t) e^{-j2\pi \frac{k}{T} t} dt$$
Defining Decomposition on $[0, 1]$

\[
g(t) = \sum_{k=-\infty}^{\infty} G[k] e^{j2\pi kt}
\]

\[
G[k] = \int_{0}^{1} g(t) e^{-j2\pi kt} \, dt
\]
Making Sense of $e^{j2\pi ft}$
Making Sense of $e^{j2\pi ft}$

$$g(t) = \sum_{k=-\infty}^{\infty} G[k]e^{j2\pi kt}$$
For those who are curious, we do not need it formally.

Problem: Fourier series for periodic function $g(t)$, what if $g(t)$ is not periodical?

Approach:
- Truncate $g(t)$ beyond $[-L/2, L/2]$ (i.e., set $= 0$) and then repeat to define $g^L(t)$.

$$g^L(t) = \sum_{k=-\infty}^{\infty} G^L[k] e^{j2\pi k t/L}$$

$$G^L[k] = \frac{1}{L} \int_{-L/2}^{L/2} g^L(t) e^{-j2\pi k t/L} dt$$
Fourier Transform

\[ G^L[k] = \frac{1}{L} \int_{-L/2}^{L/2} g^L(t) e^{-j2\pi \frac{k}{L} t} dt \]

Define \( f_k = \frac{k}{L} \) \( \Delta f = \frac{1}{L} \) \( \hat{G}(f_k) = \int_{-\infty}^{\infty} g^L(t) e^{-j2\pi f_k t} dt \)

\[ G^L[k] = \frac{1}{L} \int_{-L/2}^{L/2} g^L(t) e^{-j2\pi \frac{k}{L} t} dt \quad \Rightarrow \quad G^L[k] = \Delta f \hat{G}(f_k) \]

\[ g^L(t) = \sum_{k=-\infty}^{\infty} \hat{G}(f_k) e^{j2\pi f_k t} \Delta f \approx \int \hat{G}(f) e^{j2\pi f t} df \]

Let L grow to infinity, we derive Fourier Transform:

\[ \hat{G}(f) = \int_{-\infty}^{\infty} g(t) e^{-j2\pi f t} dt \quad g(t) = \int_{-\infty}^{\infty} \hat{G}(f) e^{j2\pi f t} df \]
Fourier Series vs Fourier Transform

- Fourier series
  - For periodical functions, e.g., \([0, 1]\)

\[
g(t) = \sum_{k=-\infty}^{\infty} G[k]e^{j2\pi kt}
\]

\[
G[k] = \int_{0}^{1} g(t)e^{-j2\pi kt} \, dt
\]

- Fourier transform
  - For non periodical functions

\[
g(t) = \int_{-\infty}^{\infty} \hat{G}(f)e^{j2\pi ft} \, df
\]

\[
\hat{G}(f) = \int_{-\infty}^{\infty} g(t)e^{-j2\pi ft} \, dt
\]

Two Domain Representations

- Two representations: time domain; frequency domain
- Knowing one can recover the other
Example: Frequency Series of sine and cosine

\[ g(t) = \cos(2\pi k_0 t) \]

\[ \cos(2\pi k_0 t) = \frac{1}{2} \left[ e^{j2\pi k_0 t} + e^{-j2\pi k_0 t} \right] \]

\[ g(t) = \sin(2\pi k_0 t) \]

\[ \sin(2\pi k_0 t) = \frac{1}{2} \left[ -je^{j2\pi k_0 t} + je^{-j2\pi k_0 t} \right] \]

\[ g(t) = \sum_{k=-\infty}^{\infty} G[k] e^{j2\pi k t} \]

\[ G[k] = \int_{0}^{1} g(t) e^{-j2\pi k t} \, dt \]
Example: Frequency Series of sine and cosine
Example: Frequency Domain's View of Euler's Formula
Example: Frequency Domain’s View of Euler’s Formula
Example: Frequency Domain’s View of Euler’s Formula

\[ e^{j2\pi f_0 t} = \cos(2\pi f_0 t) + jsin(2\pi f_0 t) \]
From Integral to Computation

\[ g(t) = \sum_{k=-\infty}^{\infty} G[k] e^{j2\pi kt} \]

\[ G[k] = \int_{0}^{1} g(t) e^{-j2\pi kt} \, dt \]

\[ \approx \sum_{n=0}^{N-1} g\left(\frac{n}{N}\right) e^{-j2\pi \frac{n}{N}} \frac{1}{N} \]
Discrete Domain Analysis

- **FFT**: Transforming a sequence of numbers $x_0, x_1, \ldots, x_{N-1}$ to another sequence of numbers $X_0, X_1, \ldots, X_{N-1}$

\[
X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \ldots, N - 1
\]

- **Interpretation**: consider $x_0, x_1, \ldots, x_{N-1}$ as sampled values of a periodical function defined on $[0, 1]$

$\Rightarrow X_k$ is the coefficient (scaled by $N$) for $k$ Hz harmonics if the FFT $N$ samples span one sec
Discrete Domain Analysis

- **FFT:**
  \[ X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \quad k = 0, \ldots, N - 1 \]

- **Inverse DFFT**
  \[ x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N}kn} \quad n = 0, \ldots, N - 1. \]