Course outline

1. Basic protocols and projects
   - 1st three weeks of lectures
   - Team of ~4 students
   - Work closely with Project TAs (weekly project meetings)

2. Analytical methods for CC
   - Develop basic mathematical tools
   - Build congestion control theory from scratch
   - Illustrate process from systems \(\rightarrow\) models \(\rightarrow\) analysis \(\rightarrow\) insights & designs

2. Overview of special topics
   - Wireless networking
   - Security and privacy
Warning

These notes are not self-contained, probably not understandable, unless you also were in the lecture

They are supplement to not replacement for class attendance
Internet applications (2006)

Telephony

Music

TV & home theatre

Finding your way

Mail

Social networking

Library at your finger tip

Games

Cloud computing
Networking: 143, 144, 145

cs/ee/ids 143 Communication Networks
- How to send information from A to B

cs/ee/ids 144 Networks: Structure & Economics
- Applications on top of infrastructure

cs/ee 145 Projects in Networking
- Advanced projects – propose your own topic

Understanding both theory and practice of networks
This satisfies project requirement for CS major
143: Communication Networks
Classes

- You are expected to attend class.
- Goal is learning with minimal lecturing.
- Ideally, classes consist of discussions, students solving problems together.
- You are expected to use the material outside class; the class is interactive.
- Turn off anything with an on/off switch. Interact! Talk!

Source: Mani Chandy
Alumni Recommendation

The Caltech experience should give engineers skills in:

- Communication
  - Writing scientific papers, proposals, plans
  - Speaking to science, government, business audiences
- Working in teams
  - Project planning and management
  - Division of labor; leadership
  - Larger, global view of the world

Source: Mani Chandy
RSRG courses and research

RSRG courses

1. **Learn by doing**
   1. Emphasis on creativity, problem-solving, research
2. Integrating courses, SURF, internships, theses, research opportunities in *multiyear programs*
3. Close interaction with faculty in and out of class
4. Interactive classes: “The lecture is dead.” [ideal]

Source: Mani Chandy
Course website

http://courses.cms.caltech.edu/cs143/

All relevant information will be posted on website
Updated through the term
Chapter 1  The Internet

Text: Walrand & Parakh, 2010

Steven Low
CMS, EE, Caltech
1.1.3 ADDRESSING

Every computer or other host attached to the Internet has a unique address specified by a 32-bit string called its IP address, for Internet Protocol Address. The addresses are conventionally written in the form a.b.c.d where a, b, c, d are the decimal value of the four bytes. For instance, 169.229.60.32 corresponds to the four bytes 10101001.11100101.00111100.00100000.

1.1.4 ROUTING

Each router determines the next hop for the packet from the destination address. While advancing towards the destination, within a network under the control of a common administrator, the packets essentially follow the shortest path. The routers regularly compute these shortest paths and record them as routing tables. A routing table specifies the next hop for each destination address, as sketched in Figure 1.2.

The packets typically go through a set of networks that belong to different organizations. The routers select this set of networks according to rules that we discuss in the Routing chapter.

More precisely, the router consults a forwarding table that indicates the output port of the packet. However, this distinction is not essential.

- Hosts, routers, links
- IP addresses, domain name
- Packet, header/trailer
Basic mechanisms

Packet switching
- No dedicated resources
- Packets may follow different paths
- Packets may be lost, error, out-of-order

Addressing
- Globally unique IP address
- Domain name
- Mapping from DN to IP may change dynamically (DNS)
Basic mechanisms

Routing
- Destination based
- Routing decision may adapt to network condition, e.g., failure

Error & loss recovery
- Retransmission of lost or erroneous pkt
- Typically done end-to-end

Flow & congestion control
- End-to-end
- Implicit feedback

class project focus
Figure 1.2: The figure sketches a router with 32 input ports (link attachments) and 32 output ports. Packets contain their source and destination addresses. A routing table specifies, for each destination, the corresponding output port for the packet.

To simplify the routing tables, the network administrators assign IP addresses to hosts based on their location. For instance, router $R_1$ in Figure 1.1 sends all the packets with a destination address whose first byte has decimal value 18 to router $R_2$ and all the packets with a destination address whose first byte has decimal value 64 to router $R_3$. Instead of having one entry for every possible destination address, a router has one entry for a set of addresses with a common initial bit string, or prefix. If one could assign addresses so that all the destinations that share the same initial five bits were reachable from the same port of a 32-port router, then the routing table of the router would need only 32 entries of 5 bits: each entry would specify the initial five bits that correspond to each port. In practice, the assignment of addresses is not perfectly regular, but it nevertheless reduces considerably the size of routing tables. This arrangement is quite similar to the organization of telephone numbers into [country code, area code, zone, number]. For instance, the number 1 510 642 1529 corresponds to a telephone set in the US (1), in Berkeley (510), the zone of the Berkeley campus (642).

The general approach to exploit location-based addressing is to find the longest common initial string of bits (called prefix) in the addresses that are reached through the same next hop. This scheme, called Classless Interdomain Routing (or CIDR), enables to arrange the addresses into subgroups identified by prefixes in a flexible way. The main difference with the telephone numbering scheme is that, in CIDR, the length of the prefix is not predetermined, thus providing more flexibility.

As an illustration of longest prefix match routing, consider how router $R_2$ in Figure 1.1 selects where to send packets. A destination address that starts with the bits 000010101 matches the first 9 bits of the prefix 18.128 = 00010010'10000000 of output link L2 but only the first 8 bits of the prefix.
CIDR notation

Example (IPv4)

18.128.33.0/24 = \{ 18.128.33.0 – 18.128.33.255 \}
= subnet mask 255.255.255.0
#addresses = 2^{32-24} = 2^8

18.128.0.0/22 = \{ 18.128.0000 0000. 0000 0000 –
18.128.0000 0011. 1111 1111 \}
= \{ 18.128.0.0 – 18.128.3.255 \}
#addresses = 2^{32-22} = 2^{10}

CIDR = classless inter-domain routing
CIDR notation
CIDR notation

0.0.0.0/1
1.0.0.0/1

0101
1010
Address blocks may be nested!

e.g. 1.0.0.0/1 contains 1.1.0.0/3
### Forwarding decision

<table>
<thead>
<tr>
<th>Destination IP</th>
<th>Output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.128.33.0/24</td>
<td>1</td>
</tr>
<tr>
<td>18.128.0.0/22</td>
<td>2</td>
</tr>
<tr>
<td>.....</td>
<td>3</td>
</tr>
<tr>
<td>.....</td>
<td>4</td>
</tr>
</tbody>
</table>

Packet's destination IP  

- entry 1: 18.128.0010 0001.00000000
- entry 2: 18.128.0000 0000.00000000
- destination: 18.128.0010 0001.0000 0101
Forwarding decision

18.128.0.0/18

0
1

18.128.0.0/22

output link 2

18.128.0.0000 0000 0000/24

18.128.33.0/24

output link 1

18.128.0010 0001.0000 0000/24
Forwarding decision

18.128.0.0/18

0

18.128.0.0/18

1

18.128.0.0/22

output link 2

18.128.33.0/24

output link 1

Packet dest: 18.128.33.5
## Longest prefix matching

<table>
<thead>
<tr>
<th>Destination IP</th>
<th>Output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.128.33.0/24</td>
<td>1</td>
</tr>
<tr>
<td>18.128.0.0/22</td>
<td>2</td>
</tr>
<tr>
<td>....</td>
<td>3</td>
</tr>
<tr>
<td>....</td>
<td>4</td>
</tr>
</tbody>
</table>

Packet’s destination IP: 18.128.33.5  →  Output link 1

- entry 1: 18.128.0010 0000 0000 0000 0000
- entry 2: 18.128.0000 0000 0000 0000 0000
- destination: 18.128.0010 0001.0000 0101

It matches only the first address block
Longest prefix matching

<table>
<thead>
<tr>
<th>Destination IP</th>
<th>Output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.128.33.0/24</td>
<td>1</td>
</tr>
<tr>
<td>18.128.0.0/22</td>
<td>2</td>
</tr>
<tr>
<td>....</td>
<td>3</td>
</tr>
<tr>
<td>....</td>
<td>4</td>
</tr>
</tbody>
</table>

Packet’s destination IP: 18.128.3.5

Output link 2

entry 1: 18.128.0010 0001.0000 0000
entry 2: 18.128.0000 0000.0000 0000
destination: 18.128.0000 0011.0000 0101

It matches only the second address block
Longest prefix matching

<table>
<thead>
<tr>
<th>Destination IP</th>
<th>Output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.52.0/22</td>
<td>1</td>
</tr>
<tr>
<td>192.168.50.0/23</td>
<td>2</td>
</tr>
<tr>
<td>192.168.48.0/22</td>
<td>3</td>
</tr>
<tr>
<td>....</td>
<td>4</td>
</tr>
</tbody>
</table>

Packet’s destination IP: 192.168.51.15

entry 1: 192.168.0011 0100.0000 0000
entry 2: 192.168.0011 0010.0000 0000
entry 3: 192.168.0011 0000.0000 0000

destination: 192.168.0011 0011.0010 0101

Even though it matches both entries 2 and 3, entry 2 has a longer prefix. entry 3 “contains” entry 2.
Longest prefix matching

Subnets in routing tables may not be disjoint!
Route to most specific address block
How to construct routing table

<table>
<thead>
<tr>
<th>Destination IP</th>
<th>Output link</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.128.33.0/24</td>
<td>1</td>
</tr>
<tr>
<td>18.128.0.0/22</td>
<td>2</td>
</tr>
<tr>
<td>....</td>
<td>3</td>
</tr>
<tr>
<td>....</td>
<td>4</td>
</tr>
</tbody>
</table>

How do we make routing decisions? (Dijkstra, Bellman-Ford, BGP, ... Ch 5 Routing)
1. THE INTERNET

1.1.3 ADDRESSING

Every computer or other host attached to the Internet has a unique address specified by a 32-bit string called its **IP address**, for Internet Protocol Address. The addresses are conventionally written in the form \texttt{a.b.c.d} where \texttt{a, b, c, d} are the decimal value of the four bytes. For instance, \texttt{169.229.60.32} corresponds to the four bytes \texttt{10101001.11100101.00111100.00100000}.

1.1.4 ROUTING

Each router determines the next hop for the packet from the destination address. While advancing towards the destination, within a network under the control of a common administrator, the packets essentially follow the shortest path. The routers regularly compute these shortest paths and record them as **routing tables**. A routing table specifies the next hop for each destination address, as sketched in Figure 1.2.

The packets typically go through a set of networks that belong to different organizations. The routers select this set of networks according to rules that we discuss in the Routing chapter. More precisely, the router consults a **forwarding table** that indicates the output port of the packet. However, this distinction is not essential.

---

Recap

- Hosts, routers, links
- IP addresses, domain name
- Packet, header/trailer
Chapter 2  Principles

Text: Walrand & Parakh, 2010

Steven Low
CMS, EE, Caltech
Principles

Sharing
Metrics
Scalability
Layering
  ■ Application & technology independence
Application topology
Packet switching, statistical multiplexing
⇒ Queueing analysis

cf circuit switching in traditional telephone network
## Sharing

<table>
<thead>
<tr>
<th>Packet switching</th>
<th>Circuit switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>Traditional telephone nk</td>
</tr>
<tr>
<td>No dedicated resources</td>
<td>Dedicated resources</td>
</tr>
<tr>
<td>No circuit setup/tear down</td>
<td>Circuit setup/tear down</td>
</tr>
<tr>
<td>Queueing delay</td>
<td>No queueing delay</td>
</tr>
<tr>
<td>Loose performance guarantee</td>
<td>Stricter performance guarantee</td>
</tr>
<tr>
<td>More efficient for bursty (data) traffic</td>
<td>Better for fixed rate traffic</td>
</tr>
</tbody>
</table>
Metrics: speed

Link rate (bps)
- DSL e.g. 768kbps down, 256kbps up

Link bandwidth (Hz)
- Size of frequency band
- FM radio, wireless spectra, ...
- If link can transmit over [300Hz, 1MHz], its bandwidth is \(~1MHz\)

Link capacity (bps)
- Maximum link rate possible

\[ C = W \log_2(1 + SNR) \]
Metrics: speed

Link capacity example

- $\text{SNR} = 2^{20}$
- $W = 1\text{MHz}$
- $C = 10^6 \log_2(1+2^{20}) \text{ bit/s} = 20 \text{ Mbps}$

link “rate” is often called link “bandwidth” in networking literature
Metrics: speed

Throughput (bps)

- Bit rate actually achieved
- Example
  - MP3 file of 3MB transferred in 2 mins
  - Throughput = 3MB / 2mins = 200 kbps
- Generally less than link rate because of transmission and protocol overheads
  - Typically lower in wireless networking than in wireline

\[
\text{throughput} \leq \text{link rate} \leq \text{link capacity}
\]
Window Flow Control

- \( \sim W \) packets per RTT
- Lost packet detected by missing ACK
Metrics: delay

Delay and delay jitter (sec)

- Some applications care more about throughput, some about delay, some about delay jitter
Queueing system

random arrival process with rate \( \lambda \) pkts/s

\[ \text{avg delay } T = \frac{1}{\mu - \lambda} \]

random service time with average \( \frac{1}{\mu} \) s/pkt

M/M/1 queue

deterministic

\( \mu \leftarrow \lambda \rightarrow \text{delay} \)
Little’s law

Little’s law \( L = \lambda T \)

- Verifies directly for M/M/1, but holds much more generally
- Extremely useful because of its generality

\( \lambda \) pkts/s

\( \mu \)

avg delay \( T \)

avg queue length \( L \)

random arrival process with rate

random service time with average \( \frac{1}{\mu} \) s/pkt
Queueing system

random arrival process with rate $\lambda$ pkts/s

random service time with average $\frac{1}{\mu}$ s/pkt

$L = \lambda T$  \hspace{1cm} $T_q = T - \frac{1}{\mu}$

$L_q = \lambda T_q$  \hspace{1cm} $T_q$
M/M/1 queue

Poisson arrival process with rate $\lambda$ pkts/s

Exponential service time with average $\frac{1}{\mu}$ s/pkt

avg total delay $T = \frac{1}{\mu - \lambda}$

avg waiting time $T_q = T - \frac{1}{\mu} = \frac{\lambda}{\mu} \frac{1}{\mu - \lambda}$

avg # pkts in system $L = \frac{\lambda}{\mu - \lambda}$
Scalability

Location-based addressing

- Example: IP is location based
- Ethernet is not

Advantages
- Reduce routing table size
- Simpler routing operation

Disadvantages
- Makes mobility hard
- Makes security hard
Scalability

Hierarchical

- Two level
- Autonomous system
  - Same administrative and/or economic domain
  - Shortest path routing: OSPF, IS-IS
- Inter-domain
  - BGP

Both simplify routing
Scalability

Best effort service
- Simple TCP/IP layer

End-to-end principle
- Simple network, intelligent hosts
- Stateless routers
  - Packet carries its own state

Both simplify router $\rightarrow$ fast big inexpensive routers

New idea: software defined networking (SDN)
Scalability

Hierarchical naming

- Easier on human
- DNS translation

Layering

- Application & technology independence
Recap

Sharing
Metrics
Scalability
Layering
  - Application & technology independence
Application topology

Little’s law: informal proof
Protocol layering

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>how to provide products and services on network</td>
</tr>
<tr>
<td>transport</td>
<td>how to recover loss and error, and control congestion</td>
</tr>
<tr>
<td>network</td>
<td>how to route a packet through the network</td>
</tr>
<tr>
<td>link</td>
<td>how to share a common transmission medium (MAC)</td>
</tr>
<tr>
<td>physical</td>
<td>how to send a single bit from A to B</td>
</tr>
</tbody>
</table>
Protocol layering

Network mechanisms implemented as protocol stack

Each layer designed separately, evolves asynchronously

- **application**
  - Many control mechanisms...
- **transport**
  - Error control, congestion control (TCP)
- **network**
  - Routing (IP)
- **link**
  - Medium access control
- **physical**
  - Coding, transmission, synchronization
Protocols are critical, yet difficult, to understand and optimize.

- **Local algorithms**, distributed spatially and vertically \(\rightarrow\) global behavior
Protocol layering

<table>
<thead>
<tr>
<th></th>
<th>application</th>
<th>transport</th>
<th>network</th>
<th>link</th>
<th>physical</th>
</tr>
</thead>
</table>

Need systematic way to understand, design, and optimize

- Their interactions
- Resultant global behavior
Scalability

Hierarchical naming
- Easier on human
- DNS translation

Layering
- Application & technology independence

Applications
- Web
- Search
- Mail
- News
- Video
- Audio
- Friends

TCP
IP

Link technologies
- Ethernet
- 802.11
- 3G/4G
- ATM
- Optical
- Satellite
- Bluetooth
Applications

Application topology

- Client/server
- CDN
- P2P
- Cloud computing
- Social networking
Appendix

Little’s theorem

- Informal proof
Little’s law

\[ L = \lambda T \]

arrival rate:
\[ \lambda \text{ pkts/s} \]

capacity:
\[ \mu \text{ pkts/s (>}\lambda) \]

\[ L \text{ pkts} \]
**Little’s law**

\[ L_t = \alpha_t - \beta_t : \text{#packets in system at } t \]

\( \alpha_t \): \#packets arrived by \( t \)

\( \beta_t \): \#packets departed by \( t \)

\( T_i \): delay of packet \( i \)

\( L(\tau) \): number of packets in system at time \( \tau \).
Little’s law

\[ \int_0^t L(\tau) d\tau \]
Little’s law

\[ \sum_{i=1}^{\beta_i} T_i \leq \int_0^t L(\tau) d\tau \]
Little’s law

\[ \sum_{i=1}^{\beta_i} T_i \leq \int_0^t L(\tau) d\tau \leq \sum_{i=1}^{\alpha_i} T_i \]
Little’s law

\[ \sum_{i=1}^{\beta_t} T_i \leq \int_{0}^{t} L(\tau) d\tau \leq \sum_{i=1}^{\alpha_t} T_i \]

\[ \frac{\beta_t}{t} \frac{1}{\beta_t} \sum_{i=1}^{\beta_t} T_i \leq \frac{1}{t} \int_{0}^{t} L(\tau) d\tau \leq \frac{\alpha_t}{t} \frac{1}{\alpha_t} \sum_{i=1}^{\alpha_t} T_i \]
Little’s law

\[ \lambda \ T \leq L \leq \lambda \ T \]

\[ \uparrow \text{ as } t \to \infty \]

\[ \frac{\beta_t}{t} \sum_{i=1}^{\beta_t} T_i \leq \frac{1}{t} \int_0^t L(\tau) \, d\tau \leq \frac{\alpha_t}{t} \sum_{i=1}^{\alpha_t} T_i \]
Queueing system

Random arrival process with rate $\lambda$ pkts/s

Random service time with average $\frac{1}{\mu}$ s/pkt

Little’s law $L = \lambda T$

- Verifies directly for M/M/1, but holds much more generally
- Extremely useful because of its generality