15-441 Computer Networks

Lecture 1

Dave Eckhardt
Peter Steenkiste

(some slides courtesy of Hui Zhang)

Outline

- First things first
- Administrative overview
- Course non-goals
- Course goals
- Key problems
- Network performance concepts

First Things First

- **Please read Chapter 1 of the text**
  - To read ahead: most of Chapter 2, but ok to skip for now:
    - 2.5, Reliable Transmission
    - 2.8, Wireless
    - Also, don't kill yourself on 2.4, Error Detection
- **Please remind me to let you stretch**
  - I haven't taught an 80-minute class in 1.5 years

People

- **Professors**
  - Peter Steenkiste (www.cs/~prs)
  - Dave Eckhardt (www.cs/~davide)
- **Teaching assistants**
  - Mike Cui
  - Josh Hailpern
  - David Murray
  - [watch this space]
- **Course secretary**
  - Barbara Grandillo, Wean Hall 8018

Information Sources

- **Watch the course web page**
  - http://www.cs.cmu.edu/~441
  - ***We expect you to read the syllabus!***
  - Handouts, readings, ..
- **We expect you to read course bboards**
  - Official announcements
    - academic.cs.15-441.announce
  - Questions/answers
    - academic.cs.15-441

Information Sources

- **Textbook**
  - Peterson and Davie, *Computer Networks: A Systems Approach, 3rd Edition*, Morgan Kaufmann, 2003

Information Sources

- **~30 lectures**
- **~3 paper homeworks**
- **1-2 lab homeworks**
  - Illustrate networking concepts
- **Mid-term and final exams**
- **1 programming assignment**
  - How to use a network
- **2 programming projects**
  - How to build a network

Grading

Homeworks	**15%**
Three projects	**45%**
Midterm exam	**15%**
Final exam	**25%**

- **Deadline means deadline**
  - Deadline is 11:59 pm on the specified date

Policy on Collaboration

- **Working together is important**
  - Discuss course material in general terms
  - Talk over tough debugging problems
- **Parts of the course must be done individually**
  - Homeworks, midterm, final, 1st programming assignment
- **Projects are done by two-student teams**
  - Learn how to collaborate
  - But each student must understand the entire project!
- **Web page has the details**

Course Non-goals

- ~~**Learn how to configure a Cisco router**~~
  - That requires a class all by itself
  - Cisco teaches those classes
  - Our perspective will be broader
- ~~**Become "Internet Experts"**~~
  - The Internet will be our frequent ***motivating example***
  - Our perspective will be broader

Why *not* an "Internet class"?

- **Is there anything other than the Internet?**

  Philosophy final exam question:

  Define "Universe". Give two examples.

Why *not* an "Internet class"?

- **Is there anything other than the Internet?**

  Philosophy final exam question

  Define "Universe". Give two examples.

- **Yes – Internet in the 1800's!**
- **Yes – The secret network?**
- **[Yes – What's next?]**

Internet in the 1800's!?!

- **Tom Standage, The Victorian Internet**
  - Telegraph!
  - Continent-spanning systems
  - Digital transmission of information
  - Crypto, cracking
  - Nerds
  - Attacks on the moral fiber of society
  - On-line dating (even an on-line wedding!)
- **Distributed message routing despite link outages**
  - Lines cut by armies in wartime!
- **Many problems; solutions eerily similar**

The Secret Network

- **With the Internet, who needs the phone system?**
  - It's a “new era”, etc.
- **One small detail...**
  - From inception, Internet has been a phone system application!
  - To connect two nodes, just ask your telco for a “circuit”...
    - ...somehow there's always copper/fiber waiting for you...
    - ...somehow when it breaks it gets fixed fast...
    - ...somehow your circuit can terminate anywhere...
  - Somehow?

Course Goals

- **Think "the network way"**
  - Distributed coordination is hard, let's go shopping
- **Learn how computer networks work**
  - Problems, approaches, protocols, software
- **Learn how to write network applications**
- **Hands-on understanding of network internals**
  - Build a simple network in software

Selected Key Problems

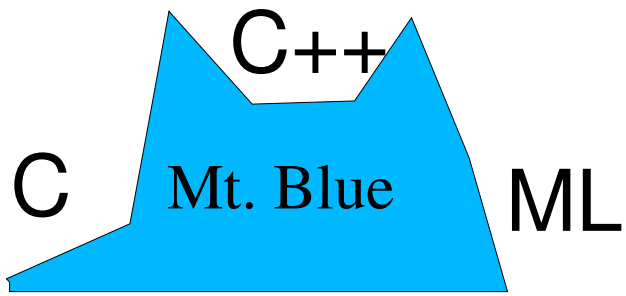
- Two Generals
- Group Membership
- Scaling

Two Generals

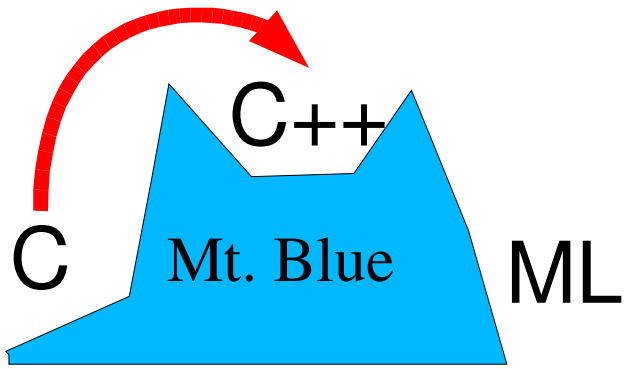
- **Problem (formulated by Jim Gray?)**
  - Two cooperating armies
    - Each size 2X
    - Separated by...
  - One opposing army
    - Size 3X
- **Idealized "combat" (think: Diplomacy, Risk, ...)**
  - 4X vs. 3X: probable win
  - 2X vs. 3X: certain loss

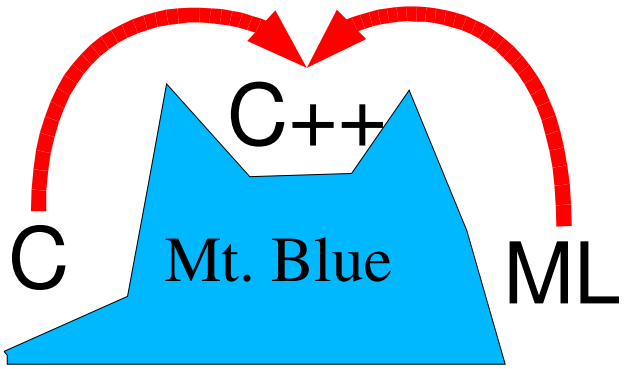
Two Generals

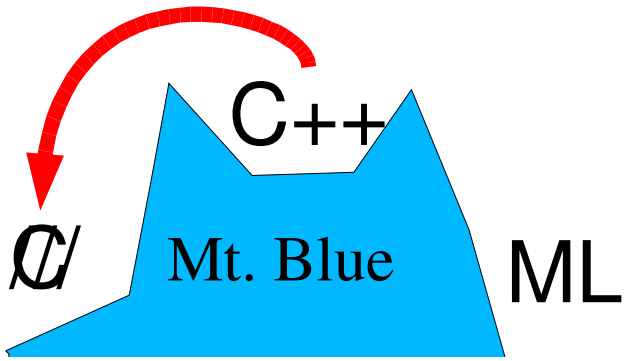
- **Problem**
  - Two cooperating armies
    - Each size 2X
    - Separated by...
  - One opposing army
    - Size 3X
- **Idealized "combat"**
  - 4X vs. 3X: win
  - 2X vs. 3X: lose

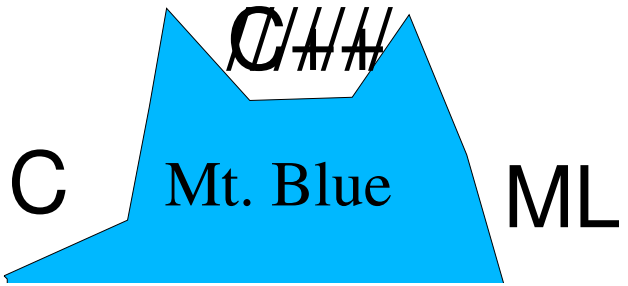


Two Generals









Two Generals

- **Necessity: *coordinated* attack**
  - Armies can communicate via messenger
- **Protocol 0**
  - C: “Attack at dawn!”
  - What if C's messenger is captured by C++?

Two Generals

- **Necessity: *coordinated* attack**
  - Armies can communicate via messenger
- **Protocol 0**
  - C: “Attack at dawn!”
  - What if C's messenger is captured by C++?
- **Protocol 1**
  - C: “Attack at dawn! Ok?”
  - ML: “Ok!”
  - What if ML's messenger is captured?

Two Generals

- **Necessity: *coordinated* attack**
  - Armies can communicate via messenger
- **Protocol 0**
  - C: “Attack at dawn!”
  - What if C's messenger is captured by C++?
- **Protocol 1**
  - C: “Attack at dawn! Ok?”
  - ML: “Ok!”
  - What if ML's messenger is captured?
- **Seemingly-trivial coordination is *impossible*!**

Group Membership

- **Group of nodes on a network**
  - Require distributed election of a “leader”
- **Sample solution**
  - “Dstributed election” algorithm chooses among group members
  - If a node enters or leaves during election, re-start algorithm
- **Results**
  - Works great for 10 nodes
  - Fails horribly for 1,000,000 nodes
    - If inter-node-join time approximates election time...
    - Election process never completes

Group Membership

- **Problem: "group membership" is undefined**
  - By the time you can compute it, it's changed
- **Lots of algorithms will run into trouble**
  - "To acquire a node number, add one to the largest current node number" – oops!
- **Key network functions must face this environment**
  - Routing, naming

Scaling

- **“DOD Standard Internet Protocol”**
  - RFC 760, 1980: Addresses are fixed length of four octets (32 bits). An address begins with a one octet network number, followed by a three octet local address. This three octet field is called the "rest" field.
  - Result: 254 networks (surely enough!)

Scaling

- **“DOD Standard Internet Protocol”**
  - RFC 760, 1980: Addresses are fixed length of four octets (32 bits). An address begins with a one octet network number, followed by a three octet local address. This three octet field is called the "rest" field.
  - Result: 254 networks (surely enough!)
- **Subsequently revised to Class A/B/C networks**
  - ~16k “Class B” networks of ~64k hosts (CMU)
  - ~4m “Class C” networks of ~255 hosts

Scaling

- **"DOD Standard Internet Protocol"**
  - RFC 760, 1980: Addresses are fixed length of four octets (32 bits). An address begins with a one octet network number, followed by a three octet local address. This three octet field is called the "rest" field.
  - Result: 254 networks (surely enough!)
- **Subsequently revised to Class A/B/C networks**
  - ~16k "Class B" networks of ~64k hosts (CMU)
  - ~4m "Class C" networks of ~255 hosts
- **Then "subnets", then "CIDR"**
- **"Surely enough" evaporates pretty fast!**

Network Performance Concepts

- **Throughput**
  - "How many things per unit time?"
  - Mb/s = megabits per second
  - KB/s = kilobytes per second
- **Latency**
  - "How long until my message arrives?"
  - ms = millisecond (10^{-6}), µs = microsecond (10^{-9})
- **Reciprocal "in theory"**
  - bits/second = (1/(seconds/bit))
- **Relationship much more complex**

Hen Performance

- **Old riddle**
  - "If a hen and a half lays an egg and a half in a day and a half, how long does it take to get a dozen eggs?"
- **Egg Latency**
  - How long does it take for one hen to lay one egg?
- **Henhouse throughput (eggs per day)**
  - Increases with number of hens
  - Does *not* mean you can build henhouse, get first egg in 1 hour
  - What is minimum time to 12 eggs?

Latency and Throughput

- **Radio a message to your friend Mike**
  - 1-megabyte photo
  - 1-megabit radio link
  - How long?

Latency and Throughput

- **Radio a message to your friend Mike**
  - 1-megabyte photo
  - 1-megabit radio link
  - How long?

$$\frac{1\,megabyte}{photo} \times \frac{8\,bits}{byte} \times \frac{second}{1\,megabit}$$

Latency and Throughput

- **Radio a message to your friend Mike**
  - 1-megabyte photo
  - 1-megabit radio link
  - How long?

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit}$$

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit}$$

Latency and Throughput

- **Radio a message to your friend Mike**
  - 1-megabyte photo
  - 1-megabit radio link
  - How long?

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit}$$

$$\frac{1\ \cancel{megabyte}}{photo} \times \frac{8\ bits}{\cancel{byte}} \times \frac{second}{1\ \cancel{megabit}}$$

Latency and Throughput

- **Radio a message to your friend Mike**
  - 1-megabyte photo
  - 1-megabit radio link
  - How long?

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit}$$

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit} = \frac{?}{?}$$

Latency and Throughput

- **Radio a message to your friend Mike**
  - 1-megabyte photo
  - 1-megabit radio link
  - How long?

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit}$$

$$\frac{1\ megabyte}{photo} \times \frac{8\ bits}{byte} \times \frac{second}{1\ megabit} = \frac{8\ seconds}{photo}$$

Two Problems

- **Small problem**
  - Mega != Mega
  - Computer people: Megabyte (MB) = 2^{20} bytes
  - Network people: Megabit (Mb) = 10^6 bits
  - It's 8.4 seconds, not 8

Two Problems

- **Small problem**
  - Mega != Mega
  - Computer people: MegaByte (MB) = 2^{20} bytes
  - Network people: Megabit (Mb) = 10^6 bits
  - It's 8.4 seconds, not 8
- **Big problem**
  - I forgot to tell you... Mike lives on the Moon
    - (Extra credit: What is Mike's last name?)
  - It takes radio waves 1.3 seconds to get there

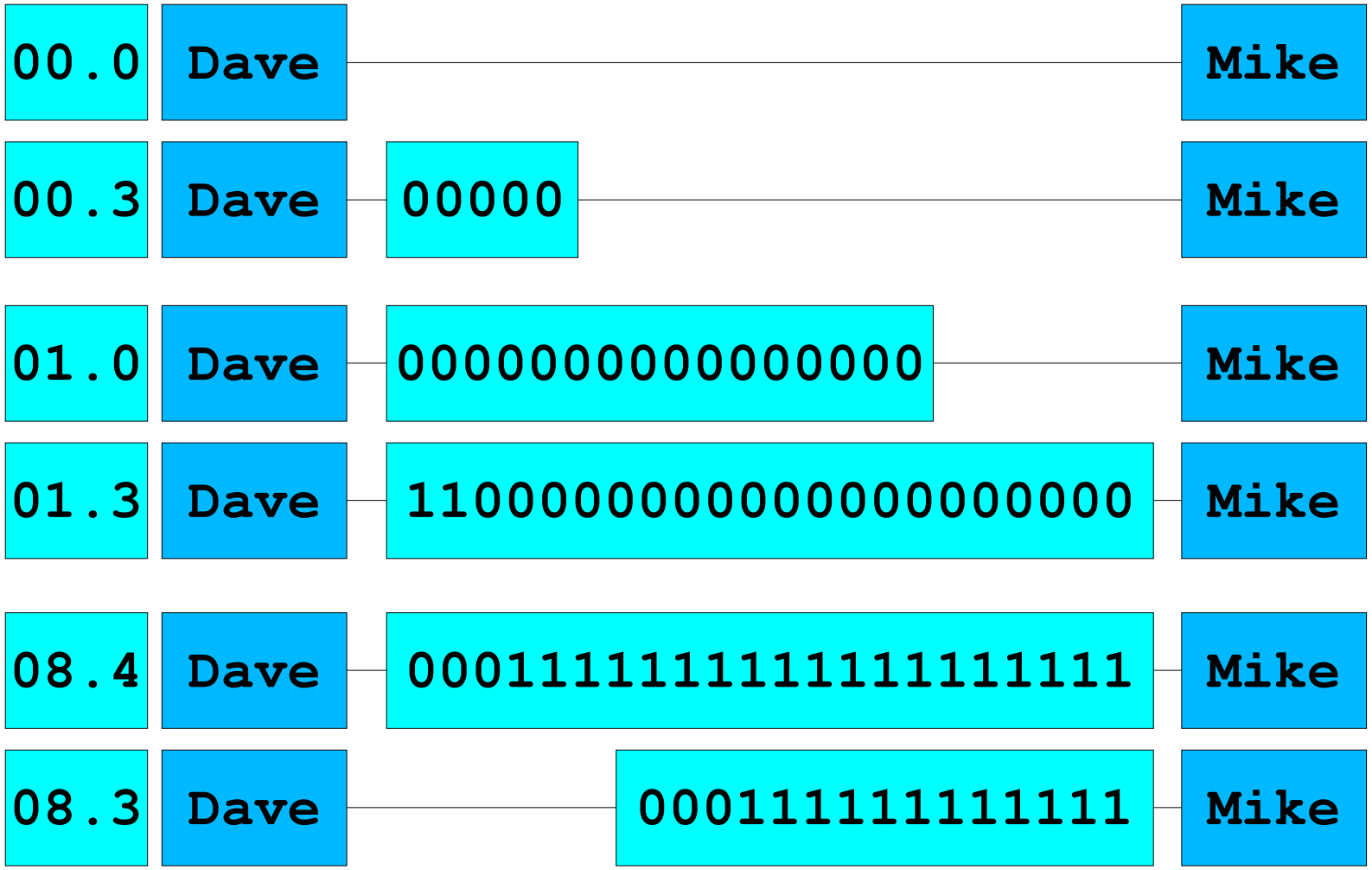
Message Latency

- **Message latency = sum of**
  - Propagation delay (distance/lightspeed)
  - Transmission time (size/throughput)
  - Queue delay (ignore for now)
- **Message to Mike**
  - Propagation delay is 1.3 seconds (one-way)
    - Also known as “link delay”
  - Transmission time is 8.4 seconds, total is 9.7 (121% of 8)
- **By the way: RTT (round-trip-time)**
  - Time to send a 0-bit message there and back: 2.6 seconds

Message Latency

- **Propagation delay vs. transmission time**
  - May vary widely
  - Earth-to-Moon is 1.3 seconds (<< 8.4)
    - Delay is a minor compared to transmission time
    - Can transmit part of message, receive back status
      - “Got that part ok” or “Oops, send it again”

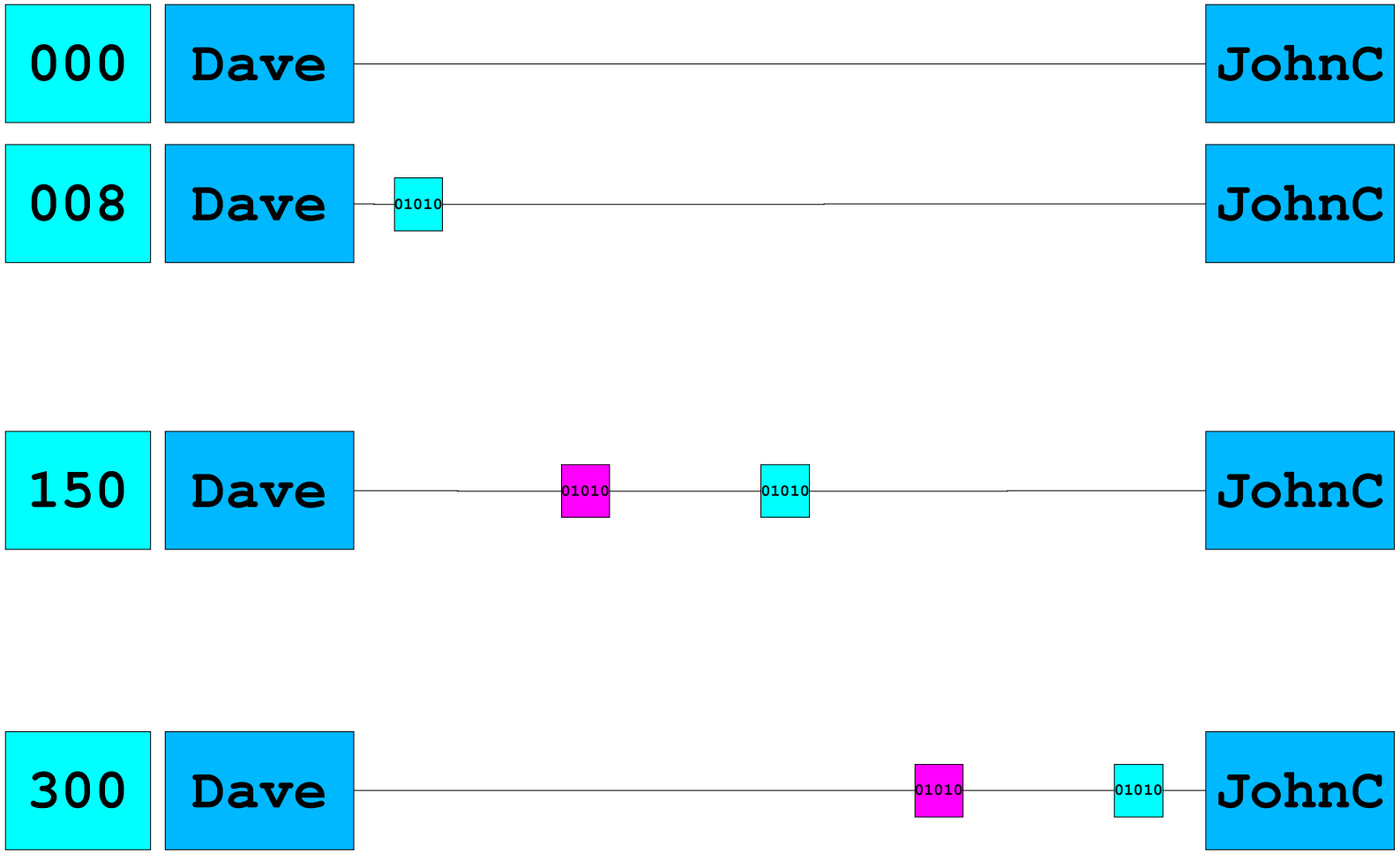
Message Latency



Message Latency

- **Propagation delay vs. transmission time**
  - May vary widely
  - Earth-to-Moon is 1.3 seconds (<< 8.4)
    - Delay is a minor compared to transmission time
    - Can transmit part of message, receive back status
      - "Got that part ok" or "Oops, send it again"
  - Earth-to-Mars is 300-1225 seconds (>> 8.4)
    - Delay vastly exceeds transmission time
    - Link holds *multiple entire messages*

Message Latency



Latency and Throughput

- **Bandwidth-delay product**
  - megabits/second X link-delay
  - This many bits are always “in flight” / “queued in link”
- **What if Mike says “Stop!! My buffer is full!”?**
  - One b-d product of bits are “in flight” to him already
  - You will queue *another* b-d product before you hear his alert!
- **Message throughput (≠ link throughput)**
  - How many messages per second can you send to Mike?
  - Depends on b-d product vs. message size
  - Depends on message protocol (= waiting protocol) you use

Latency and Throughput

- **See text for more-dignified treatment**
  - No hens, no Loonies, no Martians
- **Things to watch out for**
  - Is “delay” one-way or round-trip?
  - Mega vs. mega, kilo vs. kilo
  - Do we mean link latency or message latency?
  - Do we mean link throughput or message throughput?

Things Which Aren't Throughput

- **Bandwidth**
  - Properly, measured in Hertz
  - Difference between max & min frequency of transmission band
  - Routinely abused by CS people to mean “throughput”
- **Goodput**
  - Used to mean “productive throughput”
  - Ignore “waste” if part of a message is transmitted multiple times

Back to the Internet

- **Another reason the Internet isn't perfect**
  - Fatally overoptimized for single-planet case
  - Will work to low-Earth orbit
  - Efficiency problems talking to the Moon
  - Forget about Mars
- **InterPlanetary Internet**
  - http://www.ipnsig.org/

Summary

- **First things first**
  - Read Chapter 1
  - Study socket-programming example as a refresher
- **Project 1 (individual) out Wednesday**
- **Course non-goals, goals**
  - “Networking perspective”, Internet as running example
- **Key problems**
  - Distributed coordination; scaling
- **Network performance concepts**
  - Throughput vs. latency, ...