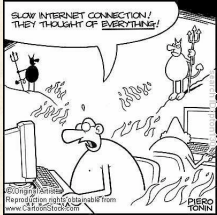




UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202 Introduction to Computation Professor Andrea Arpaci-Dusseau
Fall 2010


Lecture 34: How does a computer... send messages over the Internet?

Brief History of Networking




1970
"Internet" developed



1980

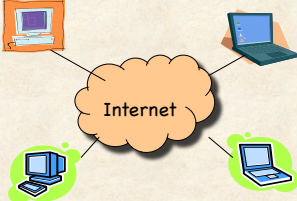
1989: World Wide Web




2000

Caveat: Internet ≠ Web

Internet:
Collection of computers connected on network
Communicate with TCP/IP protocol



Web (WWW):
Hyperlinked content (web pages) stored on servers
Request and serve pages using HTTP protocol
Built *on top of* the internet

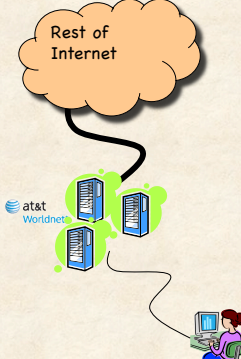


Modern Internet

Simple to connect and use

Need:

- Device capable of speaking right protocol (TCP/IP)
- IP "address" given by an Internet provider
- Connection to provider's servers (via modem, DSL, wireless, etc.)



Your PC
IP Address:
128.156.16.201

Able to access any other machine on Internet!

Today's Challenges

Interesting example of large, heterogeneous system

Challenge 1:

How to find someone when *no one knows everybody*?

Challenge 2:

How to build *reliability* on top of *unreliable* protocols?

Challenge 3:

How to *cooperate* when not in your selfish best interest?

First Challenge

Challenge 1:

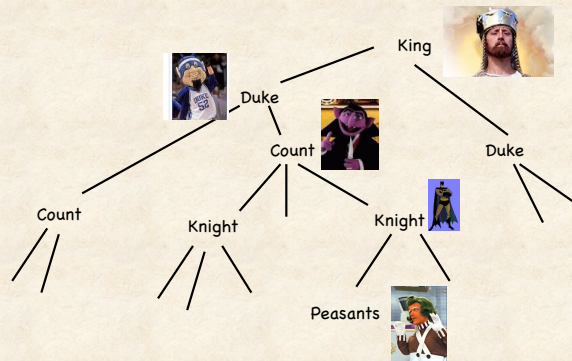
How to find someone when *no one knows everybody*?

Scenario: Imagine you're royal leader with 10,000 peasants in your kingdom

- You need to deliver message to all of them
- "We are being attacked! Come fight for me!"

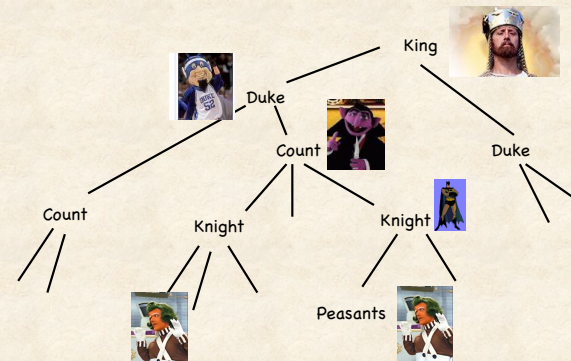
How might you do that?

Solution: Hierarchy

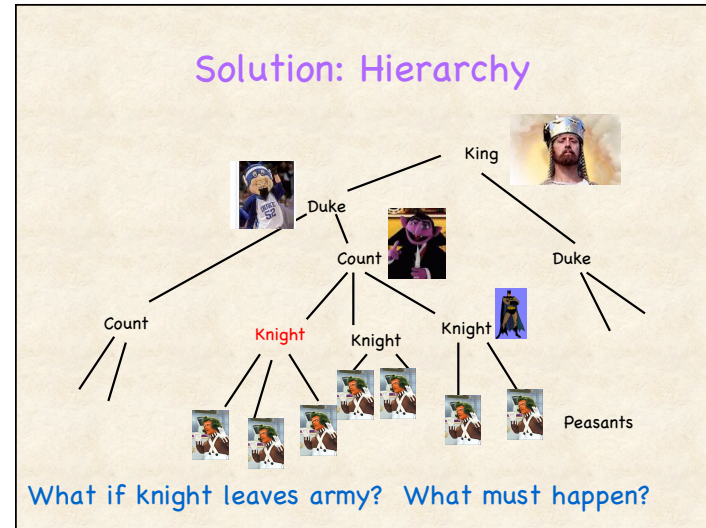
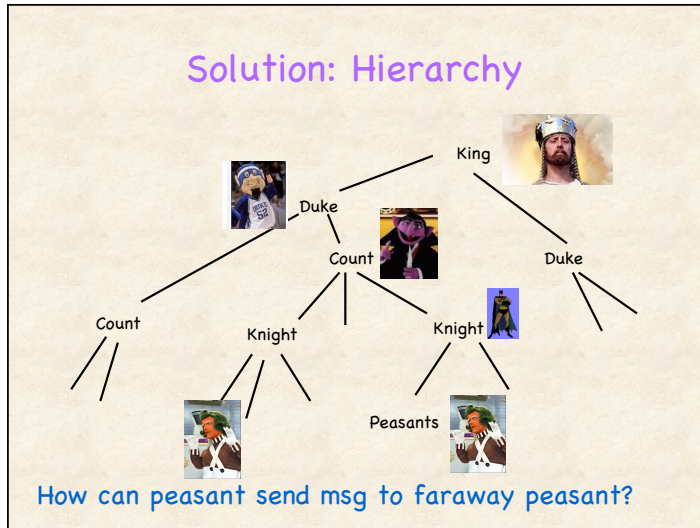


Deliver orders by broadcasting message down tree

Solution: Hierarchy



How can peasant send msg to nearby peasant?



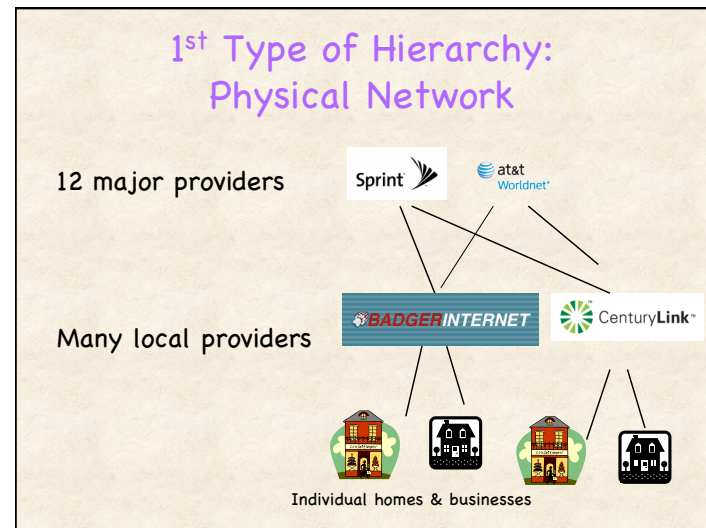
Summary of Hierarchies

Advantages

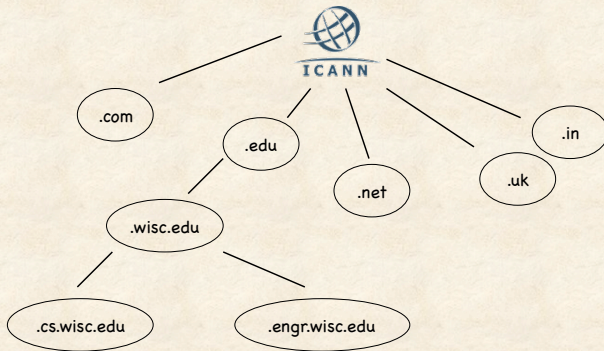
- Manages complexity
- **Decentralized**
 - No single entity knows or controls everything
- Boss doesn't know all employees
 - Manager filters info; only propagate relevant info up
- Managers/employees don't know about other branches

Disadvantages

- Can be tricky to handle new entities leaving/entering system
- Can be bad if too much info is passed up to boss (overworked if micromanage)



2nd Type of Hierarchy: Domain Name System (DNS)



How to find machine?

Scenario: Want to send a message to www.cs.wisc.edu

Must translate www.cs.wisc.edu to IP address

Ask appropriate DNS server in hierarchy

- Contact DNS server managing "edu" namespace
- Contact DNS server managing wisc.edu
- Get IP address: 128.105.7.31

Use IP address for routing messages thru physical network hierarchy based upon address

Challenge 2

How to send message **reliably** when medium might not deliver your message?

Is there some **unreliable communication device** you use everyday?



What problems do you experience w/ cellphones?

How do you deal with them?

Reliability Problems and Solutions

Can't quite understand message?

- "Could you say that again?"
- Retransmit message

Don't hear anything at all?

- "Can you hear me???" <wait> "Can you hear me???"
- Timeout and retry

Sick of listening to other person repeating themselves?

- "I heard you. Go on."
- Acknowledgements

Reliability on Unreliable Protocol

TCP/IP Protocol

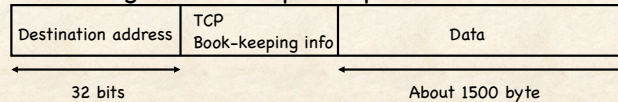
IP: Internet Protocol

- Makes best-effort to get your message to destination

TCP: Transmission Control Protocol

- Builds on top of IP
- Ensures message gets there

All messages broken up into packets



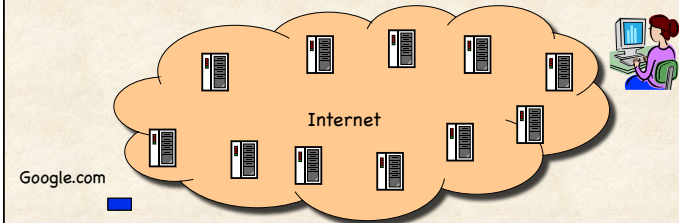
IP: Delivering Messages

Internet contains connected computers called *routers*

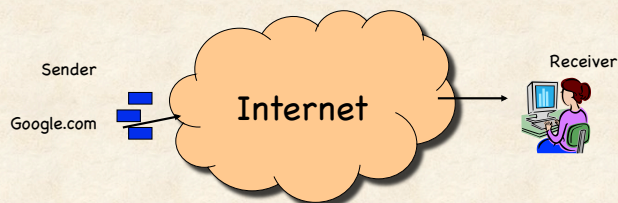
Message divided into multiple packets (Example: 3)

Packets hop from router to router to destination

- Each packet can use different routers!



What can go wrong?



Two problems for today

- Packets may arrive out of order
- Packet may be dropped and never arrive

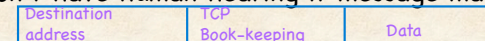
How can TCP over IP fix these problems?

Basic TCP Protocol

Concerns:

- Packets may arrive out of order
- Packet may be dropped and never arrive

Don't have human hearing if message makes sense!



What should be in book-keeping info to help?

Out of order?

Sender:

- Associate unique **sequence** number w/ each packet

Receiver

- **Sort** packets by sequence number

Basic TCP Protocol

Concerns:

- Packets may arrive out of order
- Packet may be dropped and never arrive

Don't have human hearing if message makes sense!

Destination address	TCP Book-keeping	Data
------------------------	---------------------	------

What should be in book-keeping info to help?

Packet dropped?

Receiver

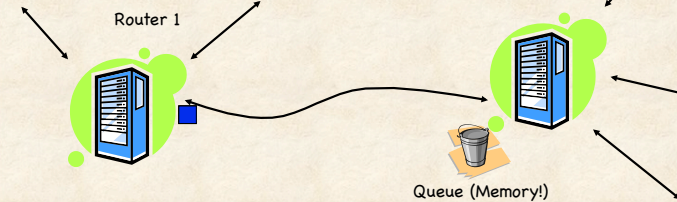
- Acknowledge packets that arrived (by sequence number)

Sender

- Resends packet if no acknowledgment in some time-out interval

Challenge 3

How to cooperate when not in your selfish best interest?



Must avoid network congestion
Queue (Fixed amount of memory) full
→ packets are dropped

How can we avoid this problem (happening too often)?

How should good sender react?

Packets getting dropped?

- → Halve the transmission rate
- How do you know your packets are being dropped?
 - Don't receive acknowledgements

All packets getting through?

- → Increase transmission rate a little

Desired behavior included in TCP/IP software

- "Congestion control"

No enforcement mechanism in Internet!

- Allows cheating, VoIP Telephony, streaming media

How does WWW work?

User-level apps run HTTP protocol on TCP/IP

Client (web browser): Sends requests to server

- Use TCP/IP to find server and ensure requests arrive
- HTTP protocol: "GET filename"

Server: Replies with requested file

- Reads file from file system; sends over network
 - Doesn't know anything about contents of file
- Easy to make your own web server!
- Implementation Issue: Speed

Client: Does work to interpret .html file, display in browser

Today's Summary

Internet: Built using TCP/IP to send packets

- Use hierarchy for decentralized control
- Build reliability (TCP) on top of unreliable layer (IP)
- Congestion control: Slow down when you see problems

Reading

- Section 7.1 – 7.3

Announcements

- Exam 2: Return Monday after Thanksgiving
- Project 2 : Create Trivia Game with Lists
 - Due Monday 12/13 (In class demo)
- Wed before Thanksgiving: Video outside lecture (no lecture)
- Bio 375-004: Spring Service-Learning Course for teaching Scratch to Kids in Madison Afterschool
 - Wed 5:30 – 7:00