General

- Instructor - Ashok K. Agrawala
  - agrawala@cs.umd.edu
  - 4149 AVW

- TA –
  - Office Hours –

- Class Meets – Tu Th 2:00 – 3:15 CSIC 2107
Prerequisite

- **Required Background**
  - must have 351 and 330 (412 or 430 would be helpful)

- **Expectations**
  - Understand the basics of Computer Architecture
  - Experience in implementing non-trivial systems-type projects
  - Should know
    - Processor
    - Memory
    - Kernel vs. user process
  - Familiar with basic probability
Expectations – After the course

- Understand the fundamentals of networking protocols, including protocol layering, basic medium access including wireless protocols, routing, addressing, congestion control
- Understand the principles behind the Internet protocols and some application layer protocols such as http, ftp, and DNS, and a few peer-to-peer systems/protocols such as Gnutella and Chord.
- Understand some of the limitations of the current Internet and its service model
- Understand the causes behind network congestion, and explain the basic methods for alleviating congestion
- Design, implement, and test substantial parts of network protocols
Announcements

- **Required Work**
  - will require about the same amount of effort as 412
    - 412 a (slightly) harder project to debug
    - 417 project is (by design) more ambiguous
Required Texts

- *Computer Networking: A Top Down Approach*
  6th edition
  Jim Kurose, Keith Ross
  Addison-Wesley
  March 2012
- *TCP/IP Sockets in C: A Practical Guide for Programmers*
  2nd Edition by Jeff Donahoo and Ken Calvert,
  Morgan Kaufmann, 2009.
  ISBN 978-0123745408
Other Material

- **Recommended Texts**

- **RFCs**
Grading

- Final 30%
- In-Term Exam(s) 30%
- Programming Assignments 35%
- Class Participation 5%
  - Pop Quizzes
What is this course all about?

- **Computer Networking**
  - ???
Chapter 1: introduction

our goal:
- get “feel” and terminology
- more depth, detail later in course
- approach:
  - use Internet as example

overview:
- what’s the Internet?
- what’s a protocol?
- network edge; hosts, access net, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- security
- protocol layers, service models
- history
Chapter 1: roadmap

1.1 what is the Internet?
1.2 network edge
   - end systems, access networks, links
1.3 network core
   - packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history
What’s the Internet: “nuts and bolts” view

- **millions of connected computing devices:**
  - *hosts = end systems*
  - *running network apps*

- **communication links**
  - fiber, copper, radio, satellite
  - transmission rate: *bandwidth*

- **Packet switches:** forward packets (chunks of data)
  - *routers and switches*
“Fun” internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster + weather forecaster

Slingbox: watch, control cable TV remotely

Tweet-a-watt: monitor energy use

Internet refrigerator

Internet phones
What’s the Internet: “nuts and bolts” view

- **Internet**: “network of networks”
  - Interconnected ISPs
- **Protocols** control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, 802.11
- **Internet standards**
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force
What’s the Internet: a service view

- **Infrastructure that provides services to applications:**
  - Web, VoIP, email, games, e-commerce, social nets, …

- **provides programming interface to apps**
  - hooks that allow sending and receiving app programs to “connect” to Internet
  - provides service options, analogous to postal service
What’s a protocol?

**Human protocols:**
- “what’s the time?”
- “I have a question”
- Introductions

... specific msgs sent
... specific actions taken when msgs received, or other events

**Network protocols:**
- Machines rather than humans
- All communication activity in Internet governed by protocols

Protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt
What’s a protocol?

a human protocol and a computer network protocol:

Q: other human protocols?
Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge
  - end systems, access networks, links

1.3 network core
  - packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

1.7 history
A closer look at network structure:

- **network edge:**
  - hosts: clients and servers
  - servers often in data centers

- **access networks, physical media:** wired, wireless communication links

- **network core:**
  - interconnected routers
  - network of networks
Access networks and physical media

Q: How to connect end systems to edge router?
- residential access nets
- institutional access networks (school, company)
- mobile access networks

keep in mind:
- bandwidth (bits per second) of access network?
- shared or dedicated?
Access net: digital subscriber line (DSL)

- use existing telephone line to central office DSLAM
  - data over DSL phone line goes to Internet
  - voice over DSL phone line goes to telephone net
- < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
- < 24 Mbps downstream transmission rate (typically < 10 Mbps)
Access net: cable network

frequency division multiplexing: different channels transmitted in different frequency bands
Access net: cable network

- **HFC**: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate

- network of cable, fiber attaches homes to ISP router
  - homes share access network to cable headend
  - unlike DSL, which has dedicated access to central office

data, TV transmitted at different frequencies over shared cable distribution network
Access net: home network

- wireless devices
- often combined in single box
- wireless access point (54 Mbps)
- to/from headend or central office
- cable or DSL modem
- router, firewall, NAT
- wired Ethernet (100 Mbps)
Enterprise access networks (Ethernet)

- typically used in companies, universities, etc
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- today, end systems typically connect into Ethernet switch
Wireless access networks

- shared wireless access network connects end system to router
  - via base station aka “access point”

**wireless LANs:**
- within building (100 ft)
- 802.11b/g (WiFi): 11, 54 Mbps transmission rate

**wide-area wireless access**
- provided by telco (cellular) operator, 10’s km
- between 1 and 10 Mbps
- 3G, 4G: LTE
Host: sends *packets* of data

host sending function:
- takes application message
- breaks into smaller chunks, known as *packets*, of length $L$ bits
- transmits packet into access network at *transmission rate* $R$
  - link transmission rate, aka link *capacity*, aka link *bandwidth*

\[
\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}
\]
Physical media

- **bit**: propagates between transmitter/receiver pairs
- **physical link**: what lies between transmitter & receiver
- **guided media**:
  - signals propagate in solid media: copper, fiber, coax
- **unguided media**:
  - signals propagate freely, e.g., radio

**twisted pair (TP)**
- two insulated copper wires
  - Category 5: 100 Mbps, 1 Gbps Ethernet
  - Category 6: 10Gbps
**Physical media: coax, fiber**

**coaxial cable:**
- two concentric copper conductors
- bidirectional
- broadband:
  - multiple channels on cable
  - HFC

**fiber optic cable:**
- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10’s-100’s Gbps transmission rate)
- low error rate:
  - repeaters spaced far apart
  - immune to electromagnetic noise
Physical media: radio

- signal carried in electromagnetic spectrum
- no physical “wire”
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

radio link types:
- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., WiFi)
  - 11 Mbps, 54 Mbps
- wide-area (e.g., cellular)
  - 3G cellular: ~ few Mbps
- satellite
  - Kbps to 45 Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude
Chapter 1: roadmap

1.1 what is the Internet?
1.2 network edge
   - end systems, access networks, links
1.3 network core
   - packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history
The network core

- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into packets
  - forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity
Packet-switching: store-and-forward

- takes $L/R$ seconds to transmit (push out) $L$-bit packet into link at $R$ bps
- **store and forward**: entire packet must arrive at router before it can be transmitted on next link
- end-end delay $= 2L/R$ (assuming zero propagation delay)

**one-hop numerical example:**
- $L = 7.5$ Mbits
- $R = 1.5$ Mbps
- one-hop transmission delay $= 5$ sec

more on delay shortly …
Packet Switching: queueing delay, loss

queuing and loss:
- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up
Two key network-core functions

*Routing*: determines source-destination route taken by packets

- *Routing Algorithms*

*Forwarding*: move packets from router’s input to appropriate router output

<table>
<thead>
<tr>
<th>Header Value</th>
<th>Output Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>3</td>
</tr>
<tr>
<td>0101</td>
<td>2</td>
</tr>
<tr>
<td>0111</td>
<td>2</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
</tr>
</tbody>
</table>

Dest address in arriving packet’s header
Alternative core: circuit switching

end-end resources allocated to, reserved for “call” between source & dest:

- In diagram, each link has four circuits.
  - call gets 2\textsuperscript{nd} circuit in top link and 1\textsuperscript{st} circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (*no sharing*)
- Commonly used in traditional telephone networks
Circuit switching: FDM versus TDM

FDM

TDM

Example:
4 users

Introduction 1-37
Packet switching versus circuit switching

*packet switching allows more users to use network!*

example:
- 1 Mb/s link
- each user:
  - 100 kb/s when “active”
  - active 10% of time

- **circuit-switching:**
  - 10 users

- **packet switching:**
  - with 35 users, probability > 10 active at same time is less than .0004 *
  
  Q: how did we get value 0.0004?

  Q: what happens if > 35 users?

* Check out the online interactive exercises for more examples
Packet switching versus circuit switching

is packet switching a “slam dunk winner?”

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still an unsolved problem (chapter 7)

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?
Internet structure: network of networks

- End systems connect to Internet via access ISPs (Internet Service Providers)
  - Residential, company and university ISPs
- Access ISPs in turn must be interconnected.
  - So that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - Evolution was driven by economics and national policies
- Let’s take a stepwise approach to describe current Internet structure
Internet structure: network of networks

**Question:** given millions of access ISPs, how to connect them together?
Internet structure: network of networks

Option: connect each access ISP to every other access ISP?

connecting each access ISP to each other directly doesn’t scale: $O(N^2)$ connections.
Internet structure: network of networks

Option: connect each access ISP to a global transit ISP? Customer and provider ISPs have economic agreement.
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors
But if one global ISP is viable business, there will be competitors … which must be interconnected
Internet structure: network of networks

... and regional networks may arise to connect access nets to ISPS
Internet structure: network of networks

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users.
Internet structure: network of networks

- at center: small # of well-connected large networks
  - “tier-1” commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - content provider network (e.g., Google): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs
Tier-1 ISP: e.g., Sprint
Chapter 1: roadmap

1.1 what is the Internet?
1.2 network edge
   - end systems, access networks, links
1.3 network core
   - packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history
How do loss and delay occur?

packets queue in router buffers

- packet arrival rate to link (temporarily) exceeds output link capacity
- packets queue, wait for turn

free (available) buffers: arriving packets dropped (loss) if no free buffers

packet being transmitted (delay)

packets queueing (delay)
Four sources of packet delay

\[ d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} \]

- **\( d_{\text{proc}} \): nodal processing**
  - check bit errors
  - determine output link
  - typically < msec

- **\( d_{\text{queue}} \): queueing delay**
  - time waiting at output link for transmission
  - depends on congestion level of router
Four sources of packet delay

trans: transmission delay:
- \( L \): packet length (bits)
- \( R \): link bandwidth (bps)
- \( d_{\text{trans}} = \frac{L}{R} \)

prop: propagation delay:
- \( d \): length of physical link
- \( s \): propagation speed in medium (~2x10^8 m/sec)
- \( d_{\text{prop}} = \frac{d}{s} \)

\( d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} \)

* Check out the Java applet for an interactive animation on trans vs. prop delay.
Caravan analogy

- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car~bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?

- time to “push” entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll booth: 100km/(100km/hr) = 1 hr
- A: 62 minutes
Caravan analogy (more)

- suppose cars now “propagate” at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- **Q:** Will cars arrive to 2nd booth before all cars serviced at first booth?
  - **A:** Yes! after 7 min, 1st car arrives at second booth; three cars still at 1st booth.
Queueing delay (revisited)

- $R$: link bandwidth (bps)
- $L$: packet length (bits)
- $a$: average packet arrival rate

- $La/R \sim 0$: avg. queueing delay small
- $La/R \rightarrow 1$: avg. queueing delay large
- $La/R > 1$: more “work” arriving than can be serviced, average delay infinite!

* Check out the Java applet for an interactive animation on queuing and loss
“Real” Internet delays and routes

- what do “real” Internet delay & loss look like?
- traceroute program: provides delay measurement from source to router along end-end Internet path towards destination. For all $i$:
  - sends three packets that will reach router $i$ on path towards destination
  - router $i$ will return packets to sender
  - sender times interval between transmission and reply.
“Real” Internet delays, routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms  5 ms  5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms  11 ms  13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms  18 ms  18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms  109 ms  106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms  102 ms  104 ms
10 de.fr1.fr.geant.net (62.40.96.50)  113 ms  121 ms  114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13 nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16 194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17 ***
18 *** * means no response (probe lost, router not replying)
19 fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms

* Do some traceroutes from exotic countries at www.traceroute.org
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all

* Check out the Java applet for an interactive animation on queuing and loss
Throughput

- **throughput**: rate (bits/time unit) at which bits transferred between sender/receiver
  - **instantaneous**: rate at given point in time
  - **average**: rate over longer period of time

Server sends bits (fluid) into pipe

Pipe that can carry fluid at rate $R_s$ bits/sec

Pipe that can carry fluid at rate $R_c$ bits/sec
Throughput (more)

- $R_s < R_c$  What is average end-end throughput?

- $R_s > R_c$  What is average end-end throughput?

`bottleneck link`

link on end-end path that constrains end-end throughput
Throughput: Internet scenario

- per-connection end-end throughput: $\min(R_c, R_s, R/10)$
- in practice: $R_c$ or $R_s$ is often bottleneck
Chapter 1: roadmap

1.1 what is the Internet?
1.2 network edge
   - end systems, access networks, links
1.3 network core
   - packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history
Protocol “layers”

Networks are complex, with many “pieces”:

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of organizing structure of network?

.... or at least our discussion of networks?
Organization of air travel

- ticket (purchase)
- baggage (check)
- gates (load)
- runway takeoff
- airplane routing

- ticket (complain)
- baggage (claim)
- gates (unload)
- runway landing
- airplane routing

- a series of steps
Layering of airline functionality

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
<th>Airport</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket (purchase)</td>
<td></td>
<td>departure airport</td>
</tr>
<tr>
<td>baggage (check)</td>
<td></td>
<td>intermediate air-traffic control centers</td>
</tr>
<tr>
<td>gates (load)</td>
<td></td>
<td>arrival airport</td>
</tr>
<tr>
<td>runway (takeoff)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>airplane routing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>airplane routing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>airplane routing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**layers:** each layer implements a service
- via its own internal-layer actions
- relying on services provided by layer below
Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system’s pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer’s service transparent to rest of system
  - e.g., change in gate procedure doesn’t affect rest of system
- layering considered harmful?
Internet protocol stack

- **application**: supporting network applications
  - FTP, SMTP, HTTP
- **transport**: process-process data transfer
  - TCP, UDP
- **network**: routing of datagrams from source to destination
  - IP, routing protocols
- **link**: data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- **physical**: bits “on the wire”
ISO/OSI reference model

- **presentation**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **session**: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
  - these services, *if needed*, must be implemented in application
  - needed?
Chapter 1: roadmap

1.1 what is the Internet?
1.2 network edge
   - end systems, access networks, links
1.3 network core
   - packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history
Network security

- field of network security:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks

- Internet not originally designed with (much) security in mind
  - original vision: “a group of mutually trusting users attached to a transparent network” 😊
  - Internet protocol designers playing “catch-up”
  - security considerations in all layers!
Bad guys: put malware into hosts via Internet

- malware can get in host from:
  - **virus**: self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - **worm**: self-replicating infection by passively receiving object that gets itself executed

- **spyware malware** can record keystrokes, web sites visited, upload info to collection site

- infected host can be enrolled in **botnet**, used for spam. DDoS attacks
Bad guys: attack server, network infrastructure

**Denial of Service (DoS):** attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts
Bad guys can sniff packets

packet “sniffing”:

- broadcast media (shared ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by

*wireshark software used for end-of-chapter labs is a (free) packet-sniffer*
Bad guys can use fake addresses

**IP spoofing:** send packet with false source address

… lots more on security (throughout, Chapter 8)
Chapter 1: roadmap

1.1 what is the Internet?
1.2 network edge
   - end systems, access networks, links
1.3 network core
   - packet switching, circuit switching, network structure
1.4 delay, loss, throughput in networks
1.5 protocol layers, service models
1.6 networks under attack: security
1.7 history
Internet history

1961-1972: Early packet-switching principles

- **1961**: Kleinrock - queueing theory shows effectiveness of packet-switching
- **1964**: Baran - packet-switching in military nets
- **1967**: ARPAnet conceived by Advanced Research Projects Agency
- **1969**: first ARPAnet node operational

- **1972**:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes
**Internet history**

**1972-1980: Internetworking, new and proprietary nets**

- **1970**: ALOHAnet satellite network in Hawaii
- **1974**: Cerf and Kahn - architecture for interconnecting networks
- **1976**: Ethernet at Xerox PARC
- **late 70’s**: proprietary architectures: DECnet, SNA, XNA
- **late 70’s**: switching fixed length packets (ATM precursor)
- **1979**: ARPAnet has 200 nodes

Cerf and Kahn’s internetworking principles:
- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today’s Internet architecture
Internet history

1980-1990: new protocols, a proliferation of networks

- **1983**: deployment of TCP/IP
- **1982**: smtp e-mail protocol defined
- **1983**: DNS defined for name-to-IP-address translation
- **1985**: ftp protocol defined
- **1988**: TCP congestion control
- **new national networks**: Csnet, BITnet, NSFnet, Minitel
- **100,000 hosts connected** to confederation of networks
Internet history

1990, 2000’s: commercialization, the Web, new apps

- early 1990’s: ARPAnet decommissioned
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960’s]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
- late 1990’s – 2000’s:
  - more killer apps: instant messaging, P2P file sharing
  - network security to forefront
  - est. 50 million host, 100 million+ users
  - backbone links running at Gbps
Internet history

2005-present

- ~750 million hosts
  - Smartphones and tablets
- Aggressive deployment of broadband access
- Increasing ubiquity of high-speed wireless access
- Emergence of online social networks:
  - Facebook: soon one billion users
- Service providers (Google, Microsoft) create their own networks
  - Bypass Internet, providing “instantaneous” access to search, email, etc.
- E-commerce, universities, enterprises running their services in “cloud” (eg, Amazon EC2)
Introduction: summary

covered a “ton” of material!

- Internet overview
- what’s a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- history

you now have:

- context, overview, “feel” of networking
- more depth, detail to follow!