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
- Tweet-a-watt: monitor energy use
- Slingbox: watch, control cable TV remotely
- 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:

Hi

Hi

Got the time?

2:00

TCP connection request

TCP connection response

Get http://www.awl.com/kurose-ross

<file>
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**

- **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
Access net: home network

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

\[
\text{packet transmission delay} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}
\]
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 ...

Introduction 1-21
Packet Switching: queueing delay, loss

queueing 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\(^{nd}\) circuit in top link and 1\(^{st}\) circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call \textit{(no sharing)}
- Commonly used in traditional telephone networks
Circuit switching: FDM versus TDM

Example:
4 users

FDM

TDM
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 50 users?
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
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.
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....
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors …. which must be interconnected

Internet exchange point
Internet structure: network of networks

…and regional networks may arise to connect access nets to ISPS
… 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 it data centers to Internet, often bypassing tier-1, regional ISPs
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

packet being transmitted (delay)

packets queueing (delay)

free (available) buffers: arriving packets dropped (loss) if no free buffers
Four sources of packet delay

- **$d_{\text{proc}}$: 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

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

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

- \( d_{\text{prop}} \): propagation delay:
  - \( d \): length of physical link
  - \( s \): propagation speed in medium (~2 \times 10^8 \text{ m/sec})
  - \( d_{\text{prop}} = d/s \)

\( d_{\text{trans}} \) and \( d_{\text{prop}} \) very different
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 both: 100km/(100km/hr)= 1 hr
- A: 62 minutes
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.
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
**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

![Diagram showing server sending bits into a pipe that can carry fluid at rate $R_s$ bits/sec, and another pipe that can carry fluid at rate $R_c$ bits/sec.](image)
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-to-end throughput: \( \min(R_c, R_s, R/10) \)
- in practice: \( R_c \) or \( R_s \) is often bottleneck

10 connections (fairly) share backbone bottleneck link \( R \) bits/sec
Throughput and Goodput

- Throughput is the average rate of successful message delivery over a communication channel.

- Goodput is the application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.
  - The amount of data considered excludes protocol overhead bits as well as retransmitted data packets.
  - Goodput is always lower than the throughput.
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?
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

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?
Why Layering Considered Harmful?

- In the data networking context structured layering implies that the functions of each layer are carried out completely before the protocol data unit is passed to the next layer.
- This means that the optimization of each layer has to be done separately.
- Such ordering constraints are in conflict with efficient implementation of data manipulation functions.
Internet protocol stack

- **application**: supporting network applications
  - FTP, SMTP, HTTP
- **transport**: process data transfer
  - TCP, UDP
- **network**: routing of datagrams from source to destination
  - IP, routing protocols
- **link**: data transfer between neighboring network elements
  - Ethernet, 802.11 (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 applications
Hourglass model
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.
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
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 is a (free) packet-sniffer
Bad guys can use fake addresses

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

... lots more on security
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
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
- **1991**: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- **early 1990s**: Web
  - hypertext [Bush 1945, Nelson 1960’s]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990’s: commercialization of the Web

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
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” (e.g., 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!