33 years of experience in the Information Technology Industry, including twelve years of experience working for leading IT consulting firms such as Computer Sciences Corporation

PhD in Computer Science from University of Colorado at Boulder

Past CEO and CTO

Held senior management and technical leadership roles in many large IT Strategy and Modernization projects for fortune 500 corporations in the insurance, banking, investment banking, pharmaceutical, retail, and information management industries

Contributed to several high-profile ARPA and NSF research projects

Played an active role as a member of the OMG, ODMG, and X3H2 standards committees and as a Professor of Computer Science at Columbia initially and New York University since 1997

Proven record of delivering business solutions on time and on budget

Original designer and developer of jcrew.com and the suite of products now known as IBM InfoSphere DataStage

Creator of the Enterprise Architecture Management Framework (EAMF) and main contributor to the creation of various maturity assessment methodology

Developed partnerships between several companies and New York University to incubate new methodologies (e.g., EA maturity assessment methodology), develop proof of concept software, recruit skilled graduates, and increase the companies’ visibility
<table>
<thead>
<tr>
<th>How to reach me?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cell</strong></td>
</tr>
<tr>
<td><strong>Email</strong></td>
</tr>
<tr>
<td>AIM, Y! IM, ICQ</td>
</tr>
<tr>
<td>MSN IM</td>
</tr>
<tr>
<td>LinkedIn</td>
</tr>
<tr>
<td>Twitter</td>
</tr>
<tr>
<td>Skype</td>
</tr>
</tbody>
</table>
What is the class about?

- **Course description and syllabus:**
  - [http://www.nyu.edu/classes/jcf/CSCI-GA.2262-001/](http://www.nyu.edu/classes/jcf/CSCI-GA.2262-001/)
  - [http://cs.nyu.edu/courses/spring16/CSCI-GA.2262-001/index.html](http://cs.nyu.edu/courses/spring16/CSCI-GA.2262-001/index.html)
  - Most of the readings will come from the required text
  - The textbook will also be useful in solving some of the assigned problems

- **Textbook(s):**
    James F. Kurose, Keith W. Ross
    Addison Wesley
Course Topics Overview

- Computer Networks and the Internet
- Application Layer
- Fundamental Data Structures: queues, ring buffers, finite state machines
- Data Encoding and Transmission
- Local Area Networks and Data Link Control
- Wireless Communications
- Packet Switching
- OSI and Internet Protocol Architecture
- Congestion Control and Flow Control Methods
- Internet Protocols (IP, ARP, UDP, TCP)
- Network (packet) Routing Algorithms (OSPF, Distance Vector)
- IP Multicast
- Sockets
Course Approach

- Introduction to Basic Networking Concepts (Network Stack)
-Origins of Naming, Addressing, and Routing (TCP, IP, DNS)
-Physical Communication Layer
-MAC Layer (Ethernet, Bridging)
-Routing Protocols (Link State, Distance Vector)
-Internet Routing (BGP, OSPF, Programmable Routers)
-TCP Basics (Reliable/Unreliable)
-Congestion Control
-QoS, Fair Queuing, and Queuing Theory
-Network Services – Multicast and Unicast
-Extensions to Internet Architecture (NATs, IPv6, Proxies)
-Network Hardware and Software (How to Build Networks, Routers)
-Overlay Networks and Services (How to Implement Network Services)
-Network Firewalls, Network Security, and Enterprise Networks
Software Requirements

- Software tools will be available from the Internet or from the course Web site under demos as a choice of freeware or commercial tools
- References will be provided on the course Web site
All students should register themselves with the class list, which is used for all technical discussions concerning the course.

To register, go to the following web page, and follow the instructions:

```
cs.nyu.edu/mailman/listinfo/csci_ga_2262_001_fa15
```

You will be notified in return that you are a list participant. Please send all of your questions to this list (not to the instructor) so that everyone can participate.
Grading

- Midterm Exam & Final Take Home Project
- Final grade will be determined by:
  - Scores on a series of homework assignments
  - Class participation
- Assignments will consist of:
  - Problems similar to those in the text
  - Small (less than 500 lines of code) programs
  - Packet analysis using Ethereal packet sniffer
- Since some assignments will be more difficult than others, the percentage that each represents in your final grade will vary
Rules for Working on Assignments

- All assignments must be done individually (see Cheating next)
- Unless stated otherwise in the assignment, all writing and coding must be original
- All assignments must be emailed to the appropriate grader
  - To avoid problems with "lost emails" (e.g., “the Internet ate my homework"), you should save a copy of your EMAILs (not simply the assignment itself)
Cheating Policy

Please do NOT:

- Copy any part of another student's homework answers
- Allow another student to copy your homework
- Copy any part of code found in a book, magazine, the Internet, or other resource
- Present the work of another as your own
- If you use the idea of another in your work, you MUST provide appropriate attribution (that is, cite the work and the author).

The penalty for first cheating offense will be a grade of **F** for the course

Computer Science Department Academic Integrity Policy
Handing in Assignments

- Homework problems must be submitted by email to the designated grader

- Please include:
  - Your name
  - Your SID
  - Assignment number (1, 2, …) in subject
    - Example, “assignment 1”

- **NO** credit will be given for **ANY** assignment after the due date
1. Instructor and Course Introduction
2. Introduction and Overview
3. Summary and Conclusion
A new wave of technology disruption is yielding a massive shift toward B2C led innovation...

The third great shift

**B2B led**
- Mainframe (1960 – 1976)
- Distributed PC (1992 – 2001)
- Desktop machine
- Departmental machine
- Divisional machine
- Enterprise machine

**B2C led**
- Internet PC (2001 – 2009)
- Handheld machine

Important Rule

- Recognize and Adopt “Paradigm Shifts”
  - https://www.youtube.com/watch?v=wOXWSg_PyTQ
  - https://www.youtube.com/watch?v=AesI6HeiwOg
  - https://www.youtube.com/watch?v=x0iRj8_9KhA&index=2&list=PLBCCA5F25EF30184C
Our goal:
- Get “feel” and terminology
- More depth, detail \textit{later} in course
- Approach:
  » Use Internet as example

Overview:
- What is the Internet?
- What is 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
1.1 What is the Internet?
1.2 Network edge
   - end systems, access networks, links
1.3 Network core
   - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched 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*
- *routers*: forward packets (chunks of data)
“Cool” internet appliances & Connected Ecosystems

Smart connected worlds across people, processes, devices, organizations, and systems

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

World’s smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Web-enabled toaster + weather forecaster

Internet phones
“Cool” internet appliances

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

World’s smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Web-enabled toaster + weather forecaster

Internet phones
**What is the Internet: “nuts and bolts” view**

- **Protocols** control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, Ethernet

- **Internet:** “network of networks”
  - loosely hierarchical
  - public Internet versus private intranet

- **Internet standards**
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force
What’s the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing

- communication services provided to apps:
  - reliable data delivery from source to destination
  - “best effort” (unreliable) data delivery
What is 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 is a protocol?

A human protocol and a computer network protocol:

Q: Other human protocols?
1.1 What *is* the Internet?

1.2 Network edge
   - end systems, access networks, links

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

1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History
A closer look at network structure

- **network edge:** applications and hosts
  - access networks, physical media: wired, wireless communication links
- **network core:**
  - interconnected routers
  - network of networks
The network edge

- **end systems (hosts):**
  - run application programs
  - e.g. Web, email
  - at “edge of network”

- **client/server model**
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server

- **peer-peer model:**
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent
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?
Dial-up Modem

- Uses existing telephony infrastructure
- Home is connected to central office
- up to 56Kbps direct access to router (often less)
- Can’t surf and phone at same time: not “always on”
Digital Subscriber Line (DSL)

- Also uses existing telephone infrastructure
- Up to 1 Mbps upstream (today typically < 256 kbps)
- Up to 8 Mbps downstream (today typically < 1 Mbps)
- Dedicated physical line to telephone central office
Residential access: cable modems

- Does not use telephone infrastructure
  - Instead uses cable TV infrastructure
- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- network of cable and fiber attaches homes to ISP router
  - homes share access to router
  - unlike DSL, which has dedicated access
Residential access: cable modems

Diagram: http://www.cableddatacomnews.com/cmic/diagram.html
Cable Network Architecture: Overview

Typically 500 to 5,000 homes

cable headend

cable distribution network (simplified)

home
Cable Network Architecture: Overview

- Cable Headend
- Cable Distribution Network (simplified)
- Home Environment
  - TV
  - Cable Modem
  - PC
  - 10 Mbps Ethernet
  - Set-Top Box
  - Splitter
  - Coax

Diagram illustrating the flow from cable headend to homes through a simplified cable distribution network, ending at home environments with connected devices such as TVs, cable modems, and personal computers.
FDM (more shortly):

Channels

1 2 3 4 5 6 7 8 9

C O N T R O L
D A T A
E E E E E E E T T T O
O O O O O O A A L

Cable headend

cable distribution network

home
- Optical links from central office to the home
- Two competing optical technologies:
  » Passive Optical network (PON)
  » Active Optical Network (PAN)
- Much higher Internet rates; fiber also carries television and phone services
- Typically used in companies, universities, etc
- 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
- 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:
  - 802.11b/g (WiFi): 11 or 54 Mbps

- wider-area wireless access
  - provided by telco operator
  - ~1Mbps over cellular system (EVDO, HSDPA)
  - next up (?): WiMAX (10’s Mbps) over wide area
Typical home network components:

- DSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point

to/from cable headend

cable modem
router/firewall

wireless access point

wireless laptops

Ethernet
Physical Media

- **Bit**: propagates between transmitter/rcvr 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 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5: 100 Mbps Ethernet
**Physical Media: coax, fiber**

**Coaxial cable:**
- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- 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 Gps)
- 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: ~ 1 Mbps
- satellite
  - Kbps to 45 Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude
1.1 What is the Internet?
1.2 Network edge
  - end systems, access networks, links
1.3 Network core
  - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History
The Network Core

- mesh of interconnected routers
- the fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete “chunks”
End-end resources reserved for “call”

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required
network resources (e.g., bandwidth) divided into “pieces”

- pieces allocated to calls
- resource piece *idle* if not used by owning call (*no sharing*)

- dividing link bandwidth into “pieces”
  - frequency division
  - time division
Circuit Switching: FDM and TDM

Example:
4 users

FDM

TDM
each end-end data stream divided into *packets*

- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

resource contention:
- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding
Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand ➞ **statistical multiplexing**.

TDM: each host gets same slot in revolving TDM frame.
Packet-switching: store-and-forward

- takes $L/R$ seconds to transmit (push out) packet of $L$ bits on to link at $R$ bps
- **store and forward**: entire packet must arrive at router before it can be transmitted on next link
- delay = $3L/R$ (assuming zero propagation delay)

**Example:**
- $L = 7.5$ Mbits
- $R = 1.5$ Mbps
- transmission delay = 15 sec

more on delay shortly ...
Packet switching allows more users to use network!

- 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?
Packet switching versus circuit switching

Is packet switching a “slam dunk winner?”

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion: 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)?
• roughly hierarchical
• at center: “tier-1” ISPs (e.g., Verizon, Sprint, AT&T, Cable and Wireless), national/international coverage
  » treat each other as equals
Tier-1 ISP: e.g., Sprint

POP: point-of-presence

to/from backbone

peering

to/from customers
“Tier-2” ISPs: smaller (often regional) ISPs

» Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

Tier-2 ISP is customer of tier-1 provider

Tier-2 ISPs also peer privately with each other.
"Tier-3" ISPs and local ISPs

- last hop ("access") network (closest to end systems)

Local and tier-3 ISPs are customers of higher tier ISPs connecting them to rest of Internet.
a packet passes through many networks!
1.1 What is the Internet?

1.2 Network edge
   - end systems, access networks, links

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

1.4 Delay, loss and throughput in packet-switched 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 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

- 1. *nodal processing*:
  - check bit errors
  - determine output link

- 2. *queueing*:
  - time waiting at output link for transmission
  - depends on congestion level of router
3. Transmission delay:
- $R =$ link bandwidth (bps)
- $L =$ packet length (bits)
- time to send bits into link = $L/R$

4. Propagation delay:
- $d =$ length of physical link
- $s =$ propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- propagation delay = $d/s$

Note: $s$ and $R$ are very different quantities!
- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service car (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
Cars now “propagate” at 1000 km/hr

Toll booth now takes 1 min to service a car

Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?

Yes! After 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.

1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!

» See Ethernet applet at AWL Web site
The nodal delay, $d_{\text{nodal}}$, is the sum of several components:

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

- $d_{\text{proc}}$ = processing delay
  - Typically a few microseconds or less
- $d_{\text{queue}}$ = queuing delay
  - Depends on congestion
- $d_{\text{trans}}$ = transmission delay
  - $= L/R$, significant for low-speed links
- $d_{\text{prop}}$ = propagation delay
  - A few microseconds to hundreds of milliseconds
Queueing delay (revisited)

- \( R = \) link bandwidth (bps)
- \( L = \) packet length (bits)
- \( a = \) average packet arrival rate

traffic intensity = \( \frac{L a}{R} \)

- \( \frac{L a}{R} \approx 0 \): average queueing delay small
- \( \frac{L a}{R} \to 1 \): delays become large
- \( \frac{L a}{R} > 1 \): more “work” arriving than can be serviced, average delay infinite!
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.
**traceroute**: `gaia.cs.umass.edu` to `www.eurecom.fr`

Three delay measurements from `gaia.cs.umass.edu` to `cs-gw.cs.umass.edu`

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 112 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. ***
19. `fantasia.eurecom.fr` (193.55.113.142) 132 ms 128 ms 136 ms

* means no response (probe lost, router not replying)
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**: 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
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
1.1 What *is* the Internet?
1.2 Network edge
   - end systems, access networks, links
1.3 Network core
   - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History
Networks are complex!

- 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>Layers: each layer implements a service</th>
</tr>
</thead>
<tbody>
<tr>
<td>» via its own internal-layer actions</td>
</tr>
<tr>
<td>» relying on services provided by layer below</td>
</tr>
</tbody>
</table>
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  
  - PPP, Ethernet

- **physical**: bits “on the wire”
- **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?
Encapsulation
Roadmap

1.1 What is the Internet?
1.2 Network edge
   - end systems, access networks, links
1.3 Network core
   - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched networks
1.5 Protocol layers, service models
1.6 Networks under attack: security
1.7 History
The field of network security is about:

» 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 can put malware into hosts via Internet

- Malware can get in host from a virus, worm, or trojan horse.

- Spyware malware can record keystrokes, web sites visited, upload info to collection site.

- Infected host can be enrolled in a botnet, used for spam and DDoS attacks.

- Malware is often self-replicating: from an infected host, seeks entry into other hosts
Bad guys can put malware into hosts via Internet

- **Trojan horse**
  - Hidden part of some otherwise useful software
  - Today often on a Web page (Active-X, plugin)

- **Virus**
  - infection by receiving object (e.g., e-mail attachment), actively executing
  - self-replicating: propagate itself to other hosts, users

- **Worm:**
  - infection by passively receiving object that gets itself executed
  - self-replicating: propagates to other hosts, users

Sapphire Worm: aggregate scans/sec in first 5 minutes of outbreak (CAIDA, UWisc data)
Bad guys can attack servers and 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 toward target from compromised hosts
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
- **IP spoofing**: send packet with false source address
The bad guys can record and playback

- **record-and-playback**: sniff sensitive info (e.g., password), and use later
  - password holder *is* that user from system point of view
more throughout this course

» Chapter 8 of the textbook focuses on security

cryptographic techniques: obvious uses and not so obvious uses
1.1 What is the Internet?
1.2 Network edge
   - end systems, access networks, links
1.3 Network core
   - circuit switching, packet switching, network structure
1.4 Delay, loss and throughput in packet-switched 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 demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

![The ARPA Network](image-url)
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
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: 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
2007 on:

- ~500+ million hosts
- Voice, Video over IP
- P2P applications: BitTorrent (file sharing), Skype (VoIP), PPLive (video)
- more applications: YouTube, gaming
- wireless, mobility
Agenda

1. Instructor and Course Introduction
2. Introduction and Overview
3. Summary and Conclusion
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!
Assignments & Readings

- Readings
  - Abstract and Chapter 1
- Assignment #1
Quiz!

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Please work it out!
Questions, Comments, Discussions?
Next Session: Application Layer