Networks and Database Systems

Professor Tom Ellman
Lecture 1
Goals of this Course

• Provide students with an overview of network and database architecture.
• Provide students with concepts and skills needed to write network / database applications.
Topic Outline

• Network Architecture.
• Java Socket Programming.
• Client Side Programming.
• Server Side Programming.
• Server / Database Interaction.
Class Wiki
https://www.cs.vassar.edu/courses/cs375-201303/top

• Overview of the course.
• Schedule of topics and readings.
• Homework assignments.
• Summary of grading policy.
• Professor Ellman’s lecture notes.
Communication

- I sometimes send email messages to the entire class.
- My messages typically contain hints or clarifications of homework assignments.
- Our Moodle site will hold an archive of these messages.
Contacting Professor Ellman

• Office: 117 OLB

• Phone: 437-5991

• Email: thellman@vassar.edu
Professor Ellman’s Office Hours

• M, W: 12pm-1pm
• Tu, Th: 1:30pm-2:30pm
• Or send email to make an appointment.
• Or just stop by my office.
Reading


• Downey, Tim, “Web Development with Java: Using Hibernate, JSPs and Servlets, Springer 2009. (Required)

Course Work.

- Programming Assignments.
- Research and Class Presentation.
- Final Project.
Grading

• Class Participation: 10%
• Programming Assignments 60%
• Class Presentation: 10%
• Final Project: 20%
Deadlines and Lateness

• Assignments are due at the start of class on the date specified.

• Late assignments will be accepted with 10% penalty, but only until the start of the next class.
Academic Integrity

- You may discuss general ideas with classmates.
- You must do each programming assignment entirely by yourself.
- You may not discuss or share programs with other students.
- Vassar regulations require the professor to report suspected violations of academic integrity to the Dean of Studies.
- Read the “Originality and Attribution” pamphlet.
Model-View-Controller
Web Application Architecture

View
Browser (Applet)

Controller
Server (Servlet)

Model
Database (MySQL)

Client Machine

Server Machine
Software & Hardware Systems

• Java.
• NetBeans IDE.
• Tomcat Servlet Container.
• MySQL Database System.
• Asprey Lab Linux Machines.
Recurring Issues

- Managing communication among diverse systems.
- Coping with unreliability of network communication.
- Interfacing (relatively) fast computers with (relatively) slow networks.
- Protecting systems and data in an hostile environment.
Chapter 1
Introduction

A note on the use of these ppt slides:
We’re making these slides freely available to all (faculty, students, readers).
They’re in PowerPoint form so you can add, modify, and delete slides
(including this one) and slide content to suit your needs. They obviously
represent a lot of work on our part. In return for use, we only ask the
following:
☑ If you use these slides (e.g., in a class) in substantially unaltered form,
that you mention their source (after all, we’d like people to use our book!)
☑ If you post any slides in substantially unaltered form on a www site, that
you note that they are adapted from (or perhaps identical to) our slides, and
note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2009
J.F Kurose and K.W. Ross, All Rights Reserved

Introduction
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)
What’s 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

![Diagram of network hierarchy with nodes labeled Mobile network, Global ISP, Regional ISP, Home network, Institutional network.]
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’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?
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
Cable Network Architecture: Overview

The cable network architecture consists of a cable headend, which houses the server(s), and a cable distribution network that connects to homes.
Cable Network Architecture: Overview
Cable Network Architecture: Overview

FDM (more shortly):
Ethernet Internet access

- 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
Home networks

Typical home network components:

- DSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point
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”
Network Core: Circuit Switching

End-end resources reserved for “call”

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required
Network Core: Circuit Switching

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

FDM

Example:
4 users

TDM

4 users
Network Core: Packet Switching

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

Bandwidth division into “pieces”
Dedicated allocation
Resource reservation
Packet Switching: Statistical Multiplexing

Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand $\Rightarrow$ 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 ...

Introduction 1-40
Packet switching versus circuit switching

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)?
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

![Diagram showing packet transmission and queueing](image)
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 = \( \frac{L}{R} \)

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

Note: \( s \) and \( R \) are very different quantities!
Nodal delay

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

- \( d_{\text{proc}} = \) processing delay
  - typically a few microsecs 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 microsecs to hundreds of msecs
Queueing delay (revisited)

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

Traffic intensity = \( \frac{La}{R} \)

- \( \frac{La}{R} \sim 0 \): average queueing delay small
- \( \frac{La}{R} \to 1 \): delays become large
- \( \frac{La}{R} > 1 \): more “work” arriving than can be serviced, average delay infinite!
“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 and routes

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

<table>
<thead>
<tr>
<th></th>
<th>Hostname and IP Address</th>
<th>Delay 1</th>
<th>Delay 2</th>
<th>Delay 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cs-gw (128.119.240.254)</td>
<td>1 ms</td>
<td>1 ms</td>
<td>2 ms</td>
</tr>
<tr>
<td>2</td>
<td>border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)</td>
<td>1 ms</td>
<td>1 ms</td>
<td>2 ms</td>
</tr>
<tr>
<td>3</td>
<td>cht-vbns.gw.umass.edu (128.119.3.130)</td>
<td>6 ms</td>
<td>5 ms</td>
<td>5 ms</td>
</tr>
<tr>
<td>4</td>
<td>jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)</td>
<td>16 ms</td>
<td>11 ms</td>
<td>13 ms</td>
</tr>
<tr>
<td>5</td>
<td>jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)</td>
<td>21 ms</td>
<td>18 ms</td>
<td>18 ms</td>
</tr>
<tr>
<td>6</td>
<td>abilene-vbns.abilene.ucaid.edu (198.32.11.9)</td>
<td>22 ms</td>
<td>18 ms</td>
<td>22 ms</td>
</tr>
<tr>
<td>7</td>
<td>nycm-wash.abilene.ucaid.edu (198.32.8.46)</td>
<td>22 ms</td>
<td>22 ms</td>
<td>22 ms</td>
</tr>
<tr>
<td>8</td>
<td>62.40.103.253 (62.40.103.253)</td>
<td>104 ms</td>
<td>109 ms</td>
<td>106 ms</td>
</tr>
<tr>
<td>9</td>
<td>de2-1.de1.de.geant.net (62.40.96.129)</td>
<td>109 ms</td>
<td>102 ms</td>
<td>104 ms</td>
</tr>
<tr>
<td>10</td>
<td>de.fr1.fr.geant.net (62.40.96.50)</td>
<td>113 ms</td>
<td>121 ms</td>
<td>114 ms</td>
</tr>
<tr>
<td>11</td>
<td>renater-gw.fr1.fr.geant.net (62.40.103.54)</td>
<td>112 ms</td>
<td>114 ms</td>
<td>112 ms</td>
</tr>
<tr>
<td>12</td>
<td>nio-n2.cssi.renater.fr (193.51.206.13)</td>
<td>111 ms</td>
<td>114 ms</td>
<td>116 ms</td>
</tr>
<tr>
<td>13</td>
<td>nice.cssi.renater.fr (195.220.98.102)</td>
<td>123 ms</td>
<td>125 ms</td>
<td>124 ms</td>
</tr>
<tr>
<td>14</td>
<td>r3t2-nice.cssi.renater.fr (195.220.98.110)</td>
<td>126 ms</td>
<td>126 ms</td>
<td>124 ms</td>
</tr>
<tr>
<td>15</td>
<td>eurecom-valbonne.r3t2.ft.net (193.48.50.54)</td>
<td>135 ms</td>
<td>128 ms</td>
<td>133 ms</td>
</tr>
<tr>
<td>16</td>
<td>194.214.211.25 (194.214.211.25)</td>
<td>126 ms</td>
<td>128 ms</td>
<td>126 ms</td>
</tr>
<tr>
<td>17</td>
<td>** ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>** ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>fantasia.eurecom.fr (193.55.113.142)</td>
<td>132 ms</td>
<td>128 ms</td>
<td>136 ms</td>
</tr>
</tbody>
</table>

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

Trans-oceanic link

* 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
Protocol “Layers”

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?
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”
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?
Network Security

• 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
The 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
The bad guys can use false source addresses

- **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
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 demonstration
  - 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
- 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
Internet History

2007:
• ~500 million hosts
• Voice, Video over IP
• P2P applications: BitTorrent (file sharing), Skype (VoIP), PPLive (video)
• more applications: YouTube, gaming
• wireless, mobility