Introduction

Behnam Momeni
Computer Engineering Department
Sharif University of Technology

Acknowledgments: Lecture slides are from Computer networks course thought by Jennifer Rexford at Princeton University. When slides are obtained from other sources, a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.
What You Learn in This Course

• **Knowledge:** how the Internet works
  – IP protocol suite
  – Internet architecture
  – Applications (Web, e-mail, P2P, VoIP, …)

• **Insight:** key concepts in networking
  – Protocols
  – Layering
  – Resource allocation
  – Naming

• **Skill:** network programming
  – Socket programming
  – Designing and implementing protocols
Okay, so let’s get started… with a crash course in data networking
Key Concepts in Networking

• Protocols
  – Speaking the same language
  – Syntax and semantics

• Layering
  – Standing on the shoulders of giants
  – A key to managing complexity

• Resource allocation
  – Dividing scarce resources among competing parties
  – Memory, link bandwidth, wireless spectrum, paths, …
  – Distributed vs. centralized algorithms

• Naming
  – What to call computers, services, protocols, …
Protocols: Calendar Service

- Making an appointment with your advisor

- Specifying the messages that go back and forth
  - And an understanding of what each party is doing
Protocols: Calendar Service

• Making an appointment with your advisor

Please meet with me for 1.5 hours starting at 1:30pm on February 8, 2006?

• Specifying the messages that go back and forth
  – And an understanding of what each party is doing
Protocols: Calendar Service

• Making an appointment with your advisor

Please meet with me for 1.5 hours starting at 1:30pm on February 8, 2006?

I can’t.

• Specifying the messages that go back and forth – And an understanding of what each party is doing
Protocols: Calendar Service

• Making an appointment with your advisor

Please meet with me for 1.5 hours starting at 3:00pm on February 8, 2006?

• Specifying the messages that go back and forth
  – And an understanding of what each party is doing
Protocols: Calendar Service

• Making an appointment with your advisor

Please meet with me for 1.5 hours starting at 3:00pm on February 8, 2006?

• Specifying the messages that go back and forth
  – And an understanding of what each party is doing

I can’t.
Protocols: Calendar Service

• Making an appointment with your advisor

Please meet with me for 1.5 hours starting at 4:30pm on February 8, 2006?

• Specifying the messages that go back and forth
  – And an understanding of what each party is doing
Protocols: Calendar Service

• Making an appointment with your advisor

Please meet with me for 1.5 hours starting at 4:30pm on February 8, 2006?

Yes!

• Specifying the messages that go back and forth
  – And an understanding of what each party is doing
Okay, So This is Getting Tedious
Okay, So This is Getting Tedious

• You: When are you free to meet for 1.5 hours during the next two weeks?
Okay, So This is Getting Tedious

• You: When are you free to meet for 1.5 hours during the next two weeks?
• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
Okay, So This is Getting Tedious

• You: When are you free to meet for 1.5 hours during the next two weeks?
• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
• You: Book me for 1.5 hours at 10:30am on Feb 8.
Okay, So This is Getting Tedious

• You: When are you free to meet for 1.5 hours during the next two weeks?

• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.

• You: Book me for 1.5 hours at 10:30am on Feb 8.

• Advisor: Yes.
Well, Not Quite Enough
Well, Not Quite Enough

• Student #1: When can you meet for 1.5 hours during the next two weeks?
Well, Not Quite Enough

• Student #1: When can you meet for 1.5 hours during the next two weeks?
• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
Well, Not Quite Enough

• Student #1: When can you meet for 1.5 hours during the next two weeks?

• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.

• Student #2: When can you meet for 1.5 hours during the next two weeks?
Well, Not Quite Enough

- Student #1: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #2: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
Well, Not Quite Enough

- Student #1: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #2: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #1: Book me for 1.5 hours at 10:30am on Feb 8.
Well, Not Quite Enough

• Student #1: When can you meet for 1.5 hours during the next two weeks?
• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.

• Student #2: When can you meet for 1.5 hours during the next two weeks?
• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.

• Student #1: Book me for 1.5 hours at 10:30am on Feb 8.
• Advisor: Yes.
Well, Not Quite Enough

- Student #1: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #2: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #1: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Yes.
- Student #2: Book me for 1.5 hours at 10:30am on Feb 8.
Well, Not Quite Enough

- Student #1: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #2: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #1: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Yes.
- Student #2: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Uh… well… I can no longer can meet then. I’m free at 1:15pm on Feb 9.
Well, Not Quite Enough

- Student #1: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #2: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #1: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Yes.
- Student #2: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Uh… well… I can no longer can meet then. I’m free at 1:15pm on Feb 9.
- Student #2: Book me for 1.5 hours at 1:15pm on Feb 9.
Well, Not Quite Enough

• Student #1: When can you meet for 1.5 hours during the next two weeks?

• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.

• Student #2: When can you meet for 1.5 hours during the next two weeks?

• Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.

• Student #1: Book me for 1.5 hours at 10:30am on Feb 8.

• Advisor: Yes.

• Student #2: Book me for 1.5 hours at 10:30am on Feb 8.

• Advisor: Uh… well… I can no longer can meet then. I’m free at 1:15pm on Feb 9.

• Student #2: Book me for 1.5 hours at 1:15pm on Feb 9.

• Advisor: Yes.
Specifying the Details

• How to identify yourself?
  – Name? Student ID?

• How to represent dates and time?
  – Time, day, month, year? In what time zone?
  – Number of seconds since Jan 1, 1970?

• What granularities of times to use?
  – Any possible start time and meeting duration?
  – Multiples of five minutes?

• How to represent the messages?
  – Strings? Record with name, start time, and duration?

• What do you do if you don’t get a response?
  – Ask again? Reply again?
Example: HyperText Transfer Protocol

GET /courses/archive/ce443/ HTTP/1.1
Host: www.cs.sharif.edu
User-Agent: Mozilla/4.03
CRLF

Request
Example: HyperText Transfer Protocol

Request

GET /courses/archive/ce443/ HTTP/1.1
Host: www.cs.sharif.edu
User-Agent: Mozilla/4.03
CRLF

Response

HTTP/1.1 200 OK
Date: Mon, 4 Feb 2010 13:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Mon, 4 Feb 2010 11:12:23 GMT
Content-Length: 21
CRLF
Site under construction
### Example: IP Packet

<table>
<thead>
<tr>
<th>4-bit Version</th>
<th>4-bit Header Length</th>
<th>8-bit Type of Service (TOS)</th>
<th>16-bit Total Length (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16-bit Identification</td>
<td>3-bit Flags</td>
<td>13-bit Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>8-bit Time to Live (TTL)</td>
<td>8-bit Protocol</td>
<td>16-bit Header Checksum</td>
<td></td>
</tr>
<tr>
<td>32-bit Source IP Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit Destination IP Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20-byte header
IP: Best-Effort Packet Delivery

• Packet switching
  – Send data in packets
  – Header with source & destination address

• Best-effort delivery
  – Packets may be lost
  – Packets may be corrupted
  – Packets may be delivered out of order
Example: Transmission Control Protocol

- Communication service (socket)
  - Ordered, reliable byte stream
  - Simultaneous transmission in both directions

- Key mechanisms at end hosts
  - Retransmit lost and corrupted packets
  - Discard duplicate packets and put packets in order
  - Flow control to avoid overloading the receiver buffer
  - Congestion control to adapt sending rate to network load
Protocol Standardization

• Communicating hosts speaking the same protocol
  – Standardization to enable multiple implementations
  – Or, the same folks have to write all the software

• Standardization: Internet Engineering Task Force
  – Based on working groups that focus on specific issues
  – Produces “Request For Comments” (RFCs)
    • Promoted to standards via rough consensus and running code
    • E.g., RFC 1945 on “HyperText Transfer Protocol – HTTP/1.0”
  – IETF Web site is http://www.ietf.org

• De facto standards: same folks writing the code
  – P2P file sharing, Skype, <your protocol here>…
Layering: A Modular Approach

• Sub-divide the problem
  – Each layer relies on services from layer below
  – Each layer exports services to layer above

• Interface between layers defines interaction
  – Hides implementation details
  – Layers can change without disturbing other layers

Application
Application-to-application channels
Host-to-host connectivity
Link hardware
IP Suite: End Hosts vs. Routers

HTTP message

TCP segment

IP packet

host

HTTP

TCP

IP

Ethernet interface

router

HTTP message

TCP segment

IP packet

host

HTTP

TCP

IP

Ethernet interface

SONET interface

Ethernet interface

router

IP packet

IP packet

IP packet
The Internet Protocol Suite

The waist facilitates interoperability
Layer Encapsulation

User A

Get index.html
Connection ID
Source/Destination
Link Address

User B
Problem: Packet size

- On Ethernet, max IP packet is 1500 bytes
- Typical Web page is 10 kbytes
What if the Data Doesn’t Fit?

Problem: Packet size

- On Ethernet, max IP packet is 1500 bytes
- Typical Web page is 10 kbytes

Solution: Split the data across multiple packets
Protocol Demultiplexing

- Multiple choices at each layer
Demultiplexing: Port Numbers

• Differentiate between multiple transfers
  – Knowing source and destination host is not enough
  – Need an id for each transfer between the hosts

• Specify a particular service running on a host
  – E.g., HTTP server running on port 80
  – E.g., FTP server running on port 21
Is Layering Harmful?

• Layer N may duplicate lower level functionality
  – E.g., error recovery to retransmit lost data

• Layers may need same information
  – E.g., timestamps, maximum transmission unit size

• Strict adherence to layering may hurt performance
  – E.g., hiding details about what is really going on

• Some layers are not always cleanly separated
  – Inter-layer dependencies for performance reasons

• Headers start to get really big
  – Sometimes more header bytes than actual content
Resource Allocation: Queues

• Sharing access to limited resources
  – E.g., a link with fixed service rate

• Simplest case: first-in-first out queue
  – Serve packets in the order they arrive
  – When busy, store arriving packets in a buffer
  – Drop packets when the queue is full
What if the Data gets Dropped?

Problem: Lost Data

GET index.html → Internet
What if the Data gets Dropped?

Problem: Lost Data

Solution: Timeout and Retransmit
What if the Data is Out of Order?

Problem: Out of Order

GET x.htindeml
What if the Data is Out of Order?

Problem: Out of Order

Solution: Add Sequence Numbers

GET x.htindeml

GET index.html
What if too many folks are sending data?
- Senders agree to slow down their sending rates
  … in response to their packets getting dropped

The essence of TCP congestion control
- Key to preventing congestion collapse of the Internet
Transmission Control Protocol

- **Flow control: window-based**
  - Sender limits number of outstanding bytes (window size)
  - *Receiver window* ensures data does not overflow receiver

- **Congestion control: adapting to packet losses**
  - *Congestion window* tries to avoid overloading the network (increase with successful delivery, decrease with loss)
  - TCP connection starts with small initial congestion window

![Graph showing slow start and congestion avoidance phases over time](graph.png)
Naming: Domain Name System (DNS)

• Properties of DNS
  – Hierarchical name space divided into zones
  – Translation of names to/from IP addresses
  – Distributed over a collection of DNS servers

• Client application
  – Extract server name (e.g., from the URL)
  – Invoke system call to trigger DNS resolver code
    • E.g., gethostbyname() on “www.cs.sharif.edu”

• Server application
  – Extract client IP address from socket
  – Optionally invoke system call to translate into name
    • E.g., gethostbyaddr() on “12.34.158.5”
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
DNS Resolver and Local DNS Server

Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
DNS Resolver and Local DNS Server

Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
DNS Resolver and Local DNS Server

Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Caching based on a time-to-live (TTL) assigned by the DNS server responsible for the host name to reduce latency in DNS translation.
Conclusions

• Course objectives
  – How the Internet works, key concepts in networking, and Network programming

• Key concepts in networking
  – Protocols, layers, resource allocation, and naming

• Next lecture:
  – Read Chapter 1 of the Peterson/Davie book
  – Skim the online reference material on sockets
  – (Re)familiarize yourself with C programming