CSC 458/2209 – Computer Networks

Handout # 2:
Course Logistics and Introduction

Professor Yashar Ganjali
Department of Computer Science
University of Toronto

yganjali@cs.toronto.edu
http://www.cs.toronto.edu/~yganjali
Today

• Outline
  • What this course is about

• Logistics
  • Course structure, assignments, evaluation
  • What is expected from you
  • What you can expect from this course

• Review
  • Simple example – mail vs. FTP
  • Foundations and basic concepts
What is This Course About?

• Undergrad course; can be taken by grads

• Computer networks
  • Basics: Layers, naming, and addressing, network (socket) programming, packet switching, routing, congestion control, ...
  • Advanced networking: HTTP, web, peer-to-peer, routers and switches, security, multimedia, online social networks, software-defined networking, ...

• Theory vs. Practice
  • CSC 358: foundation and theory
  • CSC 458: advanced networking and network programming
Logistics – Prerequisites, Readings

• Prerequisites
  • Algorithms
  • Basic probability theory
  • Strong background in C programming and Unix environment

• CSC 358 is not a prerequisite.

• Readings
  • Will be posted on course schedule web page
  • Read before class
Logistics – Textbooks

• Textbook

• Recommended books
  • “TCP/IP Illustrated, Volume 1: The Protocols”, W. Richard Stevens, 1993
Logistics – Hours, Web, Announcements

- **Office hours**
  - Tue. 3-4 PM, Wed. 3-4 PM, Bahen 5238
  - Or by appointment

- **Course web page**
  

- Please check the class web page, and the bulletin board regularly for announcements.
Logistics – Sections

• This course is offered in three sections
  • L0101: the section you are currently attending

• Other sections
  • L0201: Thu 1-3PM
  • L5101: Tue. 6-8PM

• Might have slight differences in content
  • Assignments and exams are coordinated
Logistics – Teaching Assistants

- Alireza Shekaramiz
  - alirezashekaramiz@gmail.com
- Mohsen Norouzi
  - mohsen.norouzi@gmail.com
- Hao WANG
  - wh.sjtu@gmail.com
- Saeed Arezoumand
  - saeed.arezoumand@gmail.com
- Lilin Zhang
  - llzhang@cs.toronto.edu
- Seyed-Hossein Mortazavi
  - hosein.mortazavi@mail.utoronto.ca
- Keran Yang
  - keran.yang@mail.utoronto.ca
- Shuhao Liu
  - shuhao@ece.toronto.edu
- Shiva Ketabi
  - shiva.ketabi@gmail.com
Logistics – TA hours, Tutorials

- Check class web page for office hours.

- Tutorials and discussion session
  - L0101: Fri. 1-2 PM, BA1200

- First tutorial: this week
  - Friday at 1PM for this section
Logistics – Mailing List, Bulletin Board

• Bulletin board
  • We will use Piazza for announcements and Q&A
  • Post any questions related to the course.
  • Check previous posts before asking a question.
  • We guarantee to respond within 48 hours.

• Class mailing list
  • Based on e-mail address you have defined on ROSI.
  • The TAs and I will use this list for announcements only.
  • Do not send e-mails to this list!
Logistics – Grading

- Grading for undergraduate AND graduate students
  - Assignments: 50%
    - Problem sets: 20%
    - Programming: 30%
  - Midterm exam: 20% - In class
    - L0101: October 27\textsuperscript{th}
    - Other sections: please check with your instructor
      - Different times and locations
  - Final exam: 30% - TBA

- Please note that grading is the same for graduate and undergraduate students this year.
Logistics - Deadlines

• Assignment deadlines
  • One free late submission of 24 hours
    • Use on assignment of your choice
    • E-mail TAs before the deadline
  • 10% deduction for each day late
    • Up to 20%
    • Assignment not accepted after two days
Logistics – Programming Assignments

• To be completed in groups of three students.

• You can submit your assignment during a 7-10 day period before the deadline
  • And have the results of basic tests back
  • Limited to 2-3 submissions.

• Socket Programming
• MiniNet
  • Your very own virtual network!
  • You will create and program your own network
  • VM available on CDF machines
  • More detail on this later.

• This is a heavy course, but manageable!
Logistics – Academic Integrity

• Academic Integrity
  • All submissions must present original, independent work.
  • We take academic offenses very seriously.
  • Please read
    • Handout # 1 (course information sheet)
    • “Guideline for avoiding plagiarism”
    • http://www.cs.toronto.edu/~fpitt/documents/plagiarism.html
    • “Advice about academic offenses”
    • http://www.cs.toronto.edu/~clarke/acoffences/
Logistics - Accessibility

- Accessibility Needs
  - The University of Toronto is committed to accessibility. If you require accommodations or have any accessibility concerns, please visit [http://studentlife.utoronto.ca/accessibility](http://studentlife.utoronto.ca/accessibility) as soon as possible.
Acknowledgements

- Special thanks to:
  - Nick McKeown from Stanford University
  - Jennifer Rexford from Princeton University
  - David Wetherall from University of Washington
  - Nick Feamster from Georgia Tech
Quick Survey

- Have you taken CSC358 before?
- Have you taken any networking course?
- Are you familiar with
  - Socket programming?
  - Ethernet, framing, encoding, error detection/correction?
  - UDP, TCP and congestion control?
  - DNS, SNMP, BGP?
  - BitTorrent?
  - Voice and video over IP?
  - Control plane vs. data path?
  - Network security?
  - Software-defined networking?
Questions?

What else do you want to know about this course?
Announcement

- First tutorial
  - Friday, Sep. 18\textsuperscript{th}
  - BA1200
  - 1-2 PM
- Covers \textit{socket programming}
- You’ll need this information for your first programming assignment, which will be posted next week.
Let’s Begin

- An introduction to the mail system
- An introduction to the Internet
An Introduction to the Mail System

U of T

Yashar

Admin

Stanford

Nick

Admin
Characteristics of the Mail System

- Each envelope is individually routed.
- No time guarantee for delivery.
- No guarantee of delivery in sequence.
- No guarantee of delivery at all!
  - Things get lost
  - How can we acknowledge delivery?
  - Retransmission
    - How to determine when to retransmit? Timeout?
    - Need local copies of contents of each envelope.
    - How long to keep each copy.
    - What if an acknowledgement is lost?
An Introduction to the Mail System

U of T

Yashar

Admin

Stanford

Nick

Admin

Application Layer

Transport Layer

Network Layer

Link Layer
An Introduction to the Internet

Yashar

O.S.

Datagram

Application Layer

Nick

O.S.

Transport Layer

Network Layer

Link Layer

cs.toronto.edu

leland.stanford.edu
Characteristics of the Internet

- Each packet is individually routed.
- No time guarantee for delivery.
- No guarantee of delivery in sequence.
- No guarantee of delivery at all!
  - Things get lost
  - Acknowledgements
  - Retransmission
    - How to determine when to retransmit? Timeout?
    - Need local copies of contents of each packet.
    - How long to keep each copy?
    - What if an acknowledgement is lost?
Characteristics of the Internet – Cont’d

- No guarantee of integrity of data.
- Packets can be fragmented.
- Packets may be duplicated.
Layering in the Internet

- **Transport Layer**
  - Provides reliable, in-sequence delivery of data from end-to-end on behalf of application.

- **Network Layer**
  - Provides “best-effort”, but unreliable, delivery of datagrams.

- **Link Layer**
  - Carries data over (usually) point-to-point links between hosts and routers; or between routers and routers.
An Introduction to the Mail System

Application Layer

Transport Layer

Network Layer

Link Layer
Some Questions About the Mail System

- How many sorting offices are needed and where should they be located?
- How much sorting capacity is needed?
  - Should we allocate for Mother’s Day?
- How can we guarantee timely delivery?
  - What prevents delay guarantees?
  - Or delay variation guarantees?
- How do we protect against fraudulent mail deliverers, or fraudulent senders?
Outline – Foundations & Basic Concepts

- A detailed FTP example
  - Layering
  - Packet switching and circuit switching
Example: FTP over the Internet Using TCP/IP and Ethernet
In the Sending Host

1. Application-Programming Interface (API)
   - Application requests TCP connection with “B”

2. Transmission Control Protocol (TCP)
   - Creates TCP “Connection setup” packet
   - TCP requests IP packet to be sent to “B”
In the Sending Host – Cont’d

3. Internet Protocol (IP)

- Creates IP packet with correct addresses.
- IP requests packet to be sent to router.

![Diagram of TCP and IP packet encapsulation]

- **TCP Packet**
  - TCP Data
  - TCP Header

- **IP Packet**
  - IP Data
  - IP Header

- **Encapsulation**

  - Destination Address: IP “B”
  - Source Address: IP “A”
  - Protocol = TCP
In the Sending Host – Cont’d

4. Link ("MAC" or Ethernet) Protocol
   • Creates MAC frame with Frame Check Sequence (FCS).
   • Wait for Access to the line.
   • MAC requests PHY to send each bit of the frame.

Encapsulation

<table>
<thead>
<tr>
<th>IP Packet</th>
<th>Encapsulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Data</td>
<td>Ethernet Data</td>
</tr>
<tr>
<td>IP Header</td>
<td>Ethernet Header</td>
</tr>
</tbody>
</table>

Ethernet Packet

Destination Address: MAC “R1”
Source Address: MAC “A”
Protocol = IP
In Router R1

5. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.

```
In Router R1

5. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.
```

```
IP Packet

<table>
<thead>
<tr>
<th>IP Data</th>
<th>IP Header</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decapsulation

<table>
<thead>
<tr>
<th>Ethernet FCS</th>
<th>Ethernet Data</th>
<th>Ethernet Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Address: MAC “R1”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Address: MAC “A”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol = IP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```

```
Ethernet Packet

<table>
<thead>
<tr>
<th>Ethernet FCS</th>
<th>Ethernet Data</th>
<th>Ethernet Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Address: MAC “R1”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Address: MAC “A”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol = IP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```

```
```
In Router R1

6. Internet Protocol (IP)

- Use IP destination address to decide where to send packet next (“next-hop routing”).
- Request Link Protocol to transmit packet.

![Diagram of IP packet structure]

**IP Packet**

**IP Data**

**IP Header**

**Destination Address:** IP “B”
**Source Address:** IP “A”
**Protocol:** TCP
In Router R1

7. Link ("MAC" or Ethernet) Protocol
   - Creates MAC frame with Frame Check Sequence (FCS).
   - Wait for Access to the line.
   - MAC requests PHY to send each bit of the frame.

```
Ethernet Packet

<table>
<thead>
<tr>
<th>Ethernet FCS</th>
<th>Ethernet Data</th>
<th>Ethernet Header</th>
</tr>
</thead>
</table>
```

Encapsulation

```
IP Packet

<table>
<thead>
<tr>
<th>IP Data</th>
<th>IP Header</th>
</tr>
</thead>
</table>
```

Destination Address: MAC "R2"
Source Address: MAC "R1"
Protocol = IP
16. Link (“MAC” or Ethernet) Protocol

- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.

**Encapsulation**

- **IP Packet**
  - IP Data
  - IP Header

- **Ethernet Packet**
  - Ethernet FCS
  - Ethernet Data
  - Ethernet Header

- Destination Address: **MAC “B”**
- Source Address: **MAC “R5”**
- Protocol = IP
In the Receiving Host

17. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.

**Decapsulation**

<table>
<thead>
<tr>
<th>Ethernet FCS</th>
<th>Ethernet Data</th>
<th>Ethernet Header</th>
</tr>
</thead>
</table>

**IP Packet**

- IP Data
- IP Header

**Ethernet Packet**

Destination Address: MAC “B”
Source Address: MAC “R5”
Protocol = IP
In the Receiving Host - Cont’d

18. Internet Protocol (IP)

- Verify IP address.
- Extract/decapsulate TCP packet from IP packet.
- Pass TCP packet to TCP Protocol.

![Diagram of IP and TCP packets](image)

Decapsulation

<table>
<thead>
<tr>
<th>IP Data</th>
<th>TCP Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP Header</td>
</tr>
</tbody>
</table>

Destination Address: IP “B”
Source Address: IP “A”
Protocol = TCP
In the Receiving Host - Cont’d

19. Transmission Control Protocol (TCP)
   - Accepts TCP “Connection setup” packet
   - Establishes connection by sending “Ack”.

20. Application-Programming Interface (API)
   - Application receives request for TCP connection with “A”.

TCP Packet

<table>
<thead>
<tr>
<th>TCP Data</th>
<th>TCP Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>Empty</td>
</tr>
</tbody>
</table>

Type = Connection Setup
Outline – Foundations & Basic Concepts

- A detailed FTP example
- Layering
- Packet switching and circuit switching
Layering – The OSI Model

Peering-to-peer communication

Layering – The OSI Model

Layer-to-layer communication
Layering – *Our FTP Example*

The 7-layer OSI Model

- Application
- Presentation
- Session
- Transport
- Network
- Link
- Physical

The 4-layer Internet model

- Application
- Transport
- Network
- Link

FTP

ASCII/Binary

TCP

IP

Ethernet
Outline – Foundations & Basic Concepts

- A detailed FTP example
- Layering
- Packet switching and circuit switching
Circuit Switching

- It’s the method used by the telephone network.
- A call has three phases:
  - Establish circuit from end-to-end (“dialing”),
  - Communicate,
  - Close circuit (“tear down”).
- Originally, a circuit was an end-to-end physical wire.
- Nowadays, a circuit is like a virtual private wire: each call has its own private, guaranteed data rate from end-to-end.
Circuit Switching – *Telephone Network*

Each phone call is allocated 64kb/s. So, a 10Gb/s trunk line can carry about 156,000 calls.
Packet Switching

- It’s the method used by the Internet.
- Each packet is individually routed packet-by-packet, using the router’s local routing table.
- The routers maintain no per-flow state.
- Different packets may take different paths.
- Several packets may arrive for the same output link at the same time, therefore a packet switch has buffers.
Packet Switching – **Simple Router Model**

![Diagram showing packet switching and simple router model](image)

- **Link 1**, ingress
- **Link 1**, egress
- **Link 2**, ingress
- **Link 2**, egress
- **Link 3**, ingress
- **Link 3**, egress
- **Link 4**, ingress
- **Link 4**, egress

Choose Egress

Choose Egress

Choose Egress

Choose Egress

4

4
Network traffic is bursty. i.e. the rate changes frequently.
Peaks from independent flows generally occur at different times.
Conclusion: The more flows we have, the smoother the traffic.
Because the buffer absorbs temporary bursts, the egress link need not operate at rate $N.R$.

But the buffer has finite size, $B$, so losses will occur.
Statistical Multiplexing

- **Rate**
  - **A**
  - **B**
  - **C**

- **Time**
  - **A**
  - **B**
  - **C**

Diagram showing the rate and time for signals A, B, and C.
Statistical Multiplexing Gain

Statistical multiplexing gain = \( \frac{2C}{R} \)

Other definitions of SMG: The ratio of rates that give rise to a particular queue occupancy, or particular loss probability.
Why Packet Switching in the Internet?

- Efficient use of expensive links:
  - The links are assumed to be expensive and scarce.
  - Packet switching allows many, bursty flows to share the same link efficiently.
  - “Circuit switching is rarely used for data networks, ... because of very inefficient use of the links” - Gallager

- Resilience to failure of links & routers:
  - “For high reliability, ... [the Internet] was to be a datagram subnet, so if some lines and [routers] were destroyed, messages could be ... rerouted” - Tanenbaum
Final Comments, Discussion

- Is layering the best approach?
  - Simplifies design
  - Yet, limited and inflexible

- Best effort service
  - Made the rapid growth of the Internet possible
  - Makes providing any guarantees very difficult

- Packet switching
  - Enables statistical multiplexing
  - We need extremely fast routers

- Routing
  - How does a router know which output port to send the packet to?