CSC 2229 – Software-Defined Networking

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, evaluation
  • What is expected from you
  • What you want to know

• Overview
  • Packet switching systems
  • Software-Defined Networking
CSC229: Software-Defined Networking

- Graduate level course
  - Introduction to Software-Defined Networking (SDN)
  - SDN challenges and opportunities
- Course components
  - Lectures
  - Research papers
  - Research project
- Theory + Practice
  - Switching systems are simple enough for us to prove something about them
  - Yet they are complex enough to work in practice
Outline – Part I

• Introduction
  • Packet switch systems
  • Evolution of the Internet, and Internet routers
  • Basic architectural components
  • Some example architectures

• Software-Defined Networking
  • Innovation in computer networks
  • Control vs. data plane
Outline – Part II

- SDN Challenges and Opportunities
  - Controller and switch design
  - Reliability, efficiency, and scalability
  - Architecture
  - Programming, correctness, and debugging
  - SDN security
  - New abstractions, state management
  - Network services
  - Network function virtualization
  - Network optimization
Logistics

• Office hours
  • Tue. 3-4 PM, Wed. 3-4 PM, BA5238
  • Or by appointment

• Course web page
  • Please check regularly for announcements.

• Class mailing list
  • Based on e-mail address you have defined on ROSI.
  • Send me an e-mail if you haven’t received a welcome message from that list.
  • Any (course-related!) question posted to the list will be answered within 48 hours.
Logistics

• Prerequisites
  • Any introductory course on networking
  • Algorithms
  • Basic probability theory

• Papers
  • Will be listed on class web page
  • Please read suggested papers BEFORE class
Logistics

- **Grading**
  - Active participation in class and discussions: 10%
  - Paper presentations: 30%
  - Final project: 60%
    - Proposal: 5% - 1-2 pages
    - Intermediate report: 10% - 3 pages
    - Presentation: 15% - Last week of classes
    - Final report: 30% - 6 pages

- **Deadlines**
  - 5% mark deduction for each day of delay, up to 4 days.
  - Exception: final report deadline is hard.
Logistics

• Final Project
  • Groups of three students
  • Project topic
    • Choose from the offered list; or
    • Talk to me and define your own
Logistics

• Academic Integrity
  • All submissions must present original, independent work.
  • We take academic offenses very seriously.
  • Please read
    • “Guideline for avoiding plagiarism”
    • “Advice about academic offenses”
      http://www.cs.toronto.edu/~clarke/acoффences/
Logistics

• 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 as soon as possible.
Acknowledgements

- Special thanks to:
  - Prof. Nick McKeown, Stanford University
  - Prof. Balaji Prabhakar, Stanford University
  - Prof. Jennifer Rexford, Princeton University
  - Prof. Nick Feamster, Princeton University
Questions?

What else do you like to know about this course?
Introduction to Packet Switching

Background

- What is a router?
- Router architecture

- Packet processing in routers
  - IP address lookup
  - Packet buffering
  - Switching

- Router architectures
  - The evolution
  - Software-Defined Networking
What is Routing?

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>R3</td>
</tr>
<tr>
<td>E</td>
<td>R3</td>
</tr>
<tr>
<td>F</td>
<td>R5</td>
</tr>
</tbody>
</table>
What is Routing?

### Routing Table

<table>
<thead>
<tr>
<th>Ver</th>
<th>HLen</th>
<th>T.Service</th>
<th>Total Packet Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragment ID</th>
<th>Flags</th>
<th>Fragment Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TTL</th>
<th>Protocol</th>
<th>Header Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Headers

- Source Address
- Destination Address
- Options (if any)
- Data

### Diagram

A - R1 - R3 - R4 - D

- 20 bytes

- Next Hop
- Destination Address
- Data
- Options (if any)
What is Routing?

A

B

C

R1

R2

R3

R4

R5

D

E
Points of Presence (POPs)
Where High Performance Routers are Used

R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16

(2.5 Gb/s)

(2.5 Gb/s)
What a Router Looks Like

**Cisco GSR 12416**
- **Capacity:** 160Gb/s
- **Power:** 4.2kW
- **Dimensions:** 6ft x 2ft x 19"

**Juniper M160**
- **Capacity:** 80Gb/s
- **Power:** 2.6kW
- **Dimensions:** 3ft x 2.5ft x 19"
Basic Architectural Components of an IP Router

Control Plane

Data-path
per-packet processing

Routing Protocols
Routing Table
Forwarding Table
Switching
Packet Switching – *Simple Router Model*

- **Link 1**, ingress
- **Link 1**, egress
- **Link 2**, ingress
- **Link 2**, egress
- **Link 3**, ingress
- **Link 3**, egress
- **Link 4**, ingress
- **Link 4**, egress
Introduction to Packet Switching

Background

- What is a router?
- Router architecture

- Packet processing in routers
  - IP address lookup
  - Packet buffering
  - Switching

- Router architectures
  - The evolution
  - Software-Defined Networking
Per-packet Processing in an IP Router

1. Accept packet arriving on an incoming link.
2. Lookup packet destination address in the forwarding table, to identify outgoing port(s).
3. Manipulate packet header: e.g., decrement TTL, update header checksum.
4. Send packet to the outgoing port(s).
5. Buffer packet in the queue.
6. Transmit packet onto outgoing link.
Generic Router Architecture

Header Processing

- Lookup IP Address
- Update Header
- Queue Packet

Data Hdr

- Address Table
  - ~1M prefixes
  - Off-chip DRAM

- Buffer Memory
  - ~1M packets
  - Off-chip DRAM

IP Address
Next Hop
Generic Router Architecture

Header Processing
- Lookup IP Address
- Update Header
- Address Table

Buffer Manager
- Buffer Memory

Data Header

Generic Router Architecture
Why are Fast Routers Difficult to Make?

- It’s hard to keep up with Moore’s Law:
  - The bottleneck is memory speed.
  - Memory speed is not keeping up with Moore’s Law.
- Moore’s Law is too slow:
  - Routers need to improve faster than Moore’s Law.
Router Performance Exceeds Moore’s Law

- Growth in capacity of commercial routers:
  - Capacity 1992 ~ 2Gb/s
  - Capacity 1995 ~ 10Gb/s
  - Capacity 1998 ~ 40Gb/s
  - Capacity 2001 ~ 160Gb/s
  - Capacity 2003 ~ 640Gb/s
  - Capacity 2007 ~ 4Tb/s
  - Capacity 2010 ~ 16Tb/s

Average growth rate: 2x / 18 months.
Introduction to Packet Switching

• Background
  • What is a router?
  • Router architecture

Packet processing in routers
  • IP address lookup
  • Packet buffering
  • Switching

• Router architectures
  • The evolution
  • Software-Defined Networking
Generic Router Architecture

- Header Processing
  - Lookup IP Address
  - Update Header
  - Address Table

- Buffer Manager
  - Buffer Memory
IP Address Lookup

- Why it’s thought to be hard:
  1. It’s not an exact match: it’s a longest prefix match.
  2. The table is large: about 500,000 entries today, and growing.
  3. The lookup must be fast: about 30ns for a 10Gb/s line.
Routing lookup: Find the longest matching prefix (aka the most specific route) among all prefixes that match the destination address.
IP Address Lookup

Why it’s thought to be hard:

1. It’s not an exact match: it’s a longest prefix match.
2. The table is large: about 500,000 entries today, and growing.
3. The lookup must be fast: about 30ns for a 10Gb/s line.
Address Tables are Large

Source: http://www.cidr-report.org/
IP Address Lookup

- Why it’s thought to be hard:
  1. It’s not an exact match: it’s a longest prefix match.
  2. The table is large: about 500,000 entries today, and growing.
  3. The lookup must be fast: about 30ns for a 10Gb/s line.
Lookups Must be Fast

<table>
<thead>
<tr>
<th>Year</th>
<th>Line Rate</th>
<th>40B Packets (Mpkt/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>622Mb/s</td>
<td>1.94</td>
</tr>
<tr>
<td>1999</td>
<td>2.5Gb/s</td>
<td>7.81</td>
</tr>
<tr>
<td>2001</td>
<td>10Gb/s</td>
<td>31.25</td>
</tr>
<tr>
<td>2003</td>
<td>40Gb/s</td>
<td>125</td>
</tr>
<tr>
<td>2005</td>
<td>100Gb/s</td>
<td>312</td>
</tr>
</tbody>
</table>
Introduction to Packet Switching

- Background
  - What is a router?
  - Router architecture

- Packet processing in routers
  - IP address lookup
  - Packet buffering
  - Switching

- Router architectures
  - The evolution
  - Software-Defined Networking
Generic Router Architecture

- Header Processing
  - Lookup IP Address
  - Update Header
  - Address Table

- Buffer Manager
  - Buffer Memory

CSC 2229 - Software-Defined Networking

University of Toronto – Fall 2015
Fast Packet Buffers

Example: 40Gb/s packet buffer

Size = RTT*BW = 10Gb; 40 byte packets

Write Rate, R
1 packet every 8 ns

Buffer Manager

Read Rate, R
1 packet every 8 ns

Buffer Memory

Use SRAM?
+ fast enough random access time, but
- too low density to store 10Gb of data.

Use DRAM?
+ high density means we can store data, but
- too slow (50ns random access time).
Introduction to Packet Switching

• Background
  • What is a router?
  • Router architecture

• Packet processing in routers
  • IP address lookup
  • Packet buffering
  • Switching

• Router architectures
  • The evolution
  • Software-Defined Networking
Generic Router Architecture

Data Hdr

Header Processing

Lookup IP Address

Update Header

Address Table

1

Queue Packet

Buffer Memory

N times line rate

Data Hdr

Header Processing

Lookup IP Address

Update Header

Address Table

2

N times line rate

Data Hdr

Header Processing

Lookup IP Address

Update Header

Address Table

N

Queue Packet

Buffer Memory

N times line rate

N times line rate
Generic Router Architecture

Header Processing
- Lookup IP Address
- Update Header
- Address Table

Queue Packet
- Data
- Hdr

Address Table
- Buffer Memory

Packet Buffer
- Memory

Data Hdr

Scheduler

N

1

2

N
A Router with Output Queues

The best that any queueing system can achieve.
A Router with Input Queues

Head of Line Blocking

The best that any queueing system can achieve.

\[ 2 - \sqrt{2} \approx 58\% \]
Head of Line Blocking
Virtual Output Queues
A Router with Virtual Output Queues

The best that any queueing system can achieve.
Generic Router Architecture

- Data Hdr
  - Header Processing
    - Lookup IP Address
    - Update Header
    - Address Table

- Queue Packet
  - Buffer Memory

- N times line rate

- N times line rate

- N
Introduction to Packet Switching

- Background
  - What is a router?
  - Router architecture

- Packet processing in routers
  - IP address lookup
  - Packet buffering
  - Switching

- Router architectures
  - The evolution
  - Software-Defined Networking
First Generation Routers

Typically <0.5 Gb/s aggregate capacity
Second Generation Routers

Typically <5Gb/s aggregate capacity
Third Generation Routers

Switched Backplane

Typically <50Gb/s aggregate capacity
Fourth Generation Routers/Switches
Optics inside a router for the first time

0.3 - 10Tb/s routers in development
Introduction to Packet Switching

• Background
  • What is a router?
  • Router architecture

• Packet processing in routers
  • IP address lookup
  • Packet buffering
  • Switching

• Router architectures
  • The evolution
  • Software-Defined Networking
Software Defined Networks
Software Defined Networks – Cont’d

Controller

OpenFlow Switch

Data Plane

OpenFlow Protocol

SSL

PC
Final Comments/Questions

- What other line rate functionalities can you think of (other than packet forwarding)?
  - Are they performance bottleneck in today’s networks?
  - How would they evolve over time?
  - Do they belong on data path?

- Control and management complexity
  - What control functionalities do we have in networks today?
  - How do we manage those?
  - What can we do to simplify this?