E6998-02: Internet Routing

Lecture 1
Introduction

John Ioannidis
AT&T Labs – Research

ji+ir@cs.columbia.edu
Administrivia

Instructor: John Ioannidis, pronounced *jay-eye*
Email: <ji+ir@cs.columbia.edu>
- Mail to anything else will not be answered.

Class web page: http://www.cs.columbia.edu/~ji/F02/
- Check frequently!
- Slides will be available there.
- As will additional reading material (papers, RFCs, source code, man pages, etc.).

Class BBoard: TBA
Office hours: Tuesdays 16:00-17:30 or by appointment.
TA(s): TBD
TA office hours: TBD
The Necessary Evil

- Five homeworks: 50% of grade.
- Project: 50% of grade.
- Class participation: ±20% of grade.

- No midterm or final.
Homeworks

- 1-2 weeks each.
- Look at the web site for announcements.
- Don’t start the night before, you won’t finish it.
- Involve some coding, some reading, some writing.
- Useful for your project.
- Each counts for 10% of your grade.
- Submitted over email, TBD how.
- Individual submissions.
  - Cheaters will be given an F in the course and sent to the Dean.
Project

• Sizeable project to demonstrate your mastery of the material:
  – Coding.
  – Simulation.
  – Survey or original research paper.
• Half the grade; homeworks will help.
• Individual or in groups of up to 3 people.
  – Group gets grade; members decide how to apportion the grade.
Class Participation

• Fun for me.
• Shows you are paying attention!
• May affect your grade.
• Ask questions early and often:
  – I may be assuming that you know something when you don’t (frequent).
  – I may not be explaining something well enough (rare!).
• If you are having trouble with the material, talk to me or the TA. Don’t wait until it’s too late!
Prerequisites

• Formal:
  – W4118 and W4119.

• Essential:
  – Fluency in C, C++, and Java recommended.
  – Sockets programming.
  – Some kernel programming.
  – TCP/IP operations.
  – Have heard of the concept of routing.
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Sep</td>
<td>Intro, History</td>
<td>5-Sep</td>
<td>All about addressing</td>
</tr>
<tr>
<td>10-Sep</td>
<td>Layer 2 bridging</td>
<td>12-Sep</td>
<td>Routing and forwarding</td>
</tr>
<tr>
<td>17-Sep</td>
<td>Unix routing</td>
<td>19-Sep</td>
<td>Routing implementations</td>
</tr>
<tr>
<td>24-Sep</td>
<td>About IGPs</td>
<td>26-Sep</td>
<td>OSPF</td>
</tr>
<tr>
<td>1-Oct</td>
<td>Guest lecture, TBD</td>
<td>3-Oct</td>
<td>Guest lecture, TBD</td>
</tr>
<tr>
<td>8-Oct</td>
<td>Interdomain routing</td>
<td>10-Oct</td>
<td>BGP</td>
</tr>
<tr>
<td>15-Oct</td>
<td>BGP (cont'd)</td>
<td>17-Oct</td>
<td>More BGP</td>
</tr>
<tr>
<td>22-Oct</td>
<td>Even more BGP</td>
<td>24-Oct</td>
<td>BGP operations</td>
</tr>
<tr>
<td>29-Oct</td>
<td>BGP extensions</td>
<td>31-Oct</td>
<td>BGP extensions</td>
</tr>
<tr>
<td>5-Nov</td>
<td>ELECTION DAY</td>
<td>7-Nov</td>
<td>Overlay networks</td>
</tr>
<tr>
<td>12-Nov</td>
<td>Multicast</td>
<td>14-Nov</td>
<td>Mobility</td>
</tr>
<tr>
<td>19-Nov</td>
<td>Ad hoc networks</td>
<td>21-Nov</td>
<td>More on wireless</td>
</tr>
<tr>
<td>26-Nov</td>
<td>VPNs</td>
<td>28-Nov</td>
<td>THANKSGIVING DAY</td>
</tr>
<tr>
<td>3-Dec</td>
<td>Current Research</td>
<td>5-Dec</td>
<td>Future Research</td>
</tr>
<tr>
<td>10-Dec</td>
<td>Project demos</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Networking in a Nutshell

- *Nodes* (hosts and routers) connected by *links*.
- Each node has an *address*.
  - Unique (usually).
  - Used to find node in network.
- Internet is a *packet-switched network*.
- Based on IP, a *best-effort, connectionless* protocol.
- “The Network delivers bits” (and does little else).
  - No guaranteed delivery.
  - No guaranteed in-order delivery.
  - No guaranteed correct delivery.
- Additional functionality implemented at the end nodes.
- Ancillary protocols needed to make it work.
No Such Thing as Presentation Layer!

- Forget what you learned about the 7-layer model!
- Layering as a conceptual tool.
  - Lots of “layer violations” in practice.
  - Don’t be fundamentalist!
- Several layers:
  - Application
  - Transport
  - Network
    - Internet
    - Subnet
  - Link
  - Physical

We’ll be concentrating on these
PDUs, PSUs, and other TLAs

- Protocol Data Units/Protocol Service Units.
- ISOisms, we don’t like them – we don’t use them.

- Link and below: “frame” (ethernet frame, PPP frame).
- Transport:
  - “packet” (UDP packet, ICMP packet).
  - “segment” (TCP segment).
  - Some implication about block-oriented vs. stream-oriented abstractions.
- Above transport: wrong course.
Addresses, Names, Routes

- Name: “who/what it is”
- Address: “where it is”
- Route: “how to get there”

- Human-readable?
- Hierarchical/topological?
- By layer?
- By need for protocols?

- What is “20.16.59.128.in-addr.arpa”? 
Functional Definitions

- **Name**: location-independent.
  - Aka “Identifiers”
  - “Henry Kissinger”
- **Address**: dependent on entity location only.
  - Place in the topology/geography.
  - “1 Microsoft Way, Redmond, WA”
- **Route**: depends on both source and destination.
  - Directions on how to get there.
  - “Follow path, take left fork, climb stairs, through gate”.

Sometimes we’ll use these terms loosely.

What’s an email address?

There are protocols to get one from the other.
Names/Addresses/Routes cont’d

• Various namespaces:
  – DNS names, human names, service names.
  – IP addresses, MAC addresses, phone numbers.
  – IP routes, token-ring source routes, mail relays.

• Tools to translate:
  – DNS, NIS, Hesiod.
  – DNS, ARP, NDP.
  – Routing protocols.

• Many-to-many relationships.
Routing in a Nutshell

- Node has packet to send.
- If it is for that node, send it up to higher-layer protocols.
- Figure out which network interface to send it out on and/or what the next hop is.
- Iterate.

- Forwarding: the problem of deciding which interface to send it out on.
  - Code/hardware on nodes (hosts/routers) does that.
- Routing: the problem of figuring out which interface/next hop is appropriate.
  - Combination of code on nodes and routing protocols.
Routing can happen at all layers

- Physical: move connectors around!
- Link: “bridging”
- Network: “routing”
- Transport: redirectors (avoid).
- Application: ALG

- We shall briefly examine bridging.
- We shall mostly learn about network-layer routing.
LANs

- Nodes connected to a common medium.
  - Ethernet, radio, etc.
- Nodes “hear” each other’s transmissions.
  - Broadcast/multicast.
- Nodes talk to each other directly.
- Limited in number/scope/location.

- Meaning of “node”, “medium”, “transmission” and “directly”.

- “Subnet layer”
Link-layer Addressing

• You should already know about this.
• Ethernet-like LANs, MAC address:
  – 48-bit, unique.
  – Flat namespace as far as addressing is concerned.
  – Appears at the beginning of a frame.
  – It’s a name, really.
  – Unicast/multicast addresses.
• Point-to-point connections:
  – No need for a station address.
  – Still need for service/higher layer protocol identifiers.
Connecting LANs

• Plugging them together usually not an option.
  – Distance limitations.
  – Capacity limitations.
  – Administrative/security limitations.