Learning Goals

• Be comfortable with terminology used concerning the Internet
• Understand the role of protocols, and the layering of protocols, in the architecture of the Internet. And how this layering provides levels of abstraction below which a developer need not be (too) concerned.
• Understand the basic functionality of how packets of information travel between one system and another. This will inform design and configuration choices in building and maintaining systems.
• Understand IP addressing.
Joe's Pedagogy

• In-Class:
  – Build a mental model of how things work

• Out-of-Class Studying:
  – Learn terminology
  – Memorize concepts
A Simple Example Internetwork

Network 2 (Ethernet)

Network 3 (Gigabit Ethernet)

Network 1 (Wifi)

Network 4 (point to point link)
Protocol Layering

H1
- TCP
- IP
- ETH

R1
- IP
- ETH
- GbE

R2
- IP
- GbE
- PPP

R3
- IP
- PPP
- ETH

H8
- TCP
- IP
- ETH

Protocol Layering
### Some Common Application Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
<td>Browser to server, web app to web service, etc.</td>
</tr>
<tr>
<td>HTTPS</td>
<td>HTTP Secure</td>
<td>Secure HTTP</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
<td>Server assigns IP address to a host</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
<td>Map domain name to IP address</td>
</tr>
<tr>
<td>RTP/RTCP</td>
<td>Real-time Transport Protocol / Real-time Transport Control Protocol</td>
<td>Delivers audio and video streams / statistics and control information about RTP stream</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
<td>Querying and modifying directory services information</td>
</tr>
<tr>
<td>POP</td>
<td>Post Office Protocol</td>
<td>Retrieve email from a server</td>
</tr>
<tr>
<td>IMAP</td>
<td>Internet Message Access Protocol</td>
<td>Retrieve email from a server</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
<td>Send mail from client to server, and send mail to/from server to server</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
<td>Terminal-type (shell) access to remote computers</td>
</tr>
</tbody>
</table>
• Start with Question 2 Step 3 (page 2)
• If making an HTTP request, what is an example of the start of the *Application Data*?
• How will that request get to the destination?
Two common transport level protocols

- UDP
- TCP

Both facilitate communication:
- From a port at one IP address
- To a port on another IP address

The communication metaphor between the two ports is called a “socket”.

E.g.
- A UDP socket
  - from 128.2.25.11 port 5443
  - to 204.100.2.1 port 4568
  - Commonly noted 204.100.2.1:4568

E.g.
- A TCP socket
  - Between 128.2.25.22:4567
  - And 204.100.2.2:5432
TCP/UDP Ports

• Port is a transport-layer software construct
• Port is a 16 bit integer
• 65,536 different Ports an application can communicate on
Official ports are assigned by IANA
Ports

• 0 - 1023 Well Known Ports
  – Should not be used without IANA registration
  – Loosely, system-type-services ports

• 1024 - 49151 Registered Ports
  – Also should not be used without IANA registration
  – Loosely, application-level ports

• 49152 - 65535 Dynamic and/or Private Ports
  – Anything

• Readable list of port assignments:
• Application – application communicates with another application via a sockets connection on a port
• TCP & UDP – provides a socket and port abstraction across computers
• IP – hosts communicate with routers and other hosts
Transport Layer Protocols - UDP

- Similar with UDP
- TCP and UDP don’t mix.
## TCP & UDP Comparison

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>UDP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Simple, high speed, low functionality wrapper that interfaces with IP</td>
<td>Full-featured protocol to reliably communicate data with another application across IP.</td>
</tr>
<tr>
<td>Connection</td>
<td>Connectionless, no setup</td>
<td>Connection-oriented; setup prior to transmission</td>
</tr>
<tr>
<td>Interface to Application</td>
<td>Discrete message based</td>
<td>Stream based</td>
</tr>
<tr>
<td>Reliability and ACK</td>
<td>No reliability guaranteed, best effort delivery without acknowledgement</td>
<td>Reliable, all messages acknowledged</td>
</tr>
<tr>
<td>Retransmissions</td>
<td>None. Application must detect if needed</td>
<td>All lost data retransmitted automatically</td>
</tr>
<tr>
<td>Flow control</td>
<td>None</td>
<td>Flow control on both ends (sliding window)</td>
</tr>
<tr>
<td>Overhead</td>
<td>Low</td>
<td>Low, but not as low as UDP</td>
</tr>
<tr>
<td>Speed</td>
<td>Very high</td>
<td>High, but not as high as UDP</td>
</tr>
<tr>
<td>Data quantity</td>
<td>Single datagram - up to 65K bytes</td>
<td>Small to very large (gigabytes)</td>
</tr>
<tr>
<td>Applicability</td>
<td>Speed matters more than completeness. Small discrete messages. <strong>Multicast or broadcast</strong></td>
<td>Data must be received reliably, in order.</td>
</tr>
</tbody>
</table>
Internet Addressing Practice Sheet

• What Transport Protocol will HTTP use?
• On Question 2 Step 3, fill in the TCP Segment Header information.
Protocol Layering
### Decimal representation of IP addresses

#### Class A:
- **Range of addresses:** 1.0.0.0 to 127.255.255.255
- **Example:** IBM (9.0.0.0 - 9.255.255.255)

#### Class B:
- **Range of addresses:** 128.0.0.0 to 191.255.255.255
- **Example:** CMU (128.2.0.0 - 128.2.255.255)

#### Class C:
- **Range of addresses:** 192.0.0.0 to 223.255.255.255
- **Example:** (192.0.0.0 - 192.255.255.255)

#### Class D (multicast):
- **Range of addresses:** 224.0.0.0 to 239.255.255.255
- **Example:** (224.0.0.0 - 239.255.255.255)

#### Class E (reserved):
- **Range of addresses:** 240.0.0.0 to 255.255.255.255
- **Example:** (240.0.0.0 - 255.255.255.255)

<table>
<thead>
<tr>
<th>Octet 1</th>
<th>Octet 2</th>
<th>Octet 3</th>
<th>Range of addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network ID</td>
<td>1 to 127</td>
<td>0 to 255</td>
<td>1.0.0.0 to 127.255.255.255</td>
</tr>
<tr>
<td>Network ID</td>
<td>128 to 191</td>
<td>0 to 255</td>
<td>128.0.0.0 to 191.255.255.255</td>
</tr>
<tr>
<td>Network ID</td>
<td>192 to 223</td>
<td>0 to 255</td>
<td>192.0.0.0 to 223.255.255.255</td>
</tr>
<tr>
<td>Network ID</td>
<td>224 to 239</td>
<td>0 to 255</td>
<td>224.0.0.0 to 239.255.255.255</td>
</tr>
<tr>
<td>Network ID</td>
<td>240 to 255</td>
<td>0 to 255</td>
<td>240.0.0.0 to 255.255.255.255</td>
</tr>
</tbody>
</table>

**Test question:** Who allocates IP addresses?
IP Addressing

• Every IP datagram contains the IP address of the destination host.
• The network part of an IP address uniquely identifies a single network that is part of the larger Internet.
• All hosts and routers that share the same network part of their address are connected to the same network and can thus communicate with each other by sending frames over the network, addressed with MAC addresses.
• The network part of the IP address, therefore, helps a packet find the right network.
• Every network that is part of the Internet has at least one router that, by definition, is also connected to at least one other network; this router can exchange packets with hosts or routers on either network.
• Subnetting allows an organization to break their address space into smaller networks.

Source: http://en.wikipedia.org/wiki/Subnetwork
• Similarly within CMU
  – Instead of one network 128.2 with 65K hosts
  – Network admins can use subnet masks to break it into multiple sub-networks (subnets).
    • A few top bits of the 3rd byte can be the subnet.
    • The remaining bits will be the host.
    • Note that this will not always fall only on “dot” boundaries.

<table>
<thead>
<tr>
<th></th>
<th>Dot-decimal notation</th>
<th>Binary form</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>192.168.5.130</td>
<td>11000000.10101000.00000101.10000010</td>
</tr>
<tr>
<td>Subnet Mask</td>
<td>255.255.255.0</td>
<td>11111111.11111111.11111111.00000000</td>
</tr>
<tr>
<td>Network Portion</td>
<td>192.168.5.0</td>
<td>11000000.10101000.00000101.00000000</td>
</tr>
<tr>
<td>Host Portion</td>
<td>0.0.0.130</td>
<td>00000000.00000000.00000000.10000010</td>
</tr>
</tbody>
</table>
Where do IP addresses come from?

- Where does your IP address come from?
  - Dynamic – Dynamic Host Configuration Protocol
  - Static – from a network administrator
DHCP Negotiation

To broadcast, send to IP address: 255.255.255.255 whichdatalink sends to MAC address: FF:FF:FF:FF:FF:FF

Source: http://docs.hp.com/en/B2355-90685/ch06s02.html
DHCP Offer

- IP Address being offered to the client
- Subnet mask for the local network
- Gateway IP (default router out of this network)
- DNS servers
- Lease duration
Internet Addressing Practice Sheet

• Do Question 1
• In Question 2 Step 3, add IP Datagram Header
Each networking device has a MAC address
  – Media Access Control address

Fixed by manufacturer
  – (Can be spoofed, for good or by Eve/Mallory)

A laptop my have a few:
  – Wifi: 60:fb:42:f8:b5:08
  – Bluetooth: 60:fb:42:72:08:4c

A smartphone could also have a couple:
  – Wifi
  – Bluetooth
  – And an IMEI number, which is like a mobile MAC
Important distinction

• MAC addresses are used to pass messages around a single physical network segment
  – An ethernet segment is a network
  – A wifi access point forms a wireless network segment
  – A point-to-point microwave link is a network segment

• IP addresses are used to pass messages around between networks
  – I.e. between the wifi LAN in this room and google.com
  – IP == Internet Protocol
    • Internet meaning between networks
Printing on a local network

IPP = Internet Printing Protocol, an Application Layer Protocol
• Host H1 wants to contact host H2 on the same network.
• First, H1 checks its cache to see if it already contains the IP / MAC address pair.
  – If it does then use the MAC address.
• If it does not
  – H1 broadcasts the IP address to all hosts on this network.
    • Broadcast MAC is FF:FF:FF:FF:FF:FF
  – The matching host H2 sends back its MAC address.
  – H1 then adds this mapping to its cache.
• Do Question 2 Steps 1 and 2
• Now you have info to do the Ethernet Frame Header for Question 2 Step 3
  – Complete it
• Next do Question 2 Step 4
A Simple Example Internetwork

Network 2 (Ethernet)

Network 3
(Gigabit Ethernet)

Network 1 (Wifi)

Network 4
(point to point link)
Routing?

• How did R1 know R2 was the best path to H8?

• My diagrams only showed an obvious path, but but the Internet looks like this:

Source: http://www.cheswick.com/ches/map/movie.mpeg
Network vs Internetwork

- An internetwork is an interconnected collection of networks.
- The Internet Protocol (IP) is the key tool used today to build scalable, heterogeneous internetworks.
nut:~ JoeMertz$ traceroute -I health.gov.ck
traceroute to health.gov.ck (202.65.43.185), 64 hops max, 72 byte packets
1 pod-a-weh-vl88.gw.cmu.net (128.2.80.3) 1.094 ms 0.652 ms 0.637 ms
2 core255-v1919.gw.cmu.net (128.2.255.161) 0.934 ms 0.688 ms 0.677 ms
3 pod-i-nh-vl987.gw.cmu.net (128.2.255.251) 0.929 ms 0.873 ms 0.823 ms
4 ge-7-23.carl.pittsburgh3.level3.net (4.49.108.45) 1.174 ms 1.169 ms 1.114 ms
5 ae-5-5.ebr1.washington1.level3.net (4.69.135.242) 17.289 ms 17.162 ms 17.947 ms
6 ae-91-91.csw4.washington1.level3.net (4.69.134.142) 19.895 ms 16.998 ms 18.090 ms
7 ae-4-99.edge2.washington4.level3.net (4.68.17.211) 7.646 ms 7.966 ms 7.682 ms
8 if-8-0.icore1.aeq-ashburn.as6453.net (206.82.139.65) 12.407 ms 16.747 ms 18.806 ms
9 if-0-0-0-459.core4.aeq-ashburn.as6453.net (216.6.42.21) 7.971 ms 8.485 ms 8.276 ms
10 if-3-0-0-921.core1.ct8-chicago.as6453.net (66.110.27.53) 20.337 ms 22.613 ms 20.412 ms
11 if-0-0-0-925.core1.s00-seattle.as6453.net (207.45.206.1) 91.743 ms 82.861 ms 81.951 ms
12 if-5-0.core1.vcw-vancouver.as6453.net (207.45.196.54) 83.803 ms 83.718 ms 83.970 ms
13 if-0-0.core1.lcn-lakecowichan.as6453.net (216.6.58.5) 86.043 ms 86.134 ms 86.489 ms
14 if-5-0-0.bb2.lcn-lakecowichan.as6453.net (64.86.83.39) 89.394 ms 88.493 ms 87.112 ms
15 ix-10-1-0.bb2.lcn-lakecowichan.as6453.net (64.86.84.2) 652.794 ms 605.698 ms 628.306 ms
16 202.65.32.18 (202.65.32.18) 624.766 ms 642.546 ms 621.597 ms
17 thealth.oyster.net.ck (202.65.43.185) 624.677 ms 740.438 ms 683.941 ms
• Routers maintain routing tables, indicating the best paths to other networks.
• At least every 30 seconds (or earlier when their own table changes) each router will send its routing table to each of its neighbors.
• When receiving a neighbor’s table, a router will update its own tables.
### Receive: Whenever a routing table $T_b$ is received on link $n$:

for all rows $R_b$ in $T_b$ {
  if $(R_b.link \neq n)$ {
    // Ignore the route if it was through me, for I know better
    $R_b.cost = R_b.cost + 1$;
    // This route from me would have higher cost by 1
    $R_b.link = n$;
    // and I would travel through $n$
    if $(R_b.destination$ is not in $T_a)$ add $R_b$ to $T_a$;  // if a new destination, just add to $T_a$
  } else for all rows $R_a$ in $T_a$ {
    if $(R_b.destination = R_a.destination$ and
      $(R_b.cost < R_a.cost$ or $R_a.link = n))$ $R_a = R_b$;
    // $R_b.cost < R_a.cost$ : remote node has better route
    // $R_a.link = n$ : remote node is more authoritative
  }
}}

### Routing Information Protocol (RIP)

<table>
<thead>
<tr>
<th></th>
<th>Route A</th>
<th>Route B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{To}$</td>
<td>$\text{Link}$</td>
<td>$\text{Cost}$</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

- **Ta**: Local table
- **Tb**: Remote table

**Diagram:**

- Hosts or local networks
- Links
- Routers
- **Distributed Systems**
Update A With B

Receive: Whenever a routing table Tb is received on link n:
for all rows Rb in Tb {
  if (Rb.link ≠ n) {
    // Ignore the route if it was through me, for I know better
    Rb.cost = Rb.cost + 1;
    // This route from me would have higher cost by 1
    Rb.link = n;
    // and I would travel through n
    if (Rb.destination is not in Ta) add Rb to Ta;
    // if a new destination, just add to Ta
  }
  else for all rows Ra in Ta {
    if (Rb.destination = Ra.destination and
      (Rb.cost < Ra.cost or Ra.link = n)) Ra = Rb;
    // Rb.cost < Ra.cost : remote node has better route
    // Ra.link = n : remote node is more authoritative
  }
}}
Types of Routing Algorithms

1. Interior Gateway Protocols
   Interior == Within an ISP or large organization
   a. Vector-Distance Algorithms
      • RIP is the simplest example
   b. Link-State Algorithms
      • Distribution and update of a DB at each node
      • Representing all or a substantial portion of an enterprises’ network
      • Each router has a map of the network, with which to calculate shortest path to each other node
      • Collection of shortest paths becomes the routing table

2. Exterior Gateway Protocols
   Exterior == Between ISPs or large organizations
   – Border Gateway Protocol
   – Vector-Distance Algorithm, with the addition of policy information such as “this link is cheaper” or “only use that link as a backup”
Review

- **Application layer**
  - Application code
  - Alternatively: Application protocols
    - Application code would be above them

- **Transport layer**
  - TCP and UDP

- **Internet layer**
  - IP – Internet Protocol

- **Data Link Layer**
  - MAC - ARP
1) What are valid IP addresses for H1-H4?

2) What are the range of possible IP addresses in each case?

3) How is an HTTP request passed via TCP/IP from H1 to H2? And the reply?

4) How is an HTTP request passed via TCP/IP from H1 to H4? (once H4’s IP address is known)? And the reply?

Assume Subnet masks of 255.255.255.248 (248=11111000 in binary)