Course Overview

- **Fundamental Question:**
  - How is data transferred through the Internet?

- **Principles to develop**
  - Reliable data transfer, to the correct recipient
  - Fast & error-free data transfer
  - Security and privacy safeguards

- **Implementation**
  - Network layers & Protocols
Packet delay: A packet’s trip

- A packet arrives at a router, and...
  1. The ’header’ is read for source and destination hosts (IP address), and perform error checking of the bits transmitted = 
  2. If other packets arrived first and are waiting in the output buffer, there is: 
  3. The rate at which the router can upload the bits onto the physical link = 
  4. The time to travel from one router to the next router = 

Which of these delays are constant and which are variable?

Self-review Question

- Exploring propagation delay and transmission delay.
- Consider 2 hosts, A and B, connected by a single link of rate $R$ bps. Suppose that the two hosts are separated by $d$ meters and the propagation speed is $s$ m/s. Host A sends a packet of size $L$ to Host B.
  - Find $d_{\text{prop}}$ (using what information?)
  - Find $d_{\text{trans}}$ (using what information?)
**Internet Layers: Services (first glimpse)**

- **Application layer:**
  - User interface

- **Transport layer:**
  - Reliable data transfer

- **Network layer:**
  - Find the best path through the network

- **Link layer:**
  - Transfer frames along shared links

- **Physical layer:**
  - Transfer bits along one link

---

**Network Applications**

Programs that
- run on different end systems and
- communicate over a network.
Transport services and protocols

- **Logical communication** between **application processes** running on different hosts
- Provides reliability (TCP)

Hierarchical Internet structure: network of networks

- Each packet passes through many networks
Internet structure: network of networks

Traceroute

- Provides delay measurement
- Source host sends three packets to each router $i$ on path to destination host
  - For all routers in the selected path (route)
- Router $i$ returns packets to sender
  - Sender times interval between transmission and reply.
“Real” Internet delays and routes, play around on your own

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu
1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 chf-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms
9 de2.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 rio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3l2-nice.cssi.renater.fr (195.220.96.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3l2.ft.net (193.48.50.64) 135 ms 128 ms 133 ms
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms
17 *** means no response (probe lost, router not replying)
18 ***
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

“Real” Internet delays and routes, play around on your own

- Traceroute:
  - [http://ping.eu/traceroute/](http://ping.eu/traceroute/)

- PingPlotter freeware - might be fun
  - [http://www.pingplotter.com/freeware.html](http://www.pingplotter.com/freeware.html)
Your IP is **131.229.102.127**

**Online service Traceroute**

- **Traceroute** - Traces the route of packets to destination host from our server.

**IP address or host name:** 8.8.8.8

| Traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets |
|-------------------|-------------------|-------------------|-------------------|-------------------|
| 1 static.121.168.446.de.client.your-server.de | 46.4.168.121 | 0.995 ms | 1.081 ms | 1.117 ms |
| 2 hop-t01.juniper-r135.lan.de | 213.239.224.97 | 0.135 ms | 0.135 ms | 0.135 ms |
| 3 core21.zhner.de | 213.239.245.81 | 0.405 ms | 0.398 ms | 0.398 ms |
| 4 core21.zhner.de | 213.239.245.121 | 0.379 ms | 0.379 ms | 0.379 ms |
| 5 google2браzil.net | 62.69.146.10 | 4.879 ms | 4.893 ms | 4.900 ms |
| 6 google2браzil.net | 62.69.146.10 | 4.879 ms | 4.893 ms | 4.900 ms |
| 7  |  |  |  |  |
| 8  |  |  |  |  |
| 9  |  |  |  |  |

No reply for 3 hops. Assuming we reached firewall.

---

| IP address or host name:  | web.edu |  |  |
|-------------------|-------------------|-------------------|-------------------|-------------------|
| Traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets |
| 1 static.121.168.446.de.client.your-server.de | 46.4.168.121 | 3.066 ms | 3.572 ms | 3.572 ms |
| 2 hop-t01.juniper-r135.lan.de | 213.239.224.97 | 0.399 ms | 0.399 ms | 0.399 ms |
| 3 core21.zhner.de | 213.239.245.81 | 0.367 ms | 0.367 ms | 0.367 ms |
| 4 core21.zhner.de | 213.239.245.121 | 0.350 ms | 0.350 ms | 0.350 ms |
| 5 google2браzil.net | 62.69.146.10 | 3.012 ms | 3.012 ms | 3.012 ms |
| 6 google2браzil.net | 62.69.146.10 | 3.012 ms | 3.012 ms | 3.012 ms |
| 7  |  |  |  |  |
| 8  |  |  |  |  |
| 9  |  |  |  |  |

No reply for 3 hops. Assuming we reached firewall.
Also could try...

- (a) Visit the site www.traceroute.org and perform traceroutes from two different cities in France to the same destination host in the United States. How many links are the same in the two traceroutes? Is the transatlantic link the same?

- (b) Repeat (a) but this time choose one city in France and another city in Germany.

- (c) Pick a city in the United States, and perform traceroutes to two hosts, each in a different city in China. How many links are common in the two traceroutes? Do the two traceroutes diverge before reaching China?
Vocabulary for the Application Layer

- Protocol – a set of steps to follow
- Application packet is a “message”
- Architecture
  - Client – Server vs. Peer to Peer (P2P)
- Application ≈ Process
- Port number (assigned to each application)
- TCP Connection & handshaking

What is a protocol?

A human protocol and a computer network protocol:

- Hi
- Hello
- Do you know the time?
- 2:00
- HTTP Client request
- HTTP Server response
- Get http://www.smith.edu
- <file>
At the tables now, play with some web browsers

- Open a web browser
- Type in a URL
- Brainstorm everything that happens after you press the ‘enter’ or ‘return’ key
  - What is the meaning of every element in the URL?
  - What happens at your ‘source’ computer?
  - What happens in the Internet?
  - What is the destination host and what does that host do?

HTTP Principles

- Characteristics
- Message format
  - telnet example
- Cookies
- Proxy cache
HTTP overview

Connections between Hosts

- **Persistent connections**
  - Connection from source to destination is kept open after the initial message and data exchange
  - vs. non-persistent

- **Pipelining**
  - Multiple connections are opened in order to allow sending multiple files simultaneously, such as images on a webpage
Non-persistent HTTP (for comparison)

Suppose user enters URL
www.amherst.edu/history/home.index

1a. HTTP client initiates TCP connection on port 80

2. HTTP client sends HTTP request message into TCP connection socket.

1b. HTTP server at host www.amherst.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

3a. HTTP server receives request message, forms response message containing requested object, and sends message into its socket, and...

3b. HTTP server closes TCP connection.

4. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

5. Steps 1-5 repeated for each of 10 jpeg objects
Non-Persistent HTTP: Response time

Persistent HTTP

Nonpersistent HTTP
- server closes connection after first file is transferred, and then must repeat connection procedure

Persistent HTTP
- server leaves (TCP) connection open after sending response
- subsequent HTTP messages between same client/server sent over the existing, open connection

Persistent without pipelining:
- client issues new request only when previous response has been received

Persistent with pipelining:
- default in HTTP/1.1
- client sends requests as soon as it encounters a referenced object
HTTP Review

1) Connection request to port 80 (handshaking)
2) File/data request via dedicated connection/socket

1a. HTTP client initiates TCP connection on port 80

2. HTTP client sends HTTP request message into TCP connection socket.

3. HTTP server receives request message, forms response message containing requested object, and sends message into its socket.

Client Server Review

- Client

- Server
Basic HTTP request message

- ASCII (human-readable format)

HTTP request message: general format

When is the “Entity Body” filled in with an HTTP request message?
Example: Uploading form input

GET method: (the most basic)
- The 'entity body' in the request is empty
- Also can try the HEAD method (page header info)

Post method:
- A web page often includes “form” input
- This input information is uploaded to server in the message’s “entity body”
(but also have the URL method: )
- Uses the GET method
- Input is uploaded in URL field of request line:
  www.somesite.com/animalsearch?monkeys&banana

Basic HTTP response message

HTTP/1.1 200 OK
Connection close
Date: Fri, 12 Sep 2014 2:31:16 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 2014 ...
Content-Length: 6821
Content-Type: text/html

data data data data data data ...
HTTP response status codes

In the first line of a server->client response message.
A few sample codes:

200 OK
   ▷ request succeeded, requested object later in this message

301 Moved Permanently
   ▷ requested object moved, new location specified later in this message (Location:)

400 Bad Request
   ▷ request message not understood by server

404 Not Found
   ▷ requested document not found on this server

505 HTTP Version Not Supported

---

Trying out HTTP (client side) for yourself

1. Telnet to a Web server (from ‘Terminal’ window):

   telnet www.science.smith.edu 80

   Opens TCP connection to port 80 (default HTTP server port)
   Anything typed in sent to port 80 at science.smith.edu

2. Type in a GET HTTP request:

   GET /~jcardell/ HTTP/1.1
   Host: www.science.smith.edu

   This is a minimal (but complete) GET request to an HTTP server
Trying out HTTP (client side) for yourself

3. Look at response message sent by HTTP server

HTTP/1.1 200 OK
Date: Tue, 23 Jan 2018 19:13:04 GMT
Server: Apache/2.4.7 (Ubuntu)
Last-Modified: Fri, 31 Aug 2014 20:08:20 GMT
ETag: "a95-506bd8eeec7500"
Accept-Ranges: bytes
Content-Length: 2709
Vary: Accept-Encoding
Content-Type: text/html

(... followed by the HTML file)
Trying out HTTP

What should you type to initiate an HTTP connection via telnet?

jcardell-fcap:~ jcardell$ telnet www.science.smith.edu
Trying 131.229.72.74...

telnet: connect to address 131.229.72.74: Operation timed out
telnet: Unable to connect to remote host

jcardell-fcap:~ jcardell$

Trying out HTTP

jcardell-fcap:~ jcardell$ telnet www.science.smith.edu 80
Trying 131.229.72.74...
Connected to cirrus.smith.edu.
Escape character is '^]'.
HEAD /~jcardell/ HTTP/1.1
Host: www.science.smith.edu  
HTTP/1.1 200 OK
Date: Tue, 23 Jan 2018 19:13:04 GMT
Server: Apache/2.4.7 (Ubuntu)
Last-Modified: Fri, 31 Aug 2014 20:08:20 GMT
ETag: "a95-506bd8ee7500"
Accept-Ranges: bytes
Content-Length: 2709
Vary: Accept-Encoding
Content-Type: text/html
telnet gaia.cs.umass.edu 80
Trying 128.119.245.12...
Connected to gaia.cs.umass.edu.
Escape character is '^[].'
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
Host: gaia.cs.umass.edu

HTTP/1.1 200 OK
Date: Thu, 02 Feb 2017 21:07:00 GMT

Last-Modified: Thu, 02 Feb 2017 06:59:01 GMT
ETag: "80-54786b333b730"
Accept-Ranges: bytes
Content-Length: 128
Content-Type: text/html; charset=UTF-8

<html>
Congratulations. You've downloaded the file
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html!
</html>
Connection closed by foreign host.

Fun With Telnet

- Terminal emulation, for UNIX, to log on to remote computers
- A protocol and an application (using that protocol)
- Poke around on http://www.telnet.org/
http://www.telnet.org/

Miscellaneous fun places
- rainmaker.wunderground.com :: weather via telnet!
- india.colorado.edu 13 (Get the time) :: get the time
- telehack.com 23 :: Telehack
- telehack.com :: Telehack - web
- towel.blinkenlights.nl 23 :: Star Wars asciiation
- towel.blinkenlights.nl 666 :: The Bofh Excuse Server

Other systems
- thehatshop.mudhosting.net 3000 :: Hallowed Halls
- eclipse.cs.pdx.edu 7680 :: New Moon
- batmud.bat.org 23 :: BatMUD
- forgottenkingdoms.org 4000 :: Forgotten Kingdoms
- mush.shelteringcolorado.com 2601 :: Sheltering Sky: Colorado by Night
- igormud.org 1701 :: Igor MUD/
- zombiemud.org 23 :: Zombie MUD
- achaea.com 23 :: Achaea, Dreams of Divine Lands
- galacticomm.com 23 :: Galacticomm BBS
- 1984.ws 23 :: 1984

Ping - To play with on own

- Use a terminal window, or a web client
  - Such as http://ping.eu/ping/
- Addresses to test:
  - Google DNS servers: 8.8.8.8 and 8.8.4.4.
  - Australia at 139.130.4.5
  - OpenDNS 208.67.222.222 and 208.67.220.220
  - Norton Connectsafe: 198.153.192.1 and 198.153.194.2 that respond to ICMP requests
  - Yourself: 127.0.0.1
HTTP Characteristics
- Client-server architecture
- Is “stateless”
  - Compare to FTP - read through text sections
- Persistent connection
  - and pipelining
- Push vs. pull protocol
- Uses TCP (rather than UDP)

Summary
- Recap causes of delay in the Internet
- Client-Server architecture
- Hypertext Transfer Protocol
  - Messages: request & response
  - Message format
- Playing with telnet
  - Pretending you are a web browser
  - Be careful not to make more of this than intended