CSCI-1680
Transport Layer Warmup (ish)

Nick DeMarinis

Based partly on lecture notes by Rodrigo Fonseca, Jennifer Rexford, Rob Sherwood, David Mazières, Phil Levis, John Jannotti
• IP: Due 11:59pm tonight (March 22)
  – My office hours: 3-5pm
  – Or whenever my office door says “come in”

• Updated late policy

• How to submit?
  – Just push your code to your repo
  – Want to keep a specific version for grading? Make a branch called “ip_submit”
Administrivia: Grading for IP

• Tomorrow: look for email about interactive grading
  – Some slots this week, some after break

“Between the time you’ve handed in and the demo meeting, you can continue to make minor tweaks and bug fixes, but you shouldn’t be making any major changes.”
  – OK: Fixing bugs, code cleanup
  – Not OK: Implementing RIP
Administrivia: This week

• Thursday
  – (Light) start to TCP
  – TCP assignment out (just read it before spring break)

• Grades back on Snowcast (via email), HW2
Layers, Services, Protocols

- **Application**
  - Service: user-facing application.
  - Application-defined messages

- **Transport**
  - Service: multiplexing applications to "ports"
  - Reliable byte stream to other node (TCP),
  - Unreliable datagram (UDP)

- **Network**
  - Service: move packets to any other node in the network
  - Internet Protocol (IP)

- **Link**
  - Service: move frames to other node across link.
  - May add reliability, medium access control

- **Physical**
  - Service: move bits to other node across link
The Network Layer

- Goal: move packets between hosts (anywhere on Internet)
The Transport Layer

Services for getting data to *applications* on a single host

- Multiplexing at same IP via *port numbers*
- Connected-ness
- Reliable data delivery
- Series of packets => Data stream
- Throughput management/Congestion control
What’s a port number?

- Ports define a communication endpoint, which is usually a process or service running on a host
- 16-bit unsigned number, 0-65535
- Ports numbered < 1024: “Well known port numbers”  
  - Allocated by IANA
- Higher numbers: “ephemeral ports”  
  - For general use by any application  
  - Very high numbers (>20K): usually for source ports
## Some common ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>20, 21</td>
<td>File Transfer Protocol (FTP)</td>
</tr>
<tr>
<td>22</td>
<td>Secure Shell (SSH)</td>
</tr>
<tr>
<td>23</td>
<td>Telnet (pre-SSH remote login)</td>
</tr>
<tr>
<td>25</td>
<td>SMTP (Email)</td>
</tr>
<tr>
<td>53</td>
<td>Domain Name System (DNS)</td>
</tr>
<tr>
<td>67, 68</td>
<td>DHCP</td>
</tr>
<tr>
<td>80</td>
<td>HTTP (Web traffic)</td>
</tr>
<tr>
<td>443</td>
<td>HTTPS (Secure HTTP over TLS)</td>
</tr>
</tbody>
</table>
Sockets

The kernel maps ports to sockets, which are used in applications like file descriptors to access the network

Two modes for using ports/sockets:

• Applications “bind” to a port to accept new connections

• Individual connections use 5-tuple of source-dest port
  (protocol, source IP, source port, dest IP, dest port) => connection N
Netstat

deemer@vesta ~/Development % netstat -an

Active Internet connections (including servers)

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>(state)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>10.3.146.161.51094</td>
<td>104.16.248.249.443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>10.3.146.161.51076</td>
<td>172.66.43.67.443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp6</td>
<td>0</td>
<td>0</td>
<td>2620:6e:6000:900.51074</td>
<td>2606:4700:3108::443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>10.3.146.161.51065</td>
<td>35.82.230.35.443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>10.3.146.161.51055</td>
<td>162.159.136.234.443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>10.3.146.161.51038</td>
<td>17.57.147.5.5223</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp6</td>
<td>0</td>
<td>0</td>
<td>*.51036</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>*.51036</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>127.0.0.1.14500</td>
<td><em>.</em></td>
<td>LISTEN</td>
</tr>
</tbody>
</table>
Ports are part of the transport layer

- Port numbers are the first two fields of these headers! (Not part of IP!)
An interface to applications

• Ports define an interface to applications
• If you can connect to the port, you can (usually) use it!
• Problems?
Port scanning

What can we learn if we just start connecting to well-known ports?

• Can discover things about the network
• Can learn about vulnerabilities
Large-scale port scanning

- Can reveal lots of open/insecure systems!
- Examples:
  - shodan.io
  - VNC roulette
  - Open webcam viewers..
  - ...
Disclaimer

• Network scanning is easy to detect

• Unless you are the owner of the network, it’s seen as malicious activity

• If you scan the whole Internet, the whole Internet will get mad at you (unless done very politely)

• Do NOT try this on the Brown network. I warned you.
Scanning I have done

- Scanned IPv4 space for ROS (Robot Operating System)
- Found ~200 “things” using ROS (some robots, some other stuff)