Section 1: Socket API & Traceroute

(§1.3.4, 6.1.2-6.1.4)
Outline

- Administrivia
- Project 1: Socket API
- Traceroute
Administrivia

- Sections will be recorded
- Different weeks will be led by different TA’s
- HW1 due Monday Apr 12
- Project 1 due Monday Apr 19
Network-Application Interface

- Application Layer APIs
  - Defines how apps use the network
  - Lets apps talk to each other
  - Hides the other layers of the network
Project 1

► Simple Client
  ► Send requests to attu server
  ► Wait for a reply
  ► Extract the information from the reply
  ► Continue...

► Simple Server
  ► Server handles the Client requests
  ► Multi-threaded
Project 1

- This is the basis for many apps!
  - File transfer: send name, get file (§6.1.4)
  - Web browsing: send URL, get page
  - Echo: send message, get it back

- Let’s see how to write this app …
Socket API (Generalized)

- Simple application-layer abstractions (APIs) to use the network
  - The network service API used to write all Internet applications
  - Part of all major OSes and languages; originally Berkeley (Unix) ~1983

- Two kinds of sockets
  - Streams (TCP): reliably send a stream of bytes
  - Datagrams (UDP): unreliably send separate messages
Socket API (2)

- **Sockets** let apps attach to the local network at different ports
- **Ports** are used by OS to distinguish services/apps using internet
## Socket API (3)

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOCKET</td>
<td>Create a new communication endpoint</td>
</tr>
<tr>
<td>BIND</td>
<td>Associate a local address (port) with a socket</td>
</tr>
<tr>
<td>LISTEN</td>
<td>Announce willingness to accept connections; (give queue size)</td>
</tr>
<tr>
<td>ACCEPT</td>
<td>Passively establish an incoming connection</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Actively attempt to establish a connection</td>
</tr>
<tr>
<td>SEND</td>
<td>Send some data over the connection</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>Receive some data from the connection</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Release the connection</td>
</tr>
</tbody>
</table>

[https://docs.oracle.com/javase/8/docs/api/java/net/Socket.html](https://docs.oracle.com/javase/8/docs/api/java/net/Socket.html)
[https://docs.oracle.com/javase/8/docs/api/java/net/ServerSocket.html](https://docs.oracle.com/javase/8/docs/api/java/net/ServerSocket.html)
Using Sockets

**Client**
- SOCKET
- CONNECT*
- SEND
- RECV*
- CLOSE

**Server**
- SOCKET
- BIND
- LISTEN
- ACCEPT*
- RECV*
- SEND
- CLOSE

* Denotes a blocking call

Use threads to avoid blocking
Client Program (Outline)

socket(); // create socket
getaddrinfo(); // server and port name
    // www.example.com:80
connect(); // connect to a server [blocking]
...
send(); // send data
recv(); // await reply [blocking]
... // process reply
close() // done, disconnect
Server Program (Outline)

```
socket(); // create socket
getaddrinfo(); // get info for port on this host
bind(); // associate port with socket
listen(); // start accepting connections
while (true) {
    accept(); // wait for a connection [blocking]
    { // returns a new socket
        recv(); // wait for request [blocking]
        ... // process reply
        send(); // send reply
        close(); // close connection with client
    }
}
close(); // close the server socket
```
Java Tips

- `ServerSocket` for TCP server socket
- `Socket` for TCP client socket
- `DatagramSocket` for UDP server/client socket

Some other useful utils:

- `ByteBuffer` to manipulate bytes
Python Tips

- `socket.socket(socket.AF_INET, socket.SOCK_DGRAM)` for UDP
- `socket.socket(socket.AF_INET, socket.SOCK_STREAM)` for TCP

Might be useful:

- `socketserver`

- `struct.pack()` and `struct.unpack()` to manipulate bytes
Some guidelines

- Make sure your code runs on **attu**.
  - Python users can only use packages that are available on **attu**
    (no **pip** unfortunately)

- Small portions of the grade will be awarded to robustness of your server
  - Your server should accept clients outside localhost
  - Close connection when client sends faulty packets or timeout.
    - Padding and payload length; Number of packets; Correct content; etc.
  - Multithreaded?
Traceroute

- Apps talk to other apps with no real idea of what is inside the network
  - This is good! But you may be curious ...

- Peeking inside the Network with Traceroute
Traceroute

- Widely used command-line tool to let hosts peek inside the network
- On all OSes (tracert on Windows)
- Developed by Van Jacobson ~1987
- Uses a network-network interface (IP) in ways we will explain later

Van Jacobson

Credit: Wikipedia (public domain)
Traceroute

- Probes successive hops to find network path
- Takes advantage of ICMP error messages
- Packets keep track of a Time To Live (TTL)
  - Reduced by 1 at every hop
  - Packet is not forwarded if this value reaches 0; returns ICMP error message
  - Determines the number of hops a packet can make
  - Prevents circular routing
# Using Traceroute

A traceroute is a network administration tool that is used toPing a host on the internet. It sends a series of packets to the host with increasing time-to-live (TTL) values. Each packet is sent from the source host to the destination host, and the destination host responds with an error message indicating the IP address of the router that dropped the packet due to the expired TTL.

## Using Traceroute

```
C:\Users\djw\tracert www.uw.edu
Tracing route to www.uw.edu [128.95.155.134] over a maximum of 30 hops:

1  1 ms  C1  ms  2 ms  192.168.1.1
2  8 ms  8 ms  2 ms  198.88.58-62.staticip.rima-tde.net [80.58.62.98]
3  16 ms  5 ms  41 ms  169.26-88.58-76.staticip.rima-tde.net [80.58.76.169]
4  12 ms  12 ms  13 ms  217.92-88.87-87.staticip.rima-tde.net [80.58.87.217]
5  5 ms  11 ms  6 ms  et-1-0-0-1-101-GHIDORH1.red.telefonica-wholesale.net [94.142.103.20]
11  40 ms  38 ms  38 ms  176.52.205.226
10  108 ms  106 ms  136 ms  xs-6-0-2-gateway1.red.telefonica-wholesale.net [213.140.43.93]
11  180 ms  179 ms  182 ms  xs-9-2-0-gateway2.red.telefonica-wholesale.net [94.142.110.178]
9  179 ms  175 ms  176 ms  tc-4-2.carl.SanJose2.Level3.net [4.59.8.225]
10  190 ms  186 ms  187 ms  vla00003.cw3.SanJose1.Level3.net [4.59.152.190]
11  185 ms  185 ms  187 ms  az-922-0-0-962.SanJose1.Level3.net [4.59.152.251]
12  268 ms  295 ms  289 ms  az-7-7.chi.Seattle1.Level3.net [4.59.132.501]
13  334 ms  282 ms  195 ms  az-12-51.car2.Seattle1.Level3.net [4.59.147.132]
14  195 ms  196 ms  195 ms  PACIFIC-NOR.car2.Seattle1.Level3.net [4.53.146.122]
15  197 ms  195 ms  196 ms  a88-4000.icer-stillwater-01.infra.pmn-pagpup.net [209.124.188.132]
16  195 ms  196 ms  195 ms  v148000.wbaa-ade-01.infra.washington.edu [209.124.188.133]
17  201 ms  194 ms  196 ms  Request timed out.
18  201 ms  194 ms  196 ms  a4-5301.wuar-ade-1.infra.washington.edu [128.95.155.131]
19  197 ms  196 ms  195 ms  www.cac.washington.edu [128.95.155.134]
```

<table>
<thead>
<tr>
<th>Hop</th>
<th>RTT 1</th>
<th>RTT 2</th>
<th>RTT 3</th>
<th>Reverse DNS [IP]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 ms</td>
<td>C1 ms</td>
<td>2 ms</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>2</td>
<td>8 ms</td>
<td>8 ms</td>
<td>2 ms</td>
<td>198.88.58-62.staticip.rima-tde.net [80.58.62.98]</td>
</tr>
<tr>
<td>3</td>
<td>16 ms</td>
<td>5 ms</td>
<td>41 ms</td>
<td>169.26-88.58-76.staticip.rima-tde.net [80.58.76.169]</td>
</tr>
<tr>
<td>4</td>
<td>12 ms</td>
<td>12 ms</td>
<td>13 ms</td>
<td>217.92-88.87-87.staticip.rima-tde.net [80.58.87.217]</td>
</tr>
<tr>
<td>5</td>
<td>5 ms</td>
<td>11 ms</td>
<td>6 ms</td>
<td>et-1-0-0-1-101-GHIDORH1.red.telefonica-wholesale.net [94.142.103.20]</td>
</tr>
<tr>
<td>11</td>
<td>40 ms</td>
<td>38 ms</td>
<td>38 ms</td>
<td>176.52.205.226</td>
</tr>
<tr>
<td>10</td>
<td>108 ms</td>
<td>106 ms</td>
<td>136 ms</td>
<td>xs-6-0-2-gateway1.red.telefonica-wholesale.net [213.140.43.93]</td>
</tr>
<tr>
<td>11</td>
<td>180 ms</td>
<td>179 ms</td>
<td>182 ms</td>
<td>xs-9-2-0-gateway2.red.telefonica-wholesale.net [94.142.110.178]</td>
</tr>
<tr>
<td>9</td>
<td>179 ms</td>
<td>175 ms</td>
<td>176 ms</td>
<td>tc-4-2.carl.SanJose2.Level3.net [4.59.8.225]</td>
</tr>
<tr>
<td>10</td>
<td>190 ms</td>
<td>186 ms</td>
<td>187 ms</td>
<td>vla00003.cw3.SanJose1.Level3.net [4.59.152.190]</td>
</tr>
<tr>
<td>11</td>
<td>185 ms</td>
<td>185 ms</td>
<td>187 ms</td>
<td>az-922-0-0-962.SanJose1.Level3.net [4.59.152.251]</td>
</tr>
<tr>
<td>12</td>
<td>268 ms</td>
<td>295 ms</td>
<td>289 ms</td>
<td>az-7-7.chi.Seattle1.Level3.net [4.59.132.501]</td>
</tr>
<tr>
<td>13</td>
<td>334 ms</td>
<td>282 ms</td>
<td>195 ms</td>
<td>az-12-51.car2.Seattle1.Level3.net [4.59.147.132]</td>
</tr>
<tr>
<td>14</td>
<td>195 ms</td>
<td>196 ms</td>
<td>195 ms</td>
<td>PACIFIC-NOR.car2.Seattle1.Level3.net [4.53.146.122]</td>
</tr>
<tr>
<td>15</td>
<td>197 ms</td>
<td>195 ms</td>
<td>196 ms</td>
<td>a88-4000.icer-stillwater-01.infra.pmn-pagpup.net [209.124.188.132]</td>
</tr>
<tr>
<td>16</td>
<td>195 ms</td>
<td>196 ms</td>
<td>195 ms</td>
<td>v148000.wbaa-ade-01.infra.washington.edu [209.124.188.133]</td>
</tr>
<tr>
<td>17</td>
<td>201 ms</td>
<td>194 ms</td>
<td>196 ms</td>
<td>Request timed out.</td>
</tr>
<tr>
<td>18</td>
<td>201 ms</td>
<td>194 ms</td>
<td>196 ms</td>
<td>a4-5301.wuar-ade-1.infra.washington.edu [128.95.155.131]</td>
</tr>
<tr>
<td>19</td>
<td>197 ms</td>
<td>196 ms</td>
<td>195 ms</td>
<td><a href="http://www.cac.washington.edu">www.cac.washington.edu</a> [128.95.155.134]</td>
</tr>
</tbody>
</table>

* = No Response within a certain timeout (Not all routers/servers respond to ICMP traffic)
Using Traceroute (2)

- ISP names and places are educated guesses

![Diagram showing network topology with hop counts and delays]

- Home: 1 hop
- tde: 3 hops
- Telefonica: 4 hops
- Level3: 6 hops
- pnw-gigapop: 1 hop
- UW: 3 hops

 delays:
- My computer to NYC: 100 ms
- NYC to San Jose: 180 ms
- San Jose to Seattle: >200 ms

Destinations:
- www.uw.edu
- www1.cac.washington.edu