Client-Server Interaction

Network Applications Section
Chapter 28

Network Application Section
- Chapters 28 through 41 focus on high-level services available on an internet and the application software that provides such services.
- These chapters also explain how the applications use the network to communicate.
- Applications define symbolic names to identify physical and abstract resources.
- Communication over an internet requires a pair of application programs to cooperate.

Making Contact
- There is no way for protocol software to initiate the process to inform an application that communication has arrived.
- An application that expects communication must interact with the protocol BEFORE an external source attempts to communicate.
- When another application on a different host sends the message that is expected, communication starts.
- An application that passively waits for communication is called a server.

Client Software Characteristics
- Arbitrary application that becomes a client temporarily and performs calculations locally.
- Invoked directly by a user.
- Runs locally on user’s personal computer.
- Actively initiates contact with a server.
- Can access multiple servers as needed, but contacts one at a time.
- Does not require special hardware of sophisticated operating system.

Server Software Characteristics
- Is a special-purpose privileged program dedicated to providing one service but can handle multiple remote clients at the same time.
- Is invoked automatically on boot.
- Runs on a shared (not personal) computer.
- Waits passively for contact from remote clients.
- Accepts contact from arbitrary clients but offers a single service to all.
- Usually requires powerful hardware and a sophisticated operating system.

Client and Server Using TCP/IP
- Client / Server software interacts directly with the Transport layer to establish communication.
Multiple Services on One Computer

- Why would providing multiple services on one (or a few) machines be a good idea?

Identifying a Particular Service

- Each service is assigned a unique port number.

Example Ports and Services

<table>
<thead>
<tr>
<th>Application</th>
<th>Port</th>
<th>Transport</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>daytime</td>
<td>13</td>
<td>tcp/udp</td>
<td>Server time in ASCII</td>
</tr>
<tr>
<td>ftp</td>
<td>20</td>
<td>tcp</td>
<td>File transfer</td>
</tr>
<tr>
<td>ftp_data</td>
<td>21</td>
<td>tcp</td>
<td>Send ftp commands</td>
</tr>
<tr>
<td>telnet</td>
<td>23</td>
<td>tcp</td>
<td>Remote login</td>
</tr>
<tr>
<td>smtp</td>
<td>25</td>
<td>tcp</td>
<td>Send email</td>
</tr>
<tr>
<td>time</td>
<td>37</td>
<td>tcp/udp</td>
<td>Server time in seconds</td>
</tr>
<tr>
<td>http</td>
<td>80</td>
<td>tcp</td>
<td>Web browsing</td>
</tr>
<tr>
<td>pop3</td>
<td>110</td>
<td>tcp</td>
<td>Post office protocol</td>
</tr>
</tbody>
</table>

Concurrency and Threads

- Main Thread waits for a client to connect.
- When a client connects, the main thread spins off a handler thread which then handles the remainder of the interaction between the server and that particular client.
- The main thread continues waiting for more clients.
- When a client disconnects the handler thread is closed.

Multiple Clients for the same Service

- A Computer may run Multiple Servers at the same time.
  - These are identified by port number
- A Single Server may handle many Client connections at once.
  - Each client is identified by a Socket.
  - There is a binding between the Socket and the Client (Port Number and IP Address).