Socket Data structures and How the TCP protocol works

TA: Awad A Younis
Class: CS457
Fall 2014
Outline

- Data structures for TCP socket
- Connection Establishment (Client-side)
- Socket setup (Server-side)
- Closing TCP Connection
Socket structure (TCP socket)

- Data structure associated with each socket.
- Programs refer to data structures using descriptor returned by socket:
  ```c
  servSock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)
  ```
- More than one descriptor can refer to the same socket structure.
What happens when you create and use a socket?

1. Connection Establishment (Client-side)

   - When the client creates a new socket (**socket()**), it is in the closed state:

   ![Socket Structure Diagram]

   But we did not assign any Local port or Local IP

   - When the client calls **connect()** with port number (Q) and IP address (W.X.Y.Z), the system fills in the four fields in the socket structure.
What happen when you create and use a socket?

1. **Connection Establishment (Client-side)**
   
   • TCP opens three-way handshake
     
     1. Connection request from the client to the server
     2. Acknowledgment from server to client
     3. Another Acknowledgment from client to server
     
     • The client considers the connection as established when it received AK from the server.
What happen when you create and use a socket?

- **The whole process** (client-side)

Note:
- If there is no acknowledgement received from the server, client times out and gives up.
- This may take order of minutes
What happen when you create and use a socket?

2. **Socket setup (Server-side)**

   • This step is similar as in the client when the `socket()` function is called

   ![Diagram of socket setup]

   • Using the `bind()` function the server needs to bind to port number and IP address known to the client.

   ![Diagram of socket setup with bind function]
What happen when you create and use a socket?

2. **Socket setup (Server-side)**

- When the server calls `listen()` function, the state of the socket is changed to listening (ready to accept new connection).

- Any client connection request comes to server before the call to `listen()` will be rejected.
What happen when you create and use a socket?

- The whole process (server-side): **bind()** and **listen()**
What happen when you create and use a socket?

2. **Socket setup** *(Server-side): accept()*

- When the client connection request arrives, a new socket structure is created for the connection.

  ![Diagram showing socket setup process]

  - The new socket state is set to **connecting** and it is added to not-quite-connected sockets.

  **Note:** new socket port number and IP address is the same as the listening socket *(Wait a minute).*
What happen when you create and use a socket?

2. **Socket setup** *(Server-side)*: accept()

- How an incoming packet can be matched to sockets in the same host which have the same local port number and IP Address?

```
Packet
<table>
<thead>
<tr>
<th>Source IP Address</th>
<th>172.16.1.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port</td>
<td>56789</td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>10.1.2.3</td>
</tr>
<tr>
<td>Destination port</td>
<td>99</td>
</tr>
</tbody>
</table>
```
What happen when you create and use a socket?

2. **Socket setup (Server-side): accept()**

- When the **third message** (of the three handshakes messages) comes from the client, the new socket’s state is set to Established.

- The original server socket does not change state.
What happen when you create and use a socket?

- **The whole process (server-side):** `accept()`
What happen when you create and use a socket?

3. Closing TCP Connection

- When one application calls `close()` before the other end closes (what?):
  1. TCP implementation transmits any data remaining in the SendQ
  2. Handshake message is sent
  3. Descriptor is deallocated
  4. The state is set to closing

- When the acknowledgment for the close handshake is received, the state changes to **Half-Closed** *(Remains until the other end’s HSM is received)*.
What happen when you create and use a socket?

3. Closing TCP Connection

- When the other end’s close handshake message is arrived, an acknowledgment is sent and state changes to **Time-Wait**.

- Why **Time-Wait** state?
  - The possibility of a message being delayed in a network
  - Twice the time a packet can remain in a network.
  - It stays for anywhere from 30 second to 2 minutes.
What happen when you create and use a socket?

3. Closing TCP Connection

• What will happen to the end point that does not close first?

• When the closing handshake message arrives:
  1. An acknowledgement is sent immediately
  2. The connection state becomes Close-Wait
  3. When the application calls `close()`:
     • Descriptor is deallocated
     • HSM initiated
     • Deallocate socket structure
What happen when you create and use a socket?

3. Closing TCP Connection: the whole process
Thank You

Reference

• Pocket Guide to TCP/IP Socket, by Michael J. Donahoo and Kenneth L. Calvert

• Beej’s Guide to Network Programming Using Internet Sockets, by Brian "Beej" Hall. (http://www.cs.columbia.edu/~danr/courses/6761/Fall00/hw/pa1/6761-sockhelp.pdf)