UDP Reference Slides

UDP Overview

- Best effort service, segments may be:
  - Lost
  - Delivered out of order
- Connectionless:
  - No handshaking between UDP sender and receiver
  - Each UDP segment handled independently of others

UDP Socket

UDP socket indentified by the tuple:

```
<dest IP address, dest port number>
```

When a host receives a UDP segment, it:

- Checks the destination port number
- Directs the UDP segment to the socket with that port number
    - What does this imply?

UDP Socket

- Sockets created using SOCK_DGRAM instead of SOCK_STREAM
- No need for connection establishment and termination
- No `connect-bind, listen, accept` handshaking,
- Server must still always call `bind()`
- Client doesn't need to call `connect()` though it may use `connect()` to tell kernel to "remember" the server's address and port number.

Data Transmission

No “connection” between sender and receiver

- Sender explicitly attaches IP address and port number of destination to each packet using `sendto()`
  - `send()` can still be used if address and port number have been “registered” using `connect()`
- If receiver uses `recvfrom()` it can extract IP address and port number of sender of the received packet
  - If these are not needed, `recv()` can be used instead

Transmitted data may be delivered out of order or not delivered at all

Data Transmission

UDP packets have boundaries (not a byte-stream), so `recv()` retrieves one message at a time

- No need to call `recv()` in a loop
- Call to `recv()` returns the whole packet
- Same for `recvfrom()` and `recvmsg()`