UDP Sockets

UDP Sockets

- In TCP, a connection must be established first, before data communications can occur. Because of this, when data arrives at the socket, the identity of the sender has already been established. Similarly, when data is sent over the socket, the destination has already been identified.
- In UDP this is not the case since there is no such thing as a connection. Every message is sent as a single UDP data packet. For this reason, the socket interface must identify the sender (IP address and port) whenever a new UDP packet arrives. Similarly, the destination (IP address and port) must be identified whenever a UDP packet is sent.

Python UDP Sockets

- **Create Ipv4/UDP socket:**

```
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- Receive data from a UDP socket of at most buffersize bytes. This function returns both the received data and the remote address/port, e.g.,

```
output = s.recvfrom(buffersize)
msg_bytes, address_port = output
address, port = address_port
```

- If the specified buffersize is less than the arrived UDP packet payload length, the remaining bytes are discarded!
- Send data over socket to the address specified (UDP): The bytes to be sent and the addres/port are passed to the function, e.g.,

```
address_port = ("192.168.1.10", 20000)
s.sendto(msg_bytes, address_port)
```

see EchoClientServer_UDP.py