

UDP server

UDP client

```
s = socket()
```

```
s.bind(port)
```

```
s.recvfrom()
```

...

```
s = socket()
```

```
s.sendto()
```

...

user datagram

```
graph LR; subgraph UDP_server [UDP server]; S1[s = socket()] --> B1[s.bind(port)]; B1 --> R1[s.recvfrom()]; R1 --> D1[...]; end; subgraph UDP_client [UDP client]; S2[s = socket()] --> T2[s.sendto()]; T2 --> D2[...]; end; D1 -.->|user datagram| D2;
```

The diagram illustrates the flow of a user datagram between a UDP server and a UDP client. On the server side, the process starts with creating a socket (`s = socket()`), binding it to a port (`s.bind(port)`), and then receiving data (`s.recvfrom()`). On the client side, the process starts with creating a socket (`s = socket()`) and then sending data (`s.sendto()`). A dashed arrow labeled "user datagram" connects the `s.recvfrom()` box on the server to the `s.sendto()` box on the client, indicating the direction of data transmission. Both sides have an ellipsis (...) below their respective `s.recvfrom()` or `s.sendto()` boxes, suggesting a loop or continuation of the process.