

# NETWORK PROGRAMMING

University of Tripoli  
Faculty of Information Technology  
Department of Networking

Assistant Lecturer / Marwa Ebrahim Elwakedy



# UDP: Datagrams

- Data sent in discrete packets (Datagrams)
- No concept of a "connection"
- No reliability, no ordering of data
- Datagrams may be lost, arrive in any order
- Higher performance (used in games, etc.)

# UDP: Datagrams

- Data sent in discrete packets (Datagrams)
- No concept of a "connection"
- No reliability, no ordering of data
- Datagrams may be lost, arrive in any order
- Higher performance (used in games, etc.)



# Common UDP-based Applications

- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)
- Trivial File Transfer Protocol (TFTP)
- Network File Systems (NFS)
- Network Time Protocol (NTP)
- Real Audio
- Echo
- Talk

# UDP Server

## ■ A simple datagram server

```
from socket import *
```

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

```
s.bind(("",10000))
```

```
while True:
```

```
    data, addr = s.recvfrom(maxsize)
```

```
    resp = "Get off my lawn!"
```

```
    s.sendto(resp, addr)
```

## ■ No "connection" is established

## ■ It just sends and receives packets



# UDP Client

## ■ Sending a datagram to a server

```
from socket import *
```

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

```
msg = "Hello World"
```

```
s.sendto(msg,("server.com",10000))
```

```
data, addr = s.recvfrom(maxsize)
```

## ■ Key concept: No "connection"

## ■ You just send a data packet

# UDPEchoServer

```
from socket import *  
  
import sys # Create a UDP socket  
  
s = socket(AF_INET, SOCK_DGRAM) # Bind the socket to the port  
  
server_addr = ('localhost', 10000)  
  
print('starting up on ', server_addr)  
  
s.bind(server_addr)
```

## UDPEchoServer (2)

```
while True:  
    print("Waiting to receive message")  
    data, addr = s.recvfrom(4096)  
    if data:  
        sent = s.sendto(data, addr)
```



# UDPEchoClient

```
from socket import *
```

```
import sys
```

```
# Create a UDP socket
```

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

```
server_addr = ('localhost', 10000)
```

```
message = 'This is the message.'
```

## UDPEchoClient (2)

```
try:
    # Send data
    print('sending "%s"' % message)
    sent = s.sendto(message, server_addr)

    # Receive response
    print('waiting to receive ...')
    data, server = s.recvfrom(4096)
    print('received "%s"' % data)

finally:
    print('closing socket')
    s.close()
```