

Outline

- Quick primer to sockets and network programming



Address, Ports

- Each network adapter is identifiable by an unique address – Internet Protocol Addr (IP Address)
 - E.g. gemini.cs.uga.edu = 128.192.251.5
- In a given machine, ports are used to identify data for a specific “process”
 - Well known ports: 80 for HTTP, 25 for SMTP ...
- telnet www.yahoo.com 80<cr>
GET / HTTP/1.0<cr><cr>
 - HINT: Look in /etc/services for well known services



Streaming protocol

- Reliability – against packet dropping, reordering, duplication
- In order delivery
- TCP guarantees reliable, in order delivery
 - Lower throughput
 - HTTP is TCP
- UDP offers no guarantees
 - Higher throughput
 - Streaming audio/video is UDP



Recipient group

- Unicast – only one recipient
- Broadcast – everyone on the same (subnet)
 - Bridges for not forward broadcast packets across subnets
- Multicast – only interested clients. Clients actively join multicast groups. All clients listening on a group receive messages
- Anycast – One of many clients will receive the response



Sockets

- Communications mechanism
- Behaves like a pipe – data sent on one end is received on the other end

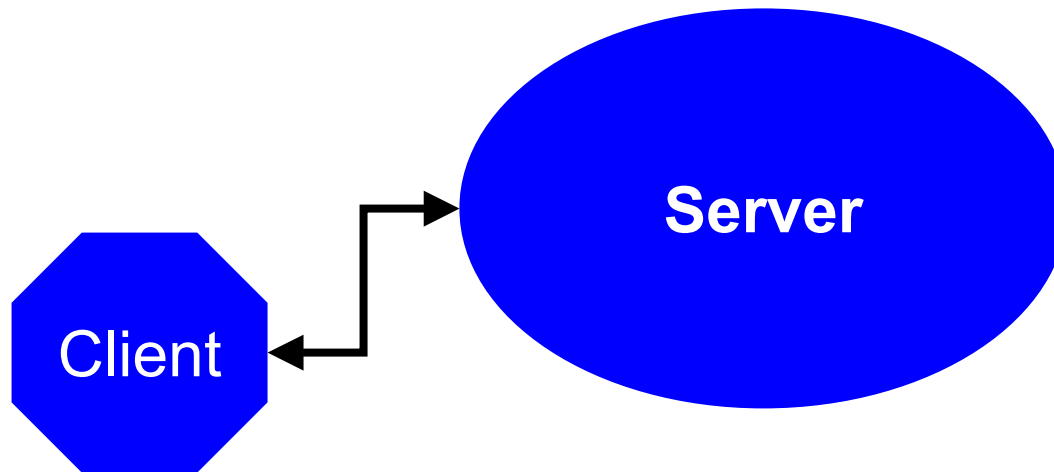
Sender

Receiver



- On a server, you can bind the socket to a port so that it listens for connection requests on that port
- On the client, you can connect to a server socket

Central server based location management



- Simple central server based approach
 - Server bind and waits on a well known port for requests
 - Clients connect to server using well known port

Connectionless

```
soc = socket(AF_INET, SOCK_STREAM, IP)
```

```
sendto(soc, messageBuffer, messageLen, flags,  
destinationSockaddr, len)
```

```
recvfrom (soc, messageBuffer, messageLen, flags,  
sourceAddr, len)
```

Client

```
soc = socket(AF_INET, SOCK_STREAM, IP);  
bzero((void *) &sAddr, sizeof(sAddr));  
sAddr.sin_family = AF_INET;  
sAddr.sin_addr = SERVER_ADDRESS;  
sAddr.sin_port = SERVER_PORT;  
connect(soc, &sAddr, sizeof(sAddr))  
←-----END-----→  
write(soc, .. , .. )  
read(soc, .. , .. )  
close(soc)
```



Server

```
soc = socket(AF_INET, SOCK_STREAM, IP);  
bzero((void *) &sAddr, sizeof(sAddr));  
sAddr.sin_family = AF_INET;  
sAddr.sin_addr = INADDR_ANY;  
sAddr.sin_port = SERVER_PORT;  
bind(soc, &sAddr, sizeof(sAddr))
```

```
socNew = accept(soc, .. , ..)
```

←-----END-----→

```
write(socNew, .. , .. )
```

```
read(socNew, .. , .. )
```

```
close(socNew)
```



Useful tools

- Tcpdump
 - Dumps network packets
- Netstat
 - Shows active connections
- Ping and traceroute
 - Verifies that “packets” can get to a machine
- Host/dig/nslookup
 - Hostname->IP mapping

