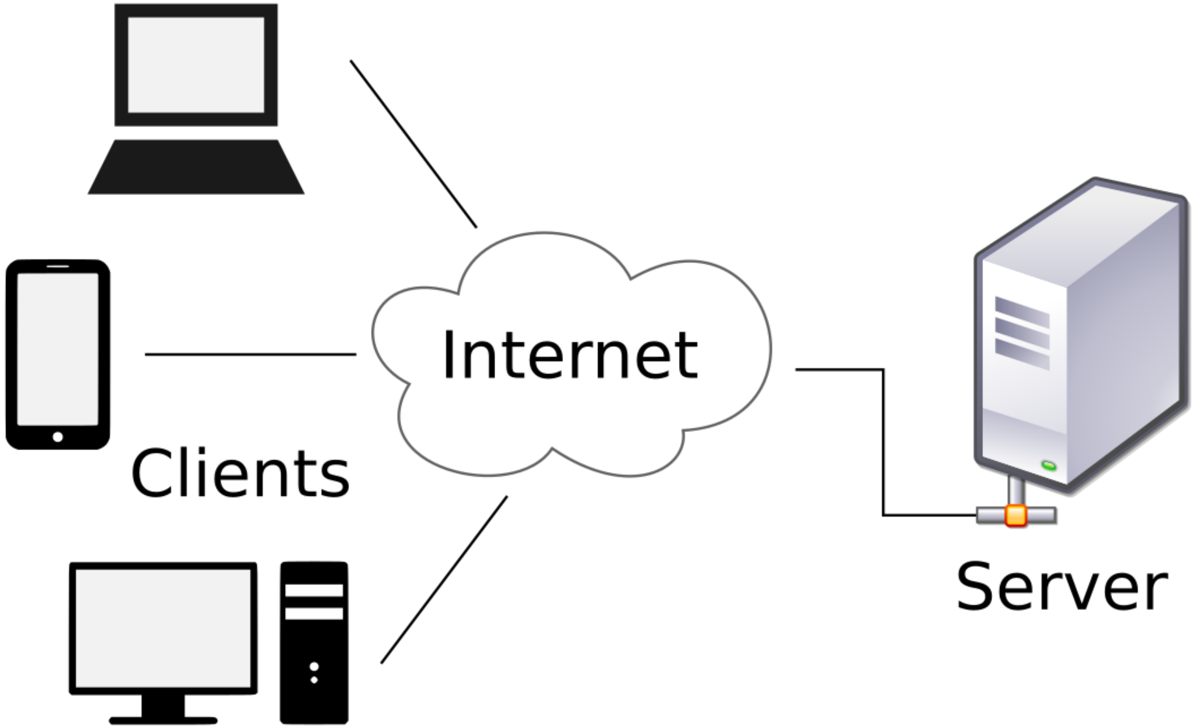
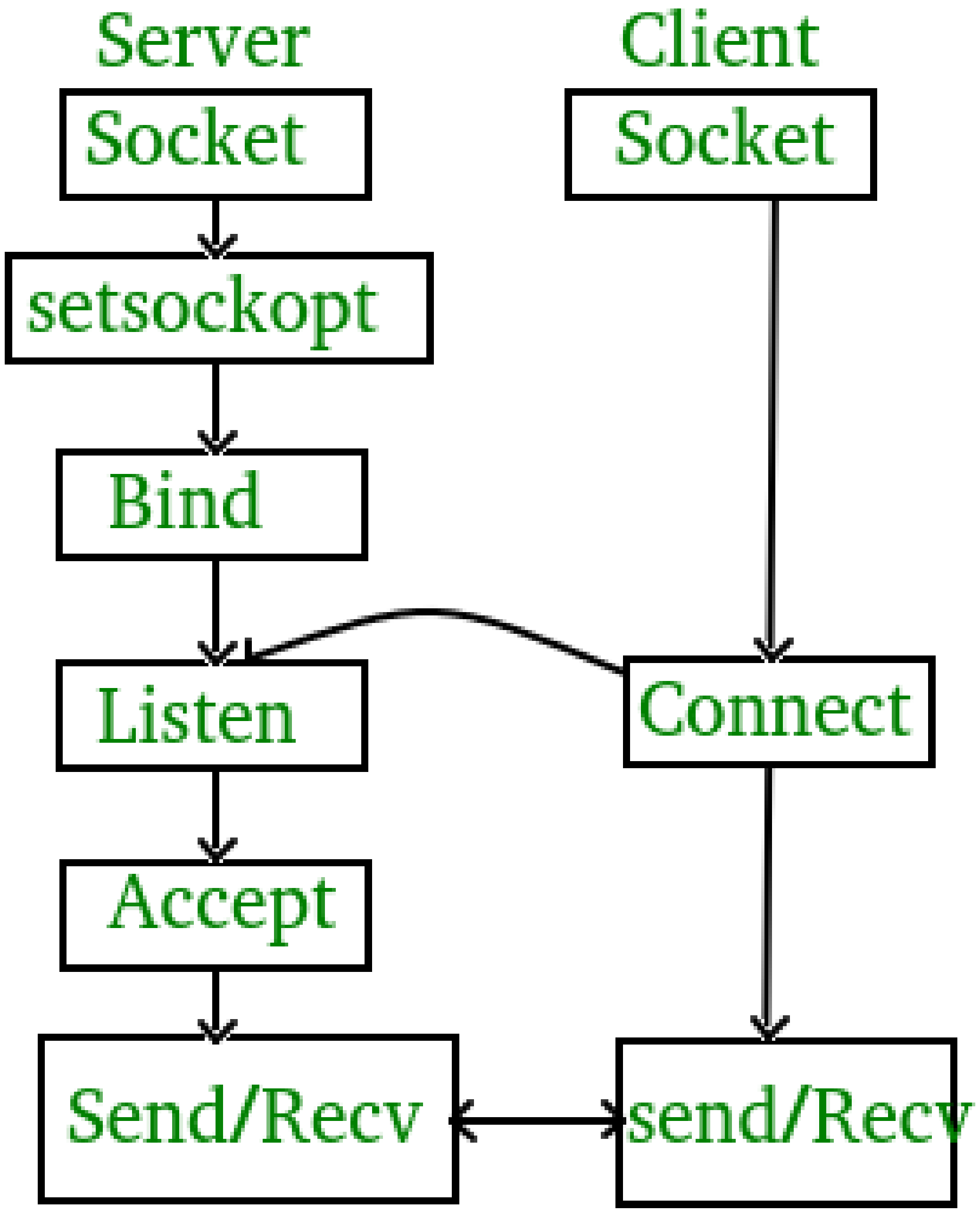
Lab1: Basic Socket Programming - Creating a Simple Client-Server Communication


Internet
Clients
Server



Server

Socket

setsockopt

Bind

Listen

Accept

Send/Recv

Client

Socket

Connect

send/Recv

Creating the Server

A **server** listens on a specific port for incoming connections and can send or receive data to/from clients.

The basic steps involved in creating a server:

- Import the socket module.
- Create a socket object using socket.socket().
- Bind the socket to an IP address and a port using bind().
- Use listen() to wait for incoming connections.
- Accept a connection using accept().
- Receive and send data using recv() and send().

Server Program

```
import socket
# Create a socket object
server_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
# Bind to a specific IP and port
server_socket.bind(('localhost', 8080))
# Start listening for connections
server_socket.listen(1)
print("Server is listening on port 8080...")
# Accept a client connection
client_socket, addr = server_socket.accept()
print(f"Connection from {addr} has been established.")
# Receive data from the client
data = client_socket.recv(1024).decode('utf-8')
print(f"Received from client: {data}")
# Send a response back to the client
client_socket.send("Hello Client".encode('utf-8'))
# Close the socket
client_socket.close()
```

Creating the Client

A **client** connects to a server on a specific IP and port and sends data.

The basic steps involved in creating a client:

- Import the socket module.
- Create a socket object.
- Connect to the server using connect().
- Send data to the server using send().
- Receive data from the server using recv().

Client Program

```
import socket
# Create a socket object
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Connect to the server
client_socket.connect(('localhost', 8080))
# Send data to the server
client_socket.send("Hello Server".encode('utf-8'))
# Receive response from the server
data = client_socket.recv(1024).decode('utf-8')
print(f"Received from server: {data}")
# Close the socket
client_socket.close()
```