

DATA COMMUNICATION AND COMPUTER NETWORKS LAB

SEMESTER:6TH



LAB REPORT # 9

Submitted By: *Zainab Khalid*
Registration No: *19PWCSE1743*
Section: **A**
Submitted to: *Engr Faiz Ullah*

**DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING
UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR**

LAB # 9

Criteria	Excellent	Marks Obtained
1. Objectives of Lab	All objectives of lab are properly covered [Marks 0.5]	
2. Introduction to Python Programming	Brief introduction of Python Programming [Marks 2]	
3. Introduction to python socket library and its various functions	Brief introduction about Socket library and its various functions used in Lab [Marks 2]	
4. Client-Server Communication using socket library	Client-Server communication, Python code and output [Marks 3]	
5. Flowchart of client server communication using python socket library	Draw a flowchart of the sequence of socket API calls and data flow for TCP [Marks 2]	
6. Conclusion	Conclusion about RC-Circuit analysis [Marks 0.5]	

LAB # 9

TCP/IP IMPLEMENTATION USING PYTHON SOCKET PROGRAMMING

Objectives of Lab:

- Introduction to python
- To learn about socket programming
- To learn about the functions of the python library
- To learn TCP/IP implementation using python programming

Introduction to Python Programming:

Python is a widely-used general-purpose, high-level programming language. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Applications:

1. GUI-based desktop applications
2. Graphic design, image processing applications, Games, and Scientific/ computational Applications
3. Web frameworks and applications
4. Enterprise and Business applications
5. Operating Systems
6. Education
7. Database Access
8. Language Development
9. Prototyping
10. Software Development

Introduction to python socket library and its various functions

Sockets and the socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical, local network to the computer or one that's physically connected to an external network, with its own connections to other networks. The obvious example is the Internet, which you connect to via your ISP.

Define socket object:

Socket () creates an endpoint for communication and returns a descriptor.

int socket(int domain, int type, int protocol);

Bind with IP address (assigning port number):

bind() system call binds an address or name with any socket in the C language.

int bind(int sockfd, const struct sockaddr *my_addr, socklen_t addrlen);

Listen ():

listen () listen for connections on a socket

int listen(int sockfd, int backlog);

Accept ():

Accept() - accept a connection on a socket

*int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);*

send ():

send() are used to transmit a message to another socket.

*ssize_t send(int sockfd, const void *buf, size_t len, int flags);*

receive ():

recv() receive a message from a socket

*ssize_t recv(int s, void *buf, size_t len, int flags);*

connect ():

Connect() - initiate a connection on a socket

*int connect(int sockfd, const struct sockaddr *serv_addr, socklen_t addrlen);*

close ():

Close() - close a socket descriptor

int close(int sockfd);

Client-Server Communication using socket library

Server file:

```
import socket

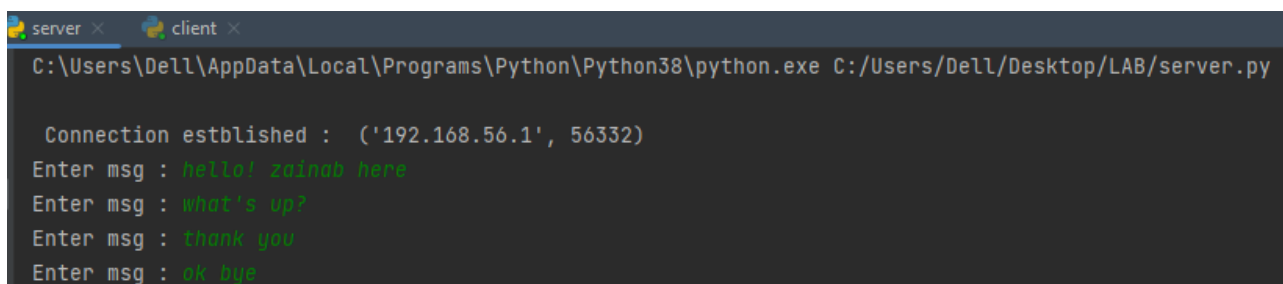
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((socket.gethostname(), 6061))
s.listen(5)

clientSoc, address = s.accept()

print("\n Connection established : ", address)
while True:
    msg = input("Enter msg : ")

    clientSoc.send(bytes(msg, 'utf-8'))
clientSoc.close()
```

Server-side output:



```
server x client x
C:\Users\Dell\AppData\Local\Programs\Python\Python38\python.exe C:/Users/Dell/Desktop/LAB/server.py

Connection established : ('192.168.56.1', 56332)
Enter msg : hello! zainab here
Enter msg : what's up?
Enter msg : thank you
Enter msg : ok bye
```

Client file:

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

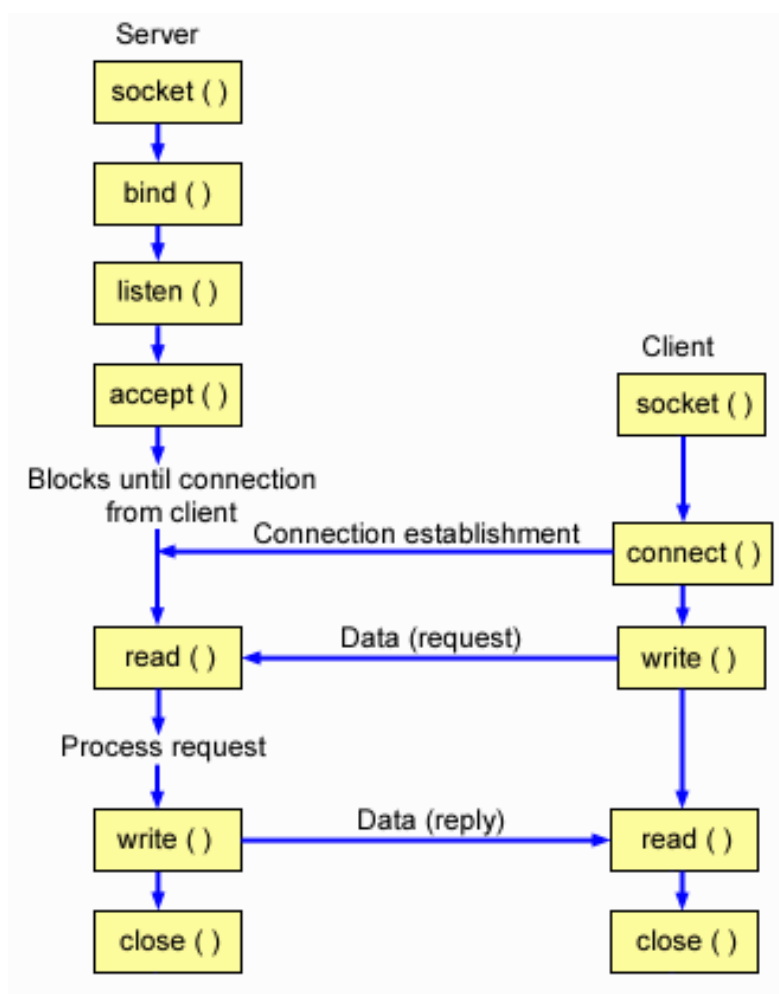
s.connect((socket.gethostname(), 6061))

while True:
    msg = s.recv(2024)
    print("Message is : ", msg.decode())
```

Client-side output:

```
server x client x
C:\Users\De11\AppData\Local\Programs\Python\Python38\python.exe C:/Users/De11/Desktop/LAB/client.py
Message is : hello! zainab here
Message is : what's up?
Message is : thank you
Message is : ok bye
|
```

Flowchart of client-server communication using python socket library



Conclusion

Client-server intercommunication is developed using socket programming. Both client and server can communicate with each other by TCP/IP protocol.