**SOFTWARE DESIGN DOCUMENT**

**FOR**

**ETHERNET BASED COMMUNICATION BETWEEN TWO COMPUTERS**

|  |  |  |
| --- | --- | --- |
| REFERENCE NO. | REV NO. | EFFECTIVE DATE |
| EBC/SDD/07/20 | 01 | 16/07/2020 |
| TITLE | **SOFTWARE DESIGN DOCUMENT**  **FOR**  **ETHERNET BASED COMMUNICATION BETWEEN TWO COMPUTERS** | |
| CUSTOMER |  | |
| PROJECT | ETHERNET BASED COMMUNICATION BETWEEN TWO COMPUTERS | |
| PREPARED BY | Niranjan Patil | |
| VERIFIED BY |  | |
| IV &V | Oak Systems Private Limited | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision#** | **Date** | **Details** | **Author initials** |
| 01 | 15/07/2020 | First draft | Niranjan Patil |
|  |  |  |  |

**TABLE OF CONTENTS**

1. Introduction…………………………………………………………..4
   1. Purpose………………………………………………………….4
   2. Scope……………………………………………………………4
   3. Abbreviations and Acronyms…………………………………...4
2. System Overview……………………………………………………..4
   1. System architecture……………………………………………...4
3. Software Overview…………………………………………………...5
   1. Software architecture……………………………………………5
   2. Stages of Server…………………………………………………6
   3. Stages of Client………………………………………………….6
4. **INTRODUCTION**
   1. **Purpose**

This is the standard “software design description document”. This document establishes the software design description for the “Ethernet Based Communication(EBC) Between Two Computers”.

Primary audiences of this document are software developers.

* 1. **Scope**

This document contains the complete description of the EBC between two computers.

**1.3 Abbreviations and Acronyms**

|  |  |
| --- | --- |
| EBC | Ethernet based communication |
| IV & V | Independent verification and validation |
|  |  |

1. **SYSTEM OVERVIEW**
   1. **System Architecture**

sockets

SERVER

CLIENT

IP address, port number

**ETHERNET**

Send/receive data

Figure: system architecture

The above figure shows the architecture of the EBC between two computers.

Basically it is an peer to peer network and it has one client and one server machine.

Both the Client and server has sockets, sockets are the internal endpoints for sending or receiving the data.

Initially the server is high and it is waiting for the client to connect, the server has an particular IP address and it is used by the client to get connect to the server.

Once the client has connected to the server the server notifies the client and accepts the connection and it also tracks the port number of the client.

Once the connection has been established, than client and server are ready to begin the communication between them.

By using the console window of client the user can enter the message and on the other side, machine which is acting as server will receive the message. User sitting on the side of server can also write an message and send it to client, now the client will receive the message.

1. **SOFTWARE OVERVIEW**

* 1. **Software Architecture**

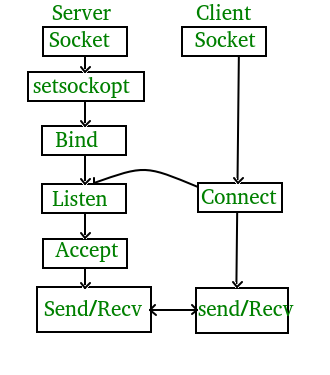


Figure: **State diagram for server and client model**

* 1. **Stages of Server**

**socket creation:**

Socket creation involves the communication domain and the type of communication.

Communication domain involve AF\_INET(IPv4 protocol)

Communication type is TCP (SOCK\_STREAM)

**Set socket:**

This helps in manipulating options for the socket referred by the file descriptor .This is completely optional, but it helps in reuse of address and port. Prevents error such as: “address already in use”.

**Bind:** After creation of the socket, we need bind the socket to the specified IP address and port number.

**Listen:**

It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The parameter of listen function, defines the maximum number to which the queue of pending connections for server may grow. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

**Accept and send/receive data:** It extracts the first connection request on the queue of pending connections for the listening socket, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer data.

* 1. **Stages of Client**

**Socket creation:**

Socket creation involves the communication domain and the type of communication.

Communication domain involve AF\_INET(IPv4 protocol)

Communication type is TCP (SOCK\_STREAM)

**Connect:**

The connect() system call connects the socket referred to by the file descriptor to the specified Server’s address and the port number is.

**Send/Receive data:**

Once the connection has been established both the server and client are ready to share the data.