IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

Review Paper on Socket Programming

Poorvika B M[1], Prajwal Gowda M M[2], Prasad R[3], Rahul R Poojary[4], Mr. Pradeep Nayak[5]
Assistant Professor, Department of Information Science and Engineering[5]
Students, Department of Information Science and Engineering[1,2,3,4]
Alva's Institute of Engineering and Technology, Mijar, Mangalore, Karnataka, India

Abstract: *The paper's objective is to introduce sockets and discuss how they are used in network programming. For client-server applications to function, sockets are essential. By sending to or reading from these sockets, the client and server can exchange data. They were created in Berkeley as a component of the BSD UNIX operating system and the Internet helped them spread like wildfire. In this paper, network programming fundamentals and socket-based network application development are covered. Because java has been used largely for establishing client-server interactions through sockets, performing the socket functions/methods is one of the most fundamental network programming tasks that a java programmer is likely to encounter. The paper's objective is to introduce sockets and discuss how they are used in network programming. For client-server applications to function, sockets are essential. These sockets allow the client and server to exchange data by writing to or reading from them. They were created in Berkeley as a component of the UNIX operating system known as BSD. Transfers that are synchronous and asynchronous. In today's globalised society, when data transmission is essential for communication within any company, it is crucial to use the network strategically to ensure efficient transmission with little traffic overhead. The two such strategies discussed in this study can be chosen to suit the requirements of every company. Socket programming and remote method invocation are the two methods.*

Keywords: 5G mobile communication network

I. INTRODUCTION

The University of California at Berkeley received funding from the Advanced Research Projects Agency (ARPA) of the US government in the 1980s to create TCP/IP protocols under the UNIX operating system. The socket interface was created by a team of Berkeley academics as part of this effort as an application programme interface (API) for TCP/IP network communications. These sockets are the TCP and UDP protocols' respective programming interfaces for stream and datagram communication of the transport layer, a component of the TCP/IP stack. The primary responsibility of a developer while designing a network application is to write the code for both the client and server programmes. A network path is a path connecting the communication of numerous entities. Data transmission services are offered on this line from the sender to the recipient. A software known as socket programming will process the network layer method that the data sent will go through. The socket interface defines several software functions or routines enabling the creation of applications for TCP/IP networks, making it an API for TCP/IP networks. The socket interface was initially incorporated into the UNIX operating system by its designers. Connecting various machines to one another for the purpose of sharing resources, performing tasks, or communicating is the essence of a network.

Two widespread network protocols are supported by the java.net package:

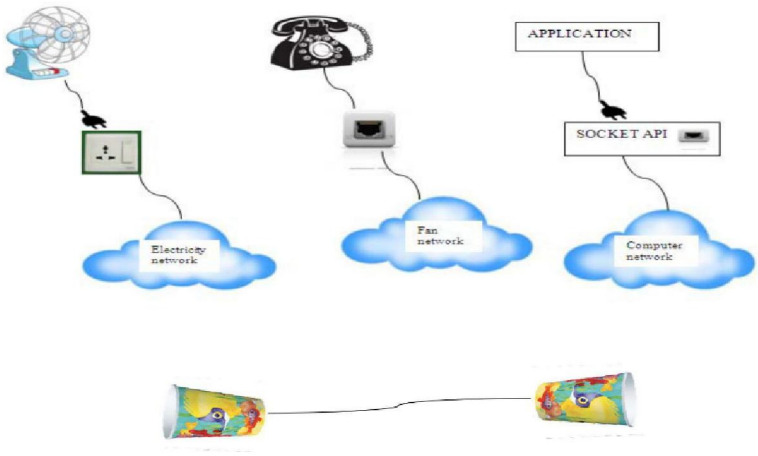
Transmission Control Protocol, or TCP for short, enables dependable communication between two programmes. TCP/IP refers to TCP when it is used with the Internet Protocol (IP).

User Datagram Protocol, also known as UDP, is a connectionless protocol that enables the transmission of data packets between applications.

II. SOCKETS

The word "socket" derives from a metaphor involving electrical and telephone connections, in which sockets serve as interfaces for connecting to one another through a network.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

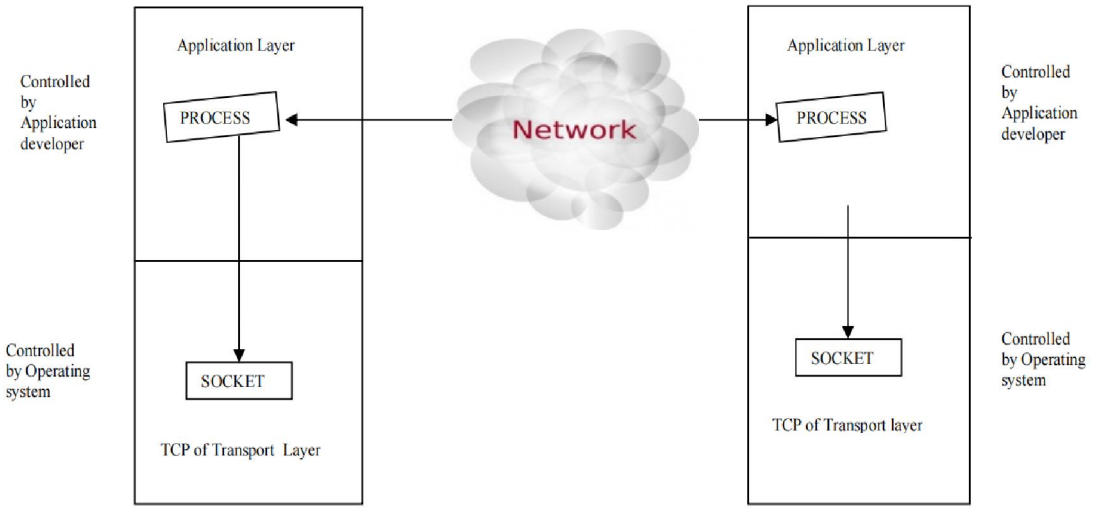


2.1. Network Programming Using Sockets

In recent years, the Internet has gained a lot of popularity. Internet network software is increasingly in demand as its popularity continues to rise. Java's powerful networking support, which is a key feature of the language, is one of the biggest benefits of using it to create Internet software. We can find classes that represent URLs, URL connections, and sockets in the java.net package. We may create complex platform-independent networking (Internet) programmes with the help of the java.io package. Sockets are used in network programming for interprocess communication. Socket programming is also known as network programming as a result. BSD type sockets are used in Java socket programming to interface with TCP/IP services. . Understanding how internet-based interprocess communication functions requires knowledge of socket programming, but not at the level of the programme itself, but rather at a higher level that is compiled into a set of socket programmes. Since Internet protocol is the foundation for computer communication in this context, sockets can also be referred to as network or Internet sockets.

2.2. Socket Programming With TCP

Since it relies on connections between clients and servers, TCP offers a connection-oriented service. Connection-oriented refers to the idea that a connection must be made before processes can communicate. The fact that a TCP client needs an acknowledgement from the server in order to communicate data to it makes the Transmission Control Protocol dependable. TCP automatically retransmits the data and waits for a longer amount of time if an acknowledgement is not received. Sending messages into sockets allows processes that are executing on different machines to communicate with one another. Every process can be compared to a home, and the process' socket to a door.



Process communicating through TCP sockets

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

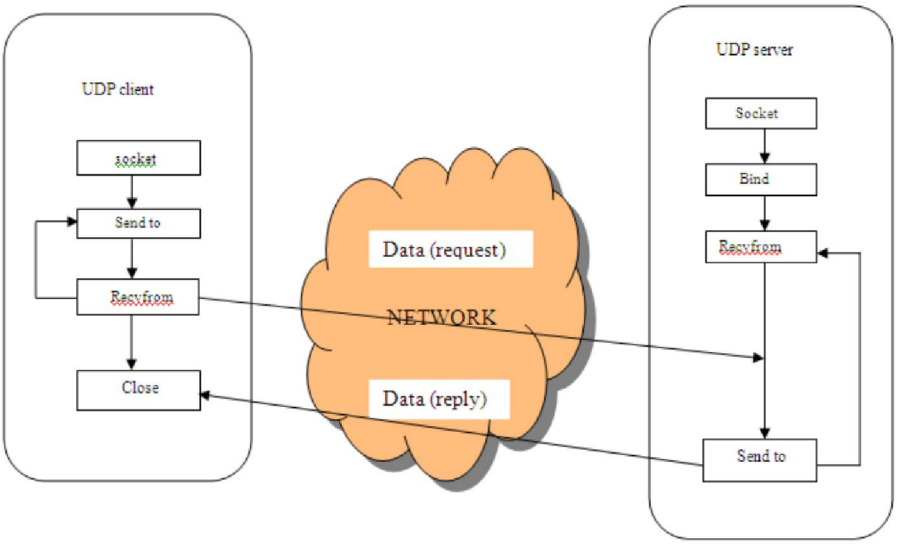
2.3. Socket Programming Over TCP

The ability to build sockets for interprocess communication has been made available by Java (IPC). As a result, when writing Java code for sockets, one must import the java.net package. The Java platform has a class called Socket that implements the client-side connection in the java.net package. The server-side connection is implemented by the class Server Socket. The Server

Socket on the server carries out the operations "bind," which involves fixing to a specific port number and IP address, "listen," which involves watching for incoming requests on the port, and "accept," which involves accepting a connection from the client.

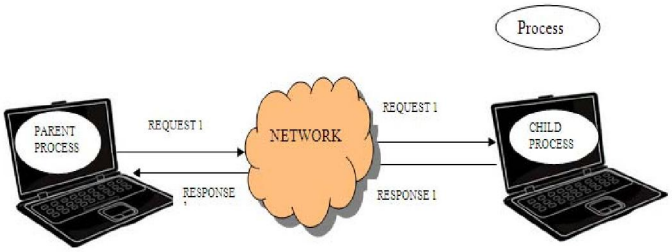
2.4. Socket Programming over UDP

A connectionless datagram protocol is UDP. Like with TCP, the client does not establish a connection with the server. Instead, the client only uses the send to function, which asks for the destination address as an argument, to send a datagram to the server.



III. CLIENT-SERVER COMMUNICATION

Computers that are either clients or servers make up a network. A client is a programme that requests a service, whereas a server is a programme that provides a service. Clients are PCs or workstations on which users run applications, whereas servers are potent machines or processes specialised to managing disc drives (file servers), printers (print servers), or network traffic (network services). Servers provide resources like files, hardware, and even computational power to clients. These sockets are the TCP and UDP protocols' respective programming interfaces for stream and datagram communication of the transport layer, a component of the TCP/IP stack. The primary responsibility of a developer while designing a network application is to write the code for both the client and server programmes. A proprietary client/server programme is what this article's client/server coverage is all about. Both the client and server applications are written by a single developer (or development team), who has complete control over the code. Other independent developers won't be able to create code that communicates with the programme, however, because the code does not implement a public-domain protocol.



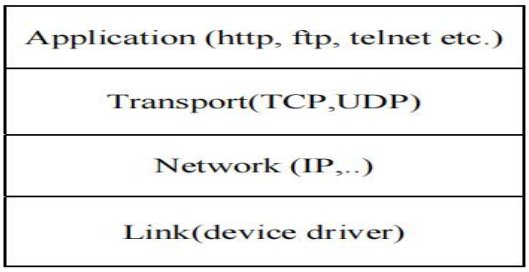
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

IV. OPERATIONS OF SOCKET

A socket can carry out the following four basic tasks:
To connect to a distant machine, follow these steps: 1. Send data; 2. Receive data; 3. Close the connection. More than one host cannot be connected to a socket at once. A socket, however, could send and receive data from the host to which it is connected. Java Network. The Java interface to a network socket is the socket class, which enables you to carry out the four basic socket actions.

V. PORT

A port is an application- or process-specific software construct acting as a communications endpoint in the host operating system of a machine in computer networking. An IP address of the host and the kind of communication protocol are both linked to a port. Ports are used to separately identify several programmes or processes that are executing on a single machine. The Transport Layer protocols, commonly referred to as TCP/IP and include the Internet software stack's Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), are the ones that use ports the most. We must select a port where both the client and server consent to meet while configuring a client or server. This port is a logical port that is identified by a 16-bit integer number rather than a physical port. The developer must take care not to utilise one of the well-known port numbers described in the RFCs while designing a proprietary client-server application as some port numbers from 0 to 1024 have been reserved to accommodate common/well-known services.



TCP/IP software stack

VI. WHAT EXACTLY IS A SOCKET?

A single computer cannot serve several clients or multiple types of information simultaneously due to the concept of a socket. This is overseen by a port is a numbered socket that is added to a specific machine. Until a client connects to it, a server process is said to "listen" to that port. Although each session is distinct, a server is permitted to accept numerous clients connecting to the same port number. [4] A server process must be multithreaded or have some method of multiplexing simultaneous I/O in order to manage many client connections.
TCP-based sockets offer the means of communication between two machines. On its end of the communication, a client application generates a connection and tries to connect that socket to a server. On its end of the conversation, the server produces a socket object when the connection is created. Now that the socket is open, the client and server can communicate by writing to and reading from it. A socket and the Java.net are represented by the socket class.The server application can establish connections with clients by using the server Socket class's technique for listening for them.

6.1 Steps for Establishing a TCP Connection Between Two Computers Using Sockets:

Step 1: The server creates a server Socket object in step one, specifying the port number on which communication is to take place.
Step 2: The accept () function of the server Socket class is called by the server. This procedure watches for a client connection to the server using the specified port.
Step 3: A client creates a Socket object, providing the server's name and port number to connect to, after the server has been waiting.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

Step 4: The Socket class function Object() { [native code] } tries to connect the client to the server and port that are supplied. The client now has a Socket object that can communicate with the server if communication has been established.
Step 5: The accept () method on the server side returns a reference to a fresh socket that is coupled with the client's socket.

6.2 Socket programming over UDP

Datagram protocol (DP) is a connectionless protocol. Like with TCP, the client does not establish a connection with the server. Rather, the client merely transmits a datagram to the server employing the sendto function, which asks for the destination's address as an input. Similar to clients, the server rejects connections from them. However, the server simply invokes the recvfrom function, which keeps waiting till some client sends data. In order for the server to respond to the client as demonstrated, recvfrom returns the IP address of the client along with the datagram. No first handshake exchange takes place. It is unreliable because you can't be sure that data or messages you transmit will arrive since they might get lost in transit. While a message is being transferred, there might be corruption.
This list is a summary of functions or methods provided by the UDP Socket given below:

- socket():Both the client and the server creates the socket() function.
- bind():It is typically used on the server side, and bounds a socket with a socket address structure, i.e. a specified local port # and IP address
- listen():It is typically used on the server side passively waiting for incoming connections from the client.
- sendto():It is on both client side and server side .It is used to send a datagram to another UDP socket
- recvfrom(): It is on both client side and server side . It is used to receive a datagram from another UDP socket.
- close():close a socket(tear down the connection).

6.3. Server Socket Class Methods

Java Network. Server programmes utilise the server Socket class to get a port and listen for client requests. [7] There are four constructors for the server Socket class:public server Socket(1) throws IO Exceptiontries to build a server socket connected to the given port. If another programme has already bound the port, an exception will happen.IO Exception is thrown by public Server Socket(int port, int backlog, Inet Address address).The Inet Address option, like the preceding functionObject(){[native code] }, defines the local IP address to bind to. When a server has more than one IP address, it can choose which IP address it will use to receive client requests by using the Inet Address. IO Exception is thrown by public Server Socket().creates a server socket that isn't bound. When using this function Object() { [native code] }, utilise the bind() method to bind the server socket when you're ready. The absence of an exception from the Server Socket function Object() { [native code] } indicates that your programme has successfully bonded to the chosen port and is prepared to receive client requests.

6.4.Socket Class Methods

Java Network. The socket that the client and server utilise to communicate with one another is represented by the socket class. The server gets a Socket object from the return result of the accept() method, whereas the client gets one by instantiating one.IO Exception, Unknown Host ExceptionThis approach tries to establish a connection with the specified server using the supplied port. The connection is successful and the client is connected to the server if this function Object() { [native code] } does not throw an error.public Socket(InetAddress host, int port) raises an IO ExceptionThe host is indicated via an InetAddress object in this method, unlike the previous function Object() { [native code] }.IO Exception is thrown by public Socket(String host, int port, InetAddress local Address, int local Port).creates a socket on the local host at the specified address and port by connecting to the specified host and port.public Socket raises an IO Exception Inet Address host, int port, InetAddress local Address, int local Port).The only difference between this method and the previous function Object() {[native code] } is that the host is indicated by an InetAddress object rather than a String.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

VII. HISTORY, OVERVIEW AND DETAILED DISCUSSION ON REMOTE METHOD INVOCATION (RMI)

The Java RMI-IIOP protocol was developed to make it easier to design CORBA applications while maintaining all its key advantages. It was created by Sun Microsystems and IBM, and it combines Java RMI technology's advantages with CORBA technology's features. [11] RMI-IIOP stands for the Java Remote Method Invocation (RMI) interface over the Internet Inter-Orb Protocol (IIOP), which gives the Java 2 platform access to CORBA's distributed computing features. It is based on CORBA/IIOP 2.3.1 and the Java Language Mapping to OMG IDL specifications.

7.1 What is RMI?

It is a way for calling a function or class object from a distance. It functions as a client-server arrangement. Client Object: A client object is an object whose method makes a remote call. Server Object: The term "server object" refers to the remote object. When a client port wants to call a remote method on a distant object, it really invokes a regular Java method, which is bundled inside an object called a stub. It is kept on a client computer rather than a server. Parameter marshalling, which is used in this packing, is a device-independent encoding for the parameter. There are various approaches to creating apps, ranging from low level socket programming through COM, DICOM, CORBA, and RMI. An object operating in one Java virtual machine can call methods on an object running in another Java virtual machine using the Java Remote Method Invocation (RMI) framework. RMI enables cross-program remote communication between Java-programmed applications. A client and a server. Typically, a server software will build some distant objects, make references to them accessible, and then wait for clients to call the objects' methods. In a typical client programme, methods are called on one or more remote objects that are located on a server by obtaining a remote reference to those objects. RMI offers the means for information transfer and communication between the server and the client. A distributed object application is another name for this type of programme.

7.2 RMI Architecture

In order to associate (or bind) a name with a remote object, as seen in Fig. 5, the server makes a call to the registry. The client first searches the server's registry for the remote object by name before calling a method on it. The illustration also demonstrates how, when necessary, the RMI system loads class definitions for objects from server to client and client to server using an already-existing web server. Understanding the RMI architecture is made possible by

Fig.

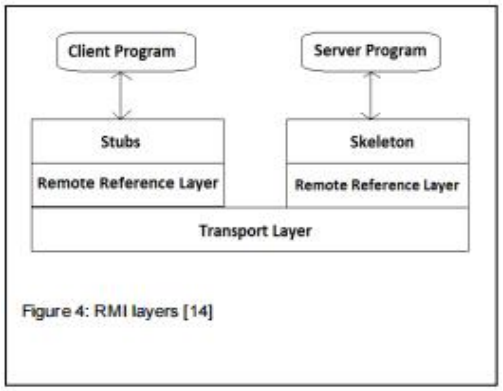


Figure 4: RMI layers [14]

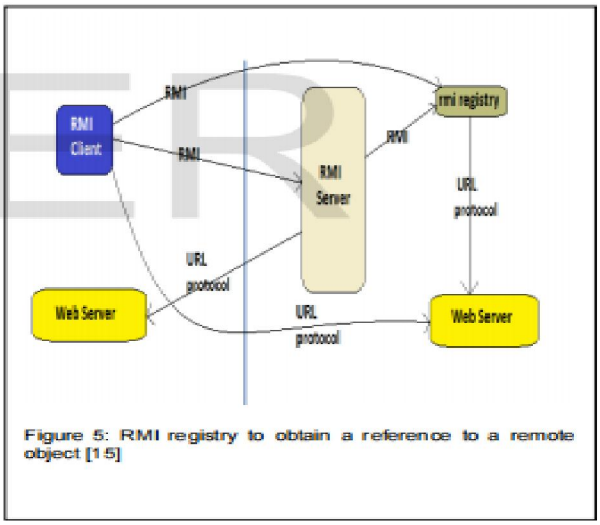


Figure 5: RMI registry to obtain a reference to a remote object [15]

RMI Registry

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

7.3 Advantages of RMI

Object-oriented: Unlike specified data types, RMI allows entire objects to be passed as arguments and returned as values. This implies that complex types, like a typical Java hash table object, can be passed as a single input. A hash table would need to be recreated on the server in existing RPC systems after the client decomposed the object into primitive data types, shipped those data types, and did so.RMI enables direct object transmission across the wire without additional client programming.

VIII. MOBILE BEHAVIOUR

RMI can transfer behaviour from a client to a server and vice versa. For instance, you may provide an interface for checking employee expense reports against existing corporate guidelines. An object that implements that interface can be obtained from the server by the client during the creation of an expenditure report. The server will start returning a different implementation of that interface that adheres to the new policies when the policies change. Because no new software needs to be installed on the user's computer, the limitations will be checked on the client side, giving the user faster feedback and reducing the load on the server. You have the most flexibility with this because adding or removing policies just requires you to build a single new Java class and instal it once on the server host.

8.1. Design Patterns

The entire potential of object-oriented technology can be utilised in distributed computing, such as two- and three-tier systems, by passing objects. Use object-oriented design patterns in your solutions where you can pass behaviour. The effectiveness of every object-oriented design pattern depends on a separate set of behaviours; without providing complete objects—both implementations and types—the movement's advantages are lost.

8.2. Safe and Secure

RMI makes use of Java's built-in security features to keep your system secure while users download implementations. RMI employs the security manager created to defend systems against malicious applets to safeguard your network and systems from potentially malicious downloaded programmes. A server may in extreme circumstances deny the download of any implementations at all.

8.3. Easy to Write/Easy to Use

Writing remote Java servers and Java clients that connect to those servers is made simple by RMI. An actual Java interface is a remote interface. A server is declared as such in about three lines of code and otherwise behaves like any other Java object. Due to its simplicity, full-scale distributed object systems can be swiftly created as well as early software versions and prototypes for testing and assessment. RMI programmes are likewise simple to maintain because they are simple to write.

8.4. Connects to Existing/Legacy Systems

Through Java's native method interface JNI, RMI communicates with already-existing systems. You can create a Java client and use your current server implementation by using RMI and JNI. You can rewrite any portions of your server in Java and take use of all Java has to offer when connecting to existing servers using RMI/JNI. Like this, RMI leverages JDBC to communicate with existing relational databases without changing the existing non-Java source that makes use of the databases.

8.5. Write Once, Run Anywhere

The "Write Once, Run Anywhere" philosophy of Java includes RMI. Any RMI-based system, as well as an RMI/JDBC system, is 100% portable to any Java Virtual Machine*. Any Java virtual machine will compile and execute JNI-written code if you utilise RMI/JNI to communicate with an existing system.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

8.6. Distributed Garbage Collection

RMI collects remote server objects that are no longer being used by any clients on the network using its distributed garbage collection function. With distributed garbage collection, which is similar to garbage collection inside a Java Virtual Machine, you can define server objects as needed while knowing that they will be deleted whenever clients no longer need to be able to access them.

IX. CONCLUSION

The following conclusions concerning the development of steganographic techniques in the network layer may be taken from the activities. Steganography and cryptography are used in the development of data security to give data security, authenticity, and integrity, as well as providing information that the receiver may trust. In-depth information regarding sockets, ports, socket programming over TCP, and a little bit of UDP, is covered in this work .Sockets are used in network programming to provide inter process communication between hosts, serving as the terminus of this communication. Since Internet protocol provides the foundation for computer communication in this context, sockets can also be referred to as network or Internet sockets. Socket programming is also a form of network programming. In this paper, sockets, ports, socket programming over TCP, and a little bit of UDP, are covered in detail. Sockets are the endpoint of interprocess communication in network programming, which uses sockets to communicate between hosts' processes. Since Internet protocol is used to communicate between computers, sockets can also be referred to as network or Internet sockets in this context. Thus, socket programming also includes network programming. Additionally, the paper discusses socket programming in Java over TCP. In this essay, a study of socket programming and remote method invocation was attempted. It is discussed how socket programming is superior to RMI. Although Socket programming is ideally suited for client-server communication, the emergence of RMI cannot be disregarded. Even more crucial, but only for extremely certain jobs. Socket programming and RMI can't be quickly described due to the paper's length and scope restrictions, but an effort has been made to explain how they compare. Since its inception, socket programming has been the subject of research. Java is replacing socket programming, although it still lacks Java's portability. Considering every factor, I think socket programming still has to develop in terms of types and how it will function in apps. All of these call for the consideration of having quicker and more dependable interactions between the server and clients.

REFERENCES

[1]. In IPv6 IEEE 2015 International Conference on Computing Communication Control and Automaton, Bobade S. and Goudar R. (2015) present Secure Data Communication Using Protocol Steganography.
[2]. Improved Smart Power Socket for Monitoring and Controlling Electrical Home Appliances, Hassan E A, Shareef H, Islam M M, Wahyudie E, and AbdrabouAA 2018, IEEE Access 6, p. 49292-49305.
[3]. Emblogic India Pvt. Ltd Noida.
[4]. http://en.wikipedia.org/wiki/Berkeley_sockets
[5]. http://www.tutorialspoint.com/java/java_networking.htm
[6]. Gellings, P. Smart Grid Planning and Implementation; River Publishers: 9260 Gistrup, Denmark, 2020.
[7]. Utilizing Digital "Micro-Mirror" Devices for Ambient Light Communication by Xu, Tapia, and Ziga. Pages. 387–400 in Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), Renton, Washington, USA, 4-6 April 2022.
[8]. Professionals in WiFi, 2022. The 4 Way Handshake is accessible online at https://www.wifi-professionals.com/2019/01/ (Retrieved on October 1, 2022)
[9]. Liquid Crystal Technologies. 2022. Available online: http://www.liquidcrystaltechnologies.com/products/lcdshutters.htm (accessed on 1 November 2022)
[10]. http://docs.oracle.com/cd/E19683-01/816-5042/6mb7bck68/index.html
[11]. http://findaccountingsoftware.com/directory/rmi-corporation/rmi-advantage
[12]. http://www.cs.rutgers.edu/~pxk/rutgers/notes/sockets/index
[13]. http://www.oracle.com/:ORACLE

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, January 2023
Impact Factor: 7.301

[14]. Lashkari, A.H.; Danesh, M.M.S.; Samadi, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i). In Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 11 August 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 48–52

[15]. Mishra, A.; Arbaugh, W.A. An Initial Security Analysis of the IEEE 802.1 X Standard; Technical Report CS-TR-4328. University of Maryland, College Park; 2002