International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 196-200
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

LAN SCAN:ERROR DETECTION & FAULT CORRECTION

Ankit Ranjan, Department of IT
AmityUniversity, India

Aditi Malviya
Dept. Of CSE
Amity University, India

Abhishek Srivastava
Amity University, India

Abstract–LAN Scanner is a small and highly configurable network scanner for a NETWORK. And it's very fast. LAN Scanner uses multithreading so you can scan nodes at a very fast speed. If used to scan ports, LAN Scanner can scan 100s of ports in less than a minute! This superfast scanning ability is only one of the amazing features of LAN Scanner. It performs very deep scans on any network you wish, extracting its users and services, share and many other useful data and it can also connect to a target machine using the default user rights, or you can specify an IP address to use. It includes powerful export options to describe your own save formats.

I have chosen java to create LAN scanner server/client applications as it provides platform independency which is an important concern for any ftp application, and its powerful but easy to use and understand socket programming unlike the socket programming in c++ using Winsock etc. or other languages. While implementing it I faced many problems and "bugs" regarding input/output stream buffer, firewall, connection establishment and last but not the least transfer of large files. Most of them were fixed but the problem of transfer of large files is not fixed yet and I am still working on it.

Keywords-- LAN Scanner; Sockets; Java; Server; Client

I. INTRODUCTION

LAN SCANNER [1] is used by the network administrator to detect whether there is an active connection in the network or not. It uses the socket programming interface in java.

The Purpose of using java is to utilize the java and socket programming for the purpose of creating client server application.

LAN Scanner is used to scan various node. Scan individual nodes. It uses the IP address for a specified node It uses the awt and applet for its implementation

The two end uses socket programming in order to communicate with each other.

LAN Scanner between two computers requires:

A transfer of files between different computer systems can be regarded as a service and it is divided into four steps:

1. access to a file, which is administrated in a File Management System (FMS) [2], residing on one computer system
2. transfer of the file data by any transport medium
3. creation of a file copy in another FMS
4. additional operations on the transferred file like print out or conversion

In order to make two computers communicate we need to first establish a connection between them, in java this can be done by socket programming.

Connection between two computers can be established by implementing two application layer protocols:

TCP/IP and UDP/IP

Datagram Communication

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 196-200
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

The datagram communication protocol, known as UDP (user datagram protocol) [3], is a connectionless protocol and it means that each time you send datagrams and you also need to sends the local socket descriptor and the receiving the desired socket's address. As you can tell, additional data must be sent each time when a communication is made.

Stream communication:

The stream communication protocol is known as TCP (transfer control protocol) [4] and unlike UDP, TCP is a connection-oriented protocol and in order to do communication over the TCP protocol, firstly a connection must be established between the pair of sockets and one of the sockets listens for a connection request (server), the other asks for a connection (client). Once two sockets have been connected and they can be used to transmit data in both (or either one of the) directions.

Sockets in Java

Like all other functionalities provided by Java and this functionality to works with sockets are also "packaged" as a package and its classes [5]. These are the following package and its main classes that help in accessing sockets:

- *Java.net package*
- *Server Socket*
- *Socket*

Among the above, Java abstracts out most of the low-level aspects of socket programming. The java.net package contains all the classes required to create network enabled applications. Server Socket and Socket are also the part of this package. Apart from these classes, it may also contain classes to connect with the web server, create secured sockets, and so forth.

The Server Socket class provides server sockets or sockets at server side and such sockets wait for requests over the network and when such requests arrive, a server socket performs operations based on the request and may return a result and the class wraps most of the options required to create server-side sockets. The Socket class provides client-side sockets or simply sockets and they are at the client side connecting to the server and also sending the request to the server and accepting the returned result. Just as Server Socket exposes only the compulsory parameters required to create a server-side sockets and similarly, Socket asks the user to provide only those parameters that are most necessary.

Java uses many stream-based mechanisms throughout the language to achieve the I/O. For examples all file I/O and memory I/O in Java is achieved through streams and most of the messaging is also carried out using streams although basic datagram implementations are also supported.

To understand where sockets fit in protocol stack we need to outline the protocol stack which looks like:

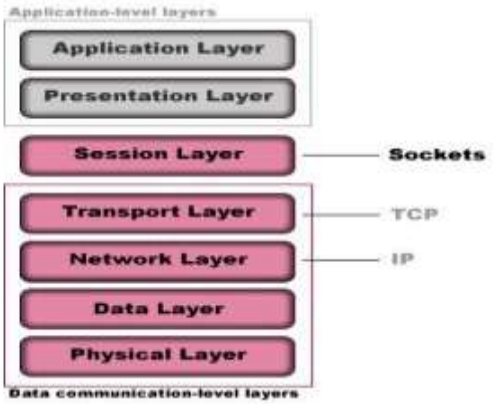

Figure 1: - Data Communication-Level Layers

Any ftp application requires two important parts: **the code that executes at client-side and the code that executes at server-side.** So by using the functionality of sockets can be partitioned into two major steps:

- **The server or the server-side code**
- **The client or the client-side code**

The multi-threaded nature of former can always be guaranteed whereas the later may or may not be multi-threaded.

The Server

The server's main function is to wait for the incoming requests, and also to service them when they come in. and so the code to implement the server can be further broken into the following steps:

1. Establish a server that monitors a particular port and this is done by creating an instance of the **Server Socket** class. There are four different ways to create an instance of Server Socket class and they are:

a. **Server Socket ()**, which simply sets the implementation that means everything is taken as default values.

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 196-200
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

b. **Server Socket (int port)**, which creates a server-side socket and binds the socket to the given Port Number.

c. **Server Socket (int port, int port)**, which not only binds the created socket to the port but also create a queue of length specified by the number passed as the backlog parameter.

So to create a socket bound to port number 5217with a backlog queue of size 5 and bound with address of local host the statement would be:

Server Socket server = new Server Socket (5217, 5);

One point is always keep in mind is that the above mentioned constructors return TCP sockets and not UDP sockets.

2. The next step is to tell the newly created server socket to listen indefinitely and accept incoming requests. This is done by using the **accept ()** method of Server Socket class. When a request comes, accept () returns a Socket object representing the connection and in the code it would be:

Socket incoming = server. accept();

3. Communicating with the socket and it means reading from and writing to the Socket object and to communicate with a Socket object, two tasks have to be performed simultaneously. First the Input and Output stream corresponding with the Socket object has to be obtained and hat can be done by using the **get Input Stream ()** and **get Output Stream ()** methods of Socket class.

The second task is to read from and write to the Socket object and since the communication has to continue until the client breaks the connection and the reading from and writing to is done within a loop.

4. Once the client breaks the connection or stops sending the request, the Socket object representing the client has to be closed. This can be done by calling **close ()** method on the Socket object and the statement should be:

Incoming. Close ();

That's how a server is coded. The next section deals with creating a client.

The Client

The main purpose of the client is to connect to the server and communicate with it by using the connection and so coding for a client requires the following steps:

1. Connect to the server. Connecting to the server can be accomplished in two steps:

a. Creating a Socket object. The socket at client side just needs to know the host name (the name of the machine where server is running) and the port where the server is listening and to create such a Socket object, there are seven constructors provided by the Socket class and the most commonly used constructor are:

- Socket (), which creates a new Socket instance without connecting to host.
- Socket (int Address address, int port), which is used to create a new Socket object and connects to this port specified given address.
- Socket (java. lang. String host, int port), which works as the same way as that of Socket (), instead of an address, the host name is used.

So to create a Socket object that connects to 'localhost' at 5217, the statement would be:

Socket s=new Socket "localhost", 5217);

b. Connecting to the server comes into picture if no argument constructor is used and it takes as the object of the Socket Address object as an argument and it also connects to the localhost at port 8888 and the code would be:

Sockets=newSocket();
S.Connect (new Socket Address ("localhost", 5217));

2. Communicating with the server and communicating with the server by using a socket at client side is no different from how the server communicates with the client. Firstly, the input and output streams connected with the Socket object.

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 196-200
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

II. MATERIALS REQUIRED AND METHODOLOGY ADAPTED

Software / OS utilized for the coding:

- Eclipse IDE
- Windows 8 (Preferable Windows XP and above
- Win-32 MS DOS.exe (or Execution)

Hardware Requirements:

- Monitor
- Router
- Core i-3 processor and above
- Input methods - Mouse, keyboard.

III. ARCHITECTURE

The image below shows the major component of a LAN Scan

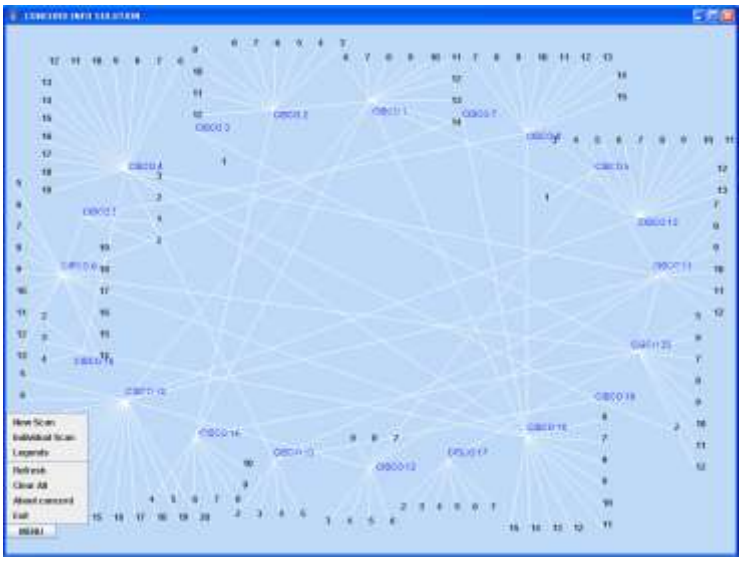

Figure 2: - Architecture of LAN Scanner

IV. HOW IT WORKS

- If the node is connected to the server, the link will be shown by a green line.
- The disconnected nodes that are not connected to the server are displayed in red.
- Yellow colour displays that the connection status is not available.

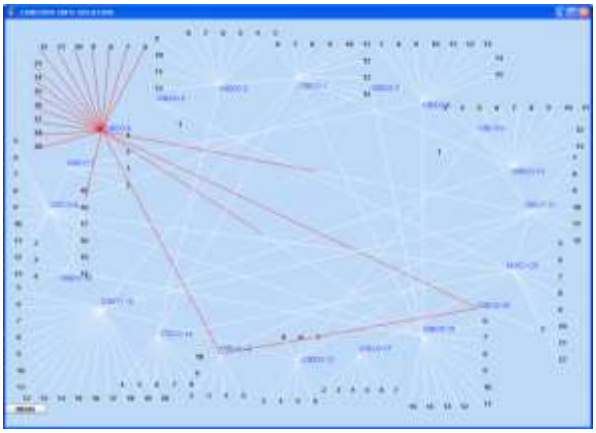

Figure 3:-Disconnected Nodes

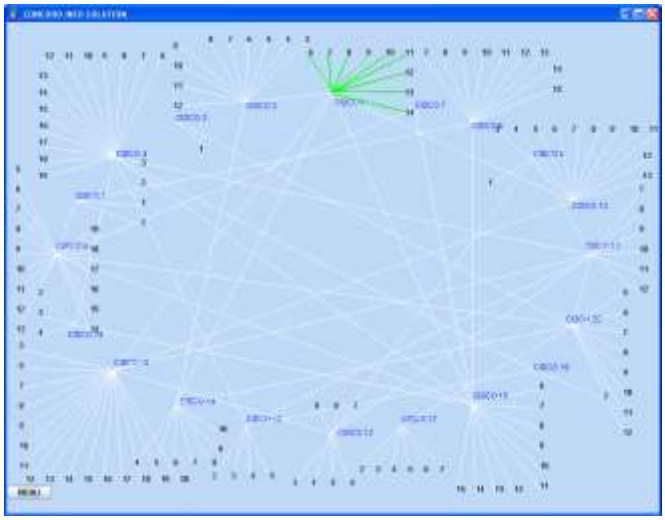

Figure 4: - Connected Nodes

V. FEATURES

- Reliability
  - Access to the scanning functionality is not altered by other possible failures of a multifunction device
  - The physical moving parts decreases likelihood of hardware malfunction
  - Straight through the paper path design helps decrease possibility of document jams
  - Network scanners inherit the attributes designed for the mission-critical document scanners

- Ease of Use
  - Eliminate the complexity and provides the simply operations with large touch screen
  - Simply the touch screen driven scanning operation eliminates the specialized training

In the unlikely event of the document jam, easy jam recovery without damaging documents should be done.

- Secure
  - Restrict access to only the authorized users with secure authentication
  - User data such as username or password or image data does not reside on the scanner
  - No external USB port is used to hijack sensitive information

VI. CONCLUSION

The paper manages LAN Scan. What Lan Scan is and what the world sees by LAN Scan. Why was it required or what lead to its advancement? How the world has imparted the information of Security and profited from it? While various methodologies, including systems and skeletons have been presented and embraced as industry measures and/or best practices that have alleviated a large portion of the issues. Security utilized within different circles/fields like Smart telephones, Social-media, E-trade, Cloud administrations, Information Security sways. LAN Scan is basic in just about in any engineering driven industry which works on workstation framework. The issues of workstation based framework and tending to their endless vulnerabilities are an indispensable piece of keep up an operational industry.

VII. REFERENCES

[1] W. A. Arbaugh, *.An inductive chosen plaintext attack against WEP/WEP2*. **IEEE Document 802.11-01/230**, May 2001.
[2] W. A. Arbaugh, N. Shankar, and Y. J. Wan..*Your 802.11wireless network has no clothes*..**http://www.cs.umd.edu/.waa/wireless.pdf**, Mar. 2001.
[3] Brian P. Crow, Indra Widjaja, Jeong Geun Kim, P. T. Sakai, *.IEEE 802.11 Wireless Local Area Networks.*, **IEEE Communications Magazine** , Sept. 1997
[4] N. Borisov, I. Goldberg, and D. Wagner, *.Intercepting Mobile Communications:*
The Insecurity of 802.11..
http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html
[5] White Paper on .Bluetooth Security., Bluetooth Special Interests Group