

# CSC 573 – Internet Protocols

## M-KODA Project Report

Arun Natarajan   Manikandan Sivanesan   Kousik Krishnakumar   Dinesh Radhakrishnan  
[anatar@ncsu.edu](mailto:anatar@ncsu.edu)   [msivane@ncsu.edu](mailto:msivane@ncsu.edu)   [kkrishn2@ncsu.edu](mailto:kkrishn2@ncsu.edu)   [dradhak@ncsu.edu](mailto:dradhak@ncsu.edu)

---

### Requirements:

- Java 6 SUN JDK
- JRE 1.6.0.16
- MySql

### Database:

This system uses MYSQL as the database in order to store and maintain User and File information published by the peers. In case of centralized approach MYSQL database is maintained in bootstrap server alone. And in case of distributed both the server and super peers maintains MYSQL database. Since Java being used in this project in order to connect with MYSQL database JDBC(**JAVA DATABASE CONNECTIVITY**) is used. Connector/J TYPE 4 , the native Java driver is used to convert JDBC calls into the network protocol used by the MYSQL database.

The bootstrap server maintains the database for IPTABLE and the FILETABLE in the centralized approach. IPTABLE maintains the information about the user and contains the user name as its primary key. Active Status field is updated by the Keep Alive Timer implemented in bootstrap server. The fields in IPTABLE are as follows:

IP Address	Host	Password	E-mail	Active Status	Port No

**Fig 1 – IPTABLE in Centralized System**

FILETABLE maintains the information about the file published by the peers. User and Message Digest are used together as primary key in the FILETABLE. Message digest is the field used as an identifier for the files using SHA-1 in order to uniquely identify a file. File Digest is the 1024byte field containing the digest of the file published by the peers using reference (1). Downloads field indicates the number of times the file downloaded by the peers but not by the owner of the file.

IP Address	User	Active Status	File Name	Message Digest	File Size	File Digest	Abstract	Downloads	Port No

**Fig 2 – FILETABLE in Centralized System**

In distributed approach IPTABLE maintained by the server contains the following fields

IP Address	Host	Password	Active Status	Email	Port No	Super Peer	SuperPeer Offer	LoadFactor

**Fig 3 – IPTABLE in Distributed System**

In addition to fields maintained in centralized tables Super Peer, Super Peer offer and Load Factor fields are added. Super Peer is used to distinguish between a normal peer and a super peer. Super Peer Offer field indicates the intention of peer to become a super peer. Load Factor field shows the number of peers under the control of super peer.

FILETABLE is created dynamically and maintained by each super peers separately and contains the following fields in the table.

IP Address	User	Active Status	File Name	Message Digest	File Size	File Digest	Abstract	Downloads	Port No	Hash Coordinate

**Fig 4 – FILETABLE in Distributed System**

### Application PDU (APDU)

The entire design of the application is done with respect to the APDU. Every request/reply and transfers take place in the form of the APDU. We use the below header format for all the packets transferred in the application. We have a “Packet Type” Field, which defines the type of packet. The contents of the packet Type field can take a value from 0-255 representing,

- 1 – Login
- 2 – Update Active Status
- 3 – Update Load Factor
- 4 – Keep alive timer
- 5 – New user Registration
- 7 – Update Super Peer Offer
- 8 – Peer connected with a Super Peer
- 9 – Peer Type Status
- 21- Request for Download
- 22- Reply for Download
- 31- Update Download Count
- 43- Update Super Peer Children
- 51- Remove File Entry
- 81-Login Reply
- 121 – Send File Search
- 122 – Result Content
- 123 -- Reset
- etc.....

The next field is “Packet ID” field, which uniquely identifies each request and reply between the peers themselves and with the server.

The third field is a “TTL” field, which is used only in the distributed approach. This field is decremented

on each hop and the request is discarded when the field reaches “0”. For the centralized approach the field is always infinity (the maximum value).

The fourth and fifth fields are the IP Address and port number used by the server and the clients. The port number is included in the header in order to support multiple clients from the same host. The sixth field is the Request/Reply field to differentiate the request packets and the reply packets. The Payload length field identifies the length of the packet. And the last field is the Payload field which carries the actual data portion.

Following this lies the data field of the APDU, it contains the details about the request/reply. This field's contents are better discussed under their respective headings.

Packet Type	Packet ID	TTL	IP Address	Port No	Request/Reply	Payload Length	Payload
-------------	-----------	-----	------------	---------	---------------	----------------	---------

**Fig 1a. APDU Format for Request / Reply**

### **Project Implementation:**

#### **New User Registration:**

A new user can be created by clicking on the Register button on the Log In Screen. Details such as user name, password, confirm password and email address are got from the user and check whether the password matches. If so, these are sent to server using an UDP packet. The payload of the UDP packet containing the user name length and user name, digest of password using SHA-1 algorithm, length of this password digest and finally the email id and its length. The server checks whether the user name (host name) already exists in the IP Table. If present, send an registration failed message to the user or else add the new user details into the IP Table.

#### **Log In Authentication:**

An existing user can log into the M-Koda search engine by entering the user name and password in the Log In screen. These informations are sent using a UDP packet containing user name, length of user name, password digest that is hashed using a SHA-1 message digest function and its length to the server. The server verifies the user name and password from its IP Table. If they match then it sets the active status field of that user to 1 and also update the active status value to 1 for that user in File Table so that the files published by that user can be accessed by other users. This makes the search better by not showing the search results of files whose active status is 0 as the other peers cant download a file from a peer who is not currently logged into the system. The Server now send the verified user name result in a UDP pack back to the client that displays result to the user of whether the login is success or failure. If the login is successful then the user logs into the M-Koda search engine.

#### **File Publish and Unpublish:**

A user can publish new files into the system and can unpublish them whenever needed. When a user needs to publish a new file they can select the file from the file chooser window. Add an abstract about the file and publish it. The client sends these information to the server in a TCP packet that contains the user name,length of user name, file name and its length, file abstract and its length, file size, signature of the file trigest and also the file digest that is hashed using SHA-1 message digest function and it is used as an unique ID for that particular file the user publishes. This restricts the user to publish a file only once. The server extracts these details of the file from the TCP stream and updates or adds the new file into the FILE Table along with IP address of user and the port number. It sets the number of downloads field to 0 initially. This field is used for

document ranking in displaying the text search results. Notification of file publish error will be sent to the user in case of file already published from the same user. If the file publish is successful the user is notified about it and a SHA\_Path file for this user will be saved at the client side having the list of files the user published and the absolute path and digest for those files. This is used to check consistency of the files that the user have published.

When the user wishes to Unpublish the files he has published, an Unpublish window opens showing all the files the user has published. This list is got from the SHA\_Path file that have been saved for each user in the client side. File consistency is done based on this list when the user logs in. The client checks whether the files are present in their absolute path and ensures they are unchanged by checking its message digest. The file entry will be deleted both from server FILE Table and the client SHA\_Path file and client will be notified. When the user unpublish a file, the user name and file's message digest is sent through an UDP packet to server where the server matches this with the FILE Table and deletes that entry.

### **Text Search:**

Searching a text in the list of files published at the server in the centralized approach has lot of issues to be dealt with. The search results must provide only the file names of those users who are currently active. The search is done based on the reference paper provided. Signature of the search query given by the user that is hashed using message digest algorithm and the user name is sent in a TCP stream using a TCP connection. The server retrieves the search query and checks in its database for those files whose host name's active status is 1. The search results are sent in a TCP stream as a set of IP address, port number, message digest, file name, size and number of downloads for each match the server finds. The user is informed if there are no results found. The client then retrieves this TCP stream and displays the search result to the user showing the file name, user name that published the file and number of downloads.

### **File Download:**

As mentioned earlier the download count is used for document ranking by ordering the files based on the number of times a file is downloaded. From the resulting search results the user can select any file and save the file into the desired folder from the file save window by clicking on the desired result. The peer now sends an UDP pack containing message digest of that file to the corresponding peer's IP Address that has published that file which in turn sends the file requested in a TCP packet. After the user completes downloading the file an UDP packet is sent to centralized Server to increase the download count of that particular file.

### **Keep Alive Timer:**

The Keep Alive timers are used for timer management to check the status of a user who has logged in. Thus as soon as an user logs into M-Koda a thread is started to hear incoming keep alive messages from the Server. The Server periodically sends keep alive message to all those peers that are currently logged in. The Server then waits till the timer expires. On the client side, whenever the client receives a keep alive message from the server, it must respond with a hello message before the timer expires at the Server. If the server doesn't receive a hello message from a peer till the timer ends then it changes its active status to 0 in its database thinking the user is down and not responding.

## **Features**

### **1) NAT Box Resolved**

This is resolved using UPNP protocol. This protocol discovers the routing devices and adds a mapping

entry for local address and the local port used in the NAT box before entering the system. And before leaving the system it deletes the mapping entry in the NAT box.

## 2) File Maintenance and Consistency

The list of files along with its details that are being published by the user is maintained in the client machine. This is done to help us in the faster download of file and the location of the file be known only to the client who have published the file. Whenever a user is logged into the system a file named Username\_SHA\_path\_file is created if it does not previously exist and this file stores the SHA 1 of the file which is the unique identifier of the file and the absolute or complete path of the file. This feature protects the privacy of the user as the location of the file is not known to anyone else.

Once a file is created it maintains all the list of files that has been published by that user. Even if a user logs out all these details are maintained. When the user logs back, the consistency of these files are checked and this done by sending the SHA 1 of all the files that are available with the user are sent to the server and it is matched with the SHA 1 that is available in the server for that corresponding user. If they are matched then the file is said to be consistent. If they do not match then there is some inconsistency and so the entry that are present in the file which are inconsistent are deleted.

## 3) Multiple Client Support

The one of the interesting feature of this project is that it supports multiple clients on the same machine. It is possible for one user to log in while some other user has already been logged in from that machine at the same time. If we have used one port number for one IP address then the multiple client support will not be possible. This is made possible only because of using the dynamic port numbers for different users. Only the server port is a well known port number and the port numbers of the clients can be anything.

## 4) User Mobility

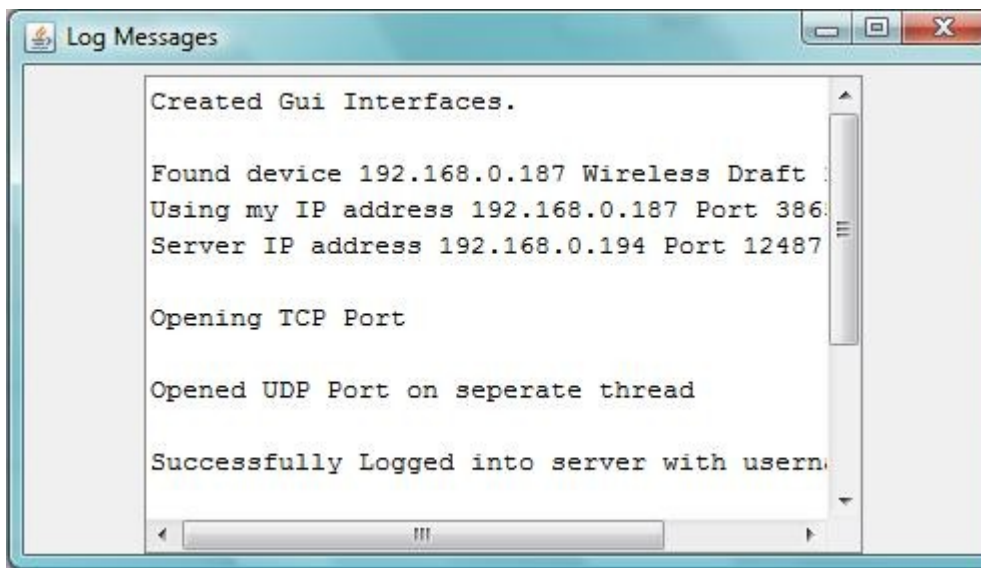
The user is not restricted to use the same system. The user can publish some files in one system and he may log out then he may sign in from some other system. In such a case a new file for storing the user's details is created. The file that is present in the previous system will remain as such and the files consistency will not be altered because of this. The user may again log back to his previously using system where the list of files he has published from this system is maintained and this provides him the ability to use any system and not restricting him/her to a particular system.

## 5) Security

Security is one of the key issues in any system and this is addressed with utmost importance. All the important data are encrypted and then only sent. The password which is got from the user for his login is encrypted and then only sent to the server.

## 6) User Friendly GUI

The user interface is designed such that it is user friendly and it will clearly take the user in a step by step process for doing what he wants to do. There is also a provision for seeing the various processes which are going on for that particular user by clicking on the messages button which displays the log messages. The diagram shown below shows the log messages.



Sample Screen shots:

A screenshot of a "Registration Panel" window. The panel has a title bar and a main content area. The title "Registration Panel" is in the top left. The main heading "Register New User" is centered. Below the heading are four labeled input fields:  
User Name: [manikandan]  
Password: [•••]  
Confirm Password: [•••]  
Email ID: [mani@abc.com]  
At the bottom of the panel are two buttons: "Register" and "Cancel".

Fig 1 : New User Registration

**USER LOGIN**

Server : 192.168.0.194 (12487)

## MKoda Search

User Name:

Password:

**Fig 2: Login Screen**

**Publish File**

File : a.c (402 bytes)  
/home/mani/a.c

Abstract / Comments :  
this is an abstract

**Fig 3 : Publish File**

Search Files

Welcome test1

Enter the text to be searched:

Search

Previous

Next

Download

Download

Download

Download

Download

Download

Download

Download

Download

Download

Download

Publish

UnPublish

Messages

Log Out

**Fig 5: M-Koda Search Engine**



Search Files

Welcome test1

Enter the text to be searched:

a

Search

Previous

Page 1/1

Next

To be done (44 bytes, 0 downloads) by : test1

Download

alg hw1.doc (59904 bytes, 0 downloads) by : test1

Download

empt\_1 (10 bytes, 0 downloads) by : test1

Download

Picasa.ini (102 bytes, 0 downloads) by : test1

Download

dump.sql (7727 bytes, 0 downloads) by : test1

Download

login.txt (164 bytes, 0 downloads) by : test1

Download

triplet\_frequency\_2 (60556 bytes, 0 downloads) by : test1

Download

abc (6 bytes, 0 downloads) by : test1

Download

a (17 bytes, 0 downloads) by : test1

Download

a.c (402 bytes, 0 downloads) by : test1

Download

this is an abstract

Publish

UnPublish

Messages

Log Out

**Fig 7 : Search Results**

**References :**

<http://www.sbbi.net/site/upnp/docs/natmappings.html>

<http://dev.mysql.com/doc/>

<http://java.sun.com/docs/books/tutorial/index.html>