**Thanks**

The report on graduation of information technology major with the topic "Building a chat software system on low bandwidth environment" is the result of our continuous efforts and the support and encouragement. Rules of teachers, friends, colleagues and relatives. Through this page, I would like to thank those who have helped me during the past time of studying.

I would like to express my deep respect and gratitude to Mr. Nguyen Tien Thanh for directly guiding and providing necessary scientific information for this project.

In the course of implementing the graduation project, I realized that I tried my best, but because the knowledge was still limited, there were still many shortcomings, I hope teachers will add more products to be more complete.

**Summary of the topic**

I use Java language to implement, Apply rmi technology in Java to simplify the design of client server model. The database I use is Mysql. The conversion of the returned data into an object is automated using reflection. To ensure safety during the exchange process, the feature of ssl will be applied to rmi

Following the recommendations of the ITU organization, the voice will be sampled at 8000Hz, a sample represented by 16 bits. I decided to use the g729 encoding standard to encode audio to send with 16 times compression. Use symmetric encryption to ensure confidentiality

While doing the topic, I had a lot of difficulties in handling input audio because this was my first time studying sound processing in real time, so the processing of the sound was still empty. The next one is difficult due to the use of Java. It does not allow me to intervene deeply into the system, so my anti-jiiter and buffer designs have not been implemented yet.

Through the process of implementing the topic. I reviewed the ability to manipulate with database, learned how to simplify the problem, learned new techniques of java.

In the future of the topic, I will overcome the remaining disadvantages, fix the beautiful interface and develop with the mobile platform

Thank you sincerely!

Dang Cong Can

Contents

[Catalog images 4](#_Toc43764831)

[Overview of the topic 6](#_Toc43764832)

[General introduction 6](#_Toc43764833)

[Theory 6](#_Toc43764834)

[About rmi 6](#_Toc43764835)

[Compare XML and Json 8](#_Toc43764836)

[What is SSL, what is an SSL certificate? 9](#_Toc43764837)

[Reflection technique 9](#_Toc43764838)

[Gson library 11](#_Toc43764839)

[Audio encoding 12](#_Toc43764840)

[Handling jitter 13](#_Toc43764841)

[Mesh Topology 15](#_Toc43764842)

[Use-case designs 16](#_Toc43764843)

[Overview of functions 16](#_Toc43764844)

[Usecase details 18](#_Toc43764845)

[Functions for group administrators 26](#_Toc43764846)

[The non-functional 28](#_Toc43764847)

[Design 30](#_Toc43764848)

[Design database for group chat 30](#_Toc43764849)

[Design Process of conversation 33](#_Toc43764850)

[Rmi interface design 35](#_Toc43764851)

[Build server 43](#_Toc43764852)

[Real time data transmission 47](#_Toc43764853)

[Support classes 50](#_Toc43764854)

[Data transfer 51](#_Toc43764855)

[User interface design 52](#_Toc43764856)

[Restrictions exist 60](#_Toc43764857)

[CONCLUDE 62](#_Toc43764858)

[Result 62](#_Toc43764859)

[Development direction of the project 62](#_Toc43764860)

[Appendix 63](#_Toc43764861)

[Structural types 63](#_Toc43764862)

[XML 63](#_Toc43764863)

[JSON 65](#_Toc43764864)

[Protocol 66](#_Toc43764865)

[Tranmission Control Protocol 66](#_Toc43764866)

[UDP (User Datagram Protocol) 68](#_Toc43764867)

[RTP 69](#_Toc43764868)

[Hibernate 72](#_Toc43764869)

[What Is Hibernate In Java? 72](#_Toc43764870)

[What Is An ORM Tool? 73](#_Toc43764871)

[Hibernate architecture 74](#_Toc43764872)

[Network Address Translation (NAT) 74](#_Toc43764873)

[Network Address Translation (NAT) working 74](#_Toc43764874)

[NAT inside and outside addresses 75](#_Toc43764875)

[Network Address Translation (NAT) Types 75](#_Toc43764876)

# Catalog images

[Fig 1 rmi architecture 7](#_Toc43764730)

[Fig 2 Json and Xml 8](#_Toc43764731)

[Fig 3 demo of json 8](#_Toc43764732)

[Fig 4 result test 9](#_Toc43764733)

[Fig 5 Java Reflection 10](#_Toc43764734)

[Fig 6 Gson 11](#_Toc43764735)

[Fig 7 Mesh Topology 15](#_Toc43764736)

[Fig 8 main funtion for user 1 16](#_Toc43764737)

[Fig 9 main funtion for user 2 16](#_Toc43764738)

[Fig 10 main funtion for user 3 17](#_Toc43764739)

[Fig 11 Use cases for group administrators 17](#_Toc43764740)

[Fig 12 Use non-functional case 18](#_Toc43764741)

[Fig 13 login 18](#_Toc43764742)

[Fig 14 Firend 19](#_Toc43764743)

[Fig 15 Information editing 20](#_Toc43764744)

[Fig 16 Messaging 20](#_Toc43764745)

[Fig 17 Join the group 21](#_Toc43764746)

[Fig 18 download 22](#_Toc43764747)

[Fig 19 Group exit 23](#_Toc43764748)

[Fig 20 friends list 23](#_Toc43764749)

[Fig 21 Delete friends 24](#_Toc43764750)

[Fig 22 create groups 24](#_Toc43764751)

[Fig 23 voice chat 25](#_Toc43764752)

[Fig 24 reset the password 25](#_Toc43764753)

[Fig 25 Admin 1 26](#_Toc43764754)

[Fig 26 Admin 2 27](#_Toc43764755)

[Fig 27 Admin 3 27](#_Toc43764756)

[Fig 28 Admin 4 28](#_Toc43764757)

[Fig 29 new message 28](#_Toc43764758)

[Fig 30 old messages 29](#_Toc43764759)

[Fig 31 list of groups 29](#_Toc43764760)

[Fig 32 Member Information Table 30](#_Toc43764761)

[Fig 33 Group information table 30](#_Toc43764762)

[Fig 34 Message table 31](#_Toc43764763)

[Fig 35 File Table 32](#_Toc43764764)

[Fig 36 Request Table 32](#_Toc43764765)

[Fig 37 Friend table 32](#_Toc43764766)

[Fig 38 Mutil Media 48](#_Toc43764767)

[Fig 39 GUI Login 52](#_Toc43764768)

[Fig 40 GUI suport 53](#_Toc43764769)

[Fig 41 GUI Sign up 53](#_Toc43764770)

[Fig 42 email 1 53](#_Toc43764771)

[Fig 43 GUI xac nhan 54](#_Toc43764772)

[Fig 44 GUI Chang pass 54](#_Toc43764773)

[Fig 45 email 2 54](#_Toc43764774)

[Fig 46 GUI pass 54](#_Toc43764775)

[Fig 47 GUI info 55](#_Toc43764776)

[Fig 48 GUI Edit info 55](#_Toc43764777)

[Fig 49 GUI add friend 56](#_Toc43764778)

[Fig 50 GUI delete friend 56](#_Toc43764779)

[Fig 51 GUI Make friente 57](#_Toc43764780)

[Fig 52 GUI team list 57](#_Toc43764781)

[Fig 53 GUI create group 57](#_Toc43764782)

[Fig 54 GUI message 57](#_Toc43764783)

[Fig 55 GUI Group member 58](#_Toc43764784)

[Fig 56 GUI get Code 58](#_Toc43764785)

[Fig 57 GUI join group 58](#_Toc43764786)

[Fig 58 GUI chat 58](#_Toc43764787)

[Fig 59 GUI Admin 59](#_Toc43764788)

[Fig 60 GUI Requets 59](#_Toc43764789)

[Fig 61 GUI set Admin 59](#_Toc43764790)

[Fig 62 GUI delete admin 60](#_Toc43764791)

[Fig 63 GUI add member 60](#_Toc43764792)

[Fig 64 GUI delete group 60](#_Toc43764793)

[Fig 65 Appendix Xml 65](#_Toc43764794)

[Fig 66 Appendix Json 65](#_Toc43764795)

[Fig 67 Appendix Tcp header 66](#_Toc43764796)

[Fig 68 Appendix Tcp work 68](#_Toc43764797)

[Fig 69 Appendix Udp header 68](#_Toc43764798)

[Fig 70 Appendix Rtp header 69](#_Toc43764799)

[Fig 71 Appendix Orm 73](#_Toc43764800)

[Fig 72 Appendix Hibenate 74](#_Toc43764801)

[Fig 73 Appendix Nat 75](#_Toc43764802)

# Overview of the topic

## General introduction

Information technology has an increasingly important role in all areas of life, especially in information transmission. Today, with the development of information technology in general, the Internet has transformed a part of the world. The development of the Internet creates close proximity among regions, countries in terms of science and technology as well as social information needs.

The need to exchange information with people is booming. Instead of having to go back and forth to exchange information, we can use the Internet to solve that problem. Internet-based communication applications are well developed

If the client / server communication model is applied to the conversation. The client sends the packet, the server receives it and sends the packet to other clients, Theoretically can be achieved. But there is a problem that the more people who talk, the bandwidth, the resources, ... the more exponentially consumed. Making such a system would cost a lot to build and maintain

That's why I decided to implement the topic Building a chat software system on low bandwidth environment to solve that problem.

## Theory

### About rmi

RMI - Remote Method Invocation is a technique of installing distributed objects in Java. RMI is part of the J2SDK suite and is a library function that supports remote method calls and returns values for distributed computing applications. To use rmi, the Java programming language must be used on both the calling and the calling methods.

To solve some problems in communication between Client / Server. RMI does not call directly but through intermediaries. This class exists on both the client and server side. The class on the Client machine is called Stub, the class on the Server machine is called Skel (Skeletion).

#### Rmi characteristics

* RMI is Java's distributed object model, RMI makes it easier to communicate between distributed objects in the internet environment.
* RMI is a high-level API built on Socket programming.
* RMI not only allows us to transfer data between objects on different computer systems, but also invokes methods in remote objects (Remote Object).
* Data transmission between different machines is handled transparently by the Java virtual machine.
* Similar to the Client / Server model, RMI maintains the concepts of Client and Server, but RMI's approach is more flexible than a Client / Server model.
* One of the most important advantages of RMI is that it provides a callbacks mechanism, which allows the server to invoke methods on the client.

#### Rmi architecture

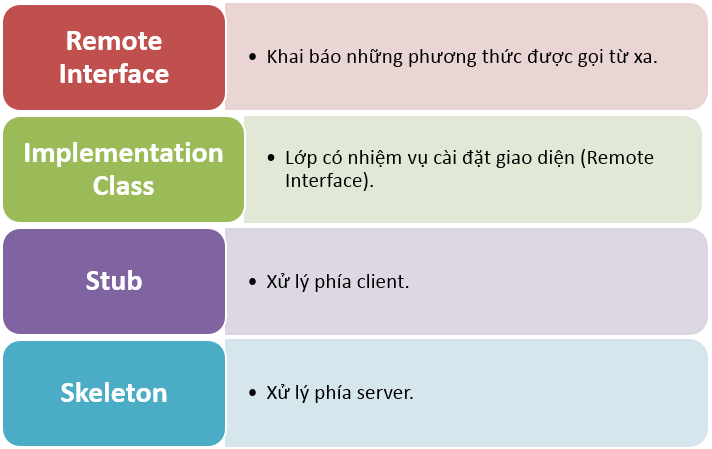


Fig 1 rmi architecture

Remote inteface: An interface that declares all the functions that the Server provides. The client can call this function as a regular call

Note: All functions in this interface throw an exception or RemoteException

Impements class: The class is based on the Remote inteface, performing the specified tasks. Usually located on the server

Stub and skeleton: Classes on the server and client for communication. automatically generated by java

#### In and out with Rmi

Another point to note is that when using rmi, the data transmitted is not changed or can be understood as programming with the transmission of value. Therefore all returned results are in the return function. Therefore it is inevitable that the object returned will have a complex structure

In rmi there is support for transferring objects, but this is a lot of restrictions such as: some objects are not implement Serializable class ,It can not be transmitted. Because Rmi will convert the object into an array of bytes to be transmitted and the sum. On the other hand, only those objects that have this class installed can send it

This makes it quite difficult to build software because there are times when it is necessary to send some objects in other libraries but the authors of those libraries often do not implement the Serializable class.

The explanation for this case is that most libraries operate on a single machine environment. And installing the Serializable interface will cause unnecessary difficulties because the internal properties must also be Serializable.

One solution is that we can download the libraries and fix the source code, but this is very time-consuming and many libraries are difficult to find the source code or the author of the library does not allow modification.

The solution is feasible in converting java objects to other structures such as XML, or Json, These two structures are widely applied. Having a clear structure suitable for object-oriented style

### Compare XML and Json

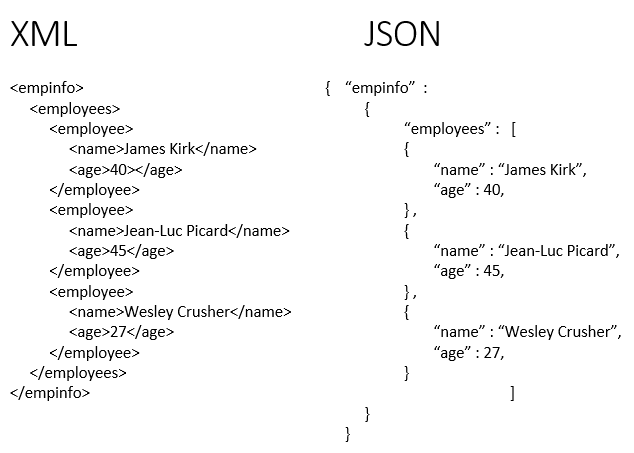


Fig 2 Json and Xml

It can be seen immediately that json has a rather tidy structure so to minimize the amount of information to be transmitted, I use json to implement the topic.

In addition, after doing some tests with the Gson library (the library that converts Object java to Json), in those tests, the string length is smaller than the conversion method bytes

Demo Code

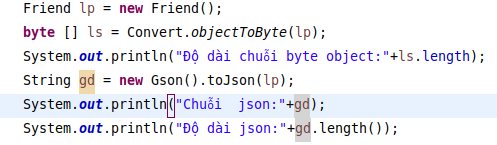


Fig 3 demo of json

Result

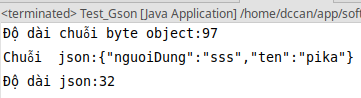


Fig 4 result test

In my project, it is often necessary to use objects with the structure similar to the above sample to transfer information, so it can be concluded that using json will reduce the amount of information to be transmitted, but it takes more. A little cost when converting from Json to java object. Fortunately, the computation time for this is very small, usually less than 1ms

More :

* And in the process of developing applications with java, one thing that we rarely pay attention to is that files compiled by the lower JDK can be run by higher JVMs but the opposite is not possible.
* The question is whether it is possible to communicate rmi between two different sets of jvm, according to my test results it is possible
* After testing with JDK 8,9,11, the drum system works smoothly

### What is SSL, what is an SSL certificate?

SSL stands for Secure Sockets Layer is a standard in security technology for establishing an encrypted connection between a web server and a browser. This connection ensures that the data sent between the web server and the browser is private and complete. SSL is an industry standard used on millions of websites to protect customers' online transactions. To create an SSL connection to a web server requires an SSL certificate.

SSL uses public encryption algorithms, when the web server is installed SSL, it will generate two encryption keys: a public key and a secret key. The encryption keys are generated when you complete a number of questions to identify your website as well as your company. The public key that is not necessarily secure should be included in the CSR (Certificate Signing Request) file. This file will be sent to the CA (Certification Authority) to authenticate and issue the SSL certificate containing the information when creating the CSR (website name, company name, address ...) and allowing the server Your is using SSL.

Typically, an SSL certificate will include some information such as domain name, company name, address, company, country, and information about the expiration date of the certificate is included. When a browser connects to a secure website, it will receive an SSL certificate, the browser will check the expiry date and who is the CA certificate issuer, if it fails to verify the browser will notify This website's user is not secured by SSL.

### Reflection technique

#### What is Java Reflection?

Java Refelection is a feature (called API or library also) in Java. Java Reflection allows access to the object's information (class names, fields, methods) and editing the object's fields (including private fields) during run time.

We can apply Java Reflection in cases where we do not know what the object is handled. (What is the class name, in which package, which fields, which method ...).

For example, I want to write a function to copy 2 objects that can be used for different types of objects. Then I need to know if the two objects are the same type, are there any fields, get and copy the value of each field.

In addition, for fields and methods with modifiers that are private, we cannot access them outside the class. In cases where it is imperative to call and access private fields and methods outside of that class, Reflection is a solution.

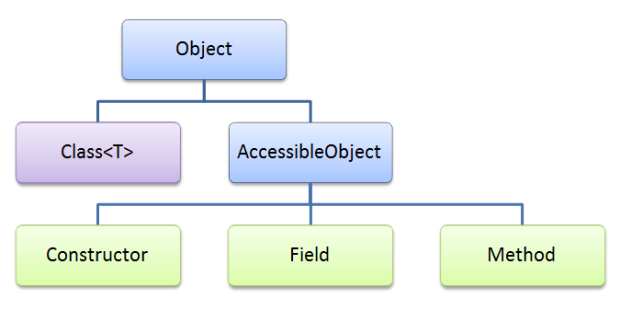


Fig 5 Java Reflection

Some frameworks use Java Reflection:

* Spring
* JUnit
* Tomcat
* Eclipse (used to autocomplete)
* ...

Limitations and disadvantages of Java Reflection

In case you already know the class structure, have access to fields and methods, you should not use Java Reflection for the following reasons:

Low performance: For example, it has to scan classpath to find class.

* Security Issues: Editing class / object during runtime can affect threads ... causing application to fail.
* Difficult to maintain: Reflection is quite confusing for newbies and not easy to debug, so it will be difficult to find errors. In addition, we also can not check some errors in the compile process (not found class, not found field ...)

Components in Java Reflection

* Corresponding to the components in a class, Java Reflection also provides the corresponding classes that we can handle:
* Class: Represents the class / interface to retrieve class information (class name, super class, class modifier, methods, fields ...)
* Constructor: Handles the constructor of the class
* Field: Handling the fields of the class (name, modifier of the field, getting values, setting values for objects ...)
* Method: Handling methods of class (listing methods, executing methods, etc.)

We can get the Class object through an object with the method getClass () or via package name + class name with the Class.forName method

To get the names of constructors, fields, and methods, we can use the method getConstructors (), getFields () or getMethods (). However, methods that do not allow to get constructors, methods, fields with modifier are private, so I will use getDeclaredConstructors (), getDeclaredFields (), getDeclaredMethods ().

The getModifier () method returns the modifier of the class, method, field, but in that form,we will write the getModifierName method to display it as String.

To get values or pass values to the object's fields you can use the method field.set () / field.get ()

Source stackjava.com

### Gson library

Gson is a Java library that can be used to convert Java objects into JSON strings. It can also be used to convert a JSON string into a corresponding Java object. Gson can work with all Java objects including objects that already exist which you don’t source code.

There are several other open source libraries that can convert Java objects to JSON. However, most of it requires you to put Annotations in your classes; or something you can't do if you don't have access to the source code. Most also do not fully support the use of Java generics.

[](https://gpcoder.com/3251-huong-dan-su-dung-thu-vien-gson/java-gson/)

Fig 6 Gson

Gson goal:

* Provide simple toJson () and fromJson () methods to convert Java objects to JSON and vice versa.
* Allow existing invariable objects to be converted to / from JSON.
* Extensive support of Java Generics.
* Allow custom for objects to convert to / from JSON.
* Support for complex objects (with hierarchies of multiple inheritance and widespread use of Generics data types)

User manual

Methods used:

* toJson (): method used to convert Java Object to Json string (this process is called Serialization). This method has one argument that is the object to convert to the Json string.
* fromJson (): method used to convert Json string to Java Object (this process is called Deserialization). This method has 2 arguments, the first argument is the json string, the second is the Java Object data type corresponding to the json string.

### Audio encoding

According to Itu (International Telecommunications Union), for voice coding, it is necessary to take samples at the frequency of 8000hz to represent the sample by 16bit. The Nyquist – Shannon theory requires sampling at 8000 Hz to recover

In summary, in 1s, it takes 16x8000 = 128kbit / s, equivalent to 16kB / s, not to mention the packet header with the current network speed, it is easy to talk to many users on the server side, but the server side will have problems. about bandwidth. So there is a need for information compression solutions to increase the ability to serve for user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Audio codecs | Tốc độ(KBit) | Độ phức tạp | Chất lượng | Độ trễ |
| G.711 PCM | 64 | Thấp | Rất tốt | Cực thấp |
| G.726 ADPCM | 40, 32, 24 | Thấp | Tốt (40k), Tồi(16k) | Rất thấp |
| G.729 CS-ACELP | 8 | Cao | Tốt | Thấp |
| G.729 A CA-ACELP | 8 | Vừa phải | Khá tốt | Thấp |
| G.723 MP-MLQ | 6.4, 5.3 | Vừa phải | Tốt(6.4k), Tồi(5.3k) | Cao |
| G.723.1 MP-MLQ | 6.4, 5.3 | Vừa phải | Tốt(6.4k), Tồi(5.3k) | Cao |
| G.728 LD-CELP | 16 | Rất cao | Tốt | Thấp |

<https://phuongot.wordpress.com/>

After studying, I decided to choose g729 for the following reasons

* Still widely applied, the quality of the output sound is still audible, is inferior to the g711 but not significantly
* Low latency
* High compression ratio, 8 times g711
* Free open source library

### Handling jitter

Jitter and latency are properties attributed to the flow in the application layer. Jitter and latency are used as metrics to measure network performance. The main difference between jitter and latency lies in their definition in that latency is nothing but network delay while jitter is a delay change.

The increase in latency and jitter have a negative effect on network performance, so it is necessary to monitor it periodically. This increase in latency and jitter occurs when the speed of the two devices do not match, congestion causes buffer overflows, traffic disruption.

Comparison chart

|  |  |  |
| --- | --- | --- |
| The basis for comparing | Jitter | Latency |
| Basic | The difference in delay between two consecutive packages | delay in package delivery on the internet |
| Reason | Cause of Network congestion | Transmission latency, serialization, data protocol, switching, routing, packet buffering |
| Precautions | Precautions Using timestamps | Many connect to the internet. |

#### Definition of Jitter

Jitter is the difference between the latency of IP packets. In other words, when the latency of the lag of the variance across the network, it causes jitter phenomenon. Can be explained by an example, suppose four packets are sent at times 0, 1, 2, 3 and received at 10, 11, 12, 13, the delay between packets is the same in all packages is 10 time units. In the other case, if these packages come 11, 13, 11 and 18, the generated delay is 11, 12, 9, 15 will be different from the above case.

The first form of delay will not affect applications like audio and video, because all packages have the same latency. However, in the second case, different lags for packets are not accepted and it also leads to the appearance of unordered packages. A high jitter indicates that the difference between lags is huge while a low jitter means that the variation is small.

#### Definition of latency

Latency is the time it takes a packet to reach its destination from the source. In network connection terms, the time spent processing a user-requested network access request and receiving a response of the request to the user. Broadly speaking, latency is the time elapsed between the execution of two events.

Latency is simply the time it takes to process messages at both the source and destination end and the latency generated in the network. There are two ways to measure network latency, the first is called unidirectional delay in which the time elapsed in the source and destination packets is measured. While in other categories, the one-way delay from node A to node B is aggregated with the one-way delay from node B back to node A and is called a round trip.

The main difference between jitter and latency

Latency is the time that a data packet requires to reach its destination from the source. In contrast, jitter refers to the variation of the delay generated by packet transmission.

Network congestion can cause jitter while latency can be generated through propagation, conversion, routing and buffer delays.

The jitter can be prevented by using timestamp. Conversely, latency can be reduced by using multiple connections to the internet.

Affect voice quality

In order to explore an information environment in practice, I have built a model as follows

Source

* Recording device
* System of sending packets with the time between each packet is 10ms
* Delay generator
* Destination
* The program creates sound in computer using java

Due to use in LAN, the ideal environment can be considered. Network congestion, packet loss does not occur

In case of fixed transmission time

Characteristics: distance between each receiving packet on the same recipient

With arbitrary latency. The sound obtained is not affected

With non-fixed transmission time (jitter)

Characteristics: distance between each receiving packet on the receiver is different

The sound produced will have distortion phenomenon if the time between receiving packets> 10ms

As for the period of receiving packets <10 ms, Quality is not affected

Conclusion: packet transmission time greatly affects the transmission quality, therefore, measures need to be taken to minimize the effects of jitter.

#### Reduce the effect of jitter

Jitter is inevitable when transmitting on the network, so we can only find ways to reduce the influence of jitter to the user's listening perception.

The method I use in this topic is to create a buffer that contains dialogues

The default number of containers is 10. This means that after at least 10 packets begin to make sound and continuously receive and broadcast, if saving 10 packets, the delay compared to continuous transmission of 0.1s can be accepted.

The use of this buffer has the benefit that within 10 ms without the incoming packet, the program still has data to broadcast while waiting for the new packet to arrive.

In case there is a packet that comes first to last, using the buffer greatly supports the re-arrangement of the packet so that the sound is seamless.

This can help conceal the jiiter better during conversations

### Mesh Topology

Mesh Topology is also known as a mesh network. Grid-based products are commonly used in networks that play an important role and cannot be deactivated. Typically, the network of an atomic power plant or the network of security and defense. For a grid, each computer device will be connected to all the other computers. It is also the familiar structure of the Internet.



Fig 7 Mesh Topology

Design

## Use-case designs

### Overview of functions

#### Main function for users

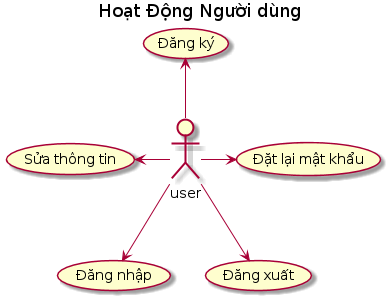


Fig 8 main funtion for user 1

This is the basic operation of users manipulating the system, including actions corresponding to the use cases

* Sign up: create a new account
* Login: Connect to the system
* Log out: log out of the account
* Edit information: correct account information
* Reset password: change password in case of user

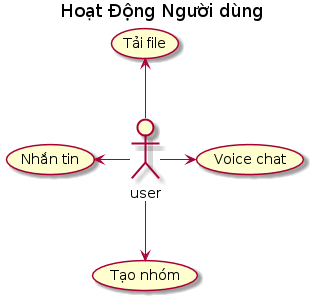


Fig 9 main funtion for user 2

These are activities related to messaging, including activities

* Messaging: send a message to the chat group
* Create group: create a new chat group
* Voicechat: open a voice conversation
* Upload file: upload the file

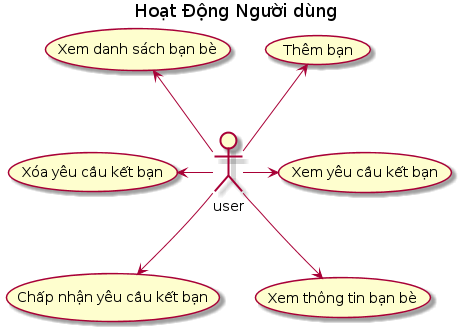


Fig 10 main funtion for user 3

The representative function for adding users to you includes activities that support users to add you in the most convenient way

#### Use cases for group administrators

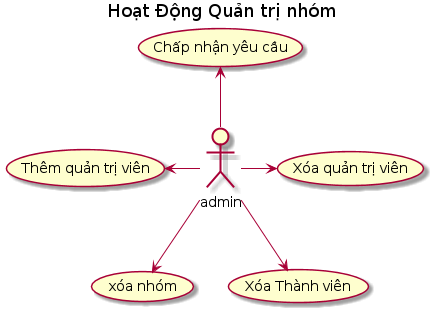


Fig 11 Use cases for group administrators

* Accept the request to join the group
* Add some members to the admin
* Removed the administrator position from some members
* Delete group
* Remove 1 member

#### Use non-functional case

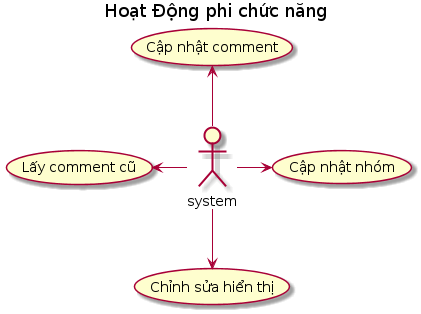


Fig 12 Use non-functional case

* Update comment: after 1 time update new cmt from other users
* Arranging the order in which order the group with the latest commet is placed first
* Update the list of participating groups
* Edit display language (not done yet)
* Get the old cmt in groups

### Usecase details

#### 1 Login

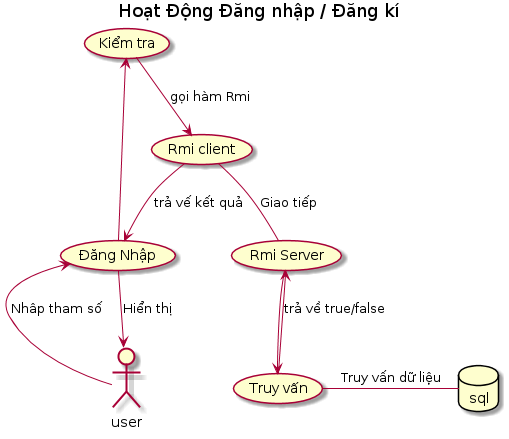


Fig 13 login

Activity details: users who open the software to perform the login / if there is no account will switch to the registration function, then the data entered will be checked for the necessary requirements, If false, must enter again

Then use the login / register method and then pass rmi to the server

The server will query and check with the database in the database management system and then return the results

Summary of functions in the interface Rmi provides

* Log in
* Registration

#### 2 Friends

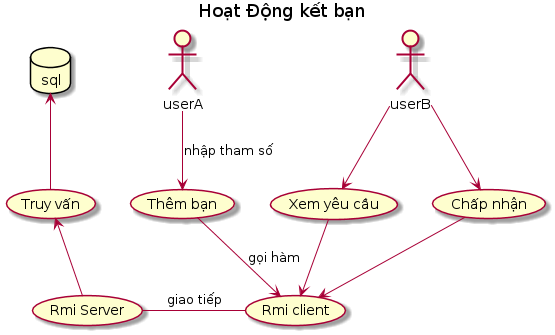


Fig 14 Firend

User A through the interface enters the name of the user who wants to add via the interface, then sends the parameter to the server using rmi. The server stores this information in the database

Users choose the function to add you. Software using rmi to retrieve friend request lists. then user b selects the person who wants to be friends or delete the friend request, then uses rmi to send to the server to handle

* Summary: the functions in the interface Rmi provides
* Add friend
* Accept the friend request
* Delete request
* See friend request

#### 3 Information editing function

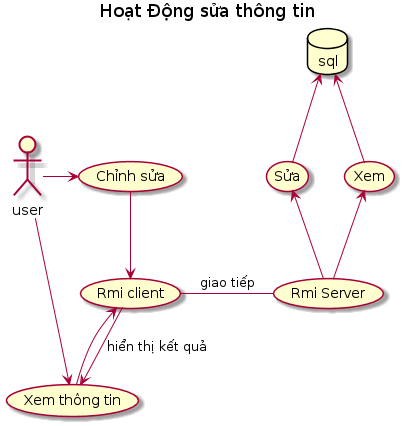


Fig 15 Information editing

Users select the function of viewing system information to get information from the server to display via rmi

From the user interface, you can edit the desired information and then update it from the rmi to the server

Summary: the functions in the interface Rmi provides

* Watch information
* Edit the display name
* Edit email

#### 4 Messaging function

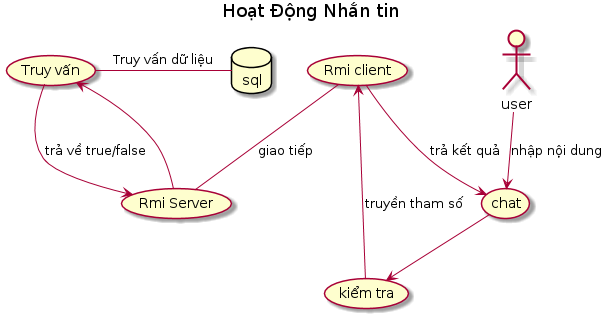


Fig 16 Messaging

Users enter the message into the interface of the software and then press enter to send. The system will check the content of the message, remove the extra characters and then send it to the server via rmi

At the server, the message will be timestamped to facilitate the identification of old and new messages, and then stored in the database.

Summary: the functions in the interface Rmi provides

* Message

#### 5 Join the group

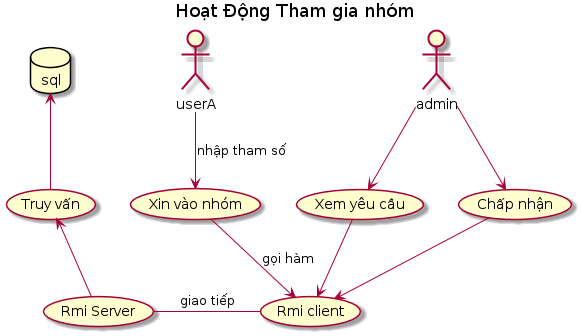


Fig 17 Join the group

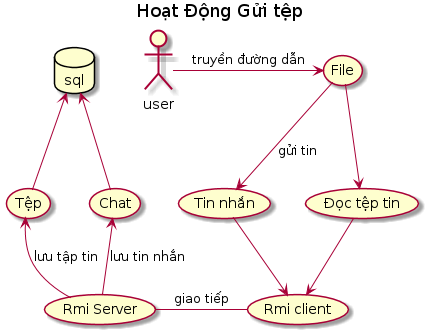
How it works is no different from adding a friend

Note here that in a group only the group administrator can view requests and add members to the group

Summary: the functions in the interface Rmi provides

* Request to join the group

#### 6 File sending activity in groups



If the user sends the file to a chat group, the software will read the contents of the file into a byte array, then send the file name as a message simultaneously. Upload the data of that file to the server. In the case of an empty file, just send a message or send nothing

On the server, the system will generate the id associated with each uploaded file and then upload it to csdl, thereby enabling us to query that file when needed.

Summary: the functions in the interface Rmi provides

* Upfile

#### 7 File download

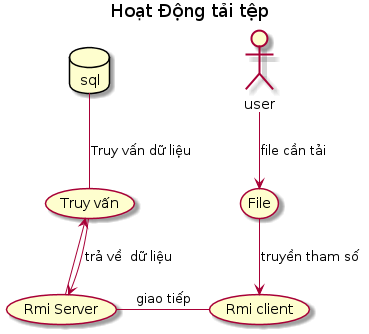


Fig 18 download

Operation description: user through the software to select the file to download then display the path for the user to choose where to store the file

Next, through rmi to get data as an array of bytes, then the software will write the file memory.

Summary: the functions in the interface Rmi provides

* Download the file

#### 8 Group exit function

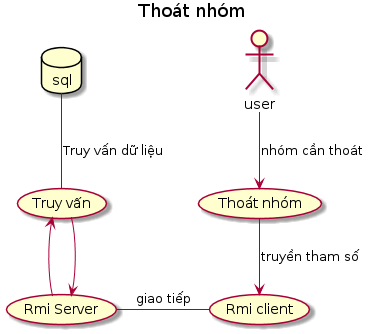


Fig 19 Group exit

The user enters the group id to exit, and rmi sends the request to the server. After the server will delete the user from the list of group members, if the process fails, the server will send an error message back to the user via return and rmi methods.

Summary: the functions in the interface Rmi provides

* Exit group

#### 9 See the friends list

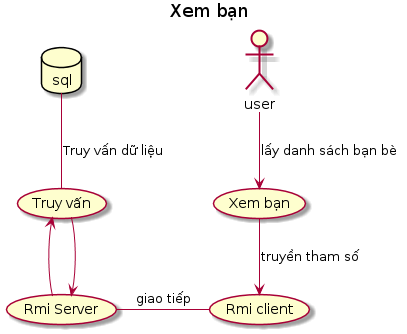


Fig 20 friends list

Users enable this function on the software. rmi will forward the message to the server. The server takes the information in the database and then converts it into an object and then returns it

Summary: the functions in the interface Rmi provides

* See the friends list

#### 10 Delete friends

This function comes with the view list of friends, when the user selects from the list of friends. The software will save the user to a list, then send it to the server.

On the server will delete the corresponding fields

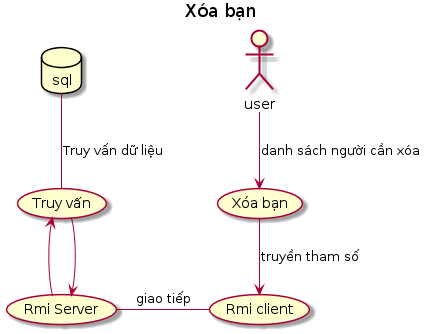


Fig 21 Delete friends

Summary: the functions in the interface Rmi provides

* Delete friend

#### 11 create groups

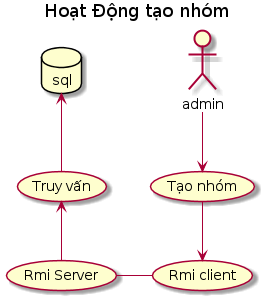


Fig 22 create groups

When the user selects this function from the interface, the user will enter the group name into the software to post via rmi

Summary: the functions in the interface Rmi provides

* Create a group

#### 12 voice chat

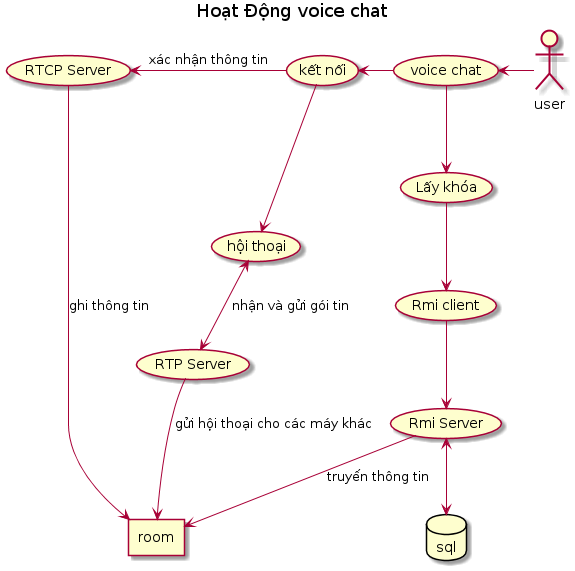


Fig 23 voice chat

Details will be in the following sections

Summary: the functions in the interface Rmi provides

* Get the conversation key

#### 13 functions to reset the password

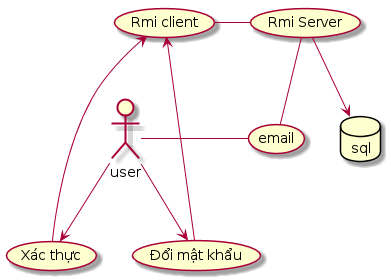


Fig 24 reset the password

Users will request this feature when they have forgotten their password. The server will send a verification code to the user email. The user re-sends this code and the new password to the server for testing. If so, change the password again and stop serving the previous login

Summary: the functions in the interface Rmi provides

* Password change request
* Validation change

#### 14 logout

When calling this function, the system will remove the corresponding token sent, which means that all activities related to this token are no longer served.

Summary: the functions in the interface Rmi provides

1. Log out

### Functions for group administrators

#### 1 Accept the request to join the group

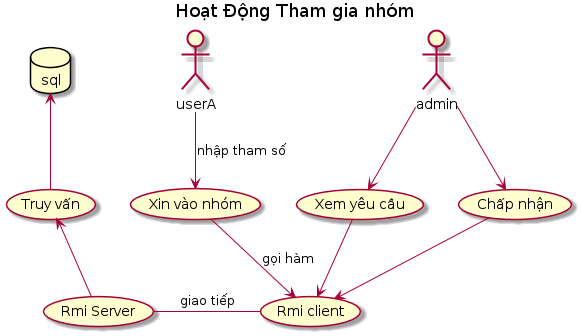


Fig 25 Admin 1

Like adding friends, administrators need to see who requests to join the group to decide who can join, then choose to add or remove those requests.

The request view function via rmi will retrieve information from the server to display a list of requesters

Then through the administrator's decision to allow or delete the request and then through rmi to execute

Summary: the functions in the interface Rmi provides

* View requests to the group
* Accept the request to join the group
* Delete request to group

#### Add an administrator

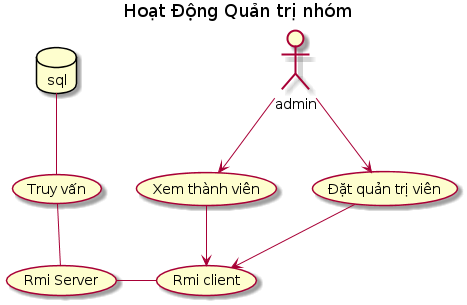


Fig 26 Admin 2

The way it works is similar to accepting a request to join a group

Function: View members ie retrieve a list of members without administrator rights to display

Summary: the functions in the interface Rmi provides

* View normal members
* Set administrator

#### Delete administrator

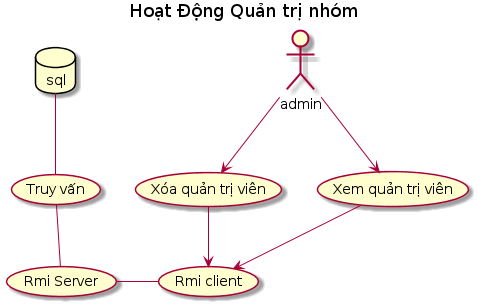


Fig 27 Admin 3

The way it works is similar to the above function

Admin view means getting a list of members with administrator rights to display

Summary: the functions in the interface Rmi provides

* See administrator
* Delete administrator

#### Delete group

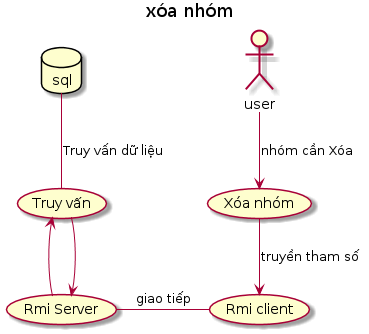


Fig 28 Admin 4

When the admin requests this function, the software will display a frame for the user to confirm and then call the methods in rmi

If this operation fails, a notification will be sent from the server

Summary: the functions in the interface Rmi provides

* Delete members

### The non-functional

#### Update new message

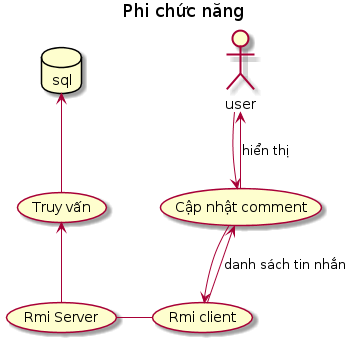


Fig 29 new message

After a period of time the software will have to automatically download new messages to be displayed to the user

Through rmi the system retrieves the list of new messages and displays them to the user

Summary: the functions in the interface Rmi provides

* Get the new message

#### Get old messages

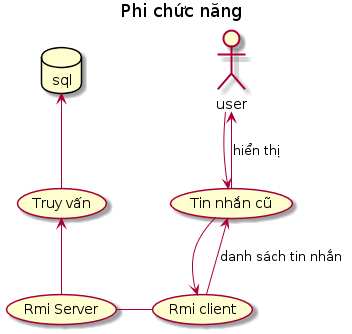


Fig 30 old messages

When users drag the scroll to the top, the system will automatically activate the function to retrieve old messages

Through rmi, the system takes a list of old messages and displays them to the user at the end of the list

Summary: the functions in the interface Rmi provides

* Get old messages

#### Get the list of groups

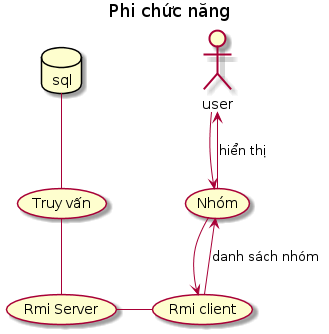


Fig 31list of groups

After a period of time the software will have to automatically update the list of groups to display to people

Summary: the functions in the interface Rmi provides

* Get the list of groups

## Design

### Design database for group chat

#### Basic

In order to create a chat group, according to my surveys on similar products like messenger, Discord, ... it needs a minimum of 3 tables to store activities.

* 1 table to store user information
* 1 table to store group member information
* 1 table to save messages in groups

**Member Information Table**

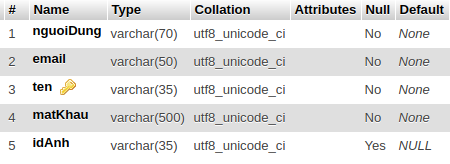


Fig 32 Member Information Table

Explain the fields

* nguoiDung: display name then information
* Email: to send some information, used for such as account recovery
* IdAnh: contains the code of the image file
* Ten: username - primary key
* MatKhau: Password has been hashed with Bcrypt

For the sources I refer to, there are many more columns for different purposes, but for the scope of this topic, I only use this information.

**Group information table**

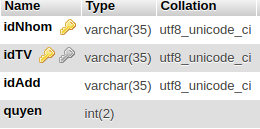


Fig 33 Group information table

* Quyen: delegated information for member 1 is admin, 0 is member
* idAdd: save information added to the group, use when trying to track information
* idTv: group member
* id: Group name

**Message table**

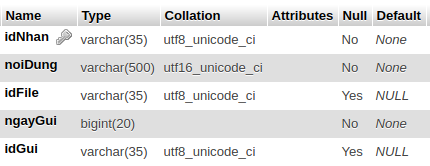


Fig 34 Message table

* idNhan: id of the group, with foreign key with idNhom (in the "nhom" table will introduce later)
* NoiDung: message content
* IdFile: id file sent to download if a member submits 1 file
* ngayGui: the time to send the file corresponds to the long type in java

Design explanation

A current messaging system requires a login, so a table containing account information is needed

Because information such as the display name, username, avatar image are made public, it is not necessary to encrypt it, and the password must be hashed with the hash function to protect in case of information leak.

Each message will be equivalent to 1 record in the message board, but if the text support is not enough for the needs to use. So I split the message into 2 types of text and multimedia

As for the text that will display the content in the chat box. As for multimedia, I will design the program to display a thumbnail image if the file is an image. Otherwise, the program will display the filename instead. There is support for downloading

However, to reduce processing, the software only allows sending files smaller than 16MB

The group membership table has a "Quyen" field for the purpose of controlling the addition and removal of group members. This is drawn from the software zoom, students can freely add members to cause vandalism

#### The tables cater for advanced functionality

The purpose of these tables is to serve the purpose of sending additional files, requesting groups, etc. These tables help to expand the functionality of the system, increasing safety and ease of use. Use

**File Table**

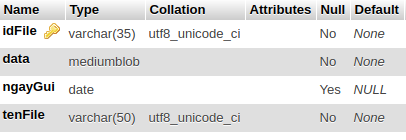


Fig 35 File Table

* Serve for the purpose of storing files
* With the primary key is idFile
* Data: is the field containing data
* ngayGui: Send time
* tenFile: the name of the file

**Request Table**

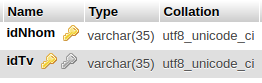


Fig 36 Request Table

* Contains requestor information into groups
* Groups are required

**Friend table**

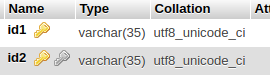


Fig 37 Friend table

There is a design similar to the required table, but because of different uses but not combined

In the message board there is also the file id field, but I do not combine it with the message board because of the increased scalability for the application. For example, adding personal photos to users also requires file storage, ..

The purpose of the request table is to contain requests into the group. This table was created to control the addition of members when an invitation code is available. It's a bit uncomfortable for groups that are highly open, but in terms of safety. as well as talking secretly with the people you want, this is necessary

Because the goal is towards group conversations, making friends is just about creating a group and making it easier to create groups and add people. However, two people can also talk privately by creating a new group, this will be fully supported

### Design Process of conversation

**Security**

To ensure confidentiality of the conversation, the security feature is indispensable so we need to have a security mechanism that encrypts packets

However, some of the issues raised here are the length of the message after encryption and the decryption time, such as using Des encryption algorithm with the standard key length of 64 bits = 8 bytes, 10 bytes. The input gets 16 bytes of output, which is an increase of 60%. The public key encryption algorithm has the same problem, not to mention the longer calculation time

The symmetric encryption algorithms such as symmetric key, advantageous permutation code are compact, but low security easy to use a computer to find the key

Chain Block cipher (CBC) are also difficult to apply in real-time communication applications due to two main reasons. 1 is that during UDP transmission it is inevitable that packet loss should be impossible to decode at the source. 2 is that the packets are constantly being sent to encrypt the blocks, which will make it more difficult to build the program.

Choosing Public Encryption is a bad option. Not to mention the cost of calculating the solution, choosing a key is a very complicated job. With multi-member conversations, exchanging each key pair for each member is a very complicated issue because when adding 1 person, it is necessary to transfer the key to all members once more. In the case of members sharing a pair of keys, while reducing the transfer of keys, many other problems arise. If all use a fixed key pair, this is almost not effective because now the reverse code to read the current source code is very simple. If creating for each group, with the user can create and cancel groups very comfortably, the job of creating a continuous key like that will be a huge burden on the system.

With 64 bit keys, the security feature of the DES algorithm is not very good, increasing the key size to 128, 256 ... will make the calculation time longer, the longer the code will be bigger. affect the transmission of the system

So I decided to use Vigenere encryption to do it, with a size of 10 bytes there would be a total of 256 ^ 10 keys. ie greater than 10 ^ 24. In addition, this is the result after the audio has passed through the g729 compressor so it will be more difficult to detect codes because old ways of decoding will be difficult to apply.

So I decided to use symmetric encryption with the key changes every time to enhance security

#### Get the key

When the user requests this function, the program will automatically check the list of existing chat groups. If no group will automatically create a new group chat object and create a group key. token for user confirmation

The system will then send both the key and token to the user via ssl

**Sign up**

The user will then send the registered Rtcp message to the server to confirm the information, which contains encrypted tokens. The server will record information about the client's IP address, connection port

The server will then send a confirmation message, containing the group number and member number

The posting process aims to register the ip address for the server to send later. This problem also stemmed from the Rmi model which does not support source address retrieval and cannot force users to open ports on their routers.

Because the design of my program is not allowed in a conversation with 2 devices on 1 account. This design makes it difficult to log in to multiple accounts, so it is only necessary to return the information to confirm whether the account can join the conversation.

**Rtcp system**

It works to keep the connection to the user in silent mode because the system will automatically remove participants in the conference if the person does not send messages to save resources

Or delete the user when he or she sends the packet to end the chat

To determine whether the user is still active or not, we must record the time to send the connection maintenance packets. This method, although it has to be repeated many times, the advantage is fast, no bandwidth is required when sending client test packets.

**Rtp system**

Receive and send messages to other members

Design clients / server in the conversation

If designed with a normal model, it requires extremely high server processing speed because this is a real-time application. Increasing the sending time between packets is also a way to reduce the pressure on the server, but it increases the latency for the user, the larger the packet size, the more lost it will cause distortion. The sound is clearer when used to listen

Therefore, I propose a model similar to a grid pattern in which each member will send information to the next members.

The difficulty of this design is that each member has dynamic ip so it is very difficult. Even if you can determine the ip address, it is not possible to connect to it without Nat port.

On the other hand, if you send a packet out, the router will automatically route for us to connect to the internet, so that we can get the port and address on the router corresponding to the packet. is that after a period of inactivity, the data table that is not used in nat will be deleted so it needs to send packets periodically to stay connected.

The downside of this design is that the client will have to send more packets than usual, but the bandwidth used on the server will decrease sharply, and the packet will not need to go through the server, reducing latency.

**Work**

The client sends the registration packet to the server, the server will store information about the ip address and connection port

The client will get information from the chat group participants to send information

After a period of time, a packet will be sent back to the server to update the connection information

### Rmi interface design

The goal of this section is to set the input parameters and the return results of the methods

Following the above sections, we have built up lists of important functions to help the system operate

The basic functions

* Log in
* Registration
* Validate registration
* Add friend
* Accept the friend request
* Delete request
* See friend request
* Watch information
* Edit the display name
* Edit email
* Message
* Request to join the group
* Exit group
* Create a group
* Upfile
* Download the file
* Delete group
* See the friends list
* Delete you
* Get the conversation key
* View requests to the group
* Accept the request to join the group
* Delete request to group
* View normal members
* Set administrator
* See administrator
* Delete administrator
* See members
* Delete members
* Get the new message
* Get old messages
* Get the list of groups
* Log out
* Change Password
* Password change authentication

**Login function**

As with any other software, a login is required to use. Usually we need a username and password to connect to the system.

To distinguish each user we can use the login name to distinguish but limit the ability of users to log on on multiple devices. Therefore the solution is to use a string of tokens (One time password) to identify the user

* Input the password string, username
* The result returns the token chain

**Registration function**

To register a new account, you need to enter your username, password, email address, display name, username, password for the login function, email used to recover accounts, and display name. is the name that the user desires to display in the conversation, and can vary. Because the username is unique

* Input password string, username, email address, display name
* Results returned: empty

**Registration authentication function**

It has the function of confirming the user's registration request, the server will send an email to the user to verify that the email is still used or not, the user needs to enter the OTP code to complete the registration.

* Input:otp string
* Output: true / false

**Add friend**

If you want to add a friend, you need to know the exact username of that friend to add it, followed by a token code to determine who is requesting the function.

If the token code is empty or does not exist then return false

* Input: token chain, the name you want to add
* The result returns” true / false

**See friend request**

Input is a token code to identify the user, without the need for additional information as this is an operation involving the user itself.

The output is a list of usernames in the form of a string list (List <String>) that can be sent via rmi

* Input: token chain
* Where to: output list

**Accept the friend request**

To make friends, you need a list of accepted people

Input is a list of people who can accept friends (List <String> can be sent via rmi) This list contains username because the username is unique to identify the data to be linked in CSDL.

Token to identify user

* Input token chain, list
* Where out: true / false

**Delete friend request**

To remove friend requests, a list of unacceptable people is required

Token to identify user

* Input: token chain, list
* Output: true / false

**Watch information**

This function to display information of yourself or a team member

To retrieve information we need to pass the username of the object

Transfer additional tokens to identify who is using, avoiding the case of people without an account can use this function

If the token code is empty or does not exist then return false

The output is the friend object introduced in the other section, which contains information about the avatar id, display name, email. Data is converted into json string

* Input token chain, username
* Json string output

**Edit the display name**

This function is used to change the display name of the user when needed

The information to change is the display name, so the input needs a display name, and the token string to identify the object to change

* Input token chain, name
* Where to output: true / false result

**Edit email**

This function is used to change the user's email when needed

The information that needs to be changed is email so the input needs a new email, and the token chain to identify the object to change

* Input token chain, email

**Message**

Used when texting in a group that the user joins. That is, the group id is needed to know which group the message is, the token to know who the sender is, the content of the message.

* Input: token, group id, message content
* Output: no

**Request to join the group**

Used when the user requests to a group. so you need to provide information about the group you want to enter, who the requester is.

In case the token does not exist in the system, the request cannot be processed

* Input: token, group id
* Output: true / false

**Exit group**

Use when the user has requested to log out of the group

So the information you need to know is which group it is, who wants to leave the group. In case the token does not exist in the system, the request cannot be processed

* Input: token, group id
* Output: true / falseWhere to output: true / false result

**Create a group**

To create a group, we need information that is the name of the group, who created it, who the original members were (refer to messenger).

* Input token, group name, member list
* Output: no

**Upfile**

Use when users want to upload files to the server. To upload a file, you need information about the file name, data of the file, the sender, and the group. Because rmi is used, it is necessary to read the data of the file and then convert it into a data type that can be sent in rmi. According to some found examples, the data type to use is a byte array.

* Input: token, group id, file name, data array
* Output: true / false

**Download the file**

* Input: token, id of file
* Output: byte array

**Delete group**

To delete a group, you need to know which group it is, followed by determining who the deletion is, is an administrator or not

* Input: token, group id
* Output: true / false

**See the friends list**

Because it is looking at your friends list, the input data only needs the token chain to identify

The output needs a list of friends' names, can add some information such as email addresses, avatars, so it will use the array of objects defined before. Due to the problem of transferring data in rmi, the result will be converted into json string

* Input: token chain
* Output: json string

**Delete friend**

Use this function when you want to remove unwanted friends. For convenience, it will support deletion at the same time so many people should input the linked list of friends' names (in Java, the list type will not have to care about the size of the array, this is convenient because we do not know. before you want to delete how many friends). The server reported success or failure

* Input: token chain, list you want to delete
* Output: boolean

**View requests to the group**

Use this function when an admin wants to review requests so the token chain is needed to identify the user or group that admin is managing.

Because the display name may overlap when displayed, the login name will be returned

* Input: token chain, group id
* Output: list of names

**Accept the request to join the group**

Use this function when you want to allow other people to join the group. For convenience, it will support the acceptance of many people at the same time, so the input is a linked list of names (in Java, the List type will not have to care about the size of the range, this is convenient because we do not know. before you want to delete how many friends). The server reported success or failure

* Input: token chain, acceptance list
* Output: boolean

**Delete request to group**

Use this function when you want to delete a request that does not meet the conditions. For convenience, it will support deletion at the same time so many people should input the linked list of their names (in Java, the List type will not have to care about the size of the range, this is convenient because we do not know in advance how many friends do you want to delete). The server reported success or failure

* Input: token chain, wish list, group id
* Output: boolean

**View normal members**

This function is used when adding other users to the board. Selected members are non-administrators

The result returned is

* Input: token chain, group id
* Output: member list

**Set administrator**

This function is used when you want to add other people to the administrator, for convenience, it will provide the ability to allow multiple people at the same time, so the input is a list of servers reporting success or failure.

* Input: token chain, new administrator list, group id
* Output: boolean

**See administrator**

This function is used when you want to remove the administrator rights for some members. It is used to display the administrators in the group. The members selected here are the administrators

* Input: token chain, group id
* Output: administrator list

**Delete administrator**

This function is used when you want to delete some administrators, for convenience, it will provide the ability to allow multiple people at the same time, so the input is a list of servers reporting success or failure.

* Input: token chain, administrator list to delete, group id
* Output: boolean

**See members**

Applies when you want to view the information of team members, Member information can be stored as a "Friend" object, however, to facilitate the data transfer will be transferred to the json string.

* Input: group id, token
* Output: json string

**Delete members**

Input and output are similar to other delete functions, except here the input list is the list of members want to delete.

Only applicable for administrators

* Input: token chain, member list to delete, group id
* Output: boolean

**Get the new message**

This function fetches messages with a later delivery time than the latest messages stored in the chat group on the client

So the information you need to know is the time of the latest message, the group is chatting, token code to determine whether the user is in the system or not.

The output is the list of "comment" objects that are passed to the Json string

The delivery time is indicated in the long form to facilitate comparison and data transmission

* Input: group id string, token, send time
* Output:Json string

**Get old messages**

This function retrieves messages that have earlier sending times than the earliest messages stored in the chat group

So it is necessary to know the time of the latest message, the group is chatting, the token code to determine whether the user is in the system or not

The output is the list of "comment" objects that are passed to the Json string

The delivery time is indicated in the long form to facilitate comparison and data transmission

* Input: group id string, token, send time
* Json string output

**Get the list of groups**

Get a list of groups the user participates in

The information of a group includes the group id, the group name is saved in the group object, The groups are saved in a list and converted into gson string

* Input: token code
* Output: json string

**Get the conversation key**

Where is the function to get the information used to confirm when participating in a group conversation, including 1 chain is the key of the group, 1 field is used to verify the member. Cannot reuse token chain because token if using token then all other tokens must be checked every time someone requests this function, if detached, the spending check is at a group size, much smaller than the entire test.

* Input id group, token hilt
* Output: security key, chain of identification

**Log out**

The information needed to log out is the token field in use. Because through this token we can identify that we're using it

* Input, token chain
* Output: no

**Change password**

We need information about who wants to change the password, the token code will only be created if the user successfully logged in, but in cases where it is impossible to log in, it is not possible to create the token, so you need to enter the username.

* Input, login name string
* Output: no

**Verify password change**

* Input, verification code, new password
* Output: true / false

### Build server

In Rmi architecture, the server built must have enough functions in rmi in order for the system to work.

Except for the login and registration functions, the server's other functions must check if the included token exists in the system, this is to avoid tampering with other accounts to sabotage.

**Log in**

To check if an account exists in the system, we must get the information in the "thongtin" table to check the username and password. Note that the password is encrypted information to protect users.

**Registration**

The system stores temporary information in memory, then sends an email containing the otp code to the user's registered email

**Registration authentication function**

The system will check the received otp code, if valid then write the information into the database with the previously saved information and the password encrypted by the hash function.

**Add friend**

To request more friends, you need to add data to the request table with the friend requesting to be saved in the idtv column, the requested person is saved in the idNhom column.

**Accept the friend request**

When accepting friend requests, the system will delete the corresponding information in the request table with idnhom as the acceptor, idTv is the requestor.

Then will inform the friend column with id1 who accepts the request, id2 is the requestor

**Delete request**

When deleting a friend request, the system will delete the corresponding information in the request table with idnhom being the requested person, member id being the requesting person.

**See friend request**

When you want to see all friend requests with the user, the system just needs to get the information corresponding to the request table with the condition idNhom with the username

**Watch information**

When you want to see the information of the person corresponding to the name entered, we need the data of the display name, the email of the user. This information is taken in the "thongtin" table provided that the username is the username

**Edit the display name**

This function is to correct the desired user information, need to update the "thongtin" table data with the user, using the new display name and the condition that the username is taken from the token

**Edit email**

This function is to correct the desired user information, need to update the "thongtin" table data with the email by new email and condition that the username is taken from the token.

**Log out**

Delete the corresponding token in the system

**Change password**

Retrieve email in database, generate code and send to retrieved email

**Verify password change**

Check the code and update the password on the Database

**Message**

When adding a message to the database, the data will be saved in the message panel with idNhom is the group that is chatting, idgui is the sender of the message, noiDung is the message content, nowGui is the time when this function is used

**Request to join the group**

Need more friends, you need to add data to the table yeuCau, the requestor is stored in the idtv column, the group is required to save in the idNhom column

**Exit group**

When a member wants to exit the group, they need to delete the corresponding data in tvNhom table, provided that idTv is the person who wants to exit the group, idNhom is the group that wants to exit.

**Create a group**

When creating a group, the system will automatically generate a string used to identify a group called id, then save information to the group table, then will add the data of each member and tvNhom table respectively. , with id created, member is a listener with Quyen equal to 0 and adds creator with permission equal to 1

**Delete group**

This function is only used for members who have administrator rights, so you need to check the permissions in the group membership table.

Delete the information in the message boards of Nann, tvNhom, group with idNhom by group string in the input. The order of deletion, in turn, is as it is related to the foreign key in the table

**See the friends list**

Detailed information about each relationship you save in the BanBe table, because the way to save friends depends on who is adding you, the username can appear in both columns id1, id2

When searching use union to get the full results

**output**

Deleting you is deleting information from friends provided that id1 = person deleted, id2 = person deleted and vice versa. Performing in turn with the people in the input list

**View requests to the group**

This function is only used for members who have administrator rights, so you need to check the permissions in the group membership table.

If the request is in the group, the data is saved in the table yeuCau with the requestor saved in the idtv column, the group is required to save in the idNhom column. so if you want to retrieve requests to the group, you need to search with idNhom = the group that is managing

**Upfile**

The system will create a string used to identify the file to be uploaded, then do the same as when sending a message with the content of the file name, idfile is the generated id, followed by saving the file to the tepTin table.

**Download the file**

Based on the file id to get the corresponding data in the file table, convert it to carry bytes and return the result

**Delete group**

This function is only used for members who have administrator rights, so you need to check the permissions in the group membership table.

Delete the information in the message boards of Nann, tvNhom, group with idNhom by group string in the input. The order of deletion, in turn, is as it is related to the foreign key in the table

**See the friends list**

Detailed information about each relationship you save in the BanBe table, because the way to save friends depends on who is adding you, the username can appear in both columns id1, id2

When searching use union to get the full results

**Delete friend**

Deleting you is deleting information from friends provided that id1 = person deleted, id2 = person deleted and vice versa. Performing in turn with the people in the input list

**View requests to the group**

This function is only used for members who have administrator rights, so you need to check the permissions in the group membership table.

If the request is in the group, the data is saved in the table yeuCau with the requestor saved in the idtv column, the group is required to save in the idNhom column. so if you want to retrieve requests to the group, you need to search with idNhom = the group that is managing

**Accept the request to join the group**

This function is only used for members who have administrator rights, so you need to check the "Quyen" information in the group membership table.

When accepting the request, the system will delete the corresponding information in the request table with idnhom as the accepting group, member id is the requestor

Then add member data to tvNhom table with idAdd as the approver's name, "quyen" equals 0 representing normal member

**Delete request to group**

This function is only used for members with administrator rights, the system will delete the corresponding requests in the table with "idNhom" from the input data, idTv are the ones in the list.

**View normal members**

This function is only used for members who have administrator rights, the system will get the members of the group in tvNhom table provided that quyen = 0

**Set administrator**

This function is only used for members with administrator rights, the system will reset the "Quyen" value of the members of the group in the input list in the tvNhom table is 1

**See administrator**

This function is only used for members who have administrator rights, the system will take members of the group in tvNhom table provided that quyen = 1

**Delete administrator**

This function is only used for members with administrator rights, the system will reset the "quyen" value of the members of the group in the input list in the tvNhom table to 0.

**See members**

Function used with all members in the group, the system will retrieve all members of the group in tvNhom table

**Delete members**

This function is only used for members with administrator rights, the system will delete the information of the members of the group in the input list on tvNhom table.

**Get the new message**

The system will retrieve information of messages in the message board, which will be sent later than the input time

**Get old messages**

The system will retrieve the information of messages in the message board

**Get the list of groups**

To get the group list, we need to get the information in the tvNhom link with the group board, end. Therefore more information is needed from the Message panel. Note that because there are some newly created groups that do not have messages yet, when linking to message boards, you need to use the left (or right) link to ensure full data.

### Real time data transmission

#### Compare UDP and TCP

Same: all network protocols TCP / IP, function to connect the machines together and can send data to each other ....

Different

| TCP | UDP |
| --- | --- |
| Connection direction | Direction is not connected |
| High reliability | Low reliability |
| Send data stream as byte stream | Send to Datagram |
| Do not allow packet loss | Allow packet loss |
| Ensure data transmission | No Guarantee for data transmission |
| Has ordered packets | Unordered packets |
| The transmission speed is lower than UDP | High transmission speed |

#### Select the protocol for real-time application

Quote verbatim

Trong những ứng dụng truyền thông đa phương tiện, yêu cầu đảm bảo khắt khe về thời gian thực (không cho phép có thời gian trễ lớn, jitter). Việc các gói tin đến không liên tục, đều đặn làm cho chất lượng hình ảnh, hoặc âm thanh thu được thấp. Rất có thể gây ra vấp hình, méo tiếng. Để đáp ứng được những yêu cầu này, một giao thức thời gian thực cần có các yếu tố:

Hộ trợ việc định tuyến muticast: Với các ứng dụng tryền thông đa phương tiện đòi hỏi thời gian thực, có sự phân phối giống dữ liệu từ một nguồn tới nhiều đầu cuối nhận dữ liệu thì việc hỗ trợ multicast là rất cần thiết. Đây là một yêu cầu rất quan trọng. Khi đó, sẽ tồn tại 1 nguồn phát và rất nhiều nguồn thu, một máy chủ xuất luồng dữ liệu thời gian thực đến rất nhiều máy khách. Nếu ta sử dụng truyền unicast, tải trọng tác động lên máy chủ rất lớn. Trong khi đó, nếu mạng có hỗ trợ truyền multicast, ta chỉ việc xuất một luồng duy nhất từ máy chủ tới một địa chỉ multicast. Sau đó tại các nút mạng, luồng dữ liệu sẽ được nhân lên và chuyển tiếp tới những địa chỉ đích.

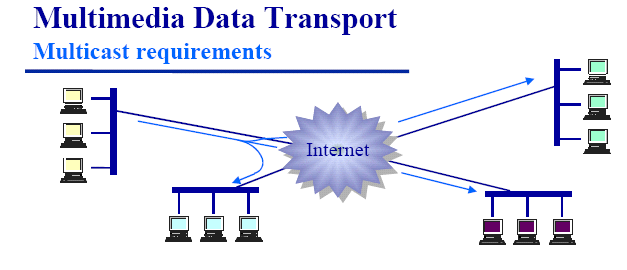


Fig 38 Mutil Media

Sử dụng Multicast trong truyền dữ liệu đa phương tiện.

Chấp nhận một số gói tin bị lỗi: Không thể đợi để truyền lại các gói, đoạn, gam dữ liệu bị thất lạc. Việc truyền lại các dữ liệu bị thất lạc hoặc bị lỗi sẽ chiếm khá nhiều thời gian. Nó sẽ làm tăng lượng tải trên đường truyền đồng thời kéo dài thời gian trễ của các gói tin.

Cần kết hợp với một thông số về thời gian (nhãn thời gian) kèm theo gói dữ liệu: Với các tín hiệu thời gian thực, đặc biệt là tín hiệu video, việc khôi phục đồng bộ tại phía thu là rất quan trọng, do đó đòi hỏi nhãn thời gian kèm theo để phục vụ cho việc tái tạo lại dữ liệu tại nơi nhận. Đặc biệt, khi tín hiệu video được mã hoá theo từng khung hình, mỗi khung hình được vận chuyển trong nhiều gói RTP. Khi đó nhãn thời gian sẽ giúp ta phân định từng nhóm gói tin tương ứng với một hình một cách dễ dàng.

Trong những giao thức ở lớp vận chuyển thì giao thức nào có thể đáp ứng được yêu cầu trên:

TCP:

Đây là một giao thức kiểu có liên kết (Connection – Oriented), tức là phải có giai đoạn thiết lập liên kết giữa một cặp thực thể TCP trước khi truyền dữ liệu. Trong khi truyền dữ liệu giao thức TCP phải đảm bảo các cơ chế xác nhận việc gởi dữ liệu, đảm bảo xắp xếp đúng thứ tự các gói tin tại bên nhận, phát lại các gói tin bị lỗi hoặc thất lạc. Do việc phải đảm bảo những cơ chế này gây lên thời gian trễ lớn, nên giao thức TCP không thể dùng được trong những ứng dụng thời gian thực.

Ngoài ra với tính chất vốn có của mình, TCP là giao thức được sử dụng để truyền dữ liệu theo kiểu điểm tới điểm, hay nói cách khác TCP chỉ được dùng cho truyền unicast, không thể sử dụng cho truyền multicast.

Với những đặc điểm trên, TCP không nên được sử dụng trong việc truyền dữ liệu mang tính thời gian thực.

UDP:

Đây là một giao thức kiểu không kết nối, được sử dụng trong một số yêu cầu ứng dụng thay thế cho TCP. Tương tự như giao thức IP, UDP không thực hiện các giai đoạn thiết lập và huỷ bỏ liên kết, không có các cơ chế báo nhận như trong TCP. UDP cung cấp các dịch vụ giao vận không đáng tin cậy. Dữ liệu có thể bị mất, bị lỗi hay bị truyền luẩn quẩn trên mạng mà không hề có thông báo lỗi đến nơi gửi hoặc nơi nhận. Do thực hiện ít chức năng hơn TCP nên UDP chạy nhanh hơn, nó thường được sử dụng trong các dịch vụ không đòi hỏi độ tin cậy cao. Ngoài ra, giao thức UDP còn có thể sử dụng cho truyền multicast.

Do vậy UDP có thể được sử dụng để truyền các dữ liệu thời gian thực.

Source “Nghiên cứu và ứng dụng Rtp”

An overview of Rtp is included in the appendix

**Packet size**

The standard rtp packet size is 16 bytes rtp header + 10 bytes of data per 1 packet is 26 bytes. The addition of Udp header and ipv4 header is 66 bytes. Information about the 2nd floor title in the osi model, I have not fully studied but estimated the total size sent is less than 100 bytes. At 1 packet / 10 ms, the bandwidth consumes about 10Kb / s. With the current network infrastructure, this speed is quite low

**Some changes on rtp protocol**

Rtp is a standard designed for a common data transfer environment, but with many topics there are many features that will not be used. If left unchanged, there will be a lot of unnecessary waste on this subject. Conversely, modifying the packet structure will lose the ability to interact with other systems. Or it could be said that it could not link to other systems

In order to facilitate the upgrade, the processing related to packet transmission will be designed into a separate packet to ensure the convenience of future upgrades if necessary.

**Change to rctp**

The goal of Rctp is to respond and evaluate the quality of the network from which the measures are taken. However, because this topic works with encrypted audio to minimize the amount of transmission needed. This means that in every case the packet size will not change. Elements related to network quality feedback ... will not be supported in this release.

Therefore I will simplify the control packet to set the highest performance

Join Package

This packet is used to register with the server that a user wants to join the group

Information fields in the group

* The type: 16 bit field contains the packet type, which can be extended later
* Field length: 16bit length of packet separately from group id and standard header
* Ntp field: 64bit time stamp
* Port field: port to receive client data
* Group field: group id code
* User: string field for member verification

Type: 1000

Bye packet

This packet is used when someone wants to log off the conversation

Information fields in the group

* The type field contains the packet type, which can be extended later 16 bits
* Field length: the length of the 16bit packet
* Ntp field: 64bit time stamp
* Group field: 32-bit group id
* User field: 32-bit member id

Type: 1111

**Live packet**

Use to stay connected when in, update information in the conversation

* The type field contains the packet type, which can be extended later 16 bits
* Field length: the length of the 16bit packet
* Ntp field: 64bit time stamp
* Group field: 32-bit group id
* User field: 32-bit member id

Type: 1001

**Rtp packet**

In the rtp packet, the sequense field is responsible for counting packets. However, since the emitting source is regular with increasing time, it is possible to use the last 32 bit timestamp field to take on the role of the sequense field, 32 bits is sufficient to represent the time corresponding to 50 days, so it can be used. optional

Currently only works with ITU-G729 audio that has been safely encrypted with cassava exchange key, so it can cut the crsc count, padding fields ...

Meaning of some schools

* SSRC: sync source corresponding to group id
* CSRC: corresponds to the membership code
* TimeStamp: Last 32bit of NTP

### Support classes

In the client server model, the communication between the two parties is indispensable. With most information being strings, organizing them into arrays will be convenient for transmission because of their compact size. However, during the implementation process, there were many times I encountered mistakes in position and meaning in those areas. Moreover, reuse or upgrade code will be difficult due to having to remember the location of each information. And above all, the error due to the wrong location that often happens causes the program to have the wrong result without the error message. Therefore, the debugging will be hard, taking a lot of time.

Because of that, I decided to use objects to store data. It will avoid the problem of memory location errors, Convenient for future upgrades. Although this will help the system consume more bandwidth to operate.

The classes

The Comment class with attributes

* idNhan: the recipient's code
* idGui: the sender's code
* noiDung: Message content
* idFile: code file sent

Group class with attributes

* idNhom: group code
* username: name

Class Friend

* nguoiDung: display name of the user
* Ten: the username of the user
* id: Image number of the avatar file
* Email: the email of the user

In the above classes, the Friend class is most often used for many purposes, containing friends list information, containing personal information, containing information of group members, and information of administrators. .. If you do not use an object then it is a whole problem because you have to remember the order of each one, because each function only requires a certain amount of information.

In the above cases the necessary information will be pinned to the object. The rest receive null values

For example: Instead of having to remember in style

Args [10] = .... // i = 10 is the username

Then we can write p.getTenNguoiDung (); More convenient in use, not having to remember spare part one

### Data transfer

**server side**

We can summarize the operation of the server as it receives data from the database management system and converts it to the object and sends it.

Executing the query is already in another section, so I don't repeat it

The result of the query is the ResutSet object through which we can get the information in the column we need by using that column name.

To convert data to objects it is required that the field names in the object must have the same name and the same type as the tables in the result.

Restrictions on naming can be eliminated by using notation in java, but because I found it unnecessary, it was not implemented.

In java, there are 2 ways to perform queries with Database: Jdbc and hibernate. However, hibernate has many advantages over JDBC. But because I have worked hard with JDBC, this project only uses Mysql. So I think the choice is that it is better with JDBC. I can use the Reflection technique to simulate the biggest advantage of Hibernate is that it automatically adds data to the object. Not to mention that using JDBC will apply the Connection pool more easily

The result is a class that can convert data flexibly. Just make sure the request is the same name as the fields in the database, and that none of the fields will have a null value

It will then be through the Gson library to convert the data into a Json sequence

**In Client side**

Just use the gson library to convert to the desired object

### User interface design

#### Basis

**Login**

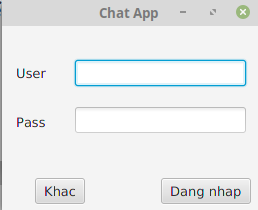


Fig 39 GUI Login

To enter the password, and username, press the Login button to complete the login process

Other buttons to cater for additional functions

**The support functions**

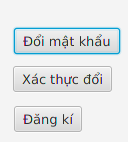


Fig 40 GUI suport

Provides features such as password change, Password change authentication, Account registration

**Sign up**

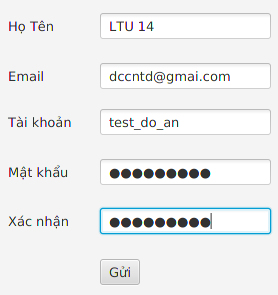


Fig 41 GUI Sign up

According to the existing rmi design, the interface should ensure to help users enter enough information on the name, email, display name and password fields to register information.

Note that you must enter the correct email for the system to send the token

**Validate registration**

After completing the registration, the system will send the token code to the email of the user just entered

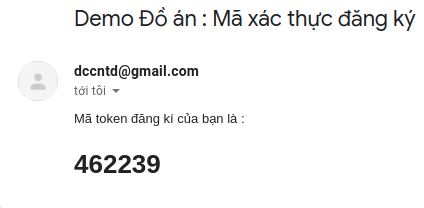


Fig 42 email 1

Then take this code to enter the fill of the window that appears to complete the registration

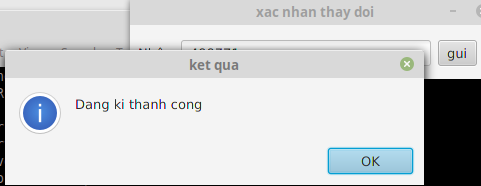


Fig 43 GUI xac nhan

**Change the password**

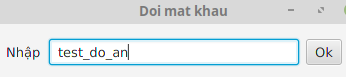


Fig 44 GUI Chang pass

The user will enter the account name you want to change into the software, then press ok

**Validation changes**

The system will send a code to previously registered email to change the password, the software does not support the password recovery function due to using the hash function to store

Result



Fig 45 email 2

The user then retype the password you want to change and the verification code sent via email

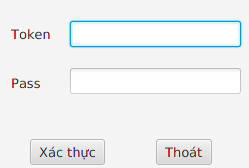


Fig 46 GUI pass

According to the rmi design, the information to be entered includes the token, and the code to be changed

**Watch information**



Fig 47 GUI info

Because there are no more functions, the part that will only display the most basic information of the user is the avatar image, username, display name.

**Edit the information**

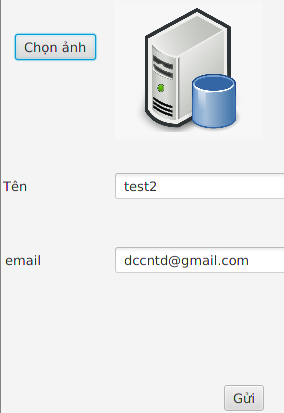


Fig 48 GUI Edit info

The system allows to correct 3 types of information: email, avatar, and display name, more specifically, When changing email, the system will send information to the old email to confirm avoiding arbitrary changes, breaking gangrene

**Add friend**

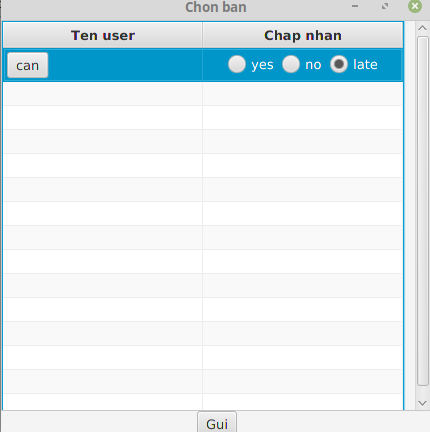


Fig 49 GUI add friend

The software will display a list of friend requests, through yes / no / late options to decide more or not.

There is also a function to view the information of the requestor for your convenience

**Delete friend**

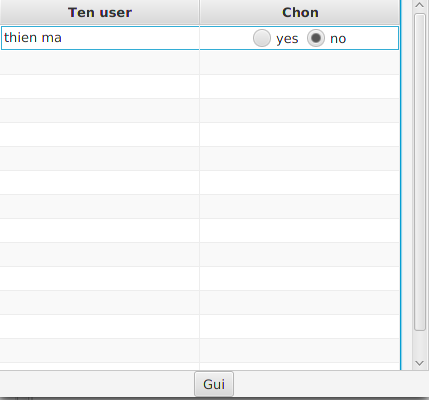


Fig 50 GUI delete friend

The software will display a list of friends so that users can choose to delete you or not

**Make friend**

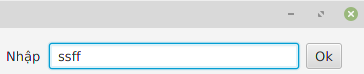


Fig 51 GUI Make friente

By entering the username, the software will send a friend request to that person

For convenience, the program will support searching for usernames

#### Group

**Team list**

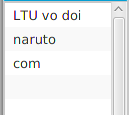


Fig 52 GUI team list

The list displays the group name that the user joins, sorted by the most recent message sent in the group

**Create a group**

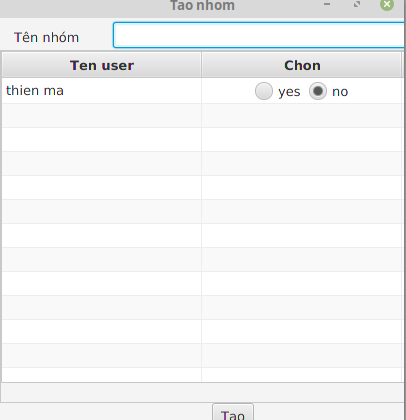


Fig 53 GUI create group

The group creation function will display the user's friends list to make a choice whether or not to include the user in the group by selecting yes / no,

Users enter the name of the group they want to create and then send it

**Message**



Fig 54 GUI message

The input field contains the message text, to send the message press the enter key to activate the send button

When the file button is pressed, it displays on the file menu to upload it to the server

**Team member**

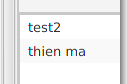


Fig 55 GUI Group member

Display the list of group members, we can view the information or make friends in the list

**Get the code to join the group**



Fig 56 GUI get Code

Display a frame to contain the invitation code, users can copy it and send it to others

**Join group**



Fig 57 GUI join group

The user will enter the invitation code into the box, press the ok button to request to the group

**Chat window**

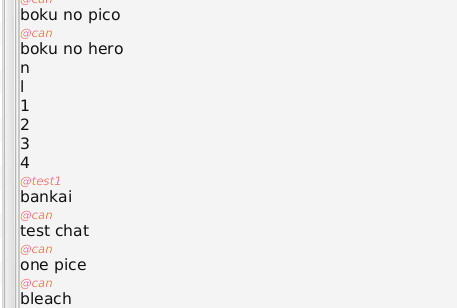


Fig 58 GUI chat

This window displays the messages of the group members, distinguishing between the message as a series of red, italic characters on each message.

#### Elevations for administrators

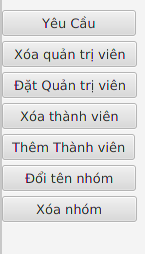


Fig 59 GUI Admin

**Approval function required**

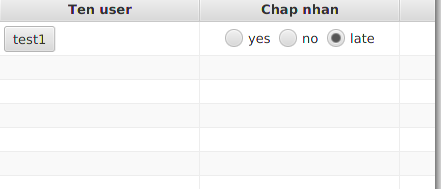


Fig 60 GUI Requets

Administrators can review requests for groups to others, yes is accepted, no is denied, late is for later review

**Set administrator**

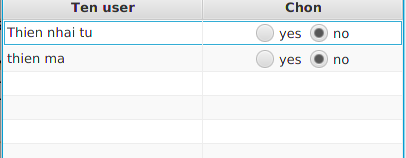


Fig 61 GUI set Admin

Administrators can choose from the list of regular members and then set administrative rights for them

**Delete administrator**

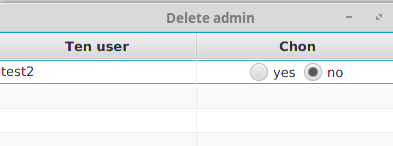


Fig 62 GUI delete admin

Actions similar to the above

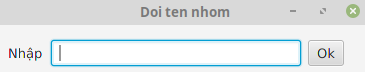
**Add group members**



Fig 63 GUI add member

The administrator can add other members from his friend list

**Rename the group**



Administrators can change the group name to their liking

**Delete group**

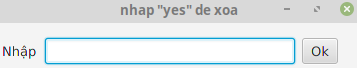


Fig 64 GUI delete group

The administrator can delete the group he / she is managing, however, to avoid possible mistaken pressures, you must enter "yes" to confirm that you want to delete the group.

### Restrictions exist

**Filter input audio**

In the process of recording the signal input containing environmental noise, the system affected the quality, but due to the knowledge of audio processing, it has not been processed. I also learned about some commonly applied filters such as Kalman, LMS ... but encountered many calculation time problems and also applied it to real-time environment.

Unlike the existing audio file filtering, the real-time environment has many limitations such as the low number of samples, only samples at the present time, which require fast computation. I've talked about the application, but for a beginner like me it's too hard to understand

**Java**

Because the goal is to build applications that run on multiple platforms, the java language is one of the top choices due to its popularity as well as in the classroom learning process, she is taught many subjects about java. However, because it is a language that runs on the platform, the operation of devices on the computer is still limited. The process of manipulating the sound of java only stops at inputting data. As for the broadcasting process, the contest has not been able to intervene

Currently, I cannot propose any solution to minimize jitter and synchronization between sources. Although it is possible to use buffer available on java, the results are not as expected

# CONCLUDE

## Result

In the process of researching and implementing the project under the guidance of Mr. Nguyen Tien Thanh, the program has completed and obtained the following results:

Based on research and research of Rtp protocol, we have initially built the Client-Server communication system.

For Client side:

* Building an interactive user interface
* Record, and compress the obtained audio data
* Send message via the system
* Revised rtp and rctp packet structure
* Play Audio

For server

* Build rmi functions
* Save information of client address table to serve transmission
* Automate the conversion of query results to objects
* Solve problems when transferring data by rmi

Through the process of implementing the project, I have learned how to design and build the client / server in the rmi model. accelerate software development speed. Next, we found a solution to solve the transmission problem via rmi, which facilitated the development of not worrying about data transmission errors anymore

## Development direction of the project

* Integrating new features
* Upgrade graphical interface
* Addressing outstanding issues is audio filtering and jitter
* Developed for mobile

# Appendix

## Structural types

### XML

#### What is Xml

XML stands for Extensible Markup Language, meaning extended markup language. This format was created by the global W3C web consortium with the purpose of developing and extending markup languages ​​other than .txt, ASCII format, etc.This way of storing XML files is quite simple, but it has Can describe many different data. Therefore, XML is relatively useful in sharing data between systems and programs. For example: XML file format can be a communication language that helps exchange data between ASP and PHP applications.

To that end, XML data specifications must follow a certain rule and syntax. Because most XML files are strictly compliant with compilation, however, the error rate during data manipulation and transfer is still about 5 - 7%. Although this number is not high, it is worth considering before using.

However, one can also be assured that the HTML hypertext markup language also uses XML syntax to create. It has element parts, non-flexible properties, so it is only effective in displaying data in the Browser browser only.

Summary

XML is the markup language used to describe data.

The tags in the XML file are not predefined. Therefore, the user must define himself in the process of creating the XML file.

#### The effect of XML files

The primary purpose of the XML format is to share data resources between platforms on different systems. In particular, networked systems, XML files will be shared more quickly. Therefore, XML is often used for exchanging and sharing between systems.

When there is data exchange between systems, those data will be organized as XML. Specific example: If a restaurant management system wants to retrieve tourist information from a travel agency system, the system needs to ensure the following operations are in place.

#### Systems need to have a consistent file structure

The tour operator needs to extract data from its system, encapsulating the data in an XML file according to the agreed structure.

The restaurant's software system will be based on conventions, conduct analysis, extract data from XML received from the travel agency's system to obtain the desired customer data source.

#### Characteristics of XML

Some salient features of XML file:

* XML is only used for structured data.
* XML has a similar structure to HTML files
* XML, although a text file format, is not readable.
* The XML file structure is usually very long
* XML file is known as a module
* XML is like a platform for semantic web and RDF.
* XML is considered as a bridge file between HTML and XHTML systems
* XML files are royalty-free, have an independent platform and are well supported for users.

**Advantages**

XML is data independent. This is also the biggest advantage of XML file. It is used to describe data in text form. Therefore, most normal software or programs can read it.

XML files can be easily read and parsed by data sources. Therefore, it is mainly used for the purpose of exchanging data between programs and systems with each other.

XML files are easily created with a few simple steps.

The XML file is used for Remote Procedure Calls for the purpose of serving website design services.

**Defect**

The error rate while using the XML file format is up to 5-7%. However, most errors when using XML files come from misconfigured users, so if you are really careful in your file settings, this error could be even lower.

#### The structure of XML

The structure of the XML file is shown as follows:

Inside:

* Content: is the attribute name, opening tag, content, the name of the tag will be determined by the user itself.
* Content: The main content of the card.
* Content: The closing tag, the name, the content of the closing and opening tags must match.

Source <https://monamedia.co/file-xml-la-gi/>

For example

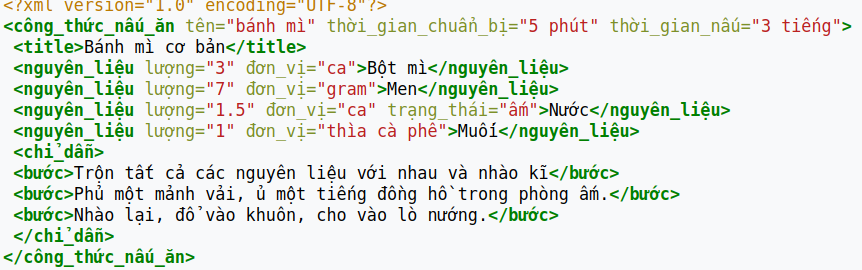


Fig 65 Appendix Xml

### JSON

#### What is JSON

JSON stands for JavaScript Object Notation, a type of data format that follows a certain rule that most programming languages can read today. JSON is an open standard for exchanging data on the web.

Define

JSON format uses key - value pairs for data usage. It supports data structures such as objects and arrays. For example, a file named topdev\_info.json with the content as below uses JSON format to store information:

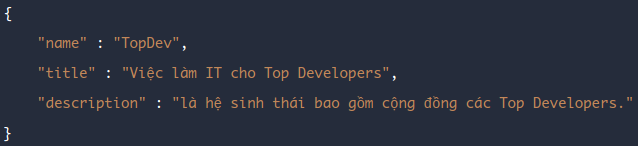


Fig 66 Appendix Json

We can see that the syntax of JSON has 2 parts: key and value:

* JSON string enclosed by curly braces {}
* JSON keys and values must be enclosed in double quotes {"}, if you put them in single quotes, this is not a standard JSON string. If the case in your value contains double quotes, then use (\) to precede it, for example \"what json is\" .

If there is a lot of data, use commas to separate.

JSON keys should contain unsigned letters or numbers, \_ and no spaces., The first character should not be numbers.

The json file can be saved with any extension, but it is usually saved under the .json or .js extension.

JSON was originally developed to cater to applications written in JavaScript. However, because JSON is a data format, it can be used in any language without restrictions.

Key values in JSON can be string (number), number (numner), empty (null), array (array), or object (object).

## Protocol

### Tranmission Control Protocol

TCP is a connection-oriented transport protocol, meaning that a connection with a remote site must be established before data transmission is made. The process of establishing a connection at TCP is called the threeway handshake.

Providing acknowledgment mechanism: When A sends data to B and B receives, it sends a packet to A to confirm receipt. If the confirmation message is not received, A will send it until B acknowledges it.

Provides a mechanism for sequential numbering (sequencing) for the data units to be transmitted, used to assemble the exact packets at the point of receipt and discard duplicate packets.

There are appropriate flow control mechanisms (flow control) to prevent clogging.

Support full-duplex mechanism (transmit and receive data at the same time)

Recover lost data on transmission line (A send B without seeing confirmation will send again).

#### TCP packet structure

Because it is a reliable protocol, TCP's packet structure is very complex

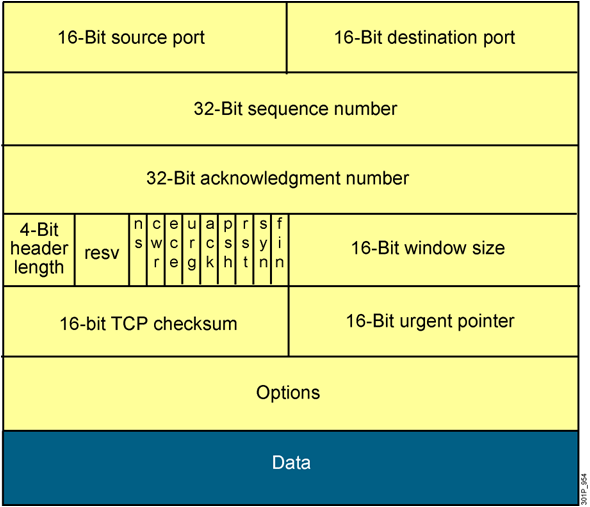


Fig 67 Appendix Tcp header

Source port and destination port (both 16 bits long): used to identify the session of the protocol on which the application layer is being transmitted in the TCP segment being considered.

Sequence number (32 bit): used to number the packet sequence (from the sequence number it will calculate the number of bytes transmitted).

Acknowledge number (32 bit): Used to indicate which packet has been received and expect to receive the byte with the next sequence number.

Header length (4 bits): shows how long the entire header is in word units (1 Word = 4 bytes).

* Reserverd bits (4 bits): are set to 0
* Bit control (9 bits): bits used to control ACK flags, Sequence flags, etc.
* Window size (16 bits): the number of bytes the device is willing to accept
* Checksum (16 bits): checks for errors of the entire TCP segment
* Urgent pointer (16 bits): used in case of data priority
* Options (up to 32 bits): allows adding other features to TCP
* Data: data of the upper layer

**How it works**

Suppose host A wants to transmit data to host B through a TCP connection. Before transmitting, host A needs to establish a TCP connection with host B this is done through a 3-step handshake as s:

* Step 1: Host A sends B a packet with the SYN flag turned on, with the sequence number numbered as 100. This first segment contains no data so there is no data, but the number of data bytes remains. is counted as a byte for the sending of the SYN flag.
* Step 2: Host B receives the packet, B sends the packet with the SYN flag turned on, accompanied by an ACK flag to confirm.

Suppose host B sets up a segment with sequence number 300. This segment reply from Host B also has no data but still counts as 1 byte for the data. When responding to host A, host B also needs to specify in the ACK sequence field the sequence number of the next byte it wants to receive from host A. Since the SYN segment sent by A is counted as 1 byte, B will expect takes the next byte as the 101st byte from A, so the ACK sequence is numbered as 101. (SEQ = 300, ACK = 101)

* Step 3: After the connection has been established, A sends the packet to meet the needs of B. The packet is numbered SEQ = 101 to meet the needs of B. ACK = 301 is used to indicate receipt Packet with SEQ = 300.

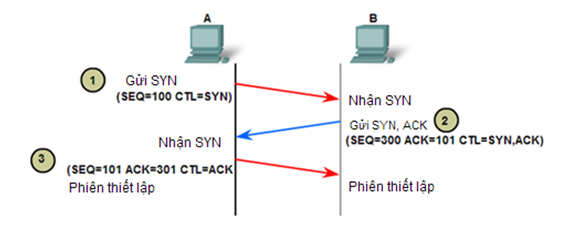


Fig 68 Appendix Tcp work

## UDP (User Datagram Protocol)

In contrast to TCP, UDP is a connectionless directional transmission protocol. It will not perform a connection-building operation before transmitting data, but will immediately transfer when there is data to be transmitted (best effort transmission mode) => very fast transfer for application layer data.

No guarantee of reliability when transmitting data and no mechanism for data recovery (it does not care if the packet arrives at its destination, does not know if the packet is lost on the way or not) => vulnerable error.

* Do not perform sequence numbering for the data units transmitted ...
* Faster and more efficient for small sized data and time-demanding.
* Being stateless, UDP is useful for answering small queries to a large number of requesters.

#### UDP packet structure

UDP packet structure is much simpler than TCP

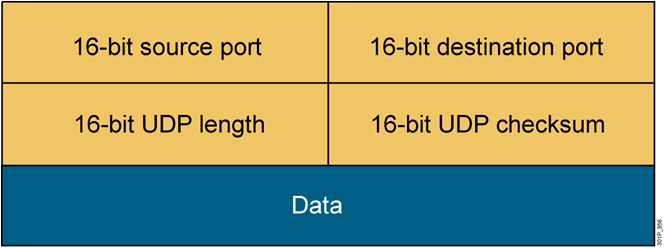


Fig 69 Appendix Udp header

* source port and destination port (both 16 bits): allow to identify a session of an application running on UDP. Can be considered the main port address of the Transport
* UDP length (16 bits): shows the length of the entire UDP datagram total how many bytes. (16 bits will have a total of 2 ^ 16 bytes = 65536 values (from 0 -> 65535 bytes)).
* UDP checksum (16 bits): uses CRC loop code algorithm to check errors for the entire UDP datagram and only checks for a limited amount
* Data: The upper layer data is packed into the UDP datagram being reviewed.

### RTP

#### RTP Header

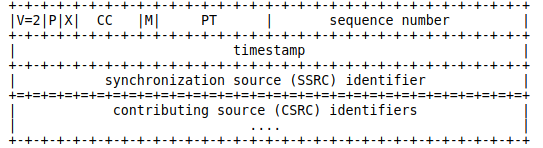


Fig 70 Appendix Rtp header

The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer.

The fields have the following meaning:

version (V): 2 bits

* This field identifies the version of RTP. The version defined by this specification is two (2). (The value 1 is used by the first draft version of RTP and the value 0 is used by the protocol initially implemented in the "vat" audio tool.)

padding (P): 1 bit

* If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the payload. The last octet of the padding contains a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.

extension (X): 1 bit

* If the extension bit is set, the fixed header is followed by exactly one header extension

CSRC count (CC): 4 bits

* The CSRC count contains the number of CSRC identifiers that follow the fixed header.

marker (M): 1 bit

* The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream. A profile may define additional marker bits or specify that there is no marker bit by changing the number of bits in the payload type field

payload type (PT): 7 bits

* This field identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats. Additional payload type codes may be defined dynamically through non-RTP means. An initial set of default mappings for audio and video is specified in the companion profile Internet-Draft draft-ietf-avt-profile, and may be extended in future editions of the Assigned Numbers RFC. An RTP sender emits a single RTP payload type at any given time; this field is not intended for multiplexing separate media streams.

sequence number: 16 bits

* The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt, because the packets may flow through a translator that does.

Timestamp: 32 bits

* The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations.The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient). The clock frequency is dependent on the format of data carried as payload and is specified statically in the profile or payload format specification that defines the format, or may be specified dynamically for payload formats defined through non-RTP means. If RTP packets are generated periodically, the nominal sampling instant as determined from the sampling clock is to be used, not a reading of the system clock. As an example, for fixed-rate audio the timestamp clock would likely increment by one for each sampling period. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent. The initial value of the timestamp is random, as for the sequence number. Several consecutive RTP packets may have equal timestamps if they are (logically) generated at once, e.g., belong to the same video frame. Consecutive RTP packets may contain timestamps that are not monotonic if the data is not transmitted in the order it was sampled, as in the case of MPEG interpolated video frames. (The sequence numbers of the packets as transmitted will still be monotonic.)

SSRC: 32 bits

* The SSRC field identifies the synchronization source. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier. Although the probability of multiple sources choosing the same identifier is low, all RTP implementations must be prepared to detect and resolve collisions. If a source changes its source transport address, it must also choose a new SSRC identifier to avoid being interpreted as a looped source.

CSRC list: 0 to 15 items, 32 bits each

* The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources. For example, for audio packets the SSRC identifiers of all sources that were mixed together to create a packet are listed, allowing correct talker indication at the receiver.

#### RTCP Packet Format

This specification defines several RTCP packet types to carry a variety of control information:

* SR: Sender report, for transmission and reception statistics from participants that are active senders
* RR: Receiver report, for reception statistics from participants that are not active senders
* SDES: Source description items, including CNAME
* BYE: Indicates end of participation
* APP: Application specific function

Each RTCP packet begins with a fixed part similar to that of RTP data packets, followed by structured elements that may be of variable length according to the packet type but always end on a 32-bit boundary. The alignment requirement and a length field in the fixed part are included to make RTCP packets "stackable". Multiple RTCP packets may be concatenated without any intervening separators to form a compound RTCP packet that is sent in a single packet of the lower layer protocol, for example UDP. There is no explicit count of individual RTCP packets in the compound packet since the lower layer protocols are expected to provide an overall length to determine the end of the compound packet.

Each individual RTCP packet in the compound packet may be processed independently with no requirements upon the order or combination of packets. However, in order to perform the functions of the protocol, the following constraints are imposed:

Reception statistics (in SR or RR) should be sent as often as bandwidth constraints will allow to maximize the resolution of the statistics, therefore each periodically transmitted compound RTCP packet should include a report packet.

New receivers need to receive the CNAME for a source as soon as possible to identify the source and to begin associating media for purposes such as lip-sync, so each compound RTCP packet should also include the SDES CNAME.

The number of packet types that may appear first in the compound packet should be limited to increase the number of constant bits in the first word and the probability of successfully validating RTCP packets against misaddressed RTP data packets or other unrelated packets.

Thus, all RTCP packets must be sent in a compound packet of at least two individual packets, with the following format recommended:

Encryption prefix: If and only if the compound packet is to be encrypted, it is prefixed by a random 32-bit quantity redrawn for every compound packet transmitted.

* SR or RR: The first RTCP packet in the compound packet must always be a report packet to facilitate header validation as described in Appendix A.2 in rfc 1889. This is true even if no data has been sent nor received, in which case an empty RR is sent, and even if the only other RTCP packet in the compound packet is a BYE.
* Additional RRs: If the number of sources for which reception statistics are being reported exceeds 31, the number that will fit into one SR or RR packet, then additional RR packets should follow the initial report packet
* SDES: An SDES packet containing a CNAME item must be included in each compound RTCP packet. Other source description items may optionally be included if required by a particular application, subject to bandwidth constraints
* BYE or APP: Other RTCP packet types, including those yet to be defined, may follow in any order, except that BYE should be the last packet sent with a given SSRC/CSRC. Packet types may appear more than once.

## Hibernate

### What Is Hibernate In Java?

Hibernate is a framework in Java which comes with an abstraction layer and handles the implementations internally. The implementations include tasks like writing a query for CRUD operations or establishing a connection with the databases, etc.

A framework is basically a software that provides abstraction on multiple technologies like JDBC, servlet, etc.

Hibernate develops persistence logic, which stores and processes the data for longer use. It is lightweight and an ORM tool, and most importantly open-source which gives it an edge over other frameworks.

### What Is An ORM Tool?

It is a technique that maps the object stored in the database. An ORM tool simplifies data creation, manipulation, and access. It internally uses the Java API to interact with the databases.

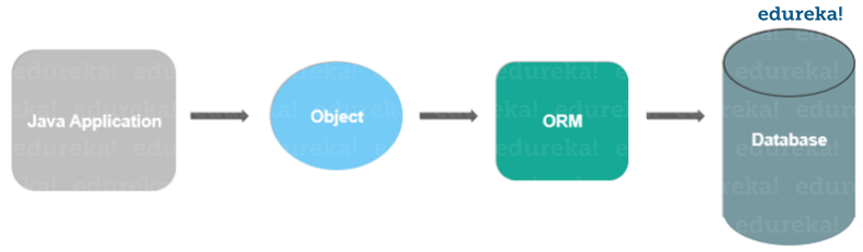


Fig 71 Appendix Orm

Advantages Of Hibernate In Java

* Lightweight and open-source – Being lightweight and open-source makes it accessible and efficient.
* Increased performance – Using cache memory helps in fast performance.
* Database Independence – Being database-independent gives it the ability to work with different databases.
* Auto DDL Operations – automatic table creation saves us from manually creating tables.
* It takes care of mapping Java classes databases using XML files without writing any code.

We can directly store and retrieve data directly from the database using simple APIs.

It does not require any application server to operate.

Minimizes database access with smart fetching strategies.

It provides simple querying of data.

Defect

Does not support complex queries

Some cases still need to use native SQL because Hibernate cannot cover all the syntaxes of database management systems.

Restricted interference with SQl statements because it is automatically generated.

Source <https://www.edureka.co/blog/what-is-hibernate-in-java/>

### Hibernate architecture

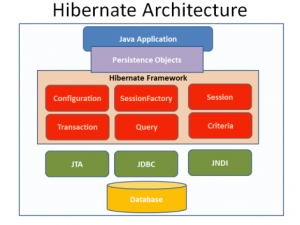


Fig 72 Appendix Hibenate

Each table in the database is an object in Hibernate. Therefore, you need to have one java bean for every table in the database. These java beans will have getters / setters and some exceptions that follow Hibernate's convention.

Each file mapping has the form \*\*\*. Hbm.xml is responsible for specifying each relationship between the properties of the object and the fields in the database table (which column corresponds to which attribute of the class, the type of price) What is the value, how does the relationship between tables transform into relationships between classes ...).

Hibernate.cfg.xml file: This is the first file to load when launching the Hibernate application. It contains information about the connection (what is connected database (MySQL, Oracle, SQL server ...), what is the username and password connected to the database, the database name is connected), the connected driver ...

## Network Address Translation (NAT)

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of private IP address to a public IP address is required. Network Address Translation (NAT) is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on router or firewall.

### Network Address Translation (NAT) working

Generally, the border router is configured for NAT i.e the router which has one interface in local (inside) network and one interface in the global (outside) network. When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address. When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.

If NAT run out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

**Why mask port numbers?** Suppose, in a network, two hosts A and B are connected. Now, both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time. If NAT does an only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network and sent to the destination. Destination will send replies on the public IP address of the router. Thus, on receiving a reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A and B are same). Hence, to avoid such a problem, NAT masks the source port number as well and makes an entry in the NAT table.

### NAT inside and outside addresses

Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organisation. These are the network Addresses in which the translation of the addresses will be done.

IMG_256

Fig 73 Appendix Nat

Inside local address – An IP address that is assigned to a host on the Inside (local) network. The address is probably not a IP address assigned by the service provider i.e., these are private IP address. This is the inside host seen from the inside network.

Inside global address – IP address that represents one or more inside local IP addresses to the outside world. This is the inside host as seen from the outside network.

Outside local address – This is the actual IP address of the destination host in the local network after translation.

Outside global address – This is the outside host as seen form the outside network. It is the IP address of the outside destination host before translation.

### Network Address Translation (NAT) Types

There are 3 ways to configure NAT:

**Static NAT** – In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global address. This is generally used for Web hosting. These are not used in organisations as there are many devices who will need Internet access and to provide Internet access, the public IP address is needed.  
Suppose, if there are 3000 devices who need access to the Internet, the organisation have to buy 3000 public addresses that will be very costly.

**Dynamic NAT** – In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP address. If the IP address of pool is not free, then the packet will be dropped as an only a fixed number of private IP address can be translated to public addresses.  
 Suppose, if there is a pool of 2 public IP addresses then only 2 private IP addresses can be translated at a given time. If 3rd private IP address wants to access Internet then the packet will be dropped therefore many private IP addresses are mapped to a pool of public IP addresses. NAT is used when the number of users who wants to access the Internet is fixed. This is also very costly as the organisation have to buy many global IP addresses to make a pool.

**Port Address Translation** (PAT) – This is also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address. Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address. This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

**Advantages of NAT**

* NAT conserves legally registered IP addresses .
* It provides privacy as the device IP address, sending and receiving the traffic, will be hidden.
* Eliminates address renumbering when a network evolves.

**Disadvantage of NAT**

* Translation results in switching path delays.
* Certain applications will not function while NAT is enabled.
* Complicates tunneling protocols such as IPsec.

Also, router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.