


Analysis of Enterprise VoIP Traffic from a Wire line IMS System

Yu Bai

Master Thesis

Supervisor: Tord Westholm, EAB

Supervisor: Dougherty Mark, DU

 HÖGSKOLAN Dalarna		<h1 style="text-align: center;">DEGREE PROJECT</h1> <h2 style="text-align: center;">Computer Engineering</h2>	
Program Master of Science in Computer Engineering		Reg. Number E3709D	Extent 15 ECTS
Name of student Yu Bai		Year-Month-Day 2009-09-17	
Supervisor Prof. Dr. Mark Dougherty		Examiner Prof. Dr. Mark Dougherty	
Company Ericsson Company		Supervisor Mr. Tord Westholm	
Department of Computer Engineering			
Title Analysis of Enterprise VoIP Traffic from a Wire line IMS System.			

Abstract

With the rapid development of telecommunication industry, the IP Multimedia Subsystem (IMS) could very well be the panacea for most telecom operators. It is originally defined as the core network for 3G mobile systems by the 3rd Generation Partnership Project (3GPP), the more recent development is merging between fixed line network and wireless network.

This report researches the characteristic of the IMS data and proposes an IMS traffic characterization analysis. We captured the IMS traffic data with 10 thousands users for about 41 hours. By analyzing the characteristics of the IMS, we know that the most important application in the IMS is VoIP call. Then we use the tool designed by Tsinghua University & Ericsson Company to recognize the data, and the results we got can be used to build the traffic models. From the results of the traffic models, I will get some reasons and conclusion. The traffic model gives out the types of session and types of VoIP call. I bring into a concept—busy hour. This concept is very important because it can help us to know which period is the peak of the VoIP call. The busy hour is from 10:00 to 11:00 in the morning. I also bring into another concept—connection ratio. This concept is significant because it can evaluate whether the VoIP call is good when it use IMS network.

By comparing the traffic model with other one's models, we found the different results from them, both the accuracy and the busy hour. From the contract, we got the advantages of our traffic models.

Acknowledgement

Firstly, I will give my heartfelt thanks to my supervisor **Mr. Tord Westholm**. He helps us to know the approximate situation of the project. He helps us about the analysis, model building and writing the report.

I owe my most sincere gratitude to my supervisor **Dougherty Mark** in Höskolen Dalarna. During the course of doing the thesis, he applies some help to us.

I will give my best acknowledgement to **Johan Gard and Kevin Wang** in EAB/FKS/D. They give me this opportunity to finish the thesis and apply the knowledge about IMS to me. With the help, we can understand the IMS deeply.

I also wish to thanks my friends, **Jiaqi Hou, Guowei Zhao, Zhenfang Wei and Jason Zhang**. They gave me some wholesome suggestions during the thesis period.

The support of my parents is gratefully acknowledged as well.

List of Abbreviations

AUC	Authentication Center
BT	British Telecom
BGCF	Breakout Gateway Control Function
BH	Busy Hour
CSCF	Call Session Control Function
HSS	Home Subscriber Server
HLR	Home Location Register
IMS	IP Multimedia Subsystem
IPTV	Internet Protocol Television
I-CSCF	Interrogating Call Session Control Function
ISDN	Integrated Service Digital Network
PoC	Proof of Concept
P-CSCF	Proxy Call Session Control Function
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RSUP	Reliable SAP Update Protocol
SIP	Session Initiation Protocol
S-CSCF	Serving Call Session Control Function
SBC	Session Border Controller
SDP	Session Description Protocol
TISPAN	Telecommunications and Internet Converged Services and Protocols for Advanced Networking
VoIP	Voice over Internet Protocol
3GPP	3rd Generation Partnership Project



Table of Contents

Abstract	1
Acknowledgement	3
List of Abbreviations.....	4
List of Figures	7
Chapter 1 Introduction.....	8
1.1 A general introduction.....	8
1.2 Related Work.....	10
1.3 Thesis Outline	11
Chapter 2 Background.....	12
2.1 Introduction to IMS.....	12
2.1.1 Introduce to core network element of IMS	13
2.1.2 The interfaces of IMS.....	14
2.1.3 The characteristic of IMS.....	15
2.2 Introduction to VoIP/SIP.....	15
2.2.1 General Introduction to SIP.....	15
2.2.2 The Type of Request Message in SIP	16
2.2.3 The header of SIP	16
2.2.4 Conditional code of SIP	18
2.2.5 General introduction of VoIP	18
2.3 Introduce to Traffic Recognition	19
2.4 IMS VoIP traffic modeling	20
Chapter 3 Analyzing Process and Tools	21
3.1 The tool design by Tsinghua	21
3.1.1 Test the tool	21
3.1.2 The problems of the tool	22
3.1.3 Theoretically result of the tool	23
3.2 The Capturing Tool	24
Chapter 4 Traffic Measurement Description.....	26
4.1 The Trace Data	26
4.2 The characteristics of the IMS data.....	27
Chapter 5 IMS VoIP Traffic Model and Analysis	29
5.1 Types distribution.....	29
5.1.1 Session types distribution.....	29
5.1.2 VoIP types distribution	31
5.2 Session duration distribution.....	32
5.2.1 Holding Time Analysis.....	33
5.2.2 Session's holding time (Max, Mean and Min)	34
5.3 Number of Session distribution.....	37
5.3.1 Active number of session by time	37
5.3.2 VoIP call flow	38



5.3.3 VoIP call number of packets.....	40
5.3.4 Number of VoIP call bytes	44
5.4 Connection Ratio.....	47
Chapter 6 Discussion	49
Chapter 7 Conclusion	53
Chapter 8 Future Work	55
References.....	56
Appendixes.....	58

List of Figures

Figure 1	The Structure of IMS.	12
Figure 2	The Conformation of IMS.	13
Figure 3	An example of SIP header.	17
Figure 4	The process of building a conversation.	18
Figure 5	General idea of the Traffic Recognition.	19
Figure 6	Flow chart of the tool.	22
Figure 7	The detail of the trace file.	23
Figure 8	The detail of the stat file.	23
Figure 9	An example of TCP-dump.	25
Figure 10	Session Types.	30
Figure 11	Packets of every session.	31
Figure 12	Bytes of every session.	31
Figure 13	VoIP Types.	32
Figure 14	Holding time probability.	33
Figure 15	Session's holding time (max, mean and min).	34
Figure 16	Average session holding time (24 hours).	36
Figure 17	Active session number by time.	38
Figure 18	Number of completed VoIP call per hour.	39
Figure 19	Percentage of VoIP call in a day.	40
Figure 20	Number of VoIP call packets per minute (1440 minutes).	41
Figure 21	Number of VoIP call packets per hour in a day.	42
Figure 22	Percentages of VoIP call packets.	43
Figure 23	Number of VoIP call bytes per minute (1440 minutes).	44
Figure 24	The number of VoIP calls bytes per hour in a day.	45
Figure 25	Percentages of VoIP call bytes.	46
Figure 26	VoIP Call Completion Ratio (24 hours).	48
Figure 27	Number of VoIP call flow.	49
Figure 28	VoIP call flow number in a week.	50
Figure 29	VoIP call holding time.	51
Figure 30	VoIP call flow duration of time in a week.	51

Chapter 1 Introduction

1.1 A general introduction

The IP Multimedia Subsystem (IMS) is an architectural framework, and it is in charge of delivering Internet Protocol (IP) multimedia services. It is open-systems architecture that supports a range of IP-based services over both wireless and wire line networks. It uses a simple and effective, low-cost and standardized network to deal with all the business in order to limit the supplier, simultaneously choose the best network configuration. In IMS, SIP (Session Initiation Protocol) is used for the voice service.

IMS has its own characteristics. IMS can concentrate the data, speech and network technology based on IP infrastructure. IMS is not intended to standardize applications but rather to aid the access of multimedia and voice applications from wireless and wire line network, such as voice, text, pictures and videos. It also provides end users with a personalized, richer communications experience. For example, in the past, if people wanted to open a videoconference, all the people must present at the same time to discuss the point of the conference. But now, using IMS, you can open a videoconference wherever you are. You can use the cell phone, the notebook, the computer and any other IMS compatible device. On the mobile net, it can also use IMS to supply multimedia value-added services, such as PoC (Proof of Concept), the instant message, video sharing and so on. About the wire line networks, some requirements of enterprise are integrated, such as VoIP. VoIP call is an important application in IMS because of its low price and convenience. It is better to talk to the friends who are in other countries using the VoIP call, because the price of the traditional phone is high and the quality of connection is not good.

We have many problems to solve in this thesis.

Firstly, we need to know the IMS structure, including its internal structure and how these structure works. Especially, we must know the SIP, because SIP protocol is very important for the IMS. Many applications are based on SIP protocol such as VoIP call and video.

Secondly, we will know the structure of IMS data. It is important for us to research the users of IMS. It is convenient for us to build the traffic model if we understand the characteristics of the IMS data, so research the structure of IMS data is essential step for us.

Thirdly, the tool designed by Tsinghua university & Ericsson AB is fit for internet data but not IMS data. So my partner updates the tool in order to recognize the IMS data.

The important motivation of the report is building the traffic models. I will find out the types of session and types of VoIP calls, which period time is the busy hour and the connection ratio. According to analyze the models, I will find out the reasons of the models. I think the traffic model is a significant element to evaluate the IMS network. I analyze busy hour in order to find out which period the users number is the biggest. I analyze the connection ratio in order to evaluate whether the quality is good when using IMS network. All the models which I build can be used to analyze the applications of IMS network. These results are also the valuable information for the research of IMS. Until now, the most important application of IMS is VoIP call. So in this project, I focus on the research of VoIP call.

The thesis work was performed at Ericsson as a research project and the opinions and results stated in the report cannot be taken as Ericsson's official view in any case. The author is responsible for the results and the opinions stated.

1.2 Related Work

A part of this thesis is to investigate the architectures of IMS and build the traffic recognition model especially VoIP call traffic models, and to see if it is possible to compare them with my results. This section will present the related work found in this area

British Telecom (BT) has identified the importance of IMS and has taken the radical step of embarking on a 10-year plan worth £10 billion to completely overhaul the core network to one based on the IMS model. Having modified the IMS model to particularly suit BT's requirements, the BT model is called the 21st Century Network (21CN). [1]

Some authors have written about use of SIP on IMS. It can find use in applications ranging from Internet telephony and conferencing to instant messaging, event notification, and the control of networked devices. [2]

Some authors analyze the SIP focus on the SIP Instant Message. [14]

Other authors intend to analyze the VoIP traffic based on internet and use about three weeks' data. [13]

Usually the model quantifies the traffic and latency for various procedures defined in IMS, starting from the basic call flows. We present the model in this paper and also

compare the IMS traffic with other traditional schemes and make conclusions on its efficiency [1].

1.3 Thesis Outline

The thesis is divided into 8 chapters. The first character is Introduction. It includes the general purpose of the thesis, the related work and the outline of the thesis. The chapter 2 is Background. In this chapter, I will give a general introduction of IMS and VoIP/SIP not only from the structure but also from the functions. In addition, I will also introduce the traffic recognition and especially IMS/VoIP modeling. Chapter 3 analyzes processes and Tools. I will introduce the tool which I used especially the Wire Shark that used to build traffic model. The 4th chapter is Traffic Measurement Description. It is an important chapter because from this chapter I will describe the traffic trace that is used in the report including how and where in the network the data was collected. I will also describe the structure and characteristics of the IMS data. Chapter 5 is the most central in this report. It is about IMS VoIP Traffic Model and Traffic Analysis. In chapter 5, I will describe in detail what and how I will model the IMS VoIP. I will also give an analysis about why such result appeared. The chapter 6 is Discussion and the chapter 7 is Conclusion. The last chapter is Future Work. At last are the References and Appendixes.

Chapter 2 Background

2.1 Introduction to IMS

IMS is an architectural framework for delivering internet protocol multimedia services. It is intended to aid the access of multimedia and voice applications from wireless and wire-line terminals. IMS was originally designed by the standards body 3GPP (3rd Generation Partnership Project) and TISPAN (Telecommunications and Internet Converged Services and Protocols for Advanced Networking). 3GPP is responsible for connecting IMS with wireless network. TISPAN has been working with 3GPP to extend the IMS architecture with capabilities required in support of wire-line access. IMS supports conversation class and none conversation class, and it is an important step to an all IP network. It is also an attempt to convergence of wire-line network and wireless network, so called Fixed Mobile Convergence (FMC).

Figure1 shows the structure of IMS.

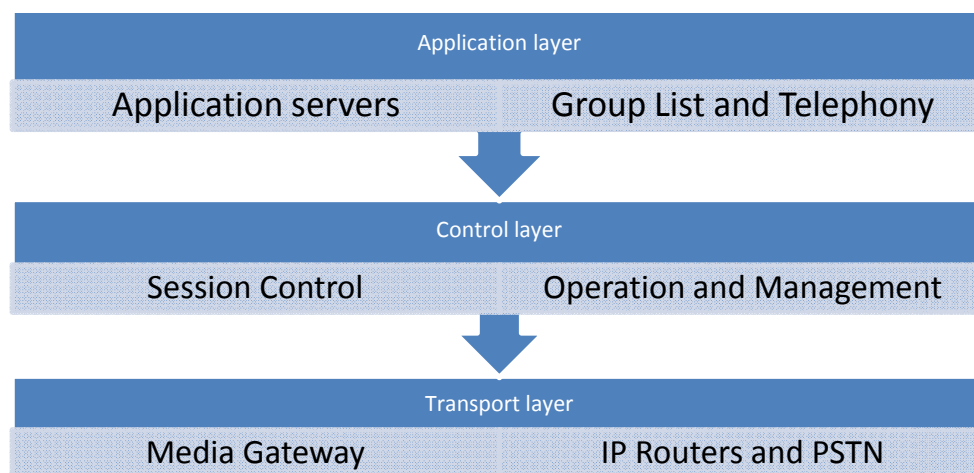


Figure 1 The Structure of IMS.

IMS is a hierarchical structure: the bottom is the transport layer, fixed network and mobile transport from here, the media gateway is in this layer. And then is the control

layer, some network elements are in this layer, such as operation, management, provisioning and charging. The top is the application layer. In this layer, it has some new application, such as video share, voice communication and transports some images.

2.1.1 Introduce to core network element of IMS

The core IMS elements are: S-CSCF; I-CSCF; P-CSCF; BGCF; MGCF; MGW; MRF; HSS; SLF and SBC. These elements play an important role in transporting the data. Each element has its own function in the system. The follow figure is about IMS conformation.

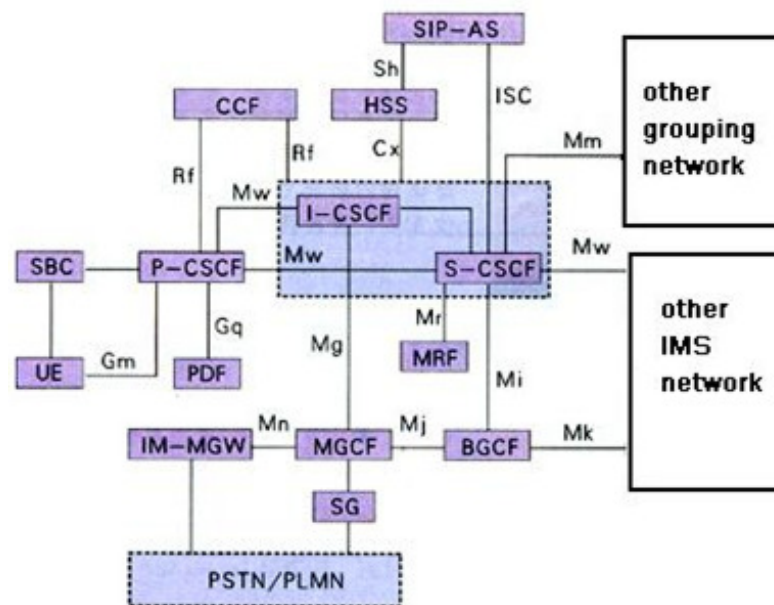


Figure 2 The Conformation of IMS.

CSCF (Call Session Control Function) is the core of the network, and it is the functional entity in IMS. The function deals with the signaling during multimedia calling. It also has other roles, such as: administer the user authentication of IMS net; handle the QoS of IMS and control of SIP when converse with other network entities

and resource allocation.

On the basis of different functions, CSCF has three roles: P-CSCF, S-CSCF and I-CSCF.

P-CSCF is an entrance when IMS calls a network, all the calling messages whatever begins or end must according to P-CSCF. P-CSCF has many functions, such as network security attack and defense, compression and decompression the SIP signaling in order to save wireless resource, user roaming control and QoS.

I-CSCF is an entrance of IMS ownership of network. In the program of registration, I-CSCF look into HSS and then choose an S-CSCF for user. In the process of calling, the calling signal goes to I-CSCF first, and then gets some information in HSS, and then goes to S-CSCF.

S-CSCF has a chief status in the IMS network call control, and it receives some registered request from P-CSCF. S-CSCF and HSS administrate user message together.

HSS is a user database, and it saves information of user. It also holds the network which service for the user. It is a new register, and it evolves by HLR and AUG.

SBC: SBC have evolved to address the wide range of issues that arise when voice and multimedia services are overlaid on IP infrastructure. SBC is like a filter and it can filter some information such as instant IP voice, the video and other conversation through by IP network. SBC can provide security and protection against.

2.1.2 The interfaces of IMS

IMS has some core interfaces, such as Cx, Mw, Rf, Sh, Gq, Gm and Mm. Next I will introduce some of them.

Cx interface: Cx is a very important interface. It is a bridge between I-CSCF or S-CSCF to HSS.

Sh interface: Sh is a connection between AS to HSS.

Gm interface: Gm is used to connect UE and P-CSCF. It is a significant interface because P-CSCF is the entrance when you want to visit the IMS network. So the

user's requirement must go through Gm.

Mw interface: Mw interface is a bridge between CSCF to CSCF.

2.1.3 The characteristic of IMS

IMS network has its own characteristics. About the wireless network, the new SIP terminal may insert in IMS directly. But conventional wire line network needs a new PSTN/ISDN artificial network to insert in IMS. The best feature is that IMS is an open architecture. The operators do not need to scrap the existing equipments in order to upgrade the network. IMS does not only have a low cost of common services, but also has powerful ability to develop the new services. IMS uses end to end technology to finish some communications. IMS has some multimedia services based on SIP. For example, it supports interactive end-to-end game, the PoC, the instant message, push-to-talk, charging and so on. IMS has the most important service—built a call bridge between mobile service and fixed internet service.

2.2 Introduction to VoIP/SIP

2.2.1 General Introduction to SIP

The SIP (Session Initiation Protocol) is a signaling protocol which is used to work up and remove multimedia communication sessions widely like voice and video calls over Internet Protocol (IP). In November 2000, SIP was accepted as a 3GPP signaling protocol and permanent element of the IP Multimedia Subsystem (IMS) architecture for IP-based streaming multimedia services in cellular systems [4]. SIP builds and controls every kind of point to point media session through a portable way. Similar with internet protocol, it is also a modular structure, request/answer model, and based on text mode. It is very simple, flexible and it is convenient to update and extent. SIP use SDP to describe the characters of the terminal operations. SIP does not only

supply QoS (Quality of Service) itself but also operated with RSUP (Reliable SAP Update Protocol).

2.2.2 The Type of Request Message in SIP

SIP protocol has many types of request message. These requests are very important for the IMS data.

INVITE	When people want to initiate a conversation.
ACK	If a person receives INVITE, he will send ACK to confirm
BYE	When people want to terminate a conversation
Register	Registered contact address of client
Cancel	It is used to cancel the ongoing INVITE request

2.2.3 The header of SIP

The header is significant for SIP, and the header connect status line directly. The header does not only provide some information about request and respond, but also provide some information about this information which included message body. Some of them can be used both in request and respond. But the others can only be used in one condition, request or respond.

The following figure illustrates an example about how to use the header in the data. And then I will introduce the header.

```
INVITE sip:Gonzalo.Camarillo@ericsson.com SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd];branch=z9hG4bKnashds7
Max-Forwards: 70
Route: <sip:pcscf1.visited1.net;lr>, <sip:scscf1.hcmel.net;lr>
From: <sip:user1_public1@hcmel.net>;tag=171828
To: <sip:Gonzalo.Camarillo@ericsson.com>
Call-ID: cb03a0s09a2sdfglkj490333
Cseq: 127 INVITE
Contact: <sip:[5555::aaa:bbb:ccc:ddd]>
Content-Type: application/sdp
Content-Length: 248
```

Figure 3 An example of SIP header.

Request-URI: The address of target users

From: Both request and response must include this title, because it explains who initiate this call. The server copies this title from request to response.

To: It is always included in a request message. But sometimes it includes the address of target user. During the total conversation, it has the same content.

Call-ID: Call-ID is the most important title in SIP. It represents a kind of sharing SIP signaling relationship between two or more users. It identifies a specific invitation and all the following services related this invitation. When deal with many conversations at the same time, the server uses Call-ID to connect the accessing information and relevant conversation. So the Call-ID can identify the different voice call.

Cseq: It is also a central title in SIP. It is used to match the request and respond. After the INVITE (except ACK and CANCEL), all the requests contain a Cseq. The result of initial request is incremental. In the other word, if you want to do other applications such as send a message or send a picture during one conversation, the result of Cseq is incremental.

Via: It saves all the proxy address of request.

Contact: Contact address of following request.

The following figure is process when build a conversation

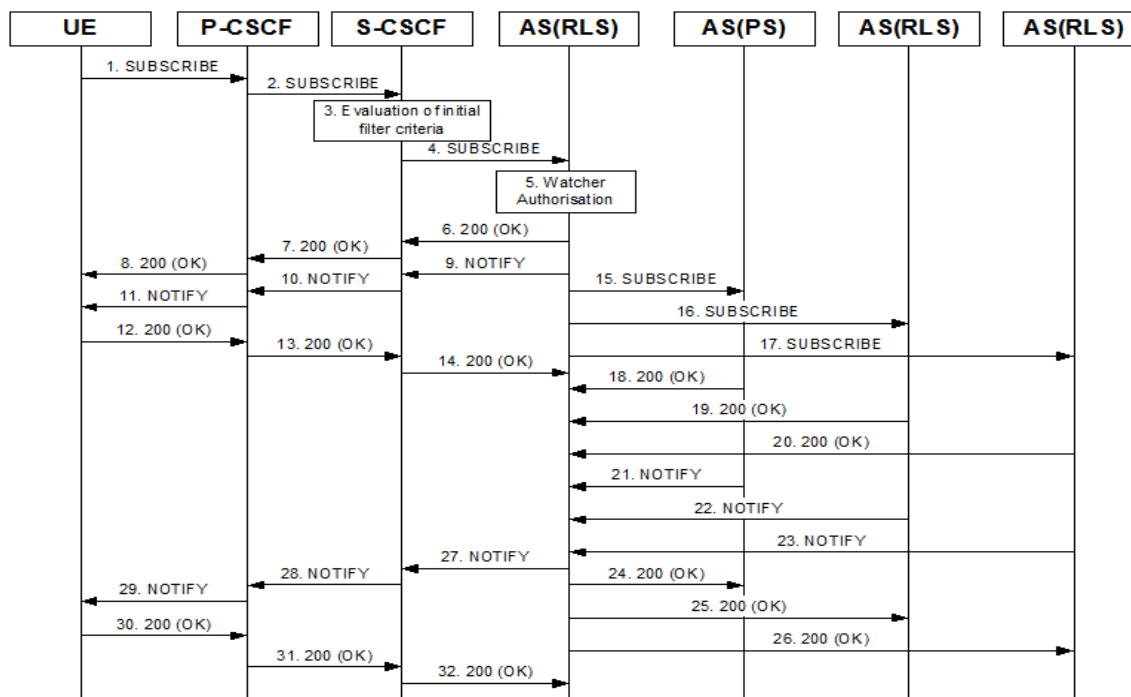


Figure 4 The process of building a conversation.

1xx: provisional responses

2xx: success

3xx: redirection

4xx: client error

5xx: server error

6xx: global failures

2.2.5 General introduction of VoIP

VoIP means voice over internet protocol, and it can supply voice communications over IP network to the transmission technologies, such as

the Internet or other packet-switched networks. Now there are several VoIP solutions, one of which is IMS VoIP and the other is Skype. In this thesis our focus is IMS VoIP. VoIP services convert your voice into a digital signal that travels over the Internet. VoIP can allow you to make a call directly from a computer, a special VoIP phone, or a traditional phone connected to a special adapter.

The price of VoIP call is very cheap. Traditional phones must charge high price when you make an international long-distance call. If your friend goes abroad to work or study, connect to his domestic parents is inconvenient. But if you use SIP you only charge the fee of local telephone service and the corresponding.

VoIP has a better call quality. Traditional telephone cannot avoid distortion because it has some defects in the technical aspect. But VoIP has a digital transmission by internet, so it has a better call quality.

2.3 Introduce to Traffic Recognition

The following figure shows the general idea of the traffic recognition and how it transforms the information into traffic flows. The approach for the traffic recognition must be link speedily, for instance, both the recognition part and the action part must be on link-speed time scale.



Figure 5 General idea of the Traffic Recognition.

This tool consists of two main parts: one is traffic recognition and signature

recognition; and the other is a protocol type. About the traffic recognition, the signature analysis is the most important element, and since the traffic in IP networks is rather complex and diverse this part must be flexible enough. Hence, the traffic recognition part must be significant for the tool because it must have enough basic functionality to deal with this complexity.

2.4 IMS VoIP traffic modeling

The IMS has diverse of applications, such as VoIP call, the videoconference, the instance message and the picture, the voice and so on. In the applications, the VoIP call and video are the most common applications. So in this report, I will build some traffic model with VoIP call and video, the population in BH (Busy Hour).

From the result, we can get that how many calls are cancelled, how many calls are completed, how many calls is rejected. We can also get that how many people use the IMS to call and how many people to have a video over a period of time. We can also know that how many people are there to use the IMS in the busy hour and how many people use IMS in the flexible time. Over all, building the traffic model is very favorable for us to improve the IMS in the future. Know the usage of the users in order to add the new applications easily.

Chapter 3 Analyzing Process and Tools

3.1 The tool design by Tsinghua

3.1.1 Test the tool

In this project, we use a tool which designed by Tsinghua University & Ericsson Company together. The tool is very useful, and it can recognize the data captured from internet. The tool must be tested because it was not designed to recognize IMS network. The tool has its own demands in order to run successfully.

Firstly, the tool must run in the Linux system. The Linux is different from windows, because it is not only an open environment but also about it is a freeware. We must write some commands when we want to install a new tool or run an application.

Secondly, the tool is written in C++, so we must know some knowledge about C++. In this project, in order to install the tool, we must install two software packets first: the libpcap and the ACE.

Libpcap is a network packet to capture function package in Linux platform. Most of the networks monitoring software are based on it. One characteristic is application framework.

ACE is the abbreviation Adaptive Communication Environment. It is available free to use, open source object-oriented framework. It also implements a number of concurrent communication software for the core model. ACE provides a rich set of reusable C++ wrapper facade and framework components and it can be used across multiple platforms to complete a common communication software tasks: it include

signal processing, service initialization, inter-process communication and so on.

3.1.2 The problems of the tool

The following figure is the flow chart of the tool.

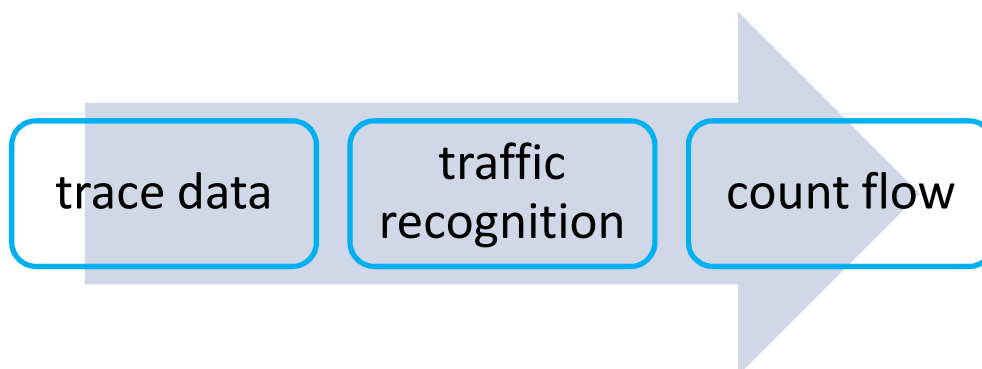


Figure 6 Flow chart of the tool.

The figure illustrates the course of the tool. We use the trace data as the input and bring it into the tool. Theoretically speaking, we can get some count flows as the output through the tool and we can also get the detail of the trace data. But in real time, we meet some difficulties.

Firstly, when we run the trace data, there are some errors. The most important is short of head file in the code. We get the detail of the data after we add some head files.

Then we met another difficult. Sometimes the tool is fit to the wired line. When we use wireless network, we will meet some errors such as eth0 sends failed. The reason is the data between ingoing and outgoing have a collision when they go through one port. In order to solve this problem we can built a fictitious network card named eth0 to let the data to get in. The system captures packets from eth0, analyzes the packets, recognize the application type, and record the statistical information.

Thirdly, we get the internet data as the trace data and bring it into the tool. From the output we could see some details of the internet data. For example, we can see the application names, the flow count and the flow percentage.

3.1.3 Theoretically result of the tool

Theoretically, when we bring the data into the tool, we can run the tool, and we could get the applications types, the packets numbers and the flow percentage after it runs successfully. We can get two files: trace file and stat file. The following figures are the trace file and the stat file.

trace.txt:

SrcIP	SrcPort	DestIP	DestPort	Proto	StartTime	EndTime	PktsCount	TotalSize	Type
192.168.237.167	1146	219.223.222.3	30604	TCP	14:55:08	14:55:08	10	416	Unknown
192.168.237.167	1153	222.28.152.141	80	TCP	14:56:05	14:56:05	56	50247	HTTP
192.168.237.202	22	192.168.237.167	1174	TCP	14:56:29	14:56:29	1	20	SSH
192.168.237.167	1200	202.108.5.100	110	TCP	15:02:41	15:02:46	53	22595	POP3
192.168.237.167	1212	202.108.5.81	25	TCP	15:03:23	15:04:01	1015	737349	SMTP
192.168.237.167	1521	162.105.239.15	21	TCP	15:14:10	15:14:10	9	61	FTP
166.111.8.238	23	192.168.237.167	2528	TCP	16:17:38	16:17:38	1	1460	TELNET

Figure 7 The detail of the trace file.

Stat Info:

App Type	Flow Num	Packet Num	Total Size	Total Flow Count	App Porp
FTP	601	645	40714	14374	4.18%
HTTP	11462	21217	8813664	14374	79.74%
POP3	106	1179	863292	14374	0.74%
SMTP	19	1033	737629	14374	0.13%
SSH	125	1654	1068166	14374	0.87%
TELNET	154	474	42218	14374	1.07%
Unknown	1907	63370	5509256	14374	13.27%

Figure 8 The detail of the stat file.

The first figure is the trace file. From the figure we could see some information of every data. We could see the source IP, destination IP, source port, destination port

and the IP-protocol. These are five tuples which are the points that can recognize the traffic data in the internet. We could also see the start time, the end time, the packet count, the total size and the application types. This file is very important because the parameters in the file are significant to identify the different applications types.

The second figure is the stat file. From the figure we could see some result of every data. We could see the applications types, the flow number, the packet number, the total size and the application percentages. In all of the parameters the most important result is the application percentage. We can know which application used universally in a period of time from the result. We could also improve the business of the internet application based on this result.

After updating the tools from Tsinghua we could get the new results. Using the new results I will build the traffic recognition models.

3.2 The Capturing Tool

The purpose of this project is about constructing traffic model for IMS, but we only have the trace data in internet. From the data we can see some applications that we are familiar with. To be able to construct an IMS traffic model, IMS traffic data is needed. The IMS data is captured by TCP-dump. In IMS, there are some new applications and protocols that different from internet.

TCP-dump is a powerful collection and analyze tool of network data in Linux operating system. If I want to use a simple word to define the TCP-dump, it means dump the traffic on a network. It can intercepted data packet on the network packets according to the definition of users.

TCP-dump has a very strong function so every senior system administrator uses it to analyze network and troubleshoots the problem. TCP-dump can also completely intercept the header of data packet sent by network and provided an analysis. It supports to the network layer, protocol, host, network or port filtering. It also provides the logic statements such as “and”, “or”, “not” to help you remove useless information. TCP-dump provides a source code, and opens the interface. Hence, it has a strong scalability. It is a very useful tool both for network maintenance and the invaders. Ordinary circumstances, the directly start TCP-dump will monitor all packets flow through the first network interface.

The following figure is an example of the TCP-dump.

```

bash-2.02# tcpdump
tcpdump: listening on eth0
11:58:47.873028 202.102.245.40.netbios-ns > 202.102.245.127.netbios-ns: udp 50
11:58:47.974331 0:10:7b:8:3a:56 > 1:80:c2:0:0:0 802.1d ui/C len=43
0000 0000 0080 0000 1007 cf08 0900 0000
0e80 0000 902b 4695 0980 8701 0014 0002
000f 0000 902b 4695 0008 00
11:58:48.373134 0:0:e8:5b:6d:85 > Broadcast sap e0 ui/C len=97
ffff 0060 0004 ffff ffff ffff ffff ffff
0452 ffff ffff 0000 e85b 6d85 4008 0002
0640 4d41 5354 4552 5f57 4542 0000 0000
0000 00

```

Figure 9 An example of TCP-dump.

From the output results we could see that the format of TCP-dump is system time, source port, destination port and parameters of data packets.

TCP-dump has an important function that can filter the needed data. TCP-dump system without any parameters will search all network interfaces, and show all the data that it captured. These data are not necessary for our needs, and too much data is disadvantage for us to analyze data. So in this project we must use the parameters in the TCP-dump to choose which data we need. And then use these data to build traffic model and have many compare with different models.

Chapter 4 Traffic Measurement Description

4.1 The Trace Data

We use the data captured from internet as the trace data. The results can indicate the statistical information of the applications in Internet. The results can also show the IP-address of each flow. But our purpose is to recognize the IMS data. Hence, we collected the IMS data to fulfill the demands. The next table shows the details of IMS data which we captured.

Capture starting time	2009/3/23 9:18
Capture ending time	2009/3/25 03:41
Number of subscribers	More than ten thousands
Total duration time	More than 41 hours
Character of subscribers	From the enterprises

Table 1 The details of capturing information

Our data is captured from 2009/3/23 to 2009/3/25. It is more than ten thousands users and the duration time is more than 41 hours. The important point is that all the subscribers are in enterprises. We know that IMS data is different from internet because of its particular structure. The structure of IMS divided into two parts: header and body. The header included some header files, and all the key words are the point for us to identify the different application types in the IMS. The header is based on SIP protocol, and SIP protocol only applies control signaling but does not apply the coding and transmission of voice data. The body includes the RTP data packets, and in these packets it encapsulates the voice material. Based on privacy problem and

capacity problem, we do not capture the RTP data, but the data include a little RTP header files. So our data only included the SIP data and a little RTP header file.

IMS data has its own applications such as VoIP calls, video conferencing, sharing the pictures and entertainment on line. The tool is fit to internet but not fit to IMS, so we must write another script language in order to run the IMS.

The following table shows the different applications between IMS network and internet.

IMS Application	Internet Application
VoIP call	MSN
Video conferencing	BT
Sharing the pictures	PPLIVE
Entertainment on line	iTunes
End-to-end game	Google Talk

Table2 Different applications between IMS and Internet.

4.2 The characteristics of the IMS data

Nowadays, the most comprehensive application in the IMS is the VoIP call. The second application is the video. So our research point is the VoIP call. We could see the details of IMS data using Wireshark, about 90% of the data uses SIP protocol. The data has many stations, such as the VoIP call, register, notify and subscribe. The VoIP call has some different sessions, such as complete, in call, cancel, rejected and the call set. These stations represent the different courses of the VoIP call. If we can search the every packet deeply, we could find some rules in it.

Every handshake course is based on SIP protocol, and the protocol included some header files. These files are very important for us to identify the different calls. The most important factors are Call-ID and Cseq. Call-ID is a title in SIP. It represents a kind of sharing SIP signaling relationship between two or more users. Cseq is also a central title in SIP. It is used to match the request and respond. After the INVITE (except ACK and CANCEL), all the requests contain a Cseq. The results of initial request are incremental. In different dialog (including the register course and the calling course) the Call-ID is different. If the same user calls to different person, their Call-ID is increasing. If during the same dialog, the user wants to do other activities (including send the picture or send a message), the Cseq (sequence number) is increasing too. In the IMS net, the video is also comprehensive. In the SIP header, all the invites are based on SDP (Session Description Protocol). From the SDP protocol we could see the media type. Using this factor, we could identify whether the packet is audio or video. All of these factors are significant for us to build the traffic recognize models and apply the facilitate conditions for the future work.

Our data is in 41 hours. The time is from the 17:00 in 23th March to 9:00 in 25th March. All the data that used to build the model is completed VoIP call because only the completed VoIP call has longer time.

Chapter 5 IMS VoIP Traffic Model and Analysis

In this chapter, I will build the IMS traffic models to analyze in four classes. The first class is about session type distribution. In this class, I can know the types of the session are divided into notify, subscribe, register and VoIP call. I can also know the courses in the VoIP call. The second class is about the session duration distribution. The focus on this part is about the holding time analysis and average session holding time. The third class is about the number of session. In this part, first I will recommend a concept—Busy hour. And then build the model with number, packets and bytes. The last class is about the connection ratio. The connection ratio is an important concept when test IMS network. It means the completed VoIP call divided by all the VoIP call including the rejected, cancel, completed call and incomplete call in one hour. I will illustrate the different models following.

5.1 Types distribution

5.1.1 Session types distribution

From the wire shark, we can see that there are four types of the session. They are notify, subscribe, register and VoIP call respectively.

Build the model

The parameters: “notify”, “subscribe”, “register”, “VoIP call”

Notify: It means return current stat information

Subscribe: When a user wishes to receive information about the status of a service session, a subscribe request is sent to a server.

Register: Used by a user to notify its current IP address

VoIP call: Build a conversation in the current time

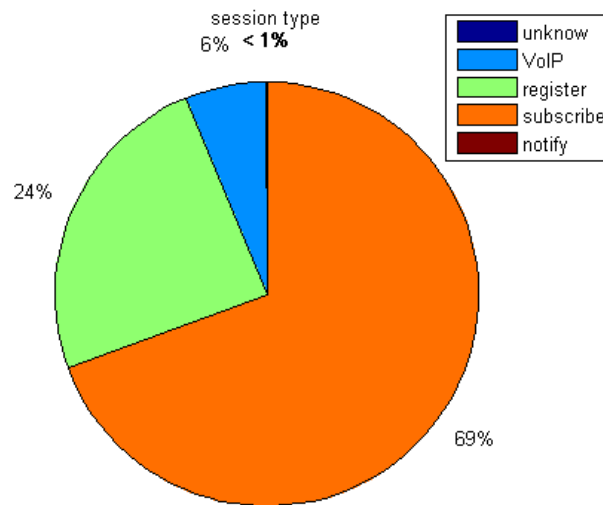


Figure 10 Session Types.

Analysis: From the figure we can see most of the session is “subscribe”. It is accounted for 69%. And next is “register” and it is about 24%. The VoIP call only build 6% and it is a small part of session. Only the small number of the session is “Unknown” and “notify”. The sum of the “unknown” and “notify” is less than 1%. It illustrates that at the present stage; most of the session is subscriber and register. They are only the courses of building a call but not an integral call. From the data which we captured we can see although VoIP is a small part of the session, it is very important for us to analyze IMS data. This part can help us to analyze the classes in the VoIP and the Busy Hour. We also see the connection rate from this part.

The following figure displays the type’s distribution with packets and bytes.

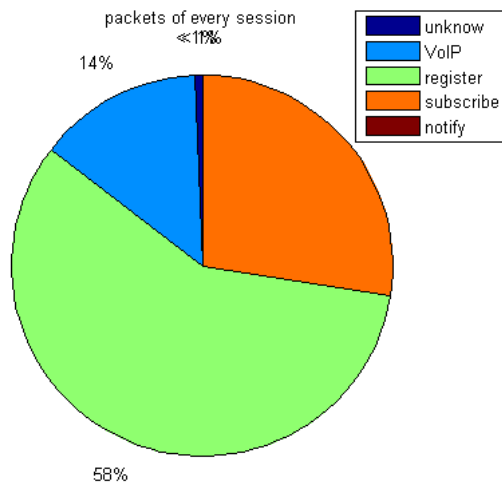


Figure 11 Packets of every session.

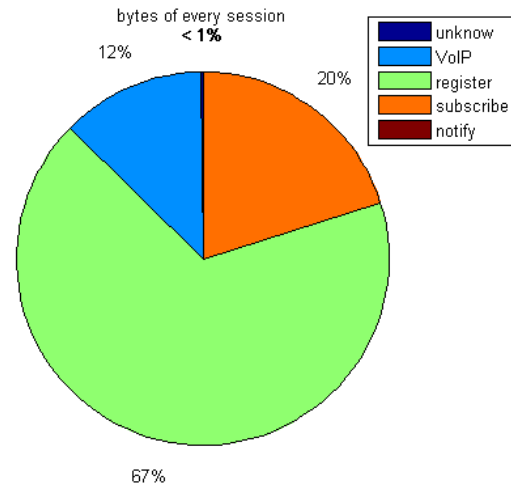


Figure 12 Bytes of every session.

The two figures display the types of the session from packets number and bytes number. We can see that the percentage distribution of the two figures is similar to the figure above. But most of the number of the packets and bytes are register, and next is subscriber. This appearance illustrates that the session named “register” included many packets and many bytes.

5.1.2 VoIP types distribution

Build the model

The parameters: rejected, cancel, incomplete and completed.

- Rejected: When the call has a busy tone or has an error, it will use rejected.
- Cancel: It is used to cancel the previous request initiated by the client.
- Incomplete: The call is not over.
- Completed: The call is over.

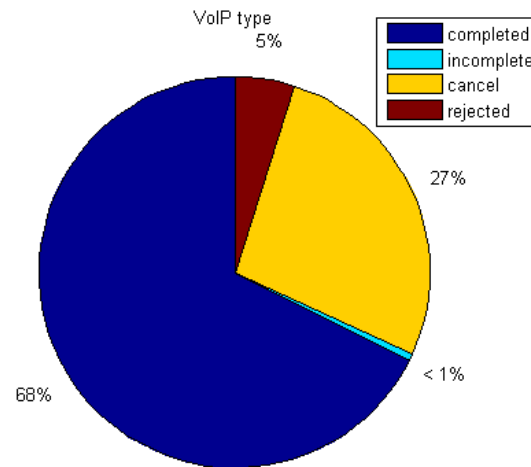


Figure 13 VoIP Types.

Analysis: From the figure we can see that most of the VoIP calls are completed. It builds about 68%. And the next is cancel and it is accounted for 27%. Only a small part of the VoIP call is rejected and incomplete. The sum of the two types accounts for less than 5%. From the percentage distribution we could get a result, at this stage most VoIP call is successful.

5.2 Session duration distribution

The second class I call it session duration distribution. In other word, it is about the holding time. I use the time difference to calculate the holding time. The monitor captured one packet using TCP-dump and then save it in the hard disk. The monitor adds some information on the header automatically. The information includes a time stamp, and it is the time of capturing the packets. One session includes many packets, so the holding time is the time difference between the first sessions to the last session. From the Wireshark, I could see that every VoIP call has its own holding time. The shortest duration is about 10^{-2} (s), and the largest duration is about 10^4 (s). In this part I also build the model about the average session holding time both in 24 hours and 41

hours, especially about the completed VoIP call.

5.2.1 Holding Time Analysis

Build the model

The parameters: the holding time and the probability of the VoIP call

The holding time: all successful completed calls' holding time

The probability of the VoIP call: the probability of all the successful completed calls

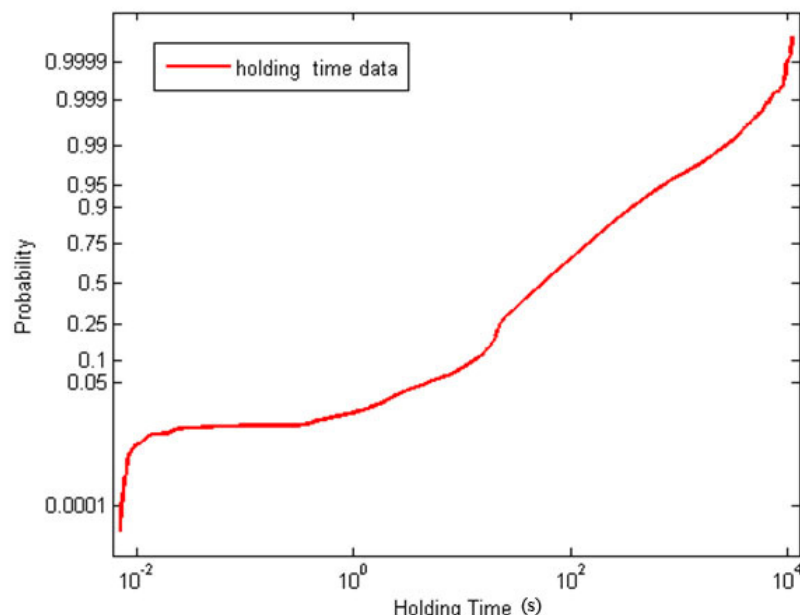


Figure 14 Holding time probability.

Analysis: The figure shows all the holding time of completed calls because only the completed calls have a longer duration. It illustrates that the shortest duration is less than 10^{-2} (s), and the largest duration is about 10^4 (s). In the data, the completed call is beginning with keyword “INVITE” and ending with keyword “BYE”. Some calls only have two packets with keyword “BYE” but not with “INVITE”. We also consider the VoIP call is completed call. In such a case, the duration time is very short. But this state is only a little part, less than 0.01%. From the figure we could also see

that about 5% VoIP call duration is in 1 second. About 10% VoIP call duration is less than 10 seconds. About 75% VoIP call duration is less than 100 second and all the VoIP duration is in 10000 second. This figure shows directly the relation between holding time and the probability of VoIP call.

At this part, the mean and median holding time can be calculated. The following is the definition of the mean and median.

$$\text{Mean} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\text{Median: } \int_{-\infty}^x f(x) dx \geq \frac{1}{2} \text{ or } \int_x^{\infty} f(x) dx \geq \frac{1}{2}$$

So the mean holding time is 164.13(s) and the median holding time is 118.17(s).

5.2.2 Session's holding time (Max, Mean and Min)

Build the model

The parameters: The duration time and the session holding time.

The duration time: The duration time of capturing the data.

The session holding time: The completed VoIP calls' holding time.

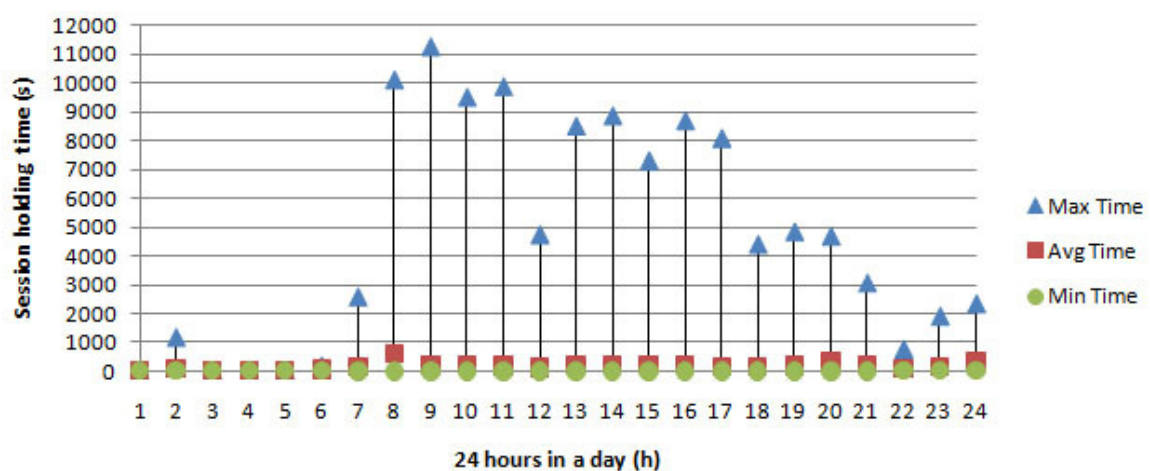


Figure 15 Session's holding time (max, mean and min).

Analysis: All the data used to build the model is completed VoIP call because only the completed VoIP call has longer time. The duration of notify and subscribe is too short to analyze. This figure is about the max session holding time, mean session holding time and min session holding time in each hour. All the statistics are in 24 hours. The beginning time in this figure is 0:00 in one day and the ending time is 24:00. We can see that the concentrated period time is from 8:00 in the morning to 17:00 in the afternoon. The reason is that this period time is the working hours, so the duration time of every completed call may be longer than the other time. The max holding time is higher than other time also. They have some business to talk about in the working hour. All the min session holding time is very low. The average session holding time is also low. From 1:00 to 7:00 in the morning, most people are sleeping and few people to use VoIP call. Even if there are people to make a VoIP call, they only talk to the point instead of chatting. So the duration is shorter than the daytime.

Build the model

The parameters: 24 hours in a day and the session time

The session time: completed calls' holding time.

24 hours of a day: a whole day.

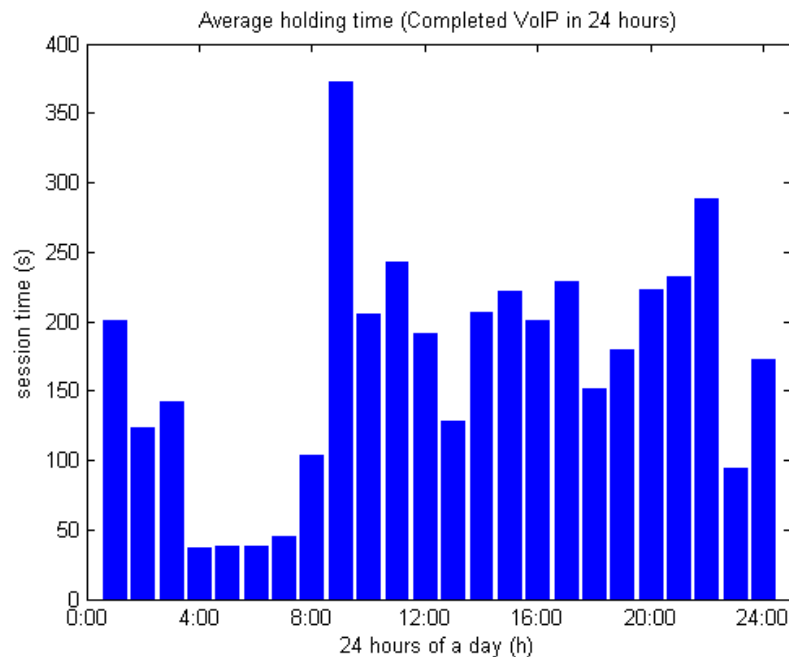


Figure 16 Average session holding time (24 hours).

Analysis: This figure is about the average holding time in 24 hours. The beginning time is the 0:00 and the ending time are 24:00. Although our data is in 41 hours, we must change it into 24 hours in a day in order to analyze the average holding time of the completed VoIP call. Firstly, the duration from 17:00 to 24:00 in the first day add to the duration from 17:00 to 24:00 in the second day, and then divided by 2. This result is the average holding time between 17:00 to 24:00. Secondly, the duration from 0:00 to 9:00 in the third day add to the duration from 0:00 to 9:00 in the second day, and then divided by 2. This result is the average holding time between 0:00 to 9:00. So the final figure is the average holding time in 24 hours in a day. From the figure we could see the longest duration is in 9:00AM, the reason is most people are working at this time, and they should use the VoIP call frequently and during a long time. The shortest duration is about at 4:00 or 5:00AM. Because at this time most people are in sleeping, so the holding time is short. The second highest duration is at 22:00 in the evening. We know that all the subscribers are enterprise. So the reason is at this time people may have some international business and use VoIP call to connect.

Hence the holding time is long also.

5.3 Number of Session distribution

I focus on the number of session distribution on this part. I will introduce a definition—Busy Hour. Busy hour is the busiest 60 minutes of the daily (24 hours) traffic. This concept plays an important role in the analysis because from this concept I can get a result about which period the number of users is the largest. So I will build the models from the following aspects: active number of session by time, VoIP call flow, number of VoIP packets and number of VoIP bytes.

5.3.1 Active number of session by time

Build the model

The parameters: 24 hours in a day and session number

24 hours in a day: a whole day

Number of session: how many sessions in an hour.

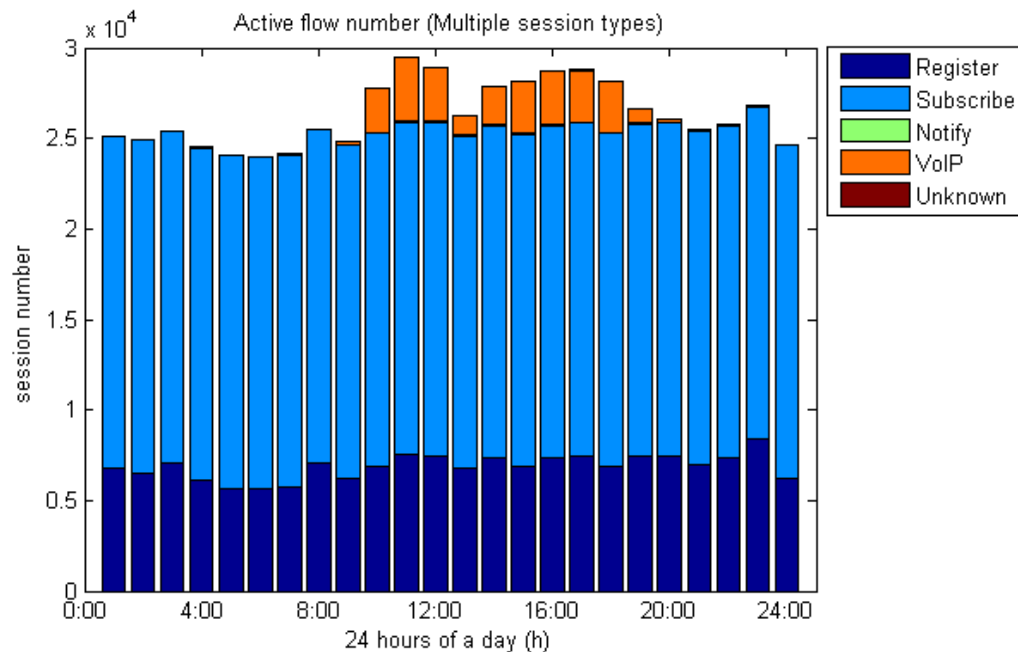


Figure 17 Active session number by time.

Analysis: We know that all the session divided by 5 classes: register, subscribe, notify, VoIP call and unknown. This figure illustrates the session number of every type in each hour. From the figure we could know that there are large number of subscribe and register, small number of notify and unknown. From 10:00 to 19:00 there are so many VoIP call but few in any other time. The reason is that our subscribers are enterprise and at this time most people are working, so they would use the VoIP frequently. First the UE sends a request of register to P-CSCF, and then UE finishes the register course. Second, the UE sends the subscribe request to S-CSCF and finishes the subscribe course. When the UE finishes the two courses, UE could finish all the activity but does not build a call. So the subscribe and register have a large number.

5.3.2 VoIP call flow

Build the model

The parameters: 24 hours of one day and number of VoIP calls per hour

24 hours of one day: a whole day.

Number of completed VoIP call per hour: how many completed VoIP call in one hour.

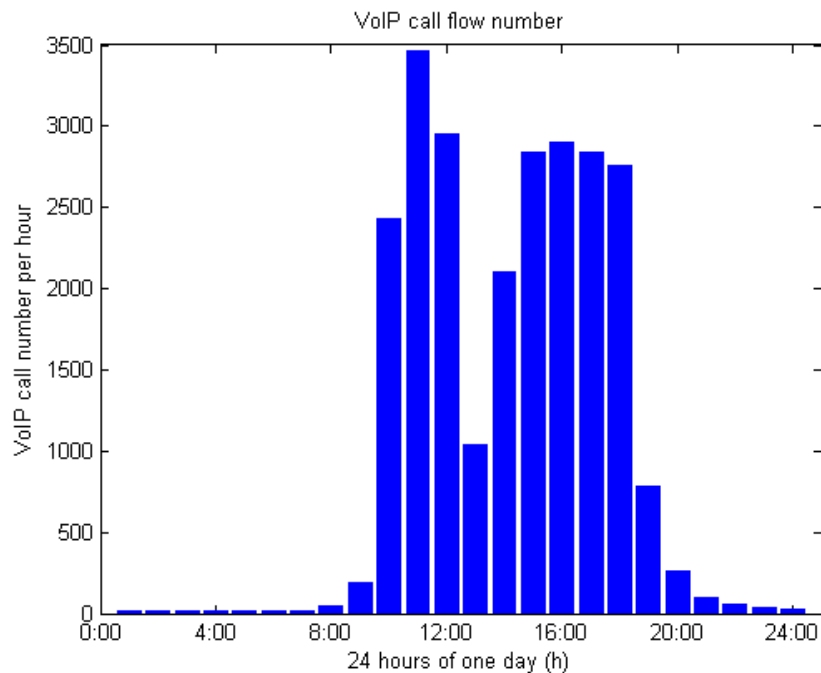


Figure 18 Number of completed VoIP call per hour.

Analysis: This figure is about the number of completed VoIP call per hour. We could see that the lowest number is less than 10 and the highest number is close to 3500. The disparity is big. We could get some results from the figure: in the morning, the peak center is on 10:00, 11:00 and 12:00. Because at this time most people are working and use the VoIP call frequently. At 13:00 and 14:00, most people have a rest time, so the number is small. In the afternoon, the peak focuses on 15:00, 16:00, 17:00 and 18:00. The reason is similar with morning. Most people use VoIP call to connect to their clients, friends and business. In the evening, the number of VoIP call is small because at this time there are so many other entertainments, so few people use VoIP call.

Build the model

The parameters: 24 hours of one day and VoIP call percentage

24 hours of one day: a whole day.

VoIP call percentage: percentage of the completed VoIP call in each hour.

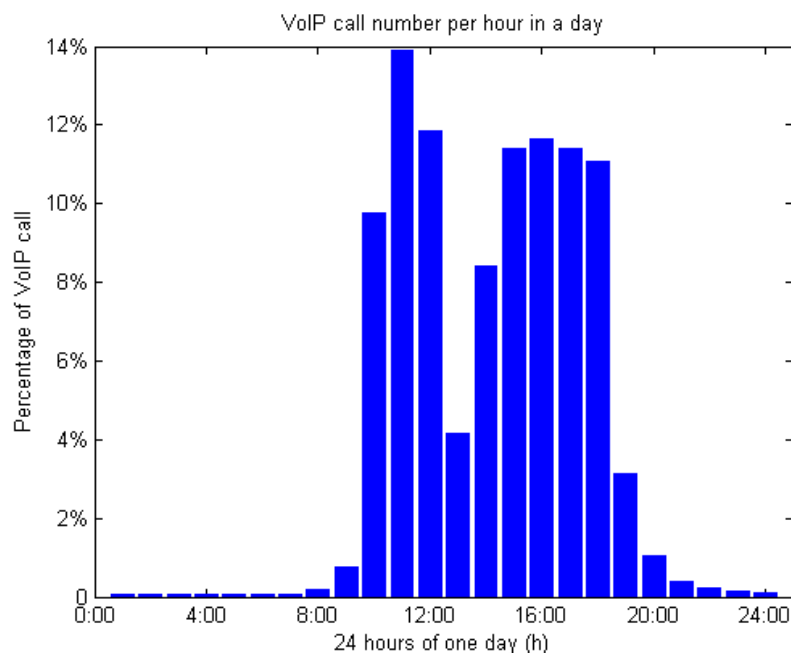


Figure 19 Percentage of VoIP call in a day.

Analysis: This figure is about the percentage of VoIP call in each hour. At this model, we could use the “Busy Hour” to evaluate the users’ number. Busy Hour means the busiest 60 minutes of the daily (24 hours) traffic. From the model, I could see that the busy hour is from 10:00 to 11:00 in the morning. The number of VoIP calls in this hour accounts for about 14% in the total VoIP calls. This hour is in the working time, so the users use the VoIP call frequently to talk to their clients about the business. The percentage of this hour is higher than the other hours. According to analyze busy hour, I could know that from 10:00 to 11:00 in the morning, the number of users is largest.

5.3.3 VoIP call number of packets

In this part, I will build the traffic model with VoIP call packets. I will analysis the

VoIP call packets per hour and per minutes. I will also analysis the packets with the percentage.

Build the model

The parameters: time of duration per minute and VoIP call number of packets

Time of duration: how long time of our data.

Number of VoIP calls packets: how many packets in an hour.

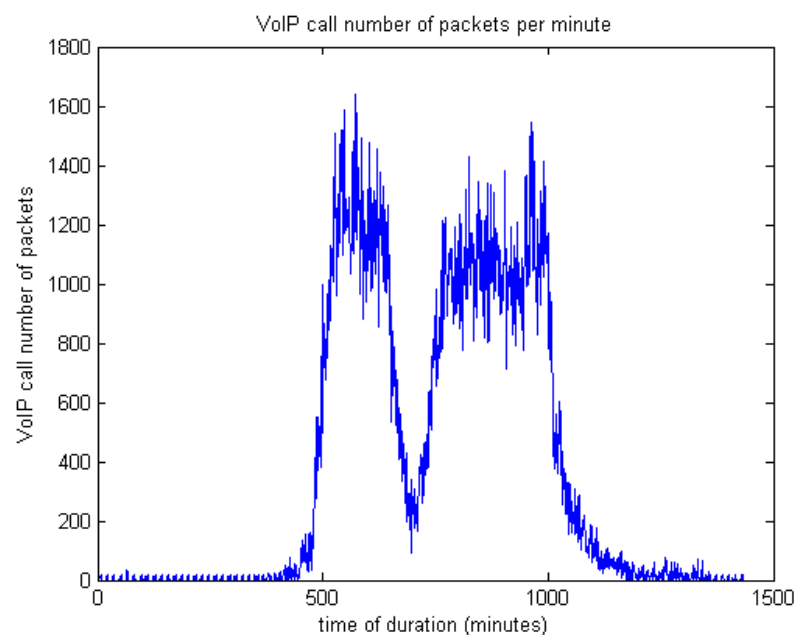


Figure 20 Number of VoIP call packets per minute (1440 minutes).

Analysis: This figure is about the number of VoIP call packets per minutes in a day. The duration time is about 1440 minutes in a day. From the figure above we could see that at 500 minutes, there is an uptrend. At the 700 minutes, it is a downtrend. At this period of time, there are too many activities and each activity is composed by many packets. So the number of packets is large at this time. The curve is very intensive. From 750 minutes to 1000 minutes, there is another peak. All of the subscribers are enterprises. This period of time is the working time and most of the users use the VoIP

call to talk with their clients. Each VoIP call includes a lot of packets. So during the peak period, the curve is also very intensive. In other time, most of the users are sleeping or have other entertainments to do, so the number of VoIP calls is little, accordingly, the number of the packets is little too.

Build the model

The parameters: 24 hours of one day and the VoIP call number of packets

24 hours of one day: a whole day.

VoIP call packets number: how many packets in an hour.

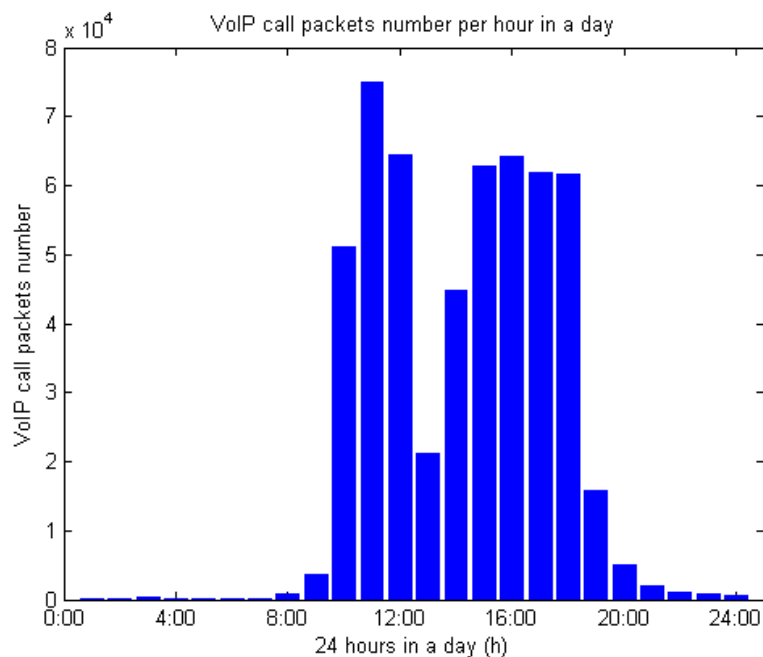


Figure 21 Number of VoIP call packets per hour in a day.

Analysis: The figure is about the number of VoIP call packets per hour in a day. It shows the lowest number of the packets is only a few hundreds and the largest number is closed 80000. We could see the large number is from 10:00 to 12:00 in the morning and from 14:00 to 18:00 in the afternoon in this figure. Because most people use the VoIP call frequently at the working time. Each VoIP call is composed by the

packets. So at this period of the time, the number of packets is big. At other time the people are in sleeping or have other entertainments to do. Even if the users use the VoIP call, they will only use the brief language to talk about the point. So the VoIP call at this time includes few packets. Hence, the number of the packets is small.

Build the model

The parameters: 24 hours in a day and percentage of the VoIP call packets

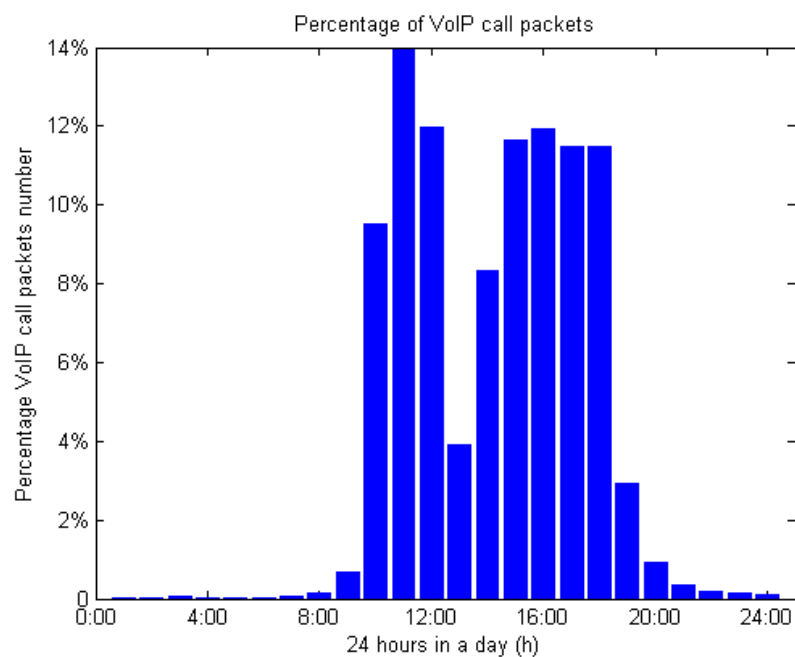


Figure 22 Percentages of VoIP call packets.

Analysis: This figure is about percentage of the VoIP call packets. By definition of the busy hour, the period is from 10:00 to 11:00 in the morning, because at this hour, the percentage of the packets is higher than the other time. Each VoIP call is composed by many packets. Large number of packets means the number of VoIP calls is large too. This hour is in the working time. All the users are from enterprise, so most of the users use VoIP call to work in this hour. Hence, this hour is the peak of one day (24 hours).

5.3.4 Number of VoIP call bytes

In this part, I will build the traffic model with VoIP call bytes. I will analyze the VoIP call bytes per hour and per minutes in a day. I will also analyze the bytes with the percentage.

Build the model

The parameters: time of duration per minute and number of VoIP call bytes

Time of duration: how long time of our data.

VoIP call bytes number: how many bytes in an hour.

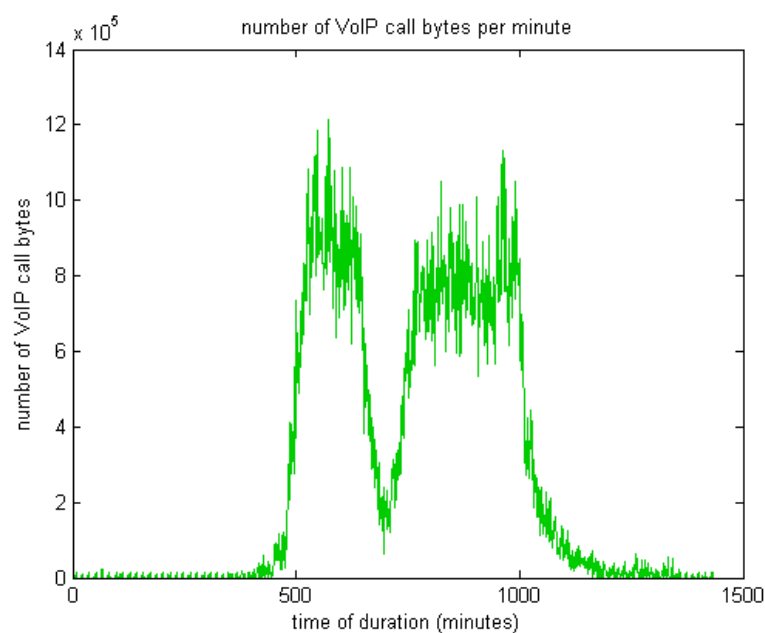


Figure 23 Number of VoIP call bytes per minute (1440 minutes).

Analysis: This figure is about the number of VoIP call bytes per minutes in a day. The duration time is about 1440 minutes in a day. From the figure above we could see that at 500 minutes, there is an uptrend. At the 700 minutes, it is a downtrend. Because at this period of time, there are too many activities and each activity takes up many bytes.

So the number of bytes is large at this time. The curve is very intensive. From 750 minutes to 1000 minutes, there is another peak. We know that all of the subscribers are enterprises. This period of time is the working time and most of the users use the VoIP call to talk with their clients. Each VoIP call takes up a lot of bytes. So during the peak period, the curve is also very intensive. In other time, most of the users are sleeping or have other entertainments to do, so the number of VoIP calls is little, accordingly, the number of the bytes is little too.

Build the model

The parameters: 24 hours of one day and the number of VoIP call bytes.

24 hours of one day: a whole day.

VoIP call bytes number: how many bytes in an hour.

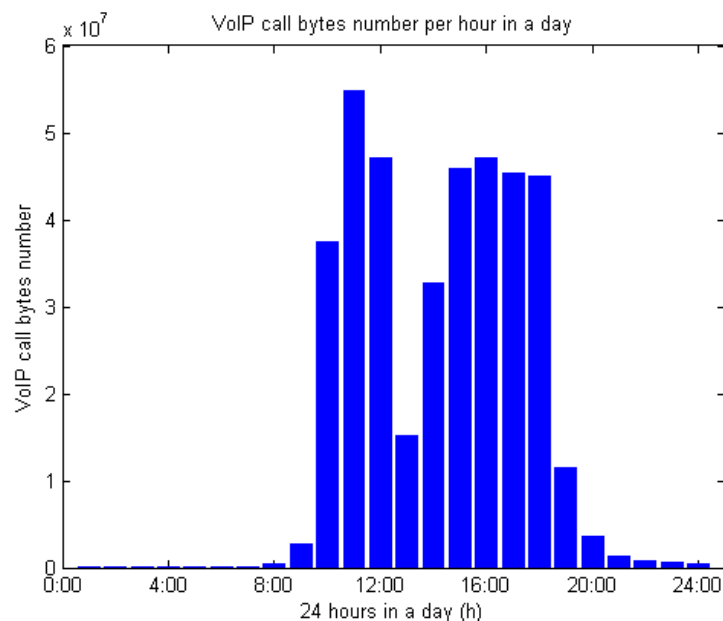


Figure 24 The number of VoIP calls bytes per hour in a day.

Analysis: The figure above is about the number of VoIP calls bytes per hour in a day. It shows that the lowest number of the packets is only a few hundreds and the largest

number is close to 6×10^7 . We could see the large number is from 10:00 to 12:00 in the morning and from 14:00 to 18:00 in the afternoon in this figure. Because most people use the VoIP call frequently at the working time. Each VoIP call takes up bytes. There are more bytes if the VoIP call duration is longer. So at this period of the time, the number of bytes is big. At other time the people are in sleeping or have other entertainments to do. Even if the users use the VoIP call, they will only use the brief language to talk about the point. So the VoIP call at this time takes up few bytes. Hence, the number of the bytes is small.

Build the model

The parameters: 24 hours in a day and percentage of the VoIP call bytes

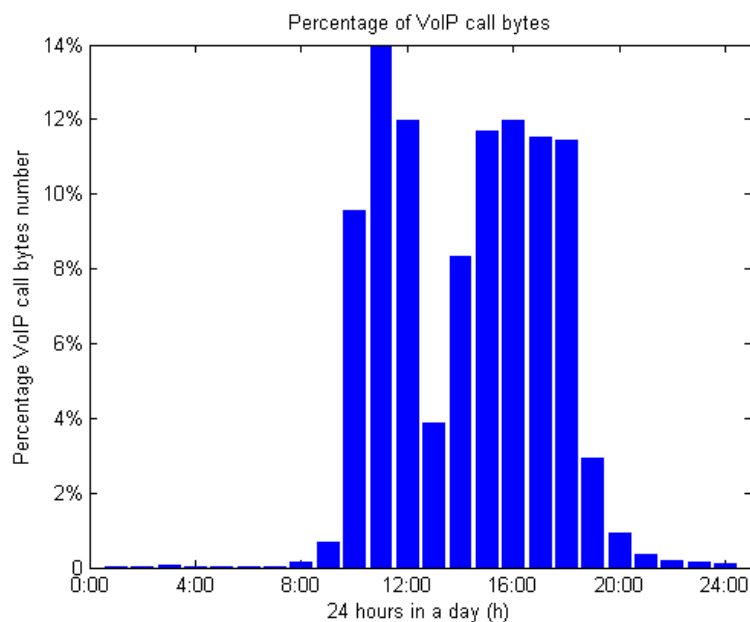


Figure 25 Percentages of VoIP call bytes.

Analysis: This figure is about percentage of the VoIP call bytes. From the model I know that the busy hour is from 10:00 to 11:00 in the morning, because at this hour, the percentage of the bytes is higher than the other time. Each VoIP call takes up many bytes. Large number of bytes means the number of VoIP calls is large too. This

hour is in the working time. All the users are from enterprise, so most of the users use VoIP call to work in this hour. Hence, this hour is the peak of one day (24 hours).

5.4 Connection Ratio

I focus on the VoIP call completion ratio analysis. In my thesis, the definition of VoIP call completion ratio is:

$$\text{Connection ratio (\%)} = \frac{\text{number of calls succeeded in conversation}}{\text{number of total calls}} \times 100\%$$

In this formula, the number of calls succeeded in conversation only means the completed calls. If one call only has a beginning or only has an ending, it is not a successful conversation. The number of total calls included the reject call, cancel calls, completed calls and incomplete calls in one hour. The connection ratio plays a very important role. It can evaluate whether the VoIP call is good when it use IMS network.

Build the model

The parameters: 24 hours in a day and the VoIP completed call rate

24 hours in a day: a whole day

VoIP completed call rate: completed call divided by total calls each hour.

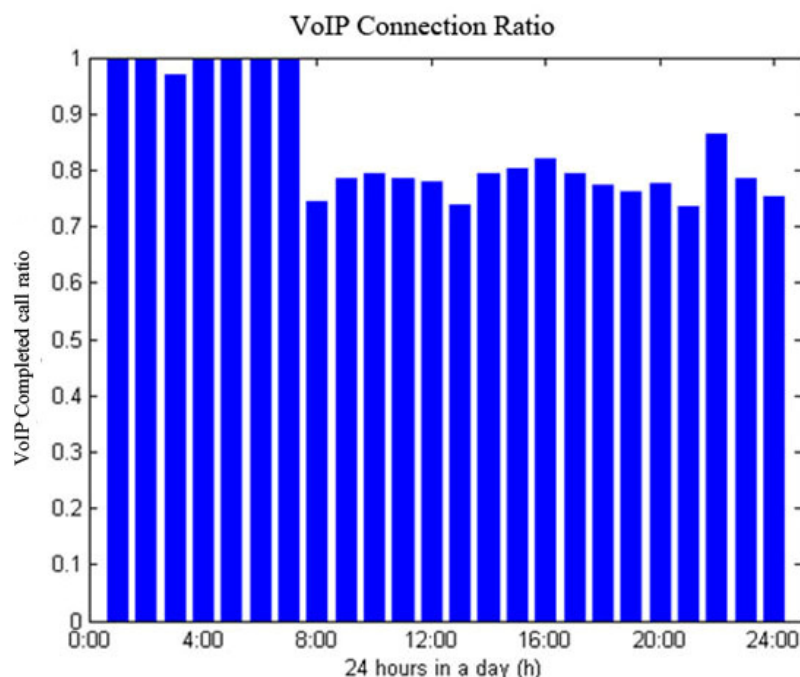


Figure 26 VoIP Call Completion Ratio (24 hours).

Analysis: From the figure above we could see that the result is similar with the 41 hours. From 0:00 to 7:00, the connection rate is close to 100%. The reason is at this time most people are in sleeping and only few people use VoIP call, so almost all the conversation between two users are successful. So the connection ratio is higher than any other time. Reversely, from 8:00 in the morning to 18:00 in the evening, the connection ratio is almost 70%. The reason is that at working time, there are many people to use the VoIP call on IMS network, so there are a lot of conversations unsuccessfully, either have a busy tone or have an error or have a cancel request. Hence, the connection ratio is low. From the figure we could see that the entire connection rate is above 70%. It illustrates that the quality of connection of VoIP call using IMS network is very good.

Chapter 6 Discussion

In my report, I got some results which are different from other articles. My data is only about 41 hours. In order to see the accurate models, I conform the data in 24 hours.

From the VoIP flow analysis.

The following figure is my model, about the number of the VoIP calls (Multiple VoIP calls types).

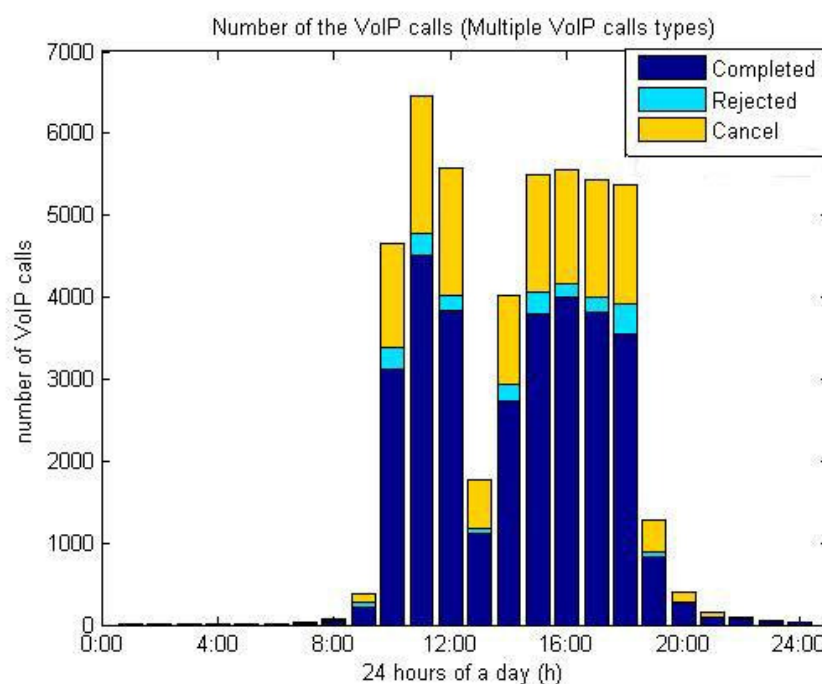


Figure 27 Number of VoIP call flow.

In the article [13], the author captured the data from January 8, 2007 to January 28, 2007. These data are about 3 weeks, so the author could analysis the VoIP flow weekly. The following figure is about VoIP call flow number in a week.

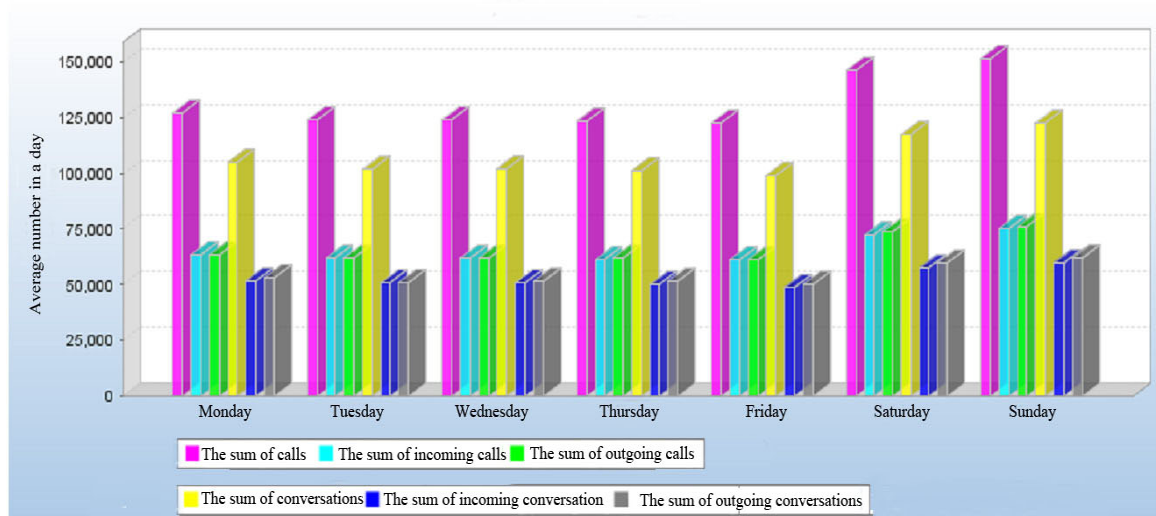


Figure 28 VoIP call flow number in a week.

From the figures above, I saw that there are some similarities and differences between them. The similarity is that both the figures are about the VoIP call number. The first figure shows that most of the VoIP calls are completed calls, then is the cancel calls in each hour. The second figure illustrates that the sum of calls and the sum of conversations. There are some differences between the two figures. The first model is in a day (24 hours), and I could see that from 9:00 in the morning to 18:00 in the afternoon, the number is larger than the other time. But the second model is in a week (7 days). From the figure we could see, the first histogram is the number of VoIP call each day in internet. The number is almost the same from the Monday to Friday. But in Saturday and Sunday, the number gets to the peak. So there are much more VoIP call in weekend.

From the VoIP call holding time.

The following figure is my model.

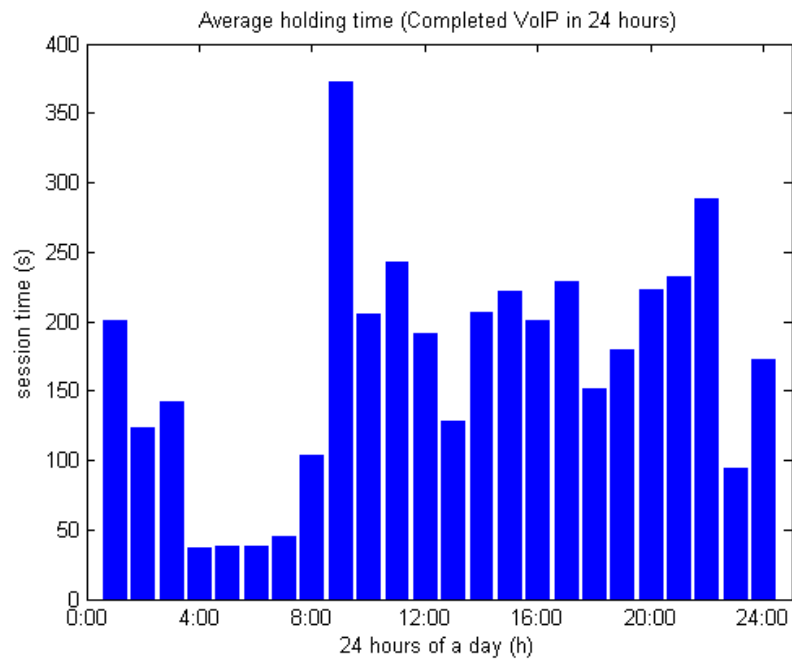


Figure 29 VoIP call holding time.

In the article[13], he author also analyzes the average holding time in a week. The following figure is about VoIP call flow duration of time in a week.

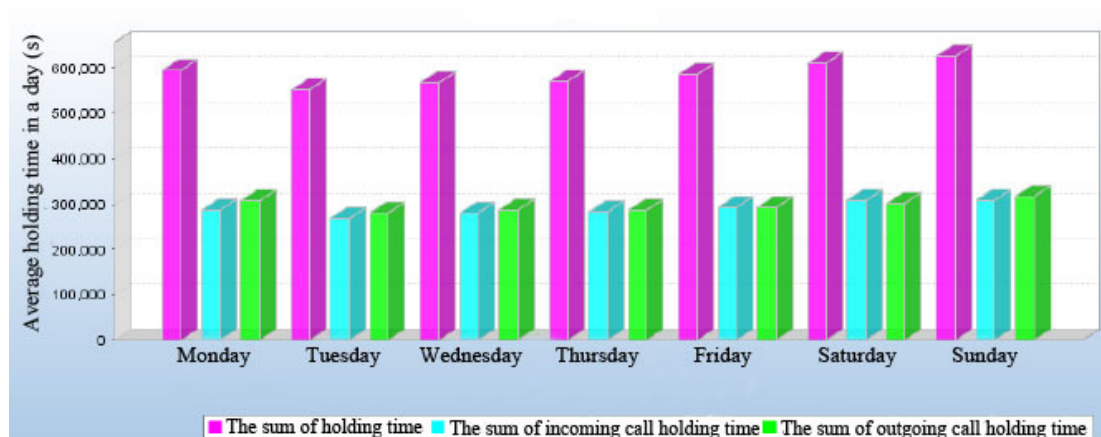


Figure 30 VoIP call flow duration of time in a week.

From the figures above, I saw that there are some similarities and differences between

them. The similarities is that both of the models are about the VoIP call holding time. They also have some differences. The duration of first one is in a day and the holding time in the working time is longer than the other time. The duration of second one is in a week, and the first histogram is the sum of the VoIP call holding time in a day. From the figure we could see that in Saturday and Sunday, the sum of the holding time is longer than other days.

We also have a discussion about average holding time. In the article[15], the author analyzes the VoIP call holding time based on internet. The author analyzes the average holding time in a day. From the 9:00 in the morning to 13:00 in the afternoon, the average holding time is 325.2(s), 345.4(s), 123.74(s), 165.54(s) and 322.54(s) separately. But in my article, the average VoIP call holding time is based on IMS network, so the average VoIP call holding time is different. In my models, from 9:00 in the morning to 13:00 in the afternoon, the average holding time is 375(s), 200(s), 240(s), 180(s) and 125(s) respectively. From the result, we could see that the result using internet is analogous with the result using IMS net. It illustrates that the IMS network applications are also universal.

We captured the data from different places. In the article[1], the author captured the data from Irland and built the IMS traffic model focus on Busy Hour. We could see that the percentage of the registration in BH is about 25%[1]. In my article, the percentage of the registration in BH is also 25%. It illustrates that no matter in Irland or other countries, the percentage of the registration is almost analogous.

Chapter 7 Conclusion

This report focuses on traffic modeling for IMS VoIP. Based on the statistics of IMS data, I build the traffic models from four aspects. The first class is about session type distribution. From the analyzing results, I found out that there are five types in IMS data—register, notify, subscribe, VoIP call and unknown. “Subscribe” dominates a large number of the session. It is accounted for 69%. “Register” takes up about 24%. The VoIP call only build 6% and it is a small part of session. The sum of the “unknown” and “notify” is less than 1%. From the analysis results, I find out the main application in IMS is VoIP call. The VoIP calls include four types: completed calls, incomplete calls, rejected and cancel. In all the statistics I know that completed call is a big part of the VoIP calls, about 68%. The cancel call accounts for about 27%. The incomplete call and rejected is account for 5% and 1% respectively.

The second class is about the duration distribution. The models in this class shows the holding time of every completed call. The model shows that the holding time in the working time is longer than the other time.

The third class is about the number of the VoIP call. In this aspect, I bring into a conception—busy hour. Busy hour means the busiest 60 minutes of the daily (24 hours) traffic. According to the models, I could see that the busy hour is from 10:00 to 11:00 in the morning. The number of VoIP calls account for about 14% in the total VoIP calls. This concept is a very important definition because it can help us to know which period is the peak in a day.

The last class is about the connection ratio. In this aspect, I bring into another important concept—connection ratio. Connection ratio means the number of calls

succeeded in conversation divided by number of total calls. According to the models I could get a summary. The connection ration is almost 100% from 0:00 to 7:00 in the morning. In another time the connection ratio is low. It plays a significant role because it can evaluate the quality of connection of VoIP call when using IMS.

According to compare the models with other author, I got that my models have their own advantages. They are more intuitive and more precise than the other models which in a week. All of my models are in 24 hours, so I could find the busy hour from it. It is better for us to analyze the IMS data flow.

Chapter 8 Future Work

I only capture IMS data in 41 hours, not enough 2 days. It is too little to build the models in a week or in a month. All of my models are in 24 hours. For example, I can build the models between weekdays and weekend if I have enough data. I can also build the models between four weeks in a month and compare the difference among them. So build the traffic models with more IMS data in the future.

My analysis only focuses on the VoIP calls. But in the IMS data, there are other applications such as Instant Message. So the analysis about IMS is not only in VoIP call but also in Instant Message in the future.

References

1. *A Traffic Model for the IP Multimedia Subsystem (IMS)*, **Abhayawardhana, V.S.Babbage, R.** 2007 IEEE 65th; Dublin, Ireland.
2. *The session initiation protocol: Internet-centric Signaling*, **H. Schulzrinne and J. Rosenberg**, IEEE Communications Magazine Oct. 2000, vol.38, no.10, pp.134-141.
3. *IPTV Traffic Measuring & Modeling*, **Geng Yu**, Stockholm: KTH Thesis, 2007.
4. http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem [online].
5. *Residential Broadband Network Traffic Analysis*, **Wei Zhenfang**, Stockholm: KTH Thesis, 2008.
6. *Signaling flows for the IP Multimedia call control based on SIP and SDP; Stage 3 (Release 5)*, **3GPP TS 24.228 v5.13.0 (2005-06)**, June, 2005.
7. *SIP: Session Initiation Protocol*, **J. Rosenberg, et al.**, IETF, RFC 3261, Jun. 2002.
8. <http://baike.baidu.com/view/95907.htm> [online].
9. <http://www.65536.cn/t/2009/VoIP-yate2-SIP.htm> [online].
10. Reference Traffic Model for User Traffic in GSM, WCDMA and Evolved Packet System (EPS).
11. Introduce to IMS, White paper, March 2007.
12. Services in the IMS ecosystem, White paper, February 2007.
13. *Analysis of VoIP Traffic*, **Liu Wenhui, Liu Fang**, Key Laboratory of Information

Processing and Intelligent Technology, Beijing University of Posts and
Telecommunications, Beijing (100876), China.

14. *SIP-based Applications in UMTS: A Performance Analysis*, **Maria Isabel Pous**,

Dirk Pesch, and **Gerry Foster**, Ireland.

15. *Modelling Approach for VoIP Traffic Aggregations for Transferring Tele-traffic*

Trunks in a QoS enabled IP-Backbone Environment, **Jörn Seger**, Faculty for
Electrical Engineering and Information Technology Department of Electronic
Systems and Switching University of Dortmund.

Appendixes

The Code

Average Holding Time

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Collections;

namespace BaiyuAnalyzer
{
    class MetaData
    {
        private HashSet<double> holdingSet;

        public MetaData()
        {
            holdingSet = new HashSet<double>();
        }

        public void updateInfo(double duration)
        {
            holdingSet.Add(duration);
        }

        private double getSetMedianVal()
        {
            List<double> holdingList = holdingSet.ToList();
            if (holdingSet.Count % 2 == 0)
            {
                int index = holdingSet.Count / 2;
                return (holdingList.ElementAt(index) + holdingList.ElementAt(index + 1)) /
2;
            }
            else
            {
                int index = (holdingSet.Count + 1) / 2;
                return holdingList.ElementAt(index);
            }
        }
    }
}
```

```
    }  
}  
  
#region properties  
public double MinHoldingTime  
{  
    get  
    {  
        return holdingSet.Min();  
    }  
}  
  
public double MaxHoldingTime  
{  
    get  
    {  
        return holdingSet.Max();  
    }  
}  
  
public double MedianHoldingTime  
{  
    get  
    {  
        return getSetMedianVal();  
    }  
}  
  
public double AvgHoldingTime  
{  
    get  
    {  
        if (holdingSet.Count == 0)  
        {  
            return 0;  
        }  
        else  
        {  
            return holdingSet.Average();  
        }  
    }  
}
```

```

    }

    #endregion
}

class AvgHoldingTmAnalyzer : Analyzer
{
    public AvgHoldingTmAnalyzer(string fileName)
        : base(fileName)
    { }

    public override void analyze()
    {

        FlowReader reader = new FlowReader(fileName);
        List<Flow> flowList = reader.readFlow();

        ArrayList list = new ArrayList(41);
        for (int i = 0; i < 41; ++i)
        {
            list.Add(new MetaData());
        }

        foreach (Flow flow in flowList)
        {
            if (!IsFlowValid(flow))
            {
                continue;
            }
            int index;
            getTimeIndex(flow, out index);
            ((MetaData)list[index]).updateInfo(flow.HoldingTime);
        }

        FileStream file = new FileStream("holding_info.txt", FileMode.OpenOrCreate,
FileAccess.Write);
        StreamWriter writer = new StreamWriter(file);
        writer.Write("Avg Time           Median Val           Max Time
Min Time\n");
        foreach (MetaData data in list)
        {
            writer.Write(data.AvgHoldingTime);

```

```

        writer.Write(" ");
        writer.Write(data.MedianHoldingTime);
        writer.Write(" ");
        writer.Write(data.MaxHoldingTime);
        writer.Write(" ");
        writer.Write(data.MinHoldingTime);
        writer.Write("\n");
    }
    writer.Close();
    file.Close();
}

private void getTimeIndex(Flow flow, out int index)
{
    DateTime startDate = new DateTime(2009, 3, 23, 10, 0, 0);
    DateTime time = flow.StartTime;
    float holdingTime = flow.HoldingTime;
    TimeSpan span = time.Subtract(startDate);

    index = (int)span.TotalHours;
}

private bool IsFlowValid(Flow flow)
{
    if (flow.Type != "VoIP Session" || flow.State != "Completed")
        return false;
    if (flow.HoldingTime < 0.000001 && flow.HoldingTime > -0.000001)
        return false;
    DateTime startDate = new DateTime(2009, 3, 23, 10, 0, 0);
    DateTime endTime = new DateTime(2009, 3, 25, 3, 0, 0);
    DateTime flowTime = flow.StartTime;

    if (flowTime.CompareTo(startDate) < 0 ||
        flowTime.CompareTo(endTime) >= 0)
    {
        return false;
    }
    return true;
}
}
}

```

Active session analyze

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
using System.IO;

namespace BaiyuAnalyzer
{
    class ActiveSessionAnalyzer : Analyzer
    {
        public ActiveSessionAnalyzer(string fileName) : base(fileName)
        {
        }

        public override void analyze()
        {
            Hashtable activeTable = new Hashtable();
            for (int i = 0; i < 5; ++i)
            {
                ArrayList activeList = new ArrayList();
                for (int j = 0; j < 41; ++j)
                {
                    activeList.Add(0);
                }
                activeTable.Add(i, activeList);
            }

            FlowReader reader = new FlowReader(fileName);
            List<Flow> flowList = reader.readFlow();
            foreach (Flow flow in flowList)
            {
                if (!IsFlowValid(flow))
                    continue;
                int index = Util.getIndexOfSessionType(flow.Type);
                ArrayList currentList = (ArrayList)activeTable[index];
                int startIndex;
                int endIndex;
```

```

        getTimeIndex(flow, out startIndex, out endIndex);
        if (endIndex > 40)
        {
            endIndex = 40;
        }

        for (int i = startIndex; i <= endIndex; ++i)
        {
            int val = Convert.ToInt32(currentList[i]);
            currentList[i] = ++val;
        }
    }

    FileStream file = new FileStream("Active.txt", FileMode.OpenOrCreate,
FileAccess.Write);
    StreamWriter writer = new StreamWriter(file);
    for (int i = 0; i < 5; ++i)
    {
        ArrayList list = (ArrayList)activeTable[i];
        foreach (int j in list)
        {
            writer.Write(j);
            writer.Write(" ");
        }
        writer.Write("\n");
    }
    writer.Close();
    file.Close();

}

private void getTimeIndex(Flow flow, out int firstIndex, out int secondIndex)
{
    DateTime startDate = new DateTime(2009, 3, 23, 10, 0, 0);
    DateTime time = flow.StartTime;
    float holdingTime = flow.HoldingTime;
    TimeSpan span = time.Subtract(startDate);

    firstIndex = (int)span.TotalHours;
    secondIndex = firstIndex + (int)(holdingTime / 3600);
}

```

```
    }

    private bool IsFlowValid(Flow flow)
    {
        DateTime startDate = new DateTime(2009, 3, 23, 10, 0, 0);
        DateTime endTime = new DateTime(2009, 3, 25, 3, 0, 0);
        DateTime flowTime = flow.StartTime;

        if (flowTime.CompareTo(startDate) < 0 ||
            flowTime.CompareTo(endTime) > 0)
        {
            return false;
        }
        return true;
    }
}
```