



The Study of MPLS VPN Billing Model

By

Ms. Vareeporn Pikulthong

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Telecommunications Science
Assumption University

October, 2003

The Study of MPLS VPN Billing Model

By

Ms. Vareeporn Pikulthong



**Submitted in Partial Fulfillment of the
Requirement for the Degree of
Master of Science in
Telecommunications Science
Assumption University**

October , 2003

The Faculty of Science and Technology


Master Thesis Approval

Thesis Title The Study of MPLS VPN Billing Model

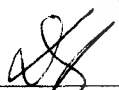
By Ms. Vareeporn Pikulthong
Thesis Advisor Dr. Surat Tanterdtid
Academic Year 1/2003


The Department of Telecommunications Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **TS7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Science.

Approval Committee:

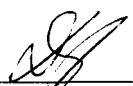

(Dr. Surat Tanterdtid)
Advisor

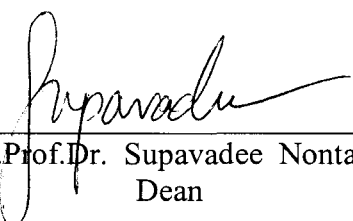

(Asst. Prof. Dr. Chanintorn J. Nukoon)
Committee Member


(Asst. Prof. Dr. Dobri Batovski)
Committee Member


(Asst. Prof. Dr. Surapong Auwatanamongkol)
Commission of Higher Education
University Affairs

Faculty Approval:


(Asst. Prof. Dr. Dobri Batovski)
Program Director


(Asst. Prof. Dr. Supavadee Nontakao)
Dean

October / 2003

ACKNOWLEDGEMENT

I would like to express my profound gratitude to my advisor Dr. Surat Tanterdtid for his invaluable advice, excellent suggestions, and constant encouragement during the thesis period. Without his guidance and help, this thesis would not have been possible. Thanks also go to Mr. Thitiroj Rojanathanes for his suggestion of programming development.

My sincere thanks also go to Asst.Prof.Dr.Chanintorn Jittawiriyakoon and Asst. Prof. Dr.Dobri Atanasov Batovski for their kind consideration to serve as my examination committee members, especially, their valuable comments and suggestions towards completion of this thesis.

Finally, I would like to take this opportunity to express my deepest gratitude to my beloved parents and friends for their consistent support, encouragement and love. This thesis is dedicated to them.

ABSTRACT

Nowadays, Internet and VPN network is seeing more popular use by users on the network. That is because they can serve the customer need in both the private network and more secure data transmission on the network. There are many types of VPN network which the user can choose that is the most appropriate type that is compatible with their needs.

Anyway, VPN network has some limitations of usage. The new solution is proposed to reduce the limitations of the network. MPLS technology is proposed for use on VPN network. When we apply MPLS Technology to VPN Network, it can improve routing distribution and quality of services on the network. But a limitation of VPN network is about dealing with billing system. The current billing system is based on the fixed-rate cost of usage; which is monthly fee or pre-paid billing for a dial-up account user. So the new billing system is proposed for reducing the limitation in billing. New billing system is based on customer usage. Customer who has high usage will pay more than customer who has low usage, while customer is paying the same monthly fee in current billing system even if they have low usage.

The computerized system is developing to help in classifying and analyzing customer usage on the network. Some calculation is shown to prove that customer can gain the different benefit from the current billing system. It's unfair for a customer who has low network usage that he pays at the same rate as a customer who has high network usage. Fairness is given by the proposed system because their payment is based on their usage. The proposed system is appropriate for customers who have low network usage

but it's not appropriate with customer who has high network usage, because their expense is increasing. Comparison between the current billing system in VPN network and new billing system that is proposed on MPLS VPN network is done. The difference in many aspects of the current billing and proposed system is discussed.

Finally we conclude that this billing system is appropriate for a customer who has low network usage than a customer who has high network usage. We can gain more benefits from this system such as fairness, full function of network equipment usage and can adapt to other IP networks, etc. Finally, recommendation for further research is shown in the last chapter.

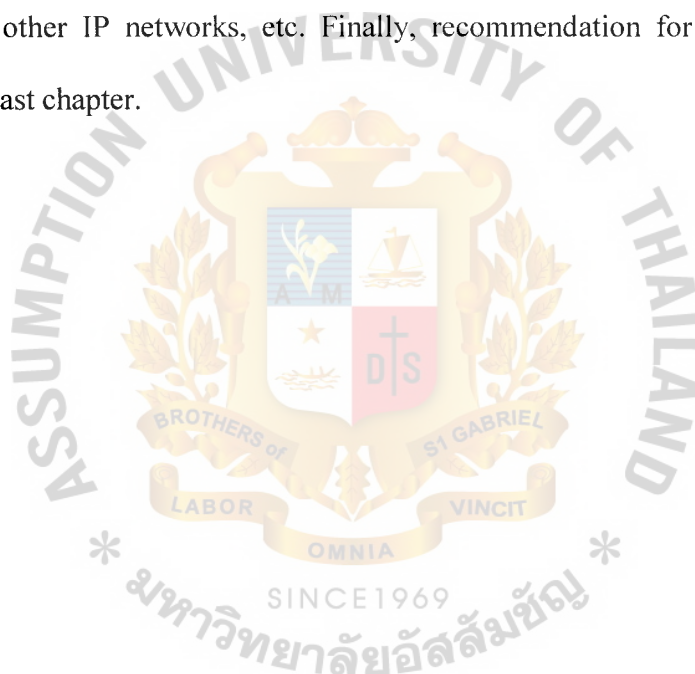


TABLE OF CONTENTS

CONTENTS	PAGE
ACKNOWLEDGMENTS	i
ABSTRACT	ii
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Objectives	2
1.3 Expected Benefits	3
1.4 Background of Technology	3
1.4.1 What is VPN?	4
CHAPTER 2 PRINCIPLE OF MPLS VPN	8
2.1 Overview of MPLS VPN Technology	8
CHAPTER 3 LITERATURE REVIEW	14
CHAPTER 4 PROBLEM STATEMENT	16
CHAPTER 5 TOOLS AND METHODOLOGY	18
5.1 Tools	18
5.2 Methodology	19
5.3 Graphic User Interface	25
CHAPTER 6 PRESENTATION AND CRITICAL DISCUSSION OF RESULTS	52

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS FOR

FURTHER RESEARCH 62

7.1 Conclusions 62

7.2 Recommendations for Further Research 63

LIST OF FIGURES iv

REFERENCES vi

APPENDIX A viii

- Abbreviations viii
- Table 6-1 : Byte Transfer Usage Calculation base on existing billing system ix
- Table 6-2 : Byte Transfer Usage Calculation base on new billing system x
- Source Code: Filter Script xi
- Source Code: Summary Script xviii
- Source Code: Port Script xxii
- Source Code: GraphByte Script xxvi
- Source Code: Graphport Script xxxii
- Source Code: Graphportpie Script xxxix
- Data Dictionary of Database xlv

CHAPTER 1:

INTRODUCTION

1.1 Overview

Life on the Internet started ages back. Since the first cable was established in 1858, it has kept growing time by time. Nowadays, it is known and used widely all over the world. Because it is fast and easy to communicate with others across the world, so every kind of business will have it implemented. From now on, the Internet will spread to the small business or even in home use.

Showing good growth, Internet then has become one kind of business. There are many ISPs (Internet Service Provider) established in the last 10 years to provide the Internet service to the users. Recently, the billing model of the Internet is divided to two types- Prepaid and Monthly Fee. In the Prepaid system, users have to pay for an Internet package before use. But in the Monthly Fee system, users can have access to the Internet before and pay the fee at the end of the month.

This thesis presents the study of billing model on MPLS VPN network. The new billing model, that is more compatible with MPLS VPN technology, will be proposed. Moreover, the new billing model will be another choice for customer who does not use high bandwidth.

This section will show the objective of this thesis. The next one will show the expected benefits that we will get from this thesis. Finally, the basics of VPN technology will be described.

1.2 Objective

The purpose is to study, evaluate and analyze the MPLS VPN Technology billing model which is compatible with the rapid change in VPN Technology and compare with the current VPN billing model.

The objectives of this thesis are:

- To study and analyze the current billing model that applies to the existing VPN Technology.
- To study the new billing model in MPLS VPN network.
- To improve the current billing method to make it compatible with the MPLS VPN network.
- To compare benefits and limitations of new billing method and the current billing method in VPN network.

1.3 Expected Benefits

The expected benefits that we will get from this thesis are

- To study VPN network and trend of this technology.
- To study MPLS and MPLS VPN technology.
- To study Accounting solution for IP network.
- To study tool for Network Utilization for ISP.
- To implement Traffic Analyzer on IP Networking.
- To implement new trend of usage-based Billing.
- Implementing new model of Accounting Based on BGP/MPLS VPN Technology.

1.4 Background of Technology

Nowadays, Internet and VPN network is increasingly important in Telecommunication network. The customer wants to keep his data private and more secure. While current technology cannot satisfy customer need and in terms of technology, they still have limitations that should be improved. So, there are rapid changes in VPN technology. MPLS technology is a technology that is applied in VPN network. Before discussing the technology of MPLS VPN, the summary of current VPN technology is described below,

1.4.1. What is VPN?

VPN stands for Virtual Private Network. Basically a VPN is a private network that uses a public network to connect to remote sites or users together. In VPN, implementation must compose of

- A main Local Area Network (LAN) that is connected from customer Head office to services provider sites; which can be dedicated connection line or can share the infrastructure in public Internet services.
- Remote access connection, which is used by remote users connecting from outside to the internal company network via provider network.

IETF classifies VPN into two models,

- The First model is based on The Customer Premise Equipment (CPE)

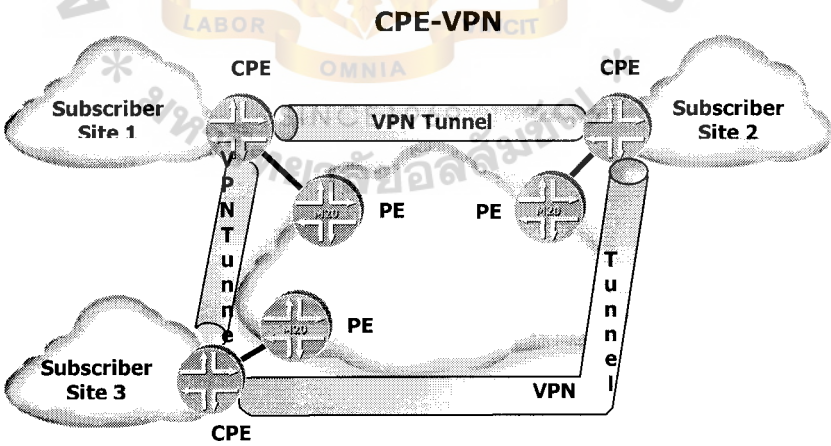


Figure 1-1: CPE-VPN model

In Figure 1, this CPE-VPN model will utilize equipment, which is located on customer sites. It can use both Layer 2 and Layer 3 technologies.

- Layer 2 Tunneling Protocol (L2TP)
- Point-to-Point Tunneling Protocol (PPTP). IPSec Tunnel. The second one is based on Network-Based (NB).

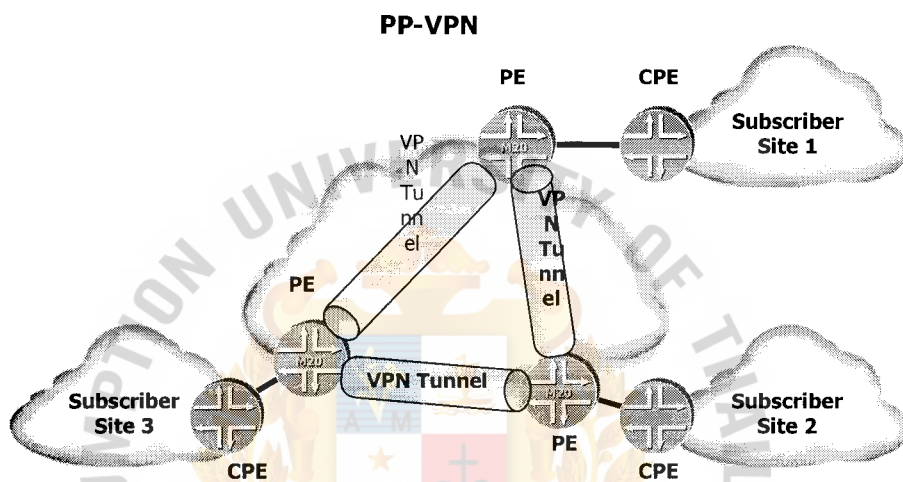


Figure 1-2: PP-VPN model

Figure 2 shows PP-VPN Model. The technologies in Layer 3 that are supported with this model can be separated to 2 solutions as below.

- Virtual Routers is used to route VPN traffic to the remote CPE.
- MPLS-Based VPNs, based on the RFC 2547bis, use Labels to switch VPN traffic between CPEs.

The VPN technology that popularly implement in existing network is,

- PPTP (Point-to-Point Tunneling Protocol) exists in Layer 2 of OSI model. It allows PPP connection to tunnel in VPN network. The encapsulation is preload in frame before sending across the network. This protocol is associated with Microsoft application.
- L2TP (Layer2 Tunneling Protocol) exists in Layer 2 of OSI model. It is developed from L2F (Layer2 Forwarding) protocol, CISCO proprietary, and best feature of PPTP to create new standard known as L2TP. The encapsulation is also preloaded in frame before sending across the network.

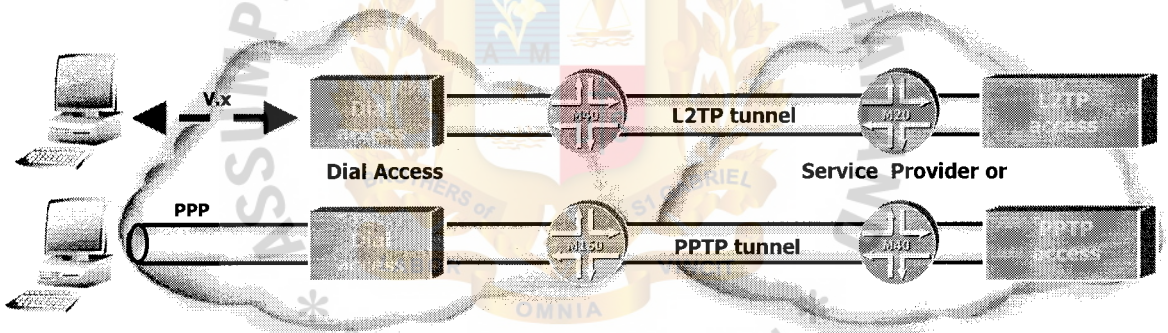


Figure 1-3: L2TP, PPTP model

In Figure 3 , a tunnel is created using either L2TP or PPTP to the remote customer site to allow access to network resources in the customer site.

- IPsec (Internet Protocol Security) exists in Layer 3 of OSI model. It is implemented in VPN network, which is more strongly concerned with secure data transfer.

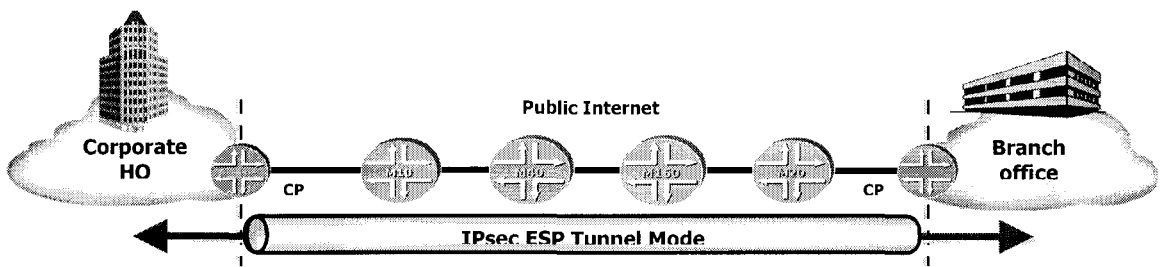


Figure 1-4: IPsec model

In Figure 4 , The packet is encapsulated and a secure tunnel is cerated before sending across the network.

Although above VPN Technology is pre-dominant today, this technology still encounters many limitations in their own technology. Several problems have been found in current VPN technology and can be summarized as following.

- Same technology must be used at both ends of VPN connection. Different technology will be used in different connections.
- Customer edge router in VPN network must deal with N routing peers, where N stands for the number of sites in VPN network.
- Configuration will be changed when a new site has been added.
- Class of services can be provided to customer and provider in IP-only layer 2.

Many VPN solutions proposed is to reduce the limitations of current VPN technology. One of the solutions is to apply MPLS technology in VPN network. This technology helps us to improve routing distribution and quality of services in VPN network.

CHAPTER 2

PRINCIPLE OF MPLS VPN

This chapter will introduce the overview of MPLS VPN technology. It will describe the main components of MPLS VPN implementation and the benefits of this technology if we apply this to the network and finally, the method with which they communicate between each node in MPLS VPN network.

2.1 Overview of MPLS VPN Technology

MPLS (Multi-protocol Label Switching) is originally used for traffic engineering and quality of services purposes. The implementation of MPLS technology in VPN network will mostly be done in the provider VPN. This thesis mainly concerns with MPLS Layer-3 VPNs which is described in “RFC 2547”. Currently, the new version is under development which is referred to 2547bis and known as “BGP/MPLS VPNs”.

The main components of BGP/MPLS VPN topology is composed of :

- Customer Edge router; the router at customer site.
- Provider Edge router; the router at the edge of provider which directly connects to customer edge router.
- Provider router; the core router in service provider cloud network.

Routing distribution on BGP/MPLS VPN can be described as below :

At customer site, the customer edge router will spread their VPN routing information to PE router nearby. These routes will then spread this information from one PE router to others, which carry the same VPN, via BGP (Border Gateway Protocol) in service provider core network. Here LSPs (Label Switching Path) will be used for forwarding packets from one PE to others. During forwarding packet, Provider routers do not need to know about customer's network in order to perform their label switching function. In the same VPN, PE routers will receive routes from other PE and then propagate the routes to the CE router, which is connected to them. Hence, the CE will also know about the networks in the remote site. Therefore, CE router will not need to learn as much routing as in the previous technology. It is good for customer because this technology will reduce responsibility on routing configuration and routing management.

The advantages that can be gained from applying BGP/MPLS VPN technology are supporting globally unique IP Address on customer site, address overlap, and VPN overlap; one customer can belong to many VPNs.

Since the BGP/MPLS VPN technology support address overlapping, so VPN routing and forwarding table (VRF) will be created on each PE router for this propose. It will be used for separating the routers, which belong to different VPNs on PE routers. For the sites belong to the same VPN and connect to the PE, a VRF will be created for each of them. But if there are multiple sites connecting to the same PE, they will share the same VRF table on that PE. And for the sites belong to multiple VPN, they will not share the VRF with others but will have their own VRF tables, which include routes from all VPNs that they are in.

Moreover, when PE routers receive BGP update from their neighbors, they might receive conflicting or overlapping routes –route belongs to different VPN. This is another problem of address overlapping. Route Distinguisher (RD) is defined to solve the addressed problem. It will be assigned to each packet in order to separate routes that belong to different VPNs on the BGP receiver side and prevent the BGP process from selecting the best route. Hence, the new IP datagram will have 12 octets address. 8 octets for Route Distinguisher and 4 octets for IP prefix. A new special 8 octets address family is called as VPN-IPv4 family.

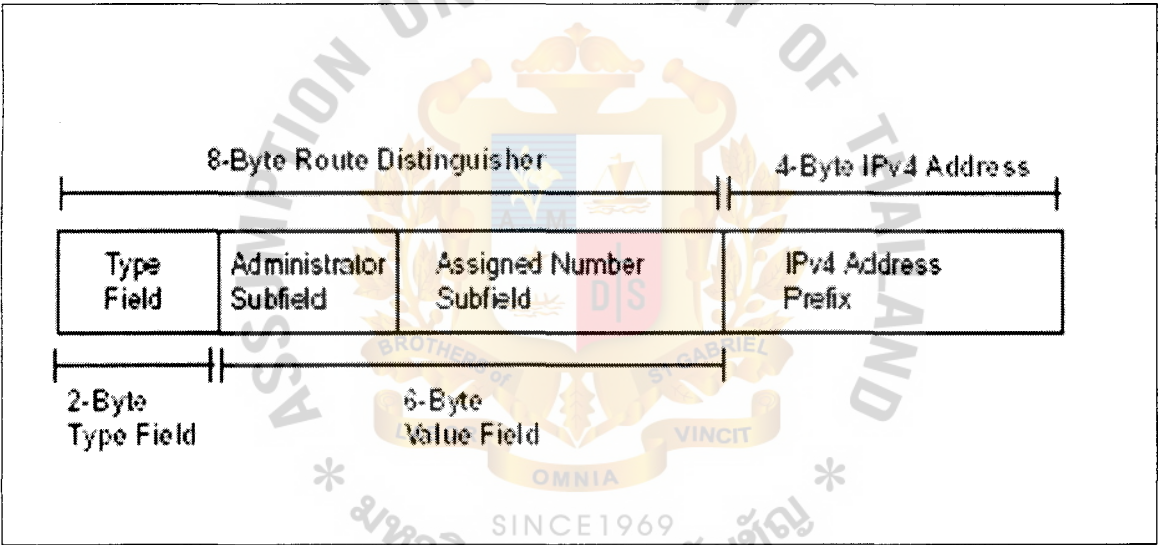


Figure 2-1: VPN-IPv4 Address

In Figure 2-1: The 8-byte RD is composed of a 2-byte Type field and a 6-byte Value field. The Type field determines the length of the Value field of two sub-fields (Administrator and Assigned Number). There are two values defined for the Type field: 0 and 1.

- For Type 0, the Administrator sub-field contains 2 bytes and the Assigned Number sub-field contains 4 bytes.
 - Adm field must contains an ASN from IANA(Internet Assigned Numbers Authority)
 - AN field is a number assigned by SP (Service Provider)
- For Type 1, the Administrator sub-field contains 4 bytes and the Assigned Number sub-field contains 2 bytes.
 - Adm field must contains an IP address assigned by IANA (Internet Assigned Numbers Authority)
 - AN field is a number assigned by SP (Service Provider)

As RD will be assigned to VRF, then every site in the same VPN will have the same RD. So all routes of that VPN will have the same distinguisher. Hence, RDs are assigned to each VPN uniquely. However, this should not mean that the site belonging to multiple VPNs will get multiple RDs. VRFs of the site need only one RD, same as those that are members of only one VPN. Hereinafter, we will describe about the way to control the advertising of routers.

As a PE router does not need to accept routes of VPN that it does not carry, so BGP extended communities will be here for controlling the distribution of routes within the provider's network. The extended community attribute has a Route Target to indicate which VPN that the route belongs to. Every customer will have a unique value for this attribute assigned. A PE router will check the Route Target value and the VPN that it carries. In the PE router, BGP process will verify the Route Target for each advertising route. If it has the same as the Route Target value of one of the VPN that it carries, the

route will be accepted. Otherwise, the route will be ignored. Then the PE routers will not waste their resource for carrying all the routes of all customers VPN.

This sample scenario can provide more understanding of BGP/MPLS VPN technology.

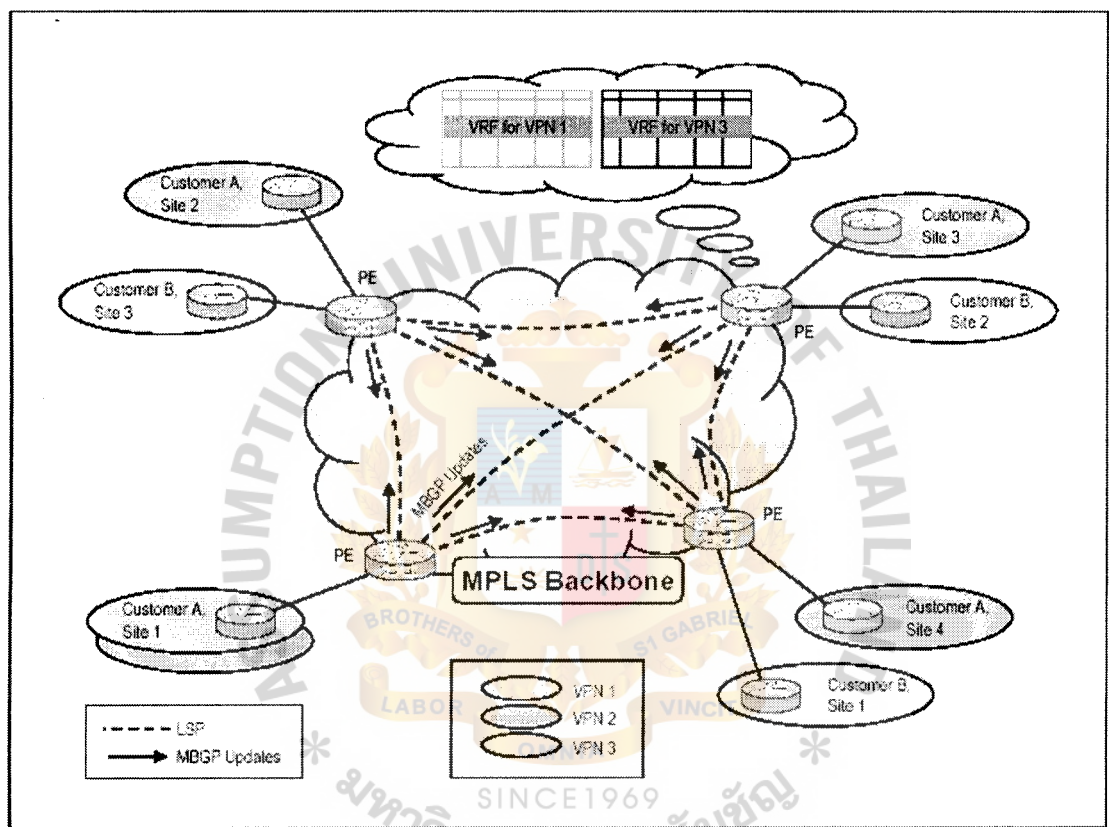


Figure 2-2: The BGP/MPLS VPN Approach

In Figure 2-2, Customer A, Site 1, lies in both VPN 1 and VPN 2. The routes are advertised from the connected PE router with one RD and two Route Target extended community attributes -one for VPN 1, the other for VPN 2. The connected PE routers also accept routes from the other PE routers, only if the routes have Route Target values equal to that value of VPN 1 or VPN 2.

For now, the provider network must have MPLS enabled, and each PE router should be capable of reaching any of the other PEs via an LSP. So PE will include the MPLS label in the BGP message while advertising a VPN-IPV4 route. And it will also set the BGP NEXT_HOP to the same as its own address. There are two MPLS labels called as outer label and inner label. These two labels will be attached anytime PE receives a packet with a destination in a remote site. After the packet has the labels, it will then be forwarded to its destination. For each labels has their own's purpose. The outer label is used to lead the LSP to the BGP NEXT_HOP. The inner label is associated with that destination. This label also identifies the destination, which was learned previously from the BGP update. Then the frame will be sent out through the connected port. But PE will do it only if that port is associated with the LSP. The frame will get label switched all the way to the remote PE. The outer label will be popped and the inner label will be examined on every hop. As the inner label is used for addressing the destination but in some cases the route summarization is done on PE, hence, the inner label will be used to determine the VRF instead. And it will reach the destination by using VRF.

CHAPTER 3

LITERATURE REVIEW

This section presents the literature survey on the topics that are related to this thesis. The first survey talks about the tool to set up price when new services is creating, Next is what is the networks architecture in the future and finally Internet Billing of department in UNSW that apply to their members.

A tool to set up price when creating new services^[9]

This paper presents a way to set up the price for a set of services. When a new service is created, it should have a good tool to set up pricing. Cash flow in, cash flow out, cost of operation, equilibrium point and expect margin are variables that must be included in this tool. A quadratic program is present as a tool in this paper. But some functions and complex price structure should be added to this program.

Building Next Generation Networks^[10]

This paper presentes a unified network architecture that can meet the trend of the business objective. Multi-services can be implemented on the network for support customer need. Architecture must be opened for multi-vendors. Low investment in implementing of new technology if its needed. And more revenue should be gained from this architecture. While meeting the business need, network simplification in implementation is met too. A unified network architecture is described in this paper.

Internet Billing^[11]

This paper describes the introduction of Internet cost of Division of information services, UNSW (The University of New South Wales, Sydney, Australia). UNSW billing the member based on connected area. The connected area can be international and domestic. The domestic can be internal UNSW usage and outside usage. This billing is intended to give fairness and help the user to control their Internet cost.



CHAPTER 4

PROBLEM STATEMENT

This chapter describes why the new billing system should be introduced. It will describes the current billing system of VPN network and the reason, why we should be proposed the new billing system.

A limitation in current VPNs network that we met is involved with billing method. In VPN network, it composes of a main private link connection to Service Provider and the other is remote access connection. From VPN network architecture, the current billing system of VPN network can be classified as following.

- **Bandwidth Based Billing**

This billing method involves charge based on link bandwidth that connects from customer office to service providers.

- **Time Based Billing**

This billing method involves charge based on time based dial up account that will be charged per hour of usage.

- **Pre-paid**

Number of access hours is pre-defined in each month. Cost is fixed.

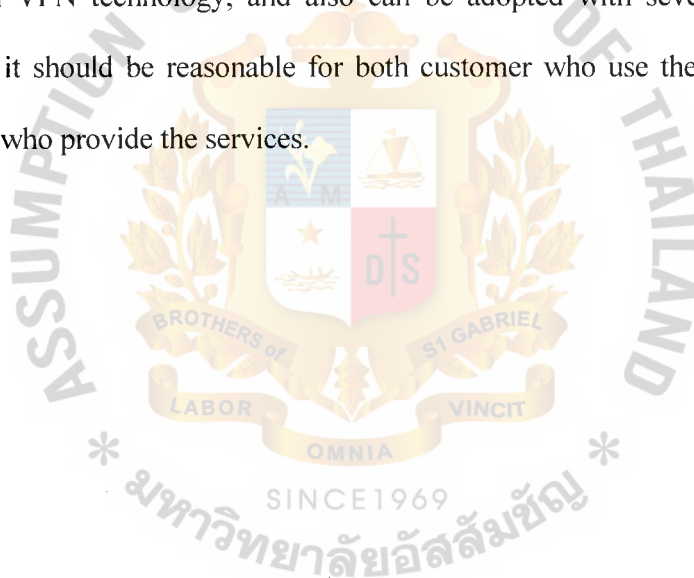
- **Pay per use**

No limited hours of access in each month. Cost is based on number of hours which user uses each month.

For rapid changes in VPN technology, accounting model that is currently used will not be compatible with the new one. The other factor is, there are many new services

occurring on IP network, such as VoIP, wireless, streaming services, broadcast media services. Currently accounting model is not compatible for the new type of services. Some of the new services consume more bandwidth than the normal services that are currently used on IP network. Another is difference of user behavior of network usage. This is unfair for both customer and ISP. The customer who uses services, which consume more bandwidth than others, should pay more. ISP should gain more revenue from customer who consumes more bandwidth than others.

The new accounting model should be implemented in which it is compatible with rapid changes in VPN technology, and also can be adopted with several new types of services. Finally it should be reasonable for both customer who use the network and the Service Provider who provide the services.



CHAPTER 5

TOOLS AND METHODOLOGY

This section presents tools that we use in implementing the new billing system; how the system is works, and finally Graph user Interface of new billing system is shown in the last section.

5.1 Tools

This billing system is more concerned with Byte transfer billing and services usage-based billing for other billing style is undergoing further development. Billing system is composed of computerized system, which is used in classified network traffic flow, summarized traffic usage and billing of usage.

To develop this billing system, tools which are used in developing of this computerized system are composed of .

5.1.1 Traffic Flow Collector System

The Application that is used as collector of network traffic is **CFLOWD Application version 2.1**. This Application is the traffic flow analysis tool. This application receives the data from the routers and keeps it in the buffer before being written into the disk.

5.1.2 Script Language

This system is a user-defined programming system. The application language that is used to generate programming script is **PERL** Language which is used to create programming script in this computerized system such as translate script, classified script and etc.

5.1.3 Database System

MySQL is the Database System, which is used to keep data of traffic flow, customer database and results of the system. Moreover **PhPMyAdmin** Application is used for managing database file. This application helps the administrator or user to maintain and manage the database easily.

5.1.4 Linux O/S

The proposed system is based on **RedHat 7.2**. Linux consumes server 's resource less and the customized script is easier to develop than Windows based system. So all the above applications are based on Linux system.

5.2 Methodology

The new billing model that we have proposed, it is divided into four distinct high-level modules. The first one is responsible for handling raw data from routers and making it available to clients on the local host. We will call it a “Receiving Module”. The second one is responsible for translating the raw data, separating them for each customer and storing them all in the database. This will work as “Translating Module”. The third one is responsible for calculating data and generating the result as html form. It is called “Calculating Module”. The last one is responsible for drawing a graph for each result. We will name it “Drawing Module”. With these four modules, we can get the information from network flows. The results will be generated in html form for easy use which will be shown later in this chapter.

Receiving Module: Basically, the data which is transferred in network cloud will be called as network flow. No matter which protocol we use or how big the data we transfer, it will be sent in form of flow. The key of the new billing model is to get flows from routers and analyze them all. Since routers have this information, then we need to export them from routers to our box. In this case, the receiving module is needed. We use CFLOWD^[12] as our daemon to receive the data flows from routers. There are three steps to collect the data which are handled by three main programs of CFLOWD. The first program, CFLOWDMUX^[12], will accept data flow packets and save them in shared memory buffers. The flow packets will arrive on UDP sockets. By using shared memory buffers, CFLOWDMUX will toggle between two packets buffers when writing packets. So the client may read the buffer that is not currently being written. With this toggling process, clients can avoid reading a buffer while it is being written by CFLOWDMUX. The second program, CFLOWD, is used for watching the shared memory packet buffer and reading a packet buffer when it becomes available. CFLOWD does not yet write raw flow data to disk. The last program, CFDCOLLECT^[12], will be in charge of retrieving data of CFLOWD and write to them to disk in ARTS form. And it also releases the shared memory buffers that were dumped to disk. The raw data file will be created every 5 minutes. In the other words, we will get the collection of 5-minute data.

Translating Module: The data we get from the Receiving Module is still unreadable. This module will translate the raw data file into the readable format. Besides, this module will separate data of each customer and store them into the proper tables. To do such a thing, it has to connect to the customer database to get some information keys in order to separate data. When we have both readable data and customer information, it will

check if each flow belongs to which customer. After matching, the usable data will be stored in each customer table and the unusable data will be ignored. The raw flow files will be flushed after translating to spare the disk space for other packets that keep coming in. Since the raw data is created every 5 minutes, so after translating if there is no other raw file left, the daemon will go to sleep mode for 5 minutes.

Calculating Model: After a while we will work with flows, now it is time to calculate the existing data, as we desire. Considering to our flow database, each customer will have their own table and each tables will have many records. Each record is definitely one real flow data. To calculate or query data from tons of records in flow database is not wise. We then summarize the flow table everyday. For further use, we summarize flow table with two different types. The first type will be summarized by customer port. If that flows use the same customer port, they will be added together. But those ports have to be in the well-known port table. Otherwise they will be considered as “others” port. The second one is based on the pair of IP addresses and port of source and destination. If the flows have the same customer port of those fields, they will be added together. By this two-types table, we can query the large amount of data efficiently. By now, we have the flow tables that are ready to use. In the very last step of this module, it will communicate with the html script and then generate the html result for on demand query.

Drawing Module: Graph is the easiest way to compare the results. That is why this module is made for. There are three kinds of graphs that are generated by this module. They are Lines, Bars and Pie graph. Lines graph will be drawn in terms of Byte Transfer. Bars and Pie graph are for Port Aggregation. All graphs will be created automatically every day or on demand manual script.

The four main modules provide us the results and necessary information. But by modules themselves they cannot show us their results. The graphic user interface then has been written for. The rest of this chapter will show the graphic result of the thesis.



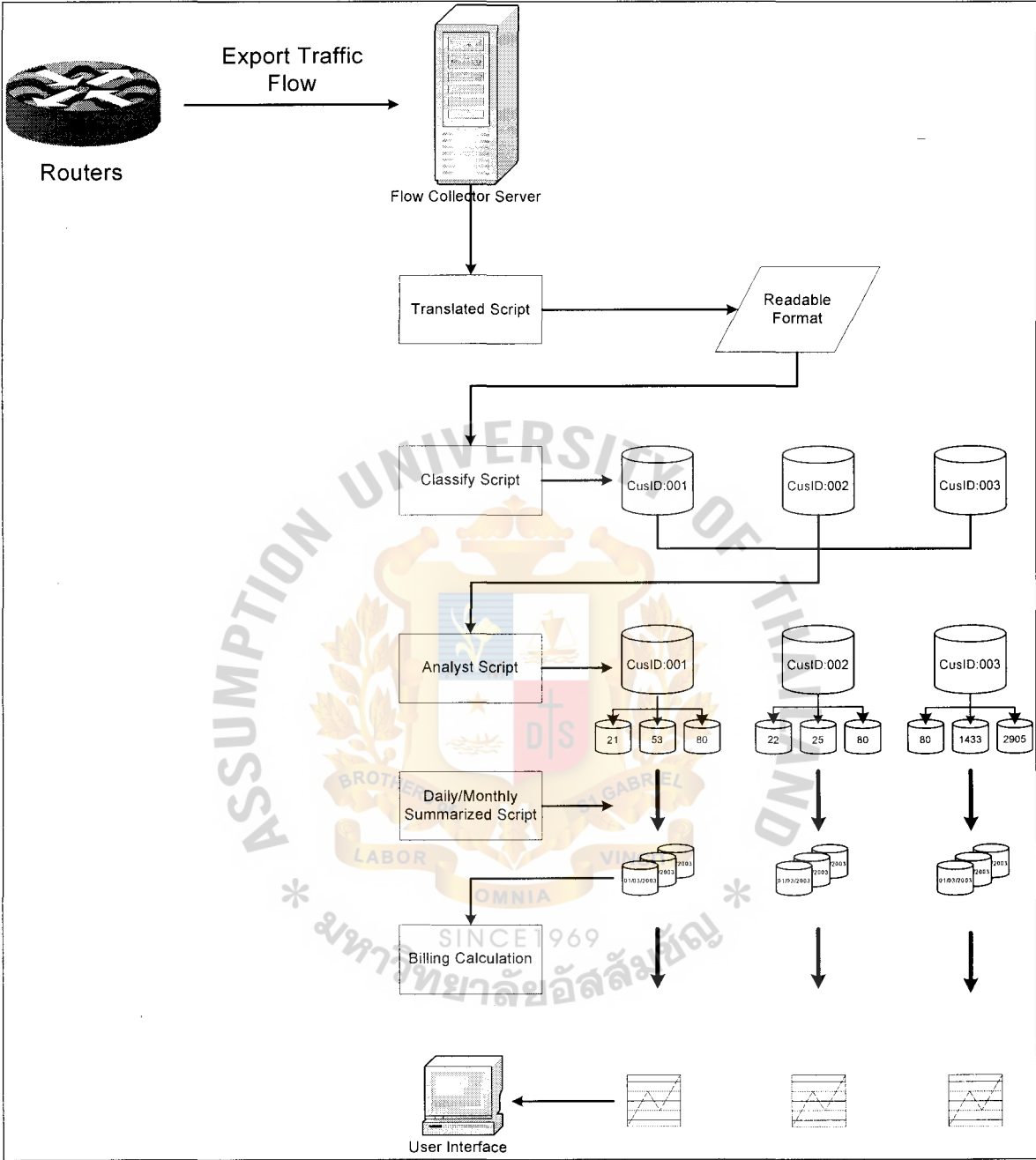


Figure 5-1. Data Flow of New Billing System

In Figure 5-1, the data flow of the new billing system is shown. Traffic data usage is exported from Router to Collector. Translating to readable format, classifying data of

each customer, analyzing bases on services port data, summarized data in each day and finally shows the results in graphic mode.



5.3. Graphic User Interface

5.3.1 .To access to the server is by using webpage.

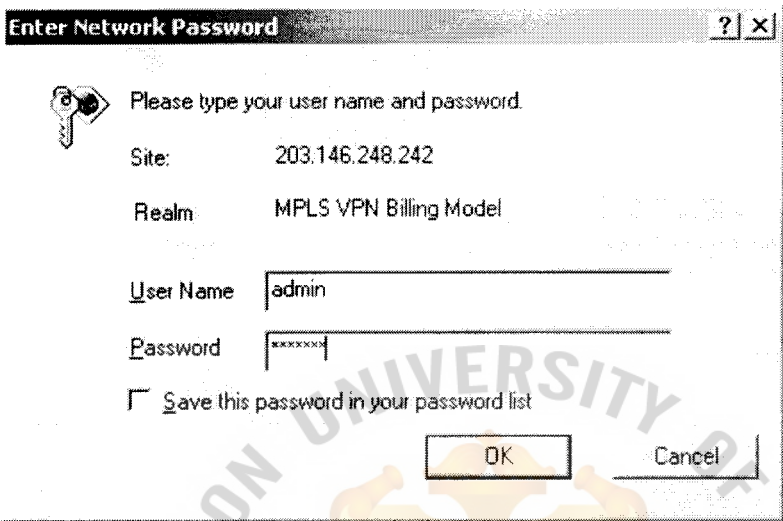


Figure 5-2: Login Windows

A login window in Figure 5-2 will show on the screen for starting access to the system. User has to enter his username and password. Only the authorized person can access the system.

5.3.2 when the user passes authorized checking at login windows, the first page will show on the screen as following:

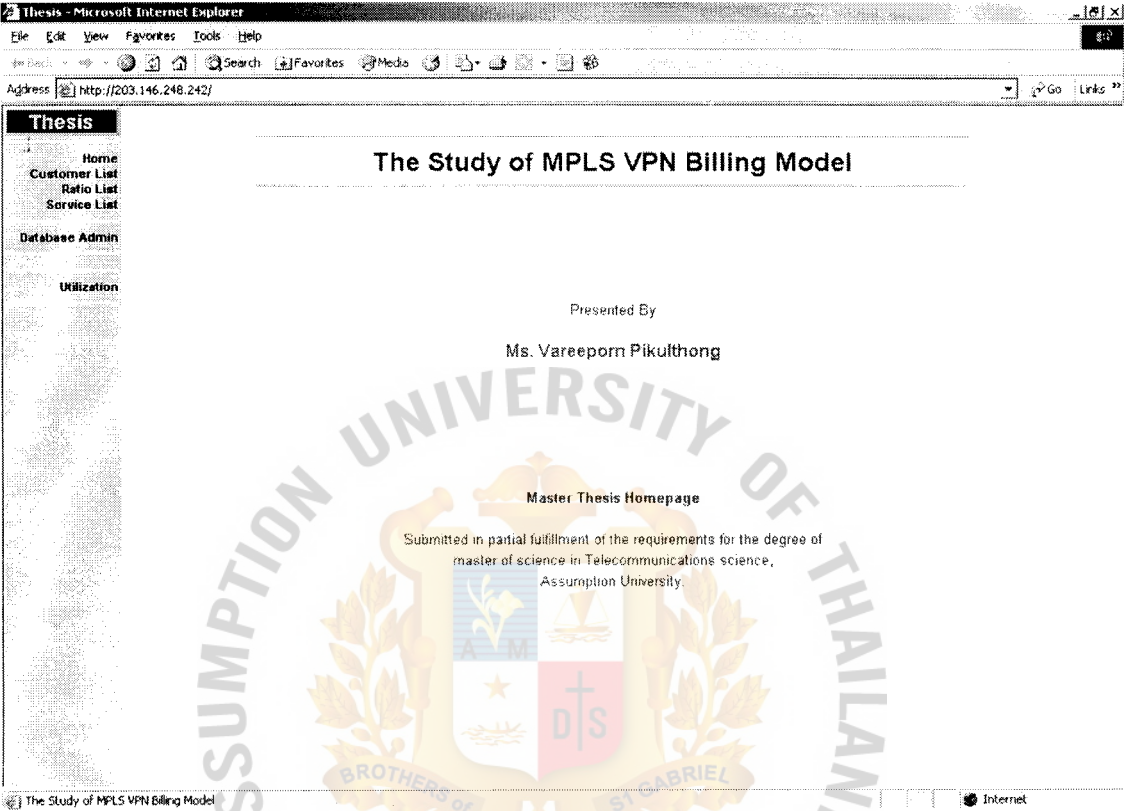


Figure 5-3: Home page

Figure 5-3 shows the first page /home page of the system. There are 3 sections of menu on the right frame.

1. Information Menu:

- “Home” menu is used for jumping back to the first page of system.
- “Customer list” menu shows the list of customers in database.
- “Ratio List” menu shows the cost ratio of each service type.
- “Services List” menu shows the services port which favors use by users in the network.

2. Administration Menu:

“Database Admin” menu is used to access to database via PhPMyAdmin application.

3. Report Menu:

“Utilization” is the menu that shows results of the system that is the summarization of usage and the other details of each customer.



5.3.3 Customer list Menu

Thesis - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Print

Address http://203.146.248.242/ Go Links

Thesis

TABLE : Customer

Home	CustomerID	CustomerName	Address	ContactPerson	EngineerName	IPDetails	ServicesTy
Customer List	DD001	Loxbit-NetFusion	11 Fl , Siripinyo Building, Sri-Ayudhya Rd., Tanon Phayathai , Rajthwee, Bkk 10400	K.Nueng Tribut ,02-201-3111	vareeporn	203.146.117.xx,203.146.102.231-232,203.146.250.64-96	Premium
Ratio List	DD002	Conner Network	606/2 Pahcachoeh,Ladyal,Chattujak,BKK,10900	K.Yedying,02-587-9747,01-844-4277	Panit	203.146.102.37-40	Economic
Service List	DD003	Sriracha Dot Net	Amper Sriracha , Chonburi	K. Wichai	Vareeporn	203.146.251.1-11	Standard
Database Admin	DD004	SCB Securities Co.,Ltd	Sintorn Building,Wireless Rd.,BKK	K.Suriya,K.Pongsak	Vareeporn	203.146.21.66	Economic
Utilization							

Customer Information Internet

Figure 5-4: Customer List Screen

Figure 5-4 shows the details of customers in the database. The shown detail is composed of Customer Name, Contact Person, Customer Address, IP address and Service Type of usage and so on. The user can only view the list of customers in database from this menu. No edit, insert and delete of customer details is allowed on this screen.

5.3.4 Ratio List Menu

Next is the list of cost ratio screen. This screen presents the details of pricing category that is used in calculating usage cost for each customer.

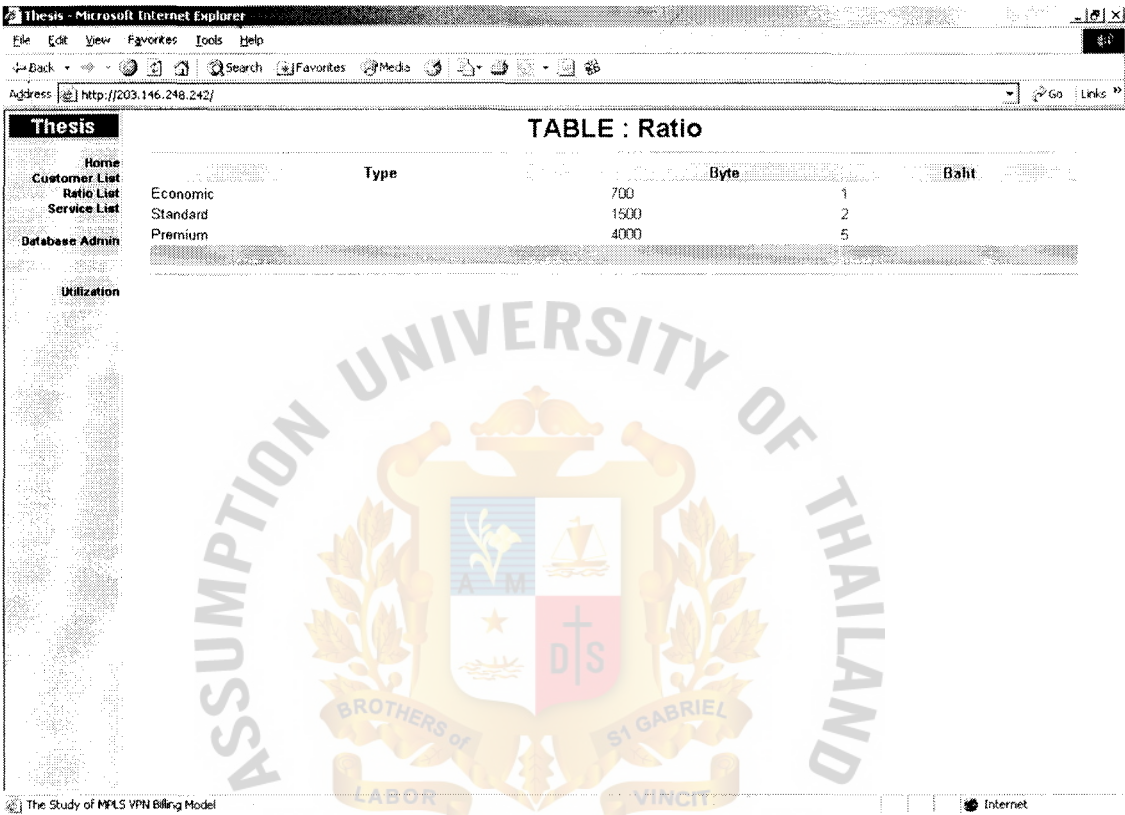


Figure 5-5: Ration List Screen

Figure 5-5, shows the cost ratio that is pre-defined in the system, All cost is applied into calculation based on services usage of customer .We can adjust or remove the cost ratio from the database via database administrator. The details of cost ratio that is shown on the screen is composed of Ratio description, Byte Transfer value that is used to calculate the cost of usage and finally cost package of each ratio.

5.3.5 Services Type Menu

Figure 5-6, shows the list of services port that are of favorite use and its descriptions. This screen will show that what is the services port type that the system will classify and show the result on the screen. Services port of usage can be Mail services (25), Web Services (80) or FTP Services (20+21) and etc.

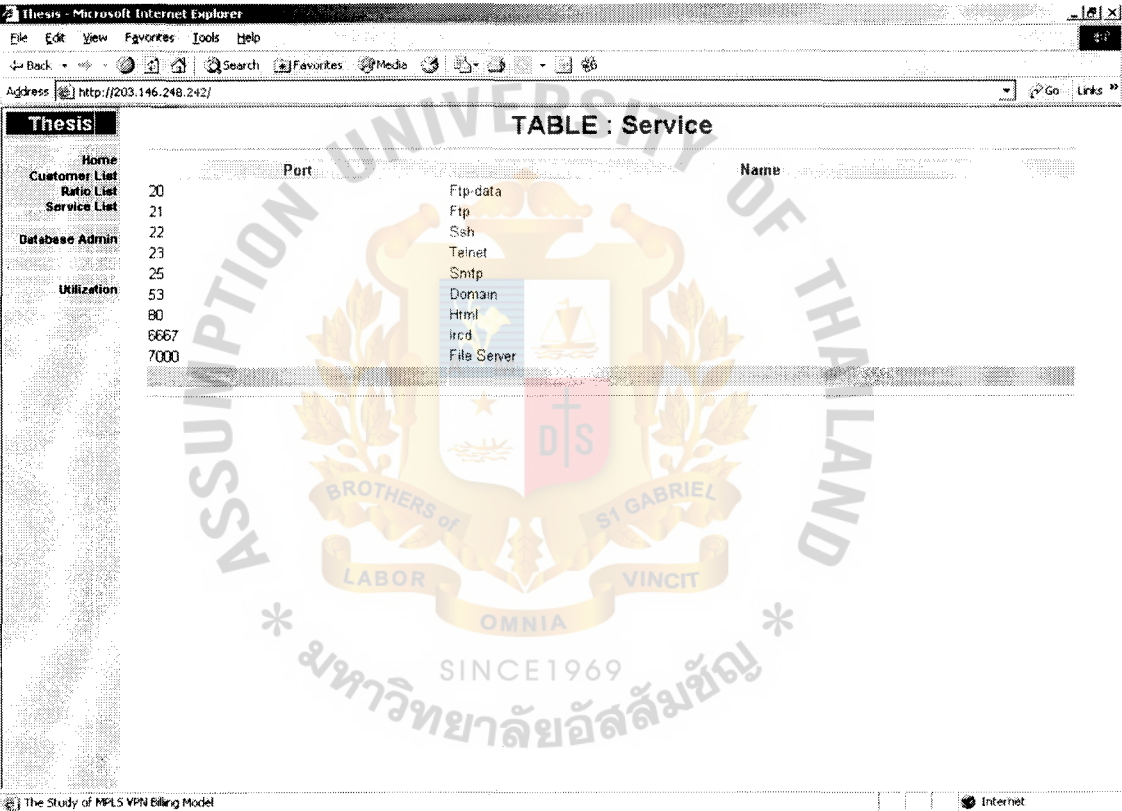


Figure 5-6: Services list Screen

5.3.6 Database Admin Menu

The main section of this system is Database Admin Menu. This menu is used to manage the database via application name called PhpMyAdmin , which help us to easily create and manage the database file.

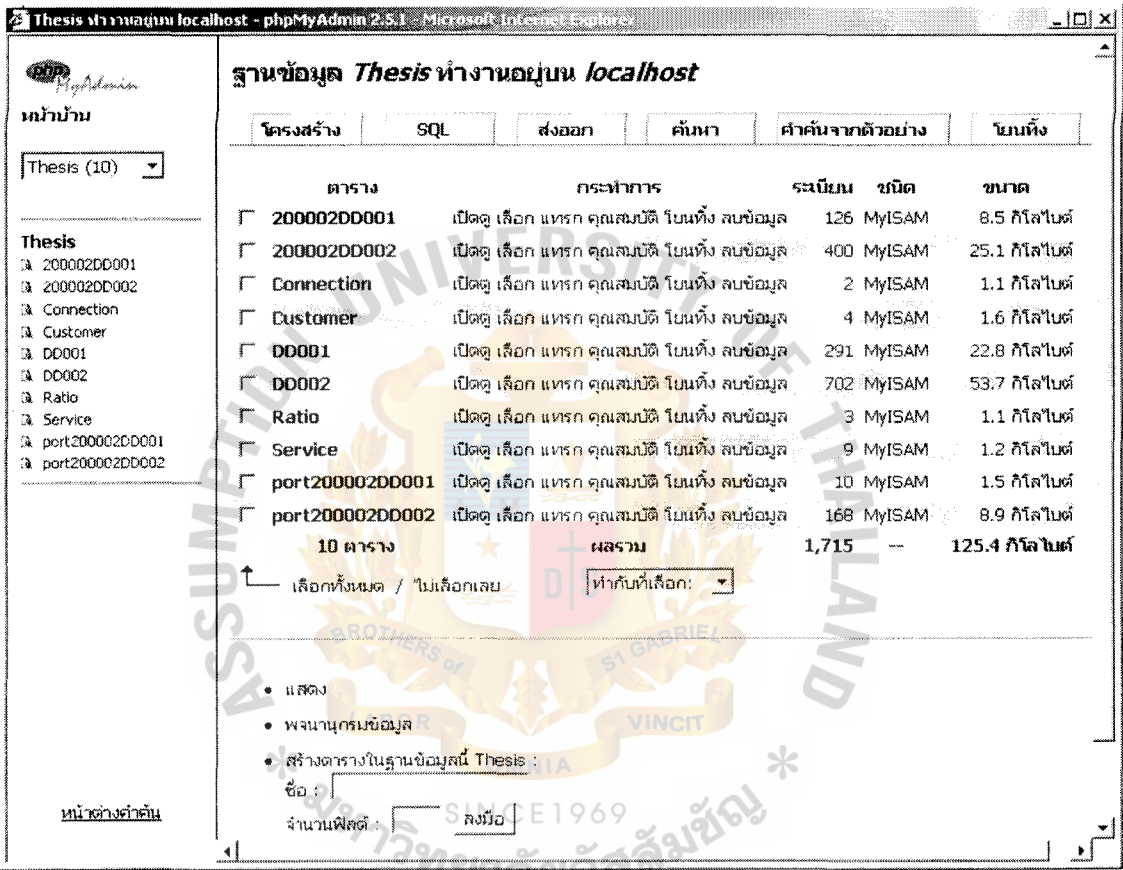


Figure 5-7: Database Management Screen

Figure 5-7,presents the main menu of database system, when user is accessing to this menu by clicking at database admin menu, next is user selects the database name that he wants to access at drop-down list box which is shown on the right side of the screen. So all tables in the database will show on the screen as in the Figure 5-7. On the main menu of database screen, at the left side of screen, will show the table name,

which was there in the database. User can access to these tables by “clicking” the mouse at the table name, The main menu of each table will show on the screen.

5.3.6.1.Customer Table Screen

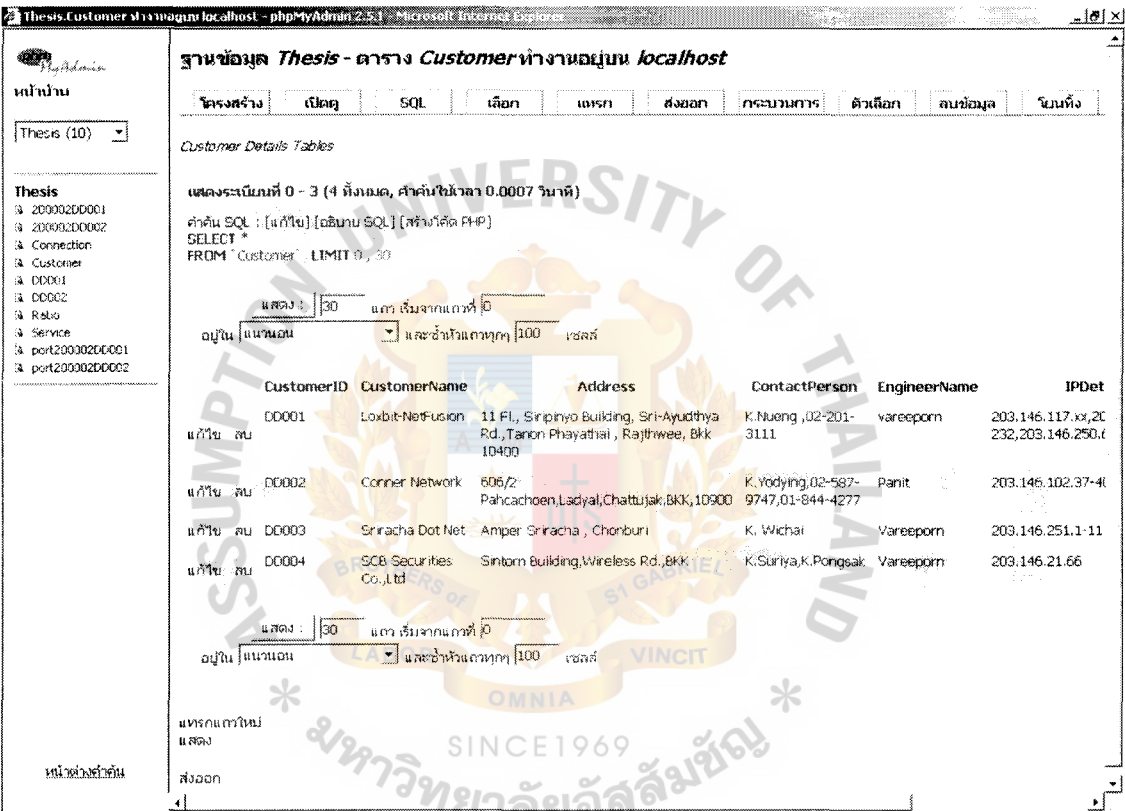


Figure 5-8: Customer Table Screen

In Figure 5-8 .The Customer Table is presented. User can access the Customer table by clicking mouse at table name, the big screen on the right will show the Customer Table main menu as in Figure 5-8. The customer detail in Customer Table is composed of Customer ID, Customer Name, Customer Address, Contact Person,

Engineer Name who takes care for each customer, IP address which is now used by customer and finally the service type that they will pay for.

User can view customer list in database by clicking at “open” sub-menu. List of customers will show as Figure 5-8.at the right side of screen. Moreover, user can edit or delete database record in this table by clicking at “edit” or “delete” button at the left side of screen in front of the data record. But for inserting the new customer record, they need to click at “insert new record” button, which is below of this screen. After inserting the new record into the database or edit the data in the table is finished, clicking “OK” button at the bottom of the screen, it completes the edit and insert process.



5.3.6.2. Connection Details Screen

Thesis.Connection ทำงานบน localhost - phpMyAdmin 2.5.1

ฐานข้อมูล Thesis - ตาราง Connection ทำงานบน localhost

โครงสร้าง เปิดดู SQL เลือกรูปแบบ แสดงผล การคำนวณ การดูเลือก ลบข้อมูล

ฟิลด์	ชนิด	ฟังก์ชัน	วางเปล่า (null)	ค่า
CustomerID	varchar(5)			DD001
Equip	varchar(15)			203.146.64.253
PortNo	varchar(10)			vlan228
IntIndex	char(3)			18

☒ บันทึก หรือ ☒ ส่งกลับ
☐ แทรกเป็นแถวใหม่ หรือ ☐ แทรกเป็นคอลัมน์ใหม่
ลงมือ ลงมือใหม่

หน้าต่างสำคัญ

Figure 5-9: Edit Screen of Connection Table

In Figure 5-9, Edit screen of Connection Table is present. This table shows the details of network equipment IP that customer is connected, the Interface Index value of interface on network equipment that customer is connected to. User can view list of connection information in database by clicking “open” sub-menu. The list of connection of each customer will be shown on the screen.

User can edit or delete database record in the table by clicking “edit” or “delete” button at the left side of screen in front of the data record. But for inserting the new record, they need to click at “insert new record” button at bottom of this screen. The Figure 5-9, shows the edit data screen that will show the details of data in from user

to modify or change. After finishing, click “OK” button at the bottom of the screen. It will return to the main menu of Connection Table.

5.3.6.3 Ratio Details Screen

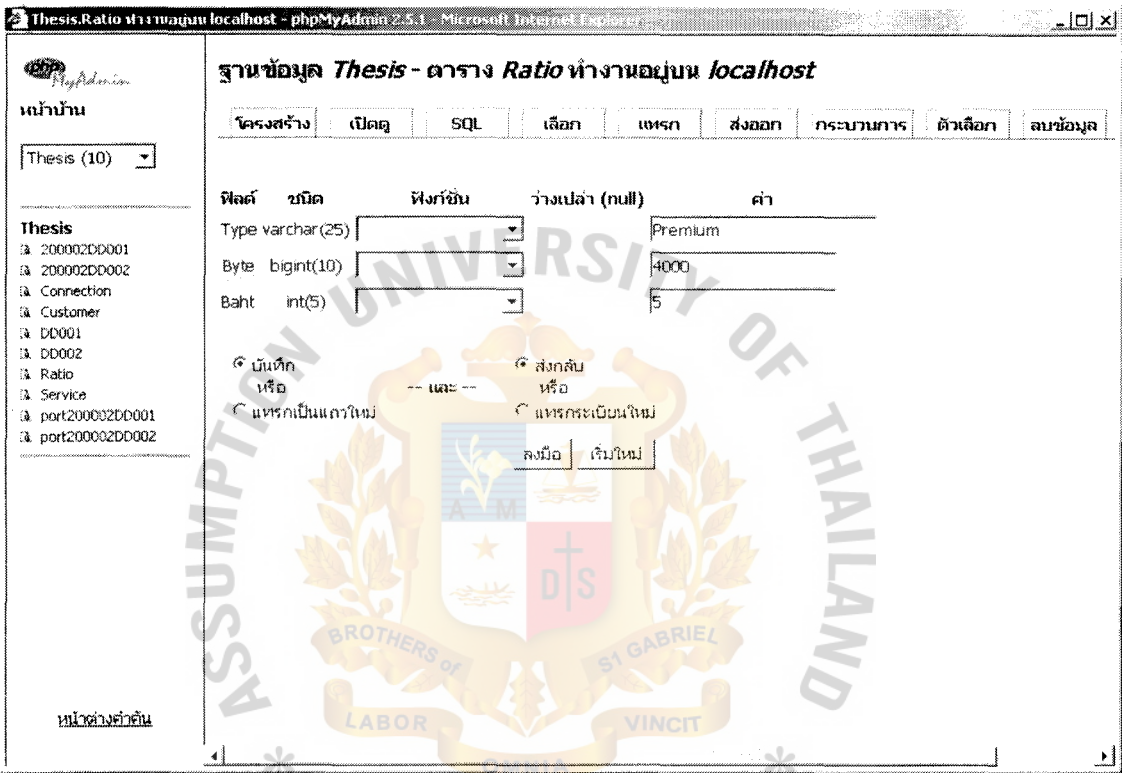


Figure 5-10: Edit Screen of Ration Table

This table keeps the data of Billing Types of each service. The data in this table is composed of Billing type description, Number of Byte Transfer that is used as the base of each billing type calculation and finally cost of each billing type.

User can insert the new record of data by clicking at “insert new record” button at bottom of main menu of this table screen. Figure 5-10 is used for editing the existing

data in the table, when editing is finished, clicking the “OK” button, will complete the process.

5.1.6.7. Services Port Screen

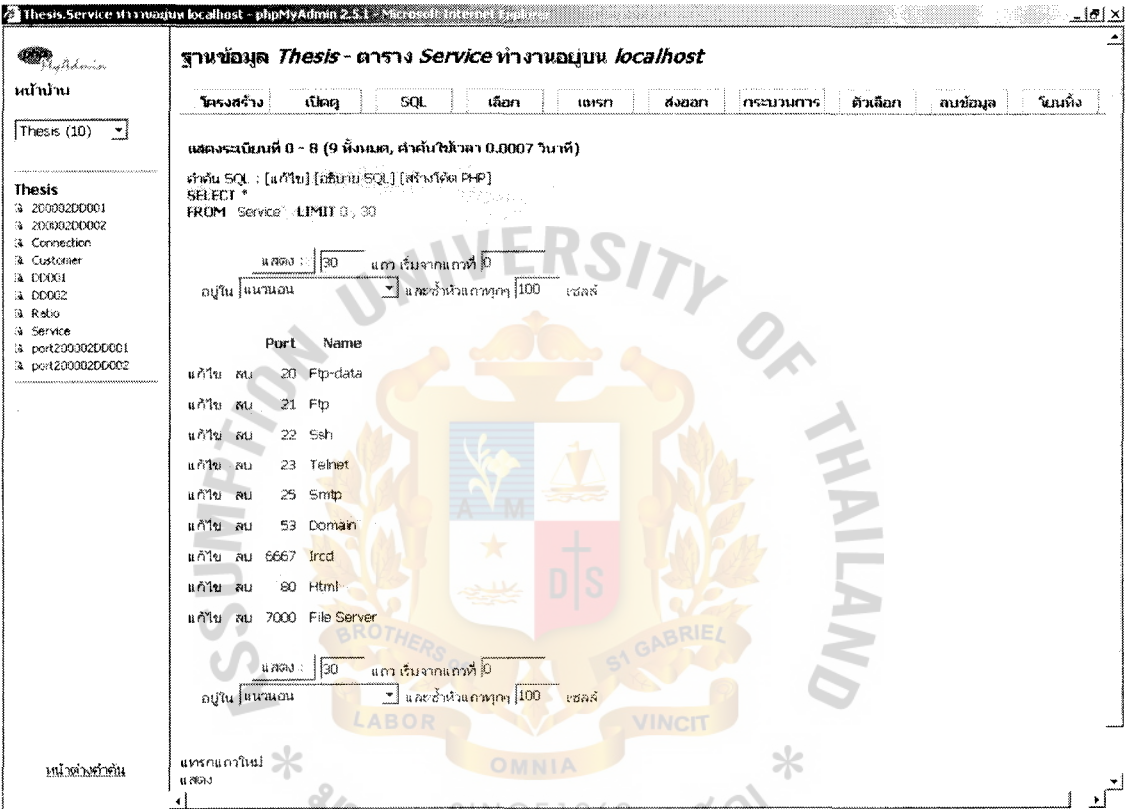


Figure 5-11: Services Type Table

In Figure 5-11, shows Services type table. This table keeps details of description and numeric number of services port that are of favorite use in the network. All services ports in this table will use as a key when the system analyses each traffic flow. Figure 5-11 shows the list of service ports in the table.

Same as above, if user wants to insert new record into the database, they need to click at “add new record” while they click at “edit” or “delete” button in front of the data

record when they want to edit or delete data in the table and click “OK” button when they finish editing data. The screen will go back to the main menu page.



5.3.6. Utilization Menu

This section presents the result of the thesis after the computerized system is finished to classify and analyze traffic flow data of each customer.

5.3.6.1 Customer List Menu

When user clicks “Utilization” menu, Figure 5-12 will show. On the screen, they will show the list of customers and other details, such as customer ID, customer name and, etc. This menu is showing the details of network usage of each customer. User start to view the result by selecting customer that they want by clicking at Customer ID or Customer name.

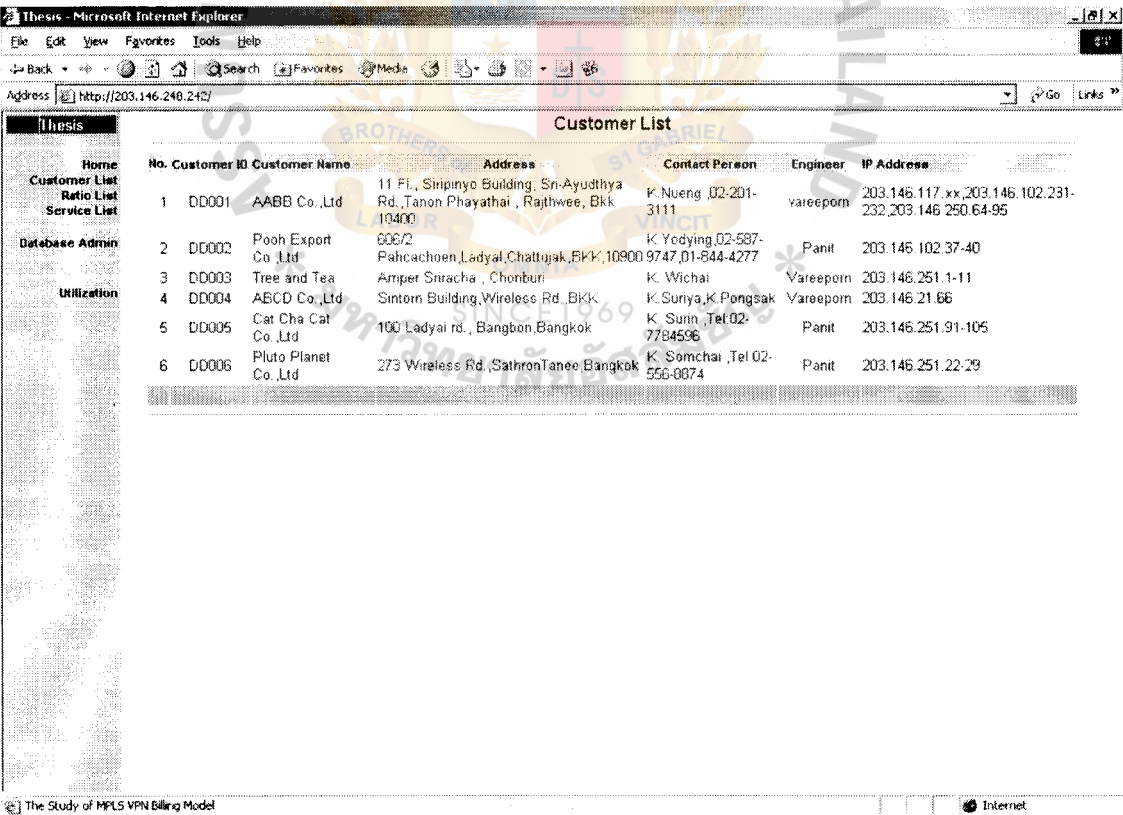


Figure 5-12: Customer list for view of usage

5.3.6.2. Customer Details Menu

Figure 5-13 is showing after customer click at Customer Name or Customer ID on this screen.

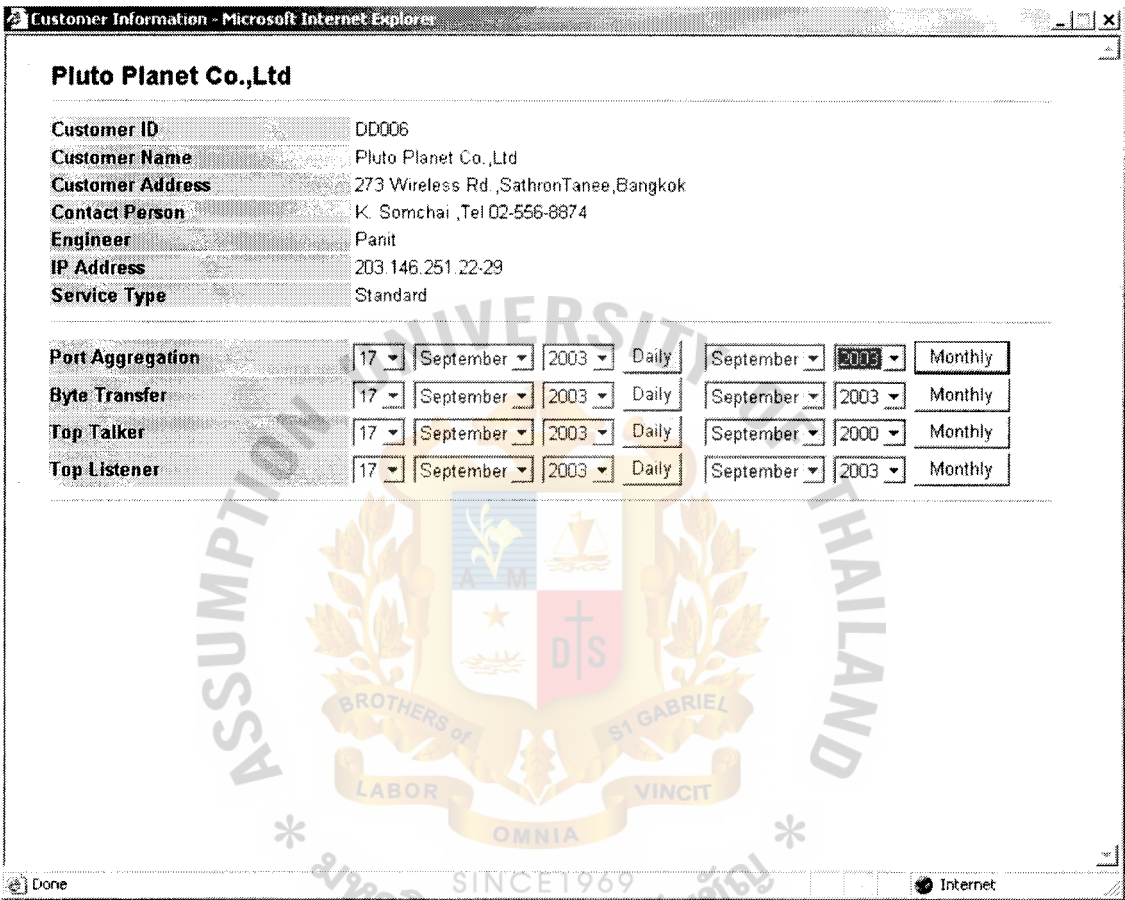


Figure 5-13: Utilization that classify in each category

Figure 5-13 shows the four sub-menus on the screen. It is composed of Port Aggregation Submenu, Byte Transfer Submenu, Top Talker Submenu and finally Top Listener Submenu. Users can select to view the results of program in each detail such as, users can views results of daily and monthly report of port aggregation, daily and monthly report of byte Transfer and cost of usage. The other is daily and monthly list of top talker and top listener.

5.3.6.3. Daily Report of Port Aggregation Screen

The first Submenu of results is Daily Port Aggregation menu. User can view daily report of port aggregation of selected customer by, specifying the period that they want to view by selecting date, month and year at drop-down list box in Port Aggregation Section. After that, he clicks at daily button to query the data of specified date. The program will query data and show the results as on Figure 5-14.

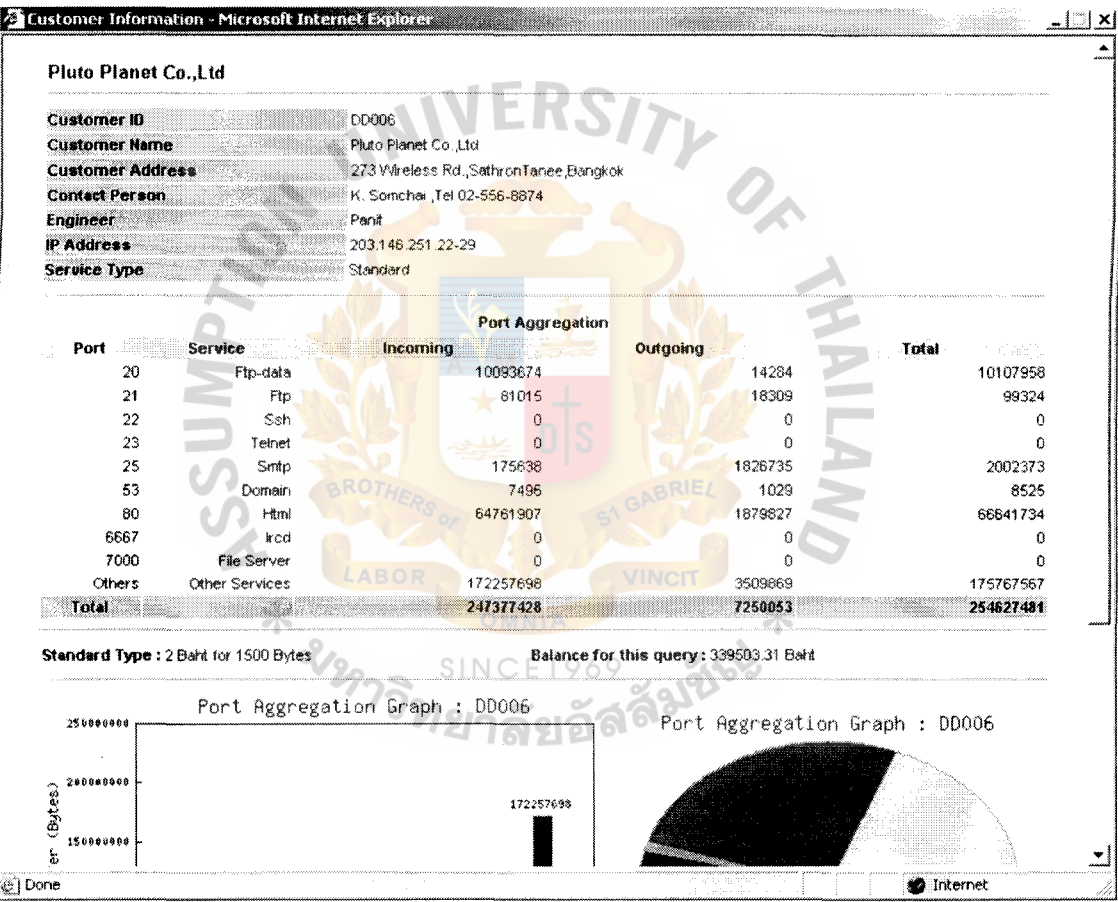


Figure 5-14: Port Aggregation results of each customer in tabular format

On the screen, result is shown about the details of Byte Transfer per each services port of usage. At the same time, the system will show the total Byte Transfer usage of both incoming and outgoing in the bottom of the table. The last information that is

shown on the table is the cost of usage. The left side is showing the Billing service type of customer and the right side is showing the total cost of usage in unit of baht. It's calculated from total byte transfer usage and Billing type service of customer.

5.3.6.3 Daily Report of Port Aggregation Screen (Continues)

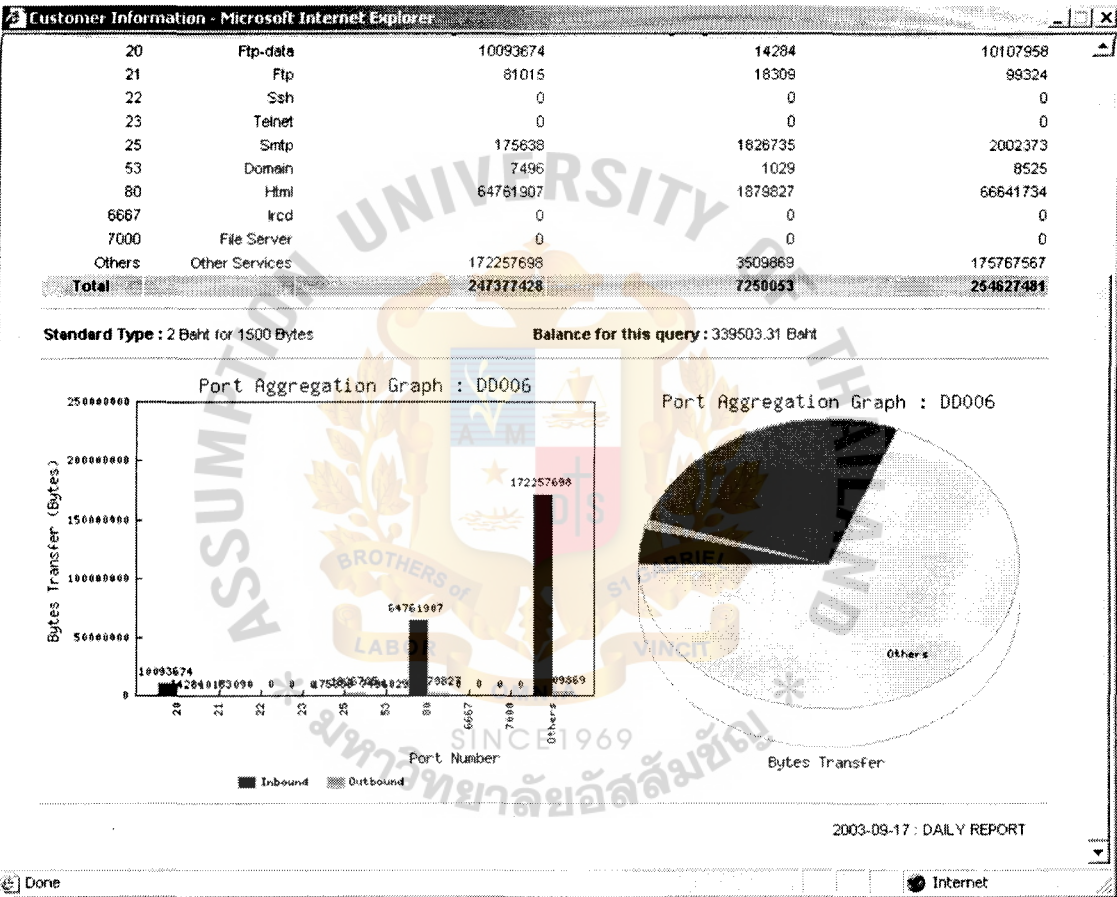


Figure 5-15: Port Aggregation results of each customer in Graph Format

Otherwise, user can view customer's usage in tabular format. The system presents daily graph of usage based on service port aggregation in the Figure 5-15, User can view graph by scrolling menu bar at the right side to the bottom of the screen.

5.5.6.4. Monthly Report of Port Aggregation Screen

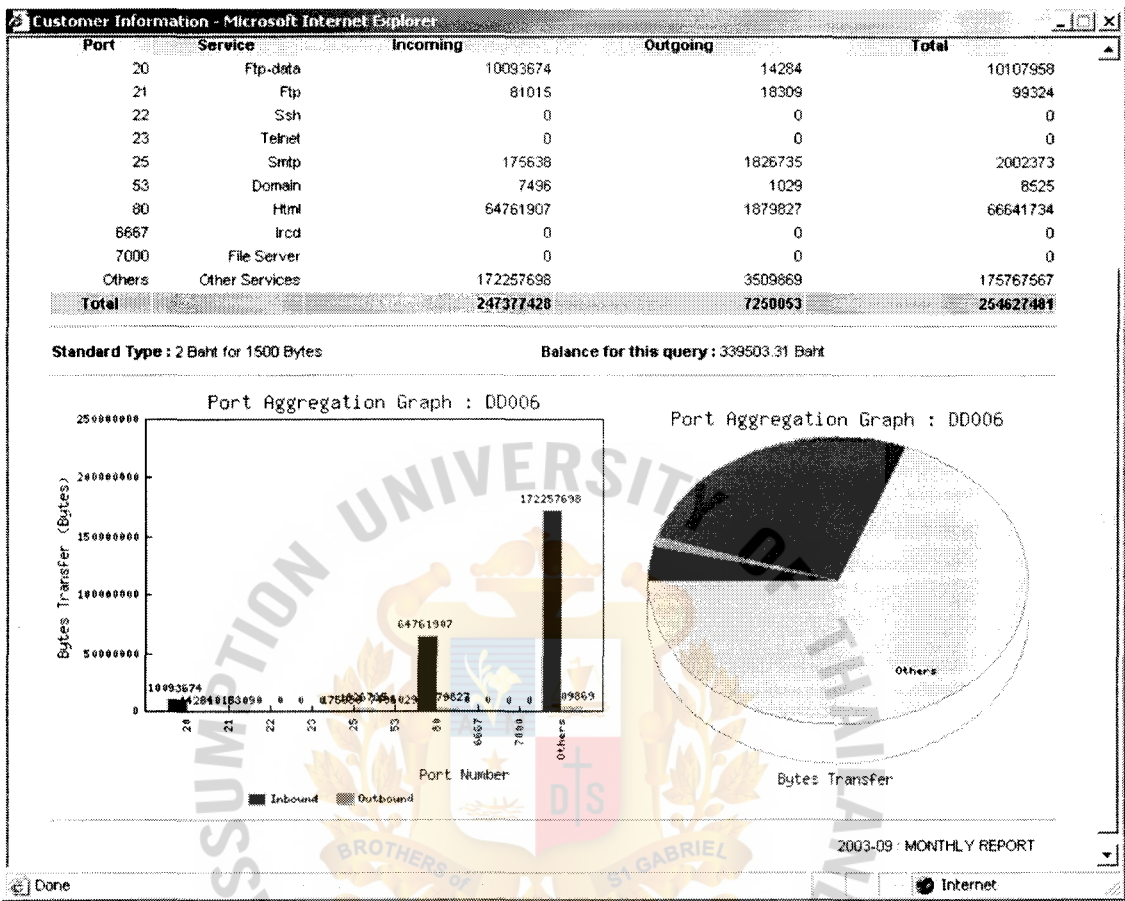


Figure 5-16: Monthly report of Port Aggregation

In Figure 5-16,shows monthly report of port aggregation screen. User can view the summarization of port aggregation usage in each month from above Figure. To access Figure 5-16, user must specify the month that they want to view the summarized data by selecting to month and year at drop-down list box, click at “monthly” button on the screen shown in Figure 5-13. After that the system will query data and show it on the screen as in Figure 5-16. The system will cumulate the service port usage of each day and store into the database. Finally, the system will

show the summarized data on the screen of monthly report. Anyway, user can see the graph of monthly report at bottom of same screen.

5.5.6.5. Daily Byte Transfers Screen

This section presents the other type of information of customer usage. It is daily Byte Transfer report.

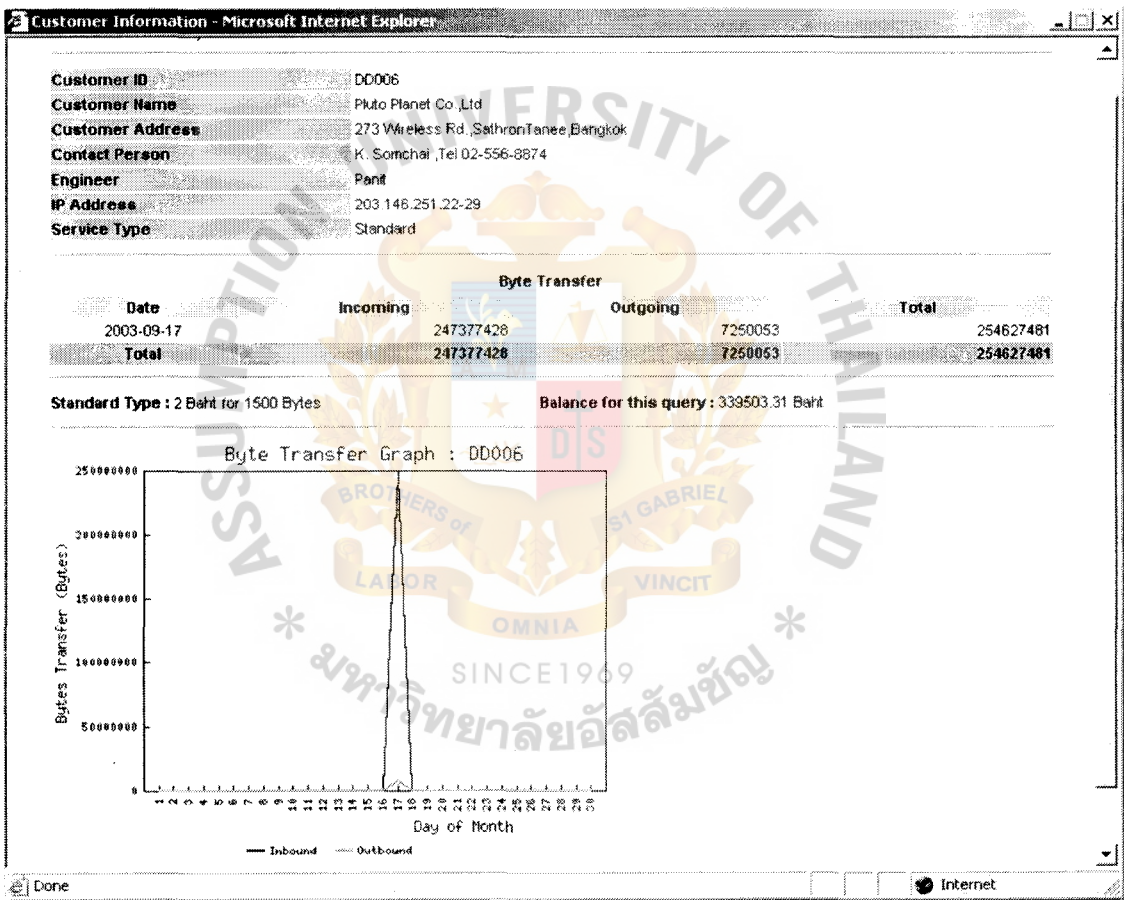


Figure 5-17: Daily summarization of Byte Transfer Format

In Figure 5-17 shows the daily report of customer usage without the other details such as shown in the report of port aggregation. On the screen, the system will summarize total Byte Transfer of Incoming Traffic and Outgoing Traffic on each day

and show it in the table and graph view. On the screen, the system will show the billing service type of customer at the left side. Cost of usage is calculated and shown on the right side, below the table.

To view daily Byte Transfer of each customer, user needs to specify the period that they want to view by selecting date, month and year at drop-down list box, after that, clicking at daily button. The program will query data and show the results as screen in Figure 5-17. Otherwise, user can view daily usage byte transfer in the graph viewed on the same page, by scrolling menu bar at the right side to the bottom of this page.

5.5.6.6. Monthly Byte Transfer Screen

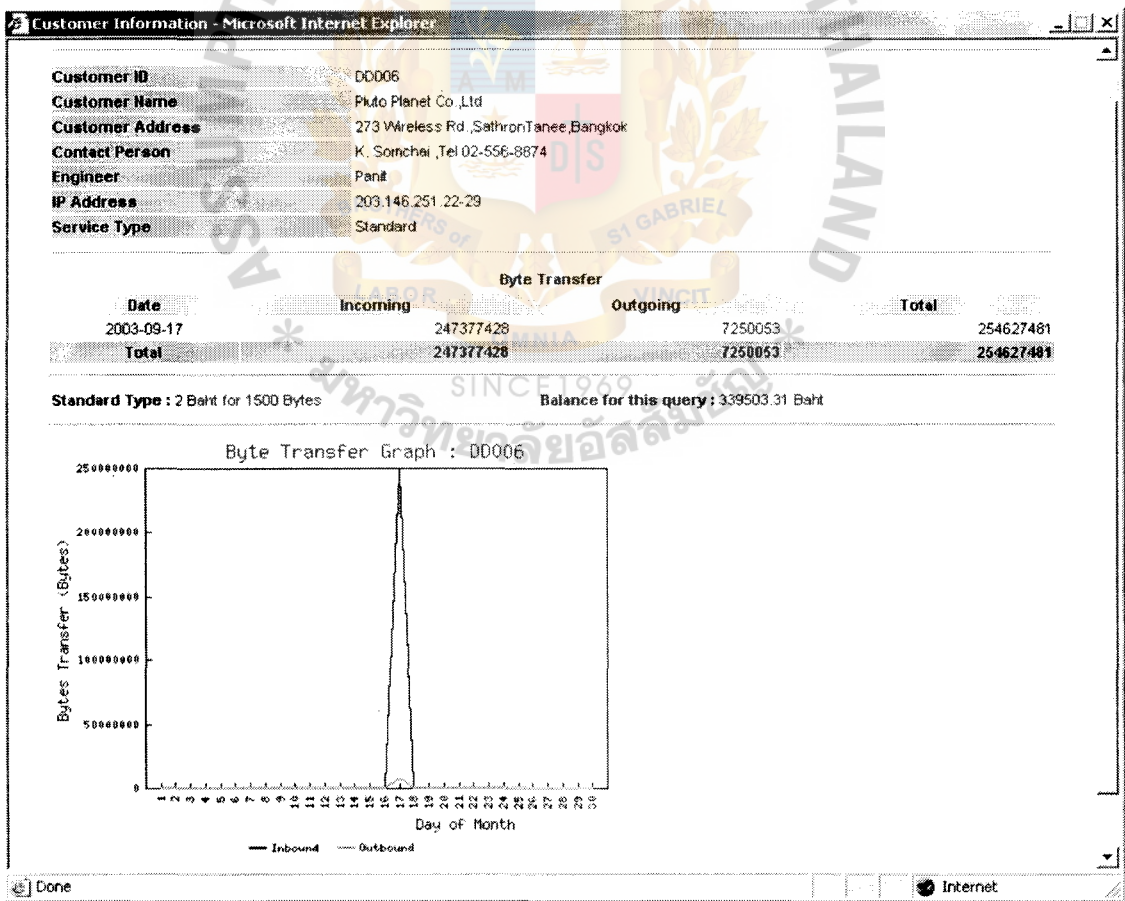


Figure 5-18: Monthly Summarization in Byte Transfer Format

The details of data that is shown on Monthly Byte Transfer screen in Figure 5-18 is same as details on daily byte transfer report. Monthly Byte Transfer report will show the total number of customer usage by summarizing incoming Traffic, outgoing Traffic and total number of them. On the screen it will show the list of incoming, outgoing byte transfer on each day and total byte transfer is also included. Finally the cost of usage is calculated and shown on the screen beside the Billing services type description of customer.

User can view the summarization of Byte Transfer in each month as shown in the Figure 5-18. by accessing screen shown in Figure 5-13, in the Byte Transfer section, user specifies month that he wants to view the data, by selecting month and year that they need at drop-down list box, click at monthly button. System will query data and show it on the screen as in Figure 5-18. Graph view is shown at bottom of the same page by scrolling menu bar at the right side to go to the bottom of this page.

5.5.6.7. Daily Top Talker Summarization Screen

In this section, the system will show the list of IP addresses on the network that communicated with IP of customer and have more of byte transfer.

Customer Information - Microsoft Internet Explorer

Contact Person

K. Somchai ,Tel 02-556-8874

Engineer

Panit

IP Address

203.146.251.22-29

Service Type

Standard

Top Talker

Customer IP	Port	Destination IP	Byte
203.146.251.42	25	203.150.14.100	1282500
203.146.251.54	0	203.118.78.35	804607
203.146.251.3	25	203.147.62.156	360756
203.146.251.54	11247	213.35.101.4	283008
203.146.251.55	0	203.144.147.174	223541
203.146.251.56	0	203.118.77.21	180534
203.146.251.56	0	210.86.160.49	154151
203.146.251.54	2144	203.107.133.238	145628
203.146.251.57	0	210.86.160.154	115032
203.146.251.57	0	203.118.71.120	86994
203.146.251.54	1645	203.107.133.238	69053
203.146.251.6	80	213.113.26.106	63691
203.146.251.55	80	203.144.172.214	57694
203.146.251.3	25	64.4.11.16	53457
203.146.251.55	80	203.113.39.10	52989
203.146.251.6	80	202.57.183.82	52528
203.146.251.6	80	203.107.162.6	47352
203.146.251.54	1657	203.107.133.238	44800
203.146.251.6	80	203.113.67.37	43582
203.146.251.2	80	203.144.143.250	43494
203.146.251.56	0	203.118.77.123	42382

Done

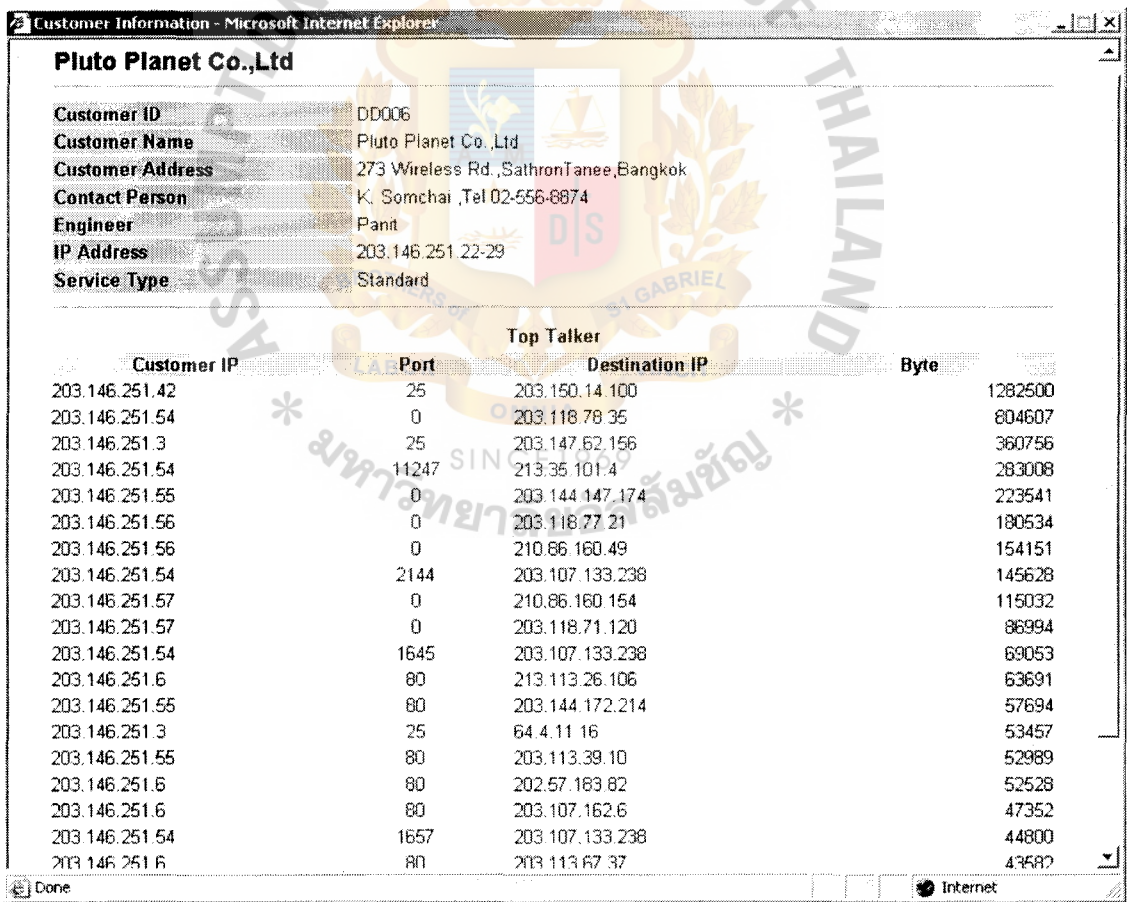
Internet

Figure 5-19: Daily Report of top talker

Daily Top Talker report is the list of customer IP address, which has many byte transfers to the destination IP Address on each day. The list of IP is ranging in descending order of Byte Transfer usage. Details that it shows on the screen are Customer’s IP address, which will show on the left side, services port of usage, destination IP address and finally number byte transfer as shown in Figure 5-19.

User can view daily Top Talker of customer, by accessing screen shown in Figure 5-13. After that specify the period that he wants to view by selecting date; month and year at drop-down list box in top talker section, next is clicking at daily button. The program will query data and show the results as the screen in Figure 5-19. This screen will show the details of customer usage deep in each IP Address. User can use view, which IP address of customer that has more byte transfer with their peers and what is the services type that they talked together.

5.5.6.8. Monthly Top Talker Screen



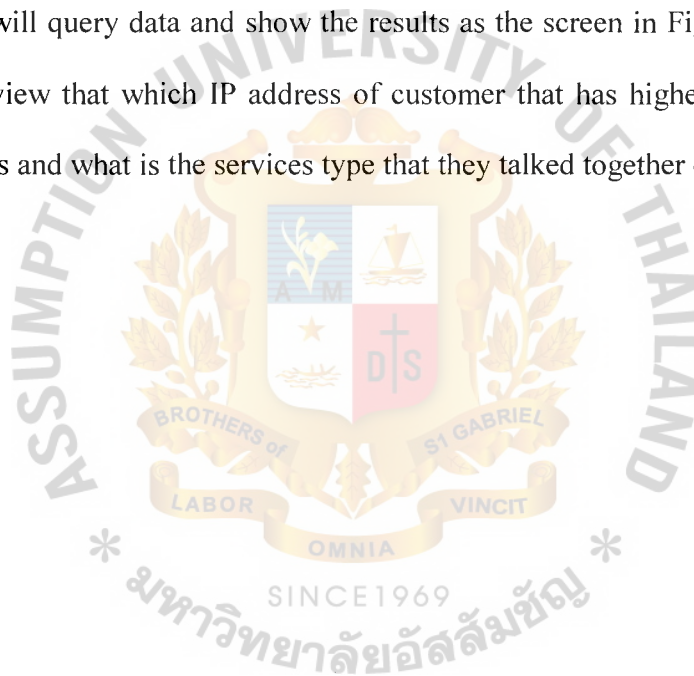
The screenshot shows a web browser window titled "Customer Information - Microsoft Internet Explorer". The page displays customer details for "Pluto Planet Co.,Ltd". Below the details is a table titled "Top Talker" showing data for various customer IP addresses, their ports, destination IP addresses, and the number of bytes transferred.

Customer IP	Port	Top Talker Destination IP	Byte
203.146.251.42	25	203.150.14.100	1282500
203.146.251.54	0	203.118.78.35	804607
203.146.251.3	25	203.147.62.156	360756
203.146.251.54	11247	213.35.101.4	283008
203.146.251.55	0	203.144.147.174	223541
203.146.251.56	0	203.118.77.21	180534
203.146.251.56	0	210.86.160.49	154151
203.146.251.54	2144	203.107.133.238	145628
203.146.251.57	0	210.86.160.154	115032
203.146.251.57	0	203.118.71.120	86994
203.146.251.54	1645	203.107.133.238	69053
203.146.251.6	80	213.113.26.106	63691
203.146.251.55	80	203.144.172.214	57694
203.146.251.3	25	64.4.11.16	53457
203.146.251.55	80	203.113.39.10	52989
203.146.251.6	80	202.57.183.82	52528
203.146.251.6	80	203.107.162.6	47352
203.146.251.54	1657	203.107.133.238	44800
203.146.251.6	80	203.113.67.37	43580

Figure 5-20: Monthly report of top talker for each customer

Monthly Top Talker report presents the customer 's IP address list, which have many of byte transfer to the destination IP Address on each month and also the services that they used. The IP Address is ranging in descending order of byte Transfer number. As show in Figure 5-20.

User can view Monthly Byte Transfer of customer by accessing to screen shown in Figure 5-13, user specifies the period that he wants to view by selecting month and year at drop-down list box in Top Talker section, after that click at “monthly” button. The program will query data and show the results as the screen in Figure 5-20. User can use it to view that which IP address of customer that has highest byte transfer with their peers and what is the services type that they talked together each month.



5.5.6.9. Daily Top Listener Screen

In this section, the system will present the list of customer IP addresses that act as the listener with the other IP on the network and have more of byte transfer.

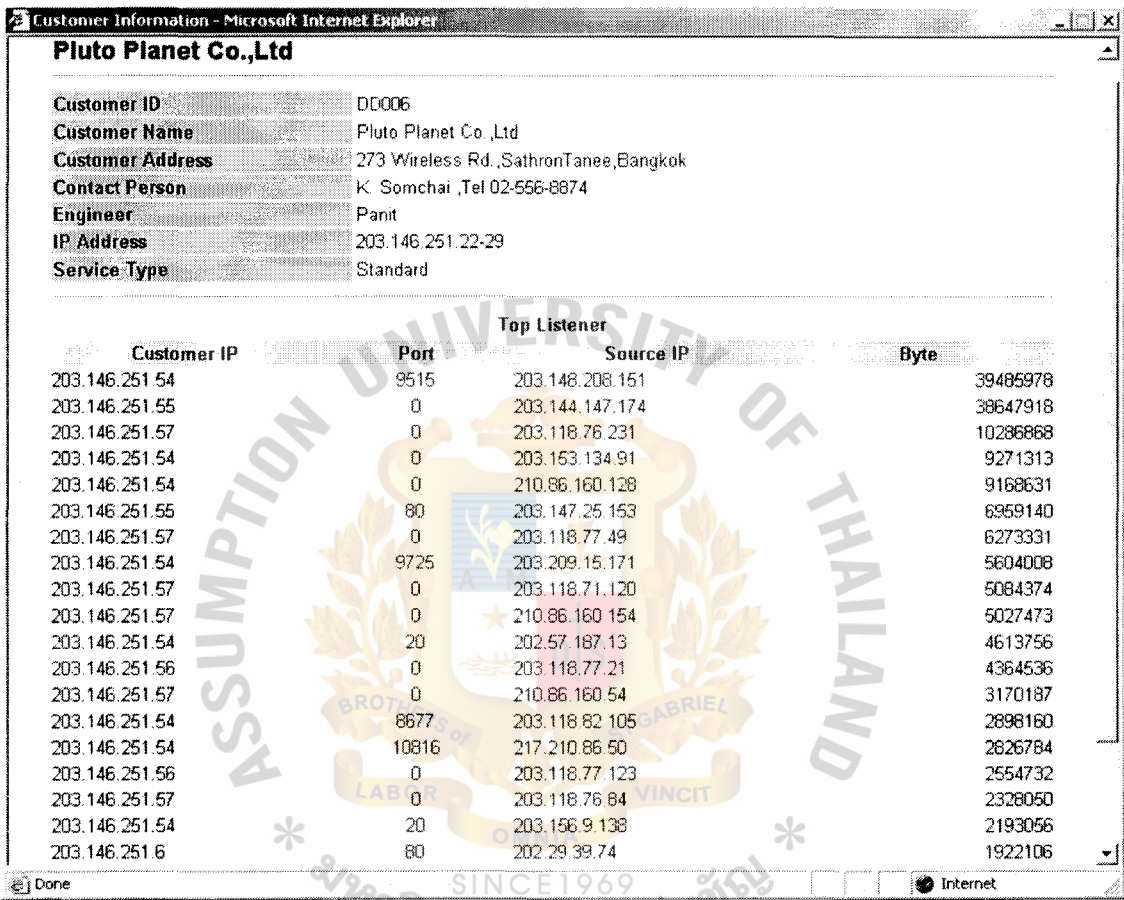
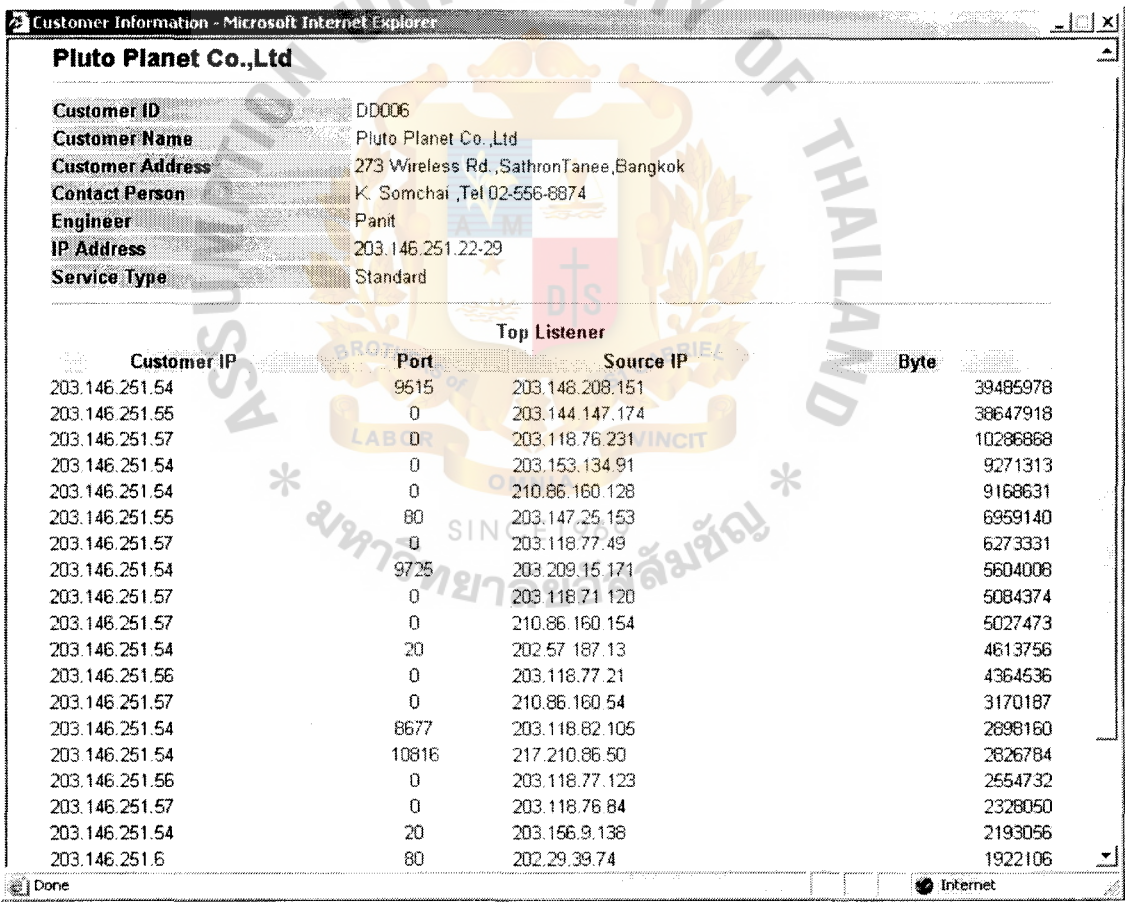


Figure 5-21: Daily report of top listener of Customer's IP

Top Listener report will show list of customer's IP address that is listening to other IP address that came in to talk with on each day. The system is ranging in descending order of byte transfer. It's same as top talker about the details that is shown on the screen. It will show Customer's IP at the left side of screen, services port of usage, destination and finally, the number of bytes transferred in each communication.

User can view daily Top Listener of each customer by accessing the screen shown in Figure 5-13, and specifying the period that he wants to view by selecting date, month and year at drop-down list box, after that clicking at daily button. The program will query data and show the results as the screen in Figure 5-21. User can view then which customer IP is listening to other IP address on the network and have many Byte Transfers on each day.

5.5.6.10. Monthly Top Listener Screen



The screenshot shows a web browser window titled "Customer Information - Microsoft Internet Explorer". The main content area displays customer details for "Pluto Planet Co.,Ltd". Below this, there is a section titled "Top Listener" which contains a table with four columns: "Customer IP", "Port", "Source IP", and "Byte". The table lists 20 rows of data, showing various IP addresses and their corresponding ports and source IPs, along with the number of bytes transferred.

Customer ID	Customer Name	Customer Address	Contact Person	Engineer	IP Address	Service Type
DD006	Pluto Planet Co.,Ltd	273 Wireless Rd.,SathonTanee,Bangkok	K. Somchai ,Tel 02-556-8874	Panit	203.146.251.22-29	Standard

Customer IP	Port	Source IP	Byte
203.146.251.54	9515	203.146.208.151	39485978
203.146.251.55	0	203.144.147.174	38647918
203.146.251.57	0	203.118.76.231	10286868
203.146.251.54	0	203.153.134.91	9271313
203.146.251.54	0	210.86.160.128	9168631
203.146.251.55	80	203.147.25.153	6959140
203.146.251.57	0	203.118.77.49	6273331
203.146.251.54	9725	203.209.15.171	5604008
203.146.251.57	0	203.118.71.120	5084374
203.146.251.57	0	210.86.160.154	5027473
203.146.251.54	20	202.57.187.13	4613756
203.146.251.56	0	203.118.77.21	4364536
203.146.251.57	0	210.86.160.54	3170187
203.146.251.54	8677	203.118.82.105	2898160
203.146.251.54	10816	217.210.86.50	2826784
203.146.251.56	0	203.118.77.123	2654732
203.146.251.57	0	203.118.76.84	2326050
203.146.251.54	20	203.156.9.138	2193056
203.146.251.6	80	202.29.39.74	1922106

Figure 5-22: Monthly report of top listener of Customer’s IP

Top Listener report will show list of customer's IP address that is listening to other IP addresses that come in to talking with in summarized of each month. The system is ranging in descending order of byte transfer. It's same as top talker about the details that are shown on the screen. It will show Customer's IP at the left side of screen, services port of usage, destination and finally, the number of byte transfer in each communication

User can view Monthly Top Listener of each customer by accessing to screen shown in Figure 5-13, and specify the period that he wants to view by selecting month and year at drop-down list box, after that clicking at "Monthly" button. The program will query data and show the results as the screen in Figure 5-22. User can view them which customer IP is listening to other IP addresses on the network and have many Byte Transfer on each month.

All the above details in this section show the methodology that we use to classify and analyze customer usage data on the network. While we have the tools to classify the customer traffic, so we should know how the cost ration comes, we will discuss about the result that we get from this computerized system and the suggestion of Byte Transfer cost calculation in the next chapter.

CHAPTER 6

PRESENTAION AND

CRITICAL DISCUSSION OF RESULTS

This section presents the discussion of results that we got from our thesis, how to apply the new billing system in use and finally the comparison of the existing VPN billing model and the new MPLS VPN billing model.

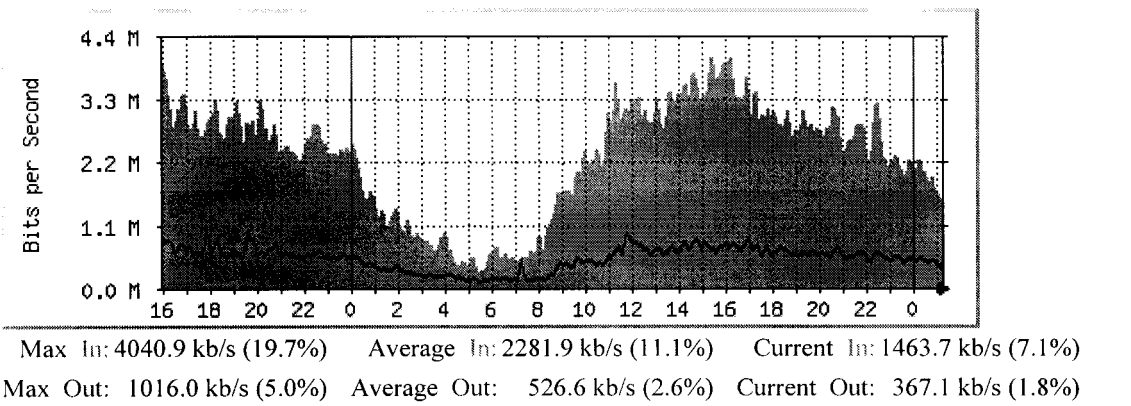
In the first section, we want to show that many customers on the network show different behavior of network usage. So we have bandwidth-based utilization graph of two customers shown below. The graph is showing utilization of network use in Daily and Monthly graph.

Bandwidth usage of Customer A

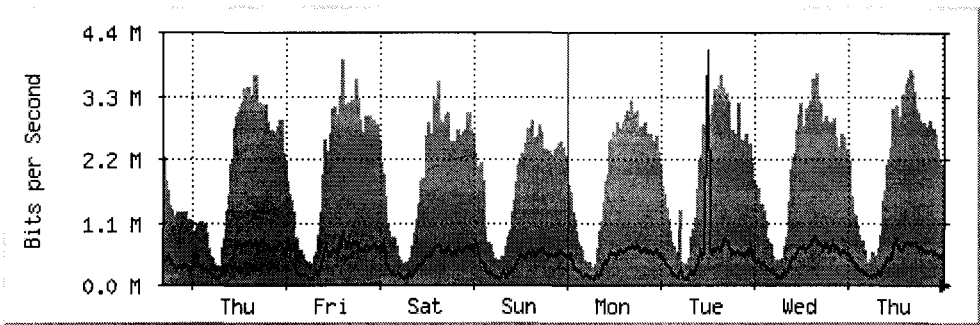
Customer A 's behavior

From utilization graph, Customer A has data transfer all the time. It consumes bandwidth during daytime and dropping after midnight. The Average usage is 2807 kb/s.

Daily' Graph (5 Minute Average)



'Weekly' Graph (30 Minute Average)



Max In: 3931.2 kb/s (19.2%) Average In: 2012.7 kb/s (9.8%) Current In: 2098.4 kb/s (10.2%)
Max Out: 4076.6 kb/s (19.9%) Average Out: 490.4 kb/s (2.4%) Current Out: 501.7 kb/s (2.4%)

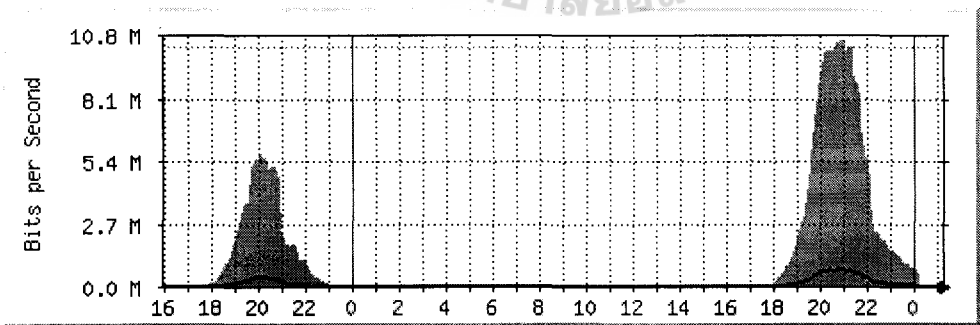
Figure 6-1 Daily and weekly Bandwidth utilization graph of Customer A

Bandwidth usage of Customer B

Customer B's behavior

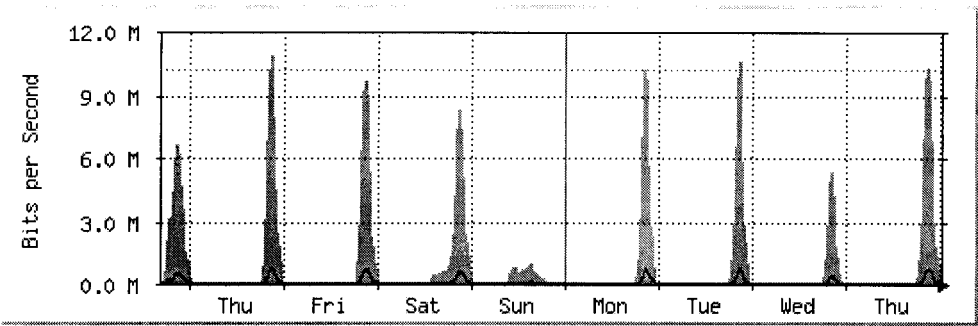
This customer transfers data only in the evening from ~18:00-24:00. No bandwidth usage during daytime period. The Average usage is 1291 kb/s.

'Daily' Graph (5 Minute Average)



Max In: 10.6 Mb/s (103.3%) Average In: 1196.8 kb/s (11.7%) Current In: 0.0 b/s (0.0%)
Max Out: 776.7 kb/s (7.6%) Average Out: 95.3 kb/s (0.9%) Current Out: 424.0 b/s (0.0%)

'Weekly' Graph (30 Minute Average)



Max In: 11.0 Mb/s (107.2%) Average In: 940.7 kb/s (9.2%) Current In: 244.3 kb/s (2.4%)
Max Out: 806.7 kb/s (7.9%) Average Out: 76.5 kb/s (0.7%) Current Out: 28.1 kb/s (0.3%)

Figure 6-2 Daily and weekly Bandwidth utilization graph of Customer B

From these graphs, both customers use the same services, but they show different behavior as we can see from usage graph. However, they have the same monthly fee. So for customers, this is unfair for them who use bandwidth periodically. So we are concerned with cost that they paid, as shown in the next.

The different behavior of network usage causes some user unable to gain the same profits, while they paid at the same monthly fee rate. For showing the difference of worth that they gain from their payment, we have the sample data of network usage of 50 customers as shown on Table 5-1; Reference data from ISP on August, 2003^[16].

In Table 6-1, shows about network usage of same service’s customer. There is customer name in the second column and the average of network usage is next. The inbound usage data is shown in “C3” and the other is outbound usage data shown in “C4”. While we have inbound and outbound usage, so we calculate the average usage of each customer at a time. The Total Average Byte Transfer is calculated from this formula.

$$X = \text{Average Total Byte Transfer (Kbits/s)} = \text{Average Inbound (Kbits/s)} + \text{Average Outbound (Kbits/s)}$$

The inbound usage is plus with the outbound usage, so the Total Byte Transfer (X) is shown in “C5”. The Total Byte Transfer that we got this is per minute of Total Byte Transfer of each customer in the units of Kilobit/s. as shown in the table 6-1. So we need to calculate the Total Byte Transfer of customer on each day by using the formula is,

$$Y = \text{byte transfer (Kbits)/month} = \text{Total Byte Transfer (Kbits)} \times 60 \times 24 \times 30$$

Daily Average Byte Transfer of each customer comes from the value of Average Byte Transfer per minute multiplied by 60 minutes. It will get Average Byte Transfer per hour, after that we multiply this with 24 hours to translate into unit of day. The result is Average Byte Transfer in a day. Next is, we multiply it by 30 days. We will get the results of Monthly Average Byte Transfer of each customer that is shown in Table 6-1 . Monthly Byte Transfer of each customer will be calculated and put into “Average Byte Transfer: Z (Mbits)/month” in “C6”.

Now, we got the Daily Average Byte Transfer of the customer, but the unit is Kilobit. For easy understanding, we will change it into the unit of Megabytes by,

$$Z = \text{Average Byte Transfer (Mbytes)/m} = (\text{Average byte transfer (Kbits/m)} / 8) / 1,000$$

From the formula, changing from bit unit to Byte is divided by 8 and divided by 1000 for changing from Kilobyte to Megabyte. While we have the Monthly Average Byte Transfer of each customer, so we can show that those customers who have different network usage

behavior, they are also different in gaining the benefit of network usage, while they pay the same rate.

Suppose that each customer has the same Monthly Fee equal to 6,500 baht/month. We multiply the monthly byte transfer of customer by the monthly fee that each customer pays.

Byte Transfer per Baht = Average Byte Transfer per month/6,500

The result of this calculation is shown in “C8” on Table 6-1. From the results, it implies that for same 1 baht, each customer has different Byte Transfer value, which depends on their usage. Customers who pay 1 baht, but their Average Byte Transfers of data are more than the other who pay the same. That cause the difference in benefits that each customer can gain from the same rate of monthly fee.

Unfairness is faced by the customer who has low bandwidth usage as we can see from above calculation. So a new Billing System is proposed. This system’s billing is based on byte transfer of each customer using in the network. So customer who uses low bandwidth pays less, while customer who consumes more bandwidth will pay more. This is fair for both customer and service provider.

But there are some issues that we should be concerned about. Byte transfer billing is fair for everyone, if we apply this billing method with all 50 sample customers, suppose that we charge a rate about 1 baht per Megabyte. Customer who has low usage of network will actually gain benefit from decreasing of the monthly fee. Otherwise customer, who has high bandwidth usage, will pay more monthly fee. Actually, that will

not attractive to Service Provider to implement this new billing system, because from the sample of calculation, mostly of the customers will pay less than the current. Revenue of service provider will immediately decrease because the revenue that is gained from low usage customer will decrease from the current. This will directly cause more affects to the service provider. For customer who has high bandwidth usage, more complaints and termination by this group of customers can occur.

However, we have some suggestions in applying our billing system in use that compromises both the service provider and customer; It is more attractive to service provider to implement this billing system and also customer still gains the fairness of billing. That is, services provider who give services to their customer, should be decide what is the target group of customers that they will apply this new billing system to.

For example, Average of Byte Transfer per month of 50 customers on Table 6-1 , “C7” , we select the target group of customers who has Average Byte Transfer of usage below 1 Gigabyte per month. From these criteria, we will get about 22 companies as the target group customers. So we find the Average Byte transfer of our target group from this formula,

$$\frac{\sum_{i=1}^n \text{Average Byte Transfer (Mbytes) per month}_i}{n}$$

N = Number of customers

From the formula, we will get,

Average Byte Transfer = 7610.0904/22 =345.91 Mbytes/month

We cumulating the Average Byte Transfer of all customers and dividing by customer number, the result is the Average of Byte transfer per month of 22 customers. After that, we get the Average Byte transfer of target group. Next is, we will calculate the number of Byte transfer per 1 baht on each month, suppose that we use the same monthly fee as above, that is, 6500 baht/m.

We will get the result as,

Byte Transfer/1 baht = 345.91/6500=0.05321741 Mbytes

In above case, we divide the Average Byte Transfer per month by Monthly Fee, so we will get the Average of Byte Transfer per 1 baht. From the result, it implies that 1 baht can transfer data about 0.05321741 Mbytes. After that we use this value to divide the Average Byte Transfer per month of each customer that is shown in “C3” on Table 6-2, we will get the expense of bandwidth usage per month of each customer in target group as shown in “C5” on Table 6-2.

After we apply the suggested billing method to our target group, we have found that 12 customers have decreasing monthly expense of network usage. It implies, that 54% of the target group has a lower expense than current when we apply the billing system to. Most of them are customers who have low bandwidth usage. But customers who have high bandwidth usage have more monthly expense of network use than the

other. But this issue it depends on service provider, that is which is their target group of customer that they can implement this billing system.

When we consider the result of calculation, we can conclude that the new billing system is not appropriate for the customer who has high bandwidth usage. Anyway the new billing system is mostly appropriate for the customer who has low bandwidth usage, because new billing system is billing based on the customer usage; it's fair for everyone who uses the network. However, this new billing system in the propose system, it depends on the provider who uses this system. They can gain more benefits in appropriate of adaptation. Finally, the supposed variable that we use in the calculation will be different for each service provider. It's based on their target group of customers. In the last section of this chapter, we show the comparison of the current VPN billing system and the new billing system in many aspects.

	Existing VPN Billing	New MPLS VPN Billing
Flexibility	Cost is based on Services Provider expenses and fix rate.	Compatible with rapidly change of VPN Technology.
Complexity	Cost is fixed and predefined.	Need computerized system and more skill of administrator in implementing the billing system
Fairness	Either high or low network usages, Customer are pays the same rate.	Cost is based on customer 's usage.
Investment	Low	High

Table 6-3 Comparison between current VPN billing system and the new billing system

In Table 6-3,shows that, in the aspect of Flexibility, the existing VPN billing has less flexibility, their billing is of fixed rate and less of tool support in changing. New billing system is based on customer usage. The system will consider deep into the details of network usage, so they can adapt to many kinds of network technologies, even if it changes.

But in the aspect of complexity, the new billing system is more complex than the existing billing system. The new billing system will need support by computerized system and strong skill of network and administrator to implement, maintain and manage it.

Good fairness is seen in the new billing system as we have shown above. The billing is based on customer usage, customers who have high usage, will pay more. Otherwise, customers who use low of bandwidth, also pay less of the expense. While the

existing system, cost is based on the service provider 's expense and pre-defined charge rate. Both customers who have high or low of bandwidth usage, they will pay the same fixed rate fee.

Investment cost of new billing system is higher than the existing VPN billing system, because new billing system will use the computerized system as main component to classify the customer usage on the network, so it needs to invest in high-end hardware to handle the huge amount of data on the network.



CHAPTER 7

CONCLUSIONS AND

RECOMMENDATIONS FOR FURTHER RESEARCH

7.1 Conclusions

In this paper, we propose a new billing system on MPLS VPN network based on customer usage while the current billing system is based on monthly fee. When we study and observe customer's usage behavior on the network, we have found that customers are of various usage types. Some customers have constant and various services usage. Another is on demand usage, They will transfer data only at a time that they want. The difference of usage types causes the customer not to gain the same profit, while both of them have to pay the monthly fee equally.

New billing system is based on data transfer that the customer used. We suppose to apply the new billing system to the customer. It shows that the propose system can achieve fairness for both the user and the services provider. Their billing will be based on their usage. Customers who have high or low usage will pay according to their uses. Customers who have low network usage will pay less than the fixed rate of current billing system because of their less usage. But customer who has high network usage will pay more than current payment because they have more network usage. Customers who have low network usage can gain more fairness than in the current system. Service Provider will gain more revenue from customers who consume more bandwidth. So we can conclude that the proposed system is better for customers who have low usage of bandwidth. Their monthly fee is decreased from the current. But for customers who have high bandwidth usage, their monthly fee will increasing from the current. The proposed system is not

appropriate for customers who have high network usage. For this group of customers, the current billing system is appropriate for them than the proposed system.

There are more benefits that we can gain from the propose system. First is the fairness, customer who use the VPN network get fairness in billing, service provider can gain more money from the user who consumes more bandwidth than others. Next is we will be able to fully use function that are already included in network equipment, no additional feature is required for the network equipment to enable Flow Export function. The other is this system can be adopted to not only in the VPN network; another IP network can also apply this system. Finally service provider can use it as a new marketing strategy to attract customers to increase their market share.

Although there are more benefits, but there are limitations that we should be concerned about. That is, this system will need a network and Unix skill administrator who will implement, maintain and manage this system to work properly.

7.2 Recommendations for Further Research

There are some recommendations that are described for further development of this system. First is server's performance. This system is tested on tiny network infrastructure. The performance of the server is enough, but if this system is implemented to the actual VPN network, such as ISP network, high performance server is needed. The other separating the Traffic Flow Collector server from Database server is recommended, for handling huge of traffic flow data in the ISP network. Another is the security aspect that should be concerned about, because this system is dealing with the data that concerns more of the sensitive data such as cost or network usage of each customer, which is confidential information of customer, than the system should be installed strictly with security that can protect the system from the hacker or criminal. Finally this system is

appropriate for and fair to some groups of customers, so the fairness of billing system for another group should be further found.



LIST OF FIGURES

Figure 1-1 : CPE-VPN model	4
Figure 1-2 : PP-VPN model	5
Figure 1-3 : L2TP,PPTP Model	6
Figure 1-4 : IPSec model	7
Figure 2-1 : VPN-IPv4 Address	10
Figure 2-2 : The BGP/MPLS VPN Approach	12
Figure 5-1 : Data Flow of Billing System	23
Figure 5-2 : Login Windows	25
Figure 5-3 : Home page	26
Figure 5-4 : Customer List Screen	28
Figure 5-5 : Ration List Screen	29
Figure 5-6 : Services list Screen	30
Figure 5-7 : Database Management Screen	31
Figure 5-8 : Customer Table Screen	32
Figure 5-9 : Connection Table	34
Figure 5-10.: Edit Screen of Ratio Table.	35
Figure 5-11 : Services Type Table	36
Figure 5-12 : Customer list for view of usage	38
Figure 5-13 : Utilization that classify in each category	39
Figure 5-14 : Port Aggregation results of each customer in tabular format	40
Figure 5-15 : Port Aggregation results of each customer in Graph Format	41
Figure 5-16 : Monthly report of Port Aggregation	42

Figure 5-17 : Daily summarization of Byte Transfer Format 43

Figure 5-18 : Monthly Summarization in Byte Transfer Format 44

Figure 5-19 : Daily Report of top talker 46

Figure 5-20 : Monthly report of top talker for each customer 47

Figure 5-21 : Daily report of top listener of Customer’s IP 49

Figure 5-22 : Monthly report of top listener of Customer’s IP 50

Figure 6-1 : Daily and weekly bandwidth utilization of Customer A 52

Figure 6-2 : Daily and weekly bandwidth utilization of Customer B 53



REFERENCES

- [1] Eric C. Rosen, Yakov Rekhter “BGP/MPLS VPNs”, RFC2574, March 1999 .
- [2] Trillium, “ Multiprotocol Label Switching [MPLS] ”, The International Engineering Consortium.
- [3] Bruce Davie, Yakov Rekhter “MPLS Technology and Application” , Morgan Kaufmann Publishers, 2000.
- [4] Jim Guichard, Ivan Pepelnjak , “MPLS & VPN Architectures - Practical Guide to Understanding, Designing and Deploying MPLS & MPLS-Enabled VPNs”, CISCO Press, 1st Edition ,2000.
- [5] Vivek Alwayn, “Advanced MPLS Design & Implementation”, CISCO Press, 1st Edition ,2001.
- [6] Chuck Semeria, “RFC 2547bis:BGP/MPLS VPN Fundamental” , Juniper Networks,2001.
- [7] Chuck Semeria, “RFC 2547bis:BGP/MPLS VPN Hierarchical and Recursive Applications” , Juniper Networks,2001.

- [8] Vijay Bollapragada, Curtis Murphy and Russ White, “Inside CISCO IOS Software Architecture”, CISCO Press, 2000.
- [9] Lesage Thierry, “A Tool to set up Price when creating new services”, Network 2002.
- [10] Dany Sylvain, “Building Next Generation Networks”, Network 2002.
- [11] Dr. Keith Burston, “Internet Billing”, Division of Information Services, University of New South Wales, August 1997.
- [12] CAIDA Tool, “CFLOWD”, Jun 2002, Version 2.1.5
Available from URL: <http://www.caida.org/tools/measurement/cflowd/>,
<http://net.doit.wisc.edu/~plonka/FlowScan/INSTALL.html>
- [13] Customized Tools, “PERL Module”, Version 5.0
Available from URL: <http://www.perl.com> , <http://www.perl.com/CPAN-local/README.html>
- [14] Database Application , “MySQL” Version 4.1,
Available from URL: <http://www.mysql.com>
- [15] Database Management Tool, “PHPMYADMIN”, Version 2.5.1
Available from [URL: http://www.phpmyadmin.net](http://www.phpmyadmin.net)
- [16] CS Loxinfo Co., Ltd : Bandwidth Utilization Graph, “MRTG Application” ,
August 2003,

APPENDIX A: Abbreviations

BGP	: Border Gateway Protocol
CE	: Customer Edge
IANA	: Internet Assigned Numbers Authority
L2TP	: Layer 2 Tunneling Protocol
L2F	: Layer 2 Forwarding
LAN	: Local Area Network
LSP	: Label Switching Path
MPLS	: Multi-Protocol Label Switching
PE	: Provider Edge
PPP	: Point-to-Point Protocol
PPTP	: Point-to-Point Tunneling Protocol
RD	: Route Distinguisher
VPN	: Virtual Private Network
VRF	: VPN Routing and Forwarding

1	Company A	11.7	5.688	17.3880	751,161.6000	93.8952	0.014445
2	Company B	293.8	117.6	411.4000	17,772,480.0000	2,221.5600	0.341778
3	Company C	157.2	67.3	224.5000	9,698,400.0000	1,212.3000	0.186508
4	Company D	10	7.552	17.5520	758,246.4000	94.7808	0.014582
5	Company E	7.752	5.544	13.2960	574,387.2000	71.7984	0.011046
6	Company F	797.9	65.7	863.6000	37,307,520.0000	4,663.4400	0.717452
7	Company G	3764.2	935.8	4,700.0000	203,040,000.0000	25,380.0000	3.904615
8	Company H	581.8	79	660.8000	28,546,560.0000	3,568.3200	0.548972
9	Company I	629	48.4	677.4000	29,263,680.0000	3,657.9600	0.562763
10	Company J	877.2	153	1,030.2000	44,504,640.0000	5,563.0800	0.855858
11	Company K	163.7	124	287.7000	12,428,640.0000	1,553.5800	0.239012
12	Company L	446.1	199.4	645.5000	27,885,600.0000	3,485.7000	0.536262
13	Company M	58.9	18.6	77.5000	3,348,000.0000	418.5000	0.064385
14	Company N	60.9	16.5	77.4000	3,343,660.0000	417.9600	0.064302
15	Company O	176.8	24.1	200.9000	8,678,880.0000	1,084.8600	0.166902
16	Company P	718.8	455.9	1,174.7000	59,747,040.0000	6,343.3800	0.975905
17	Company Q	81.5	13.7	95.2000	4,112,640.0000	514.0800	0.079089
18	Company R	107.6	37.5	145.1000	6,268,320.0000	783.5400	0.120545
19	Company S	12.5	2.376	14.8760	642,643.2000	80.3304	0.012359
20	Company T	60.4	36.2	96.6000	4,173,120.0000	521.6400	0.080252
21	Company U	9.256	16.8	26.0560	1,125,619.2000	140.7024	0.021647
22	Company V	165.4	23	188.4000	8,138,880.0000	1,017.3600	0.156517
23	Company W	1743.9	343.9	2,087.8000	90,192,960.0000	11,274.1200	1.734480
24	Company X	128.9	21.7	150.6000	6,505,920.0000	813.2400	0.125114
25	Company Y	2.024	4.824	6.8480	295,833.6000	36.9792	0.005689
26	Company Z	40.2	8.576	48.7760	2,107,123.2000	263.3904	0.040522
27	Company AA	18.3	1.896	20.1960	872,497.2000	109.0584	0.016778
28	Company BB	267.1	43.3	310.4000	13,409,280.0000	1,676.1600	0.257871
29	Company CC	10.2	10.7	20.9000	902,880.0000	112.8600	0.017363
30	Company DD	177.2	38.9	216.1000	9,335,520.0000	1,166.9400	0.179529
31	Company EE	199	72.2	271.2000	11,715,840.0000	1,464.4800	0.225305
32	Company FF	1059.5	1436.1	2,495.6000	107,809,920.0000	13,476.2400	2.073268
33	Company GG	1435.8	1060.2	2,496.0000	107,827,200.0000	13,478.4000	2.073600
34	Company HH	733.6	738.8	1,472.4000	63,607,680.0000	7,950.9600	1.223225
35	Company II	58.9	67	125.9000	5,438,880.0000	679.8600	0.104594
36	Company JJ	367.6	26.8	394.4000	17,038,080.0000	2,129.7600	0.327655
37	Company KK	155.6	294.2	419.8000	18,135,360.0000	2,266.9200	0.348757
38	Company LL	90	102.5	192.5000	8,316,000.0000	1,039.5000	0.159923
39	Company MM	1109.3	515.7	1,625.0000	70,200,000.0000	8,775.0000	1.350000
40	Company NN	650.8	28.7	679.5000	29,354,400.0000	3,669.3000	0.564508
41	Company OO	160	27.7	187.7000	8,108,640.0000	1,013.5800	0.155935
42	Company PP	7.944	8.576	16.5200	713,664.0000	89.2080	0.013724
43	Company QQ	8716.4	3711.3	12,427.7000	536,876,640.0000	67,109.5800	10.324551
44	Company RR	21.6	8.568	30.1680	1,303,257.6000	162.9072	0.025063
45	Company SS	487.7	55.8	543.5000	23,479,200.0000	2,934.9000	0.451523
46	Company TT	13.3	27.8	41.1000	1,775,520.0000	221.9400	0.034145
47	Company UU	988.7	137	1,125.7000	48,630,240.0000	6,078.7800	0.935197
48	Company VV	84.5	54.3	138.8000	5,996,160.0000	749.5200	0.115311
49	Company WW	102.4	37.5	139.9000	6,043,680.0000	755.4600	0.116225
50	Company XX	68.7	19.9	88.6000	3,827,520.0000	478.4400	0.073606

Reference : Data from ISP on August 2003

1	Company A	11.7	5.688	93.8952	6,500	1,764.3698
2	Company B	293.8	117.6	2,221.5600		
3	Company C	157.2	67.3	1,212.3000		
4	Company D	10	7.552	94.7808	6,500	1,781.0110
5	Company E	7.752	5.544	71.7984	6,500	1,349.1523
6	Company F	797.9	65.7	4,663.4400		
7	Company G	3784.2	935.8	25,380.0000		
8	Company H	581.8	79	3,568.3200		
9	Company I	629	48.4	3,657.9600		
10	Company J	877.2	153	5,563.0800		
11	Company K	163.7	124	1,553.5800		
12	Company L	446.1	199.4	3,485.7000		
13	Company M	58.9	18.6	418.5000	6,500	7,863.9670
14	Company N	60.9	16.5	417.9600	6,500	7,853.8200
15	Company O	176.8	24.1	1,084.8600		
16	Company P	718.8	455.9	6,343.3800		
17	Company Q	81.5	13.7	514.0800	6,500	9,659.9956
18	Company R	107.6	37.5	783.5400	6,500	14,723.3757
19	Company S	12.5	2.376	80.3304	6,500	1,509.4758
20	Company T	60.4	38.2	521.6400	6,500	9,802.0544
21	Company U	9.256	18.8	140.7024	6,500	2,643.9165
22	Company V	165.4	23	1,017.3600		
23	Company W	1743.9	343.9	11,274.1200		
24	Company X	128.9	21.7	813.2400	6,500	15,281.4637
25	Company Y	2.024	4.824	36.9792	6,500	694.8703
26	Company Z	40.2	8.576	263.3904	6,500	4,949.3272
27	Company AA	18.3	1.896	109.0584	6,500	2,049.2991
28	Company BB	267.1	43.3	1,676.1600		
29	Company CC	10.2	10.7	112.8600	6,500	2,120.7343
30	Company DD	177.2	38.9	1,166.9400		
31	Company EE	199	72.2	1,464.4800		
32	Company FF	1059.5	1436.1	13,476.2400		
33	Company GG	1435.8	1060.2	13,478.4000		
34	Company HH	733.6	738.8	7,950.9600		
35	Company II	58.9	67	679.8600	6,500	12,775.1413
36	Company JJ	367.6	26.8	2,129.7600		
37	Company KK	155.6	284.2	2,266.9200		
38	Company LL	90	102.5	1,039.6000		
39	Company MM	1109.3	515.7	8,775.0000		
40	Company NN	650.8	28.7	3,669.3000		
41	Company OO	160	27.7	1,013.5800		
42	Company PP	7.944	8.576	89.2080	6,500	1,676.2934
43	Company QQ	8716.4	3711.3	67,109.5800		
44	Company RR	21.6	8.568	162.9072	6,500	3,061.1633
45	Company SS	487.7	55.8	2,934.9000		
46	Company TT	13.3	27.8	221.9400	6,500	4,170.4393
47	Company UU	988.7	137	6,078.7800		
48	Company VV	84.5	54.3	749.5200	6,500	14,084.1113
49	Company WW	102.4	37.5	755.4600	6,500	14,195.7289
50	Company XX	68.7	19.9	478.4400	6,500	8,990.2900

Reference : Data from ISP on August .2003

SOURCE CODE: FILTER SCRIPT

```
#!/usr/bin/perl

#####

# FILENAME : filter #
# JOB : Separate flow and store in database #
# LAST UPDATE : 20 July 2003 #
#####

#### CALL LIBRARY ####

use Mysql;
use Cflow qw(:flowvars find);
use Date::Calc qw(Delta_Days);
use FileHandle;
use File::Basename;

#### GLOBAL VARIABLE ####

$filesset = "/usr/local/arts/data/flows.*:[0-9]";
#$filter = "/flow/data/filter";
$wait = "60";
@block;
```

```
##### CONNECT TO DATABASE #####
```

```
$host = "203.146.248.242";
```

```
$database = "Thesis";
```

```
$| = 1;
```

```
my $db = Mysql->Connect ($host, $database, 'kookae', 'kaeadmin') || die "Cannot connect :
```

```
$Mysql::db_errstr\n";
```

```
$db->SelectDB ($database) || die "Cannot connect : $Mysql::db_errstr\n";
```

```
##### FIND FLOW FILE #####
```

```
my $udp = getprotobyname('udp');
```

```
open(FILTER, $filter);
```

```
while ($line = <FILTER>)
```

```
{
```

```
if (!($line =~ /#/))
```

```
{
```

```
@tmp = split(/\./, $line);
```

```
$net = "$tmp[0].$tmp[1].$tmp[2]";
```

```
push(@block, $net);
```

```
}
```

```
}
```

```
##### GET & SORT FLOW DATA #####
```

```
while (1)
```

```
{
```

```
my @files = sort timestamp <${fileset}>;
```

```
if (@files)
```

```
{
```

```
my $file;
```

```
foreach $file (@files)
```

```
{
```

```
my $result = find(\&wanted, \&perfile, "$file");
```

```
print localtime() . " : Removing file $file ...\\n";
```

```
unlink($file);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
print localtime() . " : Input file not found. Change to sleep mode ...\\n";
```

```
sleep $wait;
```

```
}
```

```
}
```

```
sub timestamp
```

```
{
```

```
$a <=> $b;
```

```
}
```


TRANSLATE FLOW AND STORE IN DATABASE

sub wanted

```
{  
  
    $check = $db->Query ("SELECT CustomerID FROM Connection WHERE EqIP='$exporterip'  
AND IntIndex='$input_if'");  
  
    if (@line = $check->FetchRow()) { $direction = "i"; $customer=$line[0];}  
    else  
    {  
  
        $check = $db->Query ("SELECT CustomerID FROM Connection WHERE EqIP='$exporterip'  
AND IntIndex='$output_if'");  
  
        if (@line = $check->FetchRow()) { $direction = "o"; $customer=$line[0];}  
        else { $direction = "e"; }  
    }  
  
    if ($direction eq "i")  
    {  
  
        $duration = $endtime - $starttime;  
  
        ($eday, $emonth, $edate, $etime, $eyear) = split(/[s]+/, localtime($endtime));  
  
        ($ehour, $eminute, $esecond) = split(/:/, $etime);  
  
        $emonth = monthint($emonth);  
  
        $edate = fill($edate);  
  
        $ehour = fill($ehour);  
  
        $eminute = fill($eminute);  
  
        $esecond = fill($esecond);  
  
        if ($check = $db->Query("INSERT INTO $customer VALUES ('$srcip', '$sreport', '$dstip',  
'$dstport', '$direction', '$protocol', '$bytes', '$eyear-$emonth-$edate $ehour:$eminute:$esecond',  
'$duration')")) {}  
    }  
    else  
    {  

```

```

create($customer);

$check = $db->Query("INSERT INTO $customer VALUES ('$srcip', '$srcport', '$dstip',
'$dstport', '$direction', '$protocol', '$bytes', '$year-$month-$date $hour:$minute:$second',
'$duration')");
}
}
elseif ($direction eq "o")
{
$duration = Sendtime - $starttime;
($day, $month, $date, $time, $year) = split(/[\s]+/, localtime($sendtime));
($hour, $minute, $second) = split(/:/, $time);
$month = monthint($month);
$date = fill($date);
$hour = fill($hour);
$minute = fill($minute);
$second = fill($second);
if ($check = $db->Query("INSERT INTO $customer VALUES ('$dstip', '$dstport', '$srcip',
'$srcport', '$direction', '$protocol', '$bytes', '$year-$month-$date $hour:$minute:$second',
'$duration')")) {}
else
{
create($customer);

$check = $db->Query("INSERT INTO $customer VALUES ('$dstip', '$dstport', '$srcip',
'$srcport', '$direction', '$protocol', '$bytes', '$year-$month-$date $hour:$minute:$second',
'$duration')");
}
}
}

```

```

sub perfile
{
    my $file = shift;
    print localtime() . " : Working on file ${file} ... \n";
}

```

CHANGE MONTH NAME INTO NUMBER

```

sub monthint
{
    my ($a) = @_ ;
    if ($a eq "Jan") { return "01"; }
    elsif ($a eq "Feb") { return "02"; }
    elsif ($a eq "Mar") { return "03"; }
    elsif ($a eq "Apr") { return "04"; }
    elsif ($a eq "May") { return "05"; }
    elsif ($a eq "Jun") { return "06"; }
    elsif ($a eq "Jul") { return "07"; }
    elsif ($a eq "Aug") { return "08"; }
    elsif ($a eq "Sep") { return "09"; }
    elsif ($a eq "Oct") { return "10"; }
    elsif ($a eq "Nov") { return "11"; }
    elsif ($a eq "Dec") { return "12"; }
}

```

FILL ZERO

```

sub fill
{

```

```

my ($a) = @_;
if ($a eq "0") { return "00"; }
elsif ($a eq "1") { return "01"; }
elsif ($a eq "2") { return "02"; }
elsif ($a eq "3") { return "03"; }
elsif ($a eq "4") { return "04"; }
elsif ($a eq "5") { return "05"; }
elsif ($a eq "6") { return "06"; }
elsif ($a eq "7") { return "07"; }
elsif ($a eq "8") { return "08"; }
elsif ($a eq "9") { return "09"; }
else { return $a; }
}

```

CREATE TABLE

sub create

```
{
```

```
my ($a) = @_;
```

```

$check = $db->Query("CREATE TABLE $a (CIP varchar(15), CPort varchar(5), NIP varchar(15),
NPort varchar(5), IO char(1), Protocol varchar(5), Byte bigint(10), End varchar(19), Duration varchar
(4))");

```

```
}
```

SOURCE CODE: SUMMARY SCRIPT

```
#!/usr/bin/perl

#####

# FILENAME      : summary          #
# JOB           : Summarize customer's tables #
# LAST UPDATE   : 25 August 2003    #
#####

##### CALL LIBRARY #####

use Mysql;
use FileHandle;
use File::Basename;

##### GLOBAL VARIABLE #####

$date    = $ARGV[0];
$customer = $ARGV[1];

##### CHECK SYNTAX #####

if ($customer eq "")
{
    print "\nUSAGE : summary <now|date> <customerid|all>\n\n";
    print "EXAMPLE\n\n";
}
```

```

print "\t# Summarize customer id DD001 on 2 Jan 2003.\n";
print "\t summary 2003-01-02 DD001\n\n";
print "\t# Summarize customer id DD001 on the current date.\n";
print "\t summary now DD001\n\n";
print "\t# Summarize all customer on the current date.\n";
print "\t summary now all\n\n";
exit(0);
}

```

GET DATE

```

if ($date eq "now")
{
    $date = `date +%d`;
    $month = `date +%m`;
    $year = `date +%Y`;
    chomp $date;
    chomp $month;
    chomp $year;
    $date--;
    $date = "$year-$month-$date";
}

```

GET TABLE PREFIX

```

@element = split(/-/, $date);
$prefix = "$element[0]$element[1]";

```



```
##### CONNECT TO DATABASE #####
```

```
$host = "203.146.248.242";
```

```
$database = "Thesis";
```

```
$user = "kookae";
```

```
$password = "kacadmin";
```

```
$i = 1;
```

```
my $db = Mysql->Connect($host, $database, $user, $password) or die "Cannot connect :
```

```
$Mysql::db_errstr\n";
```

```
##### GET CUSTOMER ID LIST #####
```

```
if ($customer eq "all")
```

```
{
```

```
    $check = $db->Query("SELECT CustomerID FROM Customer");
```

```
}
```

```
else
```

```
{
```

```
    $check = $db->Query("SELECT CustomerID FROM Customer WHERE  
CustomerID='$customer'");
```

```
}
```

SUMMARY IF CUSTOMER TABLE EXISTS

```
while (@line = $check->FetchRow())
```

```
{
```

```
    $sumtable = "$prefix$line[0]";
```

```
    $action = $db->Query("CREATE TABLE $sumtable (CIP varchar(15), CPort varchar(5), NIP  
varchar(15), IO char(1), Protocol varchar(5), Byte bigint(10), Date varchar(19))");
```

```
    $query = $db->Query("SELECT CIP, CPort, NIP, IO, Protocol, SUM(Byte), End FROM $line[0]  
WHERE End >= '$date 00:00:00' AND End <= '$date 23:59:59' GROUP BY CIP, CPort, NIP, IO,  
Protocol");
```

```
    while (@row = $query->FetchRow())
```

```
    {
```

```
        @element = split(/ /, $row[6]);
```

```
        $date = "Selement[0]";
```

```
        $action = $db->Query("INSERT INTO $sumtable VALUES ('$row[0]', '$row[1]', '$row[2]',  
'$row[3]', '$row[4]', '$row[5]', '$date')");
```

```
    }
```

```
}
```

SOURCE CODE: PORT SCRIPT

```
#!/usr/bin/perl

#####

# FILENAME      : port                #
# JOB           : Summarize customer's tables by port #
# LAST UPDATE   : 25 August 2003      #
#####

##### CALL LIBRARY #####

use Mysql;
use FileHandle;
use File::Basename;

##### GLOBAL VARIABLE #####

$date   = $ARGV[0];
$customer = $ARGV[1];

##### CHECK SYNTAX #####

if ($customer eq "")
{
    print "\nUSAGE : port <now|date> <customerid|all>\n\n";
    print "EXAMPLE\n\n";
}
```

```

print "\t# Summarize customer id DD001 on 2 Jan 2003. (by port)\n";
print "\t port 2003-01-02 DD001\n\n";
print "\t# Summarize customer id DD001 on the current date. (by port)\n";
print "\t port now DD001\n\n";
print "\t# Summarize all customer on the current date. (by port)\n";
print "\t port now all\n\n";
exit(0);
}

```

GET DATE

```

if ($date eq "now")
{
    $date = `date +%d`;
    $month = `date +%m`;
    $year = `date +%Y`;
    chomp $date;
    chomp $month;
    chomp $year;
    $date--;
    $date = "$year-$month-$date";
}

```

GET TABLE PREFIX

```

@element = split(/-/, $date);
$prefix = "port$element[0]$element[1]";

```

```
##### CONNECT TO DATABASE #####
```

```
$host = "203.146.248.242";
```

```
$database = "Thesis";
```

```
$user = "kookae";
```

```
$password = "kacadmin";
```

```
$| = 1;
```

```
my $db = Mysql->Connect($host, $database, $user, $password) or die "Cannot connect :
```

```
$Mysql::db_errstr\n";
```

```
##### GET CUSTOMER ID LIST #####
```

```
if ($customer eq "all")
```

```
{
```

```
    $check = $db->Query("SELECT CustomerID FROM Customer");
```

```
}
```

```
else
```

```
{
```

```
    $check = $db->Query("SELECT CustomerID FROM Customer WHERE  
CustomerID='$customer'");
```

```
}
```

```
##### SUMMARY IF CUSTOMER TABLE EXISTS #####
```

```
while (@line = $check->FetchRow())
```

```
{
```

```
    $sumtable = "$prefix$line[0]";
```

```
    $saction = $db->Query("CREATE TABLE $sumtable (CIP varchar(15), CPort varchar(5), IO  
char(1), Protocol varchar(5), Byte bigint(10), Date varchar(19))");
```

```
    $query = $db->Query("SELECT CIP, CPort, IO, Protocol, SUM(Byte), End FROM $line[0]  
WHERE End >= '$date 00:00:00' AND End <= '$date 23:59:59' GROUP BY CIP, CPort, IO,  
Protocol");
```

```
    while (@row = $query->FetchRow())
```

```
    {
```

```
        @element = split(/ /, $row[5]);
```

```
        $date = "$element[0]";
```

```
        $saction = $db->Query("INSERT INTO $sumtable VALUES ('$row[0]', '$row[1]', '$row[2]',  
'$row[3]', '$row[4]', '$date')");
```

```
    }
```

```
}
```


SOURCE CODE: GRAPHBYTE SCRIPT

```
#!/usr/bin/perl

#####

# FILENAME      : graphbyte                      #
# JOB           : Generate port aggregation graph (Bar Graph) #
# LAST UPDATE   : 13 September 2003              #
#####

##### CALL LIBRARY #####

use Mysql;
use FileHandle;
use File::Basename;
use strict;
use GD::Graph::lines;

##### GLOBAL VARIABLE #####

my $date      = $ARGV[0];
my $customer  = $ARGV[1];
my $type      = "Daily";
my $host      = "203.146.248.242";
my $database  = "Thesis";
my $user      = "kookae";
my $password  = "kaeadmin";
```

```

my $storage = "/var/www/html/graph/byte";
my $check = "";
my $select = "";
my $index = "";
my $title = "";
my $table = "";
my $filename = "";
my $incoming = "";
my $outgoing = "";
my $loop = "";
my @line = ();
my @info = ();
my @inbound = ();
my @outbound = ();

##### CHECK SYNTAX #####

if ($customer eq "")
{
    print "\nUSAGE : graphbyte <date> <customerid|all>\n\n";
    print "EXAMPLE\n\n";
    print "\t# Generate byte transfer graph for customer id DD001 on February 2003";
    print "\t graphbyte 2003-01 DD001\n\n";
    print "\t# Generate byte transfer graph for all customer on February 2003\n";
    print "\t graphbyte 2003-01 all\n\n";
    exit(0);
}

```

```
##### CHECK FOR DAILY OR MONTHLY FORMAT #####
```

```
my @element = split(/-/,$date);  
if ($element[2] eq "") { $type = "Monthly"; }
```

```
##### DEFINE VARIABLE FOR EACH PARAMETER #####
```

```
my $year = $element[0];  
my $month = "";  
if ($element[1] eq "01") { $month = "January"; $loop = 31; }  
elsif ($element[1] eq "02") { $month = "February"; $loop = 29; }  
elsif ($element[1] eq "03") { $month = "March"; $loop = 31; }  
elsif ($element[1] eq "04") { $month = "April"; $loop = 30; }  
elsif ($element[1] eq "05") { $month = "May"; $loop = 31; }  
elsif ($element[1] eq "06") { $month = "June"; $loop = 30; }  
elsif ($element[1] eq "07") { $month = "July"; $loop = 31; }  
elsif ($element[1] eq "08") { $month = "August"; $loop = 31; }  
elsif ($element[1] eq "09") { $month = "September"; $loop = 30; }  
elsif ($element[1] eq "10") { $month = "October"; $loop = 31; }  
elsif ($element[1] eq "11") { $month = "November"; $loop = 30; }  
elsif ($element[1] eq "12") { $month = "December"; $loop = 31; }  
my $day = $element[2];
```

```
##### ASSIGN TABLE PREFIX #####
```

```
my $prefix = "port$element[0]$element[1]";
```

```
##### CONNECT TO DATABASE #####
```

```
my $database = Mysql->Connect($host, $database, $user, $password) or die "Cannot connect :  
$Mysql::db_errstr\n";
```

```
##### GET CUSTOMER ID LIST #####
```

```
if ($customer eq "all")
```

```
{
```

```
    $check = $database->Query("SELECT CustomerID FROM Customer");
```

```
}
```

```
else
```

```
{
```

```
    $check = $database->Query("SELECT CustomerID FROM Customer WHERE  
CustomerID='$customer'");*
```

```
}
```

```
##### CHANGE TO WORKING DIRECTORY #####
```

```
chdir $storage;
```

```
##### GENERATE GRAPH FOR EACH CUSTOMER #####
```

```
while (@line = $check->FetchRow())
```

```
{
```

```
    $incoming = 0;
```

```

$outgoing = 0;
@info    = ();
@inbound = ();
@outbound = ();

$title   = "Byte Transfer Graph : $line[0]";

if ($type eq "Daily") { $filename = "$year$element[1]$day-byte-$line[0].jpg"; }
else { $filename = "$year$element[1]-byte-$line[0].jpg"; }

for ($index = 1; $index <= $loop; $index++)
{
    if ($index == 1) { $day = "01"; }
    elseif ($index == 2) { $day = "02"; }
    elseif ($index == 3) { $day = "03"; }
    elseif ($index == 4) { $day = "04"; }
    elseif ($index == 5) { $day = "05"; }
    elseif ($index == 6) { $day = "06"; }
    elseif ($index == 7) { $day = "07"; }
    elseif ($index == 8) { $day = "08"; }
    elseif ($index == 9) { $day = "09"; }
    else { $day = $index; } *

    $info[$index-1] = $index;
    $table = "$prefix$line[0]";

    $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE IO='i' AND
Date='$year-$element[1]-$day'");

    my @value = $select->FetchRow();

    if ($value[0] eq "") { $value[0] = 0; }

    $inbound[$index-1] = $value[0];

    $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE IO='o' AND
Date='$year-$element[1]-$day'");

    my @value = $select->FetchRow();

```

```

if ($value[0] eq "") { $value[0] = 0; }
Southbound[$index-1] = $value[0];
}
my @data = ([@info], [@inbound], [@outbound]);
my $graph = new GD::Graph::lines;
$graph->set(
    x_label      => 'Day of Month',
    y_label      => 'Bytes Transfer (Bytes)',
    title        => $title,
    dclrs        => [ qw(lblue lgreen) ],
    line_width   => 2,
    x_labels_vertical => 1,
    bgclr        => '#eeeeff',
) or warn $graph->error;
$graph->set_legend('Inbound', 'Outbound' );
$graph->plot(\@data) or die $graph->error;
open(GRAPH,">$filename") || die "Cannot open $filename: $!\n";
print GRAPH $graph->gd->jpeg(100);
}

```


SOURCE CODE: GRPAHPORT SCRIPT

```
#!/usr/bin/perl

#####

# FILENAME      : graphport                      #
# JOB           : Generate port aggregation graph (Bar Graph) #
# LAST UPDATE   : 13 September 2003              #
#####

##### CALL LIBRARY #####

use Mysql;
use FileHandle;
use File::Basename;
use strict;
use GD::Graph::bars;

##### GLOBAL VARIABLE #####

my $date    = $ARGV[0];
my $customer = $ARGV[1];
my $type    = "Daily";
my $host    = "203.146.248.242";
my $database = "Thesis";
my $user    = "kookae";
my $password = "kaeadmin";
```

```

my $storage = "/var/www/html/graph/port";
my $check = "";
my $select = "";
my $index = "";
my $title = "";
my $table = "";
my $filename = "";
my $incoming = "";
my $outgoing = "";
my @line = ();
my @service = ();
my @info = ();
my @inbound = ();
my @outbound = ();

##### CHECK SYNTAX #####

if ($customer eq "")
{
    print "\nUSAGE : graphport <date> <customerid|all>\n\n";
    print "EXAMPLE\n\n";
    print "\t# Generate port aggregation graph for customer id DD001 on 2 February 2003.\n";
    print "\t graphport 2003-01-02 DD001\n\n";
    print "\t# Generate port aggregation graph for customer id DD001 on February 2003";
    print "\t graphport 2003-01 DD001\n\n";
    print "\t# Generate port aggregation graph for all customer on 2 February 2003\n";
    print "\t graphport 2003-01-02 all\n\n";
    exit(0);
}

```

```
##### CHECK FOR DAILY OR MONTHLY FORMAT #####
```

```
my @element = split(/-/ , $date);
```

```
if ($element[2] eq "") { $type = "Monthly"; }
```

```
##### DEFINE VARIABLE FOR EACH PARAMETER #####
```

```
my $year = $element[0];
```

```
my $month = "";
```

```
if ($element[1] eq "01") { $month = "January";}
```

```
elsif ($element[1] eq "02") { $month = "February";}
```

```
elsif ($element[1] eq "03") { $month = "March";}
```

```
elsif ($element[1] eq "04") { $month = "April";}
```

```
elsif ($element[1] eq "05") { $month = "May";}
```

```
elsif ($element[1] eq "06") { $month = "June";}
```

```
elsif ($element[1] eq "07") { $month = "July";}
```

```
elsif ($element[1] eq "08") { $month = "August";}
```

```
elsif ($element[1] eq "09") { $month = "September";}
```

```
elsif ($element[1] eq "10") { $month = "October";}
```

```
elsif ($element[1] eq "11") { $month = "November";}
```

```
elsif ($element[1] eq "12") { $month = "December";}
```

```
my $day = $element[2];
```

```
##### ASSIGN TABLE PREFIX #####
```

```
my $prefix = "port$element[0]$element[1]";
```

```
##### CONNECT TO DATABASE #####
```

```
my $database = Mysql->Connect($host, $database, $user, $password) or die "Cannot connect :  
$Mysql::db_errstr\n";
```

```
##### GET CUSTOMER ID LIST #####
```

```
if ($customer eq "all")
```

```
{
```

```
    $check = $database->Query("SELECT CustomerID FROM Customer");
```

```
}
```

```
else
```

```
{
```

```
    $check = $database->Query("SELECT CustomerID FROM Customer WHERE  
CustomerID='$customer'");
```

```
}
```

```
##### CHANGE TO WORKING DIRECTORY #####
```

```
chdir $storage;
```

```
##### GENERATE GRAPH FOR EACH CUSTOMER #####
```

```
while (@line = $check->FetchRow())
```

```
{
```

```
    $index = 0;
```

```
    @info = ();
```

```
    @inbound = ();
```

```
    @outbound = ();
```

```
    $title = "Port Aggregation Graph : $line[0]";
```

```

if ($type eq "Daily") { $filename = "$year$element[1]$day-port-$line[0].jpg"; }
else { $filename = "$year$element[1]-port-$line[0].jpg"; }

my $portinfo = $database->Query("SELECT Port, Name FROM Service ORDER BY Port");
while (@service = $portinfo->FetchRow())
{
    $info[$index] = $service[0];
    $table = "$prefix$line[0]";
    if ($type eq "Monthly")
    {
        $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE CPort='$service[0]'
AND IO='i' AND Date>='$year-$element[1]-01' AND Date<='$year-$month-31'");
    }
    else
    {
        $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE CPort='$service[0]'
AND IO='i' AND Date='$year-$element[1]-$day'");
    }
    my @value = $select->FetchRow();
    if ($value[0] eq "") { $value[0] = 0; }
    $incoming += $value[0];
    $inbound[$index] = $value[0];
    if ($type eq "Monthly")
    {
        $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE CPort='$service[0]'
AND IO='o' AND Date>='$year-$element[1]-01' AND Date<='$year-$month-31'");
    }
    else
    {

```

```

$select = $database->Query("SELECT SUM(Byte) FROM $table WHERE CPort='$service[0]'
AND IO='o' AND Date='$year-$element[1]-$day'");
}

my @value = $select->FetchRow();
if ($value[0] eq "") { $value[0] = 0; }

$outgoing += $value[0];
$outbound[$index] = $value[0];

$index++;
}

$info[$index] = "Others";
if ($type eq "Monthly")
{
    $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE IO='i' AND
Date>='$year-$element[1]-01' AND Date<='$year-$month-31'");
}
else
{
    $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE IO='i' AND
Date='$year-$element[1]-$day'");
}

my @value = $select->FetchRow();
if ($value[0] eq "") { $value[0] = 0; }

$inbound[$index] = $value[0] - $incoming;
if ($type eq "Monthly")
{
    $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE IO='o' AND
Date>='$year-$element[1]-01' AND Date<='$year-$month-31'");
}
else

```



```

{
    $select = $database->Query("SELECT SUM(Byte) FROM $table WHERE IO='o' AND
Date='$year-$element[1]-$day'");
}

my @value = $select->FetchRow();
if ($value[0] eq "") { $value[0] = 0; }
$outbound[$index] = $value[0]-$outgoing;
my @data = (@info, @inbound, @outbound);
my $graph = new GD::Graph::bars;
$graph->set(
    x_label      => 'Port Number',
    y_label      => 'Bytes Transfer (Bytes)',
    title        => $title,
    dclrs        => [ qw(lblue lgreen) ],
    bar_width    => 10,
    bar_spacing  => 2,
    long_ticks   => 0,
    show_values  => 1,
    x_labels_vertical => 1,
    bgcolor      => '#e0e0ff',
) or warn $graph->error;
$graph->set_legend('Inbound', 'Outbound' );
$graph->plot(\@data) or die $graph->error;
open(GRAPH,">$filename") || die "Cannot open $filename: $!\n";
print GRAPH $graph->gd->jpeg(100);

$incoming = 0;
$outgoing = 0;
}

```

SOURCE CODE: GRPAHPORTPIE SCRIPT

```
#!/usr/bin/perl

#####

# FILENAME      : graphportpie                #
# JOB           : Generate port aggregation graph (Pie Graph )#
# LAST UPDATE   : 14 September 2003           #
#####

##### CALL LIBRARY #####

use Mysql;
use FileHandle;
use File::Basename;
use strict;
use GD::Graph::pie;

##### GLOBAL VARIABLE #####

my $date      = $ARGV[0];
my $customer  = $ARGV[1];
my $type      = "Daily";
my $host      = "203.146.248.242";
my $database  = "Thesis";
my $user      = "kookae";
my $password  = "kaeadmin";
my $storage   = "/var/www/html/graph/port";
```

```

my $check = "";
my $select = "";
my $index = "";
my $title = "";
my $table = "";
my $filename = "";
my $total = "";
my @line = ();
my @service = ();
my @info = ();
my @traffic = ();

##### CHECK SYNTAX #####

if ($customer eq "")
{
    print "\nUSAGE : graphportpie <date> <customerid|all>\n\n";
    print "EXAMPLE\n\n";
    print "\t# Generate port aggregation graph for customer id DD001 on 2 February 2003.\n";
    print "\t graphportpie 2003-01-02 DD001\n\n";
    print "\t# Generate port aggregation graph for customer id DD001 on February 2003";
    print "\t graphportpie 2003-01 DD001\n\n";
    print "\t# Generate port aggregation graph for all customer on 2 February 2003\n";
    print "\t graphportpie 2003-01-02 all\n\n";
    exit(0);
}

```

```
##### CHECK FOR DAILY OR MONTHLY FORMAT #####
```

```
my @element = split(/-/,$date);
```

```
if ($element[2] eq "") { $type = "Monthly"; }
```

```
##### DEFINE VARIABLE FOR EACH PARAMETER #####
```

```
my $year = $element[0];
```

```
my $month = "";
```

```
if ($element[1] eq "01") { $month = "January"; }
```

```
elsif ($element[1] eq "02") { $month = "February"; }
```

```
elsif ($element[1] eq "03") { $month = "March"; }
```

```
elsif ($element[1] eq "04") { $month = "April"; }
```

```
elsif ($element[1] eq "05") { $month = "May"; }
```

```
elsif ($element[1] eq "06") { $month = "June"; }
```

```
elsif ($element[1] eq "07") { $month = "July"; }
```

```
elsif ($element[1] eq "08") { $month = "August"; }
```

```
elsif ($element[1] eq "09") { $month = "September"; }
```

```
elsif ($element[1] eq "10") { $month = "October"; }
```

```
elsif ($element[1] eq "11") { $month = "November"; }
```

```
elsif ($element[1] eq "12") { $month = "December"; }
```

```
my $day = $element[2];
```

```
##### ASSIGN TABLE PREFIX #####
```

```
my $prefix = "port$element[0]$element[1]";
```

```

##### CONNECT TO DATABASE #####

my $database = Mysql->Connect($host, $database, $user, $password) or die "Cannot connect :
$Mysql::db_errstr\n";

##### GET CUSTOMER ID LIST #####

if ($customer eq "all")
{
    $check = $database->Query("SELECT CustomerID FROM Customer");
}
else
{
    $check = $database->Query("SELECT CustomerID FROM Customer WHERE
CustomerID='$customer'");
}

##### CHANGE TO WORKING DIRECTORY #####

chdir $storage;

##### GENERATE GRAPH FOR EACH CUSTOMER #####

while (@line = $check->FetchRow())
{
    $index = 0;
    @info = ();
    @traffic = ();

    $title = "Port Aggregation Graph : $line[0]";
    if ($stype eq "Daily") { $filename = "$year$element[1]$day-portpie-$line[0].jpg"; }

```

```

else { $filename = "$year$selement[1]-portpie-$line[0].jpg"; }
my $portinfo = $database->Query("SELECT Port, Name FROM Service ORDER BY Port");
while (@service = $portinfo->FetchRow())
{
    $info[$index] = $service[0];
    $stable = "$prefix$line[0]";
    if ($type eq "Monthly")
    {
        $select = $database->Query("SELECT SUM(Byte) FROM $stable WHERE CPort='$service[0]'
AND Date>='$year-$selement[1]-01' AND Date<='$year-$month-31'");
    }
    else
    {
        $select = $database->Query("SELECT SUM(Byte) FROM $stable WHERE CPort='$service[0]'
AND Date='$year-$selement[1]-$day'");
    }
    my @value = $select->FetchRow();
    if ($value[0] eq "") { $value[0] = 0; }
    $total += $value[0];
    $traffice[$index] = $value[0];
    $index++;
}
$info[$index] = "Others";
if ($type eq "Monthly")
{
    $select = $database->Query("SELECT SUM(Byte) FROM $stable WHERE Date>='$year-
$selement[1]-01' AND Date<='$year-$month-31'");
}
else

```

```

}

$select = $database->Query("SELECT SUM(Byte) FROM $table WHERE Date='$year-
$element[1]-$day'");

}

my @value = $select->FetchRow();
if ($value[0] eq "") { $value[0] = 0; }

$traffic[$index] = $value[0]-$total;

my @data = ([@info], [@traffic]);

my $graph = new GD::Graph::pie(275, 275);
$graph->set(
    label      => 'Bytes Transfer',
    title      => $title,
    clrsl      => [ qw(lblue lgreen lred lyellow cyan lorange lpurple gold gray pink marine lbrown) ],
    start_angle => 90,
    suppress_angle => 5,
    '3d'       => 1,
    bgcolor    => '#eeeeff',
) or warn $graph->error;
$graph->plot(@data) or die $graph->error;
open(GRAPH,">$filename") || die "Cannot open $filename: $!\n";
print GRAPH $graph->gd->jpeg(100);

$total = 0;

}

```


DATA DICTIONARY OF DATABASE

Table Name: Customer

Field	Type	Null	Key	Default	Extra
CustomerID	varchar(5)	No		NULL	
CustomerName	varchar(255)	YES		NULL	
Address	varchar(255)	YES		NULL	
ContactPerson	varchar(255)	YES		NULL	
EngineerName	varchar(255)	YES		NULL	
IPDetails	varchar(255)	YES		NULL	
ServicesType	varchar(255)	YES		NULL	

Table Name: Connection

Field	Type	Null	Key	Default	Extra
CustomerID	varchar(5)	No		NULL	
EqIP	varchar(15)	YES		NULL	
PortNo	varchar(10)	YES		NULL	
IntIndex	char(3)	YES		NULL	

Table Name: Ratio

Field	Type	Null	Key	Default	Extra
Type	varchar(25)	YES		NULL	
Byte	bigint(10)	YES		NULL	
Baht	int(5)	YES		NULL	

Table Name: Services

Field	Type	Null	Key	Default	Extra
Port	int(5)	YES		NULL	
Name	varchar(15)	YES		NULL	

Table Name: DD001(CustomerID)

Field	Type	Null	Key	Default	Extra
CIP	varchar(15)	YES		NULL	
CPort	varchar(5)	YES		NULL	
NIP	varchar(15)	YES		NULL	
NPort	varchar(5)	YES		NULL	
IO	char(1)	YES		NULL	
Protocol	varchar(5)	YES		NULL	
Byte	bigint(10)	YES		NULL	
End	varchar(19)	YES		NULL	
Duration	varchar(4)	YES		NULL	

This table will created after running Filter Script. The Filter script will translate raw data into readable format; separate data of traffic flow into data of each customer, and keep all of data into database. This Table will create automatically to keep data of each customer after separated. Table Name will be named base on ID of each customer.

Table Name: 200309DD001(YY:MM:CustomerID)

Field	Type	Null	Key	Default	Extra
CIP	varchar(15)	YES		NULL	
CPort	varchar(5)	YES		NULL	
NIP	varchar(15)	YES		NULL	
IO	char(1)	YES		NULL	
Protocol	varchar(5)	YES		NULL	
Byte	bigint(10)	YES		NULL	
Date	varchar(19)	YES		NULL	

This table will created after running Summary Script at the end of day. The Summary script will summary traffic flow data usage of each customer in each day and keep all of data into database. This Table will create automatically to keep summary data of each customer in each day. Table Name will be named base on Year: Date: CustomerID.

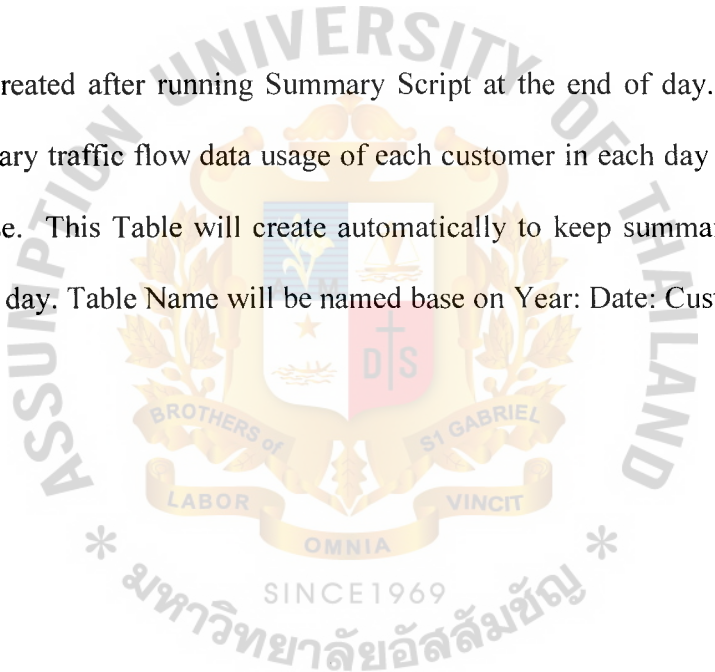


Table Name: Port200309DD001(Port:YY:MM:CustomerID)

Field	Type	Null	Key	Default	Extra
CIP	varchar(15)	YES		NULL	
CPort	varchar(5)	YES		NULL	
IO	char(1)	YES		NULL	
Protocol	varchar(5)	YES		NULL	
Byte	bigint(10)	YES		NULL	
Date	varchar(19)	YES		NULL	

This table will created after running Port Script at the end of day. The Port script will summary base on the same service port usage of each customer in each day and keep all of data into database. This Table will create automatically to keep summary services port usage of each customer in each day. Table Name will be named base on Port: Year: Month: CustomerID.

