Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

LIANG, Huan

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Minimal Cost Design of Virtual Private Network

L. Orozco-Barbosa

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

A. Karmouch

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

I. Lambadaris

O. Yang

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Minimal Cost Design of Virtual Private Network

By
Huan Liang

A thesis submitted to
the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of
Master of Applied Science

School of Information Technology and Engineering
University of Ottawa
April 2003

Your file Votre référence
ISBN: 0-612-90109-2
Our file Notre référence
ISBN: 0-612-90109-2

Abstract

VPN technology is an attractive cost-effective solution for the support of the networking needs of enterprises. In this work, we review current issues in the development of VPN technology. We then focus our research on minimal cost design, used by network-based IP VPN service providers. The interest in such solutions is generated by both customers seeking to reduce support costs and by Internet Service Providers (ISPs) seeking new revenue sources. Solving the cost minimization would allow ISPs to define and deploy new VPN services.

In this thesis, Multicommodity Min-Cost Flows (MMCF) formulations are applied to the resource allocation in network-based IP VPN, in order to develop a cost-effective routing proposal. Compared with RFC 2676, one of the Open Shortest Path First (OSPF) algorithms, various improvements in routing costs are obtained corresponding to different proposed network topologies.

Acknowledgement

I am deeply grateful to my supervisors Dr. Luis Orozco-Barbosa and Dr. Dimitrios Makrakis for their consistent knowledgeable guidance, extremely helpful comments and suggestions that helped to improve this work.

I would also like to express my appreciation to Dr. Ognian Kabranov for his fruitful discussions and help in publishing our research achievements. Many thanks to my friend Miguel Lopez-Guerrero for his encouragement and always being available to help.

Finally, a special gratitude goes to my dear family for their unfailing love and support.

Table of Contents

List of Figures

List of Tables

List of Symbols

G (N, A)	a directed network graph with N of nodes and A of connection pairs of nodes from N within the network;
u_{ij}	capacity (or upper bound) of flow from node *i* to node *j*;
b_i	net flow generated at node *i*;
K	set of demanded commodities (traffic connection requests) in the network;
h	one commodity of set K, i.e one traffic channel;
(i, j)	arc with source node *i* and destination node *j;*
c_{ij}^h	cost per unit for commodity *h* on *arc (i, j);*
x_{ij}^h	a flow of commodity *h* on *arc (i, j);*
b_i^h	commodity net flow generated at node *;*
T_D^h	The delay requirement of the VPN connection channel *h;*
T_{PD}^h	The delay of the VPN traffic flow *h* traversing path *p;*
μ	A link's service rate;
λ	A link's traffic arrival rate;
μ_{ij}	The service rate of link (*i, j*);
λ_{ij}	The arrival rate of link (*i, j*);
D_{ij}	The propagation delay on link (*i, j*);

T_{AQD}	The average queuing delay that a packet traverses the network;
T_{AD}	The average delay (including queuing and propagation delay) that a packet traverses the network;
γ	The total arrival rate in the network system;
$\bar{n}$	The average number of hops that a packet must pass through the network to reach its destination;
$\overline{n_{ij}}$	Average number of hops for a single origin-destination pair (i, j);
H	the number of hops that the pair needs to build a path;
P_H	the probability of a path with H hops;
ρ	the network's utilization.

List of Acronyms

ATM	Asynchronous Transfer Mode
CLE	Customer Located Equipment
CPE	Customer Premises Equipment
DiffServ	Differentiated Services
DSL	Digital Subscriber Lines
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	IP security protocol
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
LAN	Local Access Network
L2TP	Layer-2 Tunneling Protocol
MMCF	Multicommodity Min-Cost Flows
MPLS	Multiple Protocol Label Switch
NOC	Network Operation Center
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
POP	Points of Presence
PPTP	Point-to-Point Tunneling Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
SLA	Service Level Agreement
VPN	Virtual Private Network
VPNOC	VPN Network Operation Center
WAN	Wide Area Networks

Chapter 1. Introduction

Virtual Private Network (VPN) technology has proven to be a cost-effective solution for the deployment of private networks over wide areas [1]. However, is it possible to find a more cost-saving solution for the VPN service provider compared to what is presently available? This is the question we address in the present thesis. Throughout our work, a novel routing algorithm is proposed, simulated, analyzed and compared [2]. Our research focused on "network-based IP VPN", where the operation of the VPN is outsourced to an Internet Service Provider (ISP).

1.1 Motivation

VPN solutions generally can be classified as Frame Relay or ATM VPN, Multiple Protocol Label Switch (MPLS) VPN and Internet Protocol (IP) VPN [3]. While the public network infrastructure is used, from the customers' point of view, it appears as if they have their own private networks. Depending on different application scenarios each approach has its advantages and drawbacks. However, among these existing solutions, IP VPN is a good choice in terms of saving cost and making the private networks seamless to the rapid growing IP based applications [4]. As more and more research and development efforts are spent on IP based networking, it is important that more attention should be paid to IP VPN networking.

Most studies on VPN have focused on the technologies of security and Quality-of-Service (QoS). However, it is almost impossible that IP VPN can perform better than the traditional private networks in terms of security and QoS. Cost-effectiveness and seamless features are the main factors driving the use of IP VPNs. Actually this is a network optimization problem, where security, QoS and cost-effectiveness should all be comprehensively considered. However, the combination of these factors has not been considered well in VPN designs. In particular, the following issues still have to be addressed:

1. In single domain networks, Open Shortest Path First (OSPF) is the typical routing algorithm. A draft has been placed forward within the Internet Engineering Task Force (IETF) to address the QoS issue, the RFC2676 (OSPF extensions with QoS routing mechanisms). However, this proposal needs modifications in order to be applicable to IP VPN networks.

2. Because IP VPNs are inexpensive solutions, the optimization of their costs has been ignored. However, the lower cost is associated mostly with best effort types of networks. When service differentiation and resource reservation (in order to provide QoS support) come into the picture, the cost issue becomes significant and has to be taken into consideration. A cost efficient QoS capable VPN will provide the operator with a significant competitive advantage.

Realizing the importance of cost when operating QoS capable VPNs, we decided to focus our research in developing cost minimization optimal routing solutions for VPN networks.

- Among the several VPN solutions, we selected Network-based IP VPN as our research subject. When compared with other VPN solutions, IP VPN seems more attractive and promising. As we know, IP VPN is more scalable because of its public connectivity nature and it is also cheaper than layer 2 VPN solutions [5]. However, as it was mentioned in references [4] and [6], there are three approaches today to implement IP VPN (see the following list).

- CPE-based (Customer Premises Equipment) IP VPN: Tunnels are established only between the CPE devices, which mean that the service provider's routers are VPN-disabled.
- CLE-based (Customer Located Equipment) IP VPN: Equipment owned and operated by the service provider but located in the customer's premises.
- Network-based IP VPN: Equipment is located in the service provider's premises at the edge of its network.

Network-based IP VPN is expected to predominate because of its economic advantages to the small and medium-sized businesses, which are possibly the largest customers group of IP VPN users [4]. Therefore, among the three IP VPN approaches, Network-based IP VPN has been chosen as the research subject.

Since the research subject has been selected, the next question to be answered is: what kind of technologies will be included in our study. Security, QoS and connectivity are all critical for the service providers. In the case of IP VPN, the costs of security

technologies such as IPSec and PPTP are fixed. Therefore, except security, costs of both QoS and connectivity will be considered in this research.

In general, our research focuses on the minimal cost design for network-based IP VPN service providers with consideration to both QoS and connectivity issues.

1.2 Contributions

The Multicommodity Min-Cost Flows (MMCF) formulation was applied to the resource allocation in network-based IP VPN. A minimal cost VPN tunnel is selected by using the network flow optimization based on the proposed network management system.

MMCF minimizes the cost of operation. The novelty is that the service provider's transportation cost is integrated in the routing metric. The proposed scheme outperforms RFC 2676, based on the performance analysis (contacted through simulations). We have confirmed that it offers cost savings of as high as 9%.

In summary, using MMCF to route traffic flows in one domain proved to be more efficient than OSPF.

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 provides a literature review of VPN technologies and network cost design. It also includes a survey of VPN customers' requirements and the service provider's available support. Chapter 3 describes the proposed methodologies, used to solve the optimization problem.

In Chapter 4, both network topology and software simulation models are established. Chapter 5 provides the performance analysis results and a thorough discussion. Chapter 6 concludes this thesis.

Chapter 2. Literature Review

In this chapter, a variety of VPN issues such as VPN concepts, types, benefits, applications, requirements and implements are explained. The principles of network cost design are introduced as well.

2.1 What is a Virtual Private Network?

A private network could be understood as Intranet support services, such as electronic mail, web surfing, database and groupware to authorized users. In the 1990s, private networking services had been widely used by organizations to communicate with their communities [7].

At the beginning of the Internet's appearance, security technology was not mature enough to enable business communications on the public Internet infrastructure. This security absence was the main reason for the existence of enterprise network architectures using private and public networks, which contain firewalls that are the line of demarcation between the two domains as shown in Figure 2.1. The drawback of this architecture is the increased cost and complexity it adds to the handling of applications.

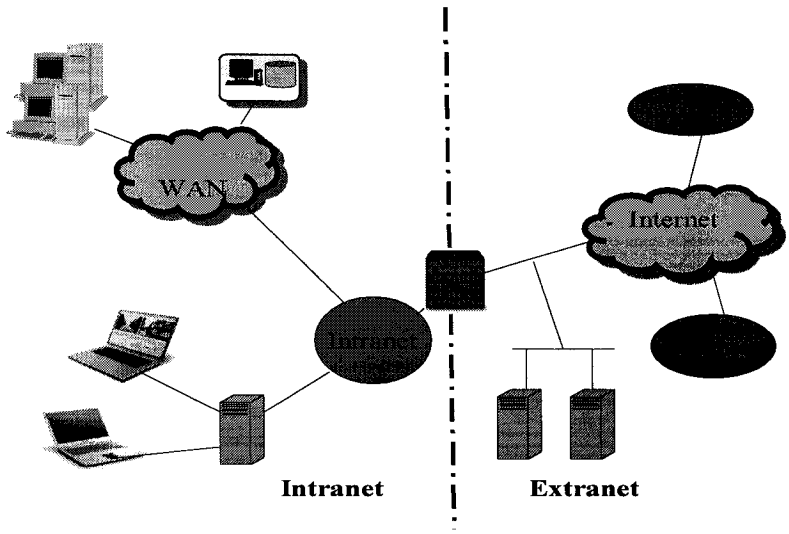


Figure 2.1: Typical Enterprise Architecture – Private and public segmentation

To simplify the application environments and reduce the leased line cost for private networks, security technologies were developed, enabling the public Internet to play the role of a private enterprise backbone. Not only IP, but also Frame Relay or Asynchronous Transfer Mode (ATM)-based networks can be used to deploy VPN.

2.2 VPN Types

Following the Open Systems Interconnection (OSI) model, VPN solutions can be classified as Link-Layer VPN (e.g. Frame Relay or ATM-based VPN) and Network-Layer VPN (e.g. IP VPN) [8]. Depending on different application scenarios, they all have their advantages and drawbacks. For instance, Frame Relay is considered as inherently secure because of the fact that it uses layer 2 technologies but compared with IP VPN it is not as scalable as IP VPN. MPLS VPN is more scalable than Frame Relay. However, it requires all sites to be tied into the same service provider and does not lend itself to remote access from remote dialup users. IP VPN is cheaper, easy to build and

has a clear advantage in remote access applications. In addition, the latency of IP connections is expected to improve [6].

2.3 VPN Benefits

Many circumstances are currently inducing separate private networks and public networks to converge towards virtual private networks because of the following reasons:

(1) Cost-effective benefits:

- Elimination of long-distance charges for leased lines and switched services;
- Companies pay only for the actual usage of data transferred;
- Minimization of end-user network design and management responsibilities [9].

(2) Flexibility benefits:

- Service providers in nearly every city create a worldwide presence;
- Connections can instantaneously be added and deleted [9];
- Permanent, periodic or temporary connectivity can be provided as needed [9].

2.4 VPN Applications and Requirements

As mentioned in [10], VPN service providers can deploy the following applications to meet various customers' requirements as shown in Figure 2.2, where all the communications are authenticated and encrypted.

- Worldwide remote LAN access [10]: suitable for mobile users, small/remote offices and home offices to access the company's VPN;

- Intranets [10]: Core part of the enterprise's connectivity. Resources such as databases, mainframes, email servers, etc. are shared among the enterprise's headquarters and branch offices;
- Extranets [10]: Provide limited access for the enterprise's partners, suppliers and customers for business-relationship purposes.

Depending on the applications, different VPNs are implemented by choosing different connections and authorities.

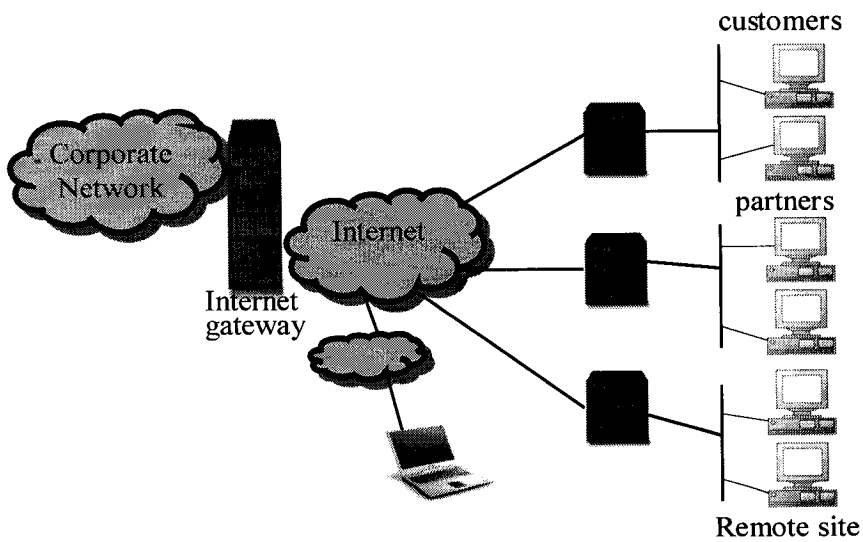


Figure 2.2: Typical VPN applications

2.4.1 VPN Connections

According to their desired applications, VPN users can choose different access methods. For instance, a remote user can select either dial-up from the Public Switched Telephone Network (PSTN) or continuous forms of access such as Digital Subscriber Lines (DSL) to connect with their headquarters via Internet. Both data and voice information between branches and headquarters may be transmitted on Frame Relay,

ATM or leased lines such as T3, OC3 and OC48, etc. Certainly, public Internet access can be used wherever it is preferred.

2.4.2 VPN Requirements

For the network, there are numerous possible requirements. However, in the case of VPN, four essential issues are being considered: compatibility, security, availability and manageability [4].

- Compatibility: Compatible with multi-protocol environments;
- Security (as mentioned in [9]):
  - Protection of resources through firewalls
  - Proof of identity through both user and packet authentication
  - Privacy of information through packet encryption;
- Availability: three dimensions – uptime, throughput and latency [11];
- Manageability: Manage the entire private/public VPN end-to-end.

2.5 VPN Implementations and QoS Considerations

According to the technologies and application requirements discussed above, VPN implementations come to the agenda. Certainly, QoS issues are the critical concerns when the VPN ideas are applied to network services.

2.5.1 VPN Implementations

As Ascend Communications, Inc mentioned in reference [9], all Virtual Private Networks are made up of five components: "Enterprise Equipment, WAN Access

Services, Network Access Switch, Backbone such as Internet and Security & Management". The VPN implementations must cover these elements and apply them properly. Generally, VPN-capable systems should include features, such as tunneling protocols, software upgrades and dynamic bandwidth management [12], among others to meet the VPN requirements discussed in Section 2.4.2.

- Enterprise Equipment [9]:

  Equipment at the customer's premise location is believed to be an end-point where a VPN begins. However, the type of the required equipment depends on the VPN architecture. The equipment includes: routers or remote access servers with tunneling and security capabilities.

- WAN Access Services [9]:

  Local WAN access services connect the enterprise with the VPN service provider's point of presence (POP). Essentially, there are three service choices: dial-up services, continuous access services and direct frame relay or ATM access for large sites.

- Network Access Switch [9]:

  Network Access Switch resides in the service provider's POPs. It is required to have a variety of local WAN access options, tunneling protocols supports, interface to frame relay or to ATM directly and to offer enough security provisions.

- Backbone/Internet [9]:

The service provider's backbone and/or Internet appear to be the wide area network for VPNs. Considering the volume of traffic flow, high performance switches and routers are required.

- Security & Management [9]:

  VPN environments require visibility and control of all equipment and links from one end to another [9]. Therefore, network management is able to view the network's entirely and its details logically and physically. Real-time network monitoring and analysis of traffic statistics are expected. Normally, VPN management is a shared responsibility between the VPN customer and its service provider [4].

2.5.2 QoS Considerations in VPN Implementations

A Service Level Agreement (SLA) signed between the VPN customer and a service provider should be guaranteed. In the SLA, bandwidth, latency, throughput and availability should all be considered. Regarding the different connection methods, the following observations could be made:

- Dedicated leased lines connection can establish fixed available resource levels;
- Layer 2-based VPN such as Frame Relay or ATM can also provide a reliable service level to VPN because of its ability to provide virtual circuits;
- IP based VPN also possesses the relative merits of various Internet QoS mechanisms. However, applying VPN on public Internet makes QoS guarantees more challengeable than the other implementations.

2.6 Network Cost Design

Network design with low cost is one of the goals. However, the cheapest design may not be acceptable for reasons of reliability and performance.

In the case of VPN network cost design, the issues to be considered are what applications the VPN will serve, the type of data, the number and classes of users and what specific restrictions apply to user groups. Moreover, we might restrict user access to some special locations or resources or assign QoS levels.

On the other hand, VPN pricing for the users should be taken into consideration in order to be more competitive in today's challenging market. The pricing factors include [13]:

- Type of VPN implementations;
- Number and speed of VPN sites;
- Remote access method;
- Network transportation;
- Equipment (hardware and software) used;
- Management services;

Therefore, ISPs also need to design the pricing model to calculate the price for their VPN users and submit a competitive service agreement to the customers. In Chapter 4, the cost design is described in detail.

Chapter 3. Methodology

In this chapter, we survey VPN technologies and introduce the cost design methodology. Regarding QoS requirements, the "QoS Routing Mechanisms and OSPF Extensions" (RFC 2676) is discussed. Then, our proposal of minimal cost design with QoS guarantees is presented.

3.1 Survey on VPN Market

For quite a long time, network service providers were focusing on providing mainly connectivity. However, the VPN technology now gives them the chance to offer a variety of value-added services and a greater return on assets. Besides, by providing low cost connectivity, service providers can also deal with the management and end-user equipment support for the benefit of the VPN customers.

3.1.1 Customer's Requirements

Security and quality of service support appear to be of the highest importance. In what follows, we explain these two issues in more detail.

(1) Security Requirements

The customer's resources must be protected from unauthorized access. Authentication should be applied to verify the user's identity. Packet encryption is required to prevent eavesdroppers from reading.

(2) QoS Requirements

The primary goal of QoS is to provide some guarantees such as dedicated bandwidth, controlled jitter and latency (required by some real-time and interactive traffic) and packet loss characteristics [5]. Currently, the customer's QoS requirements are specified in the SLA.

The customer's requirements drive today's telecommunications market. The service provider who can offer the appropriate service support can achieve a better market acceptance.

3.1.2 Survey on VPN Service Providers' Support

To meet the VPN customers' requirements as mentioned above in section 3.1.1, service providers should at least provide the following support:

(1) Security Support

Security of IP-based VPN is a major concern. VPN service providers are expected to offer highly secure tunnels through various encryption and authentication technologies [14]. Communications through the Internet interconnecting remote users, branch offices and partners to corporate headquarter networks need to be secured. Figure 3.1 shows a typical implementation of an IP VPN with security considerations, as described by Gregory J. Ciolek's in reference [15].

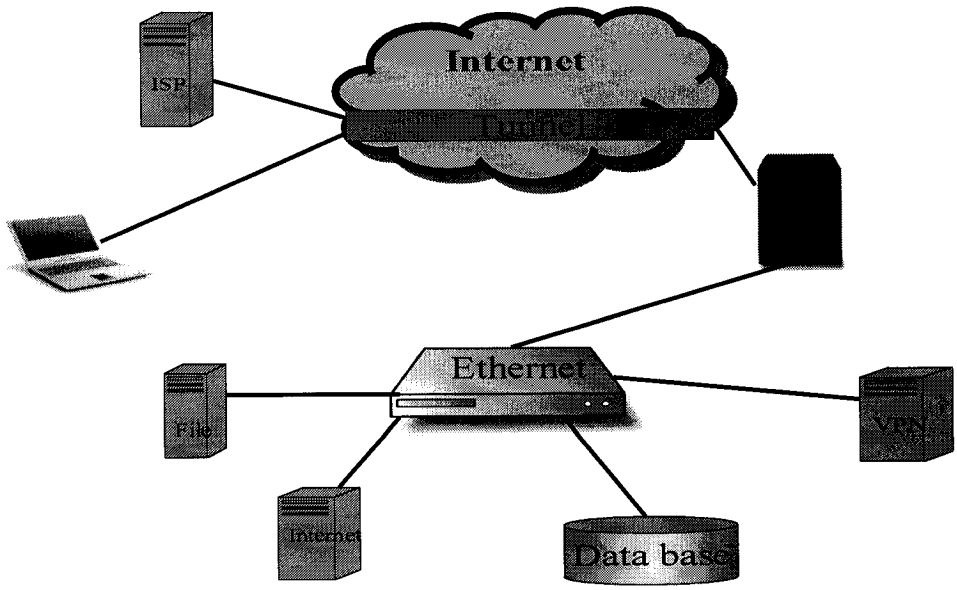


Figure 3.1: IP VPN security

Tunneling is the most popular option to make non-IP and private IP networks compatible to Internet, so that different network protocols can be integrated on a common backbone. As shown in Figure 3.1, a secure connection is established by tunneling through the Internet. Tunneling is the encapsulation of a message packet within an IP packet for transmission across the Internet [16]. The encapsulated packet is striped from the IP packet at the receiving network.

So far, three protocols are applied for tunneling through the Internet: Internet Protocol Security (IPSec) [17], Point to Point Tunneling Protocol (PPTP) [18] and Layer Two Tunneling Protocol (L2TP) [19][20]. All three protocols adhere to the basic Internet security standards [15].

(2) QoS Support

Based on our investigation on different QoS solutions, the following comments are worth mentioning:

The first generation of Internet is based on what is known as best effort service [21]. Putting it plainly, in a best effort network, all applications are treated equally, regardless of their actual needs. While this model was able to support non-real time applications such as ftp, r-login, e-mailing and web surfing, it is inadequate for supporting real time applications such as video conferencing, telephony, etc. Acknowledging the fact that the Internet is becoming a global network and having to address the need for enabling it to become a truly integrated services network, the Internet Engineering Task Force (IETF) [22] produced a series of architectures and protocols and capable of making the Internet QoS-enabled. The examples of these architectures and protocols are the Integrated Services architecture (IntServ) [23] (suitable mostly for Intranet environments due to its scalability problems), Differentiated Services [23], and provision connection-oriented capabilities supported though the Multi-Protocol label Switching (MPLS) protocol [24]. Considering the evolving needs of users, it is reasonable to develop a QoS-enabled routing algorithm in order to minimize the delay.

3.1.3 Survey on Service Provider's Revenue Sources

The main criteria where the development of successful VPN technology should be based on the provision of better quality of service, cost efficiency (resulting in savings for the users) and revenue generation. In terms of revenue, the main revenue sources are the followings:

(1) VPN management

Managing the customer's private network generates revenue potential to the service provider. VPN management tasks include network monitoring, billing, provisioning and configuration. Certainly, maintenance and other related applications can be included. The revenue generated through VPN management depends on the degree of involvement the service provider has in the management of the customer's network. Reference [25] shows that end-to-end VPN management, where the VPN customer shares the service provider's VPN equipment, can contribute to the ISP high management revenue because of its high degree of involvement.

(2) VPN Connectivity

The revenue made by VPN service providers through connectivity is the price difference between the cost of leased lines and the payments received by their customers.

3.2 OSPF Algorithm Overview

As mentioned in Chapter 1, our research is concerned with the design of an algorithm capable of minimizing the cost for a network-based IP VPN service provider that operates in one domain. OSPF (Open Shortest Path First) algorithm is the most widespread routing solution for a single-domain network. The proposed routing algorithm will be evaluated by comparing its performance to the performance of the QoS-enabled OSPF routing protocol that is described in RFC 2676 [26].

3.2.1 Evolution of the OSPF Algorithm

OSPF is an industry standard protocol developed by the Internet Engineering Task Force (IETF). As a link state routing protocol [27], OSPF maintains a topological database, which stores related information of the autonomous network state and uses it to calculate the shortest path. Link-state information is exchanged in the form of LSA (link-state advertisements). According to the OSPF version 2, described in RFC 2328 [27], LSAs are exchanged every 30 minutes, unless there is a change in the network topology. To apply the QoS mechanism to the network, " QoS Routing Mechanisms and OSPF Extensions" (RFC 2676) has been developed.

3.2.2 RFC 2676: QoS Routing Mechanisms and OSPF Extensions

The focus of RFC 2676 is on the algorithms used to compute QoS suitable routes and on the necessary modifications to OSPF in order to support this function [26]. The purpose of RFC 2676 is to identify possible approaches to allow the deployment of QoS routing capabilities with the minimum possible impact to the existing routing infrastructure [26]. It was mentioned in Section 1.1 "Overall Framework" in RFC2676 that QoS-capable routers in the network are assumed to identify and advertise resources that remain available to new QoS flows [26]. RFC 2676 assumes that each router maintains an updated database of the network topology, including the current state (available bandwidth and propagation delay) of each link [26]. Also, in Section 4.2.1 of RFC 2676, it is mentioned that their implementation relies on the assumptions such as: "the scope of QoS route computation is currently limited to a single area and all routers within the area are assumed to run a QoS enabled version of OSPF. All interfaces on a router are assumed to be QoS capable." [26].

As it was mentioned in Section 2 "Path Selection Information and Algorithms" of the RFC 2676 document [26], the metrics include link available bandwidth, link propagation delay and hop-count. Regarding the above metrics, the following explanations in the document are noteworthy:

(1) Link available bandwidth is the "current amount of available (i.e., unallocated) bandwidth"[26]. Changes in this metric should be advised in order to provide accurate information to the path selection algorithm.

(2) Link propagation delay is "meant to identify high latency links, e.g., satellite links, which may be unsuitable for real-time request" [26]. This metric is designed to prevent a delay-sensitive flow from selecting a path using a satellite link, which was mentioned in Section 1.2 "Simplifying Assumptions" of RFC 2676 [26]. Their approach in the document is "to assign delay-sensitive flows to a 'policy' that would eliminate from the network all links with high propagation delay, e.g., satellite links, before invoking the path selection algorithm." [26].

(3) Hop-count is a measure of the path cost. The smaller amount of hops a path establishes, the fewer network resources are consumed. Therefore, the path selection algorithm aims at finding the minimum hop path, which is able to satisfy the requirements of a connection request [26].

As it was discussed in Section 2.3 "Path Selection" in RFC 2676, "the cost of a path is a function of both its hop count and the amount of available bandwidth." [26]. Standard routing algorithms are typically optimized according to a single objective

(which means either minimal hop-count or maximal path bandwidth in the above cost function). RFC 2676 essentially performs a minimum hop path computation and then, it also maintains a list of equal cost hops, which are sorted by the available bandwidth. Therefore, it allows the selection of the minimum hop path with the maximum available bandwidth.

As it was mentioned earlier, RFC 2676 is currently limited to a single domain where all the routers are assumed to perform QoS-capable routing. The following algorithm we proposed to minimize the cost works also in one domain with all QoS-enabled routers.

3.3 Proposed Algorithm

As mentioned in Section 1.2, the research focuses on the development of a routing scheme that is based on the minimal cost design, as it applies to the case of an IP VPN network. According to the market survey [13][28], the IP VPN billing includes a basic service fee charge for the Customer Premises Equipment, site-to-site connectivity plus end-to-end management and monitoring. Each site within the VPN receives a monthly bill that is mostly based on the access speed (bandwidth). The provisioning and ongoing maintenance of the local loop, while part of the managed service, is not included in the monthly service fee.

These fees that apply for VPN customers are flexible because of the market competition. Therefore, if a service provider expects to survive in the price competition and maximize its revenue, it has to keep its price within the margin between the amount it charges and its cost price per unit. A way to do so is to reduce the cost of connectivity

by enabling the routing algorithm to select the lowest cost path for a certain VPN customer, while the SLA requirements are respected. Towards this end, the Multicommodity Min-Cost Formulation (MMCF) is proposed where the lines of a network are also called arcs. Please note that from now on, the term 'link' in the thesis will be replaced with the term 'arc'.

3.3.1 Multicommodity Min-Cost Flows Formulation

The minimum cost flow problem is a network optimization problem. MMCF applies to flows in networks with capacities and costs for flows through arcs. It also allows multiple sources and destinations. The purpose of the min-cost flow is to minimize the total cost subject to the availability and service demand at the nodes and its upper bound while the traffic flow travels each arc [29].

Consider a network graph, G = (N, A), it is an unemptied finite set with N of nodes and A of connection pairs of nodes from N within the network. Thus, a pair of nodes in A such as (i,j) is a flow arc. For instance, Figure 3.2 presents a network with 4 nodes, where N = { 1, 2, 3, 4} and A = { (1, 2), (1, 4), (2, 4), (3, 1), (3, 2) }.

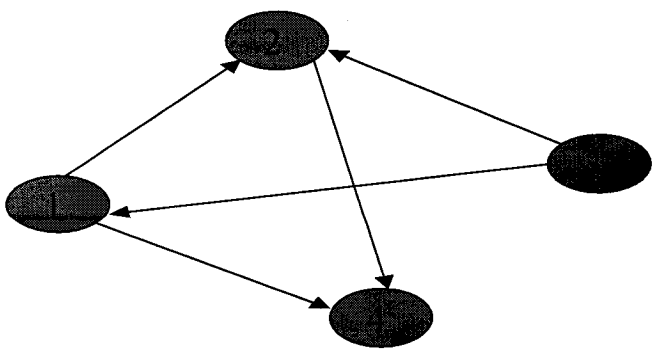


Figure 3.2 A network example

We assumed the following parameters:

u_{ij} : capacity (or upper bound) of flow from node *i* to node *j*;

b_i : net flow generated at node *i*;

K: set of demanded commodities (traffic connection requests) in the network;

h: one commodity of set K, i.e one traffic channel;

c_{ij}^h : cost per unit for commodity *h* on *arc (i, j);*

x_{ij}^h : flow of commodity *h* on *arc (i, j);*

b_i^h : commodity net flow generated at node *i.*

Using this notation we can formulate the Multicommodity Min-cost Flows problem (MMCF) as follows [2][29]:

$$\min \sum_{h \in K} \sum_{(i,j) \in A} c_{ij}^h \bullet x_{ij}^h \qquad (1)$$

$$\sum_{j:(i,j) \in A} x_{ij}^h - \sum_{j:(j,i) \in A} x_{ji}^h = b_i^h \qquad \forall i \in N, \qquad \forall h \in K \qquad (2)$$

$$x_{ij}^h \geq 0 \qquad \forall h \in K, \quad \forall (i,j) \in A \qquad (3)$$

$$\sum_{h \in K} x_{ij}^h \leq u_{ij} \qquad \forall (i,j) \in A \qquad (4)$$

Equation (1) requires the route of "commodities" on the network at a minimal total cost, respecting constraints in equation (2), (3) and (4). Equation (2) is the node balance constraint, which can be explained by the following Figure 3.3. Equation (3) prevents the flow of channel *h* that is transported on link (*i, j*) from taking a negitive value.

Equation (4) constrains the sum of all flows going from node i to node j to prevent it from exceeding the link capacity u_{ij}.

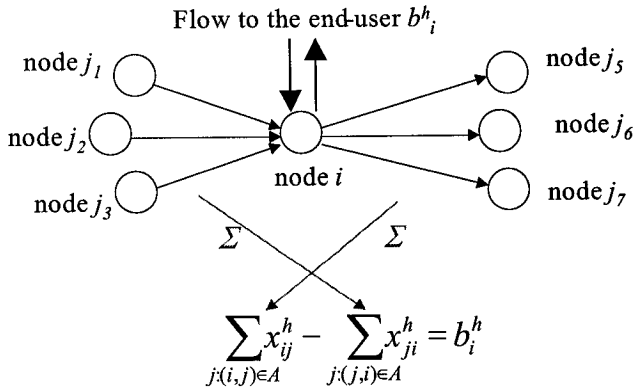


Figure 3.3 Equation (2) flow conservation at node i

The result of solving this optimization problem is the flow allocation x^h_{ij}. The allocation provides the amount of bandwidth allocated for every connection h on the corresponding physical link (i,j), which will lead to the minimization of the operational cost of the network .

3.3.2 Applying MMCF on VPN Routing with Bandwidth and Delay Considerations

In order to apply the MMCF formulation to the VPN network in a realistic manner, we assume the following constraints:

- The network links are bi-directional;
- The VPN is represented as a demand traffic matrix;
- The capacity of the physical links are limited by an upper bound;

- The Service Level Agreement between ISP and VPN customers is satisfied with bandwidth reservation and delay requirement satisfaction for the VPN connection.

Based on the recent marketing price survey dated by 2001 [30][31], we assume the cost for the ISP for 1Mbps bandwidth using T3 links is $667 per month, while the cost for using OC3 is $267 per month in the thesis. Thus, the cost per unit c_{ij}^h is defined by these specific values for their corresponding links.

As mentioned earlier in Section 3.3.1, the MMCF formulation presented in Equations (1) to (4) addresses only bandwidth allocation $x^h{}_{ij}$ and its associated constraints. In this section, we extend the formulation to include the delay constraints as well. Therefore, we define T_D^h as a delay requirement of the VPN connection channel *h*. T_{PD}^h is defined as the delay of the VPN traffic flow *h* traversing path *p*. Generally, when we mention delay in the following content, it consists of queuing delay and propagation delay. In order to incorporate both the bandwidth and delay constraints in the optimization process, the following equation (5) should also be included in the MMCF formulation:

$$\min \sum_{h \in K} \sum_{(i,j) \in A} c_{ij}^h \bullet x_{ij}^h \qquad (1)$$

$$\sum_{j:(i,j) \in A} x_{ij}^h - \sum_{j:(j,i) \in A} x_{ji}^h = b_i^h \quad \forall i \in N, \quad \forall h \in K \qquad (2)$$

$$x_{ij}^h \geq 0 \qquad \forall h \in K, \quad \forall (i,j) \in A \qquad (3)$$

$$\sum_{h \in K} x_{ij}^h \leq u_{ij} \qquad \forall (i,j) \in A \qquad (4)$$

$$T_{PD}^h \leq T_D^h \quad \forall h \in K \qquad (5)$$

We further adopt a M/M/1[1] queuing model for each link in the network to calculate the delay of traffic flow traversing a path from its source to its destination. The queuing delay of a M/M/1 model is given by $E(T) = \frac{1}{\mu - \lambda}$ [32], where μ is the service rate and λ is the traffic arrival rate. For the link (i, j), μ_{ij} is its service rate, λ_{ij} is the arrival rate and D_{ij} is the propagation delay on the link. Consequently, we obtain the equation (6) to calculate the path delay T_{PD}.

$$T_{PD} = \sum_{\substack{all \\ on}} \sum_{\substack{link(i,j) \\ the \\ path}} \left(\frac{1}{\mu_{ij} - \lambda_{ij}} + D_{ij}\right) \qquad \forall (i,j) \in A \qquad (6)$$

Thus far, the proposed MMCF algorithm can select the minimal cost path for the VPN customers by using equations (1) to (6). However, how the delay requirements or constraints T_D can be determined is another question to be answered. In other words, the appropriate delay values should be defined as the options that the customer can choose to buy. In the following Section 3.3.3, the average delay parameters for the whole network will be discussed.

3.3.3 Network Delay Parameters

According to the equations (3.91 and 3.92) given by Bertsekas [33], the average queuing delay T_{AQD} that a packet traverses the network is determined by:

$$T_{AQD} = \frac{1}{\gamma} \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} \qquad (7)$$

[1] We chose M/M/1 because it is more tractable allowing derive analytical expressions.

The average delay T_{AD} (including queuing and propagation delay) that a packet traverses the network is given by:

$$T_{AD} = \frac{1}{\gamma}\sum_{(i,j)}\left(\frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} + \lambda_{ij} D_{ij}\right) \qquad (8)$$

where γ is the total arrival rate in the system.

According to Leonard Klenrock's equations (2 and 3) in " On the Modeling and Analysis of Computer Networks" [34], for a given network such as Figure 3.4, the total internal network traffic λ running on the channels can be defined as:

$$\lambda = \sum_{(i,j)} \lambda_{ij} \qquad (9)$$

And the total external traffic γ is given by:

$$\gamma = \frac{\lambda}{\bar{n}} \qquad (10)$$

where $\bar{n}$ is the average number of hops that a packet must pass through the network to reach its destination.

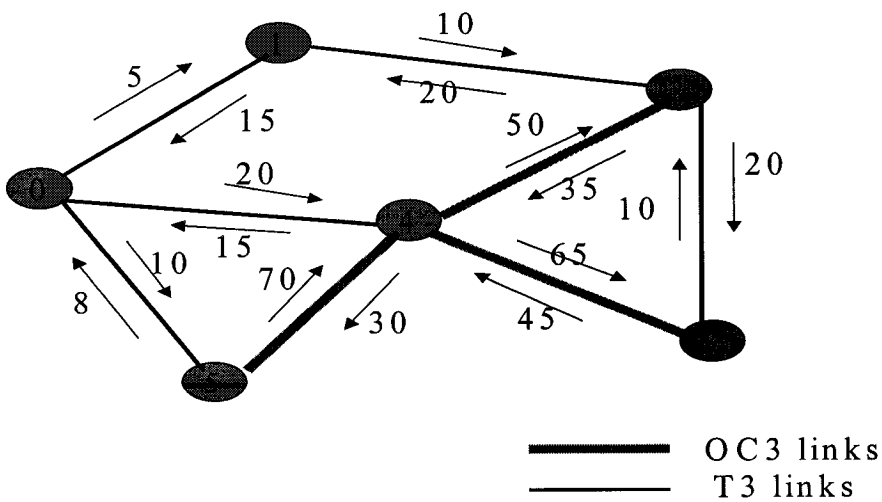


Figure 3.4 an example of a network topology

In order to determine $\bar{n}$, we first should find the average number of hops for a single origin-destination pair $\overline{n_{ij}}$, which can be given by

$$\overline{n_{ij}} = \sum_{H=1}^{N} H \times P_H \qquad (11)$$

Where H is the number of hops that the pair needs to build a path and P_H is the probability of a path with H hops.

Thus $\bar{n}$ can be given by:

$$\bar{n} = \frac{\sum_{(i,j)} \overline{n_{ij}}}{\sum_{(i,j)\in A} (i,j)} \qquad (12)$$

where A is the set of all origin-destination pairs. Using equations (9) to (12), γ can be calculated. For instance, based on these calculations in Figure 3.4, the total internal traffic is λ =428Mbps, the total number of connection pairs is 30 and $\bar{n}$ =2.48. So, $\gamma = \frac{\lambda}{\bar{n}}$ =172.58Mbps. Thus, given a non-empty network and assuming the propagation delay on each link, the average delay can be obtained by using equations (8) to (12). Thus, the ISP can estimate the network delay and offer the delay options to its customers. For example, in Figure 3.4, when the network's utilization is ρ =80%, by neglecting the propagation delay, the average delay is 1.15ms. When ρ =50%, the average delay is 0.46ms. In the following chapters, we propose several network topologies. Based on both previously mentioned algorithms, using the above equations we will attempt to determine the delay constraints and to select the QoS-routing path for each VPN connection request.

Chapter 4. Model Development

An analytical approach in terms of evaluating the proposed scheme was deemed as inappropriate, due to the complexity of the model and the realistic traffic used to load the network, which makes derivation of closed form solutions very difficult, if not impossible. An exact implementation using emulations of sources or network nodes, or even actual networking equipment was also judged as inappropriate for the purpose of this work. In addition to the considerable additional effort required to develop and place together the test bed, we felt that our analysis would have been hampered from processing, software and hardware limitations and making the number of "unaccountable" variables large, thus complicating the analysis and correct interpretation of results.

A simulation model may be of great value toward testing sensitivity and suggesting improvement in the system under investigation [35].

4.1 Network Topology Models

The network topology model covers the network architecture, consisting of the ISP network and the users' VPN topologies including the Service Level Agreement (SLA) signed between the service provider and the VPN users.

4.1.1 ISP Network Architecture

In general, ISPs build their network architecture to comply with the users' requirements and security objectives.

The network topology model is assumed for one VPN service provider who owns his/her Internet network. In addition, the following network attributes of the network model should also be considered:

- Scalability

  It is assumed that the network model consists of 11 VPN-enabled nodes and one Network Operation Center (NOC), which could function as a universal management system, able to apply the proposed multicommodity strategy. The ISP has therefore been modeled as one domain with a single NOC controlling the network globally. The specific domain could be as large as a national-wide area.

- Redundancy

  A network should have redundancy in order to be able to deal with failures of equipment. For instance, there must be at least two links between two nodes. In addition, all major nodes within the network should be reachable by one another through more than one geographically distant path. This will ensure that if a major disaster occurs within an area, it will not have a devastating impact over the entire network. However, in order to simplify the computations, the provision of extra infrastructure for the purpose of redundancy has not been incorporated in the model; only one link connection between two nodes has been assumed. For our specific

11-node ISP network models, T3 and OC3 links are chosen to interconnect the ISP's network.

4.1.2 VPN Network Topology

As mentioned earlier in Section 1.1, Network-based IP VPN is the case we are considering in our work. Therefore, for the VPN network topology, we assume the deployment of a network-based IP VPN interconnecting multiple enterprise sites, where each site has access to the nearest service provider point of presence (POP). Site-to-site traffic is carried between POPs by secure links over the service provider's managed network. However, the type of links that are used by VPN customers to access the ISP's network is beyond the range of our discussion.

For our simulation models, we assume the ISP provides Network-based IP VPN to two customers. Customer A has one headquarter A and three branches A1, A2 and A3. Customer B has one headquarter B and two branches B1 and B2.

4.1.3 Assumed Network Topology Models

Based on the above discussion, we establish the following network topologies with 11 nodes interconnected by T3 and OC3 links. The VPN customers' sites are assigned individually among these 11 nodes. Table 4.1 shows the allocations in detail. Figure 4.1 shows an example of the assumed network models.

Table 4.1 VPN Sites Allocations

Node Number	VPN customers	
	Customer A	Customer B
3	Site A3	
5		Headquarter B
6	Headquarter A	
7		Site B2
8	Site A2	
10	Site A1	Site B1

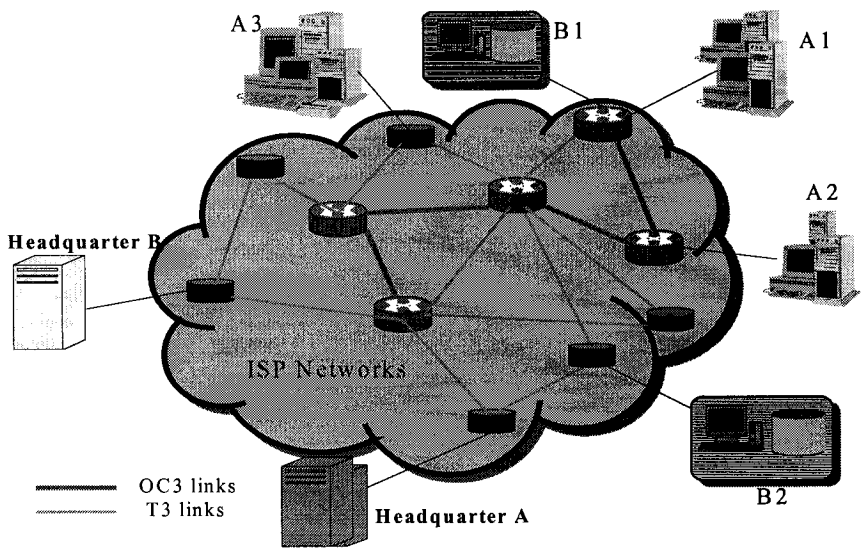


Figure 4.1: Network Topology 1

We further assume another two network topologies in Figure 4.2 and Figure 4.3 separately in order to find out whether the network architecture and link speed will effect the proposed algorithm and if it is possible, then we might ask how they effect on it. For instance, Network Topology 2 in Figure 4.2 we replace the OC3 link between node 8 and node 10 with a T3 link so as to verify if the link speed influences the simulation results. While compared with Network Topology 1 and Network Topology 2, in Figure 4.3 Network Topology 3 is a symmetric OC3-link architecture. We expect to find out

how the difference between asymmetric and symmetric architecture effects on the algorithms.

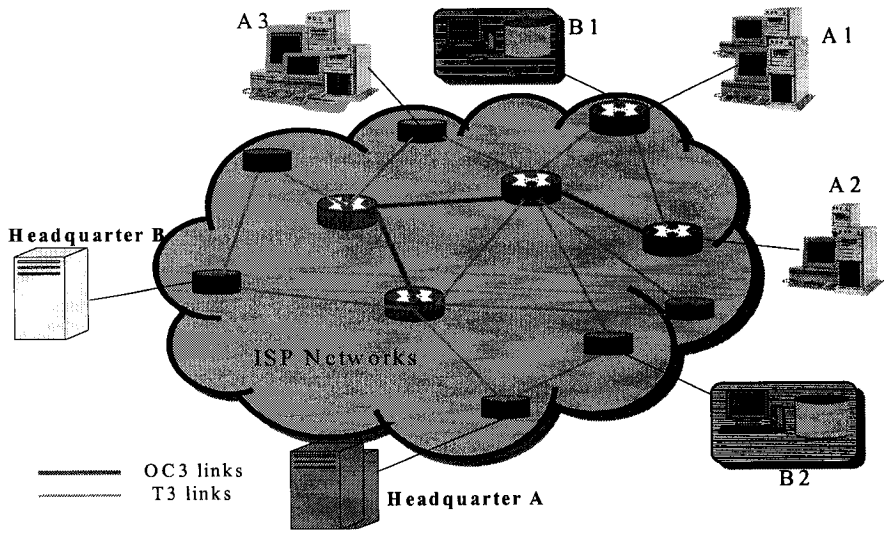


Figure 4.2: Network Topology 2

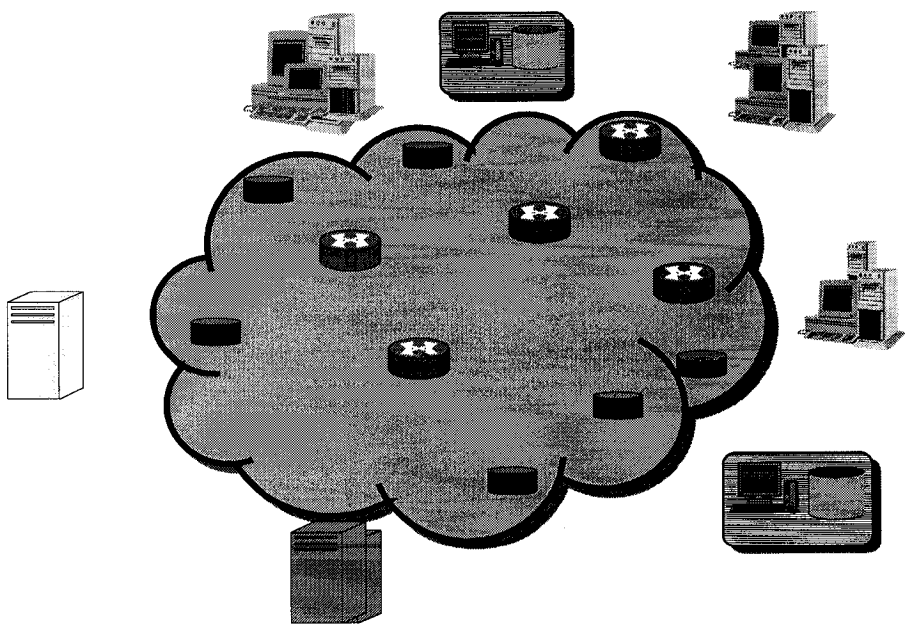
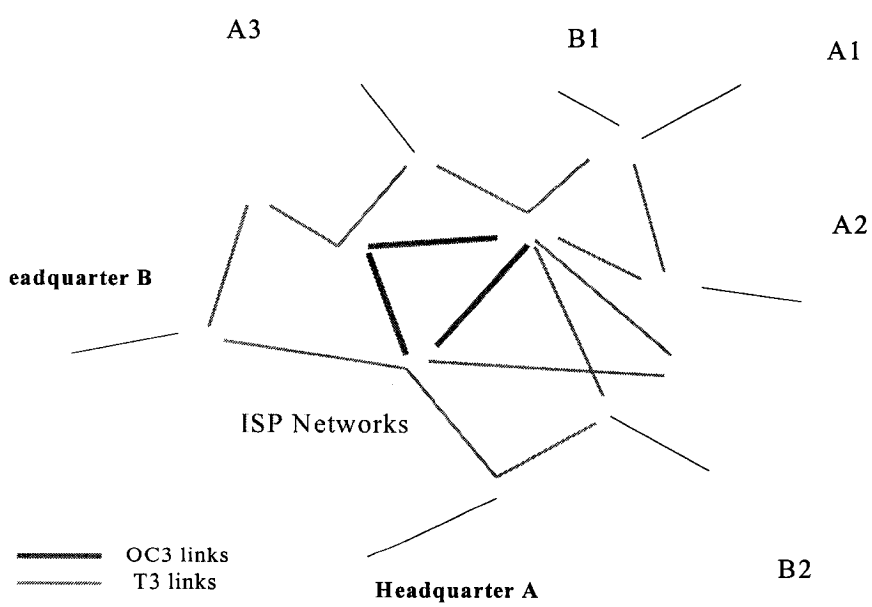


Figure 4.3: Network Topology 3


A3
B1
A1
A2
eadquarter B
ISP Networks
OC3 links
T3 links
Headquarter A
B2

4.2 Software Models

Software models have been built to accomplish the translation of the network model into a computer-recognizable format or program so as to make the data storage and simulation procedures easier. In this thesis work, C++ computer language is used to implement the simulations. The flowchart shown in Figure 4.4 gives a basic idea of the simulations.

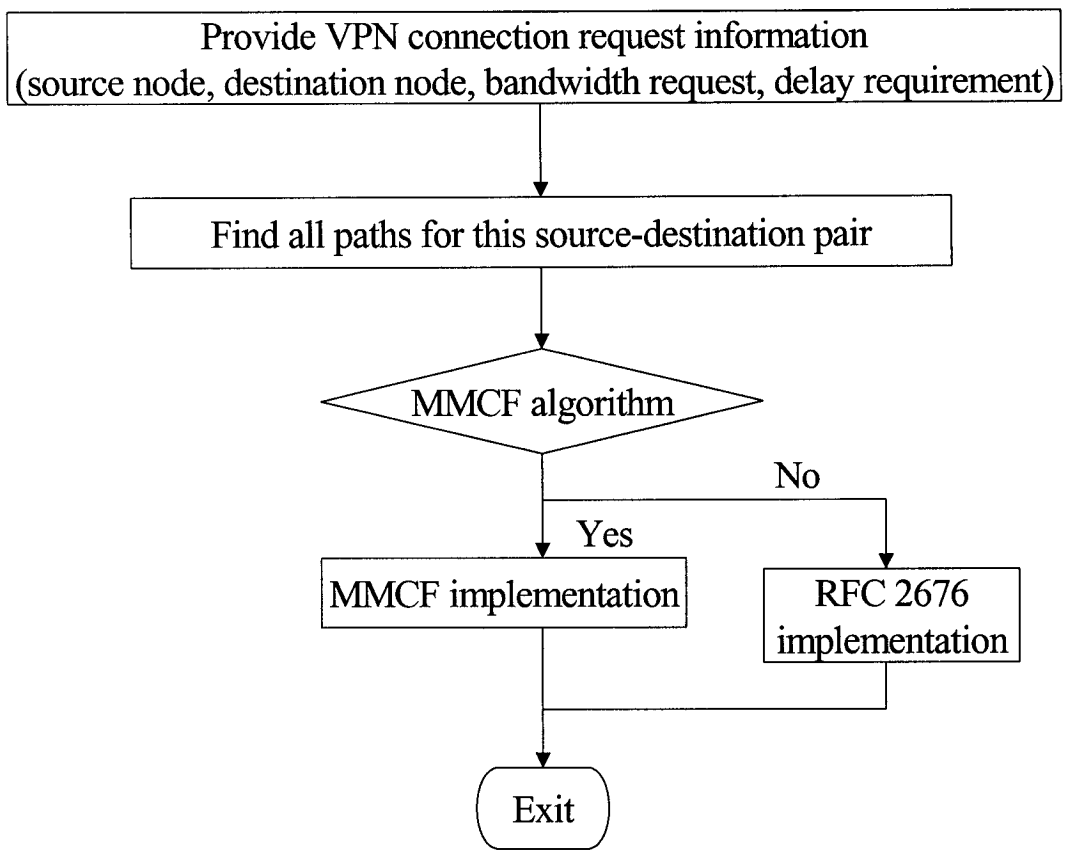


Figure 4.4 Flowchart of Simulation Implementation

As we mentioned in Chapter 3, RFC 2676 path selection is a function of both the hop count and the amount of available bandwidth. Under this implementation, a path with the minimum possible number of hops is selected among those that can provide the

requested bandwidth and satisfy its delay requirement. When several such paths are available, the preference is for the path whose available bandwidth is maximal.

On the contrary, the implementation of MMCF model selects the cheapest cost path with satisfaction of the delay requirement and the bandwidth reservation guarantee. If the cost is the same, the path comprising of less hops is selected. When both the cost and hops counts are equal, the path with more available bandwidth is chosen.

In Chapter 5, we discuss the simulation implementations of both algorithms in detail. The analysis of the simulation results is provided afterward.

Chapter 5. Simulations and Analysis

Chapter 5 provides the setup used to carry out an in-depth evaluation of the proposed scheme. We also compare the results obtained when using the RFC 2676 algorithm.

5.1 Data Collection

Currently most networks are being developed to support both best-effort traffic and guaranteed (VPN) traffic. In order to properly evaluate any QoS provisioning scheme, we have to consider that the VPN traffic and other services' traffic will share the network resources.

In the models developed in Chapter 4, there are two main data sources: 1) the VPN traffic in the service provider's network and 2) other (Internet) traffic. The VPN traffic is generated by the two headquarter offices and five branches of the VPN customers. The VPN customers sign contracts to receive the service by specifying the required access speed, which can be assumed as the bandwidth request of each simulation case. The ISP should guarantee the reservation of this bandwidth. However, if the customers try to transmit their traffic at a rate more than the speed they buy, the over-served traffic will be degraded as a best-effort service instead of a guaranteed quality of service. The delay requirement of each VPN connection request can also be assumed based on the knowledge of the average delay of the network.

For other Internet traffic, there are two approaches to describe the traffic. One option is to develop a traffic generator that can be built based on a mathematical model. The other option is to collect data from existing traffic traces taken from real networks. Having found the resource of a typical Internet traffic trace that can be representative to our case, we decided to use this traffic trace instead of generating traffic by a mathematics model.

The next question that needs to be answered is: what traffic traces should be selected? Three commonly used traffic traces were captured in October 1989 at the Bellcore Morristown Research and Engineering Facility [36]. The first trace, namely *BC-Oct89Ext*, captured 1 million external arrivals (packets headed between Bellcore and the rest of the Internet). A later trace, known as *BC-Oct89Ext4,* corresponds to a traffic trace spanning over 215 hours [37].

Another set of data collections was based on the *tcpdump* program developed by V. Paxson and S. Floydwho gathered two to four hours of such traffic at Digital's primary Internet access point in March 1995 [38].

Today, the ever-increasing Internet traffic is driving the Internet service providers to analyze whether their network resources are sufficient enough to satisfy their customers. The previously considered traffic data collection described some specific protocols or applications, which cannot reflect all current traffic through each ISP. It is therefore important to make use of more up-to-date traffic traces. One very useful resource is the WIDE (Widely Integrated Distributed Environment) project, which provides daily traffic trace data and monthly statistic traffic data as well.

5.1.1 WIDE Network

The WIDE Project, launched in 1988, created an experimental Internet environment. The WIDE Internet consists of a huge variety of computers from notebook PCs to supercomputers, incorporating all types of equipment directly and indirectly related to the information and knowledge gathering activities; from pagers to atomic clocks, from wireless LANs to satellites [39].

One of the WIDE Project's main activities is the operation of the WIDE Internet, which is an experimental network comprising an international platform for the design of a large-scale distributed environment covering the whole planet. It is composed of WIDE Network Operation Centers (NOC) in Japan and overseas. The WIDE Internet forms a shared research platform connecting approximately 140 organizations. It has also connectivity to the Internet proper, by means of two international leased lines. Therefore, the traffic trace can be a representative of Internet traffic.

5.1.3 Data obtained from the WIDE network

Statistics reported by the WIDE network provide traffic reports for each type of link in the network. In this thesis, we are particularly interested in making use of the traffic statistics of T3 and OC3 links. According to the monthly reports (see Appendix C), we can even observe the traffic statistics for the whole year. Thus we can know the peak rate of the traffic stream in the year and ensure the traffic trace data we select generally represents the link traffic status. One example of a T3 traffic trace is shown in Figure 5.1 as the red curve drawn at the very top of the graph and labeled as "total" representing the total traffic of the stream. The other curves shown in the figure are the corresponding

traffic of a variety of services. For example, the green traffic curve labeled 4:6:80 is the HTTP traffic at that address (80 is the port number assigned for HTTP service).

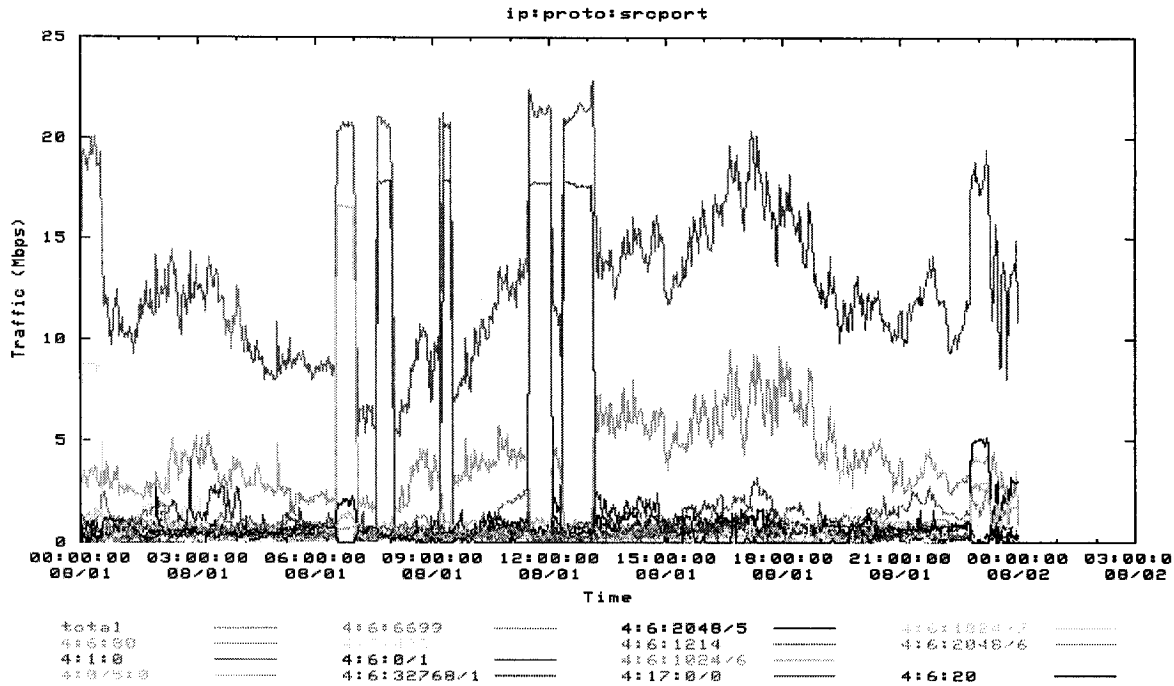


Figure 5.1 Example of the T3 Traffic Trace

Therefore, in our work, we select traffic reports (see Appendix A, B) from August 2002 as our traffic trace references. For instance, Network topology 1 in Figure 4.1 consists of 4 OC3 links and 12 T3 links and considering bi-directional traffic we need to collect 8 OC3 link traffic and 24 T3 link traffic traces individually. Therefore, 8 OC3 traffic traces from August 1st to August 8th (see the trace graphs in Appendix A Figure A1 to Figure A8) are chosen to represent OC3 Internet traffic for the corresponding link. 24 T3 traces from August 1st to August 24th (see the trace graphs in Appendix A Figure B1 to Figure B24) represent the T3 traffic for each T3 link in the Network topology 1.

Accordingly, we allocate the traffic trace collection to each OC3 and T3 link of Network Topology 1, 2 and 3 separately. Table 5.1 shows the detailed information of the

distribution where the differences of the designation between the topologies are highlighted.

Table 5.1 Traffic Trace Allocations for each Topology

Date of Traces	TOPOLOGY1		TOPOLOGY2		TOPOLOGY3	
	OC3 links	T3 links	OC3 links	T3 links	OC3 links	T3 links
Aug-01	0<->1 / 30km: 0.15ms	0<->3 / 10km: 0.05ms	0<->1 / 30km: 0.15ms	0<->3 / 10km: 0.05ms	0<->1 / 30km: 0.15ms	0<->3 / 10km: 0.05ms
Aug-02	0<->2 / 20km: 0.1ms	0<->4 / 15km: 0.075ms	0<->2 / 20km: 0.1ms	0<->4 / 15km: 0.075ms	0<->2 / 20km: 0.1ms	0<->4 / 15km: 0.075ms
Aug-03	1<->0	1<->2 / 18km: 0.09ms	1<->0	1<->2 / 18km: 0.09ms	1<->0	1<->8 / 36km: 0.18ms
Aug-04	1<->8 / 36km: 0.18ms	1<->3 / 20km: 0.1ms	1<->8 / 36km: 0.18ms	1<->3 / 20km: 0.1ms	1<->2 / 18km: 0.09ms	1<->3 / 20km: 0.1ms
Aug-05	2<->0	1<->7 / 16km: 0.08ms	2<->0	1<->7 / 16km: 0.08ms	2<->0	1<->7 / 16km: 0.08ms
Aug-06	8<->1	1<->9 / 20km: 0.1ms	8<->1	1<->9 / 20km: 0.1ms	2<->1	1<->9 / 20km: 0.1ms
Aug-07	8<->10 / 40km: 0.2ms	1<->10 / 12km: 0.06ms		1<->10 / 12km: 0.06ms		1<->10 / 12km: 0.06ms
Aug-08	10<->8	2<->1		2<->1		8<->2
Aug-09		2<->5 / 14km: 0.07ms		2<->5 / 14km: 0.07ms		2<->5 / 14km: 0.07ms
Aug-01		2<->6 / 10km: 0.05ms		2<->6 / 10km: 0.05ms		2<->6 / 10km: 0.05ms
Aug-11		2<->9 / 22km: 0.11ms		2<->9 / 22km: 0.11ms		2<->9 / 22km: 0.11ms
Aug-12		3<->0		3<->0		3<->0
Aug-13		3<->1		3<->1		3<->1
Aug-14		4<->0		4<->0		4<->0
Aug-15		4<->5 / 16km: 0.08ms		4<->5 / 16km: 0.08ms		4<->5 / 16km: 0.08ms
Aug-16		5<->2		5<->2		5<->2
Aug-17		5<->4		5<->4		5<->4
Aug-18		6<->2		6<->2		6<->2
Aug-19		6<->7 / 20km: 0.1ms		6<->7 / 20km: 0.1ms		6<->7 / 20km: 0.1ms
Aug-02		7<->1		7<->1		7<->1
Aug-21		7<->6		7<->6		7<->6
Aug-22		9<->1		9<->1		9<->1
Aug-23		9<->2		9<->2		9<->2
Aug-24		10<->1		10<->1		10<->1
Aug-25				8<->10 / 40km: 0.2ms		8<->10 / 40km: 0.2ms
Aug-26				10<->8		10<->8

Given a few of the VPN connection requests, we decided to run eight different experiments or simulation cases for each assumed network topology in order to evaluate our system. For one topology, the network traffic data of the eight experiments came from the traffic trace allocation on each link at eight different time instants. For example, for the OC3 link between node 0 and node 1, we have selected eight experimental traffic data from the trace of August 1st at time instant 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00 and 21:00. Table 5.2 shows the detailed information that the eight traffic data are

allocated to each OC3 link for the purpose of running eight simulation cases. Furthermore, Table 5.3 and 5.4 show the allocations for each T3 link.

Table 5.2 Traffic data of 8 simulation cases collected for each OC3 link

simulation	selected	Internet traffic on OC3 links (Mbps)							
case#	moments	Aug-01	Aug-02	Aug-03	Aug-04	Aug-05	Aug-06	Aug-07	Aug-08
1	0:00	12.9	95.1	3.2	4	4.8	3.8	4.1	5.7
2	3:00	6.3	48.3	8.2	16.7	2.6	5.5	4.8	3.4
3	6:00	5.5	2	4.3	1.1	2.4	1.9	3	2.8
4	9:00	95.1	3.2	6.2	1.7	20.7	3.6	7.4	4.7
5	12:00	96	10.3	5.5	4.3	12.8	43.8	8.3	7.2
6	15:00	13.8	14.9	24.6	5.7	11.2	21.9	12.4	9.5
7	18:00	12.1	8.6	10.8	7.5	10.9	11.2	8.5	14.5
8	21:00	6.6	4.6	9.8	3.2	11.7	9.3	5.7	28.3

Table 5.3 Traffic data of 8 simulation cases collected for each T3 link

Simulation	Selected	Internet traffic on T3 links (Mbps)												
case#	moments	Aug-01	Aug-02	Aug-03	Aug-04	Aug-05	Aug-06	Aug-07	Aug-08	Aug-09	Aug-10	Aug-11	Aug-12	Aug-13
1	0:00	16	12	2	9.5	15	14.2	15.8	12	10.4	10.2	7.2	8	8.7
2	3:00	13	10.8	9.8	13.6	14.2	19	12	9.8	9.2	11	9.3	7.2	4.8
3	6:00	8.8	7.2	10.6	9	9	9	8.7	10	12.8	22	7	4.4	6.8
4	9:00	9.8	12	9	8.3	14	10.5	9.2	10.4	11	5	4.2	11.8	6.4
5	12:00	22	17	8.5	10.8	19.8	23	12.8	12	9.4	8.6	7	9.5	9.3
6	15:00	14.5	16.2	16	11.2	23	17.5	20	15.8	13.2	9	12	10	10.5
7	18:00	18	16	9.5	15.1	21.6	17	19.7	20.1	17	8	9.9	10.8	12
8	21:00	11	13	14.3	12.2	17.3	12	13.8	14	11	9.2	11.2	8	11.6

Table 5.4 Traffic data of 8 simulation cases collected for each T3 link (cont'd)

Simulation	Selected	Internet traffic on T3 links (Mbps) cont'd												
case#	moments	Aug-14	Aug-15	Aug-16	Aug-17	Aug-18	Aug-19	Aug-20	Aug-21	Aug-22	Aug-23	Aug-24	Aug-25	Aug-26
1	0:00	9.3	12.2	7	4.5	5.3	6	7.4	21.8	7.4	9.3	13.2	16.2	10.1
2	3:00	8.4	8.8	7	4.1	4.7	9.3	9.2	7.7	7.6	12.2	12.4	10.4	7.9
3	6:00	9.4	6.7	4.7	3.7	5	4.8	6.5	6.3	9.4	7.9	7.2	7.3	7
4	9:00	9.9	20.2	6.4	6	4.8	11.4	14.4	23	7.8	10.9	10.7	7.3	10.8
5	12:00	10.9	10.2	9.5	5	5	11.2	7.7	6.6	12.9	12	8.8	14	13.2
6	15:00	14.8	9.5	11.9	5.8	7.7	9.1	13	12.3	16.1	13.3	11.8	13.5	14.7
7	18:00	13.3	17.4	8.4	6.9	7.1	12.5	16	11	13.6	14	13.2	11.5	15.7
8	21:00	12.2	11.7	4.6	6	5.3	7.5	14.7	11.8	8.9	8.8	10.9	7.4	12.9

According to the above traffic allocations in Table 5.2, 5.3 and 5.4, for the Network Topology 1 in Figure 5.2, its active network condition at 0:00 can be described in Figure 5.3.

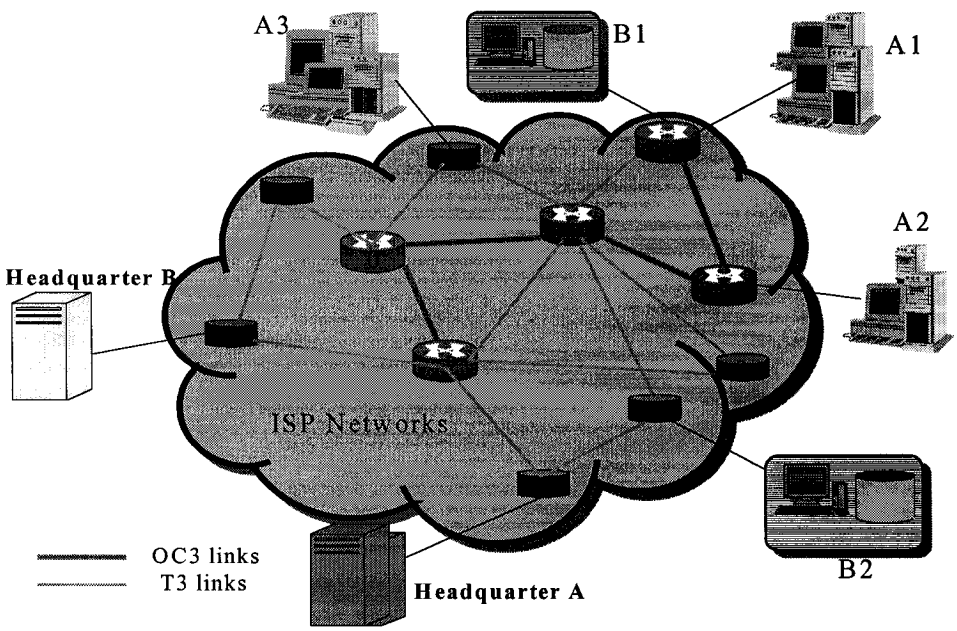


Figure 5.2 Network Topology 1

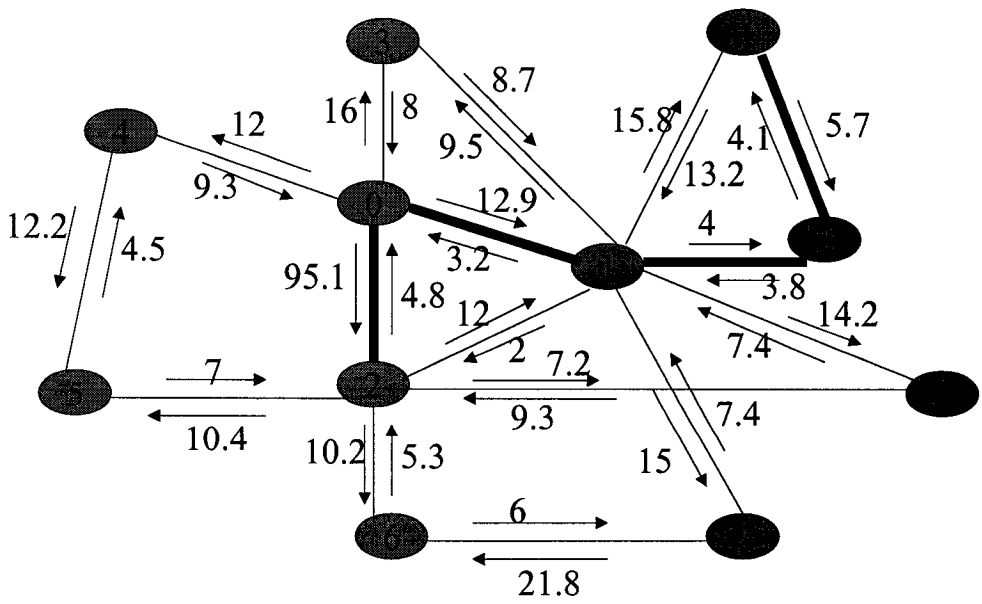


Figure 5.3 Traffic Condition of Network Topology 1 at 0:00

All of the active network conditions of the eight simulation cases for each network topology in Figure 4.1, 4.2 and 4.3 are shown in Appendix D. Thus, once we apply the VPN connection requests we can run the simulations.

5.2 Simulation Method

As previously mentioned in Section 3.3.3, the network average delay T_{AQD} (neglecting the processing and propagation delay) can be determined once we know the network conditions such as shown in Figure 5.3. By using the equations in Section 3.3.3, for Network Topology 1, when ρ equals 80%, T_{AQD} is 1.7ms and when ρ equals 50%, T_{AQD} is 0.68ms. For Network Topology 2 and 3, when ρ equals 80%, T_{AQD} is 1.88ms and when ρ equals 50%, T_{AQD} is 0.75ms. Based on these estimations of delay, we can assume that the ISP can offer three delay options (Delay-level 1, 2 and 3) to the VPN customers.

In Table 4.1, the VPN sites are allocated at nodes numbered 3, 5, 6, 7, 8 and 10. We further assume that there are five VPN connection requests in each simulation case and the VPN packet size is averagely 1000bytes. Table 5.5 shows the five requests in detail, where we can assume the value of the Delay Level (for example: Delay-level 1 = 0.70ms, Delay-level 2 = 1.0ms and Delay-level 3 = 1.2ms).

Table 5.5 Assumption of VPN connection requests

Source node	Destination node	Bandwidth reservation (Mbps)	Delay level
6	10	4	1
6	8	7	2
3	6	2	3
5	10	8	2
7	5	3	1

Consequently, Having applied the above VPN requests in Table 5.5 and having used traffic data information in Table 5.2, 5.3 and 5.4, we ran the eight simulation cases for the network topology of Figure 4.1, 4.2 and 4.3 individually. Table 5.6 shows the partial information we gathered from the simulation case 1 (at 0:00) of Network Topology 1. The first two columns indicate the source and destination nodes. The third column shows the link capacity. If there is a direct link between the source and destination node, 45Mbps is assigned for a T3 link and 155 Mbps for an OC3 link. A '0' indicates that there is no link between the two concerned nodes. The following three columns indicate the other traffic transiting over the link, the link's propagation delay and its cost. If there is no link between the two nodes, 9999 is assigned as the propagation delay. The service provider's cost information of each leased link is assigned as $267 per Mbps for OC3 link traffic, $667 per Mbps for T3 link traffic and $0 if there is no direct link connection between these two nodes. The "Initial condition" in the "Available bandwidth" block is the bandwidth that the link can offer to the VPN traffic. The bandwidth amount has been determined by subtracting the other traffic in column 4 from the link capacity in column 3. The "Following OSPF (RFC 2676)" column (column 8) is the remaining available bandwidth after VPN service reserved its required bandwidth by using the RFC 2676 algorithm. The same process applies to column 9, but now MMCF is applied to allocate the VPN traffic.

Table 5.6 Partial link information of simulation case 1 run on network Topology 1

Link nodes		Link Information				Available bandwidth (Mbps)		
Source node	Destination node	Capacity (Mbps)	Traffic (Mbps)	Propagation-delay (ms)	Cost ($)	Initial condition	Following OSPF (RFC 2676)	Following MMCF
0	0	0	0	9999	0	0	0	0
0	1	155	12.9	0.15	267	142.1	142.1	127.1
0	2	155	95.1	0.1	267	59.9	57.9	54.9
0	3	45	16	0.05	667	29	29	29
0	4	45	12	0.075	667	33	33	33
0	5	0	0	9999	0	0	0	0
0	6	0	0	9999	0	0	0	0
0	7	0	0	9999	0	0	0	0
0	8	0	0	9999	0	0	0	0
0	9	0	0	9999	0	0	0	0
0	10	0	0	9999	0	0	0	0
1	0	155	3.2	0.15	267	151.8	151.8	148.8
1	1	0	0	9999	0	0	0	0
1	2	45	2	0.09	667	43	40	43
1	3	45	9.5	0.1	667	35.5	35.5	35.5
1	4	0	0	9999	0	0	0	0
1	5	0	0	9999	0	0	0	0
1	6	0	0	9999	0	0	0	0
1	7	45	15	0.08	667	30	30	30
1	8	155	4	0.18	267	151	144	132
1	9	45	14.2	0.1	667	30.8	30.8	30.8
1	10	45	15.8	0.06	667	29.2	17.2	29.2
2	0	155	4.8	0.1	267	150.2	150.2	135.2
2	1	45	12	0.09	667	33	18	33
2	2	0	0	9999	0	0	0	0
2	3	0	0	9999	0	0	0	0
2	4	0	0	9999	0	0	0	0
2	5	45	10.4	0.07	667	34.6	31.6	31.6
2	6	45	10.2	0.05	667	34.8	32.8	32.8
2	7	0	0	9999	0	0	0	0
2	8	0	0	9999	0	0	0	0
2	9	45	7.2	0.11	667	37.8	37.8	37.8
2	10	0	0	9999	0	0	0	0

In Section 5.3 we show all the simulation results for each network topology.

5.3 Simulation results

Our simulation results show that the delay level requirement does influence the routing cost especially with the smallest delay level value. Based on the VPN connection

requests shown in Table 5.5, we provide the simulation results of three different Delay Level 1 assumptions (0.70ms, 0.63ms and 0.60ms) for each network topology.

5.3.1 Simulation Results of Network Topology 1

The Network Topology 1 graph is shown in Figure 4.1. The traffic traces collected for the eight simulation cases are shown in Appendix D.1 The VPN requests are shown in Table 5.5. We ran the simulations according to the following three scenarios:

(1) When Delay-level 1 = 0.7ms, Delay-level 2 = 1.0ms and Delay level 3 = 1.2ms

Figure 5.4 shows the costs resulting from both MMCF and RFC2676 algorithms. In this and all the following figures of cost, the X axis represents the eight simulation cases and the Y axis indicates the cost in dollars. Table 5.7 displays the detailed information of the simulation results for each case and it shows that the MMCF algorithm can save approximately 9% of the cost compared with OSPF (RFC2676) algorithm.

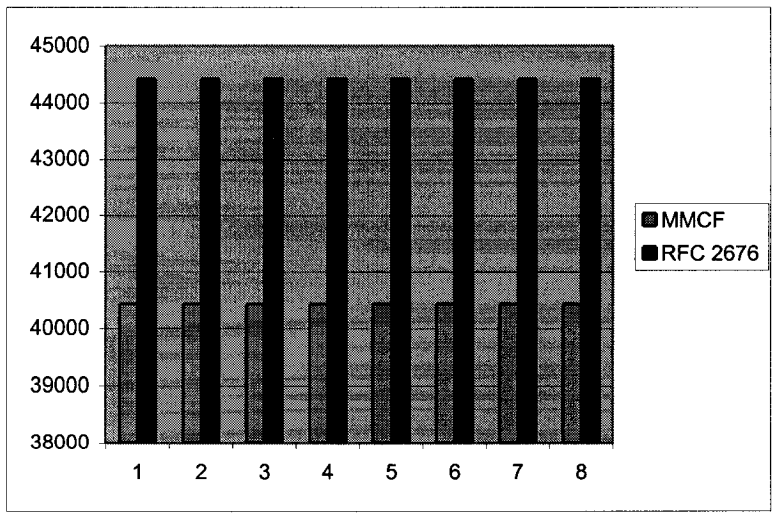


Figure 5.4 Simulation results for Network Topology 1 scenario (1)

Table 5.7 Simulation data corresponding to Topology 1 scenario (1)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.70ms)	6 7 1 8 10	0.68	0.61	0.91	6 7 1 10	0.49	0.63	0.91	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.59	0.88	6 2 1 8	0.52	0.6	0.88	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.4	0.57	0.83	3 0 2 6	0.4	0.57	0.83	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.81	0.55	0.84	5 2 1 10	0.65	0.6	0.86	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.62	0.53	0.78	7 1 2 5	0.39	0.53	0.77	
Case1 0:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.51	0.58	0.88	6 2 1 10	0.42	0.59	0.88	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.55	0.85	6 7 1 8	0.57	0.57	0.86	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.34	0.53	0.8	3 0 2 6	0.34	0.53	0.8	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.81	0.52	0.81	5 2 1 10	0.57	0.55	0.82	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.56	0.49	0.74	7 6 2 5	0.38	0.49	0.74	
Case2 3:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 7 1 8 10	0.66	0.66	0.94	6 7 1 10	0.41	0.67	0.93	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.61	0.88	6 2 1 8	0.5	0.63	0.89	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.58	0.81	3 0 2 6	0.43	0.58	0.81	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.79	0.55	0.82	5 2 1 10	0.51	0.6	0.82	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.55	0.52	0.73	7 6 2 5	0.4	0.52	0.73	
Case3 6:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 2 1 8 10	0.65	0.61	0.94	6 2 1 10	0.4	0.62	0.94	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.6	0.93	6 2 1 8	0.5	0.6	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.56	0.86	3 0 2 6	0.31	0.56	0.86	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.91	0.57	0.89	5 2 1 10	0.54	0.58	0.87	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.6	0.53	0.81	7 1 2 5	0.5	0.53	0.81	
Case4 9:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 2 1 8 10	0.67	0.6	0.95	6 2 1 10	0.45	0.61	0.95	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.59	0.93	6 7 1 8	0.57	0.59	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.32	0.56	0.88	3 0 2 6	0.32	0.56	0.88	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.94	0.57	0.9	5 2 1 10	0.63	0.58	0.9	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.53	0.53	0.83	7 6 2 5	0.37	0.53	0.83	
Case5 12:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.64	0.66	0.97	6 7 1 10	0.62	0.67	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.59	0.62	0.93	6 2 1 8	0.59	0.64	0.95	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.6	0.89	3 0 2 6	0.33	0.6	0.89	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.87	0.6	0.9	5 2 1 10	0.85	0.65	0.94	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.62	0.57	0.84	7 6 2 5	0.47	0.57	0.84	
Case6 15:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.63	0.71	1	6 2 0 10	0.66	0.73	1.02	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.58	0.67	0.97	6 7 1 8	0.69	0.7	0.99	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.65	0.92	3 0 2 6	0.33	0.65	0.92	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.83	0.63	0.93	5 2 1 10	0.9	0.71	0.98	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.68	0.61	0.87	7 6 2 5	0.49	0.61	0.87	
Case7 18:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.70ms)	6 2 1 8 10	0.69	0.62	0.94	6 2 1 10	0.48	0.63	0.94	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.58	0.9	6 7 1 8	0.61	0.61	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.56	0.84	3 0 2 6	0.31	0.56	0.84	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.79	0.55	0.85	5 2 1 10	0.63	0.59	0.87	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.6	0.52	0.79	7 6 2 5	0.43	0.53	0.79	
Case8 21:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay).

Analysis:

With the assumption of the Delay Level in scenario (1), the cost savings of each simulation case is the same (9%) although the traffic data applied to it are different. However, the path delay for every VPN connection request is different in each simulation case. Table 5.7 also shows that the MMCF algorithm saves routing costs compared with the OSPF (RFC 2676) algorithm, for the most part, with the consumption of more path delays for each connection. Figure 5.5 shows the difference of the path delay of each request among the all eight simulation cases. In this and all the following figures of path delays, the X axis represents the total of 40 requests for the eight simulation cases and the Y axis indicates the path delay for each request in milliseconds.

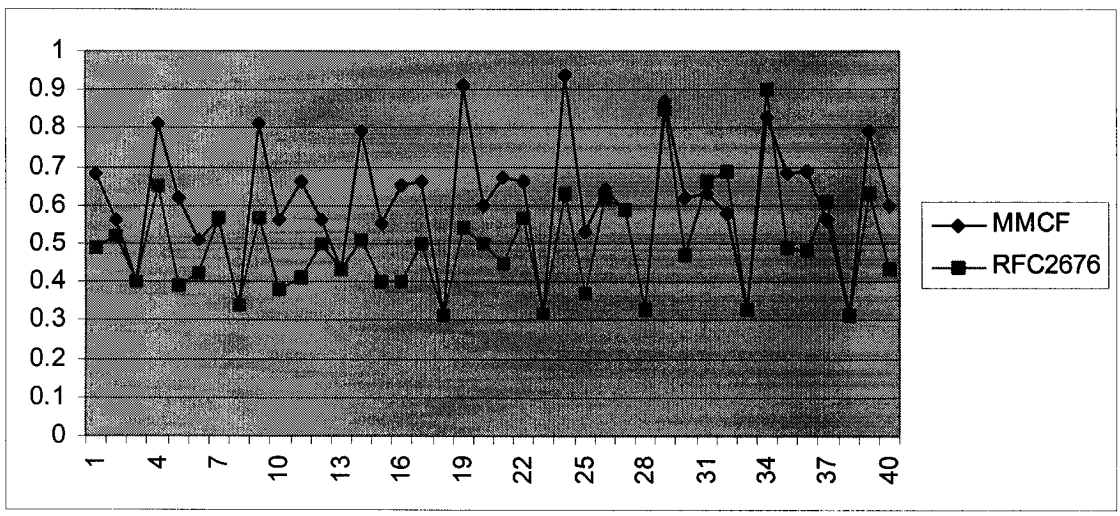


Figure 5.5 Path delays for each VPN request in the 8 simulation cases of Topology 1 scenario (1)

On the other hand, compared with the OSPF (RFC2676) algorithm, the MMCF algorithm does improve the queuing delay of the whole network. However, the MMCF algorithm spends more path delay for a specific connection than the RFC2676 (OSPF) algorithm does. Thus, Figure 5.6 indicates the improvement of the network average

queuing delay for all the simulation cases. The units in the figure are the same as in Figure 5.5.

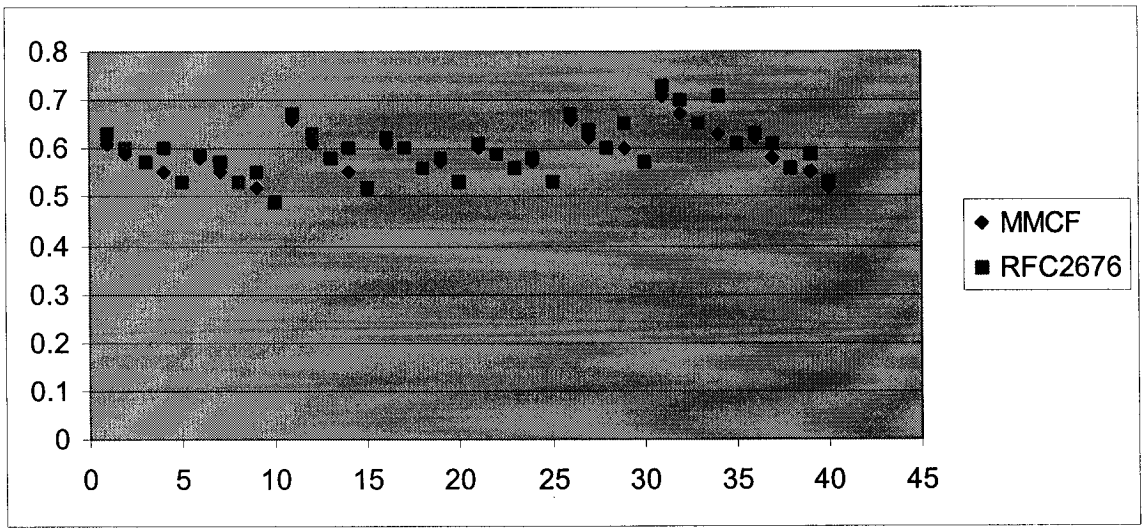


Figure 5.6 Network average queuing delay in the 8 simulations cased of Topology 1 scenario (1)

(2) Delay-level 1 = 0.63ms, Delay-level 2 = 1.0ms and Delay level 3 = 1.2ms

In the above scenario (1), the MMCF algorithm saves the same percentage cost compared with the OSPF (RFC2676) algorithm in each simulation case. Therefore, we assumed a smaller delay requirement value of Delay-level 1 such as 0.63ms in this scenario in order to find some different results. Figure 5.7 shows the costs resulting from both the MMCF and RFC2676 algorithms. Table 5.8 displays the detailed information of the simulation results for each case.

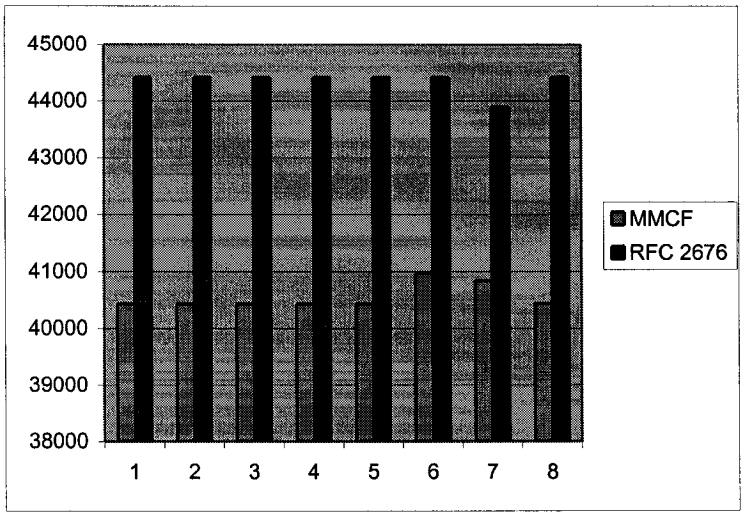


Figure 5.7 Simulation results for Network Topology 1 scenario (2)

Table 5.8 Simulation data corresponding to Topology 1 scenario (2)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.56	0.62	0.91	6 7 1 10	0.49	0.63	0.91	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.59	0.88	6 2 1 8	0.52	0.6	0.88	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.4	0.57	0.83	3 0 2 6	0.4	0.57	0.83	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.81	0.55	0.84	5 2 1 10	0.65	0.6	0.86	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.62	0.53	0.78	7 1 2 5	0.39	0.53	0.77	
Case1 0:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.51	0.58	0.88	6 2 1 10	0.42	0.59	0.88	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.55	0.85	6 7 1 8	0.57	0.57	0.86	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.34	0.53	0.8	3 0 2 6	0.34	0.53	0.8	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.81	0.52	0.81	5 2 1 10	0.57	0.55	0.82	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.56	0.49	0.74	7 6 2 5	0.38	0.49	0.74	
Case2 3:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.48	0.66	0.93	6 7 1 10	0.41	0.67	0.93	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.61	0.88	6 2 1 8	0.5	0.63	0.89	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.58	0.81	3 0 2 6	0.43	0.58	0.81	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.79	0.55	0.82	5 2 1 10	0.51	0.6	0.82	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.55	0.52	0.73	7 6 2 5	0.4	0.52	0.73	
Case3 6:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.58	0.62	0.95	6 2 1 10	0.4	0.62	0.94	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.6	0.93	6 2 1 8	0.5	0.6	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.56	0.86	3 0 2 6	0.31	0.56	0.86	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.91	0.57	0.89	5 2 1 10	0.54	0.58	0.87	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.6	0.53	0.81	7 1 2 5	0.5	0.53	0.81	
Case4 9:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.61	0.61	0.96	6 2 1 10	0.45	0.61	0.95	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.59	0.93	6 7 1 8	0.57	0.59	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.32	0.56	0.88	3 0 2 6	0.32	0.56	0.88	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.94	0.57	0.9	5 2 1 10	0.63	0.58	0.9	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.53	0.53	0.83	7 6 2 5	0.37	0.53	0.83	
Case5 12:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.63ms)	6 7 1 10	0.62	0.67	0.98	6 7 1 10	0.62	0.67	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.59	0.62	0.93	6 2 1 8	0.59	0.64	0.95	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.6	0.89	3 0 2 6	0.33	0.6	0.89	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.87	0.6	0.9	5 2 1 10	0.85	0.65	0.94	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.62	0.57	0.84	7 6 2 5	0.47	0.57	0.84	
Case6 15:00	**total_cost($): 40966**				**total_cost($): 44424**				**7.78**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.63	0.71	1	6 2 0 1 1	0.63	0.71	1.01	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.58	0.67	0.97	6 7 1 8	0.69	0.7	0.99	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.65	0.92	3 0 2 6	0.33	0.65	0.92	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.83	0.63	0.93	5 2 1 10	0.9	0.71	0.98	
(7, 5, 3M, 0.63ms)	7 1 2 5	0.58	0.62	0.87	7 6 2 5	0.49	0.61	0.87	
Case7 18:00	**total_cost($): 40833**				**total_cost($): 43892**				**6.97**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.53	0.62	0.93	6 2 1 10	0.48	0.63	0.94	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.58	0.9	6 7 1 8	0.61	0.61	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.56	0.84	3 0 2 6	0.31	0.56	0.84	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.79	0.55	0.85	5 2 1 10	0.63	0.59	0.87	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.6	0.52	0.79	7 6 2 5	0.43	0.53	0.79	
Case8 21:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay).

Figure 5.8 also shows the difference for the path delays of each request among the all eight simulation cases under this scenario. Figure 5.9 indicates the improvement of the network average queuing delay as well.

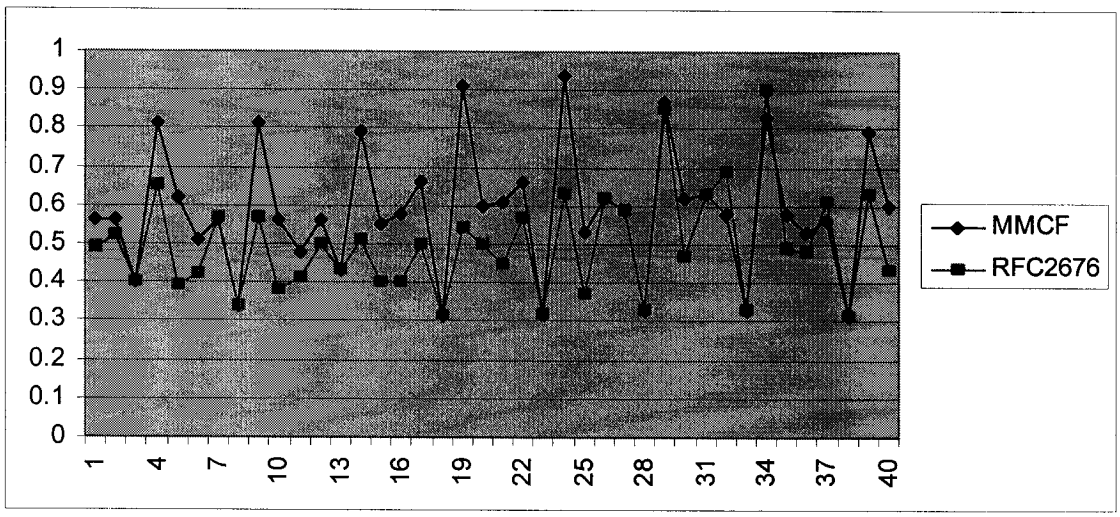


Figure 5.8 Path delays for each VPN request in the 8 simulation cases of Topology 1 scenario (2)

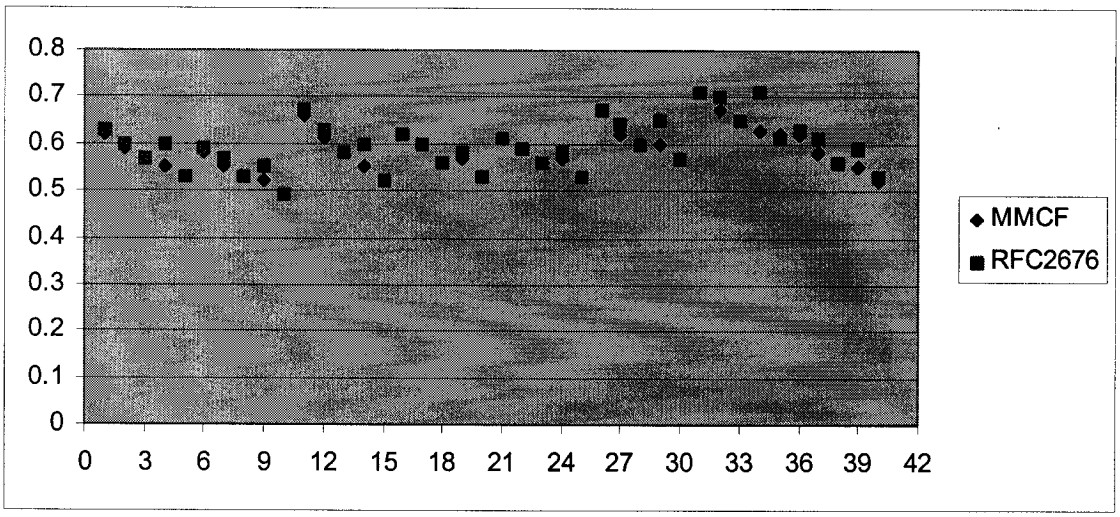


Figure 5.9 Network average queuing delay in the 8 simulations cases of Topology 1 scenario (2)

Analysis:

Instead of the value 0.7ms of Delay-level 1, we use 0.63ms in this scenario. As it is shown in Table 5.8, in case 6 and case 7, the path selections change because of the satisfaction of the decreased delay requirements. Consequently, the costs (shown in Figure 5.7) are changed as well. It shows that the MMCF algorithm can save the cost between 7% and 9% compared with OSPF (RFC2676) algorithm. Based on the above simulation results for both scenarios (1) and (2), it can be concluded that the effectiveness of cost savings is influenced by the delay requirement. Compared with the RFC2676 (OSPF) algorithm, the MMCF algorithm also improves the average network delay (neglecting the processing and propagation delay) although mostly it spends more path delay on the specific connection request.

(3) Delay-level 1 = 0.6ms, Delay-level 2 = 1.0ms and Delay level 3 = 1.2ms

In this scenario, we make Delay level 1 even smaller to evaluate the network again. Figure 5.10 shows the costs resulting from both MMCF and RFC2676 algorithms.

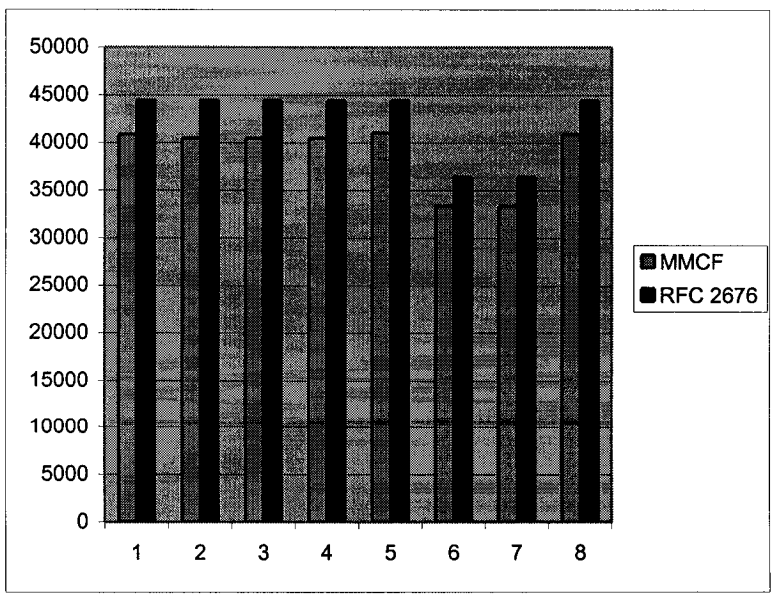


Figure 5.10 Simulation results for Network Topology 1 scenario (3)

Table 5.9 displays the detailed information of the simulation results for each case.

Table 5.9 Simulation data corresponding to Topology 1 scenario (3)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.56	0.62	0.91	6 7 1 10	0.49	0.63	0.91	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.59	0.88	6 2 1 8	0.52	0.6	0.88	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.4	0.57	0.83	3 0 2 6	0.4	0.57	0.83	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.81	0.55	0.84	5 2 1 10	0.65	0.6	0.86	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.39	0.53	0.77	7 1 2 5	0.39	0.53	0.77	
Case1 0:00	**total_cost($): 40833**				**total_cost($): 44424**				**8.08**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.51	0.58	0.88	6 2 1 10	0.42	0.59	0.88	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.55	0.85	6 7 1 8	0.57	0.57	0.86	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.34	0.53	0.8	3 0 2 6	0.34	0.53	0.8	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.81	0.52	0.81	5 2 1 10	0.57	0.55	0.82	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.56	0.49	0.74	7 6 2 5	0.38	0.49	0.74	
Case2 3:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.48	0.66	0.93	6 7 1 10	0.41	0.67	0.93	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.61	0.88	6 2 1 8	0.5	0.63	0.89	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.58	0.81	3 0 2 6	0.43	0.58	0.81	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.79	0.55	0.82	5 2 1 10	0.51	0.6	0.82	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.55	0.52	0.73	7 6 2 5	0.4	0.52	0.73	
Case3 6:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.58	0.62	0.95	6 2 1 10	0.4	0.62	0.94	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.6	0.93	6 2 1 8	0.5	0.6	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.56	0.86	3 0 2 6	0.31	0.56	0.86	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.91	0.57	0.89	5 2 1 10	0.54	0.58	0.87	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.6	0.53	0.81	7 1 2 5	0.5	0.53	0.81	
Case4 9:00	**total_cost($): 40434**				**total_cost($): 44424**				**8.98**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.45	0.61	0.95	6 2 1 10	0.45	0.61	0.95	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.59	0.93	6 7 1 8	0.57	0.59	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.32	0.56	0.88	3 0 2 6	0.32	0.56	0.88	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.94	0.57	0.9	5 2 1 10	0.63	0.58	0.9	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.53	0.53	0.83	7 6 2 5	0.37	0.53	0.83	
Case5 12:00	**total_cost($): 40966**				**total_cost($): 44424**				**7.78**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.61*	0.67	0.98	6 2 1 10	0.61 *	0.67	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.59	0.62	0.93	6 7 1 8	0.61	0.64	0.94	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.6	0.89	3 0 2 6	0.33	0.6	0.89	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.87	0.6	0.9	5 2 1 10	0.85	0.65	0.94	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.57	0.57	0.84	7 6 2 5	0.47	0.57	0.84	
Case6 15:00	**total_cost($): 33361**				**total_cost($): 36420**				**8.4**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.63 *	0.71	1	6 2 0 1 10	0.63 *	0.71	1	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.58	0.67	0.97	6 7 1 8	0.69	0.7	0.99	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.65	0.92	3 0 2 6	0.33	0.65	0.92	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.83	0.63	0.93	5 2 1 10	0.9	0.71	0.98	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.58	0.62	0.87	7 6 2 5	0.49	0.61	0.87	
Case7 18:00	**total_cost($): 33361**				**total_cost($): 36420**				**8.4**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.53	0.62	0.93	6 2 1 10	0.48	0.63	0.94	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.58	0.9	6 7 1 8	0.61	0.61	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.56	0.84	3 0 2 6	0.31	0.56	0.84	
(5, 10, 8M, 1.00ms)	5 2 0 1 8 10	0.79	0.55	0.85	5 2 1 10	0.63	0.59	0.87	
(7, 5, 3M, 0.60ms)	7 6 2 5	0.43	0.53	0.79	7 6 2 5	0.43	0.53	0.79	
Case8 21:00	**total_cost($): 40833**				**total_cost($): 44424**				**8.08**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay); *: PD exceeds delay requirement

Table 5.9 indicates that for both algorithms there is no path to satisfy the delay requirement of the VPN request (6, 10, 4M, 0.6ms). In this situation, we assume that if the delay requirement cannot be met, this request would be rejected. Thus in the path delay (Figure 5.11) and average network delay (neglecting processing and propagation delay) of Figure 5.12, the delay of this request is assigned to '0'.

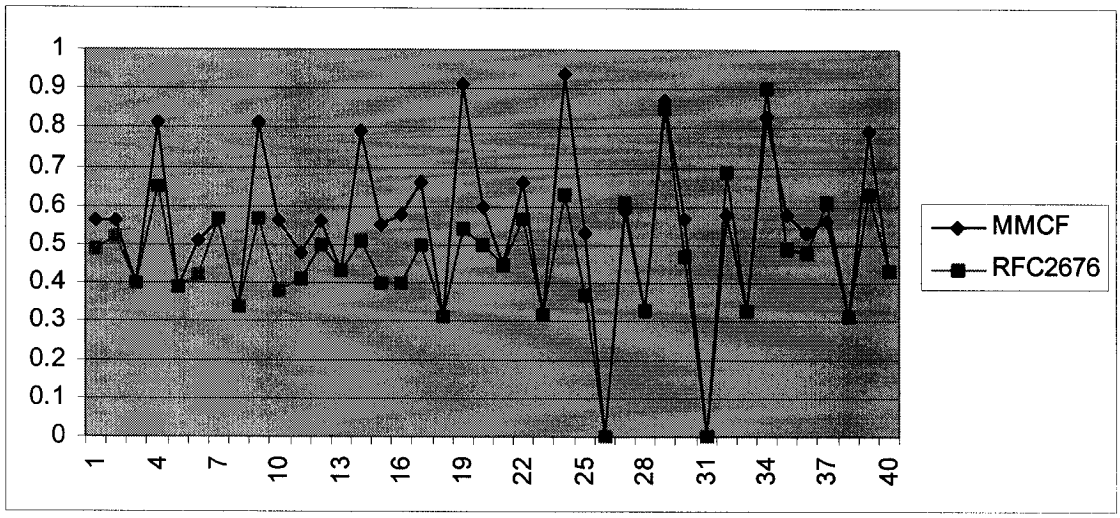


Figure 5.11 Path delays for each VPN request in the 8 simulation cases of Topology 1 scenario (3)

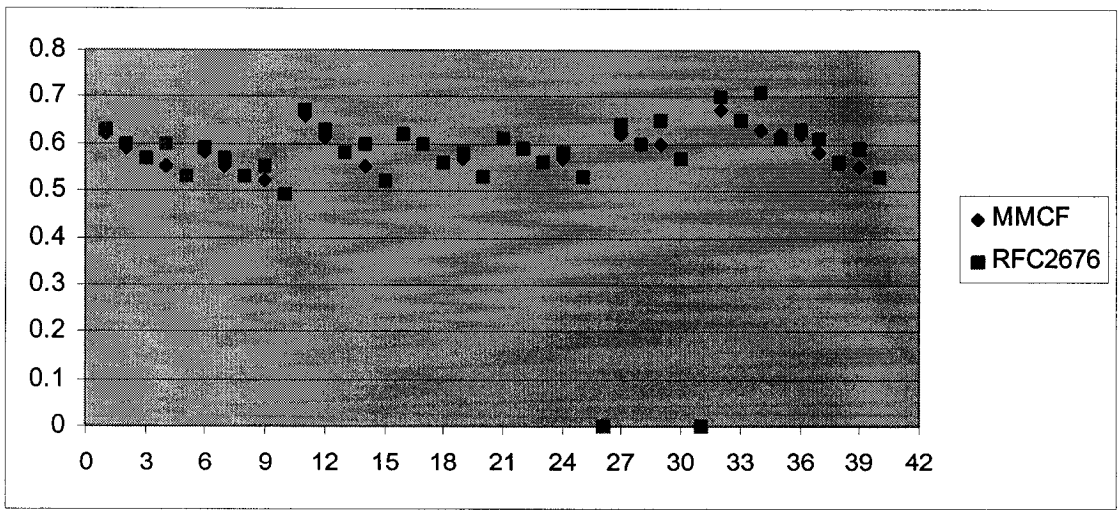


Figure 5.12 Network average queuing delay in 8 simulations cases of Topology 1 scenario (3)

5.3.2 Simulation Results of Network Topology 2

Network Topology 2 is shown in Figure 4.2. The active network status (with the traffic traces applied on each link) of each simulation case is shown in Appendix D.2. We use the same scenarios as shown in Section 5.3.1 Network Topology 1 to run the simulations. The results are the following:

(1) Delay-level 1 = 0.7ms, Delay-level 2 = 1.0ms and Delay level 3 = 1.2ms

Figure 5.13 shows the costs resulting from both the MMCF and RFC2676 algorithms. Table 5.10 displays the detailed information of the simulation results for each case. Figure 5.14 also shows the difference for the path delays of each request among the all eight simulation cases under this scenario. Figure 5.15 indicates the improvement of the network average queuing delay as well.

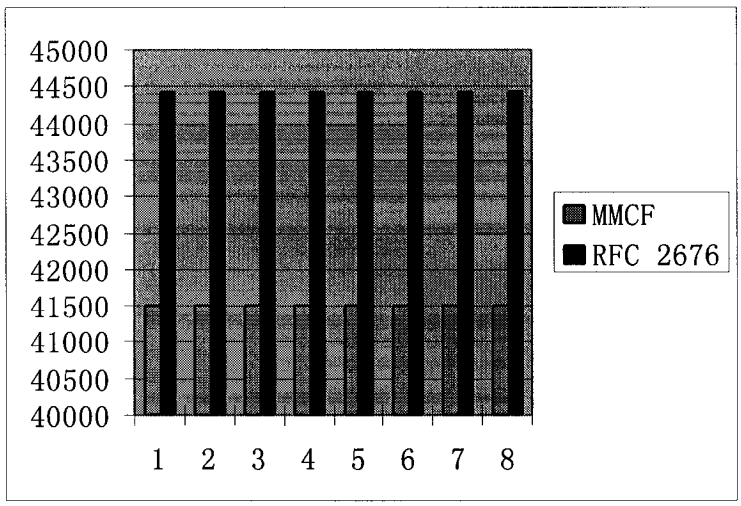


Figure 5.13 Simulation results for Network Topology 2 scenario (1)

Table 5.10 Simulation data corresponding to Topology 2 scenario (1)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.56	0.64	0.95	6 7 1 10	0.49	0.65	0.95	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.61	0.91	6 2 1 8	0.56	0.62	0.92	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.4	0.59	0.87	3 0 2 6	0.4	0.59	0.87	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.68	0.6	0.88	5 2 1 10	0.65	0.62	0.89	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.62	0.56	0.82	7 1 2 5	0.39	0.55	0.81	
Case1 0:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.51	0.6	0.91	6 2 1 10	0.42	0.61	0.91	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.57	0.87	6 7 1 8	0.57	0.58	0.88	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.34	0.54	0.82	3 0 2 6	0.34	0.54	0.82	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.62	0.55	0.83	5 2 1 10	0.57	0.57	0.84	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.56	0.51	0.77	7 6 2 5	0.38	0.51	0.76	
Case2 3:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.48	0.67	0.96	6 7 1 10	0.41	0.68	0.96	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.62	0.9	6 2 1 8	0.5	0.64	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.59	0.84	3 0 2 6	0.43	0.59	0.84	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.56	0.59	0.84	5 2 1 10	0.51	0.61	0.85	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.55	0.53	0.75	7 6 2 5	0.4	0.53	0.75	
Case3 6:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.58	0.64	0.97	6 2 1 10	0.4	0.64	0.96	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.62	0.95	6 2 1 8	0.5	0.61	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.58	0.88	3 0 2 6	0.31	0.58	0.88	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.69	0.6	0.9	5 2 1 10	0.54	0.6	0.89	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.6	0.55	0.83	7 1 2 5	0.5	0.55	0.83	
Case4 9:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 1 10	0.45	0.63	0.98	6 2 1 10	0.45	0.63	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.62	0.96	6 7 1 8	0.57	0.61	0.96	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.32	0.58	0.91	3 0 2 6	0.32	0.58	0.91	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.76	0.61	0.93	5 2 1 10	0.63	0.61	0.92	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.53	0.55	0.86	7 6 2 5	0.37	0.55	0.86	
Case5 12:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.64	0.7	1.01	6 7 1 10	0.62	0.7	1.01	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.59	0.65	0.97	6 2 1 8	0.59	0.67	0.98	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.63	0.92	3 0 2 6	0.33	0.63	0.92	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.83	0.66	0.95	5 2 1 10	0.85	0.68	0.97	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.62	0.6	0.87	7 6 2 5	0.47	0.6	0.87	
Case6 15:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.63	0.74	1.05	6 2 1 10	0.66	0.76	1.06	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.58	0.7	1	6 7 1 8	0.69	0.73	1.02	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.68	0.96	3 0 2 6	0.33	0.68	0.96	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.78	0.7	0.98	5 2 1 10	0.9	0.74	1.01	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.68	0.64	0.91	7 6 2 5	0.49	0.64	0.9	
Case7 18:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.70ms)	6 2 0 1 10	0.53	0.67	0.97	6 2 1 10	0.48	0.68	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.63	0.93	6 7 1 8	0.61	0.65	0.95	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.6	0.87	3 0 2 6	0.31	0.6	0.87	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.63	0.61	0.88	5 2 1 10	0.63	0.64	0.9	
(7, 5, 3M, 0.70ms)	7 1 0 2 5	0.6	0.56	0.82	7 6 2 5	0.43	0.56	0.81	
Case8 21:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay).

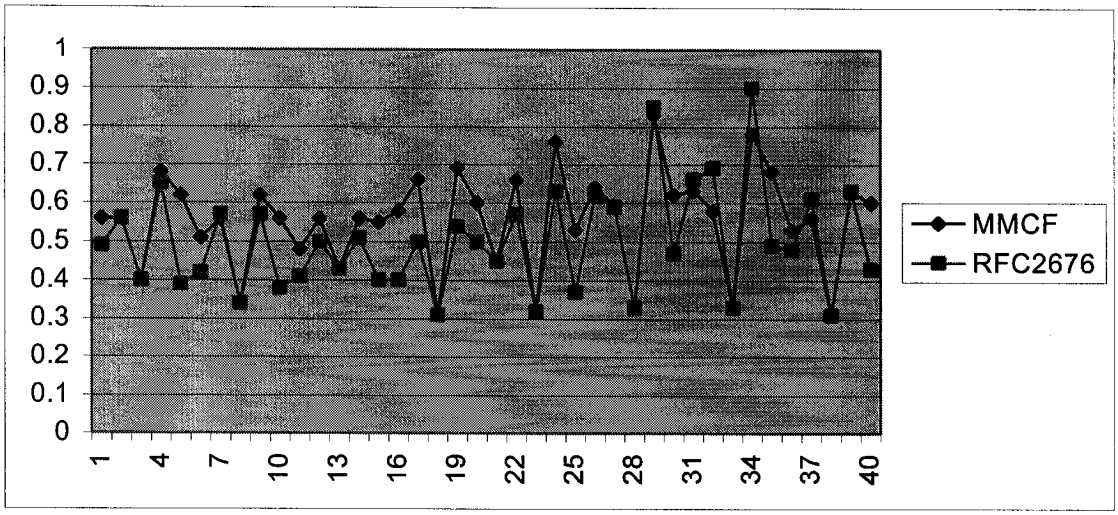


Figure 5.14 Path delays for each VPN request in the 8 simulation cases of Topology 2 scenario (1)

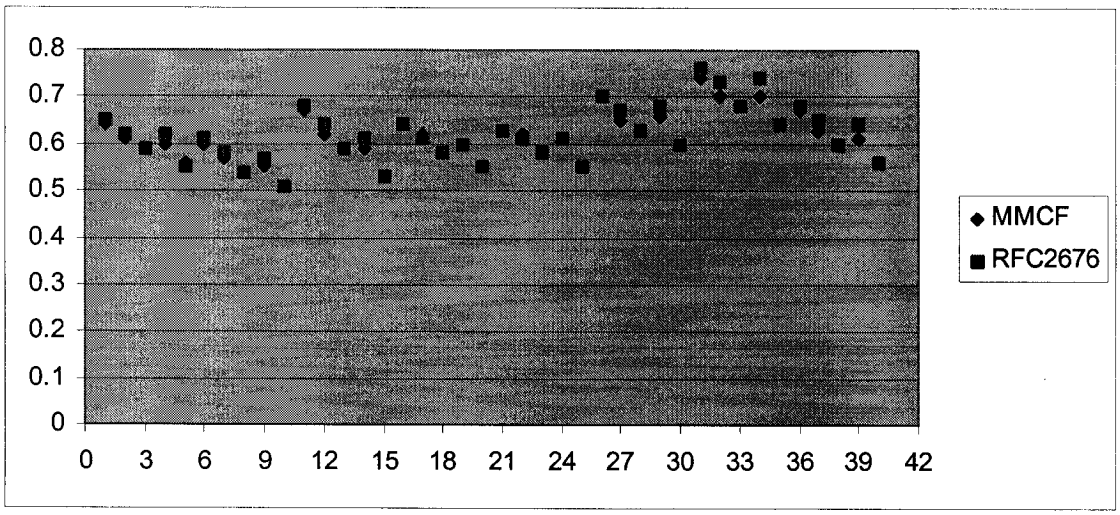


Figure 5.15 Network average queuing delay in 8 simulations cases of Topology 2 scenario (1)

(2) Delay-level 1 = 0.63ms, Delay-level 2 = 1.0ms and Delay level 3 = 1.2ms

Figure 5.16 shows the costs resulting from both the MMCF and RFC2676 algorithms. Table 5.11 displays the detailed information of the simulation results for each case. Figure 5.17 also shows the difference for the path delays of each request among the all

eight simulation cases under this scenario. Figure 5.18 indicates the improvement of the network average queuing delay as well.

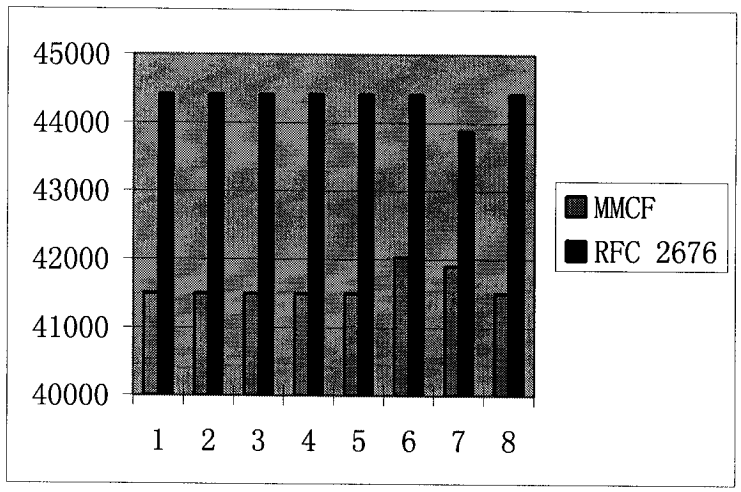


Figure 5.16 Simulation results for Network Topology 2 scenario (2)

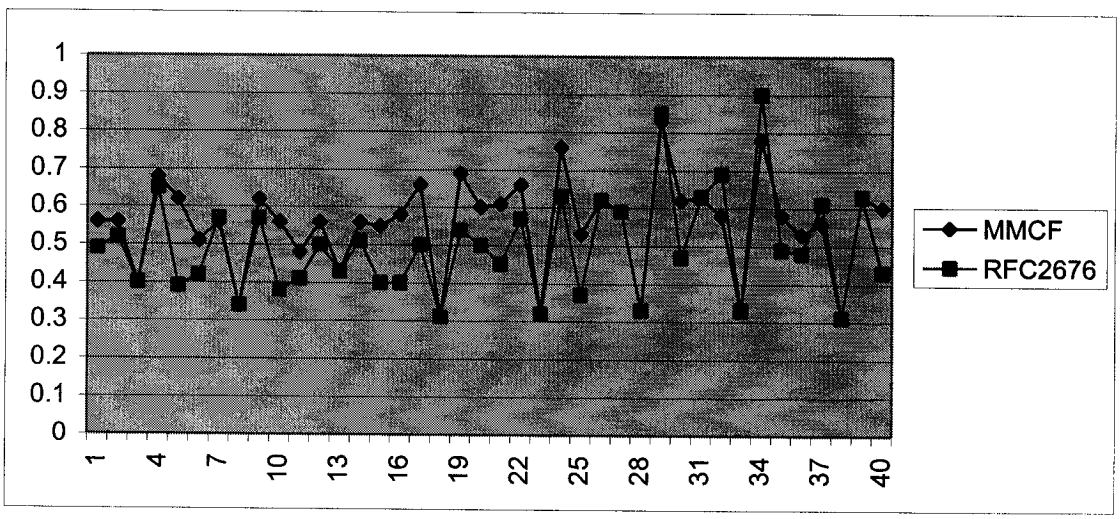


Figure 5.17 Path delays for each VPN request in the 8 simulation cases of Topology 2 scenario (2)

Table 5.11 Simulation data corresponding to Topology 2 scenario (2)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.56	0.64	0.95	6 7 1 10	0.49	0.65	0.95	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.61	0.91	6 2 1 8	0.52	0.62	0.92	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.4	0.59	0.87	3 0 2 6	0.4	0.59	0.87	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.68	0.6	0.88	5 2 1 10	0.65	0.62	0.89	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.62	0.56	0.82	7 1 2 5	0.39	0.55	0.81	
Case1 0:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.51	0.6	0.91	6 2 1 10	0.42	0.61	0.91	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.57	0.87	6 7 1 8	0.57	0.58	0.88	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.34	0.54	0.82	3 0 2 6	0.34	0.54	0.82	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.62	0.55	0.83	5 2 1 10	0.57	0.57	0.84	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.56	0.51	0.77	7 6 2 5	0.38	0.51	0.76	
Case2 3:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.48	0.67	0.96	6 7 1 10	0.41	0.68	0.96	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.62	0.9	6 2 1 8	0.5	0.64	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.59	0.84	3 0 2 6	0.43	0.59	0.84	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.56	0.59	0.84	5 2 1 10	0.51	0.61	0.85	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.55	0.53	0.75	7 6 2 5	0.4	0.53	0.75	
Case3 6:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.58	0.64	0.97	6 2 1 10	0.4	0.64	0.96	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.62	0.95	6 2 1 8	0.5	0.61	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.58	0.88	3 0 2 6	0.31	0.58	0.88	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.69	0.6	0.9	5 2 1 10	0.54	0.6	0.89	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.6	0.55	0.83	7 1 2 5	0.5	0.55	0.83	
Case4 9:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.61	0.64	0.99	6 2 1 10	0.45	0.63	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.62	0.96	6 7 1 8	0.57	0.61	0.96	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.32	0.58	0.91	3 0 2 6	0.32	0.58	0.91	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.76	0.61	0.93	5 2 1 10	0.63	0.61	0.92	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.53	0.55	0.86	7 6 2 5	0.37	0.55	0.86	
Case5 12:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.63ms)	6 7 1 10	0.62	0.7	1.01	6 7 1 10	0.62	0.7	1.01	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.59	0.65	0.97	6 2 1 8	0.59	0.67	0.98	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.63	0.92	3 0 2 6	0.33	0.63	0.92	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.83	0.66	0.95	5 2 1 10	0.85	0.68	0.97	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.62	0.6	0.87	7 6 2 5	0.47	0.6	0.87	
Case6 15:00	**total_cost($): 42030**				**total_cost($): 44424**				**5.39**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.63	0.74	1.05	6 2 0 1 1(	0.63	0.74	1.05	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.58	0.7	1	6 7 1 8	0.69	0.73	1.02	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.68	0.96	3 0 2 6	0.33	0.68	0.96	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.78	0.7	0.98	5 2 1 10	0.9	0.74	1.01	
(7, 5, 3M, 0.63ms)	7 1 2 5	0.58	0.65	0.91	7 6 2 5	0.49	0.64	0.9	
Case7 18:00	**total_cost($): 41897**				**total_cost($): 43892**				**4.55**
(6, 10, 4M, 0.63ms)	6 2 0 1 10	0.53	0.67	0.97	6 2 1 10	0.48	0.68	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.63	0.93	6 7 1 8	0.61	0.65	0.95	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.6	0.87	3 0 2 6	0.31	0.6	0.87	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.63	0.61	0.88	5 2 1 10	0.63	0.64	0.9	
(7, 5, 3M, 0.63ms)	7 1 0 2 5	0.6	0.56	0.82	7 6 2 5	0.43	0.56	0.81	
Case8 21:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay).

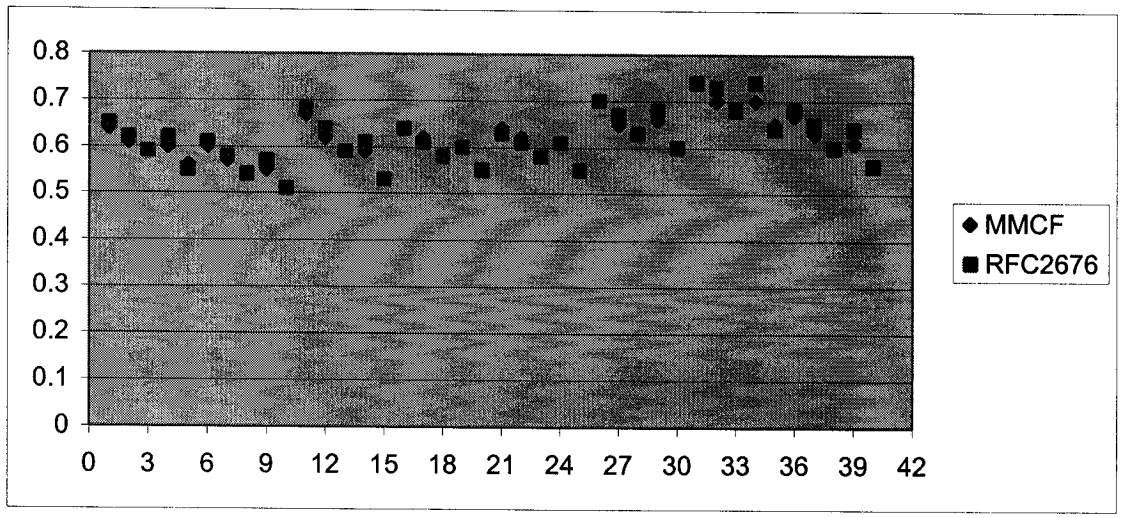


Figure 5.18 Network average queuing delay in 8 simulations cases of Topology 2 scenario (2)

(3) Delay-level 1 = 0.6ms, Delay-level 2 = 1.0ms and Delay level 3 = 1.2ms

Figure 5.19 shows the costs resulting from both MMCF and RFC2676 algorithms. Table 5.12 displays the detailed information of the simulation results for each case. Table 5.12 also indicates that for both algorithms there is no path to satisfy the delay requirement of the VPN request (6, 10, 4M, 0.6ms). As we did in Section 5.3.1 this request would also be rejected. Thus in the path delay (Figure 5.20) and average network delay (neglecting processing and propagation delay) of Figure 5.21, the delay of this request is assigned to '0' as well.

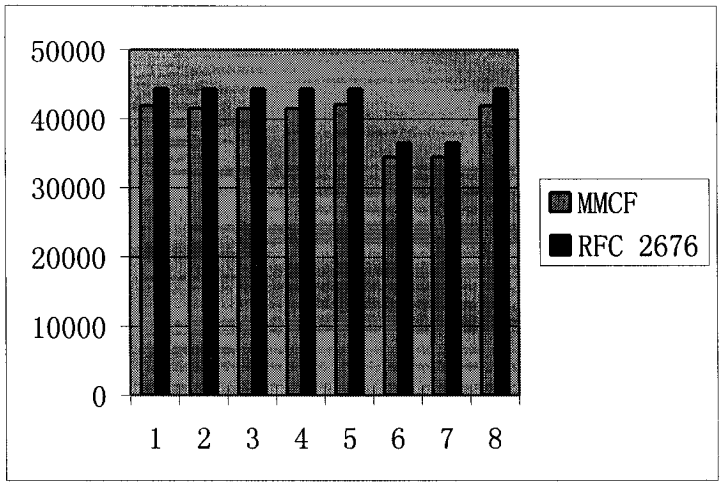


Figure 5.19 Simulation results for Network Topology 2 scenario (3)

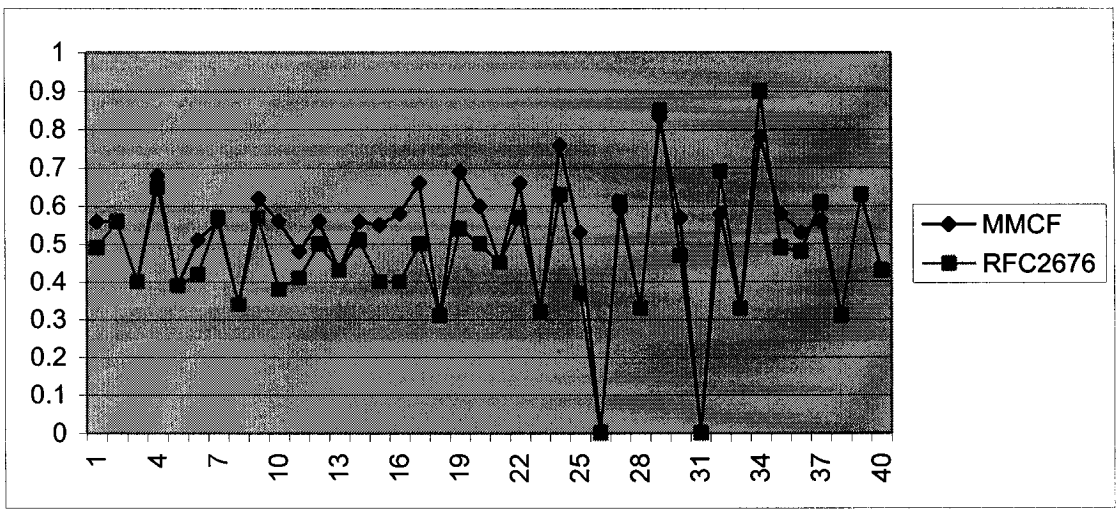


Figure 5.20 Path delays for each VPN request in the 8 simulation cases of Topology 2 scenario (3)

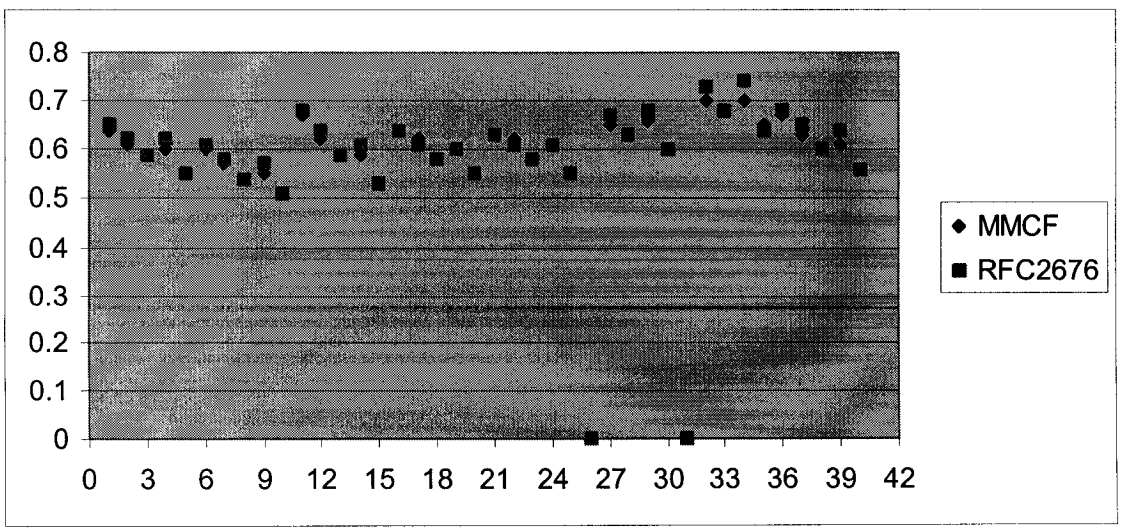


Figure 5.21 Network average queuing delay in 8 simulations cases of Topology 2 scenario (3)

Table 5.12 Simulation data corresponding to Topology 2 scenario (3)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.56	0.64	0.95	6 7 1 10	0.49	0.65	0.95	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.61	0.91	6 2 1 8	0.56	0.62	0.92	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.4	0.59	0.87	3 0 2 6	0.4	0.59	0.87	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.68	0.6	0.88	5 2 1 10	0.65	0.62	0.89	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.39	0.55	0.81	7 1 2 5	0.39	0.55	0.81	
Case1 0:00	**total_cost($): 41897**				**total_cost($): 44424**				**5.69**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.51	0.6	0.91	6 2 1 10	0.42	0.61	0.91	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.57	0.87	6 7 1 8	0.57	0.58	0.88	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.34	0.54	0.82	3 0 2 6	0.34	0.54	0.82	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.62	0.55	0.83	5 2 1 10	0.57	0.57	0.84	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.56	0.51	0.77	7 6 2 5	0.38	0.51	0.76	
Case2 3:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.48	0.67	0.96	6 7 1 10	0.41	0.68	0.96	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.62	0.9	6 2 1 8	0.5	0.64	0.91	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.59	0.84	3 0 2 6	0.43	0.59	0.84	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.56	0.59	0.84	5 2 1 10	0.51	0.61	0.85	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.55	0.53	0.75	7 6 2 5	0.4	0.53	0.75	
Case3 6:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.58	0.64	0.97	6 2 1 10	0.4	0.64	0.96	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.62	0.95	6 2 1 8	0.5	0.61	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.58	0.88	3 0 2 6	0.31	0.58	0.88	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.69	0.6	0.9	5 2 1 10	0.54	0.6	0.89	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.6	0.55	0.83	7 1 2 5	0.5	0.55	0.83	
Case4 9:00	**total_cost($): 41498**				**total_cost($): 44424**				**6.59**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.45	0.63	0.98	6 2 1 10	0.45	0.63	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.66	0.62	0.96	6 7 1 8	0.57	0.61	0.96	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.32	0.58	0.91	3 0 2 6	0.32	0.58	0.91	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.76	0.61	0.93	5 2 1 10	0.63	0.61	0.92	
(7, 5, 3M, 0.60ms)	7 1 0 2 5	0.53	0.55	0.86	7 6 2 5	0.37	0.55	0.86	
Case5 12:00	**total_cost($): 42030**				**total_cost($): 44424**				**5.39**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.61*	0.7	1.01	6 2 1 10	0.61 *	0.7	1.01	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.59	0.65	0.97	6 7 1 8	0.61	0.67	0.98	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.63	0.92	3 0 2 6	0.33	0.63	0.92	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.83	0.66	0.95	5 2 1 10	0.85	0.68	0.97	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.57	0.6	0.88	7 6 2 5	0.47	0.6	0.87	
Case6 15:00	**total_cost($): 34425**				**total_cost($): 36420**				**5.48**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.63 *	0.74	1.05	6 2 0 1 1	0.63 *	0.74	1.05	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.58	0.7	1	6 7 1 8	0.69	0.73	1.02	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.68	0.96	3 0 2 6	0.33	0.68	0.96	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.78	0.7	0.98	5 2 1 10	0.9	0.74	1.01	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.58	0.65	0.91	7 6 2 5	0.49	0.64	0.9	
Case7 18:00	**total_cost($): 34425**				**total_cost($): 36420**				**5.48**
(6, 10, 4M, 0.60ms)	6 2 0 1 10	0.53	0.67	0.97	6 2 1 10	0.48	0.68	0.98	
(6, 8, 7M, 1.00ms)	6 2 0 1 8	0.56	0.63	0.93	6 7 1 8	0.61	0.65	0.95	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.6	0.87	3 0 2 6	0.31	0.6	0.87	
(5, 10, 8M, 1.00ms)	5 2 0 1 10	0.63	0.61	0.88	5 2 1 10	0.63	0.64	0.9	
(7, 5, 3M, 0.60ms)	7 6 2 5	0.43	0.56	0.81	7 6 2 5	0.43	0.56	0.81	
Case8 21:00	**total_cost($): 41897**				**total_cost($): 44424**				**5.69**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay); *: PD exceeds delay requirement

Analysis:

Having compared the simulation results of each scenario between Network Topology 1 and Network Topology 2, we can find that the difference of the results between these two topologies comes from the different link distributions. Therefore we ran the simulations on another topology, Network Topology 3, in order to see further how extensively the link distributions influence the results.

5.3.3 Simulation Results of Network Topology 3

Network Topology 3 is shown in Figure 4.3. The active network status (with the traffic traces applied on each link) of each simulation case is shown in Appendix D.3. We ran the simulations under the same scenarios as we did in Network Topology 1 and 2. All results come from the three scenarios are identical. Thus we just show the results for scenario (3) and all the other two are the same as this. The results are the Figure 5.22 shows the costs resulting from both the MMCF and RFC2676 algorithms. Table 5.13 displays the detailed information of the simulation results for each case.

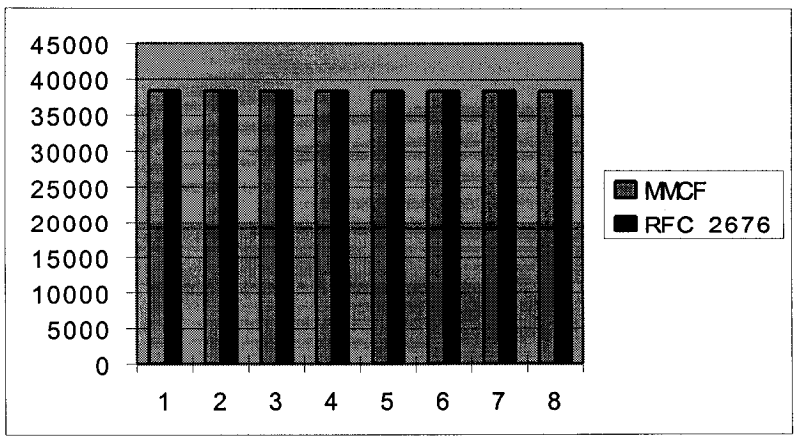


Figure 5.22 Simulation results for Network Topology 3 scenario (3)

Table 5.13 Simulation data corresponding to Topology 3 scenario (3)

Simulation cases	MMCF Model				RFC 2676 Model				Saving
	selected path	PD (ms)	AQD (ms)	AD (ms)	selected path	PD (ms)	AQD (ms)	AD (ms)	(%)
(6, 10, 4M, 0.60ms)	6 2 1 10	0.39	0.64	0.94	6 2 1 10	0.39	0.64	0.94	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.44	0.61	0.91	6 2 1 8	0.44	0.61	0.91	
(3, 6, 2M, 1.20ms)	3 1 2 6	0.36	0.59	0.87	3 1 2 6	0.36	0.59	0.87	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.51	0.6	0.87	5 2 1 10	0.51	0.6	0.87	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.37	0.55	0.81	7 1 2 5	0.37	0.55	0.81	
Case1 0:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.34	0.6	0.9	6 2 1 10	0.34	0.6	0.9	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.49	0.58	0.88	6 2 1 8	0.49	0.58	0.88	
(3, 6, 2M, 1.20ms)	3 1 2 6	0.35	0.54	0.82	3 1 2 6	0.35	0.54	0.82	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.46	0.55	0.82	5 2 1 10	0.46	0.55	0.82	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.38	0.5	0.76	7 1 2 5	0.38	0.5	0.76	
Case2 3:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.32	0.67	0.97	6 2 1 10	0.32	0.67	0.97	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.5	0.64	0.93	6 2 1 8	0.5	0.64	0.93	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.43	0.59	0.86	3 0 2 6	0.43	0.59	0.86	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.4	0.58	0.83	5 2 1 10	0.4	0.58	0.83	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.39	0.53	0.76	7 1 2 5	0.39	0.53	0.76	
Case3 6:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.32	0.63	0.96	6 2 1 10	0.32	0.63	0.96	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.49	0.61	0.94	6 2 1 8	0.49	0.61	0.94	
(3, 6, 2M, 1.20ms)	3 1 2 6	0.31	0.58	0.89	3 1 2 6	0.31	0.58	0.89	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.42	0.58	0.88	5 2 1 10	0.42	0.58	0.88	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.43	0.55	0.84	7 1 2 5	0.43	0.55	0.84	
Case4 9:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.37	0.63	0.96	6 2 1 10	0.37	0.63	0.96	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.5	0.61	0.94	6 2 1 8	0.5	0.61	0.94	
(3, 6, 2M, 1.20ms)	3 1 2 6	0.36	0.58	0.89	3 1 2 6	0.36	0.58	0.89	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.51	0.59	0.89	5 2 1 10	0.51	0.59	0.89	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.37	0.55	0.84	7 1 2 5	0.37	0.55	0.84	
Case5 12:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.48	0.69	1	6 2 1 10	0.48	0.69	1	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.6	0.67	0.98	6 2 1 8	0.6	0.67	0.98	
(3, 6, 2M, 1.20ms)	3 1 2 6	0.37	0.63	0.93	3 1 2 6	0.37	0.63	0.93	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.67	0.66	0.95	5 2 1 10	0.67	0.66	0.95	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.44	0.6	0.87	7 1 2 5	0.44	0.6	0.87	
Case6 15:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.46	0.74	1.05	6 2 1 10	0.46	0.74	1.05	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.51	0.71	1.01	6 2 1 8	0.51	0.71	1.01	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.33	0.68	0.96	3 0 2 6	0.33	0.68	0.96	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.61	0.7	0.98	5 2 1 10	0.61	0.7	0.98	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.52	0.64	0.91	7 1 2 5	0.52	0.64	0.91	
Case7 18:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**
(6, 10, 4M, 0.60ms)	6 2 1 10	0.37	0.67	0.98	6 2 1 10	0.37	0.67	0.98	
(6, 8, 7M, 1.00ms)	6 2 1 8	0.55	0.65	0.95	6 2 1 8	0.55	0.65	0.95	
(3, 6, 2M, 1.20ms)	3 0 2 6	0.31	0.6	0.88	3 0 2 6	0.31	0.6	0.88	
(5, 10, 8M, 1.00ms)	5 2 1 10	0.46	0.61	0.88	5 2 1 10	0.46	0.61	0.88	
(7, 5, 3M, 0.60ms)	7 1 2 5	0.44	0.56	0.82	7 1 2 5	0.44	0.56	0.82	
Case8 21:00	**total_cost($): 38424**				**total_cost($): 38424**				**0**

PD: Path Delay; AQD: Average Queueing Delay; AD: Average Delay (including propgation delay).

Analysis:

As the tables show that both the MMCF algorithm and the RFC2676 (OSPF) algorithm select the same path. Therefore, the costs of the two algorithms are the same. These results are astounding compared with the results we obtained from Network Topology 1 and Network Topology 2. We can ask what could be the reason that makes this difference. For a specific simulation case among the three topologies, the traffic data on each link are the same and the VPN connection requests are the same as well. Thus we can find that the reason, which results in the difference among the topologies, is the link distribution. Compared with Network Topology 1 and 2, Network Topology 3 is a symmetric network in terms of the distribution of the OC3 links. The reason that the MMCF algorithm performs better over the asymmetric networks is the MMCF focus on finding the cheapest cost path for the VPN service instead of the minimal hops. However, if the network is symmetric, no matter how many paths it can find, the link's components are all the same and therefore, the costs are the same.

Chapter 6. Conclusions and Future Work

The objective of the research is to seek a cost-savings routing algorithm for the network service provider who offers VPN services.

In this thesis, a MMCF algorithm for VPN services is proposed, simulated and compared with the currently widely used OSPF routing algorithm.

Simulations were run under the constraints of bandwidth and delay. From the analysis of the simulation results the following general conclusions can be made:

Both OSPF and MMCF schemes are designed to meet the VPN users' requirements. However, the MMCF algorithm seems to be more attractive than OSPF in terms of cost-savings for a VPN service provider. Among the three topologies evaluated, MMCF obtains the highest improvement for Topology 1 (9%) and the highest for Topology 3 (0%). It seems that asymmetric networks can benefit more from this algorithm.

The results also show that the MMCF algorithm is more beneficial when the delay constraint is not as strict (a special case is when only bandwidth is the restrictive parameter). The smaller the delay requirement value is, the lower the cost benefits from the MMCF algorithm. This should be expected, since RFC2676 (OSPF) is looking after providing short delays to the paths by its nature. Overall, MMCF is a better option than OSPF.

In this thesis, we have considered the bandwidth and delay elements. However, there are other QoS requirements such as jitter and packet loss that could be included as well. Therefore, more work can be done in the future in order to find a better solution for both VPN users and service providers.

Reference

[1] "Virtual Private Networks Solutions from Lucent Technologies" http://www.vector.kharkov.com/support/techn/lu_vpn.htm

[2] Huan Liang, Ognian Kabranov, Dimitrios Makrakis, Luis Orozco-Barbosa "Minimal Cost Design of Virtual Private Networks". Electrical and Computer Engineering, IEEE CCECE 2002. Canadian Conference, pp. 1610 –1615

[3] John Mulligan, Director of Enablement AT&T "Challenges for Deploying a Reliable IP VPN", September 23, 2002 http://yankeegroup.com/public/events/conferences/ipvpn/components/JohnMulliganPresentationDay11440.pdf

[4] Nortel Networks "The business case for IP-VPN services – Evaluating CLE-based and network-based solutions" http://www.nortelnetworks.com/products/library/collateral/55051.25-11-00.pdf

[5] Jingdi Zeng; Ansari, N. "Toward IP virtual private network quality of service: a service provider perspective" Communications Magazine, IEEE, Volume: 41 Issue: 4 , April 2003 Page(s): 113 -119

[6] A Framework for IP Based Virtual Private Networks, RFC 2764

[7] http://sunsite.dk/RFC/rfc/rfc2764.htmlTim Armstrong "How Virtual Private Networks (VPNs) Save Money and Improved *Security"* December 1998 http://www.hipaadvisory.com/tech/vpnsecurity.htm

[8] Paul Ferguson Geoff Huston "what is VPN" http://bura.iskon.hr/FAQ/white_papers/Cisco%20-%20What%20is%20VPN.pdf

[9] Ascend Communications, Inc. "Virtual Private Networks for the Enterprise" http://www.firstvpn.com/papers/ascend/vpnentrg.pdf

[10] Mahesh Rathod "VPN - A Very Personal Network", Network magazine, http://www.networkmagazineindia.com/200112/primer1.htm

[11] RFP: VPNs Across Multiple Sites In Ascend Communications' Words Solution, http://www.networkcomputing.com/912/912f13.html

[12] "VPN Cost Savings Analysis for the Enterprise" http://www.adimpleo.com/library/ascend/vpnroirg.pdf

[13] "Compare VPN products, services and prices",http://www.vpn411.com/products/

[14] "Intel Enterprise-Class VPN Solutions" http://www.intel.com/design/network/solutions/ent_vpn/

[15] Gregory J. Ciolek, "Virtual Private Network (VPN) Security", January 4,2001 http://rr.sans.org/encryption/VPN_sec.php

[16] "Petra 2001, Living In A Connected World, Part II" Petra Technology Group, http://www.petragroup.com/docs/newsletters/0102.pdf

[17] NetBSD Documentation: NetBSD IPsec http://www.netbsd.org/Documentation/network/ipsec/

[18] http://whatis.techtarget.com/definition/0,289893,sid9_gci214312,00.html

[19] http://searchnetworking.techtarget.com/sDefinition/0,sid7_gci493383,00.html

[20] Virtual Private Networks, Solutions for cost-effective, high-speed corporate extranets and wide-area networks. A Nextep Broadband White Paper May 2001

[21] Torsten Braun, Manuel Guenter, and Ibrhim Khalil, University of Bern, Switzerland "Management of Quality of Service Enabled VPNs", IEEE Communications Magazine, May 2001.

[22] The Internet Engineering Task Force, http://www.ietf.org/

[23] IP Quality of Service: IntServ and DiffServ, http://www.unc.edu/~gogan/qos/

[24] Multiprotocol Label Switching (mpls), http://www.ietf.org/html.charters/mpls-charter.html

[25] "Private Use of Public Networks for Service Providers" 3Com, Private Use of Public Networks for Service Providers

[26] *"QoS Routing Mechanisms and OSPF Extensions"*, RFC 2676

[27] OSPF version 2, Ascend Communications, Inc. April 1998, RFC 2328.

[28] T. Braun, M. Günter, I. Khalil, "Implementation of a Service Broker for Management of QoS Enabled VPNs", CATI Charging and Accounting Technology for the Internet. http://www.tik.ee.ethz.ch/~cati/deliverables.html

[29] R.Ahuja, T.Magnani, B.Orlin, *"Network Flows, Theory, Algorithms and Applications"*, Prentice Hall, 1993

[30] Infobahn IB, Broadband Internet Access Provider of T1, T3 & OC3 lines, http://www.broadband-internet-provider.com/research-information.htm

[31] Internet Access of The Computer King http://www.thecomputerking.com/internet_access_t3.aspx

[32] Alberto Leon-Garcia "Probability and Random Processes for Electrical Engineering" – 2nd edition. Addison-Wesley, 1994

[33] Dimitri Bertsekas/ Robert Gallager "Data Networks", Ptentice-Hall, 1987

[34] Kleinrock, L., "On the Modeling and Analysis of Computer Networks," Special Issue of IEEE Proceedings, pp. 1179-1191, August 1993.

[35] Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol *"Discrete-Event System Simulation"* Third edition, Prentice Hall 2000

[36] W. E. Leland and D. V. Wilson, *"High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection", Proc. IEEE INFOCOM '91*, April 1991, pp. 1360-1366

[37] http://ita.ee.lbl.gov/html/contrib/BC.html

[38] Akira KATO, Jun MURAI, Satoshi KATSUNO, Tohru ASAMI *"An Internet* Traffic Data Repository: The Architecture and the Design Policy", Internet Society INET 1999 conference http://www.isoc.org/isoc/conferences/inet/99/proceedings/4h/ref5

[39] WIDE project http://www.wide.ad.jp/

[40] MAWI Working Group Traffic Archive, Packet traces from WIDE backbone, aguri port plot data, http://tracer.csl.sony.co.jp/mawi/aguri-ports-E/2002/

[41] MAWI Working Group Traffic Archive, Packet traces from WIDE backbone, aguri port plot data, http://tracer.csl.sony.co.jp/mawi/aguri-ports-B/2002/

[42] Hong-Hsu Yen and Frank Yeong-Sung Lin, 2001 Apr., "Near-optimal Delay Constrained Routing in Virtual Circuit Networks", Proceeding of IEEE INFORM 2001, Vol 2, pp 750-756

Appendix A: OC3 Traffic Traces

The following eight OC3 link traffic trace graphs are cited from WIDE Project [39] that collects daily traffic from the experimental Internet environment. The horizon is the date and time when the traffic was collected. We are interested in the red curve labeled as "total" at the very top of the graph because it is the total traffic stream transmitted on the link.

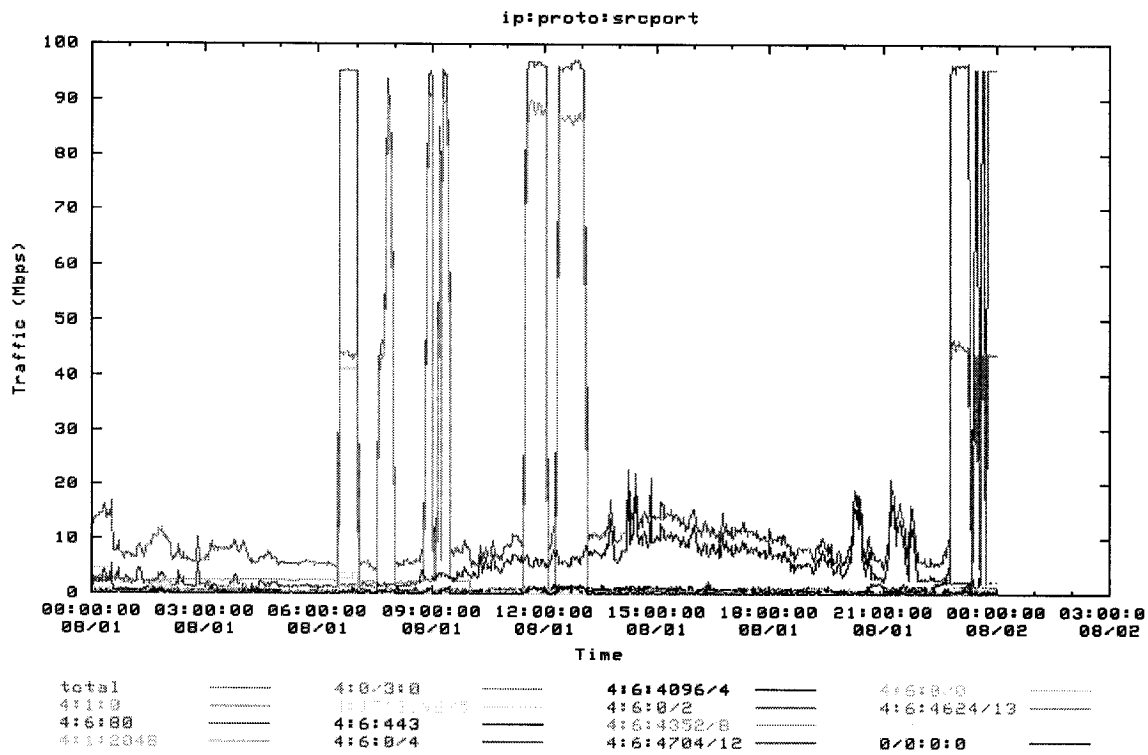


Figure A1: OC3 traffic of August 1st 2002 from WIDE network [40]

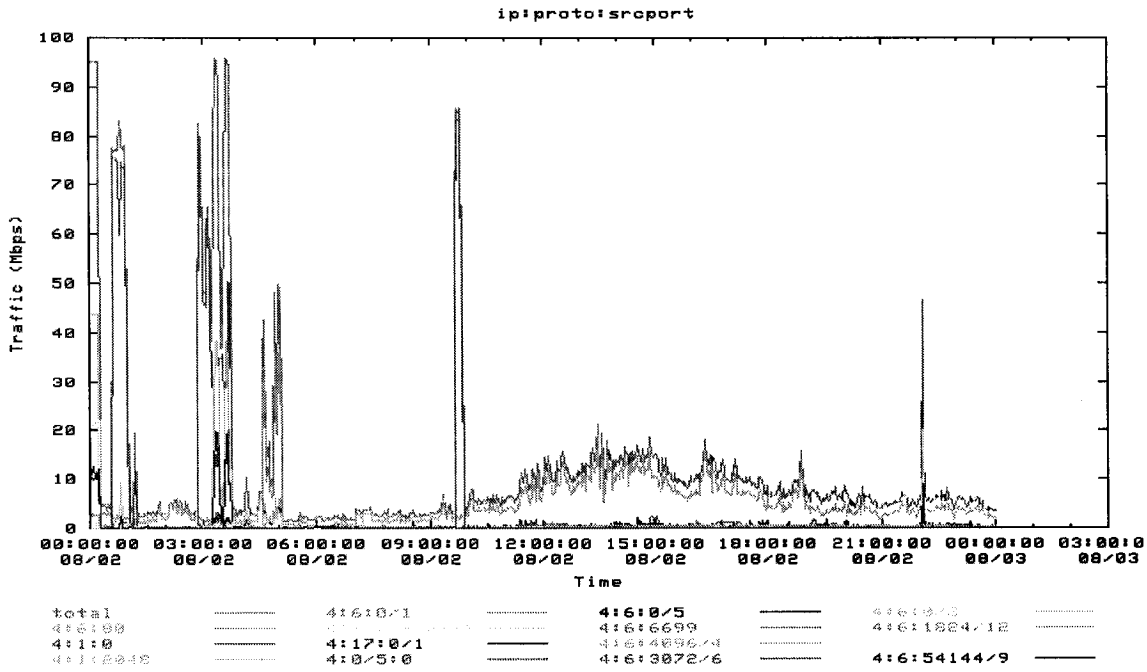


Figure A2: OC3 traffic of August 2nd 2002 from WIDE network [40]

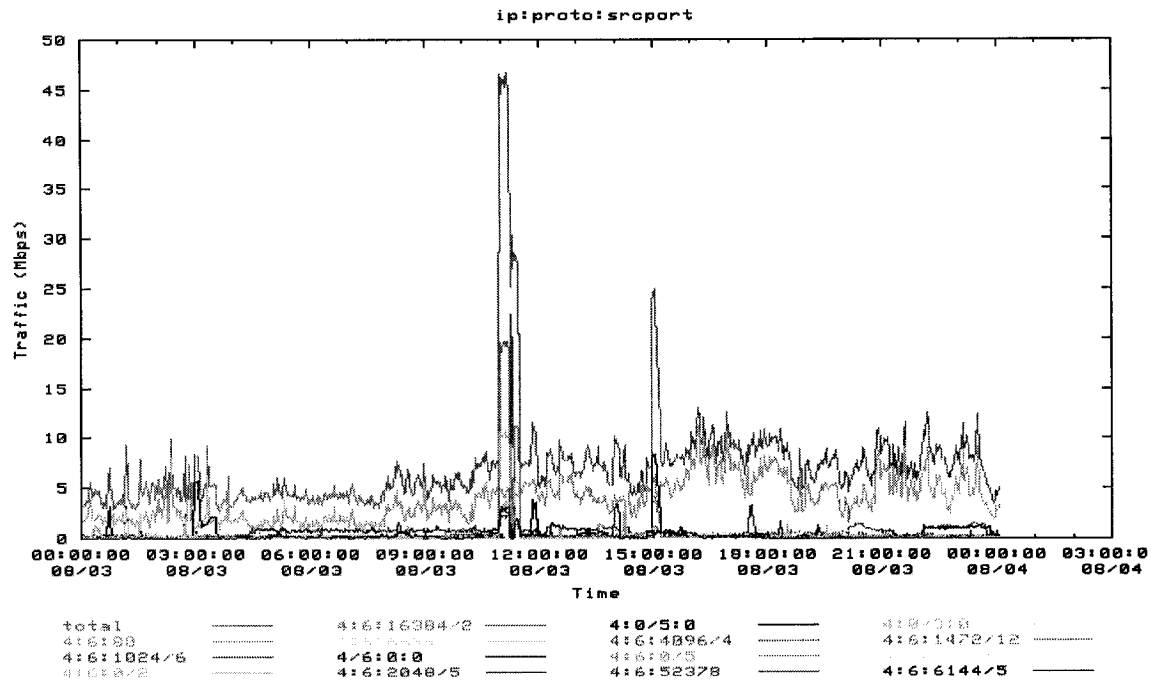


Figure A3: OC3 traffic of August 3rd 2002 from WIDE network [40]

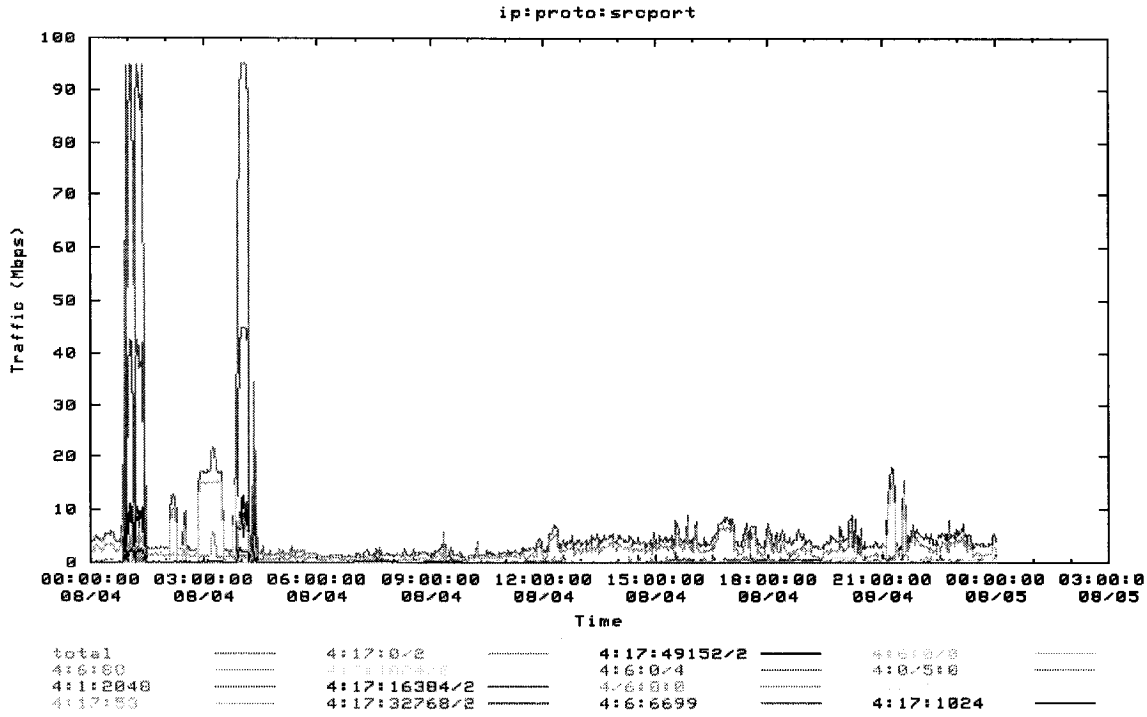


Figure A4: OC3 traffic of August 4th 2002 from WIDE network [40]

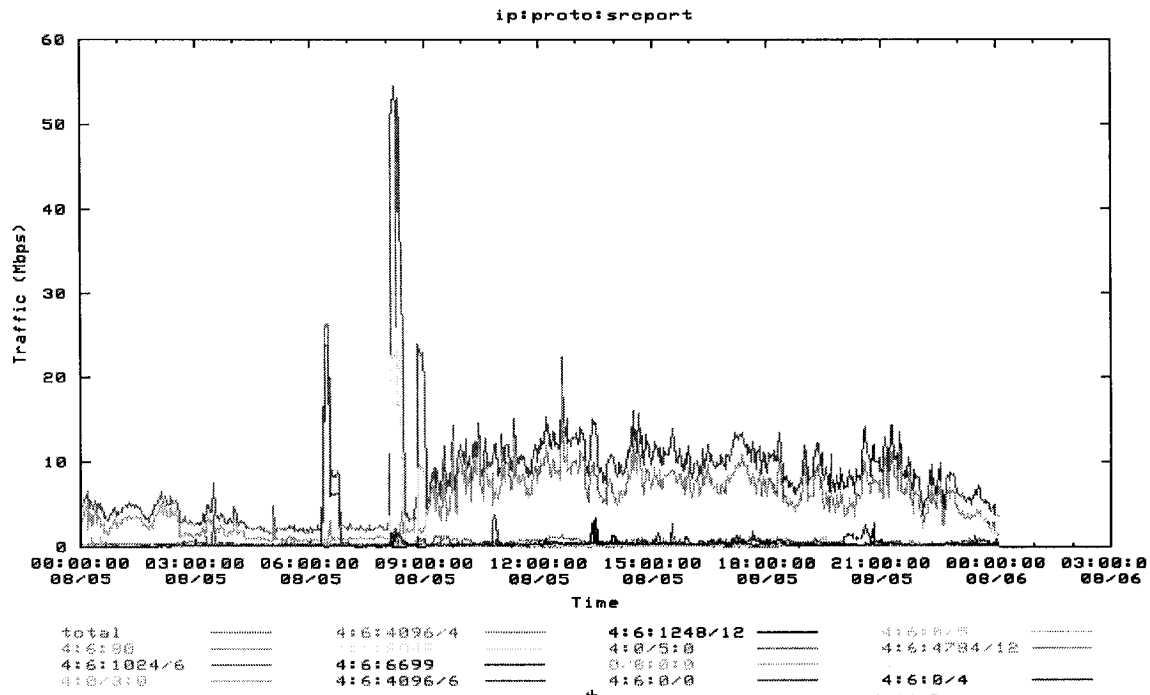


Figure A5: OC3 traffic of August 5th 2002 from WIDE network [40]

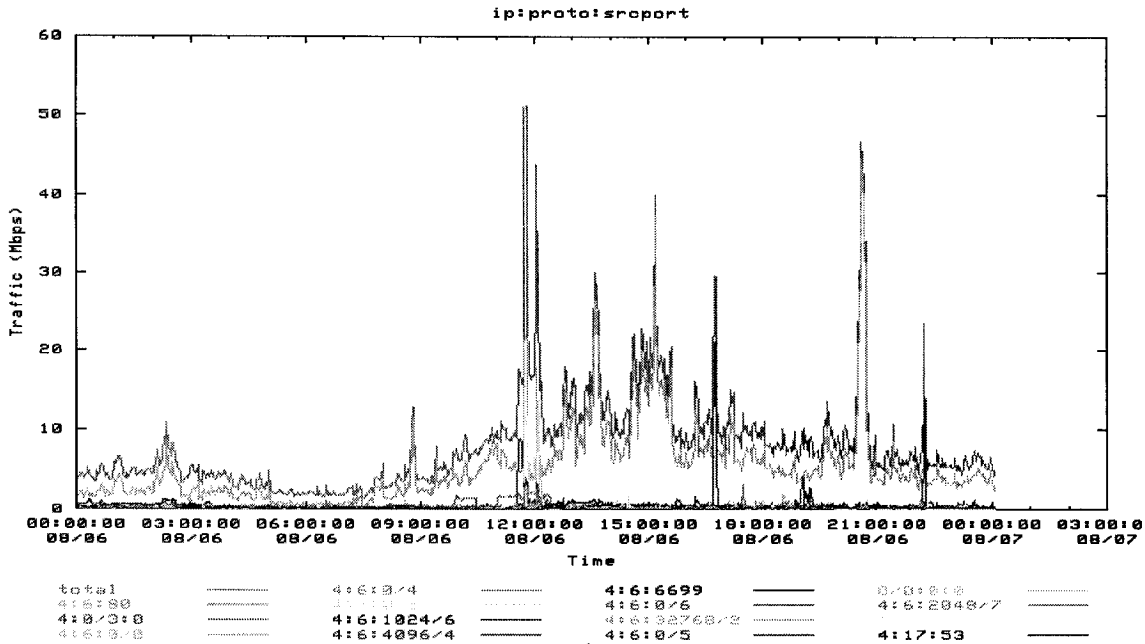


Figure A6: OC3 traffic of August 6th 2002 from WIDE network [40]

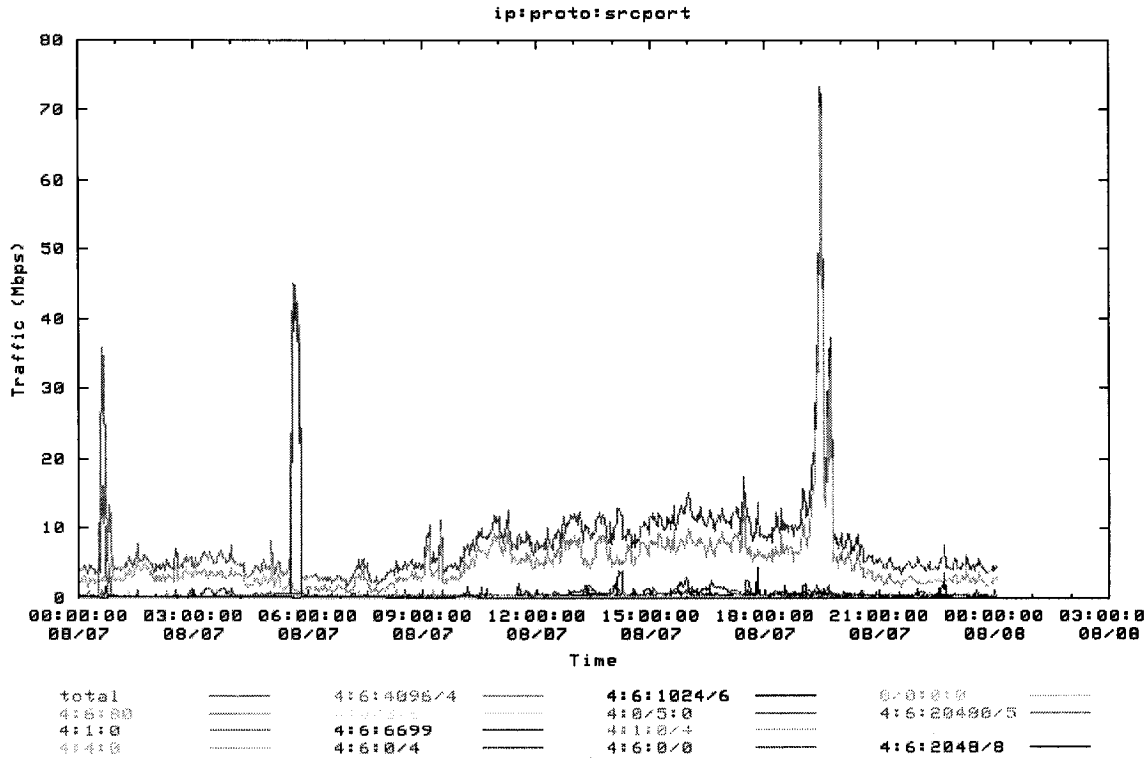


Figure A7: OC3 traffic of August 7th 2002 from WIDE network [40]

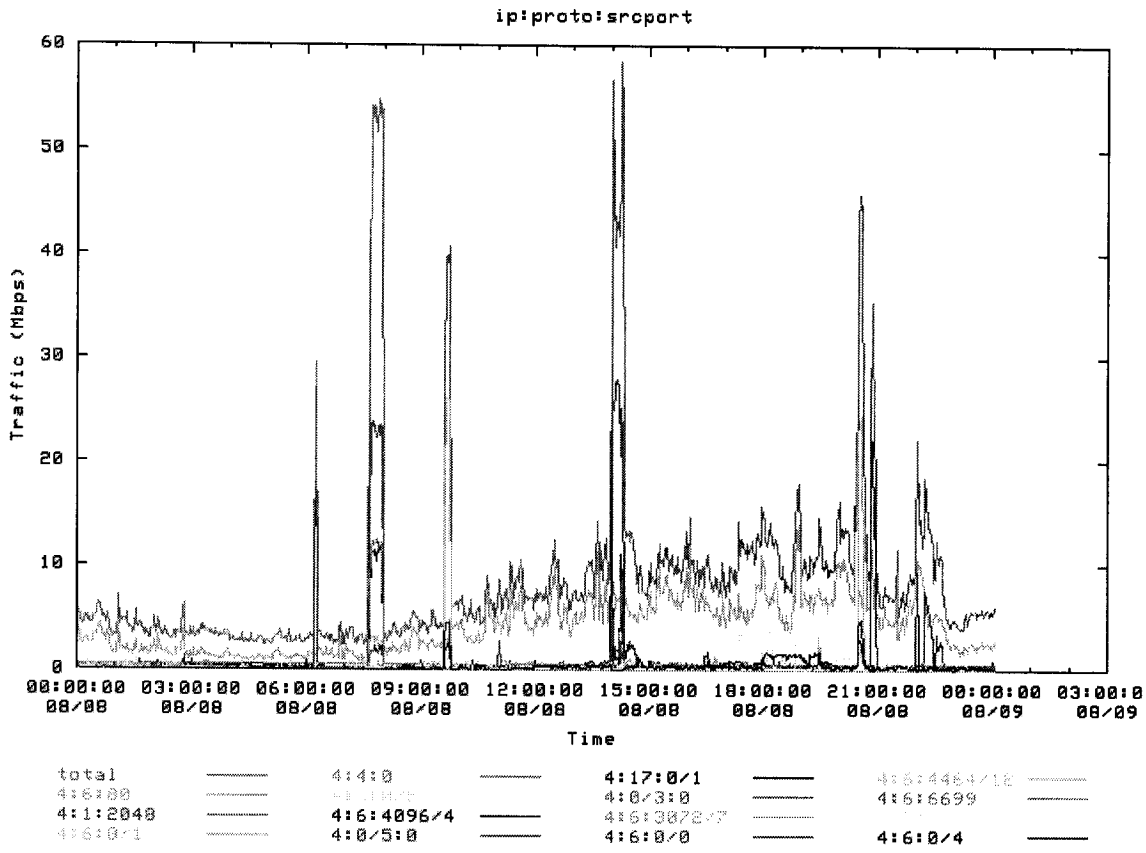


Figure A8: OC3 traffic of August 8th 2002 from WIDE network [40]

Appendix B: T3 Traffic Traces

The following twenty-six T3 link traffic trace graphs are cited from WIDE Project [39] that collects daily traffic from the experimental Internet environment. The horizon is the date and time when the traffic was collected. We are interested in the red curve labeled as "total" at the very top of the graph because it is the total traffic stream transmitted on the link.

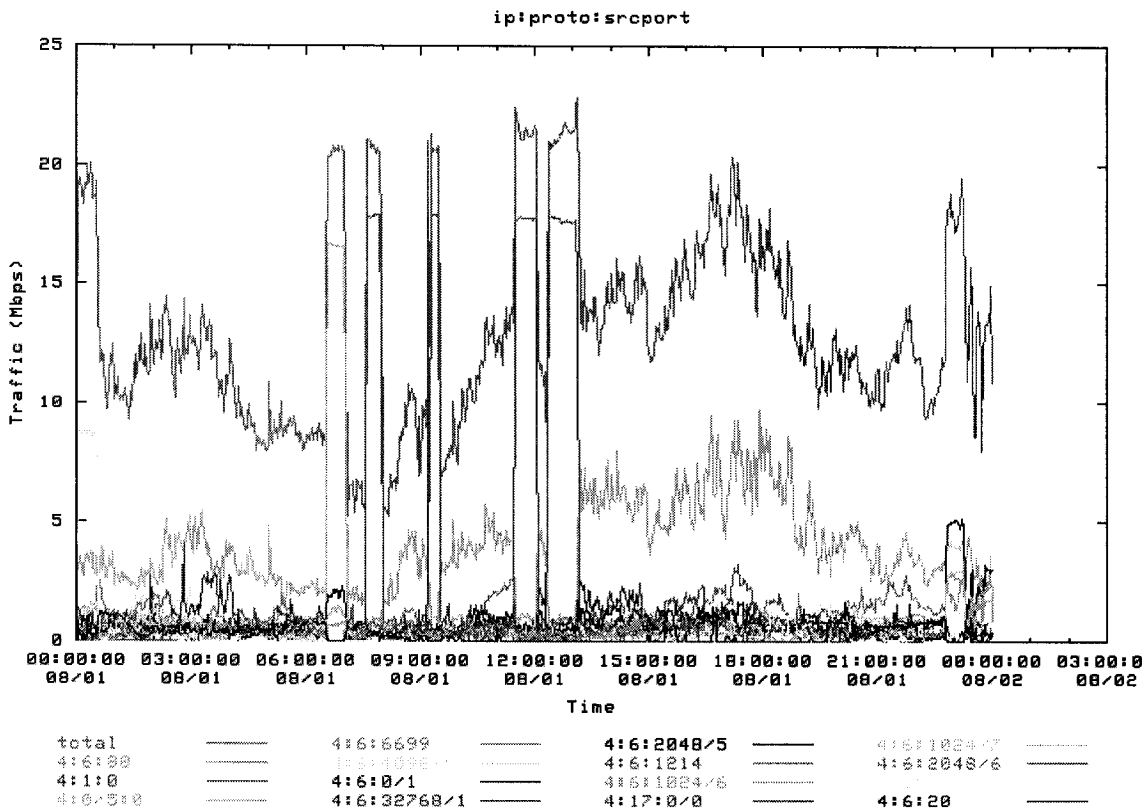


Figure B1: T3 traffic of August 1st 2002 from WIDE network [41]

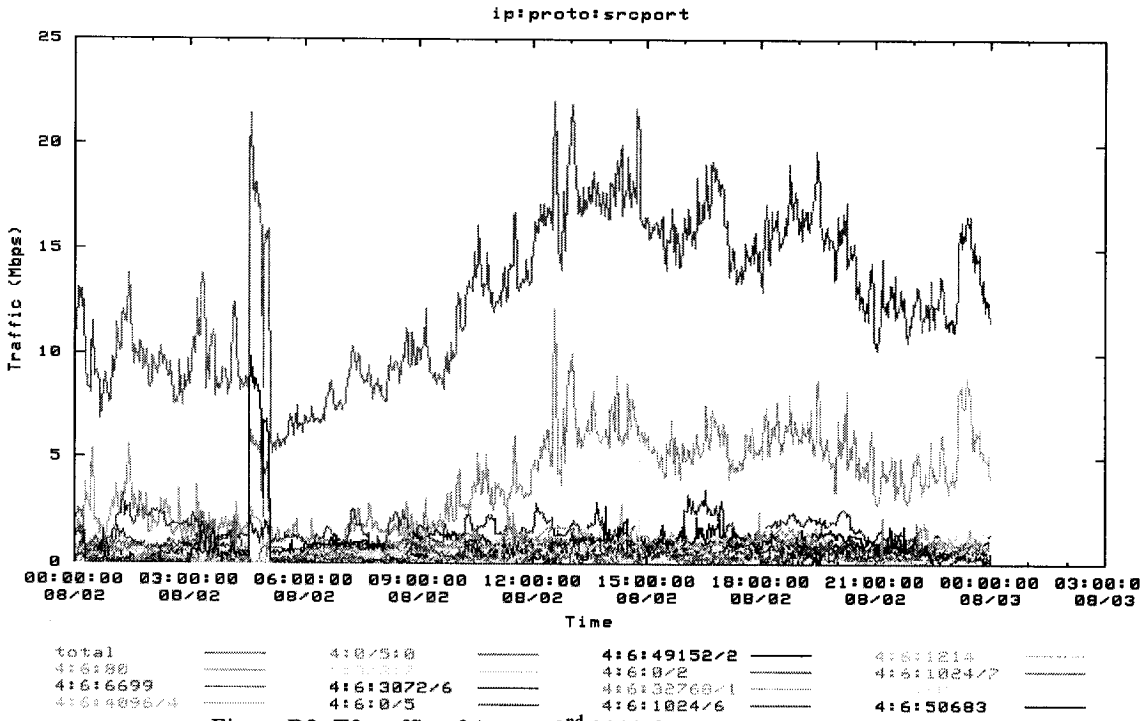


Figure B2: T3 traffic of August 2nd 2002 from WIDE network [41]

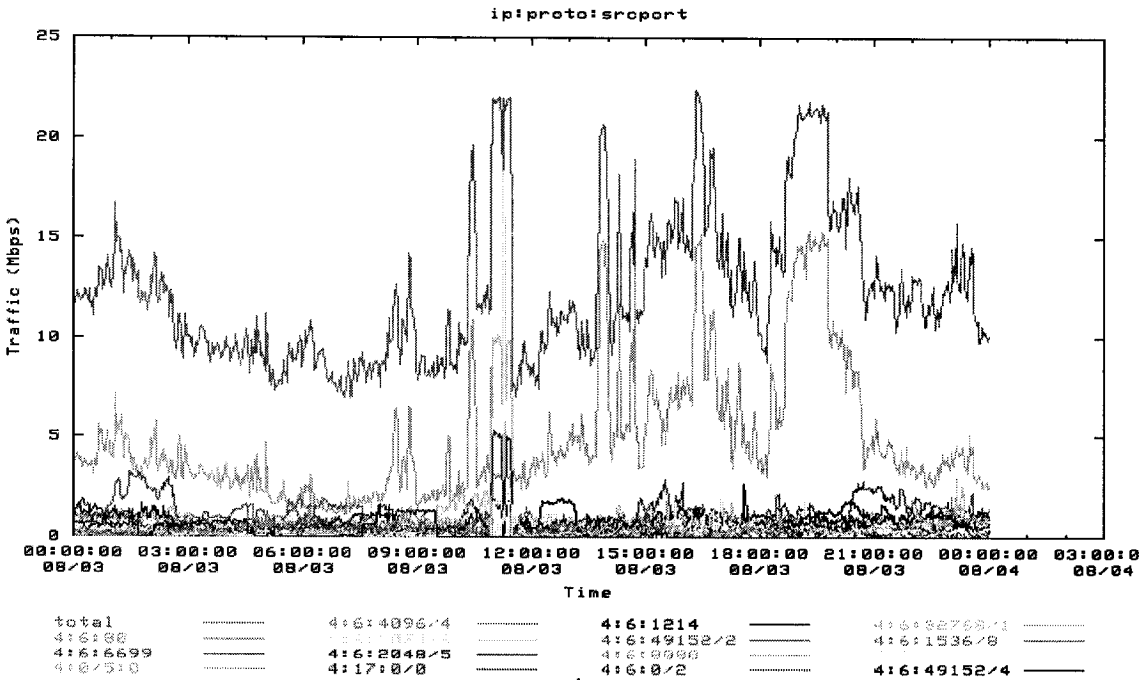


Figure B3: T3 traffic of August 3rd 2002 from WIDE network [41]

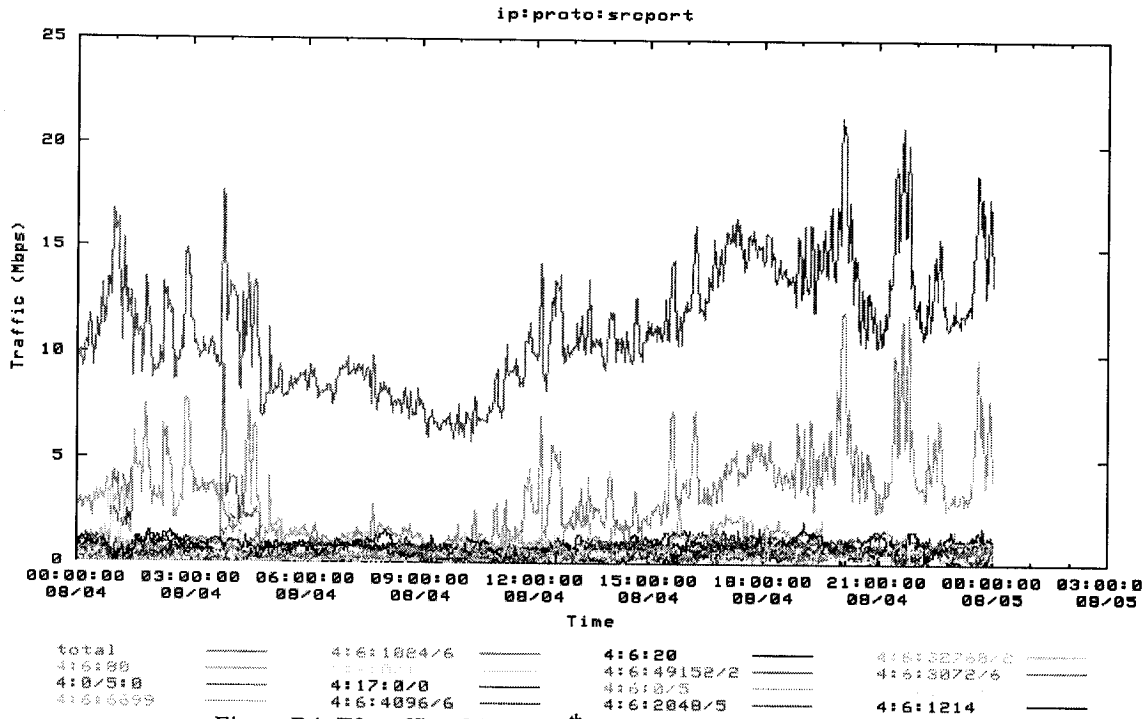


Figure B4: T3 traffic of August 4th 2002 from WIDE network [41]

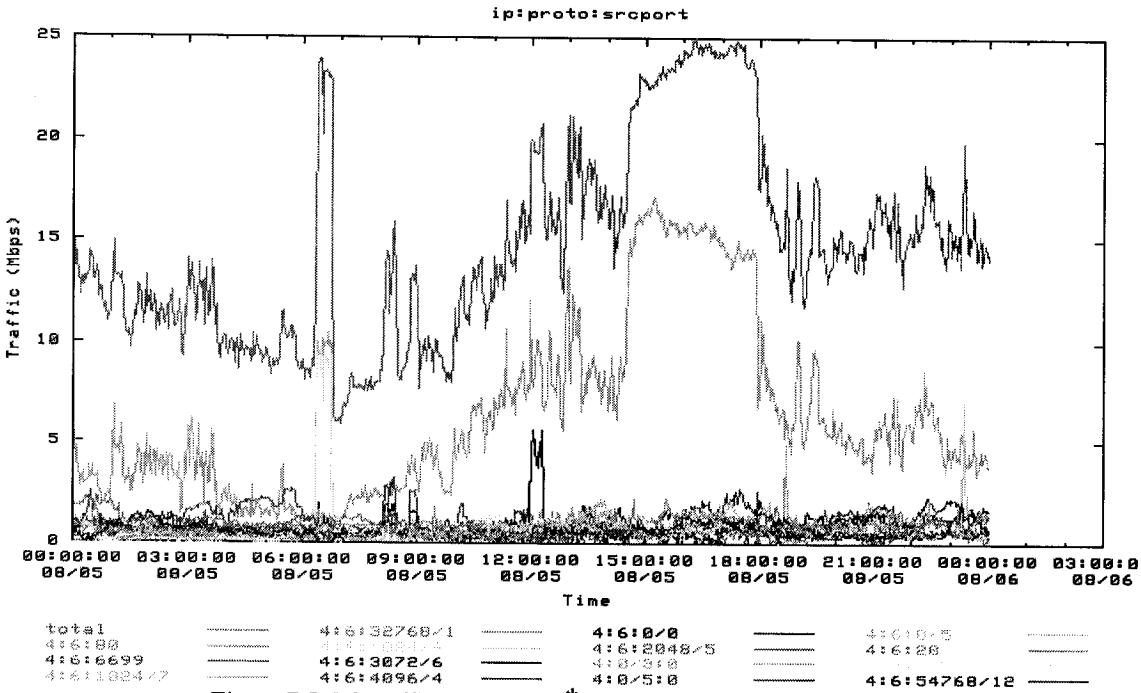


Figure B5: T3 traffic of August 5th 2002 from WIDE network [41]

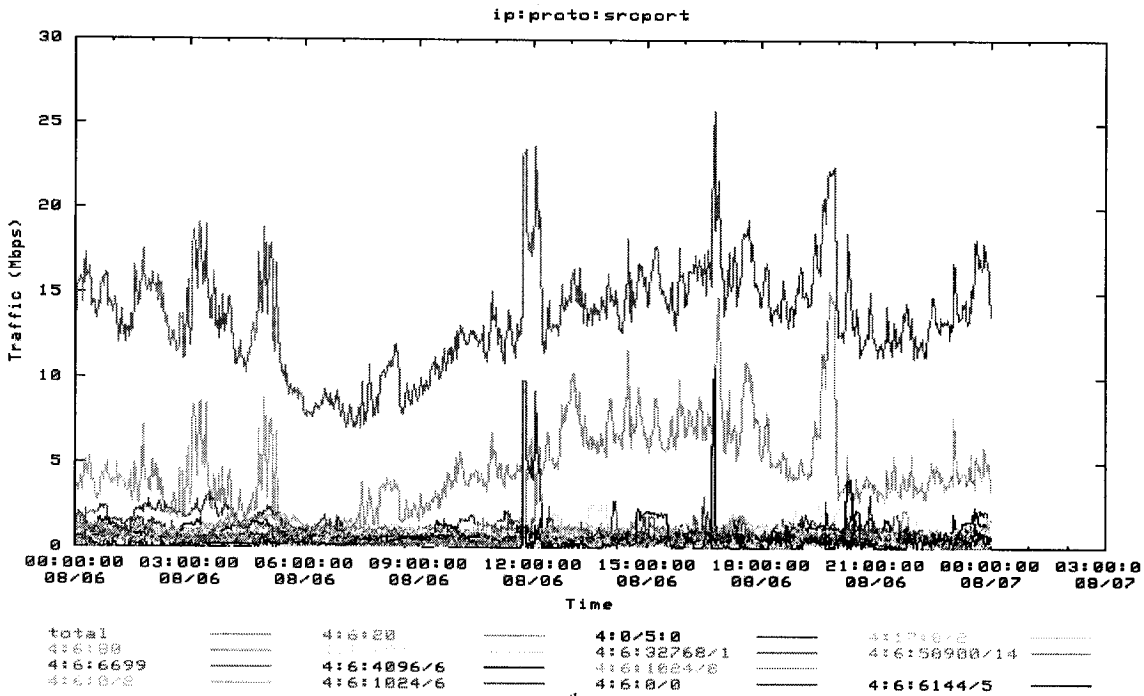


Figure B6: T3 traffic of August 6th 2002 from WIDE network [41]

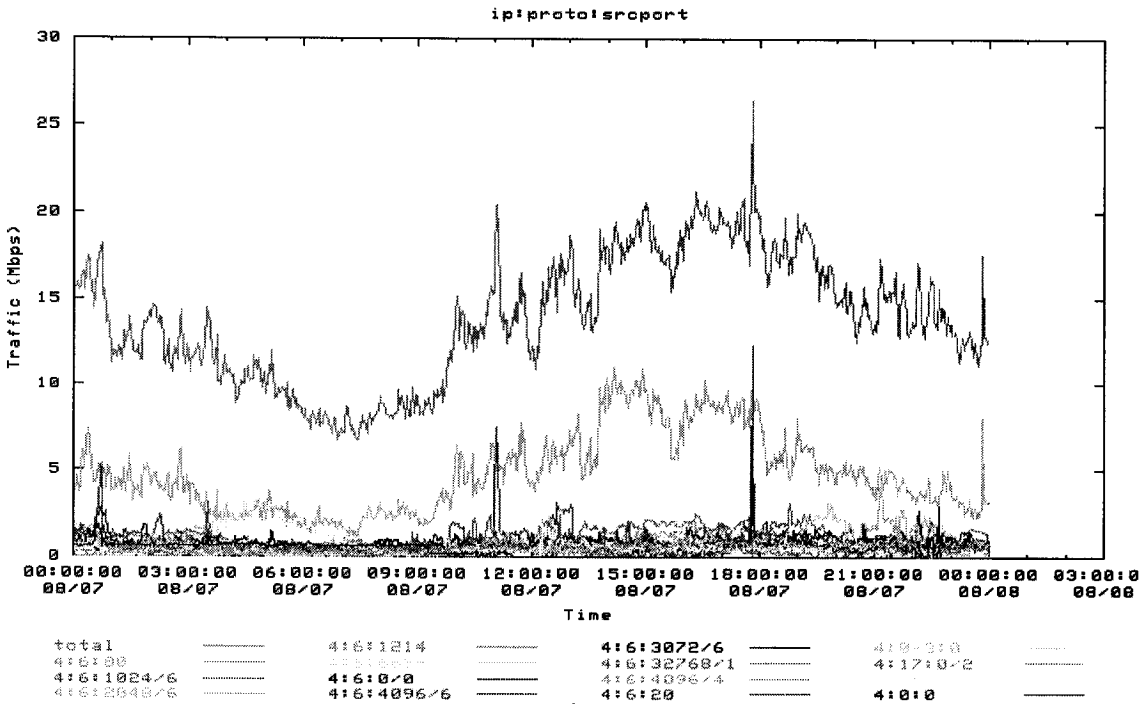


Figure B7: T3 traffic of August 7th 2002 from WIDE network [41]

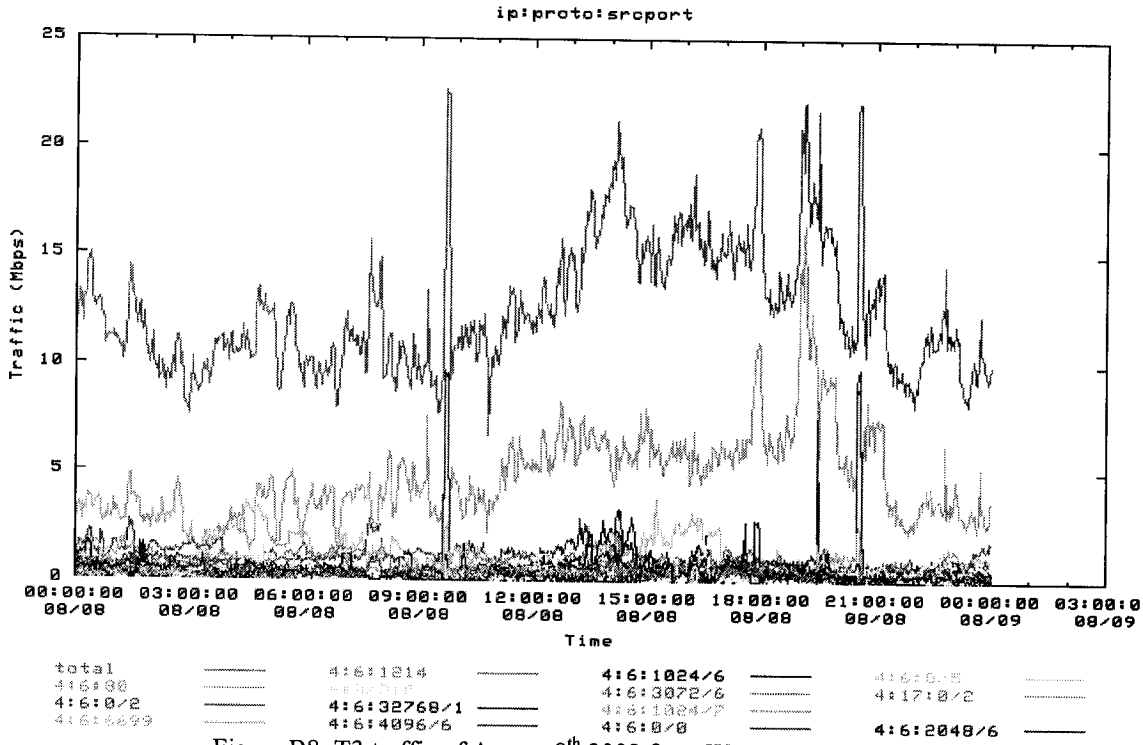


Figure B8: T3 traffic of August 8th 2002 from WIDE network [41]

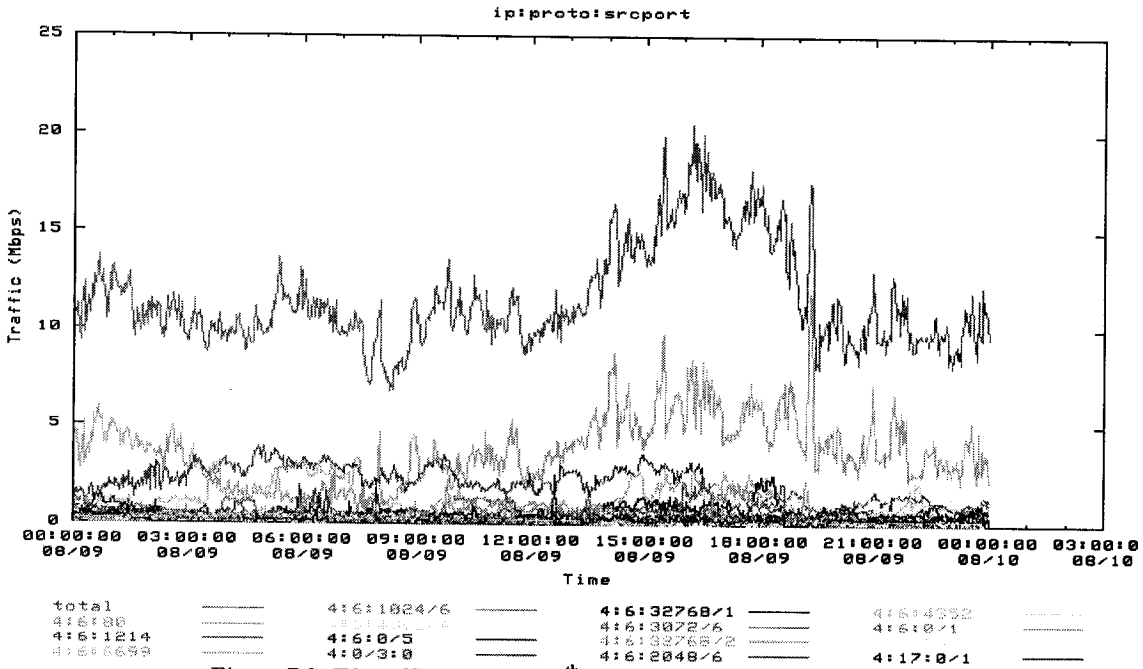


Figure B9: T3 traffic of August 9th 2002 from WIDE network [41]

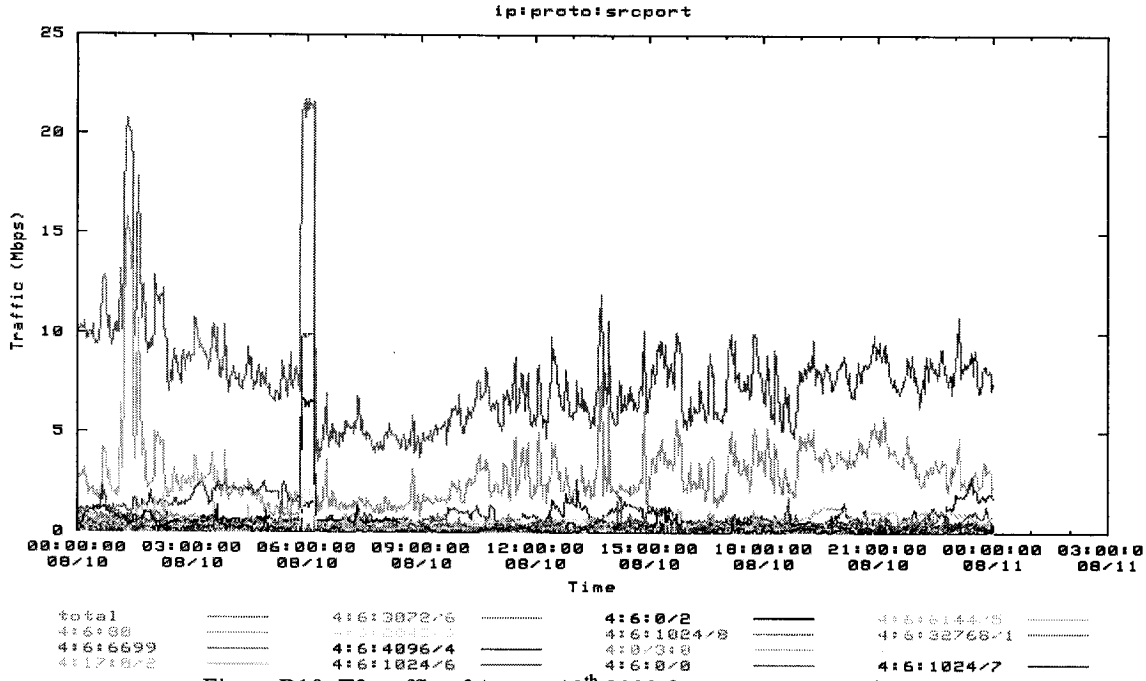


Figure B10: T3 traffic of August 10th 2002 from WIDE network [41]

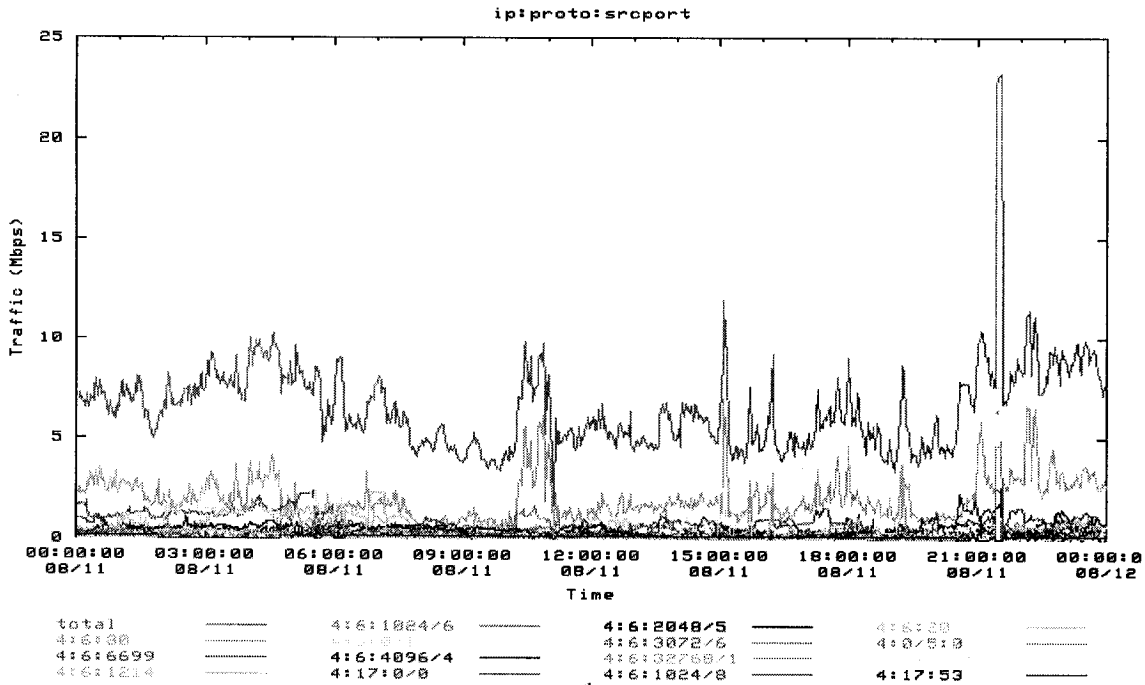


Figure B11: T3 traffic of August 11th 2002 from WIDE network [41]

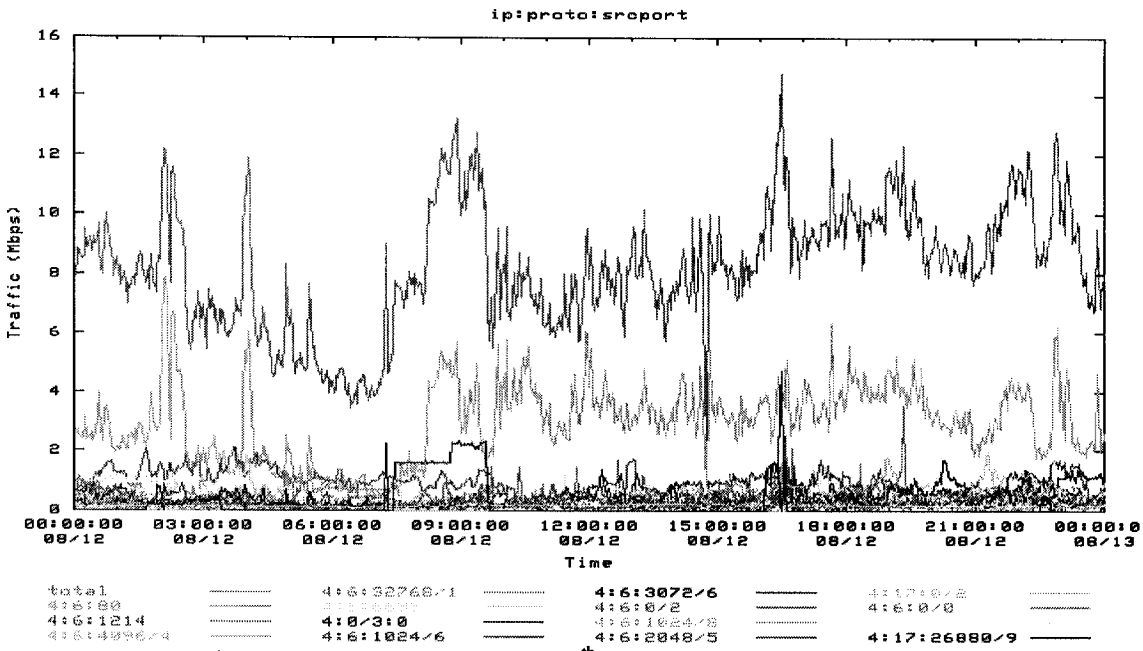


Figure B12: T3 traffic of August 12th 2002 from WIDE network [41]

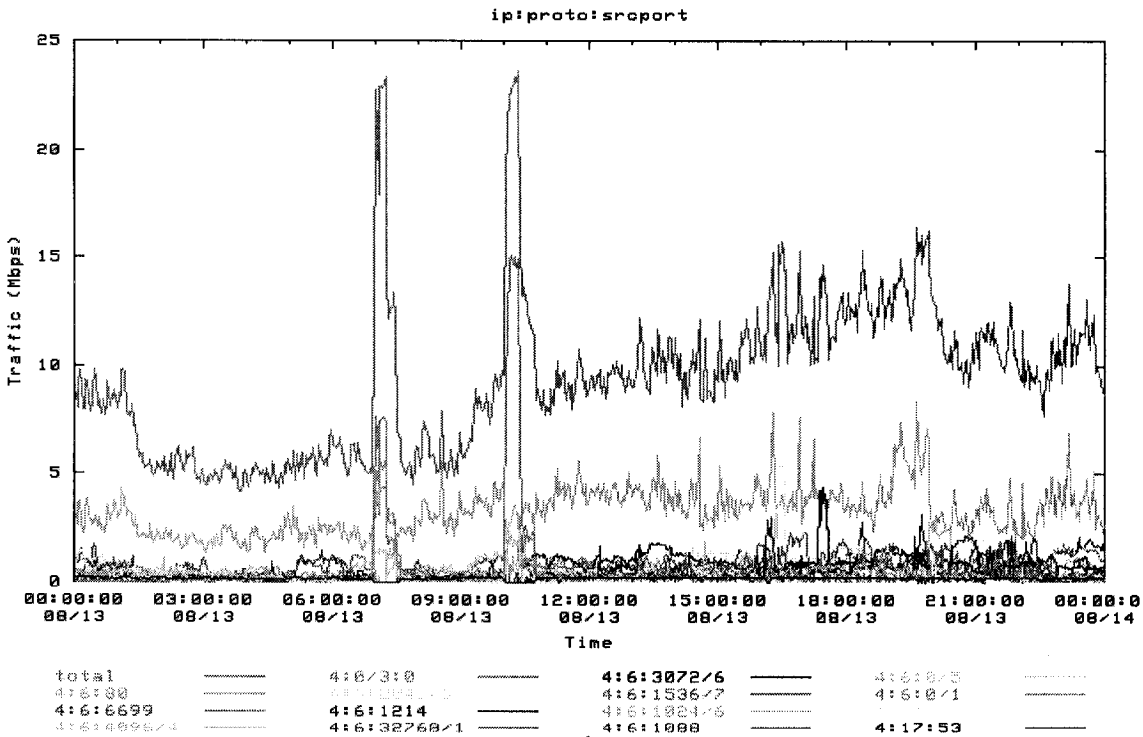


Figure B13: T3 traffic of August 13th 2002 from WIDE network [41]

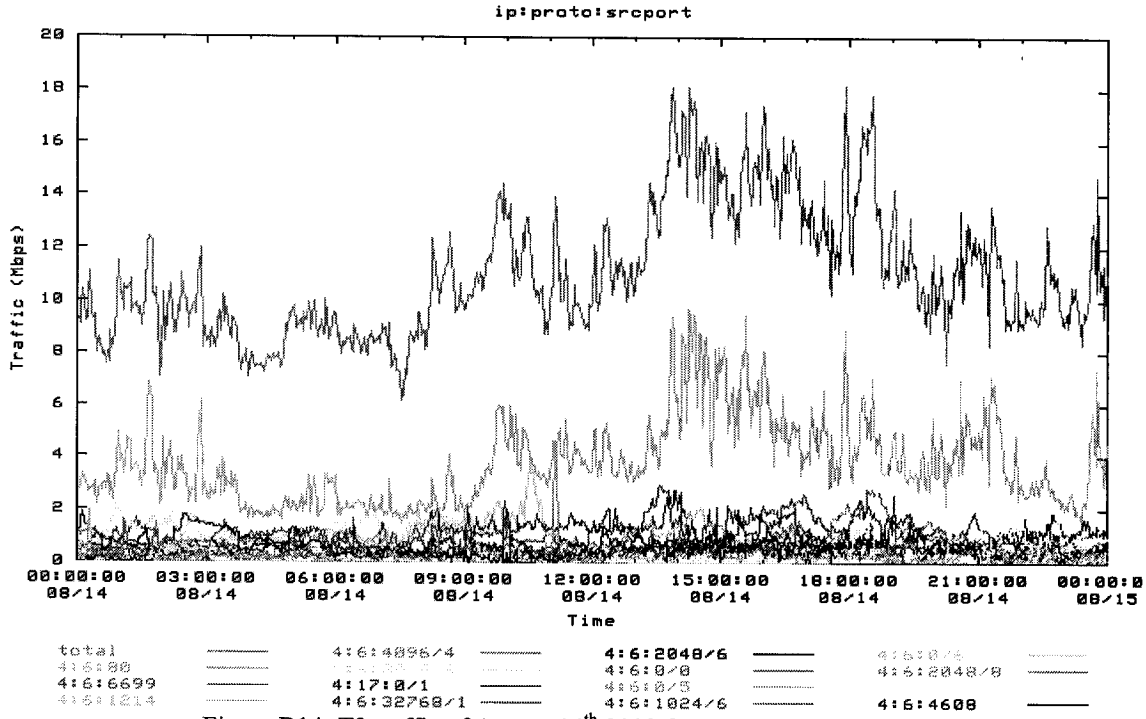


Figure B14: T3 traffic of August 14th 2002 from WIDE network [41]

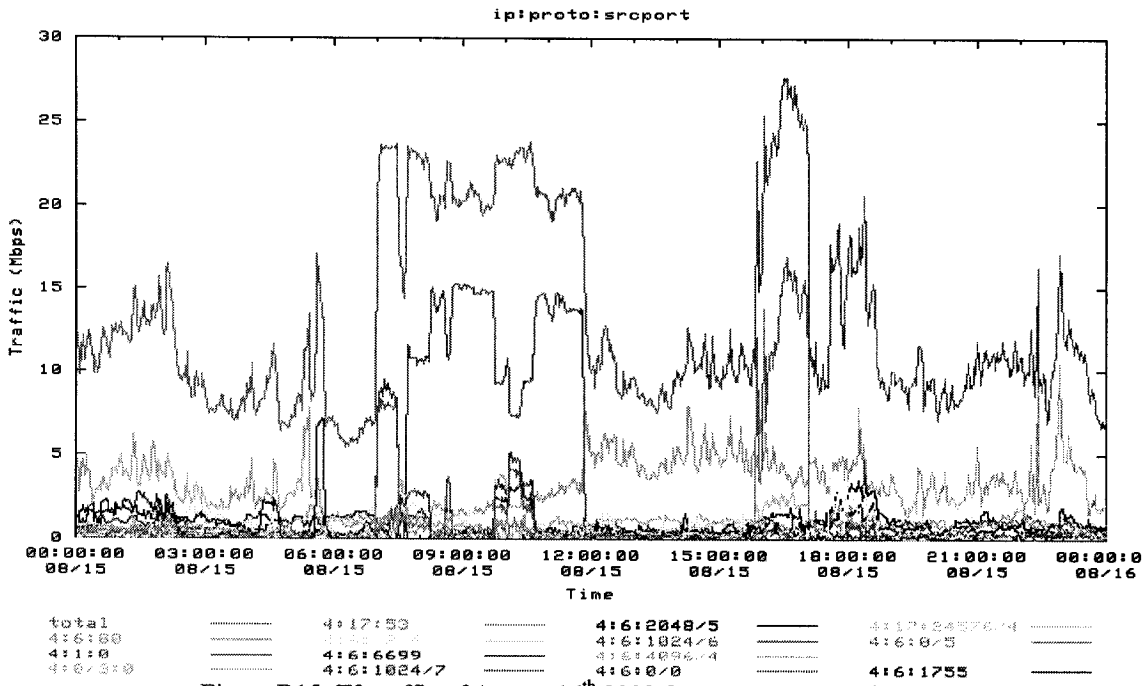


Figure B15: T3 traffic of August 16th 2002 from WIDE network [41]

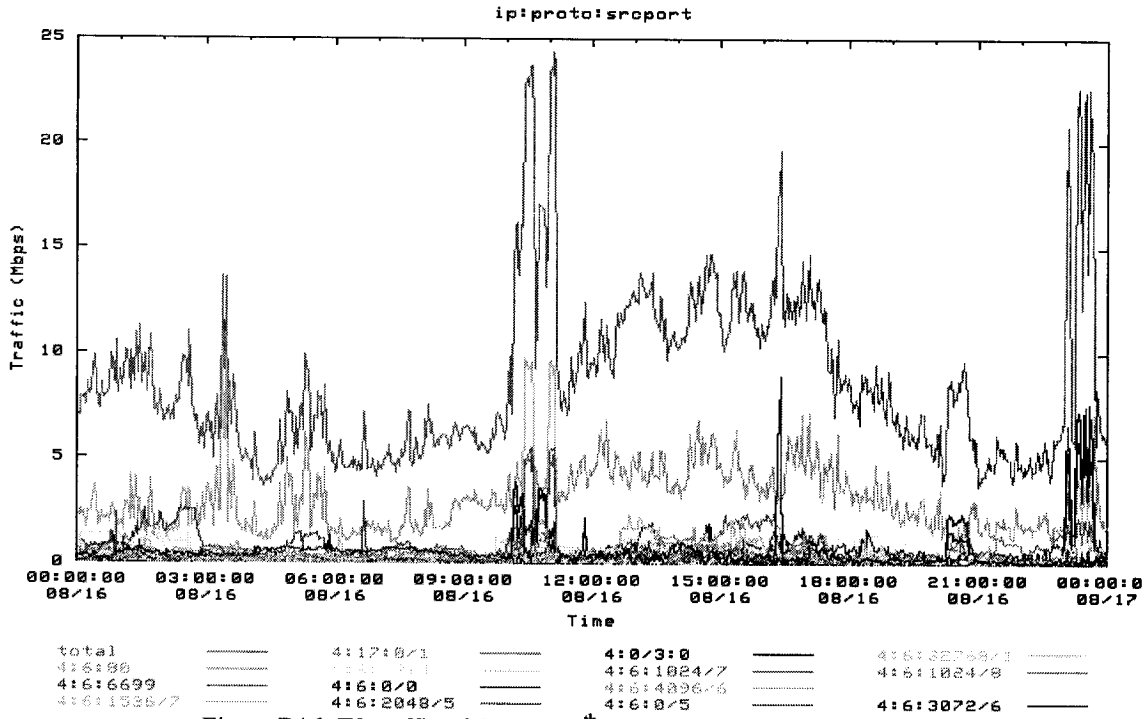


Figure B16: T3 traffic of August 16th 2002 from WIDE network [41]

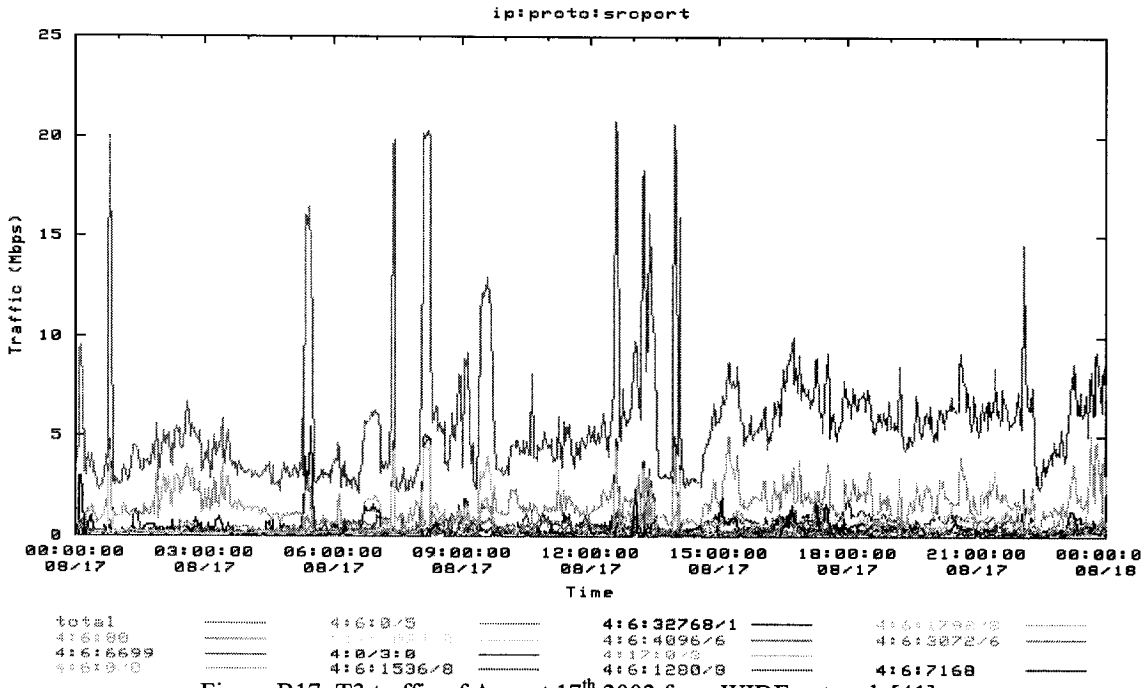


Figure B17: T3 traffic of August 17th 2002 from WIDE network [41]

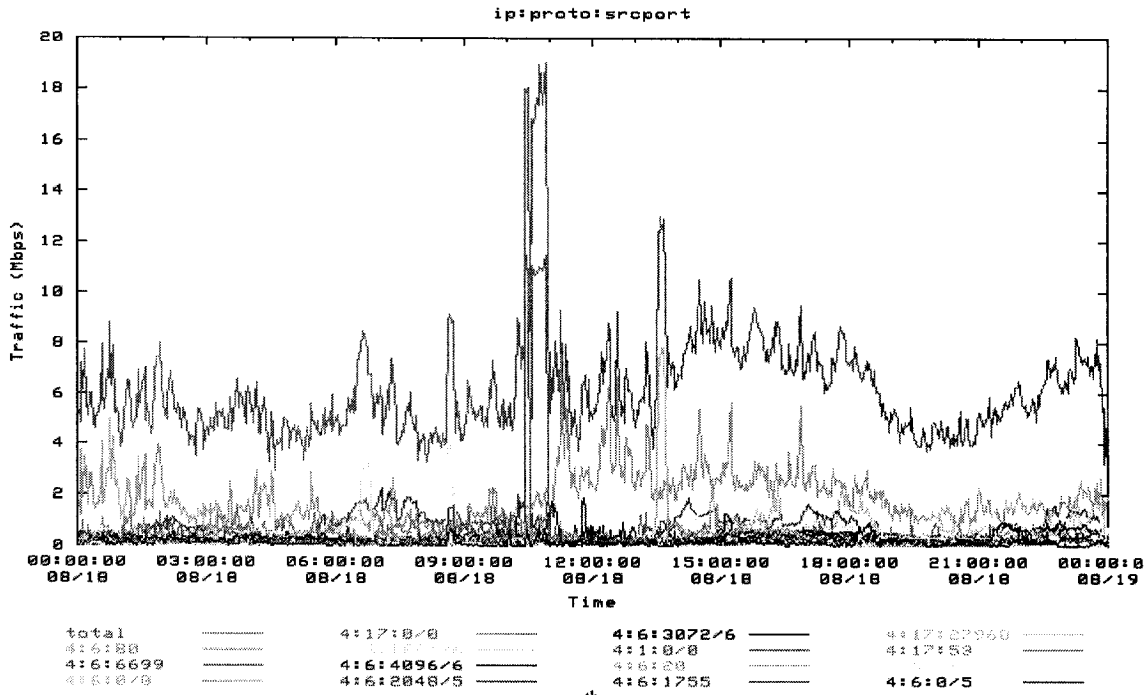


Figure B18: T3 traffic of August 18th 2002 from WIDE network [41]

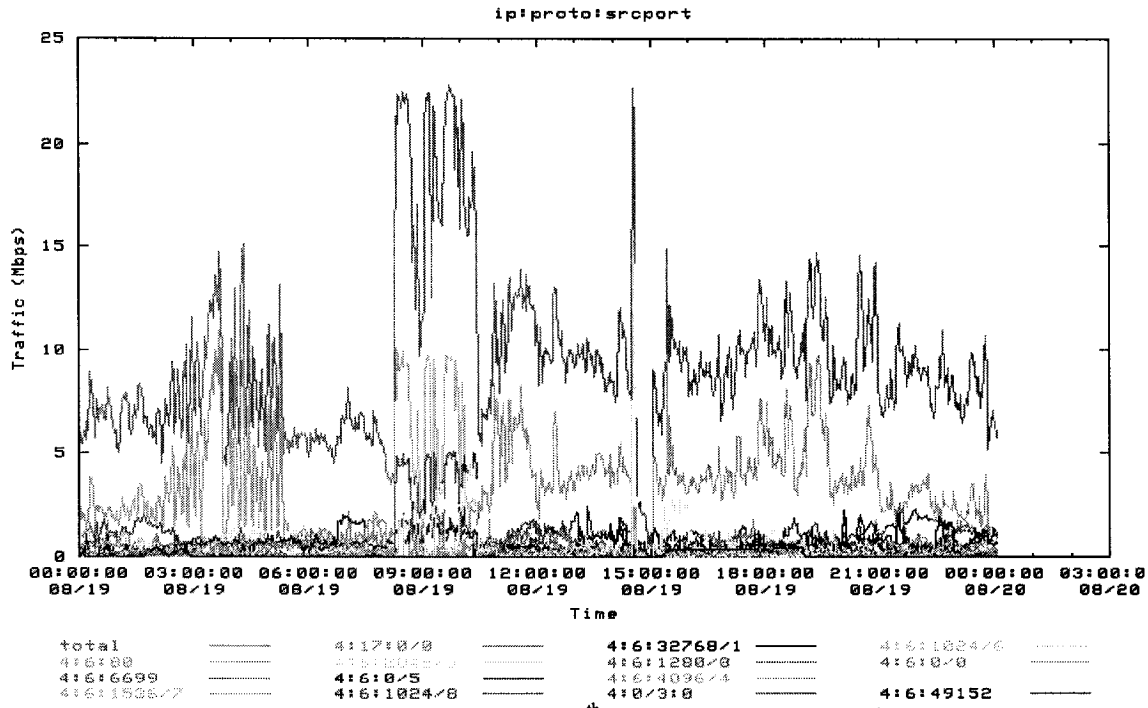


Figure B19: T3 traffic of August 19th 2002 from WIDE network [41]

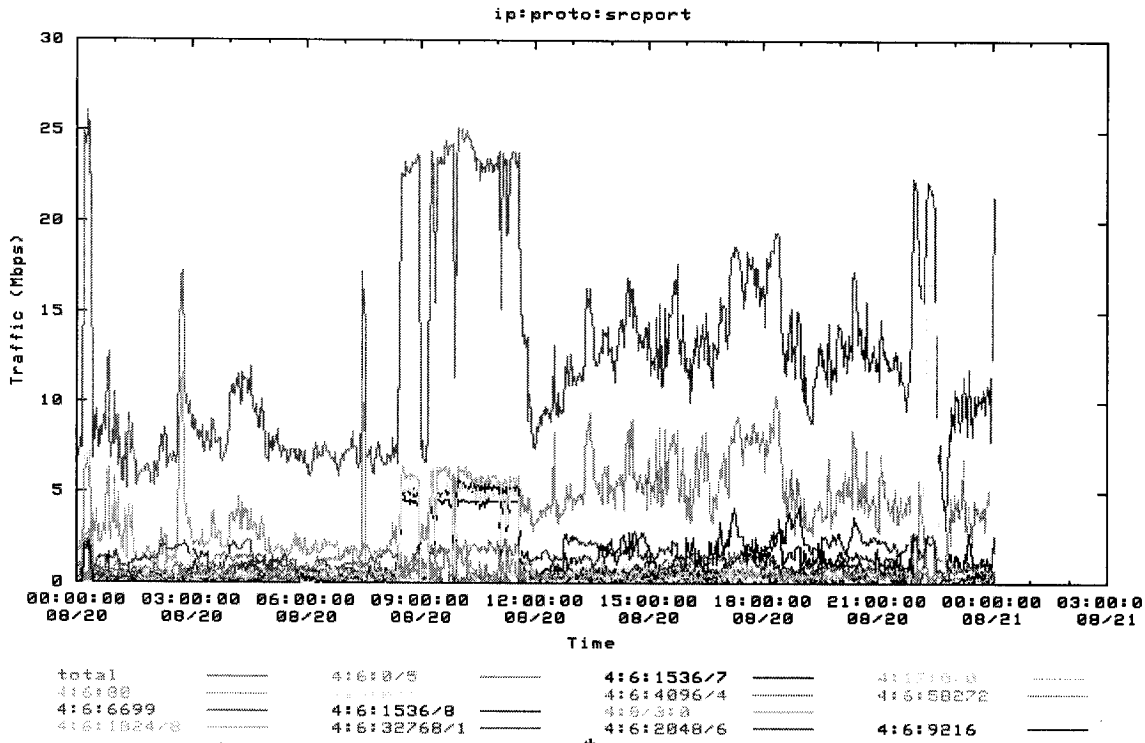


Figure B20: T3 traffic of August 20th 2002 from WIDE network [41]

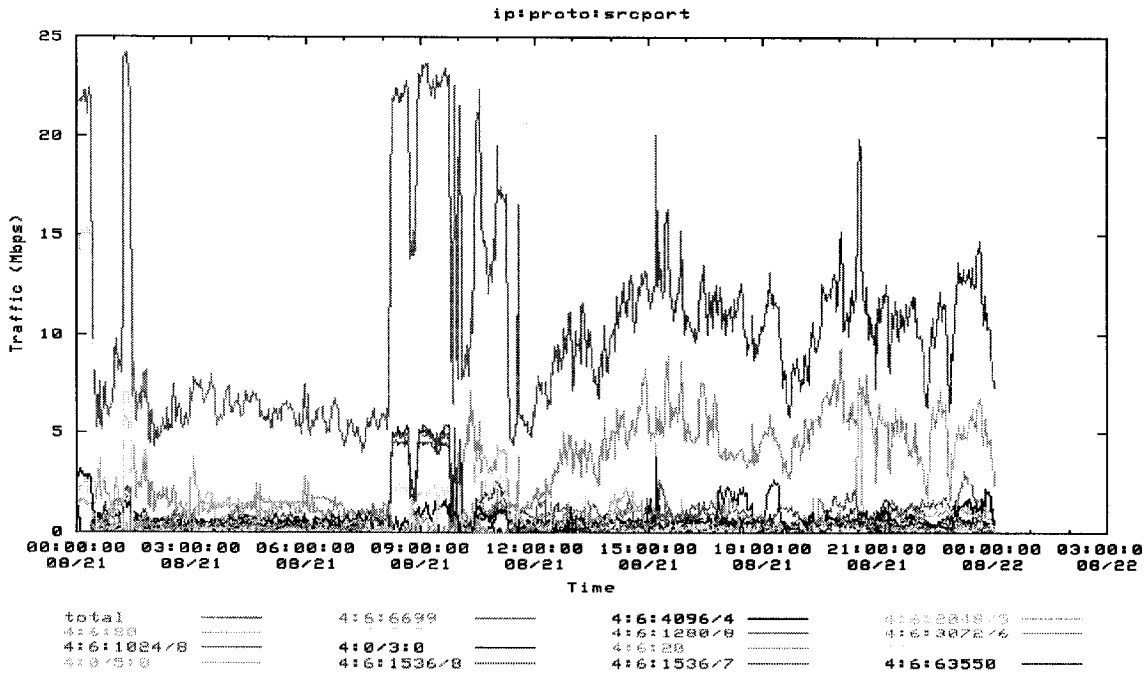


Figure B21: T3 traffic of August 21st 2002 from WIDE network [41]

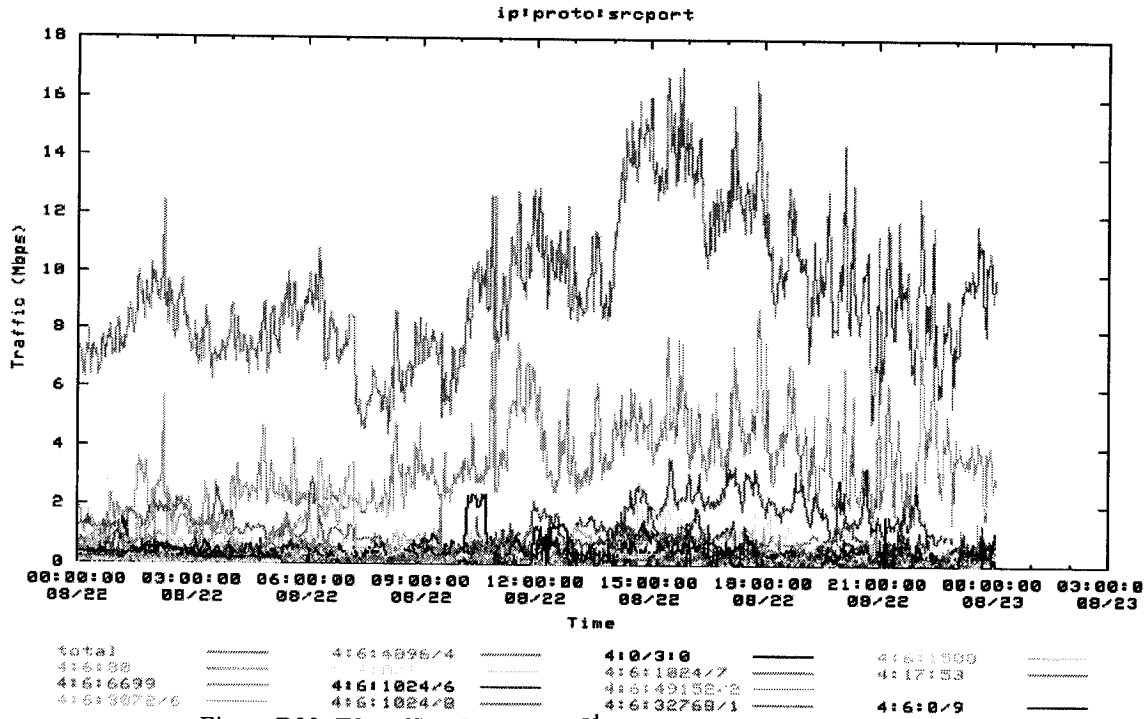
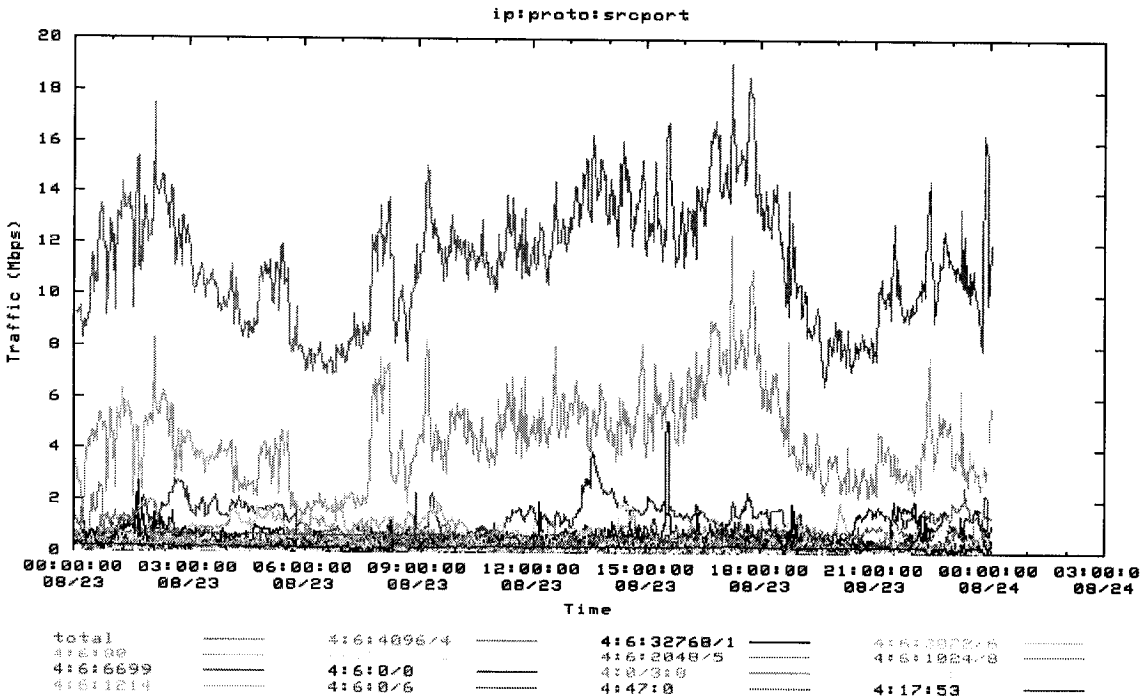


Figure B22: T3 traffic of August 22nd 2002 from WIDE network [41]



ig ure B23: T3 traffic of August 23rd 2002 from WIDE network [41]

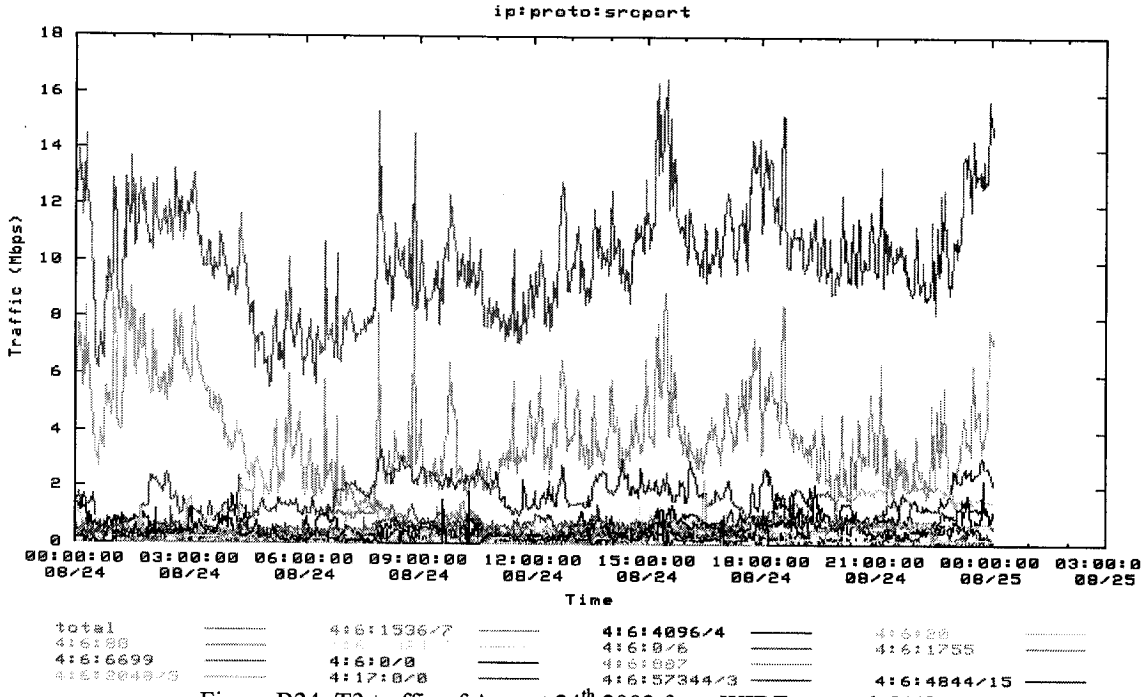


Figure B24: T3 traffic of August 24th 2002 from WIDE network [41]

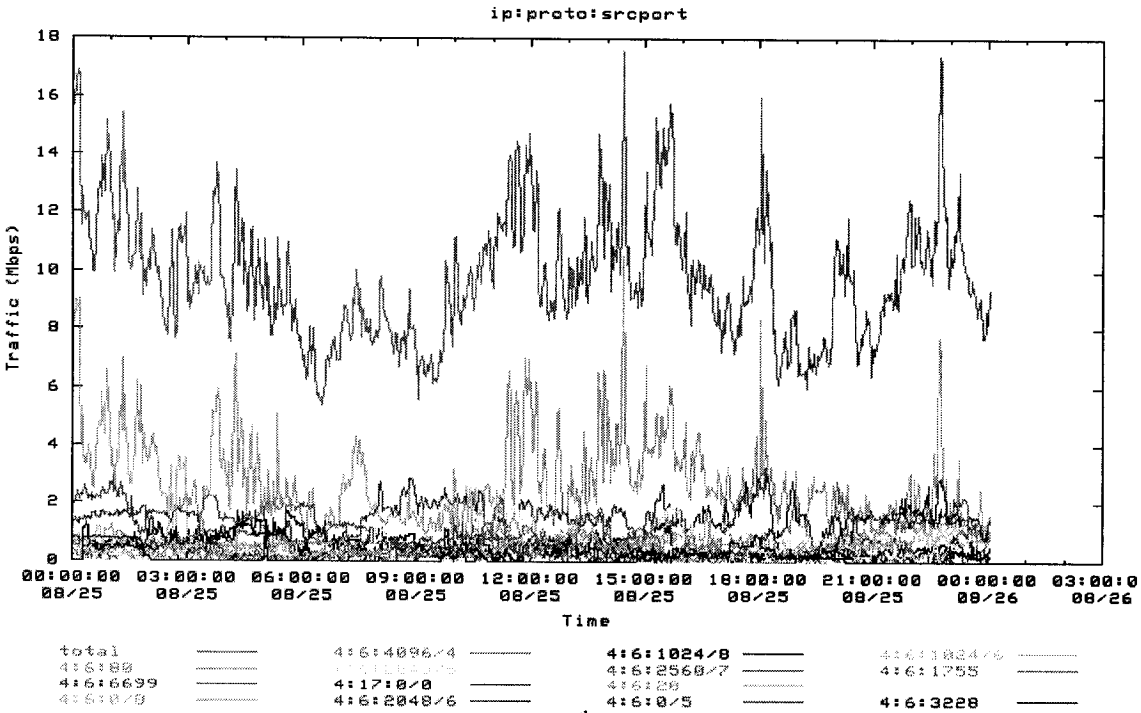


Figure B25: T3 traffic of August 25th 2002 from WIDE network [41]

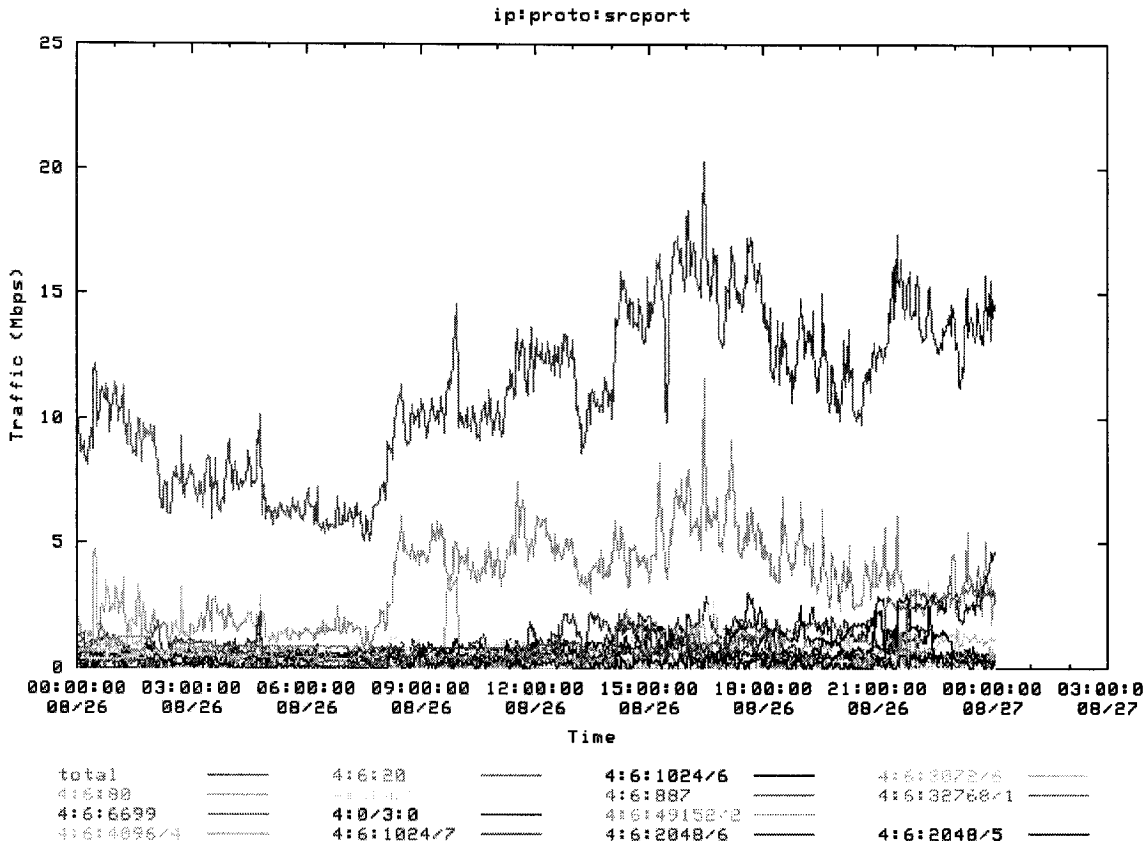


Figure B26: T3 traffic of August 26th 2002 from WIDE network [41]

Appendix C: Monthly Traffic Report in Year 2002

The monthly traffic report shows a whole picture of the traffic in a year period. Thus we can find the peak rate and ensure that the traffic data chosen from August can represent the link status in a year. Section C.1 gives the monthly traffic of OC3 link and Section C.2 gives the data of T3 links.

C.1 Monthly Traffic Report of OC3 link

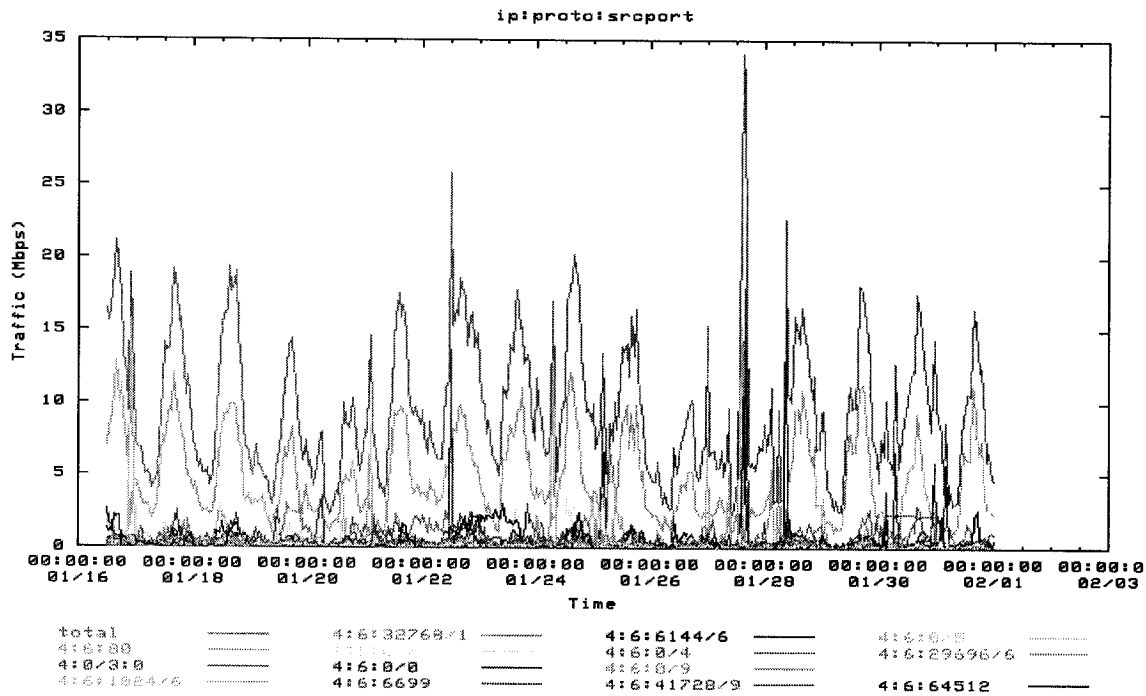


Figure C1: OC3 traffic of January 2002 from WIDE network [40]

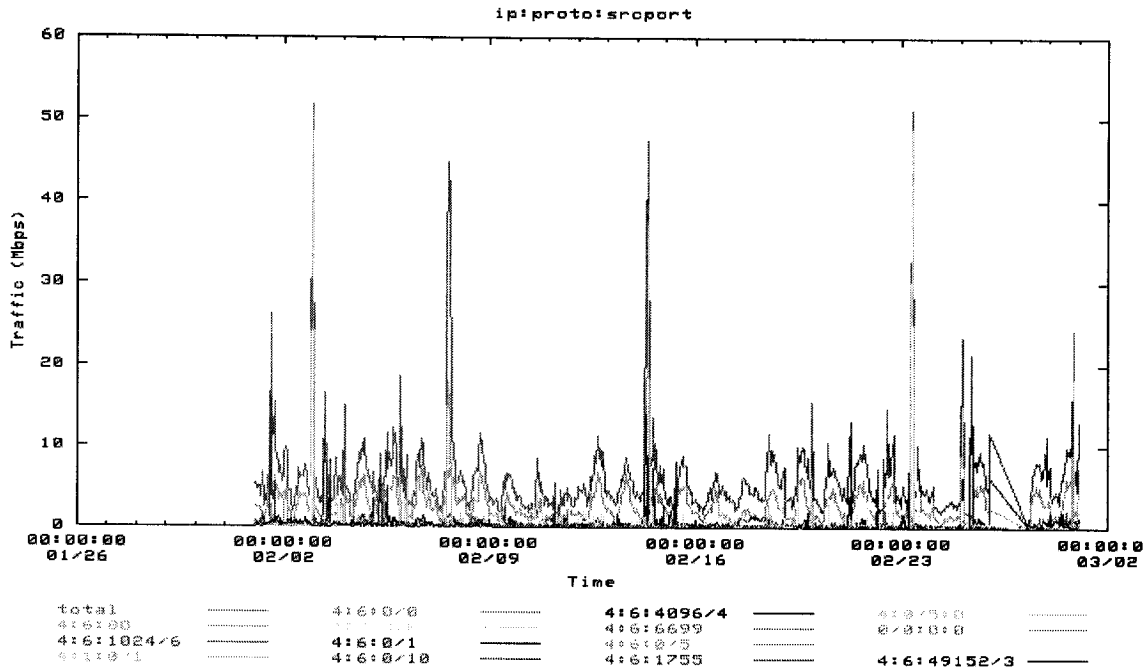


Figure C2: OC3 traffic of February 2002 from WIDE network [40]

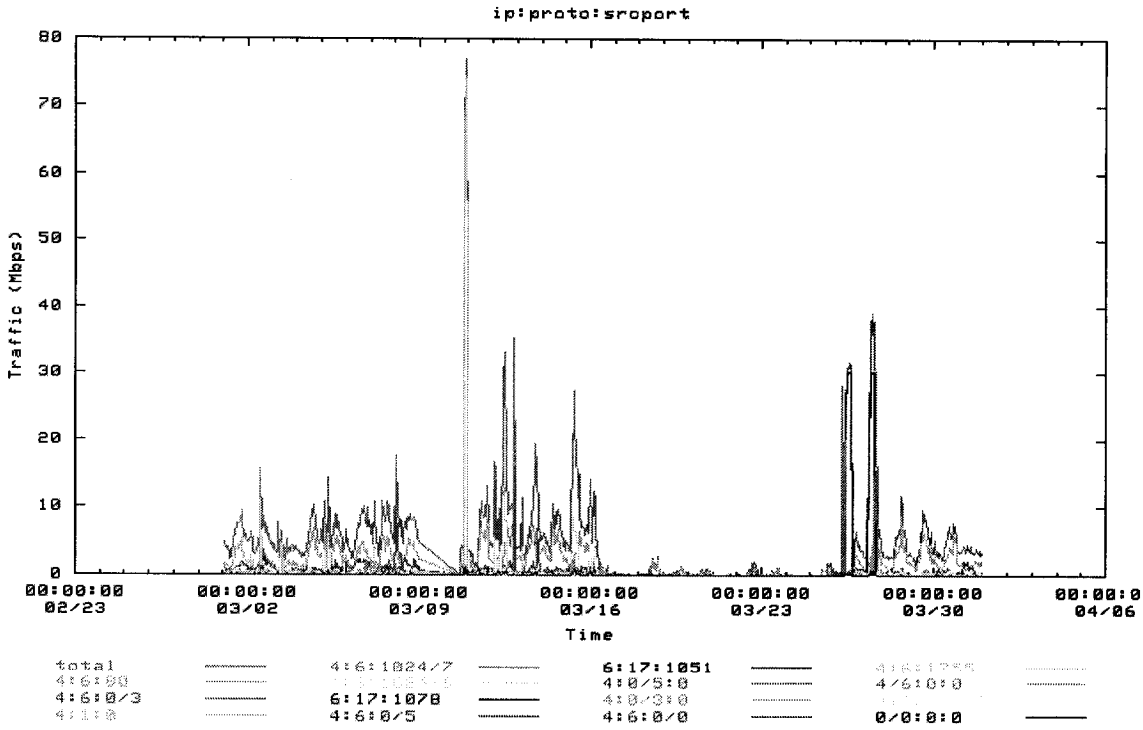


Figure C3: OC3 traffic of March 2002 from WIDE network [40]

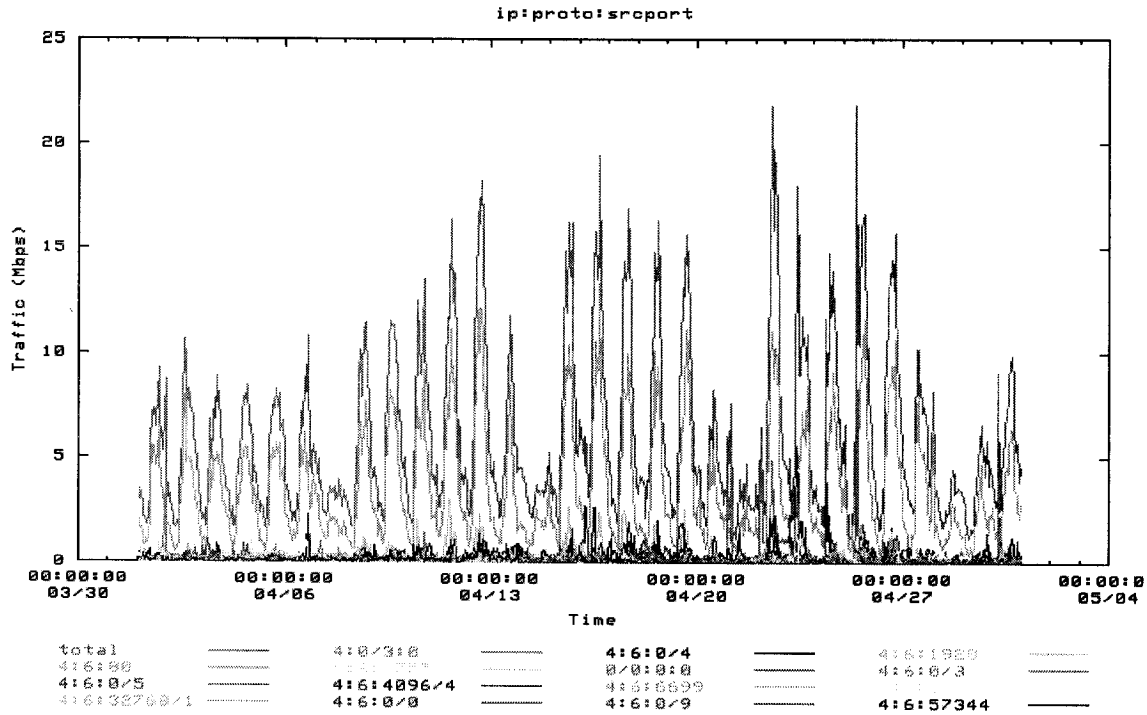


Figure C4: OC3 traffic of April 2002 from WIDE network [40]

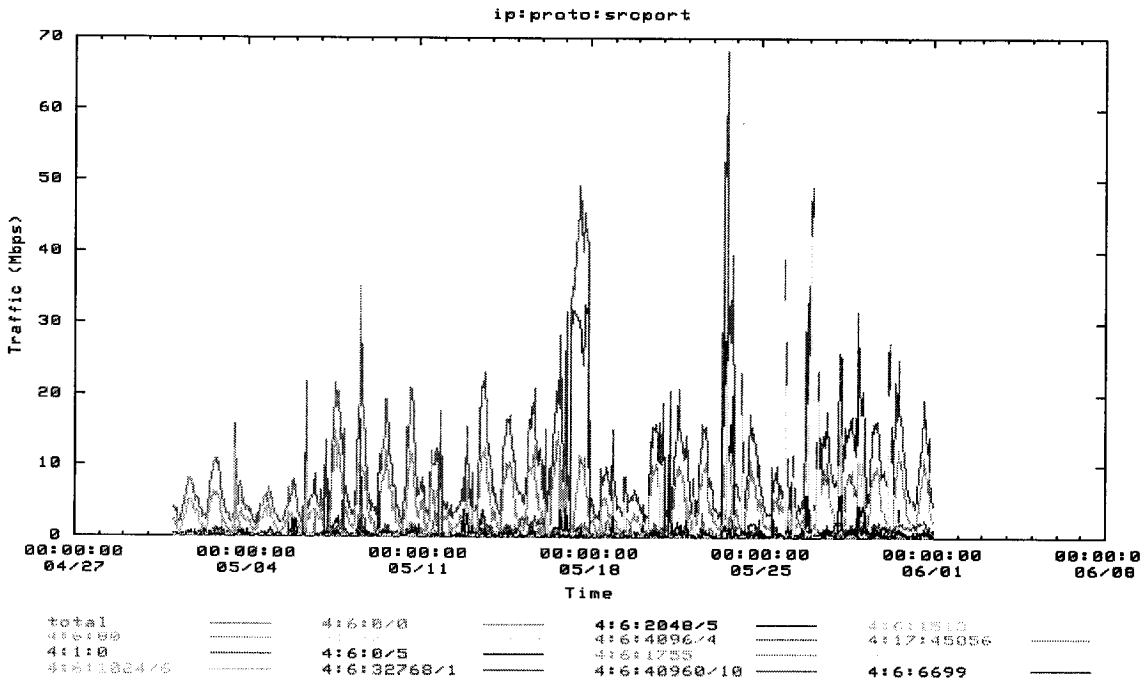


Figure C5: OC3 traffic of May 2002 from WIDE network [40]

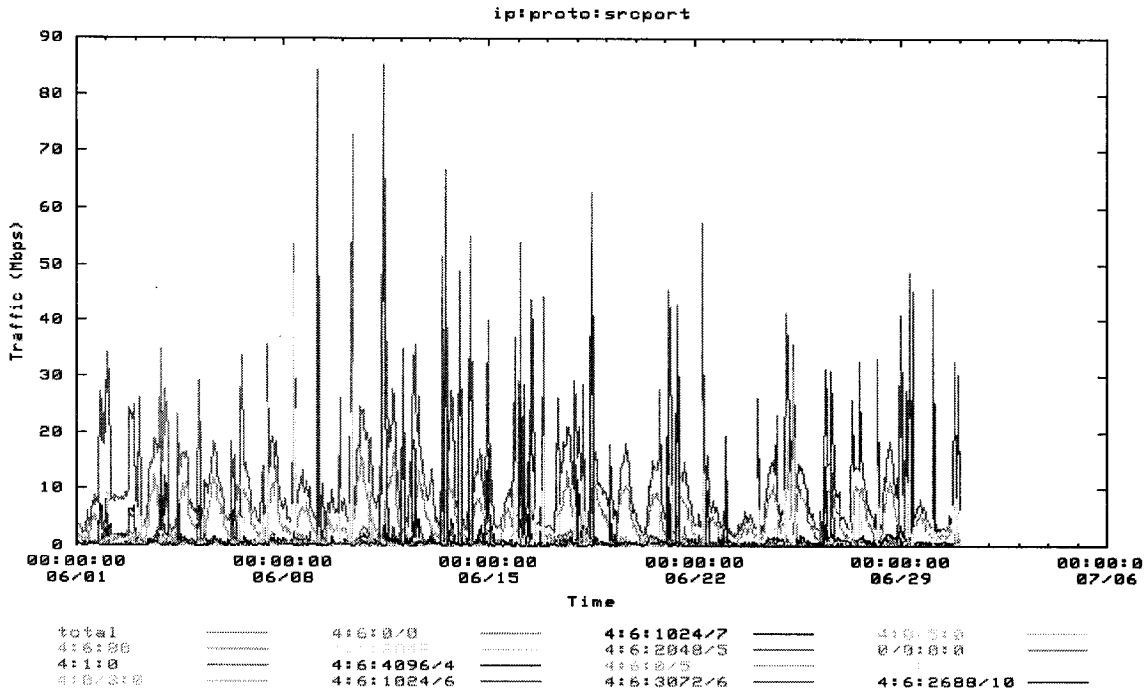


Figure C6: OC3 traffic of June 2002 from WIDE network [40]

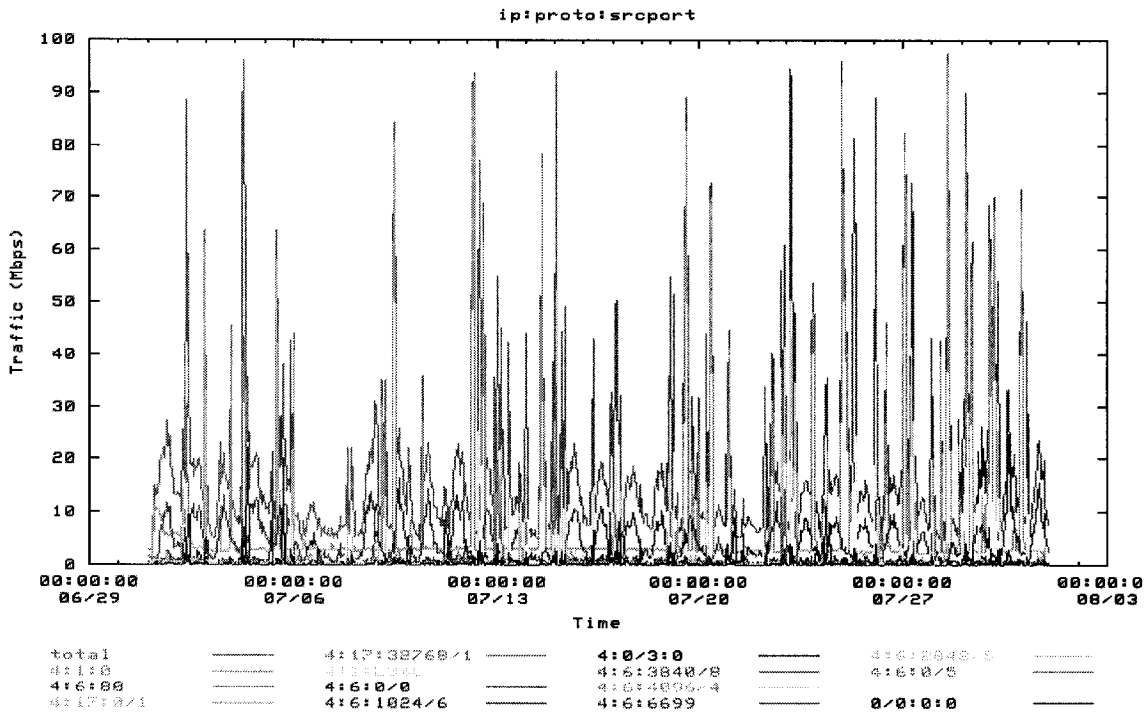


Figure C7: OC3 traffic of July 2002 from WIDE network [40]

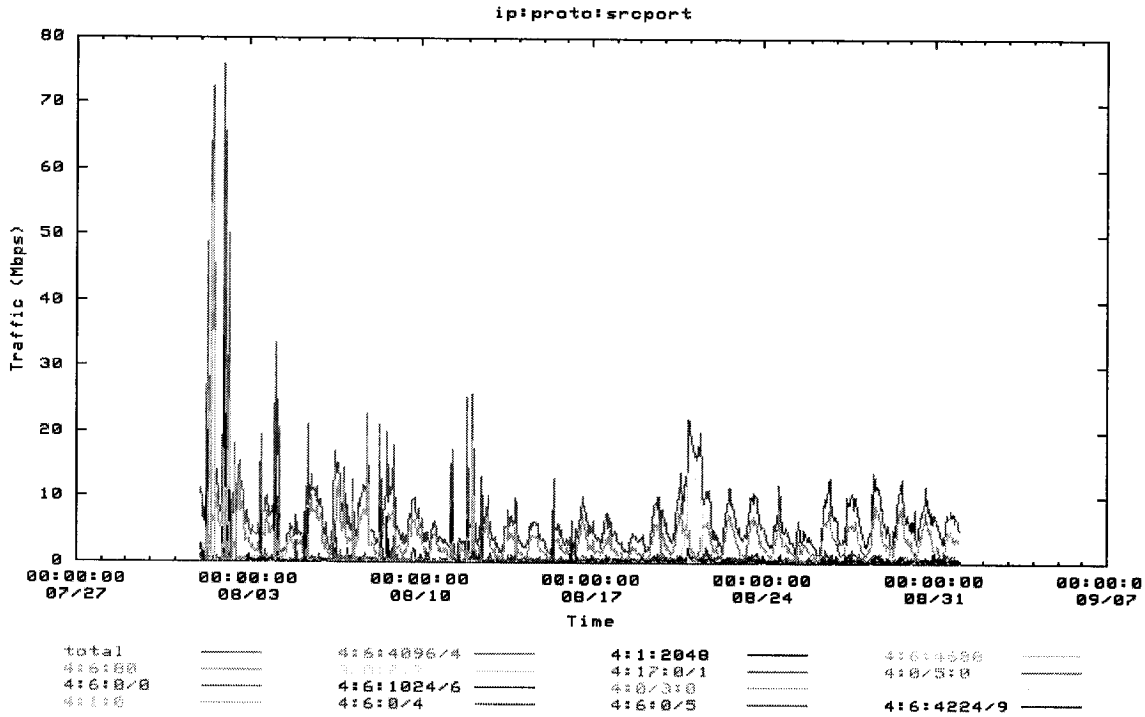


Figure C8: OC3 traffic of August 2002 from WIDE network [40]

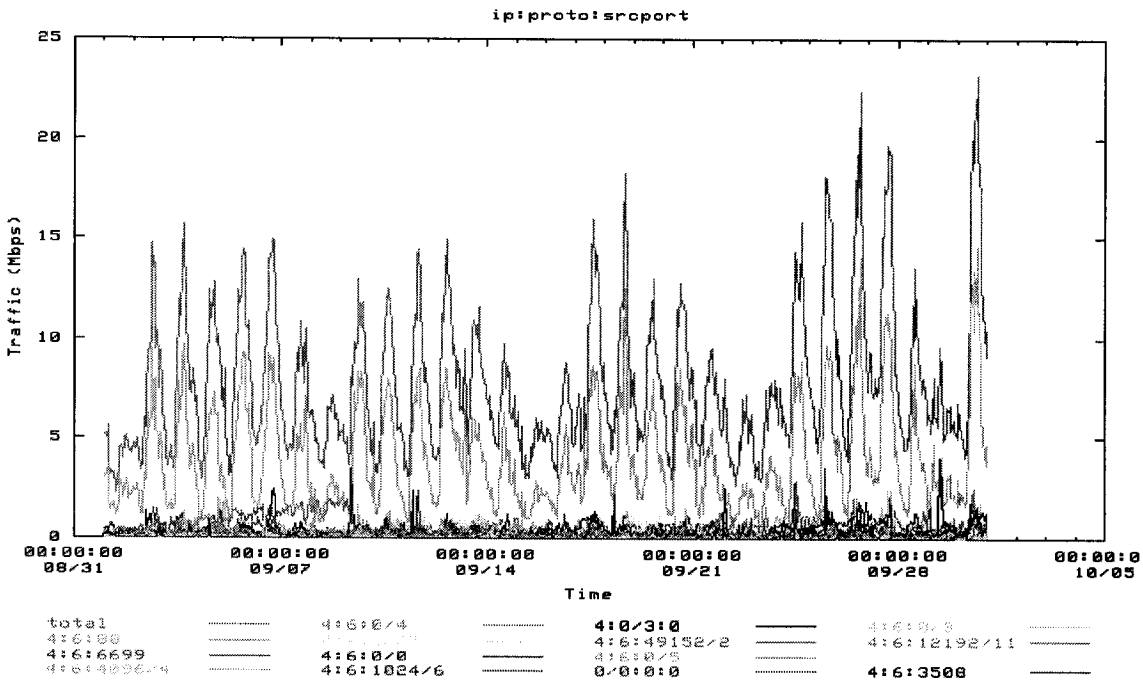


Figure C9: OC3 traffic of September 2002 from WIDE network [40]

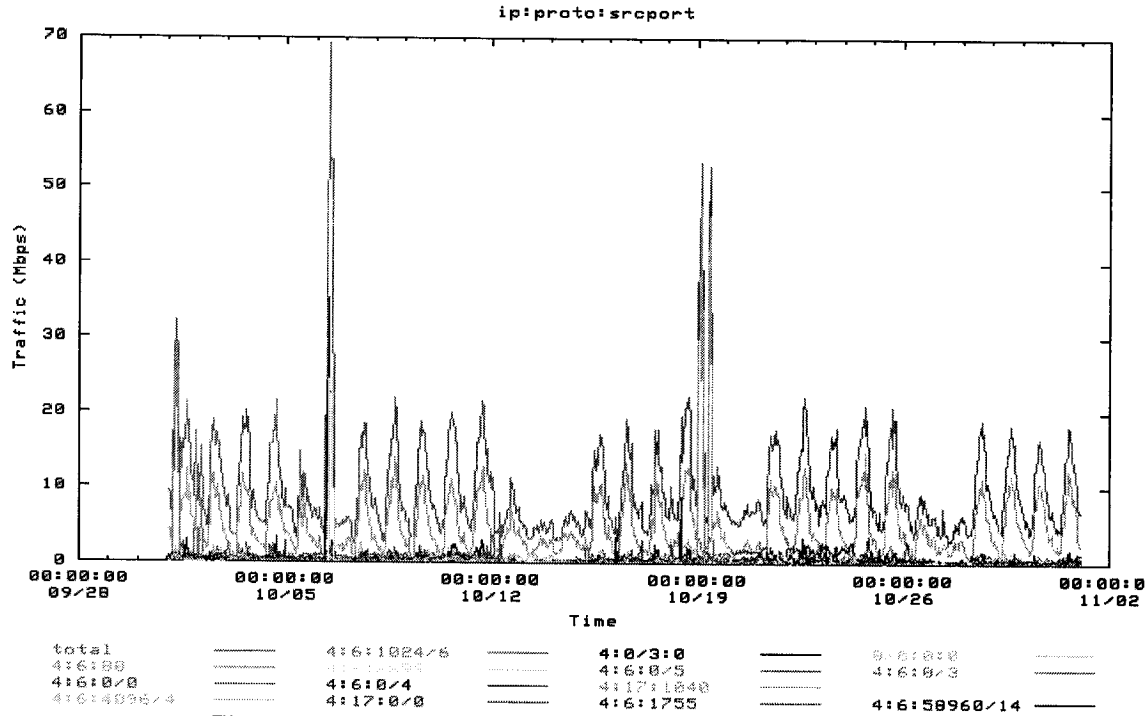


Figure C10: OC3 traffic of October 2002 from WIDE network [40]

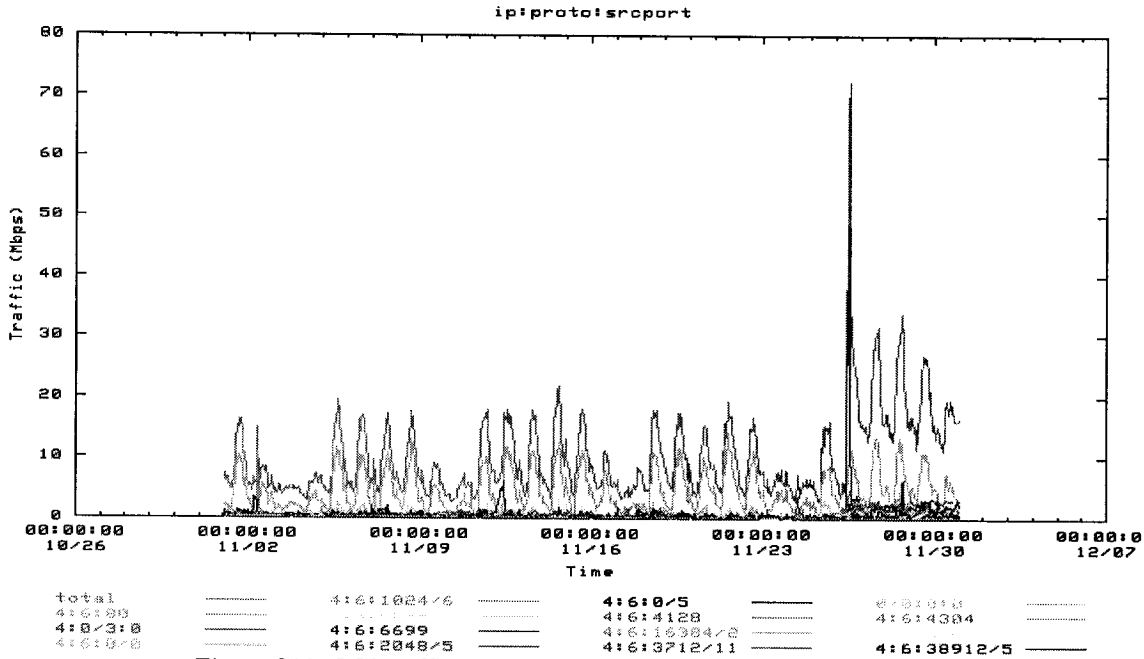


Figure C11: OC3 traffic of November 2002 from WIDE network [40]

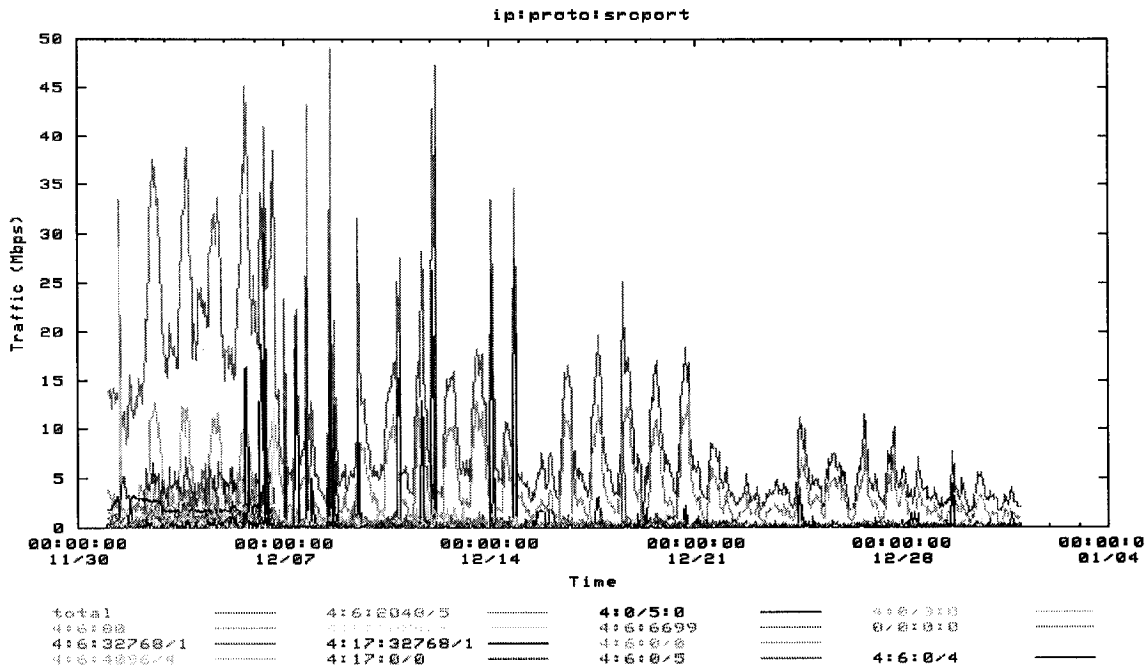


Figure C12: OC3 traffic of December 2002 from WIDE network [40]

C.2 Monthly Traffic Report of T3 link

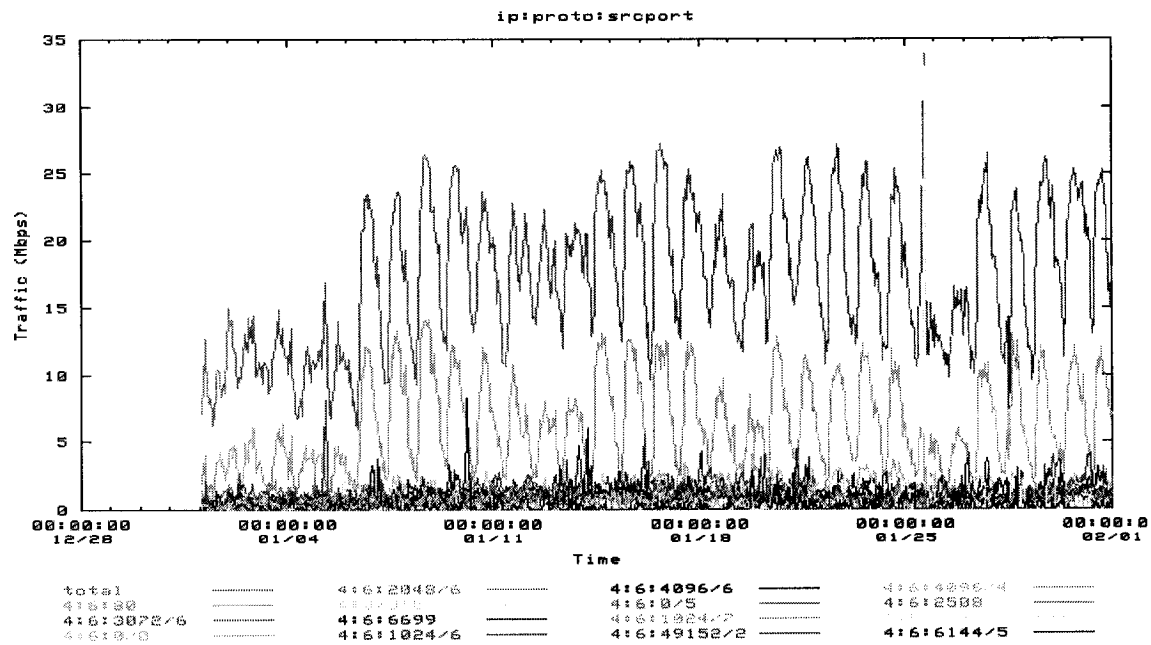


Figure C13: T3 traffic of January 2002 from WIDE network [41]

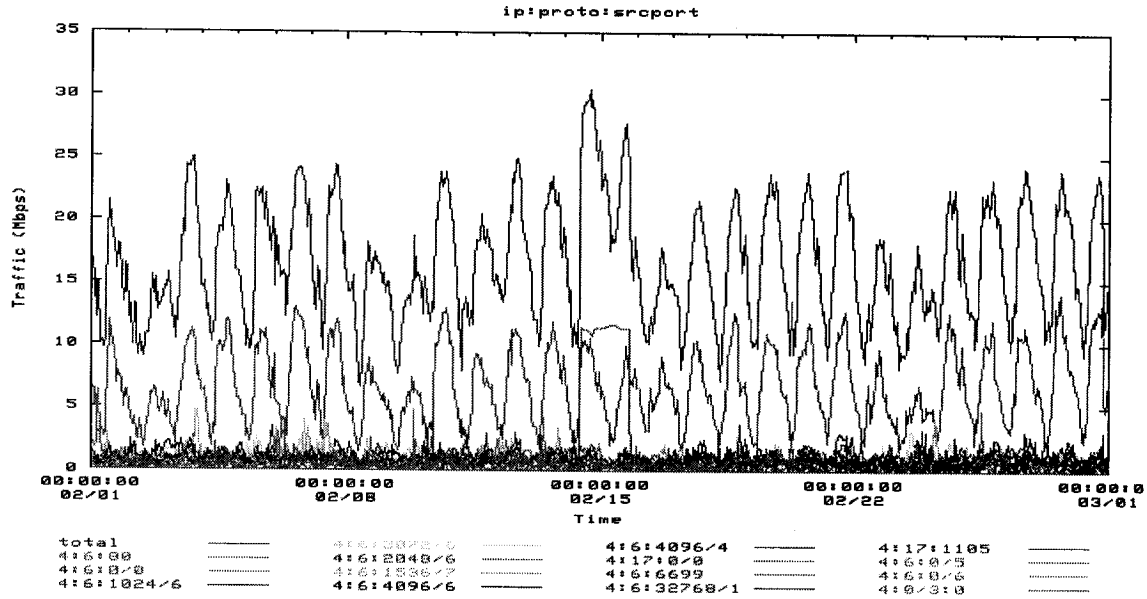


Figure C14: T3 traffic of February 2002 from WIDE network [41]

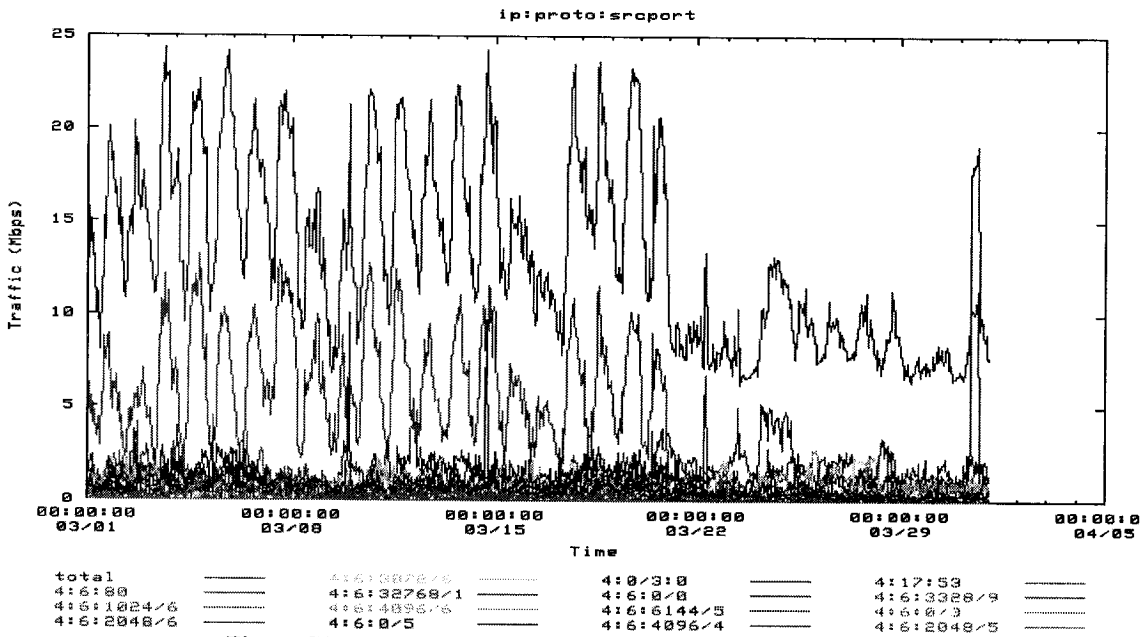


Figure C15: T3 traffic of March 2002 from WIDE network [41]

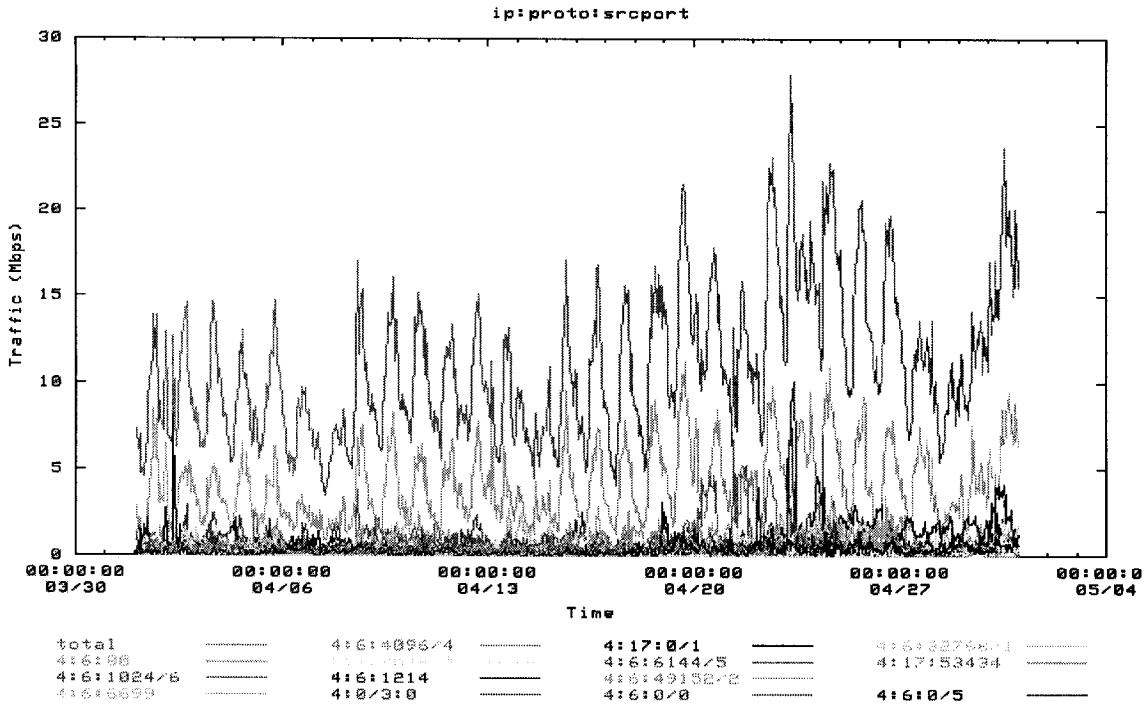


Figure C16: T3 traffic of April 2002 from WIDE network [41]

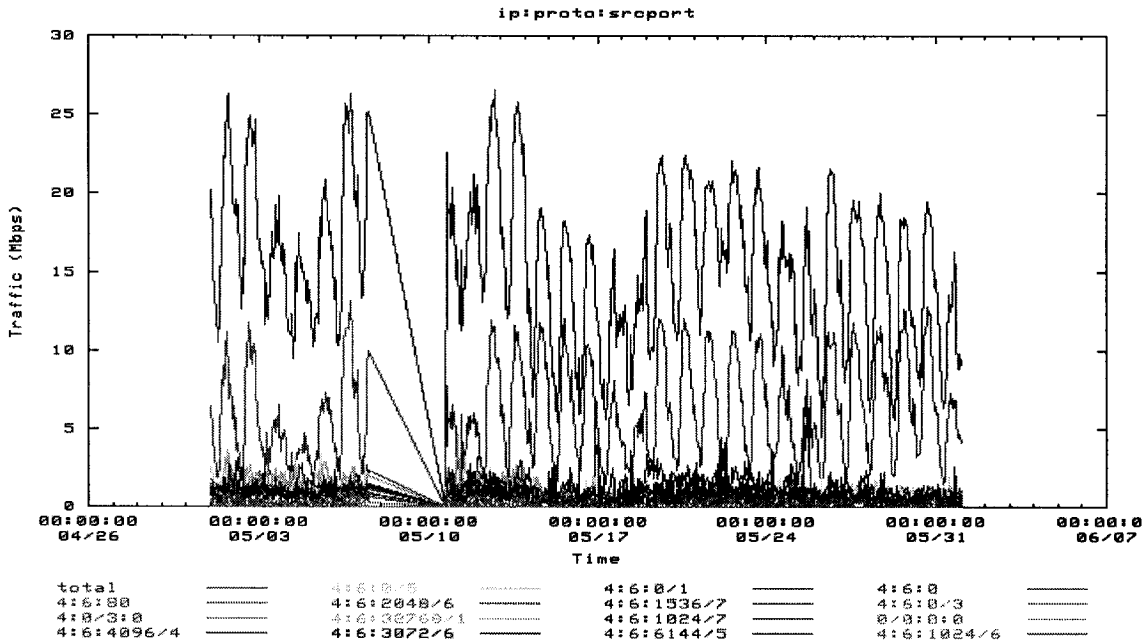


Figure C17: T3 traffic of May 2002 from WIDE network [41]

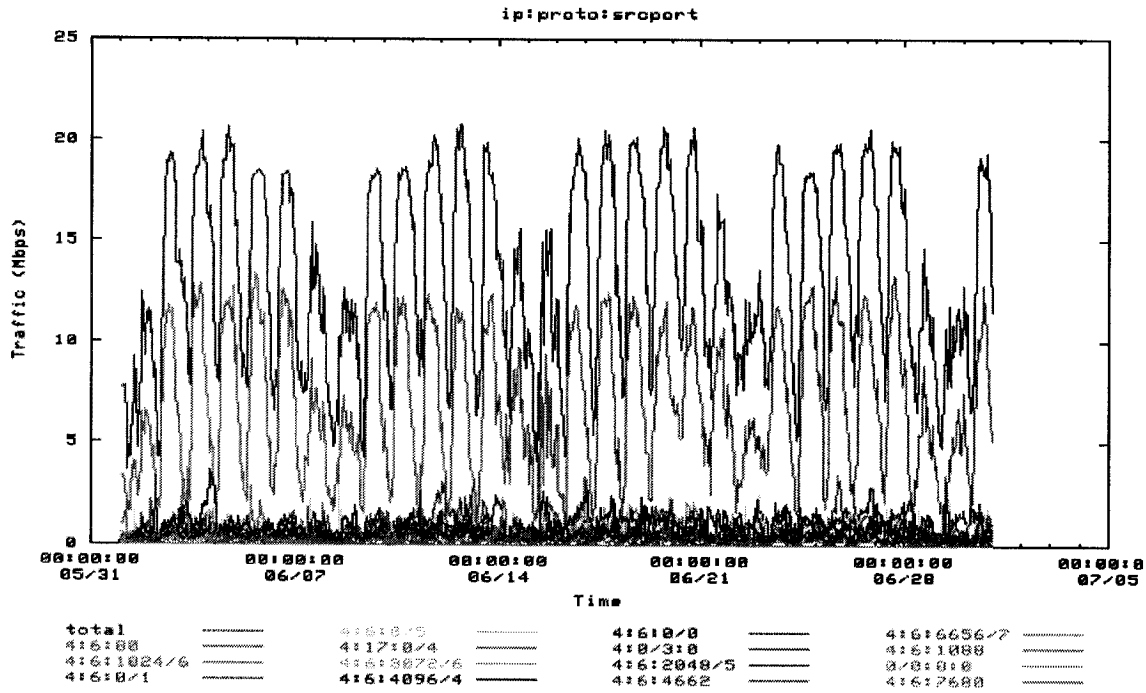


Figure C18: T3 traffic of June 2002 from WIDE network [41]

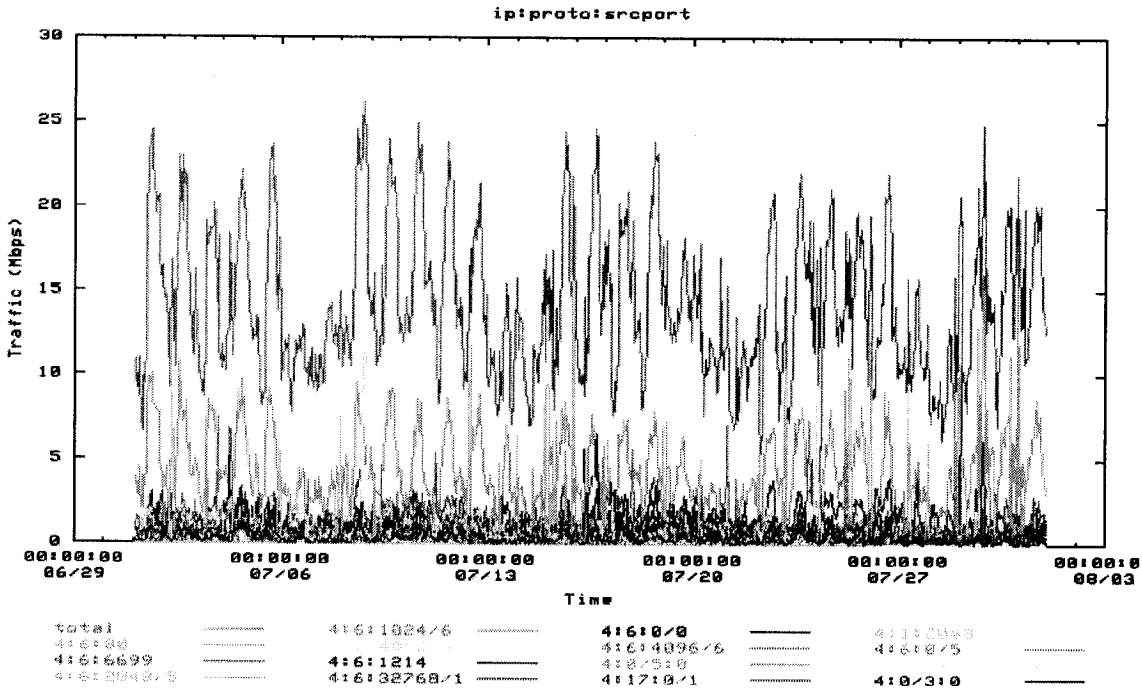


Figure C19: T3 traffic of July 2002 from WIDE network [41]

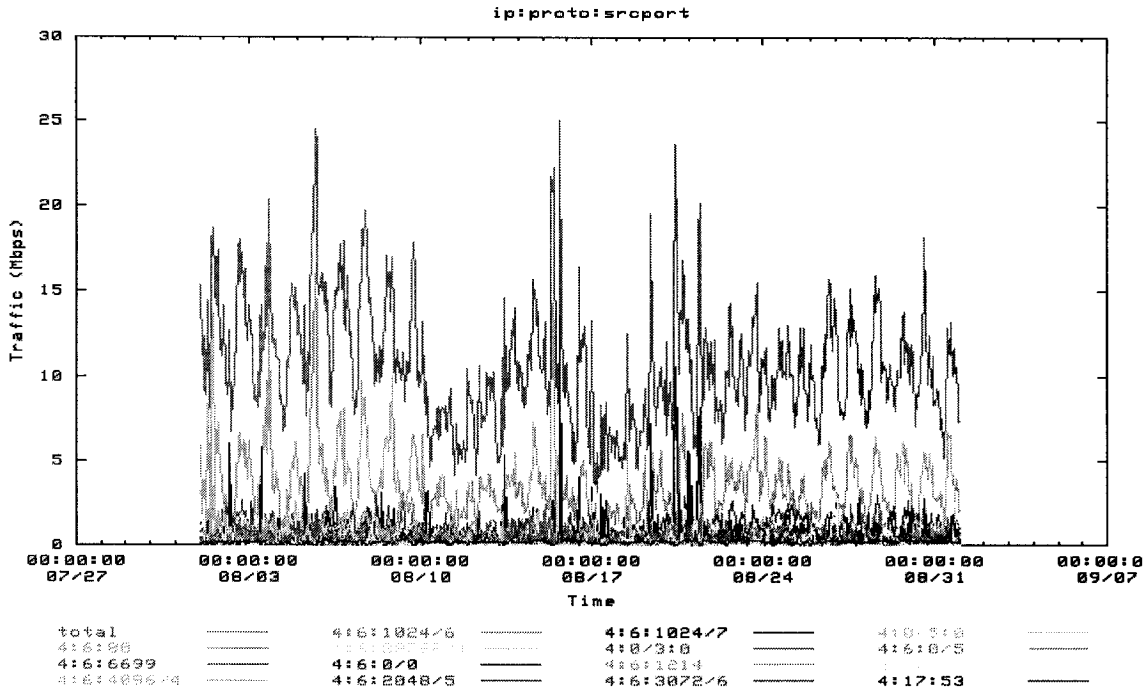


Figure C20: T3 traffic of August 2002 from WIDE network [41]

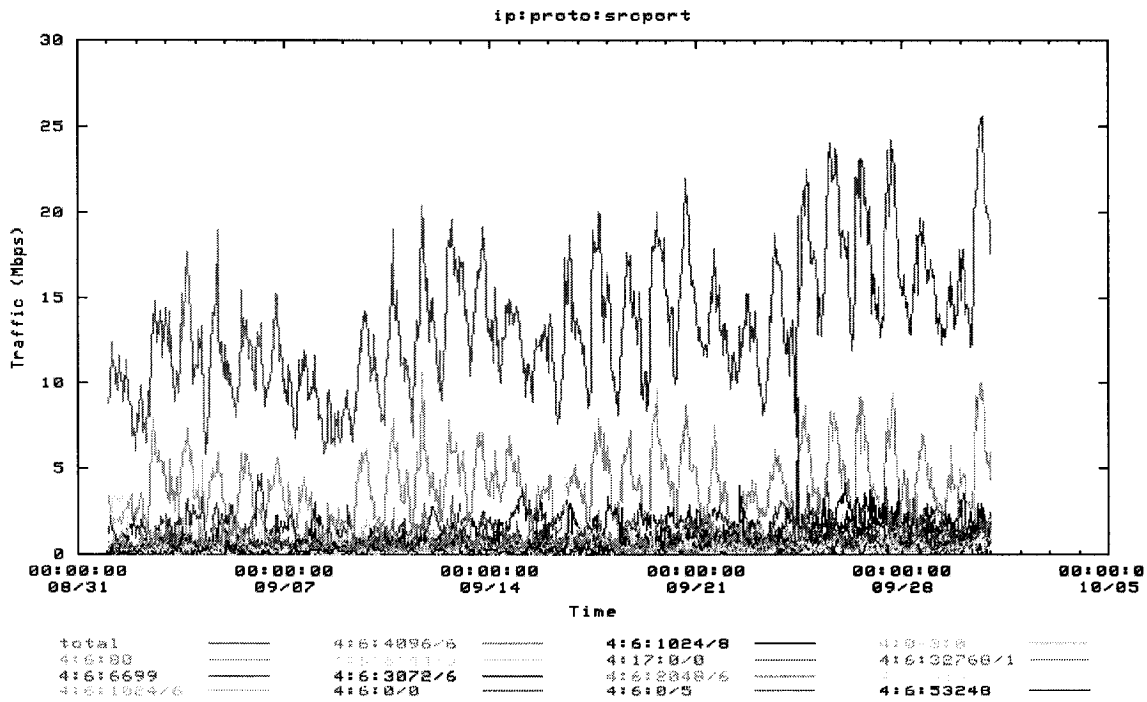


Figure C21: T3 traffic of September 2002 from WIDE network [41]

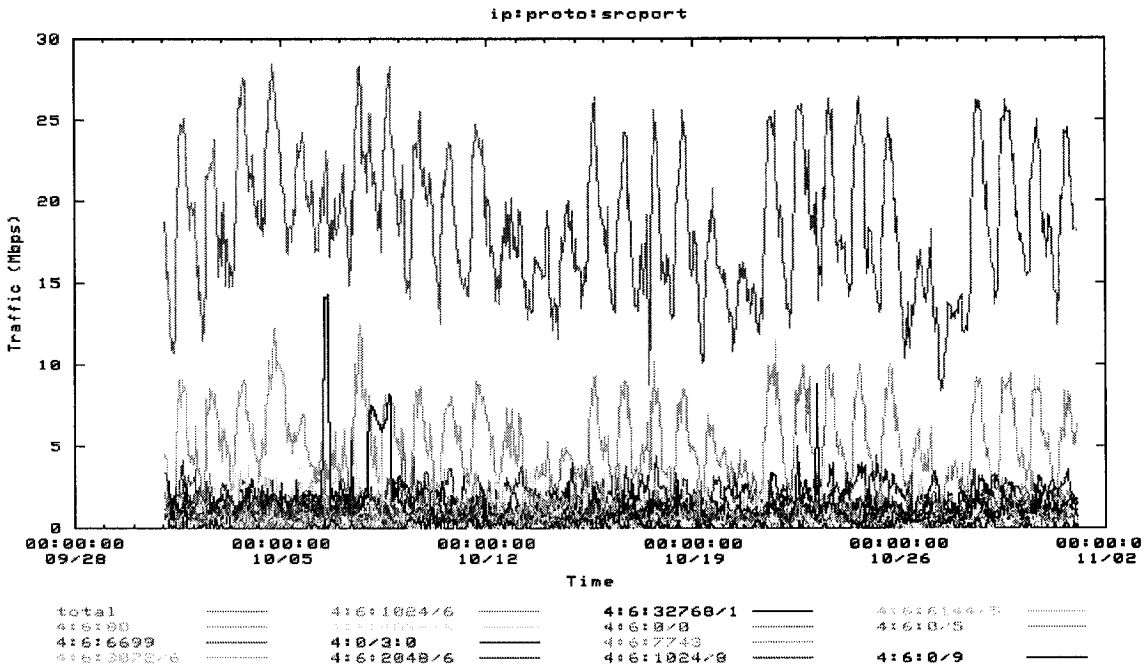


Figure C22: T3 traffic of October 2002 from WIDE network [41]

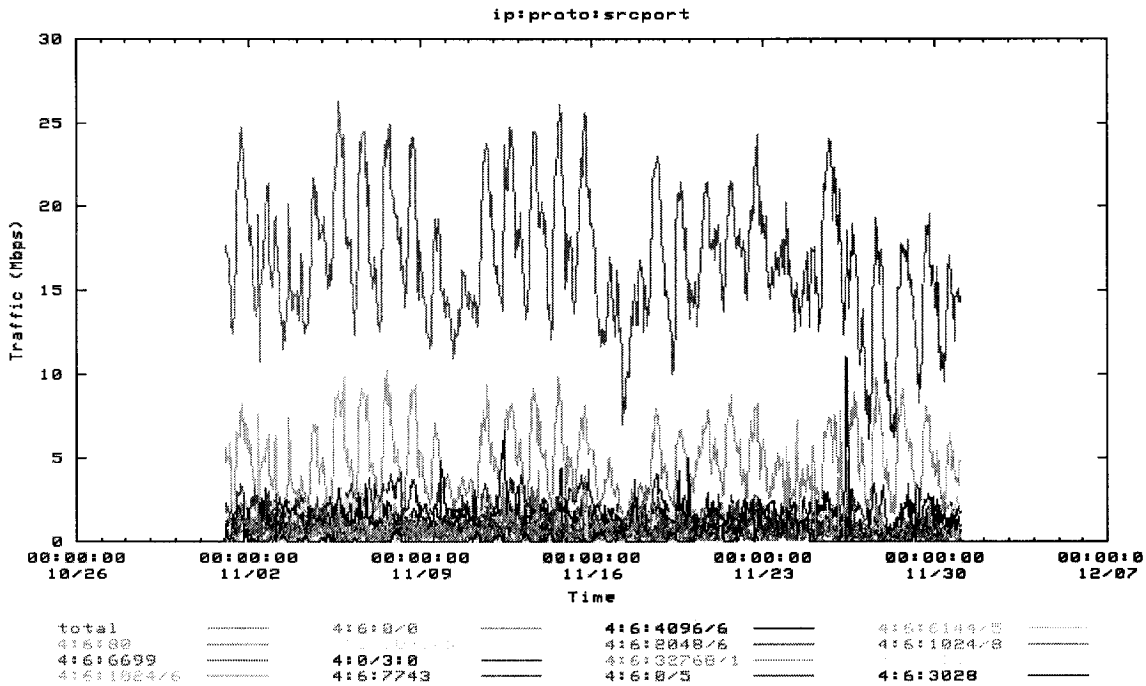


Figure C23: T3 traffic of November 2002 from WIDE network [41]

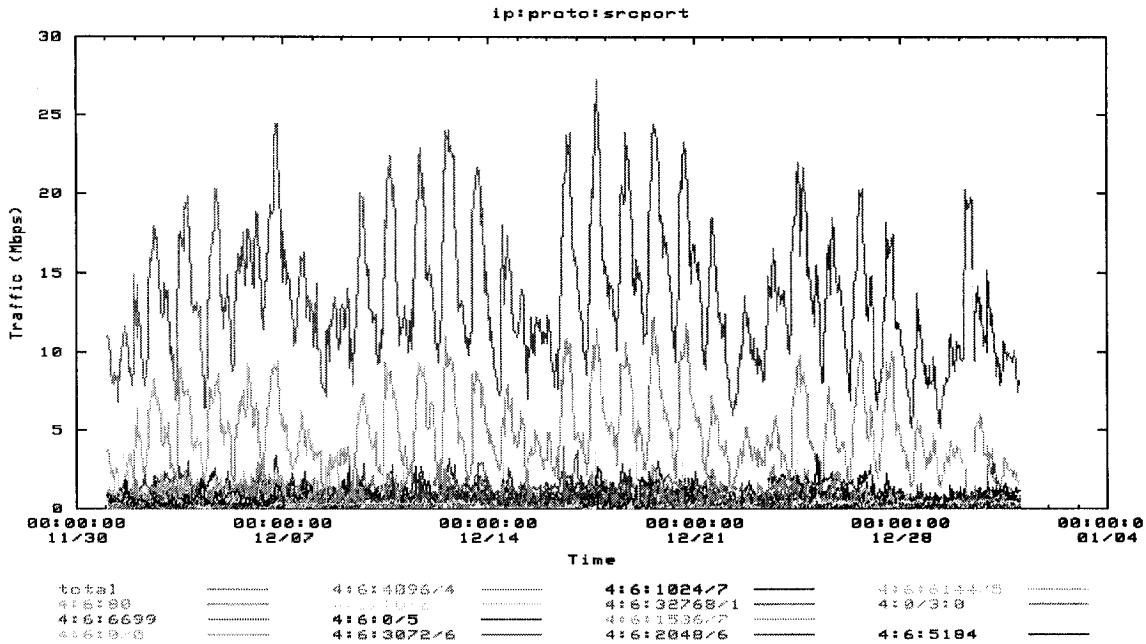


Figure C24: T3 traffic of December 2002 from WIDE network [41]

Appendix D Active Network Status

The following graphs show the active network status when the VPN requests apply to each simulation case in the three network topologies. Every two figures on each link are the bi-directional traffic data united by Mbps.

D.1 Network Topology 1 Status

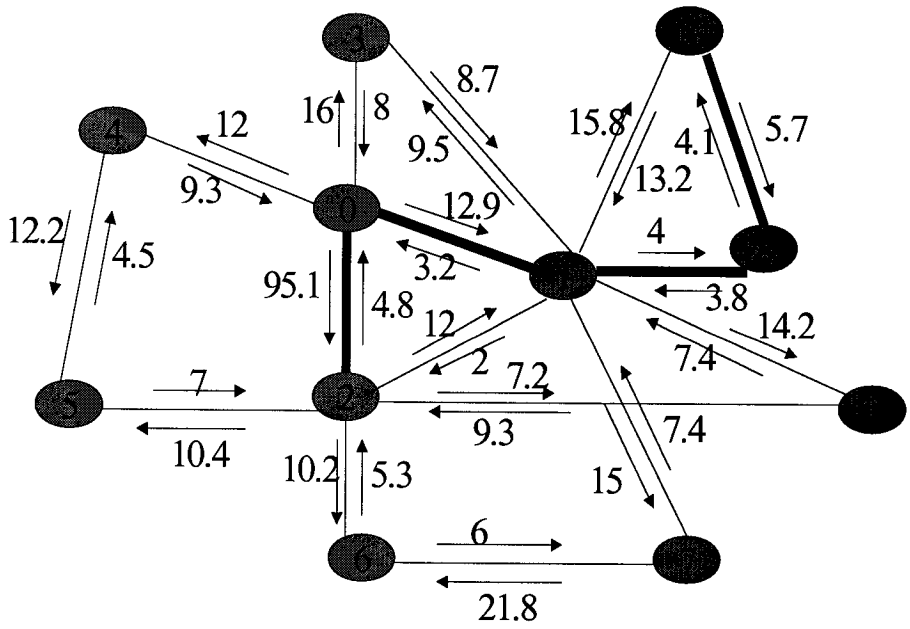


Figure D.1.1 Network status of Network Topology 1 Case 1 0:00

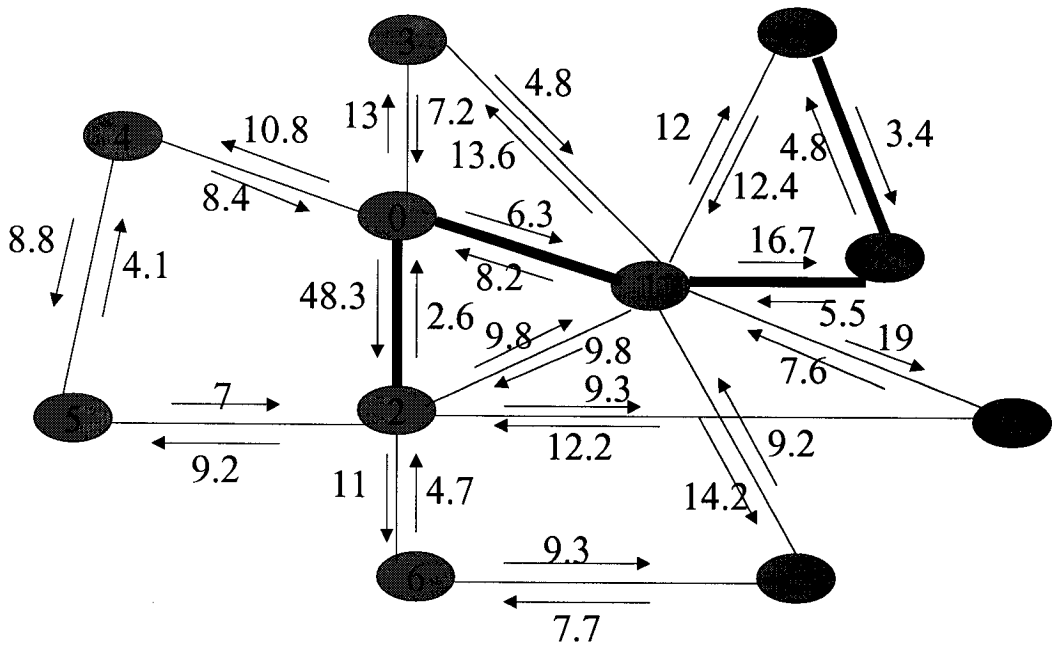


Figure D.1.2 Network status of Network Topology 1 Case 2 3:00

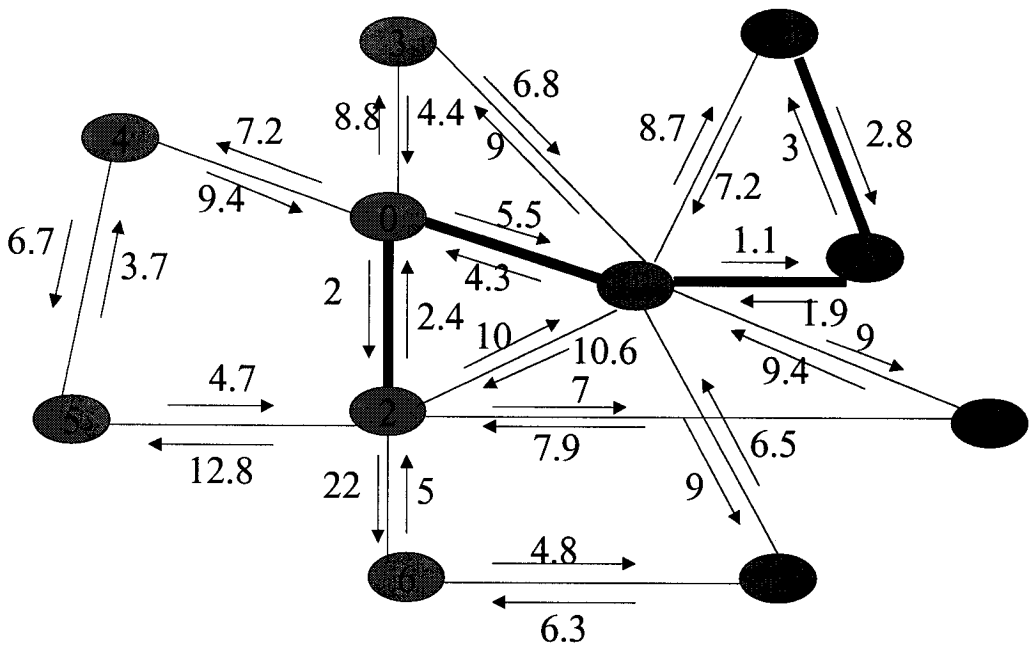


Figure D.1.3 Network status of Network Topology 1 Case 3 6:00

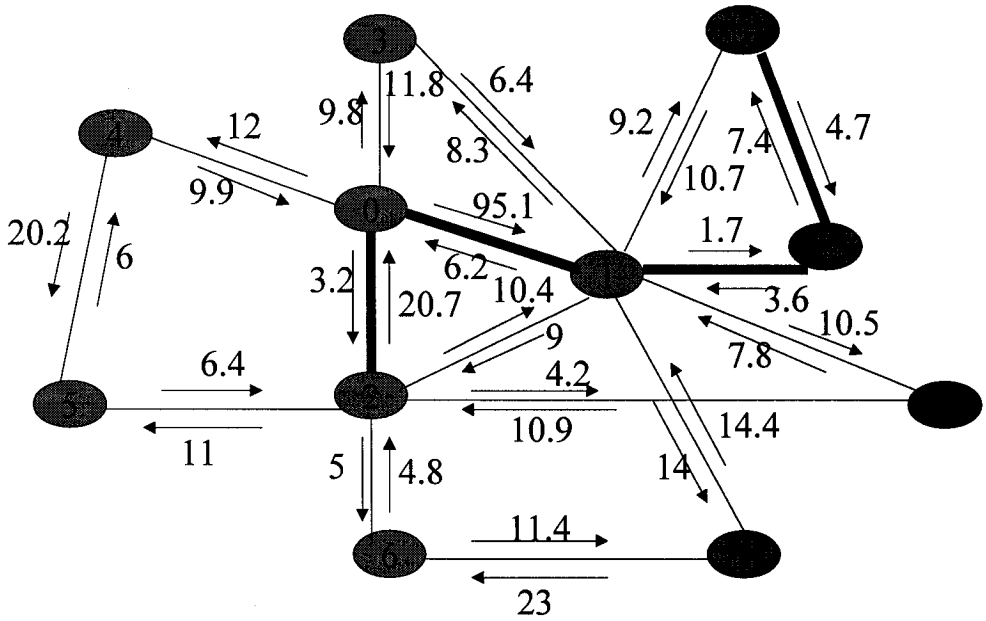


Figure D.1.4 Network status of Network Topology 1 Case 4 9:00

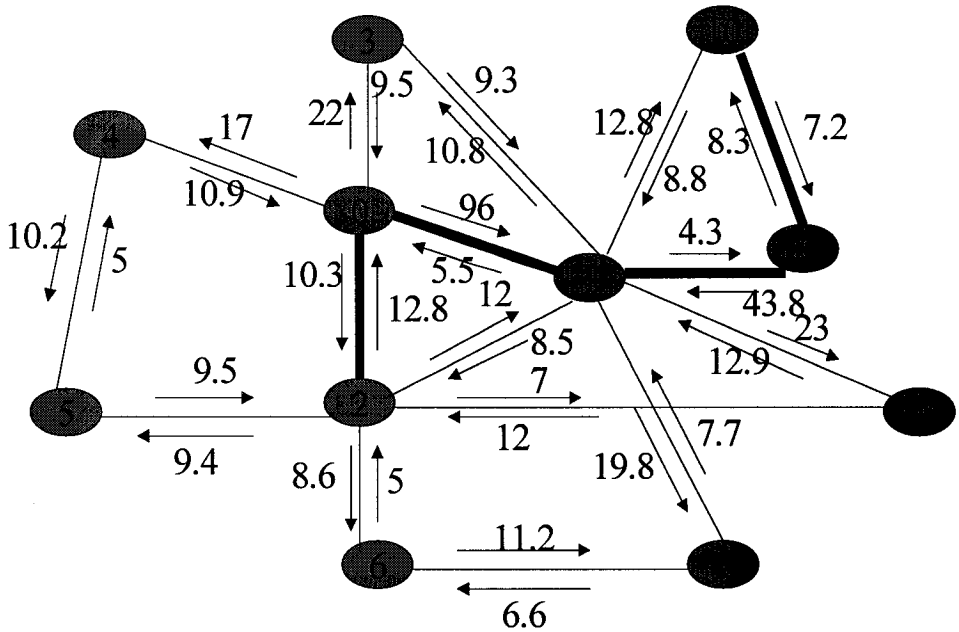


Figure D.1.5 Network status of Network Topology 1 Case 5 12:00

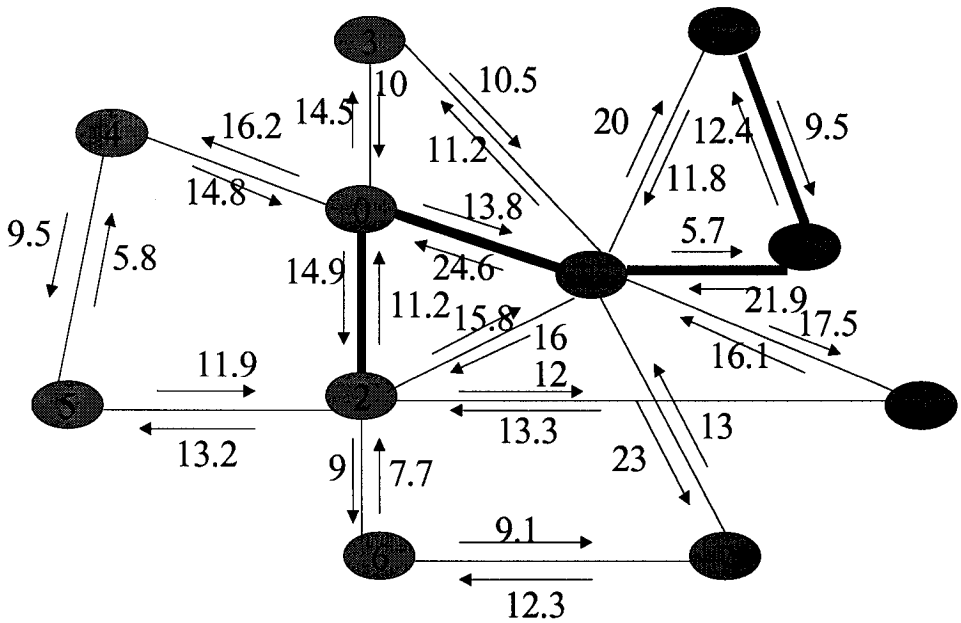


Figure D.1.6 Network status of Network Topology 1 Case 6 15:00

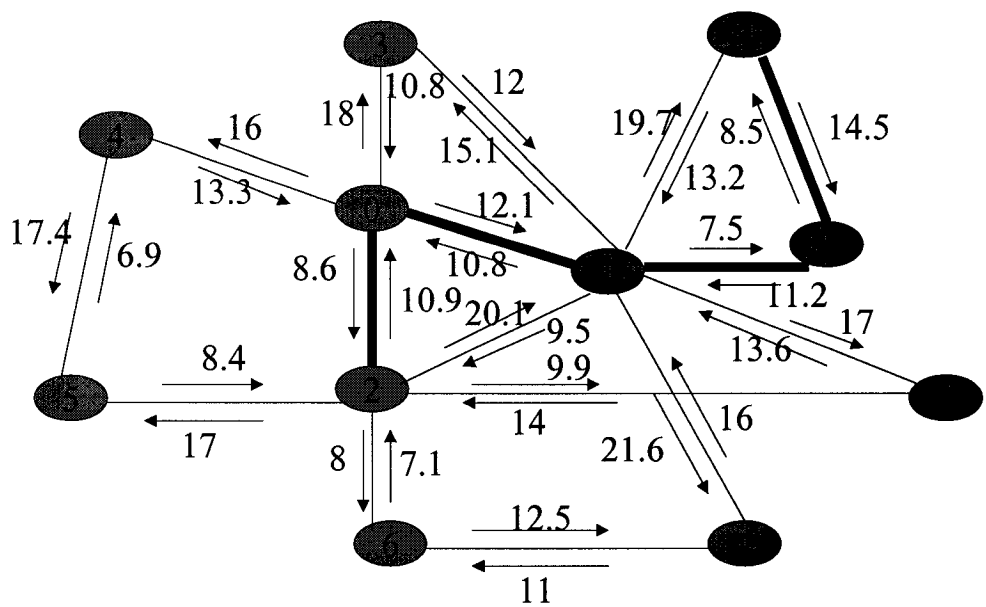


Figure D.1.7 Network status of Network Topology 1 Case 7 18:00

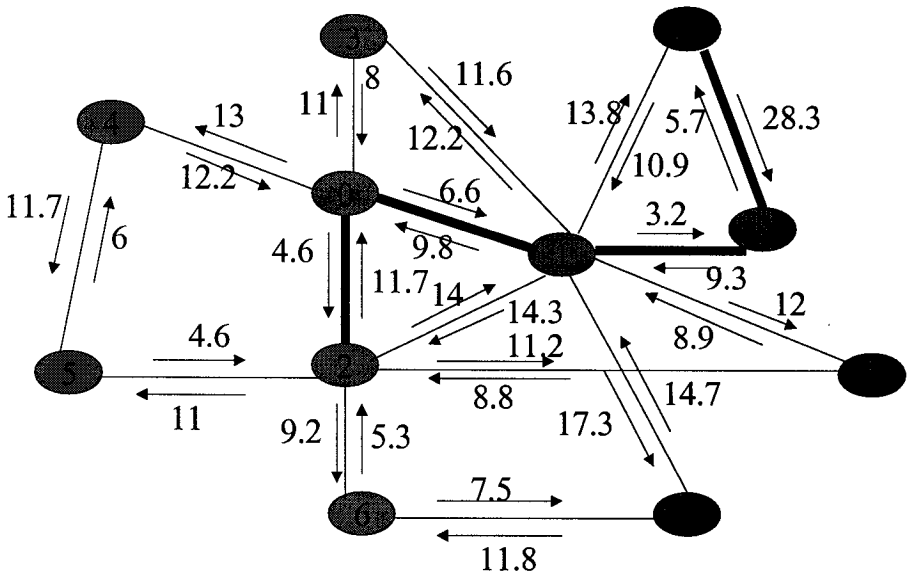


Figure D.1.8 Network status of Network Topology 1 Case 8 21:00

D.2 Network Topology 2 Status

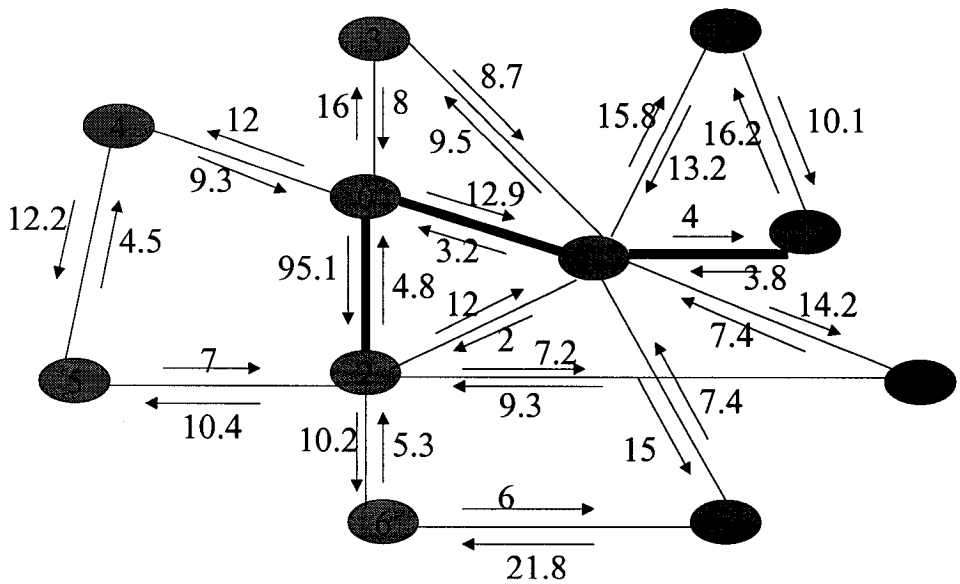


Figure D.2.1 Network status of Network Topology 2 Case 1 0:00

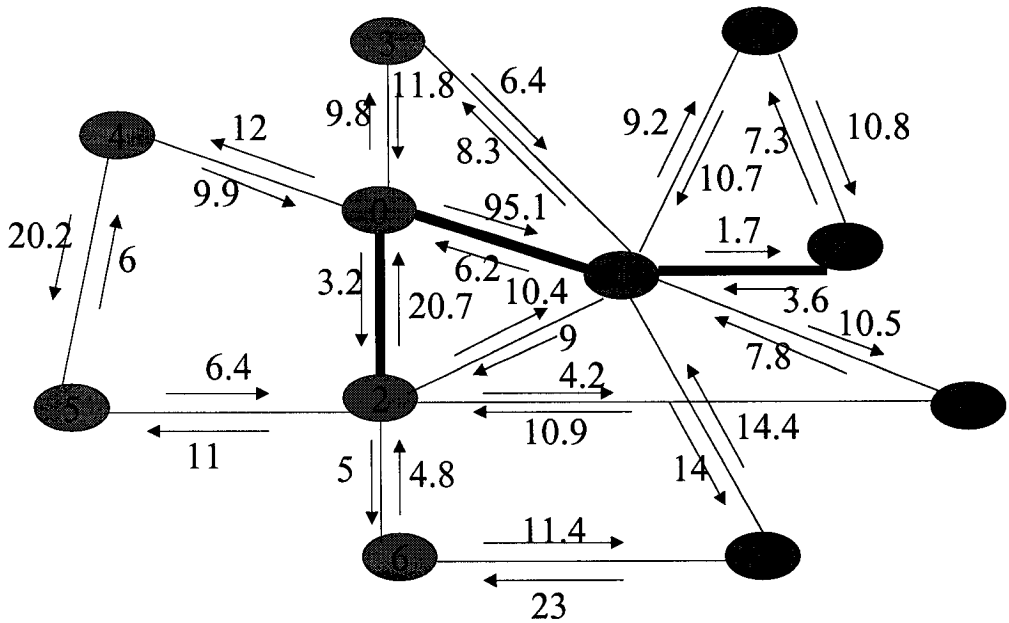


Figure D.2.4 Network status of Network Topology 2 Case 4 9:00

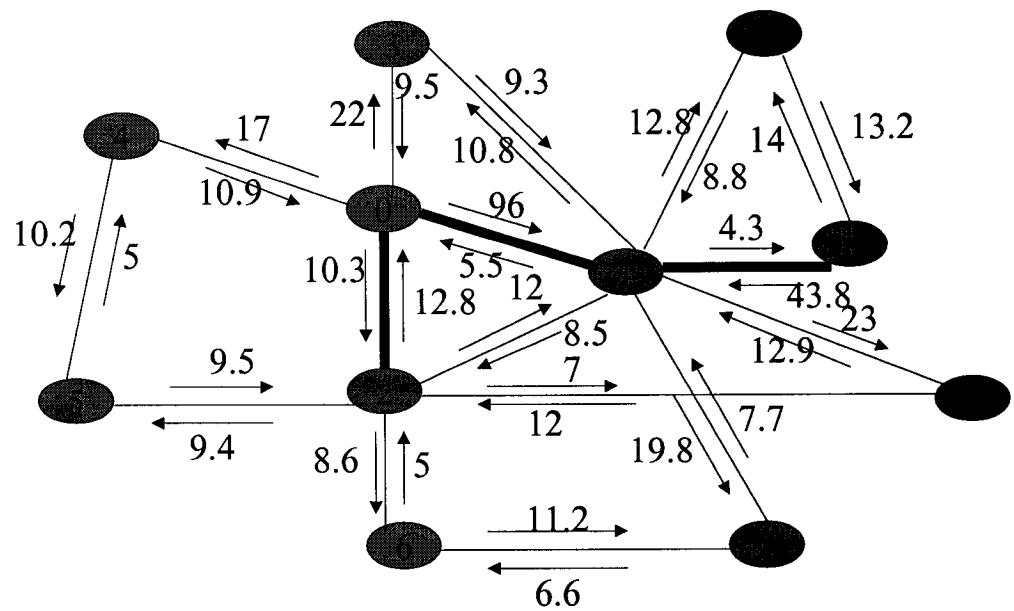


Figure D.2.5 Network status of Network Topology 2 Case 5 12:00

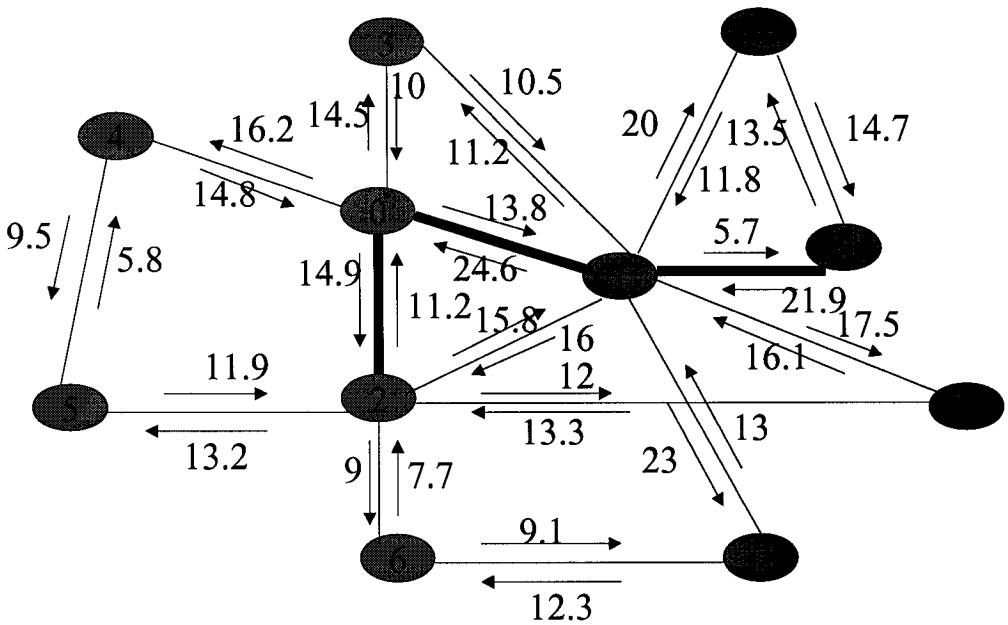


Figure D.2.6 Network status of Network Topology 2 Case 6 15:00

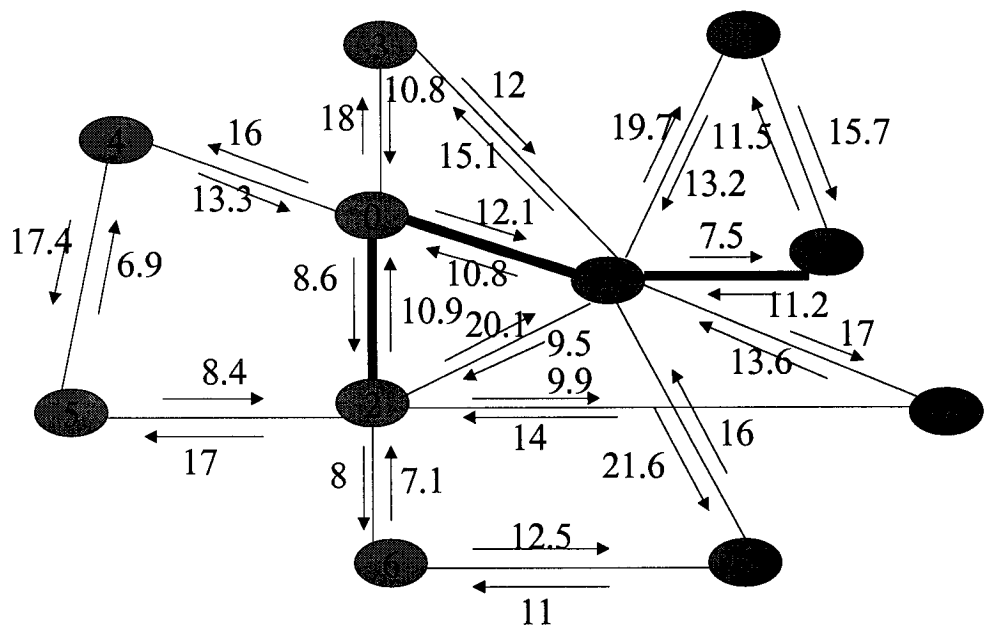


Figure D.2.7 Network status of Network Topology 2 Case 7 18:00

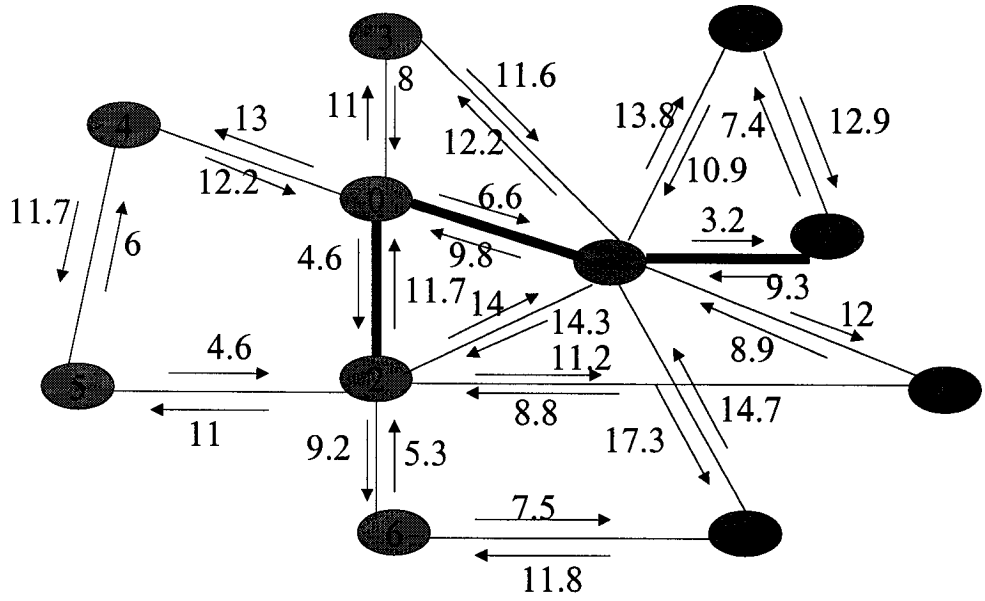


Figure D.2.8 Network status of Network Topology 2 Case 8 21:00

D.3 Network Topology 3 Status

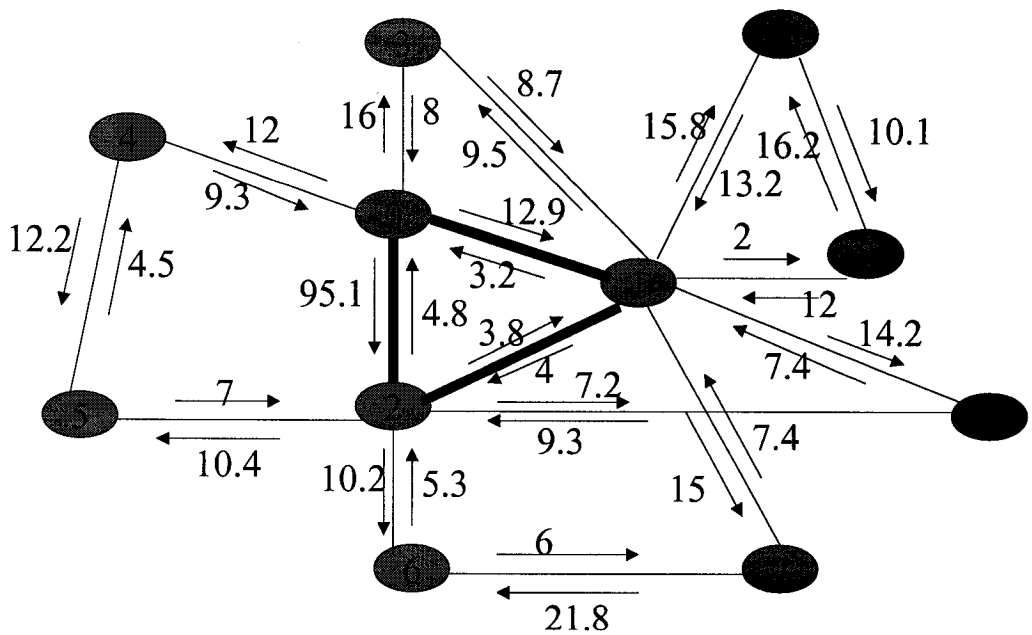


Figure D.3.1 Network status of Network Topology 3 Case1 0:00

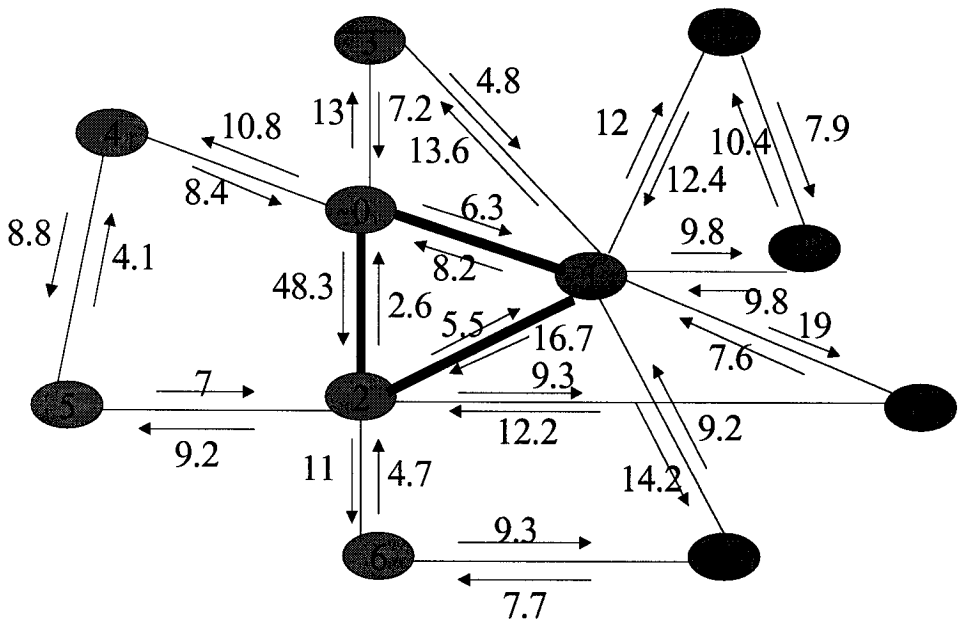


Figure D.3.2 Network status of Network Topology 3 Case2 3:00

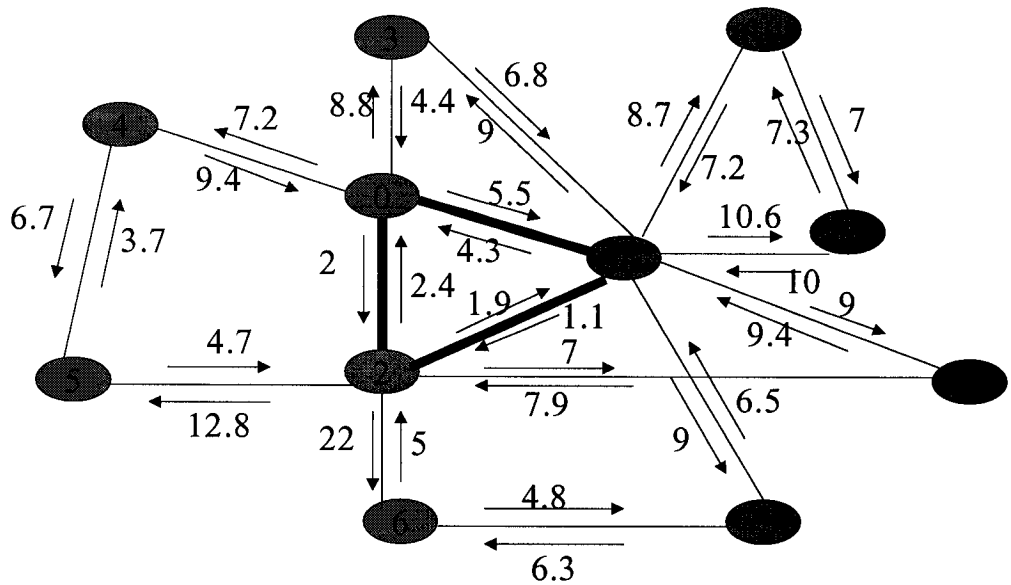


Figure D.3.3 Network status of Network Topology 3 Case3 6:00

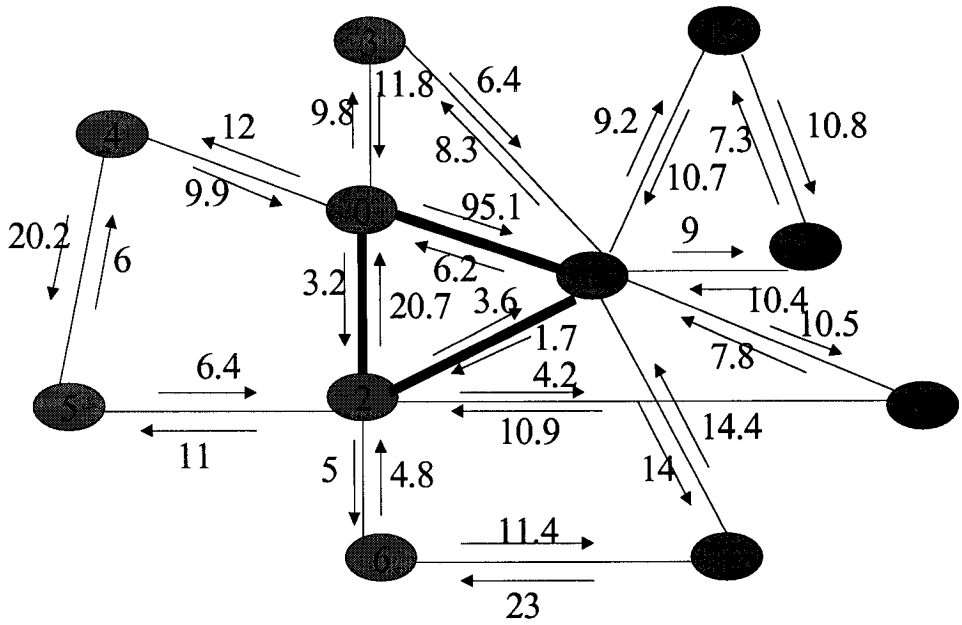


Figure D.3.4 Network status of Network Topology 3 Case4 9:00

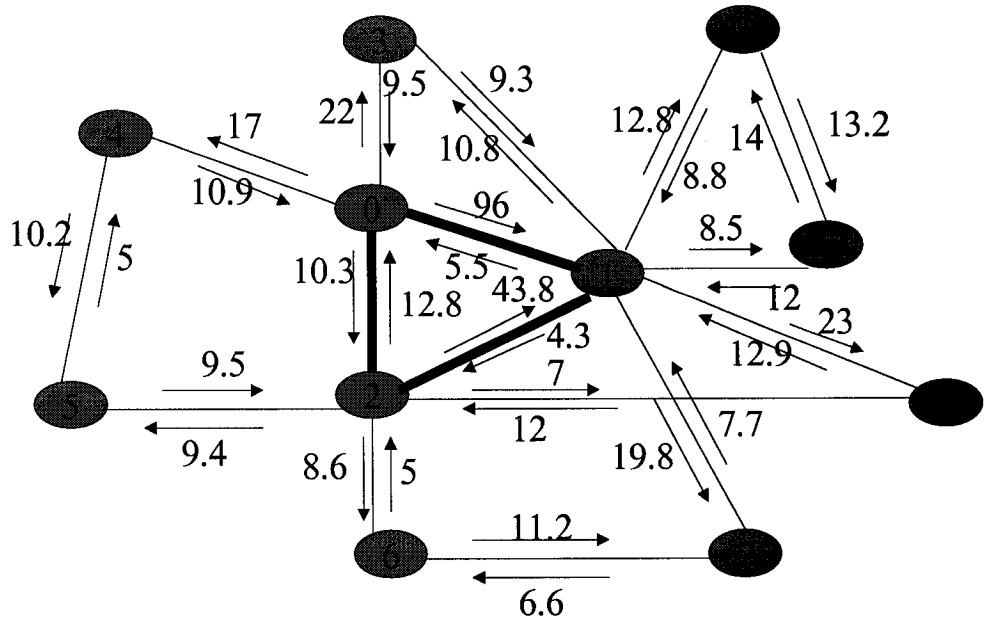


Figure D.3.5 Network status of Network Topology 3 Case5 12:00

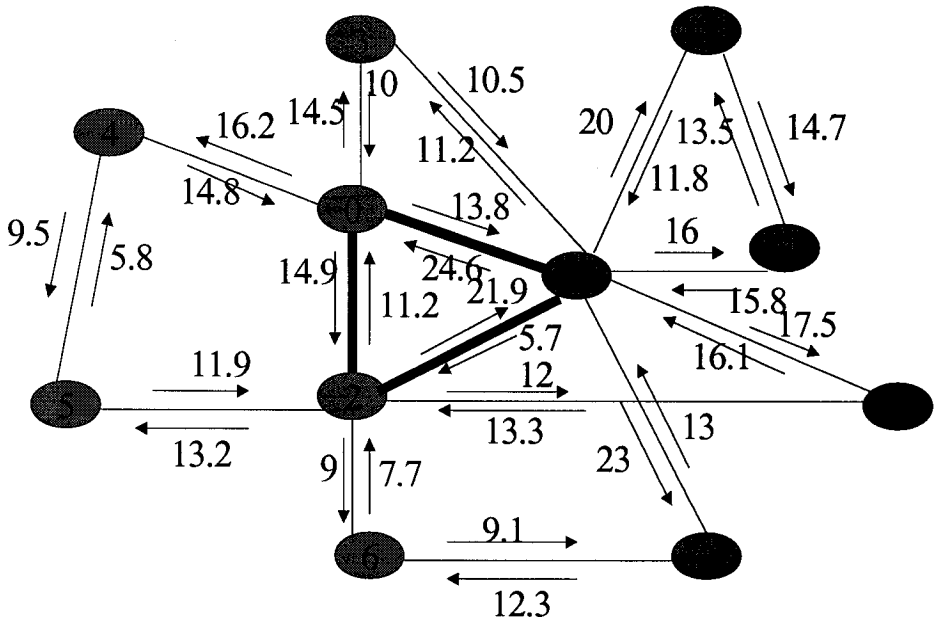


Figure D.3.6 Network status of Network Topology 3 Case6 15:00

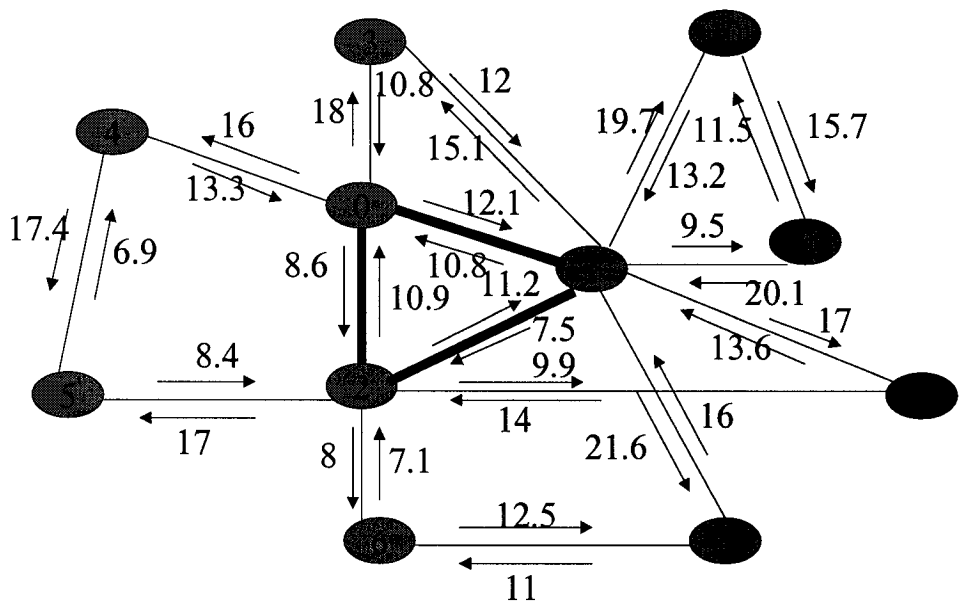


Figure D.3.7 Network status of Network Topology 3 Case7 18:00

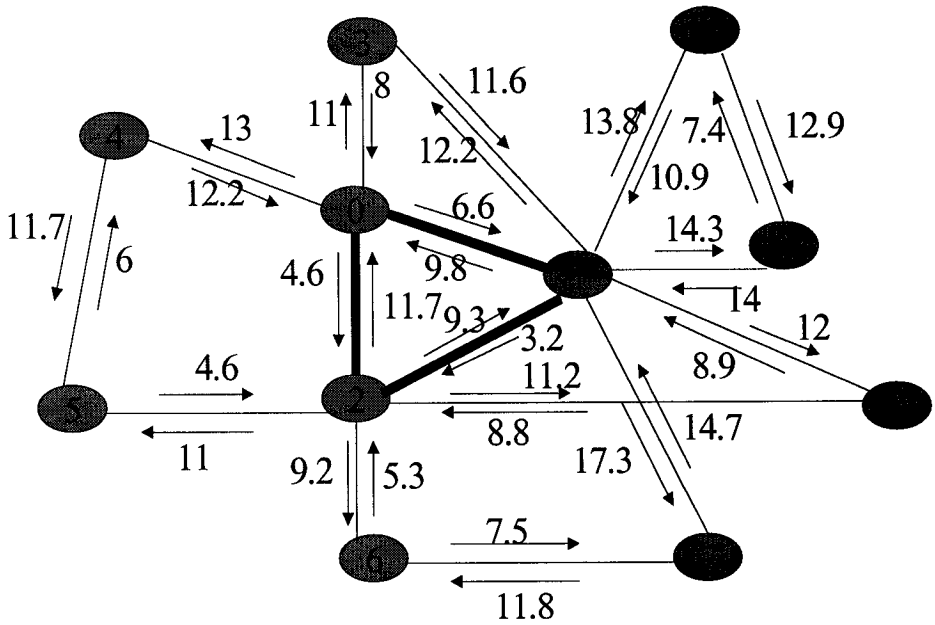


Figure D.3.8 Network status of Network Topology 3 Case8 21:00