Tunneling Based Solution to Bypass Internet Access Denial by International Internet Service Providers

BY

MOHAMMED ABDUL KHADIR KHAN ASIF

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING
DECEMBER 2010

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MOHAMMED ABDUL KHADIR KHAN ASIF** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING.**

THESIS COMMITTEE

Marwan

Dr. Marwan H. Abu Amara (Chairman)

Dr. Mohammed H. Sqalli (Co – Chairman)

Ashraf

Dr. Ashraf S. Hasan Mahmoud (Member)

Dr. Farag Ahmed Azzedin (Member)

R E AbdelAal

Dr. Radwan Abdel Aal (Member)

Department Chairman
Dr. Basem Al Madani

Dean of Graduate Studies
Dr. Mohammed Salam Zummo

29/1/11
Date

Dedicated to

My Beloved Parents and Brothers

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful

All praise and glory are due to Almighty Allah (SWT), to Him belongs the sovereignty of the heavens and the earth. I am grateful to Allah for all His favours on me since my birth, these blessings are indeed innumerable, and the greatest of His bounties on me is being a Muslim. Peace and blessings of Allah be upon the noblest of mankind, Muhammad (Peace Be upon Him), his household, his companions and the generality of the true believers to the last day.

I would like to thank my beloved parents, beloved grandparents, aunties, uncles, my brothers, cousins and all my family members for their moral support throughout my academic career. I would never have been able to pursue this without their prayers, cooperation, understanding, patience, guidance, and support.

I would like to express my profound gratitude and appreciation to my thesis committee chairman Dr. Marwan H. Abu-Amara for his support, intellectual guidance, suggestions, advice, and readiness to assist during this research. His extensive knowledge, patience, and experience made it possible to shape the thesis. He was always available when I needed him and he answered all my questions, at any time of the day. My deepest appreciation goes to my thesis committee co-chairman Dr. Mohammed Houssaini Sqalli for his support, guidance, suggestions, and advice. I am extremely grateful to him for his prompt replies and his encouragement and having trust in me. I would like to extend my deepest appreciation to my thesis committee members Dr. Ashraf S. Hasan Mahmoud , Dr. Farag Ahmed Azzedin , and Professor Radwan Abdel-Aal for their interest, invaluable cooperation and support. Their continuous support, advice and encouragement can never be forgotten.

I would like to acknowledge the support and facilities provided by King Fahd University of Petroleum and Minerals. I would like also to acknowledge the staff members for their kind support and continuous cooperation. Moreover, I would like to express my deepest thanks to every instructor who contributed in building my knowledge and experience. Thanks are also due to faculty and staff members of the Computer Engineering Department for their cooperation.

Finally, I render my profound acknowledgements to my project team members Albaiz, Alrefai, Fahd, and Khaled for the countless hours that we spent on discussing and proposing solutions for the problems that we faced. Acknowledgements are due to my friends and colleagues who made my stay at the university a precious and an unforgettable era. I would like to make special thanks to my brother Mohammed Abdul Waheed Khan and my friends Kareem bhai, Rizwan bhai, Waseem bhai, Muqtadir bhai, Fareed bhai, Saad bhai, Zahid bhai, Sujahat bhai, Sarfaraz bhai, Khalil, Nasar, Faizan, Malik, Nayeem, Eesaah, Junaid, Moiz, Ansar, Musheer, Irfan, Minhaj, Suli, Aqeel, Touseef, and Zubair. Last but not least I would like to thank everybody who contributed to this achievement in a direct or an indirect way.

Contents

List of Tables

List of Figures

THESIS ABSTRACT

Name:	Mohammed Abdul Khadir Khan Asif
Title:	Tunneling Based Solution to Bypass Internet Access Denial by International Internet Service Providers
Major Field:	COMPUTER ENGINEERING
Date of Degree:	DECEMBER 2010

The objective of this thesis is to propose effective counter measures to address the problem of the malicious act of denial of Internet access by International Internet Service Providers (IISPs), and to increase the resilience of the Internet services. Furthermore, the thesis is to address this specific problem at the routing level. The proposed solution is dependent on the use of a tunneling protocol. The proposed solution and the effect of tunneling on the problem solution is evaluated through simulation experiments. For the simulation purpose the OPNET network simulator was used. The performance of the proposed solution is evaluated in terms of throughput, end-to-end delay, and packet drop. Both the proposed solution and the simulation experiments followed the Internet standards as closely as possible.

MASTER OF SCIENCE DEGREE

King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

DECEMBER 2010

ملخص الرسالة

الاسـم:	محمد عبـد القــادر خان آصـف
عنـوان الرسالة :	حل مشكلة الحرمان المتعمد من قبل مقدمي خدمات الإنترنت الدوليين من الوصول إلى الإنترنت بإعتماد بروتوكول الإتصال النفقي
التخصــــص:	هندسة حاسب آلي
تاريخ الدرجة:	ديســـمبر2010

الهدف الرئيسي من هذه الرسالة هو اقتراح تدابير مضادة فعالة لمعالجة مشكلة الحرمان المتعمد من قبل مقدمي خدمات الإنترنت الدوليين (IISPs) من الوصول إلى الإنترنت، بالإضافة إلى زيادة مرونة خدمات الإنترنت. وفي هذه الأطروحة تم معالجة هذه المشكلة في طبقة التوجيه (Network layer). والحل المقترح يعتمد على استخدام بروتوكول الاتصال النفقي (Tunneling). حيث تم تقييم الحل وتأثيره على المشكلة من خلال محاكاة المشكلة والحل باستخدام برنامج محاكاة شبكات (OPNET). وتم تقييم أداء الحل المقترح من حيث نسبة الإنتاجية، وحساب الفترة الزمنية للتأخير بين نهاية الطرفين، وعدد البيانات المفقودة. وتم مراعاة معايير الإنترنت العالمية في كل من الحل المقترح وتجارب المحاكاة بقدر الإمكان.

درجة ماجستير في العلوم

جامعة الملك فهد للبترول والمعادن ، الظهران ، المملكة العربية السعودية

ديســـمبر2010

Chapter 1

Introduction

The Internet has become one of the most important means for reachability and communication with others. The number of people using the Internet in June 2010 exceeded 1.96 billion users which constitutes about one quarter of the entire population of the world [1]. The Internet is a huge network of hundreds of millions of nodes that exchange information with each other. Because of the dynamic nature and very large size of the network, routing of data within it takes place on two hierarchical levels. At a lower level, a group of devices are controlled by a single administrative autonomy that has a complete view of its own network and is responsible for the routing of Internet traffic within its boundaries (intra-domain routing) [2]. The network controlled by a single administrative autonomy is referred to as an autonomous system (AS). At a higher level, routing between different ASes (inter-domain routing) takes place by means of the Border Gateway Protocol (BGP).

Due to the importance of the Internet, it is crucial to ensure the availability and the resiliency of the Internet against all malicious and non-malicious activities. Accordingly, the following subsections explore a specific type of malicious activities that can affect the Internet availability at a large scale, clearly state the formal description of the problem, the objectives, the scope, the assumptions, and the limitations of the research covered in the thesis.

1.1 Internet Denial By Service Providers

From a strategic point of view for any country such as the Kingdom of Saudi Arabia (KSA), improving the country's Internet resilience including not having to rely on a single International Internet Service Provider (IISP) for providing Internet connectivity to the country is very important. The Internet resiliency can be impacted by many causes including outages. The causes for the Internet unavailability or outage can be categorized into either non-malicious or malicious including hardware and/or software failures, denial of service attacks, terrorists' attacks, and deliberate denial of Internet access by IISPs. The initial literature review in the area of Internet availability and resilience classified the Internet unavailability causes and found that they occur at three levels: application, routing, and physical [3].

The current status of the literature indicates that the subject of deliberate denial of Internet access by IISPs has not been researched, and no known commercial solutions are available to address the issue. Accordingly, an IISP can utilize BGP to maliciously block routing information from and to a particular AS that the IISP is serving. Further, the IISP can filter traffic to and from a specific AS. Therefore, the thesis addresses the problem of intentional denial of Internet access by IISPs but only at the routing level.

1.2 Research Motivation

The basic motivation for the research is to provide resilient Internet service to a local region. In this study, we are considering KSA as a local region. Internet access denial can be due to many reasons, such as a malicious IISP through the manipulation of advertised routes to ASes served by the IISP. In general, any IISP can maliciously block a certain region traffic from passing through. The blocking can happen on either direction (i.e., incoming traffic to that region or outgoing traffic from that region). Under such a case the local region becomes partially or completely isolated from the rest of the Internet. The worst case scenario could happen if the Internet access denial is caused by one or more tier-1 ISP which results in a complete Internet isolation. Tier-1 ISPs are the core of the Internet that are responsible for transfering large amounts of data, have international coverage, and are directly connected to each other and to a large number of lower tier ISPs. The victim network can be an entire region's network, an enterprise network, or a single host depending on the location of the blocking IISP.

The term Malicious Service Provider (MSP) has been used in the literature to describe a malicious activity by a service provider to hijack Internet prefixes [4]. For example, 106,089 prefixes were hijacked in December 2004 by AS 9121 [5]. Similarly, prefix 64.233.161.0/24 which includes IP addresses for Google was hijacked in May 2005 by AS 174 [6]. Such incidents can expose sensitive traffic sent by users to the hijacker, allowing the hijacker to drop, record, and/or modify the contents of the intercepted traffic.

The concept of a malicious ISP is realistic even though IISPs are supposed to provide the promised service to their customers for fear of losing them, and, ultimately, jeopardizing their reputations. However, there are many reasons that may force an IISP to become malicious and perform an Internet access denial to a specific region. For example, Internet access denial can be driven by political motivations, as governments may attempt to establish an Internet embargo on a targeted region by forcing ISPs to block Internet access to that specific region. Many large services and networks have been attacked recently for political motivations. As an example, on December 2009, Gmail had many attacks targeting email accounts of Chinese human rights activists [7]. Similarly, Twitter has also been attacked during 2009 by hackers from Iran [8]. Another prime example of political motivations of a service provider to deny Internet access to an organization are the recent attempts by many governments to pressure service providers to deny access to WikiLeaks [9]. Likewise, ISPs' routers may be hacked by attackers and reconfigured to drop traffic, causing Internet access denial.

It is possible that the Internet in KSA can become a victim of this type of problem, especially that none of the IISP or higher tier ISP reside in KSA. These reasons, and many others, can encourage IISPs to perform Internet access denial on a specific region or country. The risk and impact of Internet access denial could be critical for the government and many other organizations. Therefore, there is a need to study this problem and how to deploy an effective solution. This thesis aims at proposing a solution that can circumvent this particular problem at the routing level and study the effect on the network performance.

1.3 Problem Description

The thesis addresses the problem of the unavailability of the Internet due to the malicious act of denial of Internet access by the IISP. Furthermore, in the problem considered, we address only the routing level of the problem of the denial of Internet access.

To illustrate the problem further, Figure 1.1 provides a typical network configuration that consists of four ASes. AS100 represents the local AS (e.g. AS for KSA), AS200 represents a neighbouring country AS, AS300 is the IISP and connects both AS100 and AS200 to other ASes in the Internet, and AS400 is a possible destination AS for AS100. Figure 1.1 represents the normal behaviour of the Internet without any denial of Internet access by the IISP.

In the problem considered in this thesis, we assume that one IISP, AS300, provides the service to the local AS, AS100. While the IISP denies the service to the local AS, it still provides the service to other ASes such as AS200 and AS400. The problem is then how to provide Internet services to the local AS, even after the IISP has denied Internet access to the local AS.

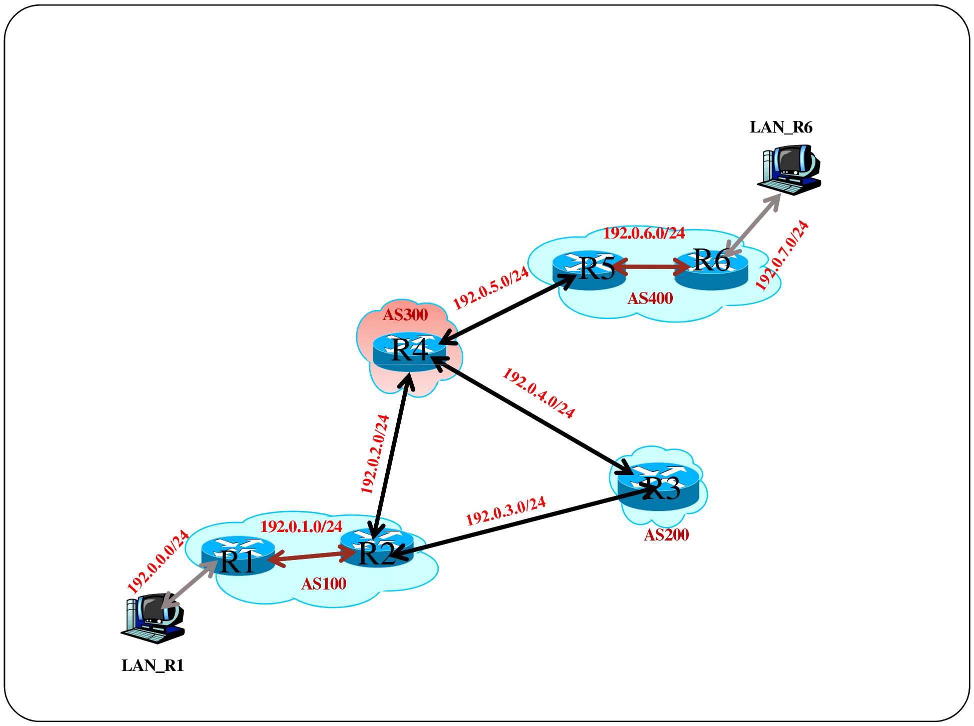


Figure 1.1: Normal behaviour of the Internet.

1.4 Thesis Objectives

The main objective of the thesis is to propose a tunneling based solution to the Internet unavailability due to the malicious act of denial of Internet access by IISPs only at the routing level. Thus, the thesis will concentrate on the following:

1. Proposing a solution to the IISP blocking problem through tunneling. This involves surveying different tunneling protocols, and choosing the most appropriate tunneling protocol for the solution.

2. Validating the proposed solution through simulations using OPNET [10].

3. Characterizing the network performance under different tunneling protocols while considering the lack as well as the presence of encryption.

1.5 Scope

The thesis takes into account the following scope for the proposed solution:

1. Only a tunnel-based solution is considered whereby a tunnel is created with the help of a neighbouring AS from the local AS to each destination AS. Note that a tunnel is needed only if traffic destined to ASes passes through the malicious IISP.

2. The proposed solution is tested and evaluated using different tunneling protocols, and with and without encryption through simulation. For the simulation

purpose, the network simulator OPNET is used [10].

3. While simulating the proposed solution we will follow the standards as closely as possible.

1.6 Assumptions

In line with the scope of the proposed solution outlined in section 1.5, the following assumptions are made:

1. Only one IISP serves the local AS, and is maliciously blocking the Internet access only to that local AS.

2. The destination point of a tunnel is known apriori, and a service level agreement with the destination ASes is already established.

3. Neighbouring ASes have service level agreements with the local AS so that a tunnel can be easily created through the neighbouring ASes.

4. The IISP malicious blocking of the Internet access to the local AS is known apriori.

1.7 Limitation

As OPNET [10] is used to simulate the proposed solution, only OPNET supported tunneling protocols and encryption algorithms will be examined. More specifically, the IPSec tunneling protocol was not considered as it is not supported by OPNET's Discrete Event Simulation (DES).

1.8 Organization of Thesis

The remainder of the thesis is organized as follows. Chapter 2 starts with background information about the Border Gateway Protocol (BGP), as well as the different types of tunnels. Chapter 3 proceeds with a literature survey related to our work. In particular, we discuss the security issues related to BGP. Afterwards we look at the different applications of tunneling protocols, and at the end of the chapter we look at the tunneling techniques and Internet resiliency. Chapter 4 provides an overview of different approaches to resolve the Internet access denial problem at the routing level. Chapter 5 provides the baseline configuration of the network and followed by the tunnel configuration of the network (i.e., the proposed solution). At the end of the chapter, we look at the scalability issue to the proposed solution. Chapter 6 provides the performance evaluation of the proposed solution. This is then followed by the discussion of the proposed solution in terms of connectivity and performance; and the chapter is concluded by some recommendations. The thesis finally concludes in chapter 7, where we discuss the overall picture of the proposed solution and the chapter is concluded by suggestions for the future work.

Chapter 2

Background

2.1 Introduction

In this chapter, we briefly present the relevant background on BGP and different types of tunneling protocols.

2.2 Border Gateway Protocol (BGP)

BGP is a de facto standard and is the only inter-AS routing protocol of the Internet. BGP is a path vector routing protocol. BGP is used in tens of thousands of Internet routers [11]. BGP views the Internet as a mesh of a number of ASes connected by inter-domain (inter-AS) links. BGP is responsible for discovery and maintenance of paths between distant ASes in the Internet. BGP exchanges information about how to reach the blocks of IP addresses. The block of IP addresses is referred to as an IP prefix. BGP routers exchange routing information via UPDATE messages. UPDATE messages can be classified into two types: route withdrawal and route announcement. When a BGP router receives an UPDATE message from its neighbouring BGP router, the message will be processed, stored, and redistributed in accordance with BGP specification [11], and the routing policies of the local AS. BGP provides reachability information to ASes and distributes external reachability internally within an AS. When an AS advertises a route to the next AS through an UPDATE message, it adds the Autonomous System Number (ASN) at the beginning of the AS-PATH of the UPDATE message.

A router that communicates directly with other routers via BGP is known as a BGP speaker. For complete Internet reachability, every AS must have at least one BGP speaker that is connected to at least one other AS. BGP uses two main protocols; Internal BGP (IBGP) and External BGP (EBGP). Note that IBGP is used inside an AS and EBGP is used to exchange network reachability information between ASes [11].

2.3 Tunneling Protocols

In networking, IP tunnels are often used to connect two disjoint IP networks. A tunneling protocol uses a protocol to encapsulate a different protocol. By using tunneling, a payload can be carried over an incompatible delivery network, or over a secure path through an untrusted network. Tunneling can be secured by using data encryption to transport insecure payload protocols over a public network such as the Internet, and hence creating a virtual private network (VPN). In the rest of this section, we will discuss different types of tunneling protocols.

The IP in IP tunneling protocol is defined in RFC 1853 [12], and it discusses the implementation techniques for using IP protocol/payload number 4 encapsulation for tunneling with IPSec and other protocols. In IP-in-IP encapsulation, the IP datagram is encapsulated by inserting an outer IP header before the original IP header [13]. The “endpoints” of the tunnel are the Source Address and Destination Address of the outer IP header. The inner IP header Source Address and Destination Address identify the original sender and recipient of the datagram respectively. Other protocol headers such as the IP Authentication header may be inserted between the outer IP header and the inner IP header. Once the encapsulated datagram arrives at the end of the tunnel destination node, it is decapsulated and the original IP datagram is delivered to the destination indicated by the original destination address field [13]. An encapsulated datagram is typically larger than the original datagram.

Another example of a tunneling protocol that runs over the network layer is the Generic Routing Encapsulation (GRE) protocol[14]. GRE is a protocol that runs over IP, with RFC 1918 private addresses, over the Internet using delivery packets with public IP addresses. In this case, the delivery and payload protocols are compatible, but the payload addresses are incompatible with those of the delivery network.

The Point to Point Protocol (PPP) [RFC 1661] defines an encapsulation mechanism for transporting multiprotocol packets across Layer 2 (L2) point to point links. Layer 2 Tunneling Protocol (L2TP) extends the PPP model by allowing the L2 and PPP endpoints to reside on different devices interconnected by a packet-switched network. With L2TP, a user has an L2 connection to an access concentrator and the concentrator then tunnels individual PPP frames to the Network Access Server (NAS) [15].

The Point to Point Tunneling Protocol (PPTP) can be used as a tunnel through an IP network. PPTP does not provide confidentiality or encryption. It relies on the protocol being tunneled to provide privacy. PPTP uses an enhanced GRE mechanism to provide a flow and congestion controlled encapsulated datagram service for carrying PPP packets [16].

Internet Protocol Security (IPSec) can be used for securing the tunnels. IPSec is a standardized framework for securing IP communications in general by encrypting and/or authenticating each IP packet in a data stream. IPSec has two modes; transport mode and tunneling mode. In transport mode, only the payload of the IP packet is encrypted. It is fully-routable since the IP header is sent as plain text. In tunnel mode, the entire IP packet is encapsulated within an IP packet to ensure that no part of the original packet is changed as it is moved through a network [17].

Chapter 3

Literature Survey

3.1 Introduction

In this chapter, we survey the literature related to our work. Initially, we discuss some BGP security issues that are relevant to the problem considered in the thesis. After that, we look at the tunneling protocols and their applications, followed by a description of the tunneling techniques and the Internet resiliency.

3.1.1 BGP Security Issues

The major concern about BGP security is that malicious BGP routers can arbitrarily falsify BGP routing messages and spread incorrect routing information [18]. As stated earlier, BGP routers exchange routing information via UPDATE messages. Incorrect UPDATES, due to either a BGP router misconfiguration or a malicious activity, may cause serious problems to routing in the Internet. For example, on May 7, 2005, Google went down for less than an hour [19]. One speculation is that DNS poisoning attacks caused this service outage because an AS falsely claimed to originate Google's prefix and some traffic sent to Google was redirected to other websites. Google later denied that it was under any attack and clarified that to their knowledge, their service outage was due to internal DNS misconfigurations [19]. Similarly, in June 2006, several prefixes were hijacked by AS 23520 [20]. Another example occurred on April 25, 1997, where a small ISP incorrectly announced all the prefixes learned from its upstream ISP as its own prefixes [21]. These faulty routes spread to the rest of the Internet. As a result, most of the Internet traffic is routed through the small ISP. The traffic overwhelmed the misconfigured and intermediate routers, many routers were affected and even crashed, and the whole Internet was unstable for hours.

As stated earlier, another serious concern about BGP is that BGP does not allow an AS to control exactly how its traffic is routed to other destinations [22]. Thus, Internet access denial can take place if the packets are routed by BGP through a malicious ISP, and if the malicious ISP drops these packets. Hence, the Internet access denial can be prevented by controlling the traffic path so that the traffic does not pass through the malicious ISP, or by preventing the traffic from being dropped at the malicious ISP by concealing the traffic identity.

A modification of BGP is needed at all routers in the Internet in order to control the traffic path so that traffic is not sent through the malicious ISP. Accordingly, Quoitin et al. [23, 24] proposed BGP tuning techniques to influence the path selection process of remote ASes. Alrefai [25] enhanced the BGP tuning techniques proposed by Quoitin et al. to achieve better scalability results.

Zhang and Zhao [18] pointed out a new type of attack, namely selective dropping attack that has not been studied before. A selective dropping attack occurs when a malicious router intentionally drops incoming and outgoing UPDATE messages, which results in data traffic being black-holed or trapped in a loop. The authors conducted research through the analysis of this type of attack and advocated that new security countermeasures should be developed to detect and prevent such an attack. However, the authors did not present any technique to detect and prevent such attacks.

The "Dropping" behavior discussed in [18] indicates that a router itself drops the incoming or the outgoing UPDATE message. Dropping an incoming UPDATE message means that when a router receives an incoming UPDATE message from a peer, it stores the route information internally but does not apply this route information to the route selection process. Similarly, dropping an outgoing UPDATE message means that when a router selects a new route, it does not send the new route to its peers. If the faulty/malicious router drops the incoming withdrawal or outgoing withdrawal messages for a particular prefix, then a black-hole for that traffic may be formed. If the router also drops incoming routes, either announcement or withdrawal, persistent packet forwarding loop can occur.

To address the selective dropping attacks in BGP, Chuah and Huang [26] proposed a scheme called Instability Analysis with Neighbour Probing (IANP) for detecting such attacks. In this scheme, an instability analysis will be triggered at an observation point, referred to as the monitor, when the number of received BGP UPDATES within a burst exceeds a certain threshold. When the monitor cannot identify the instability source, it will probe its neighbouring routers to see if they can identify the source of instability. Once the source of instability is identified, the monitor will check its stable route database to see if a selective dropping attack is embedded within this burst of BGP UPDATES. If a monitor suspects that a BGP router is conducting a selective dropping attack, then it will issue a warning message that will be flooded, with limited scope, across the BGP routers in the Internet.

3.1.2 Tunneling Protocols and Applications

Tunneling techniques have been used in various applications. For example, VPN and Mobile IP are the most widely known applications that use tunneling techniques. VPN helps to inter-connect multiple locations belonging to one enterprise or belonging to a group of related enterprises that are shared over a public network. A VPN creates a virtual tunnel that connects two endpoints of the enterprise. Accordingly, the VPN traffic sent over a public network is isolated from the rest of the traffic carried by the public network [27]. The most popular VPN tunneling protocols available are L2TP, PPTP and IPSec [15, 16, 17]. On the other hand, Mobile IP is designed to allow a mobile device to use a single permanent IP address even though the mobile device has moved from one network to another network. In Mobile IP, an extra IP address known as Care of Address (CoA) is assigned to the mobile device. It is used with a special purpose router known as the Home Agent (HA). The HA then uses an IP-in-IP tunnel to forward packets sent by a correspondent to the mobile device by using the mobile device assigned CoA [28].

Aqun et al. [29] provided a comparative study of various tunneling protocols such as L2TP, GRE, IPSec, and IP-in-IP using different criteria such as working mode, security mechanism, tunnel management and maintenance, support for multiplexing, support for multi-protocol, and Quality of Service (QoS). The paper concludes that from a working mode perspective, L2TP works in a client/server mode, whereas the other tunneling protocols work in a peer-to-peer mode. From security mechanism perspective, IPSec provides a complete built in security mechanism, whereas L2TP and GRE provide a weak security mechanism, and IP-in-IP provides no security mechanism. From a tunnel management and maintenance perspective, IP-in-IP has a soft state mechanism with no additional overhead since it relies on the Internet Control Message Protocol (ICMP) existing in the IP network, whereas L2TP provides some tunnel management and maintenance features such as sending HELLO messages periodically. On the other hand, IPSec and GRE are not considering the management and maintenance problem. From the support for multiplexing perspective, IP-in-IP does not support multiplexing, whereas the other tunneling protocols support multiplexing. With respect to the support for multi-protocols, both GRE and L2TP support multi-protocols, whereas IPSec and IP in IP do not support multi-protocols. From the QoS perspective, none of the above tunneling protocols provide support for QoS. The authors suggested that two tunneling protocols can be integrated for better performance such as IP-in-IP and IPSec.

Tunnels can be created virtually at different layers of the OSI stack. Typically, it is created either from the network layer or the data link layer. Saad et al. [30] discuss the various implementation and technical details of layer 2 tunneling techniques, such as L2TP, MPLS, etc.

In recent years, many researchers [31, 32, 33] have conducted research on the performance evaluation of different VPN tunneling protocols based on different operating systems. The most known VPN tunneling protocols are considered, namely PPTP, L2TP, IPSec, and Secure Socket Layer (SSL). IPSec is an OSI layer 3 tunneling protocol, whereas PPTP, L2TP, and SSL are OSI layer 2 tunneling protocols.

Joha et al. [31] considered the performance evaluation of remote access VPN by using Windows XP SP2 as a VPN client and Windows server 2003 and Fedora Core 6 as VPN servers. In their research, performance evaluation is provided for three commonly used VPN tunneling protocols, namely PPTP, L2TP/IPSec, and OpenVPN. The throughput, Round-Trip Time (RTT), jitter, and packet loss are used as performance metrics. The paper concludes that from the TCP throughput perspective, the PPTP on Windows server 2003 performs the best, and OpenVpn on Windows server 2003 performs the worst. On the other hand, the PPTP on Windows server 2003, PPTP on Fedora Core 6, L2TP/IPSec on Windows server 2003, and L2TP/IPSec on Fedora Core 6 have the highest UDP throughput. From the RTT perspective, PPTP on Windows server 2003 performs the best, and L2TP/IPSec on fedora core 6 performs the worst. Furthermore, the PPTP on Windows server 2003, PPTP on Fedora Core 6, L2TP/IPSec on Windows server 2003, and L2TP/IPSec on Fedora Core 6 have the lowest jitter and packet loss.

Narayan et al. [32, 33] evaluated the performance impact on different tunnel protocols and operating systems, as a result of using common encryption and data integrity algorithms. The common encryption algorithms used are Triple Data Encryption Standard (3DES) and Blowfish (BF). On the other hand, the common data integrity algorithms used are Message-Digest 5 (MD5) and Secure Hash Algorithm (SHA1). It should be noted that PPTP was designed to use only Microsoft Point to Point Encryption (MPPE).

Narayan et al. [32] provided the performance evaluation of VPN tunneling protocols in Windows 2003 environment and they extended their work in [33] with additional operating systems. The authors considered three operating systems (OS), Windows Vista, Windows Server 2003, and Linux Fedora Core 6 with three commonly used VPN tunneling protocols, namely IPSec, PPTP, and SSL. The authors also considered the use of different compatible encryption and integrity algorithms with the

VPN tunneling protocols. The statistics collected are on the throughput, CPU utilization, and initiation time as the performance of metrics. Throughput and CPU utilization without VPN were found to be 95Mbps and 5%, respectively. The authors concluded that when using Windows server 2003, the throughput of PPTP MPPE provided the highest bandwidth, i.e., 90 Mbps, while SSL 3DES SHA1 provided the least bandwidth, i.e., 40 Mbps. Further, the paper concluded that IPSec CPU usage is the least (i.e., between 5% and 12%) and SSL CPU usage is the most (i.e., up to 42%). When considering Windows Vista, the authors concluded that the throughput performance shows a similar pattern as Windows server 2003. However, the SSL average throughput is significantly lower than in Windows server 2003 (i.e., around 25Mbps). For the performance evaluation under Linux, the throughput trend is slightly different than the other OSs. The paper concluded that the IPSec, PPTP, and SSL throughput performance is not distinguishable. Further, the paper concluded that all three OS perform similarly when comparing the no VPN case to the PPTP case. However SSL values are distinctly different for each OS. Linux is the best with an average of 80Mbps and Windows Vista is the worst with an average of 20Mbps. From the CPU usage perspective, Linux consumes the highest CPU usage on both the client and the server sides. On the other hand, Windows server 2003 CPU usage is the least. From the initiation time perspective, Windows OS VPN initiation time is lower than the Linux VPN initiation time. The initiation time for Windows OS is below 0.6ms, and for Linux the range is 1ms to 1.7ms.

Khanvilkar et al. [34] studied some of the most popular Open-Source Linux-Based VPN solutions (OSLVs). In this research, they compared the network performance in terms of overhead, bandwidth utilization, and latency/jitter. The paper concludes that there is no single OSLV solution that considers all the aspects. They compared the OSLVs using UDP based tunnels and TCP based tunnels, where UDP and TCP based tunnels refer to the upper layer protocols that carry the tunnel traffic. UDP based tunnels have 50 percent lower overhead, 80 percent higher utilization, and 40-60 percent lower latency/jitter.

3.1.3 Tunneling Techniques and Internet Resiliency

The tunneling techniques have been used to improve the Internet resiliency. For example, to enhance the robustness of a network to dual link failures, Kini et al. [35] developed a technique that combines the positive aspects of the various single-link failure recovery techniques. In the proposed approach, every node is assigned three protection addresses and one normal address. The network recovers from the first failure using IP-in-IP tunneling using one of the protection addresses of the next node in the path. Packets destined to the protection address of a node are routed over a protection graph where the failed link is not present. Every protection graph is guaranteed to be two-edge connected by construction and guaranteed to tolerate a dual link failure.

Similarly, Wu et al. [36] consider a failure scenario that breaks an AS into two or more isolated parts and disrupt the connectivity among these AS partitions. The AS partitions cannot communicate between themselves, unless their neighbors provide extra connectivity to bypass the failure. Special configuration is needed, for example, tunneling, to be set as the neighbors cannot use the AS number to distinguish the partitions. The authors conclude that AS partitioning becomes equivalent to the failure of an access link.

Chapter 4

Tunnel Based Solution to Internet Access Denial

4.1 Introduction

In this chapter, we will consider an overview of the different approaches, including the tunnel-based approach, to resolve the Internet access denial problem while describing the advantages and the disadvantages to each approach.

4.2 Solution to Internet Access Denial

Different approaches can address the IISP blocking problem at the routing level. Two classes of solutions can be considered. One of the classes of solutions is based on traffic control such that traffic does not pass through the malicious IISP. The other class of solutions is based on identity hiding so that traffic does not get filtered by the malicious IISP. These two classes of solutions are discussed in the following subsections.

4.2.1 Traffic Control

This class of solutions depends on preventing the traffic from being sent to the malicious IISP. To implement this solution, we need to manipulate the existing BGP protocol to control incoming and outgoing traffic. Accordingly, the victim's traffic does not pass through the malicious IISP.

AlRefai [25] proposed a solution based on BGP tuning where in his approach the blocking problem is solved by changing the Local Preference (Loc-Pref) of the local AS so that the outgoing traffic is routed through non-malicious ASes. In addition, in order to control the incoming traffic, AS-Path shortening that advertises more specific prefixes or community techniques are considered. In the BGP tuning technique, the author is not relying on a single IISP and assumed that at least one non-malicious IISP is available. The drawback of the solution is the dependance on the existance of at least one addtional IISP beside the malicious IISP.

4.2.2 Identity Hiding

This class of solutions has two types; one that uses Network Address Translation (NAT) [37], and the other uses tunneling. The tunneling solution is proposed by this thesis and will be discussed in details later.

Network Address Translation

In the NAT approach of identity hiding [37], IP addresses of the local AS are hidden from the global Internet through the use of NAT routers. In this approach, NAT uses non-blocked IP addresses as public IP addresses to hide the identity of all the outgoing traffic. By the NAT technique, a small set of non-blocked IP addresses allows a large amount of hosts to access the Internet. As suggested by the author, multiple public IPs can be used to solve the scalability problem with the NAT routers being deployed at the gateway-level of the network to hide the identity of the traffic. The author concludes that from the performance point of view, the NAT delay is negligible. The author also proposed techniques for HTTP and SMTP servers' reachability behind NAT. A drawback of this solution is that peer-to-peer (P2P) applications require the use of a relay to function properly. However, the use of a relay causes a major hit on the performance of the P2P applications.

Tunnel-Based Solution

In the tunneling protocol approach of identity hiding, a tunnel is created from the local AS to the destination AS. To implement this solution, we can utilize available tunneling protocols like IPinIP, GRE, IPSec, etc. Before this type of solution is implemented, we need the presence of cooperating ASes before and after the malicious IISP. Afterwards, a tunnel needs to be created to each AS that is a neighbour of the malicious IISP with the local AS. By doing so, the malicious IISP can be bypassed. Once this solution is implemented, it is highly reliable. The only case where the solution does not work is when the destination host is within the malicious network. The scalability problem can be solved by creating multiple tunnels at the gateway-level of the local AS. Since we have cooperating ASes present after the malicious AS, we can utilize the existing tunnel to route the traffic to the neighbours of the cooperating ASes. This will limit the number of tunnels at the gateway-level at the local AS that need to be established. As there is no limit for the number of tunnels, and since we are creating a large number of tunnels at the gateway-level, this may degrade the performance of the router. To solve this issue we can use multiple public IPs at the gateway-level and we can distribute the tunnels on the gateway routers. The thesis focuses on identity hiding using a tunneling protocol and the solution aims to provide high Internet availability.

4.3 Qualitative Analysis and Comparison

In the Table 4.1 we provide a comparison between all the approaches considered in this chapter in terms of the following criteria: traffic filtering, setup overhead, communication overhead, difficulty to combat the method, malicious IISP services, and scalability. In the following paragraphs each criteria is discussed.

Traffic filtering refers to the amount of the traffic filtered out using the approaches discussed earlier. The methods that leads to no filtering of traffic are tunnel-based and NAT-based approaches, as both approaches use IP addresses that do not belong to the prefix that the malicious IISP has blocked. In both cases, the local region AS identity is hidden to the malicious IISP. On the other hand, the BGP tuning approach traffic will experience filtering if the traffic passes through the malicious IISP.

Table 4.1: Comparison Between Different Approaches.

	BGP Tuning	NAT-Based Solution	Tunnel-Based Solution
Filtering the traffic	Small	No	No
Setup overhead	Small	High	High
Communication overhead	No	High	Medium
Difficulty to combat the method	Easy	Difficult	Difficult
Malicious IISP services	No	Yes	No
Scalability	High	High	High

Setup overhead refers to the amount of the time needed to get all the required configurations in place to execute the approach. The setup overhead is high in the tunnel-based approach when compared to the BGP tuning approach. In the NAT based approach the setup overhead is high when the applications servers are behind NATs.

The communication overhead refers to the number of routing messages that need to be exchanged between the local region AS gateway routers before the method is effective. In the BGP tuning approach, no communication overhead is required, and the NAT-based approach required a high amount of overhead when both NATing and relaying used to facilitate peer-to-peer communication. On the other hand the tunnel-based solution requires medium communication overhead for establishing tunnels.

The difficulty to combat the method refers to the amount of effort required by the malicious IISP to overcome the solution. As such, the tunnel and the NAT based approaches are difficult to be detected by the malicious IISP. On the other hand, in the case of the BGP tuning approach, the malicious IISP can easily mimic the BGP tuning implementation performed by the local region AS to blackhole the local region AS traffic.

The services within the malicious IISP can be accessible only with the NAT-based approach. On the other hand, the BGP tuning and tunnel-based approaches do not provide accessibility to such services. Finally, in the case of scalability which refers to the easiness of extending the method to use the entire Internet, all approaches provide high scalability.

Chapter 5

Design, Implementation and Validation of Solution

5.1 Introduction

In the previous chapter, we have discussed different approaches to circumvent the malicious act of IISP to the local region while advertising reachability information to it and to the rest of the Internet. In this chapter, we will discuss the design, implementation, and validation of the tunnel based solution to Internet access denial problem using OPNET.

OPNET [10] is a commercial tool used to design, test, and simulate the performance of a network. Almost all the well known protocols are implemented in OPNET. For the purpose of design, implementation, and validation of the solution we used OP-

NET version 14.0 [38] which provides tunneling protocol support, and BGP protocol support and configuration. An OPNET user can configure autonomous systems, and interior and exterior gateway routing protocols such as EBGP, IBGP, RIP, OSPF, EIGRP, etc. The user can also configure different tunneling protocols such as IPinIP, GRE, GRE with check sum, and IPSec. While simulating the proposed solution, IPSec was not considered, as it is not supported by OPNET's Discrete Event Simulation (DES) that is needed in our simulations.

The following chapter is organised as follows. First, the baseline configuration to serve as a reference point for our tunnel based solution is shown. Second, the configuration for the tunnel based solution is presented. Finally, the validation of the proposed solution is shown and followed by a discussion on the scalability issue of the proposed solution.

5.1.1 Initial Baseline Setup

In order to validate our solution, we need to have a network setup that is suitable for validating our implementation. Figure 5.1 shows the experimental setup, and it also shows the sub-nets for each and every interface in the network. The description of the Figure 5.1 is similar to Figure 1.1 except that a tunnel is created between the router R2 and router R5. We will discuss all the devices that are used in the experimental setup in the following section.

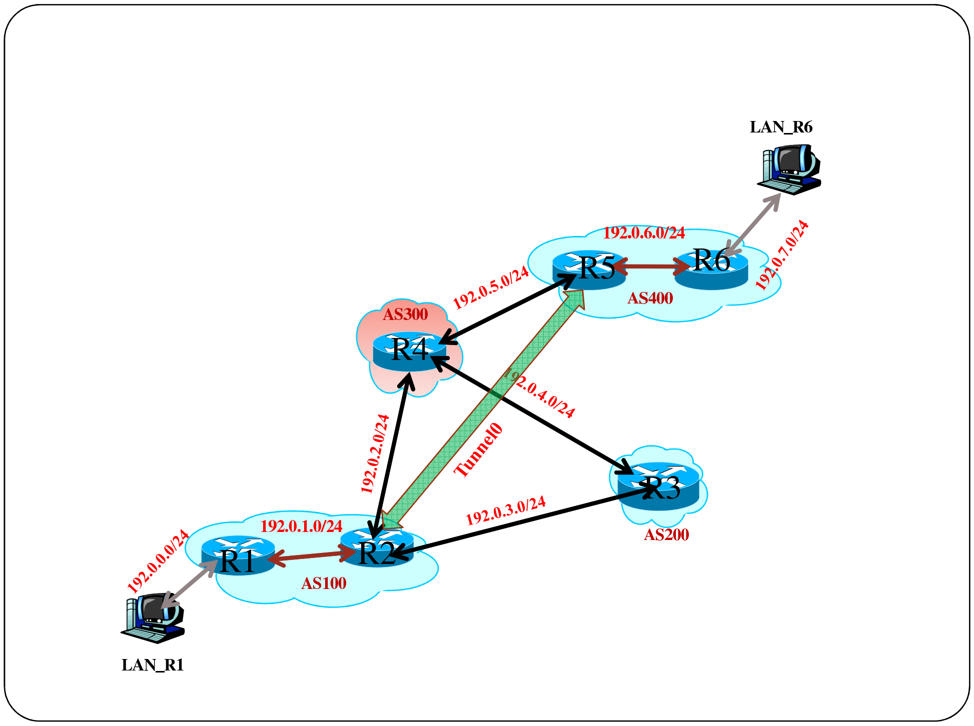


Figure 5.1: Experimental Network Setup: Initial Design of Tunnel-Based Solution with Single Tunnel

5.1.2 Devices Used

In this section, we briefly present the different OPNET devices used for the implementation of the network.

ethernet4_slip8_gtwy_adv router:

This node model represents an IP-based gateway, supports four Ethernet hub interfaces and eight serial line interfaces at selectable data rates. It also supports IP, UDP, RIP, Ethernet (IEEE 802.3), OSPF, and SLIP protocols. The router also supports the tunnel interfaces and there is no restriction on the number of tunnels that can be established. IP packets arriving on any interface are routed to the appropriate output interface based on their destination IP address. The experimental network setup consists of six such routers and are labeled from R1 to R6. These routers are configured to support BGP protocol, and a tunnel was created from the gateway router of the local region to the gateway router of the destination region. In the experimental setup, the local gateway router is R2 and one of the possible destination gateway router is R5. EBGP is configured between the ASes, and IBGP is configured within the ASes.

100BaseT_LAN object:

This is a fast Ethernet LAN in a switched topology. The object contains any number of clients as well as only one server. Client traffic can be directed to the internal server as well as to external servers. Supported applications include: FTP, Email, Database, Custom, Rlogin, Video, HTTP, etc. These applications run over TCP or UDP. The default number of workstations included in the object is 10. The baseline simulation consists of two objects named LAN_R1 and LAN_R6. In this setup, LAN_R6 is configured to serve as a server LAN and is also configured to provide all types of applications. LAN_R1 is configured to be used as a client in the local region and is also configured to request applications such as FTP, Video conferencing, and HTTP for validation of the solution.

Bidirectional PPP_DS1 link:

This is used to connect two nodes running IP protocol using IP datagram packet format over DS1 at a data rate of 1.544 Mbps. In the simulation setup, this type of links is used to connect routers with each other.

Bidirectional 100BaseT link:

This is an Ethernet link that operates at 100Mbps. In the simulation setup, this type of links is used to connect LAN objects to the router.

5.1.3 Mapping Devices Usage to our problem

In our simulation, we assume that AS 100 which contains R1 and R2 is in the local region, and LAN_R1 is one of the clients in the local region. LAN_R1 communicates with LAN_R6 that is located in AS400 to get the desired services. Router R2 acts as the local region provider and it is connected to one of the international service provider routers (i.e., router R4 in AS300). As stated earlier in section 5.1.2, a tunnel have been created between R2 and R5.

5.2 Baseline Simulation Setup

The baseline simulation considers no tunnel establishment in the network. In the baseline setup, the local traffic is routed through the malicious IISP (i.e., AS300) and the AS-path that the local traffic follows is AS300, then AS400 to reach LAN_R6. The complete path from LAN_R1 to LAN_R6 is as follows: LAN_R1, R1, R2, R4, R5, R6, then LAN_R6. This is done only for the purpose of comparing our proposed solution to baseline experimental simulation (i.e. original route).

In Figure 5.2, the X-axis represents the time in seconds and the Y-axis represents the throughput in bits per second. Figure 5.2 shows the throughput between R2 and R4, and between R2 and R3 in both directions. We notice that the traffic flows between R2 and R4 in both directions. On the other hand, traffic does not flow between R2 and R3 in both directions. The reason behind such a behavior is that local region traffic is routed through the original path assuming that IISP (i.e., AS300) is not blocking the Internet access to the local AS (i.e., AS100). This validate the baseline simulation, the baseline performance can be compared to the performance of the proposed solution.

To validate the forwarding settings of the different routers such as the entry point of the tunnel router, the exit point of the tunnel router, the malicious router, and the proper malicious router interface selection for traffic forwarding, we can examine the IP forwarding table of R2, R4, and R5. From the tables we can determine the incoming and outgoing traffic of the local region (i.e., AS100). The IP address of LAN_R6 is 192.0.7.2 and it belongs to the prefix 192.0.7.0/24. We can notice from Table 5.1 that the 'Next Hop Node' to this prefix is through router R4. Hence, the outgoing traffic is validated.

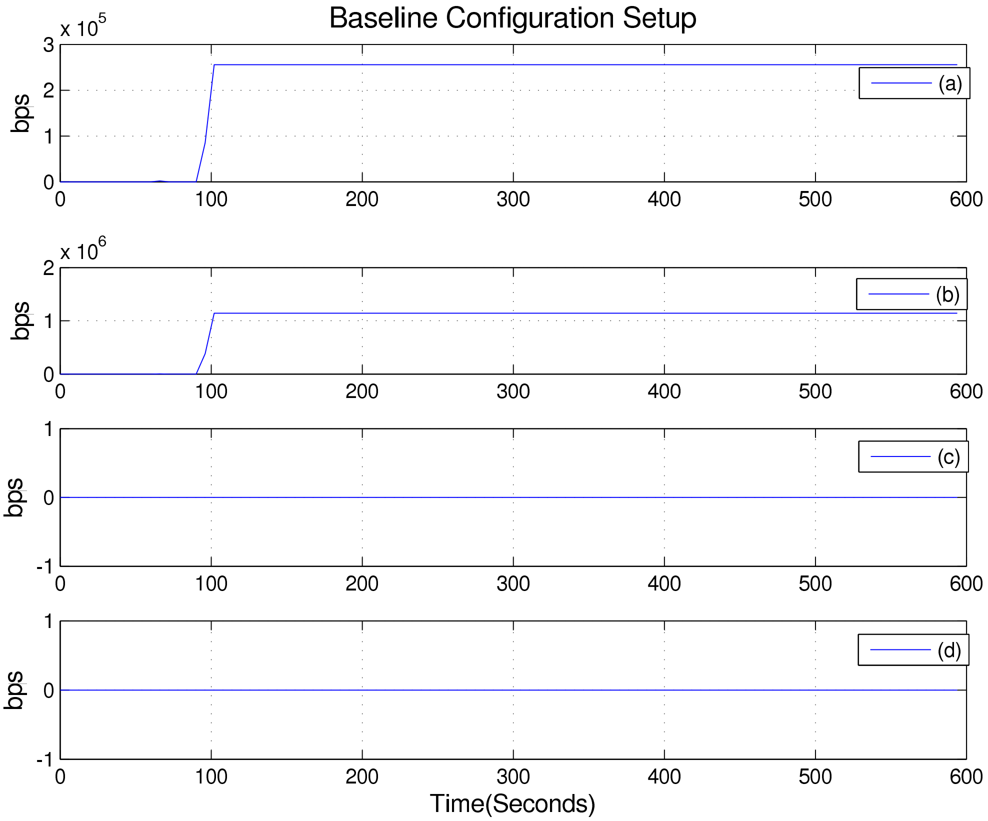


Figure 5.2: Baseline Configuration Setup: Incoming and Outgoing Traffic between R2 and R4 and between R2 and R3 (a) Throughput (bps) R2 to R4 (b) Throughput (bps) R4 to R2 (c) Throughput (bps) R3 to R2 (d) Throughput (bps) R2 to R3

The IP address of LAN_R1 is 192.0.0.2 and it belongs to the prefix 192.0.0.0/24. We can notice from Table 5.2 that the 'Next Hop Node' to this prefix is through router R2. Similarly, if we examine Table 5.3, we can notice that the 'Next Hop Node' to this prefix is through router R4. Therefore, the incoming traffic is validated.

Table 5.1: Baseline Configuration Setup: IP Forwarding Table at Router R2.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	RIP	120	1	192.0.1.1	R1	IF10	N/A	8.423
192.0.1.0/24	Direct	0	0	192.0.1.2	R2	IF10	N/A	0
192.0.2.0/24	Direct	0	0	192.0.2.1	R2	IF4	N/A	0
192.0.3.0/24	Direct	0	0	192.0.3.1	R2	IF11	N/A	0
192.0.4.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.004
192.0.5.0/24	BGP	20	0	192.0.2.2	R4	IF4	N/A	70.004
192.0.6.0/24	BGP	20	0	192.0.2.2	R4	IF4	N/A	70.008
192.0.7.0/24	BGP	20	0	192.0.2.2	R4	IF4	N/A	70.007
192.0.11.0/24	RIP	120	1	192.0.1.1	R1	IF10	N/A	8.423
192.0.12.0/24	Direct	0	0	192.0.12.1	R2	LB0	N/A	0
192.0.13.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.004
192.0.14.0/24	BGP	20	0	192.0.2.2	R4	IF4	N/A	70.004
192.0.15.0/24	BGP	20	0	192.0.2.2	R4	IF4	N/A	70.009
192.0.16.0/24	BGP	20	0	192.0.2.2	R4	IF4	N/A	70.007

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

Table 5.2: Baseline Configuration Setup: IP Forwarding Table at Router R4.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	BGP	20	1	192.0.2.1	R2	IF10	N/A	70.005
192.0.1.0/24	BGP	20	0	192.0.2.1	R2	IF10	N/A	70.004
192.0.2.0/24	Direct	0	0	192.0.2.2	R4	IF10	N/A	0
192.0.3.0/24	BGP	20	0	192.0.2.1	R2	IF10	N/A	70.007
192.0.4.0/24	Direct	0	0	192.0.4.2	R4	IF11	N/A	0
192.0.5.0/24	Direct	0	0	192.0.5.1	R4	IF4	N/A	0
192.0.6.0/24	BGP	20	0	192.0.5.2	R5	IF4	N/A	70.005
192.0.7.0/24	BGP	20	1	192.0.5.2	R5	IF4	N/A	70.004
192.0.11.0/24	BGP	20	1	192.0.2.1	R2	IF10	N/A	70.004
192.0.12.0/24	BGP	20	0	192.0.2.1	R2	IF10	N/A	70.006
192.0.13.0/24	BGP	20	0	192.0.4.1	R3	IF11	N/A	70.004
192.0.14.0/24	Direct	0	0	192.0.14.1	R4	LB0	N/A	0
192.0.15.0/24	BGP	20	0	192.0.5.2	R5	IF4	N/A	70.006
192.0.16.0/24	BGP	20	1	192.0.5.2	R5	IF4	N/A	70.004

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

Table 5.3: Baseline Configuration Setup: IP Forwarding Table at Router R5.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.008
192.0.1.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.007
192.0.2.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.005
192.0.3.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	100.005
192.0.4.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.005
192.0.5.0/24	Direct	0	0	192.0.5.2	R5	IF10	N/A	0
192.0.6.0/24	Direct	0	0	192.0.6.1	R5	IF11	N/A	0
192.0.7.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	5.975
192.0.11.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.007
192.0.12.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.009
192.0.13.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.006
192.0.14.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.004
192.0.15.0/24	Direct	0	0	192.0.15.1	R5	LB0	N/A	0
192.0.16.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	5.975

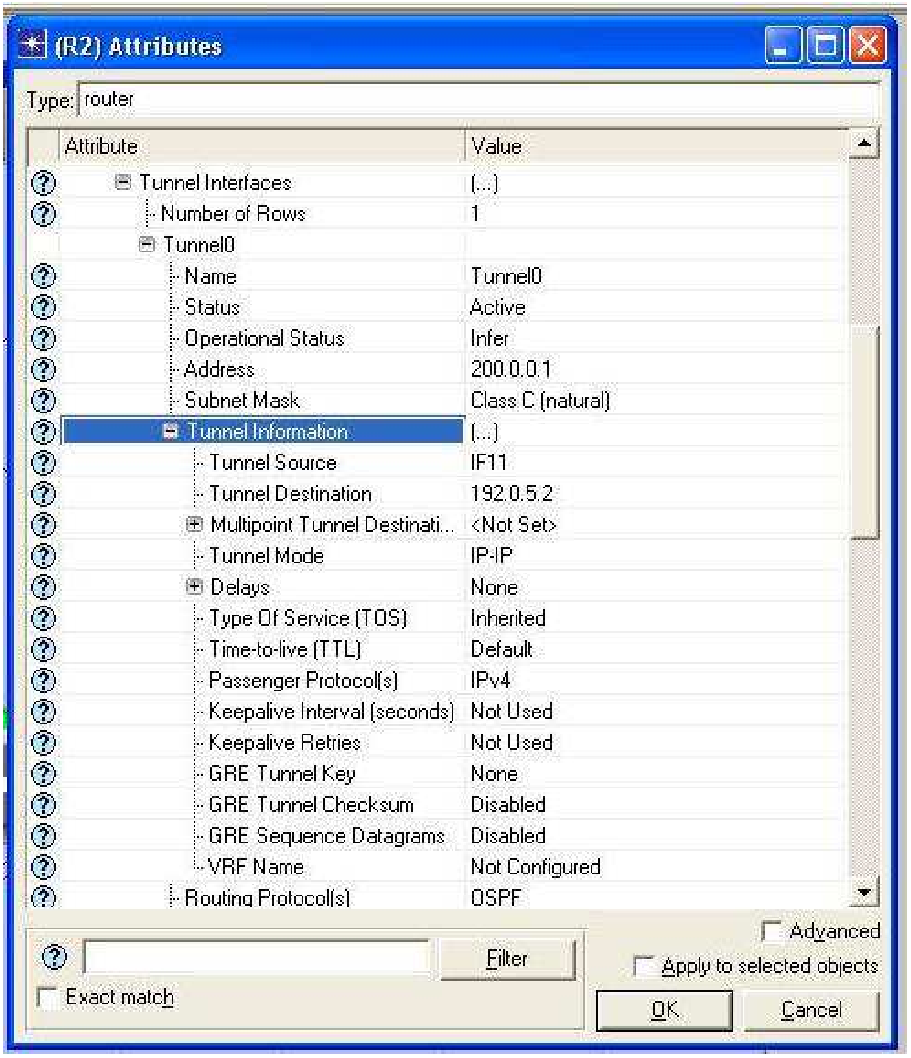
(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

5.3 Proposed Simulation Setup

In this setup, the same baseline network for simulation was used in addition to creating a tunnel between R2 and R5 that passes through R3. The non-blocked IP address that is provided by the neighbouring AS (i.e., AS200) was used to create the tunnel. Thus, with the help of a neighbouring AS, a tunnel that passes through the malicious ISP (i.e., AS300) was created. The use of a non-blocked IP address will prevent the malicious router (i.e., router R4) from dropping incoming and outgoing local region traffic.

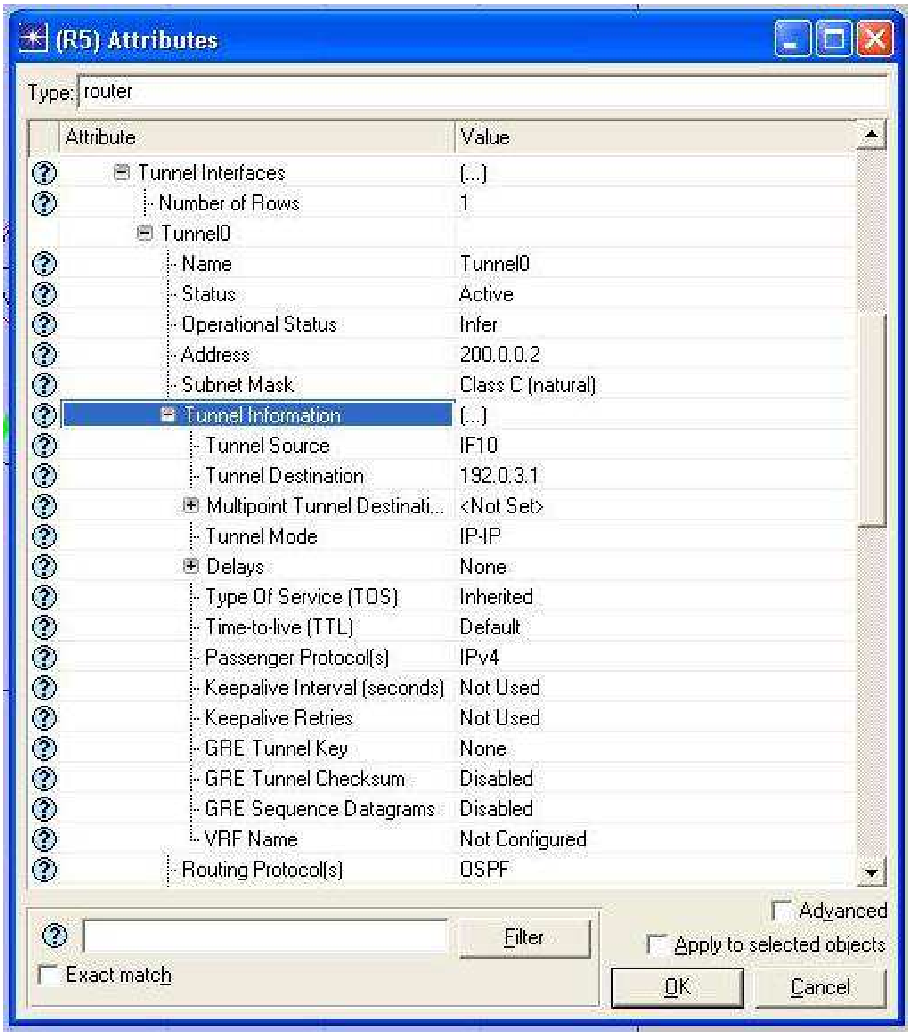
To create a tunnel, we need a prefix to be used for the tunnel interface. In the simulation, the chosen prefix belongs to subnet 200.0.0.0/24. The tunnel starting point IP address is 200.0.0.1, the tunnel ending point IP address is 200.0.0.2, and the tunnel name is Tunnel0. The starting point of the tunnel is interface IF11 of router R2, and its non-tunnel IP address is 192.0.3.1. All settings associated with configuring the starting point of the tunnel are is shown in Figure 5.3. The ending point of the tunnel is interface IF10 of router R5, and its non-tunnel IP address is 192.0.5.2. All settings associated with configuring the ending point of the tunnel are shown in Figure 5.4. The routing protocol used for the tunnel interface is OSPF, the other routing protocol that can be used for the tunnel interface is Enhanced Interior Gateway Routing Protocol (EIGRP). Only OSPF is considered as a tunnel interface routing protocol in the simulation because EIGRP is a CISCO proprietary routing protocol and the target of our solution is to use standard protocols only.

Figure 5.5 shows the IP tunnel traffic received and sent in bits per seconds on routers R2 and R5. To validate that the proposed solution is setup to forward the traffic properly through the tunnel, we can examine the IP forwarding table on both routers R2 and R5. Such tables are presented in Table 5.4 and Table 5.5. From the tables we can determine that the incoming and the outgoing traffic on router R2 and router R5, respectively, use Tunnel0. This validates the proper setup for the tunnel.



Attribute	Value
⊟ Tunnel Interfaces	(...)
Number of Rows	1
⊟ Tunnel0	
Name	Tunnel0
Status	Active
Operational Status	Infer
Address	200.0.0.1
Subnet Mask	Class C (natural)
⊟ Tunnel Information	(...)
Tunnel Source	IF11
Tunnel Destination	192.0.5.2
⊞ Multipoint Tunnel Destinati...	<Not Set>
Tunnel Mode	IP-IP
⊞ Delays	None
Type Of Service (TOS)	Inherited
Time-to-live (TTL)	Default
Passenger Protocol(s)	IPv4
Keepalive Interval (seconds)	Not Used
Keepalive Retries	Not Used
GRE Tunnel Key	None
GRE Tunnel Checksum	Disabled
GRE Sequence Datagrams	Disabled
VRF Name	Not Configured
Routing Protocol(s)	OSPF

Figure 5.3: Tunnel Configuration on Router R2



Type: router

Attribute	Value
⊟ Tunnel Interfaces	(...)
Number of Rows	1
⊟ Tunnel0	
Name	Tunnel0
Status	Active
Operational Status	Infer
Address	200.0.0.2
Subnet Mask	Class C (natural)
⊟ Tunnel Information	(...)
Tunnel Source	IF10
Tunnel Destination	192.0.3.1
⊞ Multipoint Tunnel Destinati...	<Not Set>
Tunnel Mode	IP-IP
⊞ Delays	None
Type Of Service (TOS)	Inherited
Time-to-live (TTL)	Default
Passenger Protocol(s)	IPv4
Keepalive Interval (seconds)	Not Used
Keepalive Retries	Not Used
GRE Tunnel Key	None
GRE Tunnel Checksum	Disabled
GRE Sequence Datagrams	Disabled
VRF Name	Not Configured
Routing Protocol(s)	OSPF

Figure 5.4: Tunnel Configuration on Router R5

Table 5.4: Tunnel Configuration: IP Forwarding Table at Router R2.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	RIP	120	1	192.0.1.1	R1	IF10	N/A	9.713
192.0.1.0/24	Direct	0	0	192.0.1.2	R2	IF10	N/A	0
192.0.2.0/24	Direct	0	0	192.0.2.1	R2	IF4	N/A	0
192.0.3.0/24	Direct	0	0	192.0.3.1	R2	IF11	N/A	0
192.0.4.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.004
192.0.5.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.006
192.0.6.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	94.125
192.0.7.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	94.125
192.0.11.0/24	RIP	120	1	192.0.1.1	R1	IF10	N/A	9.713
192.0.12.0/24	Direct	0	0	192.0.12.1	R2	LB0	N/A	0
192.0.13.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.004
192.0.14.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.006
192.0.15.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.009
192.0.16.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	94.125
200.0.0.0/24	Direct	0	0	200.0.0.1	R2	Tunnel0	N/A	0

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

Table 5.5: Tunnel Configuration: IP Forwarding Table at Router R5.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	OSPF 1	5	20	200.0.0.1	R2	Tunnel0	N/A	94.126
192.0.1.0/24	OSPF 1	5	20	200.0.0.1	R2	Tunnel0	N/A	94.126
192.0.2.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.005
192.0.3.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.007
192.0.4.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.005
192.0.5.0/24	Direct	0	0	192.0.5.2	R5	IF10	N/A	0
192.0.6.0/24	Direct	0	0	192.0.6.1	R5	IF11	N/A	0
192.0.7.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	6.976
192.0.11.0/24	OSPF 1	5	20	200.0.0.1	R2	Tunnel0	N/A	94.126
192.0.12.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.009
192.0.13.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.006
192.0.14.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.004
192.0.15.0/24	Direct	0	0	192.0.15.1	R5	LB0	N/A	0
192.0.16.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	6.976
200.0.0.0/24	Direct	0	0	200.0.0.2	R5	Tunnel0	N/A	0

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

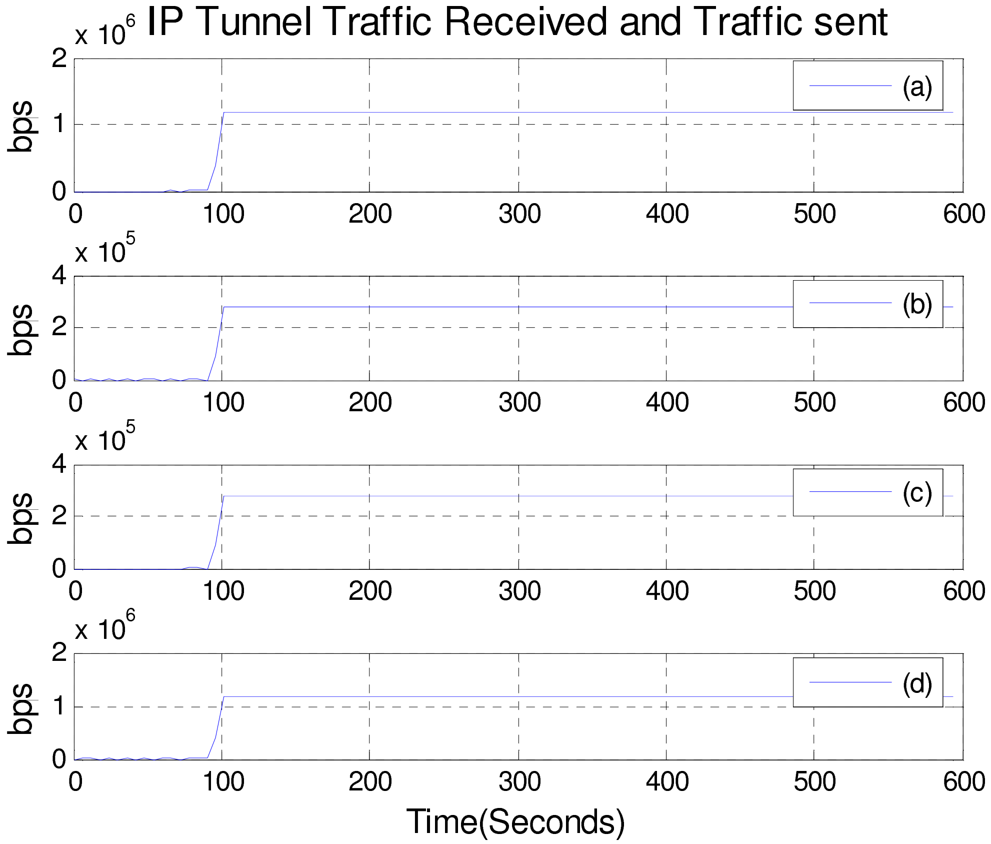


Figure 5.5: IP Tunnel Traffic Received and Sent (bits per seconds) on Tunnel0 at Router R2 and R5 (a) Tunnel Traffic Received (bps) at Router R2 (b) Tunnel Traffic Sent (bps) at Router R2 (c) Tunnel Traffic Received (bps) at Router R5 (d) Tunnel Traffic Sent (bps) at Router R5

5.4 Scalability Issue of the Propose Solution

In this section, we investigate the scalability issues related to the proposed solution. The term scalability refers to the easiness of extending the proposed method to access any part of the Internet. Different scalability designs are discussed in terms of use of multiple tunnels, using existing tunnel to route the traffic to neighbours of the tunneled AS, load balancing, and increasing the number of users in the LAN.

5.4.1 Extending Tunnel-Based Solution Design to Multiple Tunnels

There is no real limit to the number of tunnels that can be created. We can have several tunnel interfaces, as long as we do not use the same combination of source, destination, and tunnel mode more than once. For the validation purpose, we created another tunnel interface (i.e. Tunnel1) between router R2 (i.e., AS100) and router R8 (i.e., AS600), shown in Figure 5.6. We can verify the creation of multiple tunnels from the IP forwarding table of router R2, as shown in the Table 5.6. We can also verify this by examining the IP forwarding table for router R8 that is shown in Table 5.7. This confirms the creation of the second tunnel that is terminated at router R8.

5.4.2 Extending Tunnel-Based Solution Design to Multiple ASes

It is desired that the proposed solution allows the affected region to reach ASes that are not necessarily the immediate neighbors of the malicious AS. However, as it will be noted later in this section, such extension requires manual configurations at the proper routers. Accordingly, one of the concerns regarding the proposed solution that should be looked into is how frequent are the changes in the paths established between one AS and another. If the change is too frequent, then the

Table 5.6: Multiple ASes Tunnel Configuration: IP Forwarding Table at Router R2.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	RIP	120	1	192.0.1.1	R1	IF10	N/A	9.546
192.0.1.0/24	Direct	0	0	192.0.1.2	R2	IF10	N/A	0
192.0.2.0/24	Direct	0	0	192.0.2.1	R2	IF4	N/A	0
192.0.3.0/24	Direct	0	0	192.0.3.1	R2	IF11	N/A	0
192.0.4.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.005
192.0.5.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.007
192.0.6.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.7.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.8.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.9.0/24	OSPF 1	5	1	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.10.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.007
192.0.11.0/24	RIP	120	1	192.0.1.1	R1	IF10	N/A	9.546
192.0.12.0/24	Direct	0	0	192.0.12.1	R2	LB0	N/A	0
192.0.13.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.004
192.0.14.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.006
192.0.15.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.16.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.17.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.015
192.0.18.0/24	OSPF 1	5	20	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.19.0/24	OSPF 1	5	20	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.20.0/24	OSPF 1	5	20	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.21.0/24	OSPF 1	5	20	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.22.0/24	OSPF 1	5	20	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.23.0/24	OSPF 1	5	1	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.24.0/24	OSPF 1	5	1	200.0.0.2	R5	Tunnel0	N/A	86.004
192.0.25.0/24	OSPF 1	5	20	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.26.0/24	OSPF 1	5	20	200.0.1.2	R8	Tunnel01	N/A	86.004
192.0.27.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.015
192.0.28.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.097
192.0.29.0/24	BGP	20	0	192.0.3.2	R3	IF11	N/A	70.004
200.0.0.0/24	Direct	0	0	200.0.0.1	R2	Tunnel0	N/A	0
200.0.1.0/24	Direct	0	0	200.0.1.1	R2	Tunnel01	N/A	0

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

Table 5.7: Multiple ASes Tunnel Configuration: IP Forwarding Table at Router R8.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	OSPF 1	5	20	200.0.1.1	R2	Tunnel01	N/A	86.002
192.0.1.0/24	OSPF 1	5	20	200.0.1.1	R2	Tunnel01	N/A	86.002
192.0.2.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.006
192.0.3.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.009
192.0.4.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.005
192.0.5.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.005
192.0.6.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.011
192.0.7.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.01
192.0.8.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.01
192.0.9.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.014
192.0.10.0/24	Direct	0	0	192.0.10.2	R8	IF10	N/A	0
192.0.11.0/24	OSPF 1	5	20	200.0.1.1	R2	Tunnel01	N/A	86.002
192.0.12.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.014
192.0.13.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.008
192.0.14.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.004
192.0.15.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.009
192.0.16.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.008
192.0.17.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.013
192.0.18.0/24	IBGP	200	0	192.0.26.1	R9	Unresolved	N/A	70.136
192.0.19.0/24	RIP	120	1	192.0.20.2	R9	IF11	N/A	9.444
192.0.20.0/24	Direct	0	0	192.0.20.1	R8	IF11	N/A	0
192.0.21.0/24	RIP	120	1	192.0.20.2	R9	IF11	N/A	9.444
192.0.22.0/24	RIP	120	1	192.0.20.2	R9	IF11	N/A	9.444
192.0.23.0/24	IBGP	200	0	192.0.26.1	R9	Unresolved	N/A	70.008
192.0.24.0/24	IBGP	200	0	192.0.26.1	R9	Unresolved	N/A	70.109
192.0.25.0/24	Direct	0	0	192.0.25.1	R8	LB0	N/A	0
192.0.26.0/24	RIP	120	1	192.0.20.2	R9	IF11	N/A	9.444
192.0.27.0/24	IBGP	200	0	192.0.26.1	R9	Unresolved	N/A	70.007
192.0.28.0/24	IBGP	200	0	192.0.26.1	R9	Unresolved	N/A	70.095
192.0.29.0/24	BGP	20	0	192.0.10.1	R4	IF10	N/A	70.007
200.0.0.0/24	OSPF 1	5	22222	200.0.1.1	R2	Tunnel01	N/A	86.002
200.0.1.0/24	Direct	0	0	200.0.1.2	R8	Tunnel01	N/A	0

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

proposed solution becomes impractical. Thus, before discussing the extension of the proposed solution to multiple ASes, we will first examine how steady the primary paths are from one AS to others over a long period of time by considering one of the mandatory BGP attributes, i.e., BGP AS-PATH.

BGP AS-PATH Characteristics

Zhao et al. [39] examined and studied BGP AS-PATH characteristics over nearly one year. To analize BGP routing updates, data was collected from Route Views [40] from March 2002 to December 2002. During this ten month study, Zhao et al. observed 236,395 prefixes. Their focus was only on the prefixes which were announced longer than 40% of the total time that they were examining. In their study, they focused only on the path behavior of long lasting prefixes, thus they were interested in long-term AS path behavior. Some of the prefixes were falsely announced due to router misconfigurations which were expected to be corrected shortly. And, some of the falsely announced prefixes were due to traffic engineering at the BGP level that may have introduced some short lived prefixes in the global routing table. After filtering, they obtained only 116,544 prefixes. In the study of path change patterns, they assumed that initially a prefix was the primary path. Then at some time, another path may replace the primary path, and they counted time intervals for how long it takes to replace the new path with the primary path. The authors discovered that about 90% of the prefixes do not use an alternative path continuously for longer than 10 minutes. Hence, the authors conclude that primary paths do not change over a long period of time and most of the primary paths are very stable.

Multiple ASes

To make the solution scalable, we extend our initial design of the tunnel-based solution to reach multiple ASes from the affected AS as shown in Figure 5.6. In this setup, we utilize the existing tunnels established by the affected local region to send and receive traffic to and from neighbouring ASes of the end point of the tunnels. For example, from Figure 5.6, if the local region AS (i.e., AS100) wants to access some services that are located at AS500, then AS100 can utilize the existing tunnel established between R2 and R5 to send or receive the traffic to or from R5. Then, the normal routing protocols can be used to deliver the traffic from/to R5 to/from AS500.

To extend the reachability to other ASes through a tunnel route, redistribution must be used. As demonstrated in the previous section, we note that the manual redistribution of routing information is acceptable since most of the primary paths are stable over a long period of time. The purpose of the route redistribution is to propagate routes learned using one protocol into another routing protocol. For example, network 192.0.9.0/24 on LAN_R7 in the simulated network is populated as an IBGP Route in the BGP forwarding table of Router R5 as shown in Table 5.8.

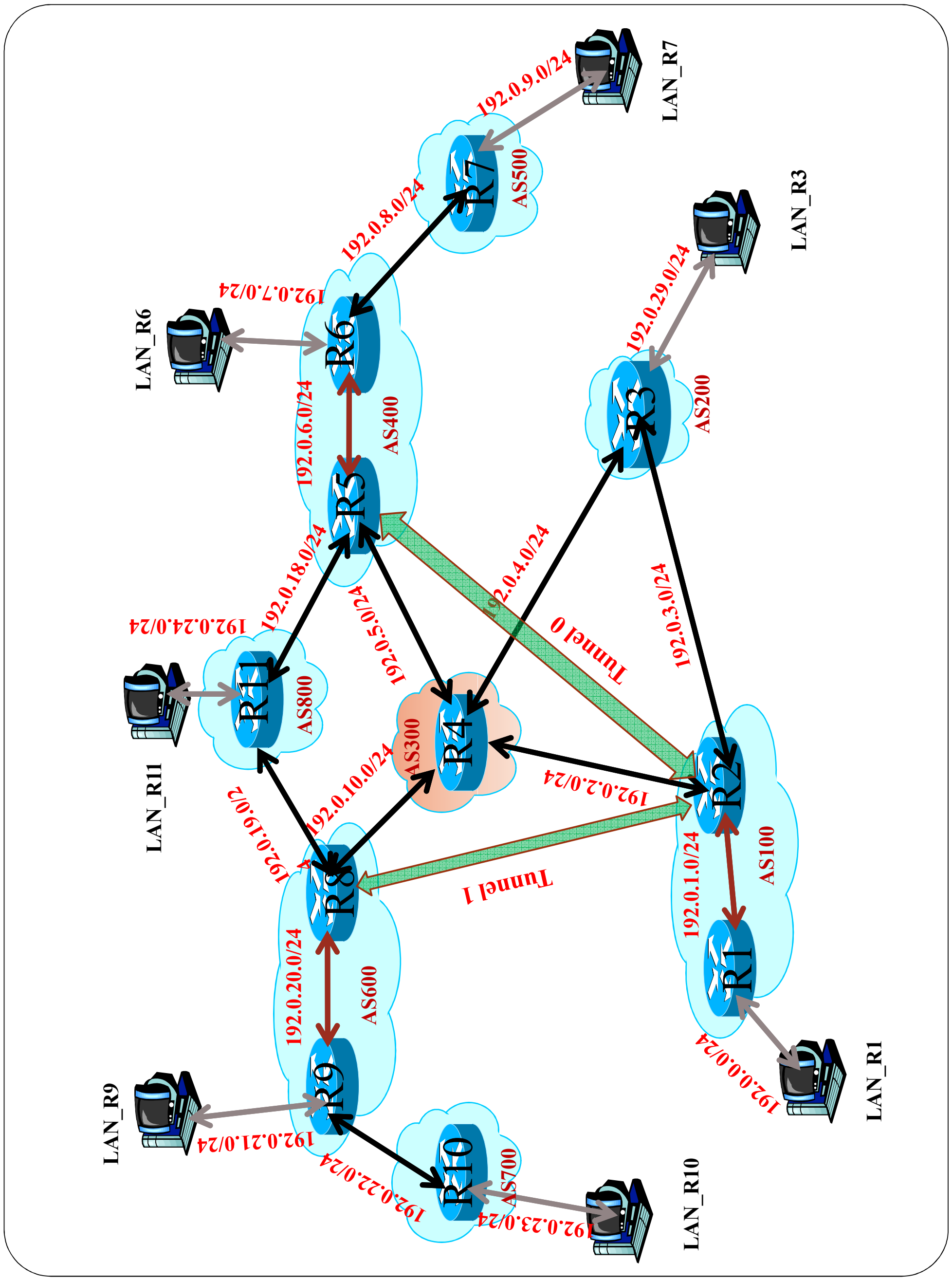


Figure 5.6: Extended Design for Tunnel Based Solution Using Multiple Tunnels and Multiple ASes

Since the prefix is known to R5 through IBGP, and since it is desired to make the same prefix reachable by R2 through the tunnel established between R2 and R5 and that uses OSPF, then the prefix must be redistributed at R5. The route redistribution value at R5 must be changed to both IBGP and EBGP so that the desired prefix gets redistributed into the tunnel through the use of the OSPF protocol. The steps on how to configure the route redistribution in OPNET are shown in Appendix A.1.

To verify the route redistribution, we can examine the IP forwarding tables of routers R2 and R5. From the routing table of router R2 shown in Table 5.6, we can determine that the local region routes traffic destined to prefix 192.0.9.0/24 through Tunnel0. In Table 5.6, we can also determine that the local region traffic destined to prefix 192.0.29.0/24 will not utilize the tunnel and, instead, will follow the normal BGP route as the tunnel is needed only if the traffic is routed through the malicious AS (i.e., AS300). Similarly, if we examine the IP forwarding table of R5 shown in Table 5.9, we can determine that Tunnel0 is used to route the traffic to the local region (i.e., AS100).

Note that in Table 5.6 and Table 5.9, we observe that some of the values of the Outgoing Interface are set to “Unresolved”. In such cases, BGP is unable to resolve the next hop and the outgoing interface for that specific prefix. To explain the reason behind such a behaviour, we note that when a BGP router receives a route, the next hop address advertised with it may not be directly connected. Under such

Table 5.8: BGP Forwarding Table at Router R5.

a	b	c	d	e	f	g	h	i	j
192.0.0.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200 100	Incomplete
192.0.1.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200 100	Incomplete
192.0.2.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300	Incomplete
192.0.3.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200	Incomplete
192.0.4.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300	Incomplete
192.0.5.0/24	IBGP	192.0.16.1	R6	IF11	1	100	0		Incomplete
192.0.6.0/24	Direct	192.0.6.1	R5	IF11	0	100	32768		Incomplete
192.0.7.0/24	RIP	192.0.6.2	R6	IF11	1	100	32768		Incomplete
192.0.8.0/24	RIP	192.0.6.2	R6	IF11	1	100	32768		Incomplete
192.0.9.0/24	IBGP	192.0.16.1	R6	IF11	0	100	0	500	Incomplete
192.0.10.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300	Incomplete
192.0.11.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200 100	Incomplete
192.0.12.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200 100	Incomplete
192.0.13.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200	Incomplete
192.0.14.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300	Incomplete
192.0.15.0/24	Direct	192.0.15.1	R5	LB0	0	100	32768		Incomplete
192.0.16.0/24	RIP	192.0.6.2	R6	IF11	1	100	32768		Incomplete
192.0.17.0/24	IBGP	192.0.16.1	R6	IF11	0	100	0	500	Incomplete
192.0.18.0/24	RIP	192.0.6.2	R6	IF11	1	100	32768		Incomplete
192.0.19.0/24	IBGP	192.0.16.1	R6	IF11	0	100	0	800	Incomplete
192.0.20.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600	Incomplete
192.0.21.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600	Incomplete
192.0.22.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600	Incomplete
192.0.23.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600 700	Incomplete
192.0.24.0/24	IBGP	192.0.16.1	R6	IF11	0	100	0	800	Incomplete
192.0.25.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600	Incomplete
192.0.26.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600	Incomplete
192.0.27.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600 700	Incomplete
192.0.28.0/24	IBGP	192.0.16.1	R6	IF11	0	100	0	800	Incomplete
192.0.29.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 200	Incomplete
200.0.0.0/24	IBGP	192.0.16.1	R6	IF11	1	100	0		Incomplete
200.0.1.0/24	EBGP	192.0.5.1	R4	IF10	0	100	0	300 600	Incomplete

(a) Destination (b) Source Protocol (c) Next Hop Address (d) Next Hop Node (e) Outgoing Interface (f) MED (g) Local preference (h) Weight (i) AS path (j) Origin

Table 5.9: Multiple ASes Tunnel Configuration: IP Forwarding Table at Router R5.

a	b	c	d	e	f	g	h	i
192.0.0.0/24	OSPF 1	5	20	200.0.0.1	R2	Tunnel0	N/A	77.832
192.0.1.0/24	OSPF 1	5	20	200.0.0.1	R2	Tunnel0	N/A	77.832
192.0.2.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.006
192.0.3.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.009
192.0.4.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.005
192.0.5.0/24	Direct	0	0	192.0.5.2	R5	IF10	N/A	0
192.0.6.0/24	Direct	0	0	192.0.6.1	R5	IF11	N/A	0
192.0.7.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	9.59
192.0.8.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	9.59
192.0.9.0/24	IBGP	200	0	192.0.16.1	R4	Unresolved	N/A	70.007
192.0.10.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.004
192.0.11.0/24	OSPF 1	5	20	200.0.0.1	R2	Tunnel0	N/A	77.832
192.0.12.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.014
192.0.13.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.008
192.0.14.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.004
192.0.15.0/24	Direct	0	0	192.0.15.1	R5	LB0	N/A	0
192.0.16.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	9.59
192.0.17.0/24	IBGP	200	0	192.0.16.1	R6	Unresolved	N/A	70.007
192.0.18.0/24	RIP	120	1	192.0.6.2	R6	IF11	N/A	9.59
192.0.19.0/24	IBGP	200	0	192.0.16.1	R6	Unresolved	N/A	70.123
192.0.20.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.01
192.0.21.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.01
192.0.22.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.009
192.0.23.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.014
192.0.24.0/24	IBGP	200	0	192.0.16.1	R6	Unresolved	N/A	70.109
192.0.25.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.008
192.0.26.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.007
192.0.27.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.013
192.0.28.0/24	IBGP	200	0	192.0.16.1	R6	Unresolved	N/A	70.095
192.0.29.0/24	BGP	20	0	192.0.5.1	R4	IF10	N/A	70.007
200.0.0.0/24	Direct	0	0	200.0.0.2	R5	Tunnel0	N/A	0
200.0.1.0/24	OSPF 1	5	22222	200.0.0.1	R2	Tunnel0	N/A	77.832

(a) Destination (b) Source Protocol (c) Route Preference (d) Metric
(e) Next Hop Address (f) Next Hop Node (g)Outgoing Interface
(h) Outgoing LSP (i) Insertion Time (sec)

a scenario, BGP does what is known as recursive lookup. If the next hop address does not exist in the router's routing table, it will then be shown as “Unresolved”.

5.4.3 Tunnel-Based Solution Design and Load Balancing Using a Pool of Public IP Addresses

Another tunnel-based solution scalability issue to be considered is the processing requirement on the gateway router. At the gateway router, every packet is sent or received through the tunnel, and must go through the encapsulation and decapsulation process. This process increases the processing time at the gateway router. However, through the use of multiple gateway routers and pools of public IP addresses, the load will be distributed on the gateway routers. A possible design for load balancing is shown in Figure 5.7. In this design, the tunnels are distributed among the gateway routers, and can potentially improve the performance.

5.4.4 Extending Tunnel-Based Solution Design with the Increase in the Number of Users

To validate our solution work for large number of users, we increase the number of users from the default value of 10 users to 40 users as shown in Figure 5.8. Figure 5.9 shows the traffic between R2 and R4, and R2 and R3 in both directions. We notice that the traffic goes through R2 and R4 in both directions and there is no traffic

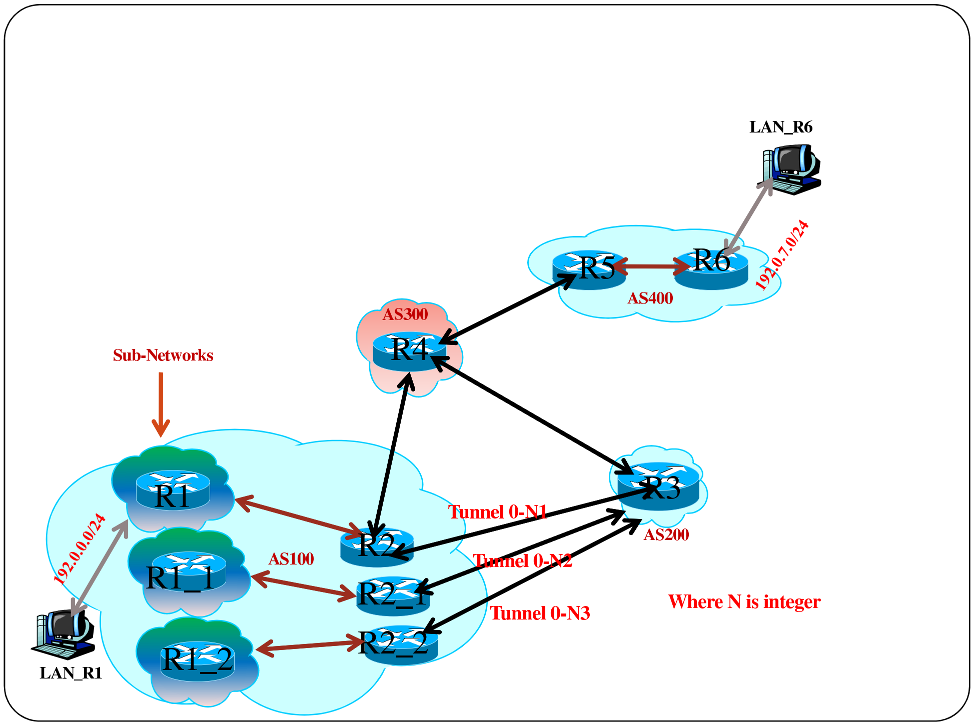
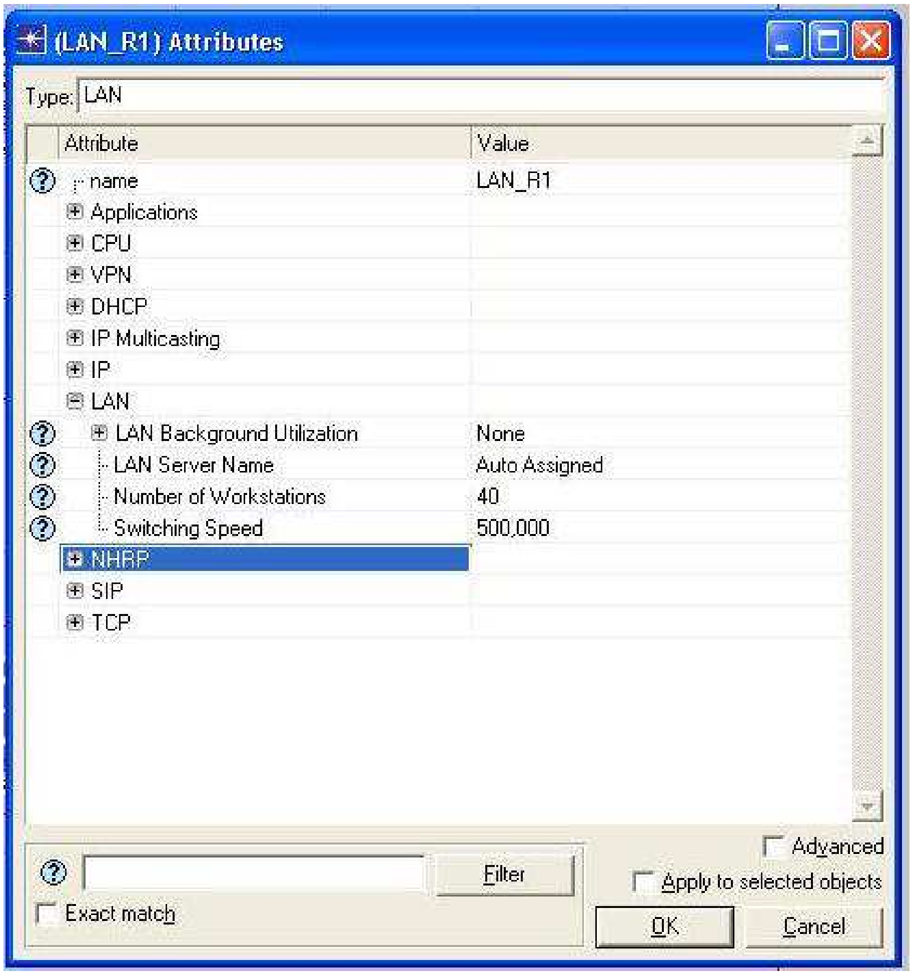


Figure 5.7: Tunnel-Based Solution Design for the Load Balancing

flowing between R2 and R3 in both directions as explained earlier. Figure 5.10 shows the IP tunnel traffic received and Sent in bits per seconds on router R2 and router R5 when the number of users is increased to 40. The proposed solution is valid for any number of users, as long as the data rate of the links of the network is increased so that large amount of traffic can be sent and/or received on these links.


(LAN_R1) Attributes

Type: LAN

Attribute	Value
name	LAN_R1
⊞ Applications	
⊞ CPU	
⊞ VPN	
⊞ DHCP	
⊞ IP Multicasting	
⊞ IP	
⊟ LAN	
⊞ LAN Background Utilization	None
LAN Server Name	Auto Assigned
Number of Workstations	40
Switching Speed	500,000
⊞ NHRP	
⊞ SIP	
⊞ TCP	

Figure 5.8: Configuration for Number of workstation or Users

5.5 Summary

In this chapter, we have discussed the design, implementation, and validation of the proposed solution. We also discussed the scalability issues related to the proposed solution in terms of the support of multiple tunnels, multiple ASes, load balancing, and the increase of the number of users.

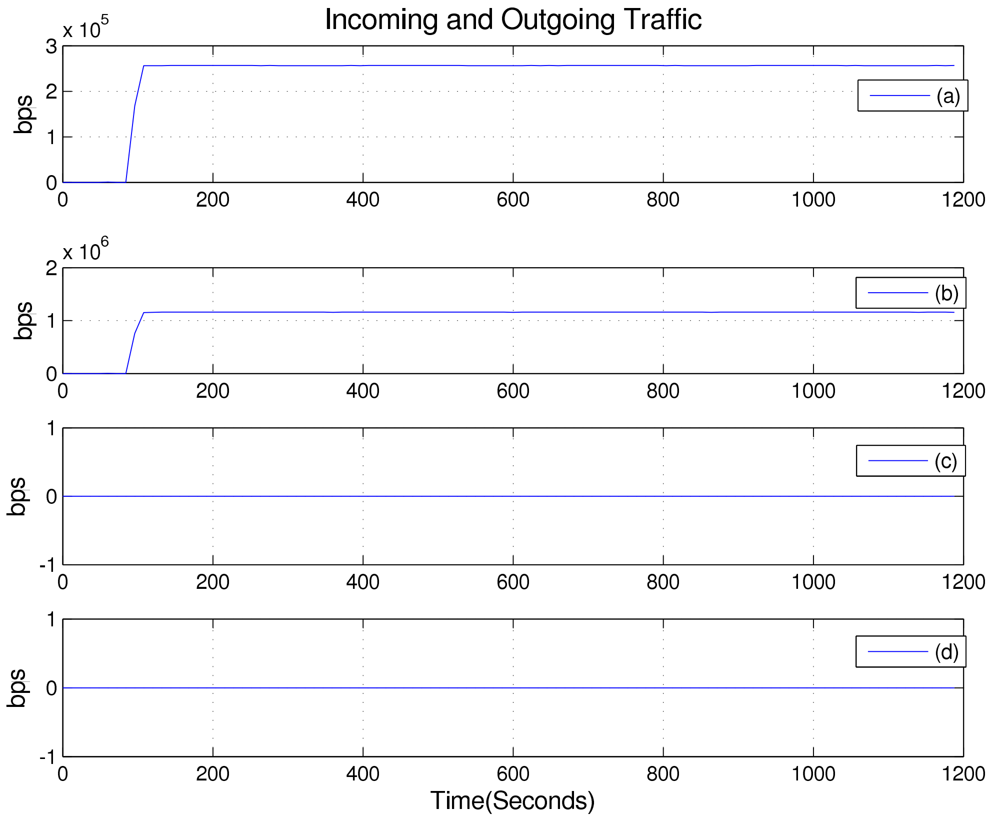


Figure 5.9: Incoming and Outgoing Traffic between R2 and R4 and between R2 and R3 When Number of Users or Workstations are Increase to 40 (a) Throughput (bps) R2 to R4 (b) Throughput (bps) R4 to R2 (c) Throughput (bps) R3 to R2 (d) Throughput (bps) R2 to R3

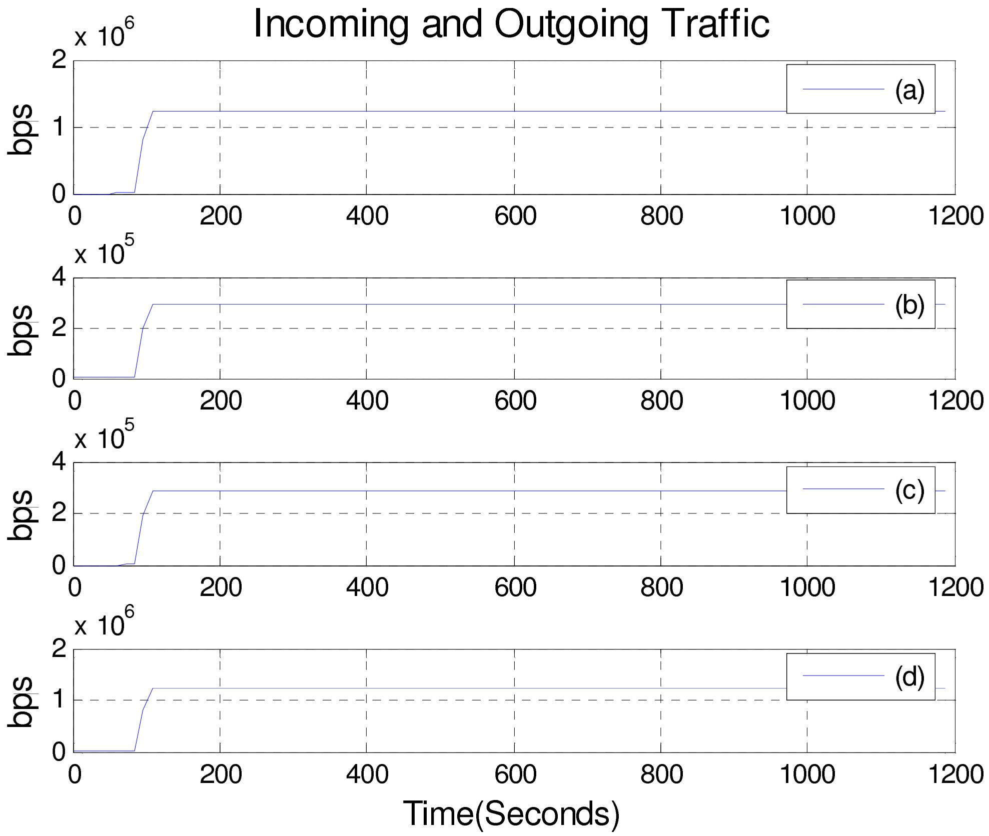


Figure 5.10: IP Tunnel Traffic Received and Sent (bits per seconds) on Tunnel0 at Router R2 and R5 When Number of Users or Workstations are Increase to 40 (a) Tunnel Traffic Received (bps) at Router R2 (b) Tunnel Traffic Sent (bps) at Router R2 (c) Tunnel Traffic Received (bps) at Router R5 (d) Tunnel Traffic Sent (bps) at Router R5

Chapter 6

Performance Evaluation of the Tunnel-Based Solution

6.1 Introduction

In the previous chapter, we have discussed the design, implementation, and validation of the tunnel-based solution to the Internet denial problem using OPNET. In this chapter, we evaluate the performance of the tunnel-based solution as the use of tunneling protocols introduces extra overhead to the packets entering the tunnel as compared to normal packets. The amount of extra overhead bytes that will be added depends on the type of tunneling protocol used. The tunneling protocols IP-in-IP, GRE, and GRE with check sum add 20, 24, and 28 of extra overhead bytes, respectively, to each packet entering the tunnel. Thus, in this chapter, we compare the performance of the tunnel-based solution to the normal operation (i.e., no malicious activity by the IISP) in terms of end-to-end delay, traffic overhead, and packet drop under different types of traffic and different network loads (number of users). The performance evaluation is conducted by means of OPNET simulations.

6.2 Simulation Setup

The baseline network model shown in Figure 5.1 is used for the performance evaluation. The total network simulation duration is set to 600 seconds. Moreover, several simulation scenarios are considered by varying the type of tunneling protocol used, the type of traffic, and the network load. Furthermore, each simulation scenario is simulated for 5 different seeds/runs and the average of the 5 results is collected. The performance statistics collected are the end-to-end delay, traffic overhead, and packet drop.

The solution is evaluated against different types of traffic generated by well known applications such as File Transfer Protocol (FTP), Video conferencing, and Hyper Text Transfer Protocol (HTTP). Both FTP and HTTP run over the TCP protocol, whereas Video conferencing runs over the UDP protocol. In the simulations conducted, we used HTTP1.1 (i.e., persistent HTTP) as it is the most commonly used version of HTTP nowadays. Each simulation is run for three different traffic loads: 25% of the available link bandwidth (about 386kbps), 50% of the available link bandwidth (about 772kbps) and 75% of the available link bandwidth (about 1158kbps).

The rest of the chapter is organized as follows. First, we discuss in details the packet fragmentation that can be caused by the addition of extra overhead bytes to each packet entering the tunnel. Second, the performance of the solution is evaluated under different network loads, traffic types, and tunneling protocols. Finally, the chapter is concluded by discussion and recommendations regarding the proposed solution.

6.3 Tunnel Fragmentation

Each path that extends between two end systems allows a maximum number of bytes to be transmitted per packet, and is considered to be the Maximum Transmission Unit (MTU) for that path. Accordingly, tunnel fragmentation occurs when the size of the encapsulated packet (i.e., original packet plus the extra tunnel overhead bytes) exceeds the MTU of the tunnel path. Under such scenario, the encapsulated packet will be divided (fragmented) into two smaller packets. The first packet is equal to the size of the MTU and the other packet size is typically smaller than the MTU. The two packets are then forwarded to the next router.

It should be noted that the configuration for the video conferencing application (i.e., UDP traffic) in the simulation uses a constant file size measured in bytes, and the file size is smaller than the MTU. In the case of video conferencing, the number of packets used is the same for all the tunneling protocol scenarios and also for the no tunnel scenario, because the requested file is smaller than the MTU. Thus, the only difference between the different video conferencing tunneling scenarios is in the size of each packet which varies in accordance with the tunneling protocol used.

On the other hand, the configuration for FTP and HTTP (i.e., TCP traffic) in the simulation also uses a constant file size, but the file size measured in bytes for no tunnel and tunnel is larger than the MTU to simulate typical file sizes under such applications. Before entering the tunnel, the server divides the file into packets equal to the size of the MTU. Once the packet is sent by the server and enters the tunnel, a tunnel overhead is added to the packet and the size of the new packet is larger than the MTU. As a result, the new packet is fragmented into two packets and forwarded to the next router.

In summary, we can conclude that a packet will be fragmented into two packets only if the size the packet after the addition of the tunnel header is larger than the MTU.

6.4 Performance Metrics and Results

A number of performance metrics are investigated in the evaluation. The first metric is the end-to-end delay, measured in seconds, at LAN_R1. The second metric is the throughput, measured in bits per second, at the link between R5 to R4. The R5 to

R4 link is selected because the tunnel overhead can be examined at this link with respect to each tunneling protocol. The third and the final metric measures the packet drop at router R5. Note that packet drop can be encountered when switching from a non-tunnel link to a tunnel link as the tunnel may cause an increase in the number of packets as a result of fragmentation, and may demand more processing from the router. Thus, router R5 is selected to measure packet drop as it is the closest router to the source of traffic (i.e., the server in LAN_R6) that implements a tunnel. The results for the metrics are generated for each traffic type, for different traffic loads, and for each tunneling protocol.

6.4.1 Simulation for End-to-End Delay

The three traffic loads (i.e., 25%, 50%, and 75%) are simulated for both TCP (FTP, HTTP) and UDP (video conferencing) based applications to measure the end-to-end delay. The end-to-end delay refers to the amount of time that a packet takes to travel from the client to the server. The end-to-end delay includes the transmission time, the propagation time, and the queuing delay.

For the purpose of the simulation, the FTP application is simulated with two different file sizes; 50KB and 100KB. On the other hand, the HTTP and video conferencing applications use only one file of size 4640B and 1172B, respectively.

To achieve the desired traffic load for FTP, HTTP, and video conferencing on the links, the number of users is increased. For 25%, 50%, and 75% traffic load, the number of users is set in the simulation to 10, 20, and 30 users, respectively. In the case of FTP and HTTP, both the file size in bytes and the inter request time in seconds are constant. Similarly, for video conferencing both the frame size in bytes and the frame inter arrival time in seconds are constant.

TCP traffic is considered first for the evaluation of the effect of the tunnel on the total end-to-end delay. For the end-to-end delay metric, all the figures are plotted for 25%, 50% and 75% traffic loads. The figures are plotted for both the no tunnel case and the tunnel case using different tunneling protocols such as IP-in-IP, GRE, and GRE with checksum enabled.

FTP is simulated as requests to download a file from the server in LAN_R6. Likewise, HTTP is simulated as requests for a web page from the server in LAN_R6. It should be noted that FTP and HTTP run over TCP which requires connection establishment. Moreover, we point out that the TCP header is considerably larger than the UDP header.

The results for the FTP application for the file sizes 50KB and 100KB are shown in Figure 6.1 and Figure 6.2, respectively. The figures show the percentage of increase in the end-to-end delay computed as $\frac{(Delay_{WithTunnel} - Delay_{NoTunnel})}{Delay_{NoTunnel}} * 100$. The percentage of increase in the end-to-end delay is caused by the introduction of the tunnel. From the figures, it can be seen that the percentage of increase in end-to-end delay for the case of 100KB file size is more than the case of 50KB file size due to the fact that the 100KB file requires more queuing in routers buffers. Furthermore, the figures display that as the traffic load increases, this percentage decreases. We also note that the IP-in-IP tunneling protocol has the least percentage of increase in the end-to-end delay. The observation is justified by noting that IP-in-IP tunneling protocol adds the least amount of overhead among the tunneling protocols considered, and therefore, produces the least amount of fragmentation. For the completeness of the results, we show in Figure 6.3 and Figure 6.4 the absolute value for the end-to-end delay for both file sizes. The absolute increase in the end-to-end delay for both file sizes is shown in Figure 6.5 and Figure 6.6 and it is computed as $(\text{Delay}_{WithTunnel} - \text{Delay}_{NoTunnel})$.

The results for the HTTP application for the percentage of increase in end-to-end delay is shown in Figure 6.7. From the figures, it can be observed that the IP-in-IP tunneling protocol has the lowest increase in the end-to-end delay. Moreover, it is clear from the figure that as the traffic load increases, the percentage of increase decreases. For the completeness of the results, we show in Figure 6.8 the absolute value for the end-to-end delay. The absolute increase in the end-to-end delay is shown in Figure 6.9. When comparing the end-to-end delay results for FTP and HTTP, we note that the HTTP application is simulated by considering HTTP1.1 persistent TCP connections. With a persistent TCP connection, the server leaves the TCP connection open after the response. Subsequent requests and responses between the same pair of client and server can be sent over the existing TCP connection. This is not the case with the FTP application, where the TCP connection

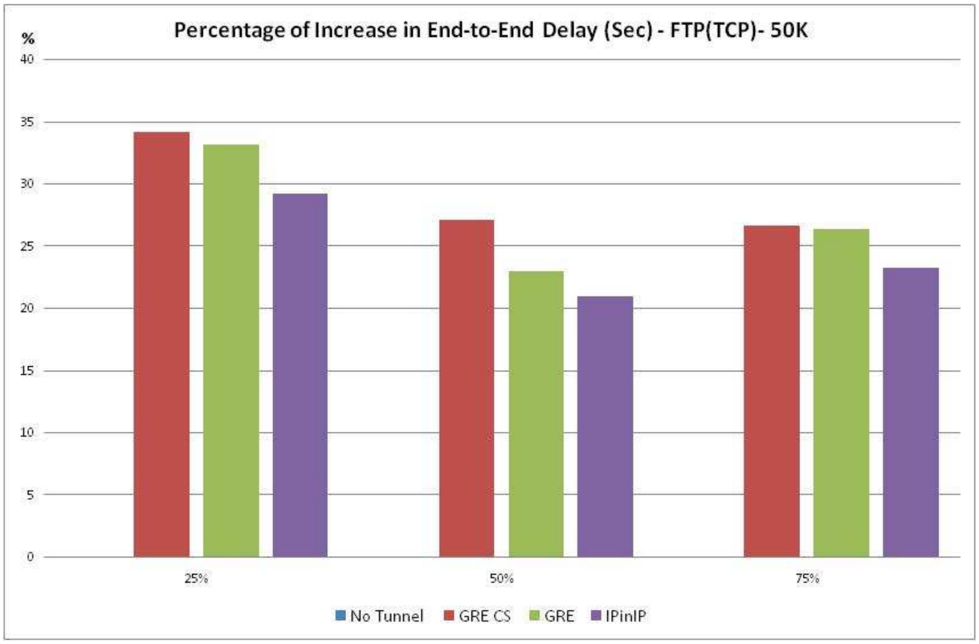


Figure 6.1: Percentage of Increase in End-to-End Delay (Sec) - FTP(TCP)- 50K

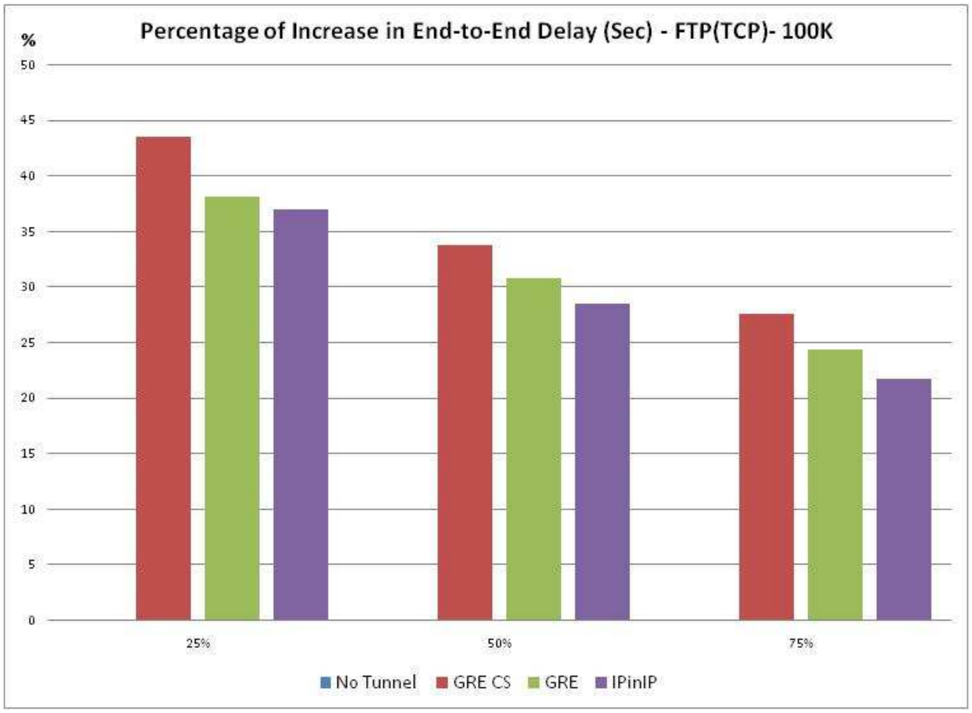


Figure 6.2: Percentage of Increase in End-to-End Delay (Sec) - FTP(TCP)- 100K

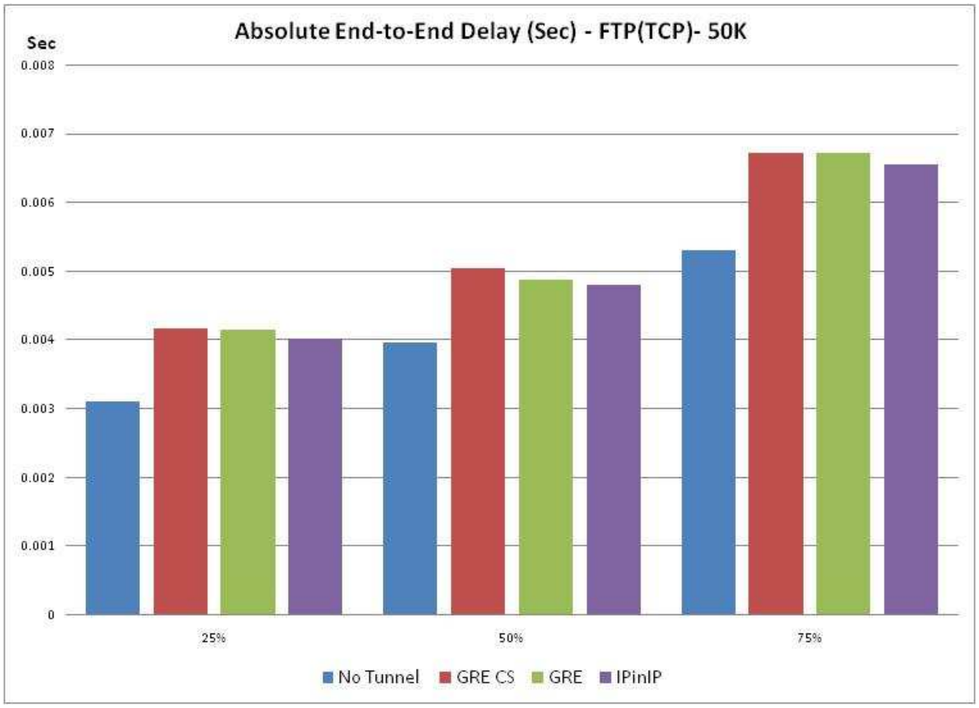


Figure 6.3: Absolute End-to-End Delay (Sec)- FTP(TCP)- 50K

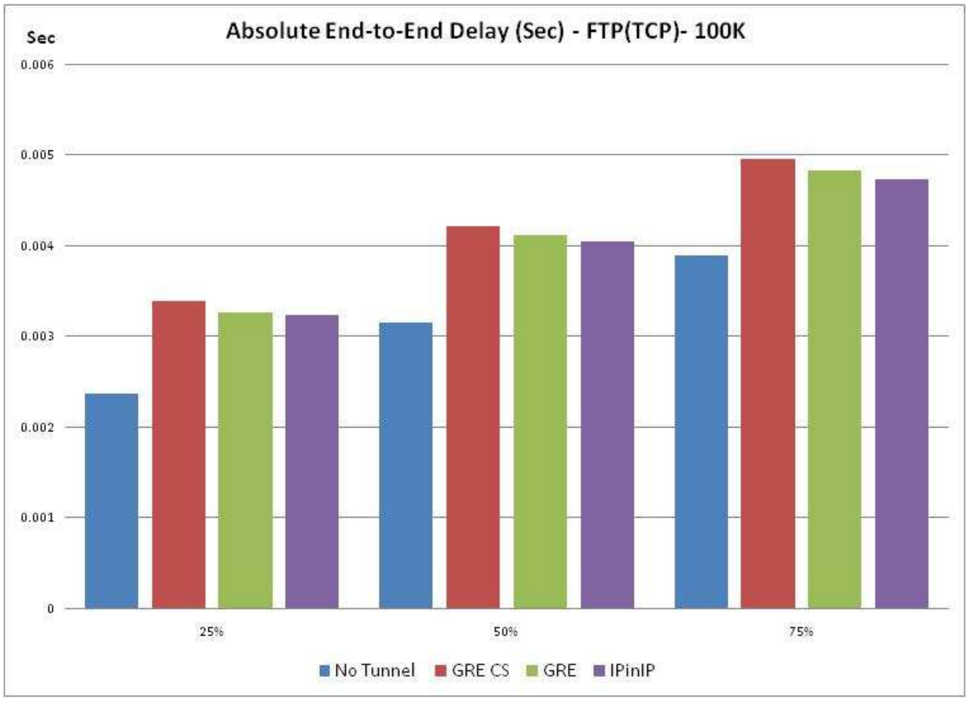


Figure 6.4: Absolute End-to-End Delay (Sec)- FTP(TCP)- 100K

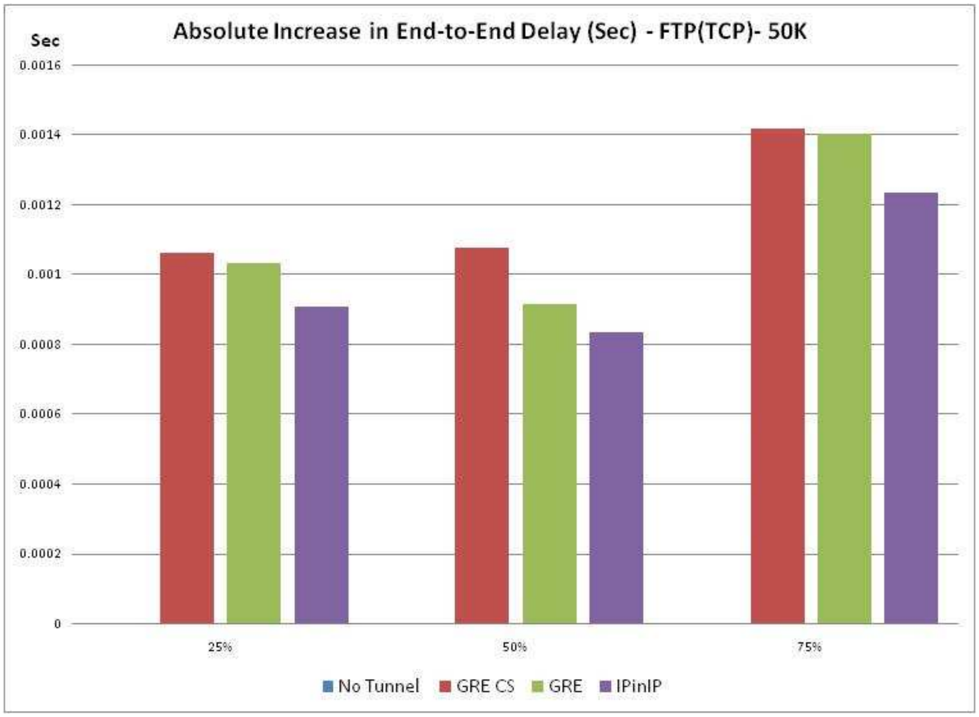


Figure 6.5: Absolute Increase in End-to-End Delay (Sec) - FTP(TCP)- 50K

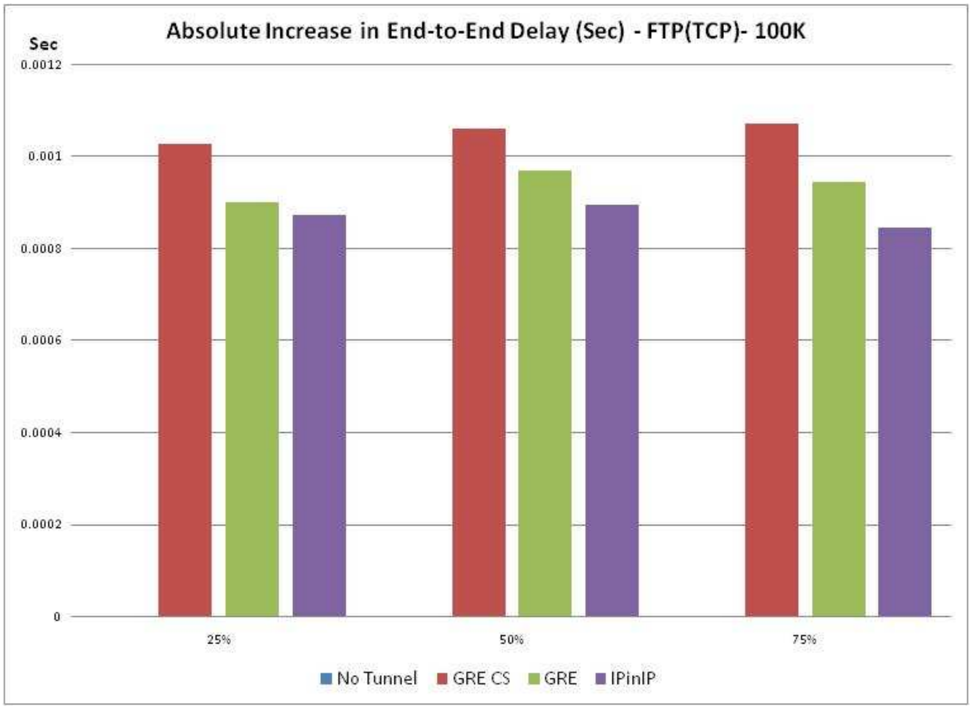


Figure 6.6: Absolute Increase in End-to-End Delay (Sec) - FTP(TCP)- 100K

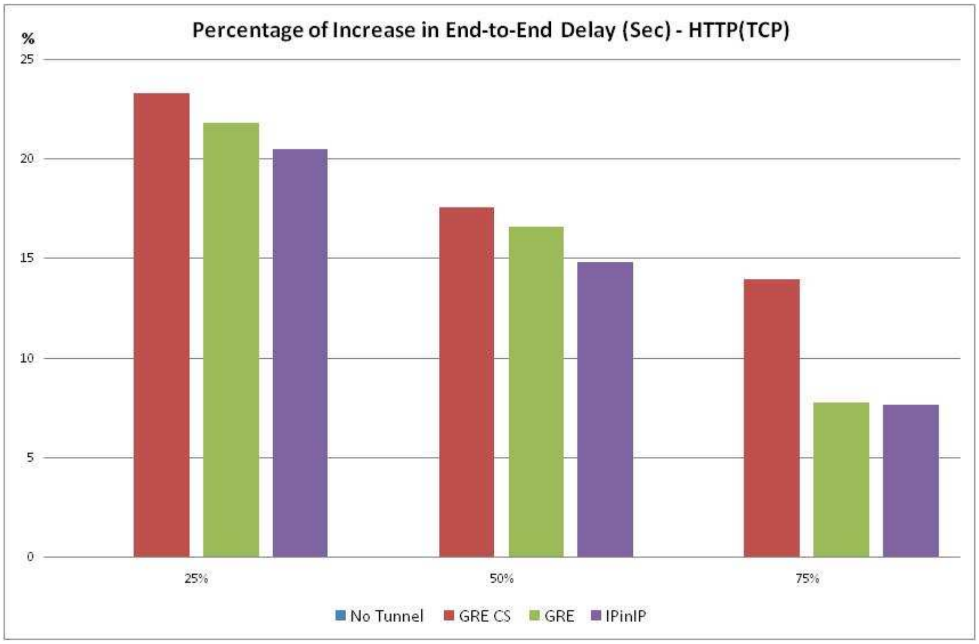


Figure 6.7: Percentage of Increase in End-to-End Delay (Sec) - HTTP(TCP)

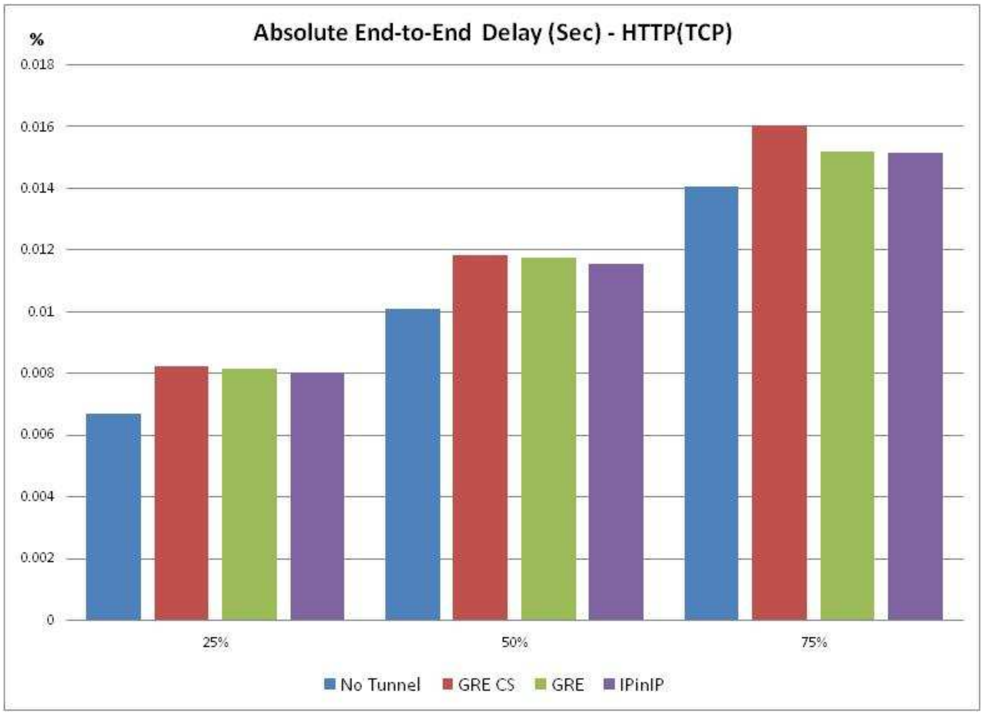


Figure 6.8: Absolute End-to-End Delay (Sec)- HTTP(TCP)

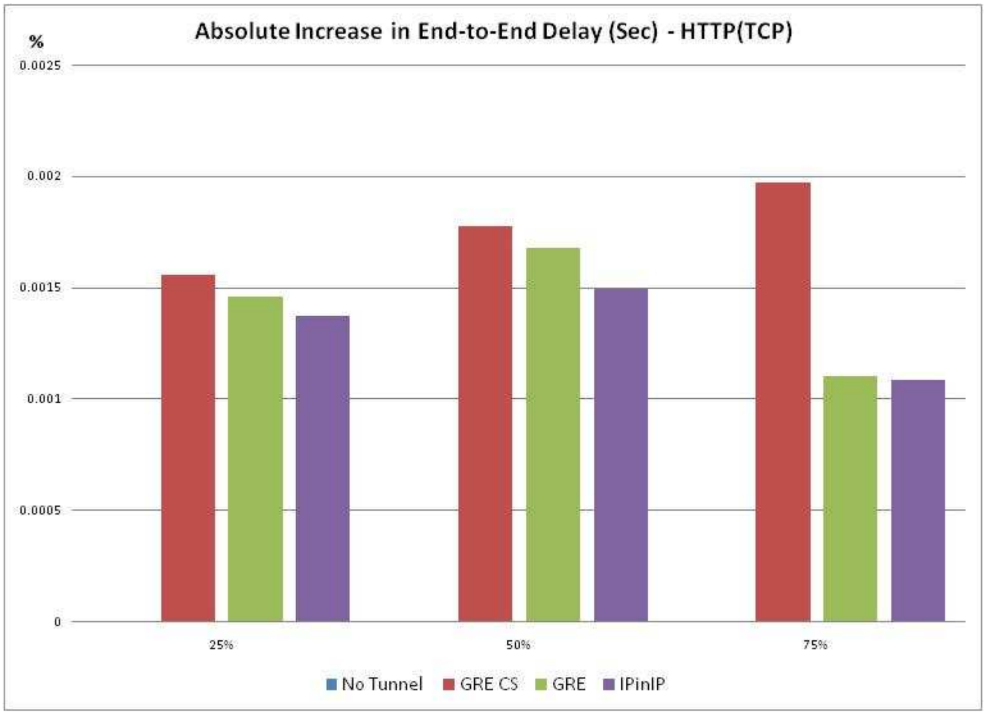


Figure 6.9: Absolute Increase in End-to-End Delay (Sec) - HTTP(TCP)

is closed after the response is sent by the server. Due to this, the percentage of increase in the end-to-end delay is less for the HTTP application when compared with the FTP application.

The last application considered in the simulation is video conferencing which runs over UDP. The end-to-end delay results show similar behaviour to the results obtained for FTP and HTTP. We can see from Figure 6.10 that the absolute value for the end-to-end delay increases as the traffic load increases. Figure 6.11 shows the absolute amount of increase in the end-to-end delay caused by the introduction of the tunnel. The absolute increase in the end-to-end delay is constant with the increase of traffic load. This is mainly due to the fact that the size of the video conferencing file is relatively small, and that the video conferencing application runs over UDP protocol (i.e., connectionless protocol) that does not wait for acknowledgements. Similar to FTP and HTTP, we see that the IP-in-IP tunneling protocol has the least amount of increase in the end-to-end delay. Moreover, the percentage of increase in the end-to-end delay is shown in Figure 6.12, and as evident it shows a lower percentage of increase in the end-to-end delay than the results for the TCP traffic. Furthermore, Figure 6.12 shows that as the traffic load increases, the percentage of increase in end-to-end delay decreases. This is mainly due to the fact that UDP has smaller overhead as compared to TCP, and that UDP is a connectionless protocol that does not wait for acknowledgements.

6.4.2 Simulation of Traffic Overhead

The traffic overhead, in bits per seconds, measures the amount of overhead bits added to each packet entering the tunnel. The same simulation setup as in section 6.4.1 is used where the three scenarios of 25%, 50%, and 75% traffic are simulated, and with different tunneling protocols. The simulation is set to measure the traffic overhead at the link R5 to R4 where the tunnel starts.

The results for the FTP application for the file sizes 50KB and 100KB are shown in Figure 6.13 and Figure 6.14, respectively. The figures show the percentage of increase in the overhead for different file sizes and it is computed as $\frac{(Overhead_{WithTunnel} - Overhead_{NoTunnel})}{Overhead_{NoTunnel}} * 100$. The percentage of increase behaviour for both file sizes follow similar trends.

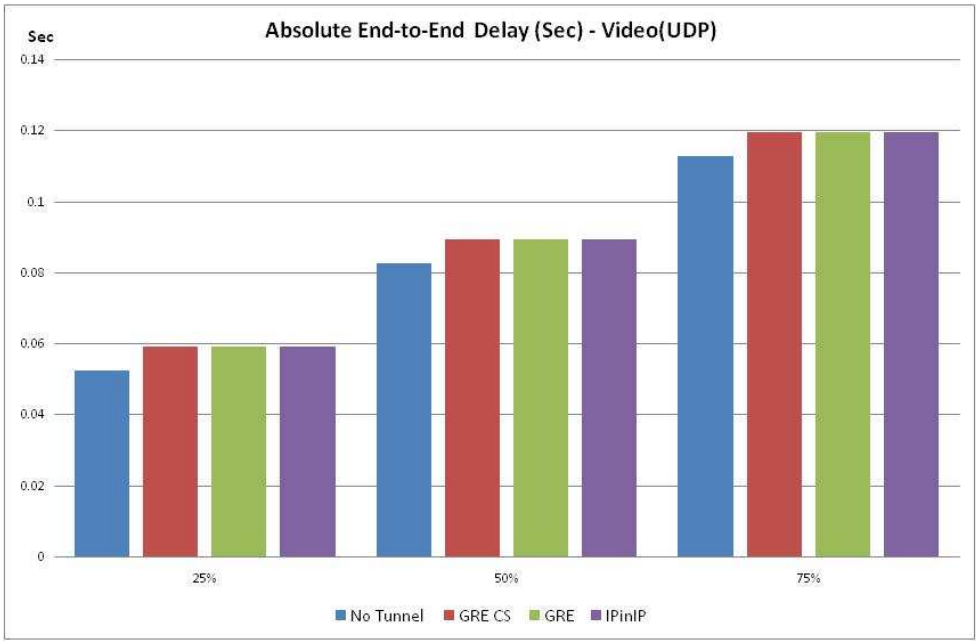


Figure 6.10: Absolute End-to-End Delay (Sec) - Video(UDP)

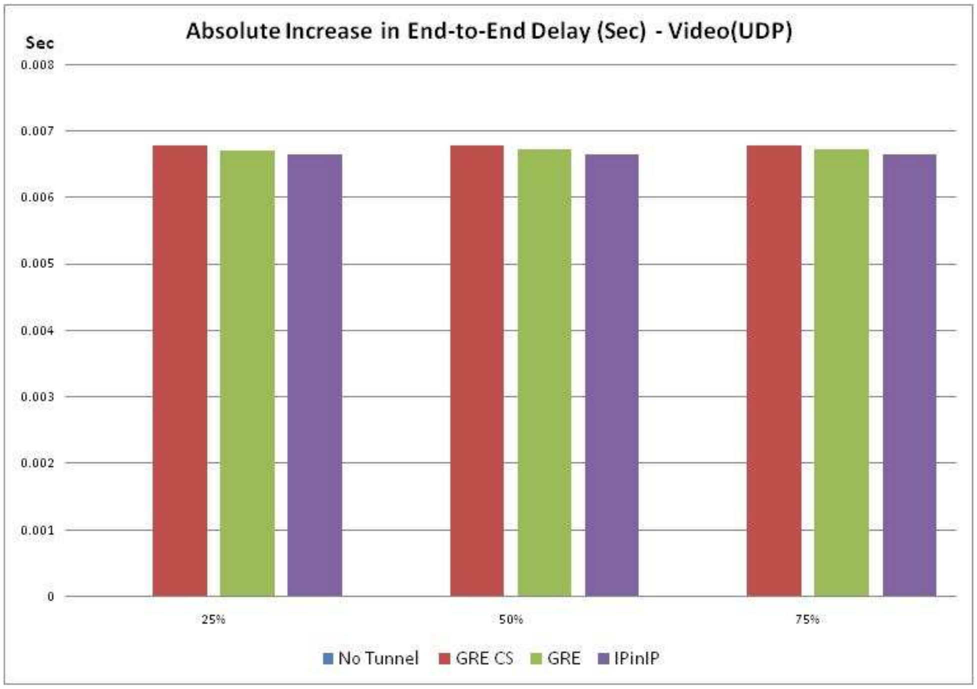


Figure 6.11: Absolute Increase in End-to-End Delay (Sec) - Video(UDP)

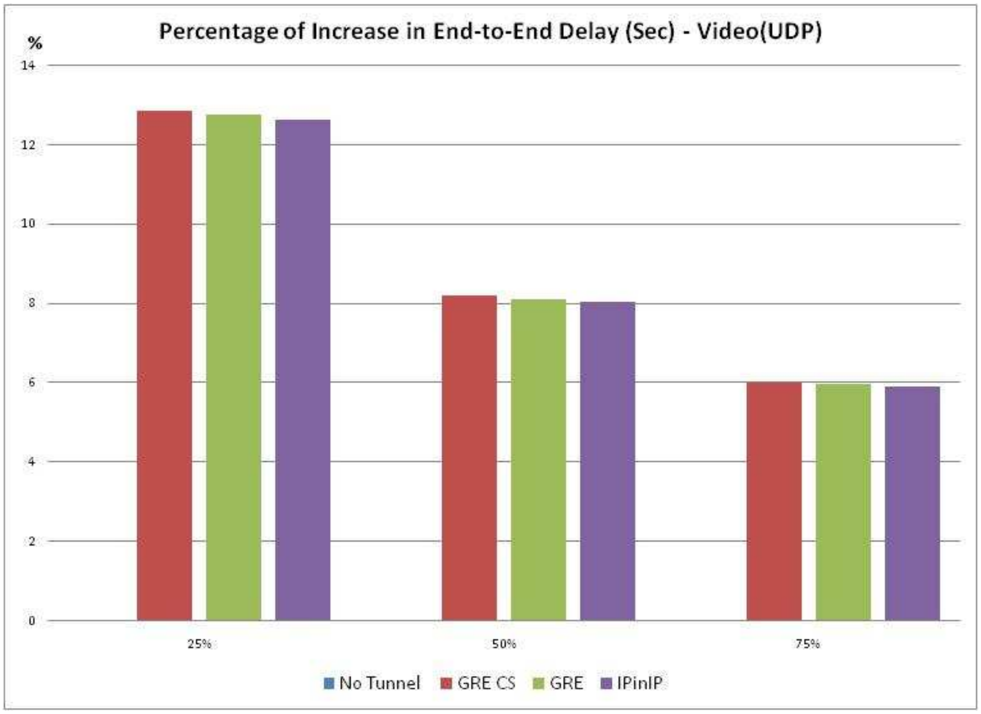


Figure 6.12: Percentage of Increase in End-to-End Delay (Sec) - Video(UDP)

For the completeness of the results, we show in Figure 6.15 and Figure 6.16 the absolute value for throughput. The absolute overhead of the tunnel is shown in Figure 6.17 and Figure 6.18 for both file sizes, and it is computed as (Throughput$_{WithTunnel}$− Throughput$_{NoTunnel}$). It represents the amount of overhead that is caused by the introduction of the tunnel.

The results for the HTTP application for the percentage of increase in the traffic overhead are shown in Figure 6.19. As observed from Figure 6.19, the percentage of increase behaviour for HTTP follows a similar trend as FTP, but the percentage of increase is comparatively less than FTP. This is due to the fact that HTTP1.1 uses persistent TCP connections. Therefore, HTTP uses less number of connections

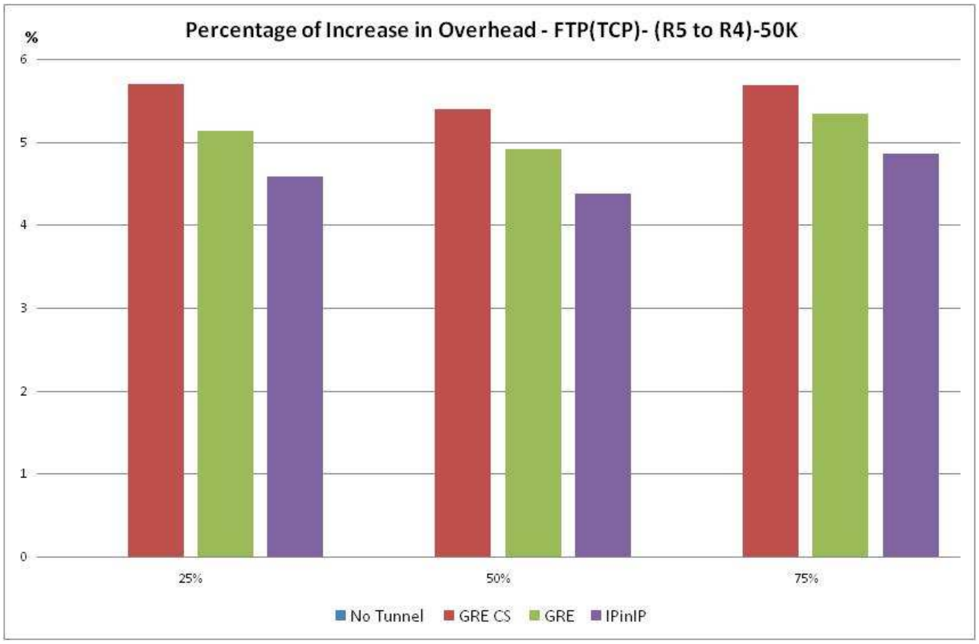


Figure 6.13: Percentage of Increase in Overhead - FTP(TCP)- (R5 to R4) -50K

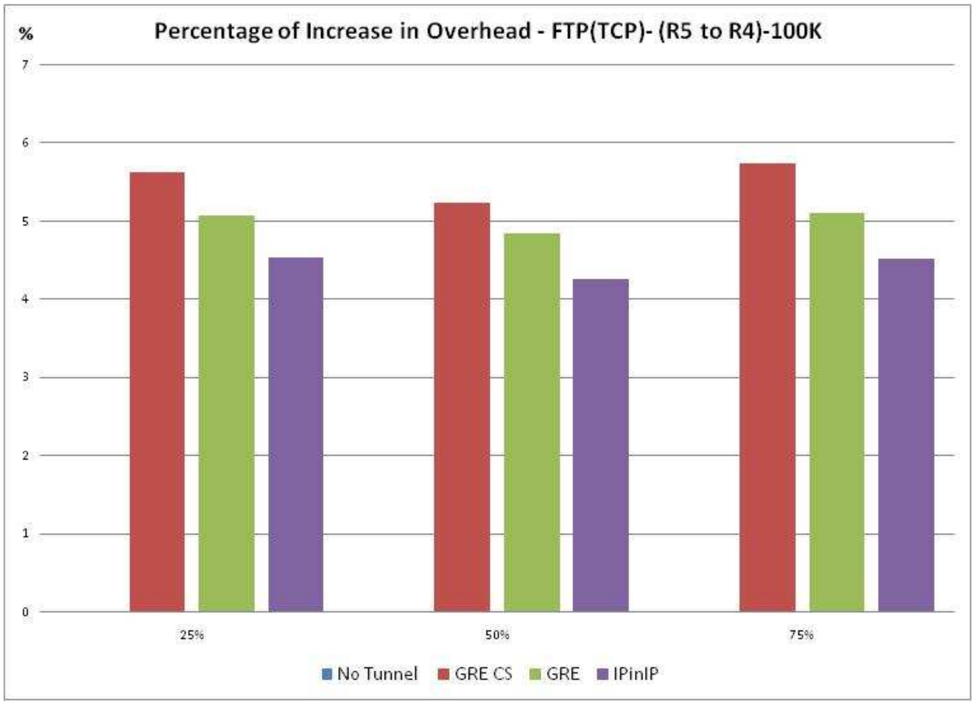


Figure 6.14: Percentage of Increase in Overhead - FTP(TCP)- (R5 to R4) -100K

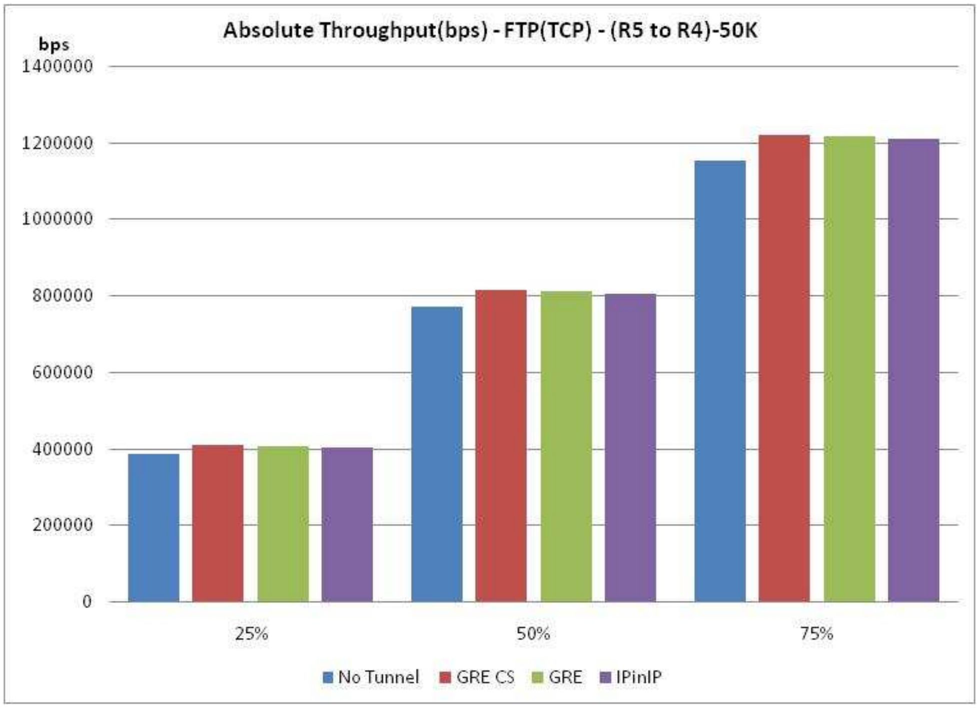


Figure 6.15: Absolute Throughput (bps) - FTP(TCP) - (R5 to R4))-50K

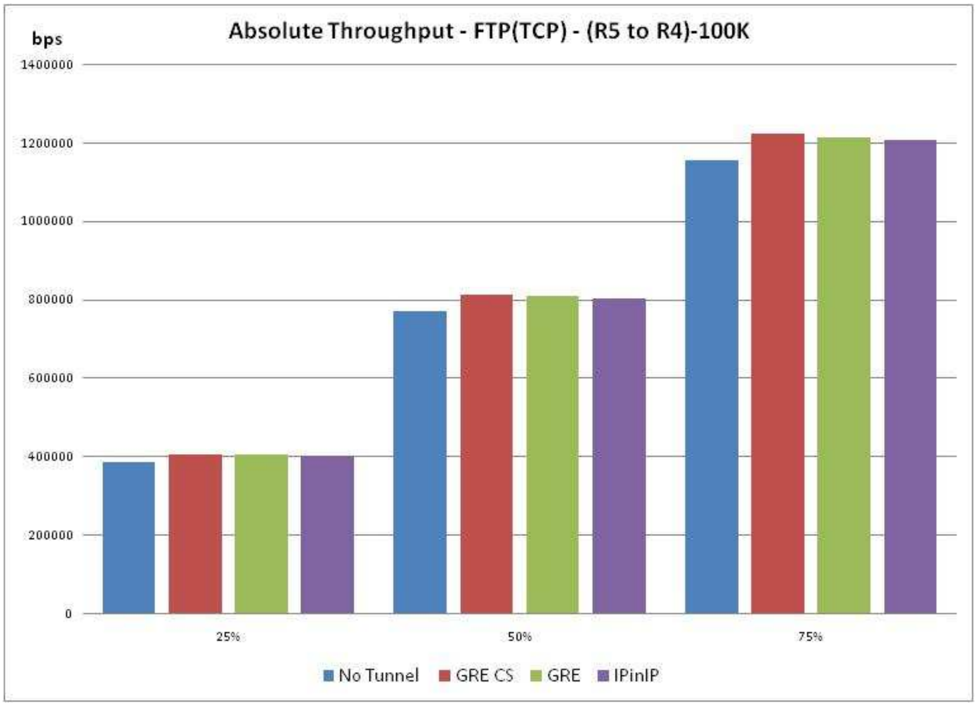


Figure 6.16: Absolute Throughput (bps) - FTP(TCP) - (R5 to R4)-100K

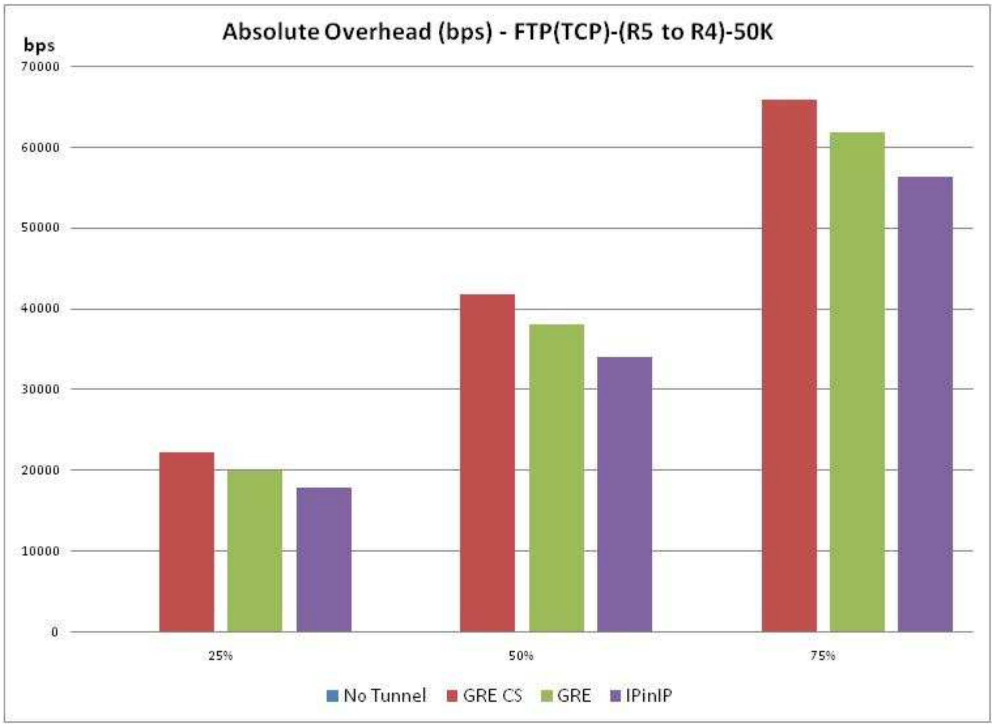


Figure 6.17: Absolute Overhead (bps) - FTP(TCP)-(R5 to R4)-50K

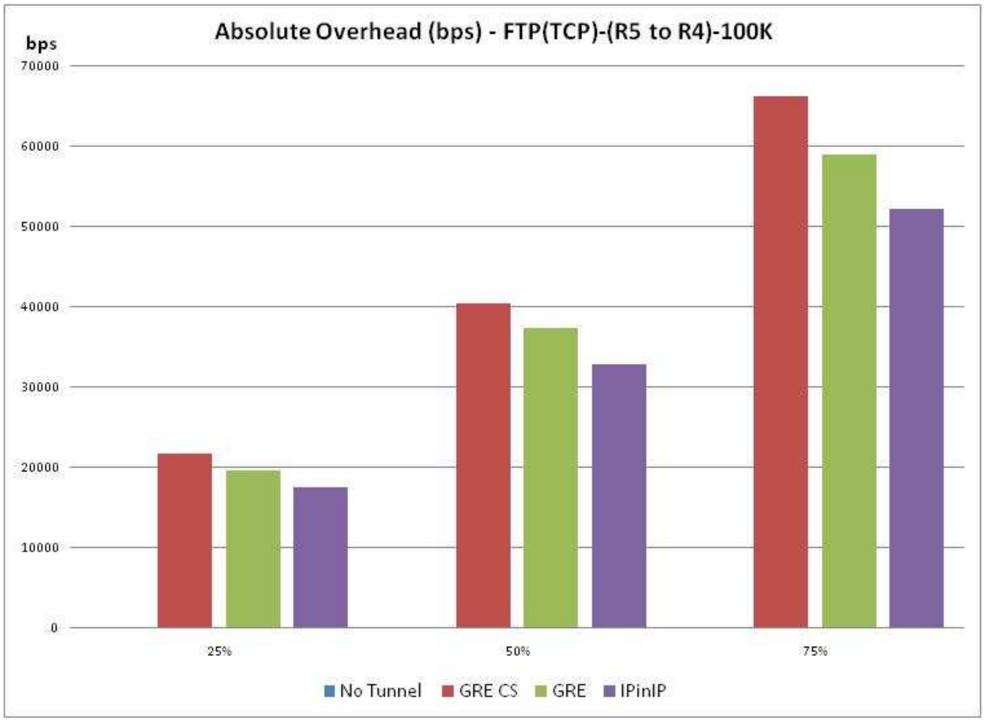


Figure 6.18: Absolute Overhead (bps) - FTP(TCP)-(R5 to R4)-100K

than FTP, and results in less overhead. For the completeness of the results, we show in Figure 6.20 the absolute value for the throughput. The absolute increase in the overhead is shown in Figure 6.21.

The last application considered in the simulation is video conferencing which runs over UDP. The traffic overhead results show similar behaviour to the results obtained for FTP and HTTP. We observe from Figure 6.22 that the percentage of increase in the traffic overhead is the same with different traffic loads. This is due to the fact that a small file size was used for the experiments per the presetting of the simulator. It can also be noted that the percentage increase in the traffic overhead, shown in Figure 6.22 is lower than the increase in throughput for the TCP traffic. As stated earlier, this is mainly caused because UDP has smaller overhead when compared to TCP, and that UDP is a connectionless protocol. The above mentioned factors cause small percentage of increase in the traffic overhead for UDP traffic. For the completeness of the results, we show in Figure 6.23 the absolute value for the throughput. Furthermore, we show in Figure 6.24 the amount of absolute traffic overhead that is caused by the introduction of the tunnel. It can be observed from the figure that the IP-in-IP tunneling protocol has the least amount of increase in traffic overhead as it adds the smallest header size among the other tunneling protocols.

In conclusion, UDP-based traffic has the least amount of increase in the tunnel overhead when compared to the TCP-based traffic. To further make the comparison

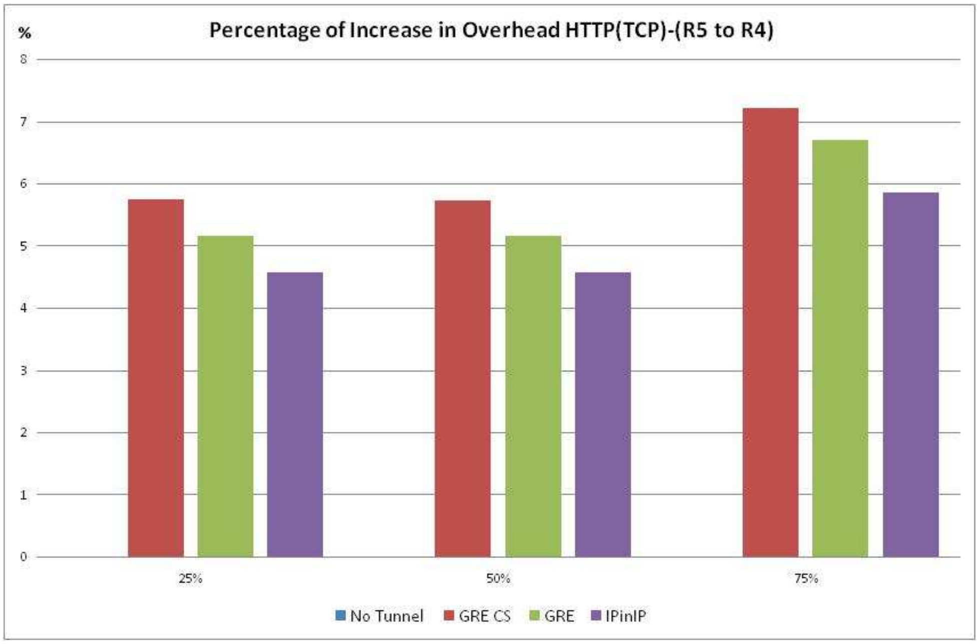


Figure 6.19: Percentage of Increase in Overhead HTTP(TCP)-(R5 to R4)

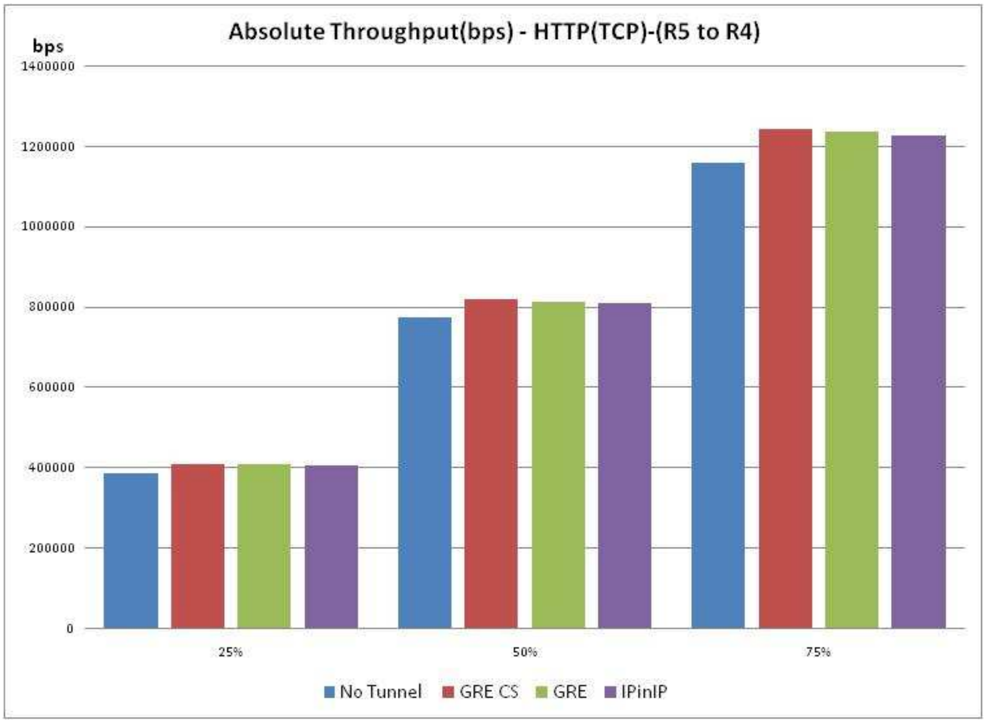


Figure 6.20: Absolute Throughput - HTTP(TCP)-(R5 to R4)

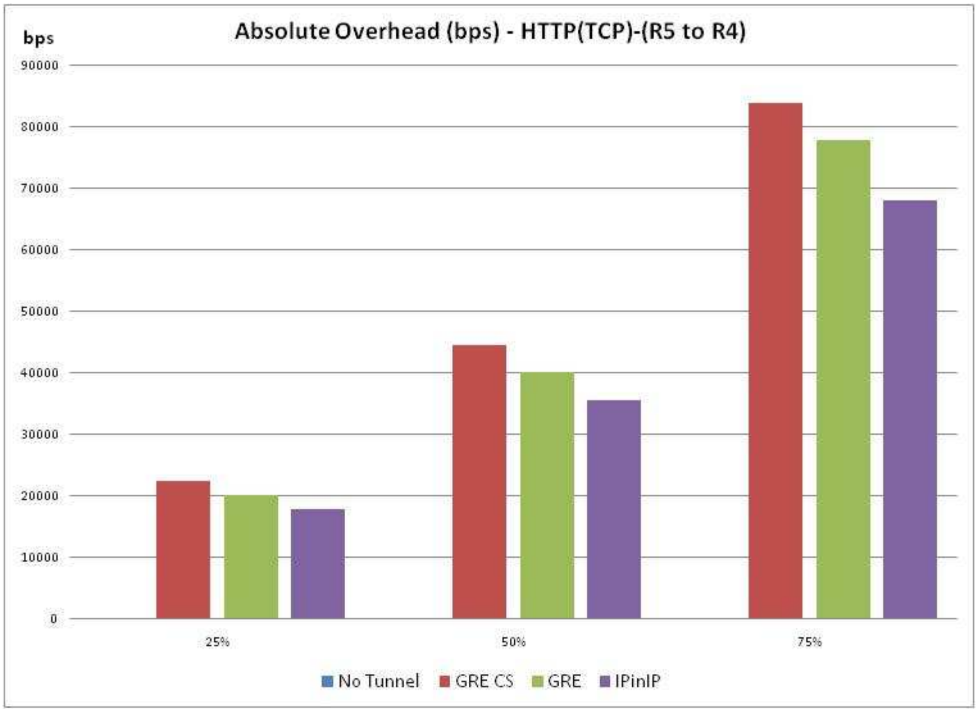


Figure 6.21: Absolute Overhead (bps) - HTTP(TCP)-(R5 to R4)

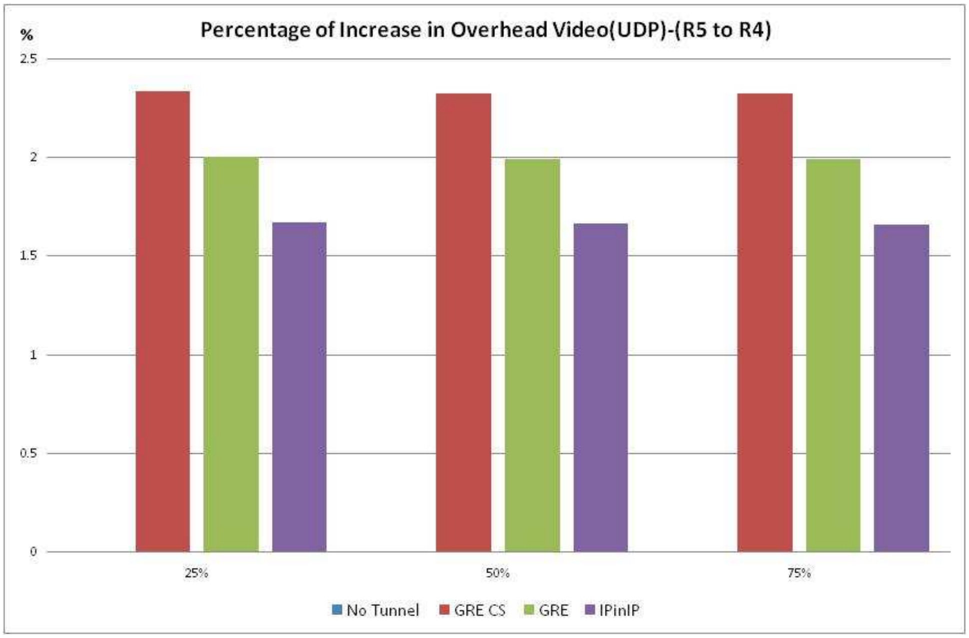


Figure 6.22: Percentage of Increase in Overhead Video(UDP)-(R5 to R4)

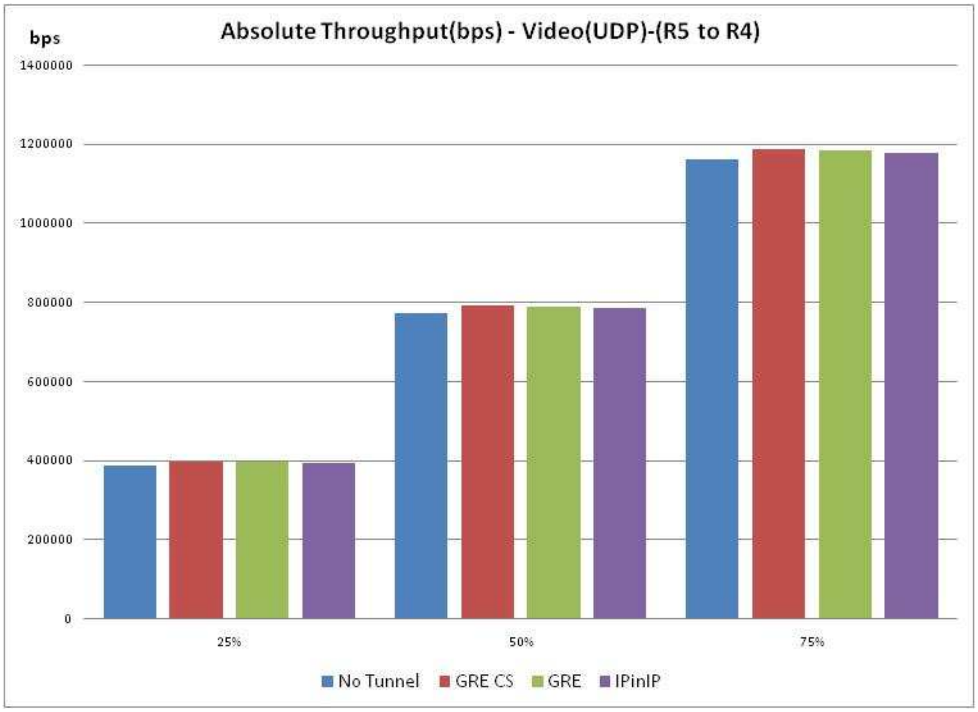


Figure 6.23: Absolute Throughput (bps)- Video(UDP)-(R5 to R4)

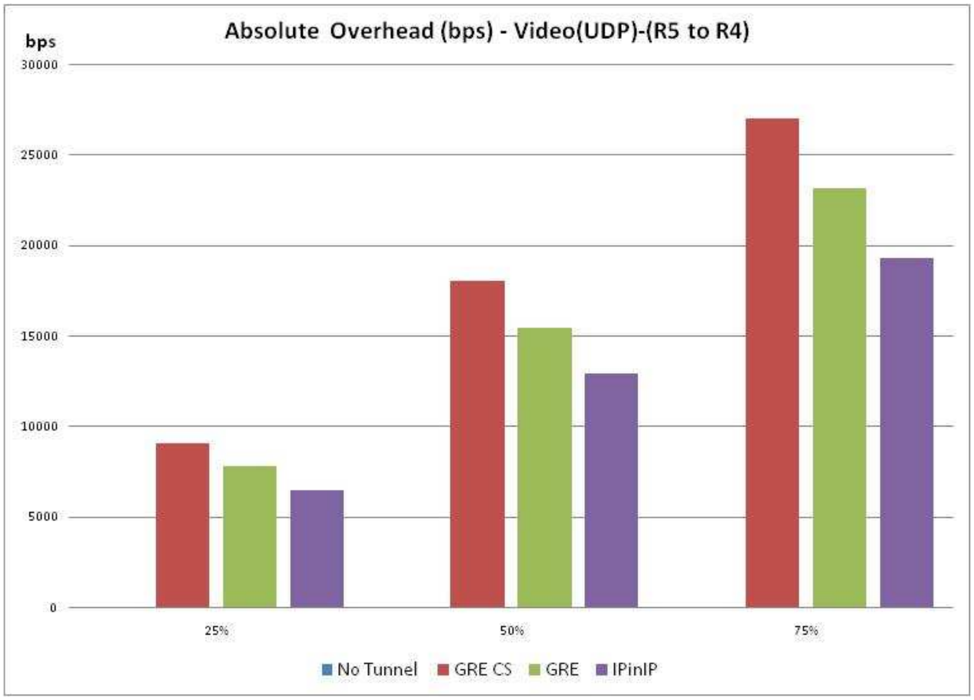


Figure 6.24: Absolute Overhead (bps) - Video(UDP)-(R5 to R4)

fair, an experiment was conducted with a small file size for the FTP application. The results of the experiment confirmed that the UDP-based traffic incurs less amount of tunnel overhead than TCP-based traffic.

6.4.3 Simulation of Packet Drop

The last metric considered in the simulations is packet drop. The main reason behind a packet drop is routers buffer over flow. In the experiments, we reduced the default value of the buffer size from 1MB to 100KB. This is done to examine the effect of tunneling protocol on the proposed solution. If the buffer size is kept at the default value, no packet drop can be observed. The experiments are done with high load traffic on the links (i.e., 94.91% and 98.80% link utilization for TCP traffic and UDP traffic, respectively) in order to examine the effects of tunneling on packet drops.

In order to simulate TCP traffic for the packet drop scenario, we have considered only the FTP application. On the other hand, the HTTP application is not considered for the simulation because, under a high traffic load scenario, the OPNET simulation of HTTP is not behaving as expected, and some of the simulated user connections requested get cancelled. According to OPNET documentation [41], a user cancelled connection occurs in OPNET when the client tries to set up a connection to the server while a connection is already open to that server, the requested page download is in progress, or the user clicks on the link on a page that is still downloading.

OPNET support was contacted for the user cancelled connection scenario of HTTP application and could not confirm that such behaviour occurs in real networks. Moreover, we have looked at the HTTP standard (RFC 2616) [42] and the standard does not have any references to user cancelled connections.

Figure 6.25 and Figure 6.26 show the maximum instantaneous percentage of traffic drop in bits per seconds and packets per seconds, respectively. The traffic drop in the case of video application is the lowest. The traffic drop for the FTP 100KB file size is less when compared to the 50KB file size because in the 100KB file size case the frequency of packet drop is more. To know the frequency of packet drops, we need to consider detailed figures that are shown in Appendix B.1. In the case of no tunnel and IP-in-IP tunneling, packet drops are not observed in all the applications, except for FTP 100KB file size case. This is due to the fact that the larger size of files being transferred will cause more queuing of packets in the router buffer, as illustrated in the detailed figures in the Appendix B.1.

Figure 6.27 and Figure 6.28 show the percentage of average traffic drop in bits per seconds and packets per seconds, respectively. Similar to the percentage of maximum instantaneous traffic drop, the average traffic drop in the case of video application is the lowest. The average traffic drop for the FTP 100KB file size is more when compared to the 50KB file size, because in the case of the 100KB file size the frequency of the packet drop is more, as demonstrated in the detailed figures in Appendix B.1.

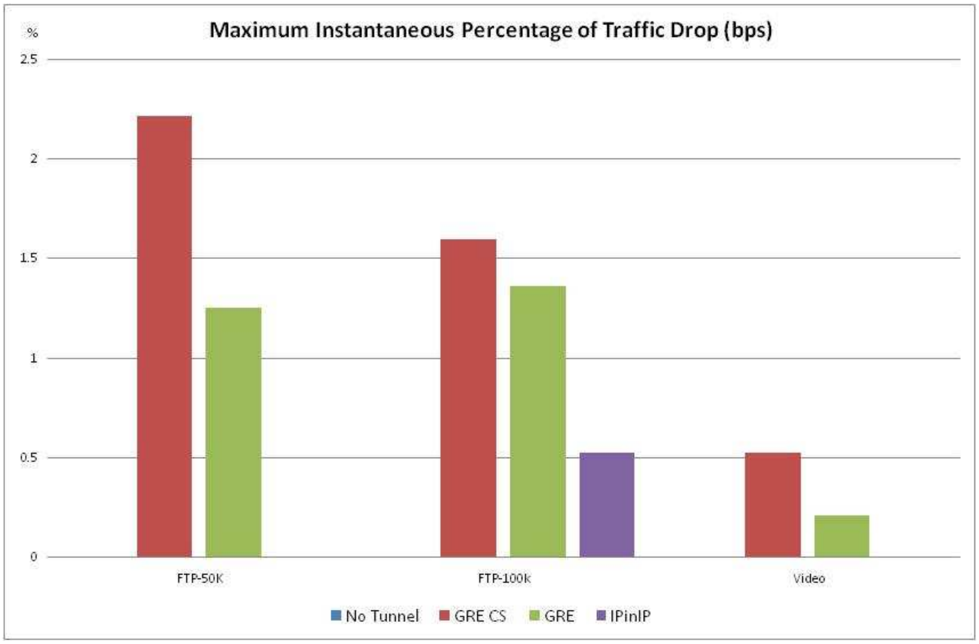


Figure 6.25: Maximum Instantaneous Percentage of Traffic Drop (bps)

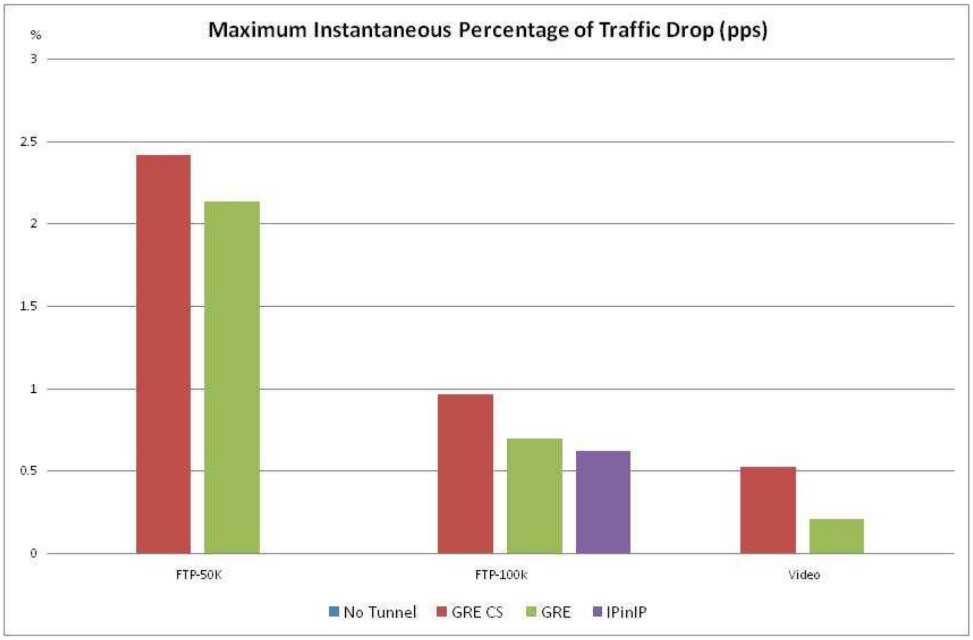


Figure 6.26: Maximum Instantaneous Percentage of Traffic Drop (pps)

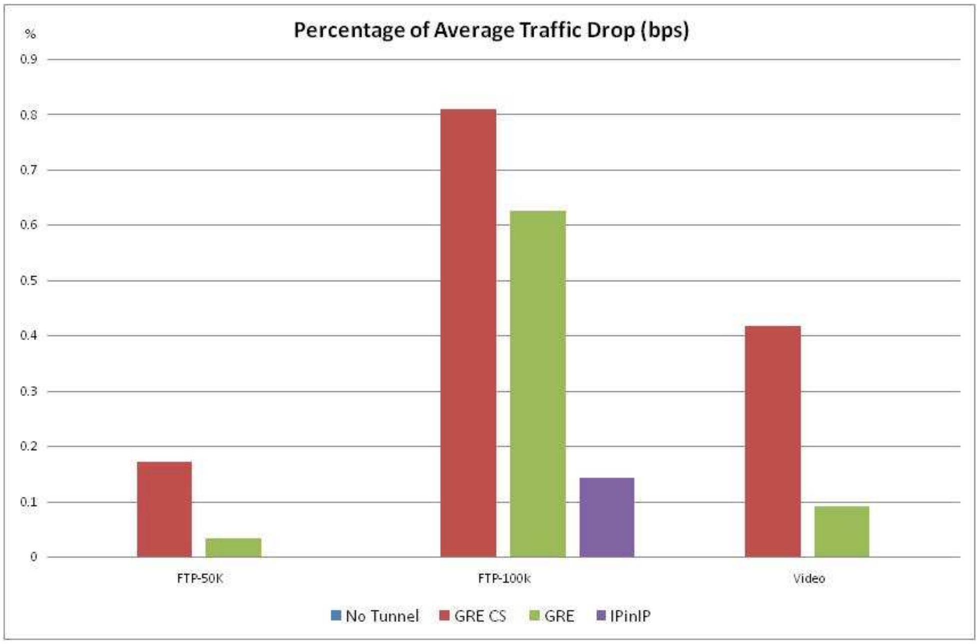


Figure 6.27: Percentage of Average Traffic Drop (bps)

6.5 Discussion and Recommendations

In this section, we discuss the suitability of the Internet denial by IISP's proposed solution, the performance consideration, and recommendations.

6.5.1 Discussion

In chapter 4, a number of solutions for the Internet denial by IISPs were introduced. The proposed tunnel-based solution required a low amount of modification to the network. In the proposed solution, once the tunnel is established, the forwarding of the packets is done normally through the tunnel, and is valid for both incoming and outgoing traffic. In terms of performance, the tunnel solution adds an amount

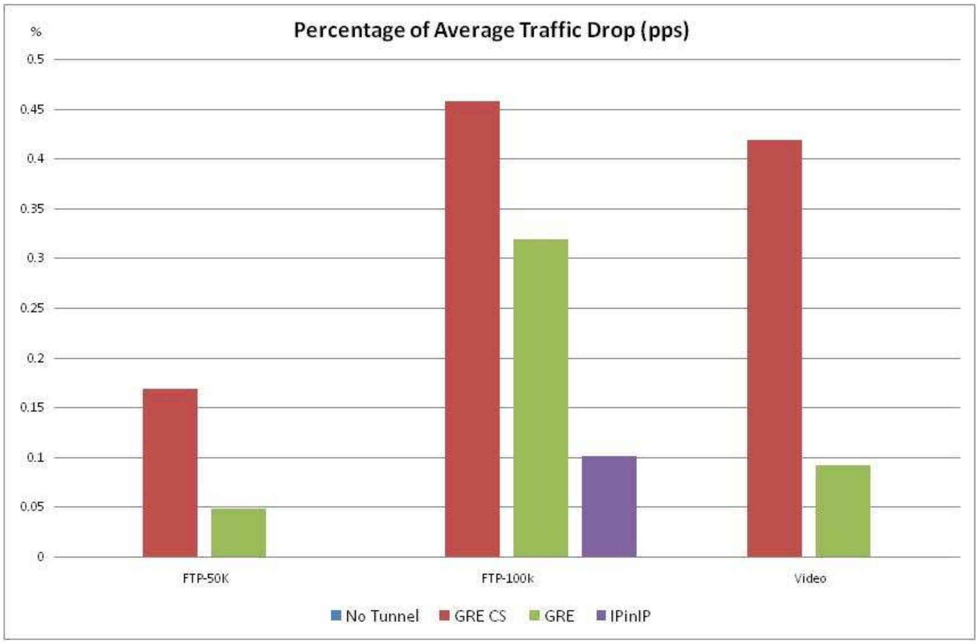


Figure 6.28: Percentage of Average Traffic Drop (pps)

of overhead traffic due to the addition of a tunnel header to each packet entering the tunnel. The packet drops can only be examined if the utilization on the links reaches above 94.91% for TCP traffic and 98.80% for UDP traffic. Furthermore, the packet drops are only seen in GRE and GRE with check sum tunneling protocols.

6.5.2 Recommendations

Based on the results of the simulations of the tunnel-based solution, we observe that the IP-in-IP tunneling protocol performs the best. Therefore, it is recommended to use IP-in-IP tunneling protocol for the deployment because it has the least amount of overhead when compared with the other tunneling protocols considered in this thesis. As discussed earlier in section 5.4, the design can be made scalable to account for the need to deploy the solution on a large scale.

6.6 Summary

In this chapter, we studied the performance aspects of the tunnel-based solution by first discussing the simulation setup, then we discussed the aspect of tunnel fragmentation and its effect on performance. Next, we discussed the results of the performance evaluation of the proposed tunnel-based solution in terms of end-to-end delay, tunnel overhead, and packet drop. Lastly, the chapter is concluded by some discussion and recommendations.

Chapter 7

Conclusion

In this thesis, we discuss the different approaches to overcome the malicious act of Internet access denial by IISP. The Internet access denial may be intentional as the case of denial imposed by IISP or unintentional denial as a result of security breach at the IISP.

The goal of this thesis work is to propose, implement, and validate the solution to the problem of Internet access denial. In this thesis, only tunnel-based solution is considered. In addition, the scalability of the design was evaluated for multiple ASes, and multiple tunnels. Furthermore, a tunnel-based solution design was proposed for load balancing using a pool of public IP addresses.

For the solution to work, traffic must pass through the tunnel in both directions such that packets do not get filtered out by the malicious IISP. Thus, the thesis described the design, implementation, and validation of the tunnel-based approach. Moreover, a performance evaluation of tunnel-based solution was provided. The evaluation was conducted using different tunneling protocols, applications, and traffic loads. The evaluation used end-to-end delay, tunnel overhead, and packet drop metrics for comparison. The results of evaluation showed that IP-in-IP tunneling protocol performed the best among all the protocols considered in this study.

This chapter concludes the thesis by outlining the contributions achieved. Moreover, it discusses some of the open problems for future work.

7.1 Summary of the Contributions

In this thesis, the following has been achieved:

1. A tunnel-based solution to the Internet denial problem was designed and evaluated using OPNET.

2. The network performance was characterized under different tunneling protocols.

3. The scalability of the design was evaluated for multiple ASes, multiple tunnels, and large number of users.

4. A tunnel-based solution design was proposed for load balancing using a pool of public IP addresses.

7.2 Future Work

Many issues in the Internet denial problem are open for further research. Some of these open issues include:

Detection of Internet denial: The detection process is important to find out when the traffic blackholing is taking place, and whether the cause of the blackholing is malicious or non-malicious. The challenge of this process is to distinguish whether the packet dropping is due to malicious causes such as Denial of Service (DoS) or by routing level Internet denial, or by non-malicious causes such as mis-configuration of intermediate routers, congestion in the network, connection time out, and/or server unavailability.

Dynamic configuration of tunnel establishment: In the proposed solution, the tunnel is established by manual configuration. One of the open research areas could be on how to establish a tunnel by dynamic configuration.

Dynamic configuration setup for redistribution: Another open research topic is related to how redistribution of routing information through the tunnel can be established dynamically.

Use of Encrypted Tunnels: Since the Provider can analyze traffic inside the tunnel and, accordingly, drop the packets, then an encryption algorithm can be used to secure tunneling protocols. Accordingly, performance impacts need to be revisited and reassessed.

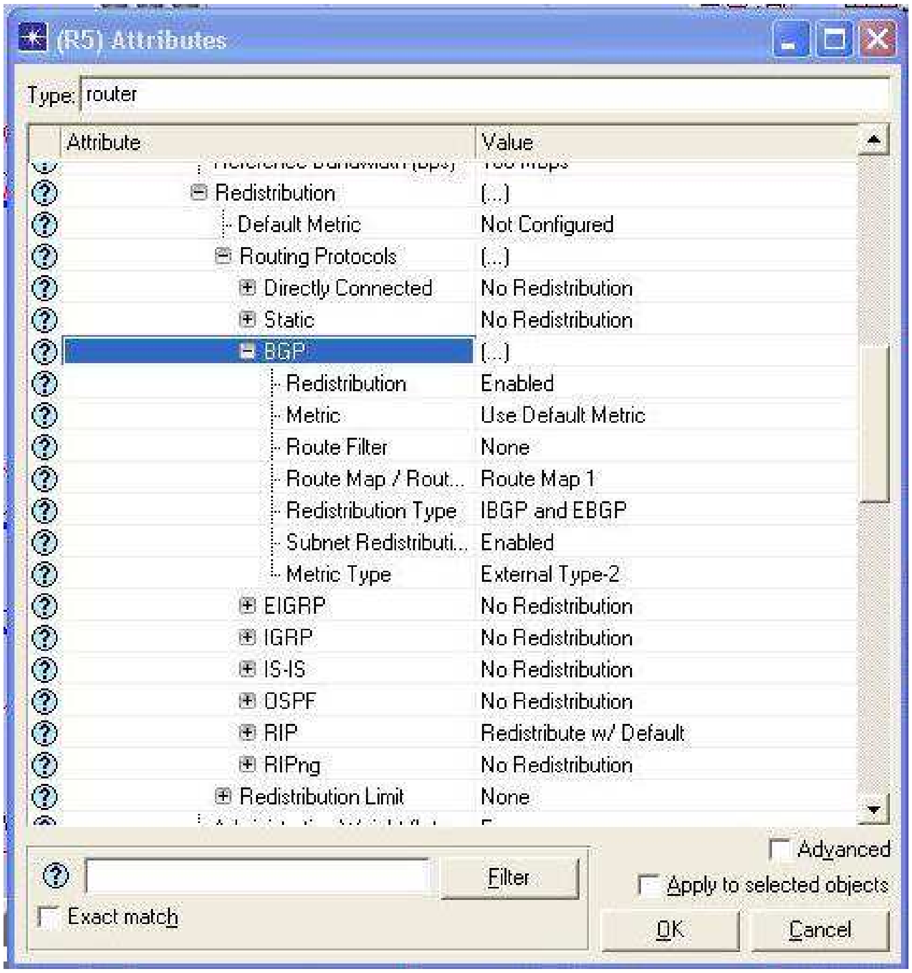
Appendix A

Route Redistribution Configuration

A.1 Route Redistribution Configuration at RouterR5

The following are the steps for the manual configuration for the route redistribution in OPNET.

The redistribution configuration on R5 is as follows: Right Click on router R5– Edit Attributes – IP Routing Protocols – OSPF Parameters – Processes – Process Parameters – Redistribution – BGP – Redistribution Type – default is set to "EBGP". Hence, only EBGP routes are redistributed. We need to change this field to both IBGP and EBGP. This is shown in Figure A.1.



(R5) Attributes

Type: router

Attribute	Value
Reference Bandwidth (bps)	100 Mbps
⊟ Redistribution	(...)
Default Metric	Not Configured
⊟ Routing Protocols	(...)
⊞ Directly Connected	No Redistribution
⊞ Static	No Redistribution
⊟ BGP	(...)
Redistribution	Enabled
Metric	Use Default Metric
Route Filter	None
Route Map / Rout...	Route Map 1
Redistribution Type	IBGP and EBGP
Subnet Redistributi...	Enabled
Metric Type	External Type-2
⊞ EIGRP	No Redistribution
⊞ IGRP	No Redistribution
⊞ IS-IS	No Redistribution
⊞ OSPF	No Redistribution
⊞ RIP	Redistribute w/ Default
⊞ RIPng	No Redistribution
⊞ Redistribution Limit	None

Figure A.1: Route Redistribution Configuration at Router R5

Appendix B

OPNET Snapshot for Buffer Usage and Traffic Drop

B.1 OPNET Snapshot for Buffer Usage and Traffic Drop at Router R5 on Interface IF10

Snapshots from OPNET for the buffer usage and traffic drop are provided in the following pages. First we examine FTP-50K followed by FTP 100K and Video conferencing. Each figure is plotted with no tunnel configuration and with tunnel configuration. The tunnel name is stated in the figure caption.

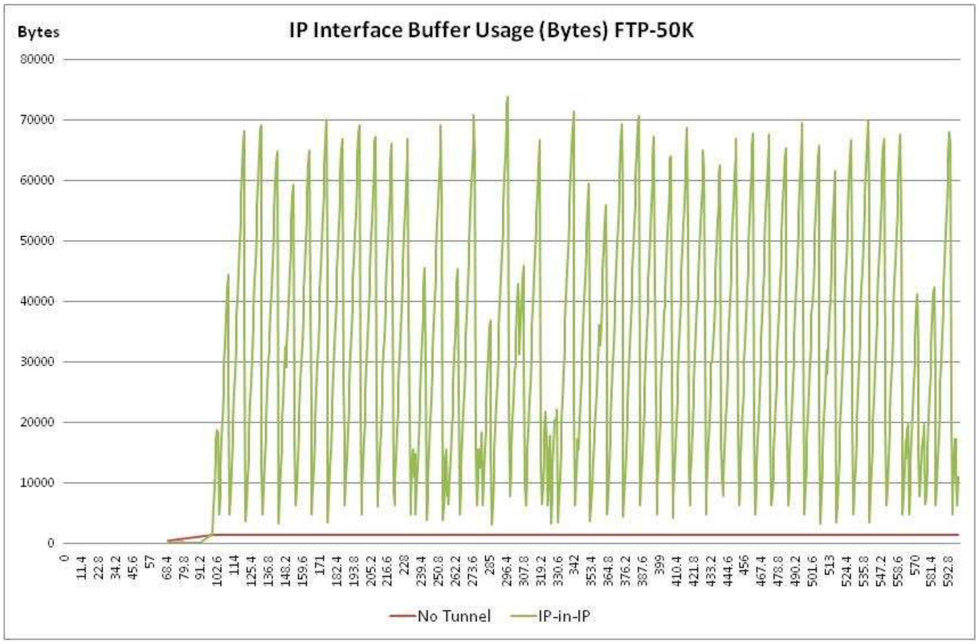


Figure B.1: IP Interface Buffer Usage (Bytes) with IP-in-IP Tunnel and with No Tunnel for FTP 50K file size

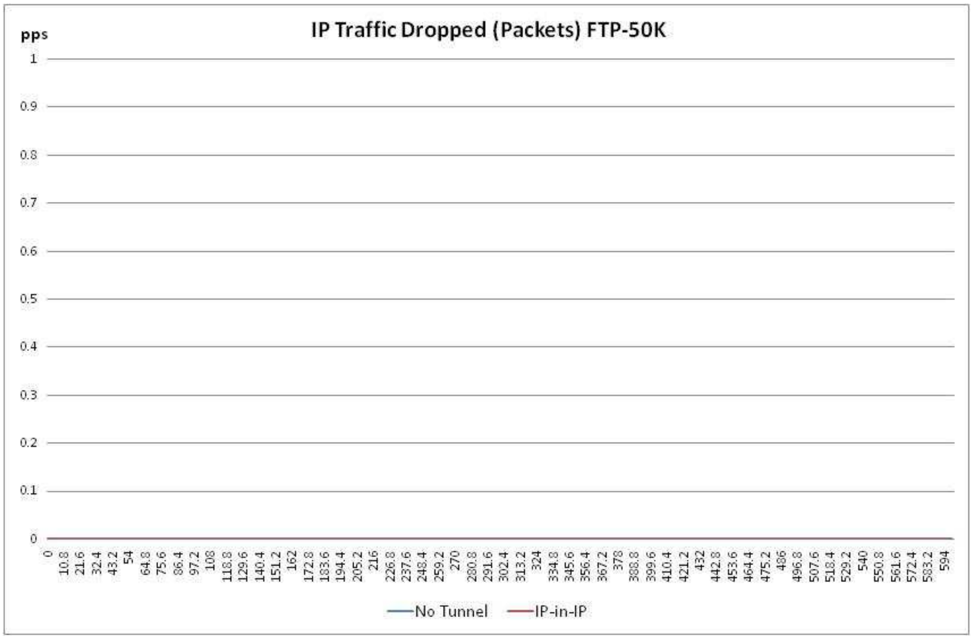


Figure B.2: IP Traffic Dropped (Packets) with IP-in-IP Tunnel and with No Tunnel for FTP 50K file size

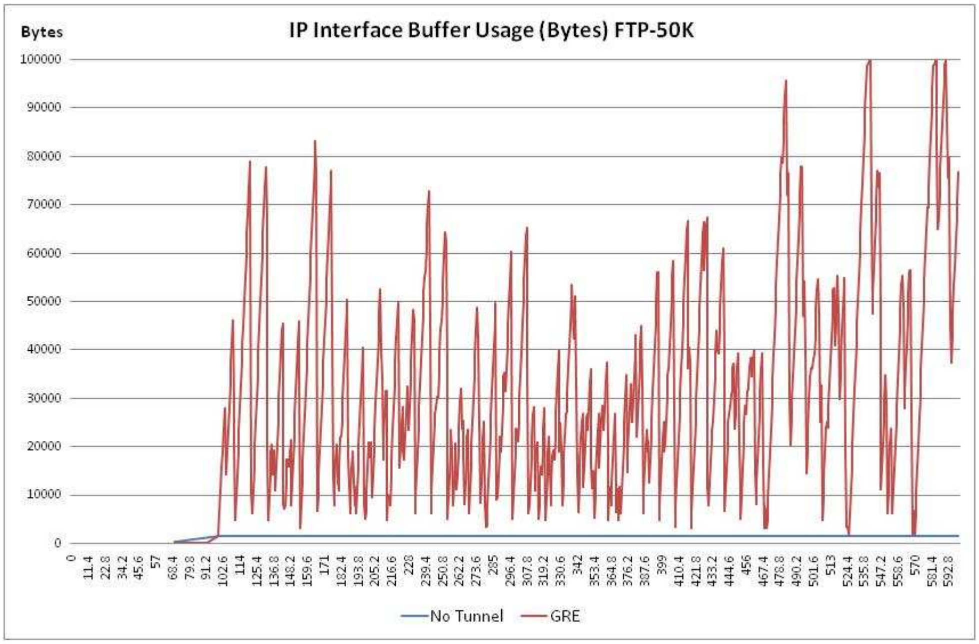


Figure B.3: IP Interface Buffer Usage (Bytes) with GRE Tunnel and with No Tunnel for FTP 50K file size

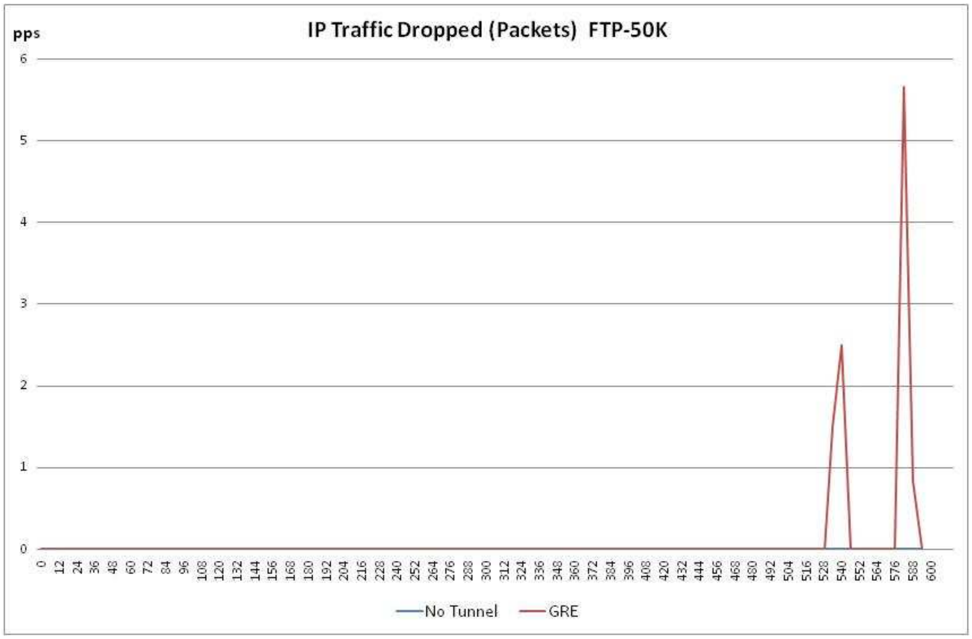


Figure B.4: IP Traffic Dropped (Packets) with GRE Tunnel and with No Tunnel for FTP 50K file size

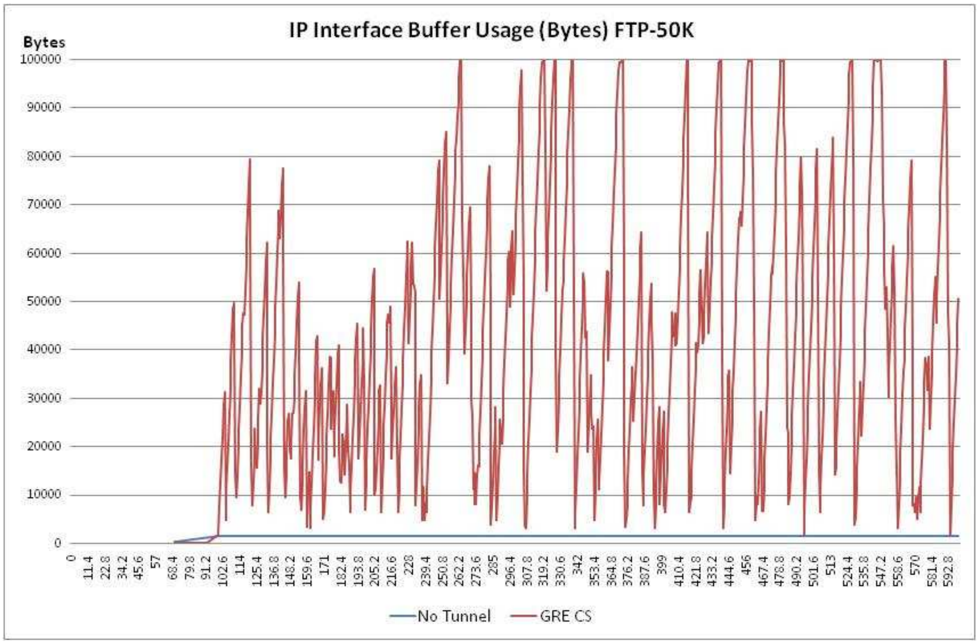


Figure B.5: IP Interface Buffer Usage (Bytes) with GRE CS Tunnel and with No Tunnel for FTP 50K file size

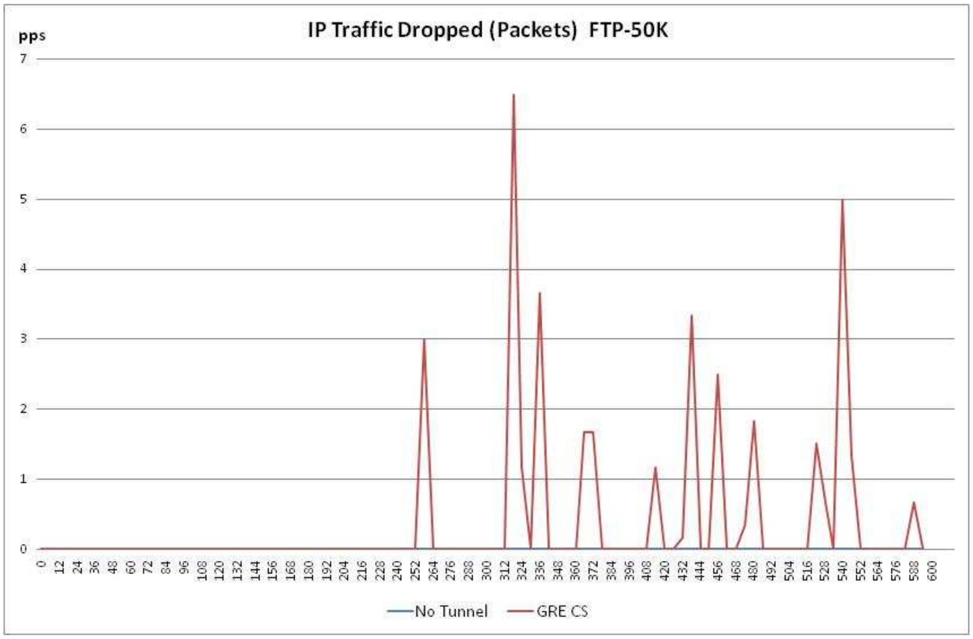


Figure B.6: IP Traffic Dropped (Packets) with GRE CS Tunnel and with No Tunnel for FTP 50K file size

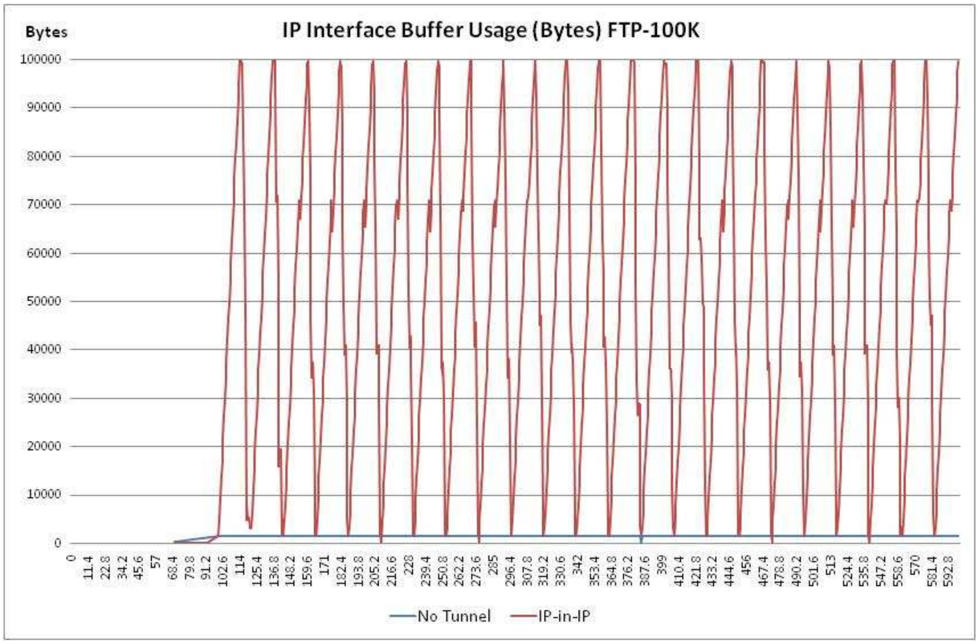


Figure B.7: IP Interface Buffer Usage (Bytes) with IP-in-IP Tunnel and with No Tunnel for FTP 100K file size

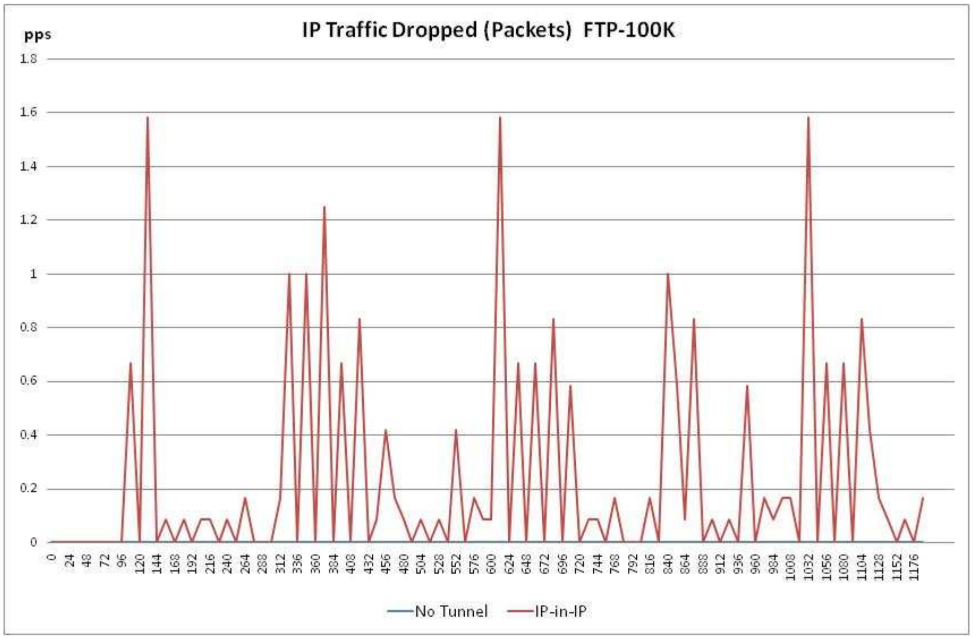


Figure B.8: IP Traffic Dropped (Packets) with IP-in-IP Tunnel and with No Tunnel for FTP 100K file size

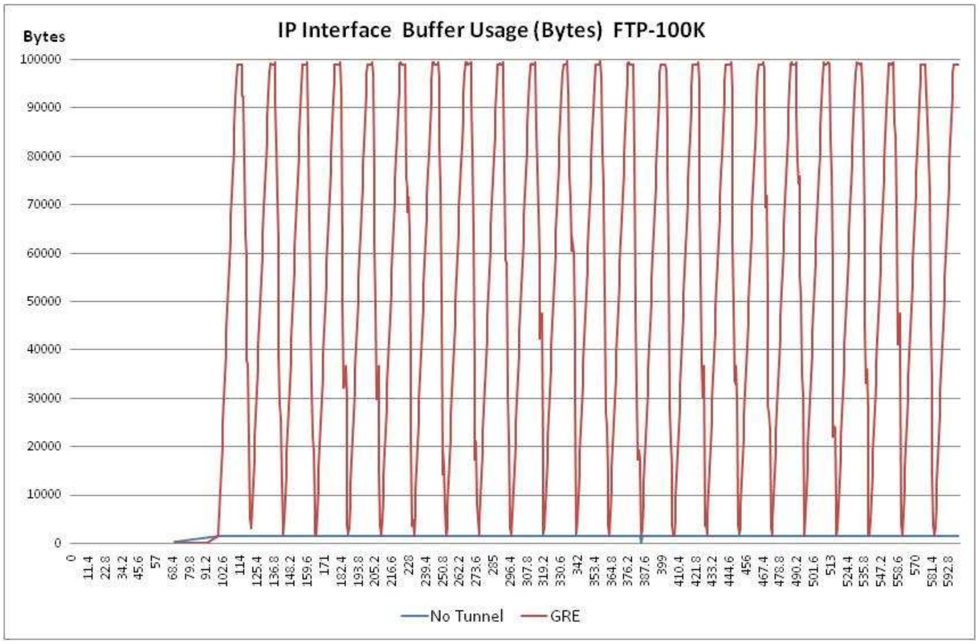


Figure B.9: IP Interface Buffer Usage (Bytes) with GRE Tunnel and with No Tunnel for FTP 100K file size

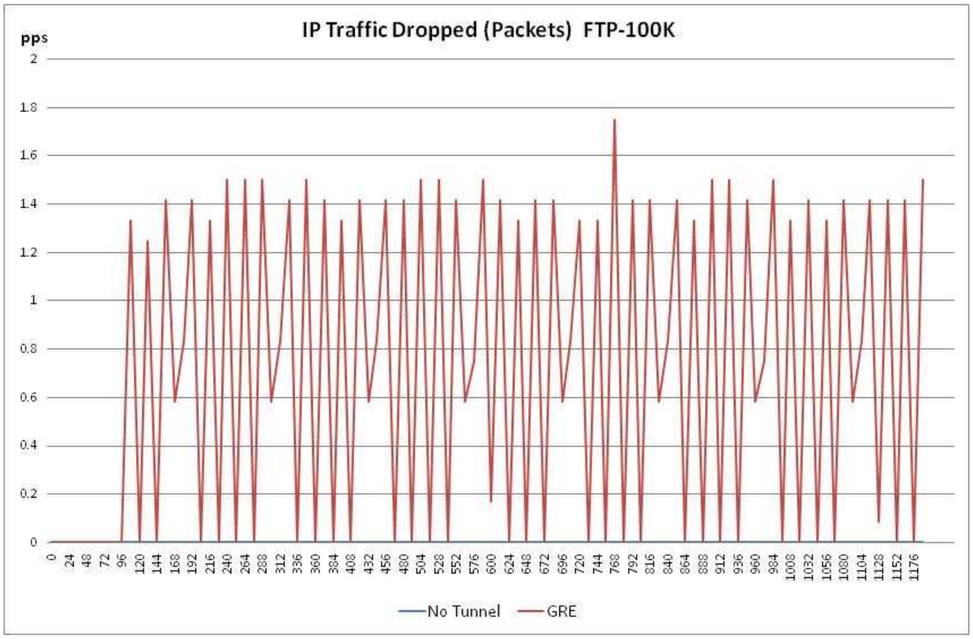


Figure B.10: IP Traffic Dropped (Packets) with GRE Tunnel and with No Tunnel for FTP 100K file size

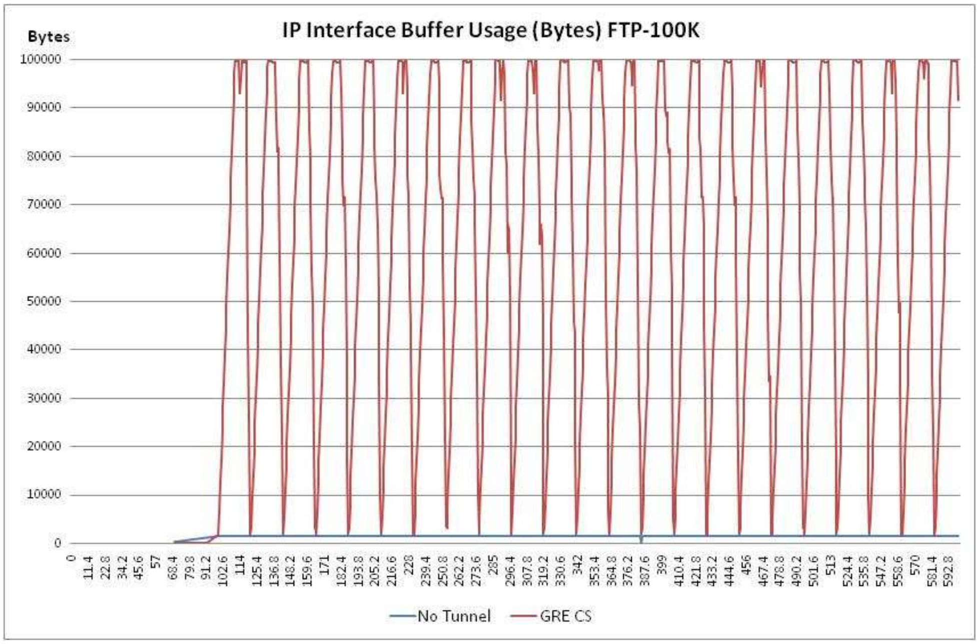


Figure B.11: Interface Buffer Usage (Bytes) with GRE CS Tunnel and with No Tunnel for FTP 100K file size

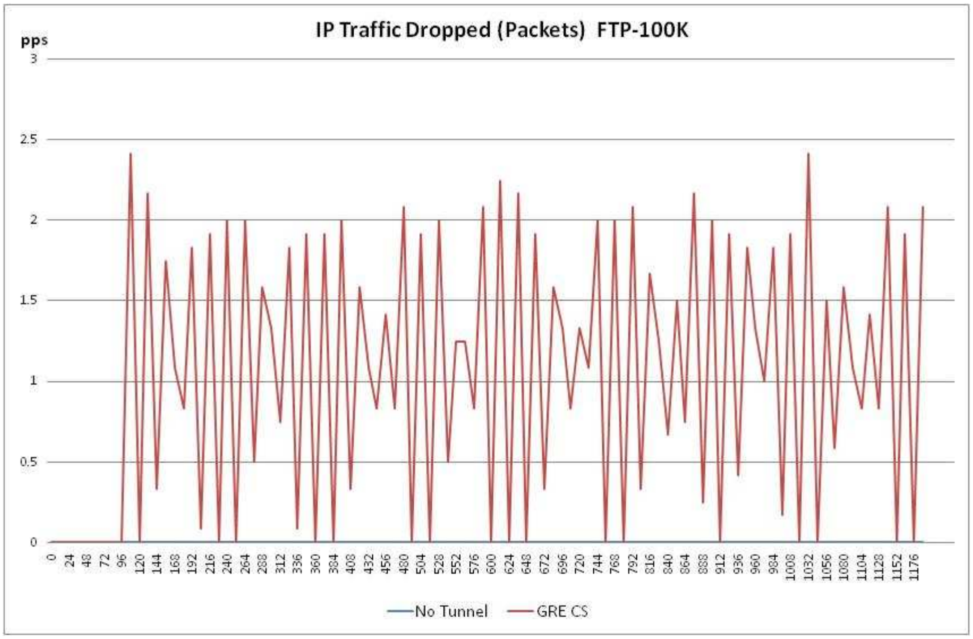


Figure B.12: IP Traffic Dropped (Packets) with GRE CS Tunnel and with No Tunnel for FTP 100K file size

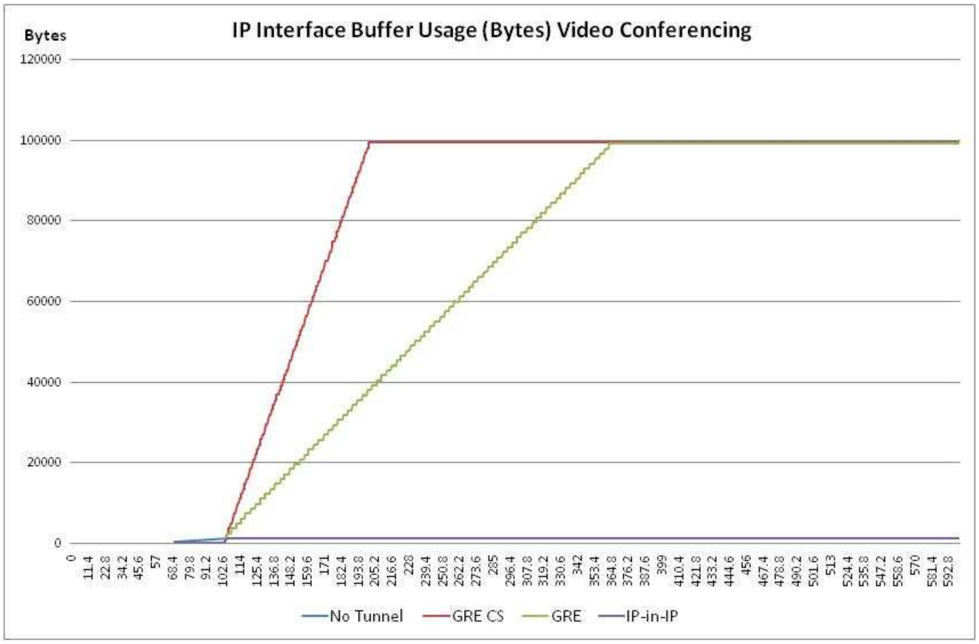


Figure B.13: IP Interface Buffer Usage (Bytes) with IP-in-IP, GRE, and GRE CS Tunnel and with No Tunnels for Video Conferencing

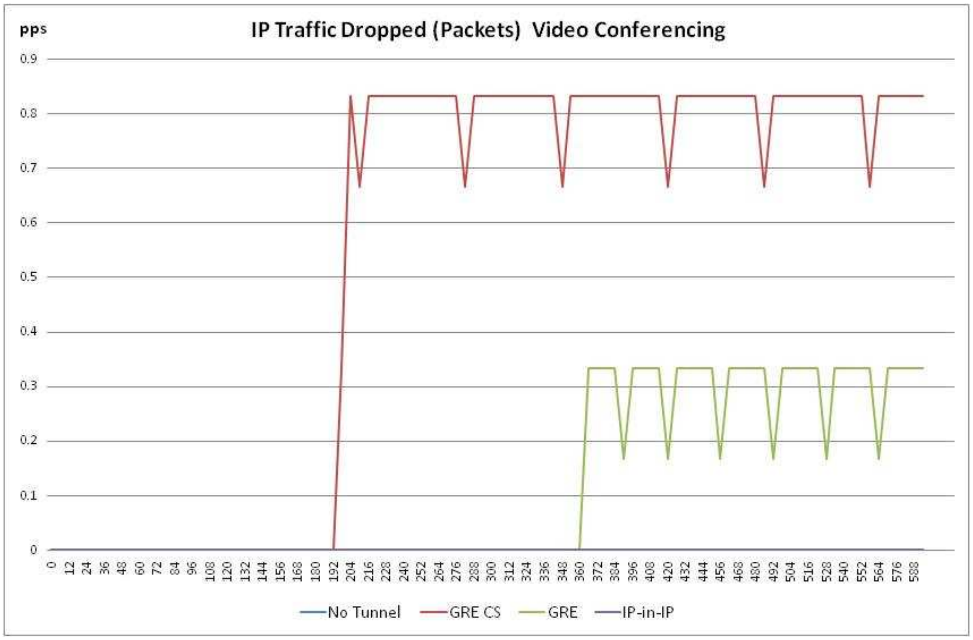


Figure B.14: IP Traffic Dropped (Packets) with IP-in-IP, GRE, and GRE CS Tunnel and with No Tunnels for Video Conferencing

Bibliography

[1] Neilsen. World internet usage statistic news and world population stats, http://www.internetworldstats.com. World Wide Web electronic publication, June 2010.

[2] Matthew Caesar and Jennifer Rexford. “BGP routing policies in ISP networks”. *IEEE Communications Society, Volume 19, Issue 6, Page(s): 5-11*, November - December 2005.

[3] Marwan H. Abu-Amara, Ashraf Mahmoud, Farag Ahmed Azzedin, and Mohammed Sqalli,. “Internet Access Denial by International Internet Service Providers: Analysis and Counter Measures”. *Research Proposal*, April 2008.

[4] Na Wang, Yingjian Zhi, Binqiang Wang. “AT: an Origin Verification Mechanism based on Assignment Track for Securing BGP”. *EEE International Conference on Communications, 2008 (ICC ’08), pp. 5739 - 5745*, 2008.

[5] P. Alin, P. Brian, T. Underwood. "Anatomy of a leak: AS9121". *http://nanog.org/mtg-0505/underwood.html*.

[6] J. Karlin, S. Forrest, and J. Rexford. "Pretty Good BGP: improving BGP by cautiously adopting routes". *in Proceedings Of ICNP,*, 2006.

[7] D. Drummond. "A new approach to china The Official Google Blog", http: //googleblog.blogspot.com/2010/01/new-approach-to-china.html, January 2010.

[8] J. Finkle and D. Bartz. "Twitter hacked, attacker claims iran link",Reuters, http://www.reuters.com/article/idUSTRE5BH2A620091218. December 2009.

[9] "WikiLeaks". *Wikipedia, http://en.wikipedia.org/wiki/WikiLeaks#cite_note-197*, 2011.

[10] OPNET Application and Network Poerformance. [Online]. http://www.opnet.com/.

[11] Y. Rekhter, T. Li, and S. Hares . "A Border Gateway Protocol 4 (BGP-4)". *RFC 4271(Draft Standard)*, January 2006.

[12] W. Simpson and Daydreamer. "IP in IP Tunneling". *IETF, RFC 1853*, October 1995.

[13] C. Perkins. “IP Encapsulation within IP”. *IETF, RFC 2003, www.ietf.org/rfc/rfc2003.txt*, October 1996.

[14] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. “Generic Routing Encapsulation (GRE)”. *IETE RFC 2784*, March 2000.

[15] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. “Layer Two Tunneling Protocol (L2TP)”. *IETF, RFC 2661*, August 1999.

[16] K. Hamzeh, G. Pall, J. Taarud, W. Little, and G. Zorn . “Point-to-Point Tunneling Protocol (PPTP)”. *IETF, RFC 2637*, July 1999.

[17] R. Atkinson S. Kent. “Security Architecture for the Internet Protocol”. *IETF, RFC 2401*, November 1998.

[18] Ke Zhang, Xiaoliang Zhao, and S.Felix Wu. “An Analysis on Selective Dropping Attack in BGP”. *IEEE International Performance, Computing, and Communications*, April 2004.

[19] Tao Wan and Paul C. van Oorschot. “Analysis of BGP prefix origins during Googles May 2005 outage”. *IEEE, Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, April 2006.

[20] J. Karlin. “A fun hijack: 1/8, 2/8, 3/8, 4/8, 5/8, 7/8, 8/8, 12/8 briefly announced by AS 23520”. *http://www.merit.edu/mail.archives/ nanog/2006-06/msg00082.html.*

[21] Kevin Butler, Toni Farley, Patrick Mcdaniel, and Jennifer Rexford . "A Survey of BGP Security". *AT&T Labs - Research, Florham Park, NJ, DRAFT VERSION, Vol. V, No. N*, April 2005.

[22] K. Butler, T. Farley, P. McDaniel, and J. Rexford. "A survey of BGP security issues and solutions". *Proceedings of the IEEE, vol. 98, pp. 100-122*, January 2010.

[23] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig. "Interdomain traffic engineering with BGP". *IEEE Communications Magazine, vol. 41, no. 5, pp. 122-128*, May 2003.

[24] B. Quoitin and O. Bonaventure. "A cooperative approach to interdomain traffic engineering". *Proceedings of Next Generation Internet Networks, Rome, Italy, pp. 450- 457, 18-20*, April 2005.

[25] Ahmad Salam Abdullatif Alrefai. "BGP based Solution for International ISP Blocking". *MS Thesis submitted to Deanship of Graduate Studies, King Fahd University of Petroleum and Minerals*, December 2009.

[26] M. Chuah and K. Huang. "Detecting Selective Dropping Attacks in BGP". *31st IEEE Proceedings Local Computer Networks 2006, Publication Date: 14-16 NoV, pp. 959-966*, November 2006.

[27] R. Younglove. “Virtual Private Network - How They Work”. *Computing & Control Engineering Journal*, December 2000.

[28] C. E. Perkins. “IP Mobility Support”. *IETF RFC 2002, Mobile IP Working Group*, October 1996.

[29] Zhao Aqun, Yuan Yuan, Ji Yi, and Gu Guangun. “Research on Tunneling Techniques in Virtual Private Networks”. *Proceedings of the International Conference on Communication Technology 2000, pp 691-697 vol.1*, 2000.

[30] Tarek Saad, Basel Alawieh, and Hussein T. Mouftah. “Tunneling Techniques for End-to-End VPNs: Generic Deployment in an Optical Testbed Environment”. *Semra Gulder, Communications Research Centre Canada, IEEE Communications Magazine*, May 2006.

[31] A. A. Joha, F. B. Shatwan, and M. Ashibani. “Performance Evaluation for Remote Access VPN on Windows Server 2003 and Fedora Core 6”. *IEEE 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Serbia, pp. 587- 592*, September 2007.

[32] S. Narayan, S. S. Kolahi, K. Brooking, and S. de Vere. “Performance Evaluation of Virtual Private Networks in Windows 2003 Environment”. *IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE)*, December 2008.

[33] S. Narayan, K. Brooking, and S. de Vere. “Network Performance Analysis of VPN Protocols: An empirical comparison on different operating systems”. *IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 2009)*, April 2009.

[34] S. Khanvilkar and A. Khokhar. “Virtual Private Networks: An Overview with Performance Evaluation”. *IEEE Communication magazine, vol. 42 no. 10, pp. 146-154*, October 2004.

[35] Shrinivasa Kini, Srinivasan Ramasubramanian, Amund Kvalbein, and Audun F. Hansen. “Fast Recovery from Dual Link Failures in IP Networks”. *IEEE INFOCOM 2009 proceedings*, 2009.

[36] Jian Wu, Ying Zhang, Z. Morley Mao, and Kang G. Shin. “Internet Routing Resilience to Failures: Analysis and Implications”. *Proceedings of CoNext 2007*, December 2007.

[37] AbdulAziz Muhammad Ali Al-Baiz. “Internet Denial by Higher-Tier ISPs: A NAT Based Solution”. *MS Thesis submitted to Deanship of Graduate Studies, King Fahd University of Petroleum and Minerals*, March 2010.

[38] OPNET Modeler., 2008.

[39] Xiaoliang Zhao, Beichuan Zhang, Dan Massey, and Lixia Zhang. “A Study on BGP AS Path Characteristics”. *USC-CSD Technical Report 04-818*, 2003.

[40] The Route Views Project. http://www.routeviews.org.

[41] "OPNET Modeler Product Documentation". *OPNET Technologies Inc.*, 2008.

[42] R. Fielding, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. "Hypertext Transfer Protocol - HTTP/1.1". *IETF, RFC 2616*, June 1999.

Vita

- Mohammed Abdul Khadir Khan Asif
- Nationality : Indian
- Born in Hyderabad, India on MAY 17th, 1985.
- Received BE (Bachelor of Engineering) degree in Electronics and Communication Engineering from Osmania University (OU), Hyderabad, India in May 2006.
- Joined Computer Engineering Department at KFUPM, Saudi Arabia as a Research Assistant in September 2007.
- Completed M.S. (Master of Science) degree requirements in Computer Engineering at KFUPM, Saudi Arabia in December 2010.
- Email: makhadirkhan@gmail.com (or) khadir@kfupm.edu.sa
- Mobile: (00966)-556493852
- Present Address : P.O. Box: 32, Dhahran, KFUPM 31261, Saudi Arabia.
- Permanent Address : 12-2-831/405 Peace Plaza, Mehdipatnam, Hyderabad-500028, Andhra Pradesh, India.