OULUN YLIOPISTO
UNIVERSITY of OULU

DEGREE PROGRAMME IN WIRELESS COMMUNICATIONS ENGINEERING

PERFORMANCE EVALUATION OF HIP-BASED NETWORK SECURITY SOLUTIONS

Thesis author	______________________ Jude Okwuibe
Thesis supervisor	______________________ Prof. Mika Ylianttila
Approved	_____ /_____ 2015
Grade	______________________

Okwuibe J. (2015) Performance Evaluation of HIP-based Network Security Solutions. University of Oulu, Department of Communications Engineering. Master's Degree Programme in Wireless Communications Engineering. Master's Thesis, 63 p.

ABSTRACT

Host Identity Protocol (HIP) is a networking technology that systematically separates the identifier and locator roles of IP addresses and introduces a Host Identity (HI) name space based on a public key security infrastructure. This modification offers a series of benefits such as mobility, multi-homing, end-to-end security, signaling, control/data plane separation, firewall security, e.t.c. Although HIP has not yet been sufficiently applied in mainstream communication networks, industry experts foresee its potential as an integral part of next generation networks.

HIP can be used in various HIP-aware applications as well as in traditional IP-address-based applications and networking technologies, taking middle boxes into account. One of such applications is in Virtual Private LAN Service (VPLS), VPLS is a widely used method of providing Ethernet-based Virtual Private Network that supports the connection of geographically separated sites into a single bridged domain over an IP/MPLS network. The popularity of VPLS among commercial and defense organizations underscores the need for robust security features to protect both data and control information.

After investigating the different approaches to HIP, a real world testbed is implemented. Two experiment scenarios were evaluated, one is performed on two open source Linux-based HIP implementations (HIPL and OpenHIP) and the other on two sets of enterprise equipment from two different companies (Tempered Networks and Byres Security). To account for a heterogeneous mix of network types, the Open source HIP implementations were evaluated on different network environments, namely Local Area Network (LAN), Wireless LAN (WLAN), and Wide Area Network (WAN). Each scenario is tested and evaluated for performance in terms of throughput, latency, and jitter.

The measurement results confirmed the assumption that no single solution is optimal in all considered aspects and scenarios. For instance, in the open source implementations, the performance penalty of security on TCP throughput for WLAN scenario is less in HIPL than in OpenHIP, while for WAN scenario the reverse is the case. A similar outcome is observed for the UDP throughput. However, on latency, HIPL showed lower latency for all three network test scenarios. For the legacy equipment experiment, the penalty of security on TCP throughput is about 19% compared with the non-secure scenario while latency is increased by about 87%. This work therefore provides viable information for researchers and decision makers on the optimal solution to securing their VPNs based on the application scenarios and the potential performance penalties that come with each approach.

Keywords: Host Identity Protocol, Security Control and Data Acquisition, Virtual Private LAN Services, Security.

Okwuibe J. (2015) HIP-pohjaisten tietoliikenneverkkojen turvallisuusratkaisujen suorituskyvyn arviointi. Oulun yliopisto, tietoliikennetekniikan osasto, Degree Programme in Wireless Communications Engineering. Diplomityö, 63 s.

TIIVISTELMÄ

Koneen identiteettiprotokolla (HIP, Host Identity Protocol) on tietoliikenneverkkoteknologia, joka käyttää erillistä kerrosta kuljetusprotokollan ja Internet-protokollan (IP) välissä TCP/IP-protokollapinossa. HIP erottaa systemaattisesti IP-osoitteen verkko- ja laite-osat, sekä käyttää koneen identiteetti (HI) -osaa perustuen julkisen avainnuksen turvallisuusrakenteeseen. Tämän hyötyjä ovat esimerkiksi mobiliteetti, moniliittyminen, päästä päähän (end-to-end) turvallisuus, kontrolli-informaation ja datan erottelu, kohtaaminen, osoitteenmuutos sekä palomuurin turvallisuus. Teollisuudessa HIP-protokolla nähdään osana seuraavan sukupolven tietoliikenneverkkoja, vaikka se ei vielä olekaan yleistynyt laajaan kaupalliseen käyttöön.

HIP–protokollaa voidaan käyttää paitsi erilaisissa HIP-tietoisissa, myös perinteisissä IP-osoitteeseen perustuvissa sovelluksissa ja verkkoteknologioissa. Eräs tällainen sovellus on virtuaalinen LAN-erillisverkko (VPLS), joka on laajasti käytössä oleva menetelmä Ethernet-pohjaisen, erillisten yksikköjen ja yhden sillan välistä yhteyttä tukevan, virtuaalisen erillisverkon luomiseen IP/MPLS-verkon yli. VPLS:n yleisyys sekä kaupallisissa- että puolustusorganisaatioissa korostaa vastustuskykyisten turvallisuusominaisuuksien tarpeellisuutta tiedon ja kontrolli-informaation suojauksessa.

Tässä työssä tutkitaan aluksi HIP-protokollan erilaisia lähestymistapoja. Teoreettisen tarkastelun jälkeen käytännön testejä suoritetaan itse rakennetulla testipenkillä. Tarkasteltavat skenaariot ovat verrata Linux-pohjaisia avoimen lähdekoodin HIP-implementaatioita (HIPL ja OpenHIP) sekä verrata kahden eri valmistajan laitteita (Tempered Networks ja Byres Security). HIP-implementaatiot arvioidaan eri verkkoympäristöissä, jota ovat LAN, WLAN sekä WAN. Kaikki testatut tapaukset arvioidaan tiedonsiirtonopeuden, sen vaihtelun (jitter) sekä latenssin perusteella

Mittaustulokset osoittavat, että sama ratkaisu ei ole optimaalinen kaikissa tarkastelluissa tapauksissa. Esimerkiksi WLAN-verkkoa käytettäessä turvallisuuden aiheuttama häviö tiedonsiirtonopeudessa on HIPL:n tapauksessa OpenHIP:iä pirnempi, kun taas WAN-verkon tapauksessa tilanne on toisinpäin. Samanlaista käyttäytymistä havaitaan myös UDP-tiedonsiirtonopeudessa. HIPL antaa kuitenkin pienimmän latenssin kaikissa testiskenaarioissa. Eri valmistajien laitteita vertailtaessa huomataan, että TCP-tiedonsiirtonopeus huononee 19 ja latenssi 87 prosenttia verrattuna tapaukseen, jossa turvallisuusratkaisua ei käytetä. Näin ollen tämän työn tuottama tärkeä tieto voi auttaa alan toimijoita optimaalisen verkkoturvallisuusratkaisun löytämisessä VPN-pohjaisiin sovelluksiin.

Avainsanat: koneen identiteettiprotokolla, SCADA, virtuaalinen LAN-erillisverkko.

TABLE OF CONTENTS

FOREWORD

This master's thesis was carried out at the Center for Wireless Communications, University of Oulu, Finland. The aim was to evaluate the performance of HIP-based security solutions on different network platforms.

I wish to express my appreciation to Professor Mika Ylianttila, for accepting to be my supervisor and for his continuous support and guidance during the course of this work. I also wish to express my gratitude to Professor Jari Iinatti for his role as my second supervisor. I am deeply grateful to my advisor, Madhusanka Liyanage for his guidance, encouragement, and unreserved help, more so for his meticulous effort in the preparation of the publications related to my thesis. Dr. Andrei Gurtov is acknowledged for his support and for being a source of inspiration through his erudite works in this field. Thanks to Ijaz Ahmed, for his pieces of advice and suggestions and for helping with proof-reading of this work, and to Nuutti Tervo for helping with the translation of my abstract. I also thank Lucas Kane and Ludwin Fuchs from Tempered Networks for their technical support all through the process of this work.

My colleagues from CWC are highly appreciated for creating a friendly working atmosphere and for their direct and indirect influence towards the content of this thesis. Finally, I would like to dedicate this thesis to all my family members for their love and motivation all through my studies.

Oulu, May. 2015

Jude Okwuibe

LIST OF ABBREVIATIONS AND SYMBOLS

VPN	Virtual Private Network
HIT	Host Identity Tag
CEK	Content Encryption Key
KEK	Key Encryption Key
KDC	Key Distribution Center
DH	Diffie–Hellman
ASes	Autonomous Systems
NAT	Network Address Translation
EPC	Evolved Packet Core
ACL	Access Control List
CDF	Cumulative Distribution Function
BEX	Base Exchange
VBO	VE Block Offset
VBS	VE Block Size
AGI	Attachment Group Identifier
LISP	Locator Identifier Separation Protocol
VSI	Virtual Switch Instance
FIB	Forwarding Information Base
FEC	Forwarding Equivalence Class
RRs	Route Reflectors
VE	VPLS Edge Device
u-PE	User-facing Provider Equipment
n-PE	Network-facing Provider Equipment
ICMP	Internet Control Message Protocol
NLRI	Network Layer Reachability information
RD	Route Distinguisher
AFIs	Address Family Identifiers
SAFIs	Subsequent Address Family Identifiers
ARP	Address Resolution Protocol
PW	Pseudowire
VC-LSPs	Virtual Circuit Label Switched Paths
SDN	Software Defined Network
LAN	Local Area Network
MAC	Medium Access Control
WAN	Wide Area Network
MPLS	Multiprotocol Label Switching
PPVPNs	Provider Provisioned VPNs
RTF	Route Target Filtering

FR	Frame Relay
ATM	Asynchronous Transfer Mode
IRTF	Internet Research Task Force
IETF	Internet Engineering Task Force
VPLS	Virtual Private LAN Services
VPWS	Virtual Private Wire Services
IPLS	IP-only LAN-like Service
CE	Customer Edge
PE	Provider Edge
H-VPLS	Hierarchical VPLS
DoS	Denial of Service
DDoS	Distributed DoS
HIP	Host Identity Protocol
H-HIPLS	Hierarchical HIP enabled virtual private LAN Service
S-HIPLS	Session key based HIP VPLS
HIPLS	HIP-enabled Virtual Private LAN Service
OSI	Open System Interconnection
L2TP	Layer 2 Tunneling Protocol
L2VPN	Layer 2 VPN
L3VPN	Layer 3 VPN
PPTP	Point-to-point Tunneling Protocol
DMZ	Demilitarized Zone
DSL	Digital Subscriber Line
BGP	Border Gateway Protocol
OAM	Operations, Administration, and Management
PDF	Probability Distribution Function
LSI	Local Scope Identifier
MTU	Maximum Transmission Unit
SCADA	Supervisory Control and Data Acquisition
$\mathrm{DH}_{\mathrm{init}}$	DH public value for initiator
$\mathrm{DH}_{\mathrm{resp}}$	DH public value for responder
K_{init}	Shared key for initiator
K_{resp}	Shared key for responder
$\circledast$	Convolution
Δt	Time difference

1. INTRODUCTION

By and large, enterprises are looking for various ways to optimize the customer experience, improve visibility, and maximize efficiency across the value chain. There is a global need for cost-effective networks that easily support an expansion plan with an adequate level of security and efficiency. To meet these objectives, enterprises operating from geographically dispersed sites require a shared Ethernet broadcast domain among different sites. Interconnecting sites through a pseudo-wire to emulate a point-to-point connection over packet-switching networks is one way to achieve this objective, however, this approach could be prohibitively expensive when enterprise offices are distributed across different continents. A growing number of enterprises are, therefore, leveraging on the use of Virtual Private LAN Service (VPLS) to offer Virtual Private Network (VPN) services across geographically dispersed sites. Such VPNs rely on already existing Internet infrastructure to provision as the public network that interconnects the geographically distributed private networks.

As an IP based network, all hosts and their locations on the Internet are identified using only their IP addresses. The idea of allowing a host and its location to be identified using only an IP address has been the design of the Internet architecture since early networks. This approach becomes somewhat inefficient with the ever increasing density of mobile hosts and will become even much more inconvenient with the advent of "The Internet of things" and next generations of networks. Mobile hosts frequently change their locations and so also their IP addresses, hence supporting real-time applications such as VoIP may experience adverse interruptions with such IP address changes. A similar limitation is experienced for multihoming networks, because upon IP address change, communicating hosts will need to re-authenticate before reestablishing their connections. Other possible challenges with this architecture is its susceptibility to Denial-of-Service (DoS) attacks and the ease of impersonation through IP address forgery [1].

HIP was developed as an internetworking architecture that separates the locator/identifier role of IP addresses and introduces a cryptographic host identity to serve the role of IP address at the transport layer [1, 2, 3, 4]. Since after it was standardized by IETF in 2000, HIP has been adopted by several telecommunications vendors and operators for internal activities. Open-source versions such as HIPL [5] and OpenHIP [6] are also available for different platforms and have been very useful to several research projects and industrial applications.

One key application area of HIP is in Virtual Private LAN Services (VPLS), as a Virtual Private Network (VPN), VPLS networks are prone to various kinds of security threats such as Denial of Service (DoS), Distributed DoS (DDoS), TCP reset attack, and IP spoofing. Several measures and architectures have been proposed and implemented to enhance the performance of VPLS services [7, 8, 9], however most of these architectures are still not able to provide the required level of security for clients on VPLS networks. Scalability is another issue of concern with early VPLS solutions, given that they were mostly flat architectures, hence could effectively support small to medium scale networks but unable to support larger networks [10].

In [10], authors proposed a hierarchical VPLS architecture based on Host Identity Protocol (HIP), named Hierarchical HIP enabled virtual private LAN Service

(H-HIPLS). This architecture provides some vital features that would enhance the security of the VPLS network, these features include authentication, confidentiality, integrity, availability, secure control protocol and robustness to the known attacks. It also sets to improve scalability on the control, forwarding, and security plane by minimizing the number of keys stored at the nodes as well as the number of established tunnels. Other proposed architectures include HIP-enabled Virtual Private LAN Service (HIPLS) [11], Session key based HIP VPLS (S-HIPLS) [9], and Hierarchical VPLS (H-VPLS) [8, 12]. Generally these architectures are still lacking in scalability in one or more planes. H-VPLS for instance is lacking in security plane, HIPLS is lack-ing in control, forwarding, and security planes, while S-HIPLS is lacking in control plane [9].

In this thesis, four separate scenarios are evaluated, two are Linux-based open source HIP implementations and the others are two sets of enterprise security equipment from separate manufactures. These network scenarios were evaluated for throughput, jitters, and latency. Hence we are able to analyze the effectiveness of these approaches and the corresponding costs on the performance of the network.

This thesis work is organized as follows: Chapter 2 provides information on the HIP architecture and its various applications, it also covers the architectural modifications of HIP and their corresponding impacts on the initial HIP design. Chapter 3 provides details on the fundamental concepts of VPN and how these functions can be implemented at different layers of the OSI. Chapter 4 focuses on VPLS as a use case of VPN, it covers the different types of VPLS, their architectural limitations, as well as security issues. It further presents information on various HIP-enabled VPLS implementations and a comparison of different architectures. Chapter 5 covers the actual testbed implementations for open-source Linux based HIP implementations and enterprise-based VPLS solutions, it also presents the analysis of experimental results for each scenario. Chapter 6 covers the discussions relative the experiment outcomes and the industrial applications of each solution, it further sheds more light on the foreseeable future trends of various HIP solutions. Chapter 7 presents an overall summary of the thesis work.

2. Host Identity Protocol (HIP)

HIP was approved by IETF as a host identification technology capable of separating the locator roles and end-point identifiers of IP addresses [2, 9]. Other than HIP, there are several other proposals on IETF designed to address similar security and mobility concerns like HIP, however these other proposals provide only partial solutions. Locator Identifier Separation Protocol (LISP) for instance, mainly provides scalabilty for the routing system. HIP however, provides a more comprehensive approach to securing end-to-end connections for mobile and multihoming devices [1, 2].

This identifier/locator split is the bed-rock of HIP and is viewed by industry experts as a major tool for revolutionizing the internet architecture. The HIP technique was significantly spurred by the advancement in public key cryptography and the increase in computational resources of hosts, hence host identities are secured using cryptographic mechanisms [1].

Technically, the use of HIP eliminates the need for IP addresses, instead of IP addresses, incoming packets use cryptographic host identifiers which are generated by the hosts themselves. This is basically straightforward in normal cases, but when it comes to meeting certain security policy requirements, then it could be a lot more complex. Moreover different network scenarios and their respective security requirements need to be taken to consideration when deciding how data is handled on the network [13].

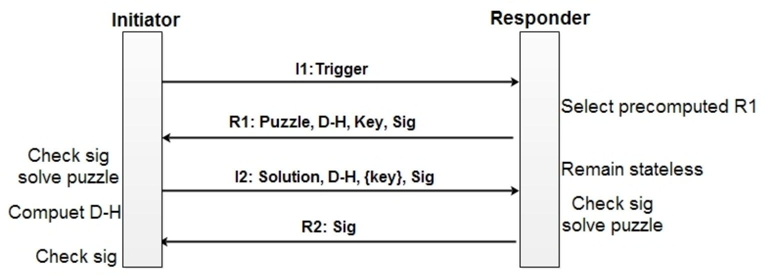


Figure 1. HIP base exchange.

HIP nodes (initiator and responder) authenticate through a four-way handshake called Base Exchange (BEX). By exchanging Diffie-Hellman keys in the 2nd and 3rd packets, and authenticating the parties in the 3rd and 4th packets as shown in Figure 1 [13], the four-way handshake makes HIP resilient to DoS attacks. The l1 packet initiates the authentication process, it uses a minimal HIP header to avoid memory intensive operation. A HIP-aware DNS provides the l1 packet with Host Identity Tags for both the initiator and the responder. The R1,l2, and R2 messages implement the standard authenticated Diffie–Hellman (DH) key exchange procedure. For a given set of peers, the DH public value is calculated as

$$\begin{aligned} \mathrm{DH}_{\mathrm{resp}} &= g^x \bmod p \\ \mathrm{DH}_{\mathrm{init}} &= g^y \bmod p \end{aligned} \tag{2.1}$$

where $g = 2$, p is a standard large prime number, and $x,y \in [1, p-2]$ are self-generated random values of the peers. The DH secrete is a shared key calculated by each peer as

$$\begin{aligned} K_{\text{resp}} &= DH_{\text{init}}^{x}, \text{and} \\ K_{\text{init}} &= DH_{\text{resp}}^{y} \end{aligned} \tag{2.2}$$

i.e., $K_{\text{init}} = K_{\text{resp}} = g^{xy} \text{mod} p$ [13, 14].

During BEX, Security Associations are exchanged between nodes before setting up an IPsec tunnel for secure data exchange [15]. The use of IPsec to implement HIP is perceived by industrial experts as the ideal HIP implementation procedure. IPsec pro-vides a cryptographic-based security for IP traffic on the network using Authentication Header (AH) and Encapsulating Security Payload (ESP) protocols. AH authenticates data origin and ensures the integrity of IP packets while ESP which is mostly applied in combination with AH, provides confidentiality and encryption services as well as integrity assurance. However, the integrity assurance from ESP does not cover the IP header fields except in cases where the header fields are already encapsulated using ESP.

HIP Architecture

The HIP architecture is based on the identity/locator split concept, hence HIP introduces a new layer between the transport and network layers of TCP/IP layered model as shown in Figure 2 [1]. The role of this new layer is mainly to perform identity/locator splitting based on public key using Host Identity (HI) name space. A Host Identity Tag (HIT) is a 128-byte harsh of HI, this could be used by applications as an alternative to IPv6 address. The fixed length of HIT provides consistent format for the protocol for all cryptographic algorithms and simplifies the management of packet size cost [4]. Notwithstanding, HITs are only routable at the overlay on top of IP and not on the IP layer itself, hence using them as replacements for IPv6 can only happen on the application layer [2].

With identity/locator split, the IP address continues to serve as locator while Host Identifiers serves for identity [4]. This architectural modification with HIP offers dif-ferent benefits to both hosts and network operators, these include easing out mobility and multihoming across different address families, end-to-end encryption, eliminating the need to re-authenticate with an end hosts when moving between two domains con-trolled by different Network Address Translation (NAT) devices. Earlier HIP architec-tures relied on end-to-end signaling for key management and IP mobility management between pairs of end-hosts. These architectures were faced with major scalability is-sues originating from the terminal-based control which requires a high network and computational overhead on both the user and the access network, hence HIP solutions were only suitable for small to medium size networks and not for larger networks. The need to extend the benefits of HIP to larger networks resulted in some modifications to the initial HIP architecture. Two of these modifications are covered in this section; the Ultra Flat Architecture and the Hierarchical architecture [16, 14, 10].

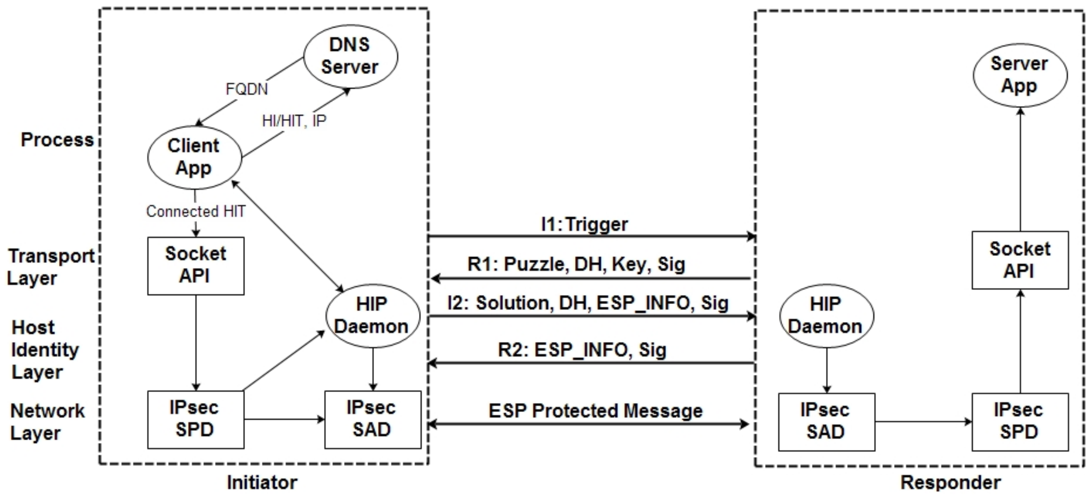


Figure 2. HIP architecture.

3. VIRTUAL PRIVATE NETWORKS (VPNs)

A Virtual Private Network (VPN) is a communication environment created through the controlled segmentation of a shared communications infrastructure to emulate the characteristics of a private network [17]. VPN supports a closed community of autho-rized users, providing them a secure access to various network related services and re-sources even when they are cast far around the world [17, 18]. As shown in Figure 3, a typical VPN consists of multiple private networks and remote users interconnected over a public network such as the internet. Early VPNs relied on the use of tradi-tional leased line technologies such as T1 and T3, however when sites are spanned across wider geographical locations, dedicated private lines become prohibitively ex-pensive to maintain. At present, VPN connections can run through almost any internet connection, these includes dial-up, Digital Subscriber Line (DSL), wireless, and satel-lite. These technologies leverage on the economics of scale to provide more affordable VPN services to customers.

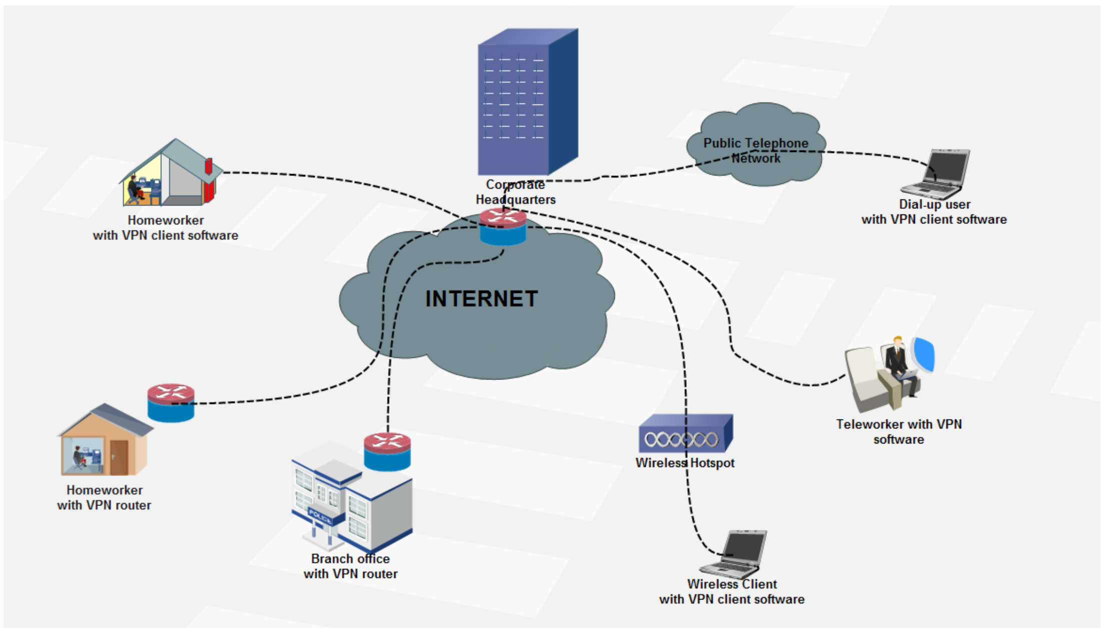

Corporate Headquarters
Public Telephone Network
Homeworker with VPN client software
Dial-up user with VPN client software
INTERNET
Teleworker with VPN software
Wireless Hotspot
Homeworker with VPN router
Branch office with VPN router
Wireless Client with VPN client software

Figure 3. VPN Connectivity.

In [17], author describes three primary types of VPN services; ***LAN Interconnect*** VPN services, ***Dial-up*** VPN services, and ***Extranet*** VPN services. LAN Interconnect VPN provides network services for geographically dispersed sites over a shared net-work infrastructure. Dial-up VPN provides point-to-point-like VPN services to mobile and telecommuting employers, allowing them to access the company's Intranet from remote locations as shown in Figure 3. Extranet VPN is an architectural hybrid of LAN interconnect and dial-in VPN, however as an extranet, it enables external ven-dors, suppliers, and customers to access specific areas of the company's intranet called the Demilitarized Zone (DMZ).

VPNs have been categorized in very many different ways [19, 20, 21, 22, 23]. In [19], author presents a comprehensive approach to categorizing VPNs, this is illustrated in Figure 4, with this approach, the provisioning agent is considered paramount, hence categorizing VPNs primarily into Provider Provisioned VPNs (PPVPNs) and Customer Provisioned VPNs.

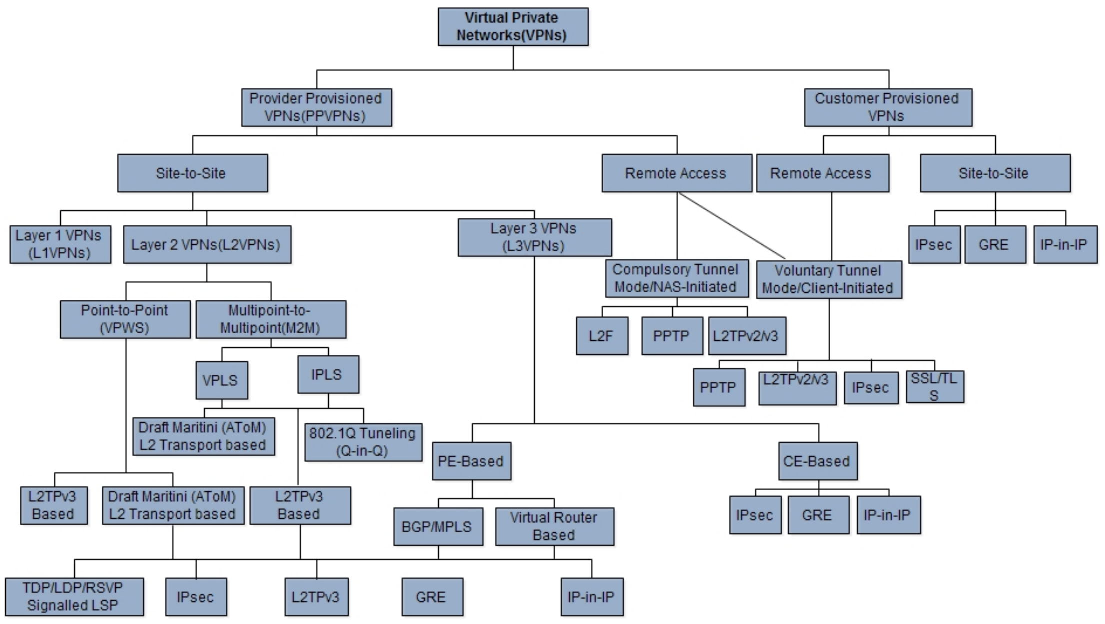


Figure 4. VPN classification.

3.1 Provider Provisioned VPNs (PPVPNs)

Provider provisioned VPN is the most common form of VPN in the industry at present. Here the core VPN functionality is configured and managed by the network service provider. Figure 5 presents a typical PPVPN connectivity showing the interconnection of multiple VPNs in an extranet. The goal is for service providers to securely deliver data and extend connectivity over one or more shared networks, with predefined s ervice l evel a ssurances. P PVPNs c ould o perate o n e ither l ayer 2 o r l ayer 3, however the generic requirement for all provider provisioned VPN technologies are described in [24]. This document was prepared by the design team in the PPVPN working group. The goal is to develop generic requirements for PPVPNs. These requirements are in three distinct categories, namely service, provider, and engineering requirements. These requirements are independent of any particular type of PPVPN technology, hence all PPVPN technologies are expected to meet the requirements stated in this document. These requirements are also independent of the deployment scenario. However, these requirements may differ based on the number of providers provisioning the VPN, i.e. single or multiple provider networks and/or Autonomous Systems (ASes). Requirements specific to layer 2 PPVPNs are contained in [25], while those specific to layer 3 PPVPNs are contained in [26].

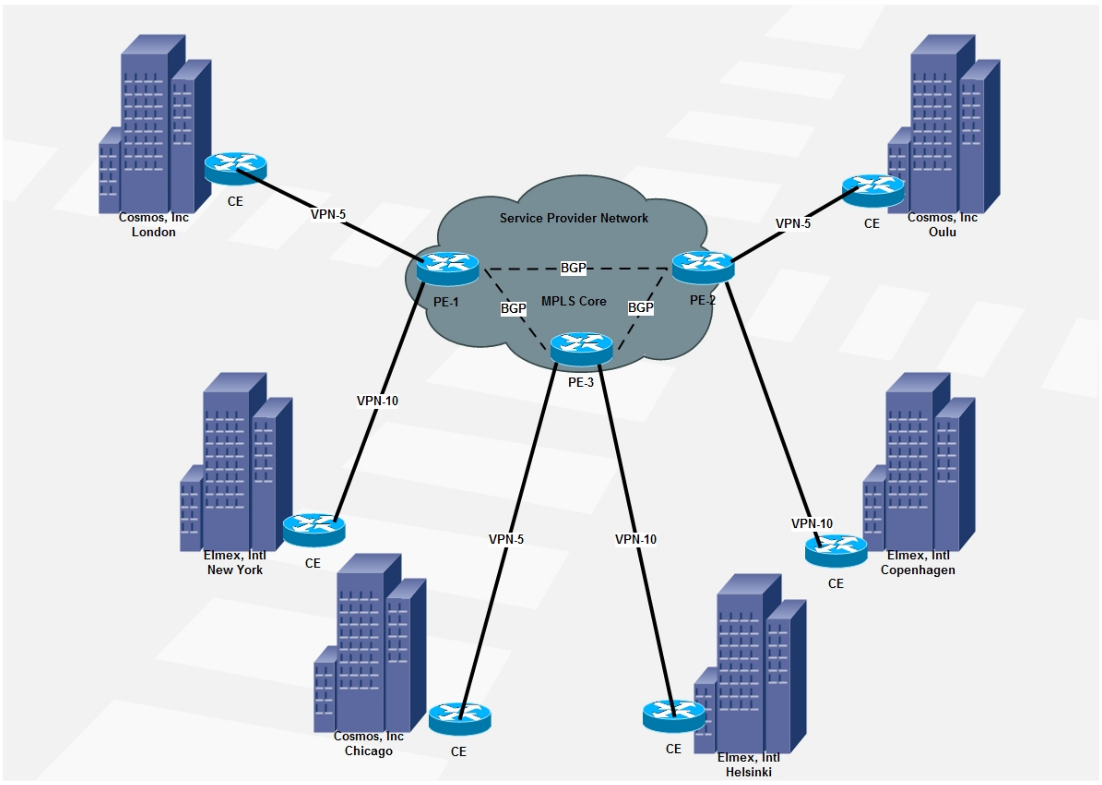


Figure 5. Provider provisioned VPN connectivity.

3.2 Customer Provisioned VPNs

A rare but evolving way to implement VPN is the Customer Provisioned VPN. Here the customer becomes the VPN service provider, thereby overseeing the configuration and management of the VPN [19]. As shown in Figure 6, a typical customer provisioned VPN is designed to technically separate the connection service of the provider from the VPN service, here instead of having the network service provider also providing the VPN services, the customer is able to use special networking equipment that allows for managements and provisioning of a VPN by the customer. A use case of this approach is evaluated in the implementation phase of this thesis. Companies like SnapGear and Tempered Networks are producing specialized hardware solutions that provide commercial-grade VPN services at commodity prices. Tempered Networks for instance provides hardware solution for protecting industrial control systems and devices, this solution consists of three main components: segmentation, secure connectivity and remote monitoring, and managing third-party connections [27]. These three together effectively eliminate the need for a service provider to be involved in the VPN management. Several benefits come with this approach. First, given the same level of security in terms of encryption and encapsulation mechanisms, this approach is invariable one step ahead in the security race, this is because it eliminates all security threats associated with a third party unlike the PPVPN approach. However secure the network is designed to be, the provider still has access to all data transmitted over their network infrastructures, hence posing a major concern on data confidentiality.

Other benefits associated with this approach include flexibility, cost-effectiveness, and high availability.

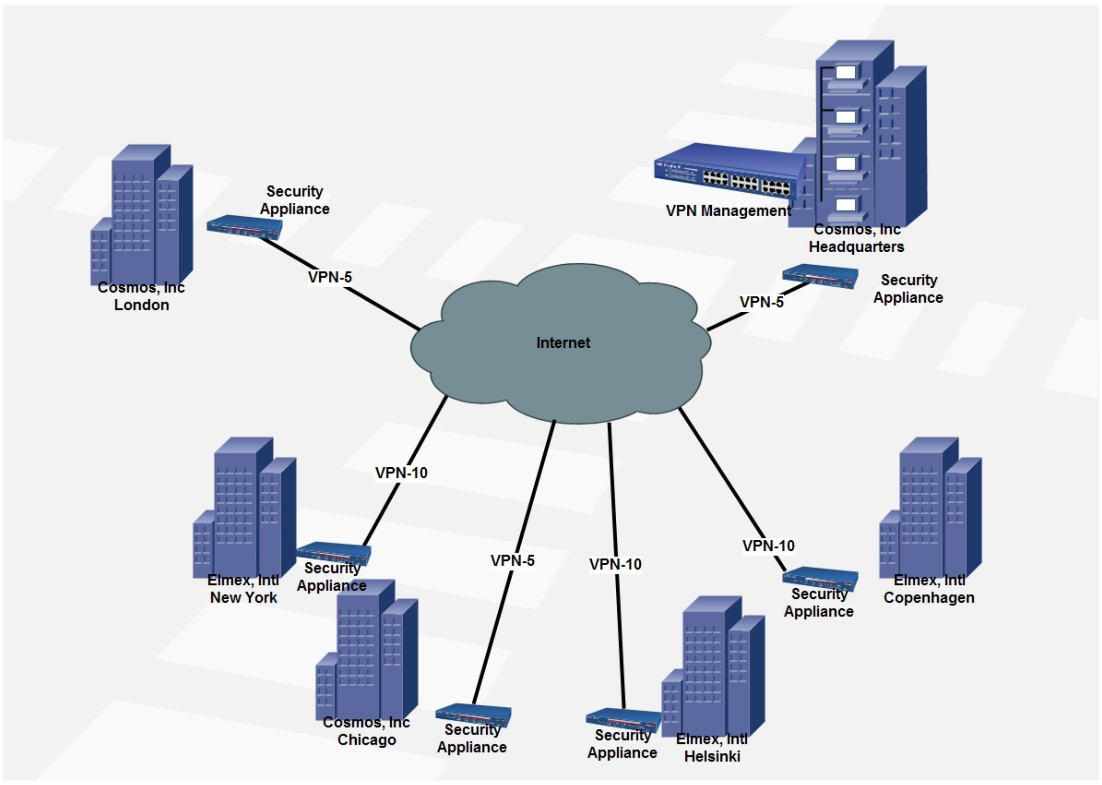


Figure 6. Customer provisioned VPN connectivity.

Another widely used paradigm to categorizing VPNs is based on the layer where they are implemented. Typical VPNs are implemented on network and data link layers, implementation in other layers such as application and physical layers are also possible. However in this section, the focus is on Layer 2 VPN (L2VPN) and Layer 3 VPN (L3VPN), we study their security and tunneling mechanisms as well as the network scenarios they are the most suitable for supporting.

3.3 Layer 2 VPN (L2VPN)

Layer 2 VPN is implemented at the link-layer of shared network infrastructure based on a switched link-layer technology like Frame Relay (FR) and Asynchronous Transfer Mode (ATM) [17]. A typical L2VPN is shown in Figure 7 [28]. In this model, L2VPN A represents a point to point service while L2VPN B represents a bridge service. There are three categories of services associated with the L2VPN framework; Virtual Private Wire Services (VPWS), Virtual Private LAN Services (VPLS), and IP-only LAN-like Service (IPLS) [20, 28]. The implementation work in this thesis is based on VPLS, hence the next chapter presents a detailed study of this group of services and why it is a more sought-after option for implementing VPNs, we also cover its architectural limitations and security concerns.

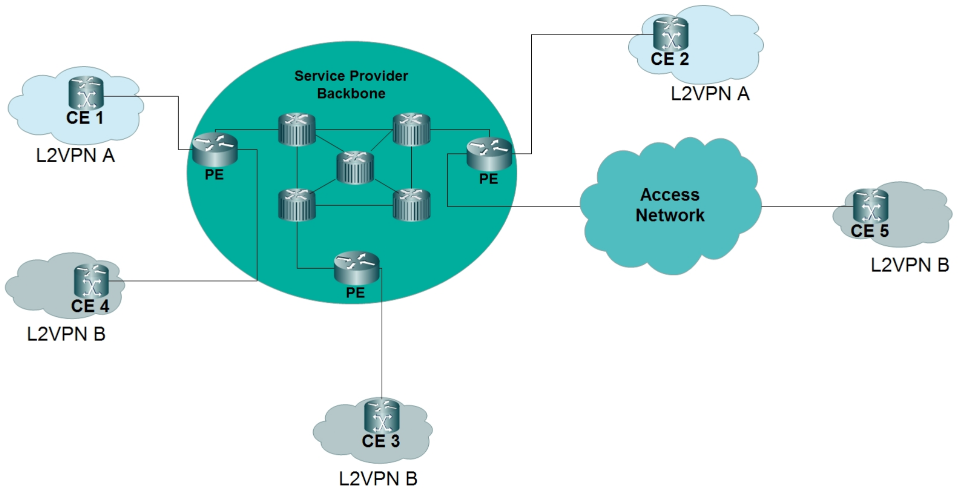


Figure 7. Layer 2 VPN Connectivity.

3.3.1 Virtual Private Wire Services(VPWS)

VPWS is a point-to-point service where Customer Edges (CEs) are interconnected via point-to-point virtual circuits. Each CE device in VPWS has a set of point-to-point vir-tual circuits which are used to interconnect with another CE device, hence frames are directly transmitted from one source CE device to a destination CE device irrespective of the content [20]. VPWS makes the integration of existing Layer 2 and Layer 3 ser-vices possible on a point-to-point basis across a service provider's IP/MPLS cloud [29]. As shown in Figure 8, VPWS is limited in its inability to support multipoint-to-multipoint communications.

3.3.2 IP-only LAN-like Service (IPLS)

IPLS is specially designed for transmitting only IP packets. It is a special case of VPLS where CE devices are hosts and routers instead of switches. IPLS services can only support IP-based packets such as Internet Control Message Protocol (ICMP) and Address Resolution Protocol (ARP) packets, hence other non-IP-based Layer 2 packets are not supported [20, 28]. However, IPLS still stands as the suitable solution for scenarios where MAC address learning capabilities are not required for the VPLS services, hence user data are restricted only to IP. PE devices interconnected using IPLS implement multipoint connectivity for IP traffic using either *discovery* where each PE device discovers IP/MAC address association for attached CE devices for each IPLS instance or *Pseudowire (PW) for Unicast Traffic*, in which case a PE device sets up a pseudo-wire called Virtual Circuit Label Switched Paths (VC-LSPs) to each of the other PEs that supports the same IPLS-instance [20].

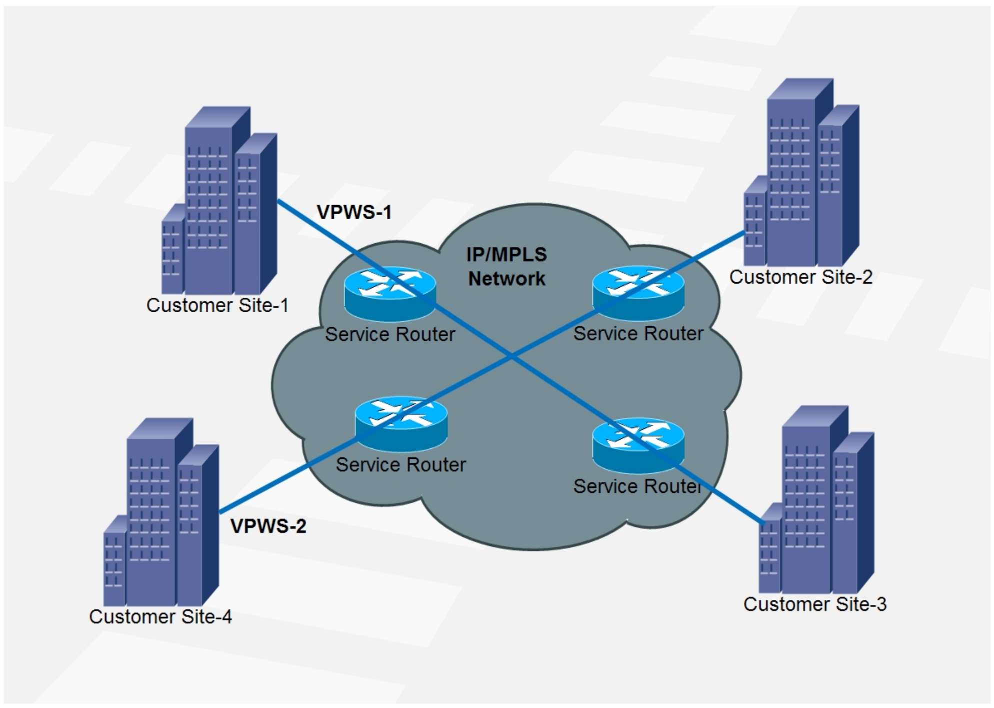


Figure 8. Virtual private wire service.

4. VIRTUAL PRIVATE LAN SERVICES (VPLS)

Virtual Private LAN Services is a multipoint layer 2 VPN that provides bridged LAN services to CEs that are members of the VPN, hence allowing CEs that belong to the same service category to behave as though they were connected through a bridged LAN. VPLS is sometimes referred to as Transparent LAN Service or Virtual Private Switched Network Service [7]. As a private LAN service, VPLS ensures that only CE devices belonging to the same VPLS instance are able to interact. PE devices per-form control driven interactions to discover all other PEs on the same VPLS, through this interaction they are able to exchange demultiplexors. In contrast to VPWS which offers point-to-point services, VPLS supports Ethernet based multipoint-to-multipoint communication over IP or MPLS (Multiprotocol Label Switching) based cooperate networks, hence most enterprises and organizations are more inclined to adopt VPLS instead of VPWS. VPLS also provides high-speed connectivity over Layer 3 (L3) provider network as well as other advanced auto discovery features [30, 20].

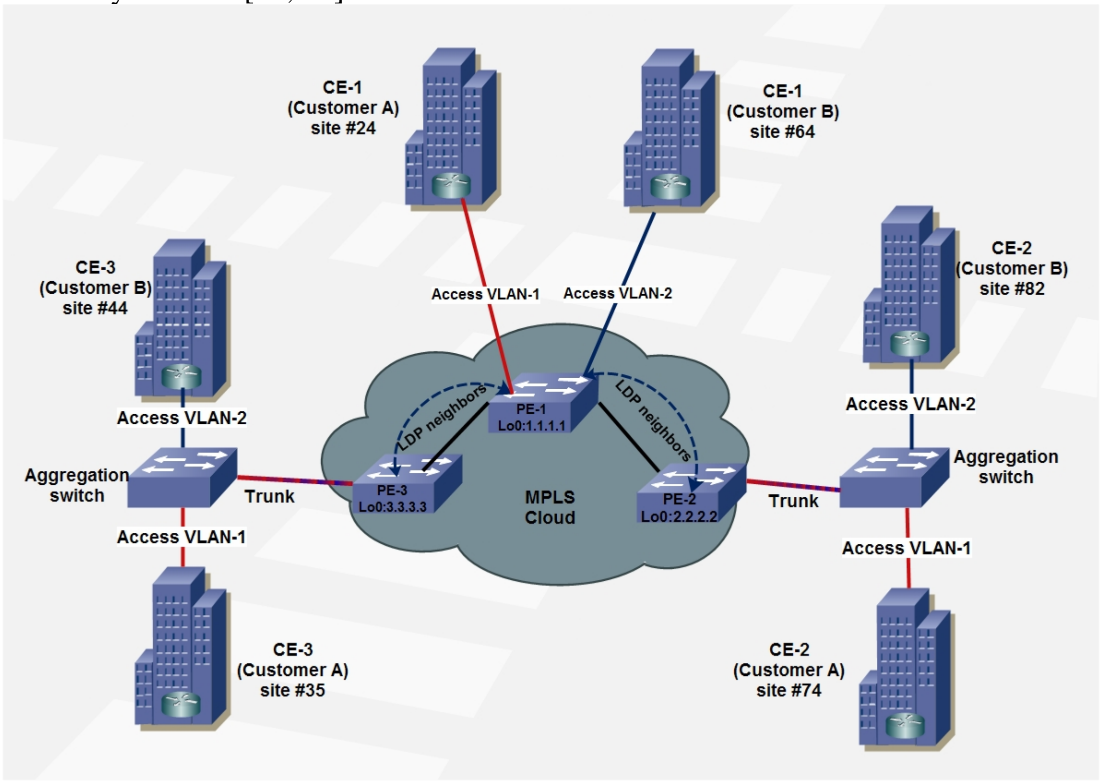


Figure 9. VPLS Connectivity.

Figure 9 is a typical VPN using LDP signaling as illustrated by Martini and Kompella in [31]. In this illustration, two separate customers (Customer A and Customer B) each operating from multiple sites spread across a metro or wide area are provided VPN services by the same provider over an MPLS cloud. The goal of the provider is to emulate the need for a dedicated switch to each customer. To accomplish this, the CE devices of two customers from different sites are grouped together and each group is assigned a single PE device. However on the management platform, each customer is assigned a particular access VLAN different from other customers, this way all traffic belonging to customer A irrespective of the site is only accessible from VLAN-1, the same holds for customer B, hence making the MPLS cloud appear like a dedicated switch to each customer. Although for some providers, the configurations might be more complex than this especially when multiple providers are involved, however the basic idea remains the same, hence giving service providers a scalability option to leverage on the economics of scale to provide VPN services to multiple customers over the same set of equipment, making services more affordable for customers.

The ability of VPLS to merge multiple LANs into a single virtual LAN is made possible through the incorporation of MAC address learning, flooding, and forwarding functions in the pseudowires that connect the individual LANs across the packet switched network [7]. VPLS could provide optimal solution for connecting multiple customer sites, especially for customers who wish to connect a relatively high number of sites with very few users per site to a corporate LAN or when high number of routers need to be interconnected without involving the provider in the customer routing [20].

To establish a VPLS, mesh of LSPs is pre-established between all the PEs participating in the VPLS instance. This could be done manually or by the use of a provisioning system or signaling protocols. Internet Engineering Task Force (IETF) currently has two standard frameworks for building VPLS, these frameworks differ predominantly in their control planes, particularly the signaling techniques i.e. the protocol used to set up the MPLS tunnel and distribute labels between PEs, and auto-discovery, i.e. what technique is used to enable multiple PE routers in the same VPLS instance to discover each other. Based on these two broad criteria, IETF have standardized *VPLS over LDP* [8] and *VPLS using BGP for auto-discovery and signaling* [7]. The follow-ing sections discuss the dynamics of these two approaches, how they differ, and the respective suitable scenarios they support.

4.1 VPLS Using BGP for Auto-Discovery and Signaling

This draft was proposed in 2007 by Kompella and Rekhter and was standardized by IETF. The draft describes the functions required to offer VPLS, a mechanism for signaling a VPLS, the auto-discovery of the endpoints of a VPLS as well as the rules for forwarding VPLS frames across a packet switched network [7]. One distinguish-ing feature of this approach is the auto-discovery and signaling mechanism which uses BGP as the control plane protocol. This document also covers the possible options for deployment with regard to the decoupling of functions across different devices. Unlike the approaches described in [32] and [33] which offer point-to-point Ethernet services across packet switched networks, this approach offers multipoint services.

4.1.1 BGP VPLS Operation

To set up a BG VPLS according to the approach in [32], a network administrator needs to first pick an RT for the VPLS, this RT will be used by all the PEs in the VPLS. To configure a particular PE, the network administrator only needs to choose the VE ID V for the PE, however for cases where the given PE is connected to a u-PE, then there may be need for multiple VE IDs. Such PE could also be configured with a Route Distinguisher (RD) to avoid it generating a unique RD for the VPLS.

4.1.2 Hierarchical BGP VPLS

Scalability is a key property that makes VPLS a viable option to offer VPN services to multiple customers. BGP allows for the scalability of the control plane in a VPLS. Scalability requires that the operations performed on the VPLS are kept to the barest possible minimum, hence for BGP to offer control plane scalability on a VPLS, three key aspects are put to consideration: 1. Minimizing the mesh connectivity among the VPLS BGP speakers. 2. Restricting the message passing of the BGP VPLS to only the intended speakers. 3. Removing the complexities involves in adding and deleting a BGP speaker on the VPLS.

One comprehensive approach to addressing these three issues is by using BGP for internet routing and IP VPNs. This technique is hierarchical and relies on designating a set of fully meshed Route Reflectors (RRs) and establishing a BGP session between each BGP speaker and one or more RRs [34]. By so doing the problem of having all BGP speakers connected in a fully meshed topology is solved. In a situation where the service provider needs a large number of RRs, this technique can also be applied recursively to create another level of RRs which still tenders the need for full mesh. This also removes the complexities involved in adding and deleting multiple BGP speakers.

4.2 VPLS Using LDP Signaling

This draft was also proposed in 2007 by Lasserre and Kompella and has equally been standardized by IETF. It is important to mention that this standard performs a similar function as the one described in the previous section, hence they are both referred to as VPLS. However, they differ in the signaling protocols they use, and thus not compatible with each other. As a VPLS, this standard also relies on the use of pseudowires to emulate a LAN segment for a given set of users by creating a Layer 2 broadcast domain through which MAC address learning and forwarding is carried out on the Ethernet. It is also possible for a single provider to offer multiple VPLS services on the same set of PE nodes. The standard proposed in this document is an extension of the pseudowire label signaling on the control plane using LDP [33].

Just as we have in traditional LANs, when multiple sites on a VPLS are set on the same broadcast domain, it is expected that all packets (unicast, multicast, broadcast) are forwarded to the right locations. Therefore there is always a need for MAC address learning and aging on each pseudowire. Flooding and packet replication as described in the previous section will still be required with this standard. In [35], authors describe how Layer 2 frames are carried over point-to-point pseudowires, this draft describes how Ethernet/802.3 and VLAN 802.1Q traffic are transported across multiple sites on the same Layer 2 broadcast domain or the same VPLS. It is equally possible to apply this model to 802.1 technologies. This proposal presents a concise and scalable way to offer VPLS without need for address resolution servers or other external servers as described in [25].

VPLS Operation and Encapsulation Actions

As shown if Figure 10, the VPLS in this example is set up between PE1, PE2, and PE3. It is set to connect customers across four depicted sites, A1 - A4 using CE1 - CE4 respectively.

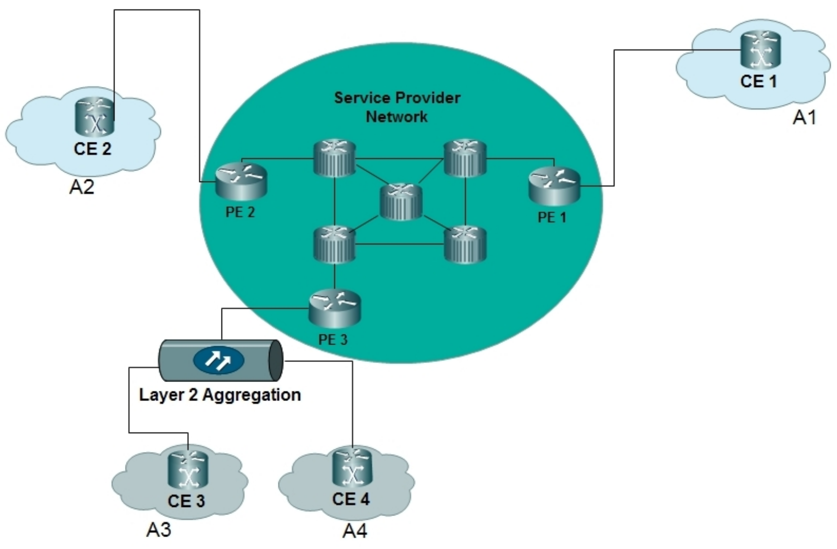


Figure 10. Operation of VPLS with layer 2 aggregation.

From this configuration, one can infer that the initial VPLS instance was between PE1, PE2, and PE3 using the full mesh of Ethernet PWs. This VPLS instance is assigned a VPLS ID accordingly. To illustrate a typical VPLS operation, we assume that PE1 signals a PW label 54 to PE2 and 55 to PE3, and PE2 signals PW label 64 to PE1 and 65 to PE3. Assume that a packet from A1 with a source MAC address M1 and destination MAC address M2 is bound for A2. On leaving C1, we assume the PE1 is unaware of the location of M2, then it must flood the packet to PE2 and P E3. On receiving this packet, PE2 will notice that it has a PW label of 64, hence the source MAC address M1 must be behind PE1, given that it distributed the label 64 to PE1 and by logical assumption it can associate the MAC address M1 to PW label 55 [33].

For this VPLS, when the customer Ethernet frame comes without a preamble, then it is encapsulated with a header as described in [35]. To determine the encapsulation need for a frame, two basic rules are applied:

1. If upon arrival at the PE the frame is already encapsulated with a customer frame and is also used by the local PE as a service delimiter, then the encapsulation is preserved as the frame is sent to the VPLS since this information is required to identify the customer and the particular service of that customer. An exception to this case is when the requested VLAN ID is signaled, in which case the VLAN tag is overwritten before the frame is sent out on the PW.

2. If upon arrival at the PE the frame is already encapsulated, but this encapsulation does not have appropriate VLAN tag, then a null tag is imposed if the VLAN ID optional parameter is not signaled.

Hence a customer frame may arrive at a customer-facing port with a VLAN tag that identifies both the customer's VPLS instance and VLAN, the encapsulation of such tag is preserved. A VPLS could have both Ethernet and Ethernet VLAN PWs, however if the PE is unable to support both, then it needs to send a label release on the PW messages that it cannot support with a status code "Unknown FEC" as required in [36][8].

4.3 Security Considerations

The VPLS standards discussed in previous sections both have general and unique security considerations. Given that VPLS is aimed at ensuring data privacy, these security concerns pose major threats to both the customer and the service provider networks. For instance, a man-in-the-middle eavesdropping can alter the transmitted data stream. For the general case, both approaches are still exposed to lack of data confidentiality and privacy which is typical of VPLS since it does not offer authentication. A more detailed description of the security issues involved in L2VPNs is covered in [37]. To secure such communication, all PE-to-PE tunnels in the provider network are encrypted with IPsec. To further improve security, the end systems in the VPLS sites may need to encrypt all data using appropriate technique before sending through the provider network. Generally, the security concerns for both VPLS standards are categorized into two main aspects; data plane security and control plane security.

4.3.1 BGP based VPLS

Securing the control plane of VPLS is very crucial to achieving data privacy, if the control plane is not properly secured, then chances are the transmitted data is misplaced on the way or worse still sent to an eavesdropper. For a BGP based VPLS, all the exchanges on the control plane are done via BGP messages. To enhance the security of such BGP messages, certain techniques such as TCP extension are are used to authenticate BGP messages as described in [38]. Using this approach, the TCP carries an MD5 digest as extension, MD5 is a widely used cryptographic harsh function when it comes to verifying data integrity. Each segment sent on a TCP will contain a 16-byte MD5 digest, this MD5 acts like a signature for the TCP segment and since it uses information known to only the end systems, it becomes an effective way for the BGP to protect itself from spoofed TCP segments that might be introduced into the stream. For a spoofing attack to be successful on this scheme, the attacker would not only have to guess the TCP sequence number, but would also need to guess the password included in the MD5 digest which is even more challenging since this password is never displayed on the connection stream and could take different forms depending on the application in use.

Not withstanding, the effectiveness of this technique is limited when it comes to concerted attacks. For instance, it may not help in keeping VPLS label private, and with full knowledge of the labels, it is possible for an intruder to eavesdrop on VPLS traffic, however such intruder will also require access to the data path within the service provider network and this underscores the need to equally secure the data plane of a VPLS [7].

For data plane security, the provider must ensure that all PE-to-PE tunnels on the VPLS are well controlled and verify that VPLS labels are coming from valid interfaces. Validity of an interface depends on the devices involved, for instance a valid interface for a PE comprises of links from P routers while for an ASBR, a valid interface is a link from an ASBR in an AS that is part of a given VPLS. In [39], authors described two key tunnel security mechanisms based on MPLS, namely MPLS-in-IP and MPLS-in-GRE, using such tunnels to secure VPLS packets according to this description require IPsec, otherwise IP address filtering becomes the only means of ensuring that packets exiting PEs were rightfully placed by the proper tunnel head node.

4.3.2 LDP based VPLS

Similar to BGP based VPLS, using LDP also requires security on both the data and control planes of the VPLS. In addition, this standard also specifies a special technique for addressing DoS attacks.

On the data plane, per VPLS L2 FIB table and per VPLS PWs are used to ensure traffic isolation between VPLS domains. Again to ensure data security, customer traffic is first encrypted and/or authenticated before getting to the provider network, since these packets remain unchanged as they move through the VPLS. The use of routers as CPE devices could help prevent broadcast storm. Another way to achieve this is by controlling and regulating the amount of broadcast traffic from each customer [8].

For control plane security, various LDP authentication methods are specified in [36]. The goal of this approach is to prevent the disruption of PEs by unauthenticated messages. LDP messages are categorized as follows [36]

1. *Discovery messages*: Used to announce and maintain an LSR in a network.
2. *Session Messages*: Used for session establishment, maintenance, and termination between LDP peers.
3. *Advertisement Messages*: Used to create, change, and delete label mappings for FECs.
4. *Notification Messages*: used for error signaling and advisory.

To avert DoS attack, it is imperative that service provider minimizes the number of MAC addresses known by each PE in the VPLS network.

4.3.3 Other Considerations

Generally, the HIP technology provides a network layer security alternative to standard security technologies like Secure Sockets Layer (SSL) and Transport Layer Security (TLS). HIP works favorably well with any transport protocol, its ability to handle host mobility and perform multihoming makes it a more desirable alternative to TLS which does not support such services. As far as security goes, technologies like TLS are more vulnerable to various TCP and DoS attacks. In addition, applications that are designed to use TLS must have the mechanism integrated into the actual system design whereas it is possible to provide the required security feature as an add-on to the already existing system through the use of HIP [1].

Another major challenge on traditional Internet architecture that the HIP solution tends to address is DoS attacks, a popular example is the TCP SYN attack. The routing of packets on TCP involves the initiator sending SYN packets to the recipient and the recipient replies with an acknowledgement packet SYN-ACK before allocating a TCP control block structure for the communication session. After the allocation of control block, there's no further way for the recipient to verify that subsequent packets are coming from the same host as contained in the SYN source address, hence creating a loophole that allows for uncontrolled flooding of SYN request from few initiators to a host in a proportion that inundates the host due to lack or memory and resources to accommodate such overwhelming demand. Using HIP however, such an attack is much more difficult to carry out, given that in HIP, the initiators identity is directly connected to the SYN packets.

Using existing IP based internet architecture, it is relatively easy for an attacker to spoof the source IP address of a packet once the host is placed under DoS attack, in addition to spoofing the IP address of the host, such attacker can further block the service of the legitimate host on the network. In [40], authors proposed a system where source addresses are validated using HIP deployed in a first-hop router. After normal HIP base exchange, the first-hop router is able to register the source IP address of the host to a database provided there is no previous record of such address in the database, that way an attempt to register an already existing IP address can be interpreted as a spoofing attempt, hence such packet is dropped. This basic idea of storing the mapping between identity and source IP address allows for tracking of duplicates and spoofed IP addresses [40].

4.4 HIP-enabled Virtual Private LAN Services (HIPLS)

HIP-enabled Virtual Private LAN Services (HIPLS) was the first p roposed secure VPLS architecture. HIP adds a cryptographic name space to name internet host and provides a HIP-enabled virtual private LAN service over an untrusted network [11, 30]. In spite of the perceived benefits of HIPLS in securing communication channels, it is still faced with a key limitation; lack of scalability in control, forwarding, and security planes, making HIPLS suitable only on unicast-only IPLS networks [30].

Typical VPNs are used to separate multiple LAN broadcast domains across shared network infrastructures. This section discusses how HIP can be used to create a VPLS overlay on standard IPv6 and IPv4 networks. In this case, Encapsulating Security Payload (ESP) tunnels are used to secure the tunnels between PEs, which are then managed using HIP signaling protocol. As shown in Figure 11, these HIP-enabled ESP tunnels constitute the pseudowires used to connect the VPLS forwarders. To accomplish this, each PE device is assigned a cryptographic HI and using certificates or parameters each HI is bound to other identifiers in the s ystem. HIP is then used to allow PE devices to authenticate one another and build secure tunnels over a public network. These PE devices may have to maintain Access Control Lists (ACLs) that they use to decide how the CEs talk to each other [11].

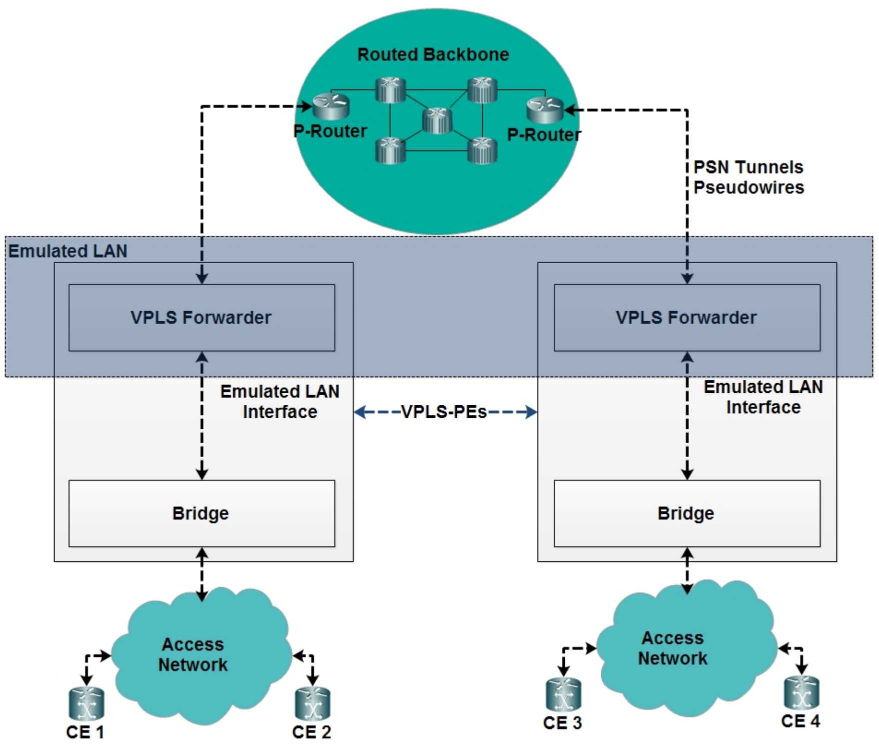


Figure 11. PE reference model.

In [10, 9, 15], authors proposed two customized versions of HIP, one is based on the use of session key and is called Session key based HIP VPLS architecture (S-HIPLS), the other is a HIP-based hierarchical VPLS architecture H-HIPLS. S-HIPLS mainly addresses the HIP limitations related to forwarding and security plane scalability, however is unable to address the control plane scalability issues. H-HIPLS on the other hand offers advanced security features on VPLS architecture such as authentication, confidentiality, integrity, availability, secure control protocol and robustness to the known attacks. The following subsections discuss more details of this proposed HIP modifications, their benefits as well as their limitations.

4.4.1 Session key based HIP VPLS (S-HIPLS)

The main idea of the proposed Session key based HIP VPLS architecture is to use a new session key mechanism together with a customized version of HIP. Two types of keys are used in this case; the Content Encryption Key (CEK) which is a symmetric key used to encrypt and decrypt all messages in a single VPN and the Key Encryption Key (KEK) which is uniquely assigned to each PE and is used to encrypt and decrypt corresponding CEKs. All keys are managed and distributed by the Key Distribution Center (KDC) which also serves as the Authentication Server (AS) and maintains the ACLs of the VPN [15, 9].

S-HIPLS architecture is categorized into four subsections, i.e. PE registration and deletion, data communication, control protocol, and key management. During PE registration, a potential PE sends a request to the KDC to seek access to the VPN. This implies that such PE requires an initial tunnel with the KDC after which a HIP base exchange (HIP BEX) process is initiated, the BEX in this scenario is similar to the traditional HIP BEX [13], except that at the fourth stage of the exchange, a certificate containing the name of the authorized VPN, QoS policies, e.t.c is sent in the R2 message if the user is a legitimate user. This authentication process is based on public key infrastructure, and when successfully completed, the PE is granted access to various network resources based on its category in the VPN ACL. On the revere end, it is also important to systematically remove inactive PEs from the VPN for security and resource optimization purposes. Inactivity notification may be passive or active depending on the initial configurations. For active notification, the PE informs inactivity by itself while for passive notification, inactivity is confirmed when PE fails to respond to periodic CSK from the KDC [9].

For data communication stage, the S-HIPLS architecture establishes HIP tunnels between PEs using IPsec. To avoid eavesdropping, all transmitted frames are encrypted using the corresponding CEK and wrapped within ESP payload before sending to a target PE.

At the control protocol, HIP tunnel is established between two intending users through a modified HIP BEX authentication process. For this scenario, there is no D-H key exchange as in the traditional HIP BEX, this is because data is already encrypted using CSK, hence allowing for less processing time than in traditional HIP BEX. Address learning is also implemented at the control protocol, given that VPLS operates on the data link layer, PEs must learn the MAC addresses of the remote CEs and the network addresses of the corresponding PEs. The third operation of the control protocol is key distribution, this is a crucial operation because erroneously issuing a key to an unauthorized PE could derail the overall security of the VPN, hence the need for an adequate key management procedure. This procedure includes the distributions of both KEK and CEK. During KEK distribution, PEs share a unique symmetric key with the KDC which is then used as the KEK for that PE, while for CEK distribution, the KDC sends CEKs to each PE [15].

Based on the simulation outcome in [15], S-HIPLS shows some remarkable benefits over traditional HIPLS architecture, such benefits include added scalability at the security plane. A key factor to this effect is reduction in the number of keys stored at the PE, unlike in HIPLS scheme where an increase in the number of keys stored is linear with respect to the number of PEs, S-HIPLS architecture ensures a constant number of keys at PEs without recourse to increase in number of PEs. At the KDC, the increment in number of keys stored in both HIPLS and S-HIPLS is linear with respect to the number of PEs, however, the S-HIPLS architecture tends to store slightly higher number of keys than HIPLS, this owes to the fact that the number of keys stored by S-HIPLS depends on both the number of PEs and VPNs whereas in HIPLS, this number is only based on the number of PEs. Summing up, the total number of keys stored in the network for the S-HIPLS scheme is substantially reduced compared with that in HIPLS.

On the forwarding plane, scalability is mainly dependent on the number of encryption per broadcast frame at the entry PE of the public network. Based on this fact, the HIPLS scheme shows a linear increase in the number of encryption per broadcast frame at a PE, while for S-HIPLS, this number remains constant with the value 1 which is a significant reduction in the number of encryptions per broadcast frame at the PE.

4.4.2 H-HIPLS

In [10], authors propose a HIP-based hierarchical VPLS architecture called Hierarchical HIPLS (H-HIPLS), the aim is to provide added security functionalities for traditional VPLS architecture. Such features as authentication, confidentiality, integrity, secure control protocol, and robustness to known attacks. In addition, this architecture promises to enhance scalability on both control and forwarding planes. On the security plane, this architecture offers the same level of scalability as in other flat VPLS architectures. These flat architectures are faced with a number of limitations including poor scalability potential due to the need to use an individual pseudowire for every pair of PE i.e. N-square scalability. They also suffer from high signaling overhead, limited number of possible hardware ingress replications as well as complex forwarding procedure.

In an attempt to resolve these issues, H-VPLS was proposed [7]. H-VPLS reduces the number of PEs connected in full mesh topology hence saving on the number of pseudowires required on the VPLS. It utilizes two types of PEs; user facing PE (u-PE) and network facing PE (n-PE). n-PE houses all the intelligence of the VPLS i.e. packet forwarding, address learning, and auto discovery functions while u-PE manages the forwarding of all packets to connected n-PE. Cost effectiveness is another advantage of the H-VPLS architecture when it comes to network expansion, this architecture allows the use of simple and cheap equipment to be used as u-PE. In addition, H-VPLS is a preferred option for use in heterogeneous service provider networks because the technology of its constituent layers is independent of each other [10].

On the down side, H-VPLS architectures suffer substantial amount of security threats such as DoS attack, TCP reset attack, and spoofing attacks. To avert these threats, encryption is used on both the control and data traffic of VPLS. Without such encryption, eavesdroppers could retrieve or alter the content of the transmitted data. Avoiding such threats also requires that the privacy of both the PE and CE is protected and that the PE is able to handle broadcast traffic.

H-HIPLS is a H-VPLS architecture that harnesses the security features of HIP to provide a secure connection across a VPN. H-HIPLS is basically a modification to the HIP-based session key mechanism which is proposed in the S-HIPLS architecture discussed in the previous Section. The goal is to facilitate hierarchical architecture, support encrypted label based forwarding and dynamic address learning mechanisms.

As proposed in [10], H-HIPLS operates based on five key mechanisms; control protocol, key management, PE management, packet forwarding, and address learning. On the control protocol, tunnels are securely established and frames encrypted using the session key of the VPN. Key management is done for both CEK and KEK, the KDC distributes CEKs and maintains ACLs for the entire VPN. PE management mechanism ensures that new PEs are duly authenticated, in this case using a HIP-based authentication mechanism. First the PE establishes a tunnel with the KDC using HIP base exchange which then establishes security associations for the tunnels and at the same time performs mutual authentication for the end nodes using their cryptographic identities. The proposed H-HIPLS advances this authentication process by incorporating the use of ACLs in authorizing new users. Hence after the identity verification at the third stage of the HIP BEX, the ACL is checked to verify the category of the new user before granting access to the VPN. In addition, the tunnel establishment for this architecture does not incorporate the D-H key exchange process as with traditional HIP BEX, hence the R1 and I2 messages do not contain any fields for D-H message instead authentication tokens provided by the KDC are used.

After PE authentication, packets are forwarded using an encrypted label mechanism, these label is inserted into the ESP header at the third step of the base exchange before forwarding the frame to the n-PE. On receiving the data frame, the u-PE first removes the header and decrypts the ESP payload using the corresponding VPN session key. The n-PE then decrypts the label and checks the nest hop for the data frame from the MAC-PE mapping table.

Address learning is normally important in VPLS given that it is a layer 2 VPN solution and relies on the use of MAC addresses for frame forwarding. Since the underlay provider network is a layer 3 network, H-HIPLS architecture proposes the use of dynamic MAC-PE mapping to map the MAC address of the CE to the corresponding network address of the PE. Using both u-PE advertisement and dynamic address request, each n-PE is able to update its MAC-PE mapping. Details of these two address learning instances are contained in [10].

Based on the simulated outcome in [10], the proposed H-HIPLS architecture has strong potential to enhance scalability and promote security on VPLS networks. For instance, the architecture significantly reduces the number of tunnels required to achieve control plane scalability. For security plane scalability, the key storage complexity of H-HIPLS at the PE is substantially optimized. At the PE, the architecture maintains a constant number of keys, similar to the S-HIPLS architecture and unlike the traditional HIPLS which maintains a linear increment in the number of keys stored and the number of PEs. At the KDC, the total number of keys stored for H-HIPLS architecture is on a linear increase, similar to HIPLS and S-HIPLS architectures, however the H-HIPLS and S-HIPLS have a slightly higher number of keys stored compared to HIPLS. On the overall, authors [10] argue that the number of keys stored on the network using the proposed H-HIPLS architecture is on a linear increase while for traditional HIPLS ar-chitecture, the number of keys stored is on an exponential increase at a relatively the same level of performance.

On security, the proposed H-HIPLS architecture also shows substantial resistance to attacks on the control plane. Going by the simulation outcome in [10], H-HIPLS shows no packet drop during a TCP SYN DoS attack session. For TCP reset attack, the outcome also shows that H-HIPLS is absolutely resistant to TCP reset attacks.

4.4.3 Comparing Different VPLS Architectures

Table 4.1 below presents a tabulated comparison of the different VPLS architectures [10, 9, 30, 15]. These architectures are compared based on various performance matrices to enable stakeholders make more informed decisions on what architecture most fits their intended VPLS solution.

Table 4.1. Comparing different VPLS architectures

	LDP based H-VPLS	**BGP based H-VPLS**	**HIPLS**	**S-HIPLS**	**H-HIPLS**
Architecture	Hierarchical	Hierarchical	Flat	Flat	Hierarchical
Scalability of forwarding plane	High	High	Low	High	High
Scalability of security plane	-	-	Low	High	High
Scalability of control plane	High	High	Low	Low	High
Protected control protocol	No	No	Yes	Yes	Yes
Data traffic encryption	No	No	Yes	Yes	Yes
IP attack protection	No	No	Yes	Yes	Yes
Efficiency of the broadcast mechanism	High	High	Low	High	High

5. Testbed Implementation and Result Analysis

Two separate testbeds were implemented, one for the Linux-based open-source HIP implementations and another for the two sets of security appliances from different vendors, one secured using the HIP protocol and the other non-secure system was equally evaluated. In all cases, these solutions are compared against a non-secure connection for network performance in terms of overhead bandwidth utilization, latency, and jitters.

5.1 Linux-based Open Source HIP Implementations

The two open-source Linux-based HIP implementations under study (HIPL and OpenHIP) were tested across three different network scenarios. First on a wired LAN, then on Wireless LAN (WLAN), and finally on a public network (Campus network connected to the Internet). The experiment testbed consists of two User Equipments (UE), i.e. Two laptops with Ubuntu 12.04 LTS (Long Term Support) operating system. We installed HIPL and later OpenHIP on both machines; one plays the initiators role and the other, the responder role. For OpenHIP implementation, we installed the OpenHIP binary (version 0.9.1) from the OpenHIP online repository [6], while for HIPL implementation, we installed HIP daemon (*hipl-daemon*) version 1.0.8 from the InfraHIP project repository [5] along with the DNS proxy (*hipl-dnsproxy*), and firewall (*hipl-firewall*) on both UEs. All installed packages are the most up-to-date versions and contain the latest codes developed.

The first experiment set-up consists of two HIP-enable UEs linked on a Local Area Network (LAN). The LAN network is established by connecting each UE to a Gigabit Ethernet bridge – Cisco Linksys WUMC710, which is further linked to a unified service router – D-link DSR-250N as shown in Figure 12.

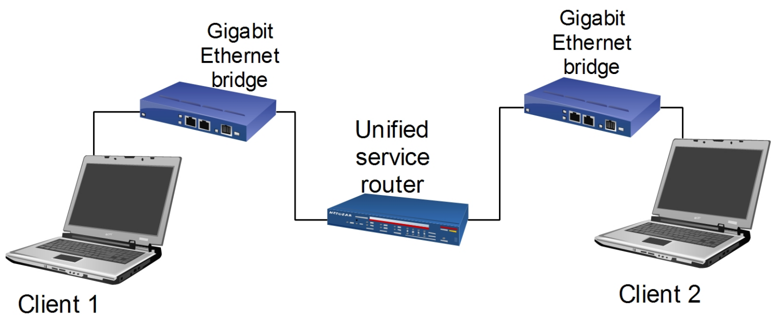


Figure 12. LAN set-up.

In the second test scenario, we set up a WLAN using the same UEs as in the first set-up, here each UE is connected to a D-link DSR-250N unified service router. These routers are further linked wirelessly through a Wireless Distribution System (WDS) service. This set-up represents a minimal case of an ideal WLAN in a typical enterprise network environment as shown in Figure 13.

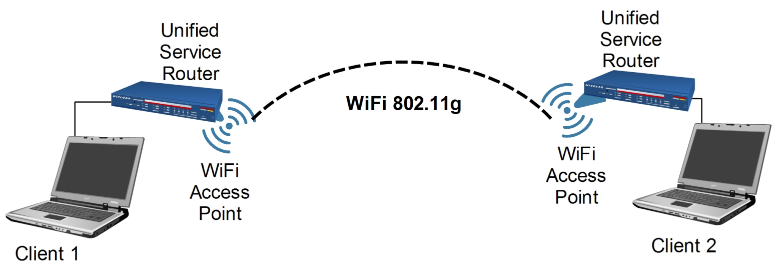


Figure 13. WLAN set-up.

The third scenario is set-up on the campus Wide Area Network (WAN), the two UEs are connected to two Cisco Linksys WUMC710 Gigabit Ethernet bridges [40] which are further linked to the campus network on Ethernet, supporting both IPv4 and IPv6 connectivity and a maximum speed of 1GBps. This test scenario which is illustrated in Figure 14 represents a typical case of running HIP over a public shared and insecure network.

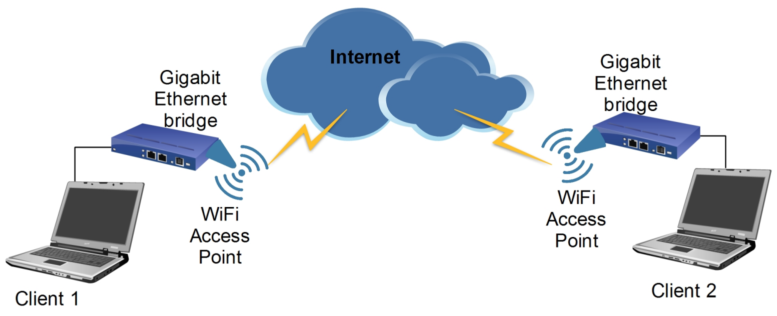


Figure 14. WAN (campus network) set-up.

For each scenario, we first test the performance of the network without any HIP implementations, then establish HIP connection between the two UEs using HIPL and later OpenHIP. For each set-up, we analyze the performance penalty of security due to HIP implementations, performance is measured in terms of throughput, jitter, and latency. We measure the performance against the non-secure scenario. The throughput and latency are measured using the IPERF networking tool [41] while jitter is measured using the basic ping tool.

5.1.1 OpenHIP Implementation

The OpenHIP project aims at developing the reference implementation of HIP for various platforms. OpenHIP is a free open-source HIP implementation developed within the IETF and the Internet Research Task Force (IRTF). It was licensed by MIT/Expat to provide researchers an experimental platform to study HIP and other related protocols [6].

In this setup, OpenHIP uses HIP to provide the rapid exchange of host identities between PC-1 and PC-2 and at the same time establishes a pair IPsec Security Associations (SA) to be used with IPsec Encapsulated Security Payload (ESP), hence OpenHIP is able to provide the various security benefits accruable to HIP such as resistance to DoS and man-in-the-middle attacks and protecting the upper layer protocols (TCP and UDP) from such attacks.

With OpenHIP installed on both UEs, a **hitgen** command is used to automatically generate the Host Identity Tags (HIT) for both UE-1 and UE-2. It is important to disable the Domain Name System (DNS) when setting up the hip configuration file. This is necessary because it enables HIP to connect the hosts to the network using the HIT tags and not allowing the DNS to view them as domain names and attempt to translate them into corresponding IP addresses as per directory services.

Next is to configure **"my_host_identity"** and the **"known_host_identities"** files in both UEs. As emphasized in [4], actual Host Identities are never used directly in any Internet protocols, instead a Host Identity Tags (HITs) or a Local Scope Identifiers (LSIs) are used. The corresponding Host Identifier public key may be stored in vari-ous DNS or Lightweight Directory Access Protocol (LDAP) directories, and they are passed in the HIP base exchange.

For our case, an LSI is used to represent the Host Identity, the parameter in the LSI is the actual IP address of the host. The known_host_identies file contains a list of the names, IP addresses, Host Identity Tags (HITs) and LSIs of all hosts that are associated with a given host.

With the above configurations, the HIP daemon is run in verbose mode using the **sudo hip -v** command. On successful execution, a HIP interface named **hip0** is added to the list of network interfaces displayed by the **ifconfig** command, hence the HIP tunnel is active and ready for use. Another important parameter to check for at this stage is the value of the Maximum Transmission Unit (MTU), this is an octet representation of the largest size of packet that can be sent over a packet based network like the internet. As noted in [2], the size of MTU could very much depend on the network path, hence different network paths tend to have different MTU sizes. It is the role of transport protocols such as TCP to discover the ideal size of MTU on the initial stage of a connection. The default MTU size for OpenHIP is 1400, however this size is inefficient for the transmission and from experiment OpenHIP is unable to automatically decide the adequate MTU size, hence the need to manually reset the MTU to a larger size. For the purpose of this experiment, the MTU size is set to 1500. With the hip daemon running, the tunnel performance is evaluated using ping and iperf tools.

5.1.2 HIPL Implementation

HIPL is an open-source software developed by InfraHIP project team, the focus is to develop the missing infrastructure pieces such as DNS, NAT, and firewall support in order to enable a widespread deployment of HIP [5]. As a HIP based project, HIPL aims at securely supporting mobility and multi-homing on the TCP/IP stack, hence providing advanced security and privacy as well as other advanced network concepts such as mobile ad hoc and moving networks.

HIPL offers infrastructural support for the application related aspects of HIP. These include APIs, OS security, rendezvous service, multiple end-points within a single host, process migrations, and other issues related to enterprise-level solutions.

In this set-up, a *hipl-daemon* package is installed and configured on both the server and client PCs. Both PCs are running 32-bit Ubuntu 12.04 LTS (Long Term Support) operating system. Other optional but useful packages (*hipl-dnsproxy* and *hipl-firewall*) were installed alongside hipl. After installation, a ***hipsd #*** command is used to run the hipl daemon. When hipl daemon is active, two additional interfaces labeled ***dummy0*** and ***dummy0:2*** are displayed among the network interfaces for both UEs.

For this experiment, we use IPv6 to establish HIP tunnel between the two hosts. It is important here to mention that both IPv4 and IPv6 are equally supported and can inter operate. The IPv6 address to be used for the HIP tunnel could be added to the LAN interface or to the WLAN interface using a similar approach. In this case, the WLAN interface was used, and the host file of the UE acting as the connection initiator was configured with the HIT and IP address of the pair UE.

Using **nc6** as the test application, a HIP connection is established between the two hosts. The HIP daemon uses automatically generated host identities from ***/etc/hip*** folder. A confirmatory test of the establishment of HIP tunnel is to type in some random texts on either UE and hit the enter key to display them on the pair UE. With this confirmation, the performance of the tunnel is accessed using the ping and iperf tools as previously done with the OpenHIP application.

Table 5.1 contains the simulation settings for IPERF testing tool, these settings are also maintained for the enterprise solution testbed. Both long and short sessions were tested, for Linux-based Open Source HIP Implementations, a short session runs for 10 seconds while the long session runs for 200 seconds, each experiment is repeated 50 times in both directions.

Table 5.1. The Simulation Settings for IPERF

Parameter	**Value**	**Value**
Protocol	TCP	UDP
Port	5001	5001
Buffer size	1470KBytes	1470KBytes
Packet size	1500Bytes	1500Bytes
TCP window size	-	85.3KBytes
Report interval	1s	1s

5.2 Analysis of Results for Linux-based Open Source HIP Implementations

The following subsections provide numerical analyses of the experiment outcome for the two HIP implementations and compares them to the non-secure scenario. The analyses show the performance penalty in terms of latency, throughput and jitter for the three tested network scenarios.

5.2.1 Performance penalty of security on latency

For all three scenarios, we measured the performance penalty of the two open-source HIP implementations on latency. Latency was measured using the basic ping test and recording the Round Trip Time (RTT) over 100 test packets transmitted in both directions. Each approach was compared against the non-secure connection.

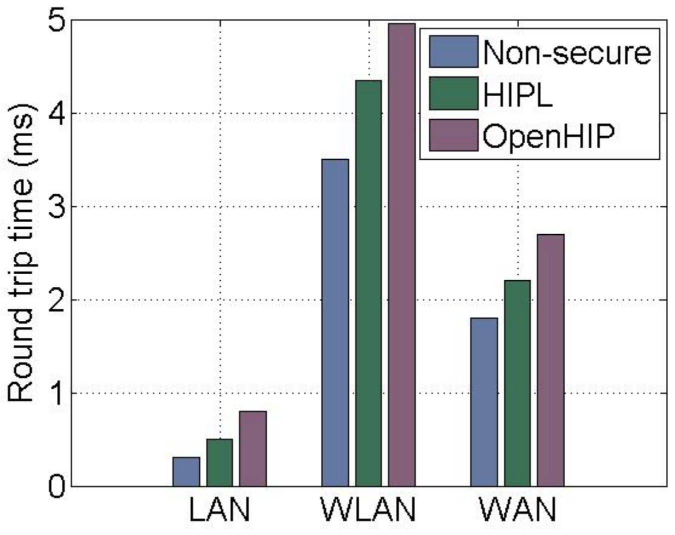


Figure 15. Performance penalty on latency.

Based on the experiment outcome shown in Figure 15, for the LAN scenario, OpenHIP implementation causes about 3 times more latency compared with the non-secure and 2 times more than HIPL implementation. The HIPL implementation increases latency by approximately 50% compared with the non-secure architecture. For the WLAN scenario, OpenHIP implementation increases latency by approximately 23% while OpenHIP increases by about 35% compared with the non-secure scenario. In the WAN set-up, HIPL increases latency approximately by 28% while OpenHIP increases by about 13% compared with the non-secure scenario. This increase in latency for both HIP implementations is due to the additional reverse lookup involved in HIP base exchange in addition to the Address Resolution Protocol (ARP) query performed upon the first connection.

5.2.2 Performance Penalty of Security on Throughput

We measured the performance penalty of the two HIP implementations on both TCP and UDP throughputs and compared with the non-secure connection. Each of the three scenarios was measured and evaluated independently. Both long and short sessions were measured as described in the experiment set-up and in all cases results are compared with the non-secure scenario.

For LAN Scenario

As shown in Figures 16 and 17, HIPL implementation performs nearly as good as the non-secure connection in terms of throughput, with a penalty of about 3% for both long and short TCP and UDP sessions, reaching approximately 820 Mbps for TCP and about 850 Mbps for UDP. OpenHIP however has its peak throughput at approximately 100 Mbps for both long and short TCP sessions, this is because OpenHIp is implemented at user-level in all three scenarios, and by design limitation, its throughput peaks at 100 Mbps [42].

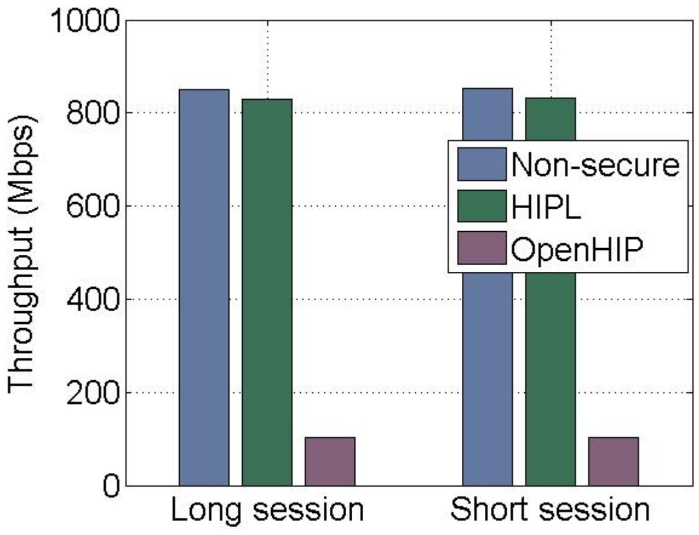


Figure 16. Average TCP throughput for LAN set-up.

On WLAN

Figures 18 and 19 show the results for the WLAN configuration, the performance penalty of security on throughput for HIPL implementation is approximately 32% for the long session and about 30% for the short session for both TCP and UDP. For this set-up OpenHIP had an average TCP throughput of about 13 Mbps and for both long and short sessions while the UDP throughput reaches an average of 14 Mbps for both long and short sessions. It is important to mention that in addition to the encryption implemented by HIP, the WLAN Access Points also introduces an additional layer of encryption for data using the WiFi Protected Access (WPA) protocol, hence causing more reduction in throughput.

On WAN (campus network)

The WAN set-up showed remarkable variations in the performance for both HIP implementations, while this could have been due to some network variables such as network density, averaging over several tests showed that both HIP implementations almost had

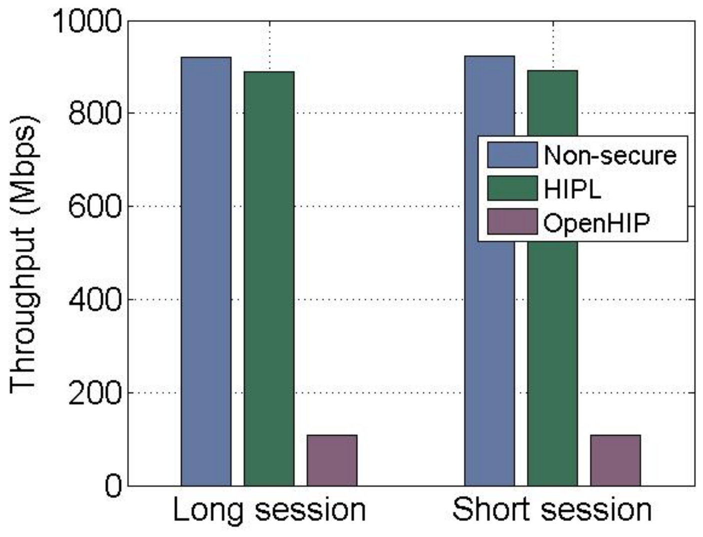


Figure 17. Average UDP throughput for LAN set-up.

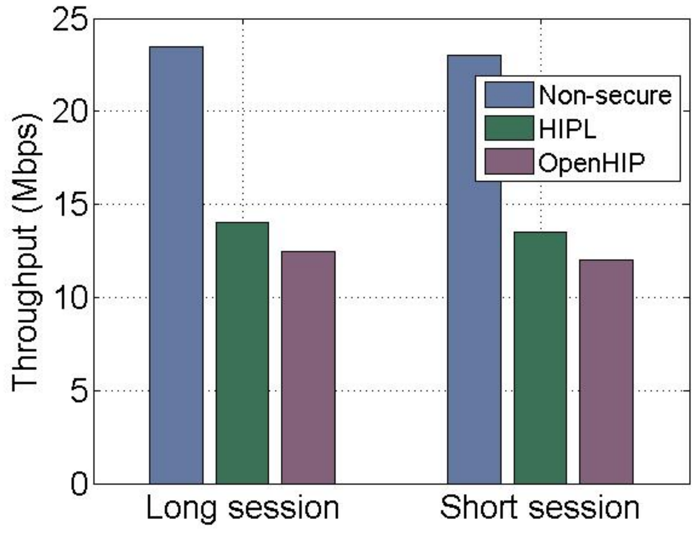


Figure 18. Average TCP throughput for WLAN set-up.

equal performance for both TCP and UDP throughput, however this is at a penalty of about 22% compared with the none-secure scenarios as shown in Figures 20 and 21.

5.2.3 Performance Penalty of Security on Jitter

In all three set-ups, we measure the performance penalty of security on UDP jitter using the IPERF tool. Figures 22, 23 and 24 illustrate the experiment results.

According to the experiment results, the average jitter on OpenHIP implementation is about 20% higher than HIPL and 75% higher than non-secure connection for the

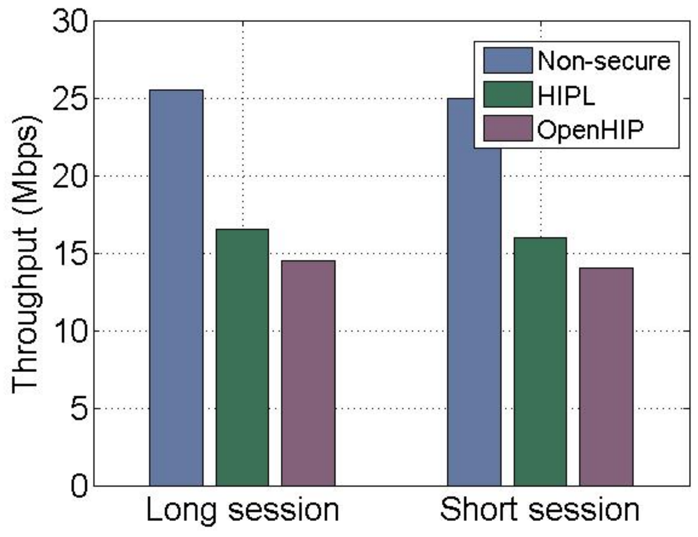


Figure 19. Average UDP throughput for WLAN set-up.

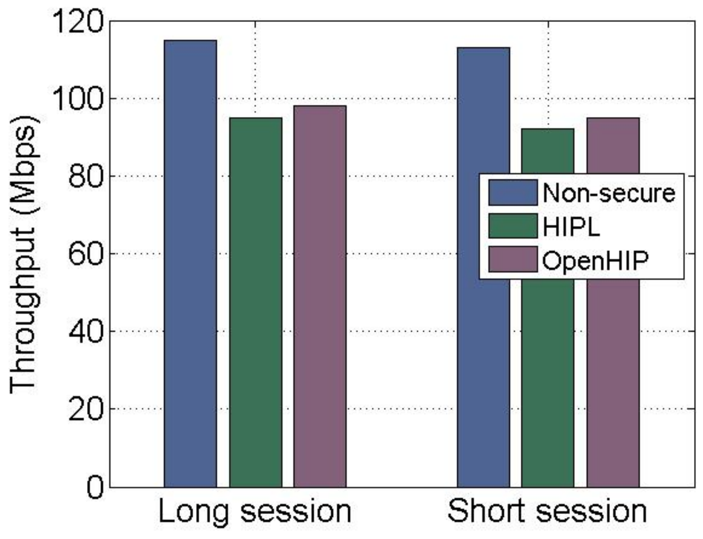


Figure 20. Average TCP throughput for WAN (campus network) set-up.

LAN scenario. On WLAN, OpenHIP experiences about 52% more jitter than none-secure connection and about 44% more than HIPL for the long sessions, while for the short sessions, OpenHIP experiences about 78% more jitter than the non-secure connection and about 52% more than HIPL. On WAN, OpenHIP experiences about 48% more jitter than the non-secure and 37% more than HIPL implementation for both long and short UDP sessions.

It is important to mention that the increased jitter for both HIP implementations does

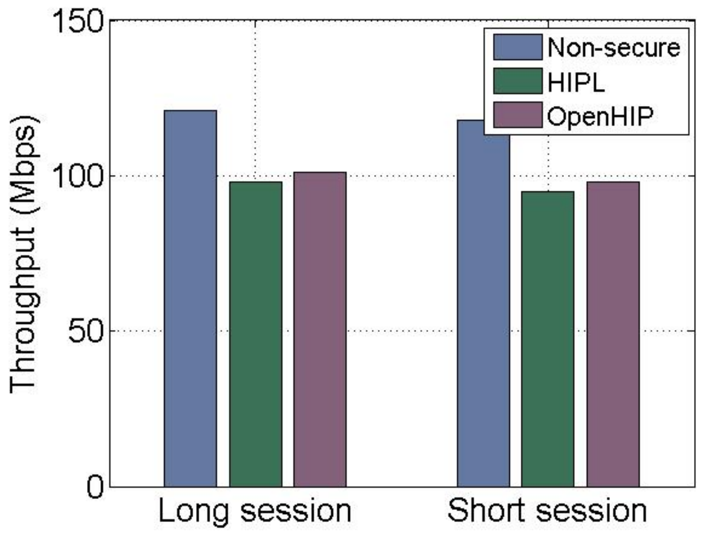


Figure 21. Average UDP throughput for WAN (campus network) set-up.

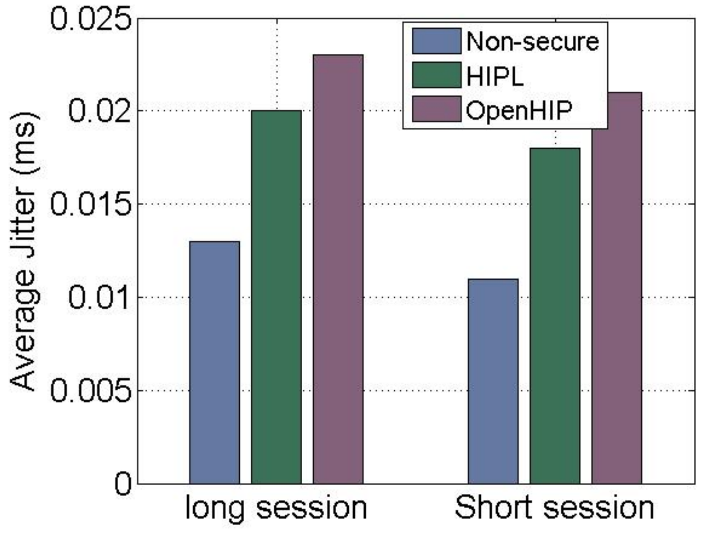


Figure 22. Average Jitter for LAN set-up.

not result from the security feature of HIP, rather it is a combined result of queuing, contention, timing drift, and serialization effects on the path through the network. For HIPL implementation, the additional *hipl-firewall* utility that accompanies the *hipl-daemon* is also a contributing factor for the increased jitter.

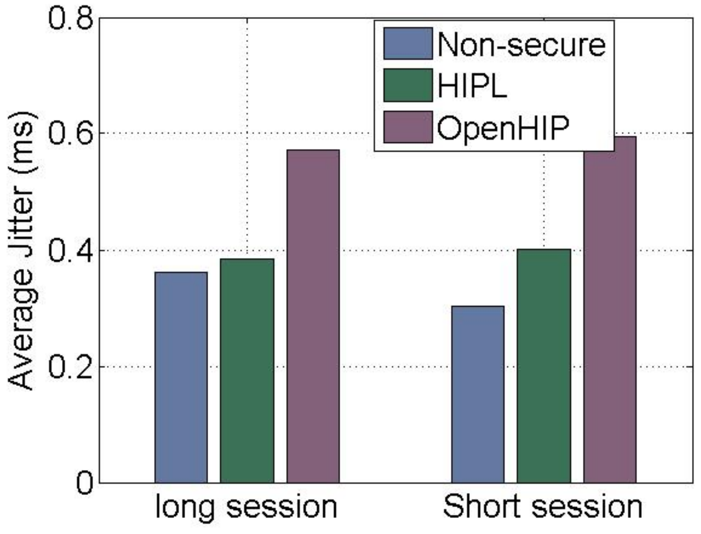


Figure 23. Average Jitter for WLAN set-up.

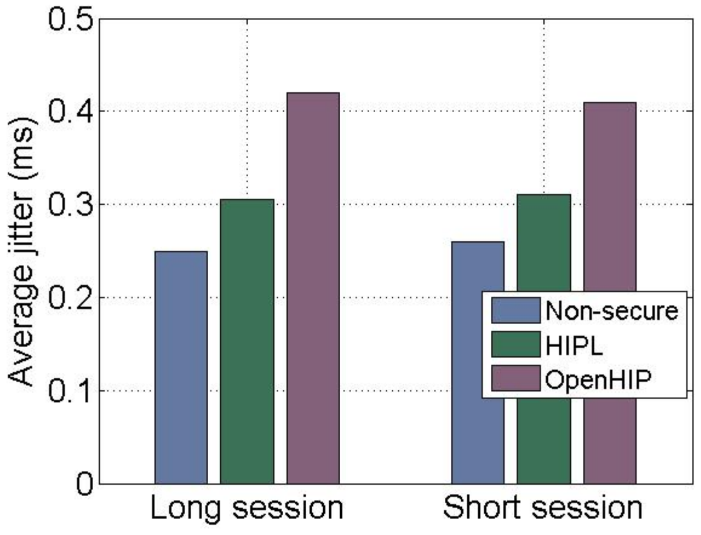


Figure 24. Average Jitter for WAN (campus network) set-up.

5.3 Enterprise-based Network Security Solution

Tempered Networks^{TM} provides a centrally managed security solution that leverages on existing network infrastructure and facilitates the provisioning and management of secure, private overlay networks over any existing network. The solution consists of two tightly integrated components namely, the HIPswitch Conductor^{TM} with SimpleConnect^{TM} User Interface and the HIPswitch^{TM} security appliances. These two are interconnected with other network equipment as shown in Figure 25 [43].

The technical specifications of the actual components used in the experiment is contained in Appendix 1.

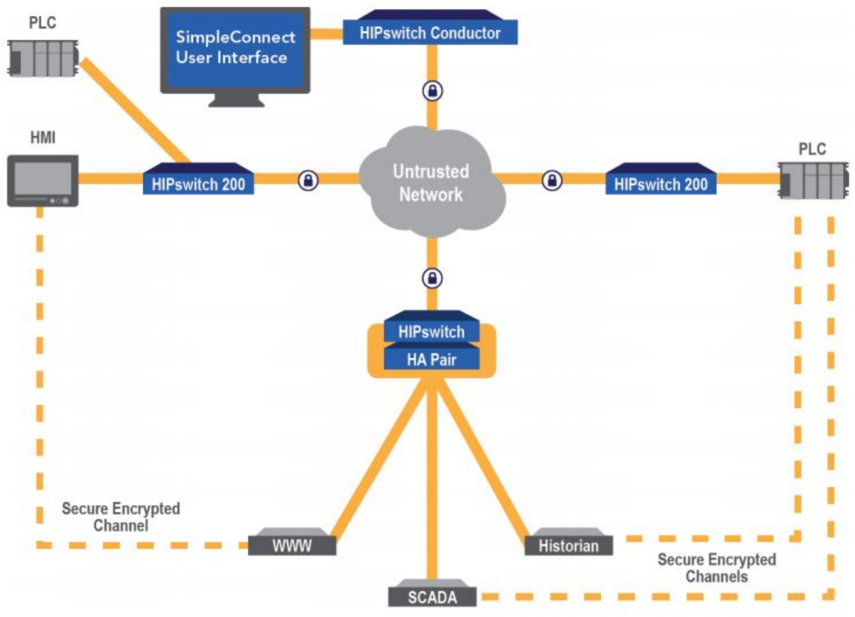
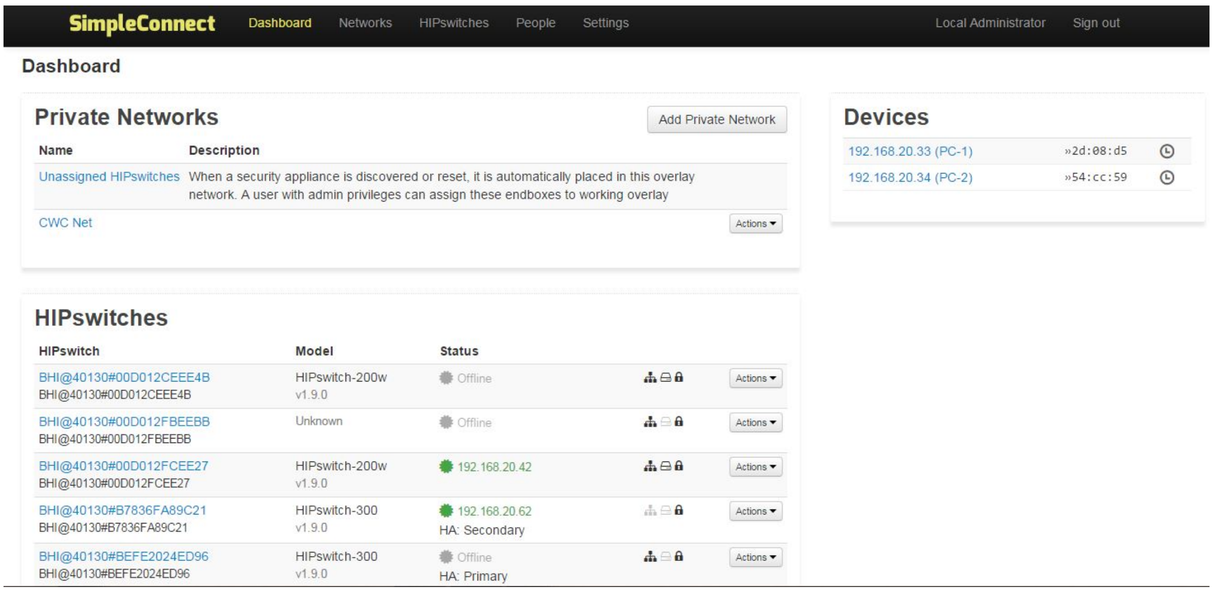


Figure 25. Tempered Networks Solution connectivity, taken from [43].

The HIPswitch conductor is the control center of the Tempered Networks solution. Through the SimpleConnect user interface, the HIPswitch conductor is able to coordinate the configuration, security policies, and trust relationships of all HIPswitches connected to the network. The conductor creates a secure private overlay network with HIPswitches and delegates management of each private network to authorized users. Other functions include defining the protected devices behind each HIPswitch, configuring communication security policies, audit and monitoring of the private networks, HIPswitches and devices [43].

In addition to security, the Tempered Networks solution also provides High Availability (HA) functionality on HIPswitch-300. To implement this feature, a pair of HIPswitch-300 are connected in a high availability mode through a dedicated Ethernet link and configured on the SimpleConnect UI to offer hardware redundancy for the secure connections. Using the HA configuration, the pair ISAs are assigned as Primary and Secondary as shown in Figure 26. Both SAs maintain a heartbeat through the Ethernet link. A failure in power or connection on the Primary HA automatically transfers the private network communications from the Primary SA to the Secondary SA [43].

For this experiment, we model a VPLS using the above described Tempered Networks solutions on the existing WiFi network. Layer 2 legacy devices are interconnected via the VPLS network. Here, the existing Wi-Fi network is provisioning as the provider network of VPLS.



Dashboard

Private Networks

Add Private Network

Name	Description	
Unassigned HIPswitches	When a security appliance is discovered or reset, it is automatically placed in this overlay network. A user with admin privileges can assign these endboxes to working overlay	
CWC Net		Actions ▾

Devices

192.168.20.33 (PC-1)	»2d:08:d5	
192.168.20.34 (PC-2)	»54:cc:59	

HIPswitches

HIPswitch	Model	Status	
BHI@40130#00D012CEEE4B BHI@40130#00D012CEEE4B	HIPswitch-200w v1.9.0	Offline	Actions ▾
BHI@40130#00D012FBEEBB BHI@40130#00D012FBEEBB	Unknown	Offline	Actions ▾
BHI@40130#00D012FCEE27 BHI@40130#00D012FCEE27	HIPswitch-200w v1.9.0	192.168.20.42	Actions ▾
BHI@40130#B7836FA89C21 BHI@40130#B7836FA89C21	HIPswitch-300 v1.9.0	192.168.20.62 HA: Secondary	Actions ▾
BHI@40130#BEFE2024ED96 BHI@40130#BEFE2024ED96	HIPswitch-300 v1.9.0	Offline HA: Primary	Actions ▾

Figure 26. SimpleConnect Dashboard.

The experiment testbed consists of two User Equipments (UEs) which are placed in two different locations. Two laptops with Ubuntu 12.04 LTS (Long Term Support) operating system are used as two UEs. We use a wireless network as the provider network. It is a WiFi 802.11g standard wireless network which supports a maximum speed of 54 Mpbs. Figure 27 illustrates the experiment testbed.

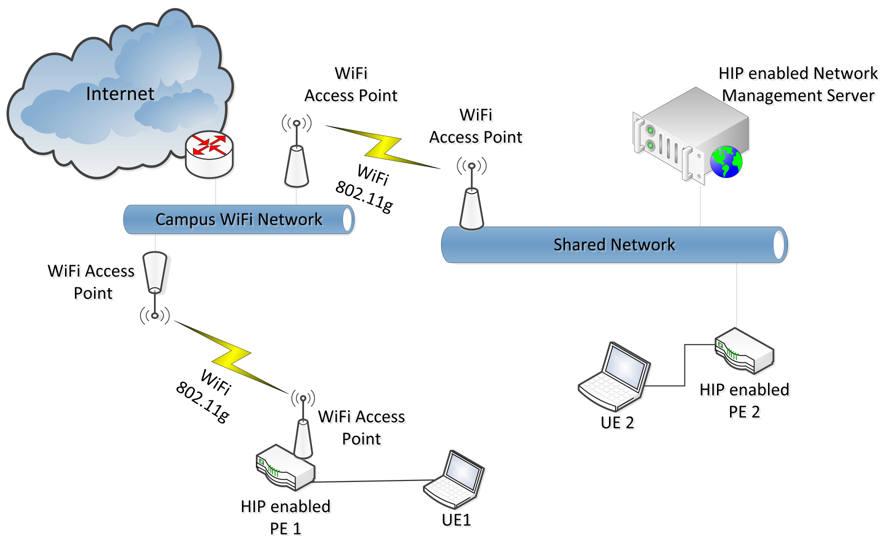


Figure 27. Experiment testbed for SCADA network.

Two HIP enabled PEs are used at the edge of the Wi-Fi network. PE-1 has the wireless connectivity to the campus network. A wired shared network is established by using a WiFi router with four Ethernet ports. PE-2 is wired connected to the WiFi router via an Ethernet port. Two HIPswitches (HIPswitch-300) shown in Figure 28 are used as PE devices. These HIPswitches are developed by Tempered Networks (formerly Asguard) [43]. Each HIP enabled network appliance called PE uses a unique public/private RSA 2048 bit key pair as its HI.

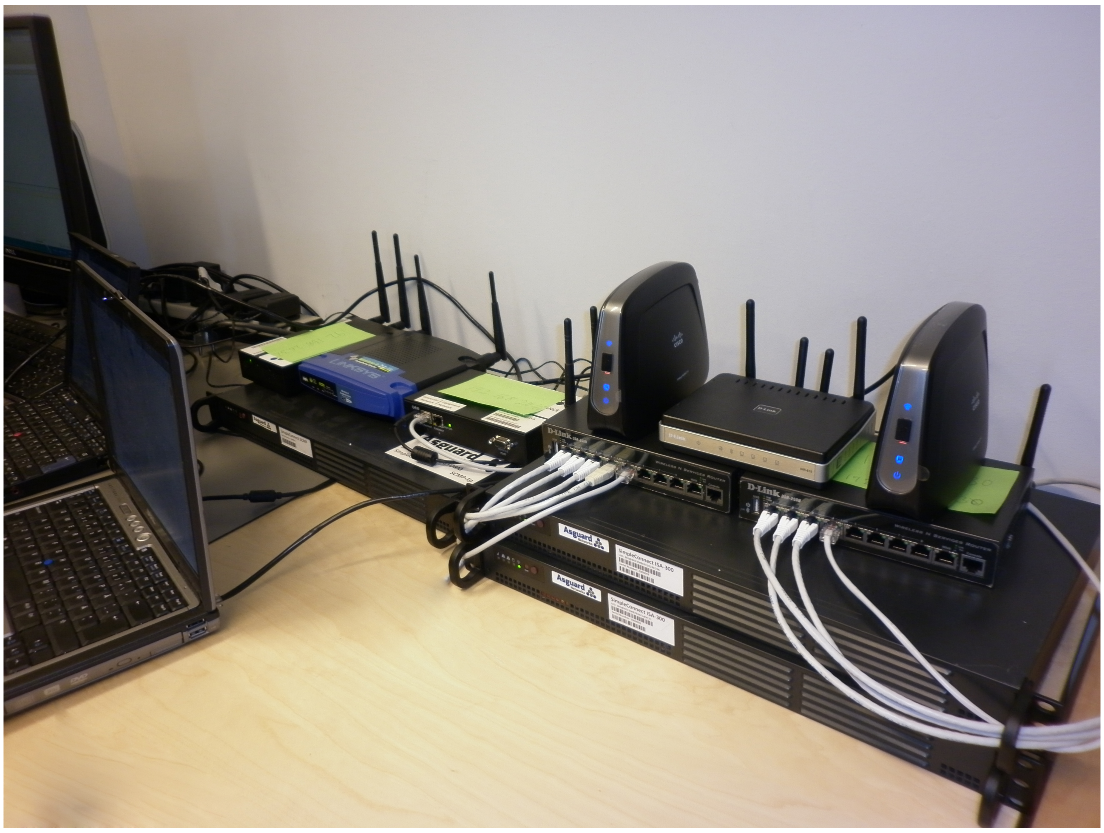


Figure 28. Experiment testbed for Tempered Network Solution.

Furthermore, a HIP enabled network management server is attached to the shared wired network. This server is responsible for the VPLS provisioning functions. It is used to assign network addresses to PEs, key management, VPN tunnel management and distribution of cryptographic credentials. Also, it supports the auto discovery functions of PEs. All customer data traffic is encrypted before transporting them over the provider network. HIP uses Advanced Encryption Standard (AES) - Cipher-block chaining (CBC) encryption. The key size of AES is 128 bits in this experiment.

To the best of our knowledge, HIP based VPLS architectures are the only secure VPLS architectures proposed yet. None of the other VPLS architectures have any dedicated security features. Thus, it is not possible to study the security penalty of other architectures. We analyze only the HIPLS architecture in our experiments. Both S-HIPLS and H-HIPLS architectures are proposed only to tackle the scalability issues by proposing a key distribution and tunnel establishment mechanisms. These changes impact only the operation of the control plane. In the steady state operation (once the VPLS network is established), both S-HIPLS and H-HIPLS architectures have exactly the same behavior as the original HIPLS. All three architectures use HIP tunnels to secure the data plane traffic. Thus, we present the experiment results based on original HIPLS architecture only. However, other secure VPLS architectures such as S-HIPLS and H-HIPLS also have the same behavior in this experiment setup.

In this experiment, we analyze the performance penalty of security due to secure VPLS architectures on throughput, jitter and latency. We measure the performance against no VPLS scenario and with non secure VPLS (LDP VPLS) scenario. The throughput and latency are measured by using the IPERF networking tool [41]. Table 5.1 contains the simulation settings for IPERF testing tool.

5.4 Analysis of Results for Enterprise-based Network Security Solution

In this section, the enterprise-based network security solution is evaluated for overhead bandwidth utilization, latency, and jitter. Two sets of enterprise equipment (Tempered Networks and Byres Security) are compared with a non-secure connection based on the listed parameters. For the TEB experiment, the same experiment set-up as in Tempered Network solution was used. For each experiment, we replace PEs in the testbed with TEBs and HIPswitches. We use two types of HIPswitches namely HIPswitch-200 and HIPswitch-300. A UDP session with the bandwidth of 54Mbps is used in each case.

5.4.1 Performance Penalty of Security on Latency

In the first set of experiments, we measure the performance penalty of security on latency due to the secure VPLS architecture. The Round Trip Time (RTT) of a packet is measured by using the basic ping test. Each experiment is run for 100 packets in both directions.

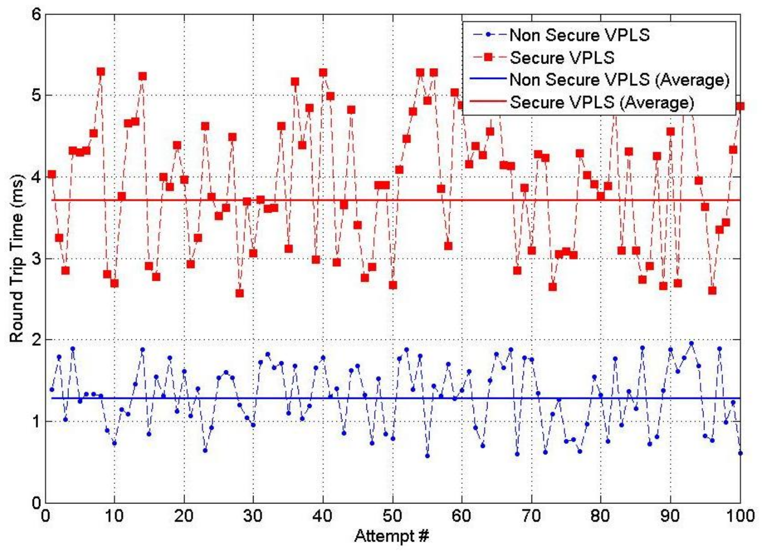


Figure 29. The performance penalty of security on latency.

Figure 29 contains the actual and average latency for secure VPLS solution compared with the non-secure connection. Latency is measured based on average RTT values. The experiment results show that the secure VPLS architecture for the enterprise equipment increases the latency approximately by 87% than the non VPLS scenario and by 68% than the non secure VPLS scenario. Similarly, non secure VPLS increases the latency approximately by 11% than the non VPLS scenario. The main reason to increase the latency is the delay in both the packet encryption and tunnel encapsulation leading to extended packet data path.

5.4.2 Performance Penalty of Security on Throughput

In the second set of experiments, we measured the performance penalty of security on throughput. Both TCP and UDP sessions were considered.

1. ***Performance Penalty of Security on TCP Throughput***

   First, we considered TCP sessions. The throughput was measured by using the IPERF networking tool. We measured the throughput of both short and long TCP sessions. A short TCP session runs for 10 seconds and a long TCP session runs for 500 seconds. Each experiment was repeated 50 times in both directions.

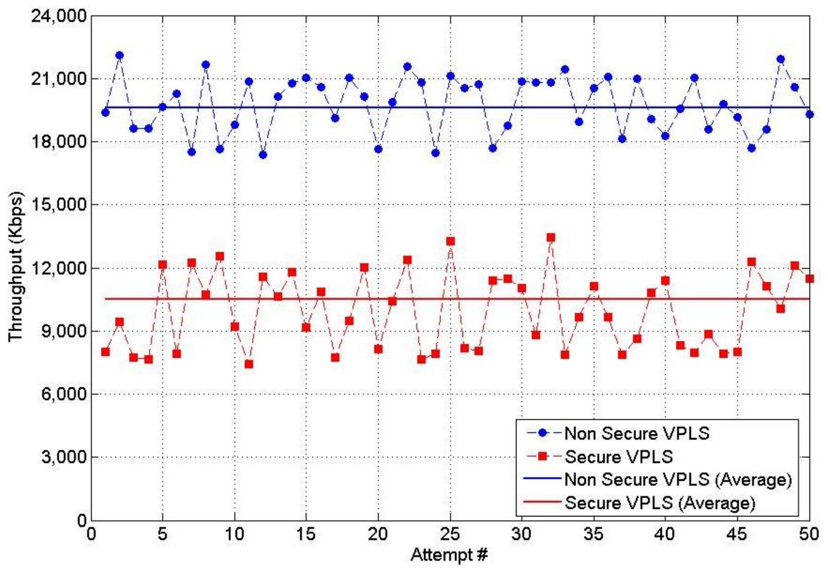
   According to experiment results shown in Figure 30, the performance penalty of security on throughput is about 19% for the short TCP session and 21% for the long TCP session compared with the non VPLS scenario. Moreover, the performance penalty of security on throughput is about 18% for a short TCP session and 20% for a long TCP session compared with the non secure VPLS architecture. Thus, we conclude that the performance penalty of security on throughput is independent of the duration of a TCP session.

   On the other hand, non secure VPLS reduces the throughput only by 1% than the non VPLS scenario in both short and long TCP sessions. Therefore, the impact of VPLS tunnel encapsulation on TCP throughput is very low. The additional layer of encryption is the main reasons for the reduced average throughput of the secure VPLS architecture.

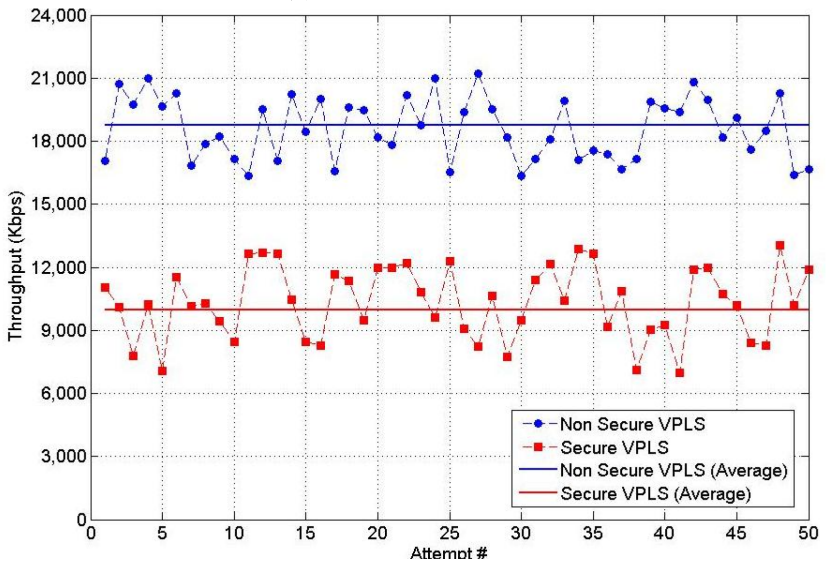
2. ***Performance Penalty of Security on UDP Throughput***

   In the next experiment, we consider UDP sessions. The throughput is measured by using the IPERF networking tool. The UDP bandwidth of IPERF is set to 54 Mbps which is equal to the bandwidth of the network. In this experiment also, we measure the throughput for both short and long UDP sessions. A short UDP session runs for 10 seconds and long UDP session runs for 500 seconds. Each experiment repeated 50 times in both directions.

   According to the experiment results shown in Figure 31, the performance penalty of security on throughput is about 20% for a short UDP session and 21% for a long UDP session compared with the non-VPLS scenario. Moreover, the performance penalty of security on throughput is about 19% for both for short and long UDP sessions compared with the non secure VPLS architecture. Thus, we conclude that for this scenario, the performance penalty of security on throughput is independent of the duration of a UDP session.



(a) Short TCP session.

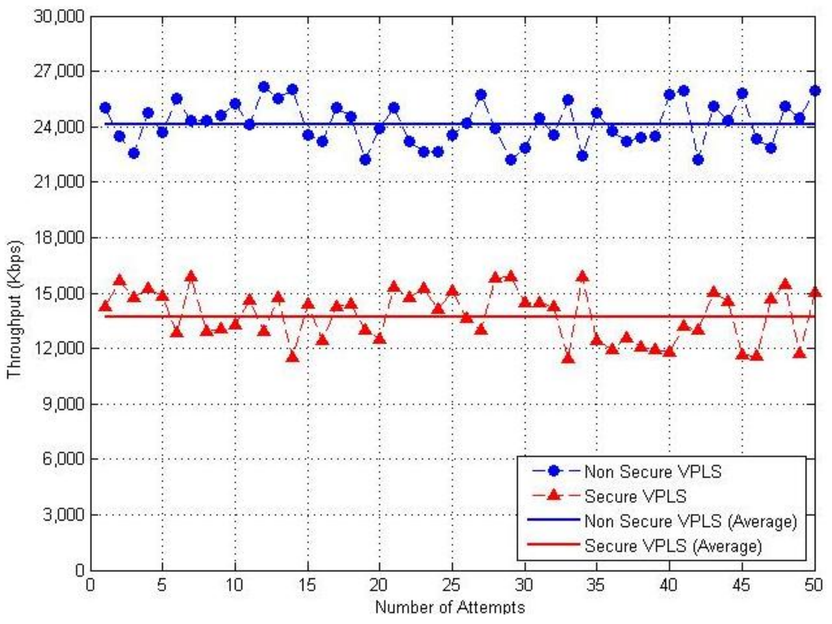


(b) Long TCP session.

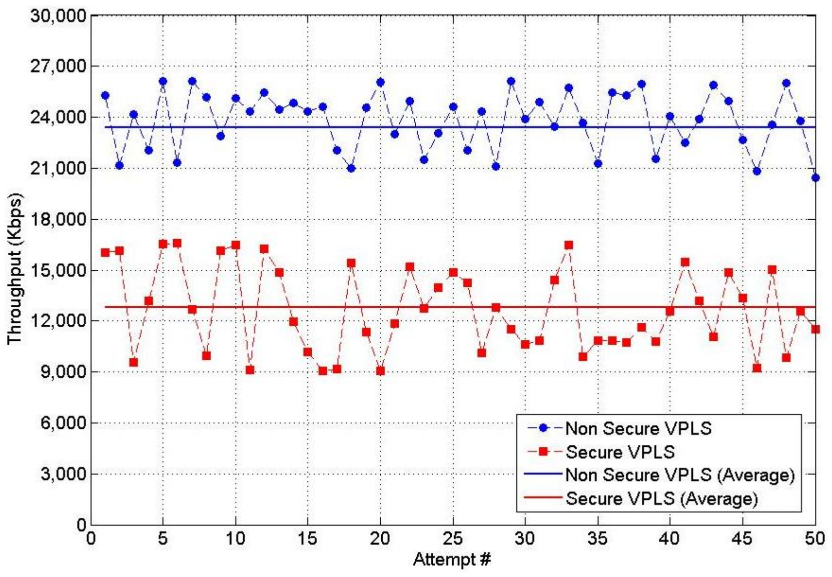
Figure 30. The performance penalty of security on TCP throughput.

On the other hand, the non-secure VPLS latency increment is below 2% compared with the non VPLS scenario in both short and long TCP sessions. Therefore, the impact of VPLS tunnel encapsulation on TCP throughput is very low. Similar to the TCP experiment, the additional layer of encryption is the main reason for the reduce average throughput of the secure VPLS architecture.

Furthermore, the experiment results reveal that UDP throughput is 14% higher for both short and long sessions than TCP throughput for secure VPLS architecture. Similarly, UDP throughput is 15% higher for short sessions and 14% higher for long sessions than TCP throughput for both non secure VPLS and non VPLS



(a) Short UDP session.



(b) Long UDP session.

Figure 31. The performance penalty of security on UDP throughput.

architectures. Moreover, the performance penalty of security on throughput is around 20% for both UDP and TCP sessions compared with both non VPLS scenario and non secure VPLS architecture. Thus, we can conclude that the performance penalty of security on throughput is independent of the transport layer protocol.

Wijesinha et al. observed that the maximum achievable UDP throughput is well below 50% (less than 27 Mbps) for 802.11g Wi-Fi connection at the maximum data rate of 54 Mbps even under ideal and controlled conditions [44]. In our experiments, the UDP throughput (26.5 Mbps) is almost similar to these findings and it verified the accuracy of our test bed.

5.4.3 Performance Penalty of Security on Jitter

In the third set of experiments, we measure the performance penalty of security on the jitter. The jitter of a UDP session is measured by using the IPERF networking tool. Each experiment repeated 50 times in both directions. Figures 32 and 33 illustrate the experiment results. The UDP bandwidth of IPERF is set to 54 Mbps.

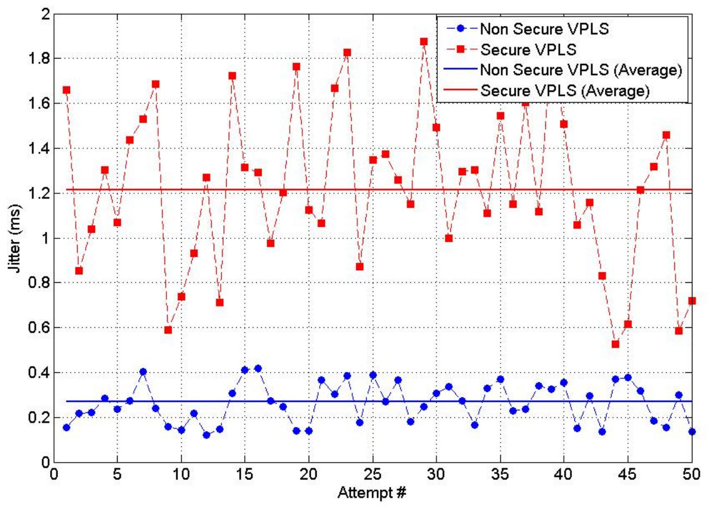


Figure 32. The performance penalty of security on jitter for short UDP session.

According to the experiment results shown in Figures 32 and 33, the average jitter of the secure VPLS architecture is about two times higher than the non-VPLS and non-secure scenarios for both short and long sessions. Thus, we conclude that the performance penalty of security on jitter is independent of the duration of the session for all scenarios. However, for the SCADA scenario, the average jitter of secure VPLS is still less than 0.5 ms. Thus, the performance penalty of security on jitter in this scenario will not affect the real time application such as VoIP video streaming in a short range network.

5.4.4 Other Comparisons

The average throughput, jitter and latency are compared based on the outcome of the experiments. The HIPswitch-300 devices have better performance than TEBs in all three performance metrics. HIPswitch-300 devices reduce the latency by 35% as shown in Figure 36 reduce jitter by 83% as shown in Figure 35, and increase the throughput by 8 times than TEB devices as shown

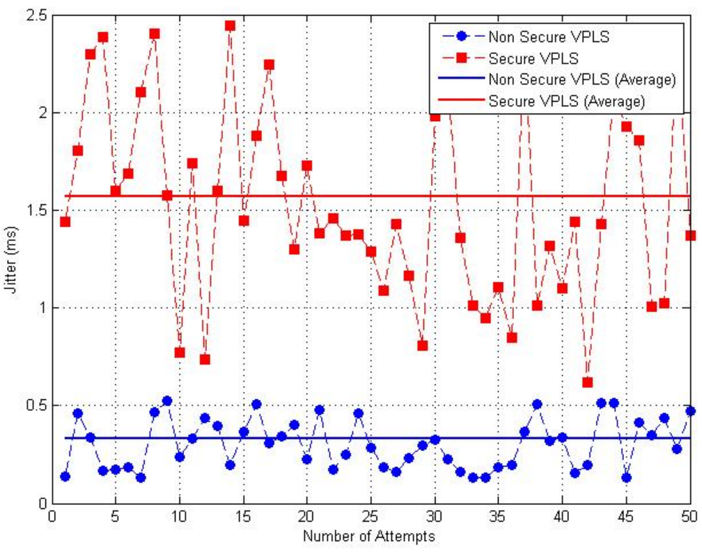


Figure 33. The performance penalty of security on jitter for long UDP session.

in Figure 34. Moreover, HIPswitch-200 devices reduce the latency by 13% and jitter by 46% and increase the throughput by 2 times than TEB devices. HIPswitch-300 devices have higher processing capabilities than TEBs, this is the main reason for the improved performance of VPLS networks using HIPswitch-300 devices. This result, therefore, opens up new opportunities for research. According to a recent Intel's white paper, IPsec acceleration is possible by using external accelerators and/or using new AES instruction sets for processors [45]. Thus, the adaptation of these techniques will further improve the performance of secure VPLS products.

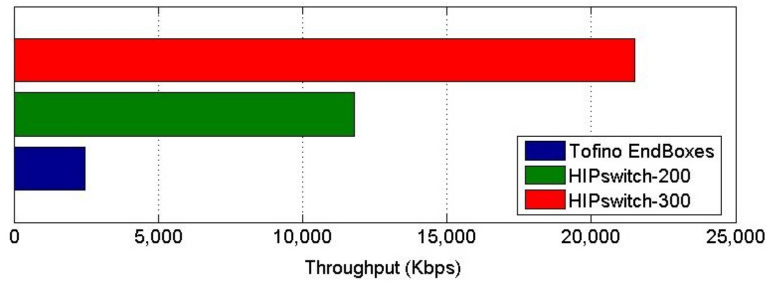


Figure 34. Average throughput comparison for VPLS network.

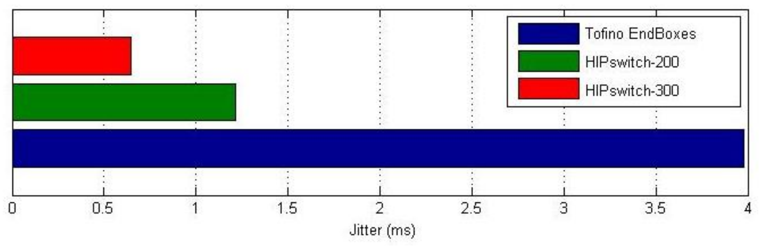


Figure 35. Average jitter comparison for VPLS network.

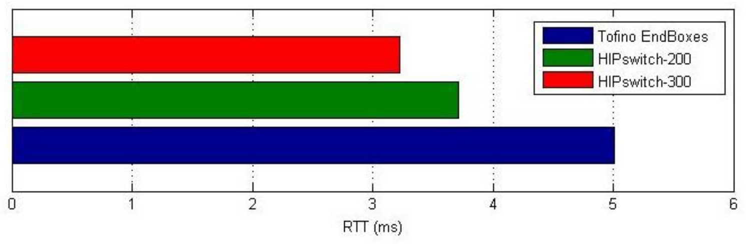


Figure 36. Average latency comparison for VPLS network.

6. Discussion and Future Works

HIP is becoming popular among many industrial applications. The HIP Working Group (HIP-WG) has published several interoperating implementations of HIP, most of which are open source. These applications spread from mobile applications to large scale cloud systems. Most of these specifications have been published as experimental RFCs, because the effects of the protocol on applications and on the Internet as a whole were unknown. Such experimental RFCs produced by the HIP WG allowed the community to experiment with HIP technologies and learn from these experiments. The HIP WG will now produce standards track versions of the main HIP RFCs taking as a base the existing Experimental RFCs. The WG will also specify certificate handling in HIP in a standards track RFC [46].

Earlier in April 2015, the IETF put up two Requests For Comments (RFC), RFC 7401 [47] and RFC 7402 [48]. RFC 7401 specifies a second version of HIP – HIPv2 which is an improvement on the earlier version defined by RFC 5201 [49], it mainly sets to address the limitations of the previous version as well as the issues raised by In-ternet Engineering Steering Group (IESG) such as crypto agility, hence ensuring more robust authentication mechanism. RFC 7402 specifies an Encapsulating Security Pay-load (ESP) Transport Format for HIP. Initial HIP implementation uses ESP only for HIP base exchange i.e. when setting up a HIP association between two hosts, however RFC 7402 specifies the use of ESP for protecting user data traffic after the HIP base exchange for integrity and optimal encryption. In addition, RFC 7402 specifies a set of HIP protocol extensions and their handling. Using these extensions, a pair of ESP Se-curity Associations (SAs) is created between the hosts during the base exchange. The resulting ESP Security Associations use keys drawn from the keying material (KEY-MAT) generated during the base exchange [48]. Other added features to the HIPv2 includes the addition of host mobility support in HIP, the extension of Domain Name System (DNS) to contain Host Identity information, using a rendezvous mechanism to contact mobile HIP hosts. This specification is intended to be an integral part of the HIPv2 [47]. In the following subsections, the industrial use cases of existing HIP implementations are discussed.

6.1 OpenHIP Implementation

VPLS security solutions are regarded by industrial networks as cost effective, high speed and multipoint connectivity model. Henderson et al. proposed a HIP based VPLS (HIPLS) architecture to secure VPLS network. HIPLS architecture is implemented by using OpenHIP implementation. HIPLS architecture is currently used in many industrial networks. Boeing, which is one of the largest airline manufactures in the world, is using HIPLS based VPLS network in the assembly line of Boeing 777 airplanes [50]. General Electronics (GE) is working on OpenHIP based security solution to secure the inter-train communication [51]. Moreover, two of the major SCADA (Supervisory Control and Data Acquisition) network appliance developing companies have already started to develop HIPLS based security solutions based on OpenHIP implementation. The first commercial product of OpenHIP implementation is released in 2009 by Canadian company Byres Security. They developed HIPLS based end-box product called Tofino Endboxes (TEBs) to establish secure VPLS network for SCADA and industrial process control systems [52]. In 2011, US based Asguard networks com-pany also released HIP enabled network appliances called Industrial Security Appli-ances (ISAs) based on HIPLS architecture by using OpenHIP implementation [53]. In 2013, Tempered Networks acquired Asguard Network and Tempered Networks is now offering a wide range of HIPLS based network appliances called HIPswitches [43]. In their second release, the HIPswitches are provided with a web based graphical user interface for the device configuration and maintenance via the HIP Conductor. Thus, the configuration process of HIPswitches is simple and user friendly. Moreover, HIP-switches 300 support inbuilt wireless connectivity as well as other utility functions like the High Availability (HA) feature. A more detailed documentation of the features of the Tempered Network HIP solution can be found in the appendix of this report.

6.2 HIPL Implementation

HIPL which started as a student project in 2001 was continued in 2002 at HIIT in Fuego-Core under InfraHIP and InfraHIP II projects. Efforts on implementation and interoperability with IndraNet, Ericsson, and Boeing provided substantial feedback tot he IETF drafts, hence the drive towards standardization by IETF. At initial stage HIPL was mainly kernelspace-oriented implementations. Not too long after, the HIPL implementation gradually moved from a research prototype towards an open source product. The first mobile implementation of HIP was developed based on HIPL implementation. In 2013, basic support for Android was included in the HIPL source code [5]. Many features were enabled in this mobile implementation over past two years. For instance, HIP based firewall functionality for Android has enabled in late 2014. HIPL implementation is also to secure cloud deployment in CERN. HIPL based security solution is utilized in cloud deployment that is used for analyzing CMS (Compact Muon Solenoid) data from CERN [54]. HIPL implementation provides the secure connectiv-ity and connection management capabilities for OpenStack based cloud.

Earlier in 2012, HIPL released its latest version, release 1.0.7, with improved code quality and prebuilt binaries for a number Linux distributions, namely Ubuntu, Fedora, and CentOS [5]. Several other features have been integrated in this release, features such as HIT or public-key based access control using HIP firewall, simple mobility to replace the need for multihoming, LSI support, IPsec heartbeat support, HIT-to-IP look up using DNS (domain specific), changing of ESP transform order, Public key to HI record conversion, DNS proxy support (to intercept and inject LSIs and HITs to apps) e.t.c [5].

6.3 Architectural Limitation of Secure VPLS Architectures

This section briefly presents the main architectural limitations in existing secure VPLS architectures.

6.3.1 Impact on L2 protocols

VPLS architectures allow the sharing of the same Ethernet broadcast domain over multiple geographically distributed sites. Generally, the legacy user equipment use various L2 network protocols such as STP (Spanning Tree Protocol), RARP Reverse Address Resolution Protocol, ARP (Address Resolution Protocol) in this Ethernet network. However, VPLS based inter-site connectivity links (VPN tunnels) are invisible to L2 devices and L2 protocol. These hidden links have very different behaviors than typical Ethernet links. Thus, many layer 2 protocols fail to function properly in VPLS based Ethernet network. For instance, STP fails to identify the loops over the provider network. It causes many issues such as broadcast storms, multiple frame transmissions, higher spanning tree convergence time and forwarding table instability.

Secure VPLS architectures are still incapable of providing a concrete solution for this issue. HIPLS proposed to evade the transmission of L2 PDUs (Protocol Data Units) over VPLS network. However, it is not a perfect solution since many SCADA and process control devices still rely on L2 protocols for their proper operation.

6.3.2 Static Tunnel Establishment and Maintenance Procedure

The existing secure VPLS architectures require to establish IPsec tunnels between PEs in the VPLS network. Moreover, authors proposed to maintain these tunnels for a long period to minimize the performance penalty due to the tunnel establishment procedure. The tunnel maintenance duration is static and predefined by the network administrators. However, some of the customer sites have very low traffic intensity between them.

As a result, some of these tunnels will not be used very frequently. The maintenance of a tunnel between such sites not only waste PE's resources such as memory, CPU, and ports but also occupy the network bandwidth for tunnel update messages. Therefore, it is necessary to fine tune the tunnel maintenance duration based on traffic demand. However, the existing secure VPLS architectures do not support dynamic parameter adjustment for IPsec tunnels.

6.4 Future Works

A potential future work to this thesis would be to extend the experiments beyond SCADA networks such as long-haul and telecommunication networks. For instance, using the Internet to interconnect two sites which are located at far away locations, possibly in different cities. Another possible extension could be an integration of HIP into the SDN architecture to provide security on the channel and then evaluating the effectiveness of such modification and the potential penalty on performance.

7. SUMMARY

This thesis work evaluated the performance penalty of security on various HIP based security solutions, using throughput, jitter, and latency as performance metrics. Two main scenarios were evaluated; the open-source Linux-based HIP implementations and the enterprise based solutions. Two separate instances of each scenario was implemented. The open-source Linux-based solutions were provided by HIPL and OpenHIP implementations, while the enterprise solutions were provided by Tempered Networks and Byres Security enterprise equipment.

For the enterprise based solutions, the performance penalty of security on throughput was around 20% for both UDP and TCP sessions compared with both non VPLS scenario and non secure VPLS architecture. Thus, it concludes that the performance penalty of security on throughput is independent of the transport layer protocol. On the other hand, non secure VPLS reduces the throughput only by 1% than the non VPLS scenario in both short and long TCP sessions. Therefore, the impact of VPLS tunnel encapsulation on TCP throughput is very low. The additional layer of encryption is the main reasons for the reduced average throughput of the secure VPLS architectures.

For open-source Linux-based implementations, the measurement results confirmed the assumption that no single solution is optimal in all considered aspects and scenarios. OpenHIP for instance experienced a major throughput limitation as shown in our experiment results, reaching a maximum of 100Mbps which does not seem very promising for future networks, hence we recommend a hardware acceleration for IPsec encryption as a possible means to remove this limitation. We also observed significant variations in the performance of both implementations when the tunnel is left idle for a prolonged duration, hence we recommend that tunnel maintenance capability be fine tuned for optimal performance after recovery from an idle state. In general, both HIP implementations have relatively low performance in terms of throughput, thus we recommend that future versions of both implementations focus on improving the network throughput.

Furthermore, experiments for enterprise based solutions revealed that UDP throughput is about 14% higher than TCP throughput for both secure VPLS and non secure VPLS traffic. Thus, UDP based applications can obtain better throughput performance than TCP based applications in a secure VPLS network as well. In general, secure VPLS architectures have relatively poor performance in terms of throughput and it opens new opportunities for research. Thus, the future secure VPLS architectures equally need to focus on improving the network throughput.

The secure VPLS architecture increases the latency approximately by 46% for the point to point approach and by about 87% for the multipoint approach, this increase is largely due to encryption and tunneling delays in PE devices. This encryption delay can be reduced by using high performing PE devices or by using efficient encryption algorithms. The IPsec acceleration can be achieved by using external accelerators and/or using new AES instruction sets for processors. Thus, the adaptation of these techniques for PEs will improve the performance of secure VPLS networks.

8. REFERENCES

[1] Gurtov A. & Komu M. (2009) Host Identity Protocol: Identifier/Locator Split for Host Mobility and Multihoming. The Internet Protocol Journal, vol.12, no. 1, pp. 27–31.

[2] Gurtov A. (2008) Host Identity Protocol: Towards the Secure Mobile Internet. Wiley, New Jersey, 332 p.

[3] Nikander P., Gurtov A., & Henderson R. T. (2010) Host Identity Protocol (HIP): Connectivity, Mobility, Multi-homing, Security, and Privacy Over IPv4 and IPv6 Networks. Communications Surveys & Tutorials, IEEE, vol. 12, no. 2, pp. 186 – 204.

[4] Nikander P. & Moskowitz R. (2006) Host Identity Protocol (HIP) Architecture. RFC4423.

[5] InfraHIPHIP: About InfraHIP. URL: http://infrahip.hiit.fi/index.php?index=about, Accessed November 24 2014.

[6] OpenHIP: About OpenHIP. URL: http://www.openhip.sourceforge.net/about.html, Accessed November 17 2014.

[7] Kompella K., & Rekhter Y. (2007). Virtual private LAN service (VPLS) using BGP for Auto-discovery and Signaling. RFC 4761.

[8] Lasserre M., & Kompella V. (2007). Virtual private LAN service (VPLS) using Label Distribution Protocol (LDP) Signaling . No. RFC 4762.

[9] Liyanage M. & Gurtov A. (2013) A Scalable and Secure VPLS Architecture for Provider Provisioned Networks. Wireless Communications and Networking Conference (WCNC) IEEE. IEEE, pp. 1115–1120.

[10] Liyanage M., Ylianttila M., & Gurtov A. (2013) Secure Hierarchical Virtual Private LAN Services for Provider Provisioned Networks. Communications and Network Security (CNS), 2013 IEEE Conference. IEEE, 2013, pp. 233–241.

[11] Mattes D., Henderson T., & Venema S. (2015) HIP-based Virtual Private LAN Service (HIPLS). HIP.

[12] Augustyn W., & Serbest Y. (2006). Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks. RFC 4665.

[13] Moskowitz R., Nikander P., Jokela P., & Henderson T. (2008). Host Identity Protocol (HIP). No. RFC 5201.

[14] Faigl Z., & Telek M. (2015). Modeling the Signaling Overhead in Host Identity Protocol-based Secure Mobile Architectures. MANAGEMENT, 11(3), 887-920.

[15] Liyanage M., & Gurtov A., (2012) Secured VPN Models for LTE Backhaul Networks. Vehicular Technology Conference (VTC Fall), IEEE. IEEE, pp. 1–5.

[16] Daoud K., Herbelin P., & Crespi N. (2008, September). UFA: Ultra Flat Architecture for High Bitrate Services in Mobile Networks. Personal, Indoor and Mobile Radio Communications. PIMRC 2008. IEEE 19th International Symposium on (pp. 1-6). IEEE.

[17] Venkateswaran R. (2001). Virtual Private Networks. Potentials, IEEE, 20(1), pp. 11-15.

[18] Scott C., Wolfe P., & Erwin M. (1999). Virtual Private Networks. O'Reilly Media, Inc.

[19] Lewis M. (2006) Comparing, Designing, and Deploying VPNs. Cisco Press, Indianapolis, 1080 p.

[20] Castoldi P., Baroncelli F., Martini B., et al (2004) Deliverable 2 Work Package 4 : Definition of Network Management and Control Requirements of Network Scenarios and Solutions Supporting Broadband Services for All.

[21] Tyson J. (2001) How Virtual Private Networks Work. Retrieved on July, vol. 31, p. 2008.

[22] Berger T. (2006) Analysis of Current VPN Technologies. Availability, Reliability and Security, 2006. ARES. The First International Conference on. IEEE, pp. 8–23.

[23] Diab W. B., Tohme S., & Bassil C. (2008) VPN Analysis and New Perspective for Securing Voice Over VPN Networks. Networking and Services, ICNS 2008. Fourth International Conference on. IEEE, pp. 73–78.

[24] Andersson L. & Madsen T. (2005) Provider Provisioned Virtual Private Network (VPN) Terminology. RFC 4026, March, Tech. Rep.

[25] Augustyn W. & Serbest Y. (2006) Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks. RFC 4665.

[26] McDysan D. & Carugi M. (2005) Service Requirements for Layer 3 Provider Provisioned Virtual Private Networks (PPVPNs). RFC 4031.

[27] Mattes D. & Moran A. (2015). Secure Connectivity Solution" Cloaks" Power Facility Networks. Power, 159(2), 39-41.

[28] Sajassi A. (2011). Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework. RFC 6136.

[29] Cisco Systems. (2004) VPLS and VPWS at a Glance.

[30] Liyanage M., Okwuibe J., Ylianttila A., & Gurtov A. (2015) Secure Virtual Private LAN Services: An Overview with Performance Evaluation. IEEE ICC 2015 - Workshop on Advanced PHY and MAC Techniques for Super Dense Wireless Networks ('ICC'15 - Workshops 13'). IEEE.

[31] Havrila P. (2012) L2 MPLS VPN Introduction and H3C Configuration Examples (Martini and Kompella VLLs/VPLS). HP Networking.

[32] Kompella K. L. M., & et al. (2003) Layer 2 VPNs Over Tunnels.

[33] Martini L. (2006) Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP).

[34] Bates T., Chen E., & Chandra R. (2006) BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP). RFC 4456, Tech. Rep.

[35] Martini L., Rosen E., El-Aawar N., & Heron G. (2006) Encapsulation Methods for Transport of Ethernet Over MPLS Networks. RFC4448.

[36] Andersson L., Doolan P., Feldman N., Fredette A., & Thomas B. (2001) LDP Specification, Work in Progress.

[37] Fang L. (2005) Security Framework for Provider-provisioned Virtual Private Networks (PPVPNs).

[38] Heffernan A. (1998) Protection of BGP Sessions via the TCP MD5 Signature Option. RCF 2385.

[39] Worster T., Rekhter Y., & Rosen E. Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE).

[40] Kuptsov D. & Gurtov A. (2009) Savah: Source Address Validation with Host Identity Protocol. Security and Privacy in Mobile Information and Communication Systems. Springer, pp. 190–201.

[41] LINKSYS WUMC710 Wireless-AC Universal Media Connector. URL: http://www.linksys.com/ca/p/P-WUMC710/. Accessed April, 2015.

[42] Iperf. URL: http://iperf.sourceforge.net/.Accessed January, 2015.

[43] Henderson T. & Gurtov A. (2014) The Host Identity Protocol (HIP) Experiment Re-port. Tech. Rep. Accessed August, 2014

[44] Tempered Networks. URL: http://www.temperednetworks.com/products. Accessed October, 2014.

[45] Wijesinha A. L., Song Y. T., Krishnan M. & et al. (2005) Throughput Measurement for UDP Traffic in an IEEE 802.11 g WLAN. Software Engineering, Artificial Intelligence, Networking and Par-allel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Con-ference on. IEEE, 2005, pp. 220–225.

[46] Carrier Cloud Telecoms - Exploring the Challenges of Deploying Virtualisation and SDN in Telecom Networks (2013). Intel Cooperation, Tech. Rep.

[47] Gonzalo C. & Manderson T. (2015) Host Identity Protocol (HIP), Charter for Working Group. URL: https://datatracker.ietf.org/wg/hip/charter/. Accessed July, 2015.

[48] Moskowitz R., Heer T., Jokela P. & Henderson T. (2015) Host Identity Protocol Version 2 (HIPv2). Tech. Rep.

[49] Jokela P., Melen J. & Moskowitz R. (2015) Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP).

[50] Moskowitz R., Nikander P., Jokela P., & Henderson T. (2008) Host Identity Protocol. RFC 5201.

[51] Henderson T. (2008) Boeing HIP Secure Mobile Architecture. Boeing SMA Team. IETF 73 HIPRG Meeting.

[52] GE Transportation. URL: http://www.getransportation.com/its. Accessed November, 2014.

[53] Tofino Security Appliance. URL: http://www.tofinosecurity.com/products/tofino-security-appliance. Accessed September, 2014.

[54] Asguard Networks. URL: http://www.asguardnetworks.com/. Accessed October, 2014.

[55] DII-HEP (CMS) cluster. URL: http://www.nordugrid.org/monitor/. Accessed March, 2015.

9. APPENDICES

Appendix 1 Technical specifications of Tempered Networks solution

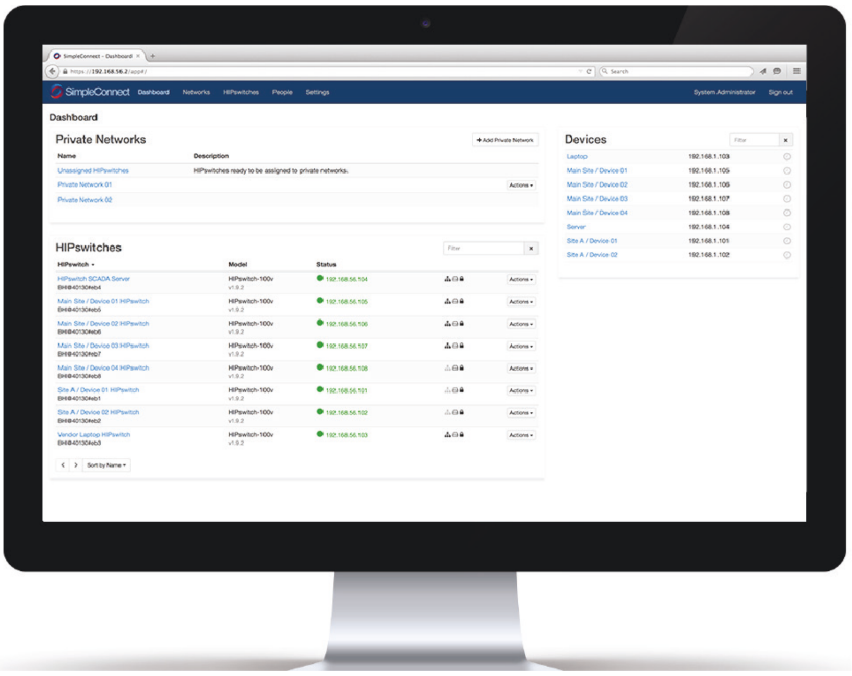
Appendix 1: Technical specifications of Tempered Networks solution

Product Line

HIPswitch Conductor & SimpleConnect Interface

The HIPswitch Conductor is a scalable orchestration engine that coordinates configuration, security policies, trust relationships, monitoring and analytics between the SimpleConnect UI and distributed HIPswitches.

The SimpleConnect web-based user interface enables single-pane-of-glass administration throughout the life cycle of a deployment.



Dashboard

Private Networks

Name	Description
Unassigned HIPswitches	HIPswitches ready to be assigned to private networks.
Private Network 01	
Private Network 02	

HIPswitches

HIPswitch ▾	Model	Status
HIPswitch SCADA Server EHIB40130#eb4	HIPswitch-100v v1.9.2	● 192.168.56.104
Main Site / Device 01 HIPswitch EHIB40130#eb5	HIPswitch-100v v1.9.2	● 192.168.56.105
Main Site / Device 02 HIPswitch EHIB40130#eb6	HIPswitch-100v v1.9.2	● 192.168.56.106
Main Site / Device 03 HIPswitch EHIB40130#eb7	HIPswitch-100v v1.9.2	● 192.168.56.107
Main Site / Device 04 HIPswitch EHIB40130#eb8	HIPswitch-100v v1.9.2	● 192.168.56.108
Site A / Device 01 HIPswitch EHIB40130#eb1	HIPswitch-100v v1.9.2	● 192.168.56.101
Site A / Device 02 HIPswitch EHIB40130#eb2	HIPswitch-100v v1.9.2	● 192.168.56.102
Vendor Laptop HIPswitch EHIB40130#eb3	HIPswitch-100v v1.9.2	● 192.168.56.103

Devices

Laptop	192.168.1.103
Main Site / Device 01	192.168.1.105
Main Site / Device 02	192.168.1.106
Main Site / Device 03	192.168.1.107
Main Site / Device 04	192.168.1.108
Server	192.168.1.104
Site A / Device 01	192.168.1.101
Site A / Device 02	192.168.1.102

Once HIPswitch security appliances register to the HIPswitch Conductor, a user who is logged into the SimpleConnect user interface is able to:

- Create secure private networks
- Delegate management of each private network to authorized users
- Add HIPswitches to private networks
- Define protected devices behind each HIPswitch
- Configure communication security policies for HIPswitches and devices
- Govern, audit and monitor private networks, HIPswitches and devices

HIPswitch Conductor

Network Connections: 2 x RJ45 Gig-E
Power: 350W Power Supply, 100-240VAC, 50-60Hz
Dimensions: 19.8”L x 17.2”W x 1.7”H
Mounting: 19” Rackmount, 1U Height
Operating Temperature: 0°C to +35°C
Humidity (non-condensing): 8% to 90%

Physical	Number of Licensed HIPswitches
HIPswitch Conductor SMB	2-9
HIPswitch Conductor ENT	Unlimited

HIPswitch-100 Series

Pictured: HIPswitch-100g (4.6"L x 1.8"W x 1.5"H)

The HIPswitch-100 physical product line consists of small form factor, industrially hardened security appliances with a secure network throughput of 5 Mbps (megabits per second). They are optionally DIN rail mountable, accept a wide range of DC input and can connect to the shared network with wired Ethernet, WiFi, or 3g cellular. The HIPswitch-100v is a virtual HIPswitch for laptop or desktop computers. Throughout is dependent on the host.

Throughput: 5 Mbps
Dimensions: 4.6”L x 1.8”W x 1.5”H inches
Mounting: Ruggedized Metal Channel Case, DIN Rail Mountable
Power: 8-42 VDC Input via Passive PoE, 2-3W Typical Operating Power
Operating Temperature: -40°C to +35°C
Humidity (non-condensing): 20% to 90%

Physical	**Network Connections**
HIPswitch-100e	2 x RJ45 Gig-E
HIPswitch-100w	1 x RJ45 Gig-E, 1 x 802.11 abg
HIPswitch-100g	1 x RJ45 Gig-E, 1 x 3g cellular
Virtual	**Host OS Requirements**
HIPswitch-100v	Windows 7 / Windows 8

Additional Specifications:
Power Options: PoE Injector
Cellular Providers: AT&T, T-mobile / Verizon / Rogers
Antenna Connections WiFi: RP SMA Female Main + Aux
Cellular: Single SMA Female
User Authentication Available: Yes

HIPswitch-200 Series

Pictured: HIPswitch-200g (4.1"L x 7.0"W x 1.6"H)

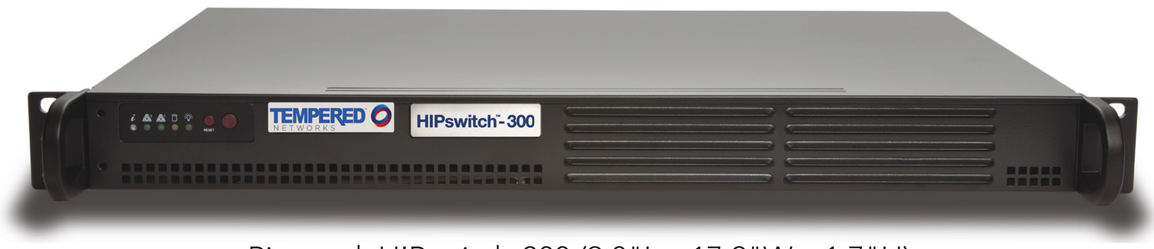
The HIPswitch-200 series of security appliances is a medium form factor, industrially hardened device with secure network throughput of 15 Mbps (megabits per second). They are optionally DIN rail mountable, accept a wide range of DC input, can connect to the untrusted shared network with wired Ethernet, WiFi, or 3g cellular connections and support failover across these different connection types.

Throughput: 15 Mbps
Dimensions: 4.1”L x 7.0”W x 1.6”H inches
Mounting: 20ga Metal Case, DIN Rail Mountable
Power: 8-48 VDC Input via multiple PoE or Barrel Jack,
5-7W Typical Operating Power
Operating Temperature: -40°C to +85°C
Humidity (non-condensing): 20% to 90%

Physical	**Network Connections**
HIPswitch-200e	2 x RJ45 Gig-E
HIPswitch-200w	1 x RJ45 Gig-E, 1 x 802.11 abg
HIPswitch-200g	1 x RJ45 Gig-E, 1 x 3g cellular

Additional Specifications:
Power Options: PoE Injector / Power Supply / Terminal Block Cellular Providers: AT&T, T-mobile / Verizon / Rogers
Antenna Connections WiFi: RP SMA Main + Aux
Cellular: Single SMA Female
User Authentication Available: Yes

HIPswitch-300 Series



Pictured: HIPswitch-300 (9.8"L x 17.2"W x 1.7"H)

The physical HIPswitch-300 is a 1U rack mount data-center grade security appliance. It has a secure network throughput of 60Mbps (megabits per second), and has 4 ports on the private network and 2 ports on the shared network. The HIPswitch-300v is a high throughput virtual security appliance that can be deployed within data-center and cloud environments. Throughput is dependent on the physical hardware.

Physical	**Network Connections**
HIPswitch-300	6 x RJ45 Gig-E
Virtual	Host OS Requirements
HIPswitch-300v	VMWare ESXi 5.x

Throughput: 60 Mbps
Dimensions: 9.8”L x 17.2”W x 1.7”H inches
Mounting: 19” Rackmount, 1U Height
Power: 200W Power Supply, 100-240VAC, 50-60Hz
Operating Temperature: 0°C to +35°C
Humidity (non-condensing): 8% to 90%
User Authentication Available: Yes

HIPswitch-400 Series

Pictured: HIPswitch-400 (25.6"L x 17.2"W x 1.7"H)

The HIPswitch-400 is a physical 1U rack mounted data-center grade security appliance. It has a secure network throughput of 850 Mbps (megabits per second), and has 4 ports on the private network and 2 ports on the shared network.

Physical	**Network Connections**
HIPswitch-400	6 x RJ45 Gig-E

Throughput: 850 Mbps
Dimensions: 25.6”L x 17.2”W x 1.7”H inches
Mounting: 19” Rackmount, 1U Height
Power: 500W Power Supply, 100-240VAC, 50-60Hz
Operating Temperature: 0°C to +35°C
Humidity (non-condensing): 8% to 90%
User Authentication Available: Yes