Enhancing security and scalability of Virtual Private LAN Services

Madhusanka Liyanage
University of Oulu, Finland

Doctoral Dissertation

Liyanage, Madhusanka, Enhancing security and scalability of Virtual Private LAN Services.
University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering; Centre for Wireless Communications; Infotech Oulu
Acta Univ. Oul. C 593, 2016
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

Ethernet based VPLS (Virtual Private LAN Service) is a transparent, protocol independent, multipoint L2VPN (Layer 2 Virtual Private Network) mechanism to interconnect remote customer sites over IP (Internet Protocol) or MPLS (Multiprotocol Label Switching) based provider networks. VPLS networks are now becoming attractive in many Enterprise applications, such as DCI (data center interconnect), voice over IP (VoIP) and videoconferencing services due to their simple, protocol-independent and cost efficient operation. However, these new VPLS applications demand additional requirements, such as elevated security, enhanced scalability, optimum utilization of network resources and further reduction in operational costs. Hence, the motivation of this thesis is to develop secure and scalable VPLS architectures for future communication networks.

First, a scalable secure flat-VPLS architecture is proposed based on a Host Identity Protocol (HIP). It contains a session key-based security mechanism and an efficient broadcast mechanism that increase the forwarding and security plane scalability of VPLS networks. Second, a secure hierarchical-VPLS architecture is proposed to achieve control plane scalability. A novel encrypted label-based secure frame forwarding mechanism is designed to transport L2 frames over a hierarchical VPLS network. Third, a novel Distributed Spanning Tree Protocol (DSTP) is designed to maintain a loop free Ethernet network over a VPLS network. With DSTP it is proposed to run a modified STP (Spanning Tree Protocol) instance in each remote segment of the VPLS network. In addition, two Redundancy Identification Mechanisms (RIMs) termed Customer Associated RIMs (CARIM) and Provider Associated RIMs (PARIM) are used to mitigate the impact of invisible loops in the provider network.

Lastly, a novel SDN (Software Defined Networking) based VPLS (Soft-VPLS) architecture is designed to overcome tunnel management limitations in legacy secure VPLS architectures. Moreover, three new mechanisms are proposed to improve the performance of legacy tunnel management functions: 1) A dynamic tunnel establishment mechanism, 2) a tunnel resumption mechanism and 3) a fast transmission mechanism. The proposed architecture utilizes a centralized controller to command VPLS tunnel establishment based on real-time network behavior.

Hence, the results of the thesis will help for more secure, scalable and efficient system design and development of VPLS networks. It will also help to optimize the utilization of network resources and further reduction in operational costs of future VPLS networks.

Keywords: Host Identity Protocol, network security, scalability, Software Defined Networking, Spanning Tree Protocol, Virtual Private LAN Services, Virtual Private Networks

Liyanage, Madhusanka, Virtuaalisten erillislähiverkkopalveluiden tietoturvan ja skaalautuvuuden tehostaminen.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta; Centre for Wireless Communications; Infotech Oulu
Acta Univ. Oul. C 593, 2016
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä

Ethernet-pohjainen VPLS (Virtual Private LAN Service) on läpinäkyvä, protokollasta riippumaton monipisteverkkomekanismi (Layer 2 Virtual Private Network, L2VPN), jolla yhdistetään asiakkaan etäkohteet IP (Internet Protocol)- tai MPLS (Multiprotocol Label Switching) -yhteyskäytäntöön pohjautuvien palveluntarjoajan verkkojen kautta. VPLS-verkoista on yksinkertaisen protokollasta riippumattoman ja kustannustehokkaan toimintatapansa ansiosta tullut kiinnostavia monien yrityssovellusten kannalta. Tällaisia sovelluksia ovat esimerkiksi DCI (Data Center Interconnect), VoIP (Voice over IP) ja videoneuvottelupalvelut. Uusilta VPLS-sovelluksilta vaaditaan kuitenkin uusia asioita, kuten parempaa tietoturvaa ja skaalautuvuutta, optimaalista verkkoresurssien hyödyntämistä ja käyttökustannusten pienentämistä entisestään. Tämän väitöskirjan tarkoituksena onkin kehittää turvallisia ja skaalautuvia VPLS-arkkitehtuureja tulevaisuuden tietoliikenneverkoille.

Ensin väitöskirjassa esitellään skaalautuva ja turvallinen flat-VPLS-arkkitehtuuri, joka perustuu Host Identity Protocol (HIP) -protokollaan. Seuraavaksi käsitellään istuntoavaimiin perustuvaa tietoturvamekanismia ja tehokasta lähetysmekanismia, joka parantaa VPLS-verkkojen edelleenlähetyksen ja tietoturvatason skaalautuvuutta. Tämän jälkeen esitellään turvallinen, hierarkkinen VPLS-arkkitehtuuri, jolla saadaan aikaan ohjaustason skaalautuvuus. Väitöskirjassa kuvataan myös uusi salattu verkkotunnuksiin perustuva tietokehysten edelleenlähetysmekanismi, jolla L2-kehykset siirretään hierarkkisessa VPLS-verkossa. Lisäksi väitöskirjassa ehdotetaan uuden Distributed Spanning Tree Protocol (DSTP) -protokollan käyttämistä vapaan Ethernet-verkkosilmukan ylläpitämiseen VPLS-verkossa. DSTP:n avulla on mahdollista ajaa muokattu STP (Spanning Tree Protocol) -esiintymä jokaisessa VPLS-verkon etäsegmentissä. Väitöskirjassa esitetään myös kaksi Redundancy Identification Mechanism (RIM) -mekanismia, Customer Associated RIM (CARIM) ja Provider Associated RIM (PARIM), joilla pienennetään näkymättömien silmukoiden vaikutusta palveluntarjoajan verkossa.

Viimeiseksi ehdotetaan uutta SDN (Software Defined Networking) -pohjaista VPLS-arkkitehtuuria (Soft-VPLS) vanhojen turvallisten VPLS-arkkitehtuurien tunnelinhallintaongelmien poistoon. Näiden lisäksi väitöskirjassa ehdotetaan kolmea uutta mekanismia, joilla voidaan parantaa vanhojen arkkitehtuurien tunnelinhallintatoimintoja: 1) dynaaminen tunnelinluontimekanismi, 2) tunnelin jatkomekanismi ja 3) nopea tiedonsiirtomekanismi. Ehdotetussa arkkitehtuurissa käytetään VPLS-tunnelin luomisen hallintaan keskitettyä ohjainta, joka perustuu reaaliaikaiseen verkon käyttäytymiseen.

Tutkimuksen tulokset auttavat suunnittelemaan ja kehittämään turvallisempia, skaalautuvampia ja tehokkaampia VLPS järjestelmiä, sekä auttavat hyödyntämään tehokkaammin verkon resursseja ja madaltamaan verkon operatiivisia kustannuksia.

Asiasanat: Host Identity Protocol, skaalautuvuus, Software-defined Networking (SDN), Spanning Tree Protocol, verkon tietoturva, VPLS, VPN-verkot

This is dedicated to my wife Ruwi and to my parents.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Professor Mika Ylianttila, for his comments, criticism, advice, endless patience, inspiration and encouragement, all of which added considerably to my graduate experience. I would like to thank Adjunct Professor Andrei Gurtov for hiring me at CWC and acting as co-supervisor for this thesis. I appreciate his vast knowledge and skills in a multitude of areas, as well as his guidance in directing me towards inspiring research projects. I would also like to thank the heads of CWC research groups, Professor Jari Iinatti and Professor Matti Latva-aho, for providing the necessary guidance, leadership and financial assistance for this project.

I would like to thank my follow-up group committee, the Chairman of the committee, Professor Savo Glisic, and Dr. Pradeep Kumar for the assistance they provided at all levels of my research. I cannot express how grateful I am to have Professor Nandana Rajatheva, who has been there for me any time I knocked on his door. I appreciate him immensely for his availability to talk about the research as well as my personal problems. He has helped me greatly in making good decisions on numerous occasions.

I am grateful to the official reviewers, Professor David Hutchison from Lancaster University, United Kingdom and Professor Aruna Prasad Seneviratne from University of New South Wales, Australia for their valuable feedback. I would also like to thank Professor Tarik Taleb from Aalto University, Finland for serving as the opponent at the defense.

I want to express my gratitude to the University of Oulu Graduate School, HPY Foundation, Nokia Foundation, Tauno Tönning foundation, FINCEAL Plus and CWC projects including MEVICO, SIGMONA, TWB, and Naked Approach for their financial support. Moreover, I would like to thank the project partners of the EU COST action projects IC1301, IC1303, CA15107 and CA15127 for their support. I would like to thank all the administrative staff: Jari Sillanpaa, Kirsi Ojutkangas, Eija Pajunen, Anu Niskanen, Elina Komminaho and Hanna Saarela who made my life lot easier at CWC. Also, I would like to thank my mentor Petri Komulainen for his advice and guidance in completing my PhD.

I am grateful to have Ijaz Ahmed, Suneth Namal, Jani Pellikka, Tanesh Kumar, Tenager Mekonnen, Erkki Harjula, Pawani Porambage and Jude Okwuibe as research group members. The regular discussions, meetings, lunches and coffee breaks with them

were all memorable. Some special words of gratitude also go to all of my col- leagues at CWC, including (but certainly not limited to) Antti Tölli, Professor Markku Juntti, Kalle Lahetkangas, Qiang Xue, Marian Codreanu, Mehdi Bennis, Shahriar Shahabuddin, Satya Joshi, Timo Koskela, Timo Braysy, Pekka Pirinen, Ari Pouttu and Zaheer Khan.

I am grateful to have such wonderful friends in Oulu. Thank you very much Abishek, Buddika, Chamari, Dhaminda, Diana, Dimuthu, Feroz, Heshani, Inosha, Kamal, Manosha, Mika, Nuwan, Praneeth, Raisa, Mrs. Rajatheva, Sandun, Saliya, Sumudu, Tharanga, Uditha and Upul. I would also like to thank the Oulu Cricket Club members and Indian Community. It was a great pleasure to play cricket at weekends to relax after a stressful research week.

I would also like to thank my sister, my brother, their families, and my parents-in-law for the support they have provided me throughout my life. Finally, I must thank my wonderful wife, the love of my life, whose love and encouragement helped me finish this dissertation. The last word goes to my parents (Amma and Thaththa) for their endless support and encouragement to achieve this target in my life.

Oulu, September, 2016 Madhusanka Liyanage

Abbreviations

AAA	*Authentication Authorization and Accounting*
AAPELE	*Algorithms, Architectures and Platforms for Enhanced Living Environments*
ACL	*Access Control List*
AES	*Advanced Encryption Standard*
AFI	*Address Family Identifiers*
AGI	*Attachment Group Identifier*
AH	*Authentication Header*
AP	*Application Plane*
ARP	*Address Resolution Protocol*
AS	*Autonomous Systems*
ATM	*Asynchronous Transfer Mode*
BEET	*Bound End-to-End Tunnel*
BEX	*Base Exchange*
BGP	*Border Gateway Protocol*
CARIM	*Customer Associated RIM*
CBC	*Cipher-block chaining*
CDF	*Cumulative Distribution Function*
CE	*Customer Edge*
CEK	*Content Encryption Key*
CEs	*Customer Edge devices*
CoS	*Class of Service*
CP	*Control Plane*
CPU	*Central Processing Unit*
CPVPN	*Customer Provisioned VPN*
DCI	*Data Center Interconnect*
DDoS	*Distributed DoS*
DH	*DiffieŰHellman*
DHCP	*Dynamic Host Configuration Protocol*
DMZ	*Demilitarized Zone*
DNS	*Domain Name Services*

DoS	*Denial of Service*
DP	*Data Plane*
DCE	*Designated CE*
DPE	*Designated PE*
DPI	*Deep Packet Inspection*
DSL	*Digital Subscriber Line*
DSTP	*Distributed STP*
EPC	*Evolved Packet Core*
ESP	*Encapsulation Security Payload*
FEC	*Forwarding Equivalence Class*
FIB	*Forwarding Information Base*
FR	*Frame Relay*
FTP	*Fast Transmission Procedure*
HDLC	*High-Level Data Link Control*
H-HIPLS	*Hierarchical HIP enabled virtual private LAN Service*
HIP	*Host Identity Protocol*
HIPL	*HIP Linux*
HIPLS	*HIP-enabled Virtual Private LAN Service*
HI	*Host Identity*
HIT	*Host Identity Tag*
HMAC	*Hash-based Message Authentication Code*
HTTP	*HyperText Transport Protocol*
H-VPLS	*Hierarchical VPLS*
H-VPLS	*Hierarchical VPLS*
I	*Initiator*
ICMP	*Internet Control Message Protocol*
IETF	*Internet Engineering Task Force*
IETF	*Internet Engineering Task Force*
IKE	*Internet Key Exchange*
IKEv2	*Internet Key Exchange version 2*
IP	*Internet Protocol*
IPLS	*IP-only LAN-like Service*
IPSec	*IP Security*
IRACON	*Inclusive Radio Communication Networks for 5G and beyond*
IRTF	*Internet Research Task Force*

ISA	*Industrial Security Appliance*
ISP	*Internet Service Provider*
IT	*Information Technology*
IV	*Initialization Vector*
KDC	*Key Distribution Center*
KEK	*Key Encryption Key*
L2	*Layer 2*
L2TP	*Layer 2 Tunneling Protocol*
L2TPv3	*Layer 2 Tunneling Protocol Version 3*
L2VPN	*Layer 2 Virtual Private Network*
L3	*Layer 3*
L3VPN	*Layer 3 Virtual Private Network*
LAN	*Local Area Network*
LDP	*Label Distribution Protocol*
LISP	*Locator Identifier Separation Protocol*
LSI	*Local Scope Identifier*
LTE	*Long Term Evolution*
LTS	*Long Term Support*
MAC	emphMessage Authentication Code
MAC	*Medium Access Control*
MEVICO	*Mobile Networks Evolution for Individual Communications Experience*
MiTM	*Man-in-the-Middle*
MOBIKE	*IKEv2 Mobility and Multihoming Protocol*
MPLS	*Multiprotocol Label Switching*
MTU	*Maximum Transmission Unit*
NAF	*Network Advertisement Frames*
NAP	*Network Advertisement Packet*
NAP	*Nordic perspective to gadget-free hyperconnected environments (The Naked Approch)*
NAT	*Network Address Translation*
NFV	*Network Function Virtualization*
NLRI	*Network Layer Reachability information*
NOS	*Network Operating System*
n-PE	*Network-facing Provider Equipment*
NSI	*Network Segment Identifier*

OAM	*Operations, Administration, and Management*
OF	*OpenFlow*
OS	*Operating System*
OSI	*Open System Interconnection*
PARIM	*Provider Associated RIM*
PDF	*Probability Distribution Function*
PDU	*Protocol Data Unit*
PE	*Provider Edge*
PE	*Provider Edge Equipment*
PKI	*Public Key Infrastructure*
PPL2VPN	*Provider Provisioned Layer 2 VPN*
PPL3VPN	*Provider Provisioned Layer 3 VPN*
PPP	*Point to Point Protocol*
PPTP	*Point-to-point Tunneling Protocol*
PPVPN	*Provider Provisioned VPN*
PW	*Pseudowire*
PVST	*Per-VLAN Spanning Tree*
PVST+	*Per-VLAN Spanning Tree Plus*
QoE	*Quality of Experience*
QoS	*Quality of Service*
R	*Responder*
RAM	*Random Access Memory*
RECODIS	*Resilient communication services protecting end-user applications from disaster-based failures*
RD	*Route Distinguisher*
RFC	*Request for Comments*
RIM	*Redundancy Identification Mechanism*
RP	*Resumption Parameter*
RPVST	*Rapid Per-VLAN Spanning Tree*
RR	*Route Reflectors*
RST	*Reset*
RTF	*Route Target Filtering*
RTT	*Round Trip Time*
SA	*Security Associations*
SAFI	*Subsequent Address Family Identifier*

SCADA	*Supervisory Control and Data Acquisition*
SDN	*Software Defined Network*
S-HIPLS	*Session key based HIP-enabled virtual private LAN Service*
SIGMONA	*SDN Concept in Generalized Mobile Network Architectures*
SLA	*Service Level Agreement*
SME	*Security Management Entity*
SPI	*Security Parameters Index*
SPM	*Session Parameter Measuring*
SSL	*Secure Socket Layer*
STP	*Spanning Tree Protocol*
SYN	*Synchronization*
TCAM	*Ternary Content-Addressable Memory*
TCN	*Topology Change Notification*
TCP	*Transmission Control Protocol*
TEB	*Tofino Endbox*
Tekes	*Finnish Funding Agency for Technology and Innovation*
TEP	*Tunnel Establishment Procedure*
TID	*Tunnel Identifier*
TLS	*Transport Layer Security*
TM App	*Tunnel Management Application*
TRP	*Tunnel Resumption Procedure*
TWB	*Train Wireless Bus, Wireless solutions for urban transit environments*
UDP	*User Datagram Protocol*
UEs	*User Equipments*
UMTS	*Universal Mobile Telecommunications System*
u-PE	*User-facing Provider Equipment*
WAN	*Wide Area Network*
VBO	*VE Block Offset*
VBS	*VE Block Size*
VC	*Virtual Circuit*
VC-LSP	*Virtual Circuit Label Switched Path*
VE	*VPLS Edge Device*
WG	*Working Group*
WIPE	*Wireless Power Transmission for Sustainable Electronics*
WLAN	*Wireless LAN*

VLAN	*Virtual LAN*
VoIP	*Voice over IP*
VPLS	*Virtual Private LAN Services*
VPN	*Virtual Private Network*
VPWS	*Virtual Private Wire Services*
VR	*Virtual Router*
VR	*Virtual Router*
VSI	*Virtual Switch Instance*
WWW	*World Wide Web*

Contents

1 Introduction

1.1 Motivation

The development of the global economy encourages many enterprises to span their services across several geographical regions. Such multi-regional expansion allows them to provide high quality services to an extensive client base. Naturally, frequent communication between multi-regional sites is mandatory for the smooth operation of the company. Specifically, regional branches regularly need to communicate with their head offices. In the past, employees had to frequently travel between company branch offices to exchange corporate information and to participate in project meetings. Since business traveling is highly time and money consuming, multinational enterprises seek out services and communication systems that enable them to interconnect their branches, so that their employees can easily access enterprise networks anytime from anywhere in the world. Such corporate clients are looking for flexible, secure, manageable, scalable, and low cost networking solutions [1–3].

Since most of the enterprises do not have their own globally spanning communication network, they usually obtain communication services from service providers. In the past, service providers used leased lines to provide such network services. However, these leased lines have significant disadvantages [4–7]. For instance, leased lines are not suitable when there are a large number of branches or when the number of branches grows quickly. Moreover, leased lines are relatively expensive and difficult to manage. As time went by, Asynchronous Transfer Mode (ATM) and Frame Relay (FR) based networks emerged and service providers started to utilize virtual circuits. With these new technologies, enterprises were able to establish Layer 3 (L3) network connections based on virtual circuits. However, the virtual circuits supported only point-to-point links. Moreover, they were difficult to configure and maintain, especially when new sites were deployed. Later, IP based packet switching networks became popular. Presently, IP based networks are used almost everywhere in the world [4, 5, 8].

Initially, L3VPNs (L3 Virtual Private Networks) were the preferred choice for many service providers. However, L3VPNs still suffer from high operating costs and compatibility issues. This motivates service providers to seek alternative solutions to provide low-cost private network services. As a result, L2VPNs (Layer 2 Virtual Private

Networks) such as VPLS (Virtual Private LAN Services) have become mainstream as cost effective and protocol independent solutions.

VPLS networks were initially utilized only in industrial networks [9]. Such networks have various SCADA (Supervisory Control and Data Acquisition) and control systems. Generally, legacy SCADA systems are designed for static network environments and they often use protocols which expect flat networks or single broadcast domains [10]. For instance, most of them support only OSI (Open System Interconnection) Layer 2 (L2) protocols. The utilization of dynamic network addressing with higher layer protocols, such as DHCP (Dynamic Host Configuration Protocol) and DNS (Domain Name Services), or IP based packet transportation is still fairly rare in many industrial networks. The use of L3VPN is challenging, and in many cases, not possible in industrial networks. Thus, industrial enterprises need L2 network virtualization techniques. These techniques can be categorized into two main types; namely, Virtual Private LAN Service (VPLS) and Virtual Private Wire Service (VPWS) [11, 12]. However, VPLS architectures are suitable for many more industrial networks than VPWS since VPLS provides Ethernet based multipoint-to-multipoint connectivity.

VPLS provides many advantages over L3 VPNs [9, 12, 13]. VPLS is an ideal alternative for the enterprises who are using non-IP protocols or who might be reluctant to move to a Layer 3 service because of concerns over sharing routing information or for other internal operational reasons. In VPLS networks, enterprises maintain complete control over the routing in their network segments. From a security perspective, the external service provider has no awareness of the enterprisesŠ private IP address space or routing protocols. VPLS also provides a transparent and protocol-independent service for companies. No L2 protocol conversion happens between LAN and WAN (Wide Area Network) interfaces.

Moreover, VPLS networks provide more flexibility and manageability for the private network segment than L3VPNs. Since enterprises have total control of their own IP routing, their IT (Information Technology) department can be highly agile in responding to varying levels of customer demands. VPLS networks allow them to conduct rapid reconfigurations themselves without informing the service provider or waiting for the provider to react to a request. Moreover, VPLS supports fast recovery during network faults by reducing the network down-time and providing higher corporate efficiency and productivity. The cost of operation is lower for VPLS customers than L3VPN users due to several reasons. First, VPLS enables convergence of network services such as VoIP, Video streaming and other cooperate network services. Thus, all network traffic can be transported over a single Ethernet interface. This means that companies can eliminate the requirement to lease multiple communication links. Second, companies do not need to provide additional training on WAN skills or hire WAN specialists to operate their network. They can operate their private network segment by using the same staff who operate their LAN network. Third, VPLS only requires low cost switches as it needs smaller and fewer routers than L3VPN solutions do.

As an L2 solution, VPLS has a zero-hop delay at the core of the network. Therefore, VPLS can achieve lower latencies and better jitter performance than L3VPNS. Furthermore, VPLS also provides the ability to add new sites without the need to reconfigure service provider equipment or the local equipment at existing sites. For these reasons, VPLS networks are now becoming attractive in many enterprise applications such as DCI (Data Center Interconnect), Voice over IP (VoIP) and videoconferencing services. According to an Alcatel-Lucent white paper [14], 47 percent of VPN traffic was operated via VPLS by 2012.

However, new VPLS applications demand additional features, such as elevated security, enhanced scalability, optimum utilization of network resources and further reduction in operational cost. Hence, in this thesis, our motivation is to design new VPLS architectures, as well as to propose modifications to the existing VPLS architecture in order to enhance security and scalability. The main findings of the thesis are presented in the following chapters.

1.2 Author’s contribution

The author’s research at the University of Oulu was directed toward three main research fields: to design novel secure VPLS architectures, to design secure communication architectures for future mobile networks and to conduct a security analysis of IoT (Internet of Things) networks. The author has published five journal/magazine articles [15–19], 15 conference papers [20–34], one patent [35], one edited book [36] and two book chapters [37, 38] since 2012. Author also contributed to one white paper [39] which was published by Nokia. During his research time at the University of Oulu, the author has received twelve awards and grants including the best Paper Award [28] at the 9th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST,2015), in Cambridge, in the UK in 2015.

In addition, the author has worked as a project manager, researcher and management committee member in various projects (MEVICO[40], SIGMONA[41], TWB[42], Naked Approach[43], AAPPLE[44], IRACON[45], WIPE[46], RECODIS[47]) at the University of Oulu during 2012-2016 with contributions in research project management, research group leadership, research project proposal preparation, project progress documentation and graduate student supervision, in addition to carrying out actual research work. In 2015, the author was awarded the best researcher award by the Centre for Wireless Communications for his excellent contribution in project management and project proposal preparation. Moreover, MEVICO project received the CELTIC Excellence Award (SILVER Star) in 2013.

However, this thesis is written based only on the work related to the design of secure VPLS architectures [15, 17, 21–23, 26, 31]. The author had the main responsibility of creating the original ideas, carrying out the analysis, developing the experiment results, generating the numerical results, and writing the research papers. Other authors provided invaluable comments, criticism, and guidance during the process.

The first contribution of this thesis is the design of a scalable secure flat-VPLS architecture based on the Host Identity Protocol (HIP). It contains a novel session key-based security mechanism to increase the forwarding and security plane scalability of secure VPLS networks. Moreover, the proposed architecture offers a higher degree of security features than other existing secure VPLS architectures. The performance of the proposed architecture is analyzed in a simulation model. Several IP based attacks are mounted in the simulation model to confirm the security features of the proposed architecture. Finally, a real test bed implementation is used to analyze the data plane performance of existing secure VPLS architectures. New results are presented in articles [15, 21] and [26].

The second contribution of this thesis is the design of a scalable and secure hierarchical-VPLS architecture based on HIP. This increases the scalability of the previously proposed flat-VPLS architecture by providing control plane scalability. A novel encrypted label based secure frame forwarding mechanism is proposed to transport L2 frames over the hierarchical VPLS network. The proposed VPLS architecture offers higher control and forwarding plane scalability than existing secure VPLS architectures and provides the same level of control and forwarding plane scalability as other unsecured hierarchical VPLS architectures. It also provides similar levels of security as other secure VPLS architectures. Moreover, the proposed architecture distributes the service provision across different PE functions to minimize the utilization of network resources, such as memory space and processing power at each node. Initially, the scalability limits of the proposed architecture are quantitatively analyzed. Then, extensive simulations are conducted to verify the findings. The simulation model is also used to confirm the security features. Finally, the proposed architecture is implemented in a real world test bed and the performance is compared with other VPLS architectures. New results are presented in articles [17] and [22].

The third contribution of this thesis is the design of a novel Distributed STP (DSTP) to maintain a loop free Ethernet network over a VPLS network. This method proposes utilizing a modified STP instance in each remote segment of the VPLS network. Furthermore, two Redundancy Identification Mechanisms (RIMs) called Customer Associated RIMs (CARIM) and Provider Associated RIMs (PARIM) are proposed to prevent functional issues due to invisible loops in the provider network. Extensive simulations are conducted to analyze the performance of the proposed DSTP. New results are presented in article [23].

The fourth contribution of this thesis is the design of a novel SDN based VPLS architecture to overcome the tunnel management limitations of legacy secure VPLS architectures. The proposed architecture utilizes IPsec enabled OpenFlow switches as VPLS nodes and the OpenFlow protocol to install flow rules. A dynamic tunnel management mechanism is proposed to estimate tunnel duration based on real time network statistics. Moreover, novel tunnel resumption and fast transmission mechanisms are proposed to reduce tunnel establishment delays of subsequent tunnel establishments. Furthermore, these mechanisms also reduce overall data transmission delays. The proposed architecture dynamically adjusts the tunnel duration by analyzing the traffic patterns of the VPLS network. Extensive simulations are conducted to reveal the expected advantages of the proposed architecture. Finally, the proposed architecture is implemented in a test bed to measure the data plane performance. New results are presented in articles [31].

1.3 Scope of the thesis

This thesis has 7 chapters. The contributions of each chapter are summarized as follows:

- **Chapter 2:** The state of the art relevant to the scope of the thesis is discussed in this chapter. The discussion is divided into four sections. First, an overview of VPNs is presented. Then, the general VPLS architecture, different VPLS types and industrial applications of VPLS networks are detailed. Subsequently, the relevant security protocols are discussed. Lastly, emerging technologies which can be used to enhance the performance of VPLS networks are discussed.

- **Chapter 3** This chapter presents a scalable secure flat-VPLS architecture based on Host Identity Protocol (HIP). It describes a proposed session key-based security mechanism and an efficient broadcast mechanism that increase the forwarding and security plane scalability of secure VPLS networks. Finally, simulations and test bed experiment results are presented to analyze the performance of this proposed architecture and to verify its security features.
- **Chapter 4:** This chapter presents a scalable secure hierarchical-VPLS architecture based on HIP. It describes a proposed encrypted label based secure frame forwarding mechanism to transport L2 frames over a hierarchical VPLS network. The security features of the proposed architecture are analyzed. Simulation results are presented to verify the performance of proposed architecture and to confirm its security features. Finally, the test bed implementation of the proposed architecture is presented.
- **Chapter 5:** This chapter presents a novel Distributed Spanning Tree Protocol (DSTP) to maintain a loop free Ethernet network over a VPLS network. With the DSTP it is proposed to run a modified STP (Spanning Tree Protocol) instance in each remote segment of the VPLS network. In addition, proposed Redundancy Identification Mechanisms (RIMs) called Customer Associated RIMs (CARIM) and Provider Associated RIMs (PARIM) are explained in detail. Finally, simulation results are presented to verify the performance advantages of the proposed DSTP mechanism.
- **Chapter 6:** This chapter presents a novel SDN (Software Defined Networking) based VPLS (SoftVPLS) architecture to overcome tunnel management limitations in legacy secure VPLS architectures. Moreover, the chapter contains detailed descriptions of proposed performance enhancing mechanisms: 1) a dynamic tunnel establishment mechanism, 2) a tunnel resumption mechanism, and 3) a fast transmission mechanism. Finally, the simulation model and the test bed implementation results are presented to analyze the performance of the proposed architecture.
- **Chapter 7:** This chapter presents the conclusions of the thesis and how these proposed architectures can be extended by future research.

2 State of the art

In this chapter, the state of art that is relevant to the scope of the thesis is discussed. The discussion is divided into four sections. First, an overview of VPNs is presented. Then, the general VPLS architecture, different VPLS types and industrial applications of VPLS networks are detailed. Next, the relevant security protocols are discussed. Lastly, we discuss the emerging technologies which can be used to enhance the VPLS network performance.

2.1 Virtual private networks

A virtual network is a collection of virtual links which logically interconnect the different network components over a physical network. These virtual links are transparent for end users and they are implemented by using the methods of network virtualization. There are two forms of network virtualization techniques, namely protocol based virtual networks such as VPNs, and the virtual device based virtual networks such as virtual machines. However, protocol based virtual networks, i.e. VPNs are more globally ubiquitous than the virtual device based virtual networks due to their simple implantation aspects. The notion of VPN has been around as long as the availability of data communication networks. However, VPNs became popular during past two decades due to the evolution of related technologies. Primarily, the implementation cost of virtual networks had dropped drastically as a result of availability of low cost network equipment and communication systems.

2.1.1 The history of VPNs

The history of virtual networks goes as far back as the history of data networks [4, 7, 48]. The first generation virtual networks were developed by using the operatorŠs dial-up connections or dedicated leased lines in the early 1970s. Then, the adaptation of X.25 introduced the virtual connection concept for data networks in the 1980s. A logical separation of the customer virtual network over the provider network was achieved by using connection oriented virtual connections. Furthermore, X.25 offered virtual networks with TCP/IP protocols. In the 1990s, new network technologies such as high speed frame relays and ATM switching were introduced. This achieved high speed virtual network connections up to 155 Mbps. Meantime, the World-Wide Web (WWW) and the Internet became popular among the network community. Thus, customers were motivated to use IP based services. On the other hand, traditional VPN services such as ATM and Frame Relay were in the decline stage of the product life cycle. Hence, the service providers also began to migrate to IP based network infrastructures. These migrations fueled the fast adoption of IP based VPNs.

Moreover, IP VPNs permit building fully meshed networks on top of provider networks. They offer significant cost reductions, high scalability and strong Class of Service (CoS) to handle the convergent traffic of voice, data and video. Due to these factors, service providers started to offer IP VPNs as value added services (VAS). IETF (Internet Engineering Task Force) formed a working group (WG) named Provider Provisioned IP VPNs (PPVPN) [49] to standardize the provider provisioned IP VPNs. Later, L2VPN [50] and L3VPN [51] WGs also paid close attention to VPN standardization.

2.1.2 The taxonomy of VPNs

Figure 1 illustrates the taxonomy of virtual networks [4–8, 48, 52].

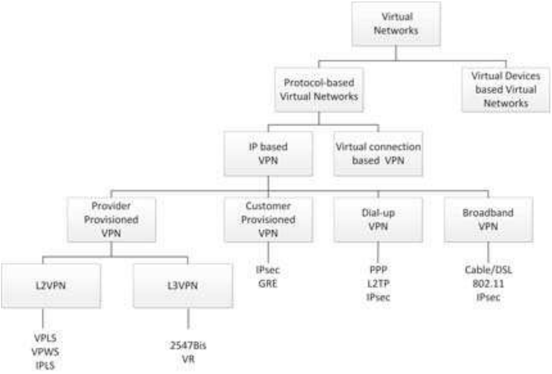


Fig. 1. The taxonomy of virtual networks.

The history of virtual networks clearly showed that the progression of virtual networks converged for IP based VPNs. There are several forms of IP based VPNs. However, Provider Provisioned VPNs (PPVPNs) and Customer Provisioned VPNs (CPVPNs) are the most common forms [4, 5]. Table 1 presents a comparison of the different IP based VPN types.

Table 1. The properties comparison of different IP based VPN architectures.

	PPVPNs	CPVPNs
Operation point	PEs	CEs,
Scalability	High	Low
Service provisioning	By the Provider	By the Customer
Membership discovery	Automatic	Manual Configuration
Security	Low	High
Ubiquitous availability	High	High
Support for lawful interception	Yes	No
Support for QoS mechanisms	Low	High
Flexibility to use different protocols	Very High	Limited
Operating layer	L2,L3	L3,L4

2.1.3 Provider provisioned VPNs

The service provider is responsible for all operational and management tasks of a provider provisioned VPN. Figure 2 illustrates a provider provisioned VPN.

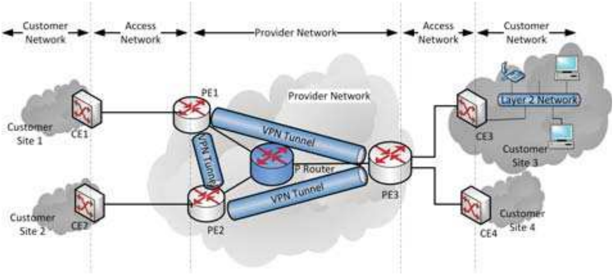


Fig. 2. A provider provisioned VPN.

Basically, these are five key elements in a provider provisioned VPN.

- **Customer Edge Equipment (CE)** : CE is host equipment or a router or a switch which is located on the customerŠs premises. It works as the middle device between the customer network and the provider network.
- **Provider Edge Equipment (PE)**: PE is the place where all the VPN intelligence resides in a provider provisioned VPN. PEs belong to the service provider and each PE maintains per-VPN routing databases. These databases contain information about VPNŠs tunneling topology, route details and forwarding tables.
- **The Provider Network** : This is the core network of the VPN which interconnects PEs by using VPN tunnels. The provider builds and maintains these VPN tunnels. The provider backbone network can be operated based on several network protocols, such as IPv4, IPv6, or MPLS (Multiprotocol Label Switching).
- **Provider(P) Routers** : P routers are located inside the provider network. They are not directly interfacing to any customer network. Hence, P routers are neither VPN-aware nor do they maintain any VPN states. Their main role is to support the traffic routing and aggregation functions of the provider network.

- **The Control Protocol**: The control protocol is responsible for the VPN tunnel establishment, maintenance and auto discovery functions of the VPN.

Provider provisioned VPNs are becoming popular among industrial customers due to several reasons [4, 5, 48, 53–57]. Primarily, the customer can outsource WAN operational functions to the network service provider. Thus, the customer requires fewer technical staff to manage their VPN. Moreover, they do not need to acquire any expensive Customer Edge devices (CEs). Hence, it directly reduces the operational and capital costs of the customers. On the other hand, the customer has flexibility to obtain a required level of service from the provider based on SLAs (Service Level Agreements) and he can change these agreements according to the current traffic demands. Moreover, the customer can change levels of service without changing his own network infrastructure.

According to the operating layer of the OSI (Open System Interconnect) model, provider provisioned virtual networks can be divided into two categories; namely, Layer 3 Virtual Private Networks (L3VPNs) and Layer 2 Virtual Private Networks (L2VPNs). An L3VPN forwards packets based on layer 3 routing information such as the IP address, TCP (Transmission Control Protocol) / UDP (User Datagram Protocol) ports and a L2VPN forwards packets based on L2 routing information such as MAC (Media Access Control) address, VC (Virtual Circuit) identifiers, switch port information. Table 2 presents a comparison of the different PPVPNs.

Table 2. Properties comparison of L2VPNs vs L3VPNs.

	L2VPNs	L3VPNs
Tunnels Support	BGP or LDP based MPLS, IPsec, GRE	BGP or LDP based MPLS MPLS, IPsec, GRE
Customer Protocol Support	Multiple Protocols	Minimum
Scalability	High	High
Ubiquitous Availability	Medium	High
Providers Network Function	L2 connectivity	L3 packet routing
Implementation Cost	Low	High
Routing Protocol Section	By the Customer	By the Provider
Control: IP routing	Customer	Provider
Change: Adding Sites	Simple, only the service provider router at the connecting site needs to change	Complex.All service provider routers need routing changes
Change: IP changes	IP addressing is simpler and there is no need to involve service provider	Service provider needs to agree all IP address changes
Faults: Management	Service provider does not need to deal with customer routing issues, fewer fault calls, quicker and cheaper fault fixing	Very difficult to fix faults, costing time and money for service provider

A VPWS provides the point-to-point connectivity over the provider network. It encapsulates the customer data and delivers through the tunnels which are built on top of the provider network. VPWS uses a different variety of tunnels including IPsec, MPLS and GRE. From the customer point of view, the operation of a VPWS is similar to a leased line as VPWS provides the point-to-point tunnels between customer sites. Hence, VPWS is the ideal VPN technology to interconnect two sites which are geographically distributed and have unique traffic patterns. On the other hand, VPLS networks offer multipoint-to-multipoint communication over the provider network. It extends the Ethernet broadcast domain to multiple sites which are geographically dispersed across the globe. A VPLS requires mesh connectivity between PEs and it uses a different variety of tunnels including IPSec (IP Security), L2TPv3 (Layer 2 Tunneling Protocol Version 3), MPLS (Multiprotocol Label Switching) and GRE (Generic Routing Encapsulation).

2.1.4 Customer provisioned VPNs

The customer is responsible for the operation and management functions of Customer Provisioned VPNs (CPVPNs). Hence, the network service provider is unaware of the customerŠs VPN schemes, routing protocols and address spaces. The VPN exchanges information only between CEs. There are two commonly used tunneling protocols to implement a CPVPN, namely IPsec and GRE. Figure 3 illustrates the basic layout of a CPVPN architecture.

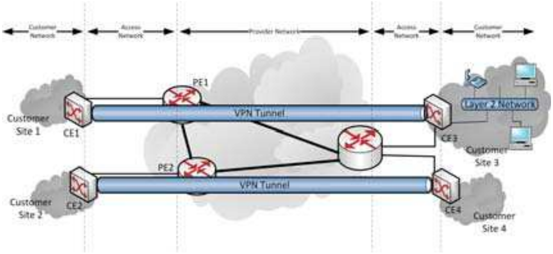


Fig. 3. A customer provisioned VPN.

Table 3 presents the comparison of the different CPVPN architectures.

Table 3. The properties comparison of CPVPNs.

	IPsec	GRE
Routing Configuration	Static Configuration	Dynamic Configuration, Based on the routing protocol
Resiliency	Low	High
Scalability	Low	High
Ubiquitous Availability	High	Low
Protocol Support	IP only	Any L3 Protocol, IP, IPx, OSPF, EIGRP
Scalability	Low	High
Broadcast Support	No	Yes
Security	High	Low

2.2 Virtual private LAN service

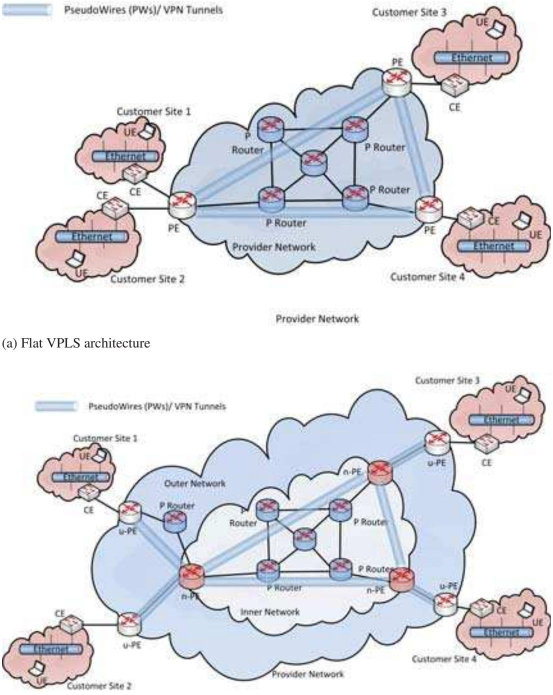
2.2.1 VPLS Categories

VPLS is a Layer 2 VPN (L2VPN) architecture which enables Ethernet based multipoint-to-multipoint communication between different sites. Thus, a VPLS allows sharing the same Ethernet broadcast domain over all the connected customer sites. VPLS end users experience the same LAN experience regardless of their locations since the provider enables rapid and flexible service provisioning by using the provider network.

There are two types of VPLS architectures available.

1. Flat VPLS architectures.
2. Hierarchical VPLS (H-VPLS) architectures.

Figure 4 illustrates the network topology of both flat and hierarchical VPLS networks.



(a) Flat VPLS architecture

(b) Hierarchical VPLS architecture ([17] ©2015 IEEE)

Fig. 4. Different VPLS architectures.

A VPLS has five main components namely, User Equipment (UEs), Customer Edge Equipment (CEs), Provider Edge Equipment (PEs), Provider (P) routers and the provider network. In contrast to flat-VPLS architectures, H-VPLS utilizes two types of PEs known as u-PE and n-PE [58]. u-PEs are the user facing PEs, while n-PEs are the network facing PEs. n-PEs contain all the intelligence of the VPLS architecture.

Specifically, they are responsible for packet forwarding, address learning and auto discovery functions. u-PEs play an aggregation role. A u-PE is attached to at least one n-PE and it forwards all the received customer frames to the attached n-PEs.

The VPLS architecture offers various benefits to both end users and service providers [2, 58–60]. Table 4 presents the key benefits of VPLS networks.

Table 4. Key benefits of VPLS networks.

Benefit	Description
L2 multipoint-to-multipoint connectivity	A VPLS can connect all the customer sites at the Ethernet level. It offers the opportunity to extended customersŠ Ethernet network across geographically distributed sites with flexible bandwidth and sophisticated SLAs (Service Level Agreements)..
High-speed connectivity	A VPLS networks offers a simple Ethernet interface for customers by simplifying the complexity of the access network and provider network. However, the service bandwidth is not tied to the physical interface and it depends only on SLAs.
Flexibility	The existence of VPLS is transparent to the customer network and its operation is independent of customer site protocols. Hence, the customer has the flexibility to use their own choice of data formats and routing protocols.
Unicast, multicast and broadcast support	In contrast to the other L2VPN solutions, a VPLS can be configured to provides any type of connectivity such as point-to-point, point-to-multipoint, multipoint-to-point and multipoint-to-multipoint. Therefore, a VPLS network supports any type of communication namely unicast, multicast, anycast and broadcast communication.
Low cost implementation	VPLS is a simpler and more cost effective solution than MPLS based VPNs.

2.2.2 The existing VPLS architectures

VPLS is a service provider provisioned L2VPN service. In provider provisioned VPNs, the service provider participates in the management and the provisioning functions of the VPN. The fundamental framework for a service provider provisioned L2VPN is defined by IETF in [61]. This framework standardizes protocols and mechanisms to support inter-operable L2VPNs. Augustyn and Serbest stated the basic implementation requirements of a provider provisioned L2VPNs in [62]. These requirements consist of topology generation, control signaling, security, redundancy and failure recovery.

Basically, a VPLS emulates a LAN and it requires full mesh connectivity. Initially, IETF defined two standard frameworks to design a VPLS network by using BGP [63] and LDP [64]. In [63], authors proposed a use-case for BGP to establish and maintain a full mesh of VPN tunnels between PEs in the VPLS. Here, BGP is used as the control protocol to provide the auto discovery and signaling functions. In [64], authors proposed to establish full mesh LDP sessions between PEs in the VPLS. Later, these LDP sessions were used to establish PWs and provide control signaling. A detailed analysis of the deployment and performance aspects of these frameworks was presented in [58]. A simplified version of VPLS is proposed as an IP-only LAN Service (IPLS) in [65]. The IPLS provides a VPLS-like service and is used exclusively for IP traffic only. All these architectures are flat VPLS architectures and they suffer from various scalability issues.

Hierarchical VPLS architectures are designed to overcome the inherent scalability issues of the flat VPLS architectures. The first functional hierarchical VPLS architecture was proposed in [64]. Some other research studies also focused on enhancing the features of H-VPLS networks [66–71].

The very first secure VPLS architecture was proposed as an HIP-enabled virtual private LAN Service (HIPLS) [72]. Here, the authors proposed a use-case of HIP [73] to provide a secure VPLS over an untrusted network. HIPLS provides the demanded level of security features such as authentication, payload encryption, secure control protocol and protection from IP based attacks.

Table 5 contains a comparison of different VPLS architectures.

Table 5. Properties comparison of different VPLS architectures([17] ©2015 IEEE).

	LDP based [64]	BGP based [63]	Hierarchical [63, 64]	IPLS [65]	HIPLS [72]
Scalability of Control Plane	Low	Low	High	Low	Low
Scalability of Forwarding Plane	High	High	High	High	Low
Scalability of Security Plane	-	-	-	-	Low
Data Traffic Encryption	No	No	No	No	Yes
IP Attack Protection	No	No	No	No	Yes
Control Protocol	LDP	BGP	LDP or BGP	LDP	HIP
Control Protocol Protection	No	No	No	No	Yes
Multiple Protocol Support	Yes	Yes	Yes	No. IP Only	Yes
Broadcast Support	Yes	Yes	Yes	Yes	No

2.3 Security protocols

2.3.1 IP security protocol

IP Security (IPsec) is a secure transport protocol that stands on top of IP to allow multiple hosts to communicate securely. It was originally standardized by the IEFT in the RFC (Request for Comments) 1825 [74] and RFC 1829 [75]. Furthermore, it was revised again in RFC 2401 [76] and RFC 2412 [77] by removing a number of incompatible aspects. An IPsec tunnel requires a mutual authentication between end nodes at the beginning of a communication session. A shared state must be maintained for a secure communication between communicating peers. Therefore, end nodes have to establish Security Associations (SA) between them. The IPsec protocol does not propose any specific secure mutual authentication and key exchange protocols. Hence, the Internet Key Exchange (IKE) protocol was initially proposed to establish this shared state in a dynamic fashion over a secure association by enabling confidentiality, data integrity, data source authentication, and access control. IKE mutually authenticates hosts and establishes an SA with a shared secret that can be used for the efficient implementation of an IPsec tunnel [78]. It was improved in IKEv2 [79]. Mobility and multihoming features were added in MOBIKE protocol [80].

Three modes of tunnel are defined by IPsec, namely: the transport mode, the tunnel mode and BEET (Bound End-to-End Tunnel) [81]. In the transport mode of operation, a new IPsec header is inserted between the IP header and the header of the higher layer, hence the original IP header remains unchanged. In the tunnel mode of operation, the entire IP packet is encapsulated in another IP datagram and a new outer IP address is added for routing purposes. Thus, there are two pair of addresses used in the tunnel mode of operation; an outer address pair for the wire and inner address pair for the application. Then, the IPsec header is inserted between these outer and inner IP headers. However, this inner address is fixed for the lifetime of an SA. The BEET mode was proposed to evade transmitting of the inner address pair as they are fixed.

IPsec supports two protocols, namely: the Authentication Header (AH) and Encapsulation Security Payload (ESP). IPSec offers integrity for the IP header and the payload with AH. It also provides user authentication. However, AH cannot encrypt the payload which is supported only by ESP. The latest ESP version has the ability to authenticate users the same was as AH. However, they are different in terms of the ability to provide authentication for the whole IP packet including both the header and payload. Although there is some IPsec software that solely relies on ESP authentication, most legacy software still uses the AH protocol.

Authentication header

AH supports two different modes; the transport mode and the tunnel mode. The tunnel mode AH appends a new IP header to the whole IP packet whereas the transport mode AH does not support this [82]. If the tunnel end-point is a gateway, the final destination could be a private network behind the gateway. At the packet gateway (tunnel ingress), in order to route the packets towards the final destination a new IP header is appended. Thus, the checksum cannot be verified, since the original IP header is modified. The transport mode is used mostly in end-to-end communication architectures, whereas AH is used to protect the integrity of a whole IP packet by calculating the checksum of a packet. Figure 5 illustrates the encryption in AH mode.

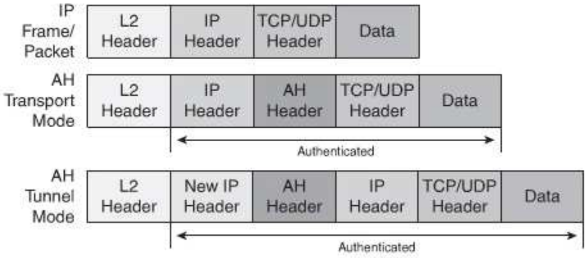


Fig. 5. Encryption in AH mode.

AH provides integrity combined with a Message Authentication Code (MAC) which is encoded with a keyed hash algorithm. The hash result is appended to the packet before it is sent to the recipient. The keyed hash algorithm produces the hash of the message and the secret key which is shared by both the sender and the receiver. On the arrival of a packet, the recipient regenerates a hash by using the same shared key and the message. The message integrity is confirmed by checking these hash values. The Security Parameters Index (SPI) is a unique identifier for a connection. It is used combined with the destination IP and IPsec protocol type to determine the Security Association (SA) being used. The sender and the recipient must have a single connection established in each direction and must assign an incrementing sequence number that provides defense against replay and denial of service attacks.

Encapsulation security payload

ESP provides integrity together with data origin authentication. Since, ESP is connectionless; integrity should be provided on a packet basis to ensure data origin authenticity, though, originally ESP was designed to enable confidentiality by encrypting the payload. ESP has several operational modes to support different combinations of services; confidentiality only, integrity only, and confidentiality and integrity [83]. ESP can either operate in tunnel or transport mode. Similarly to the AH tunnel mode, the ESP tunnel mode appends a new IP header, though the inner IP header stays unchanged and is the ultimate destination for the packet. In tunnel mode, all of the inner IP packets are secured by ESP, whereas in transport mode the ESP header is inserted after the IP header by leaving it unprotected. Figure 6 illustrates the encryption in ESP mode.

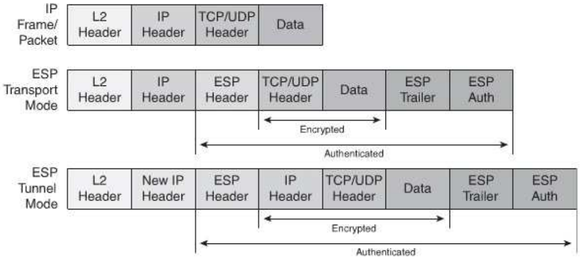


Fig. 6. Encryption in ESP mode.

ESP uses symmetric key cryptography to provide encryption with a shared key which is used by the sender and the receiver to encrypt and decrypt the messages. With ESP, the data encryption is provided with a block cipher that fragments data to be encrypted into smaller blocks. The payload of a packet contains the encrypted data and a unique Initialization Vector (IV). ESP uses padding due to three major reasons. First, to make the encrypted data an integral multiple of the fixed block sizes according to the encryption algorithm. Second, to ensure that the ESP trailer ends on a multiple of four bytes according to the specification and at last, to conceal the actual length of the payload.

2.3.2 Internet Key Exchange version 1 (IKE)

IKE version 1 has already depreciated with the introduction of version 2. IKEv1 exchange is defined in two phases where the secure association establishment is classified into the first phase and IPSec association establishment classified into a second phase [78]. IKE phase 1 has two modes of operation; the main mode and the aggressive mode. The establishment of IKE SA with three pairs of messages is considered as the main mode, whereas a non-reliable, but faster three message exchange is considered as the aggressive mode. IKE phase 2 has only one mode of operation which is known as the quick mode. It is an exchange of three messages to establish independent IPSec SAs that are unidirectional and protected with IKE SA.

2.3.3 Internet Key Exchange version 2 (IKEv2)

IKEv2 [79] is an extension to IKEv1 where the exchange is triggered by an initiator by sending a request message which is followed by a response message from the responder which has the responsibility to ensure the reliability of the message it carries. The initiator expects a response message from the responder within a time interval which is defined in the standard or resends the same request message or terminates the connection. The first two exchanges; IKE_ SA_ INIT and IKE_ AUTH negotiate cryptographic algorithms once and execute a Diffie-Hellman key exchange. The second IKE_ AUTH messages exchange certificates and identities. Figure 7 presents the IKE_ SA_ INIT and IKE_ AUTH messages and the content of the messages.

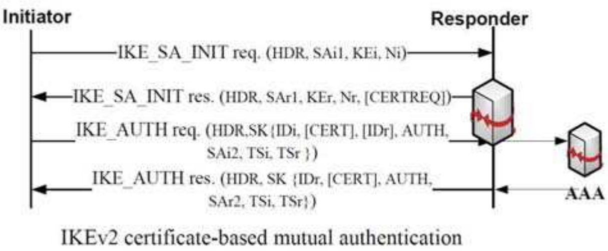

Initiator
Responder
IKE_SA_INIT req. (HDR, SAi1, KEi, Ni)
IKE_SA_INIT res. (HDR, SAr1, KEr, Nr, [CERTREQ])
IKE_AUTH req. (HDR,SK{IDi, [CERT], [IDr], AUTH, SAi2, TSi, TSr })
IKE_AUTH res. (HDR, SK {IDr, [CERT], AUTH, SAr2, TSi, TSr})
AAA
IKEv2 certificate-based mutual authentication

Fig. 7. IKEv2 based mutual authentication message exchange.

2.3.4 IKEv2 Mobility and Multihoming Protocol (MOBIKE)

IKEv2 Mobility and Multihoming Protocol (MOBIKE) [80] is an extension to IKEv2 that introduces mobility and multihoming features to legacy mobile devices. MOBIKE enables efficient management of IPsec and IKE security associations when a host is mobile and is attached to the Internet over multiple interfaces while at the same time the point of attachment to the Internet is changing. IKEv2 must negotiate its SAs after any change in the network topology, though this demands heavy processing. IKEv2 SAs are built over the IP addresses of the hosts. MOBIKE addresses the problem of changing IP addresses by suggesting it to work on top of IKEv2. However, MOBIKE is not a perfect mobility protocol since it does not support simultaneous movements or route optimization [84]. Mature mobility protocols, such as Mobile IP are being developed to address these issues.

MOBIKE solely focuses on what two peers need to agree on at the IKEv2 level and what is required for interoperability. MOBIKE uses the tunnel mode with static IP addresses on applications in order to maintain the associations even if a peer changes its location. Therefore, the already established IPsec tunnels might not detect any movement, since the IP addresses remain the same though tunnel headers may change according to mobility.

1. Inform peers of the peer address set and preferred address.
2. Detect loss of connection with the peers.
3. Inform the change of preferred address and the peer address set for the peers.
4. Provide support for NAT (Network Address Translation) enabled devices.

MOBIKE is heavily used in mobile applications with a break-before-make type of handover. Still, the protocol attempts to retain the same SAs that were established by the old point of attachment in order to prevent expensive renegotiations. MOBIKE also provides multihoming when the hosts have multiple interfaces. Though multihoming is not an option for mobility; it provides redundancy even if the primary association is broken. In a scenario where a device has multiple interfaces utilizing different technologies, such as WLAN (Wireless LAN), UMTS (Universal Mobile Telecommunications System) or Bluetooth, the MOBIKE SAs will maintain the same parameters, but with different tunnel headers when it connects through different interfaces.

2.3.5 Secure socket layer / Transport layer security

Secure Socket Layer (SSL [85] is a protocol developed by Netscape. Later, it was improved to its successor Transport Layer Security (TLS) [86] by IETF as a cryptographic protocol to provide security and data integrity. SSL/TLS can secure traffic over insecure networks such as public domains and the Internet. SSL/TLS is heavily used to secure data over the Internet for e-mail, web browsing, instant messaging, Voice-Over-IP (VOIP) and VPN. SSL is a combination of four other protocols; handshake protocol, record protocol, alert protocol, and change cipher suite protocol. The handshake protocol is responsible for negotiation of initial parameters. In order to initiate a session, the handshake protocol starts by exchanging hello messages to agree on algorithms, to exchange random values, and check for session resumption. When it is completed, the necessary cryptographic parameters and pre-master secrete are exchanged. The server and the client mutually authenticate by exchanging certificates after authenticating and agreeing on a secret which is the actual master secret that is generated from the pre-master secret and the random nonce. Followed by this, the handshake protocol will be terminated by passing the security parameters to the record protocol which verifies them. By the end of the successful connection set up and verification, a tunnel is established between the client and the server. The VPN tunnels provide security and data integrity for commonly used HyperText Transport Protocol (HTTP).

SSL/TLS provides VPN solutions, such as OpenVPN(Figure 8). SSL/TLS VPN software provides the same level of security as IPsec VPN software by means of Diffie-Hellman key exchange and ESP encapsulation [86]. SSL VPN can use as an alternative to IPsec because IPsec tunnels cause trouble with Network Address Translation (NAT) and firewall rules. SSL/TLS is designed to be transparent to higher layer protocols and to be compatible with popular web and e-commerce applications. SSL/TLS is a session based protocol that provides session oriented security between hosts, unlike permanent IPSec parameters. However, SSL/TLS lacks support for UDP traffic and requires a stateful connection.

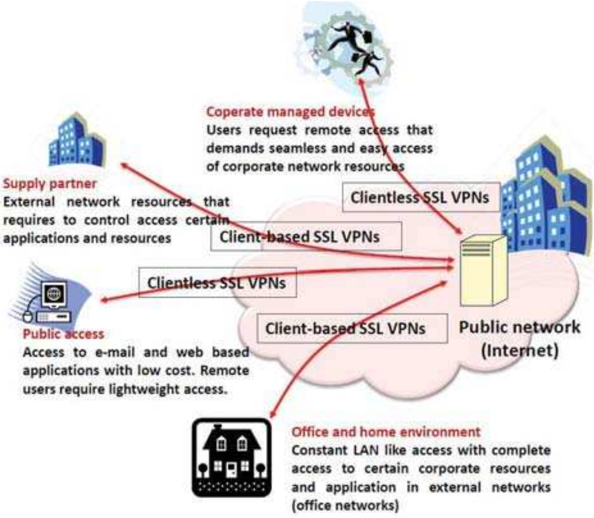

Coperate managed devices
Users request remote access that demands seamless and easy access of corporate network resources
Supply partner
External network resources that requires to control access certain applications and resources
Clientless SSL VPNs
Client-based SSL VPNs
Clientless SSL VPNs
Public access
Access to e-mail and web based applications with low cost. Remote users require lightweight access.
Client-based SSL VPNs
Public network (Internet)
Office and home environment
Constant LAN like access with complete access to certain corporate resources and application in external networks (office networks)

Fig. 8. SSL/TLS VPN scenarios.

2.3.6 Host identity protocol

Host Identity Protocol (HIP) is a novel security and mobility protocol which was standardized by IETF[87]. Generally, an IP address takes on a dual role as the locator and the host identity. The HIP separates the IP address by introducing a new namespace for the Host Identities (HIs). HIP introduces a new identity layer to the OSI model. Figure 9 illustrates the new HIP layer in a TCP/IP protocol architecture model.

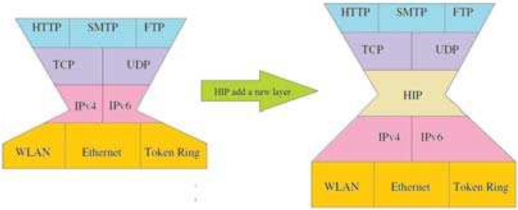


Fig. 9. New HIP layer in OSI model.

In this approach, the IP addresses are used to route packets where network sockets are bounded to Host Identifiers which are referring to individual hosts. An HI is the globally unique name of the user and it may have variable sizes. HIs are used only in the HIP handshake and a 128-bit cryptographic HI hash is used by applications. It is called the Host Identity Tag (HIT). The HIP Local Scope Identifier (LSI) is a 32-bit localized representation of HIT which serves as a bridge between HIs and IPv4 address space.

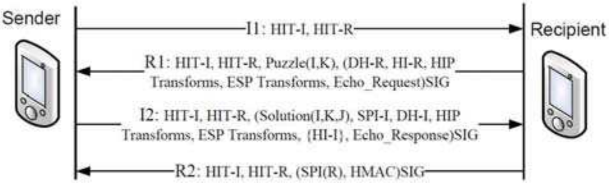

Sender
Recipient
I1: HIT-I, HIT-R
R1: HIT-I, HIT-R, Puzzle(I,K), (DH-R, HI-R, HIP Transforms, ESP Transforms, Echo_Request)SIG
I2: HIT-I, HIT-R, (Solution(I,K,J), SPI-I, DH-I, HIP Transforms, ESP Transforms, {HI-I}, Echo_Response)SIG
R2: HIT-I, HIT-R, (SPI(R), HMAC)SIG

Fig. 10. The HIP base exchange.

Base protocol or HIP Base Exchange (BEX) is the core of HIP (Figure 10). HIP BEX mutually authenticates Initiator (I) and Responder (R) based on cryptographic identities. It establishes Security Associations (SAs) for IPsec tunnels and exchanges keys by using a Diffie-Hellman (DH) key exchange procedure. Furthermore, HIP BEX includes a puzzle mechanism to prevent DoS (Denial of Service) attacks.

The I1 packet triggers the Base Exchange which may include a source IP/Host Identity Tag (HIT) and destination IP/HIT. The I1 will be replied by an R1 packet that includes a Diffie-Hellman (DH) key, cryptographic puzzle and the public key of the responder. The responder signs the R1 packet with its public key before it is sent. The cryptographic puzzle prevents DoS attacks on the responders. Upon receiving I2, the responder verifies the puzzle solution. If the solution is correct, it computes the session keys, decrypts HI-I, and verifies the signature on I2. Finally, responder sends an R2 that contains the Security Payload Index (SPI) for the SA between the initiator and responder, A Hash based Message Authentication Code (HMAC) is computed using the session key, and a signature [88]. HIP ensures that a host can be configured with multiple locators simultaneously rather than configuring several locators sequentially depreciating the previous one.

HIP uses the locator parameter to carry the information of additional locators for which a node or the preferred locator can be reached. To avoid conflicts, HIP recommends using separate ESP anti-replay windows for each interface or address to receive packets from the peer nodes when multiple locators are simultaneously used. When a mobile has more than a single locator, it must indicate the responder which is the most preferred locator. Combined mobility and multihoming approaches for future networks are expected to have the multiple interfaces for different technologies on a mobile device to enable technology convergence. Moreover, the support for multiple access technologies will improve Quality of Experience (QoE) [73].

Different HIP implementations

Two main open source HIP implementations are available, namely, HIPL (HIP Linux) and OpenHIP. These implementations have been very useful in many research projects and industrial applications. A comparison of both HIP implementations is presented in the Table 6.

Table 6. **Comparing HIPL and OpenHIP implementations([29] ©2015 IEEE).**

	HIPL	OpenHIP
Compatible platforms	Linux, Android	Linux, Mac OS X, Windows XP, DEB, RPM
Software dependencies	aptitude, hipl-firewall, hipl-dnsproxy	openssl, libxml2, libipsec (kernel mode only)
Supported IP address standards(on Linux)	IPv4, IPv6	IPv4
Length of HIT	128-bits	128-bits
Support for multiple Host Identities	Yes	Yes

– **HIPL implementation**

In 2004, the Finnish Funding Agency for Technology and Innovation (Tekes) funded the InfraHIP project; a project aimed at developing the infrastructure pieces to encourage the widespread use of HIP solutions [89]. Such infrastructures include DNS, NAT, firewall support, APIs, OS security, rendezvous service, multiple end-points within a single host, process migrations, and other issues related to enterprise-level solutions.

The InfraHIP project developed a Linux-based open-source HIP solution called HIPL. This open-source HIP solution is well designed to address the security challenges of both present and future network standards, and is hence able to support both IPv4 and IPv6 applications and allows for interoperability between the two IP versions. HIPL not only develops the listed infrastructure pieces, it also 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.

– **OpenHIP implementation**

In 2005, the OpenHIP project group released its first open source software implementing the experimental HIP solution. OpenHIP is also a Linux based piece of software. The OpenHIP implementation has been used by developers and networking firms to experiment with the HIP solution. At present, two major SCADA network appliance developing companies have already started to develop HIPLS based security solutions through the OpenHIP project [90].

The OpenHIP project aims to develop reference implementation of HIP for various platforms. OpenHIP is a free open-source HIP implementation developed by the Internet Engineering Task Force(IETF) and the Internet Research Task Force(IRTF) and licensed by MIT/Expat license to provide researchers with an experimental platform to study HIP and other related protocols [90].

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

Industrial use of different implementations

HIP is becoming popular in many industrial applications. These applications have spread from mobile applications to large scale cloud systems. Earlier in April 2015, the IETF put up two RFCs, RFC 7401[91] and RFC 7402[92]. RFC 7401 specifies a second version of HIP (HIPv2) which will be an improvement of the earlier version defined by RFC 5201[87],it mainly sets out to address the limitations of the previous version as well as the issues raised by the Internet Engineering Steering Group (IESG), such as crypto agility, hence ensuring a more robust authentication mechanism. RFC 7402 specifies an Encapsulating Security Payload (ESP) transport format for HIP. Initial HIP implementation uses ESP only for HIP base exchanges i.e. when setting up an 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. This specification is intended to be an integral part of the HIPv2 [91] [92].

VPLS based security solutions are becoming popular in industrial networks as a cost effective, high speed and multipoint connectivity model. Henderson et al. proposed an HIP based VPLS (HIPLS) architecture to secure a VPLS network. An HIPLS architecture is implemented by using an OpenHIP implementation. The HIPLS architecture is currently used in many industrial networks. Boeing, which is one of the largest airline manufactures in the world, is using an HIPLS based VPLS network in the assembly line for Boeing 777 airplanes [93]. General Electronics (GE) is working on an OpenHIP based security solution to secure inter-train communication[94]. 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 an OpenHIP implementation was released in 2009 by Canadian company Byres Security. They developed an HIPLS based end-box product called Tofino Endboxes (TEBs) to establish a secure VPLS network for industrial process control systems [95]. In 2011, the US based networks company, Asguard Networks, also released HIP enabled network appliances called Industrial Security Appliances (ISAs) based on the HIPLS architecture by using an OpenHIP implementation [96]. In 2013, Tempered Networks acquired Asguard Networks and Tempered Networks is now offering a wide range of HIPLS based network appliances called HIPswitches[97].

The first mobile implementation of HIP was developed based on an HIPL implementation. In 2013, basic support for Android was included in the HIPL source code [89]. Many features have been enabled in this mobile implementation over the past two years. For instance, HIP based firewall functionality for Android has been enabled since late 2014. An HIPL implementation is also used to secure cloud deployment in CERN. An HIPL based security solution is utilized in cloud deployment that is used for analyzing CMS (Compact Muon Solenoid) data from CERN [98]. The HIPL implementation provides secure connectivity and connection management capabilities for OpenStack based cloud systems.

2.4 Emerging technologies

2.4.1 Software-defined networking

During the past few years, Software-Defined Networking (SDN) has emerged as a new intelligent architecture for network programmability. The primary idea behind SDN is to move the control plane from switches and enable external control of data through a logical software entity called the controller. SDN offers simple abstractions to describe the components, the functions they provide, and the protocol to manage the forwarding plane from a remote controller via a secure channel. This abstraction captures the common requirements of forwarding tables for the majority of switches and their flow tables. This centralized up-to-date view makes the controller suitable for performing network management functions while allowing easy modification of the network behavior through the centralized control plane [36].

An SDN network contains three planes, namely 1) the application plane, 2) the control plane and 3) the data plane. Figure 11 depicts the overall SDN architecture.

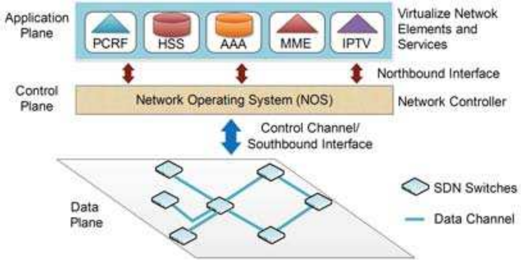


Fig. 11. The SDN architecture.

The data plane

The SDN concept separates the control plane from the data plane of the network. It pushes the network intelligence to a centralized controller. Thus, the data plane now consists of low-end switches and network links between them. Base stations, wireless access points and the Internet are connected to these DP switches. The user traffic is transported through the data plane. This communication channel is called the **data channel**.

The control plane

The control plane consists of a logically centralized controller which provides the consolidated control functionalities. Basically, the centralized controller supervises the packet forwarding functions of the network through an open interface. Moreover, it controls all the mobile backhaul functionalities such as routing, session initiations, session terminations and billing functions. The communication channel between the controller and DP switches is called the **control channel**. This control channel is implemented by using control protocols. For instance, the OpenFlow (OF) protocol is a widely used control protocol in the SDN domain [99].

The application plane

The application plane consists of the end-user business applications and other control entities. Legacy network control devices such as AAA (Authentication Authorization and Accounting) are now software applications which run on top of the Network Operating System (NOS)[100] in the application plane. Apart from that, it is possible to implement various other business applications which consume SDN communications services. In the SDN architecture, the underlying network infrastructure is abstracted from the applications. The boundary between the application and control layers is traversed by the northbound API (Application Programmable Interface). Therefore, each operator can develop his own networking applications and control entities. This helps to address specific needs of subscribers and optimizes network resources to achieve better performance. Moreover, SDN architecture further allows deploying the control plane entities on an operator cloud for more computing resources.

SDN makes it possible to manage the entire network through intelligent orchestration and provisioning systems. Thus it allows on-demand resource allocation, self-service provisioning, truly virtualized networking, and secure cloud services. Thus, a static network can evolve into an extensible vendor-independent service delivery platform, capable of responding rapidly to changing business, end-user, and market needs, which greatly simplifies the network design and operation. Consequently, the devices themselves no longer need to understand and process thousands of protocol standards but merely accept instructions from the SDN controllers.

2.4.2 Network function virtualization

One of the most interesting complementary technologies of SDN, which is virtualizing many network functions, is the so called Network Function Virtualization (NFV). The aim of NFV is to virtualize (known also as network softwarization) a set of network functions by deploying them into software packages, which can be assembled and chained to create the same services provided legacy networks [36]. The concept of NFV is inherited from the classical server virtualization by installing multiple virtual machines running different operating systems, software and processes [101, 102].

Traditionally, network operators had always preferred to use dedicated highly available black-box network equipment to deploy their networks. However, this old approach inevitably leads to long time-to-market (CapEx) periods and requires a competitive staff (OpEx) to deploy and run them. NFV technology aims to build an end-to-end infrastructure and enable the consolidation of many heterogeneous network devices by moving network functions from dedicated hardware onto general purpose computing/storage platforms such as servers. The network functions are implemented in software packages that can be deployed in a virtualized infrastructure, which will allow for new flexibilities in operating and managing mobile networks [36, 103, 104].

Another important advantage of implementing NFV in cloud infrastructures is resilience. Implementing network functions in data centers allows transparent migration between either virtual machines or real machines. Furthermore, implementing network functions in data centers will enable more flexibility in terms of resource management, assignment, and scaling. This will impact the development of eco-systems and energy efficiency of networks, as over-provisioning can be avoided by only using the necessary amount of resources [36, 104].

3 A scalable and secure flat-VPLS architecture

In this chapter, we present our proposal to design a scalable secure flat-VPLS architecture based on the Host Identity Protocol (HIP). We explain our proposed session key-based security mechanism and the efficient broadcast mechanism that increases the forwarding and security plane scalability of VPLS networks. The security features of the proposed architecture are also analyzed. Finally, simulation results are provided to verify the performance of the proposed architecture and to confirm the security features.

3.1 Related work

A framework for provider provisioned L2VPNs was presented in [61]. It discussed a number of different technical approaches to implementing L2VPNs, namely Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS) and IP-Only LAN-Like Service (IPLS). Furthermore, it revealed the correlation between different approaches and mentioned the issues associated with each approach. Service requirements for the provider provisioned L2VPNs were presented in [62]. Detailed requirements such topology generation, control signaling, control and data plane security, membership discovery, path management, redundancy and failure recovery were expressed from both the customer and service provider perspective.

There are many research studies focused on VPWS. Martini et al. specify a protocol for establishing and maintaining pseudowires by using a Label Distribution Protocol (LDP) [105]. The encapsulation methods for carrying Ethernet/802.3 Protocol Data Units (PDUs) over Multiprotocol Label Switching (MPLS) networks are discussed in [106] and the encapsulation methods for carrying Point to Point Protocol (PPP) or High-Level Data Link Control (HDLC) PDUs over a MPLS are presented in [107]. Y. Stein further proposed an extension to the MPLS pseudowire format called PWsec to enhance pseudowire user plane security [108].

However, a VPLS is more complex to implement than a VPWS as it provides multipoint to multipoint connectivity. Hence, VPLS requires full mesh connectivity between PEs. Initially, IETF proposed two methods in order to implement a full mesh establishment for a VPLS by using Border Gateway Protocol (BGP) [63] and Label Distribution Protocol (LDP) [64]. In [63], the authors proposed a VPLS architecture based on BGP. It uses BGP as the control plane protocol to provide the auto discovery and signaling for PEs. Each PE simultaneously discovers all the other PEs in the same VPN through the use of BGP and establishes a full mesh of pseudo-wires between these PEs. A VPLS architecture based on LDP was proposed in [64]. A full mesh of LDP sessions is established between PEs which belong to the same VPN. Thereafter, these LDP sessions are used to establish a full mesh of pseudo-wires between the PEs. Furthermore, the authors proposed a Hierarchical VPLS (H-VPLS) architecture to reduce the number of LDP sessions in the provider network. An analysis of LDP based and BGP based VPLS solutions was presented in [58]. Further, the authors discussed the advantages and disadvantages of each VPLS architecture.

A simplified version of VPLS was proposed as an IP-only LAN Service (IPLS) in [65]. IPLS provides a VPLS-like service and is used exclusively for IP traffic only. Furthermore, the authors explained the possible simplifications to the operation of the general VPLS. However, none of these architectures are able to provide a sufficient level of security features for the VPLS network. Specially, the control protocols of these systems are vulnerable to IP/TCP based attacks such as Denial of Service (DoS) and TCP reset attacks.

The first secure VPLS architecture was proposed as an HIP-enabled virtual private LAN Service (HIPLS). In [72], Henderson et al. described a use case of HIP which established a secure VPLS network over an untrusted network. They used an HIP protocol to build secure tunnels between PEs. This provides a secure VPLS in terms of data encapsulation, strong authentication, secure control protocol and some level of robustness to possible known attacks.

HIP based secure VPLS architectures are becoming popular among many industrial enterprises. For instance, Boeing is using HIPLS based VPLS networks in the assembly line for Boeing 777 airplanes [109]. On the other hand, two major SCADA network appliance developing companies have already started to develop HIPLS based security solutions. They used the OpenHIP project [90] which is an open source software implementation of HIP.

In 2009, a Canadian company, Byres Security, released the first HIPLS based endbox product called Tofino Endboxes (TEBs) to establish secure VPLS for SCADA and industrial process control systems [110]. However, TEBs are lacking in graphical user interface for the end-box configuration and maintenance. Thus, the configuration and maintenance procedures of TEBs are complex and time consuming tasks.

Later, Tempered Networks released the second HIPLS based network solution called HIPswitches [111]. HIPswitches provide a web based graphical user interface for the device configuration and maintenance. Thus, the configuration process of HIPswitches is simple and user friendly. Moreover, HIPswitches support inbuilt wireless connectivity as well. Tempered HIP switches are used in various use cases such as oil, gas, telecoms and production facilities [111].

3.2 Security considerations of the VPLS

A VPLS network is vulnerable to a number of security breaches and they can strain network resources, such as memory space, forwarding information tables, bandwidth and CPU processing. Hence, a VPLS architecture should be supported by a range of security features such as mutual authentication, PE authorization, data and control frame encryption, privacy protection, secure address learning and control protocols, and robustness to known attacks.

Each PE should be authenticated with other peer PEs by using some sort of cryptographic authentication procedure during the auto discovery procedures and data exchanges. Otherwise, L2VPN traffic may direct to a wrong location and malicious users can mount attacks, such as perform Denial of Service (DoS) attacks.

Furthermore, the control protocol which is used for setting up pseudowires (PWs) should be secured from knows attacks. If the control protocol uses TCP/UDP messages, it may be advisable to have some sort of protection against TCP/UDP based attacks (e.g. UDP DoS, TCP reset, TCP DoS etc.). If a PE is unable to handle high volumes of multicast or broadcast traffic efficiently for a sustained period, then it is possible to launch a DoS attack by sending a large amount of broadcast/multicast frames to a PE. These frames may have either a multicast address or an unknown MAC (Media Access Control) address in their MAC destination address fields. In these cases, the PE will not be able to process all the bogus frames and ultimately it will be unable to serve the legitimate frames as well.

If VPLS data packets are transmitted in a clear (unencrypted) form via an untrusted public network, then a man-in-the-middle can eavesdrop and may be able to inject packets into the data stream. If control packets are maliciously and undetectably altered while in flight, DoS attacks or alteration of the expected QoS level may result. Hence, it is important to provide a sufficient level of security features for a VPLS network for a proper operation.

3.2.1 HIP based virtual private LAN service

The very first secure VPLS architecture was proposed as an HIPLS (HIP based Virtual Private LAN Service) [72]. HIPLS uses HIP to create a VPLS overlay on top of a standard IPv4 and/or IPv6 provider network. HIP signaling between PE devices works as the control protocol of the VPLS which is used to establish and maintain HIP tunnels between PEs.

This application of HIP for HIPLS differs from the traditional implementation of HIP. There are two main differences. First, HIP is implemented within the middle boxes as a Şbump-in-the-wireŤ implementation not in the end hosts. Secondly, the payloads of the ESP-encrypted datagrams are layer-2 frames instead of transport Protocol Data Units (PDUs) [72]. Figure 12 illustrates the network topology and protocol stack of a simple HIPLS network.

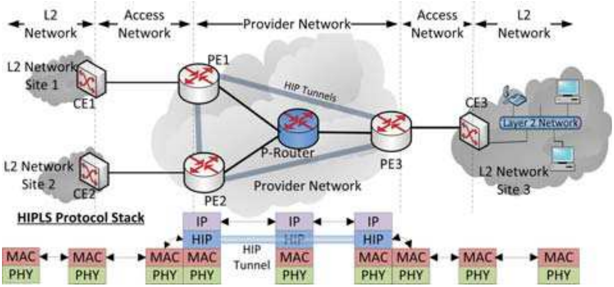


Fig. 12. The network topology of a HIPLS architecture.

HIPLS inherits the strong security properties found in HIP, including a strong authentication mechanism, data packet encapsulation and some level of robustness from attacks. The strong authentication mechanism and data packet encapsulation solve many of the above mentioned security threats in a VPLS. It provides secure PE registration, secure data/control packet exchange, a secure auto discovery mechanism, a secure control protocol and avoids several known attacks (e.g. eavesdropping, spoofing and DoS attacks).

3.2.2 The issues related to HIPLS

Although HIPLS provides a secure VPLS service, several issues can still be identified.

First, it has a massive key storage complexity. Every piece of PE equipment has to store $O(N)$ keys where N is the number of the PE equipment. As a result, the whole network has to store $O(N(N+1))$ keys. This is a serious issue as it reduces the available memory of a PE for other important functions such as routing tables. Furthermore, the extensive key searches need extra processing power at the PE and increase the packet transmission delay

Second, HIPLS has an inefficient broadcast mechanism. It performs $O(N)$ encryptions per broadcast frame at entry PE by using $O(N)$ different keys. Hence, it requires extensive processing at PEs and vulnerable to broadcast based DoS attacks.

Third, it is not possible to integrate the distribution of multicast or broadcast frames of HIPLS with the packet distribution mechanisms, such as spanning trees and multicast trees of the underlay network. Hence, the routing of these multicast and broadcast frames may not be optimal and network bandwidth is used inefficiently. Due to the above stated three reasons, HIPLS lacks both security and scalability of forwarding planes for implementation in large scale networks.

Fourth, HIPLS is vulnerable to misfeasor attacks which are originated from CEs and PEs as HIPLS do not specify any mechanism to avoid such attacks. A misfeasor is a system user who accesses unauthorized data, programs, or resources.

3.3 Proposed secure flat-VPLS architecture

We propose a novel secure VPLS architecture in this chapter. Our architecture is inspired by the HIPLS [72] architecture and uses HIP to create a VPLS overlay on top of a standard IPv4 and/or IPv6 provider network. Since most of the secure VPLS networks and industrial products are built based on HIP (by using HIPLS architecture), we also focused on improving the performance of HIPLS architecture. The main idea behind our proposal is to use a new session key mechanism with a customized version HIP. Hence, we can call our architecture as Session key based HIPLS (S-HIPLS).

Our architecture uses two types of keys: Content Encryption Keys (CEK) and Key Encryption Keys (KEK). The CEK is a symmetric key which is used to encrypt and decrypt all messages in a single VPN. Every PE has a unique KEK which is used to encrypt and decrypt the corresponding CEKs. There is a Key Distribution Center (KDC) which is the responsible entity to distribute CEKs to the PEs. Furthermore, KDC also works as the Authentication Server (AS) which maintains the Access Control Lists (ACL) of VPNs. Operators update ACLs according to the CE subscriptions. Thereafter, KDC uses these ACLs to grant the access to new PEs. HIP signaling is used as the control protocol to build and maintain the tunnels between PEs.

Our VPLS architecture can be categorized into four sub sections: PE registration and deletion, data communication, control protocol and key management.

3.3.1 PE registration and deletion

PE registration is the initial procedure for a new PE who wishes to join for the VPLS. The potential PE needs to send a request to the KDC to gain access to the VPNs. Hence, the new user has to establish an HIP tunnel with the KDC. A mandatory HIP BEX will run between the new PE and the KDC to establish this HIP tunnel. During this HIP BEX, a new PE is authenticated by a public key infrastructure (PKI) and authorized according to ACLs.

A few modifications are proposed to the original HIP Base Exchange (BEX) which was proposed in [87]. Figure 13 illustrates the modified BEX.

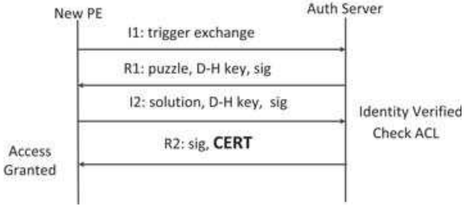


Fig. 13. Modified HIP BEX for PE registration([15] ©2013 Wiley).

The first three message exchanges are similar to the original HIP BEX. Then a HIT based authorization is sufficient enough to avoid spoofing attacks. Even if an attacker is able to generate a valid HIT, it would fail to complete the initial BEX due to lack of knowledge of the private key [112]. Additionally, a trusted third party certificates can be included in I2 for further verification of the HI. The symmetric key which is shared by Diffie–Hellman (D-H) key exchange is used as the KEK for this user.

The identity of the initiator is verified after the arrival of the I2 packet. Then KDC checks the ACL and sends a certificate in R2 which is encrypted by using the KEK key of the user. This certificate contains important information, such as the authorized VPNs, QoS policies and other VPN management details. This certificate is also important to avoid misfeasor attacks which originate from the legitimate CEs and PEs. For instance, a customer may subscribe to a VPLS service to obtain the connectivity for certain sites. However, if the customer tries to direct the traffic to a non-subscribed site via this VPLS, PEs can discard such traffic based on the information in these certificates.

The removal of inactive PEs is also important for the efficient operation and the security of the VPLS network. When a PE leaves or becomes inactive, KDC removes the KEK of that PE and no longer provides CEKs for that PE. An inactive user can be identified by using either active notification or passive notification. The PEs will actively notify their departure to the KDC before they leave or failure to acknowledge for the periodic CEK will passively notify the in-activity.

Provider VPNs

Similar to other provider networks, such as mobile networks and Internet Service Providers(ISPs), we propose to use provider VPNs to support traffic clarification in H-HIPLS. In provider networks, the service provider maintains a small set of VPN classes (provider VPNs) based on different service factors, such as supported data rates, traffic priorities and QoS classes [113]. According to the Service Level Agreements (SLAs) with each customer, the provider classifies the customer traffic into corresponding provider VPNs.

However, the number of VPNs used in the provider network is very limited. For instance, a mobile backhaul network supports 3-5 provider VPNs [113]. In industrial networks, most of the provider networks support only one provider VPN [111, 113]. However, the service provider has the flexibility to define the number of provider VPNs in the VPLS network. For instance, the control VPN of H-HIPLS is a provider VPN which transport control and signaling data.

3.3.2 Data communication

Our VPLS architecture uses HIP tunnels between PE to communicate. Hence, the sender PE must build an HIP tunnel to each target PE before any data exchange takes place. HIP uses BEET mode IPsec tunnels and all the frames are encrypted to avoid eavesdropping by the unauthorized party. The source PE encrypts L2 frames using the corresponding CEK of the VPN. Then, the frames will be wrapped within the ESP payload and sent to the remote PE. The remote PE detunnels them and places on the remote access network segment again as L2 frames.

3.3.3 Control protocol

The control protocol is used to maintain the proper operation of the VPLS network. It is responsible for three main functions, namely tunnel establishment, address learning and key distribution. Our architecture provides a secure control protocol by using secure HIP signaling.

Tunnel establishment

An HIP tunnel establishment between two users is mandatory before any type of communication in the VPLS. The HIP tunnel establishment follows a HIP BEX procedure and it mutually authenticates the users. This HIP BEX is slightly different from the original HIP BEX[87]. As we use CSKs for data encryption for ESP, D-H key exchange is omitted here. Hence, this modified HIP BEX is slightly faster and it needs less processing than the original HIP BEX. Figure 14 illustrates the modified BEX for tunnel establishment.

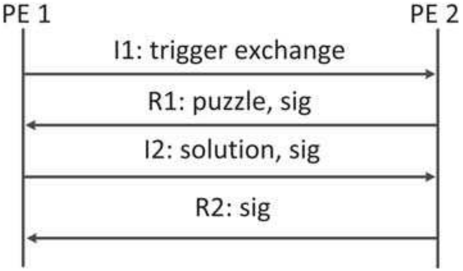


Fig. 14. Modified HIP BEX for tunnel establishment([15] ©2013 Wiley).

Address learning

VPLS is a L2VPN solution and frames are delivered based on MAC addresses. Hence, PEs should learn the MAC addresses of remote CEs and the network address of the PE which is responsible for each CE. Hence, each PE maintains a dynamic MAC-PE mapping table. The address learning procedure can be described as follows.

When a PE receives a frame from a CE access network, the PE learns that it is the responsible PE device for the source MAC address of the frame and records the entry in the MAC-PE mapping table. If the PE does not have the destination MAC address of the frame on its MAC-PE mapping table, the PE has to learn about the responsible PE. Hence, the PE broadcasts an encrypted address request frame to all PEs which belongs to the same VPN. Then the responsible PE sends a unicast frame as a reply and the PE updates its MAC-PE mapping table.

Key distribution

Key distribution is the most important process in our architecture. We present a secure key distribution to deliver the keys only to authorized PEs. Otherwise, fault traffic generation and unauthorized eavesdropping may be possible. Although key distribution is a duty of the control protocol, we describe it under the key management section.

3.3.4 Key management

The key management procedure of our VPLS architecture has three sections: KEK distribution, CEK distribution and KDC architecture.

KEK distribution

Each PE shares a unique symmetric key with the KDC which is used as the KEK for that PE. This KEK is shared using D-H key exchange during the PE registration process. It is used by the KDC to send the encrypted CEKs, certificates and any other control information to the PE. There is a periodic HIP BEX instance between each PE and KDC to renew the KEK. This process is called as rekeying and it is recommended by the HIP architecture [87]. Initially, we define the rekeying timeout as 10 s and it can be change according to the requirements of the operator.

CEK distribution

The KDC is the responsible entity for the CEK distribution. It periodically sends the CEK to PEs. These CEKs are encrypted by using the KEK of each PE. Hence, an eavesdropper cannot extract the CEK. There are three instances that a new CEK distribution is triggered.

First, the KDC periodically generates new CEKs to secure the VPLS communication session. Although an intruder may be able to capture a CEK somehow, it will not be valid after the rekeying time out. As the membership of the VPLS is dynamic, it may need to provide forward and backward confidentiality for the VPLS. Second and third CEK distribution events are used to ensure these optional security features. The KDC generates new CEKs after a new PE registration to protect the backward confidentiality. Backward confidentiality means new members cannot access old data. The KDC generates the CEKs for the VPNs which have the new PE as a member. Thereafter, it sends new CEKs to all the corresponding PEs.

The KDC generates new CEKs after a PE deletion to ensure the forward confidentiality. Forward confidentiality means an old member cannot access the new data of the VPN. It generates the CEKs of the VPNs which had the inactive PE as a member. Then, the KDC sends new CEKs to the corresponding PEs.

KDC architecture

The KDC is the heart of the key distribution mechanism. It plays a dual role as the key distributor and the authentication server. Hence, it is responsible for PE registration, PE deletion, and CEK generation and distribution. We propose a distributed KDC structure which is preferred for large scale networks. A distributed KDC architecture provides several benefits, such as reducing the work load, distributing the key storage complexity and avoiding single points of failure. We propose two distributed KDC architectures: a hierarchical server structure and a mesh server structure.

A hierarchical server structure is suitable for a provider network which has a hierarchical underlay network topology. Figure 15 illustrates a simple hierarchical KDC structure with two tiers.

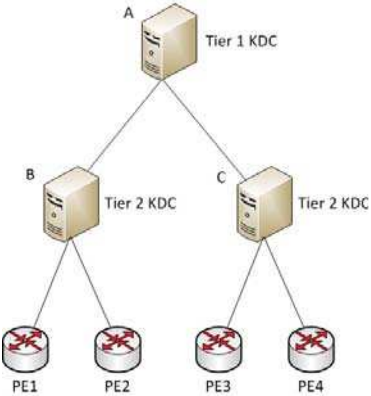


Fig. 15. A simple hierarchical KDC structure([15] ©2013 Wiley).

Each Tier 2 KDC is responsible for a section of the VPLS. It conducts the PE registration, KEK storage, CEK distribution and PE deletion functions of the particular section. For instance, B is responsible for PE1 and PE2. In addition, these Tier 2 KDCs can operate local VPNs if all the member PEs of the local VPN reside under that KDC. For example, B can operate a local VPLS only for PE1 and PE2. The Tier 1

KDC is responsible for CEK distribution (only VPNs spend at least two Tier 2 KDCs) and controls the operation of the Tier 2 KDCs. The Tier 1 KDC learns the relevant membership alternation by advisement from the Tier 2 KDC. Furthermore, the operator changes the ACL only in a Tier 1 KDC (AS) and it securely broadcasts them to Tier 2 KDCs.

The mesh structure is suitable for a provider network which has mesh underlay network. Figure 16 illustrates a simple mesh structure with two KDCs.

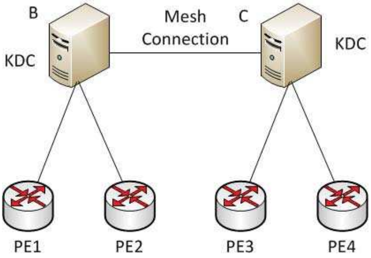


Fig. 16. A simple mesh KDC structure([15] ©2013 Wiley).

Each KDC is responsible for a section of the VPLS and all KDCs are connected to each other by forming a mesh network. Local VPNs are possible in this scenario as well. If it is necessary, the KDC has to notify regional membership changes to other KDCs. If new CSKs are required, the responsible KDC generates the new CEKs and securely transfer them to the corresponding PEs via other KDCs. Furthermore, one KDC in the mesh is elected for periodic CEK distribution.

3.3.5 Broadcast mechanism

Broadcast and multicast messages are important for a LAN service. For instance, broadcast messages are used by network protocols such as Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP) and multicast messages are used by Ethernet services like IP telephony, or television webcast.

However, HIPLS [72] does not provide an efficient broadcast/multicast mechanism and HIPLS is useful only for a constrained system like a unicast-only IPLS VPN. Hence, Henderson et al. proposed to deploy HIPLS only in systems which do not support multicast or broadcast, or bridge PDUs [72].

Our architecture provides an efficient broadcast mechanism and relaxes the unicast-only IPLS VPN constraint. We significantly reduce the number of encryptions and packet generation per broadcast frame at the entry PE. Our VPLS solution uses one session key for one VPN. Hence, it needs only one encryption per broadcast frame at the entry PE and only one encrypted broadcast frame is developed. Therefore, this packet can be replicated at the entry PE and/or at the immediate nodes according to the Spanning Tree Protocol (STP) or multicast tree. Thus, this broadcast frame distribution is exactly similar to the other non secure VPLS architectures [63] [64]. The proposed architecture is capable enough to distribute broadcast/multicast frames efficiently in the network.

3.4 Simulation results

This section contains the description of the simulation model and its most important results.

3.4.1 Security performance

We modeled and simulated our architecture in OMNET++ simulator [114] to compare the performance with other VPLS architectures. We especially conducted several extended simulations to study the performance under known attacks.

The control protocol is the vein of VPLS. It is designed based on BGP [115] or LDP [116] for most of the existing VPLS solutions (see Section 3.1). LDP based VPLS architecture proposed a full mesh of LDP sessions as the control protocol. According to the LDP specification [116], LDP sessions are built using the TCP protocol. BGP based VPLS architecture proposed to use BGP as the control plane protocol and it also uses TCP as its transport protocol [115]. However, these TCP based control protocols are vulnerable to IP/TCP based attacks. Attackers may try to perform TCP based attacks such as TCP SYN (Synchronization) DoS, TCP SYN DDoS (Distributed DoS) and TCP reset attacks to turn off the controlling plane of the VPLS network.

Hence, we compared the robustness of the control protocol of above architectures with our VPLS architecture under several known attacks. Namely TCP reset, TCP SYN DoS and TCP SYN DDoS attacks. We used the LDP based VPLS model which was presented in [64] and HIPLS [72] as the reference models to compare the performance of the proposed architecture.

Performance under TCP SYN DoS attack

A TCP SYN DoS attack is also called a TCP SYN flood attack. Basically, an attacker sends a succession of TCP SYN requests to the target system or server. As the server allocates some resources (A TCP port) for each successfully received SYN request, the attacker is able to consume server resources to make the system unresponsive for legitimate traffic.

Our simulation model contained a VPLS network with 300 nodes. We assumed that all the nodes are equivalent and the bandwidth of the network is set to 100 Mbps. We added an attacker who tries to attack a node in the network. Hence, it sends TCP SYN packets to the node under attack to occupy all the ports and IP address combinations with other nodes in the network. Then, the rest of the nodes in the network will not be able to send any data traffic for the attacked node as it does not have any port to make a connection. We assumed that the attacker also has the same bandwidth which is 100 Mbps. According to the values which were presented in [117], we used the number of ports per user as 64000 and the TCP timeout value as 270 s. We ran the simulation for 500 s and the attacks were placed between 25 s - 75 s time intervals. Figure 17 illustrates the percentage packet drop at the under attacked user over the simulation time.

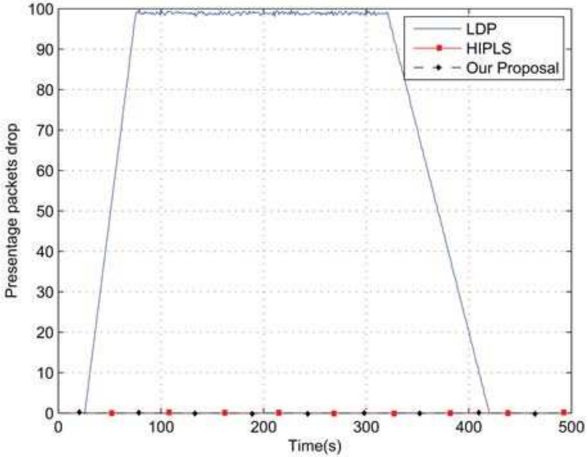


Fig. 17. Impact of a TCP SYN DoS attack([15] ©2013 Wiley).

The simulation result clearly indicates that there is no packet drop during the attack period for the HIPLS and our architecture. They have the same performance for both attacking and non-attacking periods. However, LDP architecture has dropped almost all the packets during the DoS attack period and it requires at least a TCP timeout period which is 270 s to fully recover from the attack. Hence, our architecture is capable enough to provide protection for a TCP SYN DoS attack.

Performance under TCP SYN DDoS attack

We further investigated the performance of the proposed architecture under a coordinated DDoS attack scenario. We used the same simulation setup and the attack model which was used for the previous TCP SYN DoS attack scenario. However, we increased the number of attackers and investigated the time required to successfully attack a single user in the VPLS network. We compared the performance with an LDP based VPLS architecture [64]. Figure 18 illustrates the average time required to successfully attack the user.

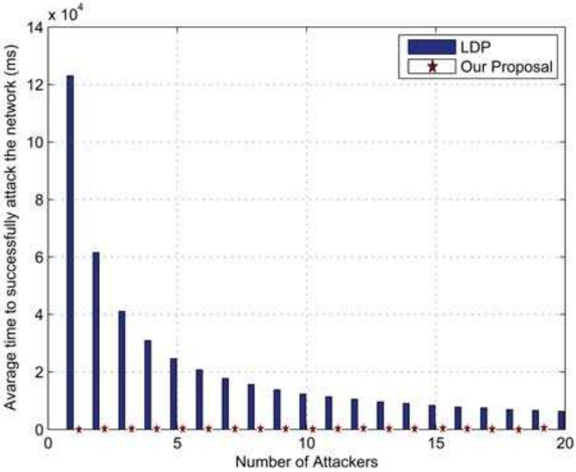


Fig. 18. Impact of a TCP SYN DDoS attack([15] ©2013 Wiley).

We ran the simulation for 500 s and placed the attack for the whole duration of the simulation time. We did not see any successful attack on our architecture. Hence, the average time required to successfully attack is remains in the initial value which is zero at the end simulation for our architecture. However, the simulation result verifies that the LDP based architecture is vulnerable to a DDoS attack as well. Furthermore, the time required to successfully attack the system is inversely proportional to the numbers of attackers.

Performance under TCP Reset attack

A TCP reset attack is a TCP/IP based attack which is used to terminate an ongoing TCP connection between two users. First, the attacker has to hijack the connection. In other words, the attacker monitors and learns the TCP header parameters such as IP addresses, port numbers, sequence numbers etc. of the TCP connection. Second, the attacker generates fake TCP packets with these TCP parameters and sends these packets with RST (Reset) Flag ON to both users or any one of the users. Since the end users do not know that an attacker has sent these packets, the end users treat these packets as legitimate packets. Therefore, they reset the TCP connection and the communication session between users is terminated.

We compared the performance of the LDP based VPLS architecture and HIPLS with our architecture under the TCP reset attack scenario. It was assumed that both VPLS users and the attacker have the same bandwidth of 100 Mbps. We calculated the probability of successful attack against the size of the file. In [118], the authors stated that the file sizes on the Internet have a Pareto distribution with a minimum file size of 4.5 KBytes and a maximum size of 20 MBytes. Hence, we changed the file sizes within this range and Figure 19 illustrates the simulation results.

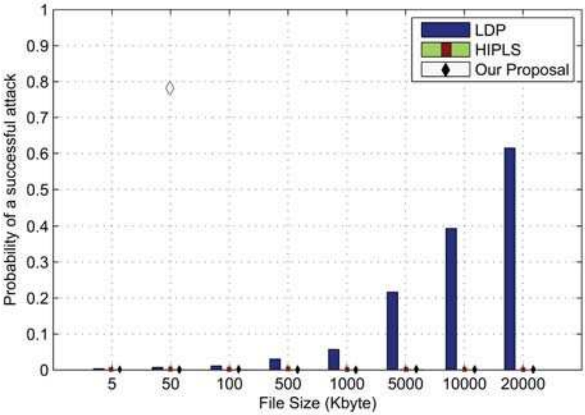


Fig. 19. Impact of a TCP reset attack([15] ©2013 Wiley).

The simulation result shows that our architecture has a zero probability of a successful attack. It verifies the protection against TCP reset attacks. However, we observe some probability of a successful attack for LDP based VPLS architectures and it is directly proportional to the file size. Larger files have longer transmission times and the attacker gets more time to reset the connection.

We further analyzed the impact of the TCP reset attack by varying the connection speed of the attacker.

Table 7. The impact of a TCP reset attack on LDP based VPLS architecture([15] ©2013 Wiley).

Attacker's Connection Speed	Average time requirement for a successfully attack (s)
1.544 Mbps (T1)	64.130
6.312 Mbps (DS2)	13.451
44.736 Mbps (DS3)	1.855
51.84 Mbps (OC1)	1.771
155.52 Mbps (OC3)	0.593

The simulations were conducted for 500 s and we did not observe any successful attacks for our proposal under any of the connection speeds. However, we obtained some values for the LDP based VPLS architecture. The experiment results (Table 7) indicate that the connection speed of the attacker is inversely propositional to the average time requirement for a successful attack.

3.4.2 Security plane scalability

Several VPLS architectures are proposed to provide efficient VPLS networks. However, HIPLS [72] is the only VPLS scheme which was able to provide a secure VPLS architecture. Furthermore, it the first and only scheme which has been proposed to use a public key authentication structure and the data traffic encryption mechanism in VPLS. We simulated the proposed architecture and HIPLS in a MATLAB environment. We considered a VPLS network which has 100 PEs and a bandwidth of 100 Mbps. The following section contains a comparison of performance security mechanisms and broadcast mechanisms in the HIPLS and our architecture.

The key storage requirements for each entity of the VPLS network has been compared. It provides major evidence to justify security plane scalability.

Key Storage at a PE

Figure 20 illustrates the key storage complexity at a PE compared to the number of PEs in the provider network. We used the number of VPNs as $M = 5$ (which is within the general number of VPNs used in an LTE backhaul network [113, 119]) and varied the PEs from 1 to 100.

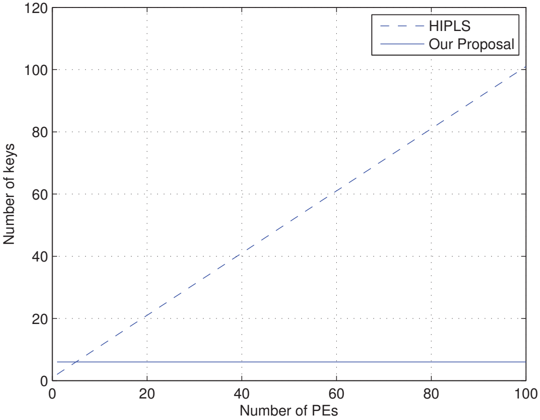


Fig. 20. The key storage complexity at a PE([15] ©2013 Wiley).

The simulation results in Figure 20 indicates a linear increment of the number of keys at a PE for the HIPLS scheme while it remains constant for the proposed architecture. Thus, the number of keys store at a PE in the proposed scheme is significantly reduced. The numerical analysis shows that the complexity of HIPLS is $O(N)$ where N is the number of PEs and the complexity of our model is $O(M+1)$ where M is the number of VPNs and it is independent of the number of PEs in the provider network. This numerical analysis explains the simulation results.

Figure 21 illustrates the key storage complexity at a PE compared to the number of VPNs in the provider network. We used the number of PEs as $N = 100$ and varied VPNs from 1 to 100.

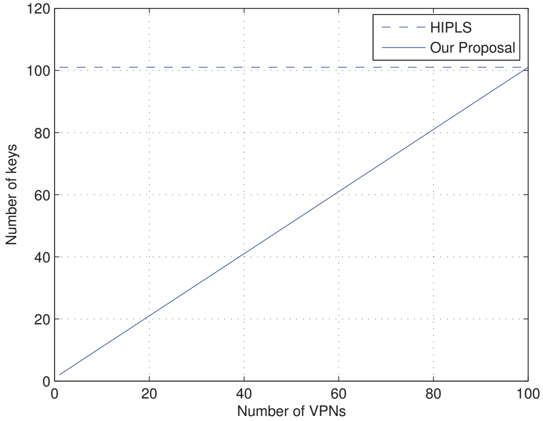


Fig. 21. The key storage complexity at a PE([15] ©2013 Wiley).

The simulation results in Figure 21 indicates a linear increment in the number of keys at a PE for our architecture while it remains constant for the HIPLS architecture. Hence, the number of keys at a PE for an HIPLS is independent of the number of VPNs in the provider network. By comparing the HIPLS scheme and the proposed scheme, the number of keys stored at a PE in the proposed scheme is reduced as long as the number of VPNs is less than the number of PEs in the network. Furthermore, we already stated that $N >>> M$ for most of the real world use cases (see Provider VPNs).

Key storage in the authentication server/Key distribution center

Figure 22 illustrates the key storage complexity at the AS/KDC compared to the number of PEs. HIPLS uses an AS and our architecture uses a KDC which provides AS services and an additional CEK distribution service. We set the number of VPNs as $M = 5$ and varied the PEs from 1 to 100.

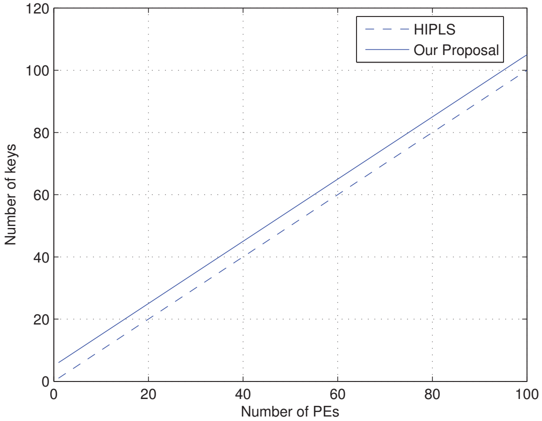


Fig. 22. The key storage complexity at the AS/KDC([15] ©2013 Wiley).

The simulation results in Figure 22 indicates a linear increment in the number of keys at the AS for the HIPLS scheme. We observed an almost similar gradual increment of the number of keys at a KDC for our scheme as well. However, the number of keys store at a KDC in the proposed scheme is slightly higher than HIPLS. This is due to the fact that the number of keys store at a KDC in the proposed scheme is dependent on both the number of PEs and VPNs, whereas it depends only on the number of PEs in the HIPLS model. However, $N >>> M$ for the most of the real world use cases ($M = 3 - 10$ and $N = 100 - 1000$) [113, 119] and and the difference is less significant for large scale networks. The numerical analysis shows that the complexity of HIPLS is $O(N)$ and the complexity of our model is $O(N + M)$. This explains the simulation results.

Figure 23 illustrates the key storage complexity at the AS/KDC compared to the number of VPNs in the provider network. We used the number of PEs as $N = 100$ and varied the PEs from 1 to 100.

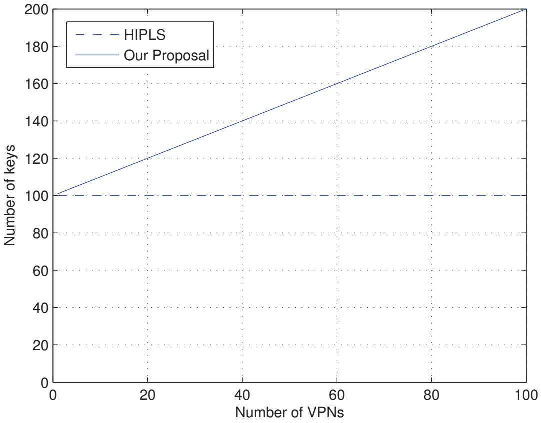


Fig. 23. The key storage complexity at the AS/KDC([15] ©2013 Wiley).

The simulation results in Figure 23 indicates a linear increment in the number of keys stored at the KDC for our architecture while it remains constant for the HIPLS architecture. Hence, the number of keys at the AS for HIPLS is independent of the number of VPNs in the provider network. By comparing the HIPLS scheme and the proposed scheme, the number of keys storage at the AS/KDC is higher in our scheme. The difference is equal to the number of VPNs in the network and as long as the numbers of VPNs are limited, the difference is less significant. Furthermore, the proposed distributed server systems can significantly reduce the key storage per server.

Total key storage in the network

Figure 24 illustrates the total key storage complexity of the VPLS network compared to the number of PEs. We set the number of VPNs as $M = 5$ and varied the PEs from 1 to 100.

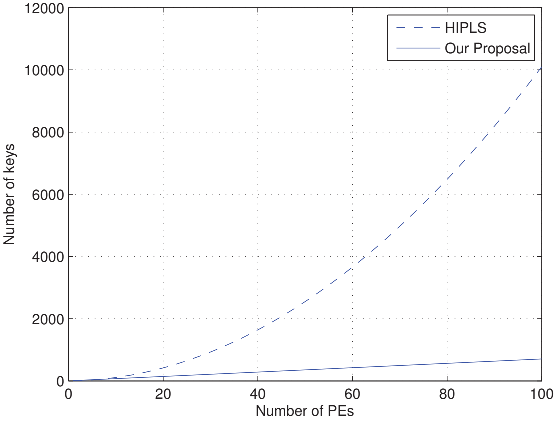


Fig. 24. Total key storage complexity of the VPLS network([15] ©2013 Wiley).

The simulation results in Figure 24 indicates an exponential increment of the total number of keys stored in the network for the HIPLS scheme while it has a linear increment for the proposed architecture. Thus, the total number of keys store in the network is significantly reduced in the proposed scheme. The complexity of HIPLS is $O(N(N+1))$ and the complexity of our model is $O(N(M+2)+M)$. This further explains the validity of the simulation results.

Figure 25 illustrates the total key storage complexity in the network compared to the number of VPNs in the provider network. We set the number of PEs as $N = 100$ and varied the PEs from 1 to 100.

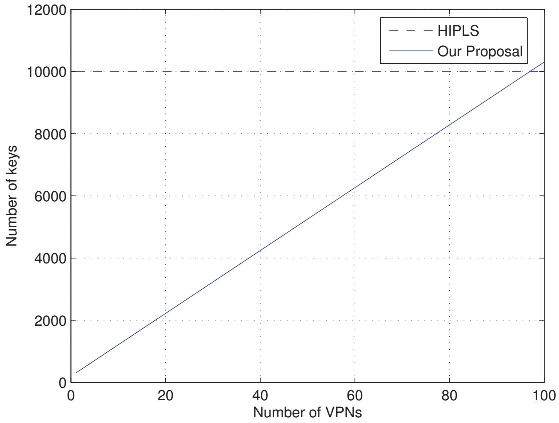


Fig. 25. Total key storage complexity of the VPLS network([15] ©2013 Wiley).

The simulation results in Figure 25 indicates a linear increment in the total number of keys stored in the network for our scheme while it remains constant the HIPLS architecture. Hence, the number of keys at an AS for HIPLS is independent of the number of VPNs in the provider network. By comparing the HIPLS scheme and the proposed scheme, the total number of keys stored in the network is reduced by the proposed scheme as long as the number of VPNs is lower than the number of PEs in the network.

The experiment results of this section verify that the proposed architecture significantly reduces the key storage complexities of the PEs and the network. Further, it shows that the security related work load on a PE is independent of the number of PEs in the network. Hence, our architecture is able to provide security plane scalability. In other words, operators can extend the network without any additional work load on top of a PE but in an AS. If it is needed, operators have the option to use resourceful and/or distributed AS systems to compensate this additional workload.

3.4.3 Forwarding plane scalability

An efficient broadcast mechanism is a key indicator to verity forwarding plane scalability. An efficient broadcast mechanism is mandatory for the proper operation of a layer 2 network, because many layer 2 network protocols such as Address Resolution Protocol (ARP) use Ethernet broadcast messages. If the broadcast mechanism is inefficient, the layer 2 nodes are unable to forward frames in larger networks. This decreases the forwarding plane scalability.

Comparison of Broadcast Mechanism

The performance of the frame broadcasting mechanism in different schemes is compared in this section. Figure 26 illustrates the number of encryptions per broadcast frame at the entry PE of the provider network.

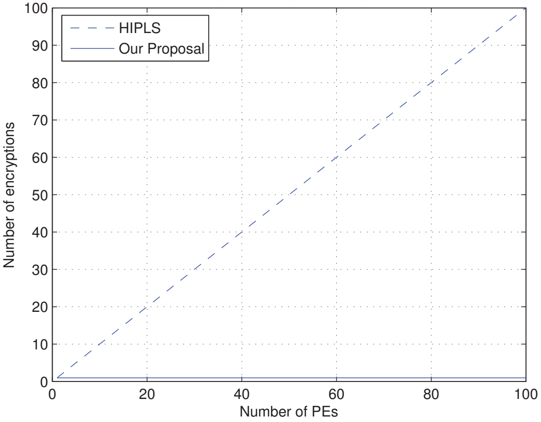


Fig. 26. The number of encryption per broadcast frame([15] ©2013 Wiley).

The simulation result in Figure 26 indicates a linear increment in the number of encryptions per broadcast frame at a PE for the HIPLS scheme while it remains constant at value 1 for the proposed architecture. Thus, the number of encryptions per broadcast frame at a PE in the proposed scheme is significantly reduced. The complexity of the HIPLS scheme is $O(N)$ and the complexity of our model is $O(1)$ and this further explains the simulation result.

The experiment clearly shows that the proposed architecture reduces the broadcast packet generation complexity at the entry PE from $O(N)$ to $O(1)$. Hence, the broadcast mechanism is almost similar to the broadcast mechanism of other non-secure VPLS architectures. Hence, this provides forwarding plane scalability.

3.5 Testbed implementation

We modeled a SCADA network which uses the existing WiFi network to interconnect L2 legacy devices via a VPLS network. Here, the existing WiFi network was provisioned as the provider network of VPLS.

The experiment test bed consists of two pieces of User Equipment (UEs) which are placed in two different locations. Two laptops with the Ubuntu 12.04 LTS (Long Term Support) operating system were used as the two UEs. Here, the provider network is a WiFi 802.11g standard wireless network which supports a maximum speed of 54 Mbps. Figure 27 illustrates the experiment testbed.

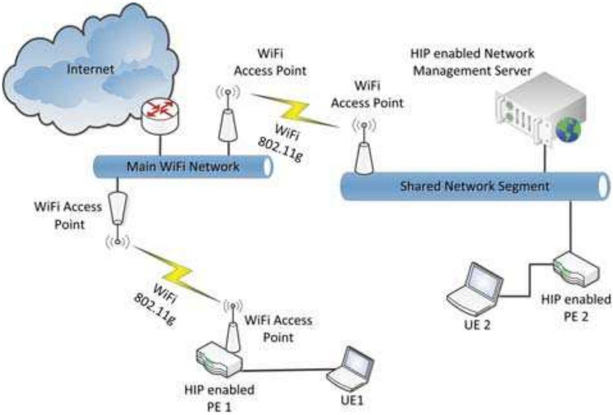


Fig. 27. The experiment testbed([26] ©2015 IEEE).

Two HIP enabled PEs are used at the edge of the Wi-Fi network. PE1 has wireless connectivity to the campus network. A wired shared network is established by using a WiFi router with four Ethernet ports. PE2 has a wired connection to the WiFi router via an Ethernet port. Two customized Industrial Security Appliances (ISAs) are used as PE devices. These ISA devices were developed by Tempered Networks [111]. Each PE uses a unique public/private RSA-2048 bit key pair as its HI.

Furthermore, an 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 the PEs, and for key management, VPN tunnel management and distribution of cryptographic credentials. It also supports the auto discovery functions of the PEs. All customer data traffic is encrypted before transporting it over the provider network. The HIP uses Advanced Encryption Standard (AES) - Cipher-block chaining (CBC) encryption. The key size of the 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 mechanisms. Thus, it is not possible to study the security penalty of other architectures. We analyzed only the HIPLS architecture in our experiments. S-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 steady state operation (once the VPLS network is established), both S-HIPLS architectures have exactly the same behavior as the original HIPLS. Both architectures use HIP tunnels to secure the data plane traffic. Thus, we present the experiment results based on original HIPLS architecture only. However, S-HIPLS also has the same behavior in this experiment setup.

In this experiment, we analyzed the security performance penalty on throughput, jitter and latency. We measured the performance against a no VPLS scenario and with non-secure VPLS (LDP VPLS [64]) scenario. The throughput and latency were measured by using the IPERF networking tool [120]. Table 8 contains the simulation settings for IPERF testing tool.

Table 8. The simulation settings for the IPERF testing tool([26] ©2015 IEEE).

Parameter	Value	Value
Protocol	UDP	TCP
Port	5004	5004
Buffer size	default (1470 kB)	default (1470 kB)
Packet size	default (1470 B)	default (1470 B)
TCP window size	-	21.0 KByte
Report interval	1 s	1 s

3.5.1 Impact on latency

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

Figure 28 contains the actual and average latency from the experiments. Based on average RTT values, the secure VPLS architecture increases the latency approximately by 87% higher than the non VPLS scenario and by 68% higher than the non secure VPLS scenario. Similarly, non secure VPLS increases the latency approximately by

11% than the non VPLS scenario. The main reason for the increase in the latency is the delay in both the packet encryption and tunnel encapsulation.

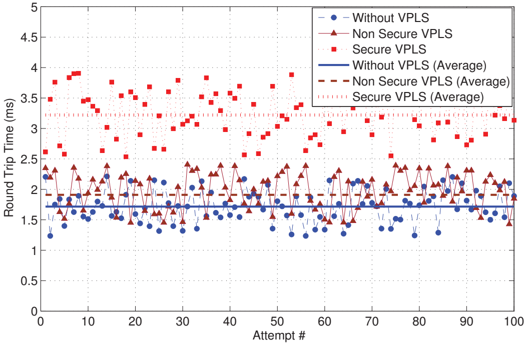


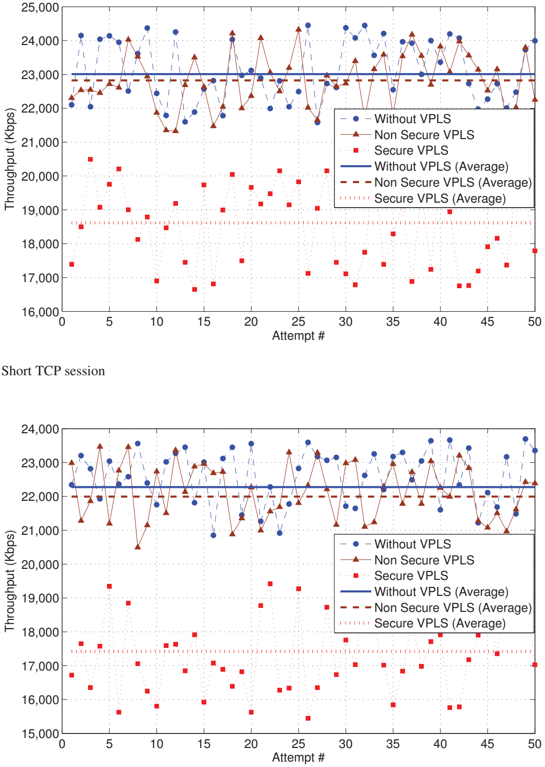
Fig. 28. The performance penalty of security on latency([26] ©2015 IEEE).

3.5.2 *Impact on throughput*

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

Impact on TCP Throughput

First, we considered the 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 s and a long TCP session runs for 500 s. Each experiment was repeated 50 times in both directions.



(a) Short TCP session

(b) Long TCP session

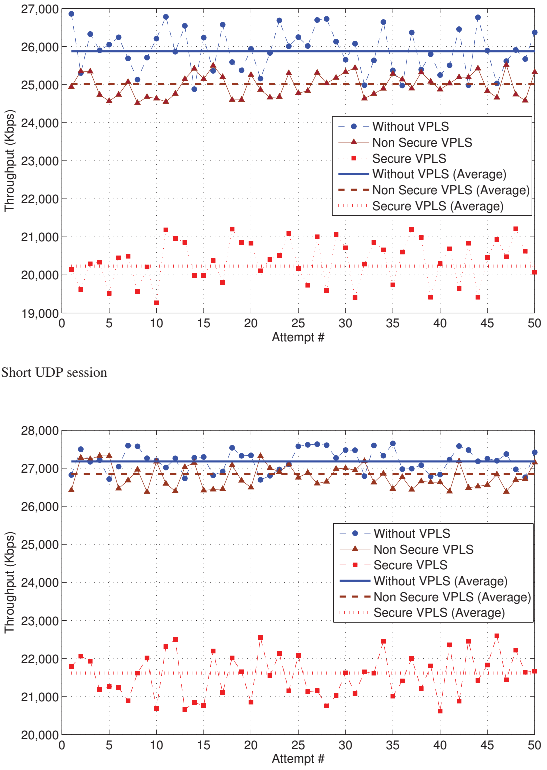
Fig. 29. The performance penalty of security on TCP throughput([26] ©2015 IEEE).

According to experiment results in Figure 29, the performance penalty of security on throughput is about 19% for a short TCP session and 21% for a long TCP session compared to 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 to 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, the non secure VPLS reduces the throughput only by 1% lower 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 reason to reduce the average throughput of the secure VPLS architecture.

Impact on UDP throughput

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



(a) Short UDP session

(b) Long UDP session

Fig. 30. The performance penalty of security on UDP throughput([26] ©2015 IEEE).

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

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

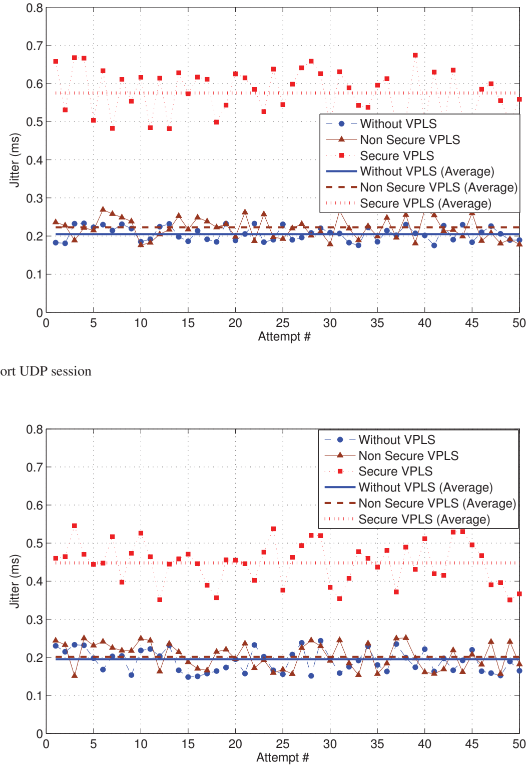
Moreover, the performance penalty of security on throughput is around 20% for both UDP and TCP sessions compared with both non VPLS and non secure VPLS scenarios. Thus, we can conclude that the performance penalty of security on throughput is independent of the transport layer protocol.

On the other hand, Wijesinha et al. observed that the maximum achievable UDP throughput is well below 50% (less than 27 Mbps) for an 802.11g Wi-Fi connection at a maximum data rate of 54 Mbps even under ideal and controlled conditions[121]. In our experiments, the UDP throughput (26.5 Mbps) is almost similar to these findings and it verified the accuracy of our test bed.

3.5.3 Impact on jitter

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

Figure 31 illustrates the experiment results.



(a) Short UDP session

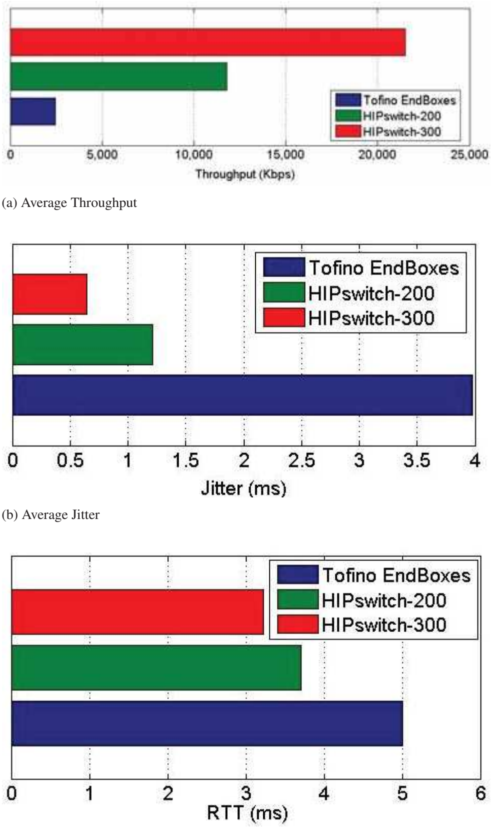
(b) Long UDP session

Fig. 31. The performance penalty of security on Jitter([26] ©2015 IEEE).

According to the experiment results in Figure 31, the average jitter of the secure VPLS architecture is 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. However, the average jitter for the secure VPLS is still less than 0.5 ms. Thus, the performance penalty of security on jitter will not affect real time applications such as VoIP, video streaming in a short range network.

3.5.4 Performance evaluation of HIPLS products

We performed experiments on the same test bed (Figure 27) to compare the performance of TEB and HIPswitches. For each experiment, we replaced PEs in the test bed with TEBs and HIPswitches. We used two type of HIPswitches namely HIPswitch-200 and HIPswitch-300. A UDP session with a bandwidth of 54 Mbps was used. The experiment results are illustrated in Figure 32.



(a) Average Throughput

(b) Average Jitter

(c) Average Latency

Fig. 32. The performance comparison of Tofino and Tempered products([26] ©2015 IEEE).

The average throughput, jitter and latency are compared. The HIPswitch-300 devices had better performance than the TEBs in all three performance metrics. HIPswitch-300 devices reduce the latency by 35% and jitter by 83% and increase the throughput by 8 times more than TEB devices. Moreover, the HIPswitch-200 devices reduce the latency by 13% and jitter by 46% and increase the throughput by 2 times more than the TEB devices. The HIPswitch-300 devices have higher processing capabilities than the TEBs. This is one of the main reasons to have higher the performance of for HIPswitches than TEBs. This result opens new opportunities for research. According to a recent Intel white paper, IPsec acceleration is possible by using external accelerators and/or using new AES instruction sets for processors [122]. Thus, the adaptation of these techniques will further improve the performance of secure VPLS products.

3.6 Security features

Our VPLS architecture is able to provide a wide range of security features for the VPLS network, namely mutual authentication, PE authorization, payload encryption, privacy protection, secure address learning and control protocol, robustness to known attacks.

Mutual authentication

The identities of the PEs are verified by using a public key authentication during HIP BEX. A HIP BEX instance is mandatory at a PE registration and prior to any data exchange between two PEs. It provides the mutual authentication between PEs and prevents outside breaches of the VPN.

PE authorization

The KDC is the responsible entity for PE authorization. It grants the access to PEs according to the ACLs which are provided by the operator. Hence, malicious users (not in the ACL) will not gain access to the VPN and it also restricts the misbehavior actions of legitimate CEs.

Payload encryption

Our VPLS scheme proposed to use IPsec ESP mode. Hence, payload is always encrypted by using the CEK of the particular VPN.

Privacy protection

Our proposal provides the privacy protection of both CEs and PEs. All the frames are encrypted and wrapped in IPsec ESP packets. Hence, the source and destination MAC address of CEs are encrypted and protected. As long as HI is exposed to the outside world, the original IP addresses of the PEs are also encrypted during the communication. This secures the privacy of the PEs.

Protection for the KDC

The KDC is a highly important element in the VPLS network. Intruders especially try to attack the KDC. However, every user who wishes to communicate with the KDC has to establish an HIP tunnel as the first step. This HIP tunnel establishment phase (HIP BEX) authenticates the users based on a PKI mechanism and authorizes it based on ACLs. Hence, intruders will not gain access. Furthermore, HIP BEX has inbuilt mechanisms to prevent DoS and spoofing attacks [73, 87]. Hence, the KDC is well secured for such attacks.

Secure control protocol

Our model uses an IPsec ESP mode to transmit the address learning and control frames. Also remote PEs are mutually authenticated before communication. Hence, fault address advertisements and eavesdropping on addresses are restricted in the VPLS. Furthermore, the control protocol uses HIP signaling though HIP tunnels which are protected from IP/TCP based attacks. Further, our simulation results verify protection for the control protocol.

Robustness to known attacks

Secure address learning and mutual authentication avoid the delivery of L2VPN traffic to the wrong destinations and access of a malicious user to the network. The secure control protocol prevents the several types of DoS attacks and unauthorized alteration of the QoS levels of VPNs. Furthermore, our proposal for an efficient broadcast/multicast frame processing mechanism at the PEs prevents the broadcast frame based DoS attacks. Payload encryption secures the VPLS traffic from unauthorized man-in-the-middle eavesdropping attacks and in flight alterations. The proposed mutual authentication mechanism in S-HIPLS uses Host Identity (A cryptographic key) to prove the identity of the user. Thus, the proposed mutual authentication mechanism is capable of verifying the identity of the entity behind the IP address and it prevents IP spoofing attacks. The proposed architecture uses HIP tunnels (IPsec BEET) in ESP mode for data communication. Thus, the original IP headers, TCP headers and payload are always encrypted. This prevents the possible eavesdropping attacks.

3.7 Comparison of our architecture and HIPLS

Both HIPLS and our VPLS architecture provide a secure VPLS architecture on top of a standard IPv4 and/or IPv6 provider network. Both architectures are based on the HIP protocol. Hence, they inherit the strong security properties found in the HIP protocol. Both proposals are able to provide strong authentication mechanisms, payload encryption, PE authorization, privacy protection, as well as robustness to several known attacks and provide a secure control protocol. However, our proposal outruns HIPLS due to its scalability and additional security features. These advantages are discussed below.

Our VPLS architecture provides a more efficient VPLS solution than HIPLS. It significantly reduces the complexity of the key storage at a VPLS node from $O(N)$ to $O(M)$ and the total key storage of the network from $O(N(N+1))$ to $O(N(M+2)+M)$ where $N >>> M$. Hence, it directly reduces the storage requirements and memory usage at PEs. Furthermore, it indirectly decreases the frame processing delay at a PE by eliminating extensive key searches and encryptions.

Our architecture provides an efficient broadcast/multicast frame processing mechanism which reduces the complexity of the number of encryption per a broadcast frame from $O(N)$ to $O(1)$ at a PE. As a result, the broadcast/multicast frame processing delay at a PE is reduced and our proposal provides some level of robustness to broadcast/multicast frame based DoS attacks. Furthermore, it needs to generate only one broadcast frame at the entry PE and it can be replicated either at the entry PE and/or at the immediate nodes according to the Spanning Tree Protocol (STP) or the multicast tree. This is a similar scenario which is used by other VPLS architectures [63][64][65]. Therefore, our VPLS scenario saves the network bandwidth and reduces the transmission delay than HIPLS. Also, our architecture relaxes the constrain to use HIPLS only for unicast-only IPLS services.

The proposed architecture provides a faster and low processing tunnel generation phase than HIPLS as it omits the extensive D-H key exchange in HIP BEX during the HIP tunnel generation between PEs (see Figure 14). The additional certificate (see Figure 13) which is sent from KDC to PEs provides some level of robustness to misfeasor attacks from CEs and PEs. Furthermore, our architecture provides the opportunity to operate separate VPNs for user and control plane traffic. Hence, it indirectly provides additional security for important system entity from the attackers. This certificate mechanism and dedicated control VPN instances were not present in HIPLS.

The only drawback of our proposal compared with HIPLS is the additional key storage requirement needed at the KDC (see Figure 23). However, it is less significant as long as the number of PEs is significantly higher than the number of VPN. In addition, a distributed KDC architecture decentralizes key storage requirement and solves this issue.

3.8 Summary and discussion

In this paper, we proposed a novel Host Identity Protocol (HIP) based VPLS architecture to provide a secure VPLS network. Our VPLS architecture is able to provide a range of security features such as mutual authentication, PE authorization, data and control frame encryption, privacy protection, secure control protocol and robustness to several types of attacks. Hence, it provides a higher degree of security than any other secure VPLS architectures. In addition, our proposal provides scalability both in security and forwarding planes. The numerical analysis verified the outrun of our architecture more than other secured VPLS solutions by significantly reducing the complexity of the key storage at a VPLS node, the total key storage of the network and the number of encryptions per broadcast frame. The simulation results further verify protection against TCP/IP based attacks such as TCP DoS, TCP DDoS and reset attacks. Hence, the proposed VPLS architecture has improve the scalability and security of existing VPLS architectures.

However, S-HIPLS is still lacking of control plane scalability. In the next chapter, we will propose a hierarchical architecture of S-HIPLS to increase the control plane scalability.

4 Secure hierarchical-VPLS architecture

In this chapter, we propose a novel hierarchical VPLS architecture based on HIP to overcome both security and scalability limitations. The proposed architecture establishes HIP tunnels between PEs in a hierarchical manner to form a VPLS network. A novel HIP signaling based control protocol is also proposed to manage the operations of the VPLS network. Hence, we name the proposed architecture a Hierarchical HIP enabled virtual private LAN Service (H-HIPLS). We also propose a novel encrypted label based secure frame forwarding mechanism to transport L2 frames over the hierarchical VPLS network.

In contrast to the typical end-to-end operation of the original HIP implementation [88], the H-HIPLS architecture uses a Şbump-in-the-wireŤ security mechanism to offer vital security features such as authentication, authorization, confidentiality, integrity, privacy protection, secure control protocol and robustness to known attacks. On the other hand, H-HIPLS also provides scalability in control, forwarding and security planes. To the best of our knowledge, this is the first secure hierarchical VPLS architecture which provides both security and scalability.

Initially, we theoretically analyze the scalability performance of the proposed architecture. Then, we conduct extended simulations to verify that the proposed architecture provides the same level of control and forwarding plane scalability as other non-secured hierarchical VPLS architectures and the same level of security plane scalability as other secure flat VPLS architectures. Furthermore, we analyze the security features of the proposed H-HIPLS architecture and its ability to protect the control protocol from IP based attacks. Finally, the data plane performance of the proposed architecture is measured in a real-world test bed implementation and compared in performance to other legacy VPLS architectures.

4.1 Related work

VPLS is a service provider provisioned L2VPN service. In provider provisioned VPNs, service providers participate in the management and provisioning functions of VPNs. The fundamental framework for a service provider provisioned L2VPN is defined by the IETF in [61]. This framework is used to standardize protocols and mechanisms to support interoperable L2VPNs. Augustyn and Serbest stated the basic implementation requirements for provider provisioned L2VPNs in [62], which are topology generation, control signaling, security, redundancy and failure recovery.

Flat VPLS Architectures: Basically, a VPLS emulates a LAN which requires a tunnel network with full mesh connectivity. Initially, the IETF defined two standard frameworks to develop a VPLS network with BGP [63] and LDP [64]. A detailed analysis of the deployment and the performance of these frameworks was presented in [58]. A simplified version of VPLS is proposed as IP-only LAN Service (IPLS) in [65]. IPLS provides a VPLS-like service and is used exclusively for IP traffic. All these architectures are flat VPLS architectures. They are lacking in scalability both in control and data planes.

Hierarchical VPLS Architectures: The first functional hierarchical VPLS architecture was proposed in [64]. Some other research studies also focused on enhancing the features of H-VPLS networks [66–71].

An L2VPN architecture that provides point-to-point and point-to-multipoint layer 2 data communication services by using a hierarchical LAN switching architecture was presented in [69]. It achieved scalability and manageability by adding some management functionality to the forwarding plane to simplify the control plane. In [68], authors proposed an H-VPLS architecture which uses a hub and spoke connectivity model to reduce the signaling and replication overhead. An enhanced H-VPLS architecture which uses a control word technique was presented in [70]. A protection scheme for an H-VPLS network was proposed in [71]. However, the IETF specified VPLS security as an indispensable factor of a VPLS since it delivers customer private frames via an untrusted public network [62], and these existing hierarchical VPLS architectures are still unable to provide the demanded level of security.

HIP-enabled virtual private LAN Service (HIPLS): The HIPLS architecture was proposed as a use-case of HIP to provide a secure VPLS over an untrusted network [72]. However, HIPLS is lacking scalability in all three planes; namely, control, forwarding and security. HIPLS is suitable only for unicast-only IPLS (IP-only Layer Services) [65] networks.

Session key based HIP-enabled virtual private LAN Service (S-HIPLS): A Session key based HIP VPLS (S-HIPLS) architecture was proposed in chapter 3. We proposed a customized version of HIP with a session key based security mechanism. S-HIPLS provides a higher degree of security plane scalability and medium degree of forwarding plane scalability for HIPLS architecture. However, S-HIPLS is still lacking control plane scalability due to the need to establish a massive number of tunnels (N-square scalability problem).

All these VPLS architectures are either lacking in scalability or requires security functions.

4.2 The proposed VPLS architecture

In this paper, we propose a secure H-VPLS architecture based on HIP to provide the demanded level of security and scalability for a VPLS network. Basically, H-HIPLS aims to establish HIP tunnels between PEs in a hierarchical manner. We propose a novel encrypted label based forwarding mechanism to facilitate the secure frame forwarding in this hierarchical architecture. H-HIPLS also supposes a dynamic address learning mechanism, a hierarchical SME (Security Management Entity) topology and a session key mechanism to facilitate the rest of the operations in the H-VPLS network. Figure 33 illustrates the protocol stack of the proposed H-HIPLS architecture.

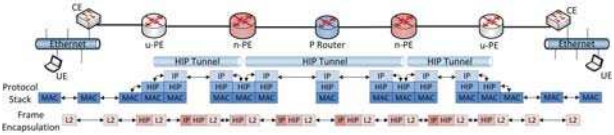


Fig. 33. The protocol stack of the proposed H-HIPLS architecture([17] ©2015 IEEE)

The operation of proposed H-HIPLS architecture is described in six sections: the control protocol, provider VPNs, PE management, the packet forwarding mechanism, an address learning mechanism and key management.

4.2.1 Control protocol

The control protocol is the heart of the VPLS network. It is mainly responsible for tunnel establishment, address learning and key management functions. The secure operation of the control protocol is mandatory to maintain the proper operation of the VPLS network[62]. The H-HIPLS proposes a secure control protocol based on HIP signaling. Basically, we define a separate provider VPN as the control VPN to securely transport the control and signaling data.

The separation of control plane traffic is motivated by several reasons. First, the control data frames require higher priority and higher QoS than user data frames. A VPN based traffic separation is an ideal mechanism to provide such services for the control traffic [113]. Second, the implementation of extra security mechanism for the control data is possible with the existence of a separate control VPN. Specifically, strong access control, firewalls and Deep Packet Inspection (DPI) mechanisms can be implemented on control VPN traffic to avoid intruder attacks on the control plane.

Provider VPNs

Similar to other provider networks such as mobile networks and Internet Service Providers (ISPs), we propose to use provider VPNs to support traffic clarification in H-HIPLS. In provider networks, the service provider maintains a small set of VPN classes (provider VPNs) based on different service factors, such as supported data rates, traffic priorities and QoS classes [113]. According to the Service Level Agreements (SLAs) with each customer, the provider classifies the customer traffic into corresponding provider VPNs.

However, the number of VPNs used in the provider network is very limited. For instance, a mobile backhaul network supports 3-5 provider VPNs [113]. In industrial networks, most of the provider networks support only one provider VPN [111, 113]. However, the service provider has the flexibility to define the number of provider VPNs in the VPLS network. For instance, the control VPN of H-HIPLS is a provider VPN which transports control and signaling data.

4.2.2 PE management

PE registration

The very first task of a newly added unregistered PE is to be registered with the VPLS network. We use an SME (Security Management Entity) to facilitate the registration process of new PEs. The proposed H-HIPLS architecture uses the SME for two tasks. 1) Authorization of unregistered PEs based on Access Control Lists (ACLs), and 2) Security key management.

We propose a novel PE registration procedure based on HIP. During this registration procedure, new PEs are authenticated based on a Public Key Infrastructure (PKI) and authorized according to ACLs. Figure 34 illustrates the PE registration procedure. Here, the initiator and responder respectively represent the unregistered newly added PE and SME.

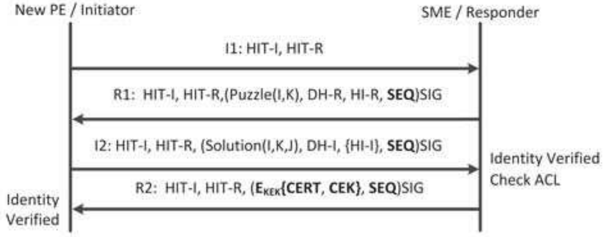

New PE / Initiator
SME / Responder
I1: HIT-I, HIT-R
R1: HIT-I, HIT-R,(Puzzle(I,K), DH-R, HI-R, SEQ)SIG
I2: HIT-I, HIT-R, (Solution(I,K,J), DH-I, {HI-I}, SEQ)SIG
Identity Verified
Check ACL
R2: HIT-I, HIT-R, (E_KEK{CERT, CEK}, SEQ)SIG
Identity
Verified

Fig. 34. PE registration procedure([17] ©2015 IEEE)

The first message (**I1**) triggers the registration procedure. It contains HITs of the initiator and responder. The SME does not allocate any resources for the new PE at the arrival of I1 message. It is a safety mechanism to avoid DoS attacks.

However, the SME sends a pre-generated **R1** message for the PE. It contains a cryptographic puzzle, Diffie-Hellman (D-H) key parameters, the public key of the SME, the sequence number and a signature. The SME includes its D-H key parameters to generate a symmetric key which is later used as a KEK (Key Encryption Key) of the PE. The R1 message also contains an HI or the public key of SME. It is used by the PE to verify the signature of the R1 message. Finally, the signature is used to verify the integrity of the R1 message. It is generated for the R1 message by using the SMEŠs private key.

The initiator sends the **I2** message after the arrival of the R1 message. It contains the solution of the puzzle, D-H key parameters, the public key of the PE, the sequence number, and a signature. The I2 message has similar obligatory fields to R1. However, the PE includes the solution of the puzzle as the puzzle parameter in R2. Finally, the signature is generated for the I2 message by using the PEŠs private key.

Upon the arrival of the I2 message, the SME subsequently checks the signature and the solution of the puzzle. Furthermore, the identity of the PE is verified after the arrival of the I2 message. Then, the SME checks the ACLs to authorize the new

PE. If the PE is a legitimate node and passes all the verification procedures, the SME completes the authentication phase by sending the R2 message. Otherwise, the SME will drop the authentication request. The **R2** message contains a CEK (Content Encryption Key), a certificate, the sequence number, and a signature. The certificate contains an authorization token (AUTH-Token), configuration information for the new PE and other VPLS management data. It is encrypted by a KEK to protect the integrity and confidentiality. The authorization token is used to establish the HIP tunnels with PEs to forward data frames. The CEK field contains the CEK of the control VPN.

The removal of PEs

It is also necessary to remove inactive PEs for the efficient operation of VPLS. H-HIPLS uses both active and passive notifications to remove the inactive PEs. In an active notification mechanism, PEs will actively notify their departure to the SME before they leave the VPLS network. In the passive notification mechanism, the SME learns the departure of a PE by the failure to acknowledge a periodic CEK distribution.

4.2.3 Tunnel establishment

According to the proposed H-HIPLS architecture, HIP tunnel establishment is mandatory between PEs before any type of data transfer. We propose a novel tunnel establishment procedure based on HIP BEX to establish these HIP tunnels. Figure 35 illustrates the tunnel establishment procedure.

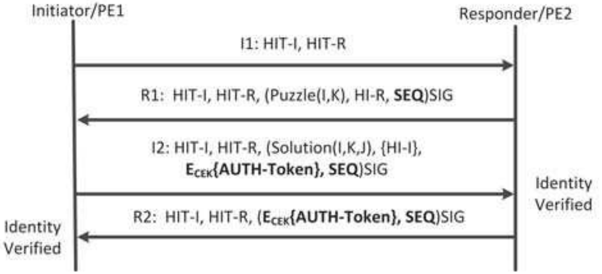

Initiator/PE1
Responder/PE2
I1: HIT-I, HIT-R
R1: HIT-I, HIT-R, (Puzzle(I,K), HI-R, SEQ)SIG
I2: HIT-I, HIT-R, (Solution(I,K,J), {HI-I}, E_CEK{AUTH-Token}, SEQ)SIG
Identity Verified
R2: HIT-I, HIT-R, (E_CEK{AUTH-Token}, SEQ)SIG
Identity Verified

Fig. 35. The tunnel establishment procedure([17] ©2015 IEEE).

The message exchanges of the proposed tunnel establishment procedure are almost similar to the message exchanges of the previous PE registration procedure. Furthermore, the functions of obligatory fields in both procedures are the same. However, there are two notable differences. First, the tunnel establishment procedure evades the D-H key exchange. Therefore, R1 and I2 messages do not contain any D-H key exchange fields. Second, we propose to exchange an authentication token during this tunnel establishment procedure. This prevents tunnel establishments by unauthorized users. The authentication token is generated as follows;

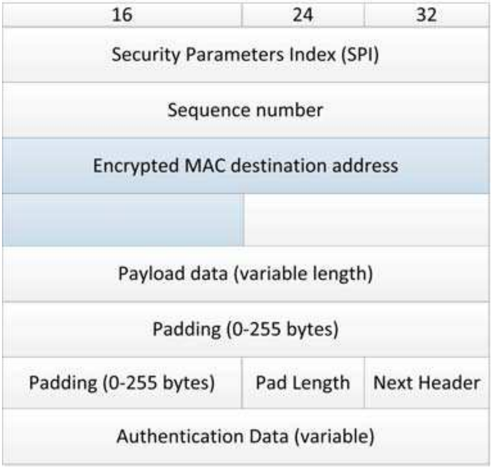
$$\text{Authentication token} = E_{CEK}(AUTH\text{-}Token||HI)$$

Each PE concatenates an AUTH-Token which is received from the SME with its HI and encrypts it using the CEK of the control VPN to generate the authentication token. The end users exchange their authentication tokens in I2 and R2 messages. The receiver of the authentication token decrypts the token and checks the respective fields by mapping with its own data. Unauthorized users cannot provide a correct authentication token since they do not have a valid AUTH-Token. Hence, connection requests with invalid authentication tokens are rejected.

4.2.4 Packet forwarding mechanism

We propose a novel encrypted label based packet forwarding mechanism and this section describes the proposal. When a u-PE receives a data frame from a CE, it follows three steps. In the first step, u-PE checks for an existing HIP tunnel with n-PE. If there is no HIP tunnel, u-PE establishes a HIP tunnel with the relevant n-PE. In the second step, the source u-PE encrypts an L2 frame using the corresponding CEK of the provider VPN. Then, it will wrap this within the ESP payload. In the third step, the source u-PE inserts the encrypted label into the standard ESP header of the packet and forwards the frame to the n-PE.

Figure 36 illustrates a modified ESP header. The encrypted label is the encrypted destination MAC (Media Access Control) address of the frame. It encrypts by using CEK of the control VPN.



16	24	32
Security Parameters Index (SPI)		
Sequence number		
Encrypted MAC destination address		
Payload data (variable length)		
Padding (0-255 bytes)		
Padding (0-255 bytes)	Pad Length	Next Header
Authentication Data (variable)		

Fig. 36. The modified ESP header([17] ©2015 IEEE).

When an u-PE receives a data frame from an n-PE, the u-PE removes the upper layer headers including the ESP header and decrypts the ESP payload using the corresponding CEK of the provider VPN. Then it will transmit to the customer access network as an L2 frame.

When a n-PE receives a data frame, it follows two steps. First, it decrypts the encrypted label and checks the MAC-PE mapping table for the next hop to forward the packet. The MAC-PE mapping table is the forwarding table of the VPLS which is used to map the destination MAC address of the data frame to the network address of the next hop PE. Second, it checks for an existing HIP tunnel between the next PE. If there is no HIP tunnel, it establishes a new tunnel. Then, it forwards the frame to the next PE.

4.2.5 Address learning mechanism

Since VPLS is a L2VPN solution, it forwards frames based on MAC addresses. On the other hand, each u-PE is responsible for a certain set of customer devices. Hence, the frames should be delivered over the provider network to reach the correct PE which is responsible for the destination device. However, the underlay provider network is an L3 network. Therefore, it is needed to map the destination MAC address of the device to the network address of the corresponding PE. We propose to maintain a dynamic MAC-PE mapping table in each n-PE to accomplish this requirement.

Each n-PE updates their MAC-PE mapping table by using two address learning instances, namely u-PE advertisements and dynamic address requests. In the first case, each u-PE advertises the MAC addresses of the responsible devices to directly connected n-PEs. Based on these advertisements, the n-PEs update their MAC-PE tables.

When an n-PE receives a frame with an unknown destination MAC, the n-PE broadcasts an encrypted address request frame (a.k.a. Dynamic Address Request) to all the other PEs to identify the responsible PE. These requests are encrypted using the CEK of the control VPN. Then, the responsible PE will send an encrypted unicast frame as a reply. Based on the reply, the requested PE updates its MAC-PE mapping table.

4.2.6 Key management

The proposed H-HIPLS architecture uses a session key mechanism instead of a per tunnel key mechanism which is proposed in classic HIP implementations [72, 73]. Similar to [15, 21], we propose to use two key types as a Content Encryption Key (CEK) and Key Encryption Key (KEK). A CEK is unique to a single provider VPN and it is used to encrypt all data frames which are belong to that VPN. The KEK is unique to a single PE and it is used to encrypt/decrypt the corresponding CEKs, certificates and any other control information. The SME is the heart of the key management system. It involves the main key management functions, namely key generation and distribution.

KEK generation and distribution

Each PE shares a unique KEK with the SME. A KEK is a symmetric key which is shared during the initial PE registration procedure. A D-H key exchange is utilized to exchange this key (see Figure 34). However, each KEK has a life time. Initially, we define the life time of a KEK as 15 minutes. However, the network operator can change the life time according to his security specifications. The SME deletes KEKs after the expiration of the life time. We propose a novel key update procedure to update KEKs after their expiration. Figure 37 illustrates the KEK update procedure.

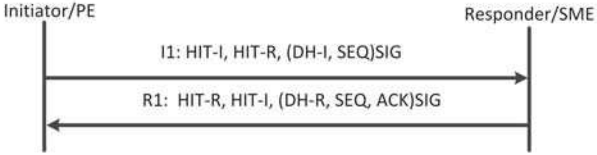

Initiator/PE
Responder/SME
I1: HIT-I, HIT-R, (DH-I, SEQ)SIG
R1: HIT-R, HIT-I, (DH-R, SEQ, ACK)SIG

Fig. 37. KEK update procedure([17] ©2015 IEEE).

The KEK update procedure has only two messages. It evades the mutual authentication phase since registered PEs already have an established HIP tunnel. Both I1 and R1 messages have a similar format as the R2 message of the previous tunnel establishment procedure. Once a KEK has expired, the PE starts a KEK update procedure. An R1 message contains an SEQ parameter. It is increased by one for each message. Furthermore, I1 contains the PEŠs D-H key parameters. The SME replies to the I1 message with an R1 message. This contains the SMEŠs D-H key parameters and the acknowledgment for the received I1 message. Based on these D-H key parameters, both the SME and PE regenerate a new KEK.

CEK generation and distribution

The SME periodically generates CEKs and securely distributes them to each PE. Before the transportation, these CEKs are encrypted by using the KEK of each PE. Similarly to the KEKs, each CEK has a life time. The SME generates new CEKs after the expiration of the life time. Apart from that, there are two possible CEK generation instances available to protect forward and backward confidentiality. Figure 38 illustrates the CEK update procedure..

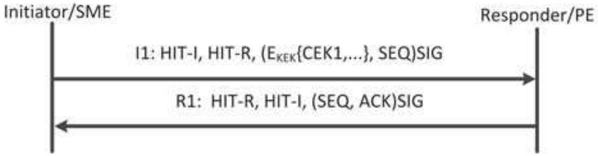

Initiator/SME
Responder/PE
I1: HIT-I, HIT-R, (E_{KEK}{CEK1,...}, SEQ)SIG
R1: HIT-R, HIT-I, (SEQ, ACK)SIG

Fig. 38. The CEK update procedure([17] ©2015 IEEE).

The CEK update procedure also has only two messages. The SME initiates the CEK update procedure. It also evades the mutual authentication phase since registered PEs have already established HIP tunnels. Both I1 and R1 messages have a similar format as the previous KEK update procedure. However, the R1 message delivers the CEKs. All these CEKs are encrypted by using the KEK of the corresponding PE.

Security management entity (SME) topology

We propose a distributed SME topology due to three reasons. First, a distributed topology eliminates a single point of failure by increasing the number of SMEs. Second, a distributed topology reduces PE registration and key distribution delays. We can keep a local SME near the PEs to reduce the communication latency. Third, a distributed topology reduces the security management work-load. A single SME utilization for a large scale network is not practical. In such networks, the SME has to maintain thousands of HIP tunnel statuses and support key management functions for thousands of PEs. Thus, a single SME utilization will increase the delay of processing security requests and the cost of a SME. Ultimately, it will limit the scalability of H-HIPLS.

Therefore, a hierarchical SME topology is proposed in H-HIPLS architecture and it is illustrated in Figure 39

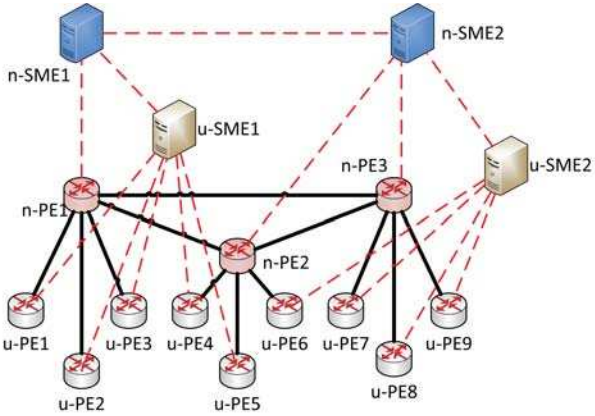


Fig. 39. Hierarchical SME topology([17] ©2015 IEEE).

Here, u-SMEs are responsible for a group of u-PEs which reside in a section of the VPLS. For instance, u-SME1 is responsible for u-PE1 to u-PE5. The n-SMEs are responsible for a group of n-PEs and u-SMEs which reside in a section of the VPLS. For instance, n-SME1 is responsible for n-PE1 and u-SME1. Finally, all the n-SMEs are mesh connected to exchange the information rapidly.

4.2.7 Broadcast mechanism

Broadcast and multicast messages play a crucial role in a layer 2 network. For instance, broadcast messages are used by VPLS address learning, Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP) services and multicast messages are used by Ethernet services, such as IP telephony, television webcast and groupware services.

On the other hand, an efficient broadcast mechanism is the key requirement to achieve forwarding plane scalability. S-HIPLS proposed an efficient broadcast and multicast mechanism by providing forwarding plane scalability for HIPLS architecture. We propose a broadcast and multicast mechanism for efficient frame distribution in the hierarchical architecture. When an u-PE receives a broadcast/multicast frame, the u-PE encrypts the frame and creates an encrypted label accordingly. Then, it directly forwards it to the n-PE. When an n-PE receives an encrypted broadcast/multicast frame, it broadcasts the frame according to a spanning or multicast tree. Hence, only n-PEs participate in the spanning or multicast tree construction.

On the other hand, it is possible to integrate the distribution of broadcast frames with the spanning or multicast tree of the underlay network since the proposed architecture generates only one encrypted frame per a native broadcast frame.

Table 9 contains a comparison of network features of different VPLS architectures. Therefore, H-HIPLS is the only VPLS architecture which satisfies both security and scalability requirements for wide area network deployment.

Table 9. A comparison of different VPLS architectures([17] ©2015 IEEE).

	IPLS	LDP based H-VPLS [64]	HIPLS [72]	S-HIPLS [15]	Proposed H-HIPLS
Architecture	Flat	Hierarchical	Flat	Flat	Hierarchical
Scalability of Forwarding Plane	Low	High	Low	High	High
Scalability of Security Plane	-	-	Low	High	High
Scalability of Control Plane	Low	High	Low	Low	High
Control Protocol Protection	No	No	Yes	Yes	Yes
Data Traffic Encryption	No	No	Yes	Yes	Yes
IP Attack Protection	No	No	Yes	Yes	Yes
Efficiency of Broadcast Mechanism	Low	High	Low	High	High

4.3 Quantitative analysis

In this section, we numerically analyze the scalability of the proposed H-HIPLS architecture. Since there are no secure hierarchical VPLS architectures, we compare the performance of our proposal with both secure flat VPLS architectures namely HIPLS [72], S-HIPLS [15, 21] and the widely use non secure H-VPLS architecture, i.e. LDP based H-VPLS (H-LDP) [64]. We use the following notation.

Number of connected customer site in VPLS network = N

Number of u-PEs in the network = N

Number of n-PEs in the network = N_n

Number of Provider VPNs in the network = M

The considered VPLS network interconnects N customer sites with N PEs/u-PEs. H-VPLS architectures use extra N_n n-PEs to form the hierarchical architecture.

4.3.1 Control plane scalability

The number of tunnels/PWs in the VPLS network is a key parameter for the comparison of the control plane scalability. The control data overhead for tunnel establishments, tunnel maintenance and topology update events can be reduced by minimizing the number of PWs in the network. Thus, the number of tunnels/PWs in the network is inversely proportional to the control plane scalability. Table 10 contains tunnels/PWs requirements for each VPLS architecture.

Table 10. The number of tunnels/PWs in the VPLS network([17] ©2015 IEEE).

	HIPLS	S-HIPLS	H-LDP	H-HIPLS
Number of PWs at a u-PE	N	N	1	2
Number of PWs at a n-PE	-	-	$N_n + \lceil \frac{N}{N_n} \rceil - 1$	$N_n + \lceil \frac{N}{N_n} \rceil$
Maximum Number of PWs at a PE	N	N	$N_n + \lceil \frac{N}{N_n} \rceil - 1$	$N_n + \lceil \frac{N}{N_n} \rceil$
Minimum Number of PWs at a PE	N	N	1	2
Number of PWs at the SME	N	N	-	N
Total number of PWs in the network	$\frac{N}{2}(N+1)$	$\frac{N}{2}(N+1)$	$\frac{N_n}{2}(N_n-1)+N$	$\frac{N_n}{2}(N_n+1)+2N$

H-HIPLS has less or an equal number of tunnels in the network to HIPLS and S-HIPLS architectures under the following condition.

$$\frac{N}{2}(N+1) \geq \frac{N_n}{2}(N_n+1)+2N \tag{1}$$

$$\frac{N}{N_n} \geq \frac{N_n+1}{N-3} \text{ for } N>3 \tag{2}$$

In large scale networks, $N > N_n$. Therefore, the RHS of equation 2 is below 1 while the LHS is above 1. Hence, H-HIPLS always has less or an equal number of tunnels per PE and in the network than HIPLS and S-HIPLS architectures in large scale networks.

On the other hand, H-HIPLS increases the number of tunnels in the network by N compared to the non-secure H-LDP architecture due to extra tunnel establishments with the SME. Thus, H-HIPLS significantly improves the scalability of the control plane than secure VPLS architectures and provides slightly deficient performance compared to existing non-secure hierarchical VPLS architectures.

4.3.2 Security plane scalability

The key storage requirement is a main parameter to compare the scalability of the security plane. If a PE needs to store a large number of keys, this uses the already scarce memory space of a PE which can be used for other functions such as forwarding tables, filters and frame buffering. On the other hand, a large number of keys can cause extensive key searches. Such procedures use extra processing power and increase the encryption delay. We evaluate the key storage requirement at different entities of the VPLS network for HIPLS, S-HIPLS and our H-HIPLS architectures. Table 11 contains the key storage requirement of each VPLS architecture.

Table 11. The key storage requirement of the VPLS network([17] ©2015 IEEE).

	HIPLS	S-HIPLS	H-HIPLS
Key storage at a u-PE	N	$M+1$	$M+1$
Key storage at a n-PE	-	-	2
Maximum number of key storage at a PE	N	$M+1$	$M+1$
Minimum number of key storage at a PE	N	$M+1$	2
Key storage at SME	N	$N+M$	$N+N_n+M$
Total key storage in the network	$N(N+1)$	$N(M+2)+M$	$N(M+2)+M+3N_n$

H-HIPLS stores fewer or equal number of keys in the network than the HIPLS architecture under the following condition.

$$N(N+1) \geq N(M+2)+M+3N_n \tag{3}$$

$$N \geq M+1+(\frac{3N_n+M}{N}) \tag{4}$$

In large scale networks where $N > M, N_n$, the condition of equation 4 is true. On the other hand, the H-HIPLS network stores an extra $3N_n$ of keys than the S-HIPLS architecture due to the additional n-PEs. However, most of these keys are stored in newly added n-PEs. Therefore, H-HIPLS will store the same number of keys per PE as S-HIPLS.

Thus, H-HIPLS significantly improves the scalability of the security plane than HIPLS through provides slightly deficient performance compared to S-HIPLS.

4.3.3 Forwarding plane scalability

An efficient broadcast mechanism is a key requirement for enhancing the scalability of the forwarding plane. Table 12 contains the performance of the broadcast mechanism for each VPLS architecture.

Table 12. The efficiency of the broadcast mechanism([17] ©2015 IEEE).

	HIPLS	S-HIPLS	H-LDP	H-HIPLS
E_{Frame}	$(N-1)$	1	-	1
L_{PE}	$(N-1)$	$(N-1)$	$\lceil \frac{N}{N_n} \rceil + N_n - 2$	$\lceil \frac{N}{N_n} \rceil + N_n - 2$

H-HIPLS always decreases the number of encryptions per broadcast frame (E_{Frame}) at a PE than HIPLS architecture. Furthermore, it has the same performance as S-HIPLS.

The maximum number of broadcast frame replications at a PE (L_{PE}) is lower in H-HIPLS than HIPLS and S-HIPLS architectures under the following condition.

$$N - \frac{N}{N_n} \geq N_n - 1 \tag{5}$$

$$N \geq N_n \text{ for } N > 1 \tag{6}$$

The above conditions are also always true since $N > N_n$ for large scale networks. Hence, H-HIPLS reduces the maximum number of broadcast frame replications at a PE more than HIPLS and S-HIPLS architectures. Furthermore, H-HIPLS has the same performance as H-LDP.

Thus, H-HIPLS significantly improves the scalability of the forwarding plane compared to secure VPLS architectures and provides similar performance as existing non-secure hierarchical VPLS architectures.

4.4 Analysis of the scalability

The proposed H-HIPLS architecture was simulated on OMNET++ and the performance of the scalability was evaluated. We compared the performance of the proposed architecture with HIPLS [72], S-HIPLS [15, 21] and H-LDP [64] architectures. A network of 125 nodes was used as the simulation model. The model network (Figure 40) was generated by using stochastic Kronecker graphs [123]. Among these nodes, 100 nodes were randomly selected as PE nodes and the rest of the nodes acted as provider routers. The tunnels between PEs were established based on the relevant VPLS architecture. Each VPLS network interconnects 100 user devices. The customer L2 devices were uniformly attached to PEs/u-PEs. We added 10 extra n-PEs for the hierarchical VPLS models, i.e. H-LDP and H-HIPLS. A maximum of 10 u-PEs were connected to each n-PE. This was equal to the number of 10 Gbps ports supported by the Cisco ASR 9001 PE router [124].

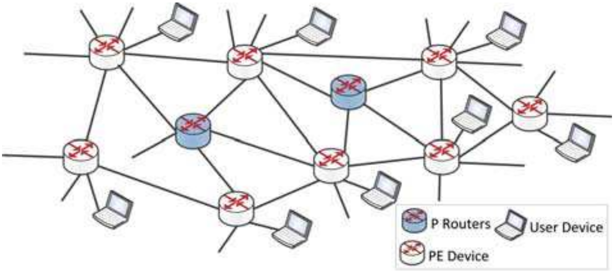


Fig. 40. The simulation model([17] ©2015 IEEE).

We assumed a scenario where each L2 device had to send average of 25 files for randomly selected L2 devices. Each device randomly selects a user device in the network to send a file and continues this process until the end of file queue. The network bandwidth at the provider network was set to 100 Mbps. We changed file sizes according to the Pareto distribution with a minimum file size of 4.5 KBytes and a maximum size of 20 MBytes [118]. Each simulation was conducted 20 times. We measured the evaluation metrics at the end of each test and average values are presented here.

4.4.1 Comparison of the control plane scalability

Total number of tunnels in the network

Figure 41 illustrates the total tunnel establishment complexity of the VPLS network compared to the number of PEs.

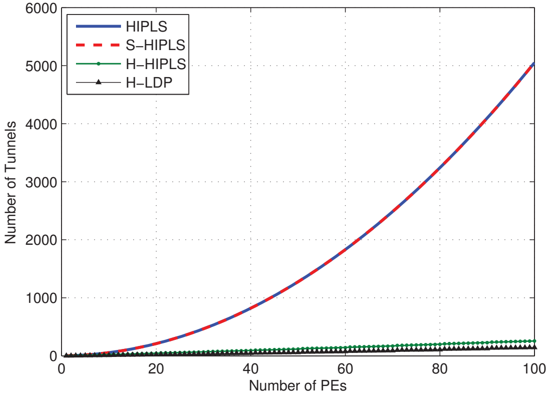


Fig. 41. The total number of VPN tunnels in the network([17] ©2015 IEEE).

A significant reduction in the total number of tunnels in hierarchical architectures was observed compared to flat architectures. There is a linear increment in the total number of tunnels with the number of PEs for both H-LDP and the proposed H-HIPLS. Comparably, H-LDP has slightly better performance than H-HIPLS since H-HIPLS needs an extra tunnel per PE for the secure key exchange with the SME. On the other hand, the total number of tunnel requirement in the network exponentially increased with the number of PEs for both HIPLS and S-HIPLS.

Therefore, the experiment results verify that the tunnel establishment complexity of the proposed H-HIPLS is significantly lower than other secured architectures i.e. HIPLS and S-HIPLS. Furthermore, H-HIPLS also offers almost similar performance as the other hierarchical architectures such as H-LDP. These simulation results match the previous numerical analysis as well. Hence, we can conclude that the H-HIPLS significantly improves the scalability of the control plane compared to other secure VPLS architectures and provides similar performance to the existing non-secure hierarchical VPLS architectures

4.4.2 Comparison of the security plane scalability

The key storage requirement is one of the main metrics to measure security plane scalability. Therefore, we evaluated the key storage requirement for different entities of the VPLS network for HIPLS, S-HIPLS and H-HIPLS architectures.

Key storage at a PE

The key storage complexity at a PE compared to the number of PEs is illustrated in Figure 42. The number of provider VPNs was set to 5 [113] and the number of PEs ranged from 1 to 100.

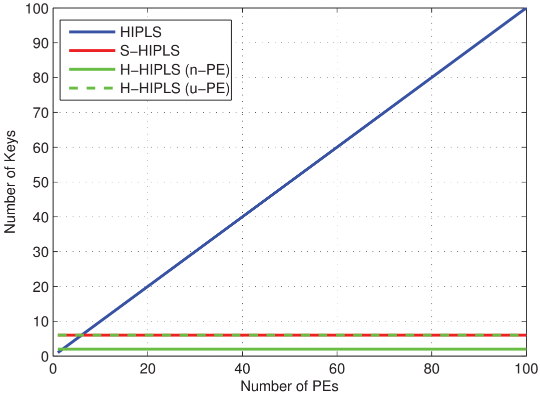


Fig. 42. The number of keys stored at a PE compared to PEs([17] ©2015 IEEE).

The simulation result indicates a linear increment in the total number of keys stored at a PE with the number of PEs for HIPLS. Both S-HIPLS and H-HIPLS (only for u-PE) have similar performance and the number of keys stored at a PE remains constant. Hence, the number of keys stored at a PE is independent of the number of PEs for both

S-HIPLS and H-HIPLS. Furthermore, the n-PE of H-HIPLS has the minimum key storage requirement as it stores only the CEK of the control VPN and its own KEK.

Figure 43 illustrates the key storage complexity at a PE compared to the number of VPNs. Here, the number of PEs is fixed to 100 and the number of VPNs ranges from 1 to 100.

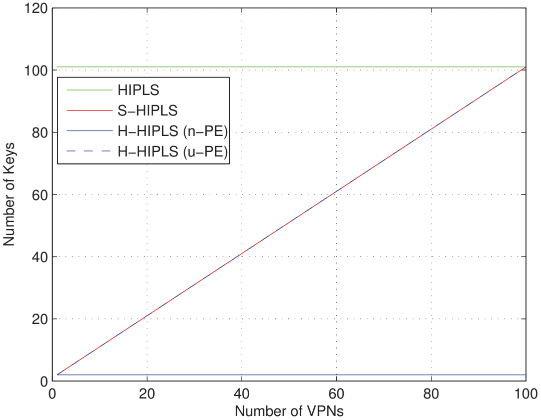


Fig. 43. The number of keys stored at a PE compared to VPNs([17] ©2015 IEEE).

A linear increment is observed in the number of keys stored at a PE with the number of PEs for S-HIPLS and u-PEs in H-HIPLS while it remains constant for HIPLS and n-PEs in H-HIPLS. Thus, the number of keys at a PE is independent of the number of VPNs for HIPLS and n-PEs in H-HIPLS. However, the performance of H-HIPLS is better than any other secure VPLS architecture as long as the number of VPNs is less than the number of PEs in the network.

Usually, the number of VPNs in a provider provisioned network is much lower than the number of PEs. In a provider provisioned network, the customer VPNs are categorized in to VPN classes based on the service level agreements. Then, the provider considers all the VPNs in a single class as a single VPN. Hence, the number of VPNs in a provider network is limited [15, 17, 113].

Key Storage in the authentication server/Security management entity (SME)

The key storage complexity in AS/SME compared to the number of PEs is illustrated in Figure 44. The number of provider VPNs is set to 5 [113] and the number of PEs ranged from 1 to 100.

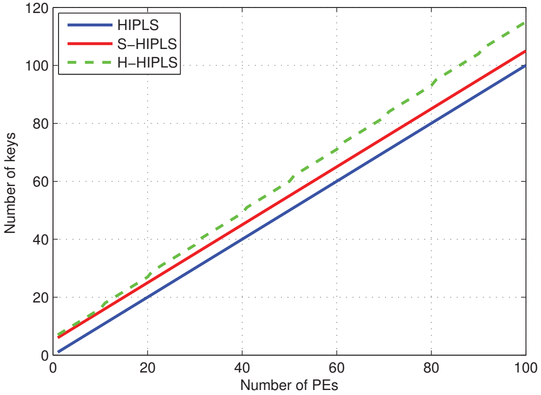


Fig. 44. The number of keys stored at the AS/SME compared to PEs([17] ©2015 IEEE).

The simulation results indicate a linear increment in the total number of keys stored at the AS/SME with the number of PEs for all three scenarios. However, the number of keys stored at an SME in H-HIPLS is slightly higher than HIPLS and S-HIPLS. The number of keys stored at an AS for HIPLS only depends on the number of PEs in the network. However, the number of keys stored at the SME for S-HIPLS and H-HIPLS architectures depends on both the number of PEs and VPNs in the network.

In addition, the number of n-PEs in the network also affects the performance of H-HIPLS. However, this deficiency in H-HIPLS can be minimized by using the proposed distributed SME topology. This weakness is less significant due the limited number of VPNs in a provider provisioned network. On the other hand, a distributed SME system can also be used to reduce the effect of this weakness.

Figure 45 illustrates the key storage complexity in AS/KDC compared to the number of VPNs. Here, the number of PEs was fixed at 100 and the number of VPNs ranged from 1 to 100.

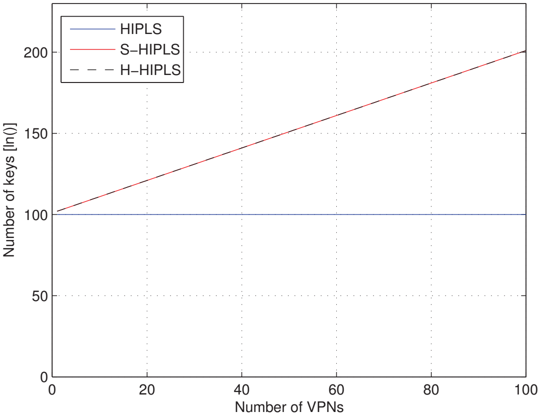


Fig. 45. The number of keys stored at a KDC/AS compared to the number of VPNs([17] ©2015 IEEE).

A linear increment is observed in the number of keys stored in the KDC with the number of VPNs for both S-HIPLS and H-HIPLS while it remains constant for HIPLS. Thus, the number of keys stored at a PE is independent of the number of VPNs for HIPLS. The performance of HIPLS is better than the proposed H-HIPLS. However, this weakness is less significant due the limited number of VPNs in a provider provisioned network. Furthermore, it can be compensated for by using a distributed KDC system.

Total key storage in the network

The total key storage in the network depends on two factors namely number of PEs and number of VPNs in the network. Hence, the performance of the proposed architecture was compared with other architectures by varying these factors.

Figure 46 illustrates the total key storage complexity of the VPLS network compared to the number of PEs. Here also, the number of provider VPNs was set to 5 [113] and the number of PEs ranged from 1 to 100.

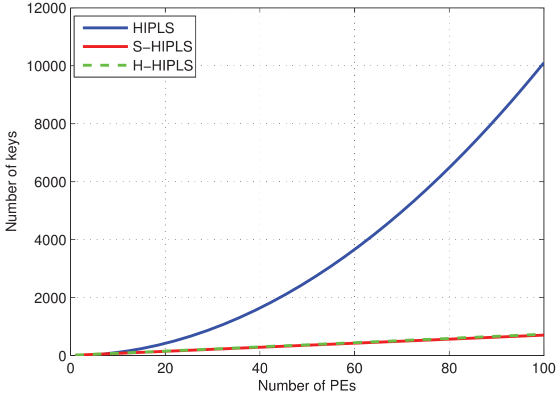


Fig. 46. The total number of keys stored in the VPLS network compared to PEs([17] ©2015 IEEE).

We observed an exponential increment in the total number of keys stored in the network with the number of PEs for HIPLS. Furthermore, S-HIPLS and H-HIPLS have almost similar performance and the number of keys stored in the network increased linearly with the number of PEs.

Figure 47 illustrates the key storage complexity at a PE compared to the number of VPNs. Here, the number of PEs was fixed to 100 and the number of VPNs ranged from 1 to 100.

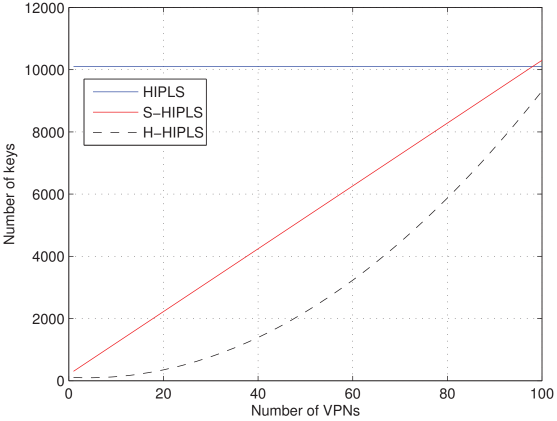


Fig. 47. The number of keys stored in the VPLS network compared to VPNs([17] ©2015 IEEE).

We observed a linear increment in the total number of keys stored in the VPLS network with the number of VPNs for S-HIPLS. Although the proposed H-HIPLS has an increment in the total number of key stored in the network, it is always less than S-HIPLS. In H-HIPLS, the CEKs are stored only in n-PEs and it reduces the total key storage in the network more than S-HIPLS. On the other hand, the number of keys stored in the network remains constant for HIPLS since it is independent of the number of VPNs. However, H-HIPLS has the best performance as long as the number of VPNs is less than the number of PEs in the network.

The experiment results clearly show that the key storage requirement in the proposed H-HIPLS is significantly lower than HIPLS and slightly higher than S-HIPLS. These simulation results match the previous numerical analysis as well. Hence, we can conclude that H-HIPLS significantly improves the security plane scalability more than HIPLS and provides similar performance to S-HIPLS.

4.4.3 *Comparison of the forwarding plane scalability*

Each PE has a maximum limit for supporting hardware ingress replications and simultaneous tunnels. This limits the forwarding plane scalability. On the other hand, an efficient broadcast mechanism is also a key requirement for improving the scalability of the forwarding plane. Hence, we compared the performance of the frame broadcasting mechanism and the number of tunnels per PE in different architectures.

Number of tunnel requirement per PE

We illustrate the total tunnel establishment complexity of a PE compared to the number of PEs in Figure 48. Here, the number of PEs ranged from 1 to 100.

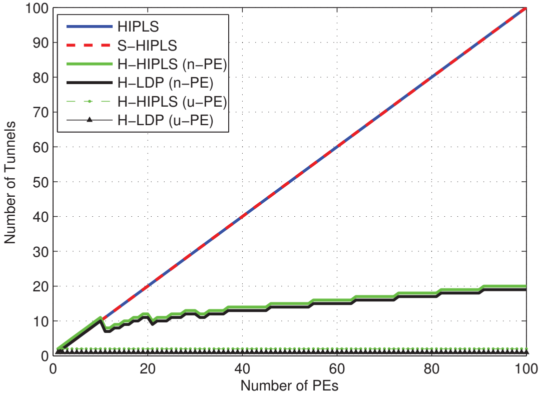


Fig. 48. The number of tunnels per PE([17] ©2015 IEEE).

The simulation results indicate a significant reduction in the number of tunnels per PE in hierarchical architectures compared to flat architectures. There is a staircase-like linear increment in the number of tunnels per n-PE with the number of PEs for both H-LDP and the proposed H-HIPLS. Furthermore, the number of tunnels per u-PE remains constant for both H-LDP and H-HIPLS as it is independent of the number of PEs. Comparably, H-LDP has slightly better performance than the proposed H-HIPLS since each PE in a H-HIPLS needs an extra tunnel for the secure key exchange. On the other hand, the number of tunnels per PE is increases linearly with the number of PEs for both HIPLS and S-HIPLS.

The maximum number of encryption per broadcast frame at a PE

We injected a single broadcast frame that should be delivered to all the PEs in the network. We measured the number of encryptions at each PE. Figure 49 illustrates the maximum number of encryptions per broadcast frame at a PE for each VPLS architecture.

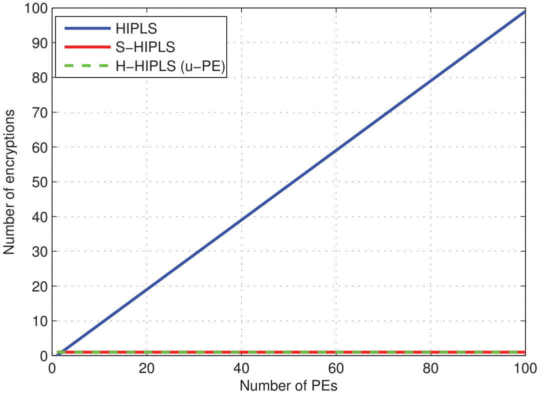


Fig. 49. The maximum number of encryptions per broadcast frame([17] ©2015 IEEE).

We can see a linear increment in the number of encryptions per broadcast frame at a PE with the number of PEs. However, the number of encryptions of both S-HIPLS and H-HIPLS remains constant at 1.

The maximum number of broadcast frame replications at a PE

Each broadcast frame needs to be replicated at intermediate the PEs to deliver it to all PEs in the network. We measured the number of replications at each PE. Figure 50 shows the maximum number of broadcast frame replications at a PE for each VPLS architecture.

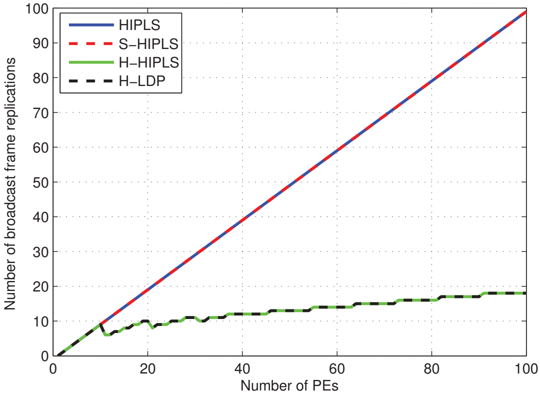


Fig. 50. The maximum number of replications per broadcast frame([17] ©2015 IEEE).

There is a linear increment in the maximum number of broadcast frame replications at a PE for both HIPLS and S-HIPLS. A significant reduction in the maximum number of broadcast frame replications at a PE is observed in hierarchical architectures compared to flat architectures. Furthermore, only n-PEs replicate the broadcast frames in H-HIPLS while all the PEs do so in HIPLS and S-HIPLS architectures.

The simulation results verify that the workload for broadcast replication at a PE and the number of tunnels per PE in H-HIPLS are significantly lower than other secure VPLS architectures. Moreover, H-HIPLS has a similar performance to H-LDP. These simulation results match the previous numerical analysis as well. Hence, we can conclude that H-HIPLS significantly improves the forwarding plane scalability more than secure VPLS architectures and provides similar performance as the existing non-secure hierarchical VPLS architectures.

4.5 IP based attack protection

The control protocol is the heart of the VPLS network. Hence, it should be protected from potential attacks. We compare the impact of IP based attacks on the control protocol of the proposed architecture. We used H-LDP [64], HIPLS [72] and S-HIPLS [15, 21] architectures as reference models to compare the performance under TCP SYN DoS, TCP SYN DDoS and TCP reset attacks.

4.5.1 The impact of attacks on tunnel establishment phase

The impact of TCP SYN DoS attack

In a TCP SYN DoS attack, an attacker sends excessive number of TCP SYN packets to the target server. The server allocates a TCP port for each successfully arrived TCP SYN packet and reserves it for a certain time period (TCP timeout). In this way, the attacker captures all ports in the server [117]. At this point, the server will stop responding to legitimate user traffic.

We use the same simulation model which was presented in Chapter 3. It has a VPLS network with 300 nodes. The bandwidth of the network is 100 Mbps. The attacker also has the same bandwidth of 100 Mbps. The number of TCP ports per user is set to 64000 and the TCP timeout is set to 270 s [117]. The simulation was run for 500 s and the attacker sends fake TCP SYN packets during 25 s - 75 s time intervals for a single user node. We measured the packet drop at the user node and Figure 51 illustrates the percentage packet drop over the simulation time.

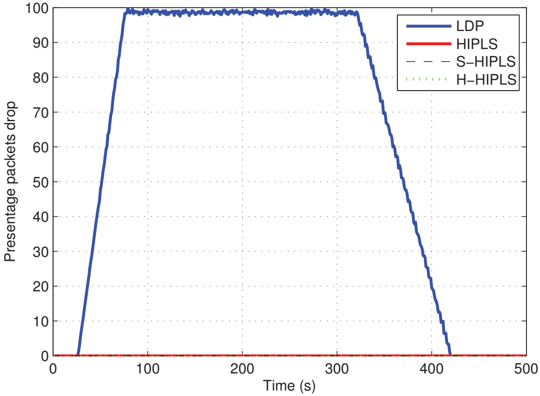


Fig. 51. The impact of a TCP SYN DoS attack([17] ©2015 IEEE).

HIPLS, S-HIPLS and the proposed H-HIPLS architecture had similar performance under the TCP SYN DoS attack. These architectures had almost zero packet drop for the whole simulation period. However, the H-LDP architecture lost almost all the packets during the attack period. Although the attack lasted for 50 s, the H-LDP architecture required an additional 270 s (a TCP timeout period) to fully recover from the attack. The simulation results verify that the control protocol of the proposed H-HIPLS is secured from TCP SYN DoS attacks.

The impact of a TCP SYN DDoS attack

A TCP SYN DoS attack was also simulated. A coordinated DDoS attack scenario was used. We used a similar simulation setup to that used to simulate the TCP SYN DoS attack. The number of attackers was gradually increased from 1 to 20. We measured the total time required to successfully attack a single user node in the VPLS network for each architecture, namely H-LDP, HIPLS, S-HIPLS and H-HIPLS. The simulation duration was 500 s for each simulation and the attackers send bogus TCP SYN packets throughout the whole duration of the simulation. The simulation was conducted 10 times and the average time required to successfully attack the user node is presented in Figure 52.

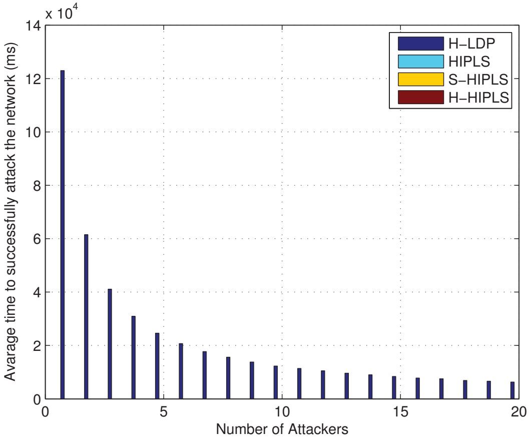


Fig. 52. The impact of TCP SYN DDoS attack([17] ©2015 IEEE).

The average time required to successfully attack all three secure VPLS architectures remained at the initial value (zero) at the end of each simulation. It verifies that there is no impact from DDoS attacks on HIPLS, S-HIPLS and H-HIPLS. However, H-LDP is vulnerable to DDoS attacks as well. The average time required to successfully attack H-LDP is decreasing with the number of attackers. Thus, the simulation results verify that the control protocol of the proposed H-HIPLS is secured from TCP SYN DDoS attacks as well.

4.5.2 The impact of attacks on the data transport phase

The impact of TCP reset attack

A TCP reset attack can terminate an ongoing TCP connection between two users by injecting fake TCP packets into the network. An attacker eavesdrops on the TCP connection and collects the TCP header information. Later, this information is used to generate fake TCP packets. The attacker sets the ŞReset BitŤ to Ş1Ť in these TCP packets. Usually, the ŞReset BitŤ is used to indicate unexpected failures on either side of the TCP connection and it makes a request to reset the connection. Since typical end users have no mechanism to identify these fake TCP packets, the end users terminate the TCP connection on the arrival of fake TCP packets [125].

We evaluated the impact of TCP reset attacks on the proposed H-HIPLS architecture and compared the performance with H-LDP, HIPLS and S-HIPLS. We used the same simulation setup which was used to evaluate the impact of TCP SYN DoS attacks. Figure 53 illustrates the probability of a successful attack compared to the size of files. We varied the size of files according to the Pareto distribution with a minimum file size of 4.5 KBytes and a maximum size of 20 MBytes [118].

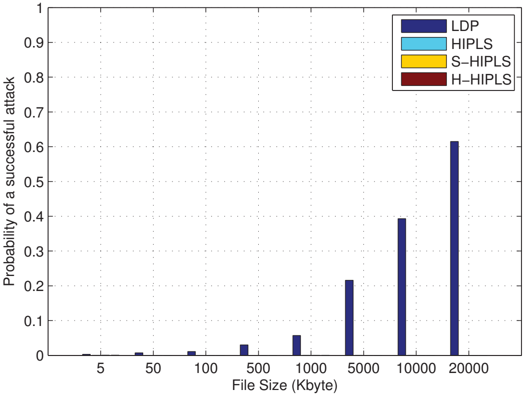


Fig. 53. The impact of a TCP reset attack([17] ©2015 IEEE).

We observed that the probability of successfully attacking the H-LDP architecture increased with the file size. The attacker gets more time to reset the connection due to the longer transmission delay of larger files. On the other hand, HIPLS, S-HIPLS and the proposed H-HIPLS have a zero probability of a successful attack. Hence, it verifies that the control protocol of the proposed H-HIPLS is secured from TCP reset attacks.

4.6 Security analysis

In this section, we analyze the security performance of proposed H-HIPLS architecture.

Protection against DoS attacks

Various DoS attacks scenarios are possible in a VPLS network. DoS attacks are very critical since they affect both control and user plane traffic. Most of VPLS DoS attacks occur at the tunnel establishment phase where the attacker sends excessive numbers of bogus connection requests to the PEs [126].

However, the H-HIPLS architecture proposes to establish an HIP tunnel before any data communication. Therefore, the attacker may attempt to send a lot of I1 messages (Figure 34 and 35) to perform a DoS attack, however, the responder will not allocate any server resources for upcoming connection requests. It just sends a precomputed R1 message for each I1 message. This R1 message contains a cryptographic puzzle to increase the commitment from the initiator. The responder only allocates server resources after the arrival of the I2 message with a correct solution to the puzzle. Therefore, the H-HIPLS architecture is protected from DoS attacks.

Protection against replay attacks

Both control and user planes of a VPLS network are vulnerable to replay attacks in various scenarios. For instance, these could occur in the tunnel establishment phase, the key update phases in the control plane and data communication sessions in the user plane.

The H-HIPLS architecture proposes the following mechanisms to counter replay attacks during tunnel establishment phase. The responder sends a virtue of the stateless response to I1 messages with pre-calculated R1 messages. It not only protects the responder against DoS attacks, but also replay attacks based on I1 messages. Monotonically increasing ŞGeneration countersŤ are included in R1, I2 and R2 messages. These counters protect the initiator from R1 and R2 based replay attacks and the responder from I2 based replay attacks.

The key exchange messages are transported over the existing HIP tunnels. HIP tunnels use the IPsec ESP (Encapsulating Security Payload) mode for data communication. The IPsec ESP mode utilizes sequence numbers to protect messages against replay attacks. Thus, the key exchange messages are automatically protected from these attacks.

HIP tunnels are used for data communication over the VPLS network. Thus, the sequence numbers mechanism which is used by the IPsec ESP mode, protects the user data frames against replay attacks. Thus, the attackerŠs replays of IPSec encrypted packet will be rejected due to sequence number mismatches with the end users.

Protection against IP spoofing attacks

IP spoofing attacks are very common in IP based networks where the attackers impersonate as legitimate users by spoofing their IP addresses. The proposed mutual authentication mechanism in H-HIPLS uses Host Identity (A cryptographic key) to prove the identity of the user. Thus, the proposed mutual authentication mechanism is capable of verifying the identity of the entity behind the IP address and it prevents IP spoofing attacks.

Protection against eavesdropping attacks

Customer data which is transmitted over a provider network is vulnerable to eavesdropping attacks. There are two types of eavesdropping attacks namely passive and active attacks. In a passive attack, the attackers try to read the ongoing customer data to capture important user information. However, the VPLS traffic of H-HIPLS architecture is encrypted by using CEKs. These keys are available only for the legitimate PEs and updated in a timely manner. An intruder cannot obtain a CEK without being authorized and authenticated by an SME. Both authorization and authentication phases enrich a wide range of security features to prevent such spoofing attacks. Without a CEK, attackers cannot eavesdrop on the communication data. Thus, user traffic in an H-HIPLS network is protected from passive attacks.

In an active attack, attackers eavesdrop on the ongoing communication channels and use the eavesdropped network information to perform various attacks such as IP spoofing, TCP reset and replay attacks. However, the proposed architecture uses HIP tunnels (IPsec BEET) in ESP mode for data communication. Thus, the original IP headers, TCP headers and payload are always encrypted. This prevents possible active eavesdropping attacks.

Protection of Security Management Entity (SME)

The SME is the responsible element for security functions in a VPLS network. It accepts only IPsec ESP packets which are sent through an HIP tunnel. Hence, a potential attacker has to establish an HIP tunnel with the SME before sending/receiving the data. However, HIP tunnel establishment follows an HIP BEX based security procedure which not only authenticates the device based on a PKI mechanism but also authorizes the initiator based on ACLs. In [112], the authors claimed that HI based authentication in an HIP BEX is sufficient enough to avoid spoofing attacks. On the other hand, the inbuilt puzzle mechanism in the tunnel establishment procedure prevents DoS and DDoS attacks [73].

Moreover, the proposed hierarchical SME topology eliminates SME's risk of single point failure.

4.7 Testbed implementation

We modelled an industrial network by using the existing wired campus network. Here, the campus network acts as the provider network of the VPLS network. Figure 54 illustrates the experiment test bed. We measured the performance penalty of the proposed H-HPLS architecture on the data plane traffic performance against the other VPLS architectures.

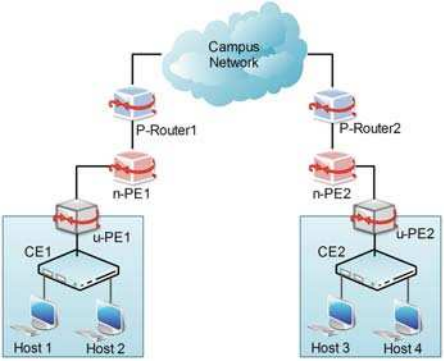


Fig. 54. The experiment testbed([17] ©2015 IEEE).

The experiment test bed consisted of two laptops and four network routers. The first laptop had an i5-3210M CPU (Central Processing Unit) of 2.5GHz and 8 Gb RAM (Random Access Memory). The second laptop had L2400 CPU of 1.66GHz and 2 GB RAM. Both laptops ran the Ubuntu 12.04 LTS (Long Term Support) Operating System

Table 13. Data Plane Performance([17] ©2015 IEEE).

	TCP Throughput (Mbps)	UDP Throughput (Mbps)	Latency (ms)	Jitter (ms)
LDP[64]	94.0567	96.2567	42.1454	0.2981
HIPLS[72]	92.8054	94.5382	45.6856	0.4025
S-HIPLS [15]	92.6435	94.3375	45.3423	0.4254
Proposed VPLS	92.4568	94.4134	47.1654	0.5456

(OS). A virtual switch (OpenVswitch version 1.10.0 [127])) was installed in each laptop. OpenVswitches acted as CEs. Two virtual hosts were connected to each OpenVswitch and they ran Lubuntu 13.10 OS. Moreover, we used an OpenHIP implementation [90] to implement virtual u-PEs on each laptop.

Two D-LINK DSR-250N [128] routers were used as n-PEs and two D-Link DIR-615 [129] routers were used as P routers in the testbed. These devices were connected as illustrated in Figure 54 by using the campus test network. The bandwidth of the network is 100 Mbps. We conducted experiments to measure the data plane performance of different VPLS architectures (i.e. H-LDP [64], HIPLS [72], TLS/SSL [130]) in terms of throughput, jitter and latency. The IPERF network measurement tool [120] was used to measure the throughput and the jitter performance. A ping test was used to measure the latency. The experiment results are presented in Table 13.

The experiment results verify that the proposed H-HIPLS architecture has almost similar throughput performance the (difference is below 1%) to HIPLS and S-HIPLS architectures. Similar to other secure VPLS architectures, H-HIPLS also has 2% lower throughput than the H-LDP architecture due to the extra layer of encryption.

The H-HPLS architecture has the highest latency. H-HIPLS has 3% higher latency than HIPLS and S-HIPLS due to the extra label encryptions at the n-PEs. Moreover, H-HPLS has 10% higher latency than H-LDP due to extra packet encryptions at u-PEs and label encryptions at n-PEs. Other secure VPLS architectures (i.e. HIPLS and S-HIPLS) have only 7% higher latency than H-LDP. According to a recent Intel white paper, IPsec acceleration is possible by using external accelerators and/or using new AES instruction sets for processors [122]. Thus, the adaptation of these techniques will further improve the latency performance of secure VPLS architectures by minimizing the encryption delays.

Although H-HIPLS architecture has the worst jitter performance, it is still less than 1ms. Thus, the performance penalty for the jitter will not affect real time applications such as VoIP, or video streaming [131].

4.8 Discussion

4.8.1 Benefits of proposed H-HIPLS architecture

Scalability

The theoretical analysis (Table 10) and simulation results verify (Figure 41 and 48) that the proposed H-HIPLS architecture establishes fewer tunnels than S-HIPLS and HIPLS. Furthermore, it requires almost the same number of tunnels as the H-LDP architecture. Hence, we can conclude that H-HIPLS significantly outruns the other secure VPLS architectures (S-HIPLS and HIPLS) in terms of control plane scalability. On the other hand, it provides almost similar scalability as non-secured hierarchical VPLS architectures.

The theoretical (Table 11 and 12) and simulation results (Figure 48, 49 and 50) verify that the number of tunnels and the workload for broadcast replication at a PE are significantly lower in H-HIPLS than other secure VPLS architectures and H-HIPLS has similar performance to H-LDP. Hence, we can conclude that H-HIPLS significantly improves the forwarding plane scalability compared to secure VPLS architectures and provides similar performance to the existing non-secure hierarchical VPLS architectures.

The theoretical (Table 11) and simulation results (Figure 42 and 46) verify that the key storage requirements in the proposed H-HIPLS are significantly lower than HIPLS and slightly higher than for S-HIPLS. Hence, H-HIPLS significantly improves security plane scalability compared to HIPLS and provides almost similar performance to S-HIPLS.

Enhanced security for the VPLS architecture

The proposed H-HIPLS architecture provides the required security features for a VPLS network, namely authentication, confidentiality, integrity, availability, secure control protocols and robustness to known attacks. The proposed tunnel establishment procedure (Figure 34 and 35) authenticates and authorizes PEs based on HIs. Thus it is capable of verifying the identity of the entity behind the IP address. IPsec based communication is used for both control and user plane communication. Thus. H-HIPLS inherits the confidentiality and integrity features of IPsec communication.

The secure control protocol and HIP based communication protect the VPLS network from many known attacks. It ensures the availability of the VPLS network. Moreover, both control and user data are protected from eavesdropping attacks due to frame encryptions. The authentication mechanism avoids spoofing, message interception and several types of DoS attacks. Moreover, the efficient broadcast mechanism prevents broadcast frame based DoS attacks.

The distribution of service provision

The proposed H-HIPLS architecture distributes the service provision among different PEs. Only u-PEs are responsible for the data encryption. Hence, they store the CEKs of all the VPNs. The intermediate n-PEs do not need to store CEKs or participate in data encryption. On the other hand, the n-PEs are responsible for other service provision functions such as dynamic address learning, forwarding table maintenance and replication of broadcast frames. Hence, the proposed H-HIPLS architecture distributes service provision functions to minimize utilization of network resources, such as memory space and processing power at PEs. Ultimately, it further enhances the control plane scalability. Table 14 illustrates the distribution of service provision of different VPLS architectures.

Table 14. The distribution of service provision([17] ©2015 IEEE).

	HIPLS	S-HIPLS	H-LDP		H-HIPLS	
	PE	PE	u-PE	n-PE	u-PE	n-PE
Number of PEs	N	N	N	N_n	N	N_n
Key storage	N	$M+1$	-	-	$M+1$	2
Number of PWs	N	N	1	$N_n+\lceil \frac{N}{N_n} \rceil -1$	2	$N_n+\lceil \frac{N}{N_n} \rceil$
E_{Frame}	$(N-1)$	1	-	-	1	-
L_{PE}	$(N-1)$	$(N-1)$	1	$\lceil \frac{N}{N_n} \rceil + N_n - 2$	1	$\lceil \frac{N}{N_n} \rceil + N_n - 2$
Complexity and Cost	High	High	Low	Medium	Low	Medium

4.8.2 Limitations of the proposed H-HIPLS architecture

In this section, we present the architectural limitations of the proposed H-HIPLS architectures.

Additional n-PEs

In hierarchical architectures, we need to use extra n-PEs in addition to u-PEs. Therefore, the number of PEs used in H-VPLS is always higher than in flat VPLS networks. This will increase the implementation cost and operational cost. However, the service distribution of the proposed H-HIPLS significantly reduces the workload of the mostly used u-PEs (See Table 14). Thus, the VPLS network can be implemented with low cost u-PEs and medium cost n-PEs. In contrast, other secure VPLS architectures need a large number of high cost PEs. The service distribution will reduce the implementation cost of H- HIPLS to some extent.

Additional encryption delay

H-HPLS architecture has slightly higher latency than other secure VPLS architectures (i.e. HIPLS and S-HIPLS) due to extra label encryptions at n-PEs. However, it is possible to reduce this delay by using IPsec acceleration [122].

Impact of volume based DoS attacks

The proposed H-HIPLS architecture is vulnerable to volume based DoS attacks such as UDP (User Datagram Protocol) floods, ICMP (Internet Control Message Protocol) floods and other spoofed-packet floods. In volume based DoS attacks, the attackers attempt to overload the network bandwidth by injecting a massive amount of junk traffic. Most types of communication networks are facing these DoS attacks[132].

However, volume based DoS attacks can be easily prevented by implementing firewalls, ingress filtering and enforcing rate bounds [132, 133]. The above security solutions are independent of our architecture and we recommend implementing them in the provider network.

Lack of dynamic tunnel parameter adjustment

Similarly to other VPLS architectures [15, 17, 63–71], the H-HIPLS architecture also proposes to maintain established IPsec tunnels between PEs and maintain them 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 wastes a PEsŠ resources such as memory, CPU and ports, but also wastes the network bandwidth for tunnel update messages. Therefore, it is necessary to fine tune the tunnel duration based on traffic demand. However, the existing secure VPLS architectures do not support dynamic parameter adjustment for tunnels.

4.9 Summary and discussion

In this paper, we proposed a novel hierarchical VPLS solution based on Host Identity Protocol (HIP). The proposed H-HIPLS architecture provides not only scalability in control, security and forwarding planes but also a wide range of security features. The theoretical and simulation results verified that

- H-HIPLS significantly increases the control and forwarding plane scalability compared to other secure VPLS architectures.
- H-HIPLS provides a similar level of security as other secure VPLS architectures.
- H-HIPLS has the same level of control and forwarding plane scalability as the insecure hierarchical VPLS architectures.
- The control protocol of H-HIPLS is protected from IP based attacks such as TCP SYN DoS and TCP reset attacks.

The proposed architecture was implemented in a real-world test bed and the performance was compared with other legacy VPLS architectures. The experiment results revealed that

- H-HIPLS has almost similar throughput performance as other secure VPLS architectures.
- H-HIPLS has only 2% lesser throughput than insecure hierarchical VPLS architectures.
- H-HIPLS has only 3% higher latency than other secure VPLS architectures.
- H-HIPLS has 10% higher latency than insecure hierarchical VPLS architectures.
- H-HIPLS has acceptable jitter performance similar to other VPLS architectures.

In the next chapter, we study the impact of VPLS architecture on L2 protocols.

5 A novel distributed spanning tree protocol for VPLS networks

In this chapter, we propose a novel Distributed STP (DSTP) to maintain a loop free Ethernet network over a VPLS network. With the DSTP we propose to run a modified STP instance in each remote segment of the VPLS network. Thus, the requirement to transport STP BPDUs through the provider network is eliminated. In addition, we propose two Redundancy Identification Mechanisms (RIMs) called Customer Associated RIM (CARIM) and Provider Associated RIM (PARIM) to mitigate the impact of invisible loops in the provider network. Thus, DSTP will be capable of establishing and maintaining a loop free Ethernet network over a VPLS network by solving the above incompatibility issues. DSTP successfully transmits broadcast frames over the VPLS network without causing any broadcast storms. Furthermore, DSTP significantly reduces the convergence time of the spanning tree and STP overhead over the provider network. Furthermore, DSTP achieves scalability by significantly reducing the number of STP messages transmitted through the provider network. We conducted several simulations to verify these features and illustrate the performance advantages of DSTP.

5.1 Spanning tree protocol

Spanning Tree Protocol (STP) is a widely used network protocol which ensures a loop free network topology for an L2 Ethernet network. STP is standardized as IEEE 802.1D [134]. Basically, STP prevents bridge loops. As a result, it prevents the broadcast radiation that results from these bridge loops. STP calculates a spanning tree which interconnects all the bridges in the network. Then, it disables all the links which are not a part of the spanning tree.

STP is the collective operation of all the bridges in the network. Thus, each bridge should have a sound knowledge of the network. STP defines special data frames called BPDUs to exchange their STP related information among the bridges. Initially, each bridge transmits configuration BPDUs to share the information with others. All bridges collaboratively select one bridge as the root bridge which is the root of the spanning tree. Then, other bridges select root ports and designated ports to establish the spanning tree. Then, each bridge blocks any other active ports which are neither a root port nor a designated port.

STP defines five port states for each port; namely blocking, listening, learning, forwarding and disabled. However, the port transition in STP is inefficient for networks which need faster convergence times. Usually, STP takes 30 to 50 seconds to respond to a topology change. Hence, the IEEE introduced the Rapid Spanning Tree Protocol (RSTP) 802.1w in 2004 [135]. It introduced new convergence behaviors and bridge port roles to STP. However, the basic features of STP remain the same for the RSTP as well. RSTP just speeds up the operation of STP by introducing new port convergence procedures.

We only consider traditional STP under this article. However, the changes which are proposed under DSTP can be directly applicable to RSTP as well.

5.2 Implementation issues of STP in a VPLS network

The traditional STP faces many issues while using it in a VPLS enabled network. Here, we briefly discuss six critical issues.

First, STP cannot eliminate the loops which are generated through the provider network. In a VPLS network, a remote network segment can be connected to multiple PEs in order to achieve network redundancy and support load balancing (e.g. Customer site2 in Figure 55). Since the VPLS builds a full mesh of PWs, there is a PW between every pair of PEs, including PEs belonging to the same network segment. These PWs are built on top of the provider network and they are invisible to STP in the customer network. As a result, broadcast frames can be looped back to the same network segment through these PWs.

Second, PWs in the provider network jeopardize the root path calculation process of STP. During the root path calculation process, each bridge selects the root path based on the accumulative cost of every link from the root bridge to the particular bridge. However, the link cost of PWs are invisible to the bridges. This may lead to invalid root path calculation and indirectly effects the performance of STP.

Third, STP is vulnerable to several security breaches. The most common attack is the Denial-of-Service (DoS) attack. Here, the attacker sends a massive amount of raw configuration BDPUs and/or Topology Change Notification (TCN) BDPUs to jeopardize the operation of STP [136, 137]. Furthermore, root role claiming and dual-home attacks also badly influence the operation of STP. Many of these attacks are initiated by eavesdropping ongoing BDPUs in the switched network. In a non-secure VPLS, an attacker can easily eavesdrop STP BPDUs on the non-secure public network.

Fourth, the STP protocol needs to transport a large number of BPDUs over the provider network during the root bridge election process. Thereafter, the root bridge broadcasts HELLO BDPUs every 2 seconds. Furthermore, every topology change in the network exchanges a lot of TCN BPDUs. This increases the overhead for the STP and consumes expensive bandwidth of the provider network. Ultimately, the customer has to pay an extra fee to transmit these STP BPDU messages.

Fifth, the extensive propagation and queuing delays in the provider network increase the spanning tree convergence time and forwarding table instabilities.

Sixth, it is highly possible to use different types of vendor switches and different types of port configurations in different remote network segments. As a result, STP BPDUs can be discarded at the provider network facing interfaces. This may affect the proper operation of STP.

5.3 Related work

STP is based on an algorithm that was invented by Radia Perlman [138]. Later, several STP versions were standardized as IEEE 802.1D [134], 802.1w [135] and 802.1s [139]. These protocol versions were designed to work only on traditional Ethernet networks. These STP specifications did not consider special network scenarios such as VPLS. Thus, all STP versions face the same set of issues in VPLS enabled Ethernet networks.

Furthermore, Cisco proposed two proprietary versions of STP as Per-VLAN (Virtual LAN) Spanning Tree (PVST) and Per-VLAN Spanning Tree Plus (PVST+). When multiple Virtual LANs are used in an Ethernet switched environment, PVST and PVST+ create a separate spanning tree for each VLAN [140]. Later, Cisco's proprietary version of RSTP to create a spanning tree for each VLAN was proposed as a Rapid Per-VLAN Spanning Tree (RPVST) [140]. However, these STP versions suffer from the same set of issues in VPLS enabled Ethernet networks since these versions did not consider the impact of VPLS networks.

However, some researchers have paid attention to common security threats to STP. Cisco proposed various security techniques such as ROOT guard [141], BPDU guard [142] and LOOP guard [143] to prevent the security threats to STP. In [136], the authors proposed an STP protection mechanism by partitioning the network into two tiers of switching networks. However, none of these techniques has considered the impact of

VPLS networks on a switched network. Hence, they are unable to prevent all attacks which are initiated from the provider network. Also, these techniques can not solve other non-security issues.

On the other hand, the IETF standardized two frameworks to develop a VPLS by using the Border Gateway Protocol (BGP) [63] and Label Distribution Protocol (LDP) [64]. Thereafter, several versions of VPLS architectures were proposed [21, 22, 65, 68, 70, 72]. However, none of these VPLS architectures paid close attention on how to use L2 customer protocols in a VPLS network. Hence, these architectures failed to solve above implementation issues of STP. Several Host Identity Protocol (HIP) based VPLS architectures were proposed to establish a secure VPLS network [21, 22, 72]. These secure architectures are able to protect STP from attacks which are initiated from the provider network. However, they still experience the other above stated issues.

Many of these architectures were proposed to evade the transmission of L2 protocol PDUs over the provider network [63, 64, 72]. This will solve some of the implementation issues of STP such as security and convergence issues. However, it is not a stable solution for two reasons. First, the broadcast mechanism will not be efficient and it will result in long packet distribution delays. Second, it will not prevent the impact of loops over the provider network such as broadcast storms, multiple frame transmissions and MAC address database instability. Therefore, none of the STP versions or VPLS architectures provides a complete solution for all the implementation issues of STP in a VPLS network.

5.4 Proposed distributed spanning tree protocol

We propose a novel Distributed Spanning Tree Protocol (DSPT). The proposed solution seeks to offer improved STP to solve the above implementation issues and enhance the scalability of STP to span provider provision networks such as VPLS. Basically, DSTP slices the spanning tree over remote customer sites. This method proposes to run a customized version of STP on each remote customer site. Thus, each network site would run a local root bridge election process and each site would elect a local root bridge for the segment. Based on this local root bridge, all other bridges in the site would decide root designation and block ports. Therefore, the PEs would not forward any STP BPDUs over the provider network.

However, the existing STP versions cannot be used as local STP instances since they are not capable of identifying loops over the provider network. Hence, we add a Redundancy Identification Mechanism (RIM) to DSTP and customize the STP accordingly. RIM identifies possible loops over the provider network and eliminates them. Two RIM implementations are proposed, namely Provider Associated RIM (PARIM) and Customer Associated RIM (CARIM).

5.4.1 Provider associate redundancy identification mechanism

The first step of the Provider Associate Redundancy Identification Mechanism (PARIM is to elect a Designated PE (DPE) for each network segment. This is a collective operation for all PEs. Initially, every PE broadcasts a Network Advertisement Packet (NAP) to all other PEs through the provider network. This NAP contains the Network Segment Identifier (NSI) of the remote network segment where the PE is connected, the priority of the PE, the link cost value, sender ID and the DPE ID. The provider assigns a unique NSI for each remote network segment and a priority value for each PE. The PE with the lowest priority value has the highest rank. The link cost value represents the bandwidth of the link that connects the PE to the provider network. The link cost is calculated according to the procedure proposed in specification IEEE 802.1D [134]. The sender ID can be the IP address of the PE or the host identity. In the beginning, each PE sets the DPE ID to its own ID.

In a similar manner to the root bridge selection process in STP [134], the PEs exchange NAPs until they select one DPE for each network site. If there are more than one PE representing a network segment, the sequence of events to determine the best DPE is to select the lowest priority value, the lowest link cost value and the lowest sender ID. Once DPEs are selected, all other PEs are set to the broadcast blocking state. In the broadcast blocking state, PEs do not forward any broadcast frames and drop all receiving broadcast frames. Only DPEs are allowed to flood broadcast frames.

Furthermore, DPE sends periodic HELLO packets via the provider network to other PEs in the same network segment to notify that the DPE is alive. An addition of a new PE or a removal of a DPE will trigger a new DPE selection process.

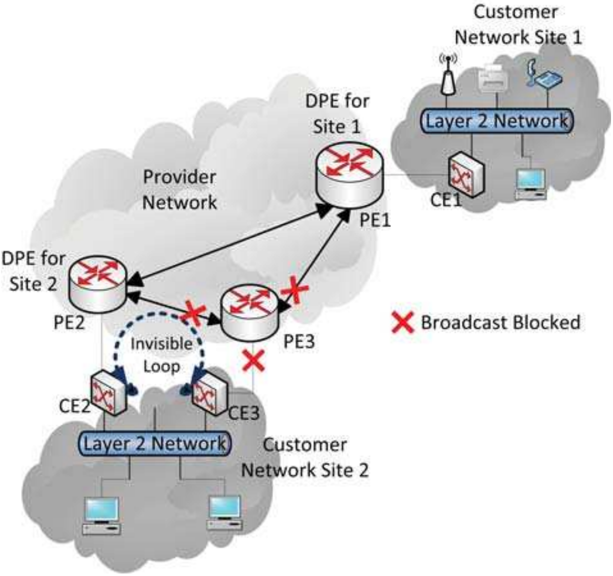


Fig. 55. Provider Associate Redundancy Identification Mechanism (PARIM)([23] ©2014 IEEE).

Figure 55 illustrates a block diagram of a VPLS network which uses the proposed DSTP with PARIM. A DSTP instance runs on each customer network site separately. Since only one PE is used to interconnect the customer network site1, PE1 is the obvious selection as the DPE for site1. Without the loss of generality, we assume that PE2 has higher priority than PE3. Hence, PE2 is selected as DPE for site2. Then, PE3 enters the broadcast blocking state.

The main advantage of PARIM is that the customer does not need to make any STP modifications in the customer network. The provider implements RIM and it is invincible for the customer. However, PARIM has some disadvantages. First, it increases the overhead of the STP on the provider network and adds an extra cost for the customer. Second, PARIM increases the complexity of the VPLS architecture.

5.4.2 Customer associated redundancy identification mechanism

Similarly to PARIM, the first step of Customer Associated Redundancy Identification Mechanism (CARIM) is to select a Designated CE (DCE) for each network segment. It is a collective operation of all CEs in a network segment. Every CE broadcasts Network Advertisement Frames (NAF) through the customer network. An NAF contains the priority of the CE, the link cost value, a MaxHop field, sender ID and DCE ID. The link cost value represents the bandwidth of the link between the CE and the PE. The sender ID can be the MAC address of the CE or the host identity. In the beginning, each CE sets the DCE ID to its own ID. MaxHop is used to avoid broadcast storms due to NAFs. The default value of MaxHop is 7 since the IEEE recommendation is to consider the maximum diameter of a network as seven bridges [134]. However, this can be changed according to the customer's requirements.

Similarly to the root bridge election process in STP [134], CEs exchange NAPs until they select one DCE for each network site. If there are more than one CE in the network segment, the sequence of events to determine the best DCE is to seek the lowest priority value, the lowest link cost and the lowest bridge ID. The bridge ID and the link cost are calculated according to the procedure proposed in IEEE 802.1D specification [134]. Furthermore, these NAFs are generated only by CEs. When any other bridge receives an NAF, it floods the NAF by reducing the MaxHop by one.

Since CEs are also L2 devices, the CEs participate in the root bridge selection process of the STP. However, only the DCEs participate in this process after the selection of the DCEs.

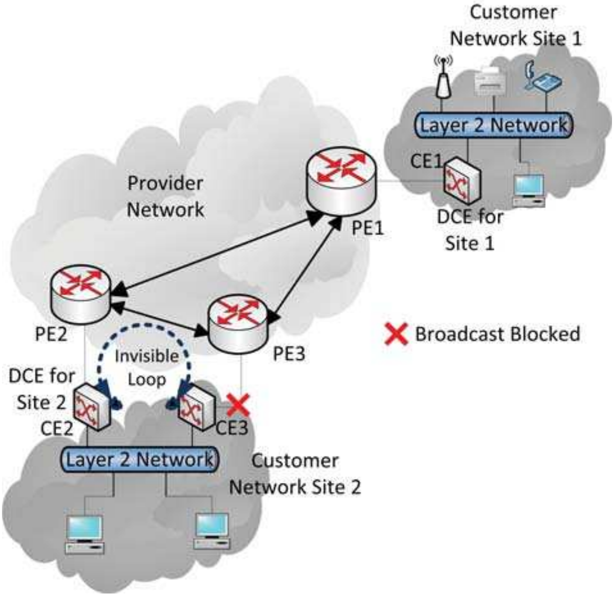


Fig. 56. Customer Associated Redundancy Identification Mechanism (CARIM)([23] ©2014 IEEE).

Figure 56 illustrates a block diagram of a VPLS network which uses the proposed DSTP with PARIM. A DSTP instance runs on each customer network site separately. Since only one CE is used to interconnect the customer network site, the CE1 is the obvious selection as DCE for site1. Without the loss of generality, we assume that CE2 has higher priority than CE3. Hence, CE2 is selected as DPE for site2 and CE3 enters the broadcast blocking state.

The main advantage of CARIM is that there is no extra STP overhead for the provider network and no additional cost for the customer. Furthermore, it does not alter the implementation of VPLS architecture. However, CARIM alters the STP behavior in the customer network.

5.4.3 Operation of the distributed spanning tree protocol

The first step of the DSTP is to run the RIM procedure. The customer has the flexibility to choose either PARIM or CARIM. RIM elects the DPEs or DCEs and sets any other PEs or CEs into the broadcast blocking state.

The second step of DSTP is to elect the root bridge. The root bridge election is local to each site. None of the CE will forward STP BPDUs to PEs and all STP BPDUs are dropped at CEs. The local root bridge election procedure is similar to the root bridge election procedure of the traditional STP [134]. However, only DCEs participate in the local root bridge election procedure in DSTP with CARIM.

Third step of DSTP is to calculate the path costs and decide the root, designated and block ports for each bridge. As the STP instance is local to each section, convergence time is comparably lower in DSTP than in traditional STP. TCN BPDUs are also local to each site. Hence, the convergence and the forwarding table updates are much faster in DSTP than the traditional STP.

When a local device transmits a broadcast frame (e.g. Address Resolution Protocol (ARP) request frame), it propagates through the spanning tree and eventually reaches the edge devices (CEs, PEs) of the customer network site. Only designated devices (DCE or DPE) transmit the received broadcast frame to all the other customer network sites through the provider network. All other edge devices drop the received broadcast frame.

When an edge device (CEs, PEs) receives a broadcast frame from any other remote customer network site, only designated boundary devices (DCE, DPE) broadcast it to the local customer network site. All other boundary devices drop the received broadcast frames.

5.5 Experiment results

We simulated the proposed DSTP on the OMNET++ network simulator [114] to analyze its performance. DSTP with PARIM, DSTP with CARIM and traditional STP [134] versions were implemented in the simulation model. Figure 57 illustrates the network model which was used for the simulations. The network model represent a basic VPLS network which has multiple connected PEs per customer site in order to achieve network redundancy and support load balancing functions.

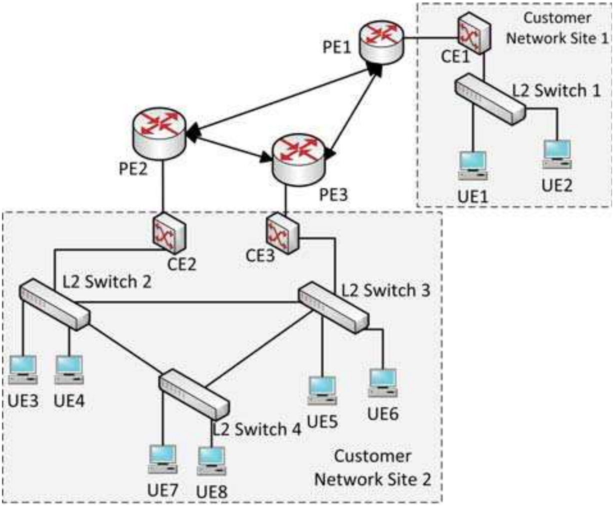


Fig. 57. The simulation model([23] ©2014 IEEE).

Various experiments were performed to measure the convergence time of the spanning tree, performance of broadcast mechanism, additional STP overhead for the provider network and scalability. The simulation model contains eight pieces of User Equipment (UEs). The links in the customer network sites had a propagation delay of 1 ms and the propagation delay between PEs was set as 5 ms. Each experiment was repeated for 10 times and only the averages are recorded.

5.5.1 The convergence time of the spanning tree

In the first experiment, the convergence time of the spanning tree was analyzed. We measured the time required to elect the root bridge by changing the propagation delay between PEs. Figure 58 illustrates the average time required to select the root bridge against the propagation delay between PEs. The root bridge selection is the first and the most time consuming step of the spanning tree establishment procedure.

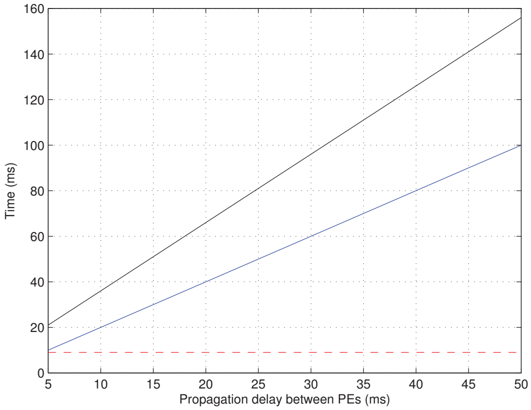


Fig. 58. The average time required to select the root bridge([23] ©2014 IEEE).

We observed a linear increment of the root bridge selection delay with the propagation delay between PEs for both STP and DSTP with PARIM. However, DSTP with PARIM significantly reduces the root bridge selection delay of the STP. On the other hand, the proposed DSTP with CARIM requires the smallest amount of time to select root bridges and the root bridge selection delay is independent of the propagation delay between PEs.

Thus, the proposed DSTP solves the impact of large propagation delays for VPN tunnels and allows customer sites to interconnect regardless of the distance between them.

5.5.2 The performance of broadcast mechanism

In the second experiment, the performance of the broadcast mechanism was analyzed. Here, UE7 broadcasts an ARP request frame. We measured the number of broadcast frames in the network.

Figure 59 illustrates the number of broadcast frames in transit compared to the simulation time. Simulation results are reported only for first 100 ms.

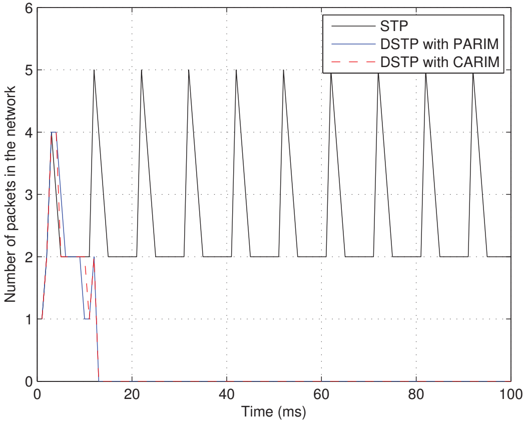


Fig. 59. The number of broadcast frames in transit([23] ©2014 IEEE).

According to the experiment results, the proposed DSTP with PARIM and DSTP with CARIM deliver the broadcast frame to all UEs after 13 ms. After that, no more broadcast frames are available in the network. However, the traditional STP generates the same broadcast frame repeatedly due to the invisible loop in the provider network. This will cause a broadcast storm.

Figure 60 illustrates the total number of broadcast frames generated compared to the simulation time. The simulation results are reported only for the first 100 ms.

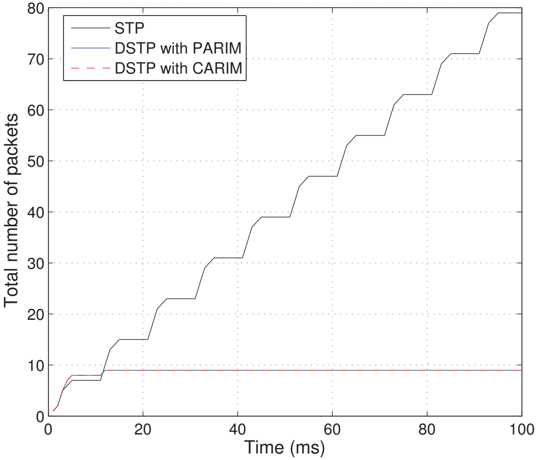


Fig. 60. Total number of broadcast frames generated in the network([23] ©2014 IEEE).

According to experiment results, DSTP with PARIM and DSTP with CARIM stop the generation of broadcast frames after 13 ms. After that, no more broadcast frames are available in the network. Also, they generate only one extra broadcast frame which is discarded at PE3 or CE3. However, the traditional STP generates the same broadcast frame repeatedly due to the invisible loop over the provider network. This will cause a broadcast storm.

Thus, we can conclude that the proposed DSTP successfully delivers broadcast frames over the VPLS and prevents the generation of broadcast storms.

5.5.3 STP overhead on the provider network

In the third experiment, the additional STP overhead on the provider network was analyzed. We measure the number of STP related messages transmitted over the provider network and Figure 61 illustrates the total number of STP messages transmitted over the provider network compared tothe simulation time. The simulations results are reported only for first 25 s.

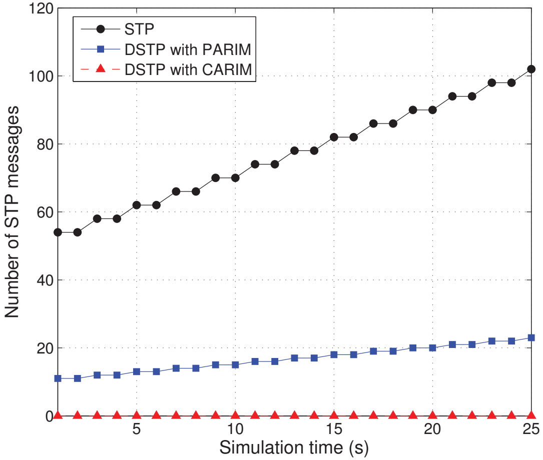


Fig. 61. Total number of STP messages transmitted over the provider network([23] ©2014 IEEE).

According to experiment results, the STP overhead on the provider network is very high for the traditional STP. It linearly increases with the simulation time due to periodic Hello BPDUs. DSTP with PARIM also has some overhead due to DPE selection messages and periodic Hello packets. The overhead linearly increases with the simulation time due to periodic Hello packets. However, DSTP with PARIM transmits the periodic Hello BPDUs only for PEs which are connected to the same customer network site. Therefore, the linear increment in DSTP with PARIM is much slower than the traditional STP. On the other hand, DSTP with CARIM has no additional overhead of STP messages since it delivers the STP BPDUs only within customer network sites.

5.5.4 Scalability

In the fourth experiment, we studied the scalability of each architecture by increasing the number of PEs in the network. Here, we assumed that each new PE connects a new customer network site to the VPLS network. Figure 62 illustrates the total number of STP messages transmitted through the provider network during the root bridge selection phase.

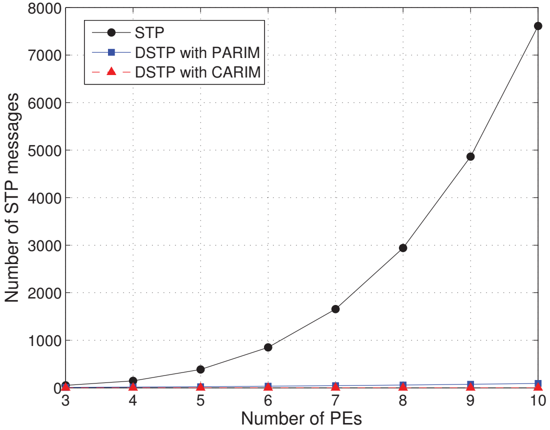


Fig. 62. The number of STP messages transmitted through the provider network during the root bridge selection phase([23] ©2014 IEEE).

The experiment results show that the number of STP messages transmitted through the provider network increases exponentially with the number PEs for the traditional STP scenario. Although the number of STP messages transmitted through the provider network had a nearly linear increment in the number of PEs for DSTP with the PARIM scenario, it was significantly lower than with the traditional STP scenario. On the other hand, DSTP with CARIM does not exchange any STP messages via the provider network.

Then, we measured the number of periodic STP messages transmitted through the provider network for each scenario. Figure 63 illustrates the average number of periodic STP messages transmitted over a 1 s time period.

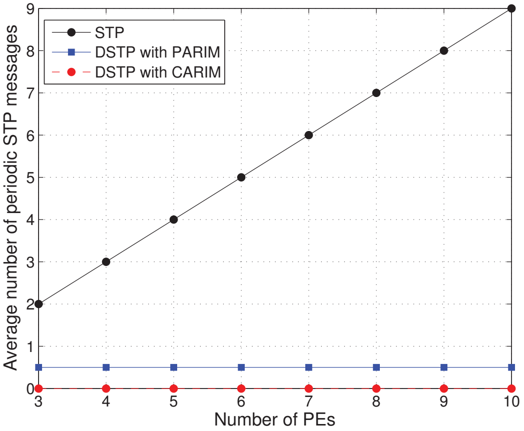


Fig. 63. The average number of periodic STP messages transmitted through the provider network per second([23] ©2014 IEEE).

The experiment results verify that the number of periodic STP messages transmitted through the provider network increases linearly with the number PEs in the network while it remains constant for the DSTP with PARIM scenario. DSTP with PARIM transmits periodic STP messages only for PEs in the same network site. Thus, the number of periodic STP messages is independent of the number PEs. It depends only on the number of PEs in the same network site. Thus, the number of periodic STP messages transmitted through the provider network is significantly lower in DSTP with PARIM than the traditional STP scenario. On the other hand, DSTP with CARIM does not exchange any periodic STP message via the provider network.

Thus, we can conclude that traditional STP is not suitable for implementation in a large scale network with a large number of PEs. DSTP offers scalability by significantly reducing the number of STP messages transmitted through the provider network in a large scale network. Ultimately, it reduces the additional overhead and STP related cost of the customer.

5.6 Summary and discussion

In this paper, we proposed a novel Distributed STP (DSTP) to maintain a loop free Ethernet network over a VPLS network. The DSTP proposes to run a modified STP instance in each remote segment of the VPLS network. Furthermore, we proposed two Redundancy Identification Mechanisms (RIMs) called a Customer Associated RIM (CARIM) and a Provider Associated RIM (PARIM) to prevent functional issues due to invisible loops in the provider network.

We conducted several simulations to verify these features and illustrate the performance advantages of DSTP. DSTP successfully transmitted broadcast frames over the VPLS network without starting any broadcast storms. Furthermore, DSTP significantly reduced the convergence time of the spanning tree and STP overhead over the provider network by outperforming existing STP versions. DSTP also increased the scalability by significantly reducing the number of STP messages transmitted through the provider network in a large scale network. Ultimately, it reduced the additional overhead on the provider network and STP related cost of the customer. Thus, DSTP is capable of establishing and maintaining a loop free Ethernet network over a VPLS network by solving existing incompatibility issues of STP.

So far, we have discussed the operation of traditional VPLS architecture. In the next chapter, we present how new technological concepts such as SDN can be used to overcome the limitations in legacy secure VPLS architectures.

6 SDN based VPLS architecture

In this chapter, we propose a novel SDN based VPLS (SoftVPLS) architecture to overcome tunnel management limitations in legacy secure VPLS architectures. Moreover, we propose three new mechanisms to improve the performance of legacy tunnel management functions.

1. A dynamic tunnel establishment mechanism: To dynamically change the tunnel parameter based on real-time network statistics.
2. A tunnel resumption mechanism : To reduce the tunnel establishment delay of subsequent tunnel establishments between authorized PEs.
3. A fast transmission mechanism : To reduce the average data transmission delay for geographically distant customer sites.

The proposed architecture utilizes IPsec enabled OpenFlow switches as PEs and the OpenFlow protocol [99] to install flow rules in PEs. A centralized controller is used to control VPLS tunnel establishment functions. The proposed dynamic tunnel establishment mechanism estimates the tunnel duration for the next tunnel instance based on network statistics provided by PEs. Therefore, the network controller can dynamically change the tunnel duration of each tunnel based on the real-time network behavior. It reduces the average number of tunnels per PE and the total number of tunnels in the network. As a result, the proposed dynamic tunnel establishment mechanism significantly increases the forwarding and the control plane scalability of VPLS networks. Moreover, this novel tunnel resumption mechanism is proposed for already authorized and previously communicated PEs. This significantly reduces the tunnel establishment delay of subsequent tunnel establishments. Finally, the performance of the proposed architecture is analyzed by using a simulation model and a testbed implementation.

6.1 Limitations of legacy VPLS tunnel management mechanisms

1. N-square scalability problem:

   Secure VPLS architectures require establishing a full mesh of IPsec tunnels between the connected customer sites [21, 22, 72]. As a result, the number of IPsec tunnels exponentially increases with the number of PEs. This is called the “N-square scalability problem”. It increases the tunnel management overhead and the operational cost. Ultimately, it reduces the control plane scalability of VPLS network.

On the other hand, every PE establishes and maintains at least $N-1$ IPsec tunnels to securely communicate with other PEs in the network (N is the number of PEs in the VPLS network). Each PE stores certain amount status data (around 20 security parameters including encryption keys, certificates, security associations [122]) to operate each tunnel. However, a PE has a limited number of ports and a limited amount of memory and processing power. This limits the number of IPsec tunnels which can be supported by a PE. For instance, if a PE is not able to support N times hardware ingress replications, the same broadcast/multi-cast frame needs to transmit N times over the same network link in a sequence. It consumes N times the expected bandwidth and adds an additional delay to subsequent frames. These limitations reduce the forwarding plane scalability of the VPLS network.

2. Static tunnel parameters:

   In legacy secure VPLS architectures, the tunnel duration of IPsec tunnels is predefined by the network administrators. Therefore, every IPsec tunnel instance is maintained for an equal and static duration regardless of the traffic demand between the customer sites. However, the periodic signaling traffic (e.g. keep alive and ping messages) for inactive tunnels introduces an additional overhead.

   On the other hand, legacy secure VPLS networks establish and maintain a full mesh of tunnels for the whole operational duration of the VPLS network [21, 22, 72]. As a result, legacy architectures reestablish IPsec tunnels at the end of each tunnel duration even without an active traffic flow through the tunnel. The reestablishment of IPsec tunnels for lightly utilized connections is a waste of network resources. It also increases the operational cost of the VPLS network.

3. Long tunnel establishment delay:

   The tunnel establishment delay highly depends on the communication link quality and distance between PEs. Legacy secure VPLS networks do not consider these physical layer constraints and all the tunnel establishments follow the same procedure. As a result, some tunnel establishment instances suffer from significantly high tunnel establishment delays (E.g. tunnels with satellite hops).

4. High transmission delay for distant sites:

   The tunnel establishment delay for geographically distant sites is very high. For instance, tunnel establishment of secure VPLS architectures [21, 22, 72] takes at least 2000 ms for sites which have 500 ms transmission delay. However, the communication session between these sites might last only 50 ms. In such cases, the customer has to wait 2000 ms just to communicate a 50 ms session. This will affect the performance of delay sensitive applications.

5. Lack of traffic engineering features:

   Legacy VPLS networks do not support traffic engineering features due to decentralized controlling and lack of network visibility.

6.2 Related work

The IETF has standardized two basic frameworks for VPLS networks by using the Border Gateway Protocol (BGP) [63] and the Label Distribution Protocol (LDP) [64]. Thereafter, several VPLS architectures were proposed to improve the performance of these frameworks [21, 22, 65, 69, 72]. A simplified version of VPLS was proposed as a IP-only LAN Service (IPLS) in [65]. IPLS provides a VPLS-like service and is used exclusively for IP traffic only. A comparison of MPLS based VPLS frameworks are presented in [58].

The very first secure VPLS architecture was proposed as a Host Identity Protocol (HIP)-enabled virtual private LAN Service (HIPLS) [72]. Here, the authors proposed a use-case with HIP to establish a secure VPLS over an untrusted network. However, the HIPLS architecture lacks in the control, data and security plane scalability. Later, two advanced HIP based VPLS architectures were proposed as Session key based HIP VPLS architecture (S-HIPLS) 3 and Hierarchical HIP VPLS architecture (H-HIPLS) 4. S-HIPLS is a flat VPLS architecture which proposed the use of a session key based security mechanism to achieve forwarding and security plane scalability. A hierarchical architecture for S-HIPLS was proposed as H-HIPLS to increase the control plane scalability as well.

Secure VPLS architectures are used in many industrial applications. For instance, Boeing uses an HIPLS based VPLS network in the assembly line for Boeing 777 airplanes [109]. Moreover, two major SCADA network appliance developing companies [110, 111] have already started to develop HIPLS based security solutions. However, all these secure VPLS architectures use static tunnel establishment procedures and they suffer from limitations such as underutilized network resources, high tunnel management overhead and lack of flexibility. Despite the H-HIPLS architecture, all other secure VPLS architectures suffer from N-square scalability problem as well.

The utilization of SDN concepts to enhance the performance of Layer 3 VPNs (virtual Private Networks) is presented in several research articles [144–148]. A use case of an SDN based VPN for data center networks was presented in [144]. A use case of an SDN based Application-Layer Traffic Optimization (ALTO) protocol for VPN traffic optimization was presented in [145]. The authors demonstrate the use of ALTO to scale-out a VPN on demand. A global wide deployment of an SDN based WAN (Wide Area Network) is presented in [146]. A survey on SDN and MPLS integration is presented in [147]. Furthermore, the possibility of implementing SDN based VPNs as a Service on an MPLS-free provider network is presented in [148]. However, none of these proposals address the limitations specific to secure VPLS architectures and the issues related to tunnel establishment procedures.

6.3 SDN based VPLS architecture

We propose a novel SDN based VPLS (SoftVPLS) architecture to enhance the tunnel management performance of existing secure VPLS architectures. This proposes three key changes. First, legacy PEs are replaced with IPsec enabled SDN switches (e.g. OpenFlow switches) and the OpenFlow protocol is utilized to install flow rules in each PE. Second, VPLS tunnel management functions are controlled by a centralized controller. Third, a Tunnel Management Application (TM App) dynamically decides the tunnel parameters based on real-time network statistics. However, the SoftVPLS architecture does not change the underlay provider network equipment other than the PEs. Therefore, existing legacy Provider Routers (P-Routers) and switches can be used as intermediate devices without making any modifications.

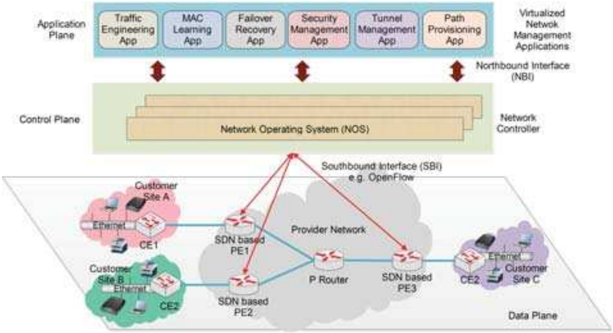


Fig. 64. The proposed SDN based VPLS architecture([31] ©2016 IEEE).

In this paper, we focus on improving only the tunnel management aspects of secure VPLS networks. However, the proposed architecture can be extended to enhance other VPLS features such as mobility management and path optimization. Moreover, we consider only flat secure VPLS architectures in this article. However, the proposed tunnel management mechanism can be applied to hierarchical architectures as well.

6.3.1 PE registration

Similar to legacy secure VPLS architectures [21, 22, 72], the first step for a new PE is to follow the registration procedure. The proposed SoftVPLS architecture also uses a HIP based registration procedure. It authorizes PEs based on ACLs (Access Control Lists) and mutually authenticates based on a PKI (Public Key Infrastructure) mechanism. Here, the network controller is responsible for registering new PEs. Moreover, an IPsec tunnel is established between the controller and PE to communicate securely after the registration.

HIP is a novel mobility and security management protocol. It separates the dual role of IP address as the locater and the Host Identity (HI). Each HIP node has a public/private key pair and the public key is used as its HI. HIP specified a Base Exchange (BEX) procedure to mutually authenticate two hosts and establish Security Associations (SAs) for an IPsec tunnel (HIP tunnel) between them[73]. Figure 65 illustrates the proposed PE registration mechanism.

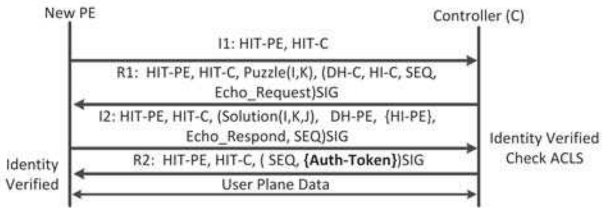

New PE
Controller (C)
I1: HIT-PE, HIT-C
R1: HIT-PE, HIT-C, Puzzle(I,K), (DH-C, HI-C, SEQ, Echo_Request)SIG
I2: HIT-PE, HIT-C, (Solution(I,K,J), DH-PE, {HI-PE}, Echo_Respond, SEQ)SIG
Identity Verified
Check ACLS
R2: HIT-PE, HIT-C, (SEQ, {Auth-Token})SIG
Identity Verified
User Plane Data

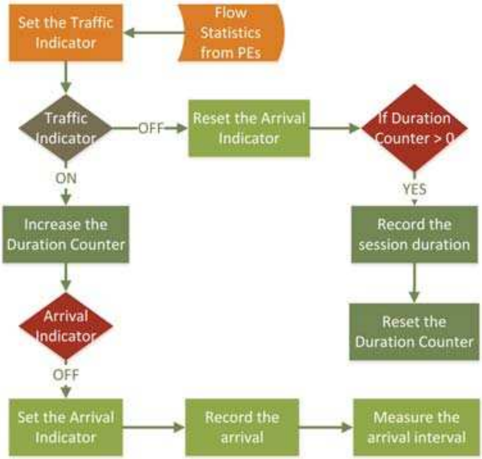
Fig. 65. PE registration mechanism([31] ©2016 IEEE).

A new PE triggers the registration procedure by sending an I1 message. Initially, the new PE and the controller are mutually authenticated by using HIs. Once the identity is verified (after receiving an I2 message), the controller checks the ACLs. These ACLs contain the list of HIs of legitimate PEs. The VPLS service operator uploads these ACLs to the network controller. The controller also sends an encrypted certificate which contains an authentication token, to the new PE. Later, this token is used to establish an IPsec tunnel with other PEs.

6.3.2 Tunnel management

In the proposed SoftVPLS architecture, the controller manages the tunnel establishment tasks based on the real-time network status, as well as network connectivity profiles and SLAs (Service Level Agreements).

The controller periodically collects the flow information from each PE to learn the real-time network status of the VPLS network. The OpenFlow protocol and switches support periodic flow information collection procedures[99]. The TM app analyzes the collected data and calculates the tunnel parameters based on the session duration and the session arrival rate for each tunnel. Therefore, we propose a Session Parameter Measuring (SPM) mechanism (Figure 66) to measure the real-time session duration and session arrival rate of each tunnel.



Set the Traffic Indicator
Flow Statistics from PEs
Traffic Indicator
OFF
Reset the Arrival Indicator
If Duration Counter > 0
YES
ON
Increase the Duration Counter
Record the session duration
Reset the Duration Counter
Arrival Indicator
OFF
Set the Arrival Indicator
Record the arrival
Measure the arrival interval

Fig. 66. The Session Parameter Measuring (SPM) mechanism([31] ©2016 IEEE).

The TM app uses two indicators for the traffic indicator and the arrival indicator. The traffic indicator contains a separate bit to represent each tunnel in the network. The TM app sets (Ş1Ť for an active session and Ş0Ť for the other case) each bit of the traffic indicator based on the received flow information from the PEs. Then, the app periodically runs the rest of the algorithm for the traffic indicator. By using this mechanism, the TM app can periodically measure the real-time session duration and the session arrival rate of each tunnel. Thereafter, the TM app periodically estimates the tunnel duration for the next tunnel instance based on the measured session characteristics.

Tunnel duration estimation

The main objective of the proper tunnel duration estimation is to minimize the average number of operational tunnels (T_{OP}) in the network. We try to minimize the active duration of each tunnel and this will ultimately reduce the T_{OP} of the network. However, the T_{OP} of legacy VPLS architectures depends only on the number of PEs in the network. They define the static tunnel duration for each tunnel regardless of the session characteristics of the tunnel. The proposed SoftVPLS architecture adjusts the tunnel duration based on dynamic characteristics, such as the session duration and the session arrival rate. Therefore, the T_{OP} of the proposed architecture depends on three factors; namely, the number of PEs in the network, the session arrival rate and the session duration of each tunnel.

The session duration of a tunnel is highly dynamic and uneven. However, we need to estimate a proper tunnel duration for the next tunnel. This should minimize premature tunnel terminations, as well as unnecessary idle tunnel durations. Thus, we propose a smoothed tunnel duration estimator based on the algorithm proposed in [149]. This algorithm considers both real time and historical average values while estimating the tunnel duration for the next tunnel instance.

$$M_{ij} \longleftarrow \alpha M_{ij} + (1-\alpha)A_{ij} \tag{7}$$

$$ETD_{ij} = \beta M_{ij} \tag{8}$$

In the above equation, A_{ij} is the measured session duration between PE_i and PE_j which is obtained from the SPM mechanism (Figure 66). M_{ij} is the smoothed mean of session durations between PE_i and PE_j. α is the smoothed factor of the mean with a recommended value of 0.9. β is the delay variance factor with a recommended value of 2[149]. The Estimated Tunnel Duration (ETD_{ij}) for the next tunnel instance between PE_i and PE_j is calculated by using the latest smoothed mean.

6.3.3 Tunnel establishment procedure

A VPLS network needs to establish PWs/tunnel between every two PEs. Similarly to legacy secure VPLS architectures, the proposed SoftVPLS architecture also establishes HIP tunnels between PEs. We propose a Tunnel Establishment Procedure (TEP) based on HIP BEX and it is illustrated in Figure 67.

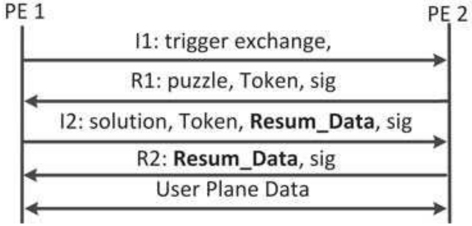


Fig. 67. Tunnel establishment procedure([31] ©2016 IEEE).

Both PEs are mutually authenticated by using their HIs. Moreover, encrypted Auth tokens which are provided by the controller, are exchanging to authorize the tunnel establishment. The proposed TEP is quite similar to the previous PE registration mechanism. However, the PEs exchange tunnel Resumption Data (Resum_data) in I2 and R2 messages. This is an optional field which is used to exchange the tunnel information related to the tunnel resumption mechanism. In other case, PEs evade the transmission of the Resum_data. The duration of each tunnel is defined by the TM app according to the tunnel duration estimation mechanism (Equation 8).

6.3.4 Tunnel resumption

The SoftVPLS architecture also proposes a novel tunnel resumption mechanism for previously communicated and authorized PEs. It is a fast tunnel re-establishment mechanism which significantly reduces the tunnel establishment delay for subsequent tunnels. During the initial tunnel establishment, PEs exchange tunnel resumption data which will later be used to support the tunnel resumption.

However, it is not economical, advantageous or necessary to support tunnel resumption for every tunnel in the network. For instance, the storage of the tunnel resumption data for every tunnel in the network will consume the already limited memory of the PEs and reduce the scalability of the network. Furthermore, not every tunnel transports delay critical user data or has long propagation delays. Therefore, we need to select a subset of tunnels which require tunnel resumption. On the other hand, tunnel resumption can be used to minimize the average tunnel establishment overhead of frequently arriving sessions.

The TM app decides which tunnels are allowed to use tunnel resumption by considering the following factors.

1. The normalized session arrival rate (F): High priority is given to the high frequent flows to reduce the tunnel establishment overhead. The session arrival rate is measured by using the SPM mechanism (Figure 66).
2. The normalized propagation delay between end PEs (D): High priority is given to tunnels with high propagation delays. The propagation delay can be measured by using flow information from the PEs.
3. The QoS (Quality of Service) requirement of the traffic flow : High Priority (P) is given to delay sensitive traffic flows. These can be defined based on SLAs between the customer and the provider.
4. The number of tunnel resumption instances supported by PE (T) : This optimizes memory utilization.

Based on the above, we calculate the Resumption Parameter (RP) for the tunnel between PE_i and PE_j.

$$RP_{i,j} = P_{i,j} + D_{i,j}F_{i,j} \tag{9}$$

The highest Priority (P) value is assigned for the highest priority flow. $F_{i,j}$ is the normalized average session arrival rate between PE_i and PE_j. $D_{i,j}$ is the normalized propagation delay between PE_i and PE_j. After calculating the RP values for all the tunnels, we sort them and select the tunnels with the highest RP values until we reach the maximum T_i for every PE. The tunnel resumption facility will be available only for the selected tunnels. RP value calculation is a periodic operation. Therefore, it is possible to change tunnel resumption parameters (e.g. priority values and SLAs) dynamically.

Tunnel resumption procedure

The Tunnel Resumption Procedure (TRP) is available only for selected and previously authorized/communicated users. Each PE creates a 32-bit Tunnel Identifier (TID) as a part of the resumption data which is exchanged during a TEP. A cache of TIDs and negotiated tunnel parameters is maintained for each selected PE. Authorized PEs can initiate a TRP instead of a TEP during the subsequent tunnel establishments. Basically, the TRP reduces the tunnel establishment delay by one round trip. However, unauthorized PEs have to follow the full TEP. The proposed TRP is illustrated in Figure 68.

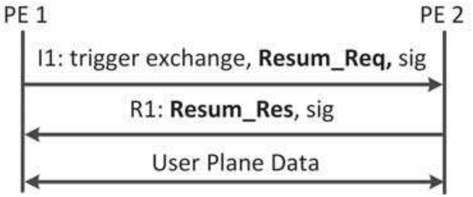


Fig. 68. The Tunnel Resumption Procedure (TRP)([31] ©2016 IEEE).

6.3.5 Fast transmission procedure

The proposed architecture supports Fast Transmission Procedure (FTP) for delay sensitive applications and tunnels with long transmission delays. Moreover, it is not necessary to support fast transmission for every tunnel in the VPLS network. For instance, not all the tunnels transport delay critical user data and not all the tunnels have long transport delays. Therefore, TM app decides which tunnels are allowed to utilize tunnel resumption by considering the following factors.

1. The traffic Transport Delay between end PEs (D) - Eliminate short tunnels - Calculate the distance by using flow information from PEs.
2. QoS requirement of traffic flow (P)- Provide the priority for delay sensitive traffic flows - Based on SLAs (Service Level Agreements) between customer and provider.

The proposed FTP is illustrated in Figure 69.

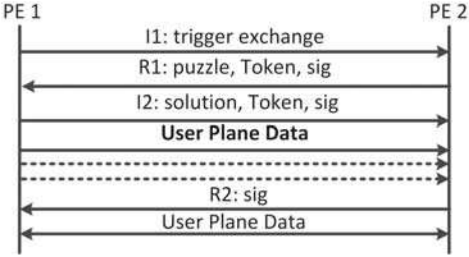


Fig. 69. The Fast Transmission Procedure (FTP).

Here, PE1 starts the transmission of user date after the I2 message. Therefore, the transmission delay will be reduced by 1RTT.

For already registered PEs, fast transmission can be utilized with TRP as well. The proposed FTP with TRP is illustrated in Figure 70.

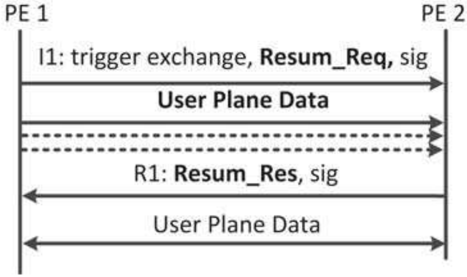


Fig. 70. The Fast Transmission Procedure with a Tunnel Resumption Procedure.

In this case, the PEs will not experience any tunnel establishment delay. The user data is transmitted because it is transmitting in a tunnel free environment. However , both PEs should be eligible for to use FTP with TRP.

6.4 Numerical results

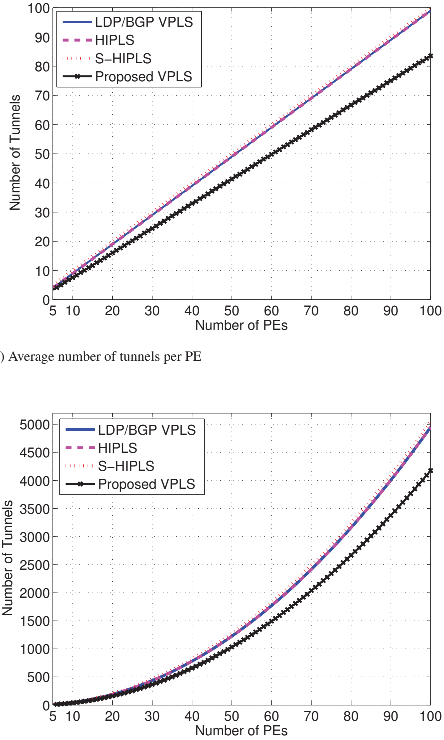
We simulated our architecture in an OMNET++ simulation environment [114] to compare the performance with other VPLS architectures. It was compared to secure flat VPLS architectures, namely HIPLS [72], S-HIPLS [21] and the most popular non secure MPLS architectures namely LDP [64] and BGP [63] based VPLS architectures. The simulation model established tunnels according to the tunnel management mechanism of each architecture. The simulation model contained a VPLS network with the equivalent of 100 PEs. The model network was generated by using stochastic Kronecker graphs [123]. The network bandwidth was set to 100 Mbps.

6.4.1 The average number of operational tunnels

The performance of the tunnel management mechanism depends on three factors; namely, the number of PEs in the network, the session duration and the session arrival rate of each tunnel. Here, we compare the performance by changing these factors.

The impact of the number of PEs

In the first experiment, we varied the number of PEs in the network. We set the mean session arrival rate (λ_a) as 1 session per minute and the mean session duration (λ_d) as 15 minutes [21, 150]. We simulated the arrival process as a Poisson process ($\lambda_a = 1$) and the session duration as an exponential distribution ($\lambda_d = 15$). Each test wass run for 100 instances. Figure 71 illustrates the simulation results.



(a) Average number of tunnels per PE

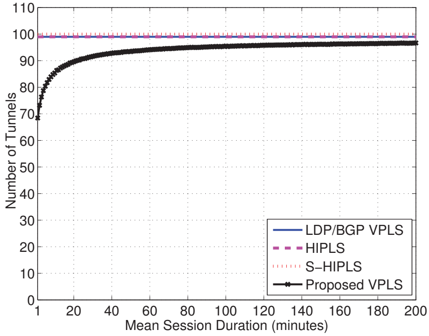
(b) Total number of tunnels in the network

Fig. 71. The impact of the number of PEs in the network([31] ©2016 IEEE).

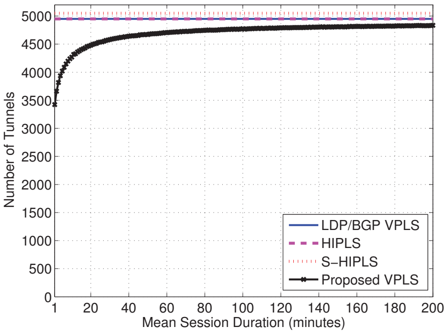
According to the simulation results (Figure 71), both the total number of tunnels in the network and the number of tunnels per PE increased with the number of PEs regardless of the architecture. As the number of PEs in the network increased, more VPN tunnels are required to interconnect them. Moreover, the proposed architecture reduces both the total number of tunnels in the network and the number of tunnels per PE compared to legacy VPLS architectures (HIPLS, S-HIPLS, LDP and BGP). The proposed architecture adjusts the tunnel duration according to the session duration and reduces idle tunnel operations. However, the performance of legacy VPLS architectures depends only on the number of PEs in the VPLS network.

The impact of session duration

In the second experiment, we changed the mean session duration (λ_d). We set the mean session arrival rate (λ_a) to 1 session per minute and used a network with 100 PEs. We simulated the arrival process as a Poisson process ($\lambda_a = 1$) and the ession duration as an exponential distribution (λ_d). Each test was run for 100 instances. Figure 72 illustrates the simulation results.



(a) Average number of tunnels per PE



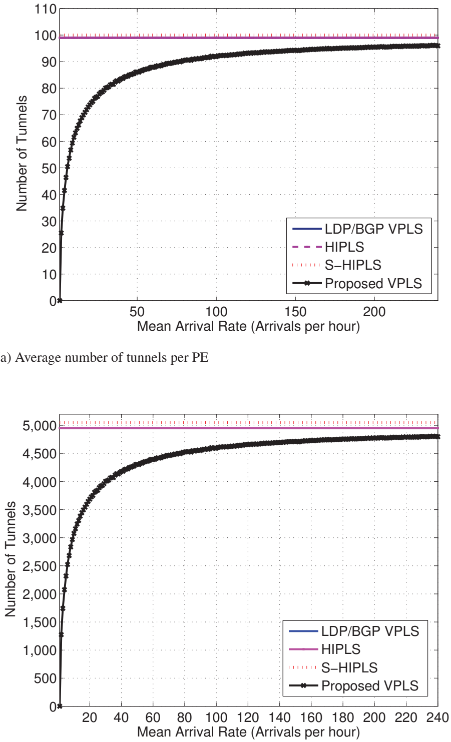
(b) Total number of tunnels in the network

Fig. 72. The impact of session duration([31] ©2016 IEEE).

The simulation results (Figure 72) verify that the proposed architecture reduces both the total number of tunnels in the network and the number of tunnels per PE compared to legacy VPLS architectures (HIPLS, S-HIPLS, LDP and BGP). It verifies that the proposed architecture dynamically adjusts the tunnel duration to minimize the average number of tunnels in the network by disconnecting inactive tunnels. The performance of legacy VPLS architectures is independent of the session duration.

The impact of arrival rate

In the third experiment, we changed the mean session arrival rate (λ_a). We set the mean session duration (λ_d) as 15 minutes and used a network with 100 PEs. We simulated the arrival process as a Poisson process (λ_a) and the session duration as an exponential distribution ($\lambda_d = 15$). Each test was run for 100 instances. Figure 73 illustrates the simulation results.



(a) Average number of tunnels per PE

(b) Total number of tunnels in the network

Fig. 73. The impact of the arrival rate([31] ©2016 IEEE).

The simulation results (Figure 73)) verify that the proposed architecture reduces both the total number of tunnels in the network and the number of tunnels per PE compared to legacy VPLS architectures (HIPLS, S-HIPLS, LDP and BGP). Moreover, the expected advantage of the proposed architecture is high when the arrival rate is low. The proposed architecture disconnects inactive tunnels quickly and avoids unnecessary tunnel maintenance operation. However, the performance of legacy VPLS architectures is independent of the arrival process.

The simulation results reveal that the expected advantage of the proposed architecture is higher with the fluctuation of highly dynamic factors, such as session duration and arrival rate than with static factors such as the number of PEs in the network. Therefore, the performance advantage of the Soft VPLS architecture is significant in highly dynamic VPLS networks.

6.4.2 Tunnel establishment delay

In the fourth experiment, we changed the propagation delay between two PEs and measured the tunnel establishment delay. Each test was run for 100 instances. Figure 74 illustrates the simulation results.

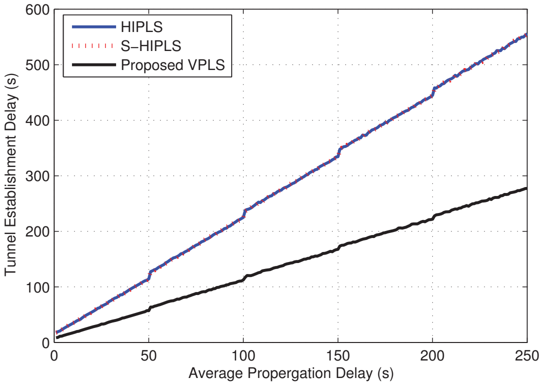
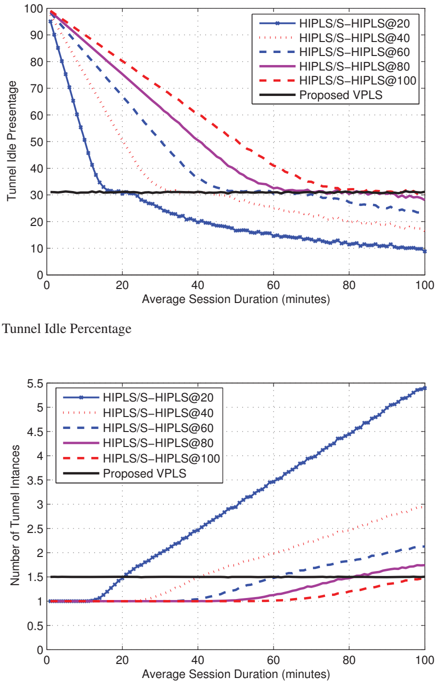


Fig. 74. Tunnel establishment delay([31] ©2016 IEEE).

The simulation results (Figure 74) verify that the proposed architecture significantly reduces the average tunnel establishment delay compared to legacy secure VPLS architectures (HIPLS and S-HIPLS). The fast tunnel resumption mechanism reduces the tunnel establishment delay of subsequence tunnel instances in the SoftVPLS architecture.

6.4.3 Tunnel idle percentage vs number of tunnel instances

In the fifth experiment, we measured the number of tunnel establishment instances and the tunnel idle percentage per active session compared to the average session duration (λ_d). We ran each experiment 100 times and average values are presented. Here, we considered five cases for HIPLS and S-HIPLS where the tunnel duration is predefined as 20, 40, 60, 80 and 100 minutes. Figure 75 illustrates the simulation results.



(a) Tunnel Idle Percentage

(b) Number of Tunnel Instances per Session

Fig. 75. The tunnel idle percentage vs number of tunnel instances([31] ©2016 IEEE).

The simulation results (Figure 75a) verify that the percentage of tunnel idle time in the proposed architecture is independent of the session duration. It reduces the tunnel idle time percentage to 30% under the utilized algorithm. However, performance can be further improved with better tunnel duration estimation algorithms.

On the other hand, the performance of legacy secure VPLS architectures highly depend on the session duration. Their performance is always lower than the proposed SoftVPLS architecture as long as the session duration is lower than the pre-defined tunnel duration. After that, the performance of legacy architectures is better than the SoftVPLS architecture.

The simulation results (Figure 75b also verify that the number of tunnel instances per session of the proposed architecture is independent of the session duration. This reduces the number of instances to 1.5 per session. Similarly, performance can be further improved with better tunnel duration estimation algorithms. However, the performance of legacy secure VPLS architectures depends highly on the session duration. Their performance was always lower than the proposed SoftVPLS architecture as long as the session duration was higher than the pre-defined tunnel duration.

Here, we considered the best possible scenario for legacy secure VPLS architectures, i.e. the impact of active sessions only. We did not consider the impact of inactive tunnel establishments. In other words, we examined the tunnel establishments of legacy VPLS architectures even without any active sessions. The performance of legacy architectures will be further reduced with these inactive tunnel establishments. However, SoftVPLS will still have better performance since it will not establish tunnels without any active traffic flows.

6.4.4 Average file transmission delay

A network with 100 PEs was used as our reference network. The model network was generated by using stochastic Kronecker graphs [123]. We compared the performance of FTM by integrating existing secure VPLS architectures, namely HIPLS [72], S-HIPLS [21] and SDN VPLS [31] architectures.

In this experiment, we measured the average waiting time before starting the user traffic transmission. We selected two PEs and gradually increased the RTT (Round Trip Time) between the PEs. We measured the waiting time at the session initiating PE before transmitting the user data. Figure 76 illustrates the simulation results.

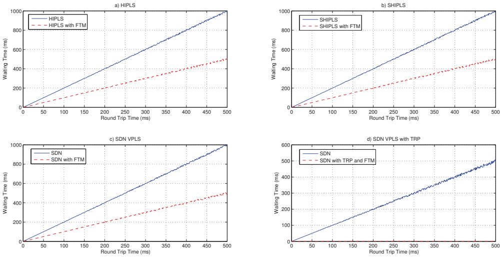


Fig. 76. The average waiting time before starting file transmission([31] ©2016 IEEE).

The simulation results (Figure 76) verify that proposed FTM has reduced the waiting time for all the architecture scenarios. The reduction of waiting by one RTT helped to achieve at-least 50% performance advantage in all scenarios.

6.5 The experiment testbed

The proposed architecture was implemented in a testbed to analyze the performance of the data plane. The experiment testbed is illustrated in Figure 77.

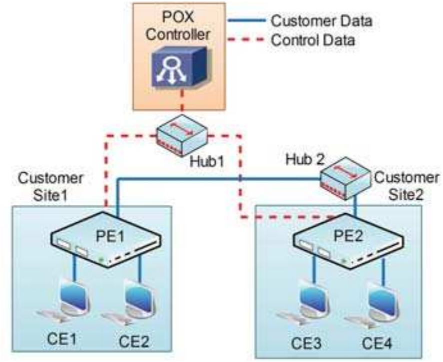


Fig. 77. The experiment testbed([31] ©2016 IEEE).

We used three laptops and two Ethernet hubs in the testbed. Two laptops with Intel i5-3210M CPU of 2.5GHz were used to implement PEs and customer sites. An OpenVswitch (OVS) version 1.10.0 [127] was installed on each of these laptops. Each OVS represents the PE for each customer site. Two CEs were connected to each PE and they ran the Lubuntu 13.10 Operating System (OS).

The third laptop with an L2400 CPU of 1.66GHz was used as the SDN controller. The latest POX controller [151] was run on this laptop. The POX controller used OpenFlow version 1.1.0[152] to control the PEs. The provider network was implemented by using two D-LINK DSR-250N routers and Ethernet links which had a bandwidth of 100 Mbps. Finally, we used an OpenHIP implementation [90] to establish IPsec tunnels between the PEs.

In our experiments, CE1 communicated with CE3 via the VPLS network. We measured the data plane throughput, jitter and tunnel establishment delay performance of the proposed architecture by using the IPERF network measurement tool [120] and Internet Control Message Protocol (ICMP) messages. Table 15 contains the simulation settings for the IPERF testing tool.

Table 15. The simulation settings for IPERF([31] ©2016 IEEE).

Parameter	Value	Value
Protocol	UDP	TCP
Port	5004	5004
Buffer size	default (1470 kB)	default (1470 kB)
Packet size	default (1470 B)	default (1470 B)
TCP window size	-	21.0 KByte
Report interval	1 s	1 s

We compared the data plane performance with non secure LDP architecture [64] and secure HIPLS architecture [72]. The experiment results (averages with confident intervals) are presented in Table 16.

Table 16. The performance comparison([31] ©2016 IEEE).

	TCP Throughput (Mbps)	UDP Throughput (Mbps)	Jitter (ms)	Tunnel Establishment Delay (ms)
LDP[64]	94.0567	96.2567	0.2981	-
HIPLS[72]	92.8054	94.3382	0.4025	80.5045
Proposed VPLS with TEP	92.8568	94.3134	0.4152	81.0947
Proposed VPLS with TRP	92.7874	94.9372	0.3975	45.1725

The data plane performance (throughputs, jitter and tunnel establishment delay) of proposed architecture is almost similar to secure HIPLS architecture. Furthermore, the utilization of the proposed SDN based VPLS architecture will not reduce the data plane performance of existing secure VPLS architectures. However, the proposed SoftVPLS architecture with TRP has significantly reduced (about 44% reduction) the tunnel establishment delay compared to the legacy VPLS architectures.

On the other hand, non secure LDP architecture performs better than both secure architectures. The extra layer of encryption and additional IPsec tunnel establishments decreased the performance of proposed architectures. However, Intel has illustrated IPsec acceleration by using external accelerators and/or using new AES instruction sets for processors [122]. Thus, the throughput and jitter performance of secure VPLS architectures can be further improved by utilizing these techniques.

In the second testbed experiment, we established communication sessions between CE1 and CE3 via the VPLS network. We measured the waiting time before transmitting the data. We compared the performance with other secure VPLS architectures, namely HIPLS [72], S-HIPLS [21] and SDN VPLS [31]. We ran the experiment 100 times and average values were calculated. The experiment results are presented in Table 17.

Table 17. The performance comparison([31] ©2016 IEEE).

	Average waiting time (ms)	Performance advantage of FTM
HIPLS[72]	80.6578	
HIPLS[72] with FTM	42.6752	47.0910%
SHIPLS [21]	81.3541	
SHIPLS [21] with FTM	43.1254	46.9905%
SDN VPLS [31]	80.5457	
SDN VPLS [31] with FTM	41.9552	47.9113%
SDN VPLS [31] with TRP	44.5646	
SDN VPLS [31] with TRP and FTM	1.8545	95.8386%

The experiment results verified that the proposed FTM reduced the waiting time of existing VPLS architectures by 46% to 47%. Moreover, FTM with TRP had almost zero waiting time. In this case, user data transmission can be started right after the transmission of the first tunnel establishment message (i.e. I1 in Figure 70).

6.6 Summary and discussion

We proposed a novel SDN based VPLS architecture to overcome the tunnel management limitations of legacy secure VPLS architectures. The proposed architecture utilizes IPsec enabled OpenFlow switches as PEs and the OpenFlow protocol to install flow rules in PEs. We further proposed a dynamic tunnel management mechanism which estimates the tunnel duration based on real time network statistics. Moreover, novel tunnel resumption and fast transmission mechanisms are proposed to reduce the tunnel delay of subsequent tunnel establishments and delays in overall data transmission.

The performance of the proposed tunnel management mechanism depends not only the number of PEs in the network but also on the session arrival rate and session duration of each tunnel. As a result, the proposed architecture reduces the average number of tunnels per PE and the total number tunnels in the network compared to secure legacy VPLS architectures. It also significantly improves the control and forwarding plane scalability. Furthermore, simulation results revealed that the expected advantage of the proposed architecture is higher with the variation of highly dynamic factors, such as the session duration and the arrival rate than with static factors such as the number of PEs in the network.

Moreover, the proposed architecture was implemented on a test bed and the experiment results showed that data plane performance (throughputs, jitter and tunnel establishment delay) of the proposed architecture is similar to existing secure VPLS architectures. This also verifies that the proposed SDN based architecture will not reduce the data plane performance of existing secure VPLS architectures. However, the proposed SoftVPLS architecture with a tunnel resumption mechanism had significantly reduced (about 44% reduction) the tunnel establishment delay compared to legacy VPLS architectures.

7 Conclusions and future directions

In this chapter, we provide the conclusion of this thesis. We also present several future directions through which the proposed architectures may further be developed for future communication networks.

7.1 Conclusions

The purpose of this thesis has been to enhance the scalability and security of VPLS networks to be used in future communication networks. The aim of Chapters 3, 4 and 6 was to propose new scalable and secure VPLS architectures. Having realized the need to support compatibility for legacy L2 protocols in new VPLS networks, Chapter 5 proposed necessary modifications to existing L2 protocols, so that they are able to work in VPLS networks. In the following, we conclude the contribution of each chapter.

In Chapter 3, we proposed a scalable and secure flat-VPLS architecture based on a Host Identity Protocol (HIP). By proposing a session key-based security mechanism and an efficient broadcast mechanism, our architecture comparatively reduced the complexity of key storage at a node and the overall key storage of the network. Further, the new mechanism also reduced the number of encryptions per broadcast frame and offered a higher degree of security features than other existing secure VPLS architectures. These features increased the forwarding and security plane scalability of the proposed VPLS architecture. The proposed architecture provides a faster and low processing tunnel generation phase than existing secure VPLS architectures as it omits the extensive Diffie-Hellman (D-H) key exchange during tunnel generation between PEs (Provider Edge Equipment). The additional certificate mechanism and separate VPNs for control plane traffic provided extra security for important system entities from attackers. As a result, the proposed architecture offers higher degrees of security than other existing secure VPLS architectures. However, the utilization of an extra control entity called a KDC (Key Distribution Center) and provider VPNs add slight burdens to the VPLS network. Nevertheless, this is not greatly significant as long as the number of PEs is higher than the number of provider VPNs in the network. In addition, a distributed KDC architecture was proposed to decentralize key storage and to solve the issues related to KDC. We ran simulations to analyze the performance of the proposed architecture and to confirm the security features. The simulation results further confirmed that our proposal was able to protect the control protocol of the VPLS from the Internet Protocol(IP)/Transmission Control Protocol(TCP) based attacks. Moreover, a test bed implementation was used to analyze the data plane performance of existing secure VPLS architectures. The performance penalty of security on throughput was about 20% for both UDP and TCP sessions compared to both non VPLS scenario and a non-secure VPLS architecture. Moreover, jitter of the secure VPLS architecture is two times higher than the non-secure and non-VPLS scenarios. The additional layer of encryption was the main reason for the reduced average throughput of the secure VPLS architecture. Moreover, the secure VPLS architecture increased the latency approximately by 87% due to encryption and tunneling delays in PE devices. However, IPsec acceleration can be achieved by using external accelerators and/or using new AES (Advanced Encryption Standard) instruction sets for the processors. Thus, the adaptation of these techniques for PEs will improve the performance of secure VPLS networks.

In Chapter 4, we proposed a scalable and secure hierarchical-VPLS architecture based on HIP. This increased the scalability of the previously proposed flat-VPLS architecture by providing control plane scalability. A novel encrypted label based secure frame forwarding mechanism was proposed to transport L2 frames over the hierarchical VPLS network. The proposed VPLS architecture increased the control and forwarding plane scalability more than other secure VPLS architectures by significantly reducing the total number of tunnels in the network, as well as the complexity of the key storage at a node, in addition to the total key storage of the network, and the number of encryptions per broadcast frame. Moreover, it had the same level of control and forwarding plane scalability as other unsecured hierarchical VPLS architectures. The proposed VPLS architecture also provided similar levels of security as other secure VPLS architectures. The control protocol of the proposed architecture was protected from IP based attacks, such as TCP SYN DoS and TCP reset attacks. The proposed architecture distributed the service provision among different PE functions to minimize the utilization of network resources, such as memory space and processing power at the PEs. We numerically analyzed the scalability limits of the proposed architecture. Then, we conducted extensive simulations to verify the findings. The simulation model was also used to confirm the security features. Finally, the proposed architecture was implemented in a real world test bed and the performance was compared with other VPLS architectures. The experiment results revealed that the proposed architecture has almost similar throughput performance to other secure VPLS architectures. It had only 2% less throughput than unsecured hierarchical VPLS architectures and only 3% higher latency than other secure VPLS architectures. However, the proposed architecture had higher latency than insecure hierarchical VPLS architectures. This performance penalty had occurred due to the extra label encryptions used for the network facing PEs.

In a VPLS network, connections through the provider network are invisible to the switching network devices and L2 protocols. However, these transparent links in the provider network cause many negative effects on L2 protocols such as STP (Spanning Tree Protocol). Hence, it is infeasible to implement the existing L2 protocols in a VPLS enabled Ethernet network. In Chapter 5, we proposed a novel Distributed STP (DSTP) to maintain a loop free Ethernet network over a VPLS network. Using this DSTP we proposed to utilize a modified STP instance in each remote segment of the VPLS network. Furthermore, we proposed two Redundancy Identification Mechanisms (RIMs) called Customer Associated RIM (CARIM) and Provider Associated RIM (PARIM) to prevent functional issues which may arise due to invisible loops in the provider network. We conducted several simulations to verify these features and illustrated the performance advantages of DSTP. DSTP successfully transmitted broadcast frames over the VPLS network without causing any broadcast storms. Furthermore, DSTP significantly reduced the convergence time of the spanning tree and STP overhead of the provider network by outperforming existing STP versions. DSTP also increased the scalability by significantly reducing the number of STP messages transmitted through the provider network in a large scale network. Ultimately, it reduced the additional overhead on the provider network and STP related costs for the customer. Thus, DSTP was capable of establishing and maintaining a loop free Ethernet network over a VPLS network by solving existing STP incompatibility.

New VPLS applications demand additional requirements such as elevated security, enhanced scalability, optimum utilization of network resources and further reduction in operational costs. Although the existing secure VPLS architectures are able to provide a sufficient level of security, they all still suffer from limitations such as low scalability, over utilization of network resources, high latency tunnel establishments and high operational costs. In Chapter 6, we proposed a novel SDN based VPLS architecture to overcome the tunnel management limitations of legacy secure VPLS architectures. The proposed architecture utilized IPsec enabled OpenFlow switches as PEs and used an Open-Flow protocol to install flow rules in the PEs. We proposed a dynamic tunnel management mechanism which estimates the tunnel duration based on real time network statistics. Moreover, novel tunnel resumption and fast transmission mechanisms were proposed to reduce the tunnel establishment delay of subsequent tunnel establishments and overall data transmission delays. The tunnel management procedures of todayŠs secure VPLS architectures depend only on the number of PEs in the network. However, the proposed architecture dynamically adjusts the tunnel duration by analyzing the traffic patterns of each tunnel. Therefore, the performance of the proposed tunnel management mechanism depends not only the number of PEs in the network but also on the session arrival rate and session duration of each tunnel. As a result, the proposed architecture reduces the average number of tunnels per PE and the total number of tunnels in the network compared to legacy secure VPLS architectures. It significantly improves control and forwarding plane scalability. Furthermore, simulation results revealed that the expected advantage of the proposed architecture was higher when there was variation in highly dynamic factors, such as the session duration and the arrival rate than there was with static factors, such as the number of PEs in the network. Finally, the proposed architecture was implemented in a test bed and the experiment results showed that the data plane performance (throughputs, jitter and tunnel establishment delay) of the proposed architecture was similar to existing secure VPLS architectures. It also verified that the proposed SDN based architecture will not reduce the data plane performance of existing secure VPLS architectures. However, the proposed SoftVPLS architecture with the tunnel resumption mechanism had significantly reduced (about 44% reduction) tunnel establishment delays compared to legacy VPLS architectures.

7.2 Discussion

The main draw back of all secure VPLS architectures (i.e. HIPLS, S-HIPLS and H-HIPLS) is the extra layer of encryption. It increases the end-to-end latency and reduces the overall throughput. None of our proposals were able to address this issue. However, it is possible to reduce this delay by using IPsec acceleration [122]. Therefore, it is important to design IPsec friendly PEs for VPLS networks. Another drawback of our proposals (S-HIPLS and H-HIPLS) is requirement of HIP enabled PEs. Such PEs need to support IPsec encryption and it increases the cost of PEs. However, most the currently available secure VPLS products are developed based on HIP. The positive involvement for industry will lead to an eventual cost reduction of HIP based PEs.

In contrast to HIPLS, our proposals (S-HIPLS and H-HIPLS) use additional Authentication Server (AS)/Key Distribution Center (KDC) in the VPLS network. It increases the vulnerability of whole VPLS network since AS/KDC is the heart of the encryption key distribution. Therefore, many attacker are targeting this entity. To solve this issue, additional security precautions (e.g. Firewalls, Security Gateways) are used for KDC/AS. Moreover, distributed AS/KDC architectures are proposed to increase resilience of AS/KDC entity. Moreover, distributed AS/KDC architecture minimize the impact of additional key storage requirement at AS/KDC by distributing the keys among several servers.

In H-HIPLS, additional use of n-PEs to interconnect u-PEs increases the total number of PEs in the network. It increases the implementation and operational cost of VPLS network. However, the service distribution of proposed H-HIPLS significantly reduces the workload of mostly used u-PEs. Thus, the VPLS network can be implemented with low cost u-PEs and medium cost n-PEs. Such implementation might reduce the implementation cost of H-HIPLS for some extend.

The proposed S-HIPLS and H-HIPLS architectures are still vulnerable to IP based attacks such as volume based DoS attacks (e.g. UDP floods, ICMP floods and other spoofed-packet floods). In volume based DoS attacks, the attackers tried to overload the network bandwidth by injecting massive amount of junk traffic. Most types of communication networks are facing to these DoS attacks [132] and they can be easily prevented by implementing firewalls, ingress filtering and enforcing rate bounds [132, 133]. Therefore, it is necessary to implement these security mechanisms together with proposed VPLS architectures.

Similarly to other VPLS architectures, the propose architectures (S-HIPLS and H-HIPLS) also have static tunnel maintenance duration which is predefined by the network administrators. These VPLS architectures do not support dynamic parameter adjustment for tunnels. Static tunnel maintenance between infrequently communicating sites not only wastes a PEsŠ resources such as memory, CPU and ports, but also wastes the network bandwidth for tunnel update messages. Therefore, it is necessary to fine tune the tunnel duration based on traffic demand between site. The proposed Soft-VPLS architecture solves this drawback by using SDN (Software Defined Networking) concept.

The proposed DSTP mechanism is able to solve the problems related to STP only. However, it is not a viable solutions to solve the performance and compatibility issues in other layer 2 protocols such as ARP (Address Resolution Protocol), RARP (Reverse ARP) while using them in a VPLS based Ethernet network. It is necessary to identify correlations between these issues and develop a common platform to implement all layer 2 protocols in a VPLS enabled Ethernet environment.

7.3 Future directions

This thesis is a step towards enhancing the scalability and security of future VPLS networks. We list several possible directions researchers may explore on this topic.

- Our test beds in Chapter 3 and 4 analyzed the performance of existing VPLS architecture spanned only over a WLAN. Expanding our test bed experiments beyond LANs and examining areas such as long-haul and telecommunication networks would be a promising area of future research.
- The IPsec acceleration can be achieved by using external accelerators and/or using new AES (Advanced Encryption Standard) instruction sets for processors. The adaptation of such techniques for PEs would improve the performance of secure VPLS networks and thus would be a promising area of research.
- Mobile devices are popular in many networks. For instance, VPLS solutions can be used for secure intra/inter-vehicular communication systems. Therefore, it would be useful to study the impact of mobile PEs on secure VPLS networks.
- We introduced new control entities (i.e. KDC, SME and SDN controllers) to secure VPLS networks. It would be useful to identify the optimum number of entities and the location for each entity. One other possible area of research would be to develop resilience control entity architecture.
- We proposed the use of u-PE and n-PE in hierarchical secure VPLS networks. It would be important to study the optimum ratio between different types of PEs. One other possible research area would be to develop load balancing mechanisms for different types of PEs.
- In this thesis, we have discussed the compatibility issues related to STP only. It would also be important to study the performance and compatibility issues in other L2 protocols, such as ARP (Address Resolution Protocol) and RARP (Reverse ARP) while using them in a VPLS based Ethernet network. By studying various L2 Protocols, it would be possible to identify correlations between these elements and explore common solutions. The ultimate goal should be to develop a common platform to implement all L2 protocols in a VPLS enabled Ethernet environment.
- The proposed SDN based VPLS architecture is used to enhance the performance of the tunnel management mechanism only. Thus, it did not improve the data plane performance (e.g. throughput, latency or jitter) of secure VPLS architectures. Hence, it would be fruitful to study the utilization of SDN to improve the data plane performance of secure VPLS architectures. Moreover, it should be possible to

use SDN concepts to improve other VPLS features such as security and mobility management, in addition to path optimization.

- On the one hand, the introduction of SDN and NFV concepts for VPLS networks helps to increase security features. On the other hand, it may introduce new security threat vectors and security challenges. A security analysis of SDN based VPLS architecture would be very important.

References

1. Scott C, Wolfe P & Erwin M (1999) Virtual Private Networks. " O'Reilly Media, Inc.".
2. Chowdhury N & Boutaba R (2010) A Survey of Network Virtualization. Computer Networks 54(5): 862–876.
3. Olifer V *et al.* (2007) Different Flavours of VPN: Technology and Applications. JNT Association.
4. Metz C (2003) The Latest in Virtual Private Networks: Part I. Internet Computing, IEEE 7(1): 87–91.
5. Metz C (2004) The Latest in VPNs: Part II. Internet Computing, IEEE 8(3): 60–65.
6. Venkateswaran R (2001) Virtual Private Networks. Potentials, IEEE 20(1): 11–15.
7. Knight P & Lewis C (2004) Layer 2 and 3 Virtual Private Networks: Taxonomy, Technology, and Standardization Efforts. Communications Magazine, IEEE 42(6): 124–131.
8. Harris S (2002) IP VPNsï£¡ An Overview for Network Executives. Technical report. URI: `http://www.onsiteaustin.com/whitepapers/VPN20justfication.pdf`.
9. Dong X & Yu S (2005) VPLS: An Effective Technology for Building Scalable Transparent LAN Services. In: Asia-Pacific Optical Communications, pp. 137–147. International Society for Optics and Photonics.
10. Boyer SA (2009) SCADA: Supervisory Control and Data Acquisition. International Society of Automation.
11. Luo W, Pignataro C, Chan A & Bokotey D (2004) Layer 2 VPN Architectures. Pearson Education.
12. Guichard J, Pepelnjak I & Apcar J (2003) MPLS and VPN Architectures, volume 2. Cisco Press.
13. Pompei S, Teodori M, Valenti A, Di Bartolo S, Incerti G & Del Buono D (2010) Experimental Implementation of an IPTV Architecture based on Content Delivery Network Managed by VPLS Technique. In: Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on, pp. 576–581. IEEE.
14. (2008) Virtual Private LAN Service (VPLS) Technical Primer for Government Agencies . Technical report, Alcatel-Lucent Inc. URI: `http://www3.alcatel-lucent.com/solutions/mpls4ips/docs/VPLS_Tech_govt_agc_twp.pdf`.
15. Liyanage M & Gurtov A (2014) Securing Virtual Private LAN Service by Efficient Key Management. Security and Communication Networks 7(1): 1–13.
16. Namal S, Liyanage M & Gurtov A (2013) Realization of Mobile Femtocells: Operational and Protocol Requirements. Wireless personal communications 71(1): 339–364.
17. Liyanage M, Ylianttila M & Gurtov A (2015) Secure Hierarchical VPLS Architecture for Provider Provisioned Networks. Access, IEEE 3: 967–984.
18. Liyanage M, Abro A, Ylianttila M & Gurtov A (2015) Opportunities and Challenges of Software-Defined Mobile Networks in Network Security Perspective. IEEE Security and Privacy Magazine .
19. Liyanage M, Kumar P, Ylianttila M & Gurtov A (2016) Novel Secure VPN Architectures for LTE Backhaul Networks. Security and Communication Networks .
20. Liyanage M & Gurtov A (2012) Secured VPN Models for LTE Backhaul Networks. In: Vehicular Technology Conference (VTC Fall), 2012 IEEE, pp. 1–5. IEEE.

21. Liyanage M & Gurtov A (2013) A Scalable and Secure VPLS Architecture for Provider Provisioned Networks. In: IEEE Wireless Communication and Networking Conference: WCNC 2013. IEEE.
22. Liyanage M, Ylianttila M & Gurtov A (2013) Secure Hierarchical Virtual Private LAN Services for Provider Provisioned Networks. In: Communications and Network Security (CNS), 2013 IEEE Conference on, pp. 233–241. IEEE.
23. Liyanage M, Ylianttila M & Gurtov A (2014) A Novel Distributed Spanning Tree Protocol for Provider Provisioned VPLS Networks. In: IEEE Conference on Communications: ICC 2014. IEEE.
24. Liyanage M, Chirkova J & Gurtov A (2014) Access Point Selection Game for Mobile Wireless Users. In: The 8th IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications. IEEE.
25. Liyanage M, Ylianttila M & Gurtov A (2014) Securing the Control Channel of Software-Defined Mobile Networks. In: A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on, pp. 1–6. IEEE.
26. Liyanage M, Okwuibe J, Ylianttila M & Gurtov A (2015) Secure Virtual Private LAN Services: An Overview with Performance Evaluation. In: IEEE ICC 2015 - Workshop on Advanced PHY and MAC Techniques for Super Dense Wireless Networks, pp. 10297–10303. IEEE.
27. Costa-Requena J, Llorente Santos J, Ferrer Guasch V, Ahokas K, Premsankar G, Luukkainen S, Ahmad I, Liyanage M, Ylianttila M, Loï£¡pez Peï£¡rez O *et al.* (2015) SDN and NFV Integration in Generalized Mobile Network Architecture. In: Networks and Communications (EuCNC), 2015 European Conference on, pp. 154–158. IEEE.
28. Liyanage M, Ahmed I, Ylianttila M, Santos JL, Kantola R, Perez OL, Itzazelaia MU, Oca EMd, Valtierra A & Jimenez C (2015) Security for Future Software Defined Mobile Networks. In: Next Generation Mobile Applications, Services and Technologies, 2015 9th International Conference on, pp. 256–264. IEEE.
29. Okwuibe J, Liyanage M & Ylianttila M (2015) Performance Analysis of Open-Source Linux-Based HIP Implementations .
30. Liyanage M, Ahmad I, Ylianttila M, Gurtov A, Abro AB & de Oca EM (2015) Leveraging LTE Security with SDN and NFV. In: IEEE 10th International Conference on Industrial and Information Systems (ICIIS). IEEE.
31. Liyanage M, Gurtov A & Ylianttila M (2016) Improving the Tunnel Management Performance of Secure VPLS Architectures with SDN. Proc. of IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA. IEEE .
32. Ahmad I, Liyanage M, Namal S, Ylianttila M, Gurtov A, Eckert M, Bauschert T, Faigl Z, Bokor L, Saygun E, Akyildiz HA, Perez OL, Itzazelaia MU, Ozbek B & Ulas A (2016) New Concepts for Traffic, Resource and Mobility Management in Software-Defined Mobile Networks. Proc. of 12th Wireless On-demand Network systems and Services Conference (WONS), Cortina d'Ampezzo, Italy. IEEE .
33. Liyanage M, Kumar P, Soderi S, Ylianttila M & Gurtov A (2016) Performance and Security Evaluation of Intra-Vehicular Communication Architecture. Proc. of IEEE ICC 2016 - Workshop on Convergent Internet of Things (Convergent IoT), Kuala Lumpur, Malaysia. IEEE .
34. Okwuibe J, Liyanage M & Ylianttila M (2016) Provider Assisted Wi-Fi Offloading Leveraging on SDN. Proc. of 22nd European Wireless conference, Oulu, Finland .

35. Liyanage M, Kumar P & Gurtov A (2015). Zone-based Security Architecture for Intra-Vehicular Wireless Communication.
36. Liyanage M, Gurtov A & Ylianttila M (2015) Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture. John Wiley & Sons.
37. Liyanage M, Ylianttila M & Gurtov A (2014) A Case Study on Security Issues in LTE Backhaul and Core Networks. Case Studies in Secure Computing: Achievements and Trends p. 167.
38. Liyanage M, Ylianttila M & Gurtov A (2014) IP-Based Virtual Private Network Implementations in Future Cellular Networks. Handbook of Research on Progressive Trends in Wireless Communications and Networking p. 44.
39. Zhang N, Lehmusvuori J, Liyanage M, Kantola R, salo J, Bauschert T, Vince Z, Ulaş A & Costa J (2016). Software-Defined and Virtualized Mobile Networks.
40. (2012-2013) Mobile Networks Evolution for Individual Communications Experience (MEVICO). Technical report. URI: `http://www.mevico.org/`.
41. (2013-2016) SDN Concept in Generalized Mobile Network Architectures (SIGMONA). Technical report. URI: `http://www.sigmona.org/`.
42. (2013) Train Wireless Bus, Wireless solutions for urban transit environments (TWB). Technical report. URI: `http://www.ee.oulu.fi/~agourtov/projects.html`.
43. (2015-2018) Nordic perspective to gadget-free hyperconnected environments (The Naked Approch). Technical report. URI: `http://nakedapproach.fi/`.
44. (2015-2017) Algorithms, Architectures and Platforms for Enhanced Living Environments (AAPELE). Technical report. URI: `http://www.cost.eu/COST_Actions/ict/IC1303`.
45. (2016-2020) Inclusive Radio Communication Networks for 5G and beyond (IRACON). Technical report. URI: `http://www.cost.eu/COST_Actions/ca/CA15104`.
46. (2015-2017) Wireless Power Transmission for Sustainable Electronics (WiPE). Technical report. URI: `http://www.cost.eu/COST_Actions/ict/IC1301`.
47. (2016-2020) Resilient communication services protecting end-user applications from disaster-based failures (RECODIS). Technical report. URI: `http://www.cost.eu/COST_Actions/ca/CA15127`.
48. Rosenbaum G, Lau W & Jha S (2003) Recent Directions in Virtual Private Network Solutions. In: Networks, 2003. ICON2003. The 11th IEEE International Conference on, pp. 217–223. IEEE.
49. Provider Provisioned Virtual Private Networks (PPVPN). Technical report. URI: `http://datatracker.ietf.org/wg/ppvpn/charter/`.
50. Layer 2 Virtual Private Networks (L2VPN). Technical report. URI: `http://datatracker.ietf.org/wg/l2vpn/`.
51. Layer 3 Virtual Private Networks (L3VPN). Technical report. URI: `http://datatracker.ietf.org/wg/l3vpn/`.
52. Rosenbaum G, Lau W & Jha S (2003) An Analysis of Virtual Private Network Solutions. provider 2: L3VPNs.
53. Carugi M & De Clercq J (2004) Virtual private network services: Scenarios, requirements and architectural constructs from a standardization perspective. Communications Magazine, IEEE 42(6): 116–122.
54. Daniel A (2004) IP Virtual Private Networks - A Service Provider Perspective. In: Communications, IEE Proceedings-, volume 151, pp. 62–70. IET.

55. Callon R & Suzuki M (2005). A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs). RFC 2401.
56. Fang L (2005) Security Framework for Provider-Provisioned Virtual Private Networks (PPVPNs) .
57. Clercq J, Paridaens O, Iyer M & Krywaniuk A (2001) A Framework for Provider Provisioned CE-based Virtual Private Networks using IPsec. Technical report.
58. Gu R, Dong J, Chen M, Zeng Q & Liu Z (2011). Analysis of Virtual Private LAN Service (VPLS) Deployment. *Internet Draft*.
59. WARNOCK G (2011) Alcatel-Lucent Network Routing Specialist II (NRS II) Self-Study Guide: Preparing for the NRS II Certification Exams Self-Study Guide (paperback) .
60. Sofia R (2009) A Survey of Advanced Ethernet Forwarding Approaches. Communications Surveys & Tutorials, IEEE 11(1): 92–115.
61. Andersson L & Rosen E (2006). Framework for Layer 2 Virtual Private Networks (L2VPNs). RFC 4664.
62. Augustyn W & Serbest Y (2006). Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks. RFC 4665.
63. Kompella K & Rekhter Y (2007). Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling. RFC 4761.
64. Lasserre M & Kompella V (2007). Virtual Private LAN Service (VPLS) using Label Distribution Protocol (LDP) Signaling. RFC 4762.
65. H Shah ER & Heron G (2007). IP-Only LAN Service (IPLS). *Internet Draft*.
66. (2011) H-VPLS N-PE Redundancy for QinQ and MPLS Access. Technical report, CISCO Cooperation. URI: `http://www.cisco.com/en/US/docs/ios/mpls/configuration/guide/`.
67. (2010) Demystifying H-VPLS. Technical report, Juniper Networks, Inc. URI: `http://www.juniper.net/us/en/local/pdf/app-notes/3500116-en.pdf`.
68. Khandekar S, Kompella V, Regan J, *et al.* (2002). Hierarchical Virtual Private LAN Service. *Internet Draft*.
69. Sodder A, Ramakrishnan K, DelRegno C, & Wils J (2003). Virtual Hierarchical LAN Services. *Internet Draft*.
70. Hu C, Yuan C, Liu K *et al.* (2009). Enhanced H-VPLS Service Architecture using Control Word. US Patent 7,570,648.
71. Zelig D, Bruckman L & Kotser Y (2007). Hierarchical Virtual Private LAN Service Protection Scheme. US Patent 7,283,465.
72. Henderson T, Venema S & Mattes D (2011). HIP-based Virtual Private LAN Service (HIPLS).
73. Gurtov A (2008) Host Identity Protocol (HIP): Towards the Secure Mobile Internet. Wiley.
74. Atkinson R (1995). Security Architecture for the Internet Protocol. RFC 1825.
75. Karn P, Metzger P & Simpson W (1995). The ESP DES-CBC Transform. RFC 1829.
76. Kent S & Atkinson R (1998). Security Architecture for the Internet Protocol. RFC 2401.
77. Orman H (1998). The OAKLEY Key Determination Protocol. RFC 2412.
78. Harkins D & Carrel D (1998). The Internet Key Exchange (IKE). RFC 2409.
79. Kaufman C (2005). Internet Key Exchange (IKEv2) Protocol. RFC 4306.
80. Eronen P (2006). IKEv2 Mobility and Multihoming Protocol (MOBIKE). RFC 4555.
81. Jokela P, Moskowitz R & Nikander P (2008). Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP). RFC 5202.

82. Kent S & Header I (2005). IP Authentication Header. RFC 4302.
83. Kent S (2005). IP Encapsulating Security Payload (ESP). RFC 4303.
84. Vaarala S & Klovning E (2008). Mobile IPv4 Traversal Across IPsec-based VPN Gateways. RFC 5265.
85. Freier A, Karlton P & Kocher P (2011). The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101.
86. Dierks T & Rescorla E (2008). The Transport Layer Security (TLS) Protocol. RFC 5246.
87. Moskowitz R, Nikander P & Jokela P (2008). Host Identity Protocol. RFC 5201.
88. Nikander P, Gurtov A & Henderson T (2010) Host Identity Protocol (HIP): Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks. Communications Surveys & Tutorials, IEEE 12(2): 186–204.
89. (2015). Infrahiphip: About infrahip. `http://infrahip.hiit.fi/index.php?index=about`.
90. (2014). Openhip: About openhip. `http://www.openhip.sourceforge.net/about.html`.
91. Moskowitz R, Heer T, Jokela P & Henderson T (2015) Host Identity Protocol Version 2 (HIPv2). Technical report.
92. Jokela P, Melen J & Moskowitz R (2015) Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP) .
93. T H Boeing HIP Secure Mobile Architecture. [online] .
94. GE Transportation. `http://www.getransportation.com/its`.
95. Tofino Security Appliance. `http://www.tofinosecurity.com/products/tofino-security-appliance`.
96. Asguard Networks. [Online]. Available:. `http://www.asguardnetworks.com/`.
97. Tempered Networks. `http://www.temperednetworks.com/products`.
98. DII-HEP (CMS) cluster. `http://www.nordugrid.org/monitor/`.
99. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S & Turner J (2008) OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review 38(2): 69–74.
100. Kolias C, Ahlawat S, Ashton C, *et al.* (2013). OpenFlow-Enabled Mobile and Wireless Networks. *White Paper*.
101. Hawilo H, Shami A, Mirahmadi M & Asal R (2014) NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC). Network, IEEE 28(6): 18–26.
102. Martins J, Ahmed M, Raiciu C, Olteanu V, Honda M, Bifulco R & Huici F (2014) ClickOS and the Art of Network Function Virtualization. In: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, pp. 459–473. USENIX Association.
103. Mijumbi R, Serrat J, Gorricho JL, Bouten N, De Turck F & Boutaba R (2015) Network Function Virtualization: State-of-the-Art and Research Challenges .
104. Jain R & Paul S (2013) Network Virtualization and Software Defined Networking for Cloud Computing: A Survey. Communications Magazine, IEEE 51(11): 24–31.
105. Martini L, Rosen E, El-Aawar N, Smith T & Heron G (2006). Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP). RFC 4447.
106. Martini L, Rosen E, El-Aawar N & Heron G (2006). Encapsulation Methods for Transport of Ethernet over MPLS Networks. RFC 4448.

107. Martini L, Rosen E, Heron G & Malis A (2006). Encapsulation Methods for Transport of PPP/High-Level Data Link Control (HDLC) over MPLS Networks. RFC 4618.
108. Stein Y (2006). Pseudowire Security (PWsec). *Internet Draft*.
109. Henderson T Boeing HIP Secure Mobile Architecture. Technical report. URI: `http://www.ietf.org/proceedings/73/slides/HIPRG-0.pdf`.
110. Tofino Security Appliance. Technical report. URI: `http://www.tofinosecurity.com/products/tofino-security-appliance`.
111. Tempered networks. Technical report. URI: `http://www.temperednetworks.com/`.
112. Kuptsov D, Khurri A & Gurtov A (2009) Distributed User Authentication in Wireless LANs. In: World of Wireless, Mobile and Multimedia Networks & Workshops. IEEE.
113. (2010) Architectural Considerations for Backhaul of 2G/3G and Long Term Evolution Networks. Technical report, CISCO Cooperation.
114. Varga A (2001) The OMNeT++ Discrete Event Simulation System. In: Proceedings of the European Simulation Multiconference (ESM-2001).
115. Rekhter Y, Li T & Hares S (2006). A Border Gateway Protocol 4 (BGP-4). RFC 4271.
116. Andersson L, Doolan P, Feldman N, Fredette A & Thomas B (2001). LDP Specification. RFC 3036.
117. Eddy W (2007). TCP SYN flooding attacks and common mitigations. RFC 4987.
118. Keller G & Beylot A (2008) Improving Flow Level Fairness and Interactivity in WLANs using Size-based Scheduling Policies. In: Proc. of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM). ACM.
119. Alvarez MA, Jounay F, Major T & Volpato P (2011) LTE Backhauling Deployment Scenarios. Technical report, Next Generation Mobile Networks Alliancen.
120. Iperf. Technical report. URI: `http://iperf.sourceforge.net/`.
121. Wijesinha AL, Song Yt & et al (2005) Throughput Measurement for UDP Traffic in an IEEE 802.11 g WLAN. In: First ACIS International Workshop on Self-Assembling Wireless Networks, SNPD/SAWN 2005. IEEE.
122. (2013) Carrier Cloud Telecoms - Exploring the Challenges of Deploying Virtualisation and SDN in Telecom Networks. Technical report, Intel Cooperation.
123. Leskovec J, Chakrabarti D, Kleinberg J, Faloutsos C & Ghahramani Z (2010) Kronecker Graphs: An Approach to Modeling Networks. The Journal of Machine Learning Research 11: 985–1042.
124. Cisco ASR 9001 Router Data Sheet. Technical report, CISCO. URI: `{http://www.cisco.com/c/en/us/products/collateral/routers/asr-9001-router/data_sheet_c78-685687.html}`.
125. Watson PA (2004) Slipping in the Window: TCP Reset attacks. Technical report.
126. Behringer MH & Morrow M (2005) MPLS VPN Security. Cisco Press.
127. Open vSwitch: An Open Virtual Switch. Technical report. URI: `\url{http://openvswitch.org/}`.
128. DSR-250N Services Router. URI: `http://www.dlink.com/us/en/business-solutions/security/services-routers/dsr-250n-wireless-n-unified-service-router`.
129. DIR-615 Wireless N300 Router. URI: `http://us.dlink.com/products/connect/wireless-n300-router/`.
130. Oppliger R (2009) SSL and TLS: Theory and Practice. Artech House.

131. Quality of Service (QoS) concept and architecture. URI: http://www.3gpp.org/dynareport/23107.htm.

132. Chang RK (2002) Defending Against Flooding-based Distributed Denial-of-Service Attacks: A Tutorial. Communications Magazine, IEEE 40(10): 42–51.

133. (2008) Protecting the Network from Denial of Service Floods. Technical report, Juniper Networks, Inc.

134. (2004) IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges. IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998) pp. 1–277.

135. Wojdak W (2003) Rapid Spanning Tree Protocol: A new solution from an old technology. Reprinted from CompactPCI Systems .

136. Yeung K, Yan F & Leung T (2006) Improving Network Infrastructure Security by Partitioning Networks Running Spanning Tree Protocol. In: Proceedings of International Conference on Internet Surveillance and Protection, 2006. ICISP'06. IEEE.

137. Marro G (2003) Attacks at the Data Link Layer. Ph.D. thesis, University of California.

138. Perlman R (1985) An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN. In: Proceedings of ACM SIGCOMM Computer Communication Review, volume 15, pp. 44–53. ACM.

139. (2002) IEEE Standards for Local and Metropolitan Area Networks— Virtual Bridged Local Area Networks- Amendment 3: Multiple Spanning Trees. IEEE Std 802.1s-2002 (Amendment to IEEE Std 802.1Q, 1998 Edition) .

140. Spanning Tree Protocol Introduction. Technical report, CISCO Cooperation. URI: http://www.cisco.com/en/US/tech/tk389/tk621/tsd_technology_support_protocol_home.html.

141. (2005) Spanning Tree Protocol Root Guard Enhancement. Technical report, CISCO Cooperation.

142. (2005) Spanning Tree Portfast BPDU Guard Enhancement. Technical report, CISCO Cooperation.

143. (2007) Cisco Spanning-Tree Protocol Enhancements using LOOP Guard and BPDU Skew Detection Features. Technical report, CISCO Cooperation.

144. Pan P (2011). Software-Defined Network (SDN) Problem Statement and Use Cases for Data Center Applications. Internet Draft.

145. Scharf M, Gurbani V, Voith T, Stein M, Roome W, Soprovich G & Hilt V (2013) Dynamic VPN optimization by ALTO guidance. In: Software Defined Networks (EWSDN), 2013 Second European Workshop on, pp. 13–18. IEEE.

146. Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Venkata S, Wanderer J, Zhou J, Zhu M *et al.* (2013) B4: Experience with a globally-deployed software defined WAN. In: ACM SIGCOMM Computer Communication Review, volume 43, pp. 3–14. ACM.

147. Casado M, Koponen T, Shenker S & Tootoonchian A (2012) Fabric: A Retrospective on Evolving SDN. In: Proceedings of the first workshop on Hot topics in software defined networks, pp. 85–90. ACM.

148. Konstantaras S & Thessalonikefs G (2014) Software Defined VPNs. Master's thesis, University of Amsterdam.

149. Jacobson V (1988) Congestion Avoidance and Control. In: ACM SIGCOMM Computer Communication Review, volume 18, pp. 314–329. ACM.

150. Gebert S, Pries R, Schlosser D & Heck K (2012) Internet Access Traffic Measurement and Analysis. Springer.

151. About POX. Technical report. URI: http://www.noxrepo.org/pox/about-pox/.
152. OpenFlow Switch Specification Version 1.1.0. Technical report. URI: http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf.