Optimization of Relay Placement for Scalable Virtual Private LAN Services

Mohammad Borhani
Linköping University
Linköping, Sweden
mohammad.borhani@liu.se

Ioannis Avgouleas
Linköping University
Linköping, Sweden
ioannisavgouleas@gmail.com

Andrei Gurtov
Linköping University
Linköping, Sweden
gurtov@acm.org

ABSTRACT

Virtual Private LAN Services are becoming popular for securely connecting geographically dispersed devices to a common protected LAN network isolated from the rest of the Internet. Traditional IP routing protocols cannot provide such connectivity; thus an overlay network of encrypted HIP/IPsec tunnels can be used instead. However, the number of full-mesh tunnels between communicating devices grows exponentially to the number of devices thereby suggesting the investigation of alternatives. The introduction of relaying, which entails selecting a subset of hub routers to retain full-mesh connectivity, allows non-hub routers, the so-called spokes, to maintain connectivity via a hub. In this work, we study the effect of relay-based routing that minimizes the number of hubs, the connection cost between spokes and hubs, the cost of connecting hubs, and the hubs deployment cost. Additionally, we prove that this minimization problem is NP-hard and, thus, intractable for large scale networks. Therefore, we propose an algorithm with provable guarantees that provides an approximate but efficient solution. Initial simulation results indicate a reduction by more than 90% in the memory required for routing tables at the expense of a minor increase in the tunnel path length.

CCS CONCEPTS

• **Networks** → **Network design principles**; Network Design; • **Theory of computation** → *Discrete optimization*; • **Mathematics of computing** → *Mathematical optimization.*

KEYWORDS

Virtual Private LAN Services, Routing, Host Identity Protocol, Approximation Algorithm

ACM Reference Format:
Mohammad Borhani, Ioannis Avgouleas, and Andrei Gurtov. 2022. Optimization of Relay Placement for Scalable Virtual Private LAN Services. In *ACM SIGCOMM 2022 Workshop on Future of Internet Routing & Addressing (FIRA '22), August 22, 2022, Amsterdam, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3527974.3545719

FIRA '22, August 22, 2022, Amsterdam, Netherlands

ACM ISBN 978-1-4503-9328-7/22/08.
https://doi.org/10.1145/3527974.3545719

1 INTRODUCTION

A massive amount of devices with heterogeneous requirements are being connected through the Internet daily. These include smart sensors, valve controls, and traffic lights, among others. Recent studies have revealed thousands of such devices online per a medium-size country, often using outdated insecure protocols such as ModBus [10]. Such devices are often hard to patch and contain known vulnerabilities that can be exploited in botnets such as Mirai. Thus, there is a clear need to hide such devices from the public Internet, yet allowing authorized connectivity for remote data management and updates.

The concept of Virtual Private LAN Services (VPLS) is based on an idea to combine islands of such devices to a single virtual local-area network using a set of encrypted tunnels [6, 11, 17]. Programmable gateways at each island intercept Address Resolution Protocol (ARP) requests targeted to other islands, capture and encapsulate LAN packets for tunneling over the Internet. The Host Identity Protocol (HIP) offers an appropriate method to establish IPsec ESP tunnels with a base exchange, maintain with keep-alive UPDATE messages and gracefully close when not needed. HIP can be viewed as one internetworking architecture aiming to implement identifier/locator split. Furthermore, other identifier/locator separation approaches exist, such as the Locator/Identifier Separation Protocol (LISP). However, since HIP encompasses end-to-end security, mobility and multi-homing, we are primarily concerned with a seamless integration of VPLS with HIP [19].

As VPLS uses a single broadcast domain, it has multiple benefits, including low communication latency, support for legacy protocols, and cost-effective installation and maintenance costs thereby reducing CAPEX and OPEX. The popularity of VPLS is encouraged by the fact that companies such as Cisco, Juniper, and Nokia are working on VPLS [2, 13, 20]. Moreover, HIP-based VPLS (HIPLS) is successfully implemented for example by Tempered Networks in USA [23]. Their deployment scenarios include securely connecting several hundred buildings of a university campus, wind generators of a electrical company, and a network of ATM machines. HIPLS allows devices to communicate in a LAN-like configuration while, at the same time, being hardly accessible for breaching their defense using the Internet. Additionally, the increasing scale of VPLS networks gives rise to challenges such as optimal tunnel management, performance and fault-tolerance.

Maintaining a full-mesh of all-to-all gateway tunnels is inefficient, since limited ternary content-addressable memory (TCAM) constraints the number of tunnels to a few thousand per gateway. Thus, it makes sense to dynamically establish and close the tunnels based on the current traffic patterns between device islands. By utilizing the concept of relaying [14], packets can be forwarded

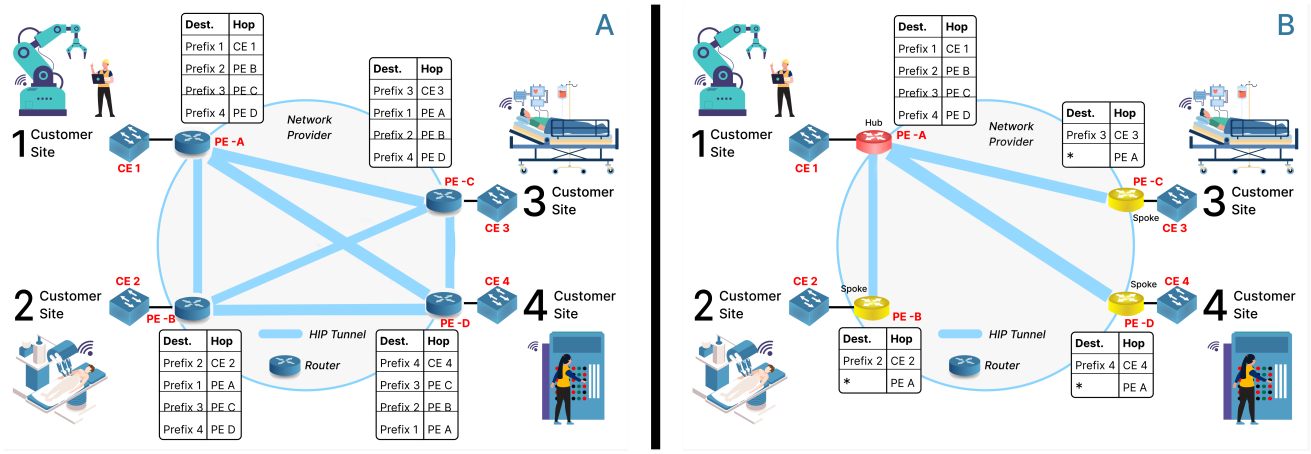

Figure 1: Memory footprint of the routing tables in an example secure HIPLS network with four PEs using: (A) full-mesh and (B) relay-based routing (the studied approach). In the latter, only the hub PEs are fully-meshed thereby realizing great memory savings for the spoke PEs at the expense of a minor increase in the tunnel path length for connecting spoke PEs.

through another gateway with an already active tunnel thereby reducing the need of activating a new gateway router. However, added latency must be considered as often VPLS traffic is of real-time nature. Finally, to address the single point of failure by using only the main tunnel, multiple existing paths with appropriate delay for fault-tolerance should be considered.

Somewhat surprisingly, even if relaying for Multi-Protocol Label Switching (MPLS) was studied e.g., in [1], this the first paper to consider relaying for encrypted tunnels. This paper makes following contributions:

- We propose and formulate the "Relay Placement Problem (RPP)" for programmable tunnel gateways to minimize the cost of activating and deploying hub and spoke routers for relay-based routing.
- We prove that RPP is NP-hard.
- We develop an approximation algorithm to offer a polynomial-time solution to the RPP problem, with a guaranteed approximation factor.
- Numerical experiments for different scenarios, including real-world topologies, demonstrate that our proposed algorithm decreases the overall amount of router memory required by more than 90% and lowers network provider costs at the expense of a minor increase in packets traversed paths.

The rest of the paper is organized as follows. In Section 2, we give a brief introduction to VPLS. Section 3 formulates the relay placement problem and develops an approximate but efficient algorithmic solution. Section 4 discusses the results. Last, Section 5 concludes the paper and gives directions for future works.

2 BRIEF INTRODUCTION TO VPLS

VPLS is a Layer 2 provider-provisioned VPN that allows multi-point-to-point connection between remote customer sites via a provider network. A usual VPLS deployment has multiple key components including:

- **Customer Network**: This is the VPLS network's user. It is composed of numerous sites that are geographically dispersed and completely managed by the user. Global manufacturer, national health care provider, energy cooperative are few examples for VPLS users.
- **Provider Network**: This is the VPLS underlay network that allows tunnels to be established. The provider networks are typically Layer 3 networks that use common network protocols such as MPLS or IP.
- **Provider Edge Equipment (PE)**: PEs are the gateways for customer network traffic and are located at the provider's network's edge. PEs also have a thorough understanding of the VPLS network. Tunnels are constructed between PEs.
- **Customer Edge Equipment (CE)**: CEs are the interconnecting devices between the customer and provider networks. CEs are owned by the customer and located on the customer's premises.

HIPLS is the secure VPLS architecture containing a logical security layer for managing VPLS security services. HIPLS offers payload encryption, secure control protocol, protection from IP attacks, and PE authentication [11, 18].

3 RELAY PLACEMENT PROBLEM (RPP)

3.1 Overview

To interconnect PEs, the provider creates a full-mesh of HIP tunnels via the IP/MPLS-based provider network. However, this reachability model forces routing tables in PEs to grow large. For the sake of illustration, we show a HIPLS network with four sites (connecting via CEs) in Figure 1. In a full-mesh setup, each PE should install four routes (prefix) to ensure connectivity. Furthermore, the fact that each PE in a VPLS design can potentially be connected to numerous CEs increases the size of the PE routing table exponentially.

The concept of relaying can be employed to decrease the growth of entries in routing tables. The relaying strategy selects a small

subset of PEs as hub nodes while retaining full-mesh reachability [14]. This allows non-hub PEs, known as spoke PEs, to reach other PEs by relaying through a specified hub PE. Relaying, as shown in Figure 1 (B), can substantially reduce the routing entries on the spoke PEs at the expense of more traffic being relayed on the provider network, which could increase latency for the customer sites.

Selecting the hub routers for relaying involves minimizing the number of hubs, as fewer hubs reduce the PE memory footprint and lowers hubs' installation and maintenance costs. Moreover, the traffic between two spoke PEs is possibly rerouted via a hub PE over an indirect path.

Raghunath et al. [21] investigated the structure of VPNs and discovered that hub-spoke architecture is employed in VPNs. Kim et al. [14] explored the scalable routing for MPLS L3VPN as an optimization problem. Our work differs from theirs as we consider secure VPLS as our target network. Furthermore, our problem formulation takes into account various costs (for instance, spokes to hub and hub installation cost) and provides a minimum cost tree that spans the hubs.

3.2 Problem Formulation

We define the *core network* within the provider network to only contain hub PEs. In other words, a core network is a set of interconnected hub PEs that are responsible nodes for relaying data. Furthermore, transferring traffic between hub PEs within the core network incurs switching costs for the network provider. The following steps are involved in deploying the relaying architecture within the HIPLS network:

- **Hub PE selection**: This step considers selecting a set of PEs as the hub for data transmission with the aim of minimizing the hubs' installation and maintenance costs. Other non-hub PEs will be considered as spokes.
- **Hub PE assignment**: Connecting each spoke PE to its designated hub PE accomplishes this step.
- **The Hubs link selection**: The last stage entails choosing links (edges) to connect all hub PEs, with the goal of picking links that lower the total cost of connection between hub nodes.

We define the Relay Placement Problem (RPP) in HIPLS as follows: the provider network is modeled as an undirected graph $G = (V, E)$, where $V = \{PE_1, PE_2, \ldots, PE_n\}$ is the set containing all PEs in the network. E denotes the set of edges (links) that connect the PEs, and $c : E \longrightarrow \mathbf{Q}^+$ represents the cost of edges. Latency, bandwidth, and cost to use specific links are among the possible metrics for edge costs.

$\mathcal{F} \subset V$ denotes the set containing possible locations for installation of hub PEs, and spoke PEs are represented by $\mathcal{D} \subset V$. The solution to RPP considers selecting a set of active hub PEs denoted by F such that $F \subset \mathcal{F}$. Then, assigning each spoke PE j ($j \in \mathcal{D}$) to some hub PE. Let x_{ij} denote a binary variable indicating whether the spoke j is connected to hub i. Additionally, $y_i = 1$ denotes whether hub i should be activated, and d_{ij} is the communication cost between spoke PE j and hub PE i (the cost of routing concerning the edge costs connecting spoke j to hub i). Moreover, a_i denotes the cost of activating hub i that is configured by the network provider, and represents deployment and maintenance cost of hubs.

Finally, the Steiner tree T, the tree with the minimum cost that spans all hub PEs, should be constructed to ensure the connectivity of hub PEs. We formulate RPP within HIPLS network as:

$$(P) \text{ minimize} \quad \sum_{i \in F} \sum_{j \in \mathcal{D}} d_{ij} x_{ij} + N \sum_{k \in T.edges} c(k) + \sum_{i \in F} a_i y_i \quad (1)$$

The term $\sum_{i \in F} a_i y_i$ in the (P) calculates the opening cost of hub PEs; the second term i.e., $N \sum_{k \in T.edges} c(k)$, captures the Steiner cost to connect all hub PEs via the Steiner tree T, in which $N \geq 1$ is a parameter to represent the cost of connecting hub PEs in *core network*. The connection cost between spoke PEs and hub PEs is demonstrated in $\sum_{i \in F} \sum_{j \in \mathcal{D}} d_{ij} x_{ij}$.

Problem (P) is a network design problem i.e., a NP-hard problem [5]. Although the RPP seems to be formulated easily, it is a difficult problem to solve efficiently (unless P = NP), making the exact solution intractable for medium to large networks. As a result, we provide an approximation schema for RPP, which produces solutions with reasonable running times in reality, as opposed to exact methods, which are computationally expensive.

3.3 Approximation Algorithm for RPP

Some of the most well-known NP-hard network design problems can be approximated using simple randomized algorithms [7]. A class of these algorithms, known as Sample-Augment (SA) algorithm, are based on the idea of selecting a random sample from the problem input, solving a subproblem, and finally augmenting the result with the solution to the original problem [8, 9].

We define the Sample-Augment problem for a minimization problem $\mathcal{P}$ as follows:

(1) Define $\mathbf{K} = \{1, \ldots, n\}$ as a set containing elements, and sampling probability for the elements as $(p_1, \ldots, p_n)$
(2) $\mathcal{P}_{sp}(K)$ is defined as subproblem for any $K \subseteq \mathbf{K}$
(3) For any $K \subseteq \mathbf{K}$ and solution to the previous step's subproblem (i.e., $Sol_{sp}(K)$), the augmentation problem is defined as $\mathcal{P}_{aug}(K, Sol_{sp}(K))$.

The SA algorithm executes the following steps:

- Obtaining independent samples from $\mathbf{K}$ based on the sampling probability
- Finding the solution to the defined subproblem and the augmented problem (for random sample it obtained in the previous step)
- The SA algorithm outputs the aggregate solution to the subproblem and augmentation problem as final solution.

Formulating RPP as a minimization problem yields a variation of the uncapacitated facility location problem (UFLP) and the Steiner tree problem. Without hubs connection requirements (i.e., removing the Steiner tree problem from (1)), the (P) problem becomes an UFLP instance, which has been shown to be NP-hard and widely investigated in the literature.

Algorithm 1 applies a well-established approximation algorithm to obtain a good solution for the Spoke-to-Hub (SH) assignment problem, which we will introduce shortly, to choose which hub PEs

to open from the list of candidate hub locations. Then, in the sampling step of algorithm 1, each spoke PE is marked independently by the probability of β, and in the solution to the SH assignment problem, we activate the hub PEs to which the marked PEs are allocated. Algorithm 1 applies connection requirements on the marked PEs by using approximated solution to the Steiner tree problem [16, 22] to link the hub PEs and extends this solution to include the open hubs (augmentation step).

Algorithm 1: Approximation Algorithm for RPP.

$\gamma \in (0, 1]$;
$F \leftarrow \emptyset$;
$\beta \leftarrow \frac{\gamma}{N}$;
/* Solving UFLP */
Execute the 3−approximation algorithm for Spokes-to-Hubs (SH) Assignment problem, and obtain the solution as $H = (F_H, x_{ij})$;
/* Sampling */
Sample (mark) a spoke PE^* at random ;
Sample every other spoke non-marked PE independently with probability β;
Let M ={set of marked PEs} ;
/* Augmentation */
for all $i' \in F_H$ **if** $(\{j | j \in \mathcal{D}$ *and* $x_{i'j} = 1\} \cap M \neq 0)$ **then**
  F.add(i');
end
Execute the 2−approximated Steiner Tree T on the set M ;
Augment T with adding the shortest paths from each spoke PE $j \in M$ and its associated hub PE;
Find a tree T'' which spans the F;
Allocate each spoke PE $j \in \mathcal{D}$ to its closest hub PE in F ;
return $\{F, T''\}$

3.4 Spokes-to-Hubs (SH) Assignment

The problem of assigning spoke to hub PEs can be formulated as follows:

$$(SH) \text{ minimize} \quad \sum_{i \in F} \sum_{j \in \mathcal{D}} d_{ij} x_{ij} + \sum_{i \in F} a_i y_i \tag{2a}$$

$$\text{subject to} \quad \sum_{i \in F} x_{ij} \geq 1, \ j \in \mathcal{D} \tag{2b}$$

$$x_{ij} \leq y_i, \ j \in \mathcal{D} \ \text{ and } \ i \in F \tag{2c}$$

$$x_{ij} \in \{0, 1\}, \ j \in \mathcal{D} \ \text{ and } \ i \in F \tag{2d}$$

$$y_i \in \{0, 1\}, \ i \in F \tag{2e}$$

Constraint (2b) forces each spoke to be assigned to at least one hub. By (2c), only active hubs should be assigned to spokes, and the last two constraints set the domain of the binary decision variables.

3.4.1 Approximation Algorithm of Spokes-to-Hubs (SH) Assignment. Since (SH) is a form of the Uncapacitated Facility Location Problem (UFLP), which has been shown to be NP-hard, we used 3−approximation algorithm based on primal-dual schema and Lagrangian relaxation to approximate its exact solution [12].

Table 1: Routing Entries for mid-size AS network with supported CE(1-10).

#Entries in Routing Tables for All PEs			
#PEs	Full-mesh	Hub-Spoke	#Hubs
100	50100	2092	3
150	109950	3794	5
200	192800	7879	7
250	302250	9889	7
300	440100	12303	9

Theorem 1. *By using 3-approximation algorithm for SH assignment problem, 2-approximation for the Steiner Tree problem, and proper choice of β [3], Algorithm 1 is an expected 6.6-approximation algorithm for the RPP problem.*

Proof. See Appendix A.

4 EVALUATION AND DISCUSSION

To evaluate the proposed algorithm's performance, we implemented Algorithm 1 on a PC running Windows 10 (4-core 2.60 GHz CPU), equipped with 8GB of RAM. For performance evaluation, we employed various types of provider network topologies, including:

- AS Network Topology: Since VPLS can be employed in large-scale networks and there exists a demand for using VPLS across multiple Autonomous Systems (AS), we generate AS network graph with properties stated in [4].
- Backbone Network Topology: We utilized backbone topologies from The Internet Topology Zoo [15] to evaluate the path traversal in hub-spoke.

Table 1 compares the number of routing entries installed in all PEs for full-mesh and hub-spoke. As motivated by the example in Figure 1, the total number of routing entries in full-mesh equals `#PE` × `#routing entries of each PE`, in which the latter term for each PE is calculated by summing the number of CEs in the network.

The number of routing entries for all PEs in hub-spoke obtained by adding the routing entries installed for each spoke and hub PE in the network. The number of routing entries for a spoke PE is comprised of the number of its supported CEs plus the number of hubs to which the spoke PE is connected. Furthermore, because the hub PE should contain all of the network's routing information, the number of routing entries in the hub PE is computed by adding all supported CEs in the network. Table 1 shows that for a random number of CEs chosen from the interval (1-10), the number of installed routing entries is significantly reduced by leveraging the hub-spoke relaying.

Figure 2 depicts the cost of the solution (i.e., summing the connection cost between spoke PEs to hub PEs, opening cost of hub PEs and cost of connecting all hub PEs in Steiner Tree) for proposed Algorithm 1 and random hub placement. In random hub placement, a subset of PE is randomly chosen to be hub PEs such that the number of hubs in both approaches (Random Hub Placement and

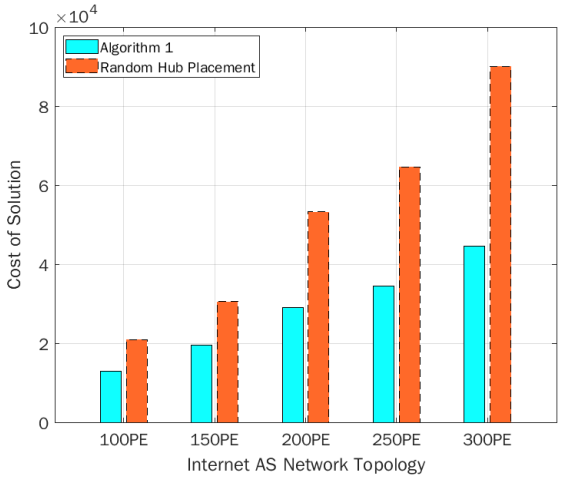

Figure 2: Solution cost for Random Hub Placement vs. Algorithm 1.

Algorithm1) is the same. Moreover, in random hub placement, random spoke PE assigned to hubs. Figure 2 shows that the Algorithm 1 generates less costly solutions for RPP than random hub placement.

Figure 3 illustrates the number of routing entries for large-scale AS networks (400 to 800 PEs) in full-mesh. Furthermore, the routing entries for hub-spoke for the same networks are depicted in Figure 4. Obviously, as the number of PEs in the network grows, the routing entries also increase. However, the increase is significantly greater with full-mesh. As a result, hub-spoke may be effectively used in large networks.

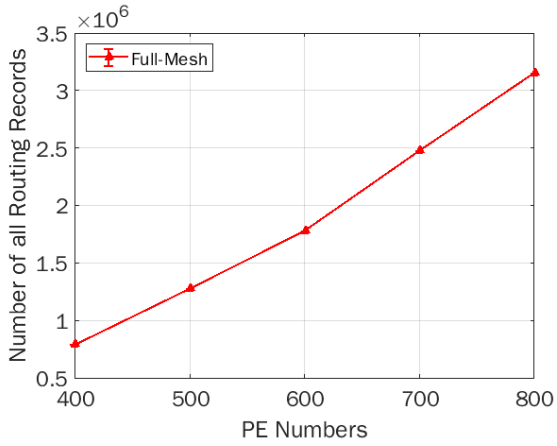

Figure 3: Routing entries for a full-mesh large-scale AS Network.

In hub-spoke data transmission, the source node transfers data to its corresponding hub. The data is then forwarded to the second hub associated with the destination PE, if necessary. Finally, the data is sent to the final PE destination through the second hub. We used the two backbone networks to evaluate the additional path taken

Table 2: Comparison of path traversed by full-mesh vs hub-spoke.

Extra Path Traversed by Hub-Spoke				
Network	Location	#Nodes	Ratio	Margin of Error
Backbone, Transit	US	51	1.387	1.3873 ±0.115 (±8.26%)
Backbone, Customer	NL	50	1.536	1.5361 ±0.0774 (±5.04%)

by the hub-spoke, in which each link (edge) of the network graph is represented by the distance between corresponding nodes (PE) creating that link in kilometers. Furthermore, a random number of

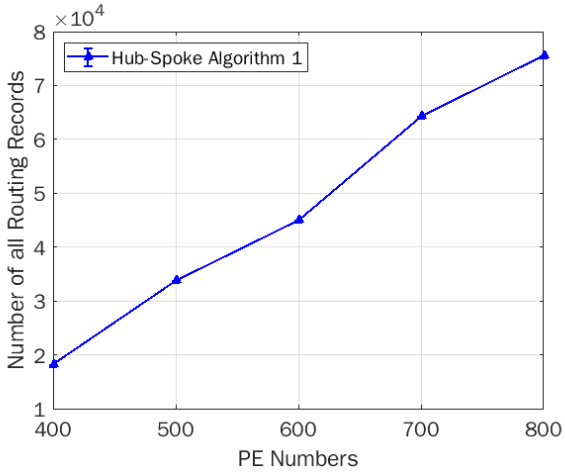

Figure 4: Routing entries for a hub-spoke Large-scale AS Network.

PE is chosen to create source_destination pairings for communication. The average amount of routing cost (i.e., path length with respect to the edge cost) in the hub-spoke divided by the same value for full-mesh is the ratio for several selections of random source_destination pairs in Table 2. The average increase in path length in traversed distance caused by hub-spoke design is represented by this ratio. Table 2 includes the ratio for both backbones with 95% confidence interval reported.

In the next experiment, we used Mininet to implement HIPLS in full-mesh and hub-spoke for a network topology in the USA. We purposefully chose a geographically dispersed network graph to examine the proposed approach in the extreme hub-spoke scenarios, in which relaying can add considerable latency [1]. In Mininet, the propagation delay of the link was estimated using the distance between nodes. Figure 5 depicts the Mininet simulation results from four distinct scenarios. In hub-spoke scenarios, algorithm 1 is given

[1] http://www.topology-zoo.org/maps/Compuserve.jpg

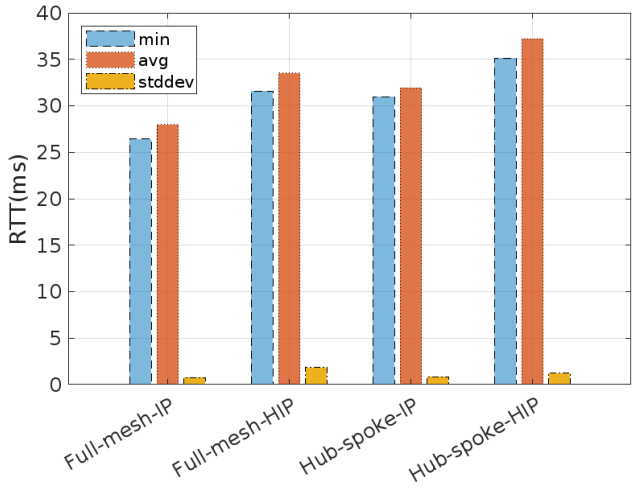

Figure 5: Mininet latency experiment for full-mesh and hub-spoke.

the network graph and the cost of placing a hub in the network as inputs, and the output contains the number of hubs and their locations, as well as the spokes' associations with hubs. Comparing the HIPLS (secure HIP-based VPLS) and IP connectivity (no security) shows the cost of the delay one should pay to secure the VPLS network (5.553*ms* more delay to secure full-mesh VPLS). Comparing HIPLS and IP connectivity in both full-mesh and hub-spoke stressed, as predicted, that offering a relaying imposes increase in path length (higher RTT delay) to decrease routing entries in PEs. For instance, the HIPLS needs to endure an extra 3.685*ms* delay in hub-spoke compared to full-mesh on average RTT.

5 CONCLUSIONS AND FUTURE WORK

We studied the relay placement problem in the context of Virtual Private LAN Services. To our knowledge, this is the first attempt to extend the VPN relaying problem to the case of encrypted tunnels between the PE nodes with Host Identity Protocol (HIP). Although the main problem is intractable due to NP-hardness, we propose a fast approximation algorithm. Initial simulations show that it can decrease fast memory demands in PE nodes up to a hundred times with proper hub-spoke relays, compared to full tunnel mesh between PE nodes. This comes at a moderate increase in the latency, as VPLS often carry real-time traffic expecting LAN-level delays.

We currently lack accurate traffic pattern and topologies data for real-world VPLS deployments. We plan to construct realistic topologies based on deployment scenarios by the Tempered company (tempered.io). One such scenario includes connecting several hundred building within a university campus to a VPLS. Another is connecting all wind generators within a single energy provider together. Obviously, traffic patterns can be also very different, ranging from all-to-all communication closer to a full-mesh of tunnels, up to strictly leaf devices reporting to a single server. We will use these data to improve the accuracy of our model and simulations.

ACKNOWLEDGMENT

This work was in part supported by the Excellence Center at Linköping – Lund in Information Technology (ELLIIT) and Graduate School in Computer Science (CUGS).

REFERENCES

[1] MohammadHossein Bateni, Alexandre Gerber, Mohammad Taghi Hajiaghayi, and Subhabrata Sen. 2009. Multi-VPN Optimization for Scalable Routing via Relaying. In *INFOCOM 2009*. IEEE, 2756–2760.
[2] Cisco. 2019. Cisco VPLS Project. https://www.cisco.com/c/en/us/products/ios-nx-os-software/virtual-private-lan-services-vpls
[3] Friedrich Eisenbrand, Fabrizio Grandoni, Thomas Rothvoß, and Guido Schäfer. 2008. Approximating Connected Facility Location Problems via Random Facility Sampling and Core Detouring. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '08)*. Society for Industrial and Applied Mathematics, USA, 1174–1183.
[4] Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. 2010. On the Scalability of BGP: The Role of Topology Growth. *IEEE Journal on Selected Areas in Communications* 28 (2010), 1250–1261.
[5] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA.
[6] Kuntal Gaur, Anshuman Kalla, Jyoti Grover, Mohammad Borhani, Andrei Gurtov, and Madhusanka Liyanage. 2021. A Survey of Virtual Private LAN Services (VPLS): Past, Present and Future. *Computer Networks* 196 (2021).
[7] Anupam Gupta, Amit Kumar, Martin P´al, and Tim Roughgarden. 2007. Approximation via Cost Sharing: Simpler and Better Approximation Algorithms for Network Design. *J. ACM* 54, 3 (2007).
[8] Anupam Gupta, Amit Kumar, and Tim Roughgarden. 2003. Simpler and Better Approximation Algorithms for Network Design. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC '03)*. Association for Computing Machinery, New York, NY, USA, 365–372.
[9] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. 2004. Boosted Sampling: Approximation Algorithms for Stochastic Optimization *(STOC '04)*. Association for Computing Machinery, New York, NY, USA, 417–426.
[10] David Hasselquist, Abhimanyu Rawat, and Andrei Gurtov. 2019. Trends and Detection Avoidance of Internet-Connected Industrial Control Systems. *IEEE Access* 7 (2019), 155504–155512.
[11] T Henderson, S Venema, and D Mattes. 2011. HIP-based virtual private LAN service (HIPLS). *Internet Draft, IETF* (2011).
[12] Kamal Jain and Vijay V. Vazirani. 2001. Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation. *J. ACM* 48, 2 (2001), 274–296.
[13] Juniper. 2019. Juniper Networks-VPLS. https://www.juniper.net/documentation/junos/topics/concept/vpls-security-overview.html
[14] Changhoon Kim, Alexandre Gerber, Carsten Lund, Dan Pei, and Subhabrata Sen. 2008. Scalable VPN Routing via Relaying. In *Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '08)*. ACM, New York, NY, USA, 61–72.
[15] Simon Knight, Hung X. Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. 2011. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications* 29 (2011), 1765–1775.
[16] L. Kou, George Markowsky, and L. Berman. 1981. A Fast Algorithm for Steiner Trees. *Acta Informatica* 15 (1981), 141–145.
[17] Madhusanka Liyanage and Andrei Gurtov. 2013. A scalable and secure VPLS architecture for provider provisioned networks. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1115–1120.
[18] Madhusanka Liyanage, Jude Okwuibe, Mika Ylianttila, and Andrei Gurtov. 2015. Secure Virtual Private LAN Services: An overview with performance evaluation. In *2015 IEEE International Conference on Communication Workshop (ICCW)*. 2231–2237.
[19] Pekka Nikander, Andrei Gurtov, and Thomas R. Henderson. 2010. Host Identity Protocol (HIP): Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks. *IEEE Communications Surveys Tutorials* 12 (2010), 186–204.
[20] Nokia. 2019. Nokia VPLS Course. https://networks.nokia.com/src/course/virtual-private-lan-services
[21] Satish Raghunath, K. K. Ramakrishnan, Shivkumar Kalyanaraman, and Chris Chase. 2004. Measurement Based Characterization and Provisioning of IP VPNs. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC '04)*. ACM, New York, NY, USA, 342–355.
[22] Gabriel Robins and Alexander Zelikovsky. 2005. Tighter Bounds for Graph Steiner Tree Approximation. *SIAM Journal on Discrete Mathematics* 19 (2005), 122–134.
[23] Tempered. 2022. Whitepaper IDN. https://www.tempered.io

A PROOFS

We need to bound the below costs:

- Cost of opening hubs
- Cost of connecting spokes to hubs
- Cost of connecting hubs through Steiner Tree

In optimal solution, we have:

- O^*: Opening cost of hub PEs
- C^*: Connection cost between spoke PEs and hub PE
- T^*: Steiner Tree cost

Moreover, in Algorithm 1, we have:

- O_{sh}: Opening cost for approximation solution to SH assignment
- C_{sh}: Connection cost for approximation solution to SH assignment

By considering $OPT = O^* + C^* + T^*$, and Section 3.4 we obtain

$$O_{sh} + C_{sh} \leq 3OPT_{sh} \leq 3OPT$$

Lemma 1 [3]: By considering $\rho_{st} = 2$ as the approximation ratio for Steiner Tree solution, the Steiner cost of T in Algorithm 1 is:

$$E[T] \leq \rho_{st}(\beta N(1 + o(1))C^* + T^*) + \beta N(1 + o(1))C_{sh})$$

Lemma 2 [3]: The connection cost of C in Algorithm 1 is:

$$E[C] \leq C_{sh} + 2C^* + \frac{T^*}{\beta N}$$

Now, we can obtain the expected approximation ratio for Algorithm 1 as (considering the approximation ratio for (sh) problem as $\rho_{sh} = 3$):

$$\begin{aligned} E[\texttt{Solution Cost}] &\leq C_{sh} + 2C^* + \frac{T^*}{\beta N} + \beta NC_{sh} \\ &\quad + \rho_{st}(\beta NC^* + T^*) + O_{sh} \\ &\leq \rho_{st}(\beta NC^* + T^*) + 2C^* + \frac{T^*}{\beta N} \\ &\quad + (O_{sh} + C_{sh})(1 + \beta N) \\ &\leq \rho_{st}(\beta NC^* + T^*) + 2C^* + \frac{T^*}{\beta N} \\ &\quad + \rho_{sh}(O^* + C^*)(1 + \beta N) \\ &\leq 6.6OPT \ \texttt{for} \ \beta = 0.33/N \end{aligned}$$