Optimization of Relay Placement for Scalable Virtual Private LAN Services

Mohammad Borhani, Ioannis Avgouleas, Andrei Gurtov
1st ACM SIGCOMM Workshop on Future of Internet Routing & Addressing (FIRA)

SIGCOMM 2022
Agenda

Introduction
- VPLS
- HIPLS

Problem Statement
- Full-mesh problem/Relaying
- Relay Placement Problem (RPP)

Proposed Solution
- Approximation Algorithm

Results

Future Works/Conclusions
Introduction

- Daily, massive amounts of heterogeneous devices are connected to the Internet
- Industry 4.0: Smart sensors/valve controls
- Outbreak of Coronavirus in 2020 [1]
  - Production facilities shut down
  - Remote works, relying on the Internet
- Such devices are often hard to patch/contain exploitable vulnerabilities: Mirai
- Hide such devices from the Internet while allowing remote data administration and updates: VPLS
Introduction- Virtual Private LAN Service (VPLS)

- Over a public network, VPLS provides Layer 2 multipoint Ethernet service
  - Support for legacy protocols [2]
  - Single broadcast domain
  - Cost-effective installation/maintenance

- HIPLS: HIP establishes/maintains/closes IPsec ESP tunnels in VPLS
Problem Statement
Full-mesh problem

- To interconnect PEs, the provider creates a full-mesh of HIP tunnels.
- This reachability model forces routing tables in PEs to grow large.
- In a full-mesh setup, each PE should install customers’ routes (prefix) to ensure connectivity.
- Supporting numerous CEs increases the size of the PE routing table exponentially - TCAM constraints.
Relaying

- Hub PEs: full-mesh reachability
- Spoke PEs: reach other PEs by relaying through hubs
  - Pros: Relaying can substantially reduce the routing entries on the spoke PEs
  - Cons: more traffic being relayed, which could increase latency for the customer sites
- Optimization problem: minimize the cost of activating hubs, hub/spokes connection, and hubs connectivity for relay-based routing
Relay Placement Problem (RPP) – Mathematical Program

- RPP considers:
  - **Hub PE selection**: Selecting a set of active hub: \( F \subset \mathcal{F} \)
  - **Hub PE assignment**: Assigning spoke \( PE_j \) to some hub \( PE_i \)
  - **Hub links selection**: selecting links that lower the total cost of connection between hub nodes

\[
(P) \minimize \sum_{i \in F} \sum_{j \in \mathcal{D}} d_{ij}x_{ij} + N \sum_{k \in T.edges} c(k) + \sum_{i \in F} a_iy_i
\]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G = (V, E) )</td>
<td>Provider network</td>
</tr>
<tr>
<td>( C )</td>
<td>Cost to use specific link (latency, bandwidth)</td>
</tr>
<tr>
<td>( \mathcal{F} \subset V )</td>
<td>Set of potential locations for installing hub PEs</td>
</tr>
<tr>
<td>( \mathcal{D} \subset V )</td>
<td>Set of spoke PEs</td>
</tr>
<tr>
<td>( x_{ij} )</td>
<td>Binary variable whether spoke ( PE_j ) is connected to hub ( PE_i )</td>
</tr>
<tr>
<td>( y_i )</td>
<td>Whether hub ( PE_i ) should be activated</td>
</tr>
<tr>
<td>( a_i )</td>
<td>Cost of activating hub ( PE_i )</td>
</tr>
<tr>
<td>( d_{ij} )</td>
<td>Communication cost between spoke ( PE_j ) and ( PE_i )</td>
</tr>
</tbody>
</table>

- Connection cost between spoke PEs and hub PEs
- Cost to connect all hub PEs via Steiner Tree
- Opening cost for hub PEs
Proposed Solution
Approximation Algorithm for RPP

- RPP is NP-hard problem [3]
- Proposed approach: Sample-Augment Algorithm:
  - Solving the subproblem: SH assignment

\[
\begin{align*}
(SH) \text{ minimize } & \sum_{i \in F} \sum_{j \in D} d_{ij} x_{ij} + \sum_{i \in F} a_i y_i \\
\text{subject to } & \sum_{i \in F} x_{ij} \geq 1, \; j \in D \\
& x_{ij} \leq y_i, \; j \in D \text{ and } i \in F \\
& x_{ij} \in \{0,1\}, \; j \in D \text{ and } i \in F \\
& y_i \in \{0,1\}, \; i \in F
\end{align*}
\]

- Random sample from the problem input
- Augmenting the result with the solution to original problem

- **Theorem 1:** using 3-approximation algorithm for SH problem, 2-approximation for the Steiner Tree problem, Algorithm 1 is an expected 6.6-approximation algorithm for RPP.

**Algorithm 1: Approximation Algorithm for RPP.**

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

- Simulation- Network Topology:
  - Backbone Network Topology: Internet Topology Zoo

Comparison of Routing Entries for mid-size AS network: full-mesh vs. hub-spoke

<table>
<thead>
<tr>
<th>#Entries in Routing Tables for All PEs</th>
<th>#PEs</th>
<th>Full-mesh</th>
<th>Hub-Spoke</th>
<th>#Hubs</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50100</td>
<td>2092</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>109950</td>
<td>3794</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>192800</td>
<td>7879</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>302250</td>
<td>9889</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>440100</td>
<td>12303</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

- The number of installed routing entries is significantly reduced by leveraging the relaying

Comparison of Routing Entries for large-scale AS network: full-mesh vs. hub-spoke

- Increase in number of PEs: the routing entries also increase. However, the increase is significantly greater with full-mesh
Results – Cont.

Solution cost: Random assignment vs. proposed Algorithm

- Proposed Algorithm generates less costly solutions for RPP than random hub placement

Comparison of path traversed by full-mesh vs hub-spoke

- The ratio of average increase in path length in traversed distance caused by hub-spoke design is less than 1.6

<table>
<thead>
<tr>
<th>Network</th>
<th>Location</th>
<th>#Nodes</th>
<th>Ratio</th>
<th>Margin of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backbone, Transit</td>
<td>US</td>
<td>51</td>
<td>1.387</td>
<td>(1.3873 \pm 0.115) ((8.26%))</td>
</tr>
<tr>
<td>Backbone, Customer</td>
<td>NL</td>
<td>50</td>
<td>1.536</td>
<td>(1.5361 \pm 0.0774) ((5.04%))</td>
</tr>
</tbody>
</table>

HIPLS needs to endure an extra ~ 3.6 ms delay in hub-spoke compared to full-mesh on average RTT
Future Works/Conclusions
Conclusions

- RPP: Reducing the total routing tables size in secure VPLS that minimizes:
  - The number of hubs (hubs deployment cost)
  - The connection cost between spokes and hubs
  - The cost of connecting hubs
- We proposed algorithm with provable guarantees: approximate but efficient solution
- Simulation results: 90% reduction in routing table entries with a slight increase in tunnel path length
- We currently lack accurate traffic pattern/topologies data for real-world VPLS deployments
- Future works:
  - Add/remove PEs in an online manner
  - Resiliency support in problem formulation
References


Thank you!
mohammad.borhani@liu.se