# *Tunnel Minimization and Relay for Managing Virtual Private Networks*

I-Wei Chen, Ying-Dar Lin and Yi-Neng Lin

Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan
{iwchen, ydlin, ynlin}@cis.nctu.edu.tw

***Abstract*—A virtual private network (VPN) is a private data network that carries traffic between remote sites. One of the most popular VPN applications is the "Intranet/Extranet VPN", which establishes network layer connections between remote intranet sites, using various tunneling protocols, to create an IP overlay network. IPSec, which is very prevalent in industry, is one of these tunneling protocols that not only provide encapsulation/decapsulation but encryption/decryption and hashing. However, an IPSec tunnel often fails to be established due to the management complexity. This work proposes the new concept of *authority* to alleviate the management overhead by reducing the number of tunnels. The problem of tunnel minimization is first formalized under three conditions - no constraint, a Tunnel Path Length constraint and a Tunnel Relay Degree constraint, and then solved using graphical models and the *Zero-One Integer Programming* algorithm. The effect of tunnel minimization is also investigated, and at most 90% of the tunnels are found to be reducible in a general enterprise VPN.**

*Keywords: VPN, optimization, tunnel reduction, IPsec, management*

## I. INTRODUCTION

The "access VPN" and the "Intranet/Extranet VPN" have been the two most popular VPN applications. An access VPN allows remote corporate users to enjoy connectivity to their corporate Intranets via ad hoc tunnels. Users can either set up PPP connections directly over a circuit-switching telephone network or using protocols such as PPTP [1] and L2TP [2] to establish PPP connections over the packet-switching Internet. In the latter case, users need only to connect to a local NAS but need not dial into the distant PPP server of the corporation. The main focus of an access VPN is to provide secure communication between end users.

However, an Intranet/Extranet VPN links the network of an enterprise headquarters to the networks of remote branches, or to networks of third parties, such as suppliers and partners. IPSec (IP Security) [3] [4] [5], which is the most popular protocol that supports this type of VPN, is used to encapsulate/de-capsulate, encrypt/decrypt and authenticate data. An IPSec tunnel is normally established by two VPN gateways which lie on the ingress/egress of corporations' networks. While supporting ordinary security, the Intranet/Extranet VPN is also associated with a complex *management overhead*.

Before an IPSec tunnel can be established between two VPN gateways, much information or several policies need to be negotiated by administrators. Consider the IKE (Internet Key Exchange) [6] for example; administrators must specify the packets to be transmitted or received through this tunnel, the negotiation mode (main or aggressive) of IKE phase 1, the required sub-protocol (ESP or AH), the encryption algorithm (DES or 3DES), the hash algorithm (MD5 or SHA1) , and the PSK (Pre-Shard Key). Tunnels are frequently *not* established for this reason, so the use of fewer tunnels, without affecting tunnel connectivity among the VPN gateways, is favored to reduce the overhead. Restated, packets that originally appear in a reduced tunnel are relayed to others, violating the requisite for private communication through an IPSec tunnel between two VPN gateways.

This study presents the concept of authority levels for different VPN gateways, to solve this problem. A VPN gateway has the privilege to relay packets, which are originally transmitted some reduced tunnel, if it has a higher authority than the two endpoints of the reduced tunnel. Figure 1 depicts an example network and its associated graph.

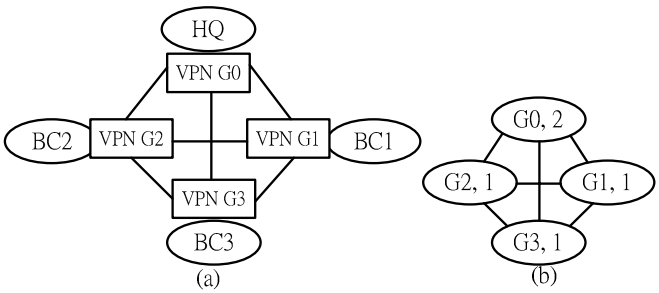


Fig.1. (a) A network example. (b) The associated graph

In this example, four corporate networks, referred to as HQ (headquarters), BC1 (branch company 1), BC2 (branch company 2), and BC3 (branch company 3), are to be connected using a VPN. The four networks are connected to each other via four VPN gateways (G0, G1, G2, G3) using six IPSec tunnels. In the graph, each network is represented by a VPN gateway, which is represented by a vertex. Each of these IPSec tunnels is represented as an edge. The number

 

associated with each vertex specifies its authority. A higher number means greater authority; naturally, headquarters has a higher authority than the branch. Hence, edge E(G1, G2) can be reduced to edges E(G1 G0) and E(G0 G2); edge E(G1, G3) can be reduced to edges E(G1 G0) and E(G0 G3), and E(G2, G3) can be reduced to edges E(G2 G0) and E(G0 G3). Fig. 2 depicts the results of the reductions.

As Fig. 2 shows, three tunnels are reduced in the minimization of tunnels since three is the minimum number of edges required to keep the graph connected. G0 is called the "tunnel relay gateway".

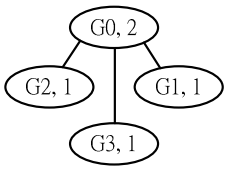


Fig. 2. Reduction of the graph in Fig. 1(b).

This study addresses the problem of minimizing VPN tunnels in a weighted VPN network topology, taking into account two interesting constraints – the first on the tunnel path length (TPL) of a reduced tunnel, and the second on the tunnel relay degree (TRD) of a VPN gateway. TPL concerns the propagation delay between the tunnel endpoints while TRD concerns the computing power and bandwidth of the tunnel relay gateway. Administrators can choose suitable upper bounds on TPL and TRD to meet these concerns. The number of tunnels reduced following minimization falls as the bounds on TPL and TRD become more restrictive. Related graphical problems, which belong to NPC [7], are defined and the *Zero-One Integer Programming (0-1 IP)* [8] [9] algorithm is applied to solve them.

The rest of the paper is organized as follows. Section II defines the problem of minimizing VPN tunnels under three conditions - no constraint, TPL constraint and TRD constraint. Under the first condition, the optimal solution can be obtained simply by reducing the tunnels one bye one. However, this approach is ineffective for problems under the second and third conditions, so 0-1 IP is used to model them. Section III presents the effect after applying tunnel minimization to the topologies that an enterprise is likely to build. Section IV concludes this work.

## II. VPN TUNNEL MINIMIZATION PROBLEM

### A. Definition and Principle

A VPN tunnel enables private communication between two endpoints. Now, a tunnel is to be reduced using other tunnels as intermediate ones.

The intermediate VPN gateways (or "*VPN tunnel relay gateway"*) must have the authority to see data that originally belong to the endpoints of the reduced tunnel. The concept of authority is the basis of our model. The principle that formally describes how one VPN tunnel can be reduced is presented below. In the graph model, an edge represents a VPN tunnel; a vertex represents a VPN gateway, and the weight of a vertex represents the authority of a VPN gateway.

Given a connected, undirected graph G(V, E) and a vertex weight function $w$: V → N, $e_{ij}$ is the edge that connects vertices $v_i$ and $v_j$, and $P_{ij}$ represents the path whose endpoints are $v_i$ and $v_j$.

*Principle 1:* Principles of VPN Tunnel Reduction

An edge $e_{ij}$ can be *reduced* if and only if there exists a path $P_{ij}$ such that

$$w(v_k) > \max\{w(v_i), w(v_j)\}\ \forall v_k \in P_{ij}, k \neq i, j\ . \qquad \square$$

$e_{ij}$ is said to *reference* $e_{mn}$ if $e_{mn} \in P_{ij}$ and $e_{ij}$ is said to *reference* $v_k$ if $v_k \in P_{ij}$, given $P_{ij}$ is the *reduction path* of $e_{ij}$.

According to Principle 1, more than one reduction path may exist for reducing an edge. Therefore, $T_{ij}$ is defined as the set of reduction paths that can be used to reduce $e_{ij}$, and $P_{ij}^y$ is used as the $y$th path in the set $T_{ij}$ to reduce $e_{ij}$. For example, consider graph G in Fig. 3, in which each vertex is associated with a name and an authority level.

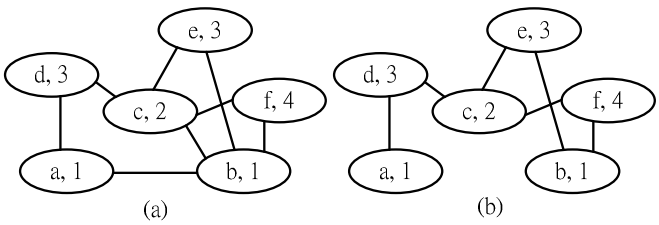


Fig. 3. (a) Graph G with four authority levels. (b) After tunnel reduction.

The following reductions can be derived from graph G.

$e_{ab}$: $P_{ab}^1$ {a, d, c, b}, $P_{ab}^2$ {a, d, c, e, b}, $P_{ab}^3$ {a, d, c, f, b}.

$e_{bc}$: $P_{bc}^1$ {b, e, c}, $P_{bc}^2$ {b, f, c}.

Three reduction paths are available by which $e_{ab}$ can be reduced, and two reduction paths exist by which $e_{bc}$ can be reduced. $T_{ab}$ is { $P_{ab}^1$, $P_{ab}^2$, $P_{ab}^3$ }, and $T_{bc}$ is { $P_{bc}^1$, $P_{bc}^2$ }. Suppose $P_{ab}^1$ is chosen to reduce $e_{ab}$ and $P_{bc}^2$ is chosen to reduce $e_{bc}$. Then, the reduction path of $e_{ab}$ becomes {a, d, c, f, b}, which is the combination of $P_{ab}^1$ and $P_{bc}^2$, and is referred to as a *recursive edge reduction* of $e_{ab}$. Also, $e_{ab}$ is said to *directly* reference $e_{ad}$, $e_{dc}$, $e_{cb}$, $v_d$ and $v_c$, and to *indirectly* reference $e_{cf}$, $e_{fb}$ and $v_f$. The first problem in this section can therefore be described as follows.

*B. VPN tunnel minimization without constraints*

*Problem 1*: VPN Tunnel Minimization.

Given a connected, undirected graph G(V, E), and a vertex weight function *w*: V → N, reduce as many edges as possible such that |E| is the minimum. □

Whether the reductions of the edges are independent, meaning that the reduction of an edge does not affect the reduction of another, must be determined to solve this problem. If reductions are independent, then the edges can be reduced one by one. The only possible situation in which the reductions of edges are mutually dependent is the *loop of edge reduction*, in which two reduced edges reference each other so that the reduction process never terminates. However, this situation cannot arise, as proven below.

*Theorem 1:* There is no loop of edge reduction in problem 1.

*Proof:* By contradiction.

Suppose there is a loop of edge caused by $e_{ij}$ and $e_{mn}$. $e_{ij}$ can reference $e_{mn}$ when $max\{w(v_i), w(v_j)\} < min\{w(v_m), w(v_n)\}$, $e_{mn}$ can reference $e_{ij}$ when $max\{w(v_m), w(v_n)\} < min\{w(v_i), w(v_j)\}$. This is a contradiction. So, there is no loop of edge reduction. □

*Algorithm for Problem 1:*

According to Theorem 1, reductions of edges are independent.

(1) For each edge $e_{ij}$,

(a) find the reduction path $P_{ij}$.

(b) If $P_{ij}$ exists, E ← E/ $e_{ij}$.

(2) G(V, E) is the desired solution. □

*C. Tunnel minimization with restrictions on TPL and TRD*

VPN tunnels can be minimized with some important restrictions. This section proposed two restrictions - one on the Tunnel Path Length, TPL, and one on the Tunnel Relay Degree, TRD. The TPL of a reduced edge $e_{ij}$ is the length of the reduction path. People may want to limit the TPL of a reduced edge to refrain from encrypting/decrypting and hashing too often, and to avoid wasting bandwidth. The TRD of a vertex $v_k$, however , represents the number of reduced edges that directly reference the vertex $v_k$ due to the load on a VPN gateway; that is, a larger TRD of a vertex indicates a heavier load on this VPN gateway. Accordingly, setting of an upper bound on TPL of every edge to TPL(G) may be desired when performing tunnel minimization on graph G. The setting of such a bound requires the TPL of every reduced edge to be less than or equal to the TPL(G). Similarly, the TRD of every vertex can be limited to TRD(G) such that no vertex has a TRD that exceeds TRD(G). Nevertheless, reductions of different tunnels are no longer independent when these two restrictions are applied to the VPN tunnel minimization problem. These two problems will be formally modeled as 0-1 IP problems. Polynomial-time approximation algorithms for solving these problems are also mentioned.

**Tunnel minimization with restriction on TPL.**

*Problem 2:* VPN Tunnel Minimization with restriction on TPL

Given a connected, undirected graph G(V, E), a vertex weight function *w*: V → N and the TPL of every reduced edge ≤ a constant, TPL(G) , minimize |E|. □

The algorithm for solving problem 2 is described below. The objective function and constraints must be determined to formulate this problem as a 0-1 IP problem. The goal is to minimize the number of edges, so variables in the objective function may represent the edges. However, the edges do not, themselves, compete with each other for reduction; rather, the reduction paths compete because every edge may be reducible by more than one reduction path . Therefore, the objective function is defined as *maximizing* $\sum_{\forall P_{ij}^y} P_{ij}^y$ (step 2). The value of $P_{ij}^y$ can be 1 or 0, where 1 means that the path is selected to reduce $e_{ij}$, while 0 implies otherwise. Clearly, the constraints $\sum_{P_{ij}^y \in T_{ij}} P_{ij}^y \le 1, \forall T_{ij}$ apply (step 3). These constraints state that only one reduction path can be used to reduce an edge at one time. The constraints derived from the restriction on TPL are required to enable the 0-1 IP algorithm to be applied. Recursive edge reduction can increase the TPL of an edge, so for each reduction path, $P_{ij}^y$, every possible recursive edge reduction must be discovered first. When the TPL of $e_{ij}$ exceeds TPL(G), the constraint $\sum_{P_{qk}^z \in S} P_{qk}^z + P_{mn}^x \le |S|$ (step 4) is derived, in which *S* is the set of reduction paths in a recursive edge reduction, and $P_{mn}^x$ represents the last reduction path in the recursive edge reduction.

*Algorithm for Problem 2:*

(1) for each $e_{ij}$, find corresponding $T_{ij}$;

(2) output the objective function which is needed by 0-1 IP;
maximize $\sum_{\forall P_{ij}^y} P_{ij}^y$;

(3) output constraints which are needed by 0-1 IP;
for each $T_{ij}$, output constraint $\sum_{P_{ij}^y \in T_{ij}} P_{ij}^y \le 1$;

(4) output constraints which are needed by 0-1 IP;
for each $P_{ij}^y$, if $|P_{ij}^y| \le$ TPL(G) then

(a) $S \leftarrow \Phi$;

(b) $S \leftarrow S \cup \{P_{ij}^y\}$;

(c) search constraints caused by recursive edge reductions of $P_{ij}^y$: *Find_Constraint* ( $P_{ij}^y$ );

(d) search constraints among those constraints generated from step (4.b).

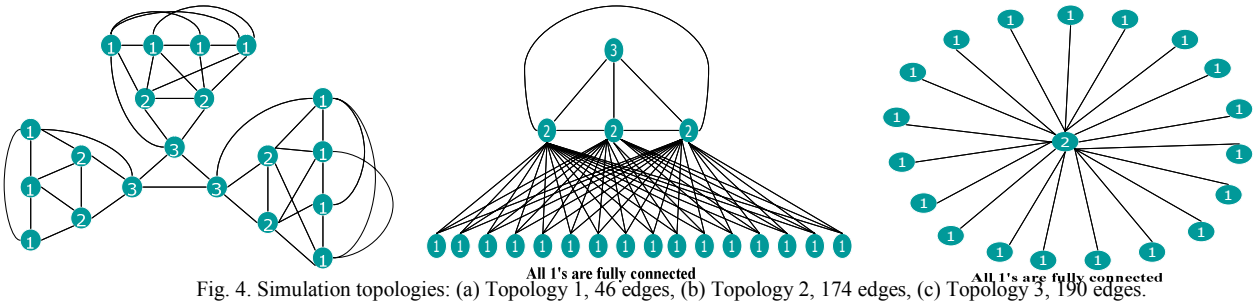

Fig. 4. Simulation topologies: (a) Topology 1, 46 edges, (b) Topology 2, 174 edges, (c) Topology 3, 190 edges.

*Find_Constraint* ( $P_{ij}^{y}$ )

(a) for each $e_{mn} \in P_{ij}^{y}$ ;

for each $P_{mn}^{x} \in T_{mn}$ ;

if $\sum_{P_{qk}^{z} \in S} | P_{qk}^{z} | + | P_{mn}^{x} | - |S| \le$ TPL(G);

$S \leftarrow S \cup \{P_{mn}^{x}\}$ ; *Find_Constraint* ( $P_{mn}^{x}$ );

else output the constraint

$\sum_{P_{qk}^{z} \in S} P_{qk}^{z} + P_{mn}^{x} \le |S|$;

(b) $S \leftarrow S / P_{ij}^{y}$ .

(5) use 0-1 IP algorithm to solve this problem;

$P_{ij}^{y} \in \{0,1\}$ , $\forall P_{ij}^{y}$ ;

*objective function*: step (2);

*constraints*: step (3) and (4);

(6) according to step (5), if $P_{ij}^{y}$ =1, $E \leftarrow E / e_{ij}$, $\forall P_{ij}^{y}$;

(7) G(V, E) is the desired solution. □

**Tunnel minimization with restriction on TRD.**

*Problem 3:* Tunnel Minimization with restriction on TRD

Given a connected, undirected graph G(V, E), a vertex weight function *w*: V → N and the TRD of every vertex ≤ a constant, TRD(G), minimize |E|. □

Problem 3 is similar to problem 2, except in that the restriction is on TRD rather than TPL. Therefore, the algorithm for solving problem 3 differs from that for solving problem 2 only in step 4, which calculates the constraints derived from the restriction on TRD. Each vertex $v_k$ is assigned a set of reduction paths that reference this vertex directly after step (4a). Therefore, the constraint derived from the restriction on TRD is $\sum_{P_{ij}^{y} \in S_k} P_{ij}^{y} \le TRD(G) \ \forall S_k$ .

*Modified step 4 for Problem 3:*

(4) output constraints which are needed by 0-1 IP;

(a) $S_k \leftarrow \Phi \ \forall v_k \in V$ ;

for each $P_{ij}^{y}$ ;

if $P_{ij}^{y}$ references $v_k$ , then $S_k \leftarrow S_k \cup \{P_{ij}^{y}\}$ , $\forall v_k \in V$ ;

(b) for each $S_k$ , output constraint

$\sum_{P_{ij}^{y} \in S_k} P_{ij}^{y} \le TRD(G)$ ;

**Restrictions on both TPL and TRD**

Notably, although the algorithms above solve problems 2 and 3, respectively, an algorithm that considers TPL and TRD restrictions simultaneously can be easily established, by combining steps (4) in the two algorithms.

## III. Effect of VPN Tunnel Minimization

Section II addresses minimizing the tunnels, based on the authority of VPN gateways. However, whether this scheme can be deployed is not determined by the minimization of tunnels itself, but rather by the percentage of tunnels that can be reduced, $\text{RT\%} = \frac{\text{reduced tunnels}}{\text{original tunnels}} \times 100\%$ . RT% is affected not only by external factors, such as *TPL* and *TRD*, but also by internal factors such as the *VPN tunnel topology,* in terms of *authority distribution* and *tunnel connectivity*. The real-world VPN tunnel topology to which the authority scheme can most suitably be applied is the enterprise topology, which exhibits the following two generalized characteristics:

*(1) vertices with similar authority levels tend to be connected, and thus exhibit a higher tunnel connectivity than others;*

*(2) number of vertices with low authority level ≥ number of vertices with high authority level.*

Accordingly, four VPN tunnel topologies, containing 20 vertices, are minimized. Each topology has a different *authority distribution* and *tunnel connectivity*. The goal of this simulation is to identify how these factors influence RT%.

### A. Simulation setup

Figure 4 (a), (b) and (c) show three tunnel topologies, involving 20 vertices. The authority distribution of topology 1, T1, including three level-3 vertices, six level-2 vertices and 11 level-1 vertices, is represented by T1(3x3, 6x2, 11x1). The authority distributions of the other topologies are T2(1x3, 3x2, 16x1) and T3(0x3, 1x2, 19x1). T1 has 46 edges; T2 has 174 edges, and T3 has 190 edges. The number of edges in each topology is minimized according to various upper bounds on TPL and TRD, using the combination of algorithms 2 and 3.

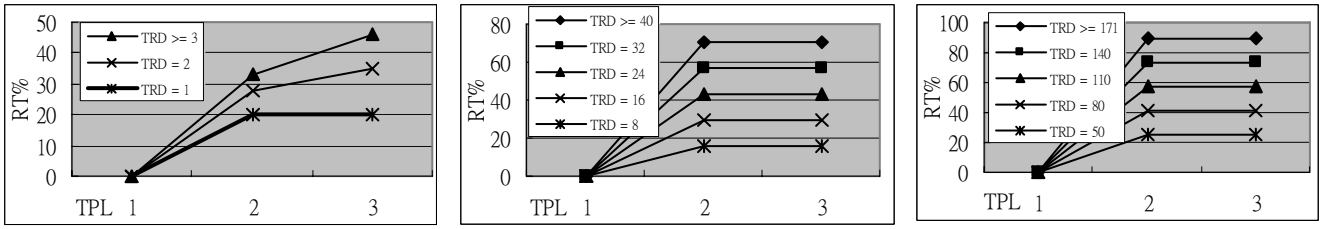


Fig. 5. Simulation result: (a) Topology 1, (b) Topology 2, (c) Topology 3.

*B. Simulation results and observations*

Some boundary conditions apply to the simulation results. First, RT% > 0 when TPL ≥ 2 and TRD ≥ 1, because the minimal tunnel path length of a reduced tunnel is 2, and some VPN tunnel relay gateway(s) must be able to relay at least one tunnel such that the reduced tunnel has the reduction path along which packets are transmitted. Second, RT% has an upper bound of $\frac{C_2^n-(n-1)}{C_2^n}\times 100\% = (1-\frac{n-1}{C_2^n})\times 100\%$, where n is the number of vertices in the graph. The proof follows.

*Theorem 2:* Given a graph with vertices n, RT% ≤ $(1-\frac{(n-1)}{C_2^n})\times 100\%$.

*Proof:* The definition of RT% is $\frac{\text{reduced tunnels}}{\text{original tunnels}}\times 100\%$, which can be rewritten as $\frac{\text{original tunnels - existing tunnels after minimization}}{\text{original tunnels}}\times 100\%$ or $(1-\frac{\text{existing tunnels after minimization}}{\text{original tunnels}})\times 100\%$.

Hence, RT% increases when the number of *original tunnels* increases or the number of e*xisting tunnels after minimization* declines. For a graph with n vertices, the maximum number of original tunnels is $C_2^n$ ; hence, when the graph is complete, the minimum number of tunnels after minimization is (n-1), which is the minimum number of edges required to keep a graph connected. □

Figure 5(a) presents the simulation results for T1. Only TPL = 3 and TRD = 3 are required to yield the maximum of RT% for this graph, which is 46%, since 21 tunnels can be reduced (21/46 = 46%). For TRD = 1, the maximum RT% is reached when TPL ≥ 2. For TRD ≥ 2, RT% is maximum when TPL ≥ 3.

Figure 5(b) displays the results of the simulation of T2. TPL = 2 and TRD = 40 are required to yield the maximum RT% for this graph, which is 71%, since 123 tunnels can be reduced (123/174 = 71%). For all TRDs, the RT% is maximum when TPL ≥ 2 because the VPN tunnel topology is so *centralized* that no tunnels can be reduced with TPL ≥ 3 following the reductions with TPL = 2.

Figure 5(c) plots the results of the simulation of T3. TPL = 3 and TRD = 171 are required to yield the maximum of RT% for this graph, which is 90%, and is also the upper bound on RT% in a graph with 20 vertices, since

$$(1-\frac{(20-1)}{C_2^{20}})\times 100\% = (1-\frac{19}{190})\times 100\% \cong 90\%.$$

This example establishes the upper bound on RT%, derived from Theorem 2, does exist. Therefore, the upper bound of RT% is minimal. This topology is like topology 2, so centralized that no tunnels can be reduced with TPL ≥ 3 following the reductions with TPL = 2 for all TRDs.

Some observations can be made. First, an enterprise with more *centralized* authority tends to have a larger maximum RT%. In these simulations, the maximum RT% (90%) in topology 3, which represents the most centralized distribution authority, is the highest of all the RT% values for the three topologies. However, TRD(G) must be ≥ 171, such that if one enterprise has such a VPN tunnel topology, it requires very powerful VPN gateways to act as a VPN tunnel relay gateway to minimize the tunnel topology. Second, an enterprise has less *centralized* authority distribution requires more gateways to implement the tunnel relay. Consequently, such an enterprise tends to have a lower TRD(G) to maximize RT%. Finally, TRD tends to more strongly affect RT% than does TPL when TPL(G) ≥ 2 because the vertices with similar authority levels tend to be connected, which is characteristic of an enterprise VPN tunnel topology.

## IV. CONCLUSION AND FUTURE WORK

This work presents the new concept of authority levels on tunneling endpoints, i.e. VPN gateways, and applies it in the tunnel minimization problem to reduce the number of tunnels. In the proposed authority scheme, VPN gateways with higher authority have the privilege of relaying the packets of the reduced tunnels, which are established by other VPN gateways with lower authority. The viability of this scheme is then theoretically proven and practically demonstrated. A graphical model is used to formalize VPN tunnel minimization problem with two interesting and useful constraints - namely the TPL and TRD. The problem is then modeled as a 0-1 Integer Programming problem, which can be solved in polynomial time by some approximation algorithms.

Although the TPL and TRD constraints are considered here to apply to two separate problems, they can be

considered together in a problem solved by uniting some constraints of 0-1 IP in algorithms for solving problems 2 and 3. Besides, the proposed algorithms can be easily modified and applied to situations in which reduced tunnels and vertices have different TPL and TRD upper bounds, respectively. Some conclusions are drawn regarding the possible features of an enterprise's VPN tunnel topology, and at most 90% of the tunnels are observed to be reducible in a VPN tunnel topology with 20 vertices.

As for future work, the TRD of a vertex may be redefined as the number of reduced edges that directly or indirectly reference that vertex. Then a polynomial-time algorithm for solving problem 3 with the redefined TRD should be obtained. If not possible, the problem should otherwise be proven to be of some complexity classes mentioned above. This new TRD of a vertex is more analogous than the original one since it considers recursive edge reduction. This problem can now be formulated as a *Zero-One Integer Nonlinear Programming* [10] problem, which is NP-hard.

## REFERENCES


[1] K. Hamzeh et al., "Point-to-Point Tunneling Protocol (PPTP)," RFC 2637, July 1999.

[2] W. Townsley et al., "Layer Two Tunneling Protocol (L2TP)," RFC 2661, August 1999.

[3] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[4] S. Kent and R. Atkinson, "IP Authentication Header," RFC 2402, November 1998.

[5] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, November 1998.

[6] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[7] Reuven Cohen and Gideon Kaempfer, "On the cost of virtual private networks," *IEEE/ACM Transactions on Networking* (TON'00), Page(s): 775-784, v.8 n.6, Dec. 2000.

[8] Thomas H. Cormen et al., *Introduction to Algorithms*, second edition, MIT Press, 2001.

[9] Linear Programming FAQ, http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html.

[10] Nonlinear Programming FAQ, http://www-unix.mcs.anl.gov/otc/Guide/faq/nonlinear-programming-faq.html.