Ring-Based Virtual Private Network Supporting a Large Number of VPNs

Osamu Honda, Hiroyuki Ohsaki, Makoto Imase
Graduate School of Information Science and Technology
Osaka University, Osaka, Japan
E-mail: o-honda@ist.osaka-u.ac.jp
oosaki@ist.osaka-u.ac.jp
imase@ist.osaka-u.ac.jp

Kazuhiro Matsuda
NTT Information Sharing Platform Laboratories
NTT Corporation, Tokyo, Japan
Email: matsuda.kazuhiro@lab.ntt.co.jp

***Abstract*— In this paper, we propose a simple but effective VPN mechanism called *RING-VPN (Ring-based Virtual Private Network)* that realizes a high scalability in terms of the number of VPNs. The key idea of our RING-VPN is to logically connect nodes in a ring topology for minimizing the number of IPsec tunnels. In our RING-VPN, each VPN node operates autonomously, making VPNs robust even in case of node and/or link failures. We also quantitatively evaluate the performance of our RING-VPN using mathematical analysis. We derive several important performance metrics of RING-VPN, such as VPN construction time, and VPN recovery time, as well as user-level performance metrics, such as minimum TCP throughput, round-trip time and packet loss probability. Furthermore, we validate our analysis by comparing numerical examples with simulation results. Through several numerical examples, we quantitatively demonstrate effectiveness of our RING-VPN in several network configurations.**

I. Introduction

VPN (Virtual Private Network) technologies, which virtually build virtual private networks, have been receiving attention as a technology to realize secure and reliable communication for these activities [1-3]. Compared to building conventional networks using dedicated lines, VPN requires substantially less expenditure to realize such communication.

At present, PP-VPN (Provider Provisioned VPN) [4] represented by MPLS (Multi Protocol Label Switching) [5] has been widely deployed. However, it has the following limitations. (1) The unit of participation in a VPN is a *site* and individual users are unable to selectively participate in a VPN. (2) Due to its limitations in protocols and/or hardware, it is only able to build a relatively small number, i.e., from tens to thousands, of VPNs. Another technology to build VPNs is IPsec VPN [6, 7], which enables each node to selectively participate in a VPN, and therefore clears the limitation (1). However, due to lack of dynamic configuration mechanism in IPsec [6, 7], IPsec tunnels must be created and/or removed manually by a network administrator. This means that, if the number of nodes participating in the VPN increases, the burden of maintaining IPsec tunnels will also progressively increase. Consequently, only a smaller number of VPNs can be built.

Several systems to build an overlay network over an existing IP network have been proposed [8-10, 10]. Such systems include the DVC (Dynamic VPN Controller) [8], which dynamically establishes and terminates IPsec tunnels between nodes to build secure and reliable VPNs. However, the DVC needs to use a full-mesh topology to establish IPsec tunnels. So, it can only build a relatively small number of VPNs, since the number of required IPsec tunnels will enormously increase as the number of nodes participating in the VPN increases.

The UMU-PBNM (the University of Murcia Policy-Based Network Management) [9] and the X-Bone [10, 11] are also systems to build VPNs. Different from [8], these systems are capable of establishing arbitrary topologies to reduce the number of IPsec tunnels. Nevertheless, the UMU-PBNM [9] is unable to achieve high reliability; since both connection and disconnection of nodes are not allowed once the topology has been established, it cannot cope with the failure of nodes [9]. On the contrary, the X-Bone [10] is unable to achieve good performance; since it adopts a two-layer VPN structure, communication delay between nodes becomes very large, leading performance degradation.

In this paper, we propose a ring-based VPN called *RING-VPN* that provides a secure and reliable network to a large number of virtual organizations. Our proposed RING-VPN dynamically builds secure and reliable VPNs with high scalability in terms of the number of VPNs by logically connecting nodes in a ring topology.

To assist various social activities over networks, VPN mechanism is required to build a VPN that connects dozens of users over a wide-area network. On the other hand, due to the ring topology of the RING-VPN, it is anticipated that performance such as transmission delay and throughput would drastically deteriorate as the number of participating nodes in the VPN increases [12, 13]. It is also expected that other important performance metrics such as the required time to build a VPN (VPN construction time) and the required time to recover from network failures (VPN recovery time) would be affected by the increase in nodes. However, the potential of RING-VPN — for example, the size of the network over which a RING-VPN can be built and the number of nodes that can participate in that VPN — have not yet been clarified.

Therefore, in this paper, we quantitatively evaluate the effectiveness of our proposed RING-VPN using mathematical

analysis. Specifically, we evaluate the impact of the number of participating nodes in a VPN as well as the impact of link bandwidth and propagation delay on the performance of a RING-VPN (e.g., VPN construction time, VPN recovery time, and minimum TCP throughput). We demonstrate the proposed RING-VPN shows good performance in terms of several performance metrics including VPN construction time and minimum TCP throughput when the number of nodes participating in the VPN is relatively small.

The structure of this paper is as follows. First, Section II explains our proposed RING-VPN. In particular, communication, connection, and disconnection controls of the RING-VPN are explained in detail. In Section III, we derive performance metrics of RING-VPN through mathematical analysis. These metrics include VPN construction time, VPN recovery time, and minimum throughput, round-trip time and packet loss probability of TCP flows. Additionally, several numerical examples are presented for demonstrating the characteristics of the RING-VPN. Simulation results are also presented for validating our approximate analysis. Finally, Section IV concludes this paper and discusses future works.

II. RING-VPN (Ring-based Virtual Private Network)

In this section, we explain our proposed RING-VPN. The key of our RING-VPN is to logically connect nodes in a ring topology for minimizing the required number of IPsec tunnels. In our proposed RING-VPN, different from other VPN technologies with a central management system such as represented by MPLS, each node autonomously finds nodes participating in a VPN, connects to the VPN, and then disconnects from the VPN. Because of this autonomy of each node, reliable communications can be realized even in cases of failures such as network devices failures and routing errors. In addition, IPsec tunnels are established between adjacent nodes to realize secure communications.

Let us define that the forward direction is counterclockwise in Fig. 1. Each VPN node establishes an IPsec tunnel to a downstream VPN node, and the IPsec tunnel is used only for unidirectional transmission from an upstream VPN node to a downstream one. Communication between VPN nodes is done via VPN nodes along the ring. For example, a packet transferred from VPN node 1 to VPN node 4 passes VPN node 2 and 3 in order and finally arrives at VPN node 4. It is anticipated that packets have to pass a large number of nodes in this type of ring-based network. Therefore, communication delay tends to be large and, as a result, TCP throughput would be degraded [12, 13].

In what follows, we will explain essential functions of the RING-VPN, i.e. control of communication between VPN nodes, control of VPN node connection to VPNs, and control of VPN node disconnection from VPNs.

A. Communication Control between VPN Nodes

In our RING-VPN, a VPN node transfers a packet to a downstream VPN node through an IPsec tunnel. The destination IP address of the encapsulated packet is always set to be the IP address of the downstream VPN node, regardless of the destination VPN node of the packet before encapsulation.

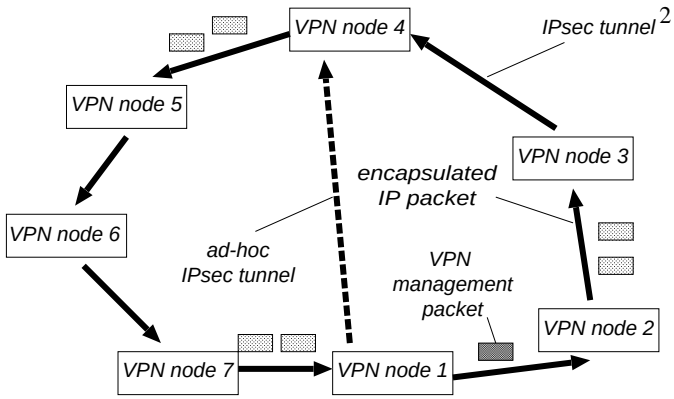

Fig. 1: Overview of our RING-VPN.

Upon the arrival of the encapsulated packet from the upstream VPN node, the VPN node compares the destination IP address of the packet before encapsulation with its own IP address. If one IP address matches the other, the VPN node decapsulates the packet and receives it. If one IP address does not match the other, the VPN node compares the source IP address of the packet before encapsulation with its own IP address, and if one IP address matches the other, the VPN node discards the encapsulated packet. The VPN node judges that the source IP address of the packet before encapsulation matches its own IP address. It is because the encapsulated packet has come back after circulating the ring-network, i.e., there existed no VPN node which was designated to receive the packet. If the source IP address does not match own IP addresses, then, the VPN node changes the destination IP address of the encapsulated packet to the IP address of its downstream VPN node, and then transfers this encapsulated packet.

B. VPN Node Connection Control

The control mechanism of the connection of a VPN node to the RING-VPN is described below. Hereafter, we denote a VPN node that is newly connecting to the RING-VPN as a *new VPN node*.

A new VPN node receives the list of the nodes already participating in the VPN and the information of the ring configuration from the VPN manager. The new VPN node then measures average round-trip time to each VPN node in VPN using ICMP Echo messages [14].

Based on the measured round-trip time, the new VPN node determines the position in the ring to which it should connect. In particular, for each adjacent node pair in the ring, the new VPN node measures the round-trip time to each of the upstream and downstream VPN nodes of the pair and, by summing up the time, calculates the total round-trip time for the pair. The VPN node pair with the minimum total round-trip time is determined as the position to which the new VPN

node connects. The new VPN node then establishes two IPsec tunnels: one to the upstream VPN node and the other one to the downstream VPN node. Lastly, the IPsec tunnel between the upstream VPN node and the downstream one, which is no longer required, is torn down.

C. VPN Node Disconnection Control

In our RING-VPN, VPN nodes autonomously re-establish the ring when the communication between VPN nodes terminated for some reason. Below is the control mechanism of the VPN node disconnection, during the re-establishment of the ring.

A VPN management packet circulates through VPN nodes along the ring. In this management packet, the list of the VPN nodes and the information of the ring configuration are recorded. Each VPN node returns the acknowledgment to its upstream VPN node every time the VPN node receives this management packet. Each VPN node judges that its downstream VPN node is not operational if it does not receive any acknowledgement after a certain period of time.

Any VPN node that detects a non-operational downstream VPN node notifies this to the further downstream VPN node connecting to the non-operational node. It then establishes a new IPsec tunnel to this further downstream VPN node. In addition, the IPsec tunnels between the VPN node and the non-operational one as well as between the further downstream VPN node and the non-operational one are terminated. Lastly, the VPN node, which detected the non-operational node, updates the list of the VPN nodes and the information of the ring configuration recorded in the management packet. Then, it transmits this packet to downstream VPN nodes.

III. Analysis

In this section, we first model the RING-VPN and derive its VPN construction time, VPN recovery time, and the minimum throughput, round-trip time and packet loss probability of TCP flows. We then quantitatively evaluate the performance of the RING-VPN through several numerical examples. Simulation results are also presented for validating our approximate analysis.

A. Analytic Model

The analytic model is shown in Fig. 2. We assume that there are N VPN nodes in the network, and each of which is called VPN node i ($1 \leq i \leq N$). Let $b_{i,j}$ be the link bandwidth between VPN node i and VPN node j, $d_{i,j}$ be the transmission delay between VPN node i and VPN node j, and $f_{i,j}$ be the number of TCP flows whose source is VPN node i and destination is VPN node j. Let $M = (m_{i,j})$ be the connectivity matrix of VPN nodes, and define $m_{i,j} = 1$ if there exists an IPsec tunnel between VPN node i and VPN node j and $m_{i,j} = 0$ otherwise.

Since the links between VPN nodes are unidirectional, the round-trip time of the VPN with N VPN nodes, R_N, is given

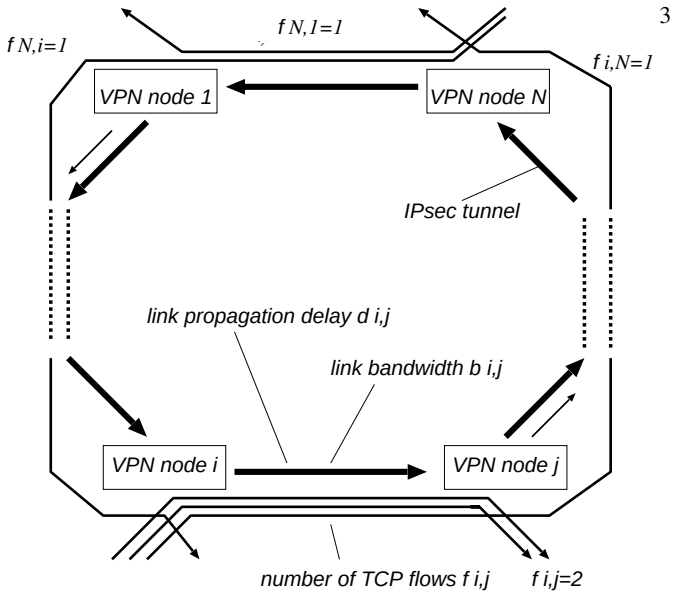

Fig. 2: Analytic model.

by

$$R_N = \sum_{i=1}^{N} \sum_{j=1}^{N} m_{i,j} d_{i,j}. \tag{1}$$

B. VPN Construction Time

We first derive VPN construction time X_N, which is defined as the time required for the N-th VPN node to join the VPN. N-th VPN node measures round-trip time to each of $N-1$ VPN nodes that already participated in the VPN. It then participates in and re-builds the VPN so that the increase in the round-trip time becomes the minimum. Specifically, let r and l denote the upstream VPN node and the downstream VPN node for a new VPN node N, respectively. Assuming that the round-trip time between VPN node i and VPN node j can be approximated by $2\,d_{i,j}$ using the transmission delay between the nodes, $d_{i,j}$, r and l are given by

$$\min_{l,r} \left(\frac{2d_{l,N} + 2d_{N,r}}{m_{l,r}} \right). \tag{2}$$

On finding the nodes to be connected, VPN node N re-builds the VPN as follows: (a) establish an IPsec tunnel to VPN node r; (b) establish another IPsec tunnel to VPN node l; and (c) terminate the IPsec tunnel between VPN node l and VPN node r. Assume that the round-trip time between VPN nodes participated in the VPN is known (i.e., the round-trip time was already measured and does not change with time). Typically, for creating an IPsec tunnel, six message exchanges between VPN nodes are required [6]. After message exchanges, an encryption key is generated. Let Δ be the time required for a VPN node to generate an encryption key. Provided that VPN node N executes the above process (a) and (b) in this order, and (c) can be executed in parallel with (a)

and (b), X_N is given by

$$\begin{aligned} X_N &= (6d_{N,r} + \Delta) + (6\,d_{l,N} + \Delta) \\ &= 6\,(d_{l,N} + d_{N,r}) + 2\Delta. \end{aligned} \tag{3}$$

Note that $X_0 = 0$ since the initial node does not establish any IPsec tunnel. Consequently, the VPN construction time is given by

$$X_N = \begin{cases} 0 & \text{if } N = 0 \\ 6\,(d_{l,N} + d_{N,r}) + 2\Delta & \text{otherwise} \end{cases} \tag{4}$$

C. VPN Recovery Time

Next, we derive the required time to re-build the VPN after the VPN node i disconnected (VPN recovery time), Y_i. Suppose the VPN node i disconnects from the VPN due to some failure. Then, an IPsec tunnel must be established between its upstream VPN node l ($m_{l,i} = 1$), and its downstream VPN node r ($m_{i,r} = 1$). In particular, detection of a VPN node failure and establishment of a new IPsec tunnel, as well as termination of no longer required IPsec tunnel, proceed as follows: (a) VPN node l detects a failure of VPN node i by non-return of the acknowledging packet. (b) VPN node l establishes an IPsec tunnel to VPN node r. (c) The IPsec tunnel between VPN node l and VPN node i is terminated. (d) The IPsec tunnel between VPN node i and VPN node r is terminated. In effect, both (c) and (d) can proceed in parallel with (b). Therefore, the VPN recovery time is given by

$$Y_i = 2d_{l,i} + 6d_{l,r} + 2\Delta. \tag{5}$$

D. Minimum Throughput, Round-Trip Time, and Packet Loss Probability of TCP Flows

We derive the minimum throughput, T_{min}, of the TCP flow, which has the minimum throughput among all TCP flows in the VPN. Hereafter, all TCP flows are assumed to continuously send data. The number of TCP flows, $n_{i,j}$, passing through VPN node i and VPN node j is given by

$$n_{i,j} = \sum_{l=0}^{N-1} \sum_{k=0}^{N-l-2} f_{i-k,j+k}, \tag{6}$$

where $f_{i,j}$ is the number of TCP flows established between VPN node i and VPN node j.

If the link between VPN node i and VPN node j is the bottleneck, VPN node i and VPN node j satisfy

$$\min_{i,j} \left(\frac{b_{i,j}}{n_{i,j} m_{i,j}} \right). \tag{7}$$

It should be noted that, since all of links between VPN nodes are unidirectional, the round-trip time of the TCP flow from VPN node i to VPN node j, $R_{i,j}$, is a constant. In other words,

$$R_{i,j} = R_N. \tag{8}$$

Therefore, T_{min} is given by

$$T_{min} = \min_{i,j} \frac{b_{i,j}}{n_{i,j}}. \tag{9}$$

Let p be the packet loss probability of TCP flows. Then, it is known that the throughput of a TCP flow, T, can be approximated by the following equation [15].

$$T \simeq \frac{1}{2\,R_N}(-3 + \frac{\sqrt{6 + 21\,p}}{\sqrt{p}}) \tag{10}$$

By letting $T = T_{min}$, the packet loss probability of TCP flows is obtained as

$$p \simeq \frac{3}{2\,R_N\,T_{min}(3 + R_N\,T_{min}) - 6}. \tag{11}$$

E. Numerical Examples

The sensitivity of the performance metrics such as VPN construction time, VPN recovery time, and the minimum throughput, round-trip time and packet loss probability of TCP flows regarding several parameters such as the number of VPN nodes, link bandwidth, link transmission delay, and the number of TCP flows is of our interest.

In particular, transmission delay and throughput are expected to decrease as the number of VPN nodes increases for ring-based networks [12, 13]. Therefore, we investigate the impact of the number of VPN nodes participating in the VPN, N, on VPN construction time, X_N, VPN recovery time, Y_N, and minimum TCP throughput, T_{min}, and round-trip time, R_N, by changing N. The transmission delay between VPN nodes, $d_{i,j}$, are set to be uniformly distributed in the range of 0–D_{max}. The link bandwidth between VPN nodes, $b_{i,j}$, is fixed at 10 [Mbit/s]. Table I shows the parameters used in numerical examples and simulation.

For simulation, we implemented the RING-VPN module in OPNET modeler version 10.0A [16]. For each parameter set, simulation is repeated 5 times with varying the seed of a random number generator, and the average of VPN construction times, VPN recovery times, and round-trip times of TCP flows are measured.

Figures 3 through 6 show the changes in VPN construction time, X_N, VPN recovery time, Y_N, minimum TCP throughput, T_{min}, and round-trip time, R_N, when the number of VPN nodes, N, is changed.

It can be seen from Fig. 3 that VPN construction time, X_N, decreases as the number of VPN nodes, N, increases. This is because of the increased density of VPN nodes. Namely, as the number of VPN nodes increases, the density of VPN nodes in the network becomes high. Hence, a new VPN node is likely to be connected to VPN nodes with short round-trip times, leading short VPN construction time. Moreover, Fig. 3 shows good agreement between numerical examples and simulation results, indicating validity of our approximate analysis.

Figure 4 shows that VPN recovery time decreases as the number of VPN nodes increases. This can be explained by the decrease in the average delay between each VPN node due to the increased number of VPN nodes. The figure also shows that VPN recovery time increases with the increase of the transmission delay between VPN nodes. This may be because the required time to establish and terminate IPsec tunnels increases as the transmission delay between nodes becomes

TABLE I

PARAMETERS USED IN NUMERICAL EXAMPLES

Number of VPN nodes	N	8–40	
Maximum transmission delay between VPN nodes	D_{max}	10, 100	[ms]
Link bandwidth between VPN nodes	$b_{i,j}$	10	[Mbit/s]
Time required for generating an encryption key	Δ	100	[ms]
Number of TCP flows between VPN nodes	$f_{i,j}$	3	

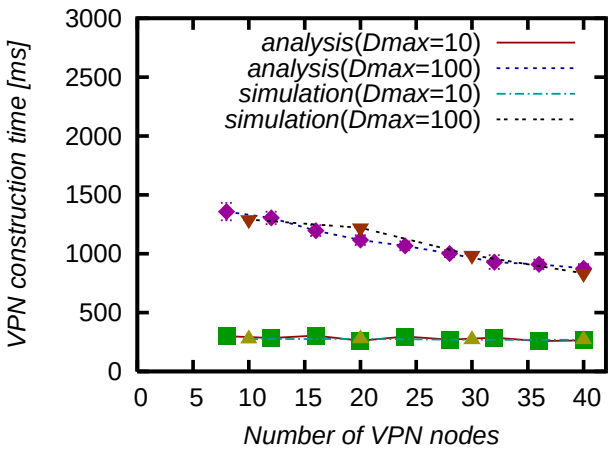

Fig. 3: Number of VPN nodes N vs. VPN construction time X_N.

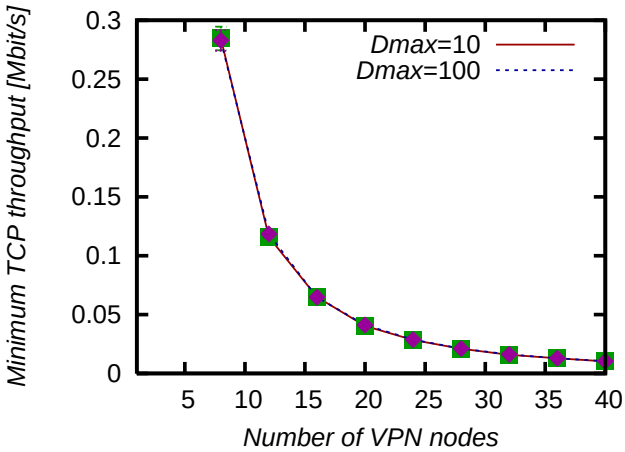

Fig. 5: Number of VPN nodes N vs. minimum TCP throughput T_{min}.

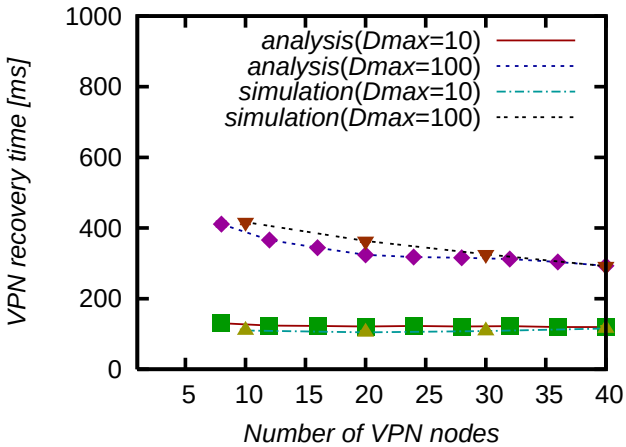

Fig. 4: Number of VPN nodes N vs. VPN recovery time Y_N.

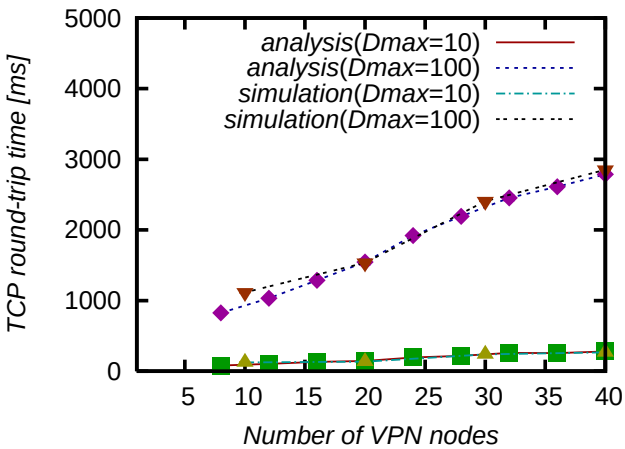

Fig. 6: Number of VPN nodes N vs. TCP round-trip time R_N.

larger, as can be seen from Eq. (5). Again, this figure shows good agreement between numerical examples and simulation results, indicating validity of our approximate analysis.

From these observations, we conclude that both the VPN construction time and the VPN recovery time are not so sensitive to the number of VPN nodes.

Figure 5 shows that the minimum TCP throughput rapidly decreases as the number of VPN nodes increases. This can be explained by the increased number of TCP flows sharing the bandwidth of the bottleneck link. Also, this figure shows that the transmission delay between VPN nodes does not affect the minimum throughput of TCP flows. This is because, as Eq. (9) indicates, the throughput of a TCP flow is dependent only on the link bandwidth and the number of TCP flows. Note that Fig. 5 shows the *minimum* TCP throughput, instead of the average TCP throughput.

Figure 6 shows that the round-trip time of TCP flows increases almost linearly as the number of VPN nodes increases. This is probably because in our RING-VPN, the round-trip time is the sum of all transmission delays (see Eq. (1). The

figure also shows that the round-trip time increases as the transmission delay between VPN nodes becomes larger.

From these observations, we conclude that the proposed RING-VPN shows good performance in terms of the minimum TCP throughput and round-trip time when the number of nodes participating in the VPN is relatively small.

IV. Conclusion

In this paper, we have proposed a ring-based VPN called RING-VPN, which builds VPNs by logically and autonomously connecting VPN nodes in a ring topology. Our RING-VPN can build a large number of VPNs over the existing network by reducing the required cost to maintain IPsec tunnels. Since each node operates autonomously, our RING-VPN can realize highly reliable communication even in case of network failures. We have mathematically analyzed the performance of the RING-VPN. In particular, we have modeled the RING-VPN and have derived VPN construction time, VPN recovery time, and the minimum throughput, round-trip time and packet loss probability of TCP flows. We have also quantitatively demonstrated the effectiveness of the RING-VPN through several numerical examples.

Our future works include evaluation of the effectiveness of advanced communication control mechanisms using ad-hoc tunnels and assessment of the impact of VPN node failures on, for instance, VPN construction time and VPN recovery time.

References

[1] M. Carugi and D. McDysan, "Service requirements for layer 3 provider provisioned virtual private ne tworks (PPVPNs)," *Request for Comments (RFC) 4031*, Apr. 2005.

[2] B. Gleeson *et al.*, "A framework for IP based virtual private networks," *Request for Comments (RFC) 2764*, Feb. 2000.

[3] A. Nagarajan, "Generic requirement for provider provisioned virtual private networks (PPVPN)," *Request for Comments (RFC) 3809*, June 2004.

[4] R. Callon and M. Suzuki, "A framework for layer 3 provider provisioned virtual pricate networks PPVPNs," *Request for Comments (RFC) 4110*, July 2005.

[5] E. Rosen and Y. Rekhter, "BGP/MPLS VPNs," *Request for Comments (RFC) 2547*, Mar. 1999.

[6] S. Kent and R. Atkinson, "Security architecture for the Internet protocol," *Request for Comments (RFC) 2401*, Nov. 1998.

[7] S. Kent and K. Seo, "Security architecture for the internet protocol," *Request for Comments (RFC) 4301*, Dec. 2005.

[8] "Dynamic VPN controller (DVC) demonstrator project report," http://www-mice.cs.ucl.ac.uk/multimedia/meetings/vpnworkshop/ documentat%ion/proceedings/2003-11-12/VPN-KMS-Issues.doc, Oct. 2003.

[9] A. Gomez, G. Martinez, and O. Canovas, "New security services based on PKI," *Future Generation Computer Systems*, vol. 19, no. 2, pp. 251–262, Jan. 2003.

[10] J. Touch, "Dynamic internet overlay deployment and management using the X-Bone," *Computer Networks*, vol. 36, no. 2, pp. 117–135, July 2001.

[11] J. Touch, Y. Wang, V. Pingali, L. Eggert, R. Zhou, and G. Finn, "A global X-Bone for network experiments," in *Proceeding in IEEE Tridentcom 2005*, Mar. 2005, pp. 194–203.

[12] M. Ergen, D. Lee, R. Attias, S. Tripakis, A. Puri, R. Sengupta, and P. Varaiya, "Wireless token ring protocol," in *Proceedings of SCI (Systems, Cybernetics and Informatics)*, vol. 4, July 2002.

[13] R. Jain, "Performance analysis of FDDI token ring networks: Effect of parameters and guidelines for setting TTRT," *IEEE Lightwave Telecommunication Systems*, vol. 20, no. 2, pp. 16–22, May 1991.

[14] J. Postel, "Internet control message protocol," *Request for Comments (RFC) 792*, Sept. 1981.

[15] H. Ohsaki, J. Ujiie, and M. Imase, "On scalable modeling of TCP congestion control mechanism for large-scale IP networks," in *Proceedings of IEEE SAINT 2005*, Feb. 2005, pp. 361–369.

[16] Opnet Technologies, Inc., "OPNET," http://www.opnet.com/.