Tunnel Gateway Satisfying Mobility and Security Requirements of Mobile and IP-Based Networks

Younchan Jung and Marnel Peradilla

Abstract: **Full-mesh IPSec tunnels pass through a black ("unsecure") network (B-NET) to any red ("secure") networks (R-NETs). These are needed in military environments, because they enable dynamically changing R-NETs to be reached from a B-NET. A dynamically reconfiguring security policy database (SPD) is very difficult to manage, since the R-NETs are mobile. This paper proposes advertisement process technologies in association with the tunnel gateway's protocol that sends 'hello' and 'prefix advertisement (ADV)' packets periodically to a multicast IP address to solve mobility and security issues. We focus on the tunnel gateway's security policy (SP) adaptation protocol that enables R-NETs to adapt to mobile environments and allows them to renew services rapidly soon after their redeployment. The prefix ADV process enables tunnel gateways to gather information associated with the dynamic changes of prefixes and the tunnel gateway's status (that is, 'down'/restart). Finally, we observe two different types of performance results. First, we explore the effects of different levels of R-NET movements on SP adaptation latency. Next, we derive the other SP adaptation latency. This can suffer from dynamic deployments of tunnel gateways, during which the protocol data traffic associated with the prefix ADV protocol data unit is expected to be severe, especially when a certain tunnel gateway restarts.**

Index Terms: **Adaption latency, IPSec tunnels, mobile internet protocol (IP), prefix advertisement, security policy, tunnel gateway.**

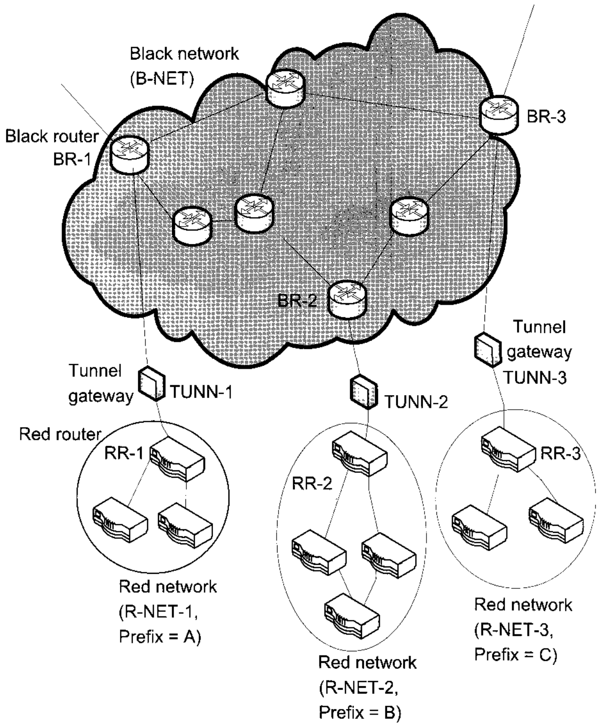


Fig. 1. Tunnel gateway and red/black networks (R-NET/B-NET).

I. INTRODUCTION

The current conduct of public protection and disaster relief operations faces significant challenges from problems resulting from incompatible communication systems. In the case of international disaster relief responses to disasters, such as earthquakes, different communication systems need to be consolidated. Mobile and internet protocol (IP)-based network usage is inevitable, because IP technology is transparent to application layer entities: a city fire department, police force, and medical emergency applications as a solution to this issue [1], [2].

Future mobile ad hoc networks will need to be able to associate moving domain networks with the IP backbone network [3]. As shown in Fig. 1, "secure" and "unsecure" networks are denoted as red and black networks, respectively. In the near future, highly mobile groups such as military users, civilian police, fire fighters and emergency rescue units will use IP-based "secure" and "unsecure" networks with few exceptions, in dynamically moving domains [4]. Such typical examples of mobile and IP-based networks need to satisfy requirements to seamlessly handle their mobility. As shown in Fig. 1, the architecture of IP-based "secure" and "unsecure" networks consists of a black network (B-NET), red networks (R-NETs) and tunnel gateways (TUNNs). The term "tactical domain" refers to a domain associated with multiple R-NETs under the control of the same tunnel gateway [5], [6]. In that case, inter-domain communication requires the use of a B-NET. When an R-NET moves between different tactical domains, the local R-NET prefix needs to be changed. In that case, the local tunnel gateway needs to be able to learn about changes in the local R-NET prefix.

It is crucial to control configuration and mobility management protocol traffic effectively in dynamically changing networks (e.g., R-NETs), to ensure seamless communication between remote peer R-NETs. The term "IP-based secure and unsecure networks" can apply to the following example network configurations. A moving domain network, which is termed a R-NET in this paper, is an autonomous system of mobile nodes in which all domain routing uses the same address space prefix. While intra-domain routing may be achieved using either open shortest path first (OSPF) or routing information proto-

Manuscript received April 24, 2011.

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0015200) and by the Research Fund, 2011 of The Catholic University of Korea.

The authors are with the School of Information, Communications and Electronics Engineering, Catholic University of Korea, Puchon, Korea, email: ycjung@catholic.ac.kr, marnel_p@yahoo.com.

1229-2370/11/$10.00 © 2011 KICS

col (RIP), inter-domain communication requires use of the IP backbone network. The IP backbone network provides a B-NET where full-mesh IPSec tunnels are included so that all R-NETs behind the tunnel gateway can be reached [4], [7], [8]. Two different R-NETs, which belong to different tunnel gateways, can communicate by passing through the tunnel gateway associated with each R-NET. When a R-NET moves between different domains, it is necessary to allocate a domain specific routing address (called an R-NET Prefix in this paper). The tunnel gateway needs to be able to learn about which R-NET associates with which tunnel gateway, since a R-NET's routing address may change the tunnel gateway associated with it. This issue can be relatively easily solved in the conventional Internet, by exchanging the IP-prefix path vector advertised between two adjacent autonomous systems [9]. This is the key role of BGP routing. The R-NETs in the tactical domain may require the highly dynamic nature of ad hoc networking [3]. Strict security and reliability requirements combined with the dynamic nature of the network, necessitate reliable capabilities: R-NET monitoring and reporting that enables a collection of R-NET status information.

In this paper, we aim to resolve two issues related to inter-domain routing between dynamically changing R-NETs. The first contribution of this paper is related to providing mobility support when an R-NET appears and disappears from a tunnel gateway. To this end, the local tunnel gateway needs to update the security policy (SP) mapping database that associates each tunnel gateway with all the SPs corresponding to its local R-NETs. In that case, the tunnel gateway will function as a multicast host, which informs other tunnel gateways of changes in its local R-NET prefix. We have called this process 'advertisement of R-NET prefixes.' The second contribution of this paper is related to the case in which the tunnel gateway goes 'down'/'restarts.' It applies to a highly mobile network architecture formed over wireless links that are susceptible to failure. Each tunnel gateway is responsible for multicasting its state in every 'hello' timer interval to the other tunnel gateways. If the hello protocol data unit (PDU) from the same tunnel gateway is not received within a predetermined time interval, the tunnel gateway is declared to be down. The protocol then removes all SPs associated with the tunnel gateway that is down. In addition, when a restart of a remote tunnel gateway is detected, all SPs associated with that tunnel gateway are removed. Every tunnel gateway multicasts all of its local R-NET prefix information, immediately after a restart of a remote tunnel gateway is detected. It is difficult to avoid stale SPs that may later cause uniformity problems in the database that associates with each tunnel gateway in dynamically changing environments. Thus, our approach to tunnel gateways with multiple attached R-NETs is to distribute the addresses of networks that have not changed as well as the changes in attached networks. Thus, the data for each prefix advertisement (ADV) includes the whole set of prefixes of the R-NET related to a certain tunnel gateway.

In Section II, we discuss related works. In Section III, we outline the advertisement process of a tunnel gateway's status and the R-NET prefixes that is the main role of the tunnel gateway. Section IV lists recommendations of our advertisement process for security association (SA) adaptation, the analysis model for the advertisement protocol data traffic and analysis results. Our conclusions follow in Section V.

II. RELATED WORKS

The IPSec tunnel gateway has been proposed to interconnect mobile ad hoc networks to a large-scale IP network [4]. Manually configuring IPSec tunnels of tunnel gateways and security policies is labor intensive and difficult to manage. In some cases, full-mesh IPSec tunnels are required to ensure that all the R-NETs behind the IPSec tunnels are reachable. The main solution of [4] is to ensure that a tunnel gateway has the capability to dynamically learn about changes of R-NET prefixes behind the Red router. When the local R-NET prefixes change, the tunnel gateway will inform the other tunnel gateways to update their SPs by sending new R-NET prefix ADV PDUs that represent the changes [10]. When the remote tunnel gateway receives the updated R-NET prefix ADV PDUs, it can either add SPs for the new prefixes learned or delete SPs for the prefixes that no longer exist.

The tunnel gateway serves as two types of multicast host [11]. The first type of host is a multicast host of fully-meshed tunnels that run in tunnel mode encapsulating security payload (ESP) for secure transmission of prefix ADV services. Since the B-NET is realized over commercial IP networks, enabling dual-use of commercial and military networks, the enemy can easily access important information such as the location of operational mobile units. Hence, for security reasons, no R-NET prefixes should be visible in the B-NET. The approach to broadcast hello packets can be used to provide a means to interconnect mobile ad hoc networks to fixed IP networks [12]. The second type of host is a multicast host for un-encrypted hello PDUs. This avoids the problem of hello PDUs being visible in the B-NET. The two types of ESP tunnels are required to support IP Multicast routing for protocol data of prefix ADV PDUs and to encrypt transmission of data traffic between R-NETs, respectively. The tunnel gateway sends hello PDUs for which the destination IP address is a multicast address. In contrast, it sends prefix ADV PDUs over an ESP tunnel for which the destination IP address is another multicast address. The tunnel gateway also will function to provide a secure communication path in the B-NET routing domain for data between R-NETs.

As shown in Fig. 2, the protocol data unit deals with two kinds of packets: Hello and R-NET prefix ADV packets. The un-encrypted hello packet is periodically sent over a multicast address for the tunnel gateway discovery process. The encrypted prefix ADV packet is sent over a multicast ESP tunnel for the R-NET discovery process. At the end of the R-NET discovery process, the tunnel gateway will have a full-mesh SAs and SP ready to encrypt and forward any packets that arrive from the local R-NETs.

The tunnel gateway detects if a remote tunnel gateway is 'down,' if the Hello Packets from the remote tunnel gateway are not received within the duration of a given number of instances of the Hello timer intervals. When a tunnel gateway that is down is detected, all policies associated with the tunnel gateway that is down are deleted from the security policy database (SPD). The

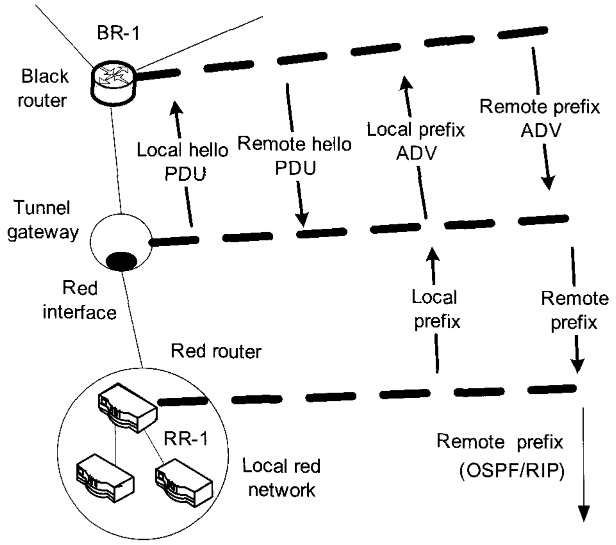


Fig. 2. Hello and prefix ADV PDUs.

tunnel gateway is required to run intra-domain routing protocols, such as OSPF or RIP, on the red interface to learn all the R-NET prefixes behind the red router to support dynamic routing between R-NETs. R-NET prefix ADV packets, which contain all the local prefixes, are periodically sent (if necessary) to the multicast IP address dedicated for the protocol data of prefix ADV PDUs. The prefix ADV packets are encrypted and sent over the multicast ESP tunnel. That is, as the local tunnel gateway monitors the local prefix for changes, in the outbound direction (red router to black router), local security policies in the tunnel gateway are updated dynamically based on the local updates of the routing table via the OSPF/RIP routing protocol. In the inbound direction (black router to red router), any security policy updates (adds or deletes) that are requested by the peer tunnel gateway will be redistributed to the routing table of the OSPF/RIP routing protocol in the local R-NET domain. Thus, the local R-NET router has a route to a remote R-NET prefix in its routing table only if its tunnel gateway has a SA in association with the remote R-NET prefix. In this case, because each SPD entry contains its SA, SA, and SP represent the same terminology in this paper.

III. Proposed Advertisement of Tunnel Gateway's Status and R-NET Prefixes

A. *R-NET Prefix Advertisements*

The major related work explored by Tran [4] dealt with a proactive protocol that provides a simple mechanism to discover a new tunnel gateway or to detect when a tunnel gateway in the network is down, focusing on scalability. However, we aim to resolve two issues: How to provide mobility support when an R-NET appears and disappears from a tunnel gateway, and how to manage status changes of tunnel gateways when one goes 'down'/'restarts.' We focus on robust operations that can provide additional benefits in order to overcome uniformity problems in the database that associates with each tunnel gateway in dynamically changing environments. After completion of the prefix ADV multicast from a certain tunnel gateway, every other tunnel gateway will update the whole set of SPs associated with that one (recall that our approach to tunnel gateways is to distribute the prefixes of R-NETs that have not changed, as well as changes in attached networks. Thus, the data of each prefix ADV includes the whole set of prefixes of the R-NET related to a certain tunnel gateway). Here, we should take note that the prefix is different from the address. An address is usually allocated to each end device, while a prefix number is assigned to each R-NET. This means that if a tunnel gateway multicasts address information, the size of the advertisement message will increase in proportion to the total number of end devices attached to a certain R-NET. However, our advertisement mechanism distributes prefix ADVs. So, if there are N R-NETs attached to a certain tunnel gateway, the tunnel gateway will multicast an advertisement message containing N prefix numbers.

If an outbound SP is found for an R-NET IP packet, it will be encrypted based on its SA information and sent over the appropriate tunnel destined to a remote tunnel gateway, otherwise, it will be dropped. If an R-NET behind a red router associated with a remote tunnel gateway is down, a local tunnel gateway continues to encrypt and forward IP packets to that R-NET prefix as long as there is an outbound SP to that R-NET prefix in the local tunnel gateway's SPD. In that case, an important issue is to ensure that a tunnel gateway has the capability to learn about dynamic changes of R-NET prefixes behind the remote red router. If the local R-NET prefixes change, the tunnel gateway needs to inform other tunnel gateways to update their SPs, by sending new R-NET prefix ADV PDUs containing the new set of all R-NET prefixes that represent the changes. When the remote tunnel gateway receives the updated R-NET prefix ADV PDU, it can either add the SPs for the new prefixes learned from the ADV PDU, or delete the SPs for prefixes that are no longer available. According to [4], one approach to ensure that a tunnel gateway has the capability is to run an intra-domain routing protocol (OSPF or RIP) on the red interface to the red router. The intra-domain routing protocol will provide the tunnel gateway with the routing information changed in its R-NETs. In that case, the tunnel gateway will inform other tunnel gateways via prefix ADV PDUs, immediately after the routing database in the local Red interface is changed. The data of the prefix ADV PDU includes all local R-NET prefix routing database information.

Fig. 3 shows that R-NET where prefix B moves from the TUNN-2 domain to the TUNN-3 domain. When TUNN-2 detects that the R-NET prefix B is no longer in its routing table as it is disconnected, TUNN-2 sends an update prefix ADV PDU to TUNN-1 and TUNN-3. This causes removal of all SPs associated with prefix B. Conversely, when the R-NET prefix B appears in the routing table of the red interface in TUNN-3, this starts multicasting an update prefix ADV PDU to TUNN-1 and TUNN-2. Then, TUNN-1 and TUNN-2 will create SPs for prefix B associated with TUNN-3.

B. *Advertisement Process for SA Adaptation*

We assume that B-NET supports IP multicast routing. Then, the tunnel gateway will function as an end host in the B-NET multicast routing domain. That requires two IP multicast addresses. One multicast address is used for the hello PDU and

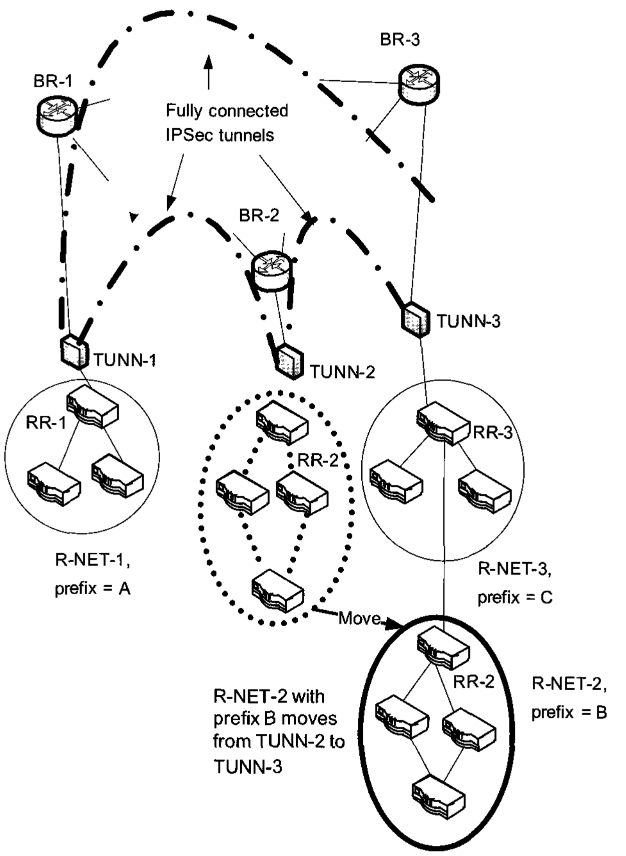


Fig. 3. Red network with prefix B moves from TUNN-2 to TUNN-3.

the other for prefix ADV PDU packets. Fig. 4 shows how a remote tunnel gateway that restarts or is down can be detected. In the active mode, the tunnel gateway sends a hello PDU and prefix ADV PDU packets every hello timer interval. According to Tran's work [4], a quiet state is defined as the period when no SPs are updated in four instances of the hello interval. The tunnel gateway reported in this paper, which was based on that study, also declares the peer tunnel gateway to be dead if a hello PDU from a certain peer tunnel gateway is not received within four instances of the hello timer interval. Then, the tunnel gateway removes all SPs associated with the peer tunnel gateway. The tunnel gateway also recognizes the event that a remote tunnel gateway has been restarted. A restart of a remote tunnel gateway can be detected via receipt of the timestamp of the hello PDUs from the remote tunnel gateway. The data of each hello PDU contains a timestamp that is taken at the instant of the initialization set-up process. This initial timestamp value can be used to detect the dead peer. When the timestamp of the hello PDUs from the same tunnel gateway is changed, that tunnel gateway declares that it has been restarted. Then, each local tunnel gateway removes all current SPs with the restarted tunnel gateway. Then, every tunnel gateway enters active mode and sends prefix ADV PDU packets during four instances of the hello timer interval. This is a necessary action to refresh all SPs associated with every tunnel gateway. That is, each tunnel gateway will have refreshed the SPD after any restarted tunnel gateway is found.

As shown in Fig. 5, the tunnel gateway alternates between active mode and quiet mode. The tunnel gateway enters quiet mode, in which only hello PDUs are sent, to conserve bandwidth in a steady state. A quiet mode state is defined as the period when no local SPs are updated or no remote tunnel gateway is restarted. This implies that the quiet mode switches to the active mode when any local R-NET prefixes are updated by the R-NET routers associated with the local tunnel gateway or any remote tunnel gateway restarts. As shown in Fig. 5, immediately after the local tunnel gateway returns to active mode, prefix ADV PDUs are sent four times in the time interval T (hello timer interval), until the condition for the quiet mode is met. Here, the prefix ADV PDU contains all prefix information associated with all local R-NETs including the fact that the R-NET changed. In this paper, T (seconds) is 30, because we assumed that any changes in the tunnel gateway's status should be detectable within 2 minutes, that is, $4T$.

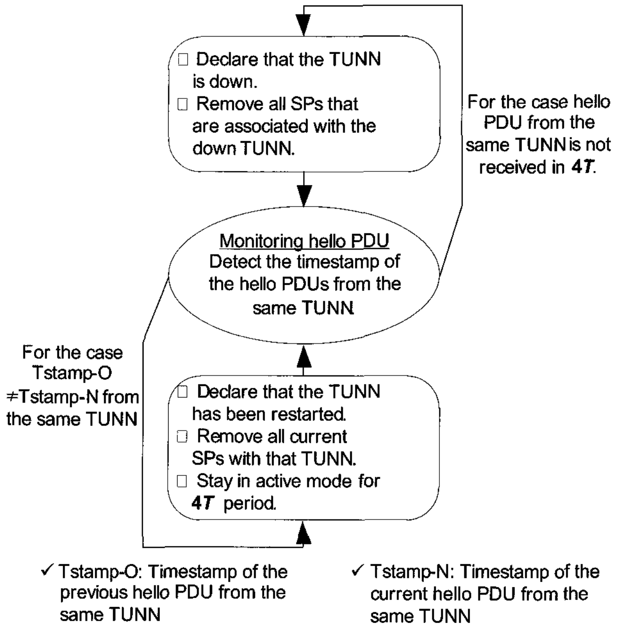
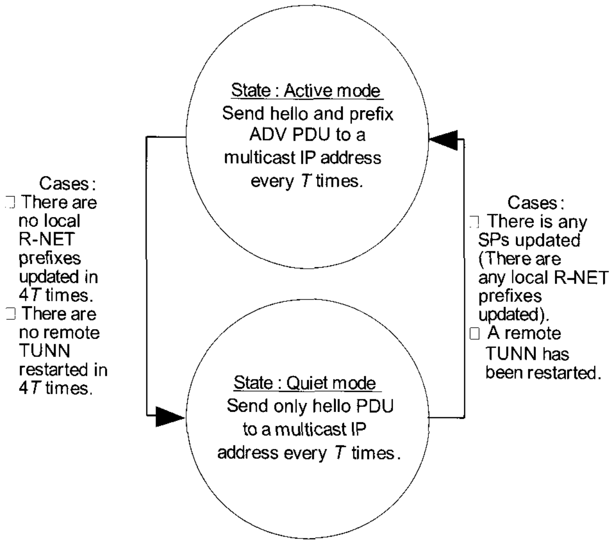

□ Declare that the TUNN is down.
□ Remove all SPs that are associated with the down TUNN.
For the case hello PDU from the same TUNN is not received in 4T.
Monitoring hello PDU
Detect the timestamp of the hello PDUs from the same TUNN.
For the case Tstamp-O ≠Tstamp-N from the same TUNN
□ Declare that the TUNN has been restarted.
□ Remove all current SPs with that TUNN.
□ Stay in active mode for 4T period.
✓ Tstamp-O: Timestamp of the previous hello PDU from the same TUNN
✓ Tstamp-N: Timestamp of the current hello PDU from the same TUNN

Fig. 4. Detecting the 'restart'/'down' status of a remote tunnel gateway.

IV. PERFORMANCE OF TUNNEL GATEWAY PROTOCOL

A. Analysis Model for the Protocol Data Traffic

The main role of the tunnel gateway is to dynamically discover the prefixes associated with R-NETs that change location. In highly dynamic R-NETs, it could be the case that the frequency of events that trigger a tunnel gateway to generate and distribute prefix advertisements might be quite high. This concern motivates the development of methods for reducing the overhead of transmitting and processing these advertisements. However, this paper used a reverse approach: Tunnel gateways distribute changes in attached networks as well as prefixes of networks that have not changed. This certainly increases the size of the prefix advertisements. However, we believe that our approach will yield additional benefits from the viewpoint of uniformity in the prefix database that associates with each tun-


State : Active mode
Send hello and prefix ADV PDU to a multicast IP address every T times.
State : Quiet mode
Send only hello PDU to a multicast IP address every T times.
Cases:
□ There are no local R-NET prefixes updated in 4T times.
□ There are no remote TUNN restarted in 4T times.
Cases:
□ There is any SPs updated (There are any local R-NET prefixes updated).
□ A remote TUNN has been restarted.

✓ T: Hello timer interval
✓ ADV PDU: Advertisement protocol data unit
✓ SPD: Security policy database
✓ TUNN: Tunnel gateway

Fig. 5. Active mode and quiet mode.

nel gateway in dynamically changing environments. The analysis model will help to ensure that the negative impact of the size increase of prefix advertisements is slight. Also, it will guarantee that our advertisement scheme will be simpler and more robust to dynamic Ad hoc environments. The system we consider is the queueing system M/D/1 as a simple-scale representation of our prefix advertisement process [13]. Our tunnel gateway system obviously belongs to a parallel system. Therefore, modeling it as an analytical queueing system is difficult. However, our proposed tunnel gateway multicasts the Hello and Advertisement PDUs. This means that the arrival patterns of PDU messages at every tunnel gateway are statistically identical. This explains why our M/D/1 queuing model is usable in our tunnel gateway scenarios. We assume the following parameters to explore how our prefix discovery process shows an SP adapting its latency to the changed condition.

- T_{move}: Average moving time of an R-NET from a cessation of connection to a connection renewal
- T_{deploy}: Average deployment time of a tunnel gateway from 'down' to restart
- N_{prefix}: Number of R-NET prefixes associated with a tunnel gateway
- N_{gate}: Number of tunnel gateways in the B-NET
- T_{hello}: Hello timer interval
- PDU_{size}: Size of prefix ADV PDU
- $\text{HeaderSize}_{\text{byte}}$: Total PDU header size in bytes
- $\text{Prefix}_{\text{byte}}$: Length of a prefix address space in bytes
- BW_{link}: Link bandwidth dedicated for protocol data
- $\bar{x}$: Service time of the prefix ADV PDU at the prefix advertisement system
- λ: Arrival rate of the prefix ADV PDU for which the interval time distribution is exponential
- D_{ADV}: Average delay time during which the prefix ADV PDU suffers from delay in the prefix advertisement system.

We focus on exploring two different types of performance result: Effects of R-NET movement and a tunnel gateway's state changes. Let us define the moving event, as either any instance of an R-NET prefix disappearing or any instance of a new R-NET prefix showing up in the routing table of the red interface in the tunnel gateway. Then, the average moving event interval in the entire network ($\text{Average}_{\text{movingtime}}$) can be computed as $T_{\text{move}}/N_{\text{prefix}}N_{\text{gate}}$. We fix the value of T_{hello} to 30 seconds. Recall that when any local R-NET prefix is updated, prefix ADV PDUs are sent four times every T_{hello}. Then, each tunnel gateway will receive the prefix ADV PDU at the rate of $1/\min(30, \frac{\text{Average}_{\text{movingtime}}}{4})$. We need to study the effects of the overhead of ESP in tunnel mode and the payload with varying size of prefix data on PDU transmission delay [14]. As shown in Fig. 6, the PDU size (PDU_{size}), which varies according to the value of N_{prefix}, can be represented by $\text{HeaderSize}_{\text{byte}} + N_{\text{prefix}}\text{Prefix}_{\text{byte}}$, that is, $(72 + 20N_{\text{prefix}})$ if we assume that $\text{Prefix}_{\text{byte}} = 20$ bytes. Here, the transmission delay of the prefix ADV PDU corresponds to the deterministic value of the PDU service time at the tunnel gateway, which depends on BW_{link}. In this case, the tunnel gateway will handle the prefix ADV PDU with 'Service time' $\bar{x} = 8\text{PDU}_{\text{size}}/BW_{\text{link}}$. Then, we can denote the analysis model as an M/D/1 queue with parameters $\lambda = 1/\min(30, \frac{\text{Average}_{\text{movingtime}}}{4})$ and $\bar{x} = 8(72 + 20N_{\text{prefix}})/BW_{\text{link}}$.

Let the utilization factor ρ be $\lambda\bar{x}$. Then, we can compute the average time in queue W as $\lambda\bar{x}/2(1-\rho)$. Here, we can derive the D_{ADV} (the average delay time during which the prefix ADV PDU suffers from delay in the queueing system). It can be computed as $\bar{x} + W$, when we consider only R-NET movement situations.

Next, we investigate the effects of the tunnel gateway's state changes. When the timestamp of the hello PDUs from the same tunnel gateway is changed, that tunnel gateway is declared to have been restarted. Then, each local tunnel gateway removes all current SPs with the restarted tunnel gateway. Then, each tunnel gateway enters active mode and sends prefix ADV PDU packets during four instances of the hello packet interval, immediately after it recognizes the 'restart' event of the remote tunnel gateway. This means that every tunnel gateways sends prefix ADV PDU packets when a certain tunnel gateway restarts. Let us define the deployment as any tunnel gateway restarts. Recall that the average deploy event interval is T_{deploy}. Then, each tunnel gateway will receive the prefix ADV PDU at the rate of $1/\min(30, \frac{T_{\text{deploy}}}{4N_{\text{gate}}})$. Recall that the PDU size (PDU_{size}) is also $(72 + 20N_{\text{prefix}})$. The tunnel gateway will serve the prefix ADV PDU with 'Service time' $\bar{x} = 8\text{PDU}_{\text{size}}/BW_{\text{link}}$. Following the same procedure as the first case, with parameters $\lambda = 1/\min(30, \frac{T_{\text{deploy}}}{4N_{\text{gate}}})$ and $\bar{x} = 8(72 + 20N_{\text{prefix}})/BW_{\text{link}}$, we can compute the prefix ADV PDU's average delay time in the queueing system (that is, $D_{\text{ADV}} = \bar{x} + W$) which is expected to be severe especially when a certain tunnel gateway restarts.

B. Analysis Results

It is desirable to optimize the protocol bandwidth for each pair of tunnel gateways. Our analysis model explored the performance (e.g., SP adaptation latency) of dynamically changing

24 bytes	24 bytes	24 bytes	N_{prefix} 20 (Prefix$_{\text{byte}}$) bytes
Outer IP header	ESP header	Inner IP header	Payload

□ N_{prefix}: Number of R-NET prefixes associated with a tunnel gateway

□ Prefix$_{\text{byte}}$: Length of a prefix address space in byte s

Fig. 6. Size of prefix ADV PDU.

networks. The SP adaption latency in this paper corresponds to the D_{ADV} value (that is, the average delay time during which the prefix ADV PDU suffers from delay in our M/D/1 queueing system) computed for a given mobile condition. The two different types of mobile situations are a movement of an R-NET and a restart of a tunnel gateway. The various R-NET movement conditions differ depending on the parameters: T_{move}, N_{prefix}, N_{gate}, T_{hello}, PDU_{size}, $\text{HeaderSize}_{\text{byte}}$, $\text{Prefix}_{\text{byte}}$ and BW_{link}. The various restart conditions differ, depending on the parameters: T_{deploy}, N_{prefix}, N_{gate}, T_{hello}, PDU_{size}, $\text{HeaderSize}_{\text{byte}}$, $\text{Prefix}_{\text{byte}}$ and BW_{link}. Recall that we fixed T_{hello} to be 30 seconds, $\text{HeaderSize}_{\text{byte}}$ to be 72 bytes and $\text{Prefix}_{\text{byte}}$ to be 20 bytes in order to reduce the computational complexity.

Strictly speaking, it is very difficult to define the boundary that can be used to identify if whether the D_{ADV} value (SP adaptation latency in this paper) meets the requirements needed to resolve the mobility issues. In this paper, the SP adaption latency does not include the delay components caused by actual physical transmission networks, such as queueing delays in the intermediate routers and transmission/propagation delays in a series of physical links. However, in our paper, we assume that if the D_{ADV} value, which is caused by the tunnel between peer tunnel gateways, exceeds 250 milliseconds, the SA and SPD structure cannot adapt to dynamically mobile networks.

Figs. 7–9 show the SP adaptation latency (D_{ADV}) in dynamically changing network configurations as T_{move} varies. Fig. 7 shows that for the case of $T_{\text{move}} = 200$ seconds, $N_{\text{gate}} = 16$ and $N_{\text{prefix}} = 10$, a BW_{link} of 0.01 Mbps causes the SP adaptation latency to increase to nearly 700 milliseconds. Thus, it can be seen that a bandwidth of 0.01 Mbps cannot meet the SP adaptation latency for protocol data traffic. Fig. 8 shows that for the conditions of $N_{\text{gate}} = 8$ and $N_{\text{prefix}} = 10$, the SP adaptation latency also cannot comply with the requirement for a latency below 250 milliseconds in dynamically network configurations as T_{move} approaches 200 seconds. Fig. 9 shows that if we assume the worst case scenario (e.g., $T_{\text{move}} = 200$ sec) of dynamic R-NETs, a BW_{link} of 0.02 Mbps can comply with the requirement for a D_{ADV} value below 250 milliseconds for the conditions of $N_{\text{gate}} = 16$ and $N_{\text{prefix}} = 10$. Therefore, it can be seen that it is appropriate to maintain the BW_{link} bandwidth within the 0.02 Mbps that is reserved for protocol data traffic.

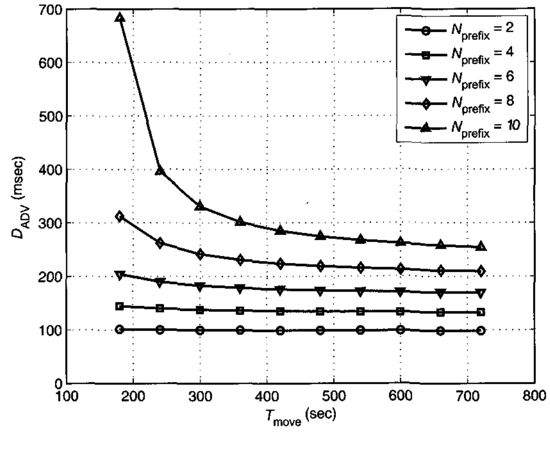


Fig. 7. D_{ADV} versus T_{move} for $N_{\text{gate}} = 16$ and $BW_{\text{link}} = 0.01$ Mbps.

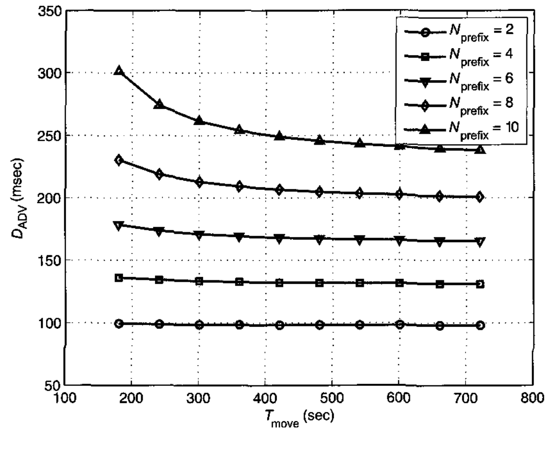


Fig. 8. D_{ADV} versus T_{move} for $N_{\text{gate}} = 8$ and $BW_{\text{link}} = 0.01$ Mbps.

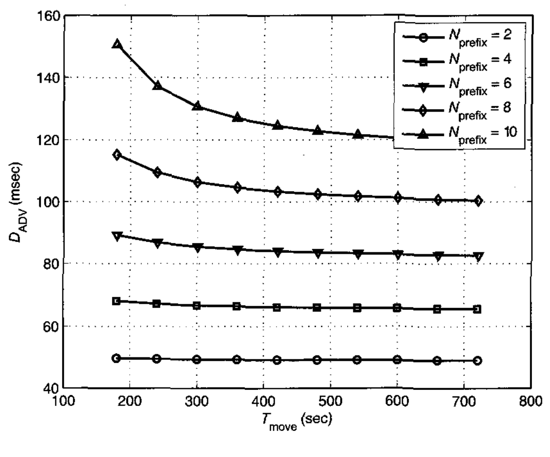


Fig. 9. D_{ADV} versus T_{move} for $N_{\text{gate}} = 16$ and $BW_{\text{link}} = 0.02$ Mbps.

Figs. 10–12 show SP adaptation latency (D_{ADV}) in the dynamically changing restart status of the tunnel gateways, with respect to T_{deploy}. It is reasonable to consider the average deployment time of a tunnel gateway from when it is down to when it is restarted (T_{deploy}) to be on the order of a couple of hours. For the worst case scenario of highly mobile tunnel gateways, we used stress testing with T_{deploy} on the order of of a time as low as several minutes. Fig. 10 shows that if we assume the worst case scenario (e.g., $T_{\text{deploy}} = 200$ seconds), a BW_{link} of 0.01 Mbps hardly complies with the requirement for a D_{ADV} value of below 250 milliseconds, where the con-

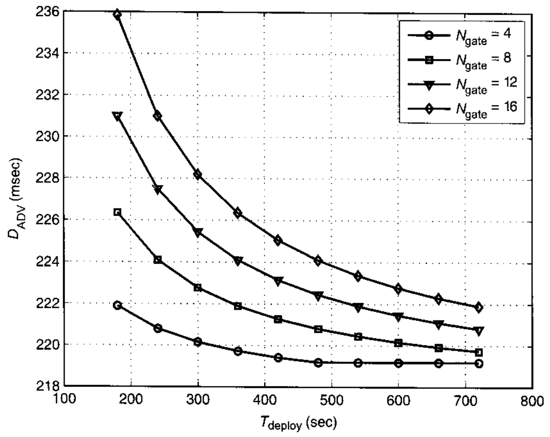


Fig. 10. D_{ADV} versus T_{deploy} for $N_{\text{prefix}} = 10$ and $BW_{\text{link}} = 0.01$ Mbps.

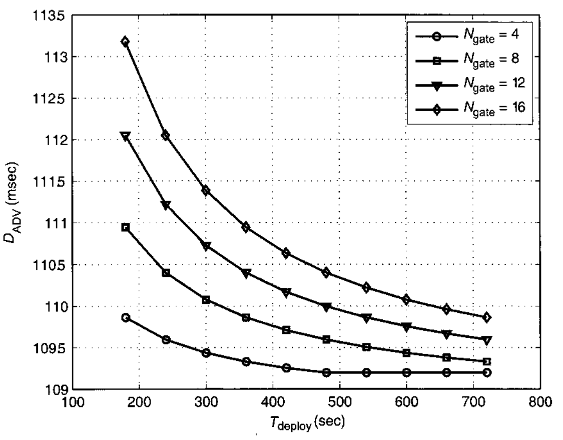


Fig. 12. D_{ADV} versus T_{deploy} for $N_{\text{prefix}} = 10$ and $BW_{\text{link}} = 0.02$ Mbps.

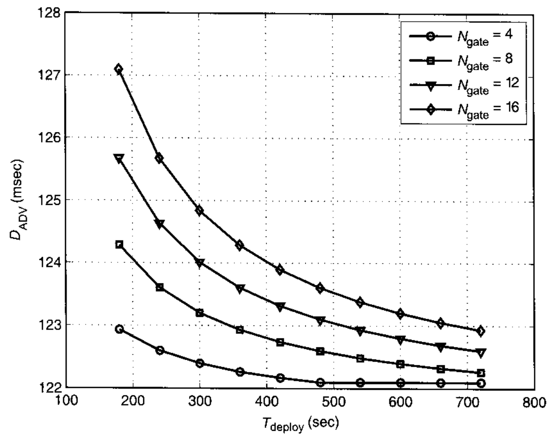


Fig. 11. D_{ADV} versus T_{deploy} for $N_{\text{prefix}} = 4$ and $BW_{\text{link}} = 0.01$ Mbps.

ditions are $N_{\text{gate}} = 16$ and $N_{\text{prefix}} = 10$. Fig. 11 shows that compliance with a D_{ADV} value of less than 250 milliseconds is possible if N_{prefix} decreases to 4, where the condition of a BW_{link} of 0.01 Mbps can meet the requirement for an SP adaptation latency of below 250 msec. The number of R-NETs under the control of a tunnel gateway will range from 4 to 10 based on the typical military operations. So, it is reasonable to consider that the worst case will occur when the number of R-NET prefixes associated with a tunnel gateway is about 10. Fig. 12 shows that when $N_{\text{gate}} = 16$ and $N_{\text{prefix}} = 10$, it is necessary to reserve a bandwidth of about 0.02 Mbps for protocol data traffic, in order to control mobility in association with R-NET movements and tunnel gateway restarts. We argue that the tunnel link bandwidth plays an important role in severely changing network environments, and about 0.02 Mbps needs to be reserved against a severe condition, such as when $T_{\text{move}} = 200$ seconds, $T_{\text{deploy}} = 200$ seconds, $N_{\text{prefix}} = 10$ and $N_{\text{gate}} = 16$.

V. Conclusion

The architecture of mobile and IP-based networks consists of a B-NET that serves as the IP backbone network, and R-NETs that can change locations. Although each R-NET domain corresponds to an autonomous system of mobile nodes, inter-domain communication between the local and remote R-NETs requires the use of the two tunnel gateways associated with the local and remote R-NETs, respectively. After completion of the prefix ADV from a certain remote tunnel gateway that has detected changes in its local R-NET prefix, every tunnel gateway will update the SPs of the R-NETs associated with the remote tunnel gateway.

In this paper, we recommended a simple and robust advertisement process for SP adaptation to mobile environments. Our advertisement process enables every tunnel gateway to renew the changed R-NET prefix status within 4 × 30 seconds, immediately after the R-NET prefix changes its location or a tunnel gateway begins to go 'down'/'restart.' This is possible, because the tunnel gateway sends the prefix ADV PDUs over a multicast address in four instances of the hello timer interval of 30 seconds when its local R-NET prefix is changed. Also, every tunnel gateway sends the prefix ADV PDUs over a multicast address as explained, when any restarted tunnel gateway is found. Using our analysis model, we explored the SP adaptation latency performance for dynamically changing networks. We found that for the worst case scenario, compliance with an SP adaptation latency of less than 250 msec is possible when the condition of $BW_{\text{link}} = 0.02$ Mbps. Therefore, it can be seen that it is appropriate to maintain the BW_{link} bandwidth of 0.02 Mbps, which is responsible for multicast of protocol data traffic. For the case of the SP adaptation latency in the dynamically changing restart status of tunnel gateways, we found that the other worst case scenario requires a BW_{link} of 0.02 Mbps to comply with the requirement for an SP adaptation latency of below 250 msec. As a result, the tunnel link bandwidth dedicated to protocol data traffic needs to be about 0.02 Mbps against a severe condition.

We argue that our advertisement approach in which tunnel gateways distribute all network prefixes including the changes in attached networks, avoids problems from the viewpoint of SP adaptation latency performance for dynamically changing networks. Our prefix ADV mechanism can yield additional benefits from the viewpoint of uniformity in prefix databases, in tunnel gateways under dynamically changing environments. This proves that our advertisement scheme is simpler and more robust to dynamic ad hoc environments. Our future works will be related to an extension of this work, which will make quantitative comparisons between our approach, focusing on robustness, and the other approach, focusing on SP adaptation latency, in which tunnel gateways distribute only the changes in attached networks rather than the addresses of networks that have not changed.

REFERENCES

[1] N. A. Surobhi, Y. Ma, and A. Jamalipour, "A semantic traffic management scheme for public safety applications in mobile ad hoc networks," in *Proc. ISWPC*, Feb. 2011, pp. 1–6.

[2] A. Ghosh, R. Talpade, M. Elaoud, and M. Bereschinsky, "Securing ad-hoc networks using IPsec," in *Proc. IEEE MILCOM*, vol. 5, Oct. 2005, pp. 2948–2953.

[3] K. Wen, W. Guo, and B. Xiao, "A mobility management scheme for hierarchical mobile ad hoc networks," *ITS Telecommun. Proc.*, pp. 671–674, June 2006.

[4] T. H. Tran, "Proactive multicast-based IPSEC discovery protocol and multicast extension," in *Proc. IEEE MILCOM*, Oct. 2006, pp. 1–7.

[5] F. Ingelrest, N. Mitton, and D. Simplot-Ryl, "A turnover based adaptive HELLO protocol for mobile ad hoc and sensor networks," in *Proc. MASCOTS*, Oct. 2007, pp. 9–14.

[6] C. E. Fossa and T. G. Macdonald, "Internetworking tactical MANETs," in *Proc. IEEE MILCOM*, Oct. 31–Nov. 3, 2010, pp. 611–616.

[7] K. Ishimura, T. Tamura, S. Mizuno, H. Sato, and T. Motono, "Dynamic IP-VPN architecture with secure IPsec tunnels," in *Proc. APSITT*, June 2010, pp. 1–5.

[8] B.-J. Kim and S. Srinivasan, "Simple mobility support for IPsec tunnel mode," in *Proc. IEEE VTC-fall*, vol. 3, Oct. 2003, pp. 1999–2003.

[9] A. Gunnar and M. Johansson, "Robust routing under BGP reroutes," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 2719–2723.

[10] S. L. Murphy, "Secure inter-domain routing standards evolution and role in the future GIG," in *Proc. IEEE MILCOM*, Oct. 2007, pp. 1–7.

[11] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A survey of multicast routing protocols for mobile ad-hoc networks," *IEEE Commun. Surveys Tut.*, vol. 11, no. 1, pp. 78–91, First Quarter 2009.

[12] S. Bin, K. Haiyan, and H. Zhonggong, "Adaptive mechanisms to enhance internet connectivity for mobile ad hoc networks," in *Proc. WiCOM*, Sept. 2006, pp. 1–4.

[13] I. Suliman and J. Lehtomaki, "Queueing analysis of opportunistic access in cognitive radios," in *Proc. CogART*, May 2009, pp. 153–157.

[14] N. Kazemi, A. L. Wijesinha, and R. Karne, "Evaluation of IPsec overhead for VoIP using a bare PC," in *Proc. ICCET*, vol. 2, Apr. 2010, pp. V2-586–V2-589.

Younchan Jung received the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1991 and 1994, respectively. From 1980 to 1989, he worked as a Senior Research Engineer in the Agency for Defense Development (ADD), Korea. He is currently a Professor at the School of Information and Communications Engineering in the Catholic University of Korea. In 2002, he was Visiting Scientist and Lecturer, Computer Science, Concordia University, Canada. Also he was Visiting Scientist at the School of Engineering, Blekinge Institute of Technology, Sweden in 2008. His research interests include wireless VoIP QoS, mobile network security, mobile ad hoc network design and robust DNS design.

Marnel Peradilla was born in Batangas, Philippines, on October 14, 1988. He received his Electronics Engineering degree from the University of the East-Caloocan, Philippines in 2010. He is currently studying Master of Science degree in the Catholic University of Korea, School of Information, Communications and Electronics Engineering. His research interests include VPN security, mobility management in the mobile ad hoc networks and robust DNS design.