IEICE TRANS. FUNDAMENTALS, VOL.E89–A, NO.7 JULY 2006

PAPER Special Section on Multi-dimensional Mobile Information Networks

Proactive Defense Mechanism against IP Spoofing Traffic on a NEMO Environment

Mihui KIM[†a)], *Student Member and* **Kijoon CHAE**[†], *Nonmember*

SUMMARY The boundary of a distributed denial of service (DDoS) attack, one of the most threatening attacks in a wired network, now extends to wireless mobile networks, following the appearance of a DDoS attack tool targeted at mobile phones. However, the existing defense mechanisms against such attacks in a wired network are not effective in a wireless mobile network, because of differences in their characteristics such as the mobile possibility of attack agents. In this paper, we propose a proactive defense mechanism against IP spoofing traffic for mobile networks. IP spoofing is one of the features of a DDoS attack against which it is most difficult to defend. Among the various mobile networks, we focus on the Network Mobility standard that is being established by the NEMO Working Group in the IETF. Our defense consists of following five processes: speedy detection, filtering of attack packets, identification of attack agents, isolation of attack agents, and notification to neighboring routers. We simulated and analyzed the effects on normal traffic of moving attack agents, and the results of applying our defense to a mobile network. Our simulation results show that our mechanism provides a robust defense.

key words: *Network Mobility (NEMO), mobile network, IP spoofing traffic, defense mechanism, neighbor graph*

1. Introduction

Currently, DDoS attacks are considered one of the most threatening of attacks against wired networks, because of consuming critical resources at the target within a short time, and causing network congestion en route from the source to the target. The damage from DDoS attacks is no longer limited to wired networks. Though the functionality of most mobile devices is extremely limited and largely non-programmable, the first virus targeted at mobile phones has already appeared. In addition, the SMS (Short Message Service) flooder has emerged as the first DDoS attack tool against mobile phones. The potential hazard is not only the blocking of communications but also the high financial cost when pricing is usage-based [1]. The damage is expected to become increasingly serious as mobile devices become high-performance machines.

Long an issue of interest, network mobility technology is now being realized with the foundation of the NEMO (Network Mobility) Working Group (WG) in the IETF (Internet Engineering Task Force). This WG is concerned with managing the mobility of an entire network that changes, as a unit, its point of attachment to the Internet. The WG defines *NEMO* as "NEtwork MObility" or "a NEtwork that is MObile." Most importantly, a NEMO can be nested in another NEMO and can be multihomed. This structure provides one or multi-hop wireless links, as well as a tree-like hierarchy.

The characteristics of a NEMO, the multi-hop wireless links and mobility, pose many security challenges, like the case of a mobile ad hoc network (MANET) [2]. Moreover, existing centralized approaches to security on a wired network are inefficient on a NEMO, because of the possible problems posed by high mobility and scalability. Cell phones and small routers can become mobile routers (MRs) that provide the features of a NEMO, and most mobile devices have limited processing power. Therefore, the security mechanisms on a NEMO should be light, for the low power consumption, but also robust. The damage by compromise of a mobile IPv6 node may become bigger due to its mobility, but especially compromise of a MR that performs basic NEMO operations can also compromise all the nodes under its charge in the NEMO, causing far more widespread damage than in a MANET.

Of the many possible features of DDoS attacks, source IP address spoofing is among the most common. IP spoofing creates particular difficulties for network managers, because it increases the number of flows by varying the source address and concealing the identity of the attack agent. DDoS attacks using IP spoofing also pose a threat in a NEMO even with IPSec (IP Security Protocol) [3], where detection of and defense against such attacks is far more difficult due to the fact that the attacker can move. Also, even though there is the trust relationship between all nodes in a NEMO through authentication, or CGA (Cryptographically Generated Addresses) approach [4] generating a random interface identifier is used, they are not enough for preventing IP spoofing attack, because the execution code of attack agent can be infiltrated into nodes through their security holes like worm virus.

In this paper, we will simulate and analyze the effects of an IP spoofing attack on a mobile network. Through the simulated results and the characteristics analysis of a DDoS attack on a NEMO, we will adapt and extend previously proposed detection and identification defense mechanisms against spoofing packets on a wired network [5], to the NEMO environment. A previously proposed mechanism was implemented, tested using strong DDoS attack tools on a real network, and confirmed to be an effective design. Initially, we set the following design goals for defense against

Manuscript received October 26, 2005.
Manuscript revised January 26, 2006.
Final manuscript received March 9, 2006.
[†]The authors are with the Dept. of Computer Science and Engineering, Ewha Womans University, Seoul, Korea.
a) E-mail: mihui@ewhain.net
DOI: 10.1093/ietfec/e89–a.7.1959

IP spoofing attacks on a wireless network, particularly on a NEMO.

- **Speedy detection and filtering** of the source-side network as soon as possible
- **Identification and isolation** of attack agents for immediate follow-up measures
- **Notification** to neighboring MRs to proactively isolate the attacking traffic

This paper is divided into five sections. In Sect. 2, we explain the basics of attack and NEMO. Section 3 introduces the proposed defense mechanism. In Sect. 4, we evaluate our mechanism and explain our analysis of the simulation results. Finally, we present a brief conclusion.

2. Background

2.1 DDoS & Spoofing Traffic

DoS/DDoS (Symbol '/' means 'AND' in this paper) attacks compromise the availability of the network resource as well as the system resource itself. Because DDoS attacks usually use normal protocol packets like legitimate users, the defense against these attacks is not enough with the intrusion prevention techniques only, such as user authentication and encryption.

The conventional mechanisms to mitigate the damage by DDoS attacks in the wired network are classified into three categories: *detection* as a first proactive defense [6], *filtering* such as ingress-based scheme [7], and *traceback* (*identification*) [8] as a reactive scheme to identify the routers, which are the closest to the true origins of attack. These various mechanisms contributed to a defense against DDoS attack, but some mechanisms have the drawback such as the requirement of wide deployment, scalability problems, limited features to only specific DDoS attacks, or a mass of wasteful overheads. Especially, because these conventional mechanisms have no consideration for the moving possibility of attack agents and wireless environment of nodes, they are insufficient for a defense against such attacks on a NEMO environment.

One reason that DDoS attacks are very threatening is the spoofing feature. It makes it difficult to find the origin of attack agents, and also increases considerably the number of flows in a few minutes through varying source IP address and source/destination port. Representative DDoS attack tools in the Internet provide even the automated spoofing feature as well as the control feature for spoofing level, to pass the attack traffic at the ingress/egress filtering routers. To date, these DDoS attack tools on the mobile IP network/NEMO did not appear, but the emergence of more intellectual and automated spoofing attack abusing mobility might be expected, as the Mobile IP/NEMO would broadly be deployed.

Although it is mandatory for MRs to perform the ingress filtering on packets received from the Mobile Network in the basic NEMO protocol [9], it may be insufficient for defending against the whole spoofing packets. Even if the router performs ACL (Access Control List) or uRPF (unicast Reverse Packet Filtering) that checks whether the source address's reverse path matches the input port, a lot of spoofing packets can actually pass through it because some DDoS tools provide "in-prefix" spoofing that uses a topologically correct prefix and non-existent interface ID. As an experimental result, we can get Fig. 1 which is the flow count gathered at the access router connecting our university network to the Internet, covering about 5000 hosts and performing the ingress filtering. We performed the in-prefix spoofing attack with TFN2k DDoS tool, one of the most threatening attack tools, as an attack scenario the attack agents were located at our internal network and a victim is located at the external network. From results, we could know the spoofed packet considerably increased the flow number, and well-spoofed attack traffic can pass the ingress filtering router besides.

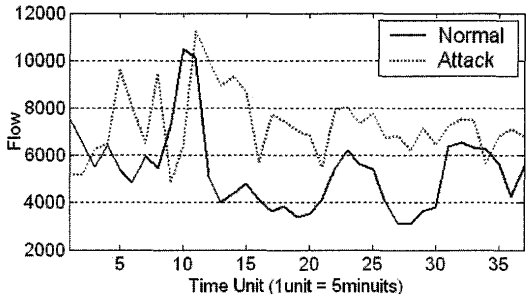


Fig. 1 Flow count in the normal/attack case.

Secondly, to examine how many flows are generated through an attack tool, we performed the TFN2k DDoS tool at the Pentium IV (CPU 1.9 GHz, 10/100 MB Ethernet). As the result, the tool generated the 6212 flows per a second in the TCP flood attack case, and the 13454 flows per a second in the UDP flood attack case. This flow count can overflow the equipment such as L4 switch within a very short time, moreover make the network down.

2.2 Mobile Network

A mobile network includes one or more MRs that connect to the global Internet in order to provide session continuity and access for all nodes in the NEMO, even as the network moves. The NEMO WG is extending Mobile IPv6 (MIPv6) [10] for network mobility, providing backward compatibility with MIPv6. There are three types of nodes in a NEMO: local fixed nodes (LFNs), local mobile nodes (LMNs), and visiting mobile nodes (VMNs). A node of any of these types may be either a host or a router. A LFN and a LMN belong to the mobile network. The LFN does not move topologically with respect to the MR, but the LMN does. A VMN can move topologically with the MR, and its home link does not belong to the mobile network. MRs access the Internet from access routers (ARs) on visited links. Figure 7 depicts the general composition of a NEMO. Our defense assumes that all types of nodes could be abused as attack agents.

To date, several drafts regarding NEMO security have been submitted; most relate to support of the secure basic NEMO protocol, or to the protection of control messages such as binding update (BU) messages. The basic requirements of AAA service for supporting network mobility have also been introduced. However, even if control messages are protected by encryption or authentication, the compromise of a MR/node can spread a spoofing attack to the whole NEMO via abnormal data traffic. Also, as the state of art research topics for NEMO, routing optimization and load sharing/session preservation mechanisms have been proposed, but the study on proper security mechanisms on NEMO is necessary.

2.3 Neighbor Graph

Mishra et al. [11], [12] introduced a novel data structure, the *Neighbor Graph*, which captures the mobility topology of a wireless network. Using this data structure, they showed the reduction in the authentication time of an IEEE 802.11 handoff by proactively distributing necessary key material one hop ahead of the mobile user [11], and the fast handoff by pre-positioning the station's context ensuring that the station's context always remains one hop ahead [12]. This proactive neighbor cashing scheme was adopted as an IEEE standard, Inter-Access Point Protocol (IAPP) specification [13], which is a standard protocol for the communications between APs.

Mishra et al. defined an *AP neighbor graph* $G=(V, E)$ where $V=\{ap_1, ap_2,..., ap_n\}$ is the set of all AP (Access Point)s, and there is an edge $e=(ap_i, ap_j)$ between ap_i and ap_j if there is some path of motion between them. They defined the *association pattern* $\Gamma(c)$ for client c as $\{(ap_1, t_1), (ap_2, t_2),..., (ap_n, t_n)\}$, where ap_i is the AP to which the station reassociates (new AP) at time t_i, and $\{(ap_i, t_i), (ap_{i+1}, t_{i+1})\}$ is such that the handoff occurs from ap_i to ap_{i+1} at time t_{i+1}. This neighbor graph can be maintained either in a distributed fashion by the APs or in a centralized manner, and autonomously learned and maintained through an 802.11 *reassociation request* frame containing the address of the old AP, or a *Move-Notify* message of IAPP (sent from an AP to the old AP during a reassociation).

3. Proactive Defense Mechanism

This section presents our overall defense architecture. Our defense against spoofing traffic on a mobile network consists of five parts: speedy detection, filtering of attack packets, identification of attack agents or NEMOs including attack agents, isolation of attack agents' traffic in the lower layer, and notification to neighboring routers. The last two steps are especially important on a mobile network in order to decrease overall damage, owing to the wireless environment and the mobility of nodes, respectively. Although there can be spoofing in several layers, we assume that the spoofing traffic is spoofing the source IP address, as in the case of DDoS attacks in wired networks. This defense could be adapted to handle spoofing of other layers in the same manner. Also, we assume that the defense mechanism is applied to the source network. Although it would be possible to use the defense mechanism in the target network, using the defense at each source network is more efficient and can decrease the damage before the target is shut down. We will explain each part of our defense against spoofing traffic in the following sub-sections.

3.1 Detection & Filtering

As it is very difficult to prevent spoofing attacks, the first priority is the rapid detection. In a NEMO environment, we cannot assume that there is at least one detection agent per router, because a NEMO can be very small, as with a PAN (Personal Area Network). Therefore, we consider two cases: in one case, all MRs perform detection and filtering; in the other, one agent per AR performs these roles.

In the first case, MRs can check the outbound packets choosing one of two different detection mechanisms, according to the number of served nodes. For the speedy detection, these mechanisms are situated at the source-side network from the attacker (spoofing-agent) point of view because the damage of most spoofing attacks starts from the outbound attack packets that are forwarded into the Internet. One method uses the configured network addresses and the other uses the rate of change of IP addresses for a single MAC address. Each attack detection condition is shown in Figs. 2 and 3.

The first method makes use of the characteristic that most spoofing attack tools rotate a specific range of IP addresses as the source address. Thus, these spoofing addresses include one or more non-configured address, such as a router address, denied address (e.g. multicast address), or a different subnet address. This method is similar to Ingress Filtering [7] and uRPF (unicast Reverse Packet Filtering)

$(sIP \notin CA_i)$ OR $(sIP \in DA_i)$ OR $(sIP \in RA_i)$

- *sIP*: Source IP address of a packet
- $CA_i=\{Ca_1, Ca_2,...,Ca_k\}$: Configurable IP addresses as source address in the shared media
- $DA_i=\{Da_1, Da_2,...,Da_l\}$: Denied IP addresses as source address in the shared media
- $RA_i=\{Ra_1, Ra_2,...,Ra_m\}$: Directly connected router IP addresses

Fig. 2 Detection condition using configurable address information.

$$| t[sIP_1, sMAC_1]Pi - t[sIP_2, sMAC_1]Pj | \leq Tcng\_rate$$

- *sMAC*: Source MAC address of a packet
- $t[sIP, sMAC]Pj$: Time of discovering the packet Pj having *sIP* and *sMAC*.
- $Tcng\_rate$: Upper threshold for change interval for spoofed packets. This is the change interval (seconds) of source IP addresses for a source MAC address.

Fig. 3 Detection condition using change rate of IP address for a MAC address.

that are representative methods for defending against spoofing attack in the recent Internet. However, mostly Ingress Filtering set by ACL (Access Control List) at the router checks the only network prefix, thus actually it may pass over in-prefix spoofing packets as results of Fig. 1. Our method is not novel, but we emphasize the necessity of monitoring configured addresses like Fig. 2 in detail. This method needs to discover configured address, router address, and denied address. In a NEMO environment, when a NEMO attempts to become nested, its network addresses can be conveyed to its parent MR or AR as a step in the handoff. The first method may not be as fast as the second, but the required storage is smaller and middle MRs in the nested structure can easily perform the detection feature using the same mechanism.

However, it is not that the first method can detect all spoofing packets at source side or be used at all cases. If privacy extension [14] or CGA approach [4] which generate a random interface identifier to protect the privacy or support the security respectively, are used, the first method could not be used due to the frequent change of source address. Also, this method has the weakness in the case that both spoofing source address and victim address are configured address, that is, both attacker and victim are included in the source side network. For the stronger detection, the second method uses the characteristic that the source MAC address does not change during an IP spoofing attack, although spoofing attack tools rapidly change the source address. Test results indicate that the rate of change is about 0.074 milliseconds for the TFN2k DDoS attack tool. Therefore, the admissible T_{cng_rate} can be set to an interval such as 2 s (seconds), considering that a person changes the IP address of a machine or [4], [14] are used, taking at least few minutes. This method provides the faster detection, but it is limited to monitoring general server hosts because routers forward packets with various source IP addresses for the same MAC address. If a middle MR in a nested NEMO uses this method as the detection feature, it should determine whether the served node is a MR or a general host, in the handoff process. Also, this method should specially manage the hosts using the IP aliasing at an interface. But, because each IPv6 router can know the multiple addresses of the hosts using the IP aliasing in the auto-configuration, the MR can treat in disregard of the change to aliased IP addresses for a MAC address. For these two methods, proper and fast detection has been established using well-known, powerful spoofing tools on a real network [5].

In the case explained above, all MRs perform detection and filtering. This assumption is ideal, but is not always the reality. The possible detection/filtering/identification processes of the second case are explained in the next section.

3.2 Identification & Isolation of Attack Nodes

Although IP spoofing makes it difficult to defend against DDoS attacks, it is important to quickly and accurately identify attack agents in order to minimize possible damage. To make the identification, that is, to find the real IP address of an attack agent, the defending agent should have an *IP2MAC* table that includes the mapping of an IP address to a MAC address.

As explained above, when all MRs perform the defense feature, a MR can quickly determine the attack agents by using an *IP2MAC* table. However, as second case if only one detection agent per AR performs the detection feature, the detection agent needs to determine the MR that includes the attack agent by means of its *IP2MAC* table. In this case, it notifies the MR of the detected attack in order to transfer the identification job, and it filters the traffic from the MR that includes the attack agent until it finds the exact attack agents. This detection and notification work is iterated over the downward path of nested NEMOs until it finds the attack agents. When the MR finds the attack agents, it notifies the parent-MRs regarding the identification, with a request to forward the normal traffic of the NEMO. If this notification is conveyed to the first detection agent, only the attack traffic is filtered.

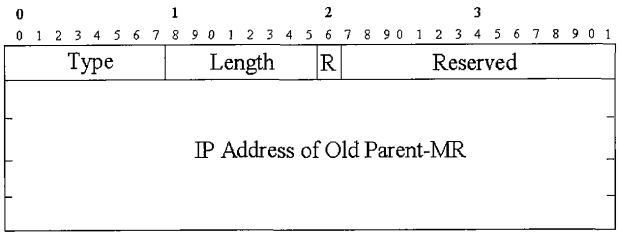
The result of isolating attack agents' traffic in the lower layer is that the MAC layer denies grant of the channel to the attack agents; for example, in the case of 802.11, it refuses to send a clear to send (CTS) message in response to a request to send (RTS) message. This is important in a wireless environment that uses a CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) like 802.11 because granting a channel to attack agents considerably affects the transmission rate of other nodes, even while filtering of attack traffic occurs at the IP layer.

3.3 Notification to Neighbor

This part of defense against attack is specific to a NEMO, and is necessary because NEMOs that include attack agents, or the attack agents themselves, can move to other MRs/AR. We assume that mobile router should authenticate a served mobile node before it is handoffed, thus all nodes in a NEMO can trust each other after successful authentication from parent node. Therefore, there is a trust relationship between neighbor MRs. To decrease the time for detection and identification at the handed-off parent MRs/AR, the first agent that detects and identifies the attack agent provides its neighboring MRs/AR with information about the attack agents, such as the real IP or MAC address. If all MRs are performing the defense feature and the attack agents are mobile, the MR handling the attack agent provides the subsequently affected MRs/AR with the attack information about the attack agent. If there is only one detection agent per AR, it provides the neighboring detection agents with the attack information.

How the neighbor graph creates

There are several ways to make the neighbor graph. First, in the case of using the IEEE 802.11 wireless LAN and IEEE 802.1f Inter-AP Protocol as the layer 2 protocols, *reassociation request* message and *move notify* mes-



Type TBA(To Be Assigned)

Length 8 bit unsigned integer indicating the length in octets of the option excluding the type and length fields. Set to 18.

R The Mobile Router Flag is set to indicate to Parent MR/AR that this message is from a Mobile Router. If the flag is set to 0, the Parent-MR/AR assumes that the Mobile Router is just behaving as a Mobile Node.

Reserved This field is unused for now. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.

IP Address of Old Parent-MR
A 16 byte field contains the Mobile Router IP Address before handoff.

Fig. 4 Old parent-MR's IP address option format.

sage can be used for creating the neighbor graph like [11], [12]. However, layer 2 protocol for constructing the NEMO can become other protocols, for example the IEEE 802.15.3 WPAN (Wireless PAN), IEEE 802.15.4 Low-bit WPAN for internetworking with sensor network, and so on. Thus, the support of higher layer is necessary, but this support should not burden on the wireless communication and wireless device. Therefore, we design this job extending the IPv6 messages.

IPv6 originally provides a way to keep track of which neighbors are reachable by *NS* (*Neighbor solicitation*), *NA* (*Neighbor Advertisement*), *RS* (*Router Solicitation*), and *RA* (*Router Advertisement*) messages [15]. The information of DDoS attack agents can be obviously notified to all of reachable neighbors in the transmission range. However, if we would know the movement pattern of attack agents or possible neighbors, we could not only minimize the required bandwidth for notification, but also minimize the processing overhead for the defense in the case that each router performs the defending mechanism after a central system triggers. Possible neighbors are reachable routers that handoff can occur, and are decided according to the layout of geography, street, or indoors. Thus, like Fig. 4, we add the old parent-MR's IP address into an option field of *RS/NS* message or *DHCP request* message in order to inform their previous parent MR's IP address, because the handoff finishes after getting the CoA (Care of Address). With this simple option, a central defense system per AR can grasp the neighboring relationship between MRs, or the movement pattern per each host in detail. However, in the case that all MRs are performing the defense feature, *move notify* message at the IP layer should transmit to the old parent MR after the handoff finishes, in order to report the "movement to" information. It can be constituted as a simple message that the new parent MR notifies its own address to the old parent MR for the new joining host.

In the case of stateless auto-configuration, if a VMN/MR moves inside the domain of another MR, it can

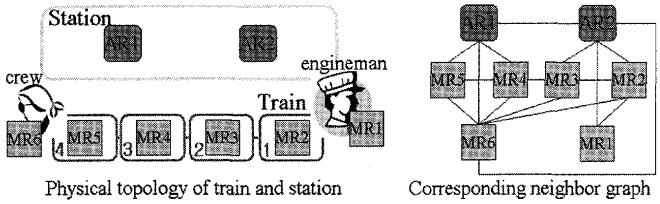


Fig. 5 An example placement of ARs/MRs and the neighbor graph.

get its new CoA from a periodic *RA* message, or a *RA* message for its *RS* message and performs the DAD (Duplicate IP Address Detection) by *NS* message in order to check the duplicate. Then, it adds the previous parent MR's IP address in the option field of *RS/NS* message in order to inform a new parent MR about old parent MR. Thus, if a parent-MR receives this information, it updates the *association pattern* $\Gamma(c)$ for client or neighbor graph, after it monitors the answer of DAD. Then the new parent-MR sends the *move notify* message to the old parent-MR in the option, as it needs. Lastly, the notified old parent-MR updates the *association pattern* $\Gamma(c)$ for client or neighbor graph.

If we consider the placement of ARs/MRs in a train and a station scenario as shown in Fig. 5, the systems of passengers and MR6 belonging to crew can move between four cars of the train or to/from station. However, MR1 belonging to engineman stays mainly at the first car of the train, or station. We can build the corresponding neighbor graph like Fig. 5, according to the path of motion. In the case of stateful auto-configuration like using DHCP, a moved mobile node can inform its old parent MR's IP address in the option field of *DHCP request* message in the process of getting CoA. Processes after receiving this information are similar to the stateless case. Also, since the space for neighbor graph is limited, the entry of table can be managed in a Least Recently Used (LRU) fashion.

According to the management capacity of each MRs, it can only manage a list of neighbor MRs instead of the movement pattern of each VMN/MR, or this neighbor graph can be managed in the centralized style by an AS (Authentication Server) or AR (Access Router). Thus, if a VMN moves to the administration domain of the AS, AS keeps track of the path information per each moving VMN through the process of authentication. In this case, if a MR detects the DDoS attack, it can request the AS to notify the candidate neighbors of the information for attack agents.

4. Evaluation

We suggested the architecture for defending the spoofing attack that consisted of five processing steps. In this section, we will evaluate the performance of each step from the various points of view.

4.1 Detection & Identification Speed

First, we consider how quickly the spoofing traffic can be detected. Our previous evaluation of detection performance

in a wired network resulted in detection and identification times of less than 50 ms (milliseconds) for all three suggested schemes [5]. Although the environment considered here is wireless rather than wired, the detection and identification mechanism for a MR is almost the same as the monitoring agent in [5]. This result is dependent on the number of received packets, the capacity of the monitoring agent, whether the IP addresses of attack packets are entered in the *IP2MAC* table in the case of scheme2/3 [5], the spoofed address, the speed of attack, and so on. In order to explain this relationship, we define the symbols that we use in the equations, as shown in Fig. 6.

There are two ways to detect a spoofing attack such as the one described in Sect. 3. One way is to use network configuration information. In this case, we can represent the relationship between the detection /identification time and influential elements as shown in Eqs. (1) and (2) in Fig. 6. Another way is to check the rate of change of source IP addresses for a given source MAC address. We can represent the relation between detection time and influential elements as shown in Eq. (3) in Fig. 6. In this case, the identification time does not need to include additional time, because the agent can immediately determine the IP address of the DDoS attack agent in the detection process. Of course, these upper bound times can decrease if the sets (e.g. CA_i) or *IP2MAC* table are managed as a tree data structure. For example, Eq. (3) can be changed into Eq. (4).

$$Tdt \leq uTcmp \cdot |CA_i| + uTcmp \cdot |DA_i| + uTcmp \cdot |RA_i| + Tsp$$
$$= uTcmp \cdot (|CA_i| + |DA_i| + |RA_i|) + Tsp \quad (1)$$
$$Tident \leq uTcmp \cdot |IP2MAC| \cdot isThere(macAdr) \quad (2)$$
$$Tdt \leq uTcmp \cdot |IP2MAC| \cdot 2 + Tsp \quad (3)$$
$$Tdt \leq uTcmp \cdot \log(|IP2MAC|) \cdot 2 + Tsp \quad (4)$$

- *Tdt*: Time for detecting spoofing attack
- *Tident*: Time for identifying the real IP address of spoofing attack agent
- *uTcmp*: Unit time for comparing IP/MAC address with an entity of address set (e.g. CA_i)
- *isThere(macAdr)*: a function that returns 1 if there is an entry for a MAC address in the *IP2MAC* table, otherwise 0. If this function outputs 0, the identification process fails.
- *Tsp*: Spent time that the first spoofing packet appears after the attack is mounted
- *IP2MAC*: IP & MAC address mapping table that the detection/identification agent manages

Fig. 6 Equations related with detection/identification speed.

4.2 Simulation Results

At first, we configure a simple but general simulation network with reference to NEMO standard document [9] as shown in Fig. 7. In order to analyze the influence on a NEMO of spoofing attack traffic and the effect of our each process, we simulate it with GloMoSim (Global Mobile Information Systems Simulation Library) that provides a scalable simulation environment for wireless and wired network systems [15]. The terrain of simulation network is (2600 meter, 400 meter). For routing, we divide the whole terrain into 104 sub-terrains as a 100-meter unit, and the italic number in a sub-terrain indicates the number of sub-terrains. Thus, the traffic of a node locating in sub-terrain *1, 27, 53, 79, 80, 81* or *82* would be forwarded to MR5, and the parent-MR of a node locating in sub-terrain *2, 3, 29, 30, 54, 55, 56, 57* or *58* is MR3. We depict a zone of sub-terrains served by each MR with a bold line.

For generating attack traffic, we configure five moving attack nodes (node 0,17,18,19 and 23) and one stationary attack node 22. Three moving attack nodes start to move at (50,50), (350,250), (950,250), (50,350), and (850,350) respectively at the start of simulation by the random-waypoint way with a minimum speed (1 meter per sec) and a maximum speed (20 meter per sec). Each attack node starts to mount the IP spoofing attack to node 1 at 10, 20, 30, 40 and 50 seconds in order of node 0,19,17,18 and 23, and they finish to attack at 120 seconds. The stationary node 22 starts to mount the attack to node 1 at 120 seconds until the end of simulation. This attack traffic is generated with 60B UDP traffic in 0.2 milliseconds interval. These parameters are oriented from the experiment of real IP spoofing attack tools at Fig. 1. Also, as normal traffic, node 21 continuously sends node 1 CBR (Constant Bit Rate) traffic of 1460B in 0.2 milliseconds interval at the start of simulation until the end of simulation.

We apply the second defense deployment. That is, AR2 and AR9 only perform the defense features at the first time, and then hand over the identification feature to child MRs when detecting the attack in order to find the exact attack agent. We set CSMA for the wired link and 802.11b for the wireless link, thus the link bandwidth is 11 Mbps.

First of all, Fig. 8 depicts the *Forwarding Packet Count* at AR2/9 and MRs as simulation time (unit is second) goes by, performing defense features step by step. In Fig. 8(a), because of no action for defense, the traffic of nodes are

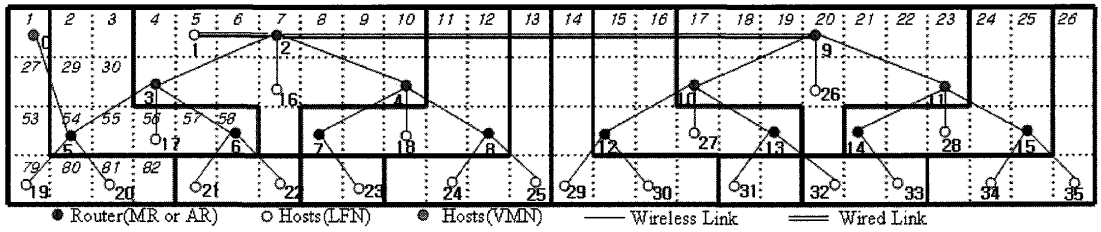
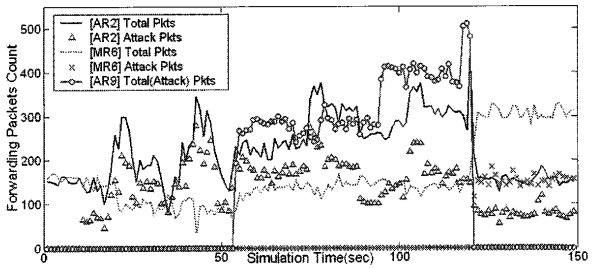
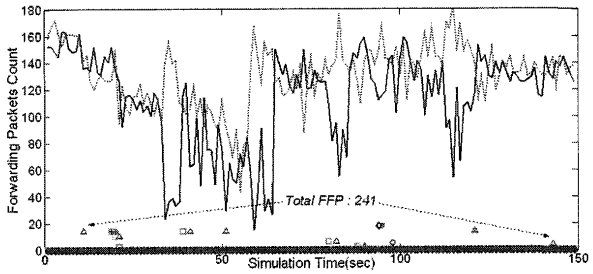


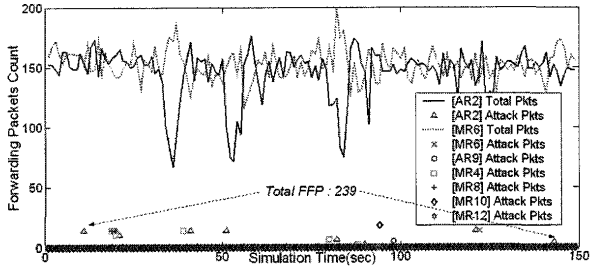
Fig. 7 Simulation network configuration.



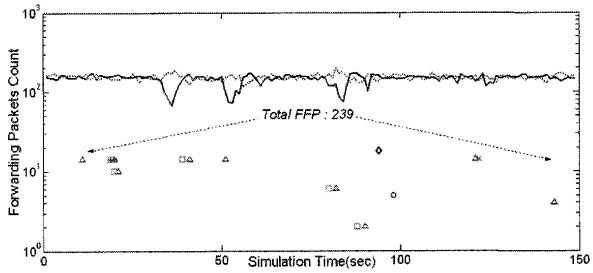
(a) No-Action (No Filtering/No Isolation/No Notification)



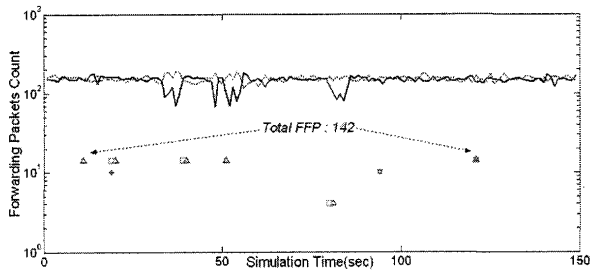
(b) L3 Filtering



(c) L3 Filtering/L2 Isolation



(d) L3 Filtering/L2 Isolation



(e) L3 Filtering/L2 Isolation/Notification

Fig. 8 *Forwarding Packet Count* at the routers (Legend of (b),(d) and (e) is the same as (c)).

fluctuated from 10 s in which node 0 starts to attack, as attack agent moves. For example, node 0 and 19 move near to MR3 at around 30 s, and their traffic through MR3 competes with the traffic of AR2. Thus, the traffic of AR2 decreases deeply at that time. Due to the similar reason, the traffic of AR2 decreases deeply in comparison with that of MR6, at around 60 s and 80 s. These decrease situations occur in the case (b) and (c) also. At around 55 s, attack agents move within the NEMO of AR9, thus the forwarding count of AR9 increases considerably at that time. As the first defense, *L3 filtering* filters out the attack traffic mostly as shown in Fig. 8(b). However, the traffic of AR2/MR6 is fluctuated also, as the start of attack and the mobility of attack agents. Figure 8(c) shows that *L2 isolation* can mitigate their influence. Straightforwardly, while the sum for forwarding normal packets of MR6 is 19579 in the (b), the sum is 22878 in the (c). Therefore, *L2 isolation* makes the performance of normal traffic enhance. For comparing results before and after the application of *Notification*, we depict (d) as a log scale graph of (c).

As the final defense, *Notification* decreases *FFP* (*Faulty Forwarding Packets*) that is a packet count of forwarding the attack traffic. Due to *Notification*, the decrease in *FFP* value is 41%, and moreover the decrease amount would enlarge in the real network due to severer attack, a number of attack agents, and more complicate network configuration. In addition to, we perform a t-test with the *FFP* value of Figs. 8(d) and 8(e) for comparing the difference between *FFP* values before and after an intervention of *Notification* feature. The t-test procedure produces a p-value of 0.0267 less than significant level 0.05, thus we can conclude that the intervention of *Notification* feature produces statistically significant improvements in the *FFP* value.

Defense Ability & Overhead of *Notification*

In order to evaluate our proactive *Notification* feature using neighbor graph, we compare it with other possible variants, NOTI1 and NOTI2. They are different in whom a detecting node is notified to. For example, if AR2 would detect an attack, it would notify to MR2/3/4/5/6/7/8 in the NOTI1, to MR3/4 in the NOTI2, and to MR3/4/6/7 in the NOTI3.

- **NOTI1**: Notification to all MRs under the AR
- **NOTI2**: Notification to its parent and child MRs
- **NOTI3**: Notification to moving possible neighbor MRs using neighbor graph

We use the following criteria to compare the performance and efficiency: *CNTL* and *FFP*. *CNTL* (Control message) measures the number of control messages to need for notification, which indicates the overhead for notification feature. As the performance of notification, *FFP* measures the sum of forwarded attack packets at each router. Basically, this experiment uses the network configuration and scenario of Fig. 8, but we simulate during 1000 s for enough simulation and vary four parameters that could affect the efficiency of *Notification*: (1) number of stationary attack agents and their deployment types, (2) number of moving attack agents, (3) movement speed of attack agent, and (4) number of nodes in a NEMO.

Firstly, to examine the effect of the number of stationary attack agents and their deployment types (Dispersion and Concentration), we gradually add stationary attack

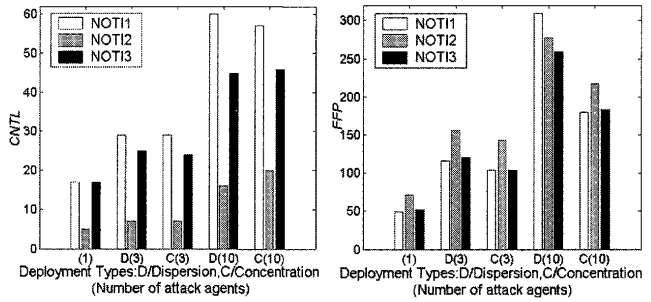


Fig. 9 *CNTL* and *FFP* vs. the number of stationary attack agents including a moving attack agent and deployment type.

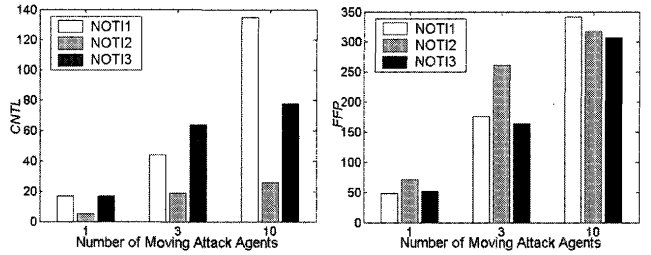
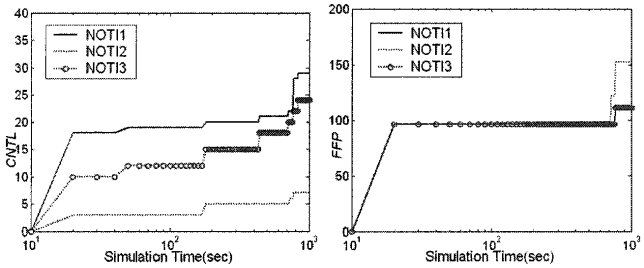


Fig. 10 *CNTL* and *FFP* vs. the number of moving attack agents.

agents to a moving attack agent at the node 0. Figure 9 depicts the effect as *CNTL* and *FFP*. The value in the parentheses indicates the number of attack agents. As the number of attack agents increases and they disperse, the *CNTL* and *FFP* increase. NOTI2 has the smallest *CNTL* value, but NOTI3 using neighbor graph has mostly the smallest *FFP* value. Also, as the number of the attack agents increases and they disperse, NOTI3 has better performance and less overhead than NOTI1. The effect on the number of moving attack agents is shown in Fig. 10. As the number of moving attack agents increases, the *CNTL*/*FFP* of NOTI1 rapidly increase. The defense performance of NOTI3 using neighbor graph is more prominent in the environment of many existing moving attack agents.

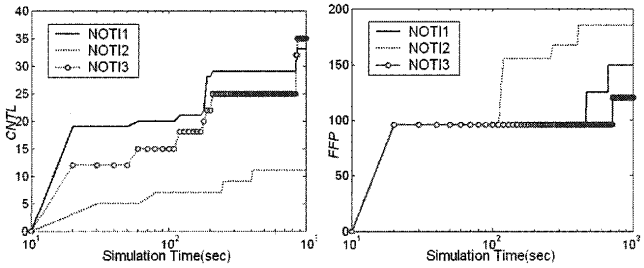
In Fig. 11, as the movement speed of two attack agents (e.g. (a) human speed, (b) motorcycle/car) increases, *CNTL* and *FFP* also increase because of frequent handoff. NOTI2 has the smallest *CNTL* value in all speeds, but the *FFP* of NOTI2 increases most steeply as the speed increases. NOTI3 using neighbor graph has the smallest *FFP* value and increases most gently as the speed increases, although resulting in the a little high *CNTL* value at the high speed. In conclusion, when we consider the obtainable benefit (decrease of *FFP*) with overhead (*CNTL*), NOTI3 is the best solution even at the high speed.

Lastly, Fig. 12 depicts the effect on the number of nodes in a NEMO. We combine two NEMOs of AR2/AR9 in Fig. 7 into a bigger NEMO in order to enlarge the network size at this experiment, and mount the attack by a moving attack agent based on similar simulation scenario using for Fig. 8. As the number of nodes/MRs is large, the *CNTL* of NOTI1 steeply increases because of the notification to all MRs under an AR, as we expect. On the other side, the increase degree of *FFP* value for three *Notifications* is similar, but



(a) Movement Speed of Attack Agent: 1~2 m/s

(a) Movement Speed of Attack Agent: 1~2 m/s



(b) Movement Speed of Attack Agent: 15~30 m/s

(b) Movement Speed of Attack Agent: 15~30 m/s

Fig. 11 *CNTL* and *FFP* vs. movement speed of attack agent.

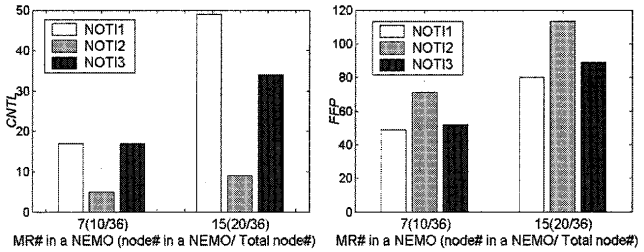


Fig. 12 *CNTL* and *FFP* vs. the number of nodes in a NEMO.

NOTI2 has the largest *FFP* value.

In conclusion, we can get two important points about notification feature. Firstly, *Notification* feature strengthens the defense mechanism in comparison with performing only three defense features (detection, L3 filtering and L2 isolation) especially against mobile attacks. Secondly, NOTI2 notifying only to parent and child MRs has the smallest overhead (*CNTL*) for all parameters, but its defense ability (*FFP*) is not good relatively. In the view of defense ability, especially NOTI3 makes a conspicuous figure in the severe attack by lots of attack agents such as DDoS attack, and in the attack by fast mobile nodes.

5. Conclusion

In this paper, we have proposed a mechanism that rapidly and proactively defends against IP spoofing attacks on a mobile network. The defense mechanism consists of speedy detection, filtering of attack packets, identification of attack agents, isolation of attack agents, and notification to neighboring routers. Of greatest importance on a mobile network are the processes of isolating attack agents in layer 2 and notifying neighboring routers. Filtering layer 2 access authorities in order to isolate attack agents can minimize their

effects on normal traffic transmission. If attack agents, or a NEMO that includes attack agents, hand off to neighboring networks, the neighboring routers can proactively cut off the attack traffic, thus minimizing the extent of the damage. For the notification to proper neighboring routers, we designed the neighbor graph managed with the extension of IPv6. We have also considered the difficulty of deploying defense agents on each NEMO, as the capability of a NEMO/MR can be small, and we have therefore suggested steps to defend against a spoofing attack when there is one representative defending agent per AR.

We have also simulated our mechanism on a mobile network, in order to analyze the influence of an attack and to prove the possibility of a speedy and proactive defense. The results showed that filtering the access authority of layer 2 to isolate the attack agents, as well as filtering the attack traffic, decreased the attack's influence on normal traffic transmission. Finally, we obtained a maximized defense result with the proactive notification feature, and compared our notification feature with two possible variants by four parameters. Especially, the notification to moving possible neighboring MRs especially made a conspicuous figure in the severe attack by lots of attack agents such as DDoS attack, and in the attack by fast mobile nodes.

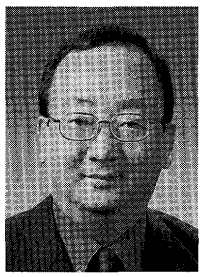
Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

[1] X. Geng, Y. Huang, and A.B. Whinston, "Defending wireless infrastructure against the challenge," Mobile Netw. Appl., vol.7, pp.213–223, 2002.
[2] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," IEEE INFOCOM 2004, pp.2404–2413, March 2004.
[3] S. Jung, F. Zhao, S.F. Wu, H. Kim, and S. Sohn, "Threat analysis on NEMO basic operations," IETF Internet Draft: draft-jung-nemo-threat-analysis-02, Work In Progress, Feb. 2004.
[4] T. Aura, "Cryptographically generated addresses (CGA)," IETF RFC 3972, March 2005.
[5] M. Kim and K. Chae, "Detection and identification mechanism against spoofed traffic using distributed agents," ICCSA2004, LNCS 3043, pp.673–682, May 2004.
[6] T. Hasegawa, S. Ano, and F. Kubota, "Programmable traffic monitoring method based on active network techniques and application to DDoS detection," IEICE Trans. Commun., vol.E87-B, no.7, pp.1890–1899, July 2004.
[7] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," IETF RFC 2827, May 2000.
[8] T. Lee, W. Wu, and T.W. Huang, "Scalable packet digesting schemes for IP traceback," Proc. ICC 2004, vol.27, no.1, pp.1008–1013, June 2004.
[9] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," IETF RFC 3963, Jan. 2005.
[10] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," IETF RFC 3775, June 2004.
[11] A. Mishra, M. Shin, N.L. Petroni, Jr., T.C. Clancy, and W.A. Arbaugh, "Proactive key distribution using neighbor graphs," IEEE Wirel. Commun., vol.11, pp.26–36, Feb. 2004.
[12] A. Mishra, M. Shin, and W.A. Arbaugh, "Context caching using neighbor graphs for fast handoffs in a wireless network," IEEE INFOCOM 2004, vol.23, no.1, pp.351–361, March 2004.
[13] T. Narten, R. Draves, and S. Krishnan, "Privacy extensions for stateless address autoconfiguration in IPv6," IETF Internet Draft: draft-ietf-ipv6-privacy-addrs-v2-04, Work In Progress, May 2005.
[14] S. Hagen, IPv6 Essentials, O'REILLY, July 2002.
[15] GloMoSim, available at http://pcl.cs.ucla.edu/projects/ glomosim/ (last visited on March 1, 2006).

Mihui Kim received the B.S. and M.S. degrees in Computer Science and Engineering from Ewha Womans University, Korea, in 1997 and 1999, respectively. During 1999–2003, she stayed in Switching & Transmission Technology Lab., Electronics and Telecommunications Research Institute (ETRI) of Korea to research and development the MPLS (Multi Protocol Label Switch) System and the 10 Gbps Ethernet System. She is now a Ph.D. student at the Graduate School of Engineering, Ewha. Her research interests include mobile network security, DDoS attack defense, and sensor network security.



Kijoon Chae received the B.S. degree in Mathematics from Yonsei University, Korea, in 1982, the M.S. degree in Computer Science from Syracuse University, U.S.A., in 1984, and Ph.D. degree in Computer Science and Engineering from North Carolina State University, U.S.A., in 1990. From 1990 to 1992, he was an Assistant Professor in the Department of Computer Science at United State Naval Academy, U.S.A. Since 1992, he has been a Professor in the Dept. of Computer Science and Engineering at Ewha Womans University, Korea. His current research interests include network security, sensor network, and performance evaluation and protocol design of wireless/mobile networks.