Attacker Traceback with Cross-layer Monitoring in Wireless Multi-hop Networks

Yongjin Kim
Dept. of Electrical Engineering – Systems
University of Southern California
Los Angeles, CA 90089-2562
yongjkim@usc.edu

Ahmed Helmy
Dept. of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2562
helmy@usc.edu

Abstract
Denial-of-Service (DoS) and Distributed DoS (DDoS) attacks can cause serious problems in wireless networks due to its limited network/host resources. Attacker traceback is a promising solution to take a proper countermeasure near the attack origin, for forensics, and to discourage attacker from launching attacks. However, attacker traceback in wireless multi-hop networks is a challenging problem, and existing attacker traceback schemes developed for the Internet cannot be directly applied to wireless multi-hop networks due to the peculiar characteristics of wireless multi-hop networks (e.g., dynamic/autonomous network topology, limited network/host resources such as memory and bandwidth). We introduce a protocol framework for attacker traceback geared toward wireless multi-hop networks with special attention to cross-layer abnormality monitoring. The basic building blocks of our protocol framework consist of *abnormality detection, abnormality characterization*, *abnormality searching*, *abnormality matching, and countermeasure*. We show that our protocol framework successfully tracks down attacker (Avg. of 100% in DoS attacker traceback, avg. of 96% in DDoS attacker traceback) under diverse network environments (e.g., high background traffic, DDoS attack, and partial node compromise) with low communication, computation, and memory overhead.

Categories and Subject Descriptors
C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General – *Security and Protection* G.3 [**PROBABILITY AND STATISTICS**]: *Statistical Computing*

General Terms: Algorithms, Security

Keywords: DoS/DDoS attack, Attacker Traceback, Cross-layer Monitoring, Wireless Multi-hop Networks

1. INTRODUCTION
Wireless multi-hop networks include Mobile Ad-hoc NETworks (MANET), wireless mesh networks, and wireless sensor networks, among others. Wireless multi-hop networks have been under active research due to their numerous promising applications and their practical deployment is near. However, security issues are not properly addressed in wireless multi-hop network research. Especially, DoS/DDoS attacks can cause a serous problem since (1) they are easy to perform using popular tools, and (2) wireless multi-hop networks are severely limited in network resources (e.g., bandwidth) and host resources (e.g., battery, and memory).

The different types of DoS/DDoS attacks can be broadly classified into software exploits and flooding attacks. In software exploits (e.g., Land attack, teardrop attack [1][15]), the attacker sends a few packets, or even single packet, to exercise specific software bugs within the target's OS or application, disabling or harming the victim. On the other hand, in flooding attacks, one or more attackers send incessant packet streams aimed at overwhelming link bandwidth or computing resources at the victim. In this paper, we mainly focus on flooding-type DoS/DDoS attacks since they cannot be fixed with software debugging. In flooding-type DoS/DDoS attacks, an attacker transmits a large number of packets towards a victim with a spoofed source address. For instance, in SYN Flood [2], at least 200-500 pps (packet per second) of SYN packets are transmitted to a single victim. UDP Echo-Chargen [4] and Smurf [3] also attacks victim using a large amount of packets with a spoofed address. It is reported that DoS attacks occur more than 4,000 times per week, and more than 600,000 pps of attack packets are used for attack in some cases [8] on the Internet. In general, we can say that the following are some characteristics of flooding-type DoS/DDoS attacks: (I) Traffic volume is abnormally increased during attack period. (II) Attackers routinely disguise their location using incorrect/spoofed addresses. (III) Such attacks may persist for tens of minutes, and in some case for several days [1].

The goal of attacker traceback is to identify the machines that directly generate attack traffic, as well as the network path this traffic subsequently follows [5]. There are several attacker traceback schemes proposed for the Internet such as packet marking [14], logging [13], ICMP traceback [6], etc [5]. Such traceback schemes developed for the fixed networks are not directly applicable to wireless multi-hop networks due to the following peculiar characteristics of wireless multi-hop networks: (1) In wireless multi-hop networks, there is no fixed infrastructure. Each node works as an autonomous terminal, acting as both host and router. (2) In general, network bandwidth and battery power are severely limited in wireless multi-hop networks compared to wired networks. (3) Nodes in wireless multi-hop networks have limited trust.

To perform efficient DoS/DDoS attacker traceback under such a harsh environment in wireless multi-hop networks, we propose a protocol framework for attacker traceback. Basically, we pay special attention to cross-layer (network layer and MAC layer) abnormality and extract useful information for attacker traceback. Every node captures protocol layer abnormality, which is observed during attack, and statistically characterizes the abnormality. The abnormality characterized by victim is called an attack signature and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SASN '06, October 30, 2006, Alexandria, Virginia, USA.
Copyright 2006 ACM 1-59593-554-1/06/0010...$5.00.

abnormalities characterized by intermediate nodes are called candidate attack signatures. Then, the victim sends a query with attack signature to intermediate nodes, to find the region that observed a similar candidate attack signature. The searching process is recursively continued to the attack origin. For searching, we use overharing capability of MAC layer to increase robustness against node compromise and reduce false negative/positive. After finding the attack origin, we take proper countermeasure to stop/lessen any attack effectively using cross-layer information (MAC layer and network layer).

The contribution of this paper can be summarized as follows:

- We use cross-layer information (i.e., network layer and MAC layer) to increase traceback accuracy. Noise factor by background traffic is largely reduced by using cross-layer information. In addition, cross-layer information provides robustness against highly Distributed DoS (DDoS) attacks.
- We use overhearing capability of MAC layer, which increases robustness against node compromise and reduce false negative/positive. In addition, it provides robustness against partial node mobility on the attack route.
- We propose traceback-assisted countermeasure, which provides optimal defense strategy against attack traffic and decreases negative impact on legitimate traffic.
- We perform extensive simulation-based analysis to show the efficacy of our proposal

The paper is organized as follows: In section 2, we briefly describe existing attacker traceback schemes. In section 3, we provide the overview of our protocol framework. We describe abnormality detection, characterization, matching, searching, and overall traceback algorithm in section 4,5,6,7, and 8 respectively. In section 9, we provide a traceback-assisted countermeasure scheme. In section 10, we provide performance analysis for our protocol framework. We conclude our paper and present future work in section 11.

2. RELATED WORK

The method in [7] using controlled flooding tests network links between routers to determine the origin of the attack traffic. Downstream node intentionally sends a burst of network traffic to the upstream network segments. At the same time, it checks incoming attack traffic for any changes. From the changes and frequency of the incoming attack traffic, the victim can determine which upstream router the attack traffic is coming from. The same process is continued a level higher until finally reaching the attacker. Since this is a reactive method, the trace needs to be completed before the attack is over.

Packet marking [14] and ICMP Traceback Message (iTrace) [6] attempt to distribute the burden of storing state and performing computation for attacker traceback at the end hosts rather than in the network. For instance, in ICMP-based notification, a router generates an ICMP message containing information about where each packet came from and where it was sent. Then, routers notify the packet destination of their presence on the route. Collection of these messages can be used to trace the attack origin. ICMP traceback message uses ICMP but limits to generating a ICMP message for every 20,000 packets (recommended). In Probabilistic Packet Marking (PPM), routers insert traceback data into each packet probabilistically, so the number of packets marked at each router is enough for the reconstruction of attack path at the victim.

Logging scheme requires the routers to log meta-data in case an incoming packet proves to be offensive. Audited packet flow is logged at various points throughout the network and then used for appropriate extraction techniques to discover the packet's path through the network. To reduce the size of the packet log and provide confidentiality, hash-based logging is proposed [13].

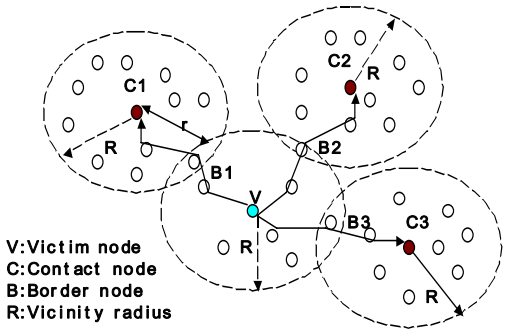
The existing schemes developed for the Internet are not directly applicable to wireless multi-hop networks due to the following reasons: (I) Intermediate relay nodes in wireless multi-hop networks can move in/out and may fail due to power outage, frequently changing network topology. In addition, each node in wireless multi-hop networks has limited trust due to the autonomous nature of nodes. Hence, traceback schemes that rely purely on relay nodes are problematic in terms of robustness and trust. (II) Storage capacity of each node is limited in wireless multi-hop networks. In packet marking and logging, a large amount of per-packet information needs to be stored at either end-host or inside the network. (III) Existing schemes incur high processing load for attack path reconstruction. For instance, in iTrace, end host first searches the database, which stores path information of packets. Then, based on the per-packet information, end-host should run reconstruction algorithm to find out attack path. On the other hand, controlled flooding consumes a lot of bandwidth for traceback, which is highly undesirable in bandwidth-constrained wireless multi-hop networks.

SWAT [11] is the first traceback protocol developed for ad-hoc networks. SWAT consists of two main building blocks: Traffic pattern/volume matching and small world construction. It uses Traffic Pattern Matching (TPM) and Traffic Volume Matching (TVM) techniques to deal with address spoofing problem and utilizes a small-world model for efficient search. However, SWAT has the following drawbacks: (1) SWAT cannot successfully trace back attacker when a high volume of background traffic exists. (2) SWAT fails to track down Distributed DoS (DDoS) attackers. (3) SWAT also shows weakness under node collusion, and false reporting, since it relies only on relay nodes of attack traffic for traceback. (4) SWAT does not provide any countermeasure mechanism after traceback.

3. ARCHITECTURAL OVERVIEW

Our traceback protocol framework consists of the following five architectural components: (1) Abnormality detection. (2) Abnormality characterization. (3) Abnormality matching. (4) Abnormality searching. (5) Countermeasure. Abnormality is detected at all nodes in the networks. Note that there exists a difference between "attack detection" and "abnormality detection." The attack detection is done by intrusion detection of victim, with application-level information. On the other hand, abnormality detection in our scheme is done by every node with available network/MAC layer information. The purpose of abnormality detection is to capture and log any abnormality as (candidate) attack signature for later traceback. Basically, each node monitors network/MAC layer activity (e.g., number of packets and frames). Once abnormality, which is largely deviated from normal profile, is detected, the information is captured. The abnormality detected by either the victim or intermediate nodes is statistically characterized and logged. In our scheme, the abnormality is characterized by cumulative distribution function of data. The data is the number of frames over monitoring timeframe. Once the attack signature is characterized, victim node initiates efficient search, and matching process is done at the nodes that observe candidate attack signature. By finding nodes in the neighbors, which observe similar or same

attack signature (high matching level), we can find the nodes or region that relayed the attack traffic. The process is continued recursively from the neighbor nodes of victim back to the attack origin. For efficient and robust attacker searching, we use small world model [9] and overhearing capability of MAC layer activity. Helmy [9][10] found that path length in wireless networks is drastically reduced by adding a few random links (resembling a small world). These random links need not be totally random, but in fact may be confined to a small fraction of the network diameter, thus reducing the overhead of creating such network. The random links can be established using contacts [10]. As shown in Fig.1, victim node, *V*, sends queries with attack signature to its vicinity nodes (nodes within radius *R*) and contacts (*C1, C2*, and *C3*). To send to the contacts, the victim node chooses three borders, *B1, B2* and *B3*, to which it sends the queries. The borders in turn choose three contacts at *r* hops away, to which the borders forward the queries. If there is no node that observed (relayed or overheard) an attack signature, it suppresses query. Otherwise, it sends the next level query to the contact of contact. In doing so, we can perform directional search for DoS attacker traceback and multi-directional search for DDoS attacker traceback, where the search process has directionality towards attacker(s). Directional and multi-directional search significantly reduces communication overhead. We will verify the reduction in the simulation section. To provide robustness against node compromise, mobility and high background traffic, we take a majority voting approach. That is, we take a region as an attack route region, if a majority of nodes observes similar abnormality. A majority of nodes that overhear MAC layer abnormality can be found in an attack route region since the wireless medium is shared by neighbor nodes. Once attack origin is identified, we take a traceback-assisted countermeasure. Traceback-assisted countermeasure has the following advantages: (1) Countermeasure can be taken at the nearest place to attack origin. Consequently, bandwidth/memory consumption of intermediate nodes between attacker(s) and victim can be minimized. (2) We use cross-layer information and abnormality matching level information to maximize the efficiency of attack traffic dropping and minimize negative impact on legitimate traffic. We describe each component of our traceback framework in the following sections.



[Figure 1] Each node has vicinity of radius *R* hops. A victim sends query with attack signature to its vicinity nodes and border nodes B_i. Then, the border nodes choose one of its borders C_i to be the contact and sends query with attack signature.

4. ABNORMALITY DETECTION

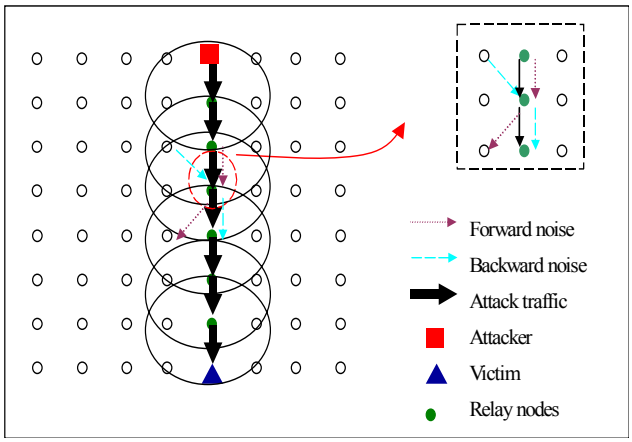
Abnormality detection is needed to start logging abnormality information. That is, each node monitors protocol layer activity and if abnormality is observed, a node logs the abnormality as candidate attack signature. The candidate attack signature is compared with attack signature, which is characterized by the victim for traceback. To detect the abnormality, we need to define normal profile. Normal profile, A_R, is defined based on information observed during period $[t_0, t_n]$. Let A_S the number of frames in a given unit time slot and A_R be the average number of frames of the long-term reference model, then the distance of the Fractional Deviation from the Mean (FDM) statistic is given as follows.

$$Dist = \frac{A_S - A_R}{A_R} \quad \text{(Eq.1)}$$

The distance, *Dist*, is defined as abnormality level. If the abnormality level is over a threshold (e.g., 0.5), it is considered suspicious, and candidate attack signature is logged.

- **Coarse-grained vs. Fine-grained detection**

We define coarse-grained detection and fine-grained detection. In coarse-grained detection, abnormality is detected in aggregate traffic level. The advantage is that it is computationally simple. However, the problem of aggregate traffic-based abnormality detection is that it is hard to detect small abnormalities accurately under the presence of large/bursty background traffic or DDoS attack. That is, a small amount of increased abnormality is not detected as an abnormality, since it is under the threshold. If we lower the threshold, there is a good chance that we could erroneously capture normal traffic as an abnormality and decrease logging efficiency. To address the problem, we define fine-grained abnormality detection with cross-layer monitoring, which uses minimal fine-grained network/MAC layer information (i.e., destination address, previous-hop MAC address). In doing this we can drastically reduce noise traffic (i.e., background traffic from non-attacker nodes) that is included in attack traffic.



[Figure 2] Illustration of forward/backward noise reduction using cross-layer monitoring

First, we can reduce noise traffic using network-layer information. That is, candidate attack signature is captured based on traffic destined to each destination (i.e., we make abnormality table indexed by destination address). We can rely on the destination address, since an attacker does not spoof destination address to achieve his goal. As shown in Fig.2, a monitoring node (inside dotted circle) can remove noise traffic that is not destined to victim

node. We call the noise traffic forward noise. In addition, we can reduce noise traffic using MAC layer information. That is, by using the previous-hop MAC address, we can filter out background traffic, which is coming from a different route than the attack traffic. We call the background traffic backward noise. By using fine-grained cross-layer information of network layer and MAC layer (i.e., destination address, previous hop MAC address), we can drastically reduce noise traffic that is included in the attack traffic. In addition using MAC layer overhearing capability, abnormality monitoring region (solid circle area in Fig.2) is largely enhanced, which increases robustness against against node compromise, false reporting and mobility.

5. ABNORMALITY CHARACTERIZATION

Once an abnormality is detected, the abnormality needs to be characterized for matching test. We characterize the abnormality as cumulative distribution function [16]. That is, when the time series data (i.e., number of frames per unit time slot) in n unit time window, $(a_1,a_2,\ldots,a_n)$, is observed, the distribution function is given in terms of the order statistic. Let $y_1<y_2,<\ldots<y_n$ be the observed values of the order statistics of a sample $a_1,a_2,\ldots,a_n$ of size n. Then, the distribution function is defined as follows.

$$F_n(x)=\begin{cases}0, x<y_1,\\ k/n, y_k\le x<y_{k+1},\\ 1, y_n\le x.\end{cases}$$

(Eq.2)

Where $k= 1,2,\ldots,n-1$. We use $F_n(x)$ as characterized (candidate) attack signature. Sampling window, D, is expressed as follows.

$$D=n\bullet d$$

(Eq.3)

Where d is unit time window length. Traceback performance varies depending on efficient characterization. For efficient characterization, parameters such as unit time windows, and total time window need to be carefully desigined. We will analyze those factors in the analysis section.

- **Coarse-grained vs. Fine-grained characterization**

For fine-grained characterization, the destination address and previous hop MAC address are used for characterization (Table 1).

Destination_addr	Source_MAC_addr	Abrnoamlity
1	2	Ξ(1,2)
1	3	Ξ(1,3)
.	.	.
.	.	.
.	.	.
.	.	.

[Table 1] Abnormality table using cross-layer information

There is obvious tradeoff between coarse-grained and fine-grained characterization. When coarse-grained characterization is used, space complexity for abnormality logging becomes $O(1)$. However, abnormality matching and consequent traceback performance becomes low. On the other hand, when fine-grained characterization is used, space complexity becomes O(N*M), where N is the number of destination_addr and M is source_MAC_addr. However, traceback back performance is improved since background traffic is drastically decreased.

6. ABNORMALITY MATCHING

We are interested in using the Kolmogorov-Smirnov (KS) statistic D_n [16] to test the hypothesis that the two abnormality, $F_n(x)$,and $F_0(x)$ is matching. $F_0(x)$ corresponds to reference abnormality (i.e., attack signature), which is included in query message, and $F_n(x)$ is the candidate attack signature observed by intermediate nodes.

$$D_n=\sup_x[\ |F_n(x)-F_0(x)|\ ] \qquad \text{(Eq.4)}$$

$$H_0 : F_n(x)=F_0(x)$$
$$H_a : F_n(x)\neq F_0(x)$$

(Eq.5)

We accept H_0 if the distribution function $F_n(x)$ is sufficiently close to $F_0(x)$, that is, if the value of D_n is sufficiently small. The hypothesis H_0 is rejected if the observed value of D_n is greater than the selected critical value that depends on the desired significance level and sample size. When the H_0 is accepted (sufficiently similar), we can infer that the abnormality is matching, meaning that the attack traffic is traversed the region of candidate attack signature. Computation overhead of the matching test is very low with $O(1)$.

Similar to detection and characterization, we use coarse-grained and fine-grained matching. By reducing noise with fine-grained information, we can increase matching accuracy. In this section, we analyze how much noise can be reduced in more detail.

- **Fine-grained matching with network-layer information**

To investigate how much noise traffic can be removed using fine-grained network-layer information, we perform connection-level analysis. We first define total noise included in coarse-grained attack signature as follows.

$$N_{TN}^n=N_D^n+N_S^n$$

(Eq.6)

Where,

N^n_D: Noise traffic (Number of connections), which is heading to different destinations from victim,

N^n_s: Noise traffic (Number of connections), which is heading to the same destination (as victim) but not coming from attacker,

Noise reduction rate that can be achieved with fine-grained network-layer information is calculated as follows:

$$N_{RN}^n=\frac{N_D^n}{N_D^n+N_S^n}=\frac{N_{TN}^n-N_S^n}{N_{TN}^n}=1-\frac{N_S^n}{N_{TN}^n}$$

(Eq.7)

The noise reduction rate depends on the congestion factor and destination diversity. If there is high volume of traffic (high congestion) coming into a node, there is a high chance that there exists normal traffic heading to the victim. In addition, if the destination of traffic is not uniformly distributed (e.g., traffic is going into several server nodes only - low destination diversity), the chance of sharing the same destination as attack traffic becomes high. Taking into the congestion and destination diversity factor, we performed an analysis to show how much noise traffic can be reduced as follows:

$$N_S^n \approx E(N_S^n) = \sum_{i=0}^{N_C^n} i \bullet \Pr\{N_S^n = i\}$$

(Eq.8)

where,

$$\Pr\{N_S^n = i\} = {}_{N_{TN}} C_i \bullet (1/M)^i (1-1/M)^{N_{TN}^n - i}$$

(Eq.9)

M: Destination diversity factor
N^n_C: congestion factor

Hence, noise reduction rate is calculated as follows.

$$N_{RN}^n = 1 - \frac{\sum_{i=0}^{N_C^n} i \bullet {}_{N_{TN}^n} C_i \bullet (1/M)^i (1-1/M)^{N_C^n - i}}{N_C^n}$$

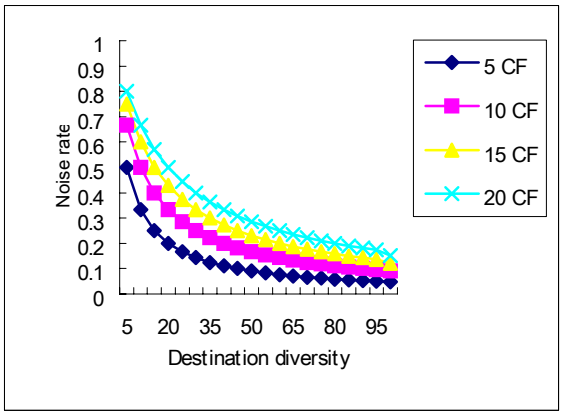
(Eq.10)

In addition, actual noise rate that is included in attack traffic is calculated as follows.

$$Noise \quad rate = \frac{N_{RN}^n}{1 + N_{RN}^n}$$

(Eq.11)

There exists a difference between noise reduction rate and noise rate. Even if we can drastically reduce relative noise rate (i.e., noise reduction rate) with the fine-grained scheme, the noise may still exist in attack traffic (noise rate >0). As we can see in Fig.3, we can drastically reduce noise rate, especially when destination diversity is high. However, noise still exists when destination diversion is low.



[Figure 3] Noise rate comparison with network-layer information

- **Fine-grained matching with MAC-layer information**

To investigate how much noise traffic can be removed using fine-grained MAC-layer information, we performed a connection-level analysis. We define total noise as follows:

$$N_{TN}^m = N_D^m + N_{SD}^m + N_{DD}^m$$

(Eq.12)

Where,
N^m_D: Noise traffic that is coming from neighbor node that does not relay attack traffic,
N^m_{SD}: Noise traffic that is coming from neighbor that relays attack traffic, and heading to the victim (but not attack traffic),
N^m_{DD}: Noise traffic that is coming from neighbor that relays attack traffic, but not heading to the victim.

Noise reduction rate that can be achieved with fine-grained MAC-layer information is calculated as follows:

$$N_{RN}^m = \frac{N_D^m}{N_D^m + N_{SD}^m + N_{DD}^m} = \frac{N_{TN}^m - N_{SD}^m - N_{DD}^m}{N_{TN}} = 1 - \left(\frac{N_{SD}^m}{N_{TN}^m} + \frac{N_{DD}^m}{N_{TN}^m}\right)$$

(Eq.13)

The noise reduction rate depends on how many one-hop neighbors exist. If there are many one-hop neighbors that generate background traffic to a node, we can reduce the background traffic noise by having separate attack signatures based on the one-hop neighbor node. The actual noise reduction rate and noise rate are calculated as follows:

$$N_{DD}^m + N_{SD}^m \approx E(N_{DD}^m + N_{SD}^m) = \sum_{i=0}^{N_C^m} i \bullet \Pr\{N_{DD}^m + N_{SD}^m = i\}$$

(Eq.14)

Where,

$$\Pr\{N_{DD}^m + N_{SD}^m = i\} = {}_{N_{TN}} C_i \bullet (1/P)^i (1-1/P)^{N_{TN} - i}$$

(Eq.15)

P: Number of one-hop neighbors
N^m_C: congestion factor

Hence, noise reduction rate is calculated as follows:

$$N_{RN}^m = 1 - \frac{\sum_{i=0}^{N_C^m} i \bullet {}_{N'_{TN}} C_i \bullet (1/P)^i (1-1/P)^{N_{TN} - i}}{N_C^m}$$

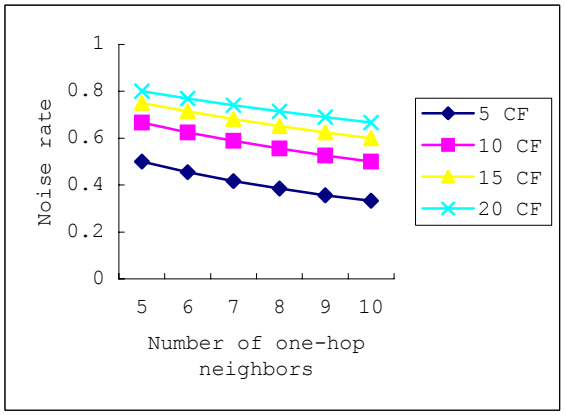
(Eq.16)

Noise rate is defined as follows:

$$Noise \quad rate = \frac{N_{RN}^m}{1 + N_{RN}^m}$$

(Eq.17)

As we can see in Fig.4, the noise rate gradually decreases as the number of one-hop neighbors increases. However, the noise rate is high when the number of one-hop neighbor is small.



[Figure 4] Noise rate comparison with MAC layer information

- **Fine-grained matching with cross-layer information**

To investigate how much noise traffic can be removed using cross-layer information (destination address, and previous-hop MAC address), we performed a connection-level analysis. Total noise with network-layer information was defined as follows:

$$N_{TN}^{n} = N_{D}^{n} + N_{S}^{n} \quad \text{(Eq.18)}$$

By further applying MAC-layer information on N^{n}_{S}, we can define the total noise as follows:

$$N_{TN}^{n,m} = N_{D}^{n} + N_{D}^{n,m} + N_{SD}^{n,m} \quad \text{(Eq.19)}$$

Where,

$N^{n,m}_{D}$: Noise traffic that is coming from neighbor node that does not relay attack traffic and heads to the victim,

$\boldsymbol{N^{n,m}_{SD}}$**:** Noise traffic that is coming from neighbor that relays attack traffic, and heading to the victim.

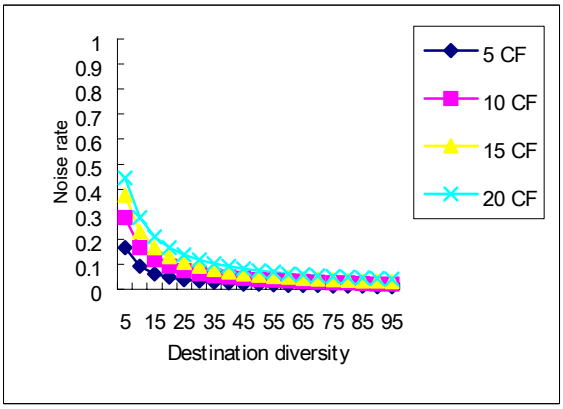
Noise reduction rate that can be achieved through cross-layer information is calculated as follows:

$$N_{RN}^{n,m} = \frac{N_{D}^{n} + N_{D}^{n,m}}{N_{D}^{n} + N_{D}^{n,m} + N_{SD}^{n,m}} = \frac{N_{TN}^{n} - N_{SD}^{n,m}}{N_{TN}^{n}} = 1 - \frac{N_{SD}^{n,m}}{N_{TN}^{n}} \quad \text{(Eq.20)}$$

Eq.20 implies that we can eliminate all the noise traffic except traffic that comes from the same one-hop previous neighbor and heads to the same destination (i.e., victim). Noise rate is defined as follows:

$$Noise \quad rate = \frac{N_{RN}^{n,m}}{1 + N_{RN}^{n,m}} \quad \text{(Eq.21)}$$

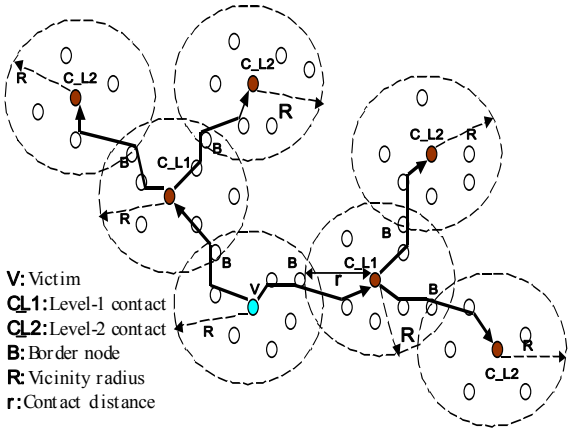
Fig.5 shows noise rate with random number of one-hop neighbors. Noise rate is reduced throughout various destination diversity, which drastically decreases the negative impact of background traffic on matching test.



[Figure 5] Noise rate comparison with cross-layer information

7. ABNORMALITY SEARCHING

For efficient and robust attacker searching, we use the small world model. Helmy [9] found that path length in wireless networks is drastically reduced by adding a few random links (resembling a small world). These random links need not be totally random, but in fact may be confined to a small fraction of the network diameter, thus reducing the overhead of creating such a network. The random links can be established using contacts [10]. Contact nodes are a set of nodes outside the vicinity, which are used as short-cut (random links) to build small world. We describe a detailed small world construction scheme in the following: Each node in the network keeps track of the number of nodes in its vicinity within *R* hops away. This defines the vicinity of a node. The vicinity information is obtained through the underlying routing protocol. Each node chooses its vicinity independently, and hence no major re-configuration is needed when a node moves or fails. There is no notion of cluster head, and no elections that require consensus among nodes.



[Figure 6] Small world construction with multi-level contacts. Victim, *v*, selects level-1 contacts. Level-1 contacts select its level-2 contacts.

The above contact selection scheme provides a mechanism to select *NoC* contacts that have distances up to *R+r* hops away from *V*. We call these contacts level-*1* contacts. To select farther contacts (contact of contact), this process is repeated as needed at the level-*1* contacts, level-*2* contacts and so on, up to a number of levels called maxDepth, *D*. SWAT also extends contact architecture for efficient traceback. However, our search policy has the following important distinctions. First, we take majority voting to find the region where attack traffic is traversed. It becomes possible since we use the overhearing capability of MAC-layer activity of neighbor nodes on the attack route. It increases robustness against node compromise, partial node mobility and reduces false positives due to a similar traffic pattern. Second, we use the signature energy concept, which is calculated by abnormality matching of KS fitness test. We will show that it increases against false negatives and false positives under diverse network environments. For robust searching, we define and use the following metrics:

- **Individual attack signature energy**

Each contact gathers individual attack signature energy for each of its vicinity nodes. The individual attack signature energy of node *i* is defined as follows:

$$E^{i} = \frac{1}{D_{i}}$$

(Eq.22)

where, D_i is defined as follows.

$$D_i = \sup_x[|F_i(x) - F_0(x)|]$$

(Eq.23)

D_i becomes small when there is high abnormality matching between the attack signature and a candidate attack signature. Consequently, E^i is increased when there is high abnormality matching.

- **Regional attack signature energy**

Regional attack signature Energy (*RE*) is defined as follows:

$$RE = \frac{E^i_{1/2}(u)}{\mu_{1/2}} \quad \text{(Eq.24)}$$

where, $E^i_{1/2}(u)$ is median of the signature energy among nodes of contact u that observe abnormality. $\mu_{1/2}$ is the median value of distance (i.e., hop counts) between contact and the nodes that observe similar abnormality. The reason we take the median value instead of an average is to prevent negative impact of false reports from malicious or compromised nodes. In addition, *RE* should satisfy the following condition:

$$\alpha = \frac{n}{N} > \delta$$

(Eq.25)

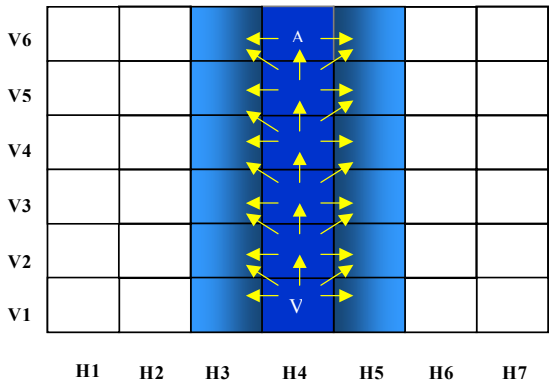
Where, α is the majority voting factor (*N*: total number of vicinity nodes of the contact, *n*: number of nodes that observe abnormality). *n* is drastically increased when we use MAC layer abnormality overhearing nodes. When, α is extremely low (e.g., α<0.1), we can infer that there is high chance of false reporting. Region around the attacker and attack route shows a high *RE* value. Intuitively, we can infer that the attacker is residing or attack traffic is traversing the region where high *RE* value is observed.

8. OVERALL ATTACKER TRACEBACK

8.1 DoS Attacker Traceback

We describe overall DoS attack traceback scheme as follows: (1) when a victim node, *V*, detects an attack such as SYN flooding, it first extracts attack signature. It then sends a query to the nodes within its vicinity and level-*1* contacts, specifying the depth of search (*D*) large enough to detect an attacker. The query contains a sequence number (*SN*) and an attack signature. (2) As the query is forwarded, each traversed node records the *SN* and *V*. If a node receives a request with the same *SN* and *V*, it drops the query. This provides for loop prevention and avoidance of re-visits to the covered parts of the network.

(3) In case a high *RE* is observed by vicinity of a victim and contacts, the first step of trace is completed. For instance, victim (*V*) sends query to the vicinity nodes and 5 level-*1* contacts in regions {(H3,V1), (H3,V2), (H4,V2), (H5,V1), (H5,V2)} around the victim in Fig. 7. Then, one level-*1* contact in region (H4,V2) reports to the victim that some of its vicinity nodes have observed high *RE*. To reduce the risk of false matching reports from vicinity nodes, the contact requests candidate attack signature observed at the vicinity nodes during given time slots instead of distributing attack signature to all vicinity nodes and waiting for individual attack signature energy response. Matching test is done at each contact.



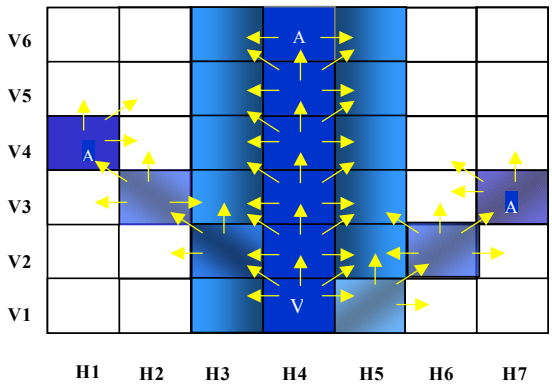
[Figure 7] Victim (*V*) sends queries with attack traffic signature to its neighbor region {(H3,V1), (H3,V2), (H4,V2), (H5,V1), (H5,V2)}. Only (H4,V2) region that observed highest *RE* sends next level queries to its own neighbor region. (Each cell corresponds to contact region, and intensity of color represents *RE*).

Although it cannot completely eliminate the risk of false matching report, it can reduce such risk. (4) Next, only the contact in region (H4,V2) that observes high signature matching in its vicinity sends next level query to level-*2* contacts, with the partial attack path appended to the query. It also reduces *D* by 1. This processing by contact is called in-network processing. Other contacts that do not have nodes that observe attack signature, suppress forwarding the query (query suppression). This results in *directional search* towards the attacker. (5) When there are no more contact reports or no other nodes outside the vicinity, the last contact reports the complete attack route to the victim.

Our scheme is based on majority voting (Eq.25). That is, even if some nodes move out from the attack route or are compromised by attackers, we can still find an attack route using available information from good nodes residing in the vicinity.

8.2 DDoS Attacker Traceback

In this section, we describe an overall DDoS attacker traceback scheme. DDoS attacks involve a sufficient number of compromised nodes to send useless packets toward a victim around the same time. The magnitude of the combined traffic is significant enough to jam, or even crash, the victim or connection links.



[Figure 8] Victim (*V*) sends queries with attack traffic signature to its neighbor region {(H3,V1), (H3,V2), (H4,V2), (H5,V1), (H5,V1)}. Regions {(H3,V2), (H4,V2), (H5,V2)} that observed highest *RE* sends next level queries to its own neighbor region.

Similar to DoS case, a victim node sends a query to its vicinity and level-*1* contacts with its characterized attack traffic signature. In

DDoS attacker traceback, multiple candidate attack signatures are observed and returned from multiple contacts. Unlike DoS attacker traceback, a combinational matching test needs to be done by a victim or lower level contact to find the branch attack route. That is, abnormality matching should be performed between the attack signature and all multiple candidate attack signatures. Then, contacts that show the highest matching level are selected as branch attack routes. For instance, in Fig.8, three responses are returned from level-*1* contacts in regions {(H3,V2), (H4,V2), (H5,V2)}. In this example, the highest abnormality matching is observed between the summation of the three candidate attack signatures from regions {(H3,V2), (H4,V2), (H5,V2)} and attack signature at the victim. As a result, a victim concludes that branch attack traffic comes from regions {(H3,V2), (H4,V2), (H5,V2)}. Contacts that are determined as the attack routes by the victim node perform next level query in a recursive manner. The searching process leads to multi-directional searching.

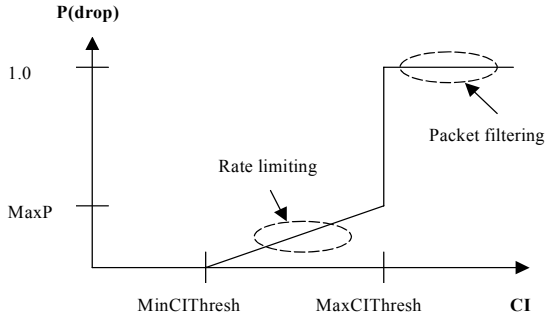
9. TRACEBACK-ASSISTED COUNTERMEASURE

Existing countermeasures against DoS/DDoS attack can be broadly classified into packet filtering and rate limiting. Current packet filtering and rate limiting techniques against DoS/DDoS attack have the following drawbacks: (1) They are taken at the nodes where an attack is detected. For instance, they are taken at the ingress point of victim. However, they are inefficient since the attack traffic already exhausts valuable network/host resources of intermediate nodes. (2) Packet filtering is challenging since it is hard to distinguish between malicious and legitimate traffic. Legitimate traffic may experience sudden QoS degradation due to packet filtering. (3) In rate limiting, it is hard to know how much rate limiting should be applied to reduce the negative impact on legitimate traffic and increase rate-limiting efficiency against attack traffic.

We propose a traceback-assisted countermeasure, which effectively users of traceback information. Basically our countermeasure mechanism finds the closest point to the attack origin and takes a countermeasure based on the abnormality matching level. We also use cross-layer information (i.e., destination address, previous MAC address) to increase countermeasure efficiency. That is, using cross-layer information, we can reduce negative impact on legitimate traffic and increase packet drop/rate-limiting efficiency against attack traffic, since we can differentiate attack traffic and legitimate traffic more accurately. Our scheme can be considered as a hybrid scheme between packet filtering and rate limiting with abnormality matching level information. That is, when abnormality matching level is the highest, we apply packet filtering. On the other hand, when abnormality matching is moderate level, we apply rate limiting based on abnormality matching level. To determine optimal rate limiting level under medium matching level, we define and use Confidence Index (*CI*). *CI* is normalized value between [0,1] of inverse of distance in KS fitness test. Rate limiting level (*P*) is determined with the following equation: (refer to Fig.9)

$$P = MaxP \bullet \frac{CI - MinCIThresh}{MaxCIThresh - MinCIThresh} \quad \text{(Eq.26)}$$

As shown in the Fig.9, when *CI* is very high it reduces to packet filtering, since it implies that there is no background traffic. On the other hand, when *CI* is medium, it becomes rate limiting based on *CI* level to reduce negative impact on legitimate traffic.



[Figure 9] CI-based Countermeasure

The advantage of using a *CI*-based countermeasure over applying fixed drop rate is multifold: (1) When *CI* is low, only a small amount of packets (either attack or legitimate packets) are dropped. Even if we cannot drop more attack packets, it does not cause a serious problem, since only a small amount of attack traffic exists. On the other hand, the negative impact on legitimate traffic is largely reduced. When *CI* is high, more packets are dropped. Even if a higher percentage of legitimate packets are also dropped, its negative impact is not significant, since only a small amount of legitimate traffic exists. We will compare our *CI*-based countermeasure with fixed rate-limiting scheme in detail in the analysis section. To further reduce the QoS degradation of legitimate traffic under this countermeasure, we use cross-layer information (e.g., MAC, network-layer information). Traffic is classified based on fine-grained information (i.e., destination address, previous-hop MAC address). When one class of traffic is identified as highly matching traffic with an attack signature, we apply rate limiting based on *CI* value for the class of traffic only.

To measure countermeasure efficiency formally, we define SDP as follows:

$$SDP = (Survived\ legitimate\ Traffic) * (Dropped\ attack\ traffic) \quad \text{(Eq.27)}$$

We will show the efficiency of our traceback-assisted countermeasure using SDP in the analysis section.

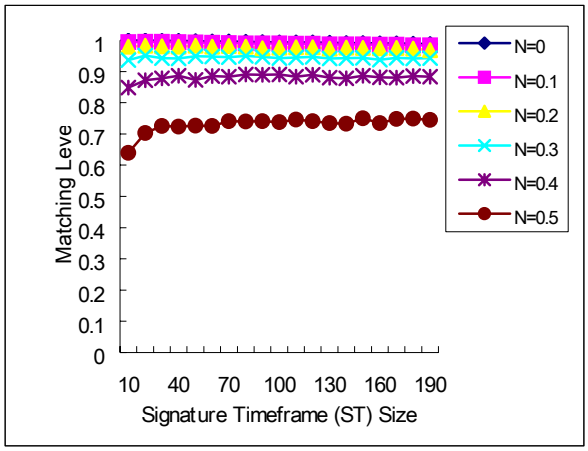
10. SIMULATION-BASED PERFORMANCE ANALYSIS

In this section, we analyze the performance of the proposed traceback protocol framework. We analyze each component of the framework, namely abnormality characterization, matching, searching and countermeasure. In addition, we analyze traceback success rate with the overall traceback mechanism. We have performed simulations using *ns-2* and *C* code. Transmission range of each node is set as 150*m*. Background traffic is generated from random source to random destination. We repeated each simulation 100 times in random topology and calculated the average value. We set *NoC* (Number of Contacts) = 6, *R* (vicinity radius) = 3, *r* (contact distance) = 3, *d* (search depth) =5 for contact selection and DSDV is used for underlying routing protocol.

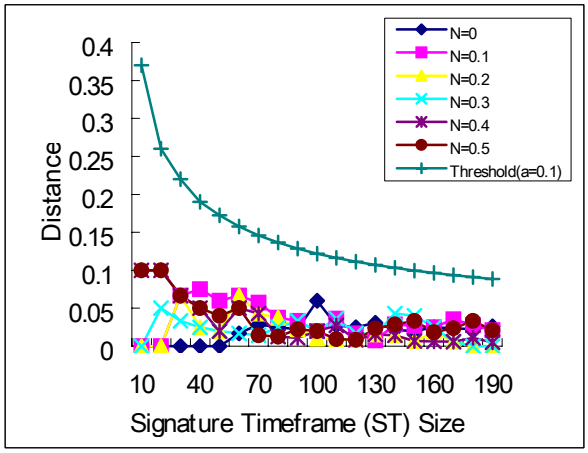
- **Abnormality characterization and matching**

Fig.10, and Fig.11 show the impact of time asynchrony on matching test. Time asynchrony represents attack signature shift among nodes, which is caused by geographically spread nodes that observe

traversing attack signature. We compare the impact of time asynchrony on matching performance between traffic pattern matching-based approach, which is used in SWAT and our scheme. In Fig.10, and Fig.11, *N* represents the percentage of time asynchrony in attack signature. For example when *N* is 0, two abnormalities (i.e., attack signature, candidate attack signature) is observed exactly at same time slot, which is unrealistic, due to propagation/transmission/queueing delay. ST size represents the total number of unit monitoring windows. As *N* becomes bigger, the matching level (i.e., correlation coefficient in SWAT, KS fitness test in our scheme) becomes lower, which may result in high false negatives. Obviously, it is because time asynchrony results in traffic pattern distortion between different observing nodes. Our scheme, which is based on KS-fitness test in Fig.11, shows less negative impact by time asynchrony. All the distances are below threshold (threshold is set with significance level of 0.1%), which represents a high matching level. This is because the KS-fitness test checks abnormality with distribution function instead of the time-series traffic pattern, which is used in SWAT.

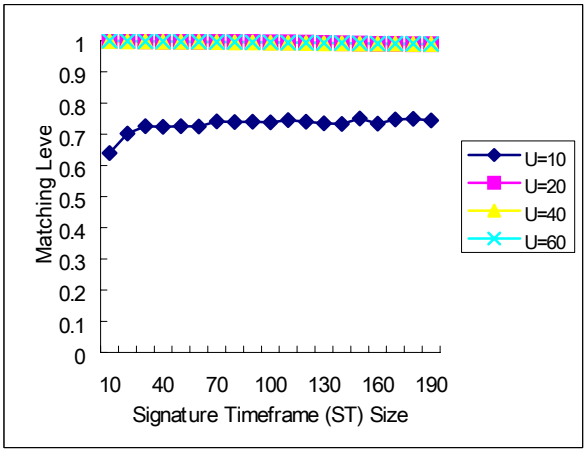


[Figure 10] Impact of time asynchrony on matching test (with pattern matching in SWAT)

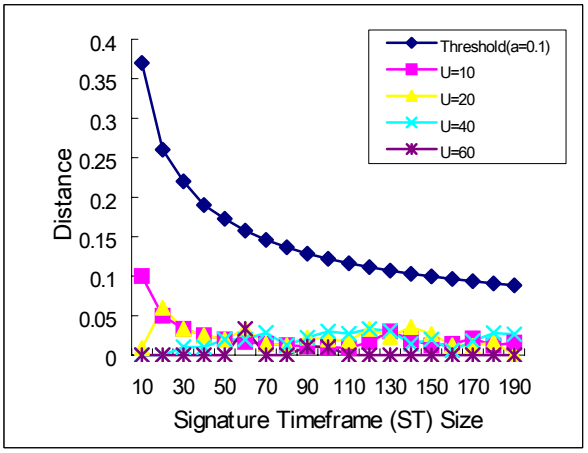


[Figure 11] Impact of time asynchrony on matching test (with our scheme). Distance represents D_n in Eq.4

In Fig.12 and Fig.13, we analyze the impact of unit monitoring window size (10 seconds, 20 seconds, 40 seconds and 60 seconds) and time asynchrony. It is shown that the negative impact of time asynchrony is increased when unit monitoring window is small in pattern matching. It is because a small distortion of traffic pattern can result in overall pattern mismatching under small unit window size. The disadvantage of long unit monitoring window is a delay in abnormality characterization. On the other hand, our scheme (Fig.13) shows stable performance across different unit window size due to the same reason in Fig.11.



[Figure 12] Impact of unit monitoring window on matching test (with pattern matching in SWAT)

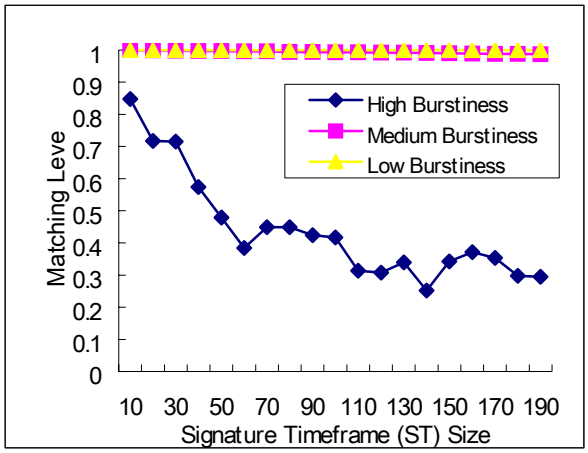


[Figure 13] Impact of unit monitoring window on matching test (with our scheme)

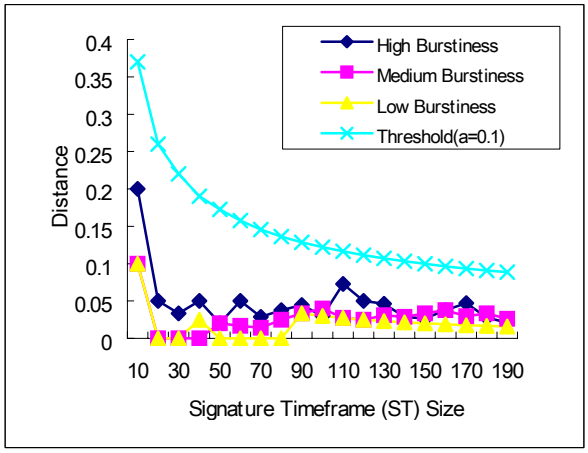
Fig.14 shows the impact of various background traffic on matching test. Unlike our initial expectation, larger ST size incurs low matching level. That is, when high bursty traffic exists and larger ST size is used, traffic matching level drastically goes down. It is because as ST size is increased, there exists more chance that the burstiness can affect the traffic pattern. Fig.15 shows the impact of background traffic on our scheme. We observe high matching level (low distance) regardless of ST size. This is because abnormality distribution of candidate attack signature is not affected by a small deviation from the reference profile (i.e., attack signature) in KS-fitness test. Fig.16 shows an abnormality matching level between an attack signature and random bursty background traffic, where *M* is the number of unit monitoring windows. High matching level (low distance) leads to a false positive. Originally, we were expecting that KS-fitness test would show a high false positive rate. However, KS-fitness shows a low false positive rate with high distance across most ST sizes.

Consequently, we can conclude that our scheme far outperforms traffic pattern-based traceback in terms of false positives and negatives under diverse parameter settings. There can be one exceptional case where a KS-test shows low performance. That is, when both traffic attack signature and candidate attack signature

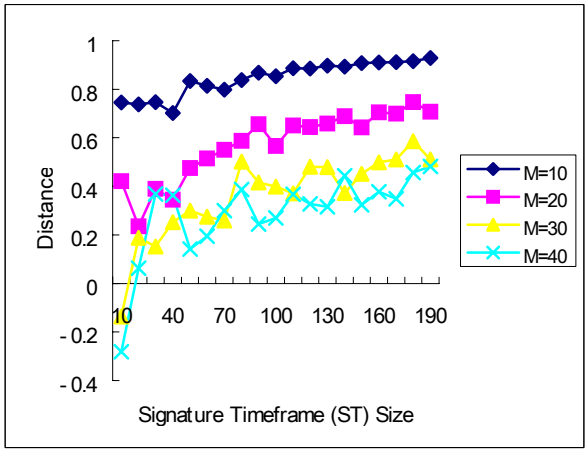
shows the same statistical characteristics (i.e., same average, variance, etc.) with different time-series traffic patterns, the KS test can cause false positives. However, this is considered a very rare case. In addition, there is no reason for an attacker to launch this kind of attack, since it can cause traceback success anyway.



[Figure 14] Impact of background traffic on matching test (with pattern matching in SWAT)



[Figure 15] Impact of background traffic on matching test (with our scheme)

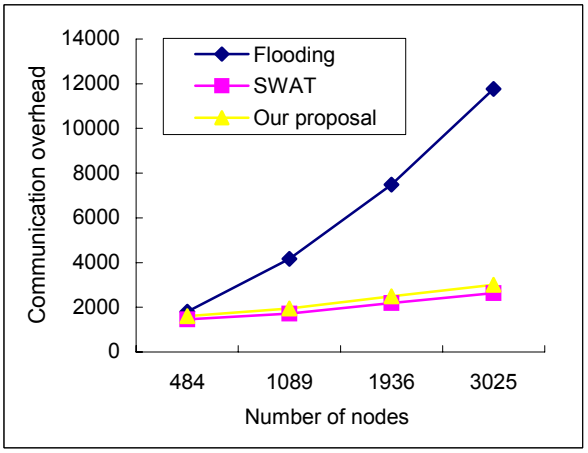


[Figure 16] False positive by background traffic (with our scheme)

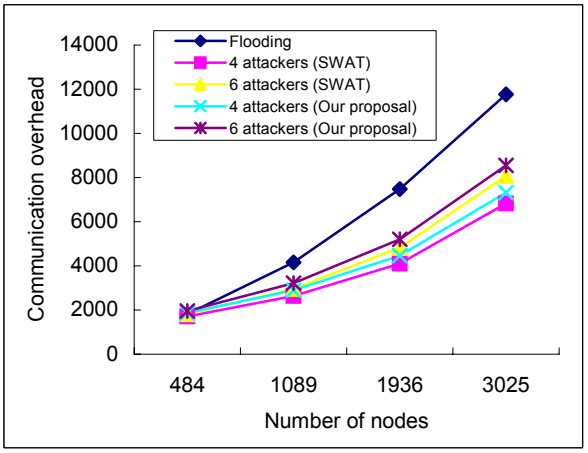
- **Abnormality Searching**

We compared communication overhead (the number of transmitted/received packets) of our protocol framework in Fig.17 and Fig.18. We varied the number of nodes 480 (area of 1680m x 1680m), 1089 (area of 2560m x 2560m), 1936 (area of 3440m x 3440m), and 3025 (area of 4320m x 4320m). A victim is located at the center of a network and an attacker is located at a random position (17 hops away in DoS and 10 hops away in DDoS) on the edge of a network. In flooding, a query message with an attack signature is flooded to the entire network. Consequently, communication overhead shows fast growth as the network size increases. Our scheme shows very low communication overhead (24% in case network size is 3025 nodes) compared to flooding, since it deploys directional search and query suppression to reduce communication overhead. Note that the energy saving becomes significant, especially when network size increases. Our scheme shows slightly higher communication overhead compared with SWAT since overhearing nodes around attack route report the candidate attack signature. However, the overhead increase is not significant. (less than 8%).

Similar to DoS case, our protocol incurs low communication overhead in DDoS attacker traceback. As the number of attackers increases, communication overhead to search distributed attackers is also increased. However, compared with flooding mechanism, our scheme incurs very low communication overhead, as shown in Fig. 18. The improvement (40% reduction in 4-attacker case) becomes significant as the network size increases. Similar to DoS case, overhead is slightly increased in our scheme compared with SWAT.

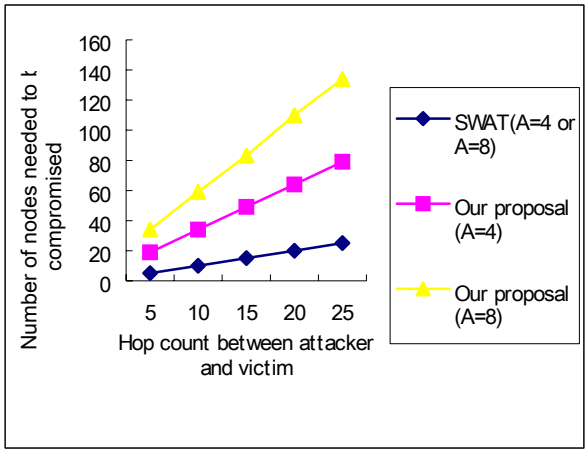


[Figure 17] Communication overhead in DoS attacker traceback comparison



[Figure 18] Communication overhead comparison in DDoS attacker traceback

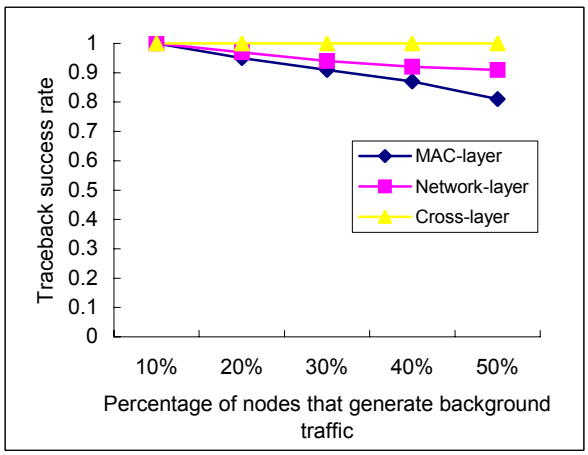
Fig.19 compares robustness against node compromise between SWAT (relay node-based scheme) and our scheme. *A* represents the number of distributed attackers. To disable traceback, an attacker needs to compromise nodes that observe abnormality and prevent them from reporting candidate attack signature. Our proposal shows much higher robustness compared with SWAT, which relies only on relay node for traceback. This is because we utilize overhearing witness node around the attack route.



[Figure 19] Robustness against node compromise

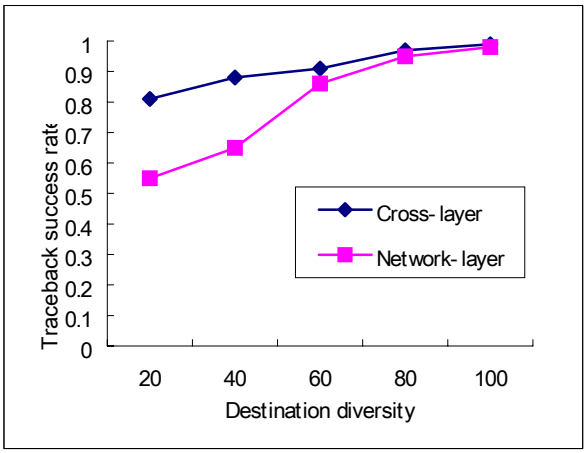
- **Overall Traceback Success Rate**

We performed a simulation and measured the overall traceback success rate with the proposed architecture. The number of nodes is set at 1089 in the network size of 2560*m* x 2560*m*. DoS attacker is performed 17 hops away from victim, and DDoS attacker is performed 10 hops away from victim. Background traffic is generated with the volume of 7.5% of attack traffic (i.e., if attack traffic=500pps, then, background traffic=(7.5*500pps)/100≈38pps) from random nodes to random destinations. Note that the background traffic is generated at the same time slots as the attack traffic. Consequently, it represents high (i.e., bursty) background traffic within short time slots. Attacker(s) and victim are randomly selected for every simulation. Fig.20 shows DoS attacker traceback success rate with MAC layer monitoring, network-layer monitoring and cross-layer monitoring. Cross-layer monitoring shows perfect traceback success even under a high volume of background traffic.

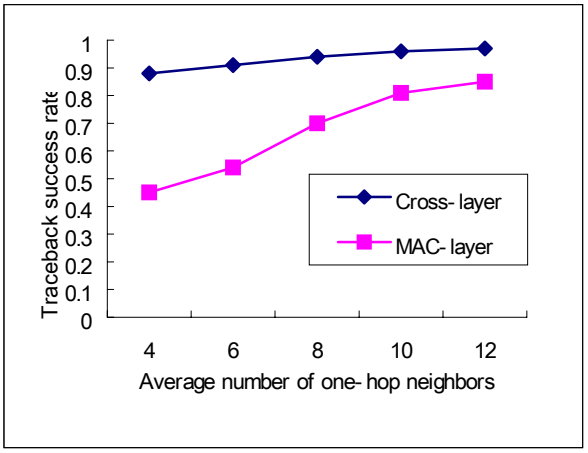


[Figure 20] Comparison of DoS attacker traceback success rate

Fig.21 shows DDoS attacker traceback success rate with various destination diversity. In this simulation, we set the number of one-hop neighbors at 6. Percentage of nodes that generate background traffic is set to 50%. When destination diversity is low (<20), traceback success rate is low with network-layer information. However, traceback with cross-layer information shows high success rate (>80%) across different diversity levels. This is because MAC layer information complements network layer information, which further reduces noise traffic. Fig. 22 shows the success rate with a various number of one-hop neighbors. Traceback with cross-layer information shows greater improvement compared with traceback with MAC layer information only.



[Figure 21] DDoS attacker traceback success rate comparison between cross-layer information and network-layer information

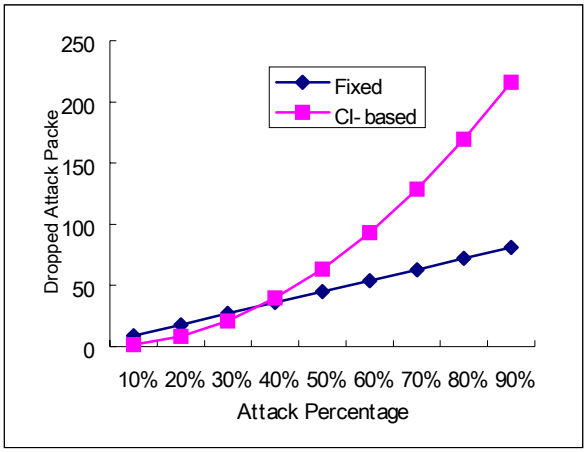


[Figure 22] DDoS attacker traceback success rate comparison between cross-layer information and MAC-layer information

- **Countermeasure**

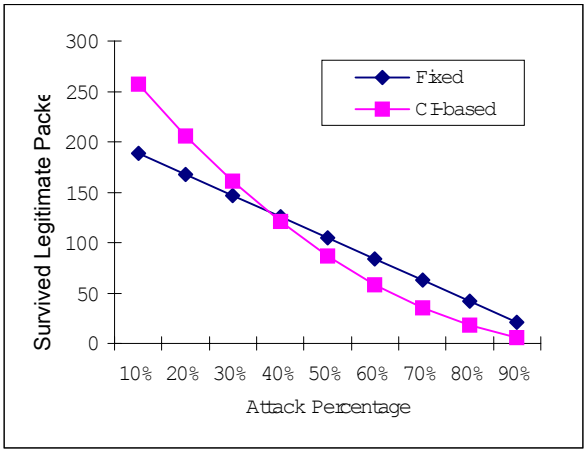
In this section, we perform analysis to verify the efficiency of our traceback-assisted countermeasure and compare it with existing countermeasure (i.e., Fixed rate limiting). We measure dropped attack packet count (Fig.23), survived legitimate packet count (Fig.24), and SDP (Fig.25). In Fig.23, attack packet dropping efficiency is increased as attack percentage is increased (Attack packet percentage represents the percentage of attack traffic in total traffic). It is because abnormality matching level is increased as attack percentage is increased. Fig.24 shows survived legitimate packet count. When attack percentage is low, more legitimate packet is survived with our scheme because matching level is low and consequently, only a small amount of packets are dropped. On the other hand, when attack percentage is high, less legitimate packets survived due to high abnormality matching level. However, the

negative impact is not significant, since there is only small amount of legitimate traffic when attack percentage is high.



[Figure 23] Attack traffic dropping efficiency

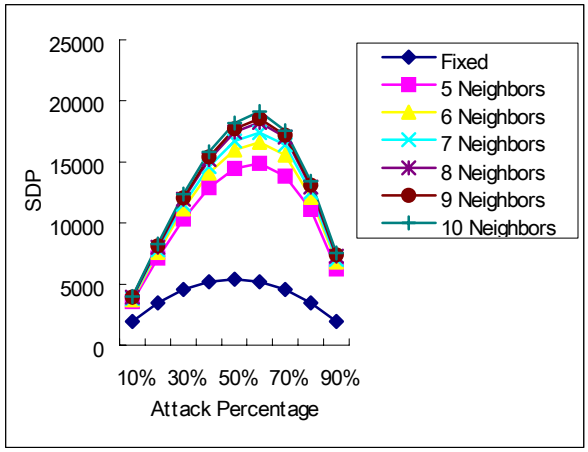
In Fig.25 we measured SDP (Eq.27) with fine-grained cross-layer information. SDP shows drastic increase when cross-layer information is considered. We compared SDP with fixed rate limiting, where no fine-grained information is considered. When the number of neighbors is increased to 10 nodes, SDP rate shows a 400% of increase. This is because we can reduce more noise traffic when there exist more one-hop neighbors.



[Figure 24] Legitimate traffic survival rate

11. CONCLUSION AND FUTURE WORK

In this paper, we proposed an efficient attacker traceback scheme geared towards wireless multi-hop networks. We paid special attention to cross-layer information (i.e., network layer and MAC layer) to increase traceback accuracy and overhearing capability of MAC layer to increases robustness against node compromise, high background traffic, mobility, and DDoS attack. In addition, we proposed a traceback-assisted countermeasure, which increases dropping efficiency against attack traffic and decreases negative impact on legitimate traffic. The efficacy of our traceback architecture is verified through extensive simulation.



[Figure 25] SDP improvement with cross-layer information

As a future work, we plan to analyze the risk and threat of mobility on traceback. Mobility of nodes can pose significant challenges on traceback, and no existing scheme considers the mobility issue in traceback. We will first systematically analyze mobility-induced risk and propose novel traceback scheme robustness under mobile scenarios.

[REFERENCES]

[1] CERT Advisory CA-97.28, IP Denial-of-Service Attacks, May 26, 1996.

[2] CERT Advisory CA-96.21, TCP SYN Flooding and IP Spoofing Attacks, Sept. 24, 1996.

[3] CERT Advisory CA-98.01, Smurf IP Denial-of-Service Attacks, Jan. 5, 1998.

[4] CERT Advisory CA-96.01, UDP Port Denial-of-Service Attack, Reb. 8, 1996.

[5] A. Belenky and Nirwan Ansari. On IP Traceback. *IEEE Communication Magazine*, July 2003.

[6] S.M.Bellovin. ICMP Traceback Messages. IETF draft 2000; http://www.research.att.com/smb/papers/draft-bellovin-itrace-00.txt.

[7] H. Burch, et al. Tracing Anonymous Packets to Their Approximate Source. In *Proceeding 2000 USENIX LISA Conf.*, pp.319-327, Dec. 2000.

[8] R.K.C.Chang. Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial. *IEEE Communication Magazine*, Oct. 2002.

[9] A.Helmy. Small World in Wireless Networks. *IEEE communication letters*, 2001.

[10] A.Helmy, et al. A Contact-based Architecture for Resource Discovery in Ad Hoc Networks. *ACM Baltzer MONET Journal*, 2004.

[11] Yongjin Kim, and A.Helmy. SWAT: Small World-based Attacker Traceback in Ad-hoc Networks. In *Proceeding of IEEE/ACM Mobiquitous*, July 2005.

[12] S.Milgram. *The small world problem*, Psychology Today 1, 61 (1967).

[13] Alex C. Snoeren, et al. Hash-Based IP Traceback, In *Proceeding of ACM SIGCOMM*, 2001.

[14] Stefan Savage, et al. Practical Network Support for IP Traceback, In *Proceeding of ACM SIGCOMM*, 2000.

[15] Microsoft Corporation. Stop 0A in tcpip.sys when receiving out of band (OOB) data, http://support.microsoft.com/support/kb/articles/Q143/4/78.asp

[16] Hogg and Tanis, *Probability and Statistical Inference*, Prentice Hall, 200