國立交通大學

資訊工程學系

碩 士 論 文

無線網路中基於動態來源繞送協定的 IP 追蹤

Wireless Traceback in Dynamic Source Routing

研 究 生：邱建熹

指導教授：謝續平 博士

中華民國九十五年六月

無線網路中基於動態來源繞送協定的 IP 追蹤

Wireless Traceback in Dynamic Source Routing

研 究 生：邱建熹　　Student: Chien-Si Chiu
指導教授：謝續平 博士　　Advisor: Dr. Shiuh-Pyng Shieh

國 立 交 通 大 學
資 訊 工 程 學 系
碩 士 論 文

A Thesis
Submitted to
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science and Information Engineering

June 2006
Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 五 年 六 月

無線網路中基於動態來源繞送協定的 IP 追蹤

研究生：邱建熹　　指導教授：謝續平 博士

國立交通大學資訊工程學系

摘要

分散式阻斷服務攻擊在網際網路上越來越盛行。同時，隨著無線網路頻寬的擴大，無線網路也成為攻擊者一個可以發動攻擊的來源。然而，相較於有線網路，在無線網路中追蹤攻擊者比在有線網路中困難，特別是在 Ad-hoc 模式下。無線網路的特性中，攻擊者可以隨意的改變他們的位置去躲避追蹤；更甚，在 Ad-hoc 模式下，只要借助中繼點的移動，就可以隱藏他們的位置。目前，現有的有線網路下的追蹤方法並不能直接套用在有移動能力和架構限制的無線網路環境。有些人試著以封包記錄的方式來解決這個問題，但在 Ad-hoc 網路中使用依靠中繼點的方式是不合理的。我們提出一個有彈性而且輕量的追蹤方式，並以封包標記的方式來針對無線隨意網路下，任何節點都可能移動的情況下，追蹤任意偽造來源位址的攻擊者。

Wireless Traceback in Dynamic Source Routing

Student: Chien-Si Chiu Advisor: Shiuh-Pyng Shieh

Department of Computer Science and Information Engineering

National Chaio Tung University

Abstract

DoS (Denial of Service) is more and more widely used in the Internet. Meanwhile, with the growth of bandwidth in Wireless Network, attackers possibly choose it to launch attacks for another choice. However, compared to wired network, tracing the attackers in wireless networks is more difficult, especially for the ad-hoc mode. With the attribute of mobility in wireless network, attackers are able to change their location to hide themselves, even more, in the ad-hoc mode, attackers can not be traced when the intermediate nodes moved. Now, conventional traceback schemes for the wired networks cannot apply directly to the wireless multi-layer network infrastructure with the mobility constraints. The packet logging method is used in an attempt to solve the problem, but in the ad-hoc network, approaches relying on intermediate hosts are not feasible. We proposed a flexible and lightweight traceback scheme by packet marking to trace the attackers that forge their IP Address in the Ad-hoc network with mobile nodes.

誌　謝

謝謝老師三年來的指導。比起其他人，我多念了一年，也多給老師帶來一些困擾。一年前，決定與無名小站的創業夥伴們暫停學業，著實是我人生的一大挑戰，如果當初沒有老師的支持，我真不確定自己是否有勇氣接受挑戰。一年後，我又回到實驗室，一樣的是熟悉的實驗室，卻大部份都是我不認識的人了。面對一批新的學弟，雖然不是很熟，也很謝謝他們沒有給我異樣的眼光，更謝謝他們在我必需兼顧學業與工作時，給我許多幫助。

另外已經任職於中原大學資訊工程系的楊明豪學長，特別要感謝他的大力指導，讓我在一個新的領域中找到題目，快速的切入，完成這份論文。而當初花不少時間跟我討論第一個題目的李富源學長，雖然沒有完成那篇論文，但從你身上學到一個研究人員的精神，令人敬佩。

也要謝謝當初實驗室的同學們，在我離開時給我一些祝福，在我面臨作論文的壓力時給我一些開導，謝謝林亞正、李卓育、陳宗逸、吳孝展這四位永遠的夥伴。

謝謝公司的幾位好朋友，當我在為論文做最後衝刺時，還不忘說些關心的話，給我精神上的安慰。

最後謝謝我的家人。感謝爸媽能理解我的所做所為，沒有另外給我更大的外在壓力。謝謝我的哥哥當我最知心的聽眾，有空就陪我聊天說說話。如果我能有比其他人多一點的成就，都要感謝身邊的人，給我勇氣完成一切。

Table of contents

List of Figures

List of Table

1 Introduction

The impact of network attacks is getting more and more significant as the Internet access becomes cheap and available everywhere. With different kind of devices and carriers, attacks, accompanied with ubiquitous Internet access, threaten the network. Attackers may try to gain access of victim hosts and compromise them as zombies. Even more, with the help of high bandwidth, attackers can easily slow down or consume resources of victim hosts, most of them are computation power, memory usage, and network bandwidth.

A kind of well known attacks, known as distributed denial-of-service (DDoS), is one that uses thousand numbers of zombies (compromised host) that in different places of the Internet to generate large traffic flows to consume the resources, such as bandwidth and CPU time, so that normal services are seriously degraded or totally denied [1]. Nowadays, many famous websites, such as Yahoo! and Amazon, are under the threat of DDoS attack, and cause thousand million dollars damage. The attack traffic could be in different type, such as SYN packet, ICMP packet ... etc and the attack traffic could be two way communicated or just sending in one way [2][3].

DDoS problem comes from the nature of IP design [4]. IP Protocol routes packets only by the destination address on IP header. Without source IP check, intended people can put any other address in the source address in IP header, if they do not care about the packet comes back or not. For the serving applications, resources are allocated for serving the connection but no future answer, because the source address of the request is not the one really request the service, then this makes the resource

consumption and slow down normal service.

Some RFCs talk about the problem, they proposed Ingress Filter [5], which checking source packets on packet forward from the edge router, since edge routers with simple routing rules, and the intended attacker could be blocked in the first moment, but the ratio of deployment rate really talks. Another RFC [6] proposed sending extra messages to the destination host, helping the destination host distinguish the source address. The packet is sent in ICMP packet, however the overhead of bandwidth and the precision and sampling rate is also challenging problem. Some advanced routers like CISCO support Unicast Reverse Path Forwarding, but the extra computation and bandwidth is needed, and the router should be equipped with full route of Internet.

The following research about IP traceback in the wired network can be mainly thought as two basic ideas, one is packet logging [7][8] and the other is packet marking[9][10][11]. The packet logging method needs extra data storage overhead and the packet marking method is information limited in 16 bit ID field, and need more packet to get precise answers. Both of them are matured research, and the deployment could depend on the environment to fit the requirement of both methods.

There are two different modes in wireless network, one is infrastructure mode and the other is Ad-hoc mode. In the infrastructure mode, the wireless nodes connect directly to the access point. In the ad-hoc mode, nodes in the same network help each other to create paths and route the packet to the correct destination. Ad-hoc network was originally made to be a self organized network.

There are some main routing protocols in the ad-hoc network. By the difference of basic routing information maintenance, they can be divided into proactive routing and reactive routing. Proactive routing protocol in ad-hoc network maintains routing table even if there are no packets sending in the network. Nowadays developed routing protocols are designed to be On-Demand, which only create routing path when need. AODV [12] (Ad-hoc On Demand Distance Vector routing protocol) and DSR [13] (Dynamic Source Routing protocol) are two famous ad-hoc routing protocols that using On-Demand method. ADOV routing protocol maintains the routing information about adjacency nodes, the intermediate node can only get local information. DSR routing protocol keeps the full routing path in the routing header, with the data in header, intermediate nodes can get the total information about the path.

IP traceback problem in the ad-hoc network will be more and more important since the growth of wireless device, and the requirement of wireless bandwidth will be larger. Attackers may generate attacks from the wireless ad-hoc network. Some research talks about it [14]. However, more attributes of ad-hoc network, such as power consumption and routing protocols, should be considered in to get resource saved and environment suitable traceback method.

We elaborate the traceback requirement under ad-hoc network in the next chapter. In chapter III, we propose a scheme for traceback in the ad-hoc network by packet marking. We evaluate the method in Section IV before concluding in Section V.

2 Related work

DSR (Dynamic Source Routing) [13] is a widely used routing protocol in Ad-hoc network. DSR protocol is composed of some parts, such as Route Discovery, Route Response, Path Salvage, Route Cache, Flow State ... etc. DSR is a routing protocol that every node should know about the full path from source to destination. Since the cost of finding all the routing path is very high, DSR is designed as on-demand. The other way to save the cost is Route Cache, which cache routes in local computer, and Flow State, which replace route to flow to save the space in the packet.

When sending one packet out, if the destination address is not listed in the routing table, the host should broadcast a Route Discovery to get how the packet should transmit, and save the routing path in the route cache. After node gets the routing path, it places the routing path in the packet header, and keeps an anchor as the current processing point. A figure tells as Figure 2-1.

Figure 2-1 DSR processing sample

Route Cache stands an important role to save the overhead to broadcast Route Discovery. However, controlling the cache is also important preventing attacker from manipulating the cache.

When a host fails to get the Route Cache from its routing table, it sends out a Route Discovery by broadcast. If the intermediate nodes know how to get to the

destination, they will reply the Route Discovery and reply a Route Cache hit rather than broadcast it to the real destination. For a node moves in a grace motion, most of the path will be the same, only some difference in the leaf node, so the Route Cache hit can save the effort about broadcast transmission. A graph explains as Figure 2-2.

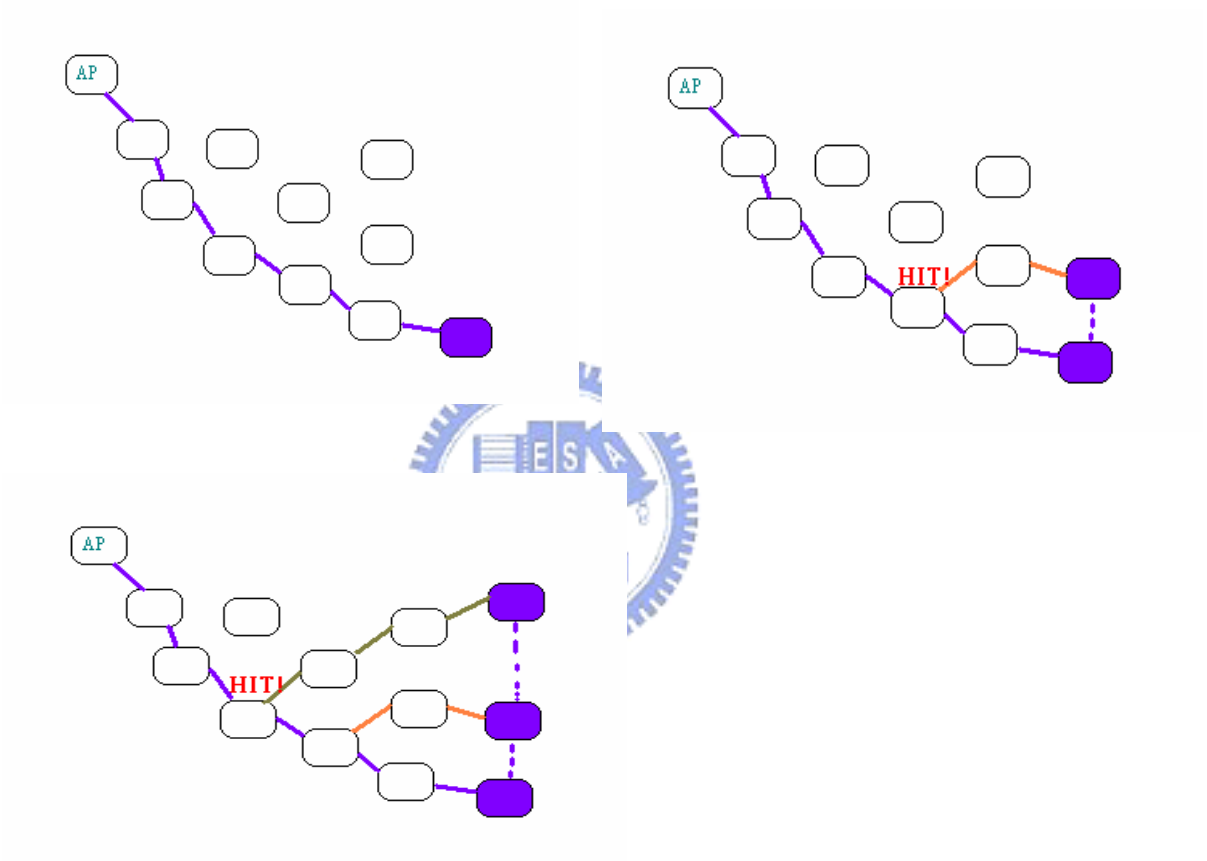


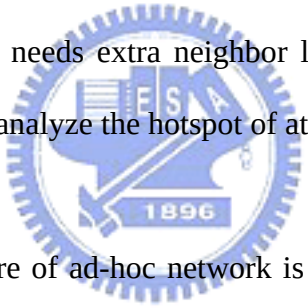
Figure 2-2 Dynamic Source Routing, Route Cache Mechanism

But the Route Cache control is so careless to the source checking. If the attacker changes source IP and sends packets out, the attacker could also take the advantage of Route Cache to hide themselves and prevent sending Route Discovery to the gateway. This kind of attack is just the same with the cause of IP Spoofing attack.

The traceback under Ad-hoc network is more and more important. Huang and

Lee proposed a hotspot traceback scheme under ad-hoc network [14]. The authors think the main difference about wired traceback and wireless traceback is the mobility. This paper tries to solve the mobility problem based on a wired traceback scheme, named SPIE [7]. The SPIE makes use of a special data structure, bloom filter [15], to release the packet logging data overhead on each supported router.

The proposed traceback scheme uses packet logging method under the wireless ad-hoc network with a special data structure named Tagged Bloom Filter (TBF). It can save storage problem just like bloom filter, and in addition, in the mobility aspect, it helps a hotspot method to distinguish where the attackers are. They save relative TTL in TBF, with the help of relative TTL, an algorithm proposed to construct the link graph. Hotspot traceback needs extra neighbor list maintenance. With the link graph and neighbor list, it can analyze the hotspot of attacker.



However, the infrastructure of ad-hoc network is not suitable for SPIE. Most of the relaying nodes in ad-hoc network are resource limited personal computer or notebook. The resources needed by SPIE are extra storage and several hash computation with packet header as input. As a power or resource limited infrastructure, this is a big challenge to choose packet logging method, such as SPIE.

We assume the attacking environment as the Figure 2-3. Attackers or zombies hide in the ad-hoc network running DSR, and they are trying to sending packets (connectionless or connection oriented) to stop the normal services of victim. The ad-hoc network gateway (access point) is able to forward packet to the Internet, and as a node with extra computation power and storage, and will not be compromised.

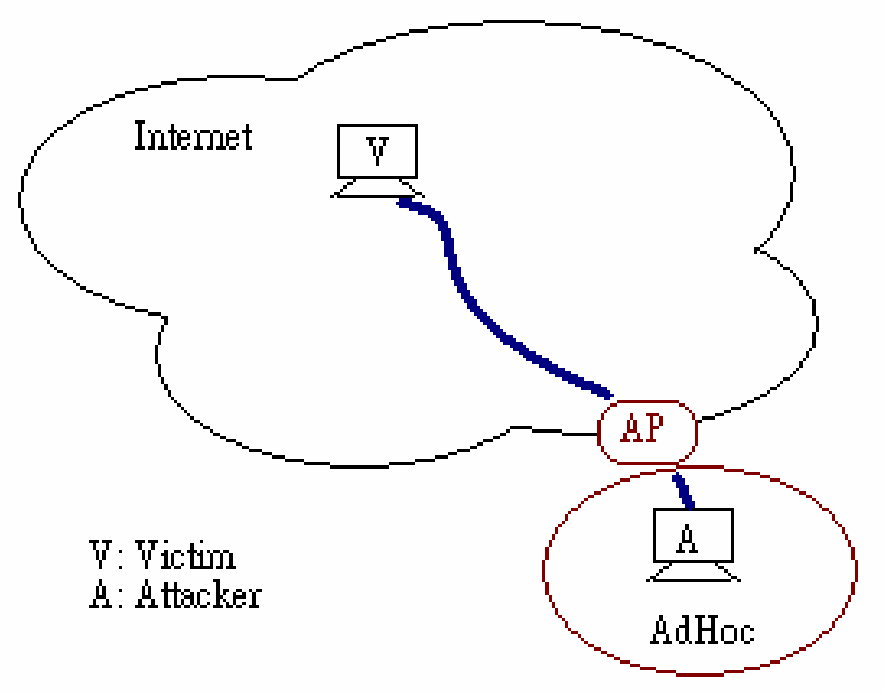


Figure 2-3 The assumed attacking environment

3 Mobile Attacker Tracing Scheme

We describe the detail of our scheme to solve the mobility problem, and how to trace the attacker that is moving and spoofing their address.

3.1 Symbol and Environment Definition

Before we talk about the scheme, we should first define the symbols and attack environment.

3.1.1 Trace attack hosts in Wired and Wireless network

The network is self-organized wireless network connected to each other in Ad-hoc mode. The network is running Dynamic Source Route protocol, and nodes forward packets for each other. In the network, there is one node with Internet access and forward packets for other Ad-hoc nodes if other nodes need to send packets to the Internet. We call the forwarding node as access point. The ability of access point is higher, in computation and memory storage, and the power problem does not affect.

The ad-hoc nodes is possible to move from one place to another place, but the moving speed is a reasonable and graceful, such a human with a notebook, walking from one place to another. The access point can not move.

In this network, there is possible for nodes to be compromised by attacker, or some nodes are attacker. The attackers generate DoS/DDoS attack the victim in the Internet. The attacker has some possible attacking methods, one is to fake one node's

IP, and the other is continuing changing spoofed IP. If the nodes are not attacker, it should run by the definition of DSR.

When DDoS attack happens, the victim in the Internet uses IDS (Intrusion Detection System) to detect some host is attacking, and the IDS is able to generate notice. The victim starts the wired traceback and find out the access point. After the access point is located, and the victim knows wireless traceback should be continued, the victim start to wireless traceback, sending attack notice to the access point.

If we can find the nodes that directly forward for the attacker, the attacker is traced.

3.1.2 Symbol definition

We give the definition about some symbols by Table 3-1, and give some simple explain about some symbols.

T_i is an integer that count for time slice, the time slice adds one when routes move. The time of T_{i+1} is later than Ti, but the interval between T_i and T_{i-1} is not the same with T_{i+1} and T_i. A graph example is as Figure 3-1. As the time ticks, the timestamp will not increment with time ticks, but increment with every path update from the received packet.

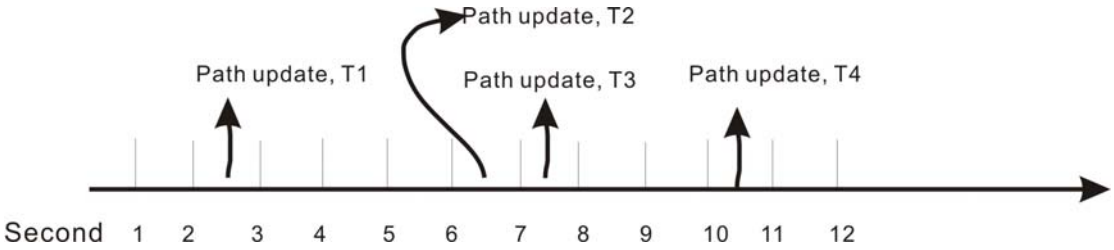


Figure 3-1 Difference between time and time slice

Table 3-1 Terminology

T_i: the i-th Timestamp, which the timestamp updates only when the Source Route of some packet changes.
N_k: stands for the k-th Node in the Ad-hoc network, with a unique identifier.
SourceRoute: composed of a list of IP address. such as $SourceRoute_i = \{N_1, N_2, N_K\}$
MD: Acronym of Marking Data, the information sent from access point to victim.
RGH: Route Graph History, a table for record SourceRoute from ad-hoc node to access point.
Access Point: A host or router that forward packet for ad-hoc nodes to the Internet.

3.1.3 Wired Traceback

Tracing in the designed attacking environment, the traceback scheme should be divided into two part as Figure 3-2, wired Internet part and wireless Ad-hoc network part. In the wired Internet part, the traceback method is already mature and many different way [7][8][9][10][11] can achieve the goal. The wired traceback schemes can trace to the last end host, in our attacking environment, a.k.a. access point (gateway). The Ad-hoc network part, there are two choices, one is combined the packet marking with the existed marking scheme, and the other is sending the tracing data in the Ad-hoc network by out-of-band strategy (such as: ICMP, UDP ... packets). We prefer to send the trace data out of band, to avoid unneeded interference to wired

traceback. Most of the wired traceback schemes use 16 bit ID field of IP packet. If the wireless traceback tries to inject more data in it, the work of wired traceback will be interfered.

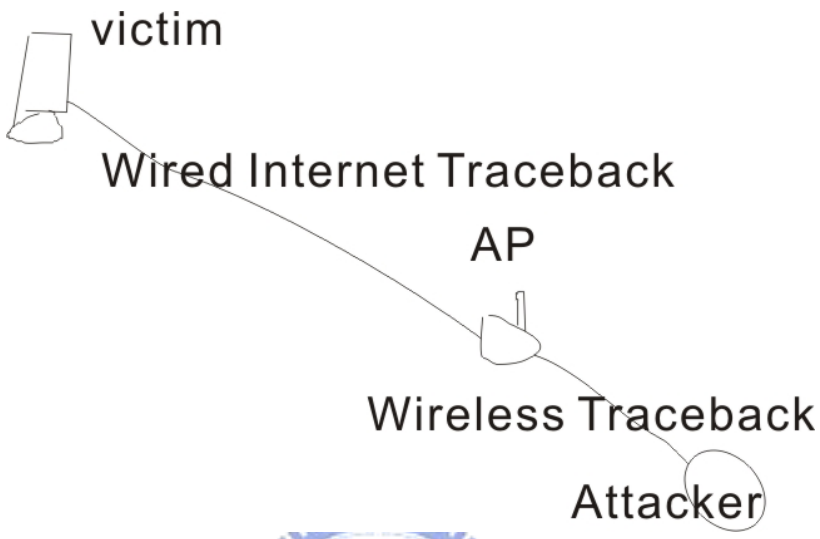


Figure 3-2 Two part of traceback

The wired traceback is trivial for nowadays ability of traceback, so we do not put much emphasis on it. Later on, we start to discuss the wireless traceback and the combination of the part of traceback.

3.2 Traceback Marking

As a gateway in Ad-hoc network, it gathers all packets forwarding to the Internet. And, by extract the source route from DSR header, gateway can get the view of nodes that sending packet to the Internet. However, the source route can not send out because the source route is relatively large compared to the original packets. Regarding to telnet protocol [16], every keyboard click might be sent out even the packet overload is only some bytes. Sending the entire source route out by out of band method is not effective at all. The information about the source should be probably

encoded to release the pressure about network bandwidth.

Marking Data (MD) is used to identify packets and timestamp relation; it should contain the timestamp of packet, and the attacker identification (Source IP Address). When attacks happen, the MD is sent back to the AP and gives the clue about the attacker. Since the MD is sent out-of-band, Marking Data should be connected to the original packet. This is easily done by simple checksum like MD5 or SHA1, and the input as packet header and some bytes of payload.

$$MD = T_i : \text{Hash(Encoded Source Route)} : \text{Digest(Digest of packet)} \quad (3.1)$$

The receiver of the out-of-band data should keep it for a small time and limited in a pre-allocated space of memory. We can reach is assumption by generate a hash table, which the space and size could be controlled by the size of hash table. The hash table with Digest of header as key saves the timestamp and encoded source route as value. The value of digest of packet should be the characteristic of the packet. Since the assumed attack is DDoS attack, so the false positive can be eliminated as more and more packets come in, so we recommend the digest of packet could use the packet header and some bytes of payload as input of a universal hash table.

Table 3-2 Destination should maintain a small table about the out-of-band traceback data.

Digest of packet	Timestamp:Encoded Source Route
.....	.....
.....	.....
.....	.....

Once the IDS system notice about the attack packet, victim can search the table by Digest of header as key, to find out the time slice and Encoded Source Route, send back an attack notice with time slice and Source Route. Timestamp helps the access point to identify the timing of attack happen, and avoid time synchronization problem. Source Route is sent in encoded format, this avoids data overhead. Source Route of source IP is also stored in the access point, sending with attack notice is for avoiding spoofed attack notice.

3.3 Mobility

Mobility of Ad-hoc network is an important feature, it allows user to move as they wish, but this also make it harder when attackers make use of this attribute. To overcome the problem of mobility, we introduce a method to save the data from forwarded packet that summarized by access point. With the help of Route Graph History, we can not only reduce the problem of mobility, but also trace the hot zone of the attacker.

3.3.1 Construct Route Graph History

Time synchronization is important in a distributed system. In attack trace, if we want to locate an attacker by time, first we need synchronize all the time in the network. However, the cost is high in a distributed system. If we can centralize the data and avoid the reliability on distributed system, we can get a synchronized time in low cost.

For getting the Ad-hoc network topology $N_1, N_2, \ldots, N_n$ in a certain period of time, a time synchronization mechanism RGH is used to main the neighborhood relationship of nodes in a certain time period. And, we will keep the information in gateway of wired and wireless networks. On every packet arrived, extract the source route on the packet and do some examination. If the source route is the same with the path of last max timestamp, mark the Path with T_{cnt} with addition to *cnt*. If the source route is different from the path last packet from the same source (a.k.a. path update), we have to increment the counter and insert the new source route to the table with new T_{cnt} marked. By the algorithm, we can maintain a time synced table about per-host RGH (Routing Graph History). The pseudo code is as Figure 3-3.

Here we define two important variables, $Path(T_i, N_k)$ and $AttackGraph_{T_i}$. $Path(T_i, N_k)$ with two input T_i and N_k, it stands for the Path (a sequence of IP address) for source node N_k to destination node access point, in the timestamp. $AttackGraph_{T_i}$ stands for in the timestamp *i*, the union of all the paths in timestamp *i*. The $AttackGraph_{T_i}$ is a directed graph, with nodes and edges.

By the algorithm RGH, we can maintain a table with time and path information

mapping. The gateway can maintain a table like Table 3-3, a relation between timestamp and route path of nodes. And, by union all the path of all nodes, we can finally get the AttackGraph by the timestamp. This method avoids the time synchronization problem and helps us extract the attacking graph from any timestamp.

Algorithm 1 algorithm for keep the ROUTE GRAPH HISTORY with time attribute.

1: **if** $Path(N_i, T_{max_i}) \neq SourceRoute_{N_i}$ **then**
2: $cnt++$
3: $Path(N_i, T_{cnt}) = SourceRoute_{N_i}$
4: **else**
5: $Path(N_i, T_{cnt}) = SourceRoute_{N_i}$
6: **end if**

Figure 3-3 Algorithm for maintain Route Graph History

Table 3-3 The table of Node, Timestamp, and Attack Graph

Time/Node	N_1	N_2	...	N_n	Union
T_1	$Path(T_1,N_1)$	$Path(T_1,N_2)$	...	$Path(T_1,N_n)$	$AttackGraph_{T_1}$
T_2	$Path(T_2,N_2)$	$Path(T_2,N_2)$	...	$Path(T_2,N_n)$	$AttackGraph_{T_2}$
...					
T_k	$Path(T_k,N_3)$	$Path(T_k,N_2)$	...	$Path(T_k,N_n)$	$AttackGraph_{T_k}$

From the RGH table is initialed in the beginning, there are more updates path rather than maintain the counter to T_{max} since the table is empty. There will be only few paths in the same timestamp, that is to say, many $Path(Ti,Nk)$ is null in the table. Figure 3-4 gives an empty RGH example. If the table contains many un-filled buckets in the table, this will be hard for us to extract the AttackGraph. But with the system goes stable (the nodes in the network appear to the gateway), empty buckets will be

less.

	N_1	N_2	N_3	N_4
T1	Route			
T2		Route	Route	
T3		Route		
T4			Route	
T5				Route

Figure 3-4 Empty Route Graph History

The other possible for update storm is when some nodes are moving drastically. In the assumed environment, this might be the attacker, either moving drastically or spoofing the source route.

In the case of many un-filled buckets, if we want to extract the Attack Graph but many *Path*(*Ti*,*Nk*) is null. We have to find the last one or two or more counter to get the possible path. But in a stable system, if the *Path*(*Ti*,*Nk*) is null for a long time, the node might be gone in the network.

By the RGH data structure, the access point can record the changing of nodes in the network. With the information, this really helps the access point to get more control about the network, and the information might be useful when the attack occurs and traceback is needed.

3.4 Traceback

With the help of above data structures and prepared data, we can start to discuss the most important part of our scheme, traceback. Packet marking can notice the access point start to trace, and RGH helps us monitoring the network. In this section, we will discuss how to trace and get the hot zone of the attacker, even if the attacker is trying to forge their IP, we also propose some method to identify they are the same attacker.

3.4.1 Tracking Attacker Movement

When we get notices about attack flow paths, we should try to identify same attacker from different source IP if there is any clue for us to tell.

All the procedures start with a RGH, a data structure collecting path/time/node association as Figure 3-3. In the normal time, the access point collects the source of all Ad-hoc nodes, and maintains the route graph with one timestamp. As every path updated, access point should update the changed path into the graph. In the normal time, the access point storage overhead is N source routes and some computation operation to extract the source route, compute MD and send.

When an access point receives a notice attack with Path(N_k, T_a), get the last IP from source route $SourceRoute_k$. If the route of Ta in the access point is not the same with the source route of attack notice, this notice will be discarded. There are still possible the access point receive the attack notice that N_k does not existed in the network, because the wired traceback is possible to be spoofed in the last (final) hop. If the N_k is not existed, just ignore it.

We divide the attackers into two kinds, one is generating DDoS attack packet,

but they do not spoof their IP addresses. For non-spoof attackers, in the assumption, if the attacker is moving and sending attack packet, they should discovery route before they send. Since the non-attack nodes will follow the DSR protocol, so we can trace the attacker just by the source route from RGH.

For spoofing attackers, after the first attack notice successfully comes to access point and validate the notice, the RGH should start to log every path change with timestamp. After on, we keep an algorithm to tell and merge attack notice as Figure 3-5.

Algorithm 2 algorithm for merging spoofing source attacker.

```
1: if ∃Path(N_k, T_a) ∈ RGH then
2:   if not N_k ∈ AttackerList then
3:     add N_k to AttackerList
4:     for ∀N_x ∈ AttackerList do
5:       if Path(N_k, T_a) and from Path(N_x, T_{a-k}) to Path(N_x, T_a) is mergeable then
6:         mark N_k = N_x
7:       else
8:         AttackNumber ++
9:       end if
10:    end for
11:  end if
12:  if N_k ∈ AttackerList then
13:    keep logging
14:  end if
15: end if
```

Figure 3-5 Algorithm for merging spoofing source attacker

As Figure 3-5, on every attack notice, we try to exam if the node is already

marked as an attacker. If he is not ever marked as attacker, one possibility is that it is a new attacker for the environment, and the other possibility is that the node is an attacker try to use another IP to attack.

In an interval, we try to back trace the existed attacker when a new attacker comes. If the two paths are able to merge, we claim they are the same node. If the two paths are not able to merge, we say that there is one more attacker in the network. The interval is left for attackers stopped for a while and normal nodes update the paths.

We define the two path is able to merge when we compare the two paths, the paths have same node in sequence.

A random spoofing tool will generate attacking packets with random source IP, and change it from time to time. In this case, the source IP will be in a wide range of diversity, but the route paths will be almost the same. Except the source IP, we can conclude the other information to merge them. The algorithm 2 describes it. By comparing all the paths with timestamp marked T_{max}, the complexity is limited to $O(n)$. If the diversity of one node is more than an unreasonable number, such as more than the number of nodes before attacks occur, then we can claim we successfully trace one of the attackers. A graph sample is like Figure 3-6.

Algorithm 3 algorithm for merging spoofing source attacker.

Given N_s by victim v is suspect

initial $AttackGraph_v = Path_(T_{max}, N_k)$

for k from 1 to n **do**

  $P_k = Path(T_{max}, N_k)$

  $Diff_{k\_s} = Path_s - (Path_k \cap Path_s)$

  if $Diff_{k\_s} = 1$ **then**

    $AttackGraph_v = AttackGraph_v \cup Diff_{k\_s}$

  end if

end for

Figure 3-6 Algorithm for merging random spoofing source attacker

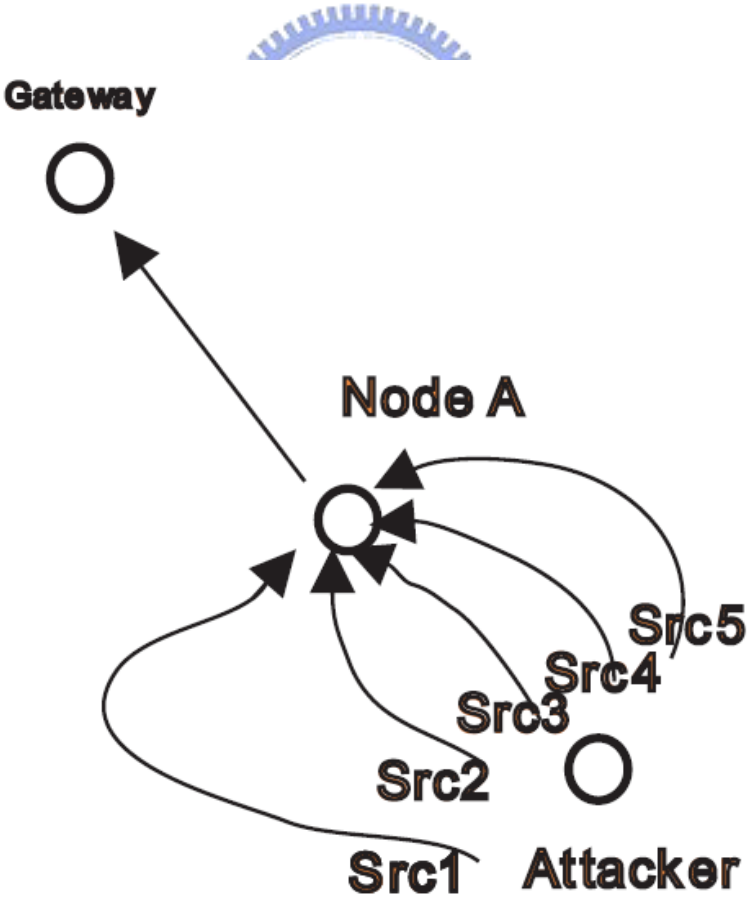


Figure 3-7 Spoofing IP Address Attacker

3.4.2 The movement of intermediate node

For minimize the effort to re-compute every route topology on the access point every timestamp, we propose a simple intermediate node movement model.

We define some possible intermediate node movements, which intermediate node defined as nodes that exclude AP and leaf node. The movement of leaf node is the target we want to trace, so we exclude it, but observe it. The movement of intermediate nodes is normal and there will be some clue to detect, so we can follow some rule to accelerate the AttackGraph re-construction.

The concept about the algorithm is that DSR guarantee loop free. So, if one node moves, and the graph loops as Figure 3-8, we can say that one path should be removed for the current timestamp (DSR also guarantee reliability, and route cache keeps two or more paths, but will not use them in the same time). Since the access point keeps updating route path of nodes by forwarded packets, it can not know about the movement of nodes.

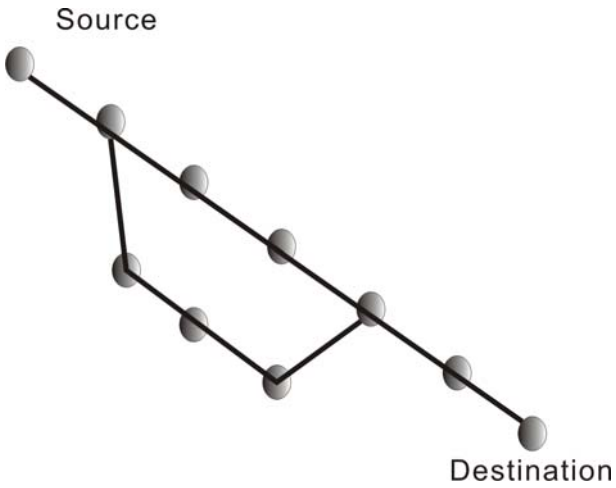
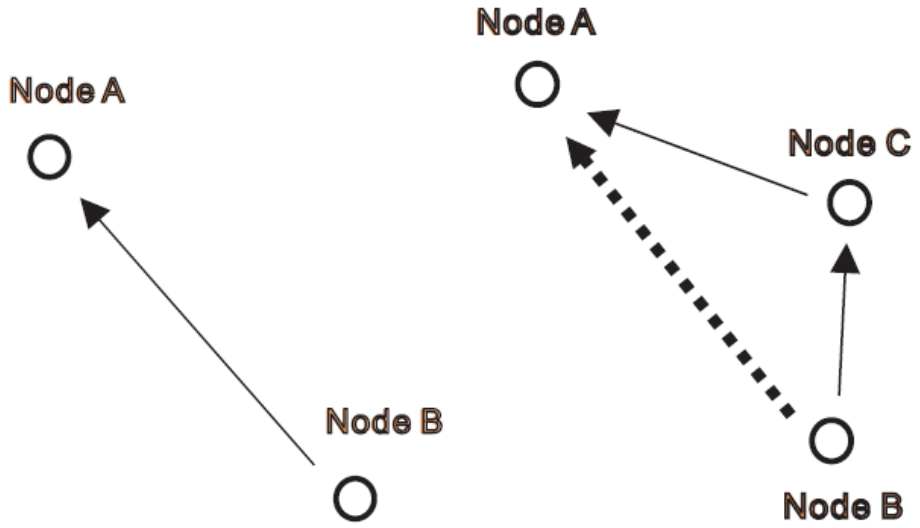


Figure 3-8 Path Move

We propose a reverse way to find out the movement by receiving the packets. Since most of the movements should be built by the combination of basic movement, if the basic movement failed to build the current attack graph, we reconstruct the attack graph from all the current paths. In this situation, either nodes move dratistically or some route path is spoofed.

This method saves some computation to construct attack graph from each paths. Choosing a suitable examination degree can reach a balance to computation and correctness. We define an examination degree to make this method flexible to the network degree. Examination degree means in two different paths, the number of different nodes after count the difference of the two paths.

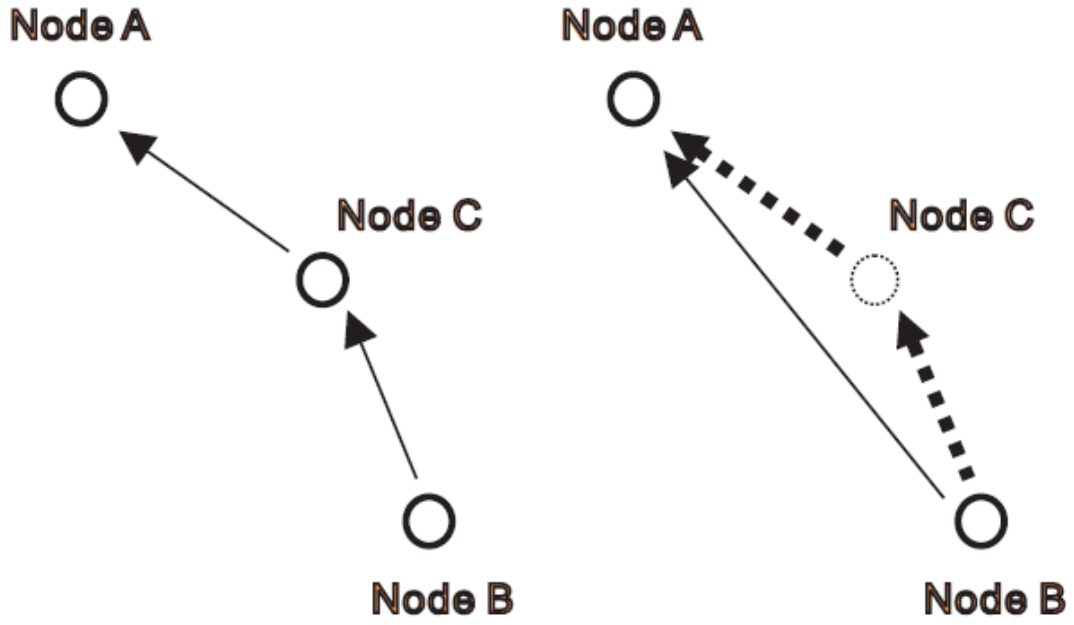
We give two example of examination degree = 1 as Figure 3-9 and Figure 3-10, and give two example of examination degree = 2 as Figure 3-11 and Figure 3-12. In the cases, we can easily find the loop occurred by a DFS (Depth First Search) algorithm, and as described above, we can balance the computation and correctness by controlling the examination degree.



(a) Forwarding Path from Node A to Node B

(b) Node C forward for Node A and Node B

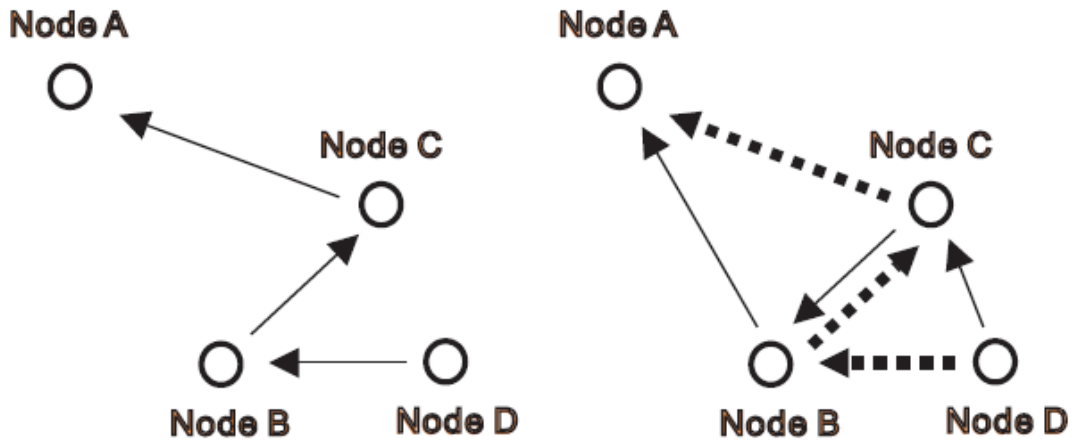
Figure 3-9 Basic Movement I, node join, two path existed in the same time



(a) Node C forward for Node A and Node B

(b) Node C disappear, Node B connect direct to Node A

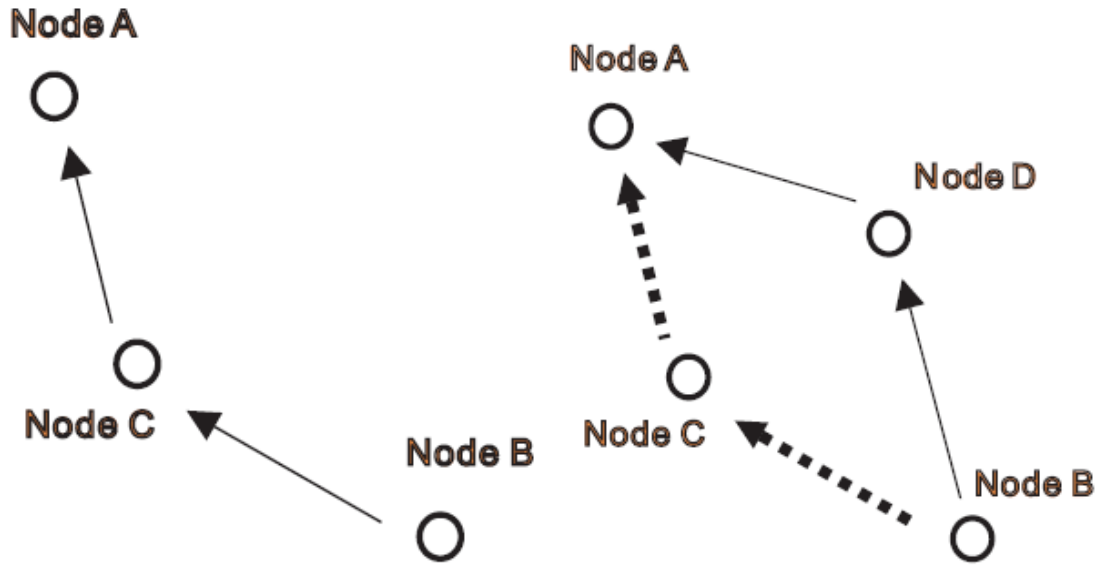
Figure 3-10 Basic Movement II, node leave, two path existed in the same time



(a) Node C forward for Node A and Node B

(b) Node C disappear, Node B connect direct to Node A

Figure 3-11 Basic Movement III, node flip flop, a loop



(a) Node C forward packet for Node A to Node B

(b) Node D replace Node C, forward packet for Node A to Node B

Figure 3-12 Basic Movement IV, node replace, a loop

3.5 Normadic Support

In the real world wireless network, the ad-hoc node is possible to roaming from one access point to another. In the proposed Mobile Attacker Tracing Scheme, we focus on single access point traceback. And in fact, the design of RGH is able to extend to *N* access point.

When the attacker is roaming in two access point, the access point should change some record with each other to maintain the correctness of packet transmission, like registration information in Mobile IP. That is to say the access point can know about the address about the roaming nodes. When the attacker tries to generate attack packet from one access point to another, the RGH of the two access point should be combined to trace the attacker.

For the case attacker roams from AP1 to AP2, we can get two RGH RGH_{AP1} and RGH_{AP2}. The most important problem of distributed RGH is that time between the RGHs is not synchronized. That is to say, if the time of two tables can not be synchronized, the only way to merge two attackers is scan all the tables for known attackers and timeslices.

So we proposed, on the exchange of two RGH_{AP1} and $RGH_{AP2,}$ the two AP should synchronize the time of two RGH by match the T_{max} of two table. By periodically exchange the RGH between RGH_{AP1} and RGH_{AP2}, we can create a mapping of T_{max} of AP1 and T_{max} of AP2. By this way, we can limit the table scan size and solve the time synchronization problem between two RGHs. As Figure 3-13, we can tell how the

two RGH keep their own counter and synchronize the time between them.

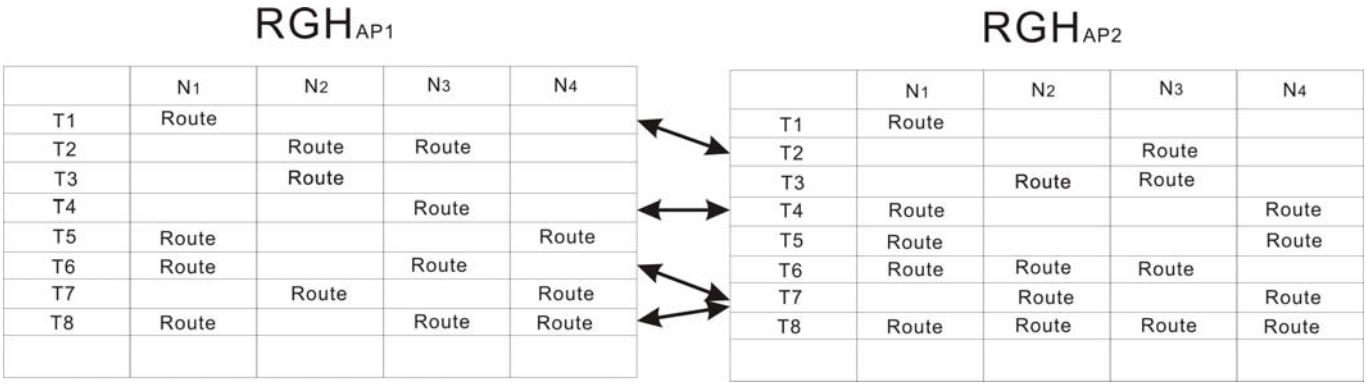


Figure 3-13 Two Route Graph History and time synchronization

On the trace of attacker, we can follow original method, but the scanning range should extend from table to two tables, and the time range should be controlled to the mapping of timeslice. In the example of Figure 3-13, if RGH_{AP1} get an attack notice on T_5 and decide to back trace to T_3, it should back trace to the last synchronized time, which is T_1 on RGH_{AP1}, and T_2 on RGH_{AP2}, and forward trace to the latest synchronized time, T_6 on RGH_{AP1}, which is T_7 on RGH_{AP2}.

We show that how to extend one RGH to two RGH, and solve the roaming problem in the real network. It is able to derive to 3, 4… to *N* distributed RGH in different APs. The mainly difference is the table scan size and the mapping of timeslice of all existed RGHs.

3.6 Analysis

In the proposed Mobile Attacker Tracing Scheme, we find some phenomenon and try to discuss it.

3.6.1 Un-successful Traceback Analysis

In most of the normal cases, the mobile attacker tracing scheme is able to locate the attacker. We found that there are some extreme cases, if the attackers are able to know the topology or know about the neighbor is going to move, the attacker is able to change the IP address in time and avoid to be traced.

In the following example in T_i, the attacker is using *Address* to generate attack packets, and before the victim trace it and the topology changed, the attacker change its address to *Address'*. If we define the Moving Distance Factor lower than the topology changed, the attacker successfully hides from the trace from T_i to T_j, and the attack graph should be reconstruct. The Figure 3-14 illustrates the case.

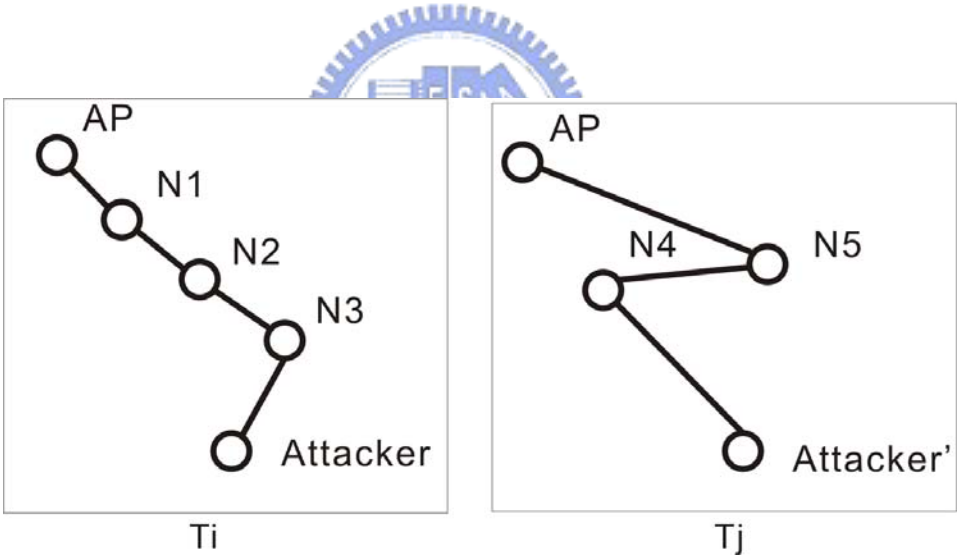


Figure 3-14 Attackers escape from trace

For the requirement to make an un-successful trace, there are two factors, MDF and route update on intermediate path change. The probability for attack to escape from trace is P(intermediate node change > MDF)*P(attacker change address). If the topology does not change, or if the topology changes less the defined MDF, attacker can not escape.

In the other way, if the attacker can predict the node moving by MAC address or other physical method, like peeping, it not the scope we talk about.

4 Simulation

All the simulation is done base on ns2 [17] simulator. Simulation setup: building ns2 and configure the wireless environment, such as channel, propagation, link layer type, interface queue type, Media Access Control, and Antenna, by the sample of ns2. We configure some special configuration on DSR agent to fit the simulation. DSR are configured with flow-state, and tune up the send buffer to reach the attacker ability.

In the simulated environment, AP is located at (300, 300) with the ability to access Internet and forwarding packets for the network. The simulated area is 600m in X-axis and 600m in Y-axis. We generate (20, 30, 50) nodes with steady-movement by ns2 indep-utils cmu-scen-gen setdest and CMUTrace to log DSR packet. Every node is trying to send UDP packet (just for testing based on IP) to AP.

Attackers in the network are able to forge IP and are able to moving in higher but reasonable speed. Attackers are trying to sending attack packets to the Internet hosts, by the forwarding of Access Point.

4.1 Simulation on Intermediate Movement Elimination

As the normal condition, every non-malicious node sends packet to the Internet and move. The access point is monitoring the Ad-hoc network and collecting Routing Information. Once the route is changed, access point will update the Route Graph History and increment counter. We want to prove that most of the movement is basic and can be eliminated by simple DFS algorithm.

After running in ns2 with the configuration above, we summarize the average path length and test if the route change can be eliminate with X = 2. Every configuration with 20, 30, 50 nodes will be run for 10 different rounds with different moving, which is generated by cmu-scen-gen setdest, and count for an average.

The figures say the simulation results. Most of the route path change can be eliminate by the proposed algorithm, the elimination ratio is from 75% to 85% when the examination degree is 4. The ratio is over 90 % when the examination degree is set to 5.

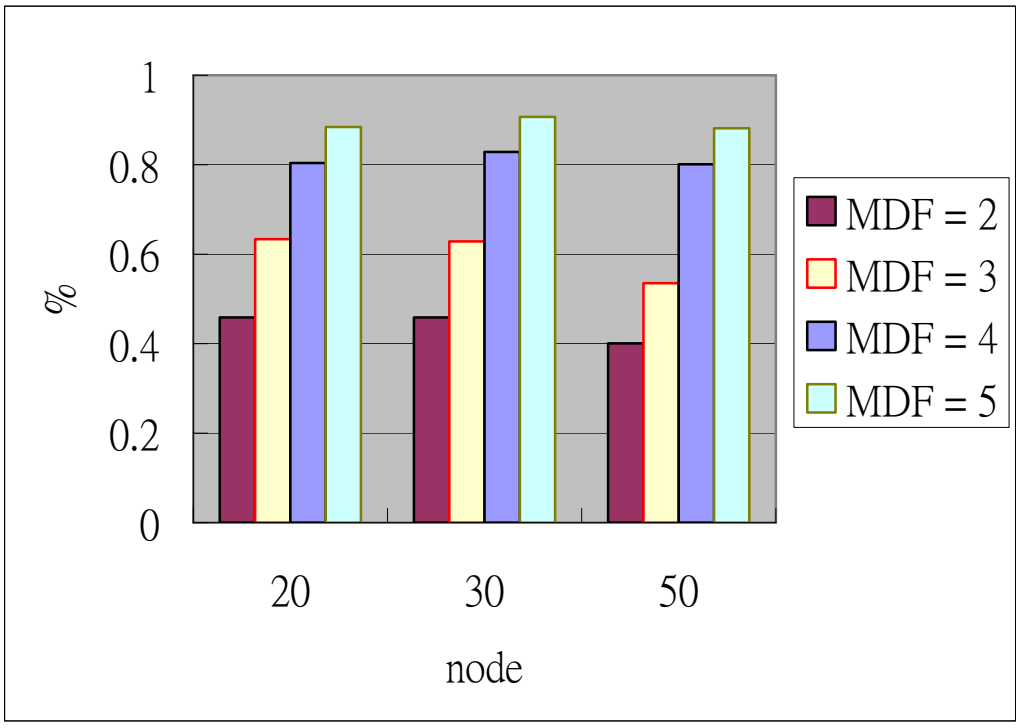


Figure 4-1 Results of basic movement elimination

4.2 Simulation on Spoofing Source Attacker

In the simulation, we assume the attacker is able to spoof their source IP Address, and with the proposed scheme, we try to find out the ratio if we can distinguish the

changing IP Attacker.

We first try to ask attacker to random change IP address and set MDF to 4, to simulate the script kid's activities, test if we can find the attacker is the same one, and this is trivial in our scheme as Figure 4-2.

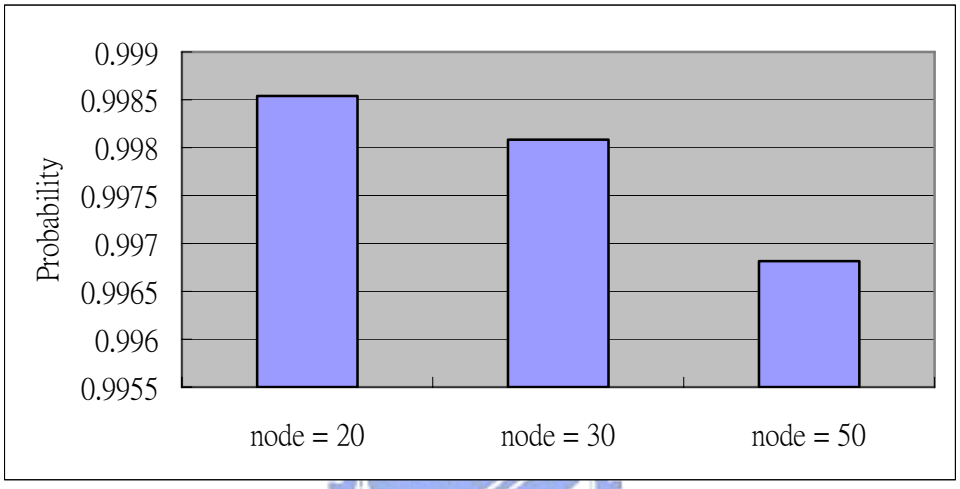


Figure 4-2 Simulation on fully random spoof IP Address

In this simulation, we run 10 different moving scenarios, and count for the probability for IP change and merge ratio. We find that most of the spoofing IP can be find.

In the next, we try to ask attacker spoof address in a selected probability and in a selected interval, after changing IP address, we want to check if the merge will tell us wrong number of attackers.

In the Figure 4-3, we found the probability of a selected ratio to spoof the address does not really help the attacker to escape from our trace. We make the spoof probability from 0.2 to 0.8, and the most of the case we can find out they are the same

attacker.

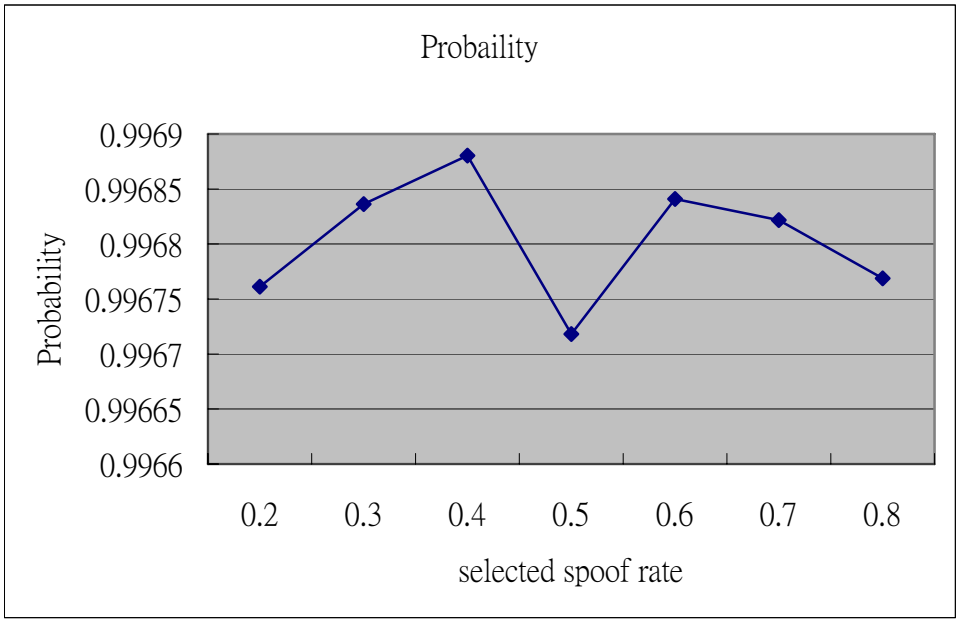
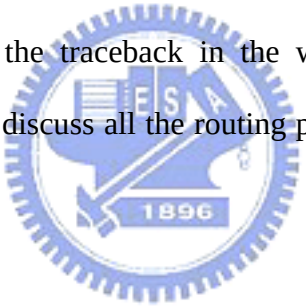


Figure 4-3 Simulation on different moving scenarios and different probability of spoof IP Address

5 Conclusion

In this paper, we introduce a traceback scheme in the wireless Ad-hoc network running Dynamic Source Routing. Compared to existed hotspot traceback in Ad-hoc network, our scheme can afford nodes lost, no extra storage overhead on Ad-hoc nodes, considering nodes moving and emphasize more on the environment of Ad-hoc network. In the simulation result, we achieve the goal to trace the attacker, and more, we leave flexible options to be fine tuned, to reach a balanced scale for exactness and computation.

The future work should can extend the existed Ad-hoc routing protocol, like AODV, DSDV… etc. Since the traceback in the wired and wireless network is important, so we hope we can discuss all the routing protocols in Ad-hoc network, to solve the problem.



References

[1] Vern Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," ACM Comp. Comm. Review, vol. 31, no. 3, 2001.

[2] David Moore, GeoRrey M. Voelker, and Stefan Savage, "Inferring internet Denial-of-Service activity," in In proceedings of the 2001 USENIX Security Symposium, 2001, pp. 9-22.

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

[4] J. Postel, "Ineternet Protocol," RFC 791, IETF, Spe 1981.

[5] Paul Ferguson and Daniel Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," RFC 2827, IETF, May 2000.

[6] Steven M. Bellovin, "Icmp traceback messages," IETF Draft, 2000, http://www.research.att.com/smb/papers/draft-bellovin-itrace-00.txt.

[7] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayers, "Hash-Based IP Traceback," in ACM SIGCOMM '01, August 2001.

[8] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayers, "Single-Packet IP Traceback," IEEE/ACM Transactions on Networking, vol. 10, no. 6, pp. 721-734, December 2002.

[9] Stefan Savage, DavidWetherall, Anna Karlin, and Tom Anderson, "Network Support for IP Traceback," IEEE/ACM Transactions on Networking, vol. 9, no. 3, pp. 226-237, June 2001.

[10] Hal Burch and Bill Cheswick, "Tracing anonymous packets to their approximate

source," In Proc. USENIX LISA '00, December 2000.

[11] D. X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback Messages," in Proc. IEEE Infocom '01, April 2001.

[12] C. Perkins, "Ad hoc on demand distance vector (aodv) routing," 1997.

[13] David B. Johnson, David A. Maltz, and Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," IETF Draft, 2004, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt.

[14] Yi an Huang and Wenke Lee, "Hotspot-based traceback for mobile ad hoc networks," in WiSe '05: Proceedings of the 4th ACM workshop on Wireless security, New York, NY, USA, 2005, pp. 43-54, ACM Press.

[15] Burton H. Bloom, "Space/Time Trade-ORs in Hash Coding with Allowable Errors," Communication of ACM, vol. 13, no. 7, pp. 422-426, July 1970.

[16] J. Reynolds J. Postel, "TELNET PROTOCOL SPECIFICATION," RFC 854, IETF, May 1983.

[17] Steven McCanne and Sally Floyd, "ns Network Simulator," http://www.isi.edu/nsnam/ns/.