http://www.eng.ukm.my

Address Resolution Protocol Optimization

J.L.Tai, Nor Adnan Yahaya dan K. Daniel Wong

Malaysia University of Science and Technology
GL33, Ground Floor, Block C, Kelana Square,
17 Jalan SS 7/26, Kelana Jaya,
47301 Petaling Jaya, Selangor,
Malaysia
E-mail: dwong@must.edu.my

Received Date: 9th November 2006 Accepted Date: 25th May 2009

ABSTRACT

This paper proposes an improved Address Resolution Protocol (ARP) for Ethernet-based networks. In the proposed alternative method, the ARP request packets are not broadcasted but instead unicasted to an ARP server which will have all the <ip, MAC> mappings of all the hosts connected to the network. This significantly reduces ARP signaling and processing overhead. The ARP server obtains the mappings through a novel passive method that does not introduce additional overhead in the network. Furthermore, the use of the ARP server makes it much easier to secure the network against certain attacks like ARP poisoning.

Keywords: ARP, protocol optimization, ARP poisoning, broadcast, unicast.

ABSTRAK

Kertas kerja ini bertujuan memperbaiki 'Address resolution Protocol' (ARP) untuk rangkaian ethernet-based. Untuk mencapai matlamat ini kaedah alternatif telah digunakan iaitu permintaan paket ARP (ARP request packet) tidak lagi menggunakan broadcast tetapi unicast kepada ARP server di mana segala pemetaan <ip, mac> untuk semua host yang berada di dalam rangkaian. Ini akan mengurangkan isyarat ARP dan pemproses overhead dengan berkesan. Server Arp akan mengekalkan pemetaan melalui kaedah novel pasif yang tidak akan memperkenalkan penambahan overhead di dalam rangkaian. Tambahan lagi, penggunaan ARP server akan memudahkan cara untuk melindungi rangkaian daripada serangan seperti ARP poisoning.

Kata kunci: ARP, protocol optimization, ARP poisoning, broadcast, unicast.

INTRODUCTION

Traditionally, when a host S running Ethernet protocol needs to communicate with a target host T on the same local area network, the network layer will pass the 32-bit IP address to the data layer and in order to send the packets to the destination host, the data layer will need to convert the IP address to the 40-bit MAC address of the host T thus the packets could be directly send over to host T through the physical media layer.

In current protocol design, each host will have a specially allocated space in memory known as an ARP cache where all hosts communicated by this host before will have their <ip, MAC> kept in the host's ARP cache memory. During the process of converting the IP address to the MAC address, if the host S does not have the <ip, MAC> in the cache mapping entry, it will send an ARP request and broadcast it to every host in the network hoping the target host T will generate an ARP reply packet with the MAC address of host T. Thus, the ARP cache reduces unnecessary ARP request broadcast messages as it is assumed that once the host S communicates with another host T, the chance for this host S to communicate further with the same host T again is high.

Every time an ARP request broadcast message is sent, all hosts including host T will receive the ARP request and each host must then process the ARP request and determine if the ARP request contains an IP address of the current host. If any host detects that the ARP request packet contains its IP address (there should be only one such host at any time as the IP address is to be unique in one network), it will then generate an ARP reply message and sends the reply directly to host S. The mapping will be kept for a short period and after some time where there are no communications between host S and host T, the mapping of <ip, MAC> for host T will be purged from the cache. This is to reduce unnecessary mappings in the ARP cache memory in view of the processing time and memory wastage if mappings are not removed.

This paper shows that it is possible to change the behavior of ARP broadcast messages to unicast messages so that the disadvantages of broadcasting ARP messages could be avoided. It will also provide evidence that this new approach is actually workable by providing the actual implementation results which had been tested in the information technology laboratory in the campus.

RELATED WORK

The Secure Address Resolution Protocol (Gouda & Huang, 2003) (S-ARP) has been designed to secure networks that use ARP against attacks like ARP poisoning. ARP poisoning is where an attacker corrupts ARP caches of hosts in a network by sending faked ARP replies. By using ARP poisoning, an attacker may, for example, cause all IP traffic meant for a victim host to be redirected to an arbitrary host that the attacker chooses. Thus, the attacker can arrange denial-of-service attacks, man-in-the-middle attacks and so on.

S-ARP uses the so called invite-accept approach to collect the ARP mappings from all hosts at an ARP server. The ARP server will broadcast an authenticated ARP request. The host whose IP address is in the ARP request can authenticate the packet before sending an authenticated ARP. The ARP server can then ensure that the message is not tampered with before updating the <ip,MAC> mapping in its the ARP table.

However, we desire a method in which the ARP server does not have to generate additional overhead (as with S-ARP) to collect the ARP mappings. In this paper, a new method will be presented that does not even generate any ARP request message in order to build up the ARP table as it simply passively obtains the IP address and MAC address mapping at the same time the IP address is assigned to the host by a DHCP server. This reduces a lot of unnecessary ARP requests during the setting up of the ARP table as well during when the host is trying to resolve the <ip,MAC> mapping. The new, improved method will be explained further in Section 3.

PROBLEM STATEMENT

Efficiency Considerations

Broadcasting behavior of ARP messages has creating some disadvantages in traditional network. First, it generates unnecessary ARP request messages that has been send to every host in the network each time one of the host in the network is trying to resolve the <ip,MAC> mapping that is not found in the ARP cache entry of the sending host. One such message is multiplied by the number of hosts connected to the network. Assuming the network has 100 hosts connected to each other, every time one of the 100 host, host S needs to send a ARP broadcast message, instead of sending the message to one host (the target host T), the ARP message is multiplied by 100 and there will be 100 ARP request messages received and processed by nodes in the network where theoretically only one message need to be received and processed. Assuming there will be one such request for each host in every 10 seconds, there will be 6000 ((60/10 packets for each host* 100 host)*100 host) such messages in one minute and only 600 such messages are actually required if ARP works in unicast mode.

Security Considerations

Another problem occurs when a host S sends an ARP broadcast messages to every host during an ARP request. Since every host in the network will receive the broadcasted messages, the identity of the host S (with IP address and MAC address) are revealed to every host in the network. In any case when one of the hosts currently connected in the network is attacked by a hacker, the hacker can use this host as a resource base to quietly collect all <ip,MAC> mappings of all hosts in the network and this allows the hacker to launch a new attack after it has collected and analyzed the information.

For example, an attacker could engage in ARP Poisoning, i.e., by providing a false MAC address into the cache, one could easily re-direct all packets that are supposed to be sent to a host located elsewhere to the hacker's host so that the latter can sniff the packets. By ARP Poisoning both hosts which are communicating with each other, the hacker can actually sniff all packets passing through both hosts by re-directing the packets towards the designated destination after receiving the packets from the victim host. This is a type of MITM (Man in the middle) attack.

All the hacker needs to do is to simply send a forge ARP reply broadcast message. In other words, when the sending host request the MAC address for a given IP addresses, the reply does not come from the host holding that IP address,; instead, the hacker replies to the request before the actual host holding that IP address replies. What if the host holding the IP address also replies? Whoever wins will get his MAC address updated in the cache. Moreover, even if the victim did not send an ARP Request, and already knows the IP to MAC mapping of the destination, an attacker can still launch an ARP poisoning attack. What the hacker does is simply send a forge ARP reply and the host will happily update that IP to MAC mapping to its cache, even thought it never requested for the MAC address. This is possible because ARP is stateless.

ARP Packet Format

To communicate mappings from <protocol, address> (<ip,MAC>) pairs to 48-bit Ethernet addresses, a packet format that embodies the Address Resolution protocol is needed. The format of the packet follows (Plummer, 1982).

```
Ethernet transmission layer (not necessarily accessible to the user):
      48.bit: Ethernet address of destination
      48.bit: Ethernet address of sender
      16.bit: Protocol type = ether_type$ADDRESS_RESOLUTION
Ethernet packet data:
      16.bit: (ar$hrd) Hardware address space (e.g., Ethernet)
      16.bit: (ar$pro) Protocol address space.
        8.bit: (ar$hln) byte length of each hardware address
        8.bit: (ar$pln) byte length of each protocol address
      16.bit: (ar$op) opcode (ares_op$REQUEST | ares_op$REPLY)
nbytes: (ar$sha) Hardware address of forget of packet, n from ar$hln.
   mbytes: (ar$spa) Protocol address of sender of packet, m from ar$pln.
    nbytes: (ar$tha) Hardware address of target of packet (if known).
   mbytes: (ar$tpa) Protocol address of target
```

Implementation Considerations

ARP request messages are broadcasted as the host S requesting the MAC address of host T does not know the MAC address of the host T, if it know, it can simply sends a ARP request direct to the host T without sending a broadcast message. Since host S only know the IP address of host T, it is thus inserted this IP address into the ARP request (ar$tpa), it will also insert its own IP address (ar$spa) and its MAC address (ar$sha) into the same packet. It will set the ar$tha to broadcast address as it does not know the target host T MAC address and set the ar$op to 1 as ARP request message and send out the message. The packet will be processed by all hosts connected to the network. Only the host carries the IP address specified in the ar$tpa will process this packet. It will extract the ar$sha from the ARP request packet and set it to ar$tha in the ARP reply packet. It will also set the ar$tpa to the IP address of host S from the field ar$spa from the ARP request message. It then sets the ar$spa with its IP address which will be the same IP address specified in the ar$tpa in the ARP request packet. Finally, it will set the ar$sha as its MAC address and ar$op to 2 as ARP reply and send this ARP reply packet to host S.

The problem lies on how the ar$tha is set to a host address while we do not know the target host address? In this case we will need to send the ARP request to one host only that is known to every host thus the ARP request message could be properly processed and ARP reply message generated. The know host in this case is what referred as ARP server. The ARP server is the host that keeps the <ip,MAC> mapping of all hosts connected to the same network. We will explain how the server is being set up in to process all the ARP requests sent by other hosts.

Protocol Design Goals

In summary:

1) To change the behavior of ARP broadcasting messages to ARP unicast.
2) To set up an ARP server to process all ARP request messages send to the server.
3) To collect all the <ip,MAC> mapping of all hosts connected to the same network for ARP server to function properly.
4) To enable the ARP server to setup the <ip,MAC> mapping of all hosts without causing extra packets being sent in the network otherwise the benefits are reduced – i.e. to reduce unnecessary messages in resolving <ip,MAC> mappings.
5) To enable the new approach to cater for manual IP assigning.

PROPOSED APPROACH

In order to grab the mapping of <ip,MAC> of any host, the DHCP protocol is the best resource as DHCP server is the one responsible in distributing the IP address to all hosts. But, DHCP never keep track of which host having which IP address. Thus, it is required to know how DHCP works in order to understand the new approach.

DHCP

Whenever any host is connected to a network, the host will need to be assigned an IP address (either manually or through DHCP). This section will explain how DHCP assignation of IP address is catered in the new approach. In any cases the IP address is assigned manually, the new approach has taken that into considerations as well. When the host gets connected to the network, it will pass through some steps in order to get an IP address from the DHCP server, the DHCP server will leases the IP address to the host, in other words the IP address is temporarily assigned to the host. This is to ensure that in any cases where the host is disconnected from the network, the IP address that is not used can be assigned to other host connected to the network later. Other the IP address, DHCP server will distribute other settings to the host as well including the subnet mask, gateway address, lease period of IP and some other information. In any network, there could be more than one DHCP sever distributing IP addresses to all hosts connected to the network. The range of IP address assigned by different DHCP servers are different, to avoid the possibility of IP conflicts. DHCP works over UDP.

DHCP Scenarios

In assigning IP addresses to hosts, there are many scenarios to be considered. This paper will show all scenarios and prove the new approach is consistent, having catered for all scenarios.

Scenario 1 – INIT

In this case, the host is connected to the network for the first time. The host S will send a DHCP DISCOVER message type as broadcast message. The DHCP server will responds with the DHCP OFFER message type. There might be more than one such message as it depends on how many DHCP servers in the network. Upon decision, the host S sends a DHCP REQUEST (broadcast) to the DHCP server. The reason the DHCP REQUEST message is broadcasted so that other DHCP server offering IP addresses will know that the client is not requesting IP addresses from them thus could assign the IP address to other hosts.

If everything is fine the DHCP server will respond with a DHCP ACK message and the client is considered bound that is, it is assigned an IP address and can start communicating with any hosts in the network.

Scenario 2 – RENEW/REBIND

In this case, the host is connected to the network and the IP address leased from the DHCP server is approaching its lease expiry time. There are two Timers in the lease expiry mechanism. First is the Time to renew, T1, and second is the Time to Rebind, T2. At T1, S will send a DHCP REQUEST (unicast) and expects a DHCP ACK from the server. If DHCP ACK is received, the lease validity of this IP address is set to maximum. If not, and T2 is almost up, S will send the DHCP Request as broadcast and expects a DHCP ACK from the DHCP server. If DHCP ACK is received, the lease validity of this IP address is set to maximum. If no DHCP ACK is received and T2 is expired, the host S will enter into INIT mode again as in Scenario 1.

Scenario 3 – INIT-REBOOT

In this case, the host is connected to the network and the host is rebooting for whatever reason. Upon booting up, it will send a DHCP Request message and expect a DHCP ACK from the server. If no DHCP ACK is received the host S will enter INIT mode again as in Scenario 1.

Scenario 4 - DHCP NAK/DHCP DECLINE

In this case, the IP address requested by the client has been given by a DHCP server but the client found that the IP address is actually assigned to someone else. The host will enter INIT mode again as in Scenario 1.

Scenario 5 - DHCP RELEASE

In this case, the IP address given to this client is no more needed by the client. The machine will send a DHCP RELEASE message to the DHCP server. The next time it wants to connect to the same network again, it will enter INIT mode again as in Scenario 1

Scenario 6 – DHCP INFORM

In this case, the IP address is being assigned manually to the host and the host will send a DHCP INFORM to the DHCP server informing the server that the IP address is taken by the host. The DHCP server might send a DHCP ACK message to the host for other information needed by the client without giving a new IP address to the host.

Summary Of All Scenarios

In any of the above scenarios, for the host to be assigned an IP address, there is one common cycle to go through in order to consider that the host is being assigned an IP address. The message cycles for all scenarios are shown in table 1. Obviously, the two messages that must be found are DHCP REQUEST message send from the client and the DHCP ACK message from the DHCP server. The message type of DHCP could be recognized from the DHCP options (Microsoft, 2006).

Table 1. Summary Of DHCP Messages For All Scenarios

Scenario 1 (INIT)	Scenario 2 (RENEW/ REBIND)	Scenario 3 (INIT/ REBOOT)	Scenario 4 (NAK/ DECLINE)	Scenario 5 (RELEASE)
1. DISCOVER	1. REQUEST *	1. REQUEST	1. REQUEST	1. RELEASE*
2. OFFER	2. ACK*	2. ACK	2. ACK	2. DISCOVER
3. REQUEST			3. DECLINE	3. OFFER
4. ACK			4. DISCOVER	4. REQUEST
			5. OFFER	5. ACK
			6. REQUEST	
			7. ACK	

* - Unicast message

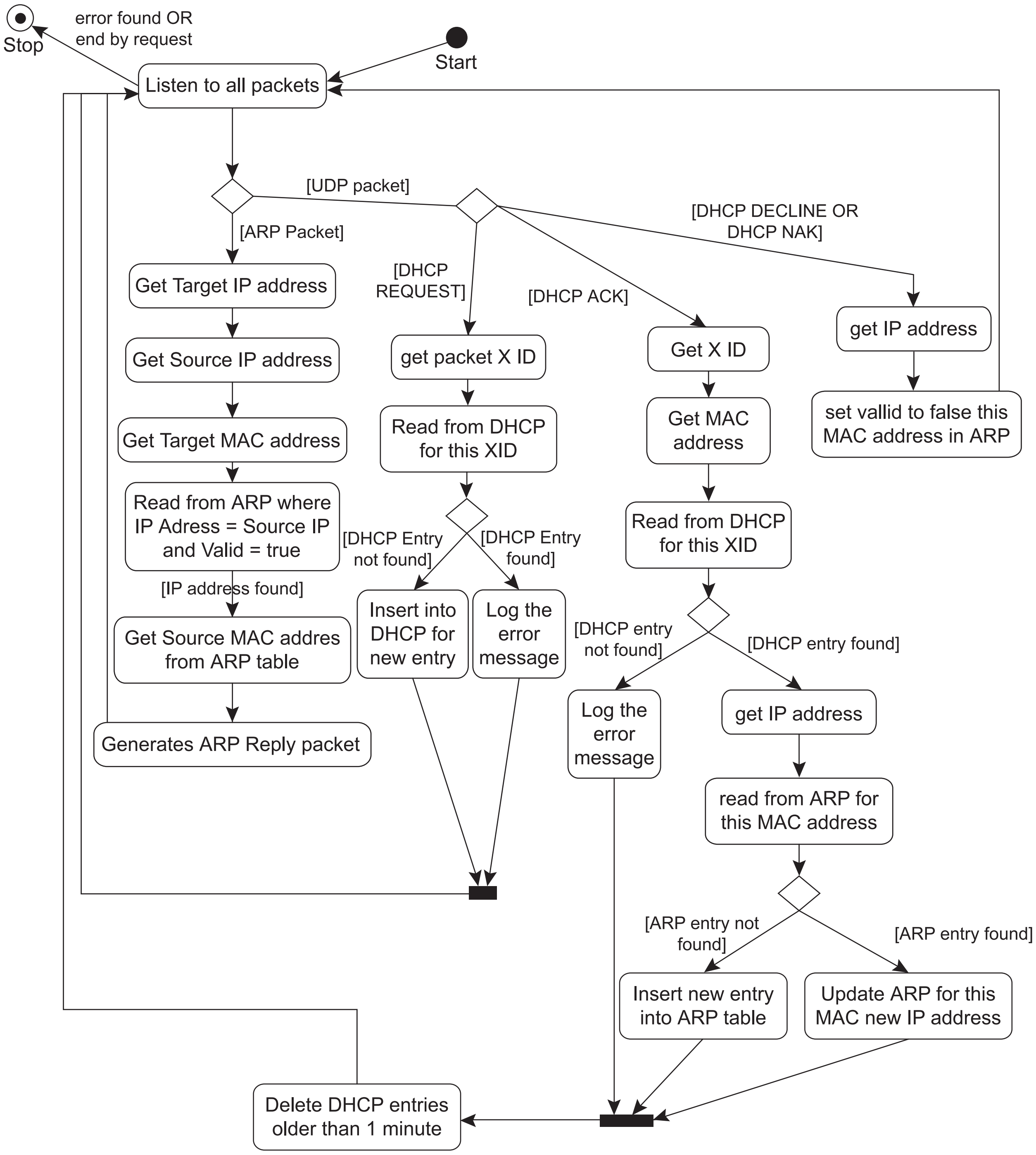

Stop
error found OR end by request
Start
Listen to all packets
[UDP packet]
[ARP Packet]
[DHCP DECLINE OR DHCP NAK]
[DHCP REQUEST]
[DHCP ACK]
Get Target IP address
Get Source IP address
Get Target MAC address
Read from ARP where IP Adress = Source IP and Valid = true
[IP address found]
Get Source MAC addres from ARP table
Generates ARP Reply packet
get packet X ID
Read from DHCP for this XID
[DHCP Entry not found]
[DHCP Entry found]
Insert into DHCP for new entry
Log the error message
Get X ID
Get MAC address
Read from DHCP for this XID
[DHCP entry not found]
[DHCP entry found]
Log the error message
get IP address
read from ARP for this MAC address
[ARP entry not found]
[ARP entry found]
Insert new entry into ARP table
Update ARP for this MAC new IP address
get IP address
set vallid to false this MAC address in ARP
Delete DHCP entries older than 1 minute

Figure 2. Activity diagram for new ARP Protocol

DESIGN ARCHITECTURE

This new approach has basically two main modules. The first module is to build up the ARP table:- This module basically listens to all packets transferred between each hosts in the network and filter out those non-interested packets and try to build up the ARP table based on the DHCP messages passed between each host and the DHCP server. The second module to generate the ARP Reply.- This module will answer all ARP requests from other hosts on the network by looking into all ARP request packets and search from the ARP table in cache if it has the MAC address of the host having the IP address specified in the ARP request packet and reply the host with an ARP reply packet should the server have that entry in the cache.

LOGIC DIAGRAM

The logic activity diagram of the protocol is shown in Figure 2.

PSEUDOCODE

Following is the pseudocode for the new approach:

```
      ?Is this an UDP or ARP packet?
Yes: an ARP Packet (Assume in the new approach there will be no more ARP  Reply message send
from other machine except from the ARP server thus this ARP message must be a ARP Request.
Even if in this classic ARP environment, the MAC of the ARP server will never exist in ARP.MAC
thus it will never find the record of this ARP server in the ARP table thus will never generates the
ARP reply bringing the system into infinity loop)
        Get the TargetIP address (ar$spa field in ARP packet format)
        Get the SourceIP address (ar$tpa field in ARP packet format)
        Get the Target MAC address (ar$sha field in ARP packet format)
Retrieve the record from ARP table for IPAddress (ARP.IPAddress)= SourceIP and valid = true
        Found:
          Get the Source MAC address (ARP.MAC)
Generates an ARP reply setting ar$spa =  SourceIP address, ar$sha =
SourceMAC address, ar$tpa = TargetIP address, ar$tha = Target MAC
address
Yes: a UDP Packet
        ?What type of DHCP Message?
         If DHCP Type = DHCP Request:
          Get the XID from the DHCP packet (xid in DHCP packet format)
           Retrieve the record from DHCP table (DHCP.XID) with this XID
           Found:
       Log the error into LOG table (DHCP record for this XID should not
       exist as XID is a unique number)
          Not Found:
Insert new record into DHCP table setting DHCP.XID=XID,
DHCP.MAC = Source MAC address from the DHCP Packet (chaddr in
DHCP packet format),  DHCP.IPAddress = Source IP address from the
DHCP Packet (yiaddr in DHCP packet format), Time1 =  Current Time
         If DHCP Type = DHCP ACK:
           Get the XID from the DHCP packet (xid in DHCP packet format)
           Get the MAC address from the DHCP packet (chaddr in
        DHCP packet format)
           Retrieve the record from DHCP table (DHCP.XID) with this XID
           Found:
              Get the IP address from the DHCP packet (yiaddr in DHCP packet
                 format)
             Retrieve the record from ARP (ARP.MAC) with this MAC address
  Found:
Update ARP table setting ARP.IPAddress to the IP address (the
machine with the MAC address has been assigned a new IP address)
             Not Found:
  Insert new record into ARP table setting ARP.MAC= MAC address,
  ARP.IPaddress = IP address, ARP.Dynamic = true, Valid = true (this
  is a new machine connected to this network)
Not Found:
  Log the error into LOG table (DHCP record for this XID should be
  exist as DHCP ACK packet would not be generated without DHCP
  Request)
        If DHCP Type = DHCP DECLINE or DHCP NAK:
Get the MAC address from the DHCP Packet (chaddr in
  DHCP packet format)
           Retrieve the record from ARP (ARP.MACAddress) with this MAC address
           Found:
              Set the valid field (ARP.Valid) for this record to false
```

ADDRESS RESOLUTION IN UNICAST MODE

The new approach to ARP accomplishes the protocol design objectives as outlined in Section 2.4 as follows:

1) **To change the behavior of ARP broadcasting messages to ARP unicast.**
   All ARP messages either ARP request or ARP reply are all in unicast mode.
2) **To set up an ARP server to process all ARP request messages send to the server.**
   The ARP server has been set up in order to prove the new approach is workable.
3) **To collect all the <ip,MAC> mapping of all hosts connected to the same network for ARP server to function properly.**
   The mechanism has shown how the <ip,MAC> mapping of all hosts connected to the same network are being collected by the ARP server
4) **To enable the ARP server to setup the <ip,MAC> mapping of all hosts without causing extra packets being send in the network otherwise forfeited the purpose of this paper – i.e. to reduce unnecessary messages in resolving <ip,MAC> mappings**
   The new approach managed to use the DHCP messages in order to build up the ARP table in the ARP server without introducing additional overhead for the same purpose.
5) **To enable the new approach to cater for manual IP assigning.**
   In order to cater for manual IP addresses, the new approach allows the administrator to key in the <ip,MAC> manually into the ARP server's ARP table and the administrator could even set the dynamic field of the table to false to indicate that this address was assigned by the administrator.

PERFORMANCE

In order to prove the new approach does not compromise the performance of ARP even as it is reducing the network load, the ARP server was tested in order to answer one ARP request from one of the host connected to the same network. In Fig. 5, we see that the machine 192.168.1.163 sends a broadcast ARP request and the machine 192.168.1.124 which the sender was trying to resolve to MAC address replies with the ARP reply immediately as shown in packet 14 above. Note that on packet 19, the ARP server sends an ARP reply from the ARP table and the machine 192.168.1.163 receives the reply. Also note that before the machine 192.168.1.163 got the reply from the ARP server, the communication between 192.168.1.163 and 192.168.1.124 were going on (packet 15 and 16), the communication keeps going on after the ARP server sends the ARP reply (packet 21 and 22). Thus, it proved that the ARP cache of machine 192.168.1.163 has got the correct entry for the MAC address of 192.168.124. Noticed that the time difference introduced is only 2.269030 – 2.128479 which is 0.140551 second. If implemented with a lower level programming language like C, the time difference is expected to be even less significant.

ARP in WAN

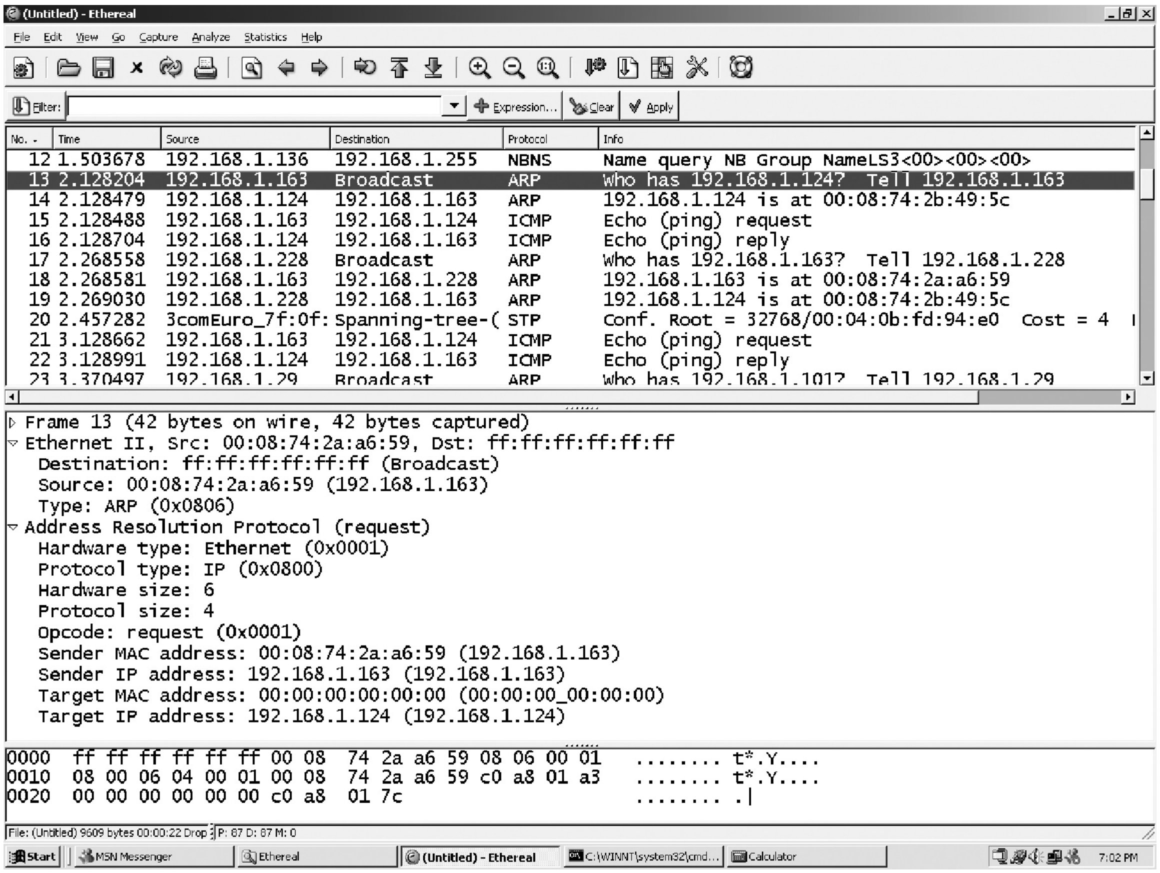
In order to bring the same concept into Wide Area Network, the new approach can cater for it easily as each network separated by the broadcast domain will each have their own ARP server. When one packet needs to travel to another broadcast domain, the router or gateway will take part in two broadcast domain and thus manage to pass the packet to another broadcast domain. The message can then be easily transported to the destination just like what happen in traditional ARP mechanism.

CONCLUSIONS

This work is derived from the point of reducing unnecessary ARP broadcast messages. After the ARP behavior is changed to unicast mode, the ARP request and reply will work in unicast mode and no more ARP messages are flying all around the network. All ARP replies will be from the ARP server and the hacker cannot perform ARP poisoning easily as any other ARP replies can be ignored in the future version of ARP. Any non-authorized host cannot be connected to the network. If the hacker would like to perform anything in the network, it must release its MAC address in the network in order to join the network as the <ip,MAC> mapping will be updated in the ARP server for all DHCP clients or inserted into ARP table by the administrator. Any hacker would not risk that by revealing themselves.

In future version of ARP, it could be easily modified to add more security into the ARP without implementing other complicated mechanisms, as the basic mechanism of ARP uncasting reduces many chances of information being easily grabbed on the network while this is avoidable, as suggested in this paper. The ARP protocol could even being modified to make the host as a honey-pot that any suspicious ARP request or reply packets being trapped and reported to administrator. ARP could be working in non-stateless mode thus no update on the cache entry is allowed without ARP request.

This paper only initiates the first approach in order to make ARP more efficient. After that, ARP could be enhanced further thus taking into considerations of security as well. Without this approach, ARP security is very hard to be implemented as ARP broadcast mechanism has caused many information being released while this could be avoided. By improving ARP, it is hoped that this will initiates a new way of redefining and enhancing many protocols being used currently and some has been used for a long time while many protocols could be enhanced to suite the current needs of networking requirements.



```
No. -  Time      Source           Destination       Protocol  Info
  12 1.503678  192.168.1.136    192.168.1.255     NBNS      Name query NB Group NameLS3<00><00><00>
  13 2.128204  192.168.1.163    Broadcast         ARP       Who has 192.168.1.124?  Tell 192.168.1.163
  14 2.128479  192.168.1.124    192.168.1.163     ARP       192.168.1.124 is at 00:08:74:2b:49:5c
  15 2.128488  192.168.1.163    192.168.1.124     ICMP      Echo (ping) request
  16 2.128704  192.168.1.124    192.168.1.163     ICMP      Echo (ping) reply
  17 2.268558  192.168.1.228    Broadcast         ARP       Who has 192.168.1.163?  Tell 192.168.1.228
  18 2.268581  192.168.1.163    192.168.1.228     ARP       192.168.1.163 is at 00:08:74:2a:a6:59
  19 2.269030  192.168.1.228    192.168.1.163     ARP       192.168.1.124 is at 00:08:74:2b:49:5c
  20 2.457282  3comEuro_7f:0f: Spanning-tree-(  STP       Conf. Root = 32768/00:04:0b:fd:94:e0  Cost = 4
  21 3.128662  192.168.1.163    192.168.1.124     ICMP      Echo (ping) request
  22 3.128991  192.168.1.124    192.168.1.163     ICMP      Echo (ping) reply
  23 3.370497  192.168.1.29     Broadcast         ARP       Who has 192.168.1.101?  Tell 192.168.1.29

▷ Frame 13 (42 bytes on wire, 42 bytes captured)
▽ Ethernet II, Src: 00:08:74:2a:a6:59, Dst: ff:ff:ff:ff:ff:ff
    Destination: ff:ff:ff:ff:ff:ff (Broadcast)
    Source: 00:08:74:2a:a6:59 (192.168.1.163)
    Type: ARP (0x0806)
▽ Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 00:08:74:2a:a6:59 (192.168.1.163)
    Sender IP address: 192.168.1.163 (192.168.1.163)
    Target MAC address: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Target IP address: 192.168.1.124 (192.168.1.124)

0000  ff ff ff ff ff ff 00 08  74 2a a6 59 08 06 00 01   ........ t*.Y....
0010  08 00 06 04 00 01 00 08  74 2a a6 59 c0 a8 01 a3   ........ t*.Y....
0020  00 00 00 00 00 00 c0 a8  01 7c                     ........ .|

File: (Untitled) 9609 bytes 00:00:22 Drop  P: 87 D: 87 M: 0
```

Figure 2. Result of ARP Request from client and ARP with ARP reply (result display in Ethereal. (Ethereal, 2006))

REFERENCES

Gouda, M.G., & Huang, C.T. 2003. A Secure Address Resolution Protocol. *Computer Networks.* 41(1): 57-71

Plummer, D.C. 1982. Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48 bit Ethernet address for transmission on Ethernet hardware. IETF RFC 826.

Montoro, M. 2006. Oxid.it – Cain and Abel. (online) http://www.oxid.it/cain.html [10th October 2006]

Ethereal. 2006. Ethereal: A Network Protocol Analyzer. (online) http://www.ethereal.com/ [10th October 2006]

Droms, R. 1997. Dynamic Host Configuration Protocol. IETF RFC 2131.

Croft, W.J. and Gilmore, J. 1985. Bootstrap Protocol. IETF RFC 951.

Microsoft (2006). DHCP Options Supported by Clients. (online) http://support.microsoft.com/default.aspx?scid=kb;en-us;121005 [10th October 2006]

Metcalfe, R.M. and Boggs, D.R. 1976. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM.* 19(5): 395-404.

Bruschi, D., Ornaghi, A., & Rosti, E. 2003. S-ARP – a Secure Address Resolution Protocol. *Proceedings of the 19th Annual Computer Security Applications Conference,* pp. 66-74.