# Passive Duplicate Address Detection for Dynamic Host Configuration Protocol (DHCP)

Sangho Shin, Andrea G. Forte, Henning Schulzrinne
Email: {sangho, andreaf, hgs}@cs.columbia.edu

***Abstract*— During a layer-3 handoff, address acquisition via DHCP is often the dominant source of handoff delay, duplicate address detection (DAD) being responsible for most of the delay. We propose a new DAD algorithm, passive DAD (pDAD), which we show to be effective, yet introduce only a few milliseconds of delay. Unlike traditional DAD, pDAD also detects the unauthorized use of an IP address before it is assigned to a DHCP client.**

## I. Introduction

**As many 802.11 wireless networks have been widely deployed in streets and parks as well as in buildings, the usage of wireless networks have increased. Typically, wrieless networks are deployed by different vendors in streets and shops, thus users very likely pass through different wireless networks when they move around. Generally, when the wireless network changes, the IP address needs to be changed through a layer 3 handoff. Typically, layer 3 handoff takes an order of seconds and this is very critical to real-time applications such as VoIP, which have become very popular since a few years ago.**

**While many efforts to improve layer 2 handoff including [] and [] have been made, very few research ( []) have been done to improve layer 3 handoff. Mobile IP [] have been introduced long ago, it is not widely deployed yet and the practical use seems to be very far becuase of a few disadvantages [].**

**Fig. 1 shows the procedure of layer 3 handoff. When a layer 2 handoff is done and a new subnet is detected, MN acquire a new IP address and update the network configuration via the DHCP protocol.**

Duplicate Address Detection (DAD) is a key feature in the DHCP [1] architecture. DAD is responsible for preventing different clients from acquiring the same IP address and therefore disrupt each other's communication. The current DAD procedure uses ICMP echo request/reply, thus incurring in a delay on the order of one second. DAD introduces the largest delay of the whole DHCP procedure. When a L3 handoff occurs, the delay introduced by DAD is responsible for most of the total handoff delay. These delays are particularly disruptive when a mobile node (MN) moves from one 802.11-based subnet to another and can interfere with on-going VoIP connections.

We introduce a novel DAD procedure called passive DAD (pDAD) which allows detecting duplicate IP addresses in an efficient manner, without introducing any significant delay. pDAD can detect duplicate IP addresses more accurately than the current ICMP approach because of the firewall in Windows XP SP2 blocking by default responses to incoming ICMP echo requests. Furthermore, it also allows the DHCP server to find out about illegally used IP addresses that have not yet caused a duplicate address.

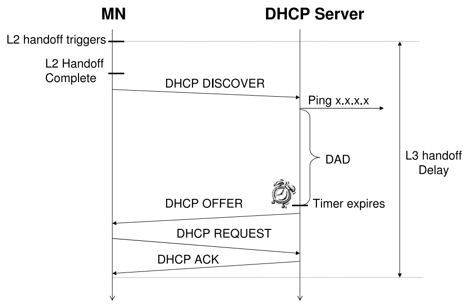

Fig. 1. Layer 3 handoff procedure

This new procedure is transparent to Mobile Nodes (MN) in the network and permits MNs to perform fast L3 handoffs.

## II. Related Work

Most of the work done in the network community for optimizing DAD, addresses DAD in the particular case of self-configurating networks such as ad-hoc networks [2] [3]. Other work has been done in the IPv6 context. In particular, the Optimistic DAD approach presented in [4], allows under certain assumptions, the use of a particular IP address that has not yet successfully completed the DAD process. Optimistic DAD is in fact based on the idea that ”in most cases DAD is far more likely to succeed than fail”. We present a new architecture that works in both IPv4 and IPv6 networks and does not require any assumption on the state of a particular IP address. The proposed architecture works for both mobile and non mobile networks, but it has been optimized for mobile environments. In particular, we have designed the new architecture with terminal mobility in mind, knowing that mobile nodes need to acquire a new IP address as fast as possible in order to prevent interruptions in their connection. Furthermore, our architecture is completely transparent to network nodes which consequentially will not have to change their *modus operandi*.

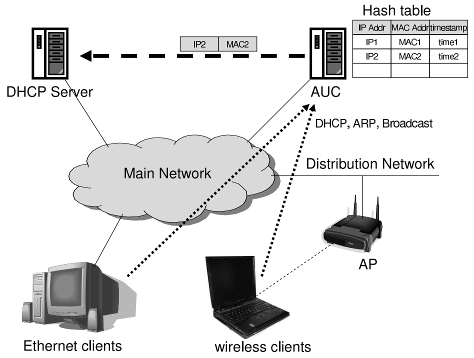

Fig. 2. Framework of PDAD

## III. Passive DAD

Passive DAD is a framework that detects IP addresses currently in use in one or more subnets. pDAD collects information on which IP addresses are in use in a specific subnet and informs the DHCP server about such addresses. In doing so, the DHCP server already knows which addresses are in use when a MN requests a new address and therefore it can assign the new address immediately without having to perform any further action during the assignment process. This allows us to remove any delay due to DAD during the address acquisition time. In the following we describe the pDAD architecture more in detail. The pDAD Internet Draft [5] contains further details.

### *A. Address Usage Collector (AUC)*

pDAD adds a new component to the DHCP architecture, namely Address Usage Collector. The AUC collects information on IP usage by monitoring ARP and broadcast traffic for a particular subnet. In order to monitor such traffic in an efficient manner, the AUC should be installed on a network component that is traversed by most of the network traffic such as a router. Usually, the AUC is installed in the DHCP Relay Agent (RA) which by default is installed on a router of a particular subnet.

By monitoring ARP and broadcast traffic, the AUC builds a table where each entry has the following information: IP address, MAC address and timestamp. Every time a new entry is added to the table, the AUC sends a packet to the DHCP server that includes the IP address and MAC address pair. This information tells the DHCP server that a node with MAC address ”MAC” is using the IP address ”IP” and therefore that IP address is already in use and should not be assigned to anyone else. Figures 3 and 4 show the structure of an entry in the AUC's table and the structure of the packet sent by the AUC to the DHCP server.

In order to keep up to date the information about IP addresses currently in use, the AUC removes an entry from the table when its timer has expired. If the IP address for that entry is still in use, a new entry for this IP address will be added to the table.

| IP Address | MAC Address | Timestamp |
|---|---|---|

Fig. 3. Structure of entries in the AUC's table

| |
|---|
| Subnet Identifier (4 B) |
| MAC Address (6 B) |
| IP Address (4 B) |

Fig. 4. Structure of packets sent by the AUC to the DHCP server

### *B. DHCP Server Behavior*

When the DHCP server receives a packet from the AUC, it checks the association IP-MAC to see if such an address was legally assigned or not, that is if the DHCP server has assigned that IP address to that client or not. If the IP address is in the unassigned IP pool, it means that such address was illegally taken, the DHCP server then removes it from the unassigned IP pool, and registers it to a bad-IP list which will also mark the IP as currently in use. In the bad-IP list there is a similar mechanism to the one used in the AUC's table where each entry has a timestamp. An IP address in the bad-IP list is removed from the list when its timer has expired, in this way the DHCP server has always up-to-date information on IP addresses currently in use.

By using pDAD, the DHCP server has also much more control on the network. For example, the DHCP server could configure packet flow rules in the egress router that perform some actions to block the IP addresses that have been illegally acquired by some malicious MNs. Furthermore, some form of intrusion detection could also be implemented.

In addition to the previous considerations, pDAD also allows the DHCP server to know about duplicate addresses as they occur and not just when an MN requests an IP address. In such a scenario, the DHCP server triggers a renew of the IP address for the legitimate user by using the DHCP FORCERENEW message [6]. No action can be initiated on the malicious user side as the malicious user does not use the DHCP infrastructure.

## IV. Implementation

We have implemented pDAD using the ISC DHCP software package [7], dhcpd is probably the most widely used DHCP server today. We have modified dhcpd to handle packets from the AUC and implemented the AUC functionality into the relay agent.

## V. Experiments

### *A. Experimental setup*

For running the experiments we installed dhcpd on a desktop machine with a 3 GHz Pentium 4 processor and 1GB RAM, the RA+AUC was installed on a linux server with a 3 GHz Pentium 4 processor and 1 GB RAM. Linux kernel 2.6 was used on all machines.

In order to check the traffic load between DHCP server and AUC, we installed our modified dhcpd and RA+AUC in the Computer Science (CS) network in Columbia University.

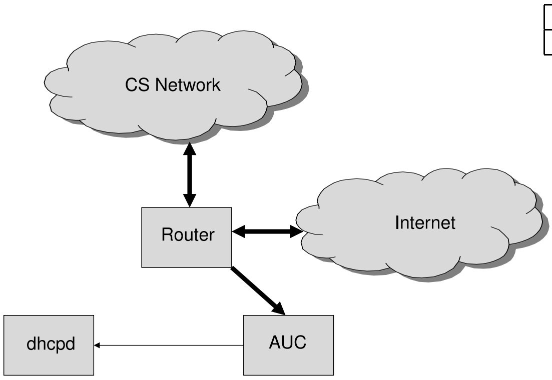

Fig. 5. Experimental setup

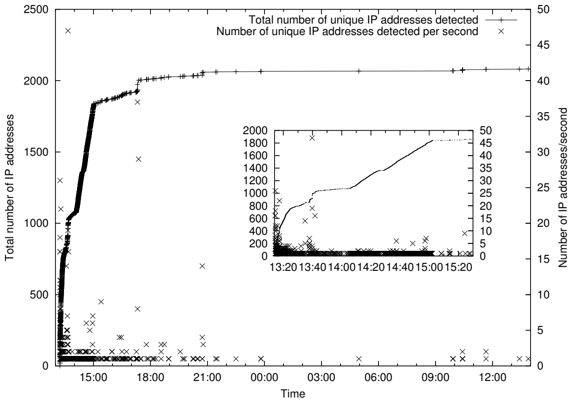

Fig. 6. Number of new IPs detected by DHCP

Dhcpd only processed packets from the AUC and the RA worked only an as AUC. No DHCP traffic was generated in the infrastructure itself. This was done in order to measure traffic and CPU load caused by pDAD only. The router of CS network forwarded all incoming and outgoing packets of CS network forwarded to the server in which AUC was installed. In order to collect IP address and MAC address information, the AUC module in the RA sniffed all broadcast and ARP packets from the router of CS network. The AUC then transmitted the address information packets to the DHCP server via Ethernet.

### B. Experimental results

We have performed the experiment a few times through two weeks, and show some results in this section.

Fig. 6 shows the distribution of number of new IPs DHCP server detected. We have detected 2092 IPs for a day from a experiment, and around 1800 IPs among 2092 IPs, about 86%, were detected within an hour and a half, and 47 IPs are detected in a second at peak time.

TABLE I
NUMBER OF MACS WITH MULTIPLE IPS

| Number of IPs mapped to a MAC | 2 | 3 | 4 | 6 | 9 | 10 | 77 |
|---|---|---|---|---|---|---|---|
| Occurrences | 13 | 3 | 1 | 3 | 1 | 1 | 1 |

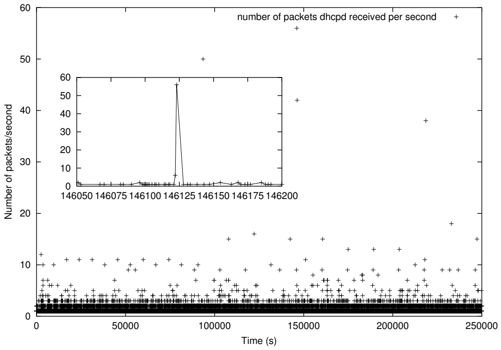

Fig. 7. Traffic volume between DHCP server and relay agent

In order to verify the measurement results, we acquired the DHCP log of a day from the administrator of CS netowkr, and confirmed from the DHCP log that we have detected all IPs assigned by the DHCP server during the day.

As shown in Table I, some MAC addresses had multiple IP address mappings: for example, 77 IP addresses were mapped to a MAC address. We have ideintified that a firewall with proxy ARP enabled was installed in the node and the node was responding to all the ARP requests to nodes inside the firewall. In another case, we have confirmed from the DHPC log we acquired that a node requested multiple IPs to the DHCP server legitimately, and it was identified as a VPN server of a lab. Also, we have detected 136 unique IP collisions caused by a node with MAC address 'ee:ee:80:xx:xx:xx', which seems to be a malicious node because the 'ee:ee:80' is not registered as an public Organizationally Unique Identifier (OUI).

*1) Overhead of DHCP server:* Fig 7 shows the traffic load between AUC and DHCP server during the experiment. The inner graph shows the same result of its peak time, where we have measured such a peak at 56 packets per second. However, we have identified that only one IP-MAC pair was a new entry

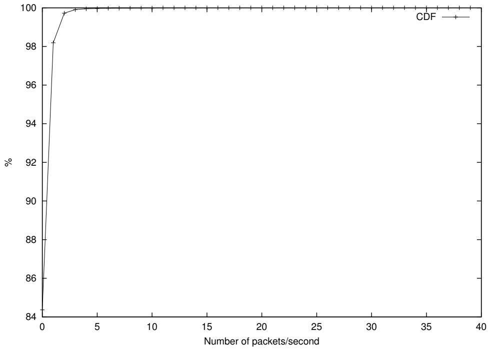

Fig. 8. Cumulative distribution function of number of packets per second DHCP server received

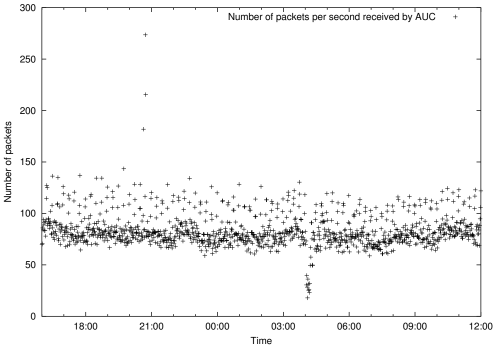

Fig. 9. ARP and broadcast traffic volume to the AUC

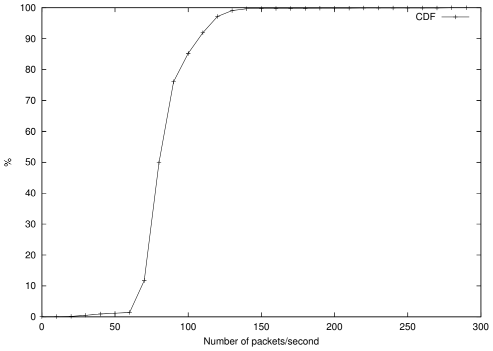

Fig. 10. CDF of the ARP and broadcast traffic to the AUC

and the rest of them were already in the table of DHCP server, which means that the timer of the 55 entries expired within a second coincidently, even though their expiration time were distributed. **Also, as we can see from the figure, DHCP received less than 10 packets from AUC most of the time and the overload to process the packets should be very small. We also confirmed that the additional CPU load to process the packets was negligible.**

Fig 8 shows that the cumulative distribution function of the number of packets per second the DHCP server received from the AUC. We can see that the DHCP server received fewer than 10 packets from the AUC for 99% of the time.

Each packet sent by the AUC to the DHCP server contains one IP, MAC pair and the RA IP address. The packet payload is 14 bytes as shown in Fig 4 in Section III-A, bringing the total (payload + headers) packet size to 80 bytes. So, the bandwidth at peak time is 4480 B/s, and usually less than 800 B/s.

*2) Overhead of AUC:* In the experiment, AUC received 11,000 packets every second in average, and the most of the packets were the unicast packets. AUC has discarded the ARP requests and responses generated by the router because they are not useful in collecting IP address usage and the traffic volume is very large. Among the 10,200 packets AUC received every second only 80 packets, which is less than 1%, were the ones that AUC processed to collect IP usage information. Fig. 9 shows the ARP and broadcast traffic volume to the AUC from the router. The peak we observed was 273 packets per second, however, Fig. 10 shows that the AUC processed less than 100 packets every second for 90% of the time.

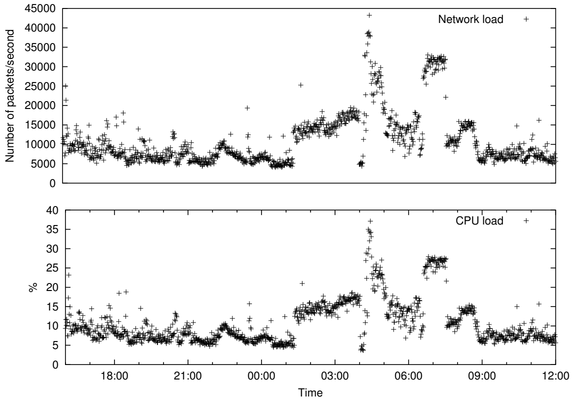

Fig. 11. Correlation of CPU load of AUC and traffic volume AUC received

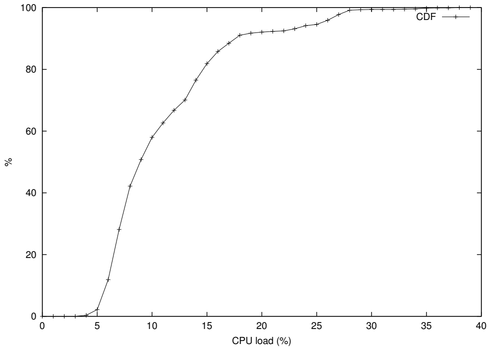

Fig. 12. Cumulative distribution function of CPU load of AUC

Figs 11 and 12 show the CPU load of the AUC and the correlation with the number of all packets AUC received every second. The AUC used around 40% of the total CPU power at peak time, but for 90% of the time, the CPU load was less than 20%. As Fig. 11 shown, the CPU load of the AUC is exactly proportional to the traffic volume to the AUC, which means that CPU was used mostly in filtering uninterested packets such as unicast and the ARP packets from the router itself. We can infer it also from the fact that only less than 1% of the packets AUC received were used by AUC to collect IP address usage. This is because AUC received both incoming and outgoing packets of CS network, even though AUC needs to monitor only the packets from the CS network. Therefore, the CPU load can be significantly reduced if AUC can receive only the outgoing packets from the CS network.

**In order to check the performance and overhead of pDAD in actual large scale wireless networks, we performed additional experiments in Columbia Univeristy Wireless Network, and we confirmed pDAD works in a large scale wireless mobile networks very well without overhead. The experiment results is summarized in a seprate technical report [] and please refer to the technical report for the detail.**

## VI. Conclusions

We propose a new protocol, pDAD which does not introduce any overhead or additional delay during the IP address acquisition time, therefore making it particularly efficient in mobile environments where handoff delays can be critical for real-time communication. Furthermore, pDAD can detect IP collision in real time, and the DHCP server can take additional actions to resolve the IP collisions. We have also shown that the traffic load between DHCP server and AUC is very small and therefore it does not interfere with the normal DHCP behavior.

## References

[1] R. Droms, "Dynamic host configuration protocol (dhcp)," RFC 2131, Mar 1997.

[2] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in *ACM MobiHoc*, June 2002, pp. 206–216.

[3] K. Weniger, "Passive duplicate address detection in mobile ad hoc networks," in *IEEE WCNC*, Mar 2003.

[4] N. S. Moore, "Optimistic Duplicate Address Detection for IPv6," December 2005.

[5] A. G. Forte, S. Shin, and H. Schulzrinne, "Passive duplicate address detection for dynamic host configuration protocol (dhcp)," Internet draft, Nov 2005.

[6] Y. T'Joens, C. Hublet, and P. D. Schrijver, "DHCP reconfiguration extension. Internet RFC 3203." December 2001.

[7] Internet System Consortium (ISC). dhcp-3.0.3. [Online]. Available: http://www.isc.org/index.pl?/sw/dhcp/