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

Andrea G. Forte, Sangho Shin, Henning Schulzrinne
Email: {andreaf, sangho, 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

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.

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

## II. 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 [2] contains further details.

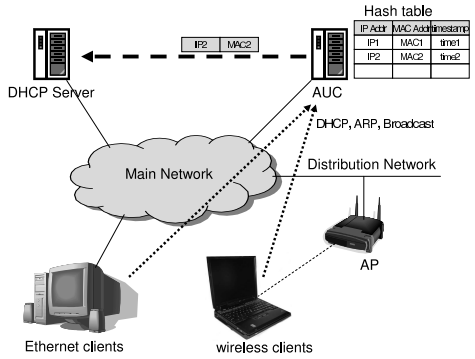


Fig. 1. Framework of PDAD

pDAD adds a new component to the DHCP architecture. The Address Usage Collector (AUC), which is usually installed in the DHCP Relay Agent (RA) [1], collects information on IP usage by monitoring ARP and broadcast traffic. The AUC then builds a table where each entry has the following information: IP address, MAC address and timestamp. When 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. In order to keep information about IP addresses currently in use, up to date, the AUC removes an entry from its hash table when the corresponding timer has expired.

The expiration time of an entry in the AUC's table should have a value close to the lease time of IP addresses in that subnet. If such expiration time were to be bigger than the lease time, the AUC could consider IP addresses as still in use even though the leases for those IPs have already expired without having been renewed. If the expiration time were to be smaller than the lease time, the AUC would check some IP addresses more frequently than it should, perhaps introducing unnecessary overhead. In this last case, however, the AUC might be able to detect unused IPs in a more timely manner, before their lease actually expires. In order to prevent all the entries in the AUC's table from expiring all at the same time, we somewhat randomize the expiration time per each entry so that the expiration time is uniformly distributed on a certain interval with a mean value close to the lease time. This also prevents the AUC from sending a burst of packets to the DHCP server everytime all the entries in the AUC's table would

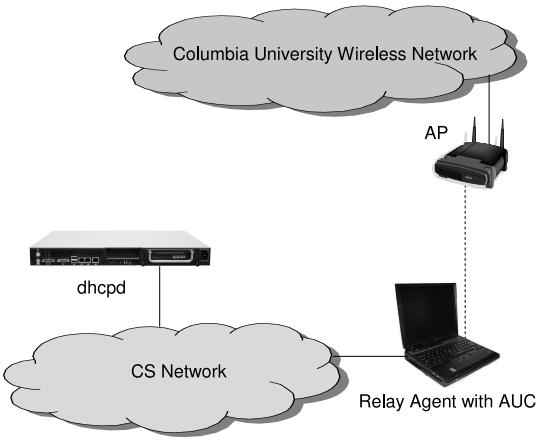


Fig. 2. Experimental setup

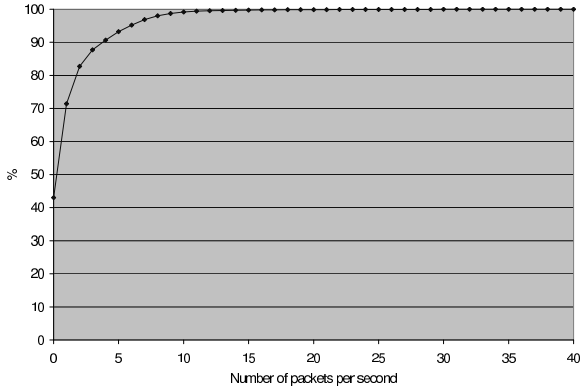


Fig. 3. Experimental results

expire.

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 by the DHCP server. If the IP address is in the unassigned IP pool, it means that it was illegally taken, the DHCP server then removes it from the unassigned IP pool, and registers it to a bad-IP list which also marks the IP as currently in use. In the bad-IP list there is a similar mechanism to the one used in the AUC's hash 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.

This mechanism also allows the DHCP server to have much more control. 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, pDAD allows the DHCP server to know about duplicate addresses as they occur and not just when an MN requests an IP address.

## III. Implementation

We have implemented pDAD using the ISC DHCP software package [3], 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.

## IV. Experiments

### A. Experimental setup

For running the experiments we installed dhcpd on an eRack Server machine with a 3 GHz Pentium 4 processor and 1GB RAM, the RA+AUC was installed on an IBM T42 laptop with a 1.7 GHz Pentium Mobile processor and 1 GB RAM. Linux kernel 2.4 was used on all machines. One Cisco AP1231G was used as Access Point (AP) in the Columbia 802.11b/g wireless network.

In order to check the traffic load between DHCP server and AUC, we installed our modified dhcpd and RA+AUC in the Columbia University 802.11b wireless network. Dhcpd only processed packets from the AUC and the RA worked only as an 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. Dhcpd was installed on an eRack server machine. The server was connected to the Computer Science (CS) network via Ethernet, the RA+AUC was installed on a laptop connected to both an AP of the Columbia University wireless network and to the CS network via Ethernet. In order to collect IP address and MAC address information, the AUC module in the RA sniffed all broadcast and ARP packets from the associated AP. Because the Columbia University wireless network is one big subnet, listening for packets from one AP allowed us to sniff most of the broadcast and ARP traffic in the subnet. The AUC then transmitted the address information packets to the DHCP server via Ethernet. Using this setup, we have measured the traffic load between AUC and DHCP server. We have performed the experiments for a period of time four to five hours long during peak time in terms of network activity (between 11:00 AM and 4:00 PM), on different weekdays.

### B. Experimental results

In our experiments the IP addresses on the wireless network had a lease time of 90 minutes. For the entries in the AUC's table we used an average expiration time of 60 minutes. During the experiments, the AUC detected about 1200 unique IP addresses within one hour. This makes the Columbia University wireless network sufficiently large to measure the overhead in terms of traffic load between DHCP server and AUC.

Fig 3 shows the cumulative distribution function of the number of packets per second that 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. Fig 4 shows the traffic load between AUC and DHCP server for a

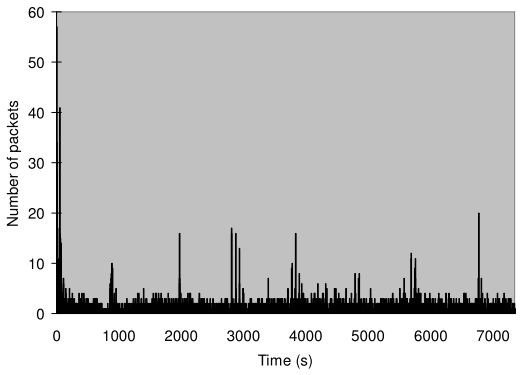


Fig. 4. Traffic volume between DHCP server and relay agent

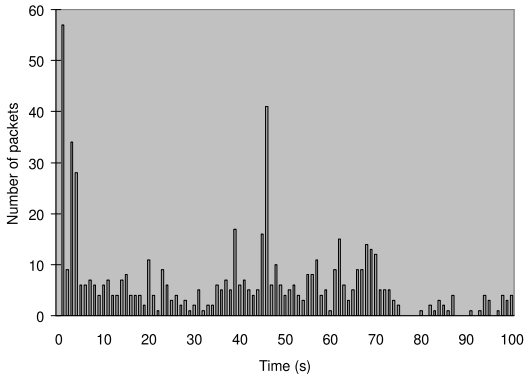


Fig. 5. Traffic volume between DHCP server and relay agent during the first 100s

two hour window of one of the experiments. Fig 5 shows the first 100 seconds of the same experiment in more detail. The peak in terms of packets exchanged between DHCP server and AUC is encountered at startup when the AUC is turned on for the first time. The AUC will have a completely empty table as no IP addresses have been collected and therefore each IP detected as in use will have to be added to the table. At the same time a packet will be sent to the DHCP server.

As Fig 5 shows, we have measured such a peak at 57 packets per second. Once the AUC's table has been built at least once, each entry will have its own timestamp that is its own expiration time. This means that there is not going to be a situation where all entries expire at the same time, thus significantly bringing down the peak of packets sent by the AUC to the DHCP server. In Fig 4 we can see that after the first expiration time in the worst case scenario the peak has a value of 20 packets per second at time 6769 s. Much lower than the initial peak of 57 packets per second.

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, bringing the total (payload + headers) packet size to 80 bytes. The first time the AUC is booted up, the maximum number of packets per second sent by the AUC to the DHCP server is 57, thus occupying a bandwidth of 4560 bytes per second. If we consider that at steady state, the maximum number of packets per second sent by the AUC to the DHCP server is 20, the amount of bandwidth used by the AUC is 1600 bytes per second at traffic peak.

## V. 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. 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, March 1997.

[2] A. G. Forte, S. Shin, and H. Schulzrinne, "Passive Duplicate Address Detection for Dynamic Host Configuration Protocol (DHCP)," Internet draft, Nov 2005.

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