Enhanced DHCP Client*

Silvia Giordano†
University of Applied Sciences of Southern Switzerland (SUPSI)
Manno, Switzerland
silvia.giordano@supsi.ch

Davide Lenzarini
Forward Information Technologies (FIT)
Manno, Switzerland
davide.lenzarini@forit.ch

Alessandro Puiatti
University of Applied Sciences of Southern Switzerland (SUPSI)
Manno, Switzerland
alessandro.puiatti@supsi.ch

Salvatore Vanini
University of Applied Sciences of Southern Switzerland (SUPSI)
Manno, Switzerland
salvatore.vanini@supsi.ch

ABSTRACT

The current network architecture for mobile devices presents applications with a synchronous model and employs end-to-end connections. In reality, devices are often disconnected from the network and, while they are disconnected, they can connect to other devices in the neighborhood and exploit the connectivity to allow the user to continue to work. This is the key concept of the Haggle project [1], a ground-up redesign of networking for mobile devices, to support the mobile user scenario. In order to benefit from users mobility, it is necessary to be able to opportunistically switch between (ad hoc) connections, whenever a new opportunity of connection arises. This imposes that connection set-up time must be minimal, considering that the nodes will probably stay in the range of each other for a short time frame. However, we saw that the 802.11 connectivity in Haggle carries significant overhead (especially when switching between Access Point (AP) mode and ad hoc mode, particularly when switching to AP mode) as the DHCP handshake takes a lot of time [2]. This suggested us to implement a new DHCP Client to speed-up the time for retrieving the IP during the connection to an AP, and to effectively implement opportunistic networking.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Network Protocols—*Applications(SMTP, FTP, etc.)*; C.4 [**Computer Systems Organization**]: Performance of Systems—*design studies, fault tolerance*

General Terms

Performance

Keywords

Opportunistic networking, DHCP

*Partially funded by Haggle project (EU 027918-FP6)

†IEEE senior member

Copyright is held by the author/owner(s).
CHANTS'07, September 14, 2007, Montréal, Québec, Canada.
ACM 978-1-59593-737-7/07/0009.

1. GOALS AND BASIC IDEA

The goal of the demo is to show the features and benefits of our enhanced DHCP Client. Indeed, our DHCP Client retrieves and sets an IP address from a DHCP server installed in commercial 802.11 access points, in a faster way than the current DHCP clients installed by default in any operating system do, in a non-atomic and controllable way. To do this, the enhanced DHCP Client can use two different paths: the **Standard DHCP path** (S-DHCPp), and/or the **ARP DHCP path** (A-DHCPp).

The S-DHCPp is the traditional way to retrieve a DHCP address as specified in [3], but with refined timeouts. The steps for allocating a new network address, using the S-DHCPp, are the following:

1. The client broadcasts a DHCPDISCOVER message on its local physical subnet;
2. Each DHCP server may respond with a DHCPOFFER message that includes an available network address;
3. The client receives one or more DHCPOFFER messages and chooses the server from which to request configuration parameters, based on the parameters contained in the DHCPOFFER message. Then it broadcasts a DHCPREQUEST message that include the indication of which server it has selected;
4. The server receives the DHCPREQUEST broadcast from the client and then commits the binding for the client to persistent storage and responds with a DHCPACK message containing the configuration parameters for the requesting client;
5. The client receives the DHCPACK message. If the client receives a DHCPNAK message or an unknown message type the DHCP process is aborted.

The A-DHCPp is a best effort way to retrieve a new network address: it exploits a particular feature of the DHCP Server to bypass the steps that compose the traditional DHCP protocol and retrieve the new network address in a faster way. When a DHCP server receives a DHCPDISCOVER message, it broadcasts an ARP request in order to know if some client already uses the proposed network address. This

request contains the new expected IP and the new expected gateway. If the ARP request is sent, our DHCP Client captures it and the new configuration parameters are derived. Since the network MASK is not present in the ARP broadcast request, a presumed value is calculated based on the following considerations:

- If the new IP address first byte belongs to an A class type, a class A subnet mask (255.0.0.0) is used.
- If the new IP address first byte belongs to a B (172) class type and the second byte value is between 16 and 31, a class B 172 subnet mask (255.255.240.0) is used.
- If the new IP address first byte belongs to a C (192) class type and the second byte value belongs to a class C type (168), a 255.255.0.0 mask is used.
- If the new IP address first byte belongs to a B (169) class type and the second byte value belongs to a class B type (254), a 255.255.0.0 mask is used.
- In all the other cases a default mask 255.255.255.0 is used.

2. S-DHCPP VS A-DHCPP

By default, our DHCP Client adopts the S-DHCPp for retrieving the new network address parameters. The user can enable the A-DHCPp by passing it as a parameter to the program. In this case the two paths are executed in parallel and, at the end of the DHCP negotiation, the parameters obtained are compared and updated if needed. In any case the parameters derived from the S-DHCPp are considered more relevant than the ones derived from the A-DHCPp. According to this assumption, if the S-DHCPp ends successfully, the possible combination of the comparison results and the corresponding actions taken could be the following:

- The S-DHCPp provides a network MASK different from the one set after the A-DHCPp. This case is considered as a best effort attempt and the MASK will not be changed.
- The IP assigned by the DHCP Server at the end of the negotiation is different from the IP contained in the ARP broadcast message. In this case the DHCP Client changes the previously assigned IP and the sockets eventually instantiated will generate a connection Exception that must be handled by the application that launches the DHCP Client.
- The Gateway assigned by the DHCP Server at the end of the negotiation is different from the IP contained in the ARP broadcast message. In this case the DHCP Client changes the previously assigned Gateway. If there were sockets instantiated between the end of the ARP path and the end of the DHCP negotiation that, in the meanwhile, were waiting for the SYN/ACK packet, they will retransmit the SIN packet and everything will work (the average time elapsed between these two stages is about 1 second as indicated in the next paragraph).

If during the S-DHCPp handshake some timeout arises, the S-DHCPp ends without success. At this point two different options can be chosen (this is an implementation choice):

1. Abort the entire DHCP process.
2. Use the parameters provided by the A-DHCPp.

3. PERFORMANCE RESULTS

We have done several tests on laptops running Windows XP and equipped with standard 802.11b/g cards. As depicted in figure 1, measurement results show that, in the best case (A-DHCPp ends successfully and the parameters inferred are correct), the average time for retrieving and setting the DHCP IP address is 300 ms. In the worst case (the A-DHCPp goes in timeout but the S-DHCPp ends successfully), the average time for retrieving the IP address is 1440 ms. Anyway, the required time for retrieving and setting the new IP obtained with the E-DHCP is significantly less than the average time requested by the Windows DHCP Client (3430 ms).

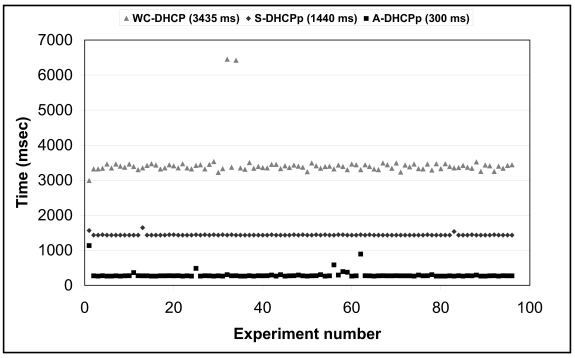

Figure 1: Average IP retrieving time - comparison among: the Windows XP client, the S-DHCPp client and the A-DHCPp client

4. FUTURE WORK

Our Enhanced DHCP Client is still under development and testing, and we plan to:

- integrate several improvements in terms of self-tuning and adaptation of network parameters (i.e. the read timeout when waiting for a packet to arrive);
- port and test the client under linux;
- develop an enhanced DHCP server customized on the basis of the E-DHCP.

5. REFERENCES

[1] HAGGLE Project. http://www.haggleproject.org.

[2] Jing Su, James Scott, Pan Hui, Eben Upton, Meng How Lim, Christophe Diot, Jon Crowcroft, Ashvin Goel, and Eyal de Lara. Haggle: Clean-Slate Networking for Mobile Devices. Technical report, Computer Laboratory - University of Cambridge, January 2007.

[3] Droms R. RFC 2131 - Dynamic Host Configuration Protocol. http://www.faqs.org/rfcs/rfc2131.html, March 1997.