Technical Report N 2: T2.1_06_TN_R01 Implementation and performance evaluation of DHCP Relay Agent module for Uni-Fu authetnication system

Mauro Brunato, Renato Lo Cigno, Albino Scalet, Danilo Severina

I. Introduction

This document describes the work that has been managed in the UdR of Trento in the year 2006 under the TWELVE Project on maintenance of the Uni-Fy system.

In this paper the description of a new feature related to assignment of IP addresses to clients is provided. The first version of Uni-Fy gate had a DHCP server that is built in the system and provides a subset of features a DHCP server keeps. The extended version of the authentication system provides a new Plug-in module that is able to interact with external DHCP servers for IP assignment.

After a short description of DHCP protocol and Relay Agent (Sec. II), in Sec. III we will describe the structure of the new Plug-in, how it is included in the system and the mandatory changes in code and in configuration parameters. In Sec. IV we will show some results related to the performance of the Uni-Fy gate for IP assignment procedure. The goal is compare the performance of the system with the two IP allocation mechanisms in terms of number of request packets that can be managed in a short time and the time to satisfy them.

II. DHCP Overview

The Dynamic Host Configuration Protocol (*DHCP*) [3] is a protocol that automates the procedures for assignment of IP addresses and required parameters for physically access to a network. The DHCP protocol is the evolution of Bootstrap Protocol (*BOOTP*) [2] and it is backward compatible to provide services to machines that implement BOOTP.

The network architecture for IP assgnment mechanism is made up of two elements: a *Client* that connects to the network and requests an IP address, and a *Server* that provides dynamic IP address to the clients and keeps taces of used and free IP addresses in the network. Furthermore, in a more complex scenario a third entity is involved in the handshake for network address requests: the *DHCP Relay Agent*. One or more DHCP Relay agent can be interposed between Client and Server and they forward packets that belong to DHCP transaction from the client to the server and viceversa.

In this section, we do not provide an in deep description of the protocols (both DHCP and BOOTP). We give an overview how the DHCP protocol is implemented and the behaviour in network configuration with and without Relay Agent devices.

A. Client – Server Scenario

The DHCP is based on UDP protocol and requires a packet handshake between the *Client* (the machine that requests an IP address) and the *Server* (the machine that manages the IP allocation for the network users). An example of correct and complete handshake is shown in Figure 1a). The handshake is always started by the Client and it is made up of 4 packets:

DHCPDiscover
is the first packet and it is sent by the Client when it is connected to the network: the packet has

source IP address equal to 0.0.0.0 and broadcast destination IP address 255.255.255.255 because the client has no information about the parameters (IP and MAC addresses) to contact the DHCP server; the packet is a request both to know information about DHCP server and to start an IP address request;

DHCPOffer
is sent by the Server and it contains information about both itself and a proposed IP address;

DHCPRequest
is sent by the Client and it contains a "formal" request of the IP address proposed by the Server in the previous packet;

DHCPAck
is the packet that ends the transaction and it confirms the assignment of proposed IP address to the Client.

The previous handshake is the simplest one. In a real scenario more than one DHCP Server can be in a network and because the DHCPDiscover packet is sent in broadcast, it is received by all the reachable DHCP Servers in the network. More than one server can answer so more than one DHCPOffer can be received by the Client: in this scenario the Client evaluates the offers and sends a DHCPRequest only to one DHCP server.

Each IP address is dynamically provided and has a *lease time* that identifies the live time of the allocated address. Before the lease expiration, the Client must send a request to renew the validity of IP. The renew procedure involves the handshake of DHCPRequest and DHCPAck and is always started by the Client.

The previous mentioned packets are the mandatory ones that are request by the standard, but other packets are foreseen. Here a short description of packets that can be sent by the Client or by the Server are listed. The Client can send the following packets over DHCPDiscover and DHCPRequest:

DHCPDecline
is sent if the Client has discovered that the offered IP address is already in use; in this case the Server must mark the network address as unavailable and inform server administrator;

DHCPRelease
the Client informs the Server that the IP address is released; the Server can mark the address as available;

DHCPInform
is sent when the Client requires more information from the Server or requests specific configuration parameters; the Server will answer with a DHCPAck packet with proper structure and information.

Instead, the Server can send the following packet over DHCPOffer and DHCPAck:

DHCPNack
is sent if something is wrong and it forces the Client to restart new request.

The packets sent by Server are only answers to request packets: the Server never starts communications.

As mentioned above, the interactions among devices use a UDP protocol and the involved ports are 67 and 68: the packets generated by the Client have source port 67 and destination port 68, while the packets received by the Client have source port 68 and destination port 67.

In a generic network one DHCP Server can manages more than one subnet IP address and it provides the correct IP address to the Client accordingly to the LAN the Client is (for instance the Client position can be identified using VLAN tag that detects the portion of LAN where cleints are connected).

B. Client – Relay Agent – Server Scenario

In a more extended scenario, a *DHCP Relay Agent* entity can be involved in the handshake for network address requests. The third entity is interposed between Client and Server and forwards packets from one to another. Unlike DHCP Server, a DHCP Relay Agent manages only one subnet IP network. The role of Relay Agent is:

- intercept the packets sent by the Client/Server;
- modify them properly;
- forward them to Server/Client.

Obviously, the Realy agent has the same behaviour both for Client packets and for Server ones.

The handshake for DHCP transaction in network with Relay Agent device is shown in Figure 1b). The renew procedure (as in previous case) involves the handshake of DHCPRequest and DHCPAck and the Client alway initiates it.

The interactions between Client and Relay Agent use UDP protocol and the ports mentioned in previous section, while interactions between Relay Agent and DHCP Server use UDP protocol but the packets are sent and received always on port 67.

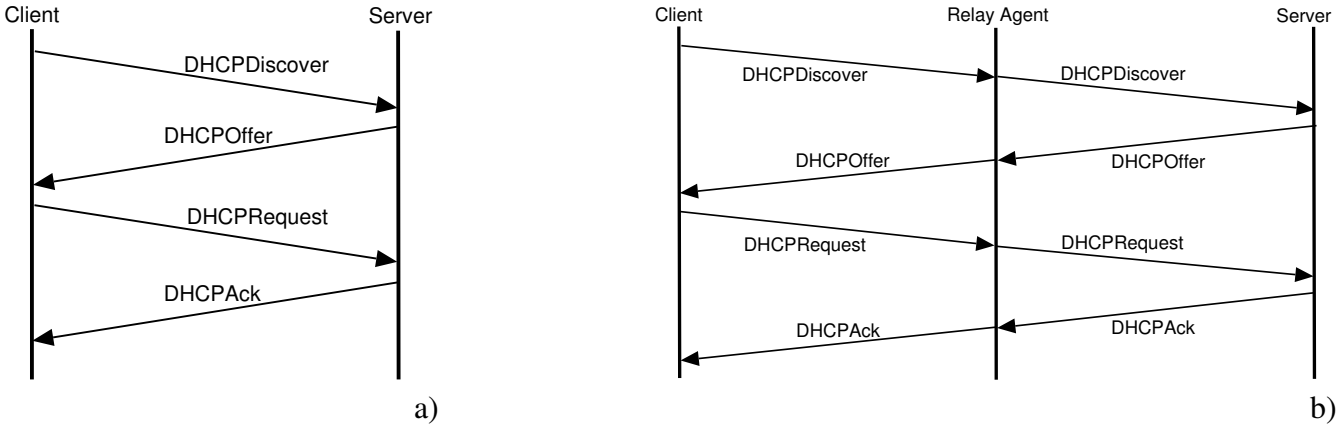

Fig. 1. Correct DHCP transaction with direct interaction between Client and Server (a) and with a Relay Agent (b).

III. RelayAgent Plugin

In this section we briefly describe the Uni-Fy gate architecture to provide information about the system where the DHCP Relay Agent module is inserted, how Relay Agent works, and the implementation of the Plugin and its interaction with the rest of the Uni-Fy system and network devices.

A. Uni-Fy gate

Uni-Fy is a Wireless LAN and HotSpot management system based on the general idea of captive portal, with distributed authentication, security, firewalling, and similar capabilities. It smoothly interfaces with DHCP management and NAT/firewalling. If external VPN access is needed, it could be supported and multiple parallel authentication data bases are natively foreseen. The high-level networking "philosophy" followed by Uni-Fy is the Open Access Network philosophy, as described, for instance, in [5], [6].

Figure 2 shows an overall view of the proposed network components:

- the blocks "**AP**" denote wireless access point devices;
- the **Gateway** component is basically a layer-3 switch with some additional ad-hoc functionalities): it checks packets that are directed from and to authorized clients and puts them into the right interface;
- the **GateKeeper** component receives packets belong to unauthorized clients from the Gateway: packets are subjected to precessing and are used to trigger events such as an authentication procedure. The authentication procedure will involve the client, the Gateway (for packet forwarding), the GateKeeper and an authentication provider, possibly chosen among many. The role of every component and the type of interaction depend on the authentication mechanism.

Figure 3 shows the basic block structure of the Uni-Fy system. It consists of two main blocks, the *Gateway* and the *GateKeeper*.

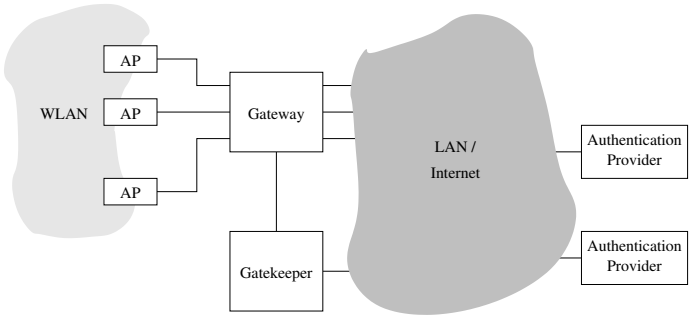

Fig. 2. An overall view of the Uni-Fy system.

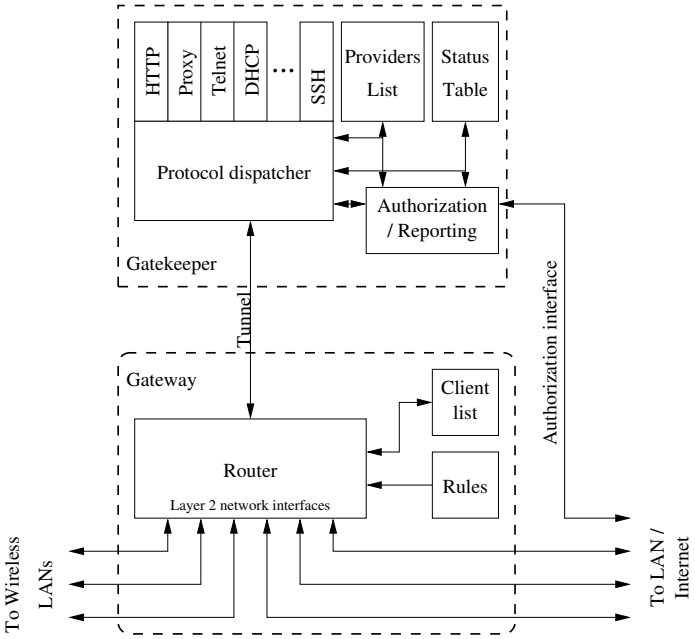

Fig. 3. Block diagram of the Uni-Fy system

The Gateway machine contains various network interfaces, some of which are directed to the wireless LAN, some to the wired LAN. Packets to and from the interfaces are managed by the *router* module accordingly to rules and list of authorized clients in the *client list*.

The GateKeeper component performs all tasks that require more processing than just looking at frame and packet headers. It basically receives all non-authorized packets from the Gateway through the "tunnel" shown in Figure 3. This tunnel is possibly implemented as a pair of UDP sockets. In case of Uni-Fy implementation within a single machine, a bidirectional IPC channel could be considered. Packets are processed with the aim of authenticating the user, and eventually sent back to the Gateway to be forwarded, or bounced.

The plug-in modules in the GateKeeper are used by the Dispatcher module in order to take decisions about specific protocols. In this work, we focus on DHCP Plug-in module that manages the DHCP service in the wireless LAN. When the Dispatcher receives a DHCP packet, it submits it to the DHCP module. To decide the appropriate action, the DHCP module can access the authorization list, where the status of every IP address is stored.

A more in depth description of the system can be found in [7], [8].

B. Relay Agent module

The goal of the work is the implementation of a new Plug-in module to extend the features of the system.

Nowadays the AAA system manages the IP address for wireless network clients. The DHCP server built in the Uni-Fy gate recevies DHCP packets and proposes a free IP address to the clients. The features supported by the built-in DHCP is a subset of features of a standard DHCP server. The aim of the development of a Relay Agent Plug-in is to allow the interaction with a DHCP server that can be located in an external network. For instance, if the authentication system is implemented in a preexistent network with its own DHCP server, the manager of the network should use it also for the wireless network. In such scenario the Realy Agent Plug-in acts as a DHCP Relay Agent and it provides interaction with the DHCP server (or DHCP servers).

The new plug-in is placed upon the Dispatcher as general Plug-in element and can be used instead of the DHCP Plug-in. The action of the Relay Agent Plug-in can be summarized as following step:

- receives a DHCP packet from the Dispatcher;
- modifies some fields of the packet;
- sends the packet to the external DHCP Server and waits for the answer;
- receives a packet from the external DHCP Server;
- modifies some fields of the packet;
- sends the packet to the Dispatcher that routes it to the Gateway and then to the client;
- interacts with the Status Table to update the status of the assigned IP address.

For optimization purpose, when the Relay Agent Plugin sends the packet to teh DHCP Server, it does not wait for the answer, because the waiting means the system is in idle mode and not other procedures can be performed. After the sending of Discover or Request packet to teh server, the Plugin ends the connection and the Offer or Ack packet forces a new connection.

IV. Tests and results

In this section we describe the bahavior of the authentication system with three DHCP netwrok configuration. First of all, the goal of the test is to underline the differences between performances with DHCP built-in servere and Relay Agent Plugin

the testbed is shown: it is configured as three different scenario where three different DHCP network configuration are implemented. An in depth description of the software to test the performances of the different network configurations follows. Finally there is a description of results of the test.

A. Scenario

The different involved scenarios have the goal to describe the performance of the system where only DHCP server is involved, Uni-Fy gate with built-in DHCP server is used, and Uni-Fy gate with DHCP Relay module is used. In all the previous cases, the systems are isolated: there are no element that can influence the performance of the system.

In Figure IV-A are shown the three different scenario:

- **Scenario I** (Figure IV-Aa):
  the direct interactions among clients and DHCP server is implemented;
- **Scenario II** (Figure IV-Ab):
  the direct interactions among users and Uni-Fy gate with built-in DHCP server is implemented;
- **Scenario III** (Figure IV-Ac):
  the interactions among users and ISC DHCP server through Uni-Fy gate Relay module is implemented.

The goal of the test is the evaluation of the performance of the assignment of network addresses when the Uni-Fy gate is used with the built-in DHCP server or with the DHCP Relay agent. To perform a complete comparison, also a configuration without the authentication system is tested.

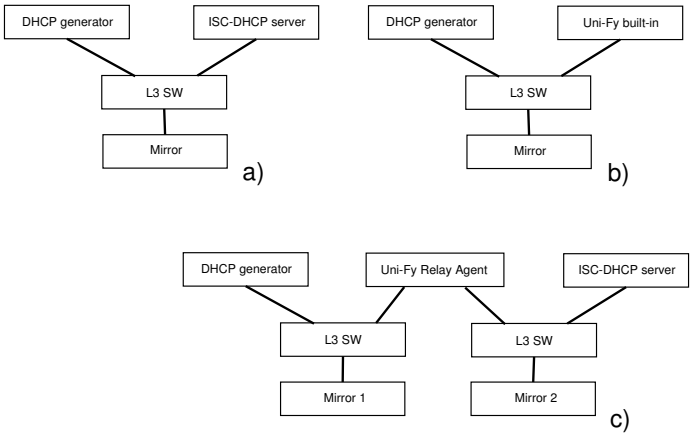

Fig. 4. Scenario for IP assignment test-bed: Scenario I (a), Scenario II (b), and Scenario III (c)

B. Software

In previous section an in depth description how the authentication system and DHCP Relay Agent module is provided. An in depth description of the other used tool to evaluate the system follows.

a) ISC server: The ISC DHCP Server [9] provides a freely redistributable reference implementation of all aspects of DHCP: Server, Client and Relay modules.

The DHCP server, client and relay agent are provided both as reference implementations of the protocol and as working, fully-featured sample implementations. Both the client and the server provide functionality that, while not strictly required by the protocol, is very useful in practice. The DHCP server also makes allowances for non-compliant clients which one might still like to support.

b) DHCP generator: In a real scenario the users request an IP address when it is connected to the network. In the testbed an high number of clients cannot be used because there are problem about both the number itself and their management. To perform tests that should evaluate under limit conditions the performances, a software that is able to mange DHCP transaction is developed. The structure of the software is shown in Figure 5 and its work can be summarized as:

- generation of DHCPDiscover packets;
- receiving of DHCPOffer packets;
- generation of DHCPReqest packets accordingly to received DHCPOffer packets;
- receiving of DHCPAck packets;
- evaluation of complete IP assignment transaction.

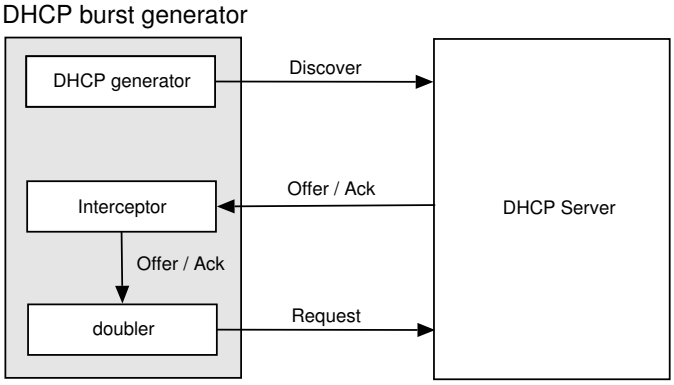

Fig. 5. Block structure of DHCP generator

As described before the request are always started by the clients: the server satisfy the requests and keeps traces about the transactions with the storage of information as MAC address of the applicant and the identifier of the transaction (*XID*). So, the DHCP generator must be able to forge packets with different source MAC address and XID to fake request: the goal is to force the DHCP server to consider all the requests as they are executed by different clients.

The generator must also keep traces about the number of sent packets and their marker (source MAC address and XID) to reconstruct correct and incorrect transactions. The generator send a burst of DHCPDiscover packets where the *burst* is a sequence of DHCPDiscovers in a short time (hundred of milliseconds) and each of them has a different XID and MAC address.

The source code of the generator is an evolution of Globber [10], a DHCP Denial of Service attack software.

C. Description of the test

For all th scenario described in the Section IV-A, the test is performed as:

for each burst intensity I_i
    for each j $\in [0, N_{MAX}]$
        sending $B_{I_i,j}$
        sleep

where I_i is the intensity of the burst and is the number of DHCPDiscovers in burst, $B_{I_i,j}$ is the j-th burst of intensity equal to I_i. Between two consecutive bursts there is a guard time to allows on the one hand the ending of the DHCP transaction and on the other hand the reset of the DHCP server. After each burst the guard interval is required to allow the expiration of the allocated network addresses, so the starting conditions of all the tests are the same. N_{MAX} is the number of the experiment for a single burst intensity and in this work it is set equal to 50. Finally the value of I_i are set in the interval between 10 and 190 with a gap of 10, so $I_i \in [10 * i, 10 + 10 * (i + 1)]$ with $i \in [0, 18]$.

The tests are performed using the following structure

set of the interface configuration on the different machines
start the tcpdump on interface of machine with DHCP generator with file output option
start the tcpdump on interface of machine with DHCP Server with file output option
for each burst intensity $I_i in\, [10, 20, 30, \cdot 190]$
    start DHCP burst generator
    for each j $\in [0, 50]$
        sending $B_{I_i,j}$
        sleep of DHCP generator and wait for ending of DHCP transaction
    stop of tcpdump

After the ending of all the tests we have two large dump files: one for Client side and one for Server side. The first one is sufficient to perform test, and the second one can be used to guarantee the results, so we can focus only to client dump file. The file must be split in smaller file and each of them must contains the handshake related to only one experiment, only one burst. For this purpose the tool *tpcslice* is used and at the end a tree of directory for storage the files is made.

Finally the data useful for the results can be provided counting the number of DHCPDiscover, DHCPRequest, DHCPOffer, and DHCPAck in the dump files. These data can be storage in well formatted file to generate graph with proper software (e.g. Gnuplot).

All the above described procedure can be implemented using script file to automatize everything.

D. Results

In this section we describe the results of the test in terms of number of received packet in the transactions and the mean times for a complete transaction.

Figures 6,7, 8 show the behavior of the DHCP transaction in terms of **cumulative** sending DHCPDiscover and DHCPRequest packets and received DHCPOffer and DHCPAck packets on the client side (dump on machine with DHCP burst generator)

Figures 9, 10, 11 show the same behavior of the previous graph, but in terms of mean correct sending and received packets.

Figures 12 shows the number of complete DHCP transaction normalized to the number of sent DHCPDiscover packets .

Figures 13, 14, 15 show the mean time required for correct and complete transactions and their confidence interval with confidence level equal to 95%. As in teh previous case, the confidence interval is computed with the repetitions of the same experiment. Finally Figure 16 show the comparison of the performance in term of time of the three system under test.

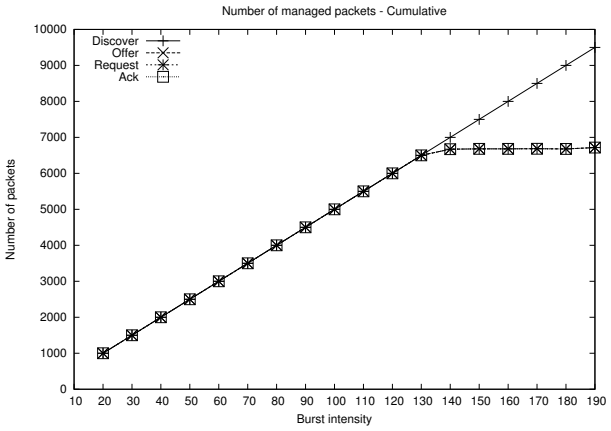

Fig. 6. Cumulative number of DHCP packets sent and received by client side with ISC DHCP Server.

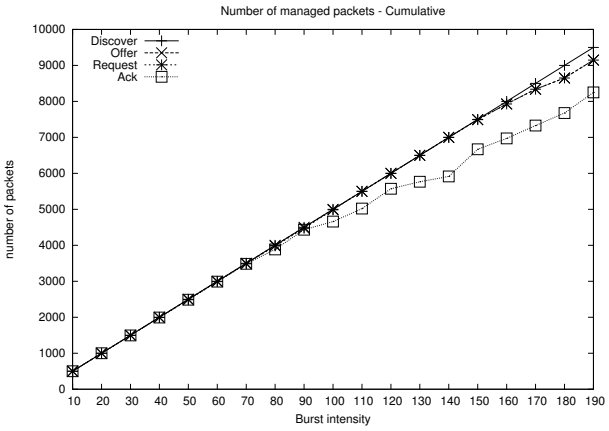

Fig. 7. Cumulative number of DHCP packets sent and received by client side with Uni-Fy DHCP Built-in.

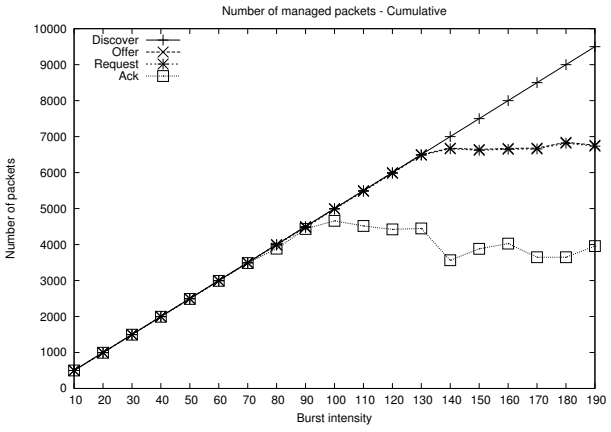

Fig. 8. Cumulative number of DHCP packets sent and received by client side with Uni-Fy DHCP Relay Agent.

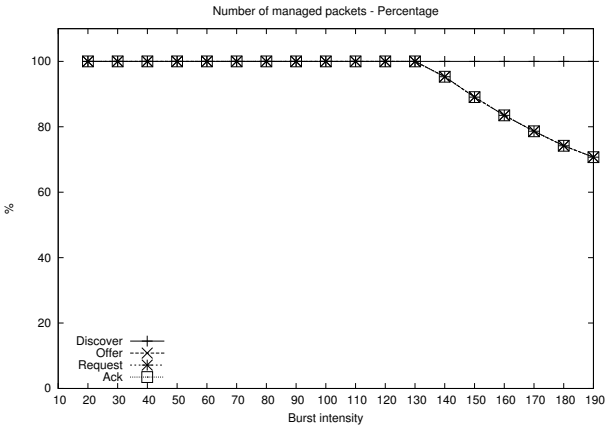

Fig. 9. Percentage of DHCP packets sent and received by client side with ISC DHCP Server.

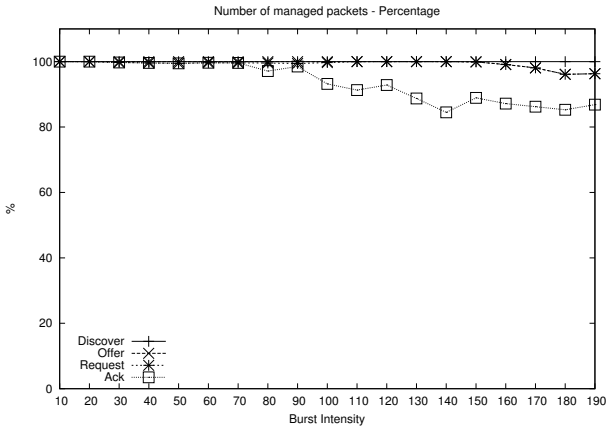

Fig. 10. Percentage of DHCP packets sent and received by client side with Uni-Fy DHCP Built-in.

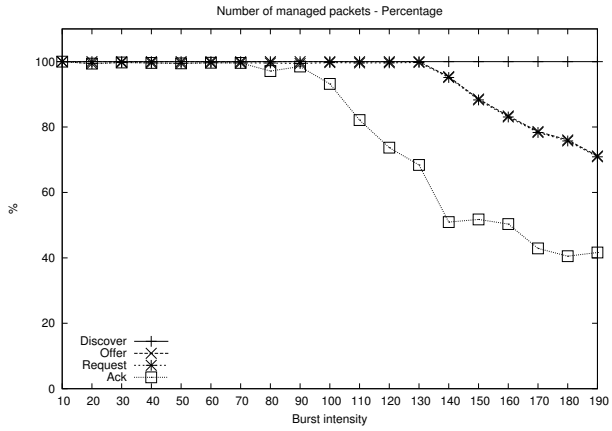

Fig. 11. Percentage of DHCP packets sent and received by client side with Uni-Fy DHCP Relay Agent.

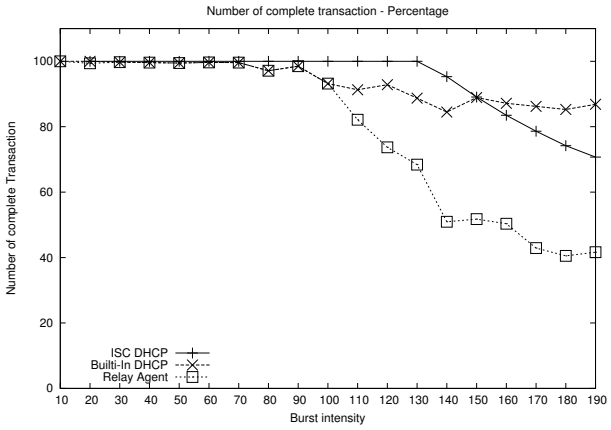

Fig. 12. Percentage of correct transaction normalized to the number of sent DHCPDiscover packets.

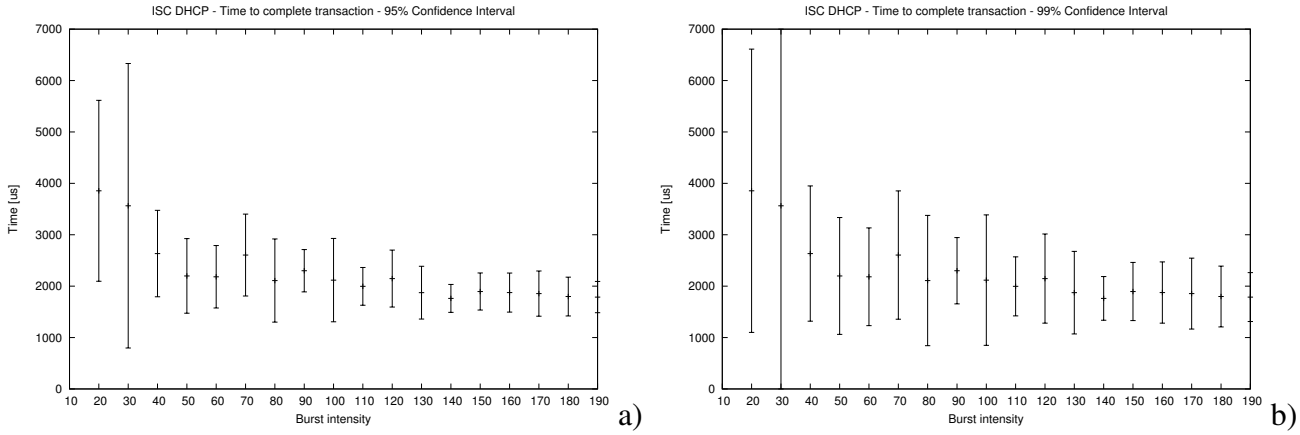

Fig. 13. Required time for complete transactions with ISC DHCP Server. Confidence level: 95% a) and 99% b)

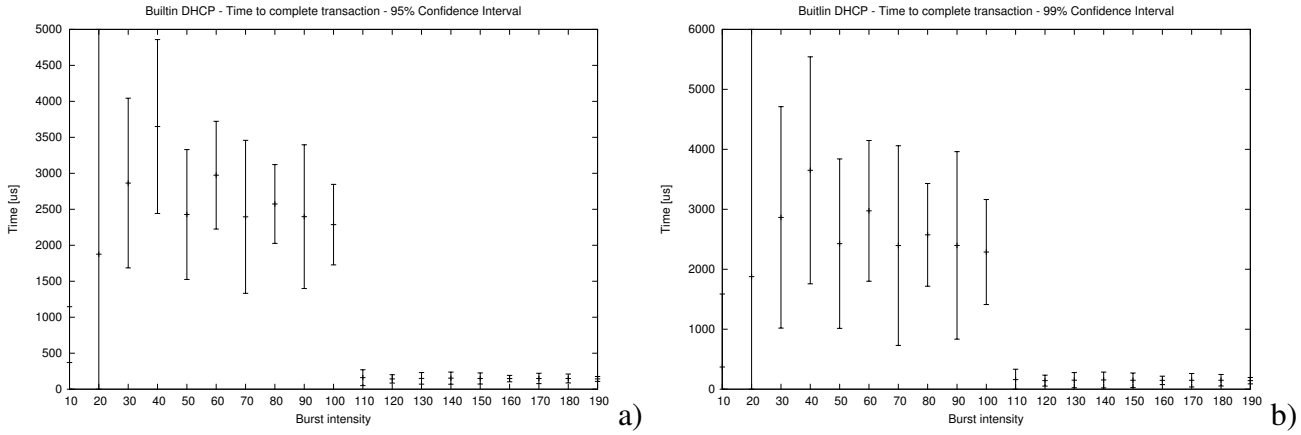

Fig. 14. Required time for complete transactions with Uni-Fy Built-in DHCP. Confidence level: 95% a) and 99% b)

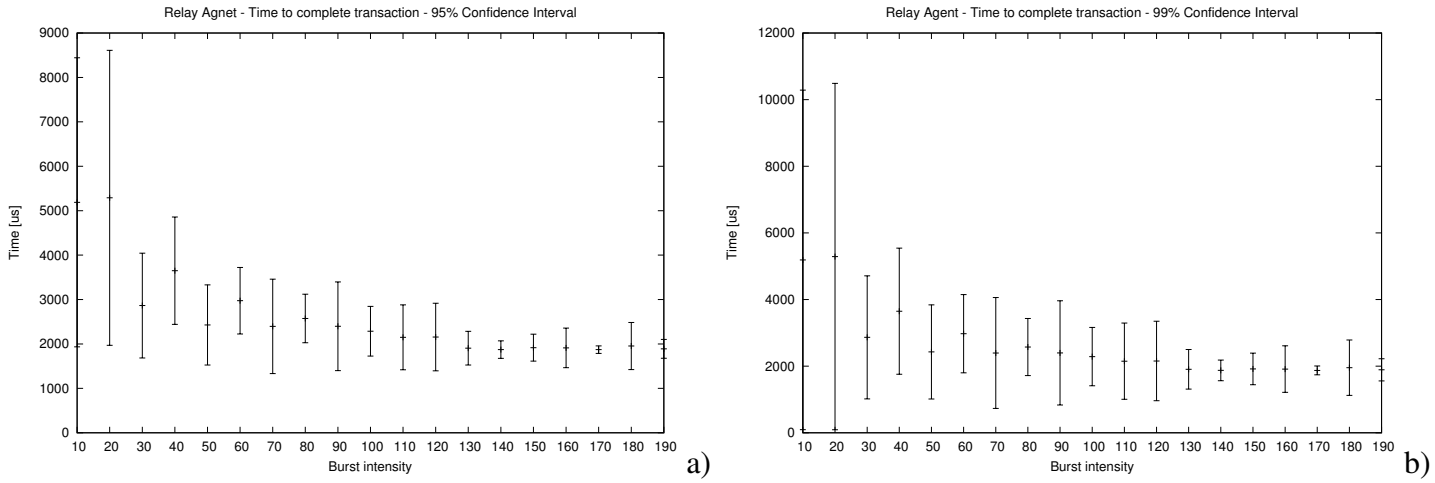

Fig. 15. Required time for complete transactions with Uni-Fy Relay Agent. Confidence level: 95% a) and 99% b)

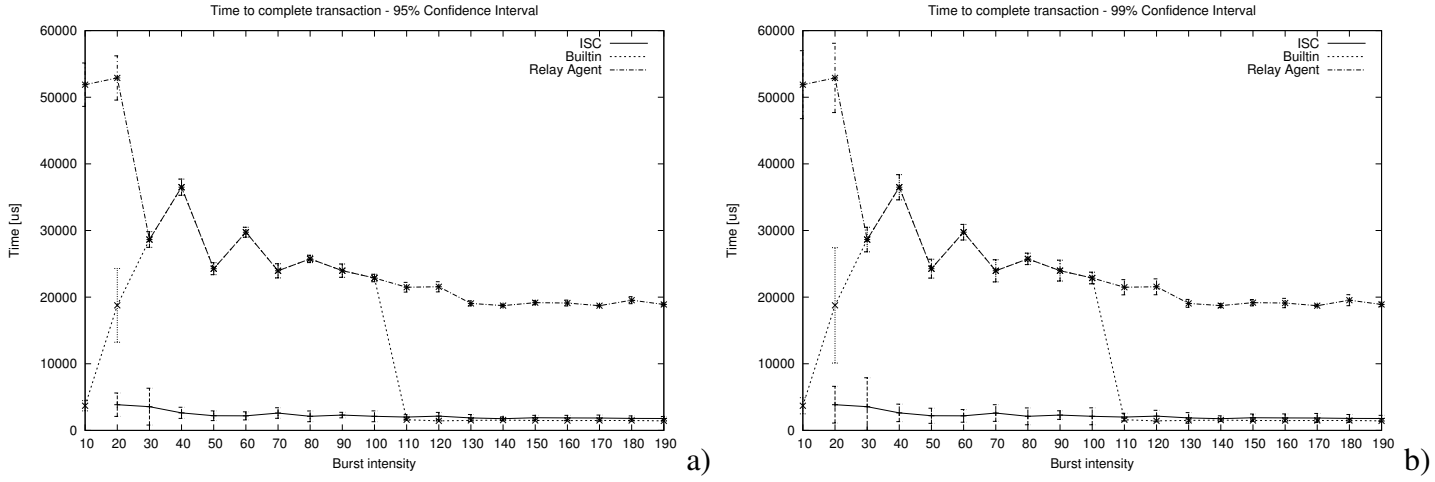

Fig. 16. Comparison among required time for complete transactions in the three tested scenario. Confidence level: 95% a) and 99% b)

V. Conclusion

TBD

References

[1] R. Battiti, M. Brunato, R. Lo Cigno, D. Severina, A. Villani, "WilmaGate: An Overview", available at http://netmob.unitn.it/files/WilmaGate-Overview.pdf

[2] RFC 951, "Bootstrap Protocol," *IETF Request for Comments*, September 1985.

[3] RFC2131, "Dynamic Host Configuration Protocol," *IETF Request for Comments*, March 1997.

[4] A. Scalet, "Implementazione ed analisi dielle prestazoini di un DHCP RELAY AGENT," Thesis, DIT - University of Trento, July 2006.

[5] M. Hedenfalk, *Access Control in an Operator Neutral Public Access Network*, MSc thesis, KTH/IMIT, Stockholm, May 2002,

[6] R. Battiti, R. Lo Cigno, M. Sabel, F. Orava, B. Pehrson, "Wireless LANs: From WarChalking to Open Access Networks," *Mobile Networks and Applications*, Vol. 10, 2005, pp. 275–287, Springer Science

[7] M. Brunato, R. Lo Cigno, D. Severina, "Uni-Fy: An Overview", available at http://twelve.dit.unitn.it/tools

[8] M. Brunato, D. Severina, "Uni-Fy: Code Overview and Product Manual", available at http://twelve.dit.unitn.it/tools

[9] ISC DHCP server, Free inplementation of DHCP server, Internet System Consortium http://www.isc.org/index.pl

[10] Globber, DHCP DoS attack http://globber.sourceforge.net/