An Improved Buddy System Auto-Configuration Protocol for Mobile Ad Hoc Networks

Julián García Matesanz[#1], Luis Javier García Villalba[*2], Ana Lucila Sandoval Orozco[*3] and
José René Fuentes Cortez[*4]

[#]*Grupo de Análisis, Seguridad y Sistemas (GASS)*
Sección Departamental de Sistemas Informáticos y Computación – LSI y CCIA –
Facultad de Ciencias Matemáticas, Despacho 310-F
Universidad Complutense de Madrid (UCM)
Plaza de Ciencias, 3
Ciudad Universitaria, 28040 Madrid, Spain
[1] julian@sip.ucm.es

[*]*Grupo de Análisis, Seguridad y Sistemas (GASS)*
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431
Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases s/n,
Ciudad Universitaria, 28040 Madrid, Spain
[2] javiergv@fdi.ucm.es

[3] asandoval@fdi.ucm.es

[4] jrfuente@fdi.ucm.es

***Abstract* - Mobile ad hoc networks (MANETs) are multihop wireless networks of mobile nodes without any fixed or preexisting infrastructure. The topology of these networks can change randomly due to unpredictable mobility of nodes and propagation characteristics. In most networks, including MANETs, each node needs a unique identifier to communicate. This work presents a distributed protocol for dynamic IP address assignment to nodes in MANETs. Nodes of a MANET synchronize from time to time to keep a record of IP address assignment in the entire network and detect any IP address leaks. The proposed stateful autoconfiguration scheme uses the OLSR proactive routing protocol for synchronization and guarantees unique IP address under a variety of network conditions including message losses and network partitioning. Simulation results show that the protocol incurs low latency and communication overhead for an IP address assignment.**

***Index Terms* — Mobile ad hoc network, MANETs, IP address assignment, autoconfiguration, dynamic host configuration, stateful protocol, synchronization, OLSR proactive routing protocol.**

I. Introduction

Mobile Ad hoc NETwork (MANET) is a set of mobile nodes which communicate themselves through wireless links. In contrast with the conventional networks, a MANET does not need a previous infrastructure, since nodes rely on each other to operate themselves, forming what is called multi-hop communication. Such networks have several disadvantages that a conventional network does not present.

The topology of this kind of network may change quickly and in an unpredictable way. Moreover, variations in the capacity of nodes and links, frequent errors in the transmission and security lack could occur.

Finally, limited resources of the nodes it must be taken into account since normally an ad hoc network will be formed by devices fed by batteries.

To communicate with each other [1] the ad hoc nodes need to configure their interfaces with local addresses which are valid inside an ad hoc network. The ad hoc nodes may also need to globally set routing addresses to communicate with other devices in Internet. From the perspective of the IP layer, an ad hoc network is presented as a multi-hop network of level 3 constituted by a link collection.

In an autonomous ad hoc mobile network the nodes can be uniquely identified across an IP address with the only premise that this address must be different than the one of any other node in the network.

The process of configuration is the set of steps through which a node obtains its IP address within the network. There are two mechanisms to set addresses: *Stateless* and *Stateful*.

The *Stateless* address configuration proposes its own node to be the one in changed of generating its IP address. The address is obtained from the concatenation of a network well-known prefix and the theoretically unique number inside the network generated by the node. This mechanism may require the inclusion of a module in responsible for verifying the uniqueness of the generated address called *Duplicate Address Detection* (DAD) [2-4].

On the other hand, the *Stateful* address configuration is based on using servers which control and assign addresses to all the nodes of the network. *Dynamic Host Configuration Protocol* (DHCP) [5] is an example of *Stateful* configuration. However, because of the multi-hop nature of the mobile ad hoc networks, this protocol cannot be applied directly.

This work proposes a *Stateful*-based auto-configuration protocol, which guarantees uniqueness of IP address under a wide variety of network conditions as message missing and network partitioning.

This work is structured in 5 sections; the first one is the present introduction. Section 2 shows the obligated references in the auto-configuration protocol scope of mobile ad hoc networks. Section 3 contains a brief specification so-called *Distributed Dynamic Host Configuration Protocol* (D2HCP), a proposal about IP address auto-configuration for Mobile Ad Hoc Network.

Section 4 presents the simulations about protocol D2HCP carried out in NS-3 [6]. Finally, Section 5 shows the main advantages of the developed new protocol as well as potential future extensions to the study.

II. Related Works

The mobile ad hoc networks present special features which must bear in mind when an address configuration protocol is implemented.

Many solutions exist for conventional networks (e.g.: RFCs 3315 [5], 4861 [7], 4862 [8] and so on) but the mobile ad hoc networks were not taken into account in its design. It is necessary, since, to give support multi-hop, support to dynamic topologies and to the *merging* and *partitioning* of networks, events that are typical in the mobile ad hoc networks.

There are numerous works that carry out proposals for the address configuration in a mobile ad hoc network using the *Stateless* as *Stateful* mechanism. Without doubt, the most representative are [2, 9-21].

Bernardos et al [22-24] carry out a rigorous study of the problems of the auto-configuration in mobile ad hoc networks, presenting an itemized review of the more representative auto-configuration protocols.

A comprehensive review of the main auto-configuration protocols can be found in [25].

Perhaps *Internet Engineering Task Force* (IETF) [26] has the most well-known work group called *Ad-Hoc Network Autoconfiguration Work Group* (Autoconf WG) [1] which its principal purpose is to describe the addressing model for ad hoc networks and how the nodes set its addresses in these networks. It is essential that such models do not cause problems to other components of an ad hoc system such as standard applications which are executed in an ad hoc node or Internet nodes connected to the ad hoc nodes. The work of this group can include the development of new protocols whether the existing IP auto-configuration mechanisms turn out to be inadequate. Nevertheless, the first task of this work group is to describe a practical addressing model for ad hoc networks.

The solutions described previously have supposed significant contributions to aid our comprehension of the problem. Nevertheless, we think that all these approaches handle only a subset of the network conditions enumerated as follows:

1. Dynamic Topology: The nodes in the network move arbitrarily and can join and leave the network dynamically.
2. Message loss and failure in the nodes: message loss can be quite frequent and can duplicate the IP address allocation if it is not managed correctly. The nodes can abruptly depart from the network due to a link failure or an accident.
3. *Partitioning* and *merging*: The network can split into multiple networks and, later, join with others. During network merging it is possible to have duplicated IP addresses in the fused network.
4. Address concurrent requests: Multiple nodes may want to join the network simultaneously.
5. Limited Energy and Bandwidth: The nodes in a mobile ad hoc network have limited energy and the links have a limited wideband. Therefore, the communication overhead which is incurred should be low.

In this work a solution similar to DAAP [27, 28] and to [29] that guarantees uniqueness in the IP address allocation under a wide set of network conditions is proposed. In our approach, the majority of address allocations imply local communication causing low communication overhead and low latency.

III. D2HCP (Distributed Dynamic Host Configuration Protocol)

The *Distributed Dynamic Host Configuration Protocol* (D2HCP) is an auto-configuration protocol that manages the joining and departing of nodes in MANET.

The protocol makes the MANET nodes collaborate with each other to manage the assignment of unique and correct IP addresses in a distributed manner. All the network nodes have the same role; there is no special type of node that centralizes the management of the same.

Nodes have a synchronization system is based on the OLSR [30] routing protocol. Thanks to this mechanism, the synchronization is done passively, monitoring the mentioned routing protocol, thus no overhead is generated in the network traffic compared to that one generated by the OLSR protocol.

Due to the fact that all the nodes are responsible for managing the joining of any new node to the network, this process can be done quickly. A node that wishes to join a network tries to contact any node still belonging to it, and may receive several responses from multiple nodes. This makes the chances of successfully joining the network high, because of the high availability and redundancy that the distributed management disposes.

Here we introduce the D2HCP specification: it begins with the used data structures, continues with an explanation of the exchanged messages between nodes for join and departure of these is continued, and then details how synchronization takes place in protocol is detailed. Finally, we explain the exchanged messages format during the auto-configuration process, detailing how to solve the possible message loss in the network using appropriate timers and performing certain actions when they expire to restore the auto-configuration process, as well as state diagrams for each operation mode that can take a node.

A. Data Structures

The data structures of this protocol can be classified into those handling the auto-configuration mechanism and those belonging to the OLSR routing protocol.

OLSR internally stores a routing table which is updated periodically. This table contains information about the route to each node, stored in the following fields:

- R_dest_addr: IP address of the destination node.
- R_next_addr: IP address of next hop in the route.
- R_dist: Distance to the destination node.
- R_iface_addr: IP address of the outgoing interface to the destination node.

The structures necessary for auto-configuration are:

- IP addresses of the node interfaces.
- Netmask.
- Free_IP_Blocks: A table of free block from each node in the network.

B. Joining and Departing of Nodes

1) Node Join: The protocol uses a specific message number for each operation. All the operations are defined looking for optimum working and low latency.

This section discusses how communication is established between the nodes and the messages transmitted during the joining and departure of nodes in the network.

The entry of a node to the network implies the need to find a node acting as a server. Once found, it will facilitate the joining by providing an IP address block and a table Free_IP_Blocks representing the state of all the nodes in the network.

Until the node does not have an assigned IP address, its communication with nodes which might act as servers will be through the MAC layer.

The configuration mechanism uses 4 types of messages in most cases. If no nodes in range with free IP addresses will be used 6 types of messages in total.

1. *SERVER_DISCOVERY*: The client node wishing to join a network starts the process with a message of this type. It is transmitted by the MAC layer, with the *broadcast* address as its destination. The message indicates the IP address number which is required (equal to the interface number).

   If the node has more than one network interface, the message is transmitted through all of them, using the ID field thus the different interfaces are not confused with several nodes.

2. *SERVER_OFFER*: The network nodes receiving the message *SERVER_DISCOVERY* reply to this message, also using the MAC layer, in which an IP address number is offered. The number of addresses offered is half of the available range.

   The *SERVER_DISCOVERY* message includes a field Count indicating how many attempts have been made by the client. Depending on its value, the server nodes will behave as follows:

   ∗ *Count* = 1: The server node will respond with a *SERVER_OFFER* if enough addresses are available and the fields R (*Ready*) and L (*Local*) will have the value 1 (it can assign the addresses provided at the moment, and they are addresses of block of own node).

   ∗ *Count* = 2: The node server will respond with a *SERVER_OFFER* if the fields can take the value R = 1 and L = 1. If not possible, it will still also respond if it is the case that there are enough addresses and R = 0, L = 1 (the server can not assign addresses at the moment, but it has them).

- *Count* > 2: If the node has addresses available and is in a capable state to do so, it will send a *SERVER_OFFER* with R = 1, L = 1. If it can, will send it with R = 0, L = 1. And finally, if it does not have enough free addresses, it will send the message with the fields R = 1, L = 0 (immediate availability of addresses, but the offered addresses are from another node in the network).

3. *SERVER_POLL*: After a time of hearing, the client node will have received several messages *SERVER_OFFER*. If not, it will try again.

It will sort received messages by the following criteria:

- The servers which are not available are discarded, i.e. with R = 0. The SERVER_OFFER with R = 0 is not used to reply with a SERVER_POLL, but they have the function of informing the client that there is a server node in the network although it cannot provide access to it at this moment.
- Priority to local addresses is given: it will prefer messages with the field L = 1.
- Finally, it is organized so that the offered address number are ranked, from highest to lowest.

According to this criteria for order preference, it will send a message *SERVER_POLL* to the first server (by the MAC layer, again) to let it know that the node has chosen this one to assign a free IP address block to it.

4. *IP_RANGE_REQUEST*. If the addresses provided by the server node were not their own, but they were from a third node in the network, with this message there will be a formal request made to that node. Since there is communication between two nodes already configured correctly, it is performed at the IP layer.
5. *IP_RANGE_RETURN*. The third network node authorizes the node that sends the message *IP_RANGE_REQUEST* to assign the address block indicated in this message to client nodes. It is also a message sent by IP.
6. *IP_ASSIGNED*. After receiving the *SERVER_POLL*, if the provided addresses were of the own server node, or after *IP_RANGE_RETURN* message if it has been necessary to request the address from a third node, the node server sends this message to the client. This message is transmitted by the MAC layer. In this message the free address block which is assigned to the client and *Free_IP_Blocks* table representing the network state is indicated. The table which is transmitted in this message does not reflect the joining of the client node.

After this message exchange, the client node chooses the first one of the block which has been assigned as its IP address. In the case of having more than one network interface, it will use the first ones of block in order, and will be the first of all which use as the primary address that identify the node.

2) Node Departure

The node departure mechanism does not require the exchange of any message. The node that wants to leave the network does not have to notify any other node of its departure, avoiding the overhead that these messages occur.

The other nodes in the network will become aware of the departure node through periodic updates of routes that OLSR protocol performs every so often. They will note that it has lost the path to that node, and therefore they removed it from its Free_IP_Blocks table, adding its free address block to the corresponding node as explained in the previous section.

C. Synchronization

The synchronization is done by monitoring the routing table of routing protocol OLSR [30]. The joining or departure of a node in the network is detected when OLSR adds a new route to its routing table, or deletes an existing one. By detecting the joining or departure of a node in the network, it is updated locally, and without exchanging any message, Free_IP_Blocks table.

For this reason, the following rules are obeyed:

- The responsibility of recovering the IP addresses that a node which leaving the network makes available is one that can be attached to the right of the free block. This will not be possible when the block to be collected contains the lowest address of the network. In that case, the node that picks the block up is one that can add to it by the left.
- By dividing the free addresses in two blocks to deliver one of them to a new node that joins the network, the node that acts as server delivers the sub-block, which does not contain its own IP address, to the client.

When the node departure is detected, its entry must be removed, and an update of the corresponding nodes, now available IP address must be recorded.

By detecting the joining of a new node, a new entry in the table for it is created, and the free address block of the node which supplied its IP address will be updated. To know who this node which acted as server was, it is simply necessary to find out which node has the IP address of the new node in its free address block.

IV. Simulations and Results

In the last network information discovery technique, the mobile node can consult a MIIS server, which stores information from several access networks and operators. To access this information, the MN must perform some steps before obtaining the desired information. It may require link-layer supports, transport protocol capability and security considerations. The main advantage of using such a technique is that the MN may have a complete and consistent view of the whole network. In addition, this approach allows MN mobility over several networks and operators. In this work, we will use this approach to demonstrate the benefits of use MIIS server technique in a heterogeneous mobile network environment.

A. Simulation Scenarios

Table 1 summarizes the main parameters used during simulations.

When performing these simulations has been remained constant the entries number in the network per unit time. This factor is important, particularly in high density networks.

Table I.

Simulation Parameters.

Parameter	Value
Simulation Area	1500 m x 1500 m
Mobile Node Number	50 to 1600
Mobility Pattern	Random Waypoint (*setdest*)
Routing Protocol	OLSR
Node Range or Coverage	125 m
Simulation Number	10
Simulation Area	1500 m x 1500 m

B. D2HCP vs Thoppian-Prakash Protocol

Figure 1 shows a comparison of the latency of D2HCP versus Thoppian and Prakash's Protocol. It is noted that in the first case the latency is lower than the second and it is very regular too (in Thoppian and Prakash the latency grows exponentially when the number of nodes is high) allowing us to conclude that D2HCP improves the results of its predecessor.

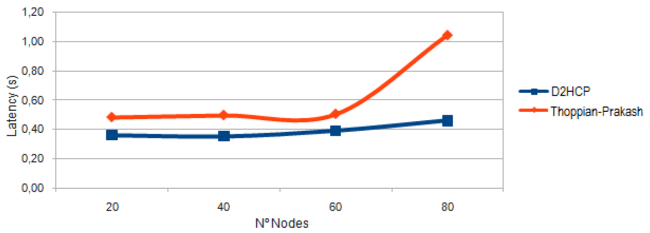

Fig. 1 D2HCP vs Thoppian-Prakash Protocol

V. Conclusion

An auto-configuration protocol for mobile ad hoc networks called D2HCP Distributed Dynamic Host Configuration Protocol (D2HCP) has been designed. This protocol is classified as a stateful protocol. This is an IPv4 address auto-configuration protocol for isolated mobile ad hoc networks. Each node is responsible for managing a range of addresses. When a new node wants to begin participating in the network, one of the nodes within the network gives half of its address range to the new node. In the case of any adjacent node not having free addresses, but free addresses do exists, a request to a network node that has free addresses is done. In this operation mode is based on distributed nature of the protocol.

To keep updated information about free addresses owned by each node, the traffic of control packets from OLSR protocol. Such protocol at each node tries to keep updated knowledge of the whole topology from the network. This protocol has been designed to work together with OLSR; although it could operate with any proactive protocol by the flexibility of its design.

D2HCP warrants uniqueness for IP addresses in a wide variety of network conditions including message loss, concurrent requests and network partition. The simulation results show that the protocol has low latency and overhead. Worth noting is the protocol scalability features compared to other proposals in the literature, its flexibility that facilitates the protocol extension with new features, as well as synchronization process introduces null overhead. Possible future work can be identified as follows:

- Detection of the merging to allow reassigning addresses that enters in conflict.
- Extension of the protocol to subordinate networks with access to the Internet or other networks, for which it should take into account the network topology to perform address auto-configuration process.
- Study protocol performance in cooperation with other proactive routing protocols.
- Add a security module that protects against different attackers to proportionate a safe auto-configuration.

ACKNOWLEDGMENTS

This work was supported by the Ministerio de Industria, Turismo y Comercio (MITyC, Spain) through the Project Avanza Competitividad I+D+I TSI-020100-2010-482 and the Ministerio de Ciencia e Innovación (MICINN, Spain) through the Project TEC2010-18894/TCM.

REFERENCES

[1]. Ad-Hoc Network Autoconfiguration Work Group (autoconf). Available online: http://tools.ietf.org/wg/autoconf/ (accessed on 25 November 2010).

[2] Perkins, C.E.; Malinen, J.T.; Wakikawa, R.; Belding-Royer, E.M.; Sun, Y. *IP Address Autoconfiguration for Ad Hoc Networks*; Internet Draft; November 2001.

[3] Weniger, K. PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 507-519.

[4] Weniger, K. Passive Duplicate Address Detection in Mobile Ad Hoc Networks. In *Proceedings of IEEE WCNC 2003*, New Orleans, Louisiana, USA, March 2003.

[5] Droms, R.; Bound, J.; Volz, B.; Lemon, T.; Perkins, C.; Carney, M. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*; RFC 3315; July 2003.

[6] The NS-3 network simulator. Available online: http://www.nsnam.org/ (accessed on 25 November 2010).

[7] Narten, T.; Nordmark, E.; Simpson, W.; Soliman, H. *Neighbor Discovery for IP version 6 (IPv6)*; RFC 4861; September 2007.

[8] Thomson, S.; Narten, T.; Jinmei, T. *IPv6 Stateless Address Autoconfiguration*; RFC 4862; September 2007.

[9] Cheshire, S.; Aboba, B.; Guttman, E. *Dynamic Configuration of IPv4 Link-Local Addresses*; RFC 3927; May 2005.

[10] Fazio, M.; Villari, M.; Puliafito, A. IP Address Autoconfiguration in Ad Hoc Networks: Design, Implementation and Measurements. *Comput. Netw.* **2005**, *50*, 898-920.

[11] Li, L.; Cai, Y.; Xu, X.; Li, Y. Agent-Based Passive Autoconfiguration for Large Scale MANETs. *Wirel. Personal Commun.* **2007**, *43*, 1741-1749.

[12] Kim, N.; Ahn, S.; Lee, Y. AROD: An Address Autoconfiguration with Address Reservation and Optimistic Duplicated Address Detection for Mobile Ad Hoc Networks. *Comput. Commun.* **2007**, *30*, 1913-1925.

[13] Nesargi, S.; Prakash, R. *DADHCP: Distributed Dynamic Configuration of Hosts in a Mobile Ad Hoc Network*; Technical Report UTDCS-04-01; University of Texas at Dallas, Department of Computer Science: Dallas, TX, USA, January 2001.

[14] Nesargi, S.; Prakash, R. MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In *Proceedings of IEEE INFOCOM 2002*, New York, USA, June 2002; pp. 1059-1068.

[15] Ros, F.; Ruiz, P.; Perkins, C.E. *Extensible MANET Auto-configuration Protocol (EMAP)*; Internet Draft; March 2006.

[16] Sheu, J.P.; Tu, S.C.; Chan, L.H. A Distributed IP Address Assignment Scheme in Ad Hoc Networks. *Int. J. Ad Hoc Ubiquitous Comput.* **2008**, *3*, 10-20.

[17] Hsu, Y.Y.; Tseng, C.C. Prime DHCP: a Prime Numbering Address Allocation Mechanism for MANETs. *IEEE Communications Letters*. **2005**, 9, 712-714.

[18] Kim, S.; Lee, J.; Yeom, I. Modeling and Performance Analysis of Address Allocation Schemes for Mobile Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*. **2008**, 57, 490-501.

[19] Chu, X.; Sun, K.; Sakander, Z.; Liu, J. Quadratic Residue Based Address Allocation for Mobile Ad Hoc Networks. In *Proceedings of IEEE International Conference on Communications (IEEE ICC 2008)*, Beijing, China, May 2008, pp. 2343-2347.

[20] McAuley, A.J.; Manousakis, K. Self-Configuring Networks. In *Proceedings of Military Communications Conference (MILCOM)*, Los Angeles, CA , USA, October 2002.

[21] Misra, A.; Das, S.; McAuley, A. Autoconfiguration, Registration and Mobility Management for Pervasive Computing. *IEEE Personal Communications*. **2001**, 8, 24-31.

[22] Bernardos, C.; Calderon, M.; Moustafa, H. *Ad-Hoc IP Autoconfiguration Solution Space Analysis*; Internet Draft; November 2008.

[23] Bernardos, C.; Calderon, M.; Moustafa, H. *Survey of IP Address Autoconfiguration Mechanisms for MANETs*; Internet Draft; November 2008.

[24] Bernardos, C.; Calderon, M.; Moustafa, H. *Evaluation Considerations for IP Autoconfiguration Mechanisms in MANETs*; Internet Draft; November 2008.

[25] García Villalba, L.J.; García Matesanz, J.; Sandoval Orozco, A.L.; Márquez Díaz, J.D. Auto-Configuration Protocols in Mobile Ad Hoc Networks. *Sensors*, **2011**, 11(4), 3652-3666.

[26] The Internet Engineering Task Force (IETF). Available online: http://www.ietf.org/ (accessed on 25 November 2010).

[27] Patchipulusu, P. *Dynamic Address Allocation Protocols for Mobile Ad Hoc Networks*. Master's Thesis, Texas A&M University: Dallas, TX, USA, August 2001.

[28] Mohsin, M.; Prakash, R. IP Address Assignment in a Mobile Ad Hoc Network. In *Proceedings of Military Communications Conference (MILCOM)*, Anaheim, California, USA, September 2002; Volume 2, pp. 856-861.

[29] Thoppian, M.R.; Prakash, R. A Distributed Protocol for Dynamic Address Assignment in Mobile Ad Hoc Networks. *IEEE Trans. Mob. Comput.* **2006**, *5*, 4-19.

[30] Clausen, T.; Jacquet, P. *Optimized Link State Routing Protocol (OLSR)*; RFC 3626; October 2003.