The Effect of DAD timeout period on Address Auto-configuration in Wireless ad-hoc networks

Murimo Bethel Mutanga[1], Pragasen Mudali[2], Tarirai Chani[3], Martin Mhlanga[4], Matthew O. Adigun[5]
Department of Computer Science
[1,2,4,5]University of ZULULAND, P. O. Box X1001, KwaDlangezwa 3886,
South Africa
Tel: +27 35 9026706
and [3]School of Information Technology
MONASH University, South Africa,
Tel +27 (0)11-950-4038
email: {bethelmutanga[1], pmudali[2], tariraichani[3], martinmafan[4], profmatthewo[5]}@gmail.com

Abstract- Lack of manual management mechanisms in wireless ad-hoc networks means that automatic configuration of IP addresses and other related network parameters are very crucial. Many IP address auto-configuration mechanisms have been proposed in literature. These approaches can be categorized as either being stateful or stateless. Stateless protocols employ a Duplicate Address Detection (DAD) mechanism during the auto-configuration process. Using this mechanism, new nodes generate their own IP address and broadcast a request packet and set a timer (DAD timeout). When the DAD timeout expires before any node using the requested IP address responds, the new node configures itself. A long period might result in unnecessary delays whilst, a short delay may result in duplicate addresses. Some solutions resort to repeating DAD for two or three times to guard against message losses that might result in assigning duplicate addresses. In this paper we assess the effect of DAD timeout period on the Duplicate Address Detection. This paper also attempts to get the optimum DAD timeout period.

Index Terms—auto-configuration, IP address, wireless ad-hoc networks

I. INTRODUCTION

The phenomenon of self organization that has its roots in biological and eco-systems is spreading in many areas of our lives. A typical example of the emergence of self-organized functions is in the field of IP address allocation. This trend will be further accelerated by the advent of ubiquitous computing, where wireless technologies interconnect an increasing number and diversity of devices [1]. This leads to increased complexity, which might become a stumbling block for further development. It is clear that a higher level of self-organization will help us to master these challenges. High levels of self-organization will help reduce administration efforts and costs.

IP address automatic configuration in wireless ad-hoc networks is one area that has received a lot of attention in recent years. Automatic configuration of hosts makes deployment and management of networks easy. So many solutions to address this issue have been proposed in literature. Basically these solutions can be categorised as stateless, stateful or hybrid. Another solution that has been considered to address the address auto-configuration problem is constructing a unique address from the Medium Access Control (MAC) address. For example, IP version 6 allows the construction of an IP address using the MAC address. A major concern with this idea is the issue of location privacy [2]. Automatic configuration using random numbers is therefore a viable solution to this problem but however, this approach has to cope with a highly dynamic network environment [3]. Some works have reported that MAC addresses can be duplicated. The work in [2] reports that there are instances of network adapters with unregistered or duplicate MAC addresses on the market, and also that some network adapters allow users to change the MAC address. It is thus clear that automatic configuration is the solution to this problem.

In stateful solutions, free IP addresses are known in advance. One or some of the nodes in the network maintain and synchronize state information. New nodes will have to rely on an already configured node to obtain a free IP address. Stateless approaches, do not maintain any state. New nodes generate their own IP address and broadcast a request packet and set a timer (DAD timeout). When the DAD timeout expires before any node using the requested IP address responds, the new node configures itself. If the DAD timeout period is not long enough, the new node configures itself before the node using the requested address can respond. On the other hand, if the DAD timeout period is unnecessarily too long, a long delay might be experienced because the new node will only be able to configure itself after the timeout has expired. If the network size is big, message losses and delays are bound to occur. An increase in network size is bound to have an effect on the effectiveness of the DAD timeout period.

This paper investigates the optimal DAD timeout period and further tries to establish the relationship between DAD timeout period and the network size. Results of this work will aid in the design of better IP address auto-configuration protocols.

The rest of the paper is organized as follows: Section II gives a brief literature review and background of automatic configuration and Duplicate address detection in wireless ad-hoc networks whilst section III outlines the setup of our

experiments. Section IV presents the simulation results whilst section V concludes this paper.

II. AUTOMATIC CONFIGURATION IN WIRELESS AD-HOC NETWORKS

IP address assignment approaches for ad hoc networks can be classified into two distinct categories namely stateless and stateful paradigms. Some schemes with characteristics of both stateful and stateless also exist under the umbrella term of hybrid approaches. The stateless auto-configuration paradigm is also referred to as conflict detection approach since protocols that follow this approach use a trial and error method to obtain a free address. On the other hand, the stateful paradigm is also known as conflict free paradigm since nodes that take part in the address assignment procedure allocate IP addresses that are known to be free in the network. In this paper we review the protocols under the stateless paradigms since DAD is the fundamental aspect of these protocols. However, some stateful approaches also employ DAD for detecting IP address conflicts.

Stateless protocols do not maintain any allocation table. The nodes generate their own IP addresses and check for possible conflicts through a Duplicate Address Detection procedure, hence most of the research in this approach is aimed at optimizing the DAD mechanism. If a conflict is detected, the new node will repeat the process. Because of this, Duplicate Address Detection is the cornerstone of the stateless paradigm. Generally, the DAD process is categorized as being either *Strong DAD* [4] or *Weak DAD* [5]. Strong DAD is a time-based DAD that checks if there is an address conflict in a network within a finite bounded time interval. Strong DAD configures nodes after the DAD procedure has been successfully completed or after a specific time interval (DAD timeout period). Weak DAD is used for the purpose of detecting IP address conflicts by making use of a key-address combination that must always match if there is no conflict in the network. When a node receives a routing control packet, it compares the address and key contained in the packet with those that appear its routing table.
A weak DAD is usually termed optimistic DAD since it configures the new node before the DAD procedure is complete. It assumes that the DAD procedure will be successful hence the name optimistic DAD. Even if the DAD is not successful, unicast communication can still take place without any problems since the nodes use the key-address combination to identify the origins or destination of a packet. In this work, we concentrate on the time based DAD because Weak DAD does not makes use of the DAD timeout period.

In Strong-DAD [4], a node randomly selects an IP address and checks whether or not it is used in the network using a DAD procedure. In fact a new node chooses two addresses: a temporary address and the actual address to use. During the IP address negotiation process, new nodes use temporary IP addresses. The temporary address is not verified for uniqueness. The network is flooded with an address request (AREQ) message containing the selected address. A node using the requested address defends it by replying with an address reply (AREP) message. If the address is currently in use, the process is started again until a free IP address is obtained. An address is assumed to be free if the timer for a DAD trial expires before receiving a conflict notification message. In [6], Strong DAD was tested using a DAD timeout period of 1.8 seconds and was seen to result in latency of more than 5 seconds. A total of 3 DAD trials were also used to guard against message losses.

Other protocols that used Strong DAD include AIPAC [6], and AROD [7]. In AIPAC new node periodically broadcasts a SendRequest message until a reply is received from at least one neighboring node (initiator). The initiator selects an address at random among the allowed addresses and sends in broadcast, a Search_IP packet. The address selected is specified in the packet. Any node receiving this packet checks whether this address belongs to it or to another node in its routing tables. If a match is detected, the node sends a Used_IP message to the Initiator. When the Initiator receives the Used_IP message, the procedure is restarted, and a new address is selected. Conversely, if no reply is received for a given time interval (DAD timeout of 1.8 seconds), the Initiator sends the Search_IP packet again (2 DAD trials) , in order to face up possible errors in wireless channels. If neither replies arrive, it means that the address is not used yet. Then the Initiator notifies the Requester with the NetID of the network and the IP address that it has to use.

In Wise-DAD [8] nodes keep state information but still performs DAD before a new node is admitted. The new node selects only one of its neighbors node to act as its negotiating agent (initiator). The initiator then generates a random IP address from the allowed addresses and checks its allocation table if there is no node in the network that has requested for or used the same IP. If the address is not known, the initiator then performs a DAD (using an address request message). All nodes receiving an address request packet update their tables and add their IP addresses to the packet before broadcasting it. Allocation tables are not actively synchronized; they are used only as an estimate of the state information. The DAD timeout used in Wise-DAD is 1.8 seconds and only one DAD trial is utilized since there is an estimate of the state information to check for address duplicates before DAD is performed.

III. EXPERIMENTAL SETUP

A. The design of DAD

We consider a DAD procedure similar to the one proposed in [4] with a slight modification. We introduce the concept of initiator and requestor proposed in ManetConf [9]. This is to guard against two nodes using the same temporary IP address. When a new node wants to be part of the MANET, it sends a *Request to join* message to its immediate neighbours. The first neighbour to respond becomes the new node's initiator. The initiator replies with *initiator_available* message and the new node will send an *acknowledgemen*t message.
The initiator then chooses a random IP address from a predetermined range and broadcasts an *Address Request* message. Any node using the requested address will defend

its address by an *Address Reply* message to the initiator; otherwise it will just forward the message.

If no response is received after the set DAD timeout period, the initiator broadcasts the Address Request message again for a predetermined number of DAD trials to guard against time delays and message losses. If after the set DAD trials, no response is received, the initiator will send an *address_packet* to the new node. In a bid to establish the optimal DAD trials , we varied the number of the DAD trials in our experiments.

Handling of network merging is not within the scope of this paper since this paper only seeks to assess the effects of DAD timeout period on the performance metrics listed below.

B. Performance Metrics

In our simulation, the following performance metrics were used for comparison:

a) Latency

This refers to the average time taken for a node to be assigned an IP address. The address assignment process must be done in as minimum time as possible.

b) Communication Overhead

The average number of address assignment packets generated and forwarded by each node during the address assignment procedure. A good scheme should use as few messages as possible and the communication should be preferably local. Flooding should always be avoided.

c) Address duplicates

The number of duplicated IP addresses in the network. A good scheme should minimize the probability of having more than one node using the same IP address.

d) IP Conflict latency

The time required for a node to receive a conflict notification message if an address duplicate is detected.

IV. Simulation Results

A. Experiment 1 : The effect of DAD timeout period on the auto-configuration protocol

The purpose of this experiment was to assess the effect of different values of DAD timeout on latency, address uniqueness and communication overhead. The nodes were spread over a rectangular 2000m x 2000m flat area for 6000 seconds of simulation time. The simulation parameters for this experiment are shown in Table 1.

TABLE I: SIMULATION PARAMETERS FOR EXPERIMENT I

Parameter	Environment
Number of nodes	30, 60, 90, 120
DAD timeout (seconds)	0.1,0.2, 0.4 , 0.6 … 2
Node arrival rate	1 node / 30 seconds
Address Range	8-bit (256)
DAD trials	1
Simulation time	6000 seconds

a) Effect of DAD timeout period on latency

Fig 1 shows that the DAD timeout period is directly proportional to the length of the configuration process. This is due to the fact that configuration only takes place after the DAD timeout period has expired.

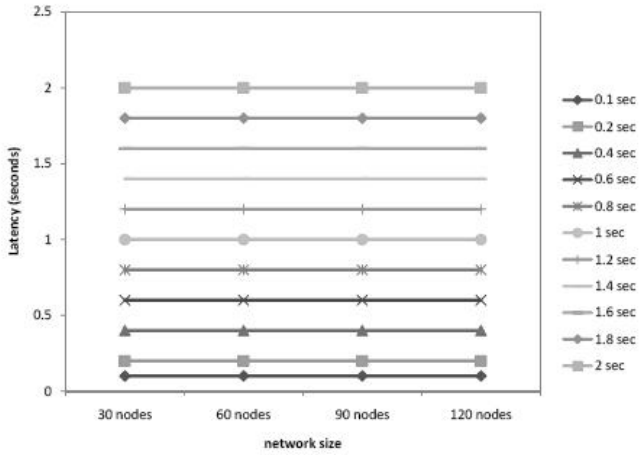

Fig. 1. The effect of DAD timeout on latency

b) Effect of DAD timeout on address uniqueness

The results obtained show that, the numbers of address duplicates are affected by DAD timeout period. Low values of DAD timeout period result in more address duplicates than larger values of DAD timeout. This can be attributed to the fact that some nodes were not able to defend their IP addresses before the DAD timeout expired, leading to address duplicates. However as the DAD timeout period was increased, nodes were able to defend their IP addresses hence address duplicates decreased. At DAD timeout of 1 second and above, the number of address duplicates did not change significantly except on the 120 node network. We conclude that at this value all nodes were able to defend their IP addresses although the same cannot be said for a 120 node network. Any value more than one second was therefore more than the required time for a node to defend its IP address.

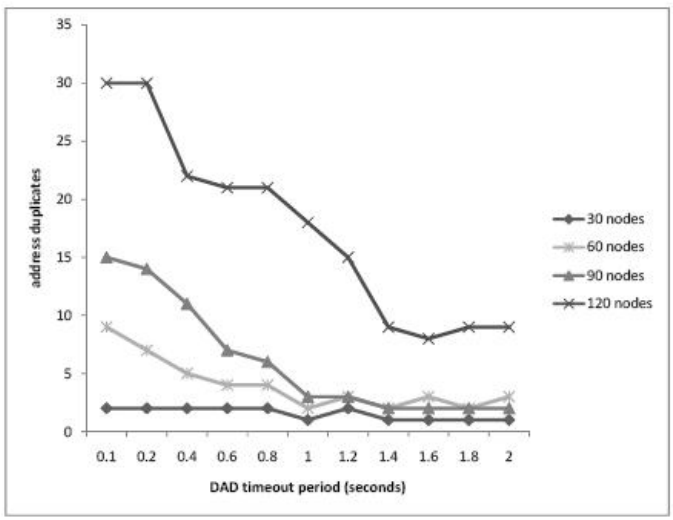

Fig. 2. The effect of DAD timeout on address duplicates.

c) Effect of DAD timeout on communication overhead

Fig 3 shows that communication overhead did not change significantly as the DAD timeout was varied. Interesting to note is the fact that at DAD timeout period of 1 second and above, communication overhead slightly increased. At the same value, Fig 2 also shows that address duplicates decreased. We conclude that the increase in communication overhead was due to the fact that nodes were able to defend their IP addresses hence Fig 2 showed a decrease in address conflicts.

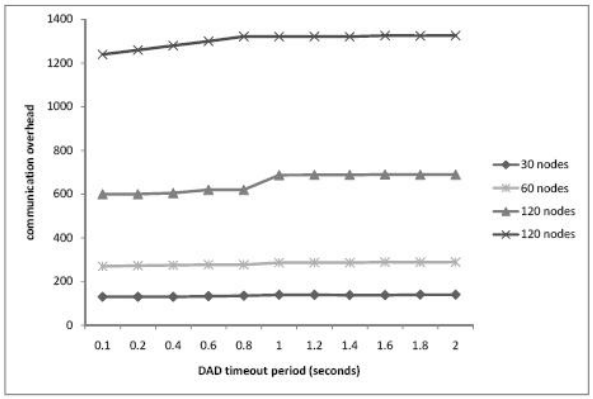

Fig. 3. The effect of DAD timeout on communication overhead

B. Experiment 2: Determining time required for conflict message delivery

The purpose of this experiment was to investigate the time that is required for an address conflict to reach the new node. We argue that the time required for an IP address conflict message to be delivered should be the minimum value that a DAD timeout should use. We created a node with a duplicate address and recorded the time that will be required to detect the duplicated address. We varied network size because different network sizes might result in different delivery times due to scalability issues. DAD timeout was set to a very high value to give enough time for the conflict message to reach the new node.

TABLE II: SIMULATION PARAMETERS FOR EXPERIMENT II

Parameter	Environment
Number of nodes	30, 90, 120
DAD timeout	5 seconds
Address Range	8-bit (256)

The results shown in Fig 4 show that the time taken for a conflict message to be delivered is at least 1 second. 120 nodes recorded slightly below 1.2 seconds in latency. These values help in determining the best value for DAD timeout period when designing an address auto-configuration protocol. From the results we can conclude that using a value which is less than 1 second will result in some nodes not able to defend their IP address. On the other hand, using a DAD timeout value that is more than 2 seconds will result in unnecessarily high latency.

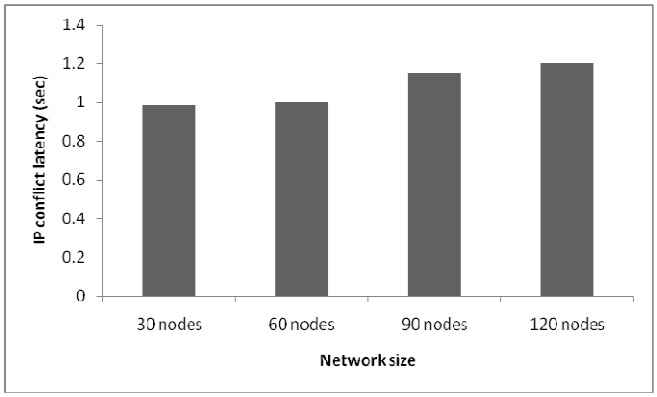

Fig. 4. Determining time required for conflict message delivery.

C. Experiment 3: Effect of DAD trials on the performance of the auto-configuration process

The purpose of this experiment was to assess the effect of the number of DAD trials on the performance of the auto-configuration protocol. Some protocols use varying DAD trials per requested address before a new node can configure itself. After a new node generates an IP address it sends a DAD message with the requested address and waits until the timeout period has expired. To guard against message losses, the new node will send the DAD message again even if it did not receive a conflict message. The number of times that the message is broadcast varies with protocols. In Strong DAD, three trials are used whilst in Wise-DAD, only one trial was used.

TABLE III: SIMULATION PARAMETERS FOR EXPERIMENT III

Parameter	Environment
Number of nodes	30, 90, 120
DAD timeout period	1 second
DAD trials	1,2,3
Address Range	8-bit (256)
Simulation time	6000 seconds

a) Effect of DAD trials on latency

Fig 5 shows that the number of DAD trials and latency were seen to be proportional to each other. This is due to the fact that each trial brings more delay hence the more the trials the more the latency.

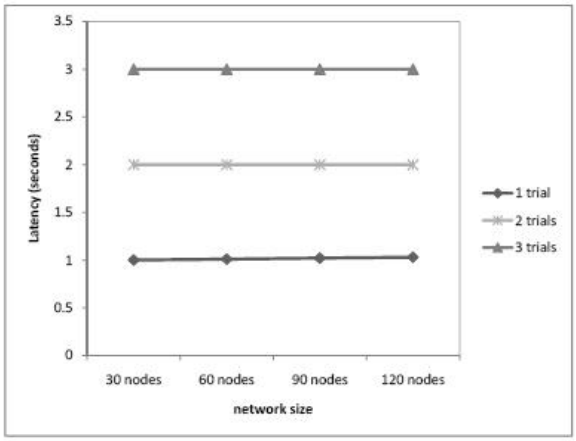

Fig. 5. The effect of DAD trials on latency

b) Effect of DAD trials on communication overhead

The communication overhead generated by the configuration process increased with DAD trials, this is due to the fact that each DAD trial generates its own traffic. However, the rate of increase of communication overhead is proportional to the number of nodes due to the broadcast storm problem.

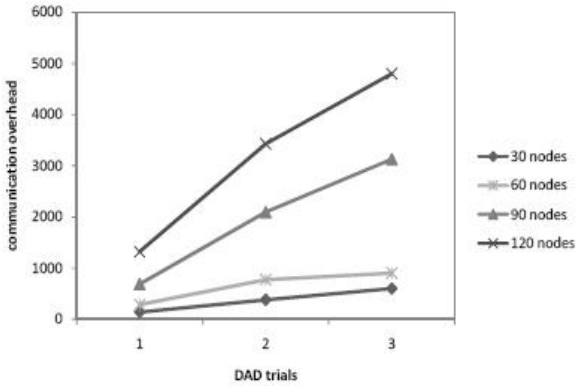

Fig. 6. The effect of DAD trials on communication overhead

c) Effect of DAD trials on address uniqueness

The results shown in Fig 7 show that address duplicates were not significantly affected by the number of DAD trials. This can be attributed to the fact that the DAD timeout period of one second that was used was is long enough for address conflicts to be reported as shown Fig 2 hence we can conclude that this is the optimal DAD timeout period. An increase in communication overhead at 1 sec that is shown in Fig 3 also suggest that more nodes were able to defend their IP addressed hence generating more packets.

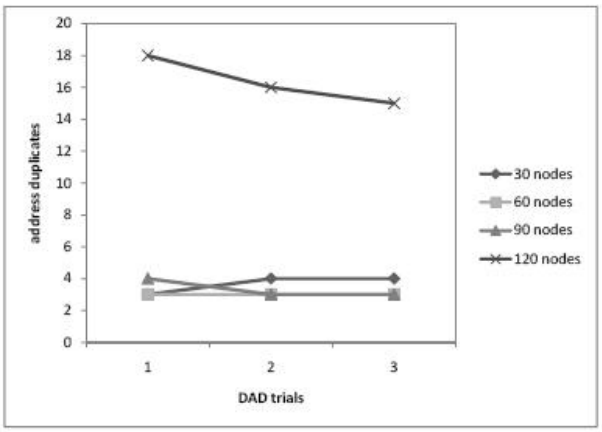

Fig. 7. The effect of DAD trials on address duplicates

V. Conclusion and Future Work

IP address auto-configuration is an area that has received attention in recent years. So many IP address auto-configuration protocols have been proposed in literature. These schemes can be categorized as either being stateless or stateful. Stateless protocols employ a Duplicate Address Detection (DAD) mechanism during the auto-configuration process to check for address duplicates. In this auto-configuration mechanism, new nodes generate their own IP address and broadcast a request packet and set a timer (DAD timeout). When the DAD timeout expires before any node using the requested IP address responds, the new node configures itself. However the optimal period of DAD timeout is not clear. A long period might result to unnecessary delays whilst, a short delay may result to duplicate addresses. Some solutions resort to repeating DAD for two or three times to guard against assigning duplicate addresses due message delays or losses.

In this paper we investigated the optimal configurations surrounding DAD. The investigation was an attempt to get the optimum DAD timeout period. We also investigate the relationship between DAD timeout period and network size.

Our results show that the DAD timeout period of 1 second is the optimal one. It is however imperative to test this timeout period on larger networks. DAD timeout period was found to affect both communication overhead and latency. The future focus of this research is developing a full auto-configuration protocol based on the results obtained in this work.

Acknowledgement

The authors appreciate the support given to the department of computer science at the University of Zululand by industry partners. Special mention goes to Telkom, THRIP, Huawei.

References

[1] Prehofer, C. and Bettstetter, C. (2005) “Self-organization in communication networks: principles and design paradigms” Communications Magazine, IEEE, Volume 43 Issue: 7 pp 78 – 85, July 2005

[2] Weniger, K. and Zitterbart, M. (2004). “Address Autoconfiguration in Mobile Ad Hoc Networks: Current Approaches and Future Directions”, IEEE Network Magazine Special issue on 'Ad hoc networking: data communications & topology control', Jul 2004.

[3] Fan, Z. and Subramani, S. (2005). “An address autoconfiguration protocol for IPv6 hosts in a mobile adhoc Network”, Computer Communications, Volume 28, Issue 4, pp: 339-350, March 2005

[4] Perkins, C. Malinen, T. Wakikawa, R. Belding-Royer, E. Sun, Y. (2001) “IP address autoconfiguration for ad hoc networks”, IETF Internet Draft 2001.

[5] Vaidya, N.H. (2002). “Weak Duplicate Address Detection in Mobile Ad Hoc Networks,” Proceedings of ACM MobiHoc 2002, Lausanne, Switzerland, pp: 206–216, June 2002

[6] Fazio, M. Villari,M. Puliafito, A. (2006) “AIPAC: Automatic IP address configuration in mobile ad hoc networks”, Computer Communications, Volume 29, Issue 8, pp 1189-1200, May 2006.

[7] Kim, N. Ahn, S. Lee, Y. (2007) “AROD: An address autoconfiguration with address reservation and optimistic duplicated address detection for mobile ad hoc networks”, Computer Communications, Volume 30, Issue 8, pp 1913-1925, June 2007.

[8] Mutanga, M.B. Nyandeni, T.C. Mudali, P. Xulu, S.S. Adigun, M.O. (2008). "Wise-DAD Auto-Configuration for Wireless Multi-hop Networks", In the proceedings of Southern Africa Telecommunication Networks and Applications Conference”, 7 -10 Sep 2008.

[9] Nesargi, S. Prakash,R. (2002) "MANETconf: configuration of hosts in a mobile ad hoc network", Proceedings of the 21st Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM 2002), New York, June 2002.

Murimo Bethel Mutanga received his undergraduate degree in 2005 from the Midlands State University in Zimbabwe. He holds a Master of Science degree in Computer Science from the University of Zululand and is presently studying towards his PhD degree at the same institution. His research interests include IP address auto-configuration, network forensics, Living Labs and telecommunications Law.