

Wise-DAD Auto-Configuration for Wireless Multi-hop Networks

MB Mutanga , TC. Nyandeni, P. Mudali, SS Xulu, MO Adigun
Department of Computer Science

University of Zululand, South Africa

Tel +27 35-9026706 Fax: +27 35-9026569

bethelmutanga, tcnyandeni, pmudali@{gmail.com}; ssxulu, madigun@ {pan.uzulu.ac.za}

Abstract— Providing unique IP addresses efficiently in ad-hoc networks is still an open research question. Many protocols to address this problem have been proposed. Duplicate Address Detection (DAD) is one of the most important components of IP address auto-configuration protocols. DAD by itself is not an auto configuration protocol but a means of checking for duplicate IP addresses before an address could be assigned to a new network node. The chance of a DAD procedure succeeding depends on the number of nodes in the network and the number of free IP addresses remaining. As the network size increases, the number of free IP addresses decreases hence the probability of a failed DAD process increases. In this paper we present Wise-DAD to address this problem. This goal was achieved by the introduction of passively updated state information. We evaluate the performance of our solution through simulation experiments.

Index Terms—Ad-hoc Network, IP address, Duplicate Address Detection, Auto-configuration

I. INTRODUCTION

A wireless multi-hop network allows nodes to communicate with each other without the required presence of a physical network infrastructure. The lack of network infrastructure and their ad-hoc nature means that auto-configuration is highly desirable in these networks.

The requirements for zero or auto-configuration in ad-hoc networking are classified under the following categories [1]: IP interface configuration, translation between host name and IP address, IP multicast address allocation and automatic service discovery.

The requirement for IP interface configuration brings about the problem of automatic, yet unique IP address allocation. Auto-configuration mechanisms should be capable of working both for IPv4 and IPv6 auto-configuration. An IP address is one of the most important network parameters as nodes require IP addresses to be able to communicate with one another.

Traditionally (in wired networks), an administrator configured computers manually. However this traditional approach can not be applied in ad-hoc networks. A first step toward automatic configuration was made with the

introduction of the Dynamic Host Configuration Protocol (DHCP) [2]. A dedicated computer called a DHCP server assigns IP addresses to the other computers hence the server must always be available. Due to characteristics of ad-hoc networks, one can not guarantee that the DHCP server will always be available.

As in wired networks, nodes in ad-hoc networks cannot take part in any type of communication unless they are configured with a unique IP address. Although routing protocols assume unique node IP addresses, the question of how these may be efficiently provided remains open. Nodes require a unique IP address for packets to be delivered to the correct destination, due to the routing side effects that may arise from nodes using duplicate addresses [3]. The autonomous nature of these networks requires the existence of an IP address auto-configuration mechanism or protocol. In ad hoc networks, such a mechanism has to cope with the highly dynamic environment and uncertain network structures [4]. A good IP address auto-configuration protocol should configure a node with an IP address whilst addressing the following goals:

Low latency for address assignment

Low communication overhead on Address Assignment

High probability of IP address uniqueness

However, these goals contradict each other e.g. an improvement towards address uniqueness is associated with high communication overhead and high latency.

In this paper we present Wise-DAD, an IP address auto-configuration protocol for wireless multi-hop networks. The idea behind Wise-DAD is to decrease the chances of DAD failing thereby reducing latency and communication overhead. This goal was achieved by the introduction of passively updated state information. Passively collecting state information reduces the number of DAD trials thereby reducing latency. Reducing communication overhead will also help conserve bandwidth thereby improving on QoS of the whole network. The rest of this paper is organized as follows. In Section II, related approaches in the area of IP address auto-configuration are described. In Section III, the basic idea behind our address auto-configuration method is outlined. In Section IV, the evaluation results of our protocol are presented. Finally, Section V concludes this paper and discusses possible future studies.

II. LITERATURE REVIEW

Address assignment paradigms or approaches for ad hoc

networks can be classified into two distinct categories namely *stateless* and *stateful* paradigms. Some schemes with characteristics of both approaches are also being developed under the umbrella term of hybrid approaches.

A. *Stateless Approaches*

Protocols following the stateless paradigm do not maintain any allocation table. An allocation table is a list of all IP addresses in use in a network at any given time. The nodes generate their own IP addresses and check for possible conflicts through a DAD procedure, hence most of the research classified under this approach is aimed at coming up with the most efficient DAD procedure. If a conflict is detected then the node will repeat the process, thus making DAD the cornerstone of the stateless paradigm.

In strong DAD [5], a node simply selects an IP address to use as its address, and checks whether or not it is used in a network using a DAD procedure. 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. During the IP address negotiation process described above, new nodes use temporary IP addresses for communication. The temporary address is not verified for uniqueness. As more nodes join the network, communication overhead increases since the number of DAD trials are likely to increase before a free IP address is obtained.

In [6] an Automatic IP address configuration (AIPAC) protocol based on strong DAD [5] was proposed. Since strong DAD does not provide a way for solving the problem of two nodes using the same temporary IP address, AIPAC uses the concept of Requester and Initiator, which is defined in [7]. The Initiator selects an address at random among the allowed addresses, and broadcasts a Search_IP packet. The address selected is specified in the packet. Any node receiving this packet checks whether the address is known (whether this address belongs to it or to another node in its routing table). If a match is detected, the node sends a reply Used_IP to the Initiator. When the Initiator receives the Used_IP message, a new IP address is selected and the process is repeated. Conversely, if no reply is received for a given time interval (Search_IP timer), the Initiator sends the Search_IP packet again, in order to face up possible errors in wireless channels. If neither reply arrives, 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. AIPAC does not specify how it handles the situation of more than one node requesting for the same IP address at the same time hence uniqueness in this scheme is not guaranteed. Also as in strong DAD communication overhead increases as more nodes join the network since the number of DAD trials are likely to increase before a free IP address is obtained.

To detect joining of two networks (network merging), AIPAC adopts the idea of a network ID (NetID) used in ManetConf [7]. The authors then proposed gradual merging when two or more networks with different NetIDs come in contact. Nodes switch from one network to another according to the evolution of the whole system.

B. *Stateful Approaches*

Protocols that follow the stateful paradigm assume that the addresses that are going to be assigned are not being used by any node in the network. This is achieved by guaranteeing that the nodes that participate in the allocation of IP addresses have disjoint address pools. In this case, performing a DAD is not necessary. Another way is to distribute the address allocation table to all network nodes so that they can configure new nodes since they know which IP addresses are free. This approach requires that the allocation tables be synchronized. In this case, a DAD is required to guard against a situation in which the same IP address is being requested for at the same time.

MANETConf [7] utilizes a distributed allocation table that needs to be maintained and actively synchronised the allocation table. A new node contacts an already configured node (initiator) for an IP address. The initiator then selects a free IP address from the allocation table and performs a DAD procedure. This is a way for checking whether the same address is being assigned in another part of the network. If all the nodes send a positive reply for this request, the address is assigned, and all the nodes of the network are notified otherwise the process is repeated until a free IP address is obtained. If a node leaves the network gracefully, it has to release the address, by broadcasting a bye message. All the other nodes update their allocation tables. However, if a node leaves abruptly, its IP address will eventually be deleted from the allocation tables since it won't be responding to subsequent address assignment procedures.

The main challenge of protocols utilizing a stateful approach with a distributed address allocation table is guaranteeing reliable state synchronization. If inconsistent states exist, they may result in address conflicts, an exhausted address space due to address leaks.

In Prophet [8], the authors proposed a novel approach that follows the stateful paradigm but the protocol does not store an allocation table. The basic idea is to predict the allocation table using a function $f(n)$ that is distributed among nodes. The first node chooses the function parameters hence it knows in advance all the IP addresses that are going to be allocated in that network. As other nodes join the network, the function $f(n)$ and a state value to generate a sequence of numbers are passed on to them so that they can also allocate new nodes with IP addresses. Unlike other stateful schemes, address reclamation is not necessary in prophet because the same address will reoccur in the sequence. However, this mechanism does not exclude the possibility of generating duplicate addresses. The authors say when a node is assigned an old address, X, the previous node with the same address (X) is likely to have left the network. This assumption is not practical because it assumes that generally the nodes leave the network using the order in which they joined the network (First in First Out). Despite this drawback, Prophet has a very low communication overhead and latency. This is mainly because assignment packets are exchanged locally.

C. Hybrid Approaches

Hybrid protocols combine elements of both stateful and stateless approaches. Protocols that follow this approach combine DAD with either a centrally maintained or a distributed common allocation table. Hybrid Centralized Query-Based Autoconfiguration (HCQA) protocol [9] utilizes strong DAD together with a centrally maintained allocation table. In HCQA [9], a node selects an address by itself and verifies its uniqueness with a DAD procedure. If the DAD is successful, it configures the address and registers with a dynamically elected Address Authority (AA), which inserts the new address in its allocation table.

The Passive Autoconfiguration for Mobile Ad Hoc Networks (PACMAN) protocol [10] combines PDAD [11] with a distributed maintenance of a common allocation table such as the one proposed in MANETConf [7]. In PACMAN a node assigns an address to itself using a probabilistic algorithm. Based on a pre-defined conflict probability, an estimation of the number of nodes and an allocation table, the algorithm calculates the size of the virtual address space, randomly selects an address from this space and ensures that the address has not already been assigned according to the local allocation table. The selected address is assigned immediately. Address uniqueness is not guaranteed although latency is properly addressed. Hybrid protocols normally inherit problems associated with the different approaches that they were derived from; for example in HCQA, if the AA crashes or leaves the network, a backup AA has to be elected just like in protocols that utilize a centralized allocation table like Centralised Auto-Configuration (CAC) [12]. In PACMAN, if many nodes join the network simultaneously, the allocation table may not be consistent and conflicts may occur. This is also the case with protocols that utilize a distributed common allocation table such as MANETConf [7]

III. THE WISE-DAD PROTOCOL

Wise-DAD was inspired by the notion that it is difficult to guarantee uniqueness of allocated addresses without DAD [6]. However a solution that decreases the chances of DAD failing in large networks is required to reduce latency and communication overhead. It is clear that such a solution will have to solve the problem of communication overhead and latency without compromising on address uniqueness. We divided the problem of IP address auto configuration and maintenance into to the following functions.

- *Network formation*
- *Node admission*
- *Node departure and address re-use*
- *Management of network merging*

A. Network formation

A new node broadcasts a request to join message and sets a timer (RTJ_Timer). The message is destined only to the immediate neighbours (one hop neighbours). When the timer expires (after 0.5 seconds), the node will rebroadcast the message and reset the timer again until it receives at least one reply from its neighbouring nodes. If two unconfigured nodes get in to each other's range, a network is formed. During this process an unconfigured node uses a temporary IP address, Host identifier (HID). Upon receiving a request

to join message, an unconfigured checks the HID of the sending node.

The node with the lowest HID chooses the Network identifier (NID) and sends it back to the other node together with an IP address that it chooses at random. The second node will also choose its own IP address and notify the other node of its chosen IP address. From that point, a network of two nodes starts to exist. If one of the configured nodes receives a request to join message, a reply with its availability to act as an initiator is sent. This process is explained in detail in the next section.

B. Node Admission

A configured node receiving a request to join message replies with a confirmation message to the sender. If a new node receives more than one confirmation message, it takes the first one and ignores the rest. The new node then replies with an initiator-selection message. The selected initiator then negotiates for an IP address on behalf of the new node.

The initiator generates a random IP address and checks if it is in its allocation table before it starts the negotiation process. If the address is in the allocation table, it generates another one otherwise it will broadcast an address request (AREQ) message and set a broadcast timer. If the broadcast timer expires (after 1.8 seconds) without any node defending the requested IP address, the initiator will send an address reply (AREP) to the new node.

On receiving an address request message other network nodes first check if the message is new or not before checking if the requested IP address has been assigned them. A message sequence number is used to determine if a message is new or not. If the address is found to be in use, an IP conflict is sent to the initiator and the process is repeated. If the message is not new, it is discarded, otherwise it will be broadcasted. Before the message is broadcast, the recipient adds its IP address to the message. As the message is passed from one node to another, a reverse path to the initiator will be contained in the packet. This allows for an IP conflict message to be sent back to the initiator. When nodes receive AREQ, they also update their allocation tables using IP addresses in the reverse path list before rebroadcast the AREQ.

C. Node departure

If a node departs gracefully, it informs the other nodes by broadcasting a goodbye message. Nodes receiving this message delete the departing node's address from their allocation tables. This allows for re-use of the address by new nodes.

On receiving this message other network nodes first check whether the message is new or not before it forwards the message. If the message is not new, it is discarded otherwise it will be broadcast. Before the message is broadcast, the node adds its IP address to the message. When nodes receive a goodbye message, they also update their allocation tables using IP addresses contained within the packet.

However node departures can be abrupt. In that case the departing node will not have the time to inform other nodes of its departure. If a node does not take part in any IP address assignment process, it is assumed to have left the

network and its IP address is deleted from address allocation tables. Its IP address will be tried in subsequent address assignment procedures.

D. Network Merging

The concept of network IDs used in MANETConf [7] was adopted to handle network merging in Wise-DAD. The first node in the network generates a random network identifier to be used by all the nodes in the network. Nodes periodically send one hop broadcast messages with their network identifiers. The network identifiers can be incorporated in the hello messages of the routing protocol. If a node receives a hello message with a different network ID, network merging is detected.

If a node receives a hello message with a network ID which is lower than its own, it initiates the network merging process by broadcasting a merger message to inform the other nodes of the impending merger. All nodes in the network with the lower network ID relinquish their IP addresses and start the process of IP address requisition afresh.

IV. SIMULATION EXPERIMENTS

Wise-DAD was implemented in Network Simulator-2 (ns-2) version 3.1 running on Ubuntu Linux 7.04 operating system with CMU extension of ns-2 to support ad-hoc networks [13]. We also simulated another stateless protocol, strong DAD [5] for comparison with Wise-DAD. In our simulation, we used DSR routing protocol. However, Wise-DAD is independent from the routing protocol used. We did not perform simulations in which nodes transfer data coming from the application layer, because we focused our attention in assessing the traffic generated by Wise-DAD independently from upper layers. The link layer model used in the simulation is based on the IEEE 802.11 MAC protocol. Experiments to analyse the effect of network size and the effect of node arrival rate were conducted.

A. Effect of network size on Wise-DAD

The aim of this experiment was to analyse the performance of Wise-DAD in different network sizes. Only one node was configured before starting the simulations and node positions were created by ns-2. We selected scenarios where every node could always communicate with the others during the entire simulation time. There were no node departures for the duration of the simulation since the aim of the experiment was to investigate the performance of Wise-DAD as network size increases. Table I below shows the other simulation parameters for the experiment.

TABLE I: SIMULATION PARAMETERS FOR EXPERIMENT I

Parameters	Environment
Number of nodes	30, 40, 50, ... 130
Area	1000m x 1000m
Simulation time	6000 seconds
Node arrival rate	1 node / 25 seconds
Address space	IPv4 class C (256)

1. Communication overhead

Fig. 1 shows the number of packet transmissions

according to the number of nodes. The result shows that the number of packets is in proportion to the number of nodes in both strong DAD and Wise-DAD. During a DAD procedure, Strong DAD floods a packet three times, and Wise-DAD only floods a packet once. That is why the number of packets recorded in strong DAD is more than the number of packets in Wise-DAD. Although both Strong DAD and Wise-DAD repeat the DAD procedure until the node succeeds in getting an unallocated address, the probability of a successful DAD in strong DAD is enhanced by soft state maintenance.

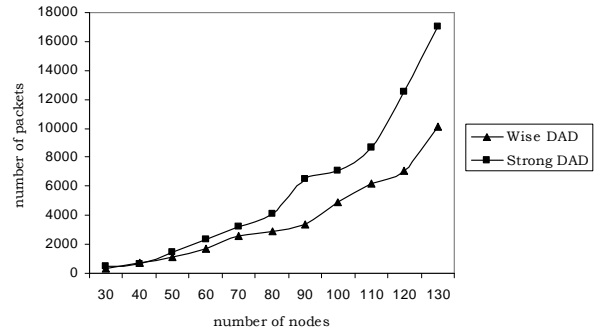


Fig.1. The effect of network size on communication overhead

2. Latency

Fig 2 shows a comparison of the average latency for address allocation according to the number of nodes. The timeouts to find an initiator for Wise-DAD was set at 0.5 seconds; and the timeout for address request (DAD procedure) is 1.8 seconds, which is calculated from the fact that the maximum hop count is 12 and the maximum one-hop round trip time is 0.15 seconds, thus the timeout must be at least 1.8 seconds [14]. Strong DAD took at least 5.5 seconds to obtain an address because it performed DAD procedures for at least three times.

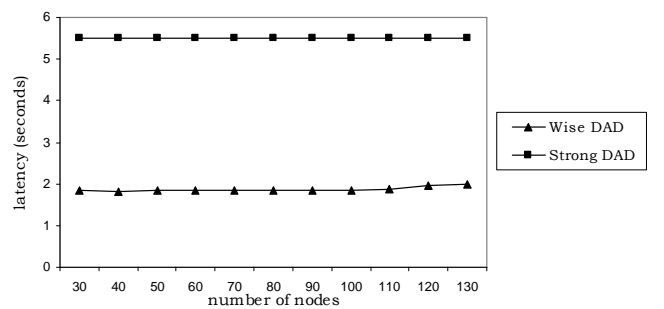


Fig.2. Effect of network size on latency

3. Address Uniqueness

One of the most important requirements in IP address auto-configuration is to guarantee the uniqueness of allocated addresses, since address conflicts may cause abnormal behavior of routing protocol and applications[8]. Fig. 3 shows the number of conflicting IP addresses according to the number of nodes. The number of IP address conflicts in both wise and strong DAD increases with network size because of the fact that the address space is a finite domain hence the probability of getting a free IP

address decreases as network size increase. The other reason could be message losses caused by MAC collisions as traffic increases (due to increase in network size) on the network.

If a class B address space was used, the number of IP address conflicts could decrease because of the large address space. Even without performing DAD, the probability of generating the same address more than once is very low. For example, if the number of nodes is 100, the probability is as low as 0.0015.

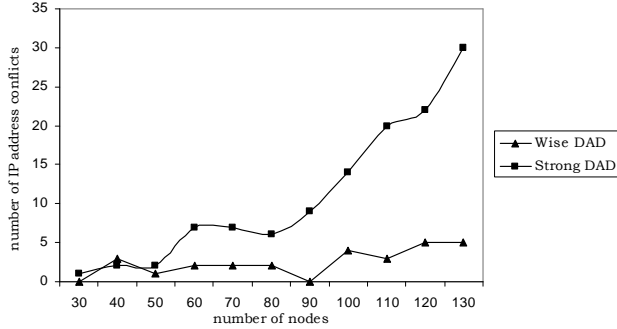


Fig.3. Effect on network size on address conflicts

B. Impact of node arrival rate

To show the impact of the node arrival rate, we varied the inter-arrival time of the nodes. The address space used was IPv4, class C with a maximum of 256 addresses, and the number of nodes was fixed at 80. Address uniqueness, communication overhead, and address allocation latency for strong DAD and Wise-DAD were recorded. Table II shows the other simulation parameters used in the experiment.

TABLE II: SIMULATION PARAMETERS FOR EXPERIMENT II

Parameters	Environment
Area	1000m x 1000m
Simulation time	6000 seconds
Node arrival rate	1 node every 5, 10, 15, 20, 25, 30 seconds

1. Address Uniqueness

Fig 4 shows that Wise-DAD did not show significant change in address conflicts as the rate of node arrival was varied. This is due to the fact that node admission helps in updating state information. Address conflicts in strong DAD on the other hand decrease as the rate of node arrival increases. This might be partially attributed to the fact the strong DAD does not provide a mechanism a situation whereby two or more nodes are requesting for the same IP address at the same time. High arrival rate are likely to results in such a situation. Message loses are likely to be another cause for high conflicts if node arrival rate is very high due to the fact that there is likely to be an increase in traffic in a short space of time.

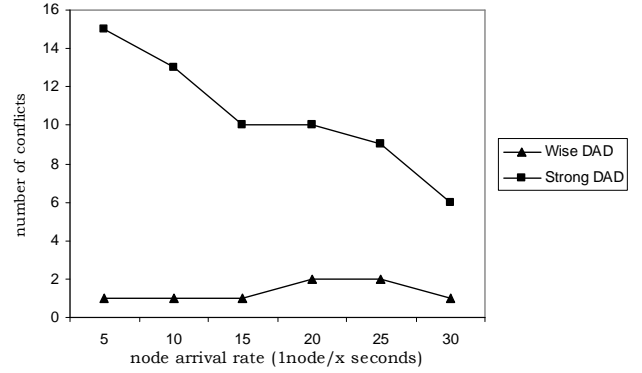


Fig.4. Effect of node arrival rate on address uniqueness

2. Communication Overhead

Fig 5 shows that communication overhead in both Wise-DAD and strong DAD is not affected by the rate at which nodes join the network. Although the number of packets is not constant, the variations are not sufficient to suggest that node arrival rate has an effect (positive or negative) on the traffic generated by both schemes. The number of packets sent by an initiator during an address assignment process depends only on the success of a DAD process. Node arrival does not have an effect on the success of DAD hence the communication overhead did not change as the node arrival rate was varied.

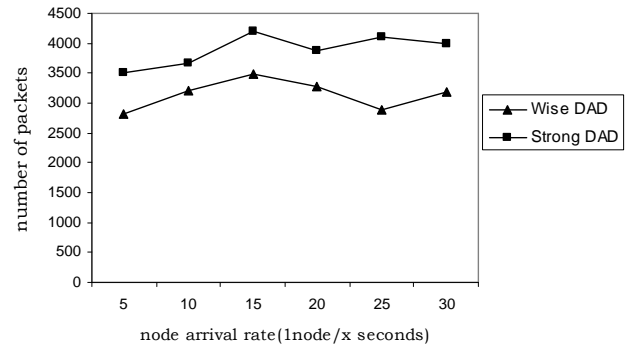


Fig.5. Effect of node arrival rate on communication overhead

3. Latency

In stateless approaches, address allocation latency is usually as affected by the number of DAD trials made. Fig 5 shows that node arrival rate does not have an effect on communication overhead hence the DAD trials were not affected by node arrival rate. As a result the latency shown in Fig 6 did not change as the node arrival rate was varied.

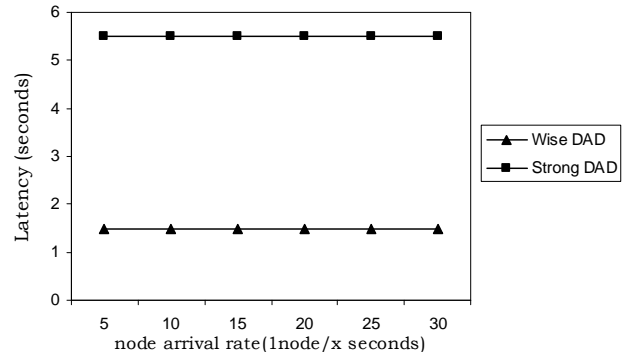


Fig.6. Effect of node arrival rate on latency

V. CONCLUSION

In this paper we presented the Wise-DAD protocol for node configuration in wireless multi-hop networks. The idea behind Wise-DAD was to decrease the chances of DAD failing thereby reducing latency and communication overhead. This goal was achieved by the introduction of passively updated state information. Simulation results show that Wise-DAD outperforms strong DAD in all the three metrics used for performance evaluation. The Wise-DAD protocol can handle network merging. Security will be the future focus of this work since it was not considered. All nodes were assumed to be authentic. However this assumption is not realistic. Address configuration schemes need a concrete solution for security since attacks on networks are becoming increasingly intelligent and more fatal.

ACKNOWLEDGMENT

The authors appreciate contributions from the Wireless Mesh Networks Group at the University of Zululand.

REFERENCES

- [1] A. Williams, Requirements for automatic configuration of IP hosts. Zero Configuration Networking, IETF Internet Draft, 2002.
- [2] R. Droms, Dynamic host configuration protocol, Network Working Group RFC 2131, March 1997.
- [3] S. Toner, D O'Mahony, Self-Organising Node Address Management in Ad-hoc Networks, in Springer Verlag Lecture notes in Computer Science 2775, Springer Verlag, Berlin, pp 476-483, 2003.
- [4] Z. Fan, S. Subramani, "An address autoconfiguration protocol for IPv6 hosts in a mobile adhoc Network", Computer Communications, Volume 28, Issue 4, , pp 339-350, March 2005
- [5] C. Perkins, T. Malinen, R. Wakikawa, E. Belding-Royer, Y. Sun, IP address autoconfiguration for ad hoc networks, IETF Internet Draft 2001.
- [6] M. Fazio, M. Villari, A. Puliafito, "AIPAC: Automatic IP address configuration in mobile ad hoc networks", Computer Communications, Volume 29, Issue 8, pp 1189-1200, May 2006.
- [7] S. Nesargi, R. Prakash, "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.
- [8] H. Zhou, L. Ni, M. Mutka, "Prophet address allocation for large scale manets", Ad Hoc Networks, Volume 1, Issue 4, pp 423-434, November 2003.
- [9] Y. Sun and E. M. Belding-Royer, "Dynamic Address Configuration in Mobile AdHoc Networks", UCSB tech. rep. 2003-11, Santa Barbara, CA, June 2003
- [10] K. Weniger, "PACMAN: Passive autoconfiguration for mobile ad hoc networks," IEEE J. Sel. Areas Commun., Special Issue on Wireless Ad Hoc Networks, vol.23, no.3, pp.507-519, March 2005.
- [11] K. Weniger, "Passive Duplicate Address Detection in Mobile Ad Hoc Networks" In IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, USA, March 2004.
- [12] M. Günes and J. Reibel, "An IP Address Configuration Algorithm for Zeroconf Mobile Multihop Ad Hoc Networks," Proc. Int'l. Wksp. Broadband Wireless Ad-Hoc Networks and Services, Sophia Antipolis, France, Sept. 2002.
- [13] K. Fall, K. Varadhan (Eds.), The ns Manual—the VINT Project, April 2008. Available: <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- [14] N.Kim, S Ahn. Y. Lee, "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.

Mutanga Murimo Bethel received his B.Sc. (Hons) in Computer Science from Midlands State University, Zimbabwe, in 2005. He is a Masters student in the Department of Computer Science at University of Zululand. Mutanga's research interests include automatic configuration and Cross-layer design in wireless multi-hop networks