Hierarchical auto-configuration addressing in mobile ad hoc networks (HAAM)

To cite this article: P. Ram Srikumar and S. Sumathy 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042017

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042017 doi:10.1088/1757-899X/263/4/042017

Hierarchical auto-configuration addressing in mobile ad hoc networks (HAAM)

Ram Srikumar. P, Sumathy. S*
School of InformationTechnology and Engineering, VIT University, Vellore-632014, India

*Email: ssumathy@vit.ac.in

Abstract. Addressing plays a vital role in networking to identify devices uniquely. A device must be assigned with a unique address in order to participate in the data communication in any network. Different protocols defining different types of addressing are proposed in literature. Address auto-configuration is a key requirement for self organizing networks. Existing auto-configuration based addressing protocols require broadcasting probes to all the nodes in the network before assigning a proper address to a new node. This needs further broadcasts to reflect the status of the acquired address in the network. Such methods incur high communication overheads due to repetitive flooding. To address this overhead, a new partially stateful address allocation scheme, namely Hierarchical Auto-configuration Addressing (HAAM) scheme is extended and proposed. Hierarchical addressing basically reduces latency and overhead caused during address configuration. Partially stateful addressing algorithm assigns addresses without the need for flooding and global state awareness, which in turn reduces the communication overhead and space complexity respectively. Nodes are assigned addresses hierarchically to maintain the graph of the network as a spanning tree which helps in effectively avoiding the broadcast storm problem. Proposed algorithm for HAAM handles network splits and merges efficiently in large scale mobile ad hoc networks incurring low communication overheads.

1. Introduction

Address auto-configuration protocols act as the backbone for Mobile Ad hoc Networks (MANET's) owing to the absence of centralized servers to coordinate the addressing of nodes. Each node must be uniquely addressed to establish communication among all nodes present in the network. Unlike IP address based networks such as IPv4 or IPv6, there are no servers running Dynamic Host Configuration Protocol (Droms, 1997), (Volz, 2006) which takes care of addressing all nodes. Nodes that are present within the wireless communication range of a particular node are able to communicate with it directly. Nodes that are not in direct communication range communicate through multiple hops along the available routes (Sharmila&Amalanathan, 2016). Nodes enter and depart at will in mobile ad hoc network, and do not follow any particularly organized pattern. There is no certainty in the continuous presence of a node which may split the network into smaller partitions and merge later. Such dynamic nature of the nodes denies them a chance to hold a permanent address to identify uniquely. Moreover, nodes have mobility which gives them the possibility to roam freely throughout the network. This causes the nodes to have different neighbours while they are in transit.

An address auto-configuration protocol needs to take care of assigning unique addresses to all nodes in the network, account for the dynamic nature of their presence in the network, resolve the

issue of nodes carrying duplicate addresses, and provide support for network partitions and merges. A novel address auto-configuration method is proposed in this paper which dynamically addresses nodes in a hierarchical pattern. In broadcast method of communication, each node has to employ neighbour discovery methods to identify nodes in its range and packets are forwarded to all the nodes in the range. Redundancy is high since a node has the possibility of receiving the same packet separately from different nodes in the range. A way to optimize the redundancy involved in broadcast communication by employing HAAM which establishes a spanning tree while addressing nodes, prior to applying any routing protocol for the network is proposed. Also, unlike other address auto-configuration protocols; this addressing method carefully avoids the problem of flooding the network with address update packets for obtaining an address on checking for presence of duplicate addresses.

The paper discusses related research works and an analysis of their shortcomings and limitations in Section II. A description of the system model in Section III explains various possible scenarios that could occur in a MANET. Section IV proposes the basic idea of how HAAM works. A detailed description of the protocol is provided in Section V, followed by its performance analysis in Section VI. Section VII provides the conclusion.

2. Related Work

Address auto-configuration protocols are majorly classified as stateful and stateless protocols (Bernardos, Calderon &Moustafa, 2007), (Garcia Villalba, Garcia Maresanz, Sandoralorozco& Marquez Diaz, 2011), (Zhou &Mutka, 2012), (Xiaonon& Shan, 2013). In the case of stateful protocols, each node in the MANET maintains a table corresponding to the logical addresses of all other nodes. Stateful protocols are also known as conflict free protocols since the addresses used for allocation are known to be free. Nodes in MANET following stateless auto-configuration protocols do not record any information about address allocations of other nodes except for its own. Stateless protocols follow trial and error method to detect unique addresses which are available for allocation; hence they are also called as conflict detection protocols. The nodes perform duplicate address detection (DAD) (Moore, 2006) on a randomly chosen address to avoid duplicity in addressing.

2.1 Stateful Auto-Configuration Protocols

Since the proposal of MANETconf in 2002 (Nesargi&Prakash, 2002) and Prophet Address allocation in 2003 (Zhou, Ni &Mutka, 2003), various stateful addressing protocols such as RSVconf (Bredy, Osafune&Lenardi, 2006), EMAP (Ros, Ruiz & Perkins, 2006), LHA(Yousef, Al-Mahdi &Mitschele-Thiely, 2007), D2HCP (Garcia villalba, Matesanz, Sandoval orozco& Marquez Diaz, 2011) and OSA (Al-mahdi, Nassar& El-Aziz, 2013) have been proposed and largely implemented. A brief on few of these to explain the basic approach followed by stateful protocols are discussed. MANETconf protocol maintains two tables: address_pending table which maintains nodes that have been initiated with an address but have not confirmed any address; address_allocated table which maintains all the addresses that have been allocated. A node selects an address for the requesting node which is not present in both of its tables. It then floods the network with a request message to confirm the address allocation. All nodes acknowledge positively on receiving this request message if the address is not found in their table, else acknowledges them negatively. Since all nodes must respond with a reply to the initiator node regardless of address duplication, it incurs high overheads for assigning a node. The whole process of initiation and flooding must be repeated even if a single negative acknowledgement is received by the initiator. Moreover, a node has to wait until it receives a reply from all the other nodes in the MANET.

In Prophet Address allocation scheme, nodes maintain a specific function. This function generates new addresses in relevance to its state. The initial address of the first node in the MANET is randomly chosen along with a random seed for the function which becomes the prophet for the MANET. On arrival of new nodes, the prophet node generates a new IP address and updates its state.

Due to the use of one hop broadcasting for establishing communication between the new node and the configured node, message overheads are very low. RSV conf protocol supports MANET which involves nodes with high mobility. A new node broadcasts proxy requests searching for a suitable proxy node that could assign an address. Proxy nodes maintain an IP database from which free IP addresses are selected for new nodes. Also, a reservation message is broadcasted to all the nodes in the network. On receiving the reservation message, each node checks for the presence of that particular IP in its database. If a conflicting match is found, the node sends a response back to the proxy node; otherwise it registers that particular IP as allocated in its database. This protocol does not allow merging multiple networks simultaneously. This may lead to formation of numerous single node networks. Moreover, RSVconf allows only two networks to merge at a time, hence, merging all the available single node network becomes tedious and leads to very high merger overheads.

2.2 Stateless Auto-configuration Protocols

Stateless protocols may function requiring the use of MAC addresses or without it. Simple DAD (Perkins, 2001), AROD (Kim, Ahn& Lee, 2007), and AIPAC (Fazio, villari&puliafito, 2006) are few which do not require the use of MAC address. IPv6 SAA (Narten, Thomson &Jinmei, 2007) and ND++ (Grajzer, Zemicki&Glabowski, 2014) which is an extended neighbor discovery protocol rely on the usage of MAC address. Nodes usually follow a naive approach of randomly choosing an address followed by duplicate address detection to check for the existence of duplicates. In Simple DAD, a new node randomly selects two addresses of which one is a temporary address while the other is the actual address it wishes to use. It broadcasts these addresses and waits for certain duration. A reply from other nodes within the interval suggests that the address has been allocated already which forces the new node to select a random address again. In AROD, each of the nodes in the network reserves an IP address in advance so that they can effectively assign new nodes with the reserved addresses followed by DAD to ascertain its uniqueness. This effectively reduces the communication overheads and latency while allocating an address to a new node. A particular choice is made by identifying the nature of the application for which the network is deployed in the protocols described above.Rakotondravelona, M. N. R et al (2015) have described a low cost, hierarchical address auto-configuration protocol that reduces the latency and overhead by inferring the address from 1-hop neighbours.

3. System Model and Methodology

Mobile ad hoc networks are self-organizing wireless networks where nodes have free mobility supporting dynamic topology and limited transmission range. A MANET with no connection to external networks like the Internet is termed as a stand-alone network. Such types of networks are often formed by devices trying to establish a network for communication when there is no possibility for an infrastructure network. Since, it is a stand-alone network; nodes have to configure themselves without the aid of external sources like DHCP servers. In order to facilitate communication between nodes which are not in direct transmission range, intermediate nodes act as routers to relay packets. For any practical application of a MANET, address allocation is mandatory. While address configurations may be done following various methods, possible scenarios of structural change in the network must be effectively handled. Few of such possible scenarios are random mobility; arrival of new nodes randomly; gracious and abrupt departure of nodes; temporary unavailability of nodes; network splits and when two separately configured networks merge.

To assign a node with an address, it is not randomly chosen from a pool of reserved addresses held by the allocating node. Instead, an address is generated following specific methods, which is discussed in the following section, to reflect the relationship between the requesting node and the allocating node (*handler*). A hierarchy is established among the addressed nodes which would help to effectively transmit packets throughout the network without the need for blind flooding. In this case, broadcasting is achieved through a series of multicasts. Conventionally, routing tables consisting of

paths of the entire network is required by each node to establish any sort of multicast communication. However, the methodology followed in the proposed HAAM eliminates this need and achieves global reach without it. Moreover, nodes do not have the necessity to have global knowledge of the network which eliminates the requirement for maintaining state tables and address allocation tables across all the other nodes.

A new node on entering the network polls for the presence of any established node in the network by sending beacon frames and starts a timer. On receiving an acknowledgement, the node sends an address request to the node which responded. The handler generates an address and sends it along with the network ID. Receiver node acknowledges this address and adds the handler to its adjacency list. After receiving the address acknowledgement, the handler adds the newly assigned node to its adjacency list as well. Unlike traditional address assignment, nodes do not flood the network to announce the newly assigned address in HAAM. Address assigned to a node comprises of a network ID and host ID. With the help of network id, presence of multiple networks within a MANET can be identified and accordingly network partitions and merges are managed. When two networks with the same network ID come in contact, those nodes which fall within the communication range are readdressed accordingly to maintain hierarchy with either of the network. Disjoint networks may be present within a MANET with the same network ID without causing any problem of identical addresses until they remain disjoint. Readdressing is done only when two such disjoint networks with the same network ID come in contact and they are not disjoint anymore.

The adjacency list used to determine the path for multicasting data packets is local to individual nodes. This method of multicasting depends only on the locally available data without requiring the nodes to share their routing tables among each other. Addressing done hierarchically following this procedure with the help of the adjacency lists ensures that links between all participating nodes are maintained as a spanning tree. A spanning tree guarantees acyclic communication paths between nodes. In practice, routing algorithms employ various spanning tree algorithms to establish an acyclic path. Finding spanning tree of a network is a NP-complete problem. HAAM efficiently eliminates the need to run such cycle detection algorithms since the established mechanism dynamically configures the assigned nodes to form a spanning tree. This reduces the overhead considerably. Existing routing algorithms like AODV, DSR, OLSR and hierarchical routing algorithms can be modified to skip the tests for cycles when HAAM is employed as the addressing mechanism, to reduce the space and runtime complexities of routing algorithms. Moreover, an optimized method for broadcasting without the need for flooding is also proposed.

4. Sample Illustration

A single node entering the MANET is considered initially. The node needs to have an address to participate in the network. Since there are no nodes available from which an address can be obtained, the first node has to initialize the network to accommodate further arrival of nodes in the range. In order to initialize the network, the addressing convention which HAAM follows is described. After initializing the network, the methodology of how the network is formed by further addition and removal of nodes is explained. Arbitrary removal and addition of nodes leads to partitioning and merging of networks which is described in the following subsections before providing algorithms to achieve all the aforementioned scenarios.

4.1 Address Space Determination

Addresses are configured in a manner to contain network ID and host ID. Although this format resembles the conventional IPv4, instead of four octets, HAAM has a 10 bit block followed by two octets making 26 bits in total (Fig. 1). The 10 bit block identifies the network ID (*net_id*) while the two octets hold host ID (*host_id*). *lvl* specifies which level the node is identified in the network. The

value of *lvl* and X together forms *host_id.* Like IPv4, the address can be represented in dotted decimal form as *net_id.lvl.X*. The following addressing pattern *net_id.(lvl)[X]* is followed for convenience.

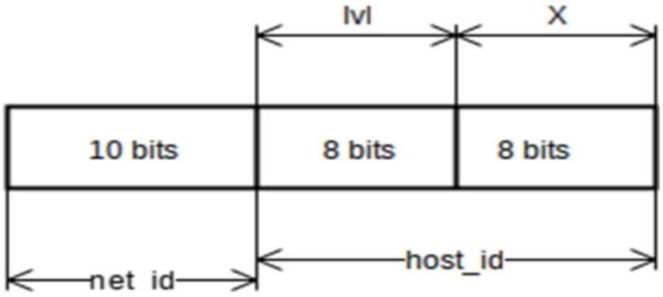

Figure 1: Address space of 26 bits

4.2 Network Initialization

A node on entering the network sets its *status*as *new***,** broadcasts *query_neighbor* message and starts the timer, *query_neighbor_timer*. If no response is received, then there is no node available in the range. When *query_neighbor_timer*runs out, node sends another *query_neighbor* message and the timer is reset. This is repeated certain number of times until the threshold limit for querying is reached. This process is termed as polling. On reaching the threshold limit, set *status* as *isolated* and continue polling. Any node trying to obtain an address is *requestor*. The node assigning an address to a requesting node is *handler*. Nodes with *isolated* status have two possible options to obtain *assigned* status: either on falling in range with a node having *assigned*status, subsequently sending a*ddr_req* message to receive an address;or on falling in range with a node having status as *new*, subsequently initiating the network addressing thereby changing its state as *assigned*. In Fig. 2, nodes with labels *A, N* and *I* have their status value as *assigned*, *new* and *isolated* respectively. Node I has been in the network for a while and carries isolated status since it is not in range with any other node. New node is in range of an already assigned node from which it can acquire an address and participate in the network. One of the two nodes labeled*A* might have had its status as *isolated* initially and later updated as *assigned* on arrival of the other node.

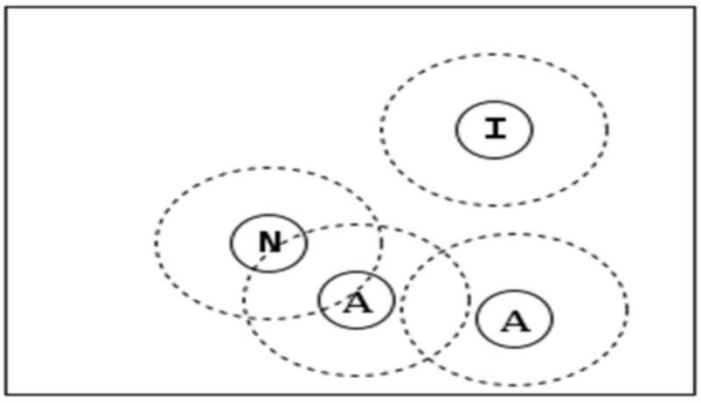

Figure 2. Scenario of Nodes with different *status* value in a network

4.3 Network Formation

Isolated nodes have the potential to be the initiator of a new network. All *isolated* nodes do not become initiators. If it meets an *assigned* node, it does not initiate a new network, instead requests for an address from its *handler***,** eventually going through the transition of being *isolated* and becoming *assigned*. If a *new*node is in range and the threshold limit for wait has exceeded after receiving *addr_req* message from the *new* node, it goes through the task of initiating the network with an address. Once an *isolated* node becomes *assigned*, *new* nodes can receive address from it. These *new* nodes on becoming *assigned* nodes, delegate more addresses by increasing the hierarchy of levels. Also, when an *assigned* node is in range of another *assigned* node having the same *net_id* but with a lower value of *lvl* than that of its handler, it clears its adjacency list, making this newly found node as its handler and requests for an address.

Since *net_id* contains 10 bits, the value it can hold ranges from 0 to *1023*. The *isolated* node can generate a random integer in this range. With *net_id*being allocated 10 bits, a MANET can have a maximum of 1024 distinct networks in it.Each*net_id* can host 256 levels and each level can contain 256 distinct nodes in it. Further increase in size of the address will eventually result in waste of data space. Considering the mobility and uncertainty in continuous availability of nodes, a single network within a MANET handing out all possible address values in range is highly improbable. Moreover, altering the address space to meet the requirements of applications employing HAAM would increase the efficiency of the system by reducing the size of the address space.

If a node which was *isolated* previously happens to meet with two other nodes, subsequently initiates a network by generating a *net_id* and delegating addresses hierarchically to those two newly arrived nodes. Concurrently, the other network in the MANET which previously had two assigned nodes and a new node is joined by two more nodes. All the nodes in the network are assigned addresses. Both the networks were able to concurrently configure the new nodes without affecting each other. Dashed lines between the nodes show the association between each other and the logical connection remains a spanning tree. Both networks have unique *net_id* (8 and 2) and they are not in direct communication range (Fig. 3). The hierarchy in addressing is clearly depicted by the *lvl* part of the address. The network with *net_id* 8 has established three different levels of addressing while the other network with*net_id* 2 has two levels of addressing.

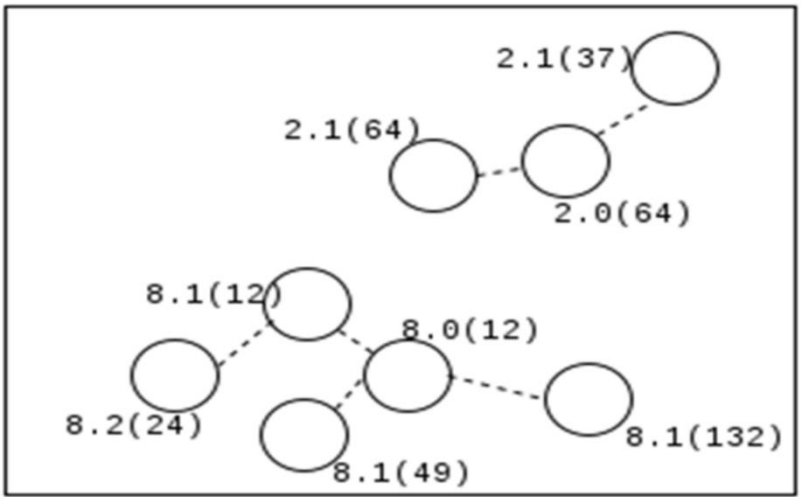

Figure 3. MANET scenario of two separate networks with distinct *net_id*

4.4 Network Partition and Merge

Nodes in MANET have random mobility which may cause a node to fall out of range from its associated network. When a node obtains an address from its *handler*, it adds the handler to the

adjacency list. A *handler* adds the nodes to its adjacency list *adj* to which it assigns an address. Even if any other node of the same network is present in the range of a particular node, it will not add that node to its adjacency list unless it had obtained an address from it or assigned an address to it. This way a link is maintained between the handler and the requesters through adjacency list. As mentioned earlier in section IV, nodes in a network are assigned accordingly to form a spanning tree. Hence, each vertex becomes a cut vertex.

When nodes fall out of the range, they split the network into partitions although they all may carry the same*net_id.* Since the nodes have mobility, structure of networks in the MANET change very often. However, regular polling through neighbor detection, helps check for available nodes in the range. This later leads to assigning appropriate addresses through a *handler* by maintaining the hierarchy termed as readdressing. Readdressing is done only among the nodes with same *net_id* to prevent nodes from having identical *host_id* within the network. Networks with different *net_id* are merged without readdressing since all the nodes are certain to carry unique addresses.

Assume two nodes with addresses *8.1(12)* and *8.2(24)* in Fig. 3 begins to move. Mobility of these nodes place both the networks in range with each other through nodes *8.1(12)* and *2.1(64)*. Fig. 4 represents the merging of the two networks without any requirement for readdressing, since both networks have unique *net_id*s. The logical link formed between the two networks is shown by a double dashed line which also depicts the spanning tree structure. If the node *2.1(64)* in Fig. 4 departs from the network, it splits and continues to exist independently as shown in Fig. 5.

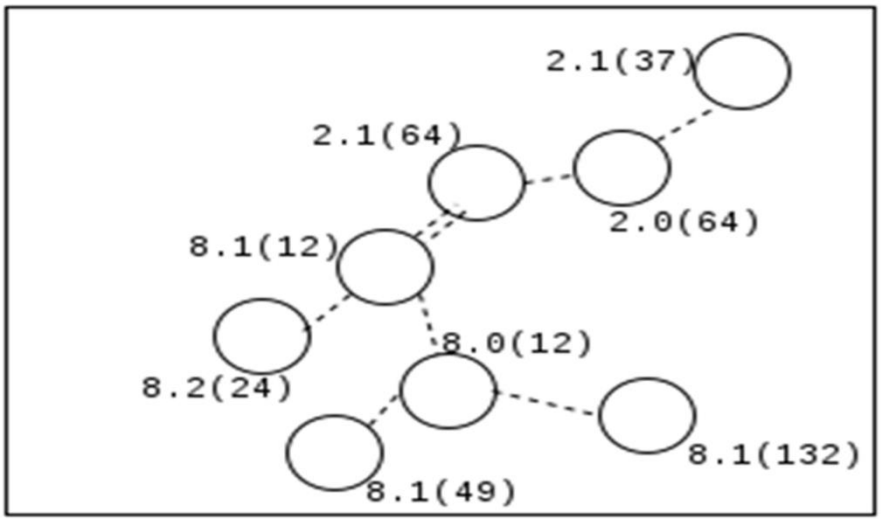

Figure 4. Scenario of Merging of two networks

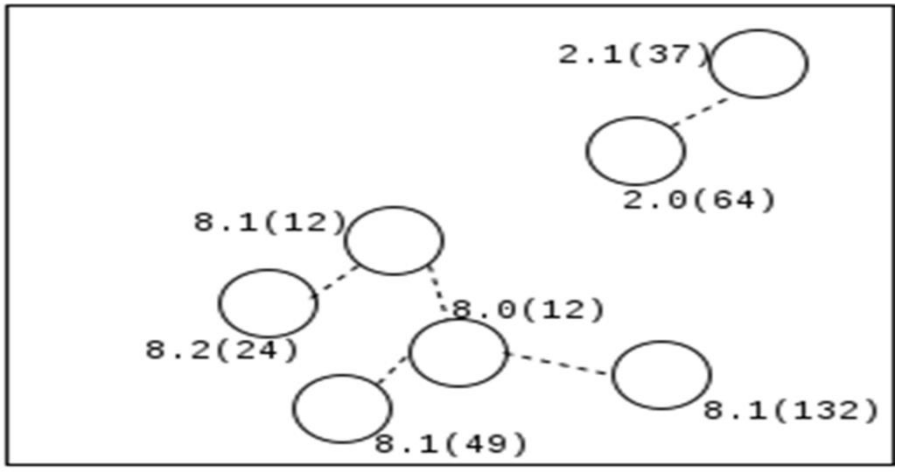

Figure 5. Scenario of the Network Split

4.5 Unique Address Assignment Procedure

The process of unique address assignment on nodes merging and departing with its impact on adjacency list creation is illustrated as different procedures.

Algorithm 1: *Addr_Alloc*procedure (for allocating address)

1. Generate random number *X* in the range [0, 255]. Repeat if *X* is present in the *assigned* list (holds the generated value which was successfully assigned).
2. Send *(lvl+1)[X]* prepended with *net_id* to the *requestor* (the requesting node).
3. On acknowledgement of address from *requestor*, update *assigned* list with the value *X*.

Algorithm 2: *Addr_Req*procedure for requesting address

1. Send addr_alloc message to the handler (the node which will allocate an address).
2. Set the received address as the node address.
3. Set pred (holds the address of the predecessor node) with the handler's address.
4. Add pred to the adjacency list, adj.
5. Send acknowledgement for the received address to the handler.
6. Set status as assigned.

Algorithm 3: Procedure for *assigned* nodes

1. On receiving addr_alloc message from isolated or new node, call Addr_Alloc procedure (Algorithm 1).
2. If a node with different net_id is in range, send join message and add the node to the adjacency list adj.
3. On receiving a join message, add the node to adjacency list adj.
4. Poll for continuous presence of nodes in the adjacency list adj,
   4.1. If present in range, do nothing
   4.2. If absent, remove the node from adjacency list along with the removal of respective *X* value from *assigned* list after the timer expires.

5. If a node with same net_id is in range with lvl lower than the pred node's lvl,
   5.1. Clear adjacency list *adj*.
   5.2. Call *addr_req* procedure

Algorithm 4: Procedure for *new* nodes joining

1. On arrival, set status as new.
2. On successful acknowledgement after polling for presence of any assigned or isolated node, send addr_req message to the handler (the node which responded positively to the poll).
3. If there is no response: set status as isolated, pred with NILL, and reset timer.

Algorithm 5: Procedure for *isolated* nodes

1. On successful acknowledgement from an assigned node after polling, call Addr_Req procedure.
2. While status is isolated on receiving add_req from a new node,
   2.1. Set*lvl* as 0.
   2.2. Set *net_id* with a random number generated in the range [0, 1023];
   2.3. Generate random number *X* in the range [0, 255] and set *host_addr* as *(lvl)[X]*,
   2.4. Set addr as net_id.(lvl)[X].
   2.5. Send *(lvl+1)[X]* prepended with *net_id* to the requesting node.
   2.6. On acknowledgement of address from *requestor*, update *assigned* list with the value *X*.

Set status as assigned.

5. Performance Analysis

5.1 Probability Analysis in assigning the Network ID

An isolated node selects a random number in the range [0, 1023] to initialize the network with a *net_id*. Let A be an event in which a node *P* has generated an integer *i* as its net_id and B be an event in which another node *Q* in the MANET has also generated the same integer *i* as its *net_id*. The probability of node *P* to generate *i* randomly from the given range is P(*A*) is 1/1024. The probability of node *Q* to generate *i* randomly from the given range is P(*B*) is also 1/1024. Intersection of P(A) and P(B) gives the probability value of both nodes generating the same *net_id* which is obtained by multiplying P(*A*) and P(*B*); [P(*A*) x P(*B*) = 0.000000953]. The resulting answer is nearly zero. This assures that no two networks in MANET will have the same *net_id*. However, even if they do have the same *net_id*, it does not cause any duplicate addressing issues in the MANET until those two networks come close enough to be in range. Even if they are in direct range, procedures discussed in previous section handles it perfectly well while merging the two networks.

5.1.1 Performance Evaluation Metrics

Certain attributes and functions of MANET become an important aspect for analyzing the performance of the network. Few such metrics are discussed to understand the extent they influence the network's efficiency.

Distributed operation: Care should be taken to prevent overloading any particular node in the MANET which will otherwise create a bottleneck for the network communications. Operations involved in the

network must be distributed throughout. Functionality of the network must not be solely dependent on specific node. HAAM is a distributed addressing approach which thrives by auto-configuring the network, keeping it robust even during failure of multiple nodes.

Complexity:Considering the limited availability of resources in mobile nodes, caution must be taken to not drain their resources. HAAM ensures that the complexity levels are kept to a minimum by maintaining only the list for holding adjacent nodes' addresses which completely eliminates the need to flood the network while addressing.

Correctness:Addresses of all nodes in the network are unique in HAAM and it is achieved without maintaining tables to obtain global knowledge of the network. All addressing procedures are localized.
Communication overheads:It is essential to avoid broadcasting to the extent possible while addressing, so as to prevent excessive bandwidth consumption. HAAM does not require broadcasting and communicates only with neighboring nodes while addressing. Consider, the number of nodes in MANET is represented by n and the number of links by l and the average transmission time between two adjacent nodes as t. Communication overhead incurred is proportional to the average number of degrees which is $2l/n$, since all address management packets make only one-hop due to localization of transactions. So, the complexity is of the order $O(2l/n)$. Latency is $O(2t)$, since it is proportional to the roundtrip time between two nodes in direct communication range.

Latency: Auto-configuration protocols must try to reduce the latency involved while assigning a node with an address. Latency is measured as the time elapsed between a node's initial request for an address and a successful configuration with that address. Since HAAM does not involve broadcasting methods while configuring nodes, the latency is kept at its minimum. Moreover, the time required for duplicate address detection is also reduced and saved.

Scalability:Protocols requiring multi-hop broadcasting for address auto-configuration do not scale well since most of the channel bandwidth is utilized for broadcasting address management packets. HAAM's scalability is high due its low latency and efficient channel utilization on reducing the number of transactions required for address management and also by keeping the transactions localized. Lower communication overhead, even distribution and shorter latency provides a better scalability factor.

Table 1. Performance metrics analysis

Parameters	**HAAM**
Network structure State maintenance	Hierarchical Partially stateful
Address conflicts	No
Address reclamation	Not required
Complexity	Low
Communication overhead	O(2l/n)
Latency	O(2t)
Scalability	High

6. Conclusion

The hierarchical address auto-configuration protocol presented enables nodes in a MANET to dynamically configure logical addresses without flooding the network with address management packets. The protocol follows a flexible addressing nomenclature which adapts to the constantly changing structure of the nodes in the network. Address delegation is done hierarchically to avoid the process of duplicate address detection while ensuring that all nodes are uniquely identified. Specifically, the need for flooding and the requirement of maintaining large tables to hold global

information is avoided. Moreover, by maintaining the logical links between the nodes as a spanning tree while addressing, the broadcast storm problem is overcome effectively. HAAM tolerates random availability of nodes accounting for their mobility, network partitions, network merges, dynamically altering state of the networks, maintaining network robustness, while providing support for high scalability. With the proposed methodology, this addressing protocol targets to determine more optimal solutions if the routing protocols are tweaked to make better use of the logical links produced between nodes which exist as a spanning tree. The further work of this proposed methodology is the formation of spanning tree during addressing, so as to reduce on running cycle detection tests while establishing routes which will serve to be the best advantage for the routing protocols in mobile ad hoc networks.

References

[1] Al-Mahdi, H., Nassar, H.; and El-Aziz, S. (2013). Performance analysis of an autoconfiguration addressing protocol for ad hoc networks. *Journal of Computer and Communications*,1, 33-40.

[2] Bernardos, C. Calderón, M.; and Moustafa, H. (2005). Survey of IP addressautoconfiguration mechanisms for MANETs. *IETF, draft-bernardosmanetautoconf-survey-05.txt (work-in-progress).(June 2010)*.

[3] Bredy, R., Osafune, T., and Lenardi, M. (2006, June). Rsvconf: Node autoconfiguration for manets. In *2006 6th IEEE International Conference on ITS Telecommunications*. 650-653.

[4] Droms, R. (1997). Rfc 2131-dynamic host configuration protocol, *Obsoletes RFC1541. Draft standard*, *3*(1).

[5] Fazio, M., Villari, M.; and Puliafito, A. (2006). AIPAC: Automatic IP address configuration in mobile ad hoc networks. *Computer communications*, *29*(8), 1189-1200.

[6] GarcíaVillalba, L. J., GarcíaMatesanz, J., Sandoval Orozco, A. L.; and MárquezDíaz, J. D. (2011). Auto-configuration protocols in mobile ad hoc networks. *Sensors*, *11*(4), 3652-3666.

[7] GarcíaVillalba, L. J., Matesanz, J. G., Sandoval Orozco, A. L., and MárquezDíaz, J. D. (2011). Distributed dynamic host configuration protocol (D2 HCP). *Sensors*, *11*(4), 4438-4461.

[8] Grajzer, M., Żernicki, T.; and Głąbowski, M. (2014). ND++–an extended IPv6 Neighbor Discovery protocol for enhanced stateless address autoconfiguration in MANETs. *International Journal of Communication Systems*, *27*(10), 2269-2288.

[9] Kim, N., Ahn, S.; and Lee, Y. (2007). AROD: An address autoconfiguration with address reservation and optimistic duplicated address detection for mobile ad hoc networks. *Computer Communications*, *30*(8), 1913-1925.

[10] Moore, N. (2006). Optimistic duplicate address detection (DAD) for IPv6, *(No. RFC 4429).*

[11] Narten, T., Thomson, S.; and Jinmei, T. (2007). IPv6 stateless address autoconfiguration, *RFC 4862*.

[12] Nesargi, S.; and Prakash, R. (2002). MANETconf: Configuration of hosts in a mobile ad hoc network. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. Vol.* 2, 1059-1068.

[13] Perkins, C. (2001). IP address autoconfiguration for ad hoc networks. *IETF draft-ietf-manet-autoconf-01.txt*.

[14] Rakotondravelona, M. N. R., Harivelo, F., &Anelli, P. (2015, September). Hierarchical Area-Based Address Autoconfiguration Protocol for Self-organized Networks. In *International Conference on Ad Hoc Networks* (pp. 173-184). Springer International Publishing.

[15] Ros, F., Ruiz, P.; and Perkins, C. E. (2006). Extensible manet auto-configuration protocol (emap). *Internet Engineering Task Force (IETF) draft*.

[16] Sharmila, C. and Amalanathan, G. (2016). Construction of Pipelined Strategic Connected Dominating Set for Mobile Ad Hoc Networks. *CIT. Journal of Computing and Information Technology*, *24*(2), 121-132.

[17] Volz, B. (2006). The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option, *(No. RFC 4704).*

[18] Xiaonan, W.; and Shan, Z. (2013). An IPv6 address configuration scheme for wireless sensor networks based on location information. *Telecommunication Systems*, *52*(1), 151-160.

[19] Yousef, A., Al-Mahdi, H.; and Mitschele-Thiel, A. (2007, October). Lha: logical hierarchical addressing protocol for mobile ad-hoc networks. In *Proceedings of the 2nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*.96-99.

[20] Zhou, H., Ni, L. M.; and Mutka, M. W. (2003). Prophet address allocation for large scale MANETs. *Ad Hoc Networks*, *1*(4), 423-434.

[21] Zhou, H.; and Mutka, M. W. (2012). Review of autoconfiguration for MANETs, *INTECH Open Access Publisher*.