PERFORMANCE AND SCALING OF WIRELESS AD HOC IPV6 STATELESS ADDRESS AUTOCONFIGURATION UNDER MOBILE GATEWAYS

Jeffrey Wildman, David Hamel, Ryan Measel, Dan Oakum, Steven Weber, and Moshe Kam
Department of Electrical and Computer Engineering
Drexel University, Philadelphia, Pennsylvania 19104

ABSTRACT

Autoconfiguration mechanisms in general, and stateless address autoconfiguration in particular, are highly desirable capabilities of military mobile ad hoc networks (MANETs). However, IPv6 stateless autoconfiguration schemes for MANETs still have to be refined, and a convincing demonstration is needed to show that these schemes can cope with the dynamic, infrastructure-free environment wherein MANETs operate. In this paper we provide a literature survey of autoconfiguration schemes designed for MANETs. In addition, we look at a specific stateless autoconfiguration scheme (by Jelger and Noel, SECON 2005). This scheme provides globally routable IPv6 prefixes to a MANET, attached to the Internet via gateways. We examine this approach through OPNET simulation, applying new mobility models to encourage squad-like clusters around gateways, introducing mobility to the gateways, and scaling the number of ad hoc nodes and the number of gateways independently. We then comment on the performance of the Jelger-Noel addressing scheme in terms of protocol overhead, autoconfiguration time, prefix hold times, and prefix stability.

I. INTRODUCTION

It has been recognized for several years that in order to provide rapidly deployable networks for military operations, address configuration must be streamlined. In particular, it is desired to reduce the level of human intervention – the manual configuration of hundreds (and in some networks, thousands) of devices – which is tedious, time consuming, and expensive. Against this background, the autoconfiguration features of IP version 6 (IPv6) appear to have significant potential to simplify the planning and managing of large-scale networks [5-13]. These features were designed so that manual configuration of hosts' addresses, before connecting them to the network, is no longer needed. Military MANETs can benefit significantly from autoconfiguration features, especially stateless autoconfiguration. However, IPv6 stateless autoconfiguration schemes for MANETs still have to be refined, and be accompanied by a convincing demonstration.

In this paper, we provide a literature survey of autoconfiguration schemes that could be applied to MANETs. One such proposed stateless autoconfiguration scheme, by Jelger and Noel (J&N) [1], provides globally routable IPv6 prefixes to a MANET, attached to the Internet via gateways. We examine the Jelger-Noel scheme through OPNET [2] simulations, taking into account several design considerations. First, we allow gateways, which provide connectivity to the Internet or a remote network, to become mobile. Second, we introduce new mobility models to the nodes, so that squad-like clusters are formed around the gateways. Third, we scale the number of ad hoc nodes and the number of gateways independently. We are interested in quantifying the following trends: (1) the performance of the protocol as the ratio of the number of ad hoc nodes to gateways changes; (2) the scaling of the protocol's overhead and initial autoconfiguration time versus network size; and (3) the protocol's performance under a group mobility regime with various cluster densities.

The rest of this paper is organized as follows. Section II contains an overview of autoconfiguration schemes for MANETs and other related research. Section III presents the motivation for the choosing the J&N auto-configuration protocol and the general environment wherein it is applied. Section IV describes the J&N protocol. Section V describes our modeling setup in OPNET – including description of protocols, their parameters, and implementation of the J&N protocol. Section VI covers performance metrics of interest and explains the simulation scenarios. In Section VII, we discuss trends observed in the simulations. Section VIII presents conclusions, and Section IX suggests possible avenues for future work on this topic.

II. AUTOCONFIGURATION APPROACHES AND RELATED WORK

Several proposals have been made concerning address autoconfiguration. These proposals can be divided into three main categories: *stateful*, *stateless*, and *hybrid*.

Stateful Autoconfiguration

Stateful autoconfiguration uses address allocation tables to maintain control over assignment of addresses.

This work is funded by the U.S. Army Communications—Electronics Research, Development and Engineering Center (CERDEC), under contract # DAAB-07-01-9-L504.

1-4244-1513-06/07/$25.00 ©2007 IEEE

This method ensures uniqueness of addresses and eliminates the need for duplicate address detection (DAD). However, in order to maintain the allocation table this approach requires either a centralized controller or a synchronized distributed system. Neither centralized control nor synchronized distributed systems are suitable for MANETs. In a centralized scheme [3], the designated *Addressing Agent* (AA) often incurs large overhead associated with handling and maintaining all addresses for the network. Moreover, a single point of failure is apparent – the entire network depends on a single node which is not guaranteed to always be reachable. This lack of robustness can be overcome by using a distributed system, such as MANETconf [4], wherein allocation tables are synchronized across multiple nodes. Some form of reliability assessment must be employed in this case, to ensure that the tables remain synchronized.

Stateless Autoconfiguration

Stateless autoconfiguration allows nodes to self-assign an IP address randomly or based on a hardware ID. To guard against duplicate addresses, the central element of a stateless proposal is some form of DAD – active or passive. In *Query-based DAD* [5] (QDAD), a node chooses two addresses on startup, a *temporary address* and a *tentative address*. The node attempts to establish communication with the tentative address from the temporary address, and then waits a specified period of time. If no response is received, the node assumes that the address is available and adopts it. Unlike QDAD, the alternative methods, *Weak DAD* and *Passive DAD*, must rely on a routing protocol. In *Weak DAD* [6] (WDAD), an initialization key is generated for each node and distributed with all routing packets. The keys are stored in the routing table which is used for comparison with the keys included in subsequent routing packets received. If different keys are received from the same address, then that address is assumed to have been duplicated. In *Passive DAD* (PDAD), proposed by Weniger [7], received routing packets are analyzed to detect any conflicts based on events that would not occur with unique addresses. By using information that is already available in the network, the amount of overhead introduced by this protocol is significantly reduced.

Hybrid Autoconfiguration

Several hybrid approaches have been proposed that use elements from both stateful and stateless autoconfiguration methods. These often provide more robust protocols but also increase complexity and overhead. The *Hybrid Centralized Query-based Autoconfiguration* (HCQA) [8] protocol uses both QDAD and a centralized allocation table on a dynamically assigned AA. The AA can then prevent duplication even if the original node is offline and not able to respond to the query. The *Passive Autoconfiguration for Mobile Ad Hoc Networks* (PACMAN) [9] protocol employs both PDAD and a distributed allocation table. This protocol does not synchronize the allocation tables actively but allows the nodes to collect the information needed for disambiguation by monitoring routing traffic passively. Additional descriptions of autoconfiguration schemes can be found in [10, 11].

Duplicate Address Detection

Several methods were proposed to prevent duplicate addresses. Most approaches used in centralized stateful autoconfiguration appear to lack the robustness necessary to compensate for the dynamic nature of MANETs and have high network flooding overhead. Approaches used in synchronized distributed stateful autoconfiguration often incur high overhead and falter in the face of high packet loss. QDAD stateless autoconfiguration methods often have high overhead and do not guarantee address uniqueness [5]. They often exhibit difficulties in accounting for network merging and partitioning. Both WDAD and PDAD have to rely on the routing protocol used by the network [6, 7].

It is not even clear whether the effort to prevent duplicate addresses preemptively is necessary in most scenarios, since in most reasonable schemes address duplication has a very low probability of occurring. Given a 128-bit address, 64-bit subnet prefix, and random address assignment, there is only a 1 in 2^{64} chance that any two nodes will adopt duplicate addresses. Even in a 10,000 node subnet, the collision probability becomes 1 in 1.84×10^{15}. It is appears that in most MANET networks any implemented preemptive DAD mechanism would add a layer of complexity and increase the risk of network failure well beyond the probability of ever encountering a duplicate address.

MANET Autoconfiguration

The increased interest in using MANETs raises questions about their integration with the Internet. Lamont *et. al.* [12] discuss an approach to integrate MANETs with the Internet which stresses minimizing handoff latency between WLANs and MANETs. In [13], “a self-organizing, self-addressing, self-routing IPv6-based MANET which supports global connectivity and IPv6 mobility” is proposed. It uses a global prefix in combination with a logical prefix to form the IP address of mobile nodes. King and Smith [14] discuss the emerging possibility of using an ad hoc network to provide the military with access to distant networks through gateways. They formulate an architecture that includes DAD, two gateway selection schemes (centralized and distributed), and MANET routing protocols. In [15], Ammari and El-

Rewini present a method to integrate Internet connectivity to MANETs using mobile gateways. Their work is based on a three-layer approach using Mobile IP and dynamic destination-sequenced distance vector (DSDV). Denko and Wei [16] present an architecture comprised of multiple mobile gateways in order to connect the Internet to MANETs using an extended AODV routing protocol. Gateway discovery is presented for both a reactive scheme (for small networks) and a hybrid scheme (for larger networks). In [17], Mo *et. al.* present new algorithms for connecting MANET nodes to the Internet; these algorithms are independent of routing protocols.

A MANET-Internet autoconfiguration scheme of particular interest is that of Jelger and Noel (J&N [1]). Their focus is on the autoconfiguration of globally routable IPv6 addresses in an ad hoc network that is connected to the Internet via one or more gateways (*i.e.*, a hybrid ad hoc network). The J&N protocol forms logical trees anchored at the gateways; the branches consist of ad hoc nodes that have selected the same prefix as the gateway. For any given ad hoc node that has selected a gateway, there is a path to that gateway such that all intermediate nodes and the gateway share the same global prefix. This property is called *prefix continuity*. Some of the benefits of prefix continuity are the avoidance of source routing and support for a dynamic network topology (including network partitioning and merging and temporal gateways) [1].

Military Networks and Mobility Models

Several studies used simulations in order to understand how IPv6 operates in hierarchical military networks. Military network representation is perhaps best exemplified in [18], where the use of an IPv6 MANET of tactical radios is explored. Other studies that dealt with military network topology include the work of Kant *et al.* [19], though their study does not focus on IPv6.

Most simulations use a random waypoint (RWP) scheme to represent node placement and movement. This approach has been criticized in the literature (e.g., [21]). In the context of our objectives it often fails to represent a realistic mobile scenario, as it is unlikely that all nodes in a military network would wander about in a manner conforming to the distributions assumed by RWP algorithms. In fact, nodes in military networks almost always move in a coordinated scheme or at least organized in groups and clusters.

III. PREFIX CONTINUITY

The J&N protocol is dependent on one major mechanism and a second, optional mechanism. The protocol's major mechanism controls how globally routable prefixes are advertised and selected, in order to ensure prefix continuity. Gateways periodically advertise their global prefixes in messages known as GW_INFO messages. These advertisements are sent out at an interval measured in seconds specified by the variable *gw_info_refresh_period*. GW_INFO messages contain such fields as the gateway (global) prefix advertised and distance to the gateway measured in hops.

If a node receives a GW_INFO message and decides to accept the advertised gateway's prefix, the sending node becomes the *upstream neighbor* of the receiving node. Nodes will only forward GW_INFO messages containing an advertised prefix that matches their selected gateway. By traveling along the path of upstream neighbors recursively, one will eventually reach the gateway that advertised the prefix that all the traversed nodes adopted. This forwarding process is key to producing prefix continuity.

Over the course of time, the ad hoc nodes in the network may receive advertisements originating from multiple gateways, and must decide which gateway to select. J&N provide several algorithms for selecting an upstream neighbor, the two most significant being the *distance* and *stability* algorithms. A node operating under the ***distance* algorithm** will change its selected global prefix if the distance in hops from itself to a newly advertised gateway is less than the distance to the current gateway; this is done in an attempt to maintain a minimum distance to the Internet (via the selected gateway). The ***stability* algorithm** seeks to maximize the amount of time spent with the same prefix and will not change gateways as long as it continues to receive GW_INFO messages advertising its currently selected prefix. When selecting a new gateway, nodes choose the global prefix that was advertised by the largest number of nodes.

A second, optional mechanism exists to verify bi-directionality in the wireless links between nodes. Under some circumstances, such as heterogeneous devices or wireless channel characteristics, links between nodes may be uni-directional. In these situations pairs of nodes may perform a three-way handshake using bi-directional (BIDIR) messages before one node can choose the other as its upstream neighbor.

As part of this scheme, each node maintains a neighbor table built from the GW_INFO and BIDIR messages received. These tables assist in the selection of new upstream neighbors. The table entries are set to expire if not refreshed by incoming GW_INFO messages.

For more details on the J&N protocol, please see [1, 20, 22].

IV. NETWORK ENVIRONMENT

We wish to apply the J&N prefix continuity protocol to a more complicated environment than the scenario studied by the original authors, namely one that better approximates a military network topology. In their original study, J&N address the performance of their protocol in a fixed-size (100 nodes) ad hoc network with four (4) stationary gateways placed in the corners of a simulation arena (2000 *m* x 2000 *m*). The ad hoc nodes were permitted to move about the entire simulation space according to a RWP mobility model.

This setup is unsuitable for the networks we are interested in studying. First, it may not be reasonable to expect the placement of gateways to bound a geographical region enclosing a MANET. Second, J&N only present the study of a fixed-size, 100 node network, therefore the scalability of the protocol cannot be clearly defined. Autoconfiguration protocols that run on military networks (which constitute from hundreds to thousands of nodes with varying densities) must be *scalable* regarding autoconfiguration time and protocol overhead. Third, military networks may consist entirely of mobile nodes, including mobile gateways. Finally, platforms in military networks are not likely to be traveling randomly about an entire geographical region; movement is always coordinated to some extent.

In applying the J&N protocol to a military-type network, Internet connectivity could be easily equated with connectivity to a larger or remote network. Specifically, we consider a hierarchal MANET, illustrated in Figure 1, consisting of two types of platforms, labeled *leaders* and *subordinates.* The set of leader platforms form a global MANET subnet, which can be considered a wireless backbone. Leader platforms and geographically close subordinate platforms produce additional wireless local MANET subnets. The gateway-node relationship from J&N is applied directly to the leader-subordinate analog here.

Additionally, we consider a group mobility model for the subordinates. It encourages cluster-formation around the gateway nodes to better mimic squad-like military formations. We further assume a simple random waypoint movement scheme for the gateways. Mobile gateways introduce additional events of interest, such as when two gateways that publish different global prefixes approach each other. It is important to observe how surrounding ad hoc nodes react to moving gateways in close proximity.

Leader platforms contain two wireless interfaces, one for longer-range communication with other leaders, and one for local communication with subordinates. Subordinate platforms have only one wireless interface for communication between local subordinates and gateways.

We assume that leader platforms are preconfigured in a manner that allows stable communication among leaders, and focus our attention on the autoconfiguration of subordinate platforms. All leaders publish their prefixes to surrounding subordinates in order to establish the logical prefix trees associated with prefix continuity as described earlier.

The prefix continuity protocol does not make any assumptions about the underlying ad hoc network topology or the number of gateways. In fact, the ad hoc network may be comprised of multiple partitions (each with one or more gateways) and still operate successfully. For military networks these properties provide flexibility in the way subordinates and leaders are arranged. For example, leaders may come online, disappear, or move to a new location; subordinates will have to adjust.

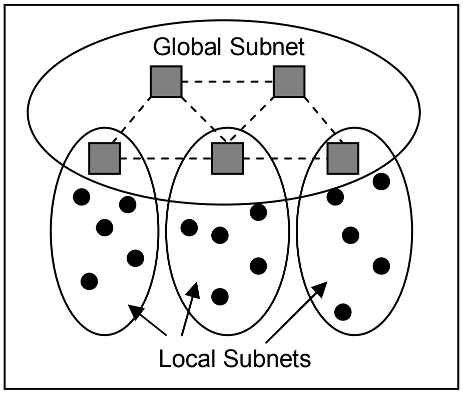

Figure 1: The logical network topology under consideration. Leaders, denoted by squares, form the backbone and serve as attachment points for the subordinates, represented as dots.

Ad hoc nodes that stray too far from their affiliated gateway will re-associate with a more appropriate gateway (if available); otherwise they become "orphaned". Re-association is accomplished through the upstream neighbor selection algorithms.

V. MODELS

This section details the general scenario representation in OPNET Modeler version 12.0 [2]. Parameter values specific to a particular scenario, as well any exceptions to the general model setup described are detailed later. Where possible, simulation parameters were set to values equal to those used in the J&N study [1].

General Description

The simulation arena was a flat 2000 *m* by 2000 *m* square. *N_ah* is used to designate the total number of ad hoc nodes in the simulation arena, while *N_gw* represents the number of gateways. Gateway and ad hoc nodes were built from the standard OPNET 'manet_station_adv' node model. Since the wireless backbone formed by all leaders was not the focus of this particular study (and therefore not actively simulated), gateways and ad hoc nodes each had only one IEEE 802.11 interface. The transmission power for nodes was set such that a transmission had a maximum range of 250 *m* under interference-free conditions. The standard, interference-based physical layer model was then used during simulation. Application traffic was not modeled in the network. A "warm-up" time of 3000 seconds was applied to the mobility models so that the spatial distribution of nodes approached steady-state [21], after which the J&N protocol was started and statistics began to be recorded.

Prefix Continuity Model

A model of the J&N prefix continuity protocol was constructed in OPNET based on [1], [20], and [22]. It was implemented as a stand-alone process model (*i.e.,* not integrated with a specific routing protocol) and interfaced with the User Datagram Protocol (UDP) transport layer protocol. BIDIR messages were disabled, as all nodes used the same transmission power. Both the *stability* and *distance* algorithms for upstream neighbor selection were studied. Prefix continuity protocol parameters, which controlled timers and neighbor table entry timeout values, were set in accordance with [20].

Mobility Models

Mobile gateways were permitted to move about the entire simulation space according to a RWP model with a constant speed of 2.5 *m/s* and a pause time of 25 seconds.

Ad hoc nodes were assigned a group mobility model that consisted of a restricted RWP model. At the beginning of the simulation, each node chose a random gateway uniformly from the set of all gateways. The restricted RWP selection algorithm would choose waypoints within a specified region of the chosen gateway. The destination selection was controlled by the parameter *xy_var*, which specified the maximum coordinate deviation in meters in both the horizontal and vertical directions from the chosen leader. Ad hoc nodes moved at a constant speed of 5 *m/s* and had a pause time of 10 seconds.

VI. PERFORMANCE METRICS & SIMULATION SCENARIOS

In this section we define the performance metrics collected from the simulations; many of them were proposed in [1]. We then describe three different scenarios used to capture the performance of the prefix continuity protocol. Each scenario combination was run ten (10) times and the mean value was used for all data presented.

Metrics

- **Total Autoconfiguration Time** (seconds): The elapsed time between the transmission of the first GW_INFO message and the time that 95% of the total number of ad hoc nodes had selected their first global prefix.
- **Autoconfiguration Overhead** (bytes/second per node): The per-node-average rate of data passed down from the J&N protocol to the UDP process. This does not include IPv6 or MAC headers.
- **Prefix Updates** (updates/node): The average number of prefix updates that a node experiences during simulation. A prefix update occurs when an ad hoc node chooses a new upstream neighbor but retains the same global prefix.
- **Prefix Changes** (changes/node): The average number of prefix changes that a particular node experiences during the simulation. A prefix change occurs when an ad hoc node chooses a new upstream neighbor and global prefix.
- **Average Prefix Hold Time** (seconds): The average period during which a prefix remained constant for any given ad hoc node. The prefix hold timer started upon configuration of a new prefix and stopped for only the following events: a prefix change occurred, or the upstream neighbor entry timed out.
- **Percentage of Time Without a Prefix** (%): The percentage of time a node spent without a prefix, starting from the transmission of the first GW_INFO message until the end of the simulation.

Model Validation

In order to extend the studies of J&N, validation of our protocol implementation with the authors' data was conducted. To this end, we replicated the scenario presented in their original paper [1]. J&N's simulation arena consisted of a flat 2000 *m* by 2000 *m* area with a stationary gateway placed in each corner, 250 *m* away from each edge. Each gateway advertised a unique prefix. One hundred nodes moved about the simulation space according to random waypoint with speeds chosen uniformly from the range [0.5, 1.5] *m/s* and pause times of 150 seconds. We simulated this scenario for both upstream neighbor algorithms and collected statistics based on the metrics described above. Each simulation run lasted 65 simulated minutes (50 minutes for mobility model "warm-up" time plus 15 minutes of protocol operation).

Scenario I

We examined the effect of *scaling* the number of ad hoc nodes, *N_ah*, for a single stationary gateway located at the center of the (square) simulation arena. *N_ah* was varied from 25 to 400 nodes. For each ad hoc network size, the group mobility parameter *xy_var* was varied from 250 to 1000 meters so that the ad hoc nodes movement became less dependent on the chosen leaders. Only the *distance* upstream neighbor selection algorithm was studied under this scenario, as the *stability* algorithm simplifies to the *distance* algorithm in the presence of one advertised prefix.

Each simulation run lasted 65 simulated minutes (using the same division as in the validation section).

Scenario II

We studied the effect of *gateway mobility for a single gateway*. *N_ah* and *xy_var* were varied as in Scenario I. The gateway node was configured to move around the simulation arena under the unrestricted random waypoint model described in Section V. Again, only the *distance* algorithm was simulated in this scenario. Each simulation run lasted 75 simulated minutes (adding 10 extra minutes to the protocol operation).

Scenario III

Finally, multiple gateways were permitted to move without restriction in the simulation space. The number of gateways, *N_gw*, was varied from 1 to 4. Each gateway was set to advertise a unique global prefix. For each selection of *N_gw*, several values for the ad hoc network size, *N_ah,* were tested while *xy_var* was fixed at 500 *m*. Each simulation run lasted 75 minutes, as in the previous scenario.

VII. RESULTS

Figures 2-6 show the principal findings and data trends exhibited in our simulations.

Validation

In general, our implementation of the J&N protocol simulated in OPNET performed as expected and was comparable to the data presented in [1], specifically autoconfiguration time, prefix updates, prefix changes, and prefix hold time. For data relating to the autoconfiguration time ("convergence" in [1]), prefix changes per node, and prefix updates per node, we find similar trends for both upstream neighbor selection algorithms. However, there is a difference in our reported average prefix hold time. Our simulations reported an average prefix hold time of the order of ten (10^1) seconds for both upstream neighbor algorithms, while J&N reports average prefix hold times of the order of one hundred (10^2) seconds for the *stability* algorithm.

After discussion with the original authors, it was found that our validation scenario did not include stationary relay nodes that were used to increase the effective transmission range of the gateways [24]. These nodes served to counteract the RWP steady-state distribution of node positions, located around the center of the simulation arena. The lack of relay nodes made it easier for the gateways to become disconnected from the MANET, potentially driving our measured prefix hold times lower.

Other potential sources for these differences include inherent differences in the simulators [23] (J&N used the network simulator, ns-2, in [1]) and differences in the interpretation of how the metric should be measured.

Scenario I

Figures 2 and 3 address the scaling of the total autoconfiguration time and the prefix continuity protocol overhead, as the number of nodes in the ad hoc network increases around a stationary gateway for various cluster sizes (controlled by *xy_var*).

Figure 2 shows the total autoconfiguration time (in seconds) versus the number of ad hoc nodes in the network, while the cluster size variable, *xy_var*, serves as a parameter. The measured autoconfiguration times show a strong dependence on the density of the cluster around the gateway. The density of the network can be increased in two ways, by adding more ad hoc nodes, *N_ah*, to the network, or by making the cluster area smaller, via *xy_var*. In both cases, the autoconfiguration times decrease for more dense networks. This is largely attributed to nodes already being within range of a potential upstream neighbor in order to autoconfigure for the first time. In less dense networks, more time is spent waiting for nodes to move within range of potential upstream neighbors, and a larger total autoconfiguration time is recorded.

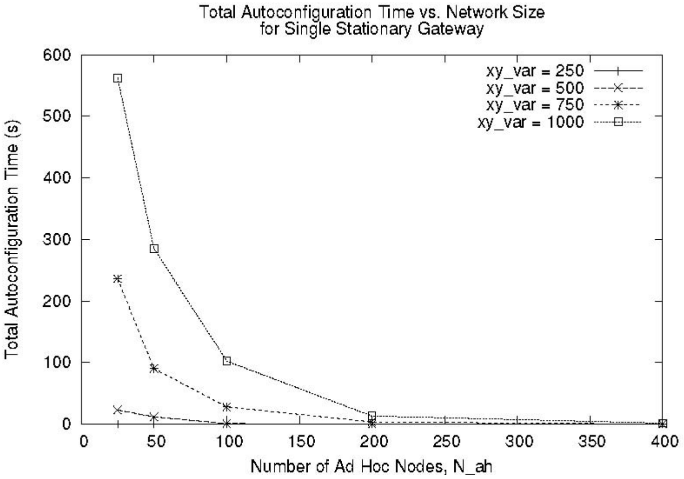

Figure 2: Total autoconfiguration time plotted against the number of ad hoc nodes for varying group mobility parameter values.

Figure 3 shows the overhead per node (in bytes/second) versus the number of nodes, *N_ah*, in the ad hoc network, while *xy_var* serves as a parameter controlling the cluster area. Increasing the density of the cluster around the gateway, either by increasing *N_ah* or by decreasing *xy_var*, tends to produce higher overhead in simulation. This behavior is explained by the connectivity of the network and the GW_INFO forwarding process. As the network becomes denser, a greater percentage of ad hoc nodes are connected to the gateway via one or more hops and will participate in the GW_INFO forwarding process, occurring once every *gw_info_refresh_period*.

However, the specific curve for *xy_var = 250 m* in Figure 2 is nearly level, showing that adding ad hoc nodes to the network has little effect on the protocol overhead. Under this specific operating region, the network has already reached full connectivity and that most, if not all nodes are already participating in the GW_INFO forwarding process every *gw_info_refresh_period*.

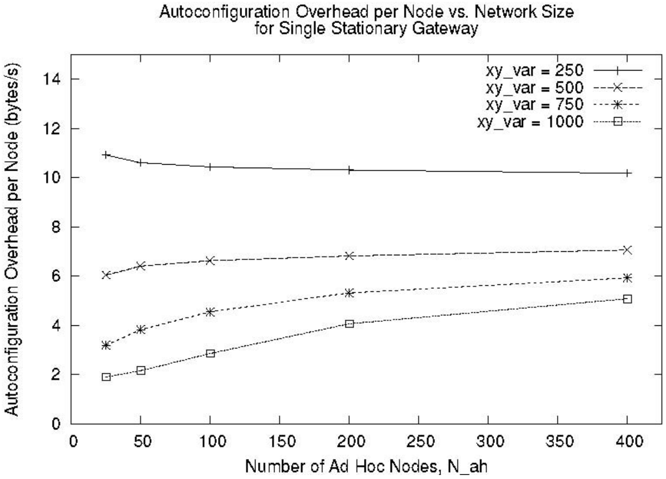

Figure 3: Autoconfiguration overhead shown versus the number of ad hoc nodes for varying group mobility parameter values.

Scenario II

Figures 4 and 5 present the performance of a single mobile gateway using the prefix continuity protocol in terms of average prefix hold time (shown in Figure 4) and fraction of time without a prefix (shown in Figure 5) when using the group mobility model described in Section V. Both dependent variables are measured against the number of nodes in the ad hoc network, while the cluster size variable, *xy_var*, serves as a parameter.

Examining the prefix hold time curves in Figure 4 shows that as the network density increases, either by increasing *N_ah* or by decreasing *xy_var*, prefix hold time decreases. It is hypothesized that for denser networks, logical prefix branches are more likely to form between the gateway and ad hoc nodes. However, when intermediate nodes in the branch move away, these branches will break, resulting in lower prefix hold times. Additionally, in larger cluster areas (*xy_var* = 1000 *m*), more nodes are likely to be affected by prefix branches breaking because longer chains are possible. As a result, larger cluster areas tend to report lower prefix hold times in Figure 4. The exception to this last statement occurs for very sparse networks (e.g., *N_ah* = 25 nodes and *xy_var* = 1000 *m*). The few nodes that actually configure a prefix (and consequently report to the average prefix hold time metric) do so usually directly from the gateway and tend to hold on to that prefix for a longer amount of time due to their proximity about the gateway. The end result is that very sparse networks tend to report higher prefix hold times than the other network configurations, seen in Figure 4.

Figure 5 shows that the percentage of time without a prefix is highly dependent on the density of the network. Increasing the network density through *N_ah* or *xy_var* will result in lower percentages of time without a prefix It is also interesting to note that just because a particular network configuration, such as the one from the previous discussion about sparse networks, may report a high average prefix hold time, it does not mean that the nodes experience a low percentage of time without a prefix. In fact, for the case where there were the fewest ad hoc nodes, *N_ah* = 25, in the largest cluster size, *xy_var* = 1000 *m*, the percentage of time without a prefix is extremely high. This supports the previous assertion that most nodes are too far away from upstream neighbors to configure a prefix and the few nodes that do configure a prefix usually do so from the gateway directly.

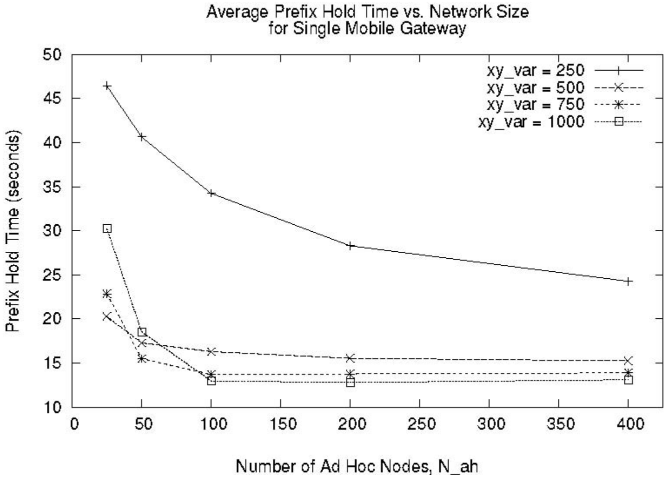

Figure 4: Average prefix hold time per node plotted against the number of ad hoc nodes for varying group mobility parameter values.

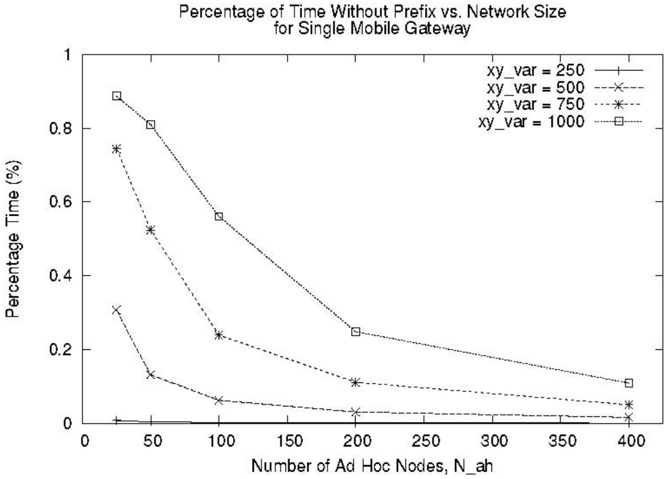

Figure 5: Percentage of time without a prefix per node plotted against various node densities.

Figures 4 and 5 together show that even though the average prefix hold time decreases with network size, the percentage of time the average node spends without a prefix also decreases. The same metrics from Scenario I show the same data trends and are not displayed. When data from Scenario I and II were directly compared, they showed the effects of making the gateway mobile. To summarize them, gateway mobility tends to lower the average prefix hold time, increase the percentage of time without a prefix, and make the logical prefix branches become harder to maintain.

Scenario III

Figure 6 displays both the prefix updates per node (left vertical axis, marked as UP in the legend) and the prefix changes per node (right vertical axis, marked as CH in the legend), and plotted against the number of gateways, *N_gw*. Prefix updates and changes are shown for the distance and stability upstream neighbor selection algorithms, marked with initials D and S respectively in the legend. The plot is shown for an ad hoc network size of 200 nodes and *xy_var* = 500 *m*, as the data demonstrated similar trends for the other network sizes.

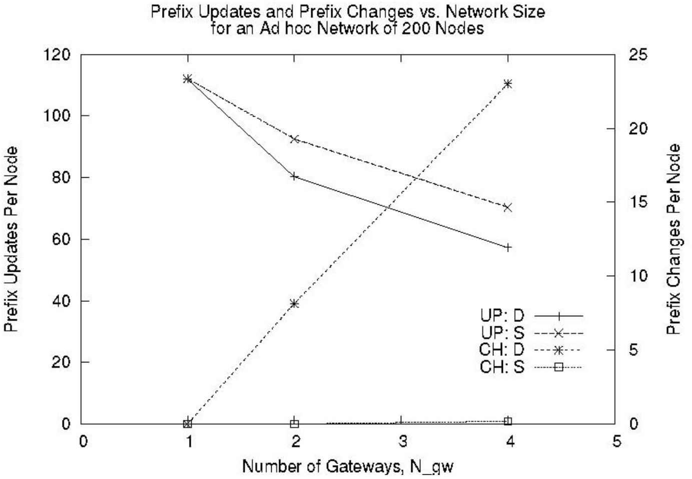

Figure 6: Number of prefix updates (left axis) and prefix changes (right axis) versus the number of mobile gateways.

As seen in Figure 6, the number of prefix updates for both algorithms decrease as gateways are added. This tendency is due to the increased partitioning of the 200 nodes, causing smaller groups of nodes to form around each gateway. Conversely, the number of prefix changes increases for both algorithms (albeit very little for the *stability* algorithm) as the number of gateways increases. This tendency is attributed to the fact that each gateway advertises a unique prefix. By making more prefixes available in the network, it is more likely that nodes will experience prefix changes during the simulation.

Comparing the two algorithms in Figure 6, the *stability* algorithm was found to produce more prefix updates and fewer prefix changes than the *distance* algorithm. This tendency is due to the way each algorithm selects upstream neighbors. The *stability* algorithm attempts to minimize prefix changes by choosing upstream neighbors that preserve the current prefix; the *distance* algorithm chooses an upstream neighbor solely on the distance (in hops) of that neighbor to the advertised gateway.

VIII. CONCLUSIONS

We investigated the performance and scaling properties of an IPv6 stateless address autoconfiguration scheme, the J&N prefix continuity protocol [1], as applied to military-type MANETs. We first validated our protocol model against the original data [1] to ensure proper operation. We noted that in terms of most metrics our simulations and those of [1] showed the same trends. However, we noted differences in prefix hold times and provided several explanations for such differences. Next, we examined the scalability of a single stationary gateway. Subsequently, we focused on protocol performance under a single mobile gateway. Finally, general performance characteristics were gathered when multiple mobile gateways were present in the network.

In the single stationary scenario (Scenario I), we found that large network densities achieved rapid autoconfiguration times. Although the autoconfiguration overhead is higher for dense clusters, the overhead in general is very low when BIDIR messages are disabled. Hence the protocol shows potential for military networks where bandwidth availability is a concern and routing protocols can validate bi-directionality in links.

By permitting gateways to move in a simple random waypoint scheme in Scenario II, we observed an impact on the protocol’s performance. Gateway mobility increased the percentage of time without a prefix and decreased the prefix hold time. It was also found that adding more ad hoc nodes to the network tended to decrease the percentage of time without a prefix as well as the prefix hold time.

By allowing multiple clusters to move around freely in the arena, we were able to observe how the two neighbor selection algorithms handled gateway mobility. The *stability* algorithm resulted in a multitude of prefix updates and very few prefix changes, as it was designed to maximize average prefix hold time. The *distance* algorithm registered more prefix changes per node as a result of its focus on minimizing distance (in hops) to the chosen gateway.

IX. FUTURE WORK

Future studies should address the feasibility of a hierarchical layering of the prefix continuity protocol, such that mobile gateways would first autoconfigure their prefixes from one or more status-elevated gateways.

Autoconfiguration of the ad hoc nodes underneath would then take place after a gateway completed its own autoconfiguration.

Allowing multiple gateways in the same mobile group to advertise the same prefix is another scenario of interest. This capability could provide faster autoconfiguration time while possibly also increasing prefix stability.

Third, it is worthwhile to investigate the performance of the prefix continuity protocol in light of the "parking lot" problem. A network may specifically autoconfigure in a static topology, or "parked" state, and mobilize afterward.

Attention should also be given to the notion of an optimal gateway-to-ad hoc node ratio. Such a study should seek to find a threshold where the selected number of gateways balance metrics such as percentage of time without a prefix and prefix hold time.

X. REFERENCES

[1] C. Jelger and T. Noel, "Proactive Address Autoconfiguration and Prefix Continuity in IPv6 Hybrid Ad-Hoc Networks," Proceedings of the IEEE Sensor and Ad Hoc Communications and Networks (IEEE SECON '05), pp. 107-117, Sept. 2005.

[2] OPNET, "OPNET Technologies, Inc. - Modeler." on-line: http://www.opnet.com , April 2007.

[3] M. Günes and J. Reibel, "An IP Address Configuration Algorithm for Zeroconf Mobile Multihop Ad Hoc Networks," Proceedings of the International Workshop on Broadband Wireless Ad Hoc Networks and Services, Sophia Antipolis, France, Sept. 2002.

[4] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network", Proceedings of the 21st IEEE Conference on Computer Communications (IEEE INFOCOM '02), Vol. 2, pp. 1059-1068, June 2002.

[5] C. Perkins, J. Malinen, R. Wakikawa, Y. Sun, E. M. Belding-Royer, "IP Address Autoconfiguration for Ad Hoc Networks", IETF Internet Draft marked draft-perkins-manet-autoconf-01.txt, 2001.

[6] N. Vaidya, "Weak Duplicate Address Detection in Mobile Multihop Ad Hoc Networks," Proceedings the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '02), pp. 206–216, Lausanne, Switzerland, June 2002.

[7] K. Weniger, "Passive Duplicate Address Detection in Mobile Ad Hoc Networks", Proceedings of the IEEE Wireless Communications and Networking Conference (IEEE WCNC '03), Vol. 3, pp. 1504-1509, March 2003.

[8] Y. Sun, E.M. Belding-Royer, "A Study of Dynamic Addressing Techniques in Mobile Ad hoc Networks", Wireless Communications and Mobile Computing, Vol. 4, Issue 3, pp. 315-329, April 2004.

[9] K. Weniger, "PACMAN: Passive Autoconfiguration for Mobile Ad-Hoc Networks," IEEE Journal on Selected Areas in Communications, Vol. 23, Issue 3, pp. 507-519, March 2005.

[10] B. Wehbi, "Address Autoconfiguration in Ad Hoc Networks", Institut National des Telecommunications Internal Report, on-line: http://www.bachwehbi.net/autoconf_report.pdf, May 2005, accessed April 2007.

[11] K. Weniger and M. Zitterbatt, "Address Autoconfiguration in Mobile Ad Hoc Networks: Current Approaches and Future Directions," IEEE Network, Vol. 18, Issue 4, pp. 6-11, Aug. 2004.

[12] L. Lamont, M. Wang, L. Villasenor, T. Randhawa, and S. Hardy, "Integrating WLANs and MANETs to the IPv6 based Internet," Proceedings of the IEEE International Conference on Communications (IEEE ICC '03), Vol. 2, pp. 1090-1095, May 2003.

[13] C. Wang, C. Li, R. Hwang, and Y. Chen, "Global connectivity for mobile IPv6-based ad hoc networks," Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05), Vol. 2, pp. 807–812, March 2005.

[14] K. S. King and N. Smith, "Dynamic Addressing and Mobility in Tactical Hybrid Ad Hoc Networks", Proceedings of IEEE Military Communications Conference (IEEE MILCOM '05), Vol. 5, pp. 2930-2935, Oct. 2005.

[15] H. Ammari and H. El-Rewini, "Integration of Mobile Ad Hoc Networks and the Internet Using Mobile Gateways," Proceedings of the 18th International Parallel and Distributed Processing Symposium, pp. 218-225, April 2004.

[16] M. K. Denko and C. Wei, "An Architecture for Integrating Mobile Ad Hoc Networks with the Internet using Multiple Mobile Gateways," Canadian Conference on Electrical and Computer Engineering, pp. 1097–1102, May 2005.

[17] Y. Mo, J. Huang, and B. Huang, "MANET Node Based Mobile Gateway with Unspecific MANET Routing Protocol," International Symposium on Communications and Information Technologies (ISCIT '06), pp. 886-889, Oct. 2006.

[18] D. B. Green and R. Reddy, "Stryker Brigade Combat Team IPv6 Transition Modeling and Simulation Study", IEEE Military Communications Conference (IEEE MILCOM '05), Vol. 4, pp. 2275-2281, Oct. 2005.

[19] L. Kant, F. Anjum, and K.Young, "Design and Analysis of Scalable Network Centric Warfare Mechanisms", Proceedings of the 25th Army Science Conference (ASC '06), PO-03, Nov. 2006.

[20] C. Jelger, T. Noel, and A. Frey, "Gateway and Address Autoconfiguration for IPv6 Adhoc Networks," IETF Internet Draft marked draft-jelgermanet-gateway-autoconf-v6-02.txt, April 2004.

[21] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," Proceedings of the 22nd IEEE Conference on Computer Communications (IEEE INFOCOM '03), Vol. 2, pp. 1312-1321, March 2003.

[22] A. Frey, "Autoconfiguration in IPv6 Wireless Ad Hoc Networks – Stand Alone Daemon." Oct. 12, 2004, on-line: http://www-r2.u-strasbg.fr/~frey/safari/autoconf.html, accessed April 2007.

[23] M. Takai, J. Martin, and R. Bagrodia, "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks," Proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '01), pp. 87-94, Long Beach, California, USA, Oct. 2001.

[24] C. Jelger, "Gestion des équipments mobiles et communications de group dans l'Internet Nouvelle Génération," Ph.D. Thesis (in French), Université Louis Pasteur, Strasbourg, France, Oct. 2004.