A Study of Dynamic Addressing Techniques in Mobile Ad hoc Networks

Yuan Sun Elizabeth M. Belding-Royer
Department of Computer Science
University of California, Santa Barbara
{suny, ebelding}@cs.ucsb.edu

***Abstract*—Dynamic address assignment enables nodes in mobile ad hoc networks to obtain a routable address without the need for any explicit configuration. It provides a means for nodes to communicate without any centralized infrastructure, and provides a mechanism for dynamic network membership. Recently, a considerable number of dynamic addressing protocols have been proposed. While these approaches bear some similarities to each other, they also differ in some important characteristics. To understand the benefits of these different approaches, it is necessary to test the protocols in a wide range of network conditions so that their performance and suitability can be predicted. This paper studies existing solutions by categorizing and qualitatively analyzing the scalability and other performance properties of the approaches. We also introduce a new addressing approach the provides both quick and efficient unique address assignment. We then compare selected protocols through quantitative analysis based on extensive simulations. Based on the simulation results, we point out the applicability of the protocols and offer suggestions to improve protocol performance.**

I. Introduction

An ad hoc network is comprised of mobile nodes that communicate solely over the wireless medium. It provides a flexible means of communication when there is little or no infrastructure, or the existing infrastructure is inconvenient or expensive to use. Significant research in this area has focused on the design of efficient routing protocols that are suitable for the characteristics of mobile ad hoc networks, including mobility, limited power and limited transmission range. Many of the current routing protocols can be divided into two general categories [17]: proactive routing protocols such as DSDV [13] and OLSR [3], and reactive routing protocols such as AODV [15] and DSR [10]. Other routing protocols, including ZRP [8], combine both proactive and reactive approaches.

The majority of these routing protocols assume that mobile nodes in ad hoc networks are configured *a priori* with a unique address before they communicate in the network. However, not all nodes have pre-assigned static addresses if they usually obtain their addresses through a dynamic mechanism on the wired network. Also, because mobile nodes may frequently move from one network to another, it is desirable for them to obtain addresses via dynamic configuration. Hence, dynamic addressing is an essential component of building such a self-organizing network system.

Recently, a number of addressing schemes for ad hoc networks have been proposed, all of which aim to provide efficient address assignment in a dynamic network environment so as to enable correct communication in the network. These approaches bear many similarities to each other, such as self-organizing, self-healing behavior in order to better adapt to the dynamic and resource-constrained environment of the ad hoc network. However, these approaches also differ in a wide range of aspects, such as address format, usage of centralized servers or full decentralization, hierarchical structure or flat network organization and explicit or implicit duplicate address detection. As a consequence, these approaches are likely to have different performance properties under varying network conditions. The different performance properties, such as scalability, play an important role on the protocol applicability. For example, due to the limited resources of the mobile devices as well as of the wireless network, the address assignment process should not incur significant traffic load on the network. Limiting the control traffic generated by the assignment process enables the protocol to more easily scale to larger networks. On the other hand, a node should obtain a valid address in a timely fashion, regardless of the network size, so that its communication is not delayed. To deploy large-scale mobile ad hoc networks, it is important to understand and analyze these different approaches under a wide range of network conditions so that their performance and suitability can be predicted.

This paper studies a number of proposed addressing solutions by categorizing and qualitatively analyzing the

performance properties, including the scalability of the approaches. Further, we compare representative protocols from each category and provide quantitative analysis based on extensive simulations. Some of the approaches have been evaluated separately in their original publications. However, due to differing experimental setups, the evaluation results are not directly comparable. Our simulations are designed to test the protocols under network conditions of varying network sizes, node mobility and traffic rates, as well as network events such as merges and partitions. Based on the simulation results, we point out the applicability of the protocols and offer suggestions to improve protocol performance for specific problems. Specifically, the contributions of our paper are the following:

- Overview and categorization of many of the proposed approaches for address autoconfiguration.
- Introduction of a new addressing solution that offers quick address assignment with little control overhead.
- Quantitative evaluation of the proposed solutions through analysis and simulation.
- Discussion of modifications to improve the performance of the studied protocols.

The remainder of the paper is organized as follows. Section II examines the requirements and objectives for addressing protocols in mobile ad hoc networks. Section III categorizes and summarizes previously proposed addressing mechanisms. Section IV analyzes common protocol characteristics as well as compares their different features. Next, our evaluation of the protocols and performance results are presented in section V. Finally, section VI concludes the paper.

II. Requirements and Objectives

In this section, we first examine the need to avoid or detect duplicate addresses for dynamic addressing. Then the requirements to solve the addressing problem in mobile ad hoc networks are analyzed. Finally, we state the objectives of an optimal addressing solution.

A. Duplicate Address Detection

An important requirement of address autoconfiguration is to provide duplicate-free address assignment and to be able to detect duplicate addresses if they occur. Duplicate addresses may arise during the initialization of a group of nodes if the nodes are not in direct transmission range of each other. Also, nodes in different networks may independently obtain the same address, and duplicate addresses will result if these networks later merge. If two nodes in a single network have the same address, it becomes impossible to distinguish the nodes solely by their addresses. Hence, if communication with one of the nodes is desired, a route to the correct address may not necessarily be a route to the correct node.

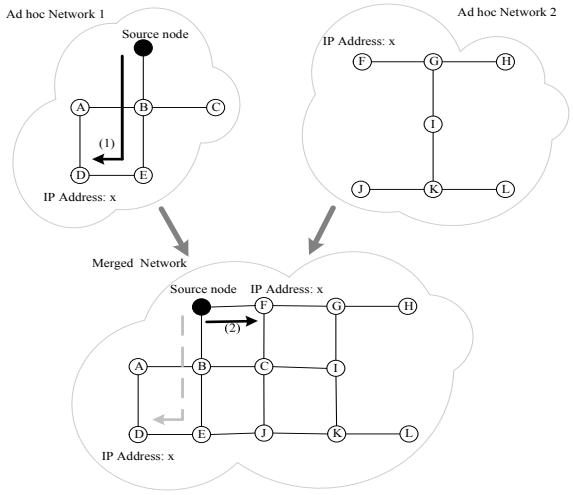


Fig. 1. Example of duplicate addresses and erroneous routing. Nodes D and F have the same IP address *x*. Erroneous routing occurs when these two networks merge.

Figure 1 shows an example of the existence of duplicate addresses and the erroneous routing that can result. Nodes D and F obtain the same address, **x**, independently in different networks. A source node in network 1 initiates a session and communicates with node D through the routing path indicated by arrow (1). This path has a length of three. When the two networks move towards each other and subsequently merge, node F enters the direct transmission range of the source node. If proactive routing protocols are used, after receiving the periodic routing information update the source node will update its routing entry for address **x** to the direct path to node F with path length of one. If reactive routing protocols are used, a new route discovery process will also result in the route to address **x** through the new path indicated by arrow (2). Thus, erroneous routing prevents the source from communicating with the correct destination.

In [22], the notion of Strong and Weak DAD are introduced. Strong DAD implies no duplicate addresses exist in the network at any time. On the other hand, the goal of Weak DAD is to prevent a packet from being routed to a wrong destination even if two nodes in the network happen to have chosen the same address. In [22], the author states that *Strong DAD cannot be guaranteed if message delays between at least one pair of nodes in the network are unbounded*, which is a likely occurrence in dynamic ad hoc networks; however, Weak DAD can be achieved through modifications of routing protocols. In [23], the author proposed Passive DAD, in which the detection of duplicate addresses is accomplished passively by continu-

ously monitoring routing protocol traffic. Proactive routing information, such as link state exchange, is needed to detect the address duplicates.

Each of the existing approaches tries to ensure the correctness of address assignment by utilizing one of the duplicate address detection techniques above. For approaches that seek Strong DAD, a single address assignment will require approval from all the nodes in the network. The scalability of the protocol is therefor strongly correlated with the organization of the network, i.e., a flat structure, or a hierarchical one. If the former case, every address acquisition will result in extra traffic throughout the whole network; however, only group leaders need to take action in the latter case. For the Weak and Passive DAD approaches, the overhead and latency for address resolution is correlated with the overhead of the routing protocols. Weak DAD can be integrated with both proactive and reactive routing protocols; however, passive DAD is likely to only work with proactive routing protocols. Hence Passive DAD may suffer performance degradation in large networks or high mobility [2].

B. Ad hoc Network Dynamics

Due to the inherent mobility of ad hoc networks, membership is highly dynamic; nodes may join or leave the network at any time. Also, because of their limited power, the energy of a node may become depleted and thus the node will become disconnected. In some scenarios, such as a conference, a group of people with mobile devices may gather together in one large group or split into smaller separate groups. Hence, network partitions and merges are likely to be frequent occurrences in mobile ad hoc networks. As indicated in [9], the following are common network events that should be handled by an address autoconfiguration protocol:

- **Independent Initializations**: A node independently boots and acquires an address. This is the simplest scenario. It provides the foundation upon which other scenarios build.
- **Group Initializations**: In this case, a group of mobile nodes initialize together within a short time frame. This is a typical scenario where multiple mobile nodes simultaneously boot to form a network and cooperate with each other to accomplish a certain task.
- **Network Joins**: This scenario indicates that a new node, without a pre-configured address, enters an existing ad hoc network and obtains an address.
- **Network Partitions**: In this case, a network splits into two or more partitions. After the network partition, new nodes may join either partition and acquire addresses. Thus, duplicate addresses may occur if these partitions later merge.
- **Stand-alone Networks Merge**: Two networks may move into the transmission range of each other and a merge occurs. Duplicate addresses may exist before the merge, resulting in erroneous routing after the merge.

C. Objectives

Based on the various ad hoc network dynamics described in section II-B and the importance of eventual duplicate address detection, we now list the objectives of an optimal ad hoc network address configuration protocol:

- **Dynamic Address Configuration**: Nodes should be able to dynamically obtain IP addresses without manual or static configuration.
- **Uniqueness**: Nodes should obtain unique addresses for correct routing and communication.
- **Robustness**: The addressing protocol should adapt to the dynamics of the network, including partitions and merges.
- **Scalability**: The protocol should avoid significant performance degradation as the size of the network increases.

There are three properties included in the scalability requirement. First, the address space should not be entirely consumed, even as the network size grows. Address reclamation is an important feature to ensure address reuse. Second, nodes should obtain valid addresses in a timely fashion. Finally, nodes should obtain routable addresses with minimum control overhead, without having an adverse affect on ongoing communication in the network.

III. Categorization of Approaches

In this section, we first summarize many of the proposed addressing mechanisms for ad hoc networks and categorize them into three groups: *decentralized* approaches, *leader-based* approaches, and *best effort* approaches. In addition, we describe approaches that utilize IP addresses, as well as approaches that diverge from IP-based addressing.

The decentralized protocols approach the problem of dynamic address assignment in ad hoc networks as a distributed agreement problem. Each node independently proposes a candidate IP address, and the address is validated upon agreement from *all* nodes in the network. In leader-based approaches, on the other hand, nodes obtain valid IP addresses from an elected leader of the network. Both decentralized and leader-based approaches include methods to avoid or detect duplicate addresses during address assignment, as well as to detect partitions and merges. Best effort approaches, however, allow nodes to assign their own addresses without the involvement of all

other nodes in the network. Hence, the potential for duplicate addresses exists. A detection mechanism is often utilized to avoid duplicates and ensure correct communication.

A. Decentralized Approaches

Ad hoc Address Autoconfiguration (AAA)*:* In this Internet Draft [14], addresses are randomly selected from the address range 169.254/16. Duplicate address detection (DAD) is performed by each node to guarantee the uniqueness of the selected address. During this process, a node floods an Address Request message in the network to query for the usage of its tentative address. If the address is already in use, an Address Reply message is unicast back to the requesting node so that a different address can be selected. The absence of an Address Reply indicates the availability of the requested address. This approach does not consider complex scenarios such as network partitions and merges.

MANETconf*:* In MANETconf [11], each node maintains a list of all IP addresses in use in the network. A new node obtains an IP address through an existing node in the network; the latter performs an address query throughout the network on the new node's behalf. This address allocation requires a positive acknowledgment (ACK) from all known nodes indicating the address is available for use. Each node in the network also agrees on a partition ID to detect partitions and merges. A network partition is detected when the node performing address assignment for a new node fails to obtain ACKs from all other nodes in the network. After the detection, the set of nodes from whom an ACK was not received is deleted from each node's list of in-use addresses. The nodes then agrees on a new partition identifier. When partitions merge, nodes in different partitions are required to exchange their set of allocated addresses so that duplicates can be detected.

Other Approaches*:* In [21], the authors propose an addressing scheme (which we call the Buddy approach) based on the buddy system used for memory management. Each node maintains a disjoint free-IP set. A configured node, i.e., a node that has an address, picks one address from its free set and assign it to a newly joined node, as well as allots half of its free set to the new node. Similar to MANETconf, a network identifier is maintained to detect network partitions and merges.

The IPv6 Stateless Address Autoconfiguration mechanism for wired networks is described in [20]. In this mechanism every node creates a link-local address and verifies its uniqueness on a link.

B. Leader-Based Approaches

Dynamic Address Configuration Protocol (DACP)*:* To ensure duplicate address detection while minimizing the participation of network nodes, we introduce a new approach, first described in [19] that utilizes an elected *Address Authority (AA)* to maintain the state information of the network, such as the node addresses, as well as their lease lifetime and a unique network identifier. However, this approach does not rely on a single leader to assign addresses. Instead, the address assignment is accomplished in a distributed fashion, similar to the AAA approach, in which nodes independently obtain a candidate address through a network-wide Address Request. They then register this address with an address authority (AA) if no rejection is received. The AA periodically broadcasts Network Identifier Advertisement messages so that partitions and merges can be detected in a timely fashion. Specifically, when nodes do not receive their AA advertisement for consecutive intervals, they detect the partition and elect a new AA. When the AAs of different partitions hear each other's advertisement, a network merge is detected and the AAs take the responsibility of detecting duplicate addresses in both partitions. Address changes only occur when a duplicate exists; only the nodes with the duplicate address must obtain a new address.

Optimized DACP (ODACP)*:* To isolate the overhead of the DACP approach resulting from the broadcast of the Address Request, we also introduce another version of DACP without duplicate address detection. The result is a pure leader based approach, which we call Optimized DACP (ODACP). The leader is elected in the same manner as in DACP, and every node registers its tentative address with the leader without flooding address requests. If the address is not already in use in the network, the leader verifies the registration; otherwise, it denies the registration request and indicates to the node that it must select a new address. As in DACP, the detection of merges and partitions is implemented by the leader advertisement.

Dynamic Address Allocation Protocol (DAAP)*:* In [12], the authors propose DAAP based on the concept of address assignment by a leader. The functionality of the leader is shared among all the nodes in the network. When a new node joins the network, it becomes the leader until the next node joins. The leader maintains the highest IP address within the ad hoc network, as well as a unique identifier associated with the network. Each node stores the highest address, which is the address of the leader, and sends hello messages periodically to its neighbors. These hello messages include the network identifier so that merges and partitions can be detected. When a node receives a hello message with a different partition identifier, it detects a merge; if a node does not

receive any messages containing the current partition identifier, then, after some timeout, a partition is detected.

Other Approaches*:* Two agent-based IPv6 autoconfiguration mechanisms in mobile ad hoc networks are presented in [7] and [24]. In these approaches, each node acquires a subnet ID from the agent, and then generates a link-local address based on its MAC address. The latter approach further creates a hierarchical network structure based on the leader management. Because these approaches are based on IPv6, a MAC address can easily fit into the IP address. Assignment of unique addresses is therefore trivial.

In addition to these approaches, the DHCP [4] protocol created for wired networks is another example of a leader-based approach.

C. Best Effort Approaches

Prophet Addressing*:* The prophet addressing approach [26] utilizes a stateful function *f(n)* to generate a series of random numbers. The first node *X* in the network sets its IP address and chooses a random state value as the seed for its *f(n)* to compute a sequence of addresses locally for the network. Another node can obtain an IP address from *X*, as well as a state value as the seed for its *f(n)*. The same process continues as nodes join the network. The function *f(n)* is designed in such a way that the possibility of duplicates is kept low.

Weak DAD*:* The basis of Weak DAD, presented in [22], is to prevent a packet from being routed to a wrong destination, even if duplicate addresses exist. A unique per-node key is included in the routing control packets and in the routing table entries. Then, if two nodes happen to have chosen the same IP address, they can still be identified by the use of their unique keys. Hence every node is identified by a unique tuple $< address, key >$. The authors of [22] suggest using a node's MAC address as its key.

In our experimental study, we introduce a new addressing approach based on this idea that modifies current routing protocols to provide best effort address assignment and a resolution mechanism when duplicate addresses are detected. Specifically, each node randomly picks an IP address in a certain range without requesting approval from other nodes in the network. Duplicate addresses are detected using *lazy detection*, whereby only when traffic is sent and routing information is exchanged can duplicates be detected. As a result, this approach does not explicitly detect network merges and partitions. It also allows nodes with duplicate addresses to co-exist until one of the nodes is utilized for routing.

Other Approaches*:* Passive DAD, in which the detection of duplicate addresses is accomplished passively by continuous monitoring of the routing protocol control traffic, is proposed in [23]. The approach is based on the properties of link state routing protocols, whereby nodes use periodic link state routing information to notify other nodes about their neighborhood. The paper suggests three techniques for utilizing link state information to detect duplicates. These techniques are: sequence number based, locality principle based, and neighborhood based detection. The basic idea of these techniques is to provide unique addresses within the two hop neighborhood. Then, if there exists at least one node with a unique address, duplicates can be detected. Although this approach does not incur extra overhead in the network, it relies heavily on the underlying routing protocol; the correctness and effectiveness are affected by the particular parameter settings of the routing protocol. Further, the approach only considers networks with low node mobility.

D. Non-IP approaches

Non-IP approaches can also be placed into the above categories. Specifically, [1] utilizes variable length addresses through a mechanism similar to MANETconf. Like MANETconf, it also belongs to the decentralized category. The addressing scheme presented in the MMWN system [16] utilizes a leader-based method and creates a hierarchical structure. Finally, the Address Free Architecture presented in [5] falls into the best effort category.

IV. Characteristics Analysis

In this section, narrow our focus by providing a qualitative analysis and comparison of all the proposed approaches that utilize IPv4 addressing. Quantitative evaluation of selected protocols is presented in section V.

A. Common Characteristics

Because ad hoc networks typically consist of mobile devices with limited resources, limited bandwidth and no pre-existing infrastructure, all the protocols presented in section III support the features of self-organization and self-repair. In this section, we describe four common attributes that are exhibited in many (or all) of the described approaches.

1) Unique Keys: In almost all solutions, a unique key is utilized as a secondary identifier of a node. These include the MAC-keys in [22] and the Random Source ID (RSID) in [23]. By the use of a key, in the event two nodes have the same address, the nodes can still be distinguished. An important issue is determining a mechanism to obtain this key. One possibility utilized by many

Categories	Decentralized			Leader Based		Best Effort	
Approaches	AAA	MANETconf	Buddy	DACP	DAAP	Weak DAD	Prophet
Duplicate Possibility during Assignment	Yes	No	No	No	No	Yes	Yes
State Maintenance	No	Yes	Yes	Yes	Yes	No	No
Critical Nodes	No	No	No	Yes	Yes	No	No
Periodic Message	No	Yes	Yes	Yes	Yes	Yes	No
Change of Addresses During Merges	Node with Duplicate	Node with Duplicate	Node with Duplicate	Node with Duplicate	Node with Duplicate	Node with Duplicate	All Nodes
Communication Overhead	$O(r \times N^2)$	$O(r \times N^2)$	$O(r \times h \times N)$ *	$O(r \times N^2)$ †	$O(r \times h \times N)$	0	$O(d \times N)$
Addressing Latency	$O(r \times D \times t)$	$O(r \times D \times t)$	$O(r \times h \times t)$ ‡	$O(r \times D \times t)$ §	$O(r \times h \times t)$	0	O(t)

TABLE I

COMPARISON OF THE DYNAMIC ADDRESSING APPROACH CHARACTERISTICS.

Abbreviation:

N = Number of nodes in the network D = Network diameter r = Number of retries

d = Average node degree t = Average 1-hop latency h = Average hop number

approaches is to utilize the MAC address as the key. Although it is possible for MAC addresses to be duplicates, the combination of the IP and MAC address is unique with very high probability. Further, duplicate MAC addresses can be detected by the solution described in [18].

2) Distributed Communication: Nearly all of the proposed approaches use a distributed communication method. For leader based approaches, the leader is either elected through distributed communication as in DACP, or the role of leader is distributed to each node in the network as in DAAP. In best effort approaches such as Weak DAD, distributed routing information exchange is utilized to detect and avoid duplicate addresses.

3) Detection of Network Events: Most approaches use explicit mechanisms to detect network events such as partitions and merges. The detection is normally accomplished by utilizing a unique network identifier. This identifier is either broadcast throughout the network by a leader node, or it is contained in periodic hello messages exchanged between neighbors.

4) Soft State Information: Most of the approaches utilize periodic signaling as well as timeout mechanisms to handle network events. The higher the signaling frequency, the more quickly the network can detect and adapt to network events. However, the overhead increases as the signaling rate increases. Given the scalability requirement, finding an optimal value is non-trivial with different network environments.

B. Qualitative Comparison

As described in section IV-A, all the approaches share some common characteristics. However, they also differ from each other in a wide range of aspects. Table I presents a comparison of the characteristics of the IPv4 addressing protocols.

In decentralized approaches, the assignment of an address typically requires permission from all nodes in the network by either an explicit positive acknowledgment (*ACK*) or the absence of negative acknowledgment (*NACK*). Network-wide flooding is often utilized and leads to high communication overhead ($O(N^2)$) and high latency ($O(D \times t)$). In a mobile wireless network, packet loss and link breaks are potentially frequent occurrences. Hence performance may degrade or duplicates may exist if critical messages are lost due to collisions or the changing network topology.

In leader-based mechanisms, the leader assumes the responsibility of address assignment. The communication overhead is therefore comparatively low. The leader is typically elected in a distributed fashion that is robust to dynamic topology changes. However, node movement can also cause frequent leader and network identifier changes, thereby bringing instability to the network. In addition, in order for the leaders to maintain complete knowledge of assigned addresses, significant communication overhead may be required. Duplicate addresses may also exist if leaders do not have a correct view of the network.

Best effort approaches, on the other hand, allow nodes autonomous assignment of addresses. This process has little overhead. However, since address assignment is not based on network-wide view, duplicate addresses may exist after address allocation.

The described approaches have varying susceptabilities to security attacks; these are briefly summarized in table II. One major issue is *denial of service* attacks, in which a malicious node can return false information so as to block certain addresses. A malicious node can also repeatedly flood messages to request different addresses and

* worst case is $r \times N^2$.

† ODACP has overhead O(r×h×N).

‡ worst case is r×D×t.

§ ODACP has latency O(r×h×t).

Approaches	Decentralized	Leader-Based	Best Effort
Denial of Service	✓	✓	×
Depletion of Network Resources	✓	×	×

TABLE II
SECURITY PROBLEMS FOR EACH PROTOCOL CATEGORY

deplete the limited network resources. Currently, there are no solutions to prevent these types of attacks; however, there are mechanisms to detect these attacks. For instance, to detect these problems, an intrusion detection system, as described in [25], can be deployed.

V. PERFORMANCE STUDY

In section IV-B, we presented a qualitative comparison of the protocols. In order to obtain further understanding of the strengths and weaknesses of the protocols, we simulate representative protocols from each category described in section III. Specifically, we focus on IPv4 addressing solutions to enable a fair comparison. The simulations are performed in a variety of networks. Quantitative evaluation allows us to draw conclusions regarding the applicability of the protocols within each protocol family. Specifically, our performance study has the following goals:

- Compare the protocols over a wide range of network scenarios.
- Determine the scenarios in which each protocol performs well, as well as environments where the protocols exhibit performance degradation.
- Propose protocol enhancements to improve performance.

In the following sections, we first justify our selection of protocols to evaluate. We then provide a description of our experiment setup. Performance results are then presented, followed by our observations.

A. Selected Protocols

MANETconf and AAA are chosen to represent the decentralized approaches. In general, decentralized approaches seek consistency of the network state by obtaining an agreement from all nodes in the network for each address assignment. This consistency is accomplished by either ACKs from all nodes, or the absence of a NACK from any node. MANETconf utilizes the first mechanism, while the AAA approach exercises the latter. We use the implementation of MANETconf from the authors of [11], with modification to support detection of merges and partitions as specified in their original paper. Because MANETconf requires information provided by routing protocols, we simulate MANETconf with DSDV and AODV as representatives of both proactive and reactive routing protocols. We also implement the AAA approach according to the Internet Draft [14]. We do not study the Buddy approach in [21] because it has the potential problem of address leakage when message loss occurs during free set exchanges.

DACP is selected as a representative of the leader based approaches because it has even address distribution. We use the implementation of DACP described in [19]. The DACP approach utilizes a duplicate address detection process that can incur significant overhead. To understand the overhead generated by the broadcast Address Request, we also study the ODACP approach. ODACP only utilizes address registration with the Address Authority, and hence is likely to require less overhead than DACP. We do not study DAAP because it is not well-specified how to maintain correct information about the leader to deal with a changing network topology [12]. Further, their mechanism to detect partition only works when the partition contains no more than one node.

Weak DAD is chosen to represent best effort approaches. We utilize the basic idea proposed in [22] and implement random address generation. We also modify the routing tables and the route discovery process by including a key associated with each node address. We choose not to implement Prophet addressing because the method is essentially a random number generation process with a specially designed function *f(n)*. We do not study Passive DAD because it only potentially works with link state routing protocols and the performance is tightly dependent on the parameter setting of the underlying routing protocols.

To enable a fair comparison, we do not study IPv6 or variable length addressing approaches; we focus solely on IPv4 solutions and leave the other categories as future work. In summary, we perform simulations on following six protocols: MANETconf-DSDV, MANETconf-AODV, AAA, DACP, ODACP, and Weak DAD.

B. Evaluation Methodology

To achieve the evaluation goals presented earlier in section V, we focus on a three step study. Each step is outlined below. All approaches are implemented in the NS-2 [6] simulator with the CMU mobility extensions [2]. Each data point represents an average value of 10 runs with the same settings, but different randomly generated mobility scenarios. The simulation time for the first two sets of experiments is 100 seconds, while the third set is 250 seconds. The maximum number of retransmissions of addressing control packets is set to two.

The protocols are evaluated using the following criteria:

Number of Nodes	Network Size	Time Interval
10	330m × 330m	5 seconds
20	670m × 670m	10 seconds
50	1000m × 1000m	25 seconds
100	1500m × 1500m	50 seconds
150	1800m × 1800m	75 seconds
200	2100m × 2100m	100 seconds

TABLE III
SIMULATION PARAMETERS FOR STUDY 1.

- **Correctness**: Correctness implies no duplicate addresses exist after address assignment, and no erroneous routing occurs. We check for duplicates in the assigned addresses and subsequent routing table entries during the simulation to verify correctness.
- **Efficiency**: Efficiency indicates that a node can obtain a valid address in a short time without incurring significant overhead. This includes two performance metrics:
  - **Latency for Address Autoconfiguration**: This measurement represents the average latency for a node to obtain a unique IP address within the network. This includes all possible delays caused by the message exchanges and timeouts.
  - **Communication Overhead**: This measurement is the number of control packets transmitted during the initialization of a node. This includes all broadcast messages, neighbor exchanges and unicast control packets. Each hop-wise packet transmission is counted as one transmission.
- **Scalability**: Scalability requires that the performance of the protocols does not experience significant degradation as the network size increases, or as the mobility and traffic load increase. We evaluate the performance metrics described above with increasing network size, varying node mobility and network traffic load to investigate the scalability of each protocol.
- **Stability**: Stability implies that the network address assignment should be stable and should not change frequently. In other words, the protocol should be resilient to topology changes. We also measure the frequency of network identifier changes during the simulation as an indicator of protocol stability.

We now describe the studies used in our evaluation.

1) Study 1 - Static Networks: The first set of simulations considers a group of nodes initializing together within a short time interval to form a network. In this study, we focus on static networks with varying network sizes; no mobility or traffic is introduced into the simulations. The number of nodes in the network with the sizes of simulated area is shown in table III. These parameters investigate the scalability of the approaches by studying a variety of network sizes, while maintaining an average node density of seven. All nodes initialize uniformly within a specific time interval as indicated in the table. Through this set of simulations, the protocol correctness, efficiency, and scalability in a static environment are examined.

2) Study 2 - Mobile Networks: In this study, we focus on the effect of mobility on the network initialization and maintenance. Due to mobility and subsequent link breaks, control packets may be lost or suffer long latency, which will potentially affect the correctness and efficiency of the protocols. Mobility will also affect the stability and robustness criteria during the maintenance of the network. Specifically, the random mobility of the nodes is likely to cause network partitions and merges.

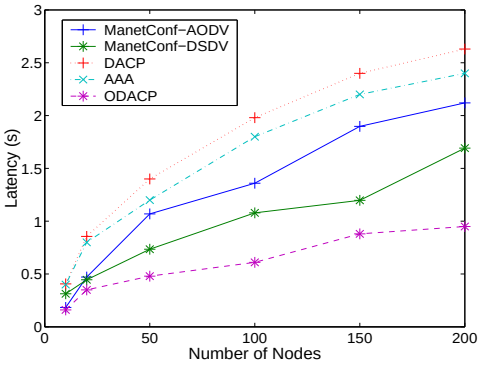
In this set of simulations, group initialization, as described in study 1, occurs in a fixed network size of 50 nodes. We vary the maximum node speed between 0 and 20 m/s; pause time is set to 0. The protocol efficiency is first examined, as well as the latency and communication overhead when the mobility of nodes increases. We then investigate the performance during the detection of network merges and partitions to determine the stability and robustness in a mobile environment.

3) Study 3 - Congested Networks: None of the previously published simulation studies by the protocol developers investigates the effect of varying traffic load on the protocols. However, data traffic is likely to introduce more collisions and message loss into a network. Hence, traffic load will have significant impact on the protocol performance.

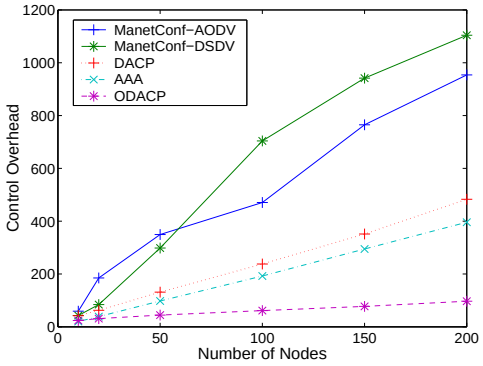
In this study, the effect of data traffic on the protocol performance, specifically protocol correctness, efficiency and robustness to message loss, is investigated. Because nodes can only conduct data transmissions after they obtain an address, we do not simulate group initialization as in the first two studies. Instead, we have a specified number of pre-configured nodes in the network that serve as traffic sources and destinations. New nodes then join the network with uniformly distributed inter-arrival times. We utilize CBR traffic with 10, 20, 30, and 40

Simulation Parameters	Value
Number of Nodes	50
Preconfigured Nodes	30
Inter-arrival Time Interval	20 seconds
Data Packet Size	512 bytes
Data Sending Rate	4 packets/second
Maximum Node Speed	5 m/s

TABLE IV
SIMULATION PARAMETERS FOR STUDY 3.



(a) Address Allocation Latency per Node.



(b) Control Overhead per Node.

Fig. 2. Address Allocation in a Static Network with Varying Number of Network Nodes.

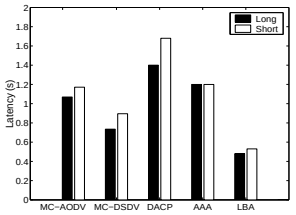
sources. The specific simulation parameters are shown in table IV.

C. Performance Results

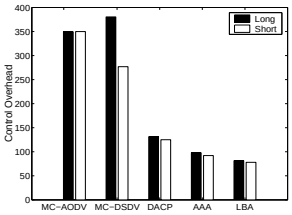
1) Study 1 - Static Networks: Figure 2 shows the efficiency of the different protocols as the number of nodes increases in a static network. As presented in figure 2(a), the average address acquisition latency of all approaches increases as the number of nodes increases. This is due to both longer paths that control messages need to traverse, as well as an increase in the number of collisions. ODACP has the lowest increase of latency because a new node must only contact the leader to obtain an address. On the other hand, AAA and DACP both have a more significant increase in latency due to the duplicate address detection process that floods AREQ messages throughout the network. The timeout value for reception of an AREP is based on the network diameter. Hence nodes wait longer for an AREP as the network size increases. DACP has consistently higher latency than AAA because of the node registration latency with the leader.

The addressing latency of MANETconf is higher than ODACP because every node must receive an ACK from every known node in the network. On the other hand, MANETconf outperforms DACP and AAA because once a node collects all ACKs, it does not need to wait for a timeout to validate the address. As shown in figure 2(a), using a reactive routing protocol increases the address acquisition latency for MANETconf due to the route discovery needed to route ACK messages back to the source.

The average number of control packets per node is shown in figure 2(b). Similar to the previous figure, ODACP has lower control overhead per node because each node only contacts the leader to obtain an address. In this case, the control packets include the overhead for leader lookup, address assignment request and reply between the node and the leader, as well as the periodic



(a) Addressing Latency.

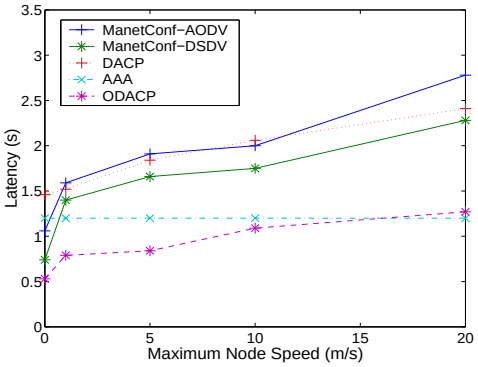


(b) Control Overhead.

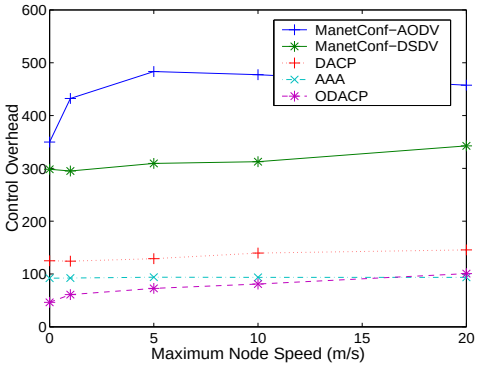
Fig. 3. Protocol Effi ciency with Different Initialization Time.

leader advertisement. The beacon interval for leader advertisement is set to 5 seconds. AAA and DACP have higher overhead because of the flooding of AREQ messages throughout the network. DACP additionally includes the registration messages, as well as the periodic leader advertisement messages. MANETconf has the highest communication overhead because for every address allocation, address request messages are sent to all known nodes in the network, and ACKs are returned from all nodes. MANETconf with DSDV has higher overhead than using AODV when the network size is large. This is because with a large network, the routing information generated by DSDV has not yet converged. This results in more message retransmissions. MANETconf with DSDV is particularly problematic because nodes need route information to transmit ACKs so that addresses can be assigned; however, DSDV routing information cannot converge until each node has an address.

Figure 3 shows the protocol performance when we apply different time intervals within which all nodes initialize. Here, the network is a 50 node static network. The Long time interval indicates that all nodes initialize within the range shown in table III, while the Short time interval values are 1/5th of the long values. All approaches except AAA have increased addressing latency when the time interval decreases, due to the increase in collisions. The overhead remains roughly constant with different time



(a) Address Allocation Latency per Node.



(b) Control Overhead per Node.

Fig. 4. Address Allocation of 50 Node Network with Varying Node Mobility.

factors, except for MANETconf with DSDV. In this scenario, complete routing information is not available for control message transmission during initialization.

In addition to these protocols, we also simulated the Weak DAD approach. Because each node randomly picks an address without communication with the other nodes, the addressing latency and control overhead are both zero. Hence, we do not include it in the figures.

We have examined the efficiency of all approaches in terms of addressing latency and communication overhead. During the simulations, all nodes were verified to acquire unique IP addresses in the networks, indicating that all approaches can correctly perform address allocation in a static network. In figure 2, we can see that the decentralized approaches do not scale well when the network size increases because the approaches utilize network-wide broadcast; if ACKs from all nodes are required, the control overhead is even greater. On the other hand, the leader-based approach has the best scalability in both latency and overhead in a static network.

2) Study 2 - Mobile Networks: Figure 4 shows the protocol performance in a network of 50 mobile nodes. As shown in figure 4(a), the addressing latency of all approaches increases with increased mobility, except AAA. When mobility increases, there are more broken paths, resulting in higher message loss. For the leader-based approaches, nodes spend more time determining a path to the leader to obtain an address. For MANETconf, when a node requests an ACK from all nodes for an address allocation, broken paths result in either additional route discoveries or packet retransmissions. Hence the addressing latency is also increased. In the AAA approach, a node validates its tentative address after the timeout for receiving a NACK. Hence the latency is constant for all mobilities and is dependent on the timeout value. Finally, as in the static network, the addressing latency for DACP is still one of the highest because of the duplicate address detection process plus address registration latency.

Figure 4(b) presents the effect of mobility on communication overhead. ODACP has the greatest percentage of increase in overhead because nodes must retransmit registration packets to the leader when the path to the leader breaks. This introduces more packet transmissions to obtain an address assignment. The AAA and DACP approaches both have only a slight increase in control traffic when mobility increases. This is because most messages are broadcast, and so broken paths do not have a significant impact on the control packet exchange. The message transmission between nodes and the leader also benefits from the flooding advertisement. Hence mobility does not significantly affect the control overhead of DACP. For MANETconf, the overhead of the protocol is not significantly affected by the mobility; however, while not shown in the figure, it does result in a significant increase in the overhead of underlying routing protocols.

During the simulation, the network may initially consist of several partitions. As mobility increases, nodes in different partitions move into transmission range of each other; network merges then occur. As described in section IV, most of the approaches utilize a network identifier and advertisement messages to detect network merges

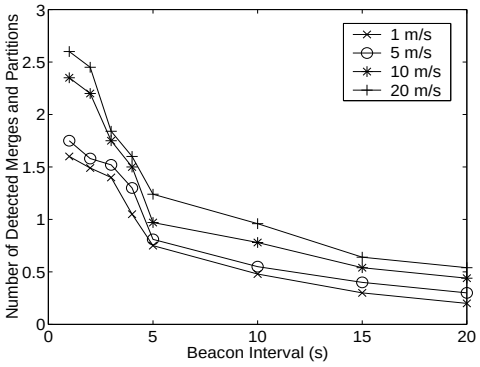
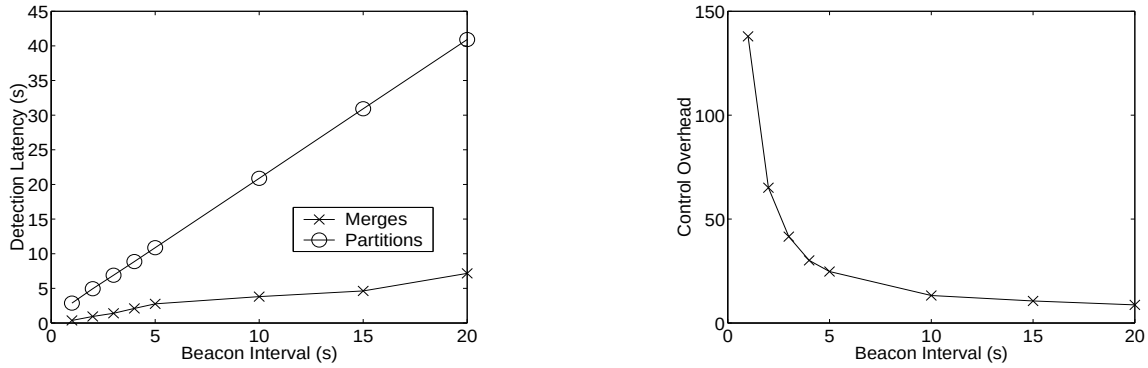


Fig. 5. Effects of varying beacon intervals on network merges.



(a) Latency. (b) Control Overhead.

Fig. 6. Event Detection with Varying Beacon Intervals.

and partitions. The advertisement frequency determines the timeliness to detect and adapt to these events. Figure 5 shows the number of merges and partitions detected in a 50 node network using different network identifier advertisement intervals. When the interval increases, the number of merges and partitions detected in the network decreases, indicating that the protocols are less sensitive to the network events.

Figure 6 presents the latency and overhead of detecting merges and partitions with different advertisement intervals at maximum speed of 20 m/s. Detection of network merges represents a tradeoff between assurance of unique addresses and overhead, and potential network instability. The number of network merges that are detected and the timeliness in which they are detected is directly influenced by the network identifier, or hello message, advertisement frequency. The more often these messages are sent, the quicker merges can be detected and address duplicates can be eliminated. However, a greater advertisement frequency results in greater overhead. Further, the action taken after a merge will influence the stability of the network. Some protocols, such as Prophet, require all network nodes to obtain a new address after a merge occurs. Hence, while duplicates will be prevented, the network-wide change of address will disrupt all open connections and result in substantial overhead. On the other hand, other protocols that only require nodes with duplicate addresses to obtain new addresses will be more stable and will not be detrimentally impacted by the number of detected merges.

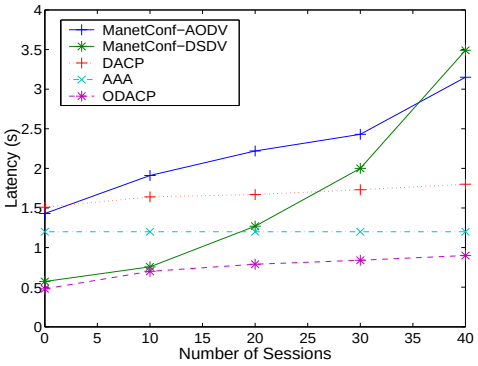
We now analyze the latency and communication overhead in the worst case to detect a network merge. Detection can either be done through periodic advertisement messages or through hello messages.

If advertisement messages are utilized, one node in each partition must maintain the network identifier. This node periodically broadcasts the identifier. When two such nodes from different networks receive each other's beacon message, they detect that the network has merged. They then exchange their IP lists to determine whether duplicate addresses exist. On the other hand, if hello messages are utilized, the hello messages contain the network identifier. When two nodes from different partitions receive hello messages from each other, the merge is detected through the different identifiers. The two nodes then exchange IP address lists and flood the lists throughout their own partitions so that all nodes in the network can detect address duplicates.

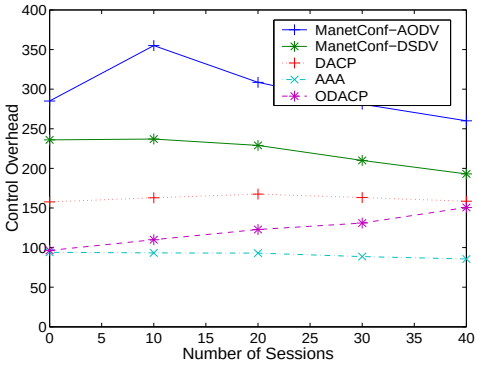
Let the interval for advertisements be Interval_{adv}, and the hello interval Interval_{hello}. Let the one-hop packet transmission latency be *t*, number of nodes in the merged network be *N*, and let *D* be the diameter of the merged network. Then, in the worst case, the latency for detecting network merges using periodic advertisements is the advertisement interval plus the latency of the advertisement propagation from one network to another. The latency for the detection of merges utilizing hello messages is the hello interval. The overhead for detecting network merges and duplicate addresses when using advertisement messages is the unicast message to exchange IP lists, which in the worst case must traverse the network diameter; when hello messages are used, the overhead is caused by flooding the IP lists through the network. Hence, we have

$$\begin{aligned} \mathbf{T}_{adv} &= \text{Interval}_{adv} + D \times t \\ \mathbf{T}_{hello} &= \text{Interval}_{hello} \\ \mathbf{Overhead}_{adv} &= 2 \times D \\ \mathbf{Overhead}_{hello} &= N \end{aligned}$$

We can see that utilizing leader advertisement messages to detect the merges may result in longer latency, and the overhead is dependent on the advertisement interval. Note here we do not include the periodic messaging in the overhead calculation because in both cases they are equal.



(a) Address Allocation Latency per Node.



(b) Normalized Control Overhead per Node.

Fig. 7. Address Allocation of 50 Node Network with Varying Number of Sessions.

3) Study 3 - Congested Networks: Figure 7 shows the efficiency of the protocols in a congested network. When the number of sessions increases and more traffic is introduced into the network, more collisions and message loss occurs. As shown in figure 7(a), MANETconf has the most significant increase of addressing latency when the number of sessions increases. Because it requires ACKs from all known nodes in the network, a single message loss requires significant retransmission of MANETconf control packets. Other approaches do not have significant control overhead, and so network congestion only causes a slight increase in latency.

The effect of traffic on control overhead is shown in figure 7(b). ODACP has the most significant increase in control overhead due to loss and retransmission of the registration messages. For MANETconf, because it already has high control overhead for address allocation without data traffic, the introduced data traffic prevents nodes from acquiring the wireless channel and transmitting packets. This causes packet drops at the local interface queue to increase. Hence, because some control packets are not even transmitted, MANETconf suffers from message loss and the control overhead actually decreases when the number of sessions increases. Further, we noticed that there are more partitions detected using MANETconf in a congested network because nodes suffer persistent message loss due to continuous congestion on certain paths. ACKs cannot be received from certain nodes and a partition is assumed to have occurred. For AAA and DACP, because they take the absence of a NACK as an indication of an available address, the overhead does not increase significantly. However, although not shown in the results, the message loss results in the risk of duplicate address assignment.

In all the experiments, Weak DAD does not incur extra latency and communication overhead for address allocation. Hence it is not shown in the performance result figures. The one drawback of Weak DAD, however, is that it does increase the size of routing control packets. Table V shows the size increase using the AODV routing protocol, where x is the size of the original RERR packet (the size of the RERR packet is not fixed), and n is the number of destinations that utilize the broken path indicated in RERR packets.

Packet	AODV	Weak DAD Enhanced
RREQ	28 bytes	36 bytes
RREP	24 bytes	32 bytes
RERR	x bytes	$(x + 4 \times n)$ bytes
HELLO	24 bytes	32 bytes

TABLE V
PACKET SIZE FOR AODV AND WEAK DAD ENHANCED AODV.

D. Observations

In our experiments, it is observed that all approaches correctly allocate addresses to nodes in an ad hoc network in static, mobile, and congested scenarios. However, in real environments, message losses can result in duplicate address with the DAD and DACP approaches. Leader-based approaches have the best scalability in terms of protocol efficiency of address latency and communication overhead. To improve scalability, a hierarchical addressing structure based on the leader assignment can be utilized. Decentralized approaches have comparatively poor scalability due to the network-wide message flooding. When mobility increases, most approaches have decreased protocol efficiency. When the network becomes congested, control messages are prone to be lost, resulting in the performance degradation of decentralized approaches. Particularly when ACKs are required, the protocols do not show good robustness to message loss.

Weak DAD, as a best effort approach, takes the risk of assigning duplicate addresses to nodes and tries to resolve conflicts during the routing process. This does not intro-

duce extra latency or overhead into addressing; however, it requires modification of routing protocols and increases control packet size.

During the course of the experiments, we observed modifications that would improve protocol performance. For MANETconf, relying on existing routing protocols such as DSDV or AODV leads to high routing overhead, particularly with high mobility. We suggest that MANETconf be integrated with routing protocols so that routes can be learned during the flood of the address query. For DACP, the duplicate address detection process incurs significant overhead and latency. When mobility is low, this option can be eliminated and ODACP can be utilized. On the other hand, if mobility is high, the flooding of AREQ messages can actually help refresh the route to the leader. Finally, Weak DAD can be applied to both proactive and reactive protocols by indicating per-node keys as secondary identifiers.

VI. Conclusions

In this paper, we have investigated the problem of dynamic addressing in mobile ad hoc networks. We studied current solutions by categorizing and qualitatively analyzing scalability and other performance properties of the approaches. We introduced a new address assignment approach, the Leader-based Approach, that was shown to have low overhead while still ensuring the timely assignment of unique addresses. Further, we compared selected protocols and provide quantitative analysis based on extensive simulations. Through the experiments, we examined the applicability of the protocols in different network environments and offer suggestions to improve protocol performance.

In our comparison, we focused on the performance of IPv4-based addressing techniques. An important issue of further research would be to include a comparison of IPv6 approaches, as well as approaches that are not IP-centric. An additional problem of dynamic addressing is how the address of a destination node can be determined after the address assignment so that traffic can be initiated. It is likely that this can be combined with a service discovery protocol. The address can become a service, and a node can perform a look up to obtain the destination's address.

References

[1] J. Boleng. Efficient Network Layer Addressing for Mobile Ad Hoc Networks. In *Proceedings of the International Conference on Wireless Networks (ICWN'02)*, pages 271–277, Las Vegas, NV, June 2002.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the* 4^{th} *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM'98)*, pages 85–97, Dallas, TX, October 1998.

[3] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. In *Proceedings of IEEE INMIC*, Lahore, Pakistan, December 2001.

[4] R. Droms. Dynamic Host Configuration Protocol. http://www.ietf.org/rfc/rfc2131.txt, March 1997.

[5] J. Elson and D. Estrin. An Address-free Architecture for Dynamic Sensor Networks. Technical Report 00-724, Computer Science Department, USC, January 2000.

[6] K. Fall and K. Varadhan. ns Manual. *http://www.isi.edu/nsnam/ns/doc/*. The VINT Project.

[7] M. Gunes and J. Reibel. An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-Hoc Networks. In *Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services*, Sophia Antipolis, France, September 2002.

[8] Z. J. Haas and M. R. Pearlman. The Performance of Query Control Schemes for Zone Routing Protocol. In *Proceedings of the ACM SIGCOMM'98 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication*, pages 167–177, Vancouver, Canada, August/September 1998.

[9] J. P. Jeong, J.-S. Park, K. Mase, Y.-H. Han, B. Hakim, and J.-M. Orset. Dynamic Configuration of IPv4 Link-Local Addresses. *IETF Internet Draft, draft-jeong-manet-addr-autoocnf-reqts-00.txt*, August 2003. (Work in Progress).

[10] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[11] S. Mesargi and R. Prakash. MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, New York, NY, June 2002.

[12] P. Patchipulusu. Dynamic Address Allocation Protocols for Mobile Ad Hoc Networks. Master' s thesis, Comupter Science, Texas A&M University, August 1997.

[13] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, London, England, August/September 1994.

[14] C. E. Perkins, J. T. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun. Ad hoc Address Autoconfiguration. *IETF Internet Draft, draft-ietf-manet-autoconf-01.txt*, November 2001. (Work in Progress).

[15] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the* 2^{nd} *IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.

[16] R. Ramanathan and M. Steenstrup. Hierarchically-organized, Multihop Mobile Wireless Networksfor Quality-of-Service Support. *Mobile Networks and Applications*, 3(1):101–119, 1998.

[17] E. M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, 6(2):46–55, April 1999.

[18] C. Schurgers, G. Kulkarni, and M. B. Srivastava. Distributed Assignment of Encoded MAC Addresses in Sensor Networks. In *Proceedings of the* 2^{nd} *ACM International Symposium on Mobile*

Ad Hoc Networking and Computing (MobiHoc'01), pages 295–298, Long Beach, CA, October 2001.

[19] Y. Sun and E. M. Belding-Royer. Dynamic Address Configuration in Mobile Ad hoc Networks. Technical Report 2003-11, Computer Science Department, UCSB, March 2003.

[20] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. Request for Comments (Draft Standard) 2462, Internet Engineering Task Force, December 1998.

[21] M. R. Thoppian. A Protocol for Dynamic Configuration of Nodes in Manets. Master' s thesis, Comupter Science, University of Texas at Dallas, August 2002.

[22] N. H. Vaidya. Weak Duplicate Address Detection in Mobile Ad Hoc Networks. In *Proceedings of the* 3^{rd} *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, pages 206–216, Lausanne, Switzerland, June 2002.

[23] K. Weniger. Passive Duplicate Address Detection in Mobile Ad Hoc Networks. In *WCNC*, Florence, Italy, February 2003.

[24] K. Weniger and M. Zitterbart. IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks. In *Proceedings of European Wireless 2002*, Florence, Italy, February 2002.

[25] Y. Zhang and W. Lee. Intrusion Detection in Wireless Ad-hoc Networks. In *Proceedings of the* 6^{th} *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM'00)*, pages 275–283, Boston, MA, August 2000.

[26] H. Zhou, L. Ni, and M. Mutka. Prophet Address Allocation for Large Scale MANETs. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 2003.