Quorum Based IP Address Autoconfiguration in Mobile Ad Hoc Networks

Tinghui Xu and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
Email: txu@fau.edu, jie@cse.fau.edu

***Abstract*— *IP address autoconfiguration poses a challenge for mobile ad hoc networks (MANETs) since it has to be done to ensure correct routing. Protocols for IP address autoconfiguration can be classified into two categories. In the* **stateless** *category, nodes in the network do not keep IP allocation information, thus duplicate address detection (DAD) is used to resolve possible address conflicts. Protocols in the* **stateful** *category require nodes, cooperatively or separately, to maintain state information and ensure address uniqueness. They use either a full replication or no replication scheme to maintain IP information. In this paper, we propose a quorum based IP address autoconfiguration protocol in MANETs using partial replication. Making the compromise between message overhead and data consistency,* **quorum voting** *enforces data consistency by ensuring a fresh read upon every access so that each node is configured with a unique IP address. The protocol is scalable and no central server is involved. Extensive experiments are implemented comparing the configuration latency, message overhead and address reclamation cost between our protocol and existing stateful protocols. The simulation results show that nodes are configured in lower latency and the message overhead for maintaining the network is fairly low. Moreover, the proposed protocol greatly enhances the address availability by keeping proper redundancy.*

Keywords: ***IP Autoconfiguration, mobile ad-hoc networks (MANETs), quorum voting, scalability.***

I. Introduction

A mobile ad-hoc network (MANET) consists of a set of mobile nodes communicating via multi-hop wireless links and operating in the absence of a supporting infrastructure. MANETs can be stand-alone networks or connected to an external network, such as the Internet. A node in the network can either communicate directly with its neighbors within transmission range, or with distant nodes out of transmission range via multi-hop routing. Most routing protocols assume that mobile nodes in MANETs are configured with a unique identifier (ID) before routing can be initiated. An IP address, among a number of protocol parameters that are needed to be configured, can uniquely identify a node within a network. It ensures correct package delivery.

Autoconfiguration protocols are intended to configure each node in the MANET with a unique IP address and handle dynamic network situations. Existing autoconfiguration protocols can be classified into two categories: *stateless* and *stateful*. For stateless protocols [9][11][14], nodes in the network do not keep IP allocation information, thus duplicate address detection (DAD) is used to resolve possible address conflicts. Stateful protocols [1][2][3] require nodes, cooperatively or separately, to maintain state information and ensure address uniqueness.

Research on IP autoconfiguration typcally focuses on stateful protocols, which differ in the mechanism used to maintain IP state information. MANETConf [1] deploys full replication; each node keeps IP allocation information of the entire network. The confirmations from all the nodes in the MANET are required to perform IP address configuration. It provides high address availability and network reliability, by scarifying memory usage and configuration latency.

Using the opposite scheme, protocols [2] and [3] require no data replication and only collect global information in a periodic manner. While each node updates global state information from periodic synchronization in [2], only the first node that enters the network (C-root) maintains the global information in [3]. Keeping disjoint IP address blocks, protocol [2] manages to reduce the configuration latency by allowing one node to perform configuration for new nodes. The distributed IP assignment scheme [3] achieves even lower memory usage since only a portion of nodes are maintaining the disjoint IP address blocks. However, it does not provide solutions for address borrowing, and the global information is maintained by a single node.

Both full replication and non-replication have advantages and disadvantages. Partial replication is intended to achieve better performance by balancing the replication level. It makes compromise between full replication and non-replication, and brings the protocol the following benefits:

1) *Enhanced reliability:* Nodes can access the IP state information stored in one node even if it failed or the network partitioned.
2) *Improved responsiveness:* New nodes can acquire IP addresses even if most of them enter the network at the same spot, since replication extends the IP address block of an allocator.

Replicated IP state information is maintained locally at different nodes, therefore it is necessary to use a *consistency control* protocol which manages the replicas in the presence of configuration and node departure. Usually the consistency is maintained by making sure that each read is fresh and that updates are performed by at most one node. *Quorum voting* ensures the consistency of IP state information by using mutual exclusion. Votes are collected from nodes in a quorum on every read and updates are committed in the quorum. The formal

definition of quorum is provided in section II. Properties of our protocol by using quorum voting are:

1) *Address uniqueness:* Two nodes cannot be configured with the same IP address because only one allocator is able to collect enough votes from the same set of cluster heads.
2) *Data consistency:* Once network partition occurs, an IP address can be used in only one of the partitioned networks since majority votes are to be collected on configuration.
3) *High address availability:* The IP space of an abruptly vacated node is usable as long as enough copies are maintained in the network.

One potential problem of quorum voting is that the number of nodes required in a quorum for performing an operation increases linearly with the number of replicas. One way to reduce a quorum is by organizing the nodes into clusters and storing the physical copies at the cluster heads. In our protocol, only *cluster heads* maintain the IP address block and configure entering nodes. They are at least two hops away from other cluster heads. Other than cluster heads, a *common node* joins a cluster and acquires one IP address from the cluster head after it enters the MANET. We refer to a node requesting configuration as the "requestor" and the node allocating the address for the requestor as the "allocator". The cluster heads of adjacent clusters are referred to as adjacent cluster heads.

The configuration process requires the allocator to collect a quorum from its adjacent cluster heads for a proposed IP address upon the configuration request. The information with the latest time stamp is chosen to determine the availability of the address. After configuring the requestor, the allocator shall update the IP address status.

The remainder of this paper is organized as follows. Section II presents background information. Related work is provided in Section III, followed by a detailed description of the proposed protocol and its extension in Sections IV and V, respectively. The simulation results are presented and discussed in Section VI. Finally we concluded our work in Section VII.

II. Background Information

A MANET is a self-configuring network of mobile nodes (and associated hosts) connected by wireless links, the union of which forms an arbitrary topology. The nodes are free to move randomly and organize themselves arbitrarily; thus the network topology may change rapidly and unpredictably.

A. Properties and Challenges

IP address autoconfiguration requires nodes in the MANET to be configured with unique addresses, deal with migration, operate cooperatively to reclaim unused addresses, and detect address conflicts after network partition and merge. There is no centralized server. Instead, all nodes collectively perform the functionality of a server. The main challenges of IP autoconfiguration are:

IP uniqueness: Depending on how the IP addresses are distributed and maintained among nodes, allocating a unique IP address to a newly entered node could be accomplished in a global or local manner. If each node keeps the whole IP address space (*fully replicated*), the allocator is required to flood the entire network for each configuration. The scalability could be improved if each node kept disjoint IP address space (*unreplicated*), which would enable the allocator to independently configure new nodes. However, maintaining disjoint IP address space incurs the reliability issue that the departure of a single allocator would lead to IP address leaking. The proposed protocol deploys a *partial replicated* approach. It configures nodes locally based on quorum collection and provides increased network reliability.

Address reclamation: Nodes might leave the MANET after accomplishing their tasks or, due to the energy limitation, they might abruptly leave, which causes address leaking. The address reclamation process is designed to reclaim unused addresses and resolve address leaking. In former protocols [1][2][3], address reclamation is initiated reactively (event-driven) such as certain nodes fail to reply to the configuration request. Usually address reclamation involves *broadcasting* address collection messages globally, thus incurring high message overhead. A node failing to respond is considered to have left abruptly, and its address shall be collected for future use.

Network partition and merge: Network partition occurs when one or more nodes moves out of the transmission range of the rest of the nodes in the network. A unique ID is assigned to each partition so that the network partition can be detected when a package with a different partition ID is received. Reusing the addresses of departed nodes increases the available address space. However, it leads to possible address conflicts once partitioned networks merge. In the quorum-based protocol, only the majority partition is able to collect a quorum on configuration, through which it ensures consistency of IP state information.

B. Clustering

A cluster structure is a two-layer hierarchical network that partitions the network into a group of clusters. Unlike traditional clustering algorithms, clusters are dynamically formed as the nodes enter the network in our protocol. If there exists a cluster head that is within two hops, the entered node becomes the common node. Otherwise it becomes a new cluster head. Each cluster has one cluster head that configures all other members in the cluster. Two cluster heads cannot be neighbors. Figure 1 shows the network hierarchy after clustering.

C. Quorum Voting

Quorum voting is deployed in order to maintain the consistency of replicated IP information kept by different nodes in the presence of configuration, node movement, address reclamation and network partition. Quorum voting is simple to implement and its correctness is easy to prove.

The quorum voting algorithm is implemented by storing a physical copy of an allocator's IP space at its adjacent cluster head. Each copy of an IP address is associated with a *time stamp* which is equal to zero initially and is incrementally increased each time the copy is updated. An allocator can

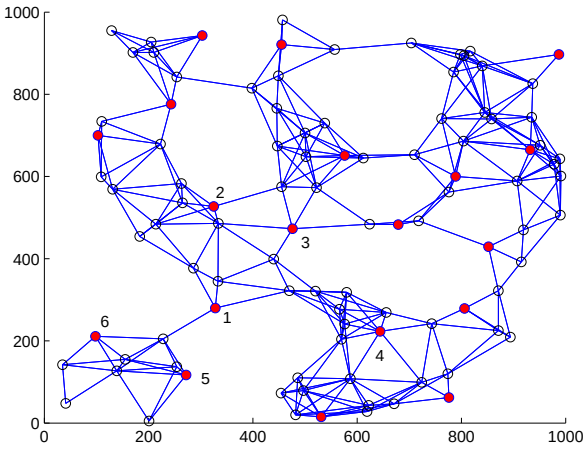

Fig. 1. **Clustering and quorum set**

process configuration requests provided it can collect votes from at least $\lfloor n/2 \rfloor$ other nodes, where n is the number of nodes. The set of these nodes is referred to as a quorum.

Definition 1: Let U be a universe of n elements. A set system $S = \{S_1, S_2, \ldots, S_m\}$ is said to be a quorum system over the universe U if $\forall i$, $S_i \subseteq U$ and $\forall i, j$ $S_i \cap S_j \neq \emptyset$. Each set S_i is referred to as a quorum set or simply as a quorum.

Applications of quorum systems concentrate on distributed control and management problems such as mutual exclusion, data replication, and secure access control. Implementing mutual exclusion using quorum systems requires nodes collecting permissions from all members in a quorum set while configuring new nodes.

In the typical application of data replication, quorum sets are divided into reading quorums r, and writing quorums w, where each reading quorum intersects each writing quorum. By satisfying the following conditions, we ensure that no two nodes can be configured with the same IP address.

$$w > v/2 \text{ and } r + w > v$$

where v is the total number of votes.

Referring to Figure 1, red nodes are cluster heads and white nodes are common nodes. Node 1 backs-up the copies of its IP address block in node 2, 3, 4, 5, 6, and at the same time stores the replicas from these nodes. The quorum system contains possible quorum sets: $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5\}$, $\{2, 3, 4, 5\}$.

D. Dynamic Linear Voting

A set containing exactly half the nodes in the quorum system does not constitute a quorum (majority set) under the voting algorithm. Otherwise, two quorums could coexist without intersection, possibly destroying the consistency of the replicated data. By choosing a *distinguished node*, the set constitutes a quorum if it contains the distinguished node. It applies only when the number of copies is even.

Definition 2: When reading or updating the allocation information of an IP address, a distinguished node [19] is the cluster head that has the address in its IPSpace.

By applying dynamic linear voting algorithm to Figure 1 when cluster head 1 is the distinguished node, we could dynamically adjust the quorum sets to: $\{1, 2, 3\}$, $\{1, 4, 6\}$, which increases the probability of successful vote collection.

III. Related Work

In this section we review the IP autoconfiguration protocols in literature. Stateless and stateful protocols are introduced. Query-based DAD (duplicate address detection) approach [9] for address autoconfiguration in ad hoc networks proposed by Perkins *et al.* deploys a stateless scheme. A newly entered node selects a random IP address that has not been used by another node (from the view of allocator) and verifies its availability with all the other nodes in the network by *broadcasting* the *Address Request* (AREQ) message. The node with the same IP address is supposed to reply the message. The new node is configured with the address if no *Address Reply* (AREP) is received after *AREQ_RETRIES* time broadcasts. This approach has low complexity and ensures even distribution of IP addresses. However, the latency and message overhead of the configuring can be very high. Besides, network merging is not considered in this approach.

Sanket N. *et al.* propose a distributed dynamic protocol, MANETconf [1], for autoconfiguration in the MANET. In this protocol, each node maintains the whole IP address pool, and global *flooding* is required on configuration of each new node. Address leaking is detected when nodes fail to reply to the verification request in the process of configuring new nodes. This protocol addressed problems involving network partition and merging. However, global flooding may lead to scalability issues as the network expands.

Mohsin and Prakash proposed a proactive IP address assignment approach [2], in which nodes maintain disjoint IP address space. Each node is capable of configuring a new node independently. Although the communication overhead for configuring new nodes is reduced, each node has to maintain the IP allocation table of the whole network and synchronize the information periodically. A node keeps track of its buddy node. In doing so address leaking can be detected when it loses track of its buddy node.

Sheu *et al.* propose a distributed IP address assignment scheme [3] for Ad hoc networks. In this scheme, only coordinators maintain IP address pools and are responsible for configuring new nodes with IP addresses. Coordinators in the network form a virtual C-tree [3] in order to periodically update their address allocation information to C-root, which is the first node that entered the network. It configures nodes in lower latency and incurs less control overhead than [2] when the number of nodes is above 140. However, it does not support network partition and merge. Detection of departed nodes and maintenance of the IP allocation table of the entire network relies on the C-root, thus C-root becomes the mainstay, but also the bottleneck of the protocol.

Boleng [10] proposes a variable length address assignment approach in MANET. A newly entered node is configured with an address higher than the current highest address in the network. Each node keeps track of the number of bits

currently used for addressing and the highest address. These two addressing parameters are embedded in every packet and updated proactively. It supports network partition and merging. However variable length addresses reduce storage resource consumption, realizing the assignment could be a challenge. Moreover, we cannot neglect the message overhead for proactively maintaining addressing parameters.

In order to tackle the problems of unbounded message delay and undetected address conflicts in the presence of partition, Vaidya proposes Weak DAD [11] for IP address autoconfiguration in MANETs based on link state routing. It allows two nodes to be configured with the same IP address as long as messages are routed to the correct destination. In addition to IP address, the scheme assumes every node is configured with a unique key based on MAC address or hardware ID. However, an address conflict cannot be detected if two nodes with the same IP address choose the same key.

In passive DAD [14], each node analyzes incoming routing packets to detect possible conflicts. Continuous monitoring on routing traffic is required in order to achieve bandwidth efficient DAD. In contrast to Active Duplicate Address Detection (ADAD), which is applied by most of the distributed IP configuration protocols, PDAD generates no additional protocol overhead on detecting duplicates. Address conflicts are detected by deriving hints from events which rarely occur in cases of unique addresses but often occur with duplicate addresses.

IV. Solution Description

Making a tradeoff between message overhead and memory storage, the IP state information is replicated locally at adjacent cluster heads, and configuring new nodes is accomplished through quorum voting. The protocol configures each node with a unique IP address and supports node movement, departure, network partition, and address reclamation. Address borrowing and quorum adjustment are provided as well, to supply higher performance.

A. Data Structure

The IP space maintained by a cluster head, *U* consists of the following data structure:

- *IPSpace*: IP address block that is assigned to a cluster head during configuring. Only *IPSpace* of the allocator is divided and assigned during configuration.
- *QuorumSpace*: IP address block maintained by adjacent cluster heads. Addresses in this set can also be used by *U* when all addresses in *IPSpace* are occupied.

Each cluster head *U* maintains the routes to the cluster heads in its *QDSet*, which contains adjacent cluster heads of *U* within three hops. *QDSet* is initialized during configuration and updated whenever new votes are distributed.

B. Network Initialization and Address Configuration

Our protocol assumes reliable delivery of messages within transmission range. The first node entering the network broadcasts a configuration request for a free IP address. If it does not receive response after a period of $\mathbf{T}_e$, it rebroadcasts the request until the threshold $\mathbf{Max}_r$ times is reached. It becomes the first cluster head and obtains the whole IP address space of the network.

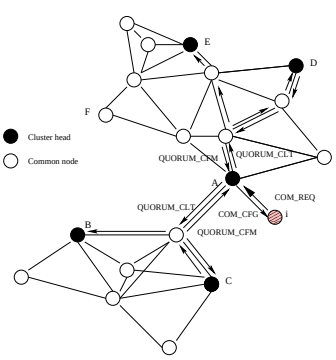

Fig. 2. Common node

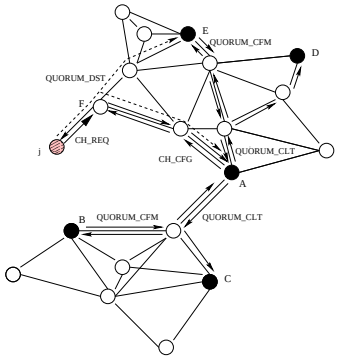

Fig. 3. Cluster head

The following nodes entering the network first listen to the periodic hello message to obtain neighbor information. The hello message contains the IP addresses of the sender and cluster heads within three hops. Say node i enters the network, if there is an allocator less than two hops away, it sends message *COM_REQ* requesting a free IP address. The cluster head A proposes an IP address and verifies whether the address is occupied with adjacent cluster heads in a quorum set. The latest time stamp is used to decide the availability of the address. Allocator A configures node i with the proposed IP address if the address is free. Otherwise it starts a new round of quorum collection on another IP address. After it is configured to be a common node, A updates the IP state information for cluster heads in the quorum. Figure 2 shows the message exchange of the common node configuration process.

If there is no cluster head within two hops, node j sends *CH_REQ* message to its nearest cluster head for configuration. The allocator assigns half its IP block after quorum collection. The newly entered node is configured with the first address of the IP block and declares itself a new cluster head. Figure 3 demonstrates the configuration process for cluster head j.

We also apply an alternative to enable even distribution of IP addresses. Instead of unicasting the allocator, the entering node first requests the information of the IP block size of cluster heads in the neighborhood. It chooses the allocator with the largest available IP block to be its configurer. The message exchanges are demonstrated in Table 1.

Requestor		Allocator		Adjacent CHs
CH_REQ	$\longrightarrow$	$\diamond$		
$\diamond$	$\longleftarrow$	CH_PRP		
CH_CNF	$\longrightarrow$	$\diamond$		
		QUORUM_CLT	$\longrightarrow$	$\diamond$
		$\diamond$	$\longleftarrow$	QUORUM_CFM
$\diamond$	$\longleftarrow$	CH_CFG		
CH_ACK	$\longrightarrow$	$\diamond$		

Table 1: **Cluster head configuration**

C. Node Movement and Departure

Nodes in MANETs have limited energy capacity and move randomly. Movement of nodes will not lead to the changing of their IP addresses, however, it would incur message overhead for location updates. After fulfilling their task, nodes may leave the network. A node moving out of the transmission range of all the other nodes in the network is considered as having left the current network.

Each node keeps the IP address of the cluster head by which it has been configured, which we refer to as a *configurer*. A node could gracefully leave the MANET by returning its IP address or address space to its configurer before its departure, or abruptly depart without notice. We first consider graceful departure, the simpler scenario. Location updates for common nodes and cluster heads are explained below.

1) Common Node: Periodic update is applied for common node location update. The node *u* moving out of three hops of its configurer informs the current nearest cluster head *C* of its existence with message *UPDATE_LOC (configurer , IP)*. The node *C* is recorded as *u*'s *administrator* and its IP address will be included in the next *UPDATE_LOC* message when it is more than three hops away from its administrator. When leaving the network, it returns its IP address to the nearest cluster head *D* and leaves the network once the acknowledgement from *D* is received. By using the information of message *UPDATE_LOC*, the IP address will be returned to its allocator, *A*, or a cluster head *E*, which belongs to the *QDSet* of the configurer. *A* or *C* updates the quorum set in the *QDSet*, and sets the address to vacant.

Periodic updates incur extra message overhead when nodes are highly mobile. If message overhead is the main issue of the network, we deploy the *upon-leave update* scheme by which nodes only send message *RETURN_ADDR* with information *(configurer, IP)* to the nearest cluster head upon departure. The address allocation update is the same as *periodic update* except that the message is broadcasted to adjacent cluster heads.

2) Cluster Head: After being configured, each cluster head keeps its IP space replication among its adjacent cluster heads. Each of them has their own corresponding quorum and keeps track of the existence of adjacent cluster heads from hello messages. As the network grows, quorum sets are updated whenever a new cluster head enters the neighborhood.

The movement of a cluster head will not change its quorum. Suppose cluster head *U* is about to leave the MANET. If its configurer *A* is within three hops, it returns its IP block to *A*. Otherwise, the address block is returned to the cluster head with the smallest IP block in its *QDSet*, *S*. Either *A* or *S* shall acknowledge *U* upon receiving its IP block. Before its departure, *U* also informs the cluster heads in *QDSet*, thus resigning itself in their *QDSet*. Cluster head *A* or *S* will inform each node configured by *U* the change of their allocator accordingly.

D. Address Reclamation

Nodes that leave the network without returning their addresses cause address leaking. The IP space of one cluster head could be reclaimed by inquiring its adjacent cluster heads and the latest IP allocation table is used. However, it involves global flooding when a large amount of nodes abruptly leave the network.

The cluster head, *U* detecting the abrupt leave of its adjacent cluster head, or running out of IP addresses in both *IPSpace* and *QuorumSpace*, initiates the address reclamation process. It broadcasts *ADDR_REC* message. Each common node configured by *U* that receives this message is supposed to send message *REC_REP* to their closest cluster head, *V*, with the IP address of *U* informing it of their existence. If *V* has the copy of *U*'s *QDSet*, it updates the quorum set in the *QDSet* and set the address to occupied. If *V* is not an item in *U*'s *QDSet*, it will forward the message to its adjacent cluster heads until the allocation information is updated.

V. Protocol Extension

A. Address Borrowing

In the former approach [2], borrowing IP addresses requires each node to maintain a global IP allocation table for the entire network. By looking up the table, the allocator requests an IP address from the node that has the largest block of IP address.

In our scheme, each cluster head maintains the *IPSpace* of its own and the nodes in *QDSet*. It extends the IP space of a cluster head by up to 5.5 times according to the simulation results. Replication greatly increases the size of IP space maintained by one cluster head, hence increasing the address availability. A cluster head first configures new nodes with addresses in *IPSpace*. Once it runs out of addresses in *IPSpace*, it starts to use addresses in *QuorumSpace* as long as enough votes from a quorum can be collected.

Although replication reduces the probability of address depletion, it still could happen. In this case, instead of initiating an address reclamation process, the cluster head acts as an agent and forwards the message exchanged for configuration between its configurer and the requester.

B. Quorum Adjustment

The quorum set of a cluster head is initially set to the current adjacent cluster heads when configured. Once there are new cluster heads entering the neighborhood, the quorum set is expanded to cover these new cluster heads and the quorum size increases as more IP block replicas are distributed.

If adjacent cluster heads take a leave or migrate elsewhere, cluster head *U* may not be able to collect the votes from a quorum in its *QDSet*, which can delay the configuration of new nodes. When *U* fails to contact certain cluster heads in *QDSet*, it starts timer $\mathbf{T}_d$. Once the timer expires, it dynamically shrinks its quorum set excluding the cluster head, *V* from which no response is received. This adjustment increases the probability of successful configuration in the event that cluster heads decrease dramatically. Accordingly, *U* unicasts *V* with message *REP_REQ* in order to verify its existence. If *U* does not receive a reply from *V* after a time $\mathbf{T}_r$ passes, it initiates an address reclamation process for cluster head *V*.

Keeping the replication of IP space at adjacent cluster heads increases the network's immunity to the multiple failure of cluster heads. Since shrinking the quorum set decreases the

number of replicas, it also increases the probability that all the cluster heads in the *QDSet* would shut down. Therefore, cluster heads begin to increase replicas once $|QDSet|$ is lower than 3.

C. Network Partition and Merging

Network merging occurs when nodes from one network move into the transmission range of nodes in another network. Network merging is handled by joining the nodes from one network to the other network one by one.

On the contrary, network partition occurs when one or more nodes move out of the transmission range of the rest of the nodes in the network and form two or more separate networks. In order to identify the partitioned networks, each of them regards the lowest IP address in the network as its network ID. It is generated as the network initiates, and is passed over whenever a new node enters the network. Network partition is detected when one node receives a message with a different network ID. All the nodes in the network with the larger network ID are required to acquire new IP addresses from the other network.

Because of the voting algorithm, two partitions cannot collect majority votes at the same time for configuring nodes with an IP address. The majority partition continues to configure new nodes with the IP addresses available. For the minority partition, the protocol deals with the following two situations:

- *Isolated cluster head:* The cluster head partitioned from all the other cluster heads in the network. It cannot configure nodes with IP address in either *IPSpace* or *QuorumSpace*. After partition, it becomes the first cluster head in the network and regains all the addresses for a network to configure existing common nodes with new IP addresses.
- *Partitioned cluster head:* The cluster head is partitioned with some of its adjacent cluster heads. It configures new nodes with IP addresses in *QuorumSpace* if majority votes for *IPSpace* do not exist.

VI. Simulation

We implement the simulation experiments using C programming language. The evaluations are focused on the configuration latency, maintenance message overhead, and address reclamation cost between our protocol and existing stateful protocols [1][2][3]. The properties of the quorum-based scheme are demonstrated as well.

A. Simulation Setup

Simulations are performed on a MANET with nodes moving to a random destination at the speed of 20m/s after its configuration with the network. Networks with a maximum of 50 - 200 nodes are simulated and the simulation area is 1km1km. Nodes arrive on a sequential manner and are randomly chosen to depart gracefully or abruptly. The probability of abrupt departure varies between 5% - 50%. The total rounds of 1000 are executed on collecting data. Figure 4 shows an example of a randomly generated network layout with 100 nodes in an area of 1km1km.

B. Configuration Latency

The comparison of configuration time is demonstrated between our protocol and MANETConf [1] in Figure 5. The hop counts include messages exchanged during the configuration. One message sent from one node to its one hop neighbor is considered to be one hop. The network size varies between 50 and 200, and the transmission range (tr) is 150m. The configuration latency is reduced by half by deploying our protocol.

The configuration time for different transmission ranges is compared in Figure 6. The configuration latency of our protocol remains below 10 hops for different transmission ranges, while it stays above 15 hops for MANETConf. Figure 7 shows the impact of transmission range and network size on configuration time for our protocol. The proposed protocol configures nodes with less latency as the transmission range decreases. However, a larger transmission range increases the average size of *QDSet* for cluster heads as we will show in following sections. This grants the network with higher reliability and increased address availability.

C. Message Overhead

The message overheads for configuration and node departure are compared between our protocol and the protocol proposed by Mohsin and Prakash [2] in Figures 8 and 9. Message overhead is based on hop counts. The parameters for the simulation are: tr = 150, area = 1km1km, nn = 50 - 200. Our protocol requires less message overhead for node configuration and departure as the network size increases since we do not require periodical synchronization of global IP allocation tables.

Figure 10 compares the maintenance overhead for node movement and departure between our protocol and the distributed protocol [3] at the speed of 20m/s. The alternative scheme that does not use location update is demonstrated as well, which greatly reduces message overhead. The comparison shows that our protocol and [3] achieve similar performance for maintenance. However, while our protocol requires that each IP address be returned to its original allocator, it is not realized for protocol [3]. Therefore after a long period of time, our protocol would not suffer from address fragmentation. Figure 11 shows the message overhead for different node speeds in the MANET with 150 nodes. Because the location update is committed when a node moves out of three hops away from its configurer or administrator, higher node mobility incurs higher message overhead.

D. Partial Replication

In order to increase network reliability on abrupt leave of cluster heads, replicated IP state information is stored for each cluster head. Communication overhead and configuration latency are increased because of quorum collection and update. However, we achieved superior performance in two aspects:

1) *Extended IP space*: partial replication simplifies local address borrowing by not requiring each node to keep global address allocation information. It enables the

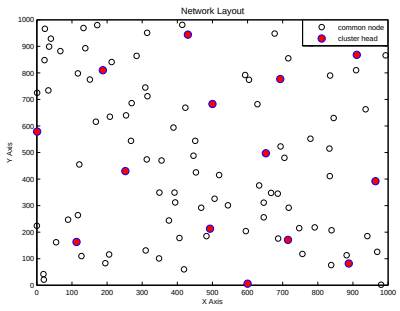

Fig. 4. Network Layout

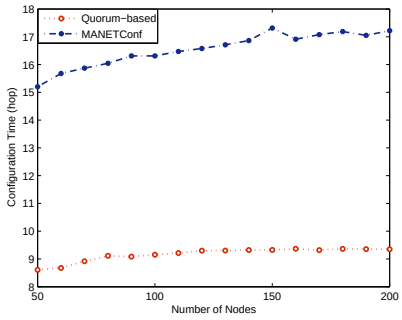

Fig. 5. Configuration latency comparison (nn)

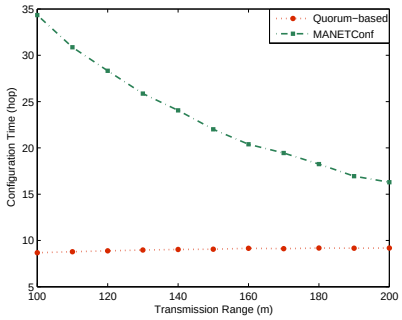

Fig. 6. Configuration latency comparison (tr)

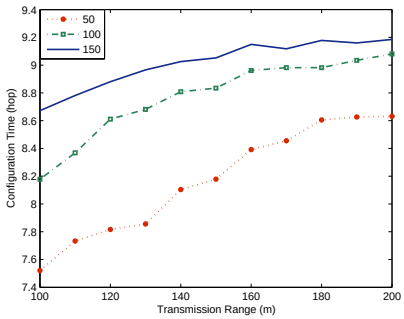

Fig. 7. configuration latency (tr)

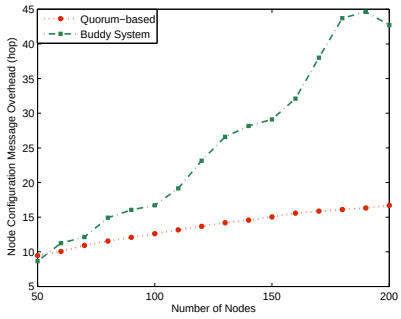

Fig. 8. Message Overhead (Configuration)

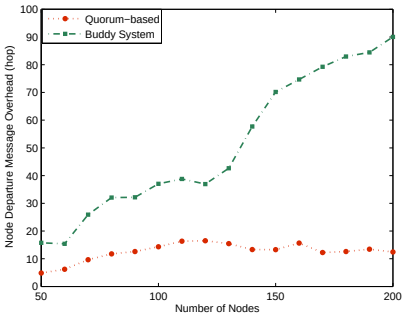

Fig. 9. Message Overhead (Departure)

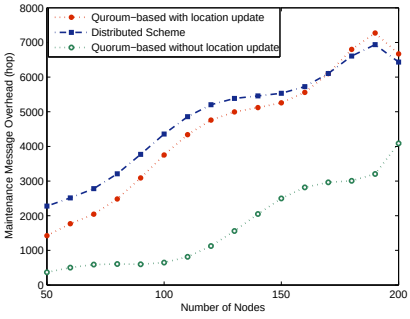

Fig. 10. Maintenance Message Overhead

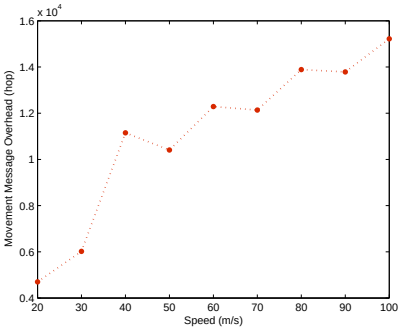

Fig. 11. Movement Message Overhead (speed)

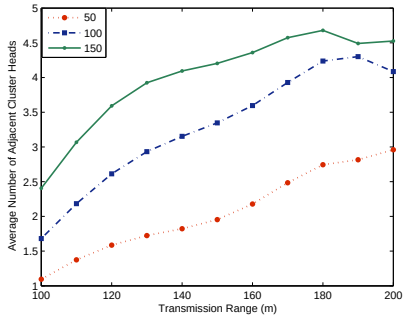

Fig. 12. Quorum size (nn & tr)

protocol to respond in lower delay when a large amount of nodes enter the network at the same spot. Simulation results show that our protocol could extend the IP space of a cluster head by up to 5.5 times its original size. Figure 12 shows the comparison of IP space size for different transmission ranges and network sizes between our protocol and [3]. As the transmission range increases, the IP space size ratio of our protocol to [3] increases.

2) *Increased network reliability:* replication increases the probability of preserving state information when a large number of nodes leave the network abruptly and simultaneously. Figure 13 compares the percentage of information loss for different abrupt leave ratios between our protocol and [3]. Suppose cluster head U left without notice. As long as half of the cluster heads in its *QDSet* exist, the IP addresses maintained by U are still usable by these cluster heads since at least one quorum remains. Otherwise we consider U to have failed after its abrupt leave. Its IP state information shall be regained during the address reclamation process. Simulation results show that replication enables the network to preserve up to 99% of IP state information of cluster heads when the abrupt leave percentage is less than 30%.

E. Address Reclamation

Address reclamation is realized locally for our protocol. Figure 14 compares the message overhead of address reclamation for our protocol and the distributed protocol [3]. They achieve similar performance when number of nodes(nn) equals to 80 and 170. Our protocol incurs less message overhead for network size over 170. For the distributed protocol [3], each coordinator periodically updates IP state information to C-root, who initiates address reclamation once certain coordinators fail

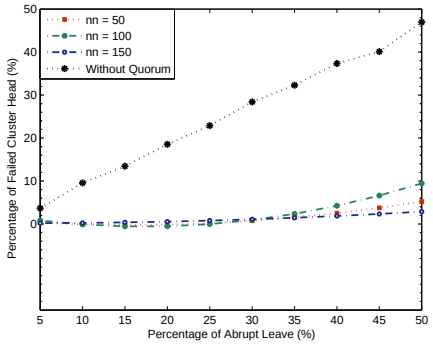

Fig. 13. Failed Cluster Heads Percentage

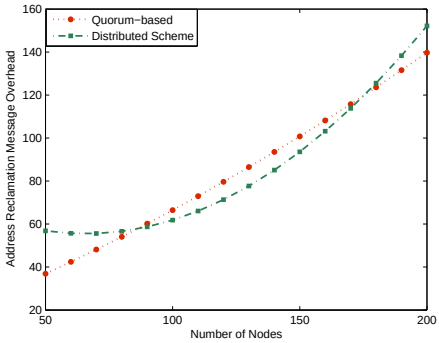

Fig. 14. Address Reclamation Overhead

to report. Since reclamation cost is relatively high, quorum-based protocol manages to postpone the process by extending the IP space size of each cluster head. From another perspective, it reduces the frequency of address reclamation.

VII. Conclusions

This paper proposes a quorum-based IP address autoconfiguration protocol in MANETs. The main contributions of the paper are as follows. First, we introduce the concepts of partial replication and quorum voting in the field of IP autoconfiguration. Partial replication increases the network reliability and the availability of free addresses at the same time. Quorum voting ensures correct configuration and consistency of IP state information in the presence of configuration and network partition. The protocol specifies the solutions of node configuration, movement and departure, network partition and merge, address reclamation and quorum adjustment. Second, a survey of the state-of-the-art IP address autoconfiguration protocols is accomplished and evaluated on their pros and cons. Third, extensive experiments are carried out to compare the performance of our protocol and three existing autoconfiguration protocols on configuration latency, message overheads, and address reclamation. Simulation results show that our protocol configures nodes with shorter latency and incurs less overall message overhead for node movement and address reclamation when compared with MANETConf [1] and [2]. Since a central server is not involved, by deploying partial replication we managed to achieve higher network reliability and address accessibility than [3].

VIII. Acknowledgement

This paper was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410, CNS 0626240, and a grant from Motorola Inc.

References

[1] S.Nesargi and R.Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," in *Proceedings of IEEE INFOCOM*, volume 2, pages 23-27, New York, USA, June 2002.

[2] M.Mohsin and R.Prakash, "IP Address Assignment in a Mobile Ad Hoc Network," in *Proceedings of Military Communications Conference (MILCOM 2002)*, volume 2, pages 856-861, Anaheim, California, USA, October 2002.

[3] J.P.Sheu, S.H.Tu and L.H.Chan, "A Distributed IP Address Assignment Scheme for Ad Hoc Networks," in *Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, volume 1, pages 439-445 Vol. 1, July 2005.

[4] M.Fazio, M.Villari and A.Puliafito, "Autoconfiguration and maintenance of the IP address in ad-hoc mobile networks," in *Australian Telecommunications, Networks and Applications Conference (ATNAC 2003)*, Melbourne, 8-10 December 2003.

[5] A.P.Tayal and L.M.Patnaik, "An address assignment for the automatic configuration of mobile ad hoc networks," in *Personal Ubiquitous Comput.*, volume 8, pages 47-54, London, UK, 2004.

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

[7] P.R.Wilson, M.S.Johnstone, M.Neely, and D.Boles, "Dynamic Storage Allocation: A Survey and Critical Review", International Workshop on Memory Management, September 1995.

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

[9] C.E.Perkins, J.T. Malinen, R.Wakikawa, E.M.Belding-Royer and Y.Sun, "IP Address Autoconfiguration for Ad Hoc Networks, draft-ietfmanet-autoconf-01.txt," *Internet Engineering Task Force, MANETWorking Group*, July 2000.

[10] J.Boleng, "Efficient network layer addressing for mobile ad hoc networks",in *Proc. of International Conference on Wireless Networks (ICWN'02)*, Las Vegas, USA, June 2002, pages 271-277.

[11] N.H.Vaidya, "Weak duplicate address detection in mobile ad hoc networks", tech. rep., University of Illinois at Urbana-Champaign, January 2002.

[12] S.Thomson and T.Narten, "IPv6 Stateless Address Autoconfiguration",

[13] Y.Sun, E.M.Belding-Royer, "A Study of Dynamic Addressing Techniques in Mobile Ad Hoc Networks", in *Wireless Communications and Mobile Computing*, 2004, pages 315-329.

[14] K.Weniger, "Passive Duplicate Address Detection in Mobile Ad Hoc Networks", in *Proc. of IEEE WCNC 2003*, New Orleans, USA, March 2003.

[15] K.Weniger, "PACMAN: Passive Autoconfiguration for Mobile Ad Hoc Networks", in *IEEE Journal on Selected Areas in Communications (JSAC) Special Issue*, March 2005.

[16] E.Guttman, "Autoconfiguration for IP Networking: Enabling Local Communication", in *IEEE Internet Computing*, volume 5, pages 81-86, Piscataway, NJ, USA, 2001.

[17] A.Misra, S.Das, and A.McAuley, "Autoconfiguration, Registration and Mobility Management for Pervasive Computing", in *IEEE Personal Communications, (Special Issue on Pervasive Computing)*, volume 8, pages 24-31, August 2001.

[18] R.Droms, et al. "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", July 2003.

[19] S.Jajodia and D.Mutchler "Enhancements to the Voting Algorithm", in *VLDB '87: Proceedings of the 13th International Conference on Very Large Data Bases* page 399-406, San Francisco, CA, USA, 1987.