On Efficient Network Planning and Routing in Large-Scale MANETs

Wassim El-Hajj, Ala Al-Fuqaha, *Member, IEEE*,
Mohsen Guizani, *Senior Member, IEEE*, and Hsiao-Hwa Chen

***Abstract*—In mobile ad hoc networks (MANETs), hierarchical architecture and distributed approaches are more practical than flat architecture and centralized approaches. In this paper, we propose a suite of protocols that achieve a distributed planning and routing scheme for MANETs. The proposed suite, which is composed of three protocols, offers scalability and extends network lifetime. The first protocol, i.e., the fast distributed connected dominating set (FDDS), constructs the virtual backbone by designing a fast distributed hierarchical algorithm that finds a connected dominating set (CDS) in the network graph. The constructed virtual backbone takes into account the node's limited energy, mobility, and traffic pattern. The second protocol, i.e., FDDS-M, proposes a distributed maintenance protocol that preserves the integrity of the hierarchical structure constructed by FDDS. The third protocol, i.e., FDDS-R, uses an intelligent path-selection fuzzy logic controller that can easily be incorporated in any existing link state routing protocol to select energy-efficient routes. We conducted extensive simulations that compare the operational properties (energy efficiency and network lifetime) of our schemes with others. The results show that our proposed schemes can achieve scalability and energy efficiency and outperform some of well-known approaches in the literature.**

***Index Terms*—Dominating set, energy efficiency, fuzzy logic, mobile ad hoc networks (MANETs), scalability, virtual backbone.**

I. Introduction

A mobile ad hoc network (MANET) [1], [2] is a collection of arbitrarily located wireless hosts in which an infrastructure is absent. MANETs can be represented as a graph $G = (V, E^t)$, where V is a set of vertices, and E is a set of time-varying edges used to represent the dynamic nature of MANETs. Two vertices are joined by a link if they are within each other's transmission coverage. Two different architectures exist for MANETs: 1) flat and 2) hierarchical.

Since flat networks have poor scalability [4], [5], we proposed an efficient hierarchical network design approach called the fast distributed connected dominating set (FDDS) [6]. Since FDDS works on the design of the initial topology, two extensions are proposed here: 1) FDDS-M and 2) FDDS-R. FDDS-M maintains the connectivity of the network, and FDDS-R takes care of the routing part. Both extensions strive to prolong the network lifetime. Our proposed protocol works well in applications where mobility is high and connectivity is important, such as a military battle field.

FDDS [6] is used to handle the initial hierarchical architecture in a distributed way. It creates a virtual backbone composed of cluster heads (CHs) and connects the network nodes to the CHs. FDDS uses a customized fuzzy logic controller, i.e., a network setup controller, that gives higher quality for nodes that can survive longer. All four steps

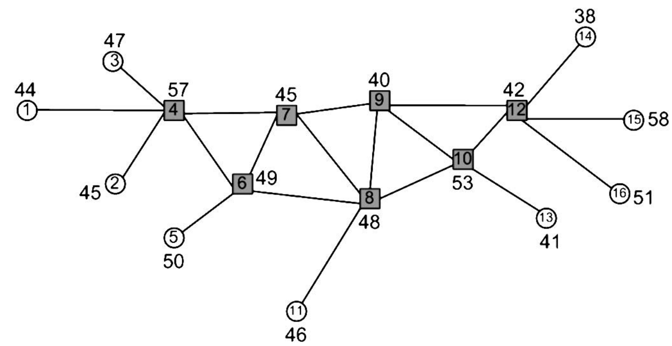

Fig. 1. MANET architecture using FDDS.

of FDDS use the output of the controller. FDDS assumes that message collisions are handled by the medium-access-control layer.

In our work, allocation of channel access occurs via a distributed contention-based mode called the distributed coordination function (DCF). DCF is the primary access method for IEEE 802.11 and is based on carrier sense multiple access with collision avoidance.

The radio model we use here follows the traditional power-attenuation model [8]. The signal power falls with $1/r^k$, where r is the distance between the transmitter and receiver nodes, and k is a real constant dependent on the wireless environment, whose value is typically between two and four. In our design, when we estimate the distance according to the Received Signal Strength (RSS), we do consider the communication environment, and this is done by varying the value of k.

Fig. 1 shows a MANET topology that was designed using FDDS. The number next to the node ID represents the quality of the mobile node SQ. FDDS constructs the virtual backbone using a low message complexity of $O(n)$ and a low time complexity of $O(\Delta^2)$. In our analysis, the time complexity is the number of steps that it takes to solve a problem as a function of the size of the input. The message complexity refers to the total number of messages exchanged between all nodes. Simulation results conducted in [6] assure the validity of the preceding discussion.

Next, we discuss virtual backbone construction using a connected dominating set (CDS). We list some of the important approximating algorithms used to find the CDS. Recall that the construction of a virtual backbone in a MANET is a primary application of CDS. Alzoubi *et al.* [12]–[14], [16] proposed a distributed CDS construction based on computing and then connecting a maximal independent set. Wu and Li [17]–[20] proposed an energy-efficient clustering based on the dominating-set marking algorithm. Parthasarathy and Gandhi [21] proposed two algorithms to find the virtual backbone based on the assumption that they require some kind of flooding, which increases the complexity of the algorithm. Stojmenovic *et al.* [22] proposed a scheme that is very similar to Wu's scheme [17]–[20], except that it requires a neighborhood topology, which may be achieved by the Global Positioning System or other geolocation techniques. In [23], Chang and Tassiulas proposed a protocol that routes data through a path whose nodes have the largest residual energy (RE).

Table I presents the complexities of the approaches discussed in the literature. It is noted that FDDS and FDDS-M provide better message complexity $O(n)$ and time complexity $O(\Delta^2)$ than the approaches presented in Table I.

A major contribution of our work lies in the experimental evaluation part. We compared our approach with other known approaches under different network settings. In some scenarios, we considered networks with outspread nodes (small node coverage). In other scenarios, we

Manuscript received January 8, 2008; revised June 21, 2008, November 13, 2008, and January 20, 2009. First published January 20, 2009; current version published August 14, 2009. The review of this paper was coordinated by Dr. E. Hossain.

W. El-Hajj is with the College of Information Technology, United Arab Emirates University, Al Ain 17551, United Arab Emirates (e-mail: welhajj@uaeu.ac.ae).

A. Al-Fuqaha and M. Guizani are with the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008-5201 USA (e-mail: alfuqaha@cs.wmich.edu; mguizani@ieee.org).

H.-H. Chen is with the Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan (e-mail: hshwchen@ieee.org).

Digital Object Identifier 10.1109/TVT.2009.2013354

TABLE I
COMPLEXITIES OF WELL-KNOWN APPROACHES SUGGESTED IN THE LITERATURE

approach	[16]	[24]	[21]	[22]	[18]
msg complexity	$O(n)$	$O(n\Delta)$	$O(n \log n)$	$O(n^2)$	$\Theta(m)$
time complexity	$O(n)$	$O(n)$	$O(\log^2 n)$	$\Omega(n)$	$O(\Delta^3)$

considered dense networks (large node coverage). We also considered networks under high traffic load and networks with and without fuzzy logic controllers. Four well-known protocols, including FDDS, were simulated under the scenarios previously mentioned; FDDS outperformed all of them and increased the network lifetime.

II. MAINTENANCE COMPONENT: FDDS-M

As time elapses, the configuration of a MANET changes, and some nodes might move away from their CH's proximity. To preserve the hierarchical structure, such nodes should switch to a new CH. In most cases, a moving node will be able to hear signals from many new CHs. The node evaluates the CHs it can hear and then decides to join one of them. The evaluation criterion is based on the following parameters: 1) the SQ of the CH; 2) the remaining capacity that the CH can handle RC; and 3) the received signal strength of the CH RSS. The moving node uses a fuzzy logic controller, i.e., a network maintenance controller, to combine these parameters and produce a single value defining the maintenance quality MQ of each CH. The CH with the highest MQ is chosen as the node's new parent.

A. Network Maintenance Controller

SQ is very important to the maintenance phase, because it indicates how powerful the CH is. However, this is not sufficient. The CH might have a high SQ, but it might not be able to handle any extra traffic. Moreover, the CH might have a high SQ, but it might be far away from the node. Thus, we consider the remaining capacity and the received signal strength of the CH as deciding factors. A high value of RC is preferred since it indicates that the CH can easily accommodate more nodes. In addition, the higher the RSS, the closer the node is to the CH, and the lower the energy it consumes to communicate with it. When SQ, RC, and RSS are combined, MQ is produced to identify the quality of each CH.

The network maintenance controller has five rules that were designed to indicate what a candidate CH should have, i.e., high SQ, high RC, and high RSS. A sample rule is "If SQ is low or RSS is weak or RC is low, MQ is bad." Next, we discuss the cluster maintenance algorithm and how it uses the network maintenance controller to preserve the network architecture and prolong its life.

B. Cluster Maintenance

Two factors that cause the network configuration to change are when CHs retire/move and when nodes move. The protocol that handles cluster maintenance is FDDS-M, which is responsible for preserving the hierarchical structure of the network. Periodically (after T seconds), each CH sends a control message to its neighbors to ensure that it has up-to-date information. The periodic message contains the CH ID, its weight SQ, and its remaining capacity RC. Each node/CH receiving the message evaluates the quality MQ of the CH that sent the message.

In our model, the node cannot be displaced from its original location by more than its own coverage area. Assume that the maximum speed is $v_{\max} = 5$ m/s and that the node coverage area is $d_{\max} = R = 200$ m. Then, the update message should be sent after every $T = (200/5) = 40$ s. In this case, we know for sure that, after 40 s, the node did not move more than 200 m. This helps us in developing the cluster maintenance actions. If one of the CHs moves a distance R in one direction and its neighboring CH moves a distance R in the opposite direction, and since the two CHs were originally R distance apart, the two CHs cannot be more than $3R$ distance (three hops) away from each other. The same argument applies for nodes that are moving.

FDDS-M is composed of two parts: One part is used to handle normal node maintenance, and the other is used to handle CH maintenance. In the succeeding sections, we explain both parts and analyze their complexities.

C. Node Maintenance

Let B_s be the total number of CHs in a MANET. Assume that CH j is the parent of normal node i. CH j sends a periodic message to its neighbors (B_s messages in total). Let t be a predefined threshold that is used by each node to determine whether a received message has a strong or a weak signal. It is noted that the algorithm is distributed; thus, measuring the time complexity of one node is sufficient to measure the time complexity of the whole algorithm. The actions that node i might take when itself or its parent CH moves are given here.

1) Node i receives a message from its parent CH j. Two possibilities might occur.
   a) Node i receives only one message from its parent CH j. In this case, node i does nothing and stays with its parent CH. In total, only B_s messages are sent by the CHs.
   b) Node i receives a message from its parent CH j and messages from other CHs. Node i measures the signal strength RSS_j of the message that originated from its parent CH j. If $RSS_j > t$, i.e., node i is within the coverage area of CH j, node i does not invoke FDDS-M and extends its association with its parent CH. (The complexity analysis is the same as that in the previous step.) If $RSS_j \leq t$, i.e., node i is barely covered by CH j, node i invokes FDDS-M. Let the CH with the highest MQ be CH k. Node i changes its association from CH j to CH k. It sends one message that informs CHs k and j about its decision. In total, B_s messages are sent by the CHs, and $(n - B_s)$ messages are sent by the normal nodes. This leads to a total of n messages exchanged in the network. Since node i evaluates a maximum of Δ CHs (all the neighbors of i), the time complexity is $O(\Delta)$.
2) Node i receives no message from its parent CH j. CH j will time out and remove node i's entry from its table. Two possibilities might exist.
   a) Node i receives messages from other CHs. Node i takes the same actions presented in the previous section when $RSS_j \leq t$; in this case, $RSS_j = 0$. n messages are exchanged in the network. The time complexity is $O(\Delta)$ (with the same analysis as that presented in Step 1b).
   b) Node i does not receive any message from any CH. Node i sends a message to its neighbors and waits for an ACK (the first message). Each ACK covers one-hop neighbors of the node that is sending the ACK (Δ messages).
      i) When one or more ACKs are received by node i, the ACKs' contents are searched for CHs. Note that the contents of the ACKs represent the two-hop neighbors of node i. Node i connects to one of the discovered CHs by electing the one-hop neighbor with the highest weight (the second message). A total of $\Delta + 2$ messages are exchanged in this scenario. The time complexity is $O(\Delta^2)$ since node i makes the decision, this time

according to its one- and two-hop neighbors. It is noted that this scenario (in most cases) can only be applicable for a constant number of nodes, because most nodes have at least one CH in their proximity. In total, B_s messages are sent by the CHs, and $c \times (\Delta + 2)$ messages are sent by the normal nodes, where c is a constant.

To highlight our claim that this scenario (*in most cases*) is applicable for a constant number of nodes, we consider a topology where all the nodes in the network are initially within each other's range. In this case, FDDS elects three CHs to represent all the nodes in the network. If a large number of nodes exit the communication range of the CHs within time period T, they either can connect to the already elected CHs using two hops or will completely be disconnected from the network. In both cases, the maintenance algorithm preserves the network.

ii) If no ACKs are received, node i becomes completely disconnected from the network. CH j will time out and remove node i from its table. When node i again connects to the network, it assigns itself to a new parent CH and resumes communication. No extra messages are sent, except for the B_s messages sent by the CHs.

Combining the complexity analysis previously made, we conclude that the node maintenance algorithm has $O(n)$ message complexity and $O(\Delta^2)$ time complexity. These are the same complexities achieved by FDDS.

D. CH Maintenance

A CH can invoke cluster maintenance, depending on two conditions: 1) It wants to retire, because its RE becomes low ("CH retires"). 2) It receives a weak or no signal from a neighboring CH ("CH moves").

1) CH Retires: Let $J_1 = \{j_1, j_2, \ldots, j_\Delta\}$ be the original CH neighbors of CH j. CH j starts by sending J_1 to all its neighbors (the first message). Two scenarios exist.

a) Let i be one of j's neighbors, such that i can cover $j \cup J_1$. If such an i exists, i changes its status to become CH and sends a message to CH j, notifying that it becomes a normal node (the second message). Finally, node j sends a message to its neighbors, telling them to stop their search since a node that can cover $j \cup J_1$ was found (the third message). In such a scenario, three messages were sent by various nodes. If all the CHs in the network were to retire at the same time (which is very much unlikely), a total of $3B_s$ messages will be exchanged in the whole network. The time complexity is $O(\Delta^2)$ since node i searches its neighbors (maximum Δ) to check if they cover all the neighbors of j (maximum Δ).

b) If none of j's one-hop neighbors can cover $j \cup J_1$, then FDDS-M searches for more than one node that can cover $j \cup J_1$. Let i be a one-hop neighbor of j such that $j \cup J_1$ belongs to the one- and two-hop neighbors of i. Let $\{k_1, k_2, \ldots, k_\Delta\}$ be the connectors that connect the one-hop nodes of i with the CHs that belong to $j \cup J_1$ (excluding those that belong to i's one-hop neighbors). i changes its status to CH and sends a message to the connector nodes, asking them to become CHs (the second message). It also asks CH j to become a normal node. Node j, in return, sends a message to its neighbors, telling them to stop their search since a node that can cover $j \cup J_1$ was found (the third message). This way, the backbone connectivity is preserved.

The procedure previously discussed sends, in the worst case, three messages. One message was sent by the CH who wants to retire. Another message was sent by the node that can cover $j \cup J_1$. The same message asks the connector nodes to become CHs. In addition, a third message to stop any node from doing extra work was sent. In total, a maximum of $3B_s$ messages will be exchanged in the network. The procedure requires $O(\Delta^2)$ time complexity, because a neighbor of j needs to check whether its one-hop neighbors can cover $j \cup J_1$. Thus, the overall message and time complexities of the "CH retires" procedure are $O(n)$ and $O(\Delta^2)$, respectively.

2) CH Moves: Let J_1 be the set containing the original CH neighbors of CH j. CH j starts by sending a periodic message to its neighbors. Each neighbor that receives the message replies with an ACK. Three scenarios might occur.

a) All the CHs that belong to J_1 send an ACK to CH j. In this case, the original connections of CH j are preserved, and no action is needed. When all the CHs in the network move, they send a maximum of B_s messages. Assume that all the nodes/CHs in the network acknowledge these messages. A maximum of n messages will be sent. This leads to a total of $B_s + n$ messages exchanged. Since CH j checks if the acknowledgments cover all the items in J_1 (maximum Δ items), the time complexity of this scenario is $O(\Delta)$.

b) Only part of the CHs that belong to J_1 send an ACK to CH j. CH j might also receive some ACKs from CHs that do not belong to J_1. If all the CHs send a message and all the CHs/nodes acknowledge these messages, a total of $B_s + n$ messages will be sent. CH j starts by establishing connections with the new CHs that send it an ACK (if any). Then, CH j sends a message destined to the CHs in J_1. Note that, according to our model (In Section II-B, the maximum node displacement is equal to the coverage area), all the CHs in J_1 are reached by j through a maximum of two to three hops. If all the CHs send such a message for their three-hop neighbors, a maximum of $3B_s$ messages will be exchanged in the network. If all the CHs in J_1 reply with an ACK, the backbone would be preserved, and no action is needed. Since these ACKs have to travel three hops back to their original CH, another $3B_s$ messages are needed. This yields a total of exchanged messages to $n + 7B_s$. The time complexity of the algorithm is $O(\Delta)$, because j checks if all the items in J_1 are covered.

If only part or none of the CHs in J_1 sends an ACK to CH j, the backbone is disconnected. New CHs must be elected to connect CH j with the CHs in J_1 that got disconnected from j. CH j executes Step 3 of FDDS as an attempt to connect to the CHs in J_1 that are not one hop away from CH j. Step 3 of FDDS requires a CH to send a maximum of two messages. All the CHs will need to send $2B_s$ messages. The nodes elected by executing Step 3 of FDDS change their status to CH and update their tables. CH j also updates it own tables, and the backbone is preserved. Each newly elected CH sends a message to its neighbors, updating its status (B_s messages). Therefore, the total number of the messages exchanged in this scenario is $n + 4B_s$. The time complexity of the algorithm is $O(\Delta^2)$, because a CH needs to loop across its one- and two-hop neighbors.

c) If neither CHs nor normal nodes send an ACK to CH j, CH j becomes disconnected from the network. One of j's neighbors times out after Δt and sends a message to its neighbors, telling them that CH j has been disconnected from the network. CH j changes its status to normal node. When CH j again connects to the network, it is treated as a normal node.

In summary, cluster maintenance (FDDS-M) is composed of node maintenance and CH maintenance, each having message and time complexities of $O(n)$ and $O(\Delta^2)$, respectively. These are the same complexities achieved by FDDS. Therefore, the low complexities are still preserved.

E. FDDS-M Is Scalable and Energy Efficient

The network maintenance controller used by FDDS-M gives more weight for CHs that have high SQ, RC, and RSS values, each of which can be translated to high energy efficiency. A high SQ means that the CH is powerful and more survivable than other CHs. A high RSS means that the CH is close to the node; thus, less power is needed for communication. A high RC means that the CH can handle extra traffic and can balance the traffic load on each CH. Balancing the traffic load among all CHs results in a more survivable network [17], [19], [20], [25]–[27]. The scalability of FDDS-M comes from the fact that it is completely distributed and has low message and time complexities.

III. Routing Component: FDDS-R

To complete our distributed scheme, we design an energy-efficient routing protocol called FDDS-R, which is, in fact, an extension of the well-known Optimized Link State Routing Protocol (OLSR). We extended OLSR by designing an intelligent path selection controller, i.e., a routing controller that chooses an appropriate path to route messages between (s, d) pairs. In this section, we first discuss the controller and then explain how the routing is done.

A. Routing Controller

Suppose that a path exists between a source–destination pair. Define path length PL to be the number of hops along the path. In addition, define path cost to be the RE of the weakest node (with the minimum RE) along the path. Since choosing the shortest path to route messages between source–destination pairs is not a good idea, particularly when energy conservation is needed, a preferable path is suggested, i.e., a path that has a fewer number of hops consisting of nodes with higher RE, i.e., lower PL and higher path cost. To achieve such balance, we design a fuzzy logic controller, i.e., a routing controller that combines the path cost and path length and produces a single output that defines route quality RQ. The path with the highest RQ is chosen to route messages between (s, d) pairs.

The routing controller has five rules that are designed to give higher values for paths that have a higher cost (nodes along the path have higher RE) and a fewer number of hops (short PL). A sample rule is "If the path cost is medium and PL is short, RQ is good."

B. Routing

The routing controller produces a single output that defines route quality RQ. The path with the highest RQ is chosen to route messages between (s, d) pairs. The flooding mechanisms [multipoint relays (MPRs)] used in OLSR is performed only once when the network structure is complete, i.e., when the clusters are formed. The flooding only happens in the virtual backbone and not in the whole network. As a result of the flooding, the routing tables are filled. From that time on, no flooding will be done. The routing tables are updated by the maintenance protocol. When a particular path is selected, the routing is composed of three steps: 1) If the source is not a CH, it sends a message to its representative CH; else, it does nothing. 2) Then, the CH acts as the new source and forwards the message to its destination CH. 3) If the destination CH is not the final destination, it relays the message to the final destination node in its cluster.

C. FDDS-R Is Scalable and Energy Efficient

Even though the distance-vector routing protocol is simple and efficient in small networks and requires little management, they do not scale well (due to the count-to-infinity problem) and have poor convergence properties. For this reason, we add the routing controller module to a link state routing protocol (i.e., OLSR). The scalability of FDDS-R comes from using OLSR. OLSR uses MPR, which minimizes the flooding of control messages in the network. In addition, OLSR is known to perform well in large-scale and dense networks. Even though the controller tries to balance between the energy and length of the path, it is more biased to higher energy paths (paths that are shorter with more powerful nodes). This fact directly contributes to the energy efficiency of FDDS-R. Our simulation results show that networks that use the routing controller are more survivable than those without it.

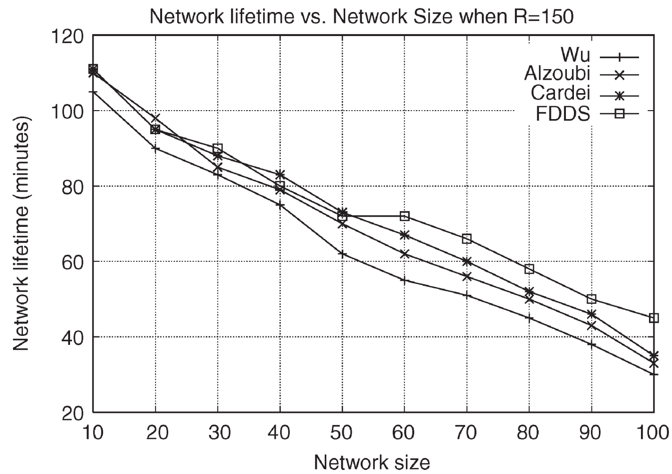

Fig. 2. Network lifetime of networks of different sizes with $R = 150$. FDDS and approaches suggested by Alzoubi *et al.* [28], Cardei *et al.* [24], and Wu [26] are compared.

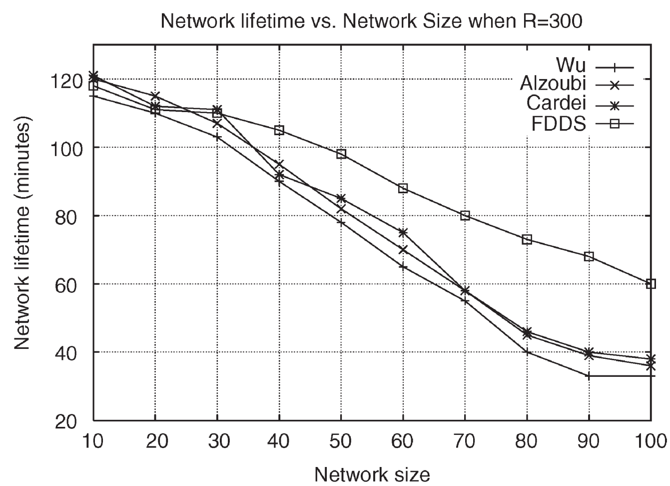

Fig. 3. Network lifetime of networks of different sizes with $R = 300$. FDDS and approaches suggested by Alzoubi *et al.* [28], Cardei *et al.* [24], and Wu [26] are compared.

IV. Simulation Result

In this section, we analyze the energy efficiency of our distributed approach. We refer to the whole approach as FDDS, which includes FDDS, FDDS-M, and FDDS-R. We focus on evaluating the operational properties (energy efficiency and lifetime) of FDDS.

In our simulations, we use the Monte Carlo method, where random numbers are utilized to simulate traffic and mobility. For mobility, we use the Gauss–Markov Mobility Model. The main advantage of this model is allowing past velocities (and directions) to influence

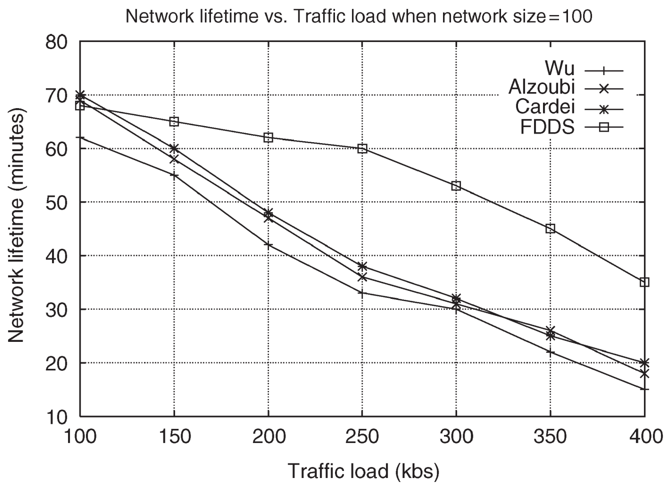

Fig. 4. Network lifetime of a network of 100 nodes under increasing traffic load. FDDS and approaches suggested by Alzoubi *et al.* [28], Cardei *et al.* [24], and Wu [26] are compared.

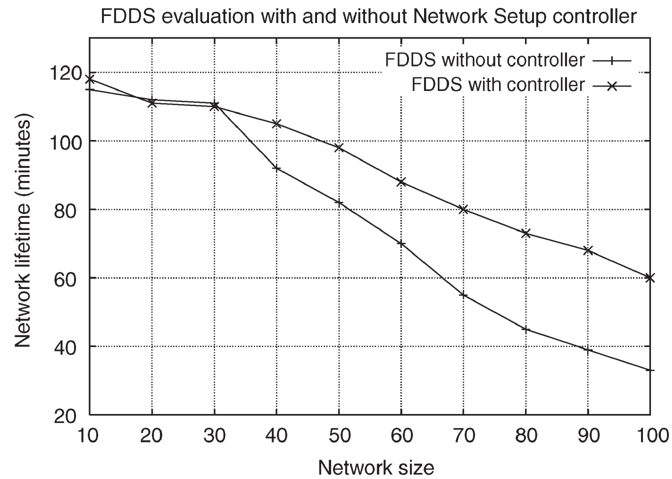

Fig. 5. Network lifetime of networks that use the network setup and network maintenance controllers versus networks without them.

future velocities (and directions). Traffic generation was simulated using Poisson distribution. Traffic modeled by Poisson distribution has exponential interarrival time and holding time. For our simulations, we developed a customized simulator using Java. We used a package called SimJava to facilitate the development of our simulator. To design the fuzzy logic controllers, we used MATLAB. Since Java cannot directly interact with the fuzzy logic toolbox in MATLAB, we used a C program that converts an *.fis* file to a *.dll* file (which can directly be accessed from Java). This way, we were able to achieve direct communication between the Java simulation engine and the fuzzy logic controllers.

A. Lifetime

Networks with sizes in the range of 10–100 nodes are uniformly generated in a 500-square-unit grid. Out of n existing nodes, $n/2$ sources and destinations are randomly generated. The power consumed by the battery is 1749 and 930 mW for transmitting and receiving, respectively. The battery of each node has energy that varies between 1000 and 10 000 J.

For this experiment, we implemented the approaches suggested by Alzoubi *et al.* [28], Cardei *et al.* [24], and Wu [26]. For a given network, we vary R and calculate the lifetime of each network. R can take values of 150 and 300. Figs. 2 and 3 show the lifetime of networks with different sizes when the aforementioned approaches are used. The bit rate used is 250 kb/s.

Fig. 2 shows the lifetime of different network sizes when the following approaches are used: FDDS and the approaches suggested by Alzoubi *et al.* [28], Cardei *et al.* [24], and Wu [26]. The coverage of each node is set to 150 units. When the network size is small (< 40), all approaches perform the same. For networks of a large size, FDDS has the longest lifetime. In fact, FDDS has a lifetime that is almost 10 min longer than that of the other approaches. Fig. 3 shows the same experiment but with the node coverage set to 300 units. Such a large coverage produces a dense network. FDDS achieves a much longer lifetime (almost double) when the network size is 100. We can conclude from this experiment that FDDS performs better when the network is large and dense.

B. Lifetime Under Increasing Traffic Load

In this experiment, we evaluate the lifetime of a network with 100 nodes under various traffic loads. The coverage of each node is set to $R = 300$. The approaches compared are the same as those used in the previous section. Fig. 4 shows the network's lifetime with increasing traffic load. The traffic loads vary between 100 and 400 kb/s. Fig. 4 shows that the other approaches perform better than FDDS when the traffic load is low. However, as the traffic load increases, FDDS outperforms the other approaches. When the traffic load is 400 kb/s, FDDS's lifetime is almost double that of the other approaches. This experiment proves that FDDS performs well when the traffic load is high.

C. Lifetime With and Without Fuzzy Logic Controllers

In this experiment, we evaluate the effect of using the network setup controller and the network maintenance controller on the performance of FDDS. Once we calculate the lifetime of a particular network, each node's weight is calculated using the network setup controller. Then, we calculate the lifetime of the same network, when each node's weight is set to be its own RE. The bit rate is set to 250 kb/s, and the coverage area of each node is $R = 300$. Fig. 5 shows that networks with small sizes are not affected by the fuzzy logic controller. However, as the network size increases, networks that use the network setup fuzzy logic controller survive longer than networks without it (with survivability of almost twice when the network size is greater than 80). This experiment shows that the fuzzy logic controller plays an important role in increasing the network lifetime.

D. Lifetime With and Without the Routing Controller

In this experiment, we evaluate the effect of using the routing controller on the performance of FDDS. Two routing schemes are considered in this experiment. The first scheme uses the routing controller to evaluate each path. It chooses the path with the highest weight. The second scheme calculates the path cost (the RE of the weakest node) of each path. It chooses the path with the highest path cost. We calculate the lifetime of various networks when both schemes are used. Let the bit rate be 250 kb/s and the node coverage area be $R = 150$. Fig. 6 shows the results. When the network size increases, our scheme extends the network lifetime. This is because the routing controller tries to balance the path length and cost, whereas the other routing schemes only consider the path cost. This experiment shows that the routing controller also plays an important role in increasing the network lifetime.

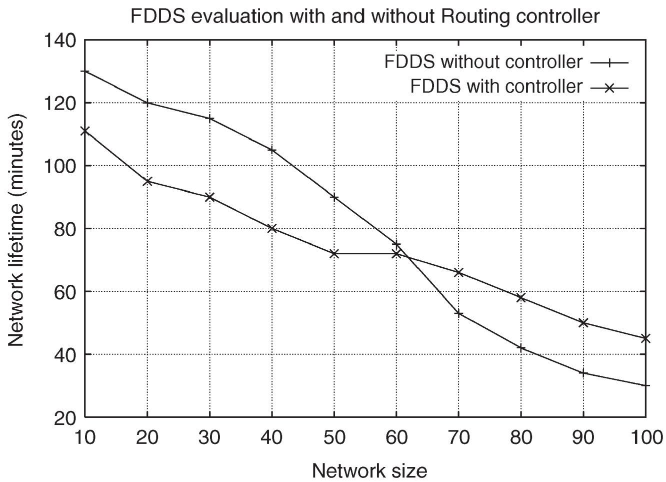

Fig. 6. Network lifetime of networks that use the routing controller versus networks without it.

V. Conclusion

To handle scalability and energy efficiency in large-scale MANETs, we have designed a hierarchical energy-efficient scheme that can easily be set up and maintained. The scheme consists of three components: 1) the clustering component (FDDS), 2) the maintenance component (FDDS-M), and 3) the routing component (FDDS-R). The contribution that has been made in this paper is the proposal of FDDS-M and FDDS-R. FDDS-M is used to handle network maintenance in a distributed manner, and FDDS-R is used to handle network routing. The customized fuzzy logic controllers that we have used in the aforementioned components and the efficient algorithms that we have used to cluster/maintain the network distinguish our scheme from the other schemes that have the same objective. The experimental evaluation that we have conducted show that, when all three components are used to handle MANET communication, the scalability and energy efficiency of the network are improved.

References

[1] P. Mohapatra and S. Krishnamurthy, *Ad Hoc Networks: Technologies and Protocols*. New York: Springer-Verlag, 2005.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluations considerations," *IETF RFC*, no. 2501, Jan. 1999.

[4] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[5] P. Gupta, R. Gray, and P. R. Kumar, "An experimental scaling law for ad hoc networks," Univ. Illinois at Urbana-Champaign, Tech. Rep., May 2001.

[6] W. El-Hajj, Z. Trabelsi, and D. Kountanis, "Fast distributed dominating set based routing in large scale MANETs," *Comput. Commun.*, vol. 30, no. 14/15, pp. 2880–2891, Oct. 2007.

[7] W. El-Hajj, D. Kountanis, A. Al-Fuqaha, and S. Guizani, "A fuzzy-based virtual backbone routing for large scale MANETs," *Int. J. Sensor Netw. (IJSNet)*, vol. 4, no. 4, pp. 250–259, Jan. 2006.

[8] T. S. Rappaport, *Wireless Communications: Principles and Practices*. Englewood Cliffs, NJ: Prentice–Hall, 1996.

[9] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1–3, pp. 165–177, Dec. 1990.

[10] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[11] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, Apr. 1998.

[12] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *J. Commun. Netw.*, vol. 4, no. 1, pp. 22–29, Mar. 2002.

[13] K. Alzoubi, P.-J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *Proc. 35th Hawaii Int. Conf. Syst. Sci.*. Big Island, HI, 2002, p. 297.

[14] P.-J. Wan, K. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, 2002, pp. 1597–1604.

[15] I. Cidon and O. Mokryn, "Propagation and leader election in multihop broadcast environment," in *Proc. 12th Int. Symp. DISC*, Andros, Greece, Sep. 1998, pp. 104–119.

[16] K. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *Proc. MOBIHOC*. Lausanne, Switzerland, 2002, pp. 157–164.

[17] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," in *Proc. ICPP*, 2001, pp. 346–356.

[18] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. 3rd Int. Workshop Discrete Algorithms Methods Mobile Comput. Commun.*, Aug. 1999, pp. 7–14.

[19] J. Wu, B. Wu, and I. Stojmenovic, "Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets," *Wireless Commun. Mobile Comput.*, vol. 3, no. 4, pp. 425–438, Jun. 2003.

[20] J. Wu, B. Wu, and I. Stojmenovic, "Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets," *Wireless Commun. Mobile Comput.—Special Issue Res. Ad Hoc Netw., Smart Sensing, Pervasive Comput.*, vol. 3, no. 4, pp. 425–438, Jun. 2003.

[21] S. Parthasarathy and R. Gandhi, "Fast distributed well connected dominating sets for ad hoc networks," Univ. Maryland, College Park, MD, Tech. Rep. CS-TR-4559, 2004.

[22] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 1, pp. 14–25, Jan. 2002.

[23] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 609–619, Aug. 2004.

[24] M. Cardei, X. Cheng, X. Cheng, and D.-Z. Du, "Connected domination in multihop ad hoc wireless networks," in *Proc. 6th Int. Conf. Comput. Sci. Informat.*, NC, 2002, pp. 251–255.

[25] J. Shaikh, J. Solano, I. Stojmenovic, and J. Wu, "New metrics for dominating set based energy efficient activity scheduling in ad hoc networks," in *Proc. WLN Workshop (in conjunction IEEE Conf. Local Comput. Netw.)*, Oct. 2003, pp. 726–735.

[26] J. Wu, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," *J. Commun. Netw.*, vol. 4, no. 1, pp. 59–70, Mar. 2002.

[27] J. Wu and B. Wu, "A transmission range reduction scheme for poweraware broadcasting in ad hoc networks using connected dominating sets," in *Proc. IEEE Semiannual VTC—Fall*, Oct. 2003, pp. 2906–2909.

[28] K. M. Alzoubi, P.-J. Wan, and O. Frieder, *Message Efficient Distributed Algorithms for Connected Dominating Set in Wireless Ad Hoc Networks*, 2001.