Understanding and Exploiting the Trade-Offs between Broadcasting and Multicasting in Mobile Ad Hoc Networks

Lap Kong Law, Srikanth V. Krishnamurthy, and Michalis Faloutsos

Abstract—We find that current group communications protocols are far from "one size fits all," they are typically geared toward and optimized for particular scenarios. Multicasting, in general, works well if the density of group members is sparse and in low mobility; broadcasting, in contrast, works well with a high density of group members and in high mobility. Due to the dynamics of the network, one strategy may be preferable to the other at different times and in different localized regions. In this paper, we first quantify the trade-offs between broadcasting and multicasting and evaluate the suitability of a strategy in various scenarios of deployment. Based on the lessons learned, we design a protocol that **adapts** in response to the dynamics of the network. We named our protocol Fireworks. Fireworks is a hybrid two-tier multicast/broadcast protocol that provides efficient and lightweight multicast dissemination and self-adapts in response to variations in the density and distribution of group members to provide efficient performance. Fireworks creates pockets of broadcast distribution in areas with many members, while it creates and maintains a multicast backbone to interconnect these dense pockets. Fireworks offers packet delivery statistics comparable to that of a pure multicast scheme but with significantly lower overheads. We also show that Fireworks has a lower level of degrading influence on the performance of coexisting unicast sessions than either traditional multicast or broadcast methods.

Index Terms—Group communications, ad hoc networks, multicast, broadcast.

1 Introduction

Group communications is an essential component in mobile ad hoc networks (MANETs). Its use is not only limited to typical ad hoc network applications such as tactical deployments, electronic classrooms, or disaster recovery missions, but it is also an indispensable component for disseminating control information in many ad hoc routing protocols. Due to the importance of group communications in MANETs, many protocols for achieving the same purpose have been widely explored [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Recent surveys on these protocols appear in [14] and [15].

Typically, group communications protocols are classified into either broadcast or multicast protocols. Traditionally, a broadcast protocol is typically thought of as a method that disseminates data to the entire network, while a multicast protocol targets only a subset of nodes that are called group members. A second distinguishing feature between broadcast and multicast protocols (a feature that is of importance in this work) is that, while multicast protocols create and maintain some sort of a data dissemination structure (such as a tree or a mesh), broadcast protocols do not; they are typically derived from the simple flooding strategy.

We find that typical multicast protocols are far from "one size fits all," i.e., they are typically geared toward and optimized for particular scenarios. Therefore, when they are deployed in different scenarios, their performance may vary significantly. Furthermore, they may incur unreasonable amounts of overheads in certain scenarios. The creation and maintenance of the multicast structure could be heavyweight as their operations require control messages to be exchanged among the constituent nodes in the network. As one might expect, in cases of high mobility, wherein the constructed multicast structure tends to stale fairly quickly, there is a need for the periodic invocation of control messages with high frequency.

Broadcasting provides several intrinsic advantages. First, it does not require the creation of any delivery structure. Second, there is an inherent redundancy in broadcasting due to multiple rebroadcasters. This redundancy provides extra robustness in conditions of mobility. Therefore, broadcasting is preferable for use in the scenarios with many group members or in high mobility. On the negative side, broadcasting would attempt to deliver the packet to all the nodes in the network regardless of who the intended recipients are. This property of broadcasting leads to many redundant data transmissions and renders it an unsuitable choice in scenarios with a small number of group members.

This discussion suggests that there are trade-offs between the use of multicasting and broadcasting for providing group communications in MANETs. Our first objective in this paper is to comprehensively understand and quantify these trade-offs. Toward this, we choose a candidate protocol from each class: the On-Demand Multicast Routing Protocol (ODMRP) [4] for multicast and the Scalable Broadcast Algorithm (SBA) [16] for broadcast. The reason for choosing these two protocols is that they have been shown to outperform most other protocols in their

• *The authors are with the Department of Computer Science and Engineering, Engineering BU2, Room 367, University of California, Riverside, CA 92521. E-mail: {lklaw, krish, michalis}@cs.ucr.edu.*

Manuscript received 18 Jan. 2006; revised 4 May 2006; accepted 18 July 2006; published online 16 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0018-0106.

respective classes [17], [18]. In general, our results suggest that, while multicasting seems to be the preferable choice in scenarios of low to moderate mobility and when the group size is small (less than 40 percent of the nodes are group members), broadcasting appears to be the winner in high mobility and if the group size is relatively large (more than 40 percent of the nodes are group members).

Armed with this understanding of the trade-offs between broadcasting and multicasting, our second goal in this paper is to design an adaptive group communications protocol for MANETs. The key idea in designing the new protocol, which we call Fireworks (for reasons to be detailed later), is that it may be desirable to simply perform broadcasts in localized regimes of the network in which there is a dense clustering of multicast group members. Constructing and maintaining a structure in these localized regimes may simply yield negligible dividends over this approach and, furthermore, at a significant overhead cost.

Simply put, Fireworks is an adaptive, multicast/broadcast protocol that exploits group members' affinity to simplify multicast routing and invoke broadcast operations in appropriate localized regimes. By reducing the number of group members that participate in the construction of the multicast structure and by providing robustness to mobility by performing broadcasts in densely clustered local regions, Fireworks achieves packet delivery statistics that are comparable to that with a pure multicast protocol but with significantly lower overheads. We compare the performance of Fireworks with the performance of a fairly exhaustive set of group communications protocols; in particular, we use the Multicast Ad-hoc On-demand Distance Vector (MAODV) [19], Hierarchical Differential Destination Multicast (HDDM) [20], ODMRP, and SBA in our comparison studies. Fireworks outperforms the protocols considered in our studies. In particular, our performance evaluations demonstrate that:

1. Fireworks strikes the balance between packet delivery performance and overhead with various group sizes and mobility.
2. Fireworks can withstand an increased amount of traffic load better by providing a higher packet delivery ratio with lower overhead.
3. Fireworks incurs lower overhead in scenarios with clustered motion, while maintaining a very good packet delivery performance.

In addition, Fireworks also produces lower levels of interference on coexisting unicast sessions when compared to both ODMRP and SBA.

The rest of the paper is organized as follows: In Section 2, we provide brief overviews of our chosen protocols, ODMRP and SBA. Then, in Section 3, we present an evaluation of the trade-offs between broadcasting and multicasting under various considered scenarios by running each of the chosen protocols in isolation. In Section 4, we provide a detailed description of our proposed protocol, Fireworks. In Section 5, we present our performance evaluations of Fireworks and discuss the observations. We discuss related work in Section 6. Finally, we conclude the paper in Section 7.

2 Description of the Chosen Multicast and Broadcast Protocols

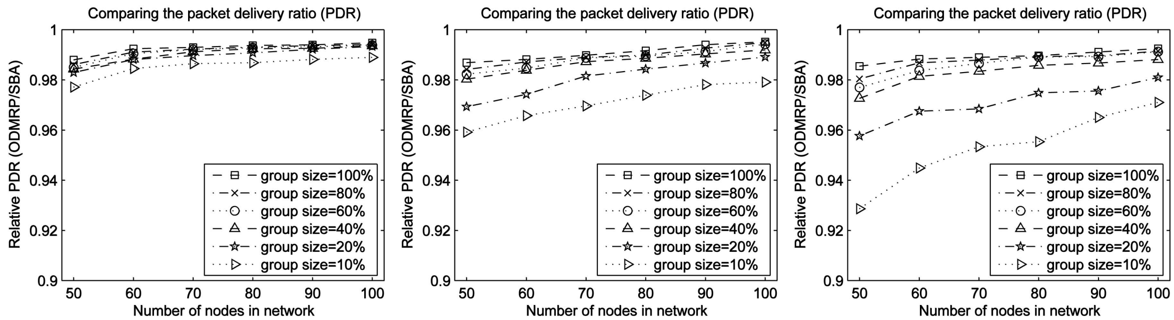
In this section, we provide an overview of the candidate multicast and broadcast protocols that we have chosen for our evaluation. As mentioned earlier, our choice was based on prior efforts [17], [18] that demonstrate that these protocols outperform most of their competitors protocols in terms of performance.

2.1 On-Demand Multicast Routing Protocol (ODMRP)

The On-Demand Multicast Routing Protocol (ODMRP) [4] is a mesh-based multicast protocol. When a multicast source has a packet to send and the multicast group members are yet to be identified, it floods a Join Query message in the network. The Join Query message is also periodically flooded to refresh group membership information and update routes as long as the source still has packets to send. When a node receives a Join Query message, it stores the *source id* and *sequence number* indicated in the message in its message cache; duplicate receptions of the same Join Query are discarded. If the message received is not a duplicate instance of a previous message and if the Time-to-live (TTL) value indicated in the message is greater than zero, the recipient node rebroadcasts the Join Query. When the Join Query reaches a multicast receiver, it creates a Join Reply message and broadcasts it to its neighbors. When a node receives a Join Reply, it checks if it is identified to be the next hop entry. If it is, the node is a forwarding node and the forwarding group flag is set. It then rebroadcasts its own Join Reply. Finally, the Join Reply reaches the multicast source and the routes are established. From then on, until information is further updated, a node will forward the packet only if it is in the forwarding group. In contrast with traditional tree-based protocols, this mesh-based protocol can potentially construct multiple routes from the source to each of the group members. Thus, ODMRP can tolerate mobility much better than most of its counterparts (as identified in [17]).

2.2 Scalable Broadcast Algorithm (SBA)

The Scalable Broadcast Algorithm (SBA) [16] is an intelligent broadcast protocol in the sense that it considerably reduces the number of rebroadcasts as compared with flooding. Furthermore, it has been shown in previous work [18] that SBA outperforms most of the other broadcast schemes such as the counter-based scheme and the location-based scheme. It reduces the effects of a broadcast storm [21] by using a simple technique that we briefly discuss below. SBA incorporates the exchange of periodic Hello messages between neighbors to enable the acquisition of local neighborhood information by each node. Each Hello message contains a list of the one-hop neighbors of the broadcasting node and, thus, finally, every node in the network will have its two-hop neighborhood information. The collected neighborhood information is used to decide whether or not a received data packet should be rebroadcasted. The decision is made by determining, by means of the neighborhood information table, if there exists any node that is not covered by previous broadcasts. If all the neighbors of the node are already covered, the node will not rebroadcast



(a) (b) (c)

Fig. 1. Relative packet delivery ratio of ODMRP and SBA under different node densities (by increasing the number of nodes), group sizes, and node mobility. (a) Node mobility = 5 m/s. (b) Node mobility = 10 m/s. (c) Node mobility = 15 m/s.

the packet; otherwise, the node will schedule a time to rebroadcast the packet based on the number of neighbors that it has. The higher the number of neighbors, the sooner the node will rebroadcast the packet. This would therefore make nodes with higher degrees broadcast earlier than lower degree nodes. Thus, this can potentially enable the coverage of a large fraction of nodes with relatively few broadcasts.

3 Evaluating the Use of Broadcasting and Multicasting in MANETs

As discussed in Section 1, one can envision that trade-offs exist between the use of broadcasting and multicasting for group communications in MANETs. Depending on the scenario under consideration, one strategy may be preferable to the other. In this section, we perform extensive simulations using the ns-2 [22] simulator and consider a fairly large set of scenarios to evaluate and understand the trade-offs. From an intuitive standpoint, these studies are motivated from the observation that the construction and maintenance of a multicast structure could in fact be overhead intensive and may not provide any benefits as compared to simple broadcasting in certain scenarios. Specifically, we address the following question with regard to the suitability of using broadcast or multicast in various group communications scenarios: With what conditions is broadcasting favorable (in terms of packet delivery ratio, control overhead, and forwarding overhead) to multicasting and vice versa?

3.1 Trade-Offs between Broadcasting and Multicasting

As mentioned earlier, the candidate protocols under consideration are ODMRP and SBA. To evaluate the protocols, we consider a $1{,}250\ \mathrm{m} \times 1{,}250\ \mathrm{m}$ simulation area. Each node's transmission range is $250\ \mathrm{m}$. Nodes move as per the random waypoint model with constant speed and zero pause time until a total of 100 simulation seconds have elapsed. The final results are obtained by averaging the values measured over 30 simulation runs with different seeds. The parameters that we vary include 1) multicast group size, 2) node density, and 3) node mobility. By varying these parameters, we construct a large set of scenarios.

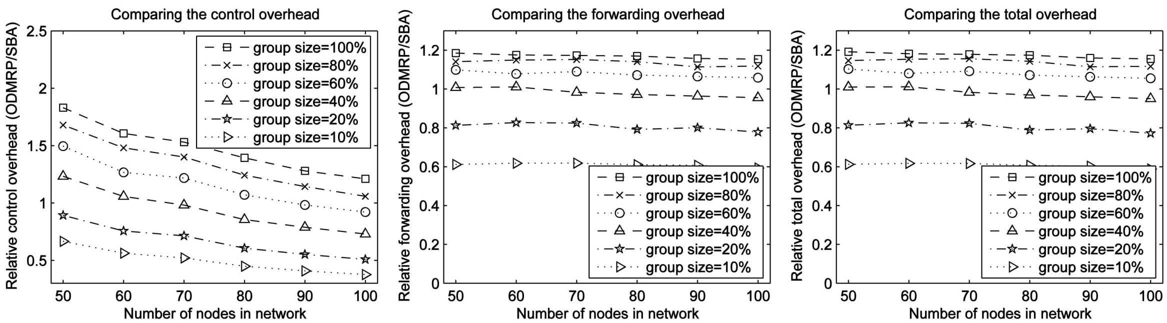
We define the multicast group size to be the ratio of the number of receivers to the total number of nodes. In the simulations, we use six different group sizes that range from 10 percent to 100 percent. The multicast group members are picked randomly from among the nodes in the network. The node density is varied by varying the number of nodes from 50 to a maximum of 100. We use three different *constant* speeds (5 m/s, 10 m/s, and 15 m/s) of nodes. The packet size is 512 bytes.

We compare the candidate protocols in terms of the achieved packet delivery ratio and the incurred overhead. When we examine the incurred overhead, we explicitly compare the **control overheads** due to the transmissions of *protocol-specific* control packets (expended in order to either construct or maintain a structure with the multicast approach and for the Hello messages in broadcast approach [4], [16]); **forwarding overheads** (due to redundant DATA packet transmissions) are also accounted for.

3.1.1 Observations and Interpretations

We present the results of our simulations experiments in Fig. 1 and Fig. 2. Fig. 1 depicts the comparison of the packet delivery ratio observed with the two protocols with varying node densities, group sizes, and node mobilities. Fig. 2 depicts the comparison of overhead observed with the two protocols under varying node densities and group sizes with a node mobility of 10 m/s. To aid comparison, we present the results in terms of the *relative* performance of ODMRP to SBA rather than presenting their respective raw results.

As seen in Fig. 1, SBA has a higher packet delivery ratio (by about 1-8 percent) than ODMRP in all of the considered scenarios. The higher packet delivery ratio of SBA is due to the inherent redundant rebroadcasts, which help SBA achieve a higher packet delivery ratio. However, one can also see (in Fig. 2c) that SBA, in scenarios with small group sizes (for group sizes below 40 percent), generates a much higher overhead than that of ODMRP. Specifically, when the group size is 10 percent, ODMRP incurs only around 60 percent overhead of that with SBA. The higher overhead of SBA in scenarios with small group sizes is mainly due to the high data forwarding overhead (see Fig. 2b) since SBA attempts to deliver data packets to the entire network regardless of the group size and potentially performs more rebroadcasts than what is needed in order to reach only the group members. Even though this broadcast redundancy



(a) (b) (c)

Fig. 2. Relative overheads of ODMRP and SBA under different node densities (by increasing the number of nodes) and group sizes. (a) Control overhead. (b) Forwarding overhead. (c) Total overhead.

provides SBA with a higher packet delivery ratio, its excessive overhead also renders it an unsuitable choice when the group membership size of the network is small. **These observations suggest that, for small group membership sizes ($<$ 40 percent of nodes are group members), ODMRP (or, in general, multicast) is preferable.**

When the group membership size is large (for group sizes above 40 percent), ODMRP incurs a much higher overhead than that with SBA (see Fig. 2c). Specifically, when the group size is 100 percent, ODMRP incurs 20 percent more overhead than that with SBA. This is because, when the number of multicast group receivers increases, a higher number of Join-Reply messages are sent by ODMRP and, thus, a higher number of forwarding nodes are set up. This creates additional redundant routes from the source to most of the destinations. This causes the overheads generated by ODMRP to be much higher than that of SBA. In contrast to ODMRP, SBA tries to disseminate the data packets to spawn the entire network with as few rebroadcasts as possible. By thus quelling unnecessary rebroadcasts, the overhead is significantly reduced. **The above observation suggests that, for large group sizes ($\geq$ 40 percent of nodes are group members), SBA (or, in general, broadcast) is preferable.**

We also observe in Fig. 1 that **increasing node mobility hurts ODMRP performance significantly, especially if the group membership size is small**. The node mobility is increased from 5 m/s for experiments whose results are shown in Fig. 1a to 10 m/s for those in Fig. 1b and to 15 m/s for those in Fig. 1c. When the node mobility increases, the packet delivery performance of ODMRP gradually degrades, especially when the group size is small. This is because the multicast structure stales faster with higher node mobility. In effect, this reduces the delivery of the right packets to the correct destinations. This is especially the case when the group size is small since there are fewer forwarding nodes, meaning that there exist fewer redundant routes. SBA, on the other hand, is relatively unaffected since the number of nodes that rebroadcast the packet is relatively unchanged with both mobility and with the multicast group size.

3.1.2 Effects of Data Packet Size

As we see in Fig. 2, the relative total overhead between ODMRP and SBA depends heavily on the forwarding overhead. This is due to the fact that the size of the data packets is much larger than the size of the control packets in the scenarios considered. In effect, the total overhead is dominated by the forwarding overhead. In order to gain a better understanding on the effects of data packet size on the tradeoffs between ODMRP and SBA, we conduct simulations with the same simulation settings as above but with different data packet sizes. The simulation results are shown in Fig. 3. When the data packet size decreases, the relative total overhead of ODMRP as compared to that of SBA increases. This is due to the fact that, in these scenarios, the advantage of having small overheads with SBA becomes more pronounced. Under the extreme cases where the data packet size is very small (say, 16-64 bytes), SBA may be further attractive for use when the multicast group membership size in the network is 20-30 percent. However, one might expect that, in typical scenarios, the size of the data packets is likely to be larger than the size of the control packets. Therefore, the trade-offs between broadcasting and multicasting that were observed with our previous set of studies (40 percent threshold) still hold in general.

In summary, our results suggest that there is no clear winner between the two schemes considered and that the scenario may in fact dictate the choice of multicast or broadcast. Our studies suggest that, in general, broadcasting is preferable in scenarios wherein a large fraction of the nodes are group members ($\geq$ 40 percent nodes are group members) and in high mobility. On the other hand, multicasting is preferable if the group membership is sparse ($<$ 40 percent node are group members) and with low to

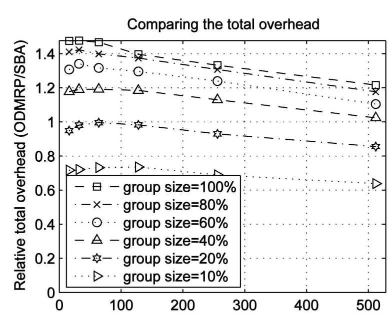


Fig. 3. The effects of data packet size on the relative total overhead between ODMRP and SBA.

moderate mobility. These are the features that we try to incorporate in our proposed new adaptive group communications protocol that we discuss next.

4 Fireworks: An Adaptive Group Communications Protocol

The design of Fireworks is mainly motivated by two high level observations from our studies discussed earlier. First, a simple broadcast scheme can significantly reduce the control overhead in scenarios wherein the density of group members is high. Second, many current protocols cannot adapt to local variations in network properties. Most of these protocols have static, globally predefined parameters that cannot be adjusted dynamically within localized regimes. Our objective then is to design a new protocol that 1) exploits the advantages of broadcasting in high densities and 2) provides localized flexibility in response to changing network conditions.

Fireworks dynamically identifies and organizes the group members into *cohorts* which correspond to areas of high group member affinity. In each of these "dense" neighborhoods, one of the group members is selected to be a *cohort leader*. Cohort leaders have two main functions: 1) they establish a sparse multicast structure among themselves and the source, and 2) they use broadcasting (with adaptive scope) to deliver the packets to other group members in their cohort.

The advantages of this approach are the high adaptability to local properties leading to significantly reduced overheads. This is achieved for the following four reasons:

1. Fireworks reduces the number of group members that participate in the formation and maintenance of the multicast structure (since only cohort leaders are involved in the process) and, in turn, lowers the control overhead,
2. the use of broadcasting in the member-intensive cohort region maximizes the "wireless broadcast advantage" [23],
3. the local broadcasts are resistant to changes in the local neighborhood due to mobility, and
4. constraining the broadcast to local neighborhoods of dense member affinity limits data redundancy overhead due to broadcasts.

4.1 High-Level Description

Fireworks, as its name implies, forms a *fireworks-like*[1] group communications structure for data packet delivery. Specifically, it constructs a 2-tier hierarchical structure (see Fig. 4), where the *upper tier* is formed by a multicast source (S in Fig. 4) and cohort leaders (A-E in Fig. 4) that represent groups of multicast members that form a *cohort*, and the *lower tier* consists of the members in a cohort. Since each cohort demonstrates a high density of group members, a cohort leader simply invokes an adaptive localized broadcast within its cohort to disseminate multicast packets received from the source. This would reduce the consumed overhead while ensuring efficient data delivery as discussed in Section 3.

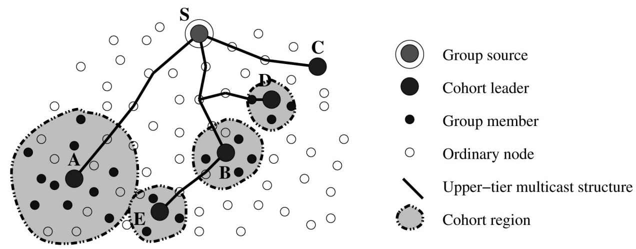


Fig. 4. Fireworks 2-tier multicast hierarchy structure.

4.2 Definitions of Protocol States and Data Structures

Fireworks employs a set of data structures and comprises multiple protocol states, which we define below. These definitions are used later when we detail protocol operations.

1. **Role** (*role*). Each group member in Fireworks has a *role*: It could either be in a transient mode wherein it is *JOINING* the session, it could be a cohort *LEADER*, or it could simply be the *CHILD* of a cohort leader.
2. **MGroup** (*mg*). This state variable, maintained by each group member, indicates the current multicast group of the group member.
3. **Leader** (*ldr*). This variable maintains the address of the cohort leader with which the group member is affiliated (if the group member is a child). If the group member is a cohort leader itself, this value is set to NULL.
4. **Distance** (*d*). The distance to the cohort leader is maintained by this state variable. If the group member is a cohort leader itself, this value could simply set to a very high value (i.e., infinity).
5. **Cohesiveness** (*c*). This is a state variable that maintains the affinity of group members within a node's k-hop[2] radius; it is computed as follows: The cohesiveness of a node, say i, is defined as:
$$c_i = \sum_{\forall n \in N_i^k} (k - distance_{i,n} + 1), \qquad (1)$$
where N_i^k is the set of group members that are within a k-hop radius from node i; the $distance_{i,n}$ is the hop distance from node i to node n. The higher the number and the closer the group members in its proximity, the greater will be the cohesiveness of a node.
6. **Join Group Table** (*JGTable*). This table, maintained at each node, maintains information with regard to the *JOINING* group members and the existing cohort leaders that are nearby. Each entry in the table contains the *address, mcast-address, role, distance*, and *cohesiveness* as it pertains to the nearby group member or cohort leader. The information maintained in this table is obtained by means of the

1. The transmission of data packets from the source to cohort leaders is analogous to emission of firework shells to some predefined spots in the sky; the broadcast of data packets by each leader in the cohort is analogous to the explosion of the fireworks at the predefined spots.

2. k is a system parameter. We consider the case when $k = 2$ since it gives the optimal trade-offs between performance and overhead.

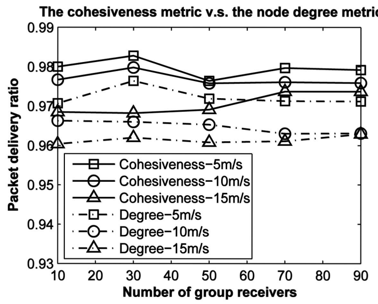


Fig. 5. Comparing two clustering metrics: cohesiveness and node degree.

ADVERTISE and the LEADER messages (to be discussed in Section 4.3).

7. **Cohort Member Table** (*CMTable*). This table is maintained *only* by cohort leaders. It contains information with regard to all the group members of the cohort (called children or cohort members) that are associated with the cohort leader. Each entry in the table contains the *address, mcast-address*, and the *distance* of each child. The information is obtained via the reception of CHILD messages that are sent out by each cohort member.

Remark 1. The aforementioned **cohesiveness** is used as the primary clustering metric for Fireworks since it helps Fireworks form cohorts with higher group member *affinity* and *stability* that cannot be provided by the other commonly used clustering metrics such as the ID-based [24] and the degree-based clustering metrics [25]. With cohesiveness defined in this way, Fireworks not only can form cohorts with a large number of group members, but also ensures that the group members are as close to the cohort leader as possible. This characteristic allows the formation of cohorts that maximizes the wireless broadcast advantage. Since group members are more concentrated around their respective cohort leader, group members are expected to stay longer within their respective cohort and thereby increase the stability of the cohort. To justify the above claim, we perform a simple experiment to compare the metric with other clustering metrics. Since the use of the ID-based clustering metric does not aim to create dense clusters, we only compare our cohesiveness metric with the node degree clustering metric (as in [25]). With the node degree clustering metric, the group members that have the highest number of group members in their k-hop neighborhood become candidates of cohort leader. In the experiment, the simulation area is 1,250 m $\times$ 1,250 m and the number of nodes is 100. We vary the group size from 10 percent to 90 percent and the mobility from 5 m/s to 15 m/s. All nodes are randomly distributed and they move according to the Random Waypoint model. Fig. 5 compares the packet delivery ratio of Fireworks with the two different clustering metrics. We see that, in all of the considered scenarios, using the cohesiveness as the clustering metric gives us a better packet delivery ratio than when using the node degree as the clustering metric.

4.3 Construction of the Fireworks Multicast Structure

The construction of our *fireworks-like* structure consists of three steps: 1) The determination of roles by group members, 2) the creation of the upper tier multicast structure, and 3) the employment of adaptive broadcast in the lower tier multicast structure (i.e., within a cohort). These steps are described below.

4.3.1 Role Determination of Group Members

The determination of the role of a group member is composed of two phases:

1. **Discovery Phase**. In this phase, the joining node discovers the other joining group members and cohort leaders in its vicinity. When a node decides to join a multicast group, it enters this phase and advertises its presence to its k-hop neighborhood by broadcasting an ADVERTISE message. The ADVERTISE message has a scope of k hops and contains the *address, mcast-address, hopcount*, and *cohesiveness* of the node. Upon the reception of a unique ADVERTISE message, nodes update their *JGTable* as per the contents in the message. After this phase, each joining node would have obtained the k-hop local topology information in their *JGTable*s (in the absence of packet losses). This information is used (if needed) in the decision phase (to be discussed) to determine the cohort leaders. Packet losses can result in a reduction in the accuracy of the topology information. However, our studies show that, due to the inherent redundancy provided by broadcasting, such losses are rare and have negligible effects on the performance of Fireworks. This phase may be triggered again when the connection to the cohort leader is lost.
2. **Decision Phase**. In this phase, the joining node determines if it should choose to be the cohort leader for its k-hop neighborhood. After the discovery phase, if a joining node cannot still find any cohort leader in its vicinity, it will enter this phase.[3] If the cohesiveness value of a node is the highest when compared to its k-hop neighbors, it will elect itself as a cohort leader and serve a cohort. It then changes its *role* to *LEADER* and broadcasts a LEADER message containing its *address, mcast-address, cohesiveness*, and *hopcount*. The TTL value of this message is set to k so as to notify the node's k-hop neighbors of the presence of a new cohort leader.[4] Nodes that are within the broadcast scope of the LEADER message update their *JGTable* to reflect the contents of the message.

During these phases, a joining node may receive several LEADER messages. If this is the case, the joining node will pick the best cohort leader to join (the best cohort leader is

3. Note that the first decision phase (during initialization) is started after at least two ADVERTISE messages have been sent. This is due to the fact that the first ADVERTISE message initially has a cohesiveness value of zero since, in the beginning, nodes are unaware of their neighborhoods.

4. As discussed later, the distribution scope of the subsequence LEADER messages could be dynamically adjusted.

the one that has the shortest distance and highest cohesiveness; further ties are broken by selecting the one with the highest nodeID) by unicasting a CHILD message containing its *address, mcast-address*, and *hopcount* to the selected cohort leader to notify the cohort leader of its intention to join the cohort. The cohort leader would then update its *CMTable* accordingly.

Note that, if a joining node is unable to find any cohort leader in its vicinity *and*, based on the above criteria, is unable to elect itself as a cohort leader, it will invoke additional instances of the discovery and the decision phases. Consequently, after the completion of the above phases, a joining node *must* either become a cohort leader or a child of a cohort leader. From then on, each cohort formed becomes a single routing entity as represented by its cohort leader. Only the relatively small number of cohort leaders will then participate in the construction and maintenance of the multicast structure. This role determination procedure is sufficient for a node to join the multicast group no matter the state of the network (either a multicast structure is in the initialization state or is already constructed).

In the scenarios where all the multicast group members are isolated, Fireworks is reduced to a pure multicast scheme. In this case, Fireworks would incur a slightly higher control overhead than a pure multicast scheme due to the transmissions of the ADVERTISE messages. However, the size and number of these messages is small (16 bytes); only two ADVERTISE messages are sent for each group member. Thus, the extra overhead incurred is not significant.

4.3.2 Creation of Upper Tier Multicast Structure

To enable the construction of the upper tier of the Fireworks multicast structure, the multicast source periodically broadcasts a SOURCE-QUERY message containing its *address* and *mcast-group* to the network. Intermediate nodes forward unique SOURCE-QUERY messages further and set up pointers backward toward the source. When a cohort leader receives the SOURCE-QUERY message, it unicasts a SOURCE-REPLY message back to the source via the route established by the aforementioned backward pointers. The nodes along the unicast path toward the source become the forwarding nodes for the group and are identified by the (*source, mcast-group*) attribute pair. From then on, data packets are multicast from the source to the cohort leaders via a tree constructed by coalescing the constructed reverse unicast paths. Note that this is not a source tree. Forwarding nodes, upon the receipt of SOURCE-REPLY from more than one cohort leader, conclude that they are the root of a multicast subtree and forward packets to their multiple children on the tree.

4.3.3 Adaptive Broadcast within Cohort

Once the cohort leader receives a data packet from the source, it performs a broadcast within its cohort to deliver the data packet to the associated group members. Note that the broadcast operation performed is adaptive in the sense that the maximum broadcast scope is not simply set to k hops, but instead depends on the furthest child of the cohort leader. In other words, the broadcast scope could be reduced as per the *distance* information of each furthest child, which is contained in the *CMTable*. This adaptability could reduce unnecessary transmissions of data packets that could result due to setting the broadcast scope too large. An example is illustrated in Fig. 4, where cohort leaders may have different broadcast scopes. The cohort leaders (B, D, and E) maintain cohorts of radius 1-hop since there are no children that are beyond this distance. In the extreme case, when a group member is isolated (Node C in Fig. 4), the isolated group member will become a cohort leader at the conclusion of the aforementioned phases. Such a singular leader has no children and, thus, will not perform any local broadcast.

4.4 Joining a Multicast Group

A node is considered to have joined a multicast group if its role is either that of the cohort leader or if it is deemed a child of a cohort leader. The process of joining a multicast group is described below.

When a node decides to join a multicast group, it simply changes its *role* to *JOINING* and enters the discovery and decision phase as described in Section 4.3.1. If the joining node has cohort leaders in its k-hop vicinity, it would possibly receive LEADER messages before entering the decision phase. If this is the case, the joining node will simply pick the best cohort leader to join (become a child of a cohort leader) as described in Section 4.3.1. If the joining node has no cohort leader present in its vicinity and its cohesiveness is the highest as compared to its k-hop neighbors, it will become a cohort leader and serve a cohort.

4.5 Leaving a Multicast Group

Group members could leave a multicast group at any time. A group member that has the *role* of *CHILD* simply stops unicasting the CHILD message to its cohort leader. Fireworks is based on maintaining soft-state and, after a predefined timeout, entries are purged from the tables listed earlier.

When a cohort leader decides to leave the multicast group, it simply stops transmitting the LEADER message. Cohort members, upon discovering the absence of a leader, will first try to quickly rejoin another cohort by looking for other leaders in their *JGTable*. If no cohort leader is present in a member's vicinity, the cohort member will switch its *role* to *JOINING* and invoke the discovery and decision phases to find another cohort or to become a cohort leader as described in Section 4.3.1.

4.6 Maintaining the Multicast Structure

Due to node mobility, the upper tier multicast structure and the formation of cohorts will have to be continually updated. We describe below the maintenance functionalities of different entities with Fireworks.

4.6.1 Source Functions

The source periodically refreshes the upper tier multicast structure (the tree to the cohort leaders) by triggering the exchange of SOURCE-QUERY and SOURCE-REPLY messages as described in Section 4.3.2. By means of this, the multicast tree structure might be refined. Stale routes may be purged and new ones created due to changes that occur as a result of mobility.

4.6.2 Cohort Leader Functions

Each cohort leader periodically broadcasts a LEADER message to its cohort. The purpose of this periodic announcement is to indicate its continued existence to the associated cohort members. In addition, this broadcast acts as an invitation to the leader's nearby *new* group members that are not currently associated with the cohort. Each cohort member ($role = CHILD$) sends updates that contain the distance of the member to its cohort leader regularly (to be discussed in detail). Using this, a cohort leader is able to dynamically adjust the scope of the local broadcast as mentioned earlier. The broadcast scope of the LEADER message is set to 2 hops if the number of cohort members (as recorded in *CMTable*) and the estimated number of new cohort members (specified in the *JGTable*) together is greater than a predefined threshold.[5] If these conditions do not hold, the LEADER message broadcast scope is set to 1 hop. The reason for reducing the LEADER message broadcast scope is that, when the number of cohort members becomes small, the advantages of performing local broadcasts are lost (as discussed earlier). This reduction of the broadcast scope of the LEADER message to a single hop is akin to simply resorting to unicast transmissions (by using the broadcast channel) from the source to the associated members of the cohort via the leader. Note that, in this case, the members are simply a hop away from the cohort leader.

4.6.3 Cohort Member Functions

Each cohort member periodically indicates its existence and updates its distance to its cohort leader so that the cohort leader can dynamically adjust its broadcast scope as discussed previously. This is done by unicasting a CHILD message to the cohort leader. The cohort leader will update its *CMTable* as per the contents of this message. Since the probability of a given cohort member implicitly leaving the associated cohort depends on the member's distance to the cohort leader (i.e., the closer the cohort member to its leader, the less possible it is that it moves out of scope), the frequency of these unicast updates from a member depends on this distance of the member from the leader. Our simulation results show that reducing the update frequency of the 1-hop cohort members has negligible effects on the performance of Fireworks in terms of the packet delivery ratio but significantly reduces the incurred control overhead.[6]

Sometimes, a cohort member may overhear LEADER messages of leaders from other cohorts. When this happens, the cohort member will see if the cohort leader that transmits the LEADER message is closer than its current cohort leader. If it is, the cohort member will switch to the new cohort by updating its state variables (ldr and d) and unicasting a CHILD message to the new cohort leader.

The connection between a cohort member to the cohort leader is deemed lost if the cohort member misses three consecutive LEADER messages from the cohort leader (via a time-out that accounts for this). In this case, the disconnected cohort member will, at first, try to rejoin a different cohort by looking for other leaders in its *JGTable*. If other cohort leaders are available, the disconnected cohort member will join the best leader as described in Section 4.3.1. If no leaders are found in the table, the disconnected cohort member will try to rejoin the group by invoking the discovery and decision phases as described in Section 4.3.1.

4.6.4 Relinquishing Cohort Leader Functionalities

A cohort leader will give up its *LEADER* role when it determines that it is no longer necessary to maintain itself as a leader. In Fireworks, a cohort leader that has no children is required to regularly check for the presence of other cohort leaders in its vicinity. Upon finding a leader, it will give up its own *LEADER* role and switch to a *CHILD* role by joining the discovered leader.

A second scenario that may lead to the relinquishment of cohort leader is when two or more cohort leaders come within the range (within k hops) of each other due to mobility. Even though Fireworks does not strictly enforce the existence of only a single leader within a k hop radius (since this may complicate the operation of Fireworks), cohort leaders may give up their roles if this was to happen. This is because members tend to migrate to the "best" cohort leader among the cohort leaders that drift together. This may cause some of the cohort leaders under discussion to lose all their cohort members. Such members would then relinquish their *LEADER* roles as discussed earlier.

Remark 2. Fireworks implicity takes mobility into account when constructing the data dissemination structure. Mobility of nodes is manifested as a continuous change of group memberships. Fireworks adapts to these changes by examining the group membership in each cohort and reforming cohorts as per the aforementioned operations. Note that Fireworks is robust to mobility due to the use of the cohesiveness as the clustering metric and the use of local broadcasts within cohorts. The use of the cohesiveness metric, as discussed, forms cohorts with high group member density and high stability. Combined with the use of local broadcasts within cohorts, the data dissemination structure of Fireworks is relatively resistant to changes. For instance, when $k = 2$, the maximum cohort radius is $250 \text{ m} \times 2 = 500 \text{ m}$. Even in high mobility, say 20 m/s (vehicular speed), the average amount of time that a group member resides in a cohort is of the order of tens of seconds. Data delivery at a high rate may be expected to take at most a few hundred milliseconds; the structure is thus fairly resistant to topological changes due to mobility.

5 Evaluations of Fireworks

In order to provide an extensive performance evaluation of Fireworks, we implement and simulate the protocol in ns-2 [22] and compare the obtained performance with that of various multicast and broadcast protocols. These protocols include ODMRP [4], MAODV [19], HDDM [20], and SBA

5. This threshold is set to five throughout our evaluations. This has been seen to be an appropriate threshold as per our previous studies [1].

6. In our evaluations, all 2-hop cohort members update at 3 seconds intervals and all 1-hop cohort members update at 9 seconds intervals. Small changes to these values did not cause the performance to change by much.

[16], and they are the representative protocols for mesh-based multicast, tree-based multicast, 2-tiered multicast, and broadcast, respectively. Since, for HDDM, the performance depends heavily on the accuracy and availability of the unicast routing information, we compare with Fireworks two versions of HDDM: 1) HDDM with omniscient unicast routing (HDDM-omniscient) and 2) HDDM with AODV (HDDM-AODV). With omniscient unicast routing, HDDM can obtain, immediately, the shortest route between any pair of nodes without the need to perform any route computations or disseminating queries. On the other hand, if AODV is used as the underlying unicast routing protocol, the routing information is either immediately provided by AODV (cached route due to the previous search) or a route request operation is performed in order to find a route to a specified destination.

The protocol parameters of each protocol are selected to conform with the settings in the original papers that describe them. For Fireworks and ODMRP, the source refresh interval is set to 3 seconds and the timeout for forward group is set to 4.5 seconds. For SBA, the hello message interval is 3 seconds. For MAODV, the hello message interval is set to 1 second and the group hello interval is set to 4 seconds. For HDDM, a hello message is broadcast for every 15 packets sent (it corresponds to approximately 3 seconds in most of the simulations). Note that the hello message interval used in SBA, MAODV, and HDDM have different meanings. In SBA, it is the interval for which a node broadcasts its 2-hop information. In MAODV, it is the interval for which a node broadcasts a beacon when it did not broadcast anything within the interval. In HDDM, it is the interval for which a source polls the roots of the subgroups. We believe that the parameters are chosen so as to evoke the best performance for each of the chosen protocols.

We divide our evaluations into three parts: In the first part, we evaluate the performance of Fireworks under *randomly constructed network scenarios*. In these scenarios, all nodes are uniformly and randomly distributed throughout the simulation area at the beginning of the simulation. The movements of nodes are guided by the random waypoint model. In the second part, our objective is to demonstrate the adaptability of the Fireworks under *clustered network scenarios*. The scenarios in question are similar to the random network scenarios but we intentionally include group formations to reflect clustered group members (cohorts) in the networks. The motion of these clustered group members are defined by the Reference Point Group Mobility (RPGM) model [26]. In this model, logical groups are defined and their movements are correlated with the motion of their so-called respective *reference points*. In our evaluation, we pick one node from each logical group to be the *reference node* and its position and speed is used to guide the motion of the members in its logical group. In the third part, we demonstrate that the extent to which Fireworks degrades the performance of coexisting unicast sessions in the network is much lower than the degradation experienced by such sessions due to concurrent pure multicast or broadcast schemes.

In the simulations, nodes have a transmission range of 250 meters and a maximum transmission rate of 2 Mb/s. The total simulation time is 100 seconds and we repeat the simulations 40 times with different seeds and obtain the average results. The first source (randomly chosen among the source nodes) begins the transmission of data at time 20 s and, if additional sources are present, they start transmitting data one after another (again, randomly chosen) with the starting instances separated by 0.5 s. Group members randomly join the group between [0, number of group members $\times$ 0.01) seconds. The data packet size is set to 512 bytes. Note that these generic parameters and scenario specific parameters (specified later) are, for the most part, conformant with these used in prior studies of the protocols with which we compare Fireworks [17], [4], [20], [27].

5.1 Simulating Random Network Scenarios

In these experiments, the parameters that we vary in order to evaluate the performance of Fireworks under different settings are: *group sizes, node mobility, number of sources*, and *traffic load*. The performance metrics that we are interested in are: *packet delivery ratio, data forwarding overhead*, and *control overhead*. These metrics are commonly used to evaluate the performance of a group communications protocol as in [20], [4], [17]. Note that the definitions of these performance metrics are the same as those we defined in Section 3.1.

The common simulation settings that are used in these experiments are the simulation area ($1{,}250\ \mathrm{m} \times 1{,}250\ \mathrm{m}$), the number of nodes (100) and the number of multicast groups (1).

5.1.1 Scenario 1: Varying Group Size and Node Mobility

First, we examine the effects of the group size and node mobility on the performance of Fireworks and compare the performance with that of ODMRP, HDDM, MAODV, and SBA. The common fixed parameters are the traffic load (5 pkts/s) and the number of sources (1).

The performances of the protocols under scenario 1 are shown in Figs. 6 and 7. The packet delivery ratio (see Fig. 6) with both Fireworks, ODMRP, and SBA approach 100 percent for all group sizes and node mobilities. The poor delivery performance with MAODV and HDDM (even with the aid of the omniscient routing information) is the consequence of the use of the tree-based multicast structure which does not provide enough robustness to withstand route breakages due to mobility. The performance of HDDM-AODV is particularly poor, especially with large group sizes, since it puts too much stress on the AODV routing protocol to find routes to the large number of group members. The heavy traffic generated from the route search process reduces the accuracy of the process and the throughput achieved for data dissemination.

In terms of the data forwarding overhead (see Fig. 7a), Fireworks incurs lower overhead as compared to ODMRP, HDDM-AODV, and SBA for all group sizes. The reason is that Fireworks adaptively uses multicast and broadcast (based on local network information) to disseminate data packets; this optimally reduces the number of broadcast operations performed. Even though Fireworks performs

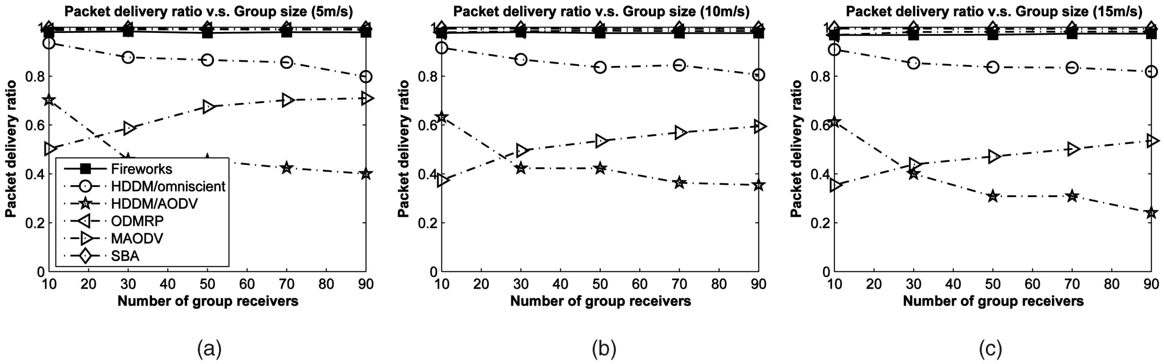


Fig. 6. Packet delivery ratio versus group size. Different plots show different node mobilities. (a) Mobility = 5 m/s. (b) Mobility = 10 m/s. (c) Mobility = 15 m/s.

broadcasts within each cohort, the incurred data forwarding overhead is still lower; this in turn implies that performing broadcasting in local cohorts is very effective. Even though HDDM-omniscient and MAODV has lower forwarding overhead than Fireworks due to the use of the tree-based approach, they also suffer a much worse delivery performance due to limitations of the same approach.

In terms of control overhead (see Fig. 7b), Fireworks is the clear winner. Both MAODV and HDDM-AODV incur a very high amount of control overhead. This is due to the fact that a large number of multicast update messages needs to be sent due to frequent changes in the multicast structure in the case of MAODV; a large number of AODV route query and route reply message exchanges are needed in order to discover/maintain the multicast structure in the case of HDDM-AODV. The discrepancy in terms of control overhead between HDDM-omniscient and HDDM-AODV lies in the inclusion of control overhead induced by HDDM on the AODV routing. As we see from the result, the large amount of control overhead produced from using a nonomniscient (realistic) unicast routing protocol causes HDDM to produce a considerable amount of control overhead. On the other hand, Fireworks, SBA, and ODMRP incur much lower amounts of control overhead. In addition, the simpler multicast structure provided with Fireworks results in less control overhead than with both ODMRP and SBA. Specifically, Fireworks incurs around 30 percent less control overhead than ODMRP when the group size is 10 percent (69 percent less when compared with SBA) and up to 50 percent less overhead when the group size is increased to 90 percent (36 percent less when compared with SBA).

The above discussion generally holds in typical cases where the source sending rate is larger than the rate of exchange of control messages and the data packet size is larger than the control packet size. For the cases where the traffic load is low (either because of low source sending rate or small data packet size), Fireworks would still be a better choice over both ODMRP and SBA, as evinced by the above results. This is because Fireworks attempts to find the sweet spot in terms of performance by judiciously invokes broadcasts in specific areas.

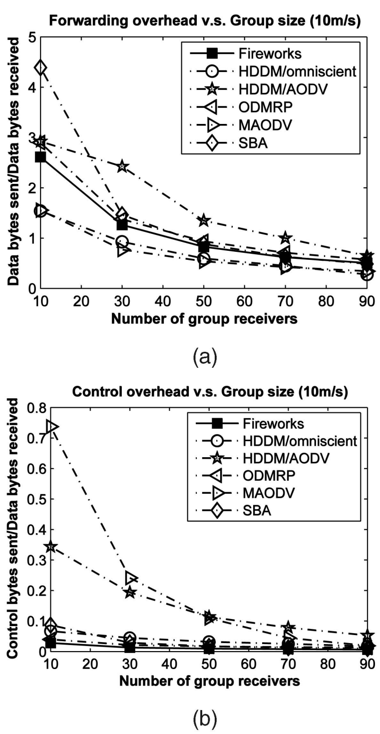


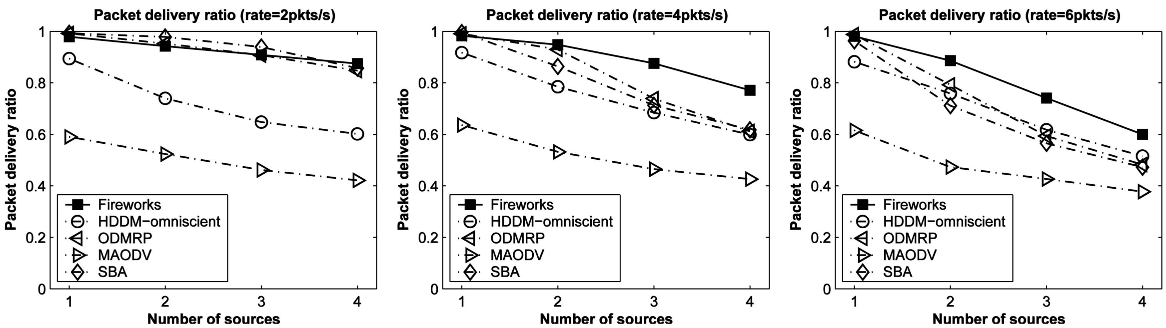
Fig. 7. Control and forwarding overhead versus group size with constant node mobility of 10m/s. (a) Forwarding overhead. (b) Control overhead.

5.1.2 *Scenario 2: Varying the Number of Sources and Traffic Load*

In this experiment, our objective is to study the effects of the number of sources and traffic load on the performance of Fireworks and compare the performance with the selected candidate protocols. The commonly fixed parameters are the number of group members (30) and the constant node mobility (5 m/s).

Note that, due to the poor performance of HDDM-AODV in these experiments, for purposes of clarity, we do not present results pertinent to HDDM-AODV in the rest of this section. The results were similar in nature and Fireworks outperformed HDDM-AODV in all the scenarios considered.

The performances of the protocols under scenario 2 are shown in Figs. 8 and 9, respectively. Even with an increase in the number of sources and traffic load, Fireworks is still

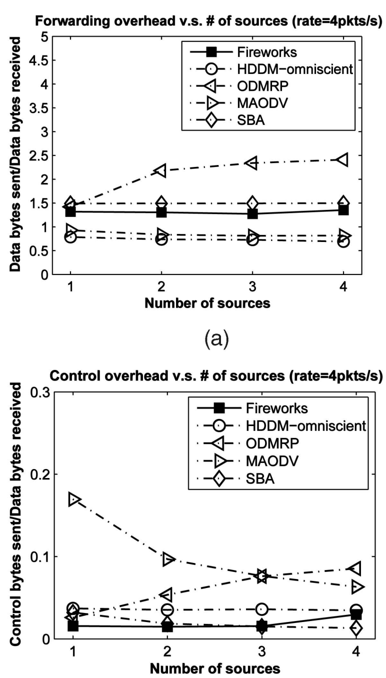


(a) (b) (c)

Fig. 8. Packet delivery ratio versus the number of source. Different plots show different source rates. (a) Source rate = 2 pkts/s. (b) Source rate = 4 pkts/s. (c) Source rate = 6 pkts/s.

able to maintain a better delivery ratio than any other candidate protocols in most of the cases (see Fig. 8). This is because Fireworks generates, in general, lower data forwarding and control message overhead. The contention and, therefore, collisions are thus less severe in Fireworks-enabled networks than in networks with the other candidate protocols. This is elucidated in Figs. 9a and 9b.

The data forwarding overhead and control overhead are in general much lower with Fireworks than with the other candidate protocols in all scenarios considered (HDDM-omniscient and MAODV have less forwarding overhead than Fireworks due to the inherent property of the tree-based approach. However, the reduced forwarding overhead also causes these protocols to achieve a very poor packet delivery ratio.). When there are more than one source, ODMRP incurs the highest amount of forwarding overhead since it creates a group-based mesh (the forwarding nodes that are created by any source of the group forward data packets for the group). The excessive redundancy created by ODMRP is not seen in Fireworks, as the created forwarding nodes are attributed by a specific (source, mcast-group) pair. Furthermore, the cohorts formed in Fireworks are shared between all the sources of the same group and, thus, the control overhead incurred by the cohorts will not be affected by the number of sources.



(a)

(b)

Fig. 9. Control and forwarding overhead versus group size with source rate of four packets per second. (a) Forwarding overhead. (b) Control overhead.

5.2 Simulating Clustered Network Scenarios

In these experiments, we want to further emphasize the benefits that Fireworks can offer due to its having considered group member affinity in constructing the multicast structure. Before we discuss our simulation experiments, let us discuss how Fireworks constructs the dissemination structure by adapting to the changing environment. We perform a simple experiment to illustrate the idea. In the experiment, we initially distribute 40 group members randomly throughout the $2{,}000 \text{ m} \times 2{,}000 \text{ m}$ simulation area. The group members move as per the RPGM model with a constant speed of 10 m/s. Therefore, no physical clusters are expected to be formed at the beginning of the simulation. As time progresses, these 40 group members will gradually move together to form two physical clusters. We take snapshots periodically during the simulation run and count the number of cohorts and the number of group members in the cohorts. Table 1 shows the average results of 20 simulation runs.

In the table, we see that the average number of cohorts decreases with time. Besides, the average number of group members in cohorts increases. This is because, at the beginning of the simulation, since group members are sparely distributed across the simulation area, Fireworks may potentially create many cohorts in different regions. However, the cohorts thus formed tend to be small in size (in terms of the number of group members) due to the low density of group members. As time progresses, the group members gradually move together and Fireworks adapts to the change by constructing fewer cohorts. These cohorts tend to include a fairly large number of group members. In effect, the multicast structure constructed by Fireworks is much simpler and is more efficient in disseminating data packets. In the following simulation experiments, we show that the adaptability provided by Fireworks leads to excellent packet delivery performance

TABLE 1
A Sample Trace of Our Simulations Showing Cohorts Statistics

Time	50s	100s	150s	200s	250s	300s
Average # of cohorts	9.20	8.35	6.10	5.40	5.15	4.25
Average # of group members/cohort	2.370	2.820	4.311	5.167	5.676	7.753

and incurs relatively lower overhead than any other candidate protocol considered.

In the following simulation experiments, clustered group members are introduced as discussed earlier. The main parameters of interest are 1) the density of group members in a cluster, 2) the distribution of group members, and 3) the size of the network (in terms of network dimension or number of nodes). We enumerate the performance of Fireworks in terms of the reduction in overhead as compared with ODMRP, MAODV and SBA (SBA is omitted in scenario 5 since it is clear that its overheads are far greater than the other candidate protocols under these simulation settings). In these scenarios, we are interested in comparing the overheads incurred by the candidate protocols.

Some common simulation parameters that are relevant to these experiments are the constant node mobility (5 m/s), the number of groups (1), and the number of sources (1).

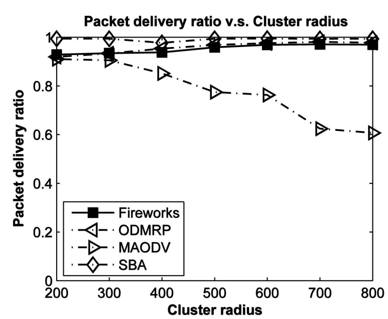
5.2.1 Scenario 3: Varying the Density of Group Members of a Cluster

In this experiment, we examine the effects of the density of group members of a cluster on the performance of Fireworks, ODMRP, MAODV, and SBA. We have a total of 300 nodes moving around in the $2{,}000 \text{ m} \times 2{,}000 \text{ m}$ simulation area. We construct one physical cluster that consists of one multicast group and 30 group members. The traffic rate is 2 pkts/s and the mobility of nodes is 5 m/s (constant). The parameter that we vary is the density of group members of the cluster.[7] Since the number of group members is fixed, the density of group members of the cluster can be varied by varying the size of the cluster. In this experiment, we consider that a cluster is a circular region where group members reside. The density can thus be varied by varying the radius of the cluster. We vary the cluster radius from 200 m to 800 m. Note that group members are evenly distributed over the cluster region.

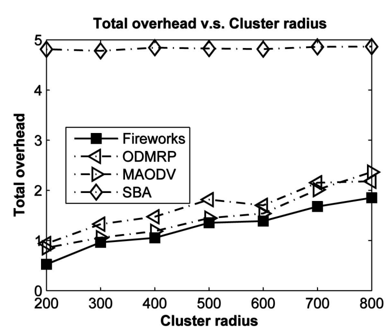
The comparisons of the packet delivery ratio and the total overhead of the protocols are shown in Fig. 10. As seen in Fig. 10a, the packet delivery ratio of Fireworks is much higher than with MAODV, is very similar to that of ODMRP, and almost approaches 100 percent as with SBA. However, the total overhead incurred by Fireworks is much lower than that of ODMRP, MAODV, and SBA (see Fig. 10b). This is because Fireworks is able to identify the cluster and construct a more efficient and simple multicast structure to disseminate packets to the group members. We see that, when the density of group members in the cluster increases (when the cluster radius decreases), the total overhead incurred by Fireworks decreases. For instance, when the cluster radius is 200 m, Fireworks constructs only one cohort. In this case, when a packet from the source arrives at the cohort leader, the cohort leader only requires broadcasting the packet once to disseminate the packet to all the group members in the cohort. When the cluster radius increases, Fireworks may construct more than one cohort since the physical cluster size is larger than the maximum cohort size (k-hop radius). However, Fireworks is still be able to construct a fewer number of routes from the source to the cluster so that the total overhead is lower than that of the other candidate protocols.

5.2.2 Scenario 4: Varying the Distribution of Group Members

In this evaluation, we examine the effects of the distribution of group members on the total overhead of Fireworks, ODMRP, MAODV, and SBA. The fixed parameters are the simulation area ($2{,}000 \text{ m} \times 2{,}000 \text{ m}$), the number of nodes (300), the traffic rate (2 pkts/s) and the group size (40). The distribution of group members varies from a purely random distribution to a complete clustered distribution. Specifically, we increase the number of logical clustered groups from 0 to 4. Each logical group consists of 10 group members and these group members move as per the RPGM



(a)



(b)

Fig. 10. Comparing the performance of protocols with different densities of group member in a cluster. (a) Packet delivery ratio. (b) Total overhead.

7. We wish to point out the difference between "cluster" and "cohort." "Cluster" here means the physical region where group members reside. "Cohort" is the logical grouping of group members that is constructed by Fireworks.

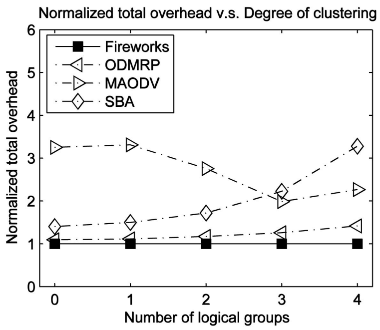


Fig. 11. Comparing the adaptability of FIREWORKS, ODMRP, MAODV, and SBA.

model. For those group members that are not in any logical group, the motion is as per the random waypoint model.

The overhead of the protocols are shown in Fig. 11. The overhead of each protocol is normalized with respect to that of Fireworks. As more logical groups are defined, the network becomes more clustered which means that group members move together (motion is correlated). We see that Fireworks is able to adapt to clustered motion far better than all the other protocols due to its inherent features, i.e., clustered regions with high concentration of group members can be covered by a small number of broadcast packets in Fireworks.

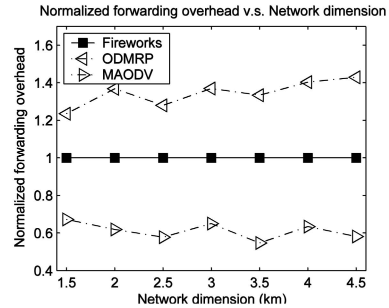
5.2.3 *Scenario 5: Varying the Network Size*

In this experiment, we examine effects of varying the network size on the overheads of Fireworks, ODMRP, and MAODV. The common fixed parameters are the traffic load (5 pkts/s) and the group size (40). We introduce two logical groups, each with 20 group members within a circular area of 400 m radius. The number of nodes increases when the physical network size increases such that the density of nodes is maintained. The number of nodes under various physical network sizes are: 180 in 1.5 km^2, 320 in 2.0 km^2, 500 in 2.5 km^2, 720 in 3.0 km^2, 980 in 3.5 km^2, 1,280 in 4.0 km^2, and 1,620 in 4.5 km^2.

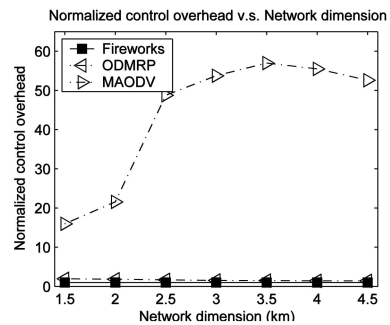
The overheads of the protocols are shown in Fig. 12. To aid the comparisons, we normalize the overhead of all protocols with respect to the overhead of Fireworks. Note that in this experiment, the average packet delivery ratio with both Fireworks and ODMRP approach around 90 percent or more but the average packet delivery ratio with MAODV is only around 45 percent.

As we see, Fireworks has much lower forwarding overhead and control overhead as compared to ODMRP (as the normalized overheads of ODMRP are both greater than 1). These results indicate that Fireworks is able to adapt to the environment better by identifying the logical groups and appropriately constructing fewer routes that are targeted toward the groups. As the network size increases, the average path length from the source to each multicast destination increases and treating each destination independently to construct a mesh (as with ODMRP) can lead to increased overheads.

Even though MAODV has a lower forwarding overhead than Fireworks, it incurs a much greater control overhead than Fireworks due to the frequent multicast structure updates due to the vulnerability of the structure to route changes. Given that MAODV fails to achieve a reasonably good packet delivery ratio (less than 50 percent), we do not consider MAODV as outperforming Fireworks in forwarding overhead.



(a)



(b)

Fig. 12. Comparing the overheads of Fireworks, ODMRP, and MAODV. The packet delivery ratio of Fireworks and ODMRP is around 90 percent and MAODV is only around 45 percent. (a) Normalized forwarding overhead. (b) Normalized control overhead.

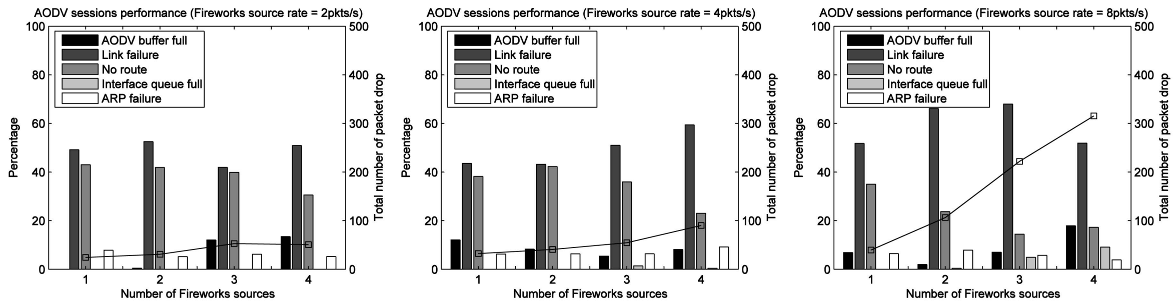
5.3 Effects of Fireworks on the Performance of Concurrent Coexisting Unicast Sessions

The adaptability provided by Fireworks has just been shown to significantly reduce the communication overheads. We further claim that such a reduction in the communication overheads could potentially reduce the impact on coexisting unicast sessions' performance. In order to validate our claims, we perform experiments to evaluate the impact of Fireworks on coexisting unicast sessions' performance. We also compare this impact with that of the impact of a pure multicast and a pure broadcast session on coexisting unicast sessions.

We choose the popular Ad-hoc On-Demand Distance Vector Routing (AODV) [28] as the representative unicast routing protocol and we use SBA and ODMRP as the representative broadcast and multicast protocols, respectively.

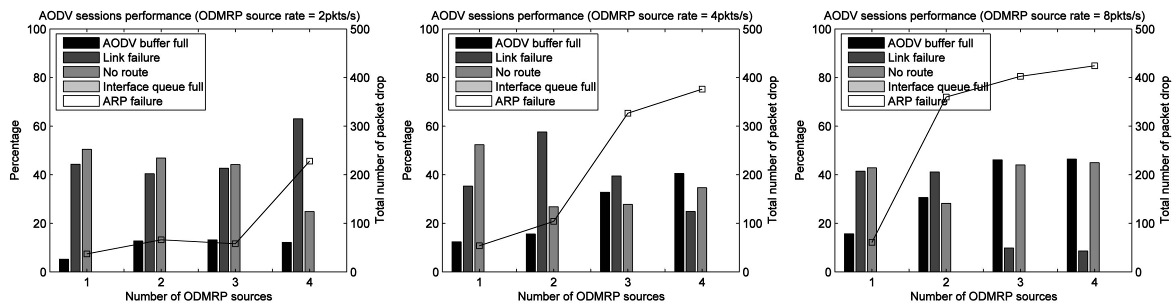
The simulation settings are as follows: There are 100 nodes in the $500\ \mathrm{m} \times 500\ \mathrm{m}$ simulation area and the node transmission range is 100 m. The total simulation time is 60 seconds. Nodes move at a constant speed of 5 m/s. The group membership size is fixed at 40 percent[8] (i.e., 40 randomly chosen nodes are group members). The AODV buffer size and the MAC layer queue size are set to 64 and 50 packets, respectively. In this experiment, we run two unicast sessions with a session rate of 5 packets/s (Therefore, a total of

8. We pick 40 percent as the group membership size as we have shown in Section 3.1 that, with this group membership size, broadcasting and multicasting perform comparatively.



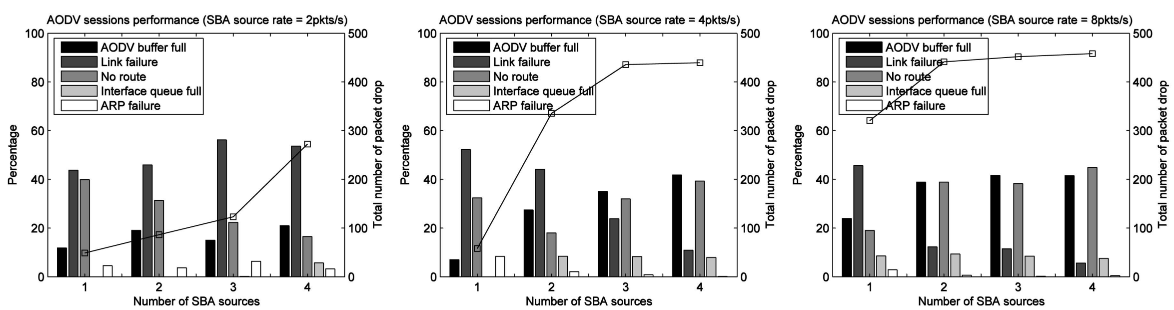
(a) (b) (c)

Fig. 13. Distributions of AODV packet losses with different number of Fireworks sources and traffic loads. Line plot indicates the total number of dropped packets. (a) rate = 2 pkts/s. (b) rate = 4 pkts/s. (c) rate = 8 pkts/s.



(a) (b) (c)

Fig. 14. Percentage distribution of reasons for AODV packet losses with different number of ODMRP sources and source data rates. Line plot indicates the total number of dropped unicast data packets. (a) ODMRP source rate = 2 pkts/s. (b) ODMRP source rate = 4 pkts/s. (c) ODMRP source rate = 8 pkts/s.



(a) (b) (c)

Fig. 15. Percentage distribution of reasons for AODV packet losses with different number of SBA sources and source data rates. Line plot indicates the total number of dropped unicast data packets. (a) SBA source rate = 2 pkts/s. (b) SBA source rate = 4 pkts/s. (c) SBA source rate = 8 pkts/s.

600 data packets are expected to be transmitted). In the mean time, we vary the number of Fireworks, ODMRP, or SBA sources (1, 2, 3, and 4 source(s)) in the network and also vary their rates (2, 4, and 8 packets/s) of transmission.

The performance of the unicast sessions in the presence of Fireworks, ODMRP, and SBA traffic are depicted in Fig. 13, Fig. 14, and Fig. 15, respectively. In these figures, we plot the total number of unicast packet drops (lines) and the distributions in percentage for the cause for these packet drops (bars) while varying the number of sources and source data rates. From these figures, we see that the number of unicast packet drops increases drastically with both ODMRP and SBA when the amount of traffic increases (due to the increase in the number of sources and source data rates). However, the number of unicast packet drops with Fireworks remains at a low level even when the amount of traffic is high. For instance, when there are four sources and the source rate is 4 packets/s, the percentages of unicast packet drops with Fireworks, ODMRP, and SBA are around 16 percent, 64 percent, and 71 percent, respectively (the total expected number of packets received is 600). We notice that there is a large number of unicast packet drops at the AODV buffer queue and at the MAC layer queue with both ODMRP and SBA (which account for around 40 percent of the total packet drops). This large number of unicast packet drops with ODMRP and SBA is due to the high overheads incurred with the protocols as discussed earlier. Essentially, these packet drops occur when there is a link failure upon which a large number of packets waiting in these queues that rely on that link are dropped. As the source traffic increases, the number of such link failures are also seen to increase because of what are called "false link" failures [29], [30]; these occur due to the deployment of the IEEE 802.11 MAC protocol [31]. Specifically, with the IEEE 802.11 MAC protocol, if the intended recipient of a Request-To-Send (RTS) packet is within the sensing range (interference range) of some other node, it does not respond to the sender of the RTS message with a Clear-To-Send (CTS) message. After seven consecutive attempts, the sender deems the link to have failed, although, in reality, it still exists.

In summary, we see that, due to the lower overhead incurred with Fireworks, its impact on coexisting unicast sessions is significantly smaller than that with ODMRP and SBA under all of the considered traffic pattern combinations, validating our claim.

6 Related Efforts

Numerous multicast protocols have been developed for use in MANETs. MAODV [19] is a multicast extension of its unicast counterpart. The operation of MAODV is analogous to the operation of AODV. Multicast routes are discovered on demand by broadcasting route request messages in a manner that is similar to the dissemination of unicast route requests; the route reply propagates back from the group members of the group to the source. Thus, a tree is constructed and data is propagated on the tree to the group members. ODMRP [4] is a mesh-based multicast protocol which creates a mesh structure for reliable data delivery. CAMP [5] constructs a group-shared mesh which makes use of a core node to reduce the control traffic needed for receivers to join the multicast group. AMRIS [6] makes use of ID numbers to guide the construction of a tree-based shared multicast structure, which supports multiple senders and receivers. AMRoute [7] is a hybrid multicast protocol which constructs a virtual multicast tree on top of the virtual mesh links established between group members. All of these protocols create a flat routing topology and are unaware of the topological characteristics of the structure. In [17], it was shown that ODMRP compares favorably to most of the other aforementioned multicast schemes; this motivated us to use ODMRP for the purposes of comparison with Fireworks and SBA. Note that, unlike Fireworks, none of the above schemes adopt broadcast features to adapt to local conditions.

Recently, a hierarchical multicast protocol called HDDM has been proposed in [20]. It is targeted to provide scalable multicasting in MANETs. The idea of the protocol is to extend the scalability of the Differential Destination Multicast (DDM) [32] protocol which was used to support multicasting in small groups. The protocol divides the entire network into different subgroups by selecting suitable subroots that are responsible for delivering data packets using the DDM protocol to their respective subgroup members. While HDDM requires the source to have a complete list of group members and requires an underlying unicast protocol to provide routing information, Fireworks does not. The unicast routing information is used by the HDDM source to determine its subroots. Each subgroup is basically a multicast tree that consists of subgroup members rooted at a selected subroot. Although Fireworks constructs a hierarchical structure, the criteria for the creation of the tiers and the purpose of the subgroups (cohorts in Fireworks) are substantially different in the two protocols. Fireworks constructs cohorts based on group member affinity, which aims at maximizing the wireless broadcast advantage. HDDM aims at providing a suitable sized subgroup for efficient DDM protocol deployment.

7 Conclusions

In this paper, we examine the impact of scenario-specific parameters on the performance of a group communications protocols. Our studies show that, in certain scenarios, a simple broadcast scheme can yield a packet delivery performance that is similar to that of a multicast protocol but with significantly lower overheads. We also observe that group communications sessions can have a drastic negative impact on coexisting unicast sessions. Our understanding motivated us to design a new hybrid adaptive group communications protocol that we name Fireworks. Fireworks exploits the property that the use of a broadcast scheme in an area of densely distributed group members could significantly reduce protocol overhead. It takes the group members affinity into account in constructing the data delivery structure and dynamically partitions a multicast group into several smaller cohorts in such a way that the formed cohorts manifest a high level of group affinity. A simple broadcast scheme is then used to provide a low-overhead data delivery service within these cohorts. From our simulation results, the *fireworks-like* data delivery structure constructed is shown to be lightweight in terms of the control and data forwarding overheads of the protocol. Since Fireworks employs broadcasting within a cohort, the inherent redundancy provides reliability and achieves a packet delivery performance that is comparable with that of a pure multicast and broadcast protocol. Moreover, due to the reduction in the communication overheads, Fireworks has lower levels of degrading influence on coexisting unicast sessions performance when compared with pure multicast or pure broadcast schemes. Even though Fireworks is specially designed for clustered networks, our results also demonstrate its superior performance as compared with various multicast and broadcast protocols under random network deployment scenarios.

Acknowledgments

This paper is an extended version of the combination of papers [1], [2], and [3]. It was prepared through collaborative participation in the Communications and Networks Consortium sponsored by the US Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory or the US Government.

REFERENCES

[1] L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, "On Evaluating the Trade-Offs between Broadcasting and Multicasting in Ad Hoc Networks," *Proc. IEEE Military Comm. Conf. (MILCOM '04),* 2004.

[2] L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, "Understanding the Interactions between Unicast and Group Communications Sessions in Ad Hoc Networks," *Proc. IFIP/IEEE Int'l Conf. Mobile and Wireless Comm. Networks (MWCN '04)*, pp. 1-13, 2004.

[3] L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, "Fireworks: An Adaptive Group Communications Protocol for Mobile Ad Hoc Networks," *Proc. IFIP Networking,* pp. 853-868, 2005.

[4] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '99),* pp. 1298-1304, 1999.

[5] E. Madruga and J. Garcia-Luna-Aceves, "Scalable Multicasting: The Core-Assisted Mesh Protocol," *ACM/Baltzer Mobile Networks and Applications,* special issue on management of mobility, vol. 6, no. 2, pp. 151-165, 2001.

[6] C. Wu and Y. Tay, "AMRIS: A Mulitcast Protocol for Ad Hoc Wireless Networks," *Proc. IEEE Military Comm. Conf. (MILCOM '99),* 1999.

[7] J. Xie, R. Talpade, A. Mcauley, and M. Liu, "AMRoute: Ad Hoc Multicast Routing Protocol," *Mobile Networks and Applications,* vol. 7, no. 6, pp. 429-439, 2002.

[8] K. Chen and K. Nahrstedt, "Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET," *Proc. IEEE INFOCOM,* pp. 1180-1189, 2002.

[9] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad Hoc Routing," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '99),* pp. 1313-1317, 1999.

[10] Y.-B. Ko and N. Vaidya, "Flooding-Based Geocasting Protocols for Mobile Ad Hoc Networks," *Mobile Networks and Applications,* vol. 7, no. 6, pp. 471-480, 2002.

[11] S.K. Das, B.S. Manoj, and C.S.R. Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks," *Proc. ACM MobiHoc,* pp. 24-35, 2002.

[12] S. Park and D. Park, "Adaptive Core Multicast Routing Protocol," *Wireless Networks,* vol. 10, no. 1, pp. 53-60, 2004.

[13] C. Jaikaeo and C.-C. Shen, "Adaptive Backbone-Based Multicast for Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '02),* 2002.

[14] C. de Morais Cordeiro, H. Gossain, and D.P. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," *IEEE Network,* Jan./Feb. 2003.

[15] P. Mohapatra, C. Gui, and J. Li, "Group Communications in Mobile Ad Hoc Networks," *Computer,* special issue on ad hoc networks, pp. 70-77, Feb. 2004.

[16] W. Peng and X.-C. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," *Proc. ACM MobiHoc,* pp. 129-130, 2000.

[17] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *Proc. IEEE INFOCOM,* pp. 565-574, 2000.

[18] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *Proc. ACM MobiHoc,* pp. 194-205, 2002.

[19] E. Royer and C. Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol," *Proc. ACM/IEEE MobiCom,* pp. 207-218, 1999.

[20] C. Gui and P. Mohapatra, "Scalable Multicasting in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2004.

[21] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. ACM/IEEE MobiCom,* pp. 151-162, 1999.

[22] S. McCanne and S. Floyd, "Ns-2 Simulator," http://www.isi.edu/nsnam/ns/, 2006.

[23] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Proc. IEEE INFOCOM,* 2000.

[24] M. Gerla and J.-C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *Wireless Networks,* vol. 1, no. 3, pp. 255-265, 1995.

[25] C. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *J. Selected Areas in Comm.,* vol. 15, no. 7, pp. 1265-1275, 1997.

[26] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," *Proc. ACM/IEEE Int'l Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '99),* pp. 53-60, 1999.

[27] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for Reliable Multicast in Multi-Hop Ad Hoc Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm.,* pp. 64-71, 1999.

[28] C.E. Perkins and E.M. Royer, "Ad Hoc On-Demand Distance Vector Routing," *Proc. IEEE Workshop Mobile Computing Systems and Applications (WMCSA '99),* pp. 90-100, 1999.

[29] S. Xu and T. Saadawi, "Does IEEE 802.11 MAC Protocol Work Well in Multi-Hop Wireless Ad Hoc Networks?" *IEEE Comm. Magazine,* vol. 39, no. 6, pp. 130-137, June 2001.

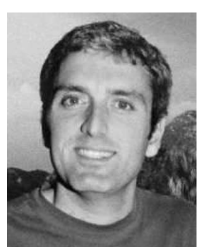
[30] F. Klemm, Z. Ye, S.V. Krishnamurthy, and S.K. Tripathi, "Improving TCP Performance in Ad Hoc Networks Using Signal Strength Based Link Management," *Ad Hoc Networks J.,* Mar. 2005.

[31] *ANSI/IEEE std 802.11,* IEEE, 1999, standards.ieee.org/getieee802/download/802.11-1999.pdf.

[32] L. Ji and M. Corson, "Differential Destination Multicast—A MANET Multicast Routing Protocol for Small Groups," *Proc. IEEE INFOCOM,* 2001.

Lap Kong Law received the BS degree in computer science from the Chinese University of Hong Kong in 2002. He is currently a PhD candidate in the Department of Computer Science and Engineering, University of California, Riverside. His research interests include routing protocols for unicast, broadcast, and multicast communications in ad hoc networks, TCP performance over wireless networks, and capacity analysis of cellular/ad hoc hybrid networks.

Srikanth V. Krishnamurthy received the PhD degree in electrical and computer engineering from the University of California at San Diego in 1997. From 1998 to 2000, he was a research staff scientist at the Information Sciences Laboratory, HRL Laboratories, LLC, Malibu, California. Currently, he is an associate professor of computer science at the University of California, Riverside. His research interests span CDMA and TDMA technologies, medium access control protocols for satellite and wireless networks, routing and multicasting in wireless networks, power control, the use of smart antennas, and security in wireless networks. Dr. Krishnamurthy has been a PI or a project lead on projects from various DARPA programs including the Fault Tolerant Networks program, the Next Generation Internet program, and the Small Unit Operations program. He is the recipient of the US National Science Foundation CAREER Award from ANI in 2003. He has also coedited the book *Ad Hoc Networks: Technologies and Protocols* (Springer Verlag, 2005). He has served on the program committees of INFOCOM, MobiHoc, and ICC and is the associate editor-in-chief for ACM MC2R.



Michalis Faloutsos received the bachelor's degree from the National Technical University of Athens and the MSc and PhD degrees from the University of Toronto. He is currently a faculty member in the Computer Science Department at the University of California, Riverside. His interests include Internet protocols and measurements, multicasting, and cellular and ad hoc networks. With his two brothers, he coauthored a paper on powerlaws of Internet topology (SIGCOMM '99), which is in the top 15 most cited papers of 1999. His work has been supported by several US National Science Foundation (NSF) and DAPRA grants, including the prestigious NSF CAREER award. He is actively involved in the community as a reviewer and a TPC member in many conferences and journals.