Fireworks: An Adaptive Group Communications Protocol for Mobile Ad Hoc Networks

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

Department of Computer Science & Engineering,
University of California,
Riverside, CA 92521, USA
{lklaw, krish, michalis}@cs.ucr.edu

Abstract. In group communications, we find that current multicast protocols are far from "one size fits all": they are typically geared towards and optimized for particular scenarios. As a result, when deployed in different scenarios, their performance and overhead often degrades significantly. A common problem is that most of these protocols incur high overheads with a high density of group members and in high mobility. Our objective is to design a protocol that *adapts* in response to the dynamics of the network. In particular, our objective is to provide efficient and lightweight multicast data dissemination irrespective of the density of group members and node density. Our work is motivated by two observations. First, broadcasting in some cases is more efficient than multicasting. Second, member and node layout distributions are not necessarily homogeneous. For example, many MANET applications result in a topological clustering of group members that move together. Thus, we develop **Fireworks**, an adaptive approach for group communications in mobile ad hoc networks. Fireworks is a hybrid two-tier multicast/broadcast protocol that adapts to maintain performance given the dynamics of the network topology and group density. In a nutshell, our protocol creates pockets of broadcast distribution in areas with many members, while it develops 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.

1 Introduction

Nodes can be distributed and move in clustered patterns in several MANET applications such as disaster recovery missions and military operations. We believe that this formation of the clustered topology could potentially be exploited to

* Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

R. Boutaba et al. (Eds.): NETWORKING 2005, LNCS 3462, pp. 853–868, 2005.

reduce the overhead incurred in multicasting. Such clustered topologies are typically characterized by sets of densely packed multicast group members within localized regions. Thus, one could simplify routing by regarding these sets as independent routing entities.

Most of the existing protocols that have been developed thus far [1, 2, 3, 8, 9, 10] do not take the affinity of group members into consideration when constructing their multicast delivery structure. A common problem with these protocols is that the control overhead could be unreasonably large, when the network manifests a dense distribution of group members globally or locally. The control overhead can be high when all the group members are required to participate in the construction and maintenance of the multicast structure. Furthermore, in a mobile environment, a large structure can be frequently under repair[1].

This work is motivated by two observations. First, we observe that a simple broadcast scheme can significantly reduce the control overhead in scenarios wherein the density of group members is high[4]. As a rule of thumb, broadcasting seems more efficient when 40% or more of the nodes in the network are group members. Second, many current protocols cannot adapt to local variations in network properties. Most of these protocols have static, globally pre-defined, parameters that cannot be adjusted dynamically within localized regimes. Our objective then is to design a new protocol that (a) exploits the advantages of broadcasting in high densities and (b) provides localized flexibility in response to changing network conditions.

We propose *Fireworks*, an adaptive multicast/broadcast protocol that exploits group members affinity to simplify routing and invoke broadcast operations in appropriate localized regimes. 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: (a) they establish a sparse multicast tree among themselves and the source, and (b) 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 three reasons: (a) Fireworks reduces the number of group members that participate in the formation and maintenance of the multicast structure and in turn lowers the control overhead, (b) the use of broadcasting in the cohort region maximizes the "wireless broadcast advantage"[2] [13], and (c) the local broadcasts are resistant to changes in the local neighborhood due to mobility.

We perform extensive simulations to evaluate the performance of Fireworks. We study a wide range of scenarios by varying the group sizes, the node mobilities, the number of sources, the traffic load, and the spatial distributions of group members in order to understand the limits of the performance of group commu-

[1] We discuss related work on efforts that address scalable multicasting in section 4.
[2] The fact that every transmission is a broadcast and thus heard by all neighbors.

nications. More specifically, we compare the performance of Fireworks with that of ODMRP[2]. ODMRP has been shown previously to compare favorably with most previously proposed multicast protocols[5]. Fireworks is shown to perform comparably with ODMRP in terms of the packet delivery statistics but with significantly lower overheads. In the presence of multiple group communications sessions or other unicast connections in the network (as the case would be in typical deployment scenarios), this reduction in overheads is especially desirable.

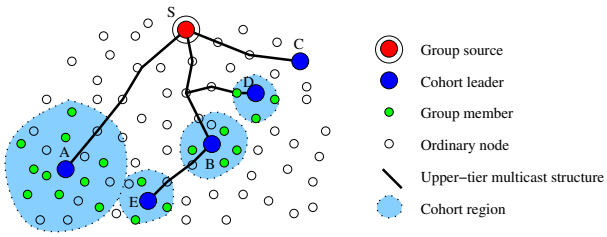


Fig. 1. Fireworks 2-tier multicast hierarchy structure

The rest of the paper is organized as follows: In the next section, we provide a detailed description of our proposed protocol. In section 3, we present our simulation framework and discuss the observed results. Comparisons with ODRMP are also deliberated in this section. We discuss some related works in section 4 and conclude the paper in section 5.

2 Protocol Description

Fireworks, as its name implies, forms a *fireworks-like*[3] group communications structure for data packet delivery. Specifically, it constructs a 2-tier hierarchical structure (see Fig. 1) where the *upper tier* is formed by a multicast source (S in Fig. 1) and cohort leaders (A-E in Fig. 1) 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 observed in [4].

In the following subsections, we describe the details of Fireworks. We deliberate on the construction of the two tier hierarchy and the actual use of the construction for data delivery.

[3] 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 fireworks at the predefined spots.

2.1 Definitions of Protocol States and Data Structures

Prior to our detailed discussion of Fireworks, we define a few data structures and protocol states that are employed. These definitions are used throughout the latter sub-sections that 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, could be a cohort *LEADER* or 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[4] radius; it is computed as follows:

   The cohesiveness of a node, say i, is defined as:

   $$c_i = \sum_{\forall n \in N_i^k} (k - distance_{i,n} + 1) \tag{1}$$

   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 the cohesiveness of a node has.
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 ADVERTISE and the LEADER messages (to be discussed in section 2.2).
7. **Cohort Member Table** ($CMTable$). This table is maintained *only* by cohort leaders. It maintains 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.

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

2.2 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:

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 an 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 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. If after the discovery phase, if a joining node cannot still find any cohort leader in its vicinity, it will enter this phase. [5] If its cohesiveness value is the highest as compared to its k-hop neighbors[6], 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[7]. Nodes that are within the broadcast scope of the LEADER message update their *JGTable* to reflect the content of the message.

[5] 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.

[6] This is indicated by the entries built up in its *JGTable*.

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

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[8] 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.

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 towards 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 towards 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 sub-tree and forward packets to their multiple children on the tree.

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 as a result of setting the broadcast scope too large. An example is illustrated in Fig. 1 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

[8] The best cohort leader is the one that has the shortest distance and highest cohesiveness (in that order); further ties are broken by selecting the one with the highest nodeID.

member is isolated (Node C in Fig. 1), 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.

2.3 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 2.2. 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 2.2. 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.

2.4 Leaving a Multicast Group

Group members could leave a multicast group at anytime. 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 2.2.

2.5 Maintaining the Multicast Structure

Due to the 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.

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 8. 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.

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 rebroadcast 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[9]. If these conditions do not hold, the LEADER message broadcast scope is set to 1 hop. The reason of reducing the LEADER message broadcast scope is that when the number of cohort members becomes small, the advantages of performing local broadcasts are lost. 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.

Cohort Member Functions. Each cohort member periodically indicates its existence and updates its distance to its cohort leader so that the cohort leader could 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[10].

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 3 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 2.2. If no leaders

[9] This threshold is set to 5 throughout our evaluations. This has been seen to be an appropriate threshold as per our previous studies [4].

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

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 2.2.

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.

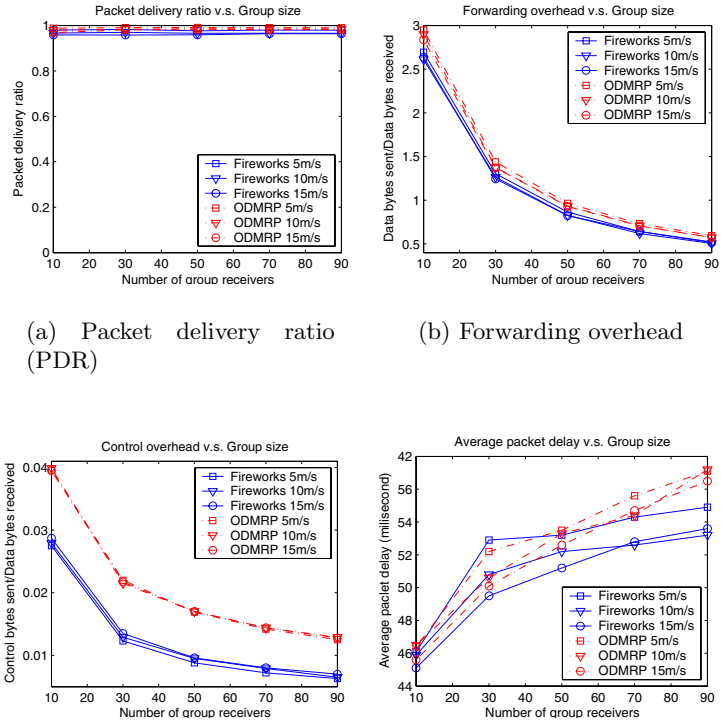
A second scenario that may lead to the relinquishment of cohort leader is when two or more cohort leaders come within the range (within k hops) of each other due to mobility. Even though Fireworks does not strictly enforce the existence of only a single leader within a k hop radius (since this may complicate the operation of Fireworks), cohort leaders may give up their roles if this were 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.

3 Performance Evaluation

In order to evaluate the performance of Fireworks, we implement and simulate the protocol in NS-2[6] and compare the obtained performance with that of the well-known multicast protocol, the On-Demand Multicast Routing Protocol (ODMRP). We pick ODMRP as a baseline protocol since it has shown to be one of the elite protocols in its class[5].

We divide our evaluations into two 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 formations to reflect clustered group members (cohorts) in the networks. The motion of these clustered groups members are defined by the Reference Point Group Mobility (RPGM) model[7]. 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 simulations, nodes have a transmission range of 250 meters and a maximum transmission rate of 2Mb/s. The total simulation time is 100 seconds and we repeat the simulations for 40 times and obtain the average results. The first source (randomly chosen among the source nodes) begins the transmission of data at time 20s and if additional sources are present, they start transmitting



(a) Packet delivery ratio (PDR)

(b) Forwarding overhead

(c) Control overhead

(d) Average packet delay

Fig. 2. Comparing the performance of Fireworks and ODMRP under random network scenarios with varying group sizes and mobilities

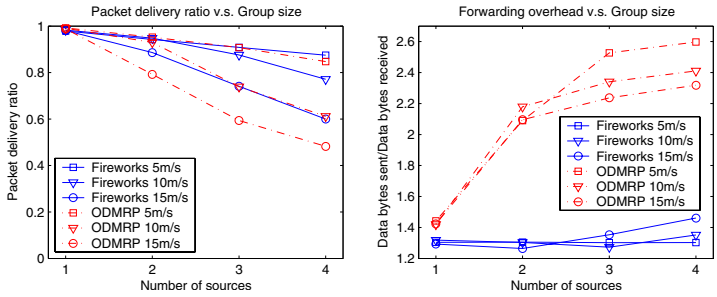
data one after another (again randomly chosen) with the starting instances separated by 0.5s. Group members randomly join the group between [0, number of group members × 0.01) seconds. The data packet size is set to 512 bytes.

3.1 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*, *control overhead* and *average packet delay*.

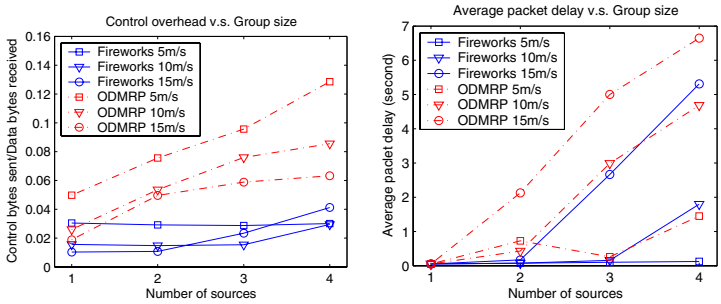
The common simulation settings that are used in these experiments are the simulation area (1250m×1250m), the number of nodes (100) and the number of multicast groups (1).

Simulation 1: Group Size and Node Mobility. First, we examine the effects of the group sizes and node mobility on the performance of Fireworks and



(a) Packet delivery ratio (PDR)

(b) Forwarding overhead



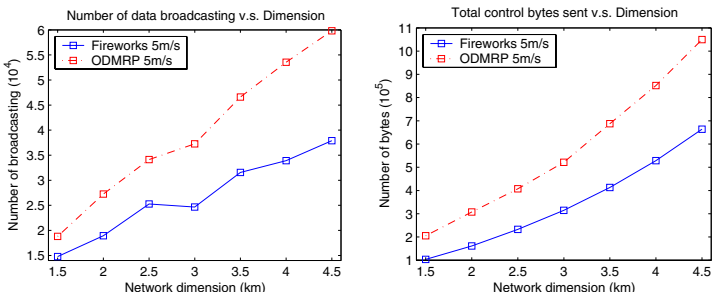
(c) Control overhead

(d) Average packet delay

Fig. 3. Comparing the performance of Fireworks and ODMRP under random network scenarios with varying number of sources and traffic rates

compare the performance with that of ODMRP. The common fixed parameters are the traffic rate (5 pkts/s) and the number of sources (1).

The results of these simulations are shown in Fig. 2. The packet delivery ratios with both Fireworks and ODMRP approach a 100% for all group sizes and node mobilities. In terms of the data forwarding overhead, Fireworks incurs around 10% less overhead as compared to ODMRP. This reduction in data forwarding overhead is due to the simpler multicast structure constructed by Fireworks as compared with ODMRP. Although Fireworks does perform 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. In terms of control overhead, Fireworks is the clear winner. Fireworks incurs 30% less overhead when the group size is 10% (10 % of the nodes in the network are group members) and up to 50% less overhead when the group size is increased to 90%. The reduction in control overhead is especially significant when the data packet size is small (comparable to the control packet size), since the total incurred overhead would then be dominated by the control overhead (as opposed to data overhead due to redundant data transmissions). In terms of average packet delay,



(a) Total number of data broadcasting

(b) Total number of control bytes sent

Fig. 4. Comparing the overheads of Fireworks and ODMRP

Fireworks also tends to have lower delay than ODMRP. This could potentially be due to the lower queuing delay thanks to the reduction in the load due to the overhead, with Fireworks.

Simulation 2: 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 that of ODMRP. The common fixed parameters are the number of group members (30) and the node mobility (5m/s).

The simulation results are shown in Fig. 3. With an increase in the number of sources from 1 to 4 and with traffic loads varying from 2pkts/s to 6pkts/s, the packet delivery ratios of both ODMRP and Fireworks decrease. However, as observed, Fireworks is able to maintain a better delivery ratio under higher traffic load and with a higher number of sources. This is because Fireworks generates, in general, lower data forwarding and control message overhead than ODMRP. The contention and therefore, collisions are thus less severe in Fireworks enabled networks than in ODMRP enabled networks. This is elucidated in Figures 3(b) and 3(c). The data forwarding overhead and control overhead are much lower with Fireworks than with ODMRP in all scenarios considered. ODMRP creates a group-based mesh, and the excessive redundancy created by the mesh 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. Note that due to the lower congestion level in the network, the average packet delay incurred by using Fireworks is much smaller than that of ODMRP in all scenarios.

3.2 Simulating Clustered Network Scenarios

In these experiments, we want to further emphasize the benefits Fireworks can offer due to its having considered group member affinity in constructing the multicast structure. In the scenarios considered, clustered group members are

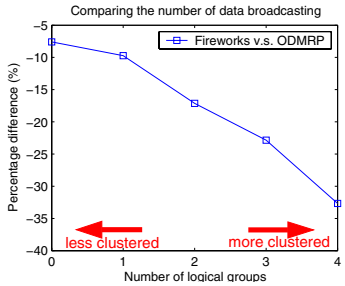


Fig. 5. Comparing the adaptability of FIREOWKRS to ODMRP

introduced as discussed earlier[11]. The main parameters of interest are (i) the size of the network (in terms of network dimension or number of nodes) and the correlation between group members' motion patterns. We enumerate the performance of Fireworks in terms of the reduction in overhead as compared with ODMRP. The performance metrics that we consider are *the number of data forwarding* instances and *the number of control bytes* incurred.

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

Simulation 3: Varying the Network Size. In this experiment, we would like to see the effects of varying the network size on the performance of Fireworks and ODMRP. The common fixed parameters are the traffic rate (5 pkts/s) and the group size (40). We introduce 2 logical groups, each with 20 group members within a circular area of 400m 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 1.5km^2, 320 in 2.0km^2, 500 in 2.5km^2, 720 in 3.0km^2, 980 in 3.5km^2, 1280 in 4.0km^2 and 1620 in 4.5km^2.

The results are shown in Fig. 4. As we see, increasing the physical network size (number of nodes) increases the number of data broadcasting performed and the amount of control bytes sent in both Fireworks and ODMRP. However, Fireworks has much smaller number of data broadcasting and control bytes sent as compared to ODMRP. These results indicate that Fireworks is able to adapt the environment better by identifying the logical groups and appropriately constructing fewer routes that are targeted towards 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 overhead.

[11] As mentioned, this could represent a group of firefighters or rescue workers operating in the proximity of each other or a team of soldiers in a tactical mission.

Simulation 4: Examining Clustered Motion. In this evaluation, we examine the effects of clustered motion (as discussed) on Fireworks and ODMRP. The fixed parameters are the simulation area (2000m×2000m), the number of nodes (300), the traffic rate (2 pkts/s) and the group size (40). 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 model. For those group members that are not in any logical group, the motion is as per the random waypoint model.

The results of the simulations are shown in Fig. 5. The results represent the percentage differences in the number of data broadcasting instances between Fireworks and ODMRP. 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 ODMRP due to its inherent features as discussed earlier.

4 Related Works

Numerous multicast protocols have been developed for use in MANETs. MAODV[1] is a multicast extension of its unicast counterpart. ODMRP[2] is a mesh-based multicast protocol which creates a mesh structure for reliable data delivery. CAMP[8] constructs a group-shared mesh which makes use of a core node to reduce the control traffic needed for receivers to join group. AMRIS[9] makes use of ID number to guide the construction of a tree-based shared multicast structure which supports multiple senders and receivers. AMRoute[10] 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. Unlike Fireworks, none of the previous schemes adopt broadcast features to adapt to local conditions.

Recently, a hierarchical multicast protocol called HDDM has been proposed in [11]. It is targeted to provide scalable multicasting in MANETs. The idea of the protocol is to extend the scalability of the DDM[12] multicast protocol which was used to support multicasting in small groups. The protocol divides the whole network into different sub-groups by selecting suitable sub-roots that are responsible for delivering data packets using DDM protocol to their respective sub-group 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 sub-roots. Each sub-group is basically a multicast tree that consists of sub-group members rooted at a selected sub-root. Although Fireworks constructs a hierarchical structure, the criteria for the creation of the tiers and the purpose of the sub-groups (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 sub-group for efficient DDM protocol deployment.

5 Conclusions

In this paper, we propose a new hybrid multicast/broadcast scheme for MANETs. The construction is primarily geared towards reducing overheads incurred with group communications in MANETs. 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 a packet delivery performance that is comparable with that of a pure multicast protocol (ODMRP) is achieved. Even though Fireworks is specially designed for clustered networks, our results also demonstrate its superior performance as compared with ODMRP under random network deployment scenarios as well.

Note: 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 Army Research Laboratory or the U. S. Government.

References

1. E.M. Royer and C.E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in Proceedings of the 5th annual ACM/IEEE MOBICOM, 1999, pp. 207-218.
2. S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," in Proceedings of IEEE WCNC, 1999, pp. 1298-1304.
3. C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," in Cluster Computing, vol.1, no.2, 1998, pp. 187-196.
4. L.K. Law, S.V. Krishnamurthy, and M. Faloutsos, "On Evaluating the Trade-offs between Broadcasting and Multicasting in Ad Hoc Networks," in Proceedings of the IEEE MILCOM, 2004.
5. S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," in Proceedings of IEEE INFOCOM, 2000, pp. 565-574.
6. S. McCanne and S. Floyd, "Ns-2 simulator." http://www.isi.edu/nsnam/ns/
7. X. Hong, M. Gerla, G. Pei and C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," in Proceedings of ACM/IEEE MSWiM, 1999, pp. 53-60.
8. E.L. Madruga and J.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, Apr. 2001.
9. C. Wu and Y. Tay, "AMRIS: A Mulitcast Protocol for Ad Hoc Wireless Networks," in Proceedings of IEEE MILCOM'99, Atlantic City, NJ, Nov. 1999.

10. J. Xie, R.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.
11. C. Gui and P. Mohapatra, "Scalable Multicasting in Mobile Ad Hoc Networks," in Proceedings of IEEE INFOCOM, 2004, Hong Kong.
12. L. Ji and M.S. Corson, "Differential Destination Multicast - A MANET Multicast Routing Protocol for Small Groups," in Proceedings of IEEE INFOCOM, 2001, Anchorage, Alaska.
13. J. Wieselthier, G. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," in Proceedings of IEEE INFOCOM, 2000.