Team Communications among Autonomous Sensor Swarms

Mario Gerla and Yunjung Yi
{gerla, yjyi}@cs.ucla.edu
University of California, Los Angeles, CA 90095

Abstract--[1] *In this paper, we consider team (swarm) of unmanned vehicles (UVs) equipped with various sensors (videos, chemicals, etc). Those swarms need efficient communication to feed sensed data, communicate data to other swarms, to navigate and, more generally, to carry out complex mission autonomously. We focus on a particular aspect of mission oriented communications, namely,* ***team multicast****. In team multicast, the multicast group does not consist of individual members, rather, of teams. In our case, the teams may consist of special UVs that have been established to launch a search and rescue mission. Simulation results illustrate the performance benefits of the team multicast solution as compared with more traditional multicast approaches.*

1. Introduction

With the advances in wireless ad hoc communications, robotics and microflyer technology, it will be possible to organize teams (swarms) of small unmanned ground, sea and airborne vehicles and to launch complex missions that comprise several such teams [1]. The typical application scenario is a disaster area that requires the intervention of police, firemen, paramedics etc, but where the unfriendly environment bars direct access [2]. The swarm operates in a completely distributed, autonomous manner, establishing communications network between the rescue teams and all critical fixed and mobile sensors and actuators in the disaster area. It allows the police to "see", probe and manipulate the environment remotely before they can safely enter. Possible emergency scenarios include: chemical, nuclear plant disaster/sabotage; fire on a ship; explosion/fire on high rise building, etc. Other non-emergency application domains can benefit of the proposed swarm technology. For instance, space and planetary exploration (e.g., Mars), collection of scientific data in remote, sparsely instrumented regions, etc.

In the aftermath of a disaster, we may assume that some "networked islands" of sensors, monitors and actuators have survived in the plant. For example, sensors may have been installed in strategic locations in the plant, building, ship etc; they had been preplanned for such an emergency. However, full sensor coverage and networking are generally not available after the accident, either because it was not practical or too costly to establish a complete infrastructure or because the infrastructure has been partially destroyed. Consequently, the entire area becomes partitioned into islands. The rapid deployment of a swarm of air/ground agents will reestablish network connectivity, restore access to critical sensor probes, install new probes as necessary and help the collection and filterind of relevant data.

As an example, consider a chemical attack in the subway system. Police, fire dept and paramedics have been alerted, but the environment is unfriendly. The rescuers must first probe the environment to predict how the chemical cloud will propagate in the system and decide when and from where to enter and rescue the people trapped in the tunnels. They send in swarms of UGVs and UAVs to collect data samples. The prediction of chemical cloud diffusion may be aided by computer models and by simulation which are fed directly by field measurements, allowing through this iterative approach a better assessment of the situation. The simulator can also be used to ask "what if" questions, for instance, how open vent and move the trains in order to avoid casualties, when is it safe to let the firemen enter the scene. Clearly, efficient communications from swarm to commander are essential for the success of this task.

The paramedics can also use virtual reality techniques to do a "virtual walk through" of the facilities. The virtual reality database was likely set up in advance. Fresh inputs, collected by swarms of UAVs (Unmanned Airborne Vehicles) and UGVs (Unmanned Ground Vehicles), can be used to overlay chemical diffusion data on top of the existing subway database. Note that parts of this database (e.g., detailed blueprints) are used also by the swarms to navigate through the system. Such blueprints can be preloaded in the UAVs/UGVs before injecting them in the subway, or (if the data is too large for the limited memory on board) each swarm can prefetch (on demand) the part of the database it needs as it navigates through the tunnels.

In this paper, the main focus will be on UV (Unmanned Vehicle) teams. We first review the communication requirements of a system of "swarms" working collectively to perform a mission. We then focus on a particular aspect of mission oriented communications, namely, **team multicast**. In team multicast, the multicast

---

[1] This work is supported in part by ONR ``MINUTEMAN" project under contract N00014 - 01 - C - 0016

group does not consist of individual members, rather, of teams (swarms). In particular, nodes in the same team (swarm) will have coordinated motion. We call this model the "affinity team model". For example, a team may be a special task force that is part of a search and rescue mission. The message then must be broadcasted to the various teams that are part of the multicast group, and, to all UVs within each team. For example, a weapon carrying airborne UV may broadcast an image of the target (say, a poison gas plant) to the reconnaissance and sensor teams in front of the formation, in order to get a more precise fix on the location of the target. The sensor UV team(s) that have acquired such information will return the precise location. As another example, suppose N teams with chemical sensors are assessing the "plume" of a chemical spilled from different directions. It will be important for each team to broadcast its findings step by step to the other teams using team multicast. In general, team multicast will be common place in ad hoc networks designed to support collective tasks, such as occur in emergency recovery or in the battlefield. Naturally, a two-tier communication can be achieved in team multicast where a source sends data packet to a representative of each subscribed team in multicast group first and then the representative forwards the packet within a team.

As a first prototype of team multicast, we develop Multicast-enabled Landmark Ad Hoc Routing (denoted as M-LANMAR). M-LANMAR is an extension of LANMAR (Landmark Ad hoc routing) [4, 5] protocol enabling multicast. M-LANMAR creates a tunnel from the multicast source to each representative node (say, landmark) of the subscribed multicast group (of teams). It then sends a separate copy of the packet to each landmark (i.e., multiple unicast). Once the packet has reached the target landmark, it is broadcasted to all nodes within the team using restricted flooding.

At first glance, the M-LANMAR approach may seem inefficient. However, this approach has a few advantages over traditional multicast protocols such as ODMRP (On-Demand Multicast Routing Protocol) [8] and MAODV (Multicast extension of AODV) [10]. First, M-LANMAR proactively maintains the membership information. This removes the setup latency (up to seconds) to build a multicast mesh or tree that may be unacceptable in the real time coordination and control of a mission (for example, multiple sensor beamforming).

Secondly, M-LANMAR can achieve the better reliability over traditional multicast protocol using broadcast by simply using a robust, unicast MAC layer, and by running TCP-like protocol on the tunnel from source to Landmark. The final distribution of the multicast packet within the team is very reliable as it uses local scoped "flooding".

The last, but not least benefit of M-LANMAR is the protection against congestion. M-LANMAR congestion can be controlled in various different ways. One way is to use TCP-like protocol. The TCP-like congestion control window automatically guarantees congestion protection.

The main contributions are expected from this research: (1) use of the swarms for quick establishment of connectivity among "islands of sensors" embedded in the system – namely, methodology for maintaining connectivity with the minimum number of UAVs/UGVs located in strategic positions; (2) development of a team-multicast protocol, M-LANMAR, which is suitable for our target scenarios with UAVs/UGVs; (3) the implementation and evaluation of M-LANMAR

The rest of our paper is organized as follows. We will present background in the next section and our protocol M-LANMAR in Section III. And, simulation study will follow. Finally, we conclude our paper in Section 5.

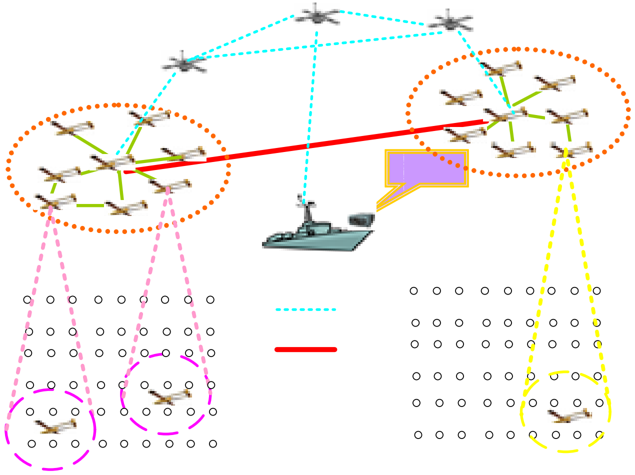


Figure 1. A proposed communication architecture

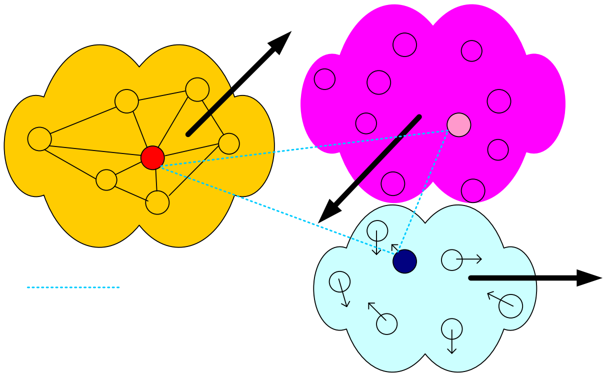


Figure 2. The overview of swarm network

2. Background

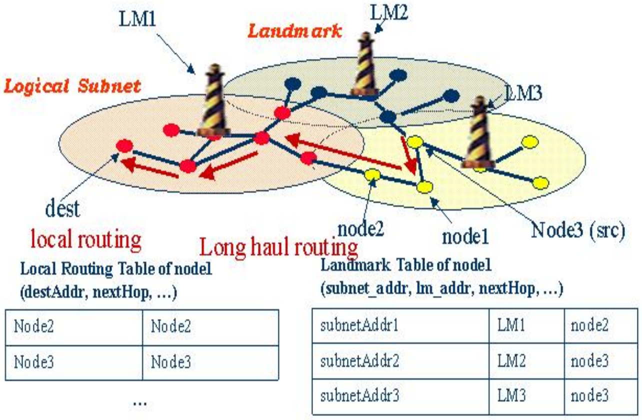
2.1. Communications architecture

The overview of our target system is shown in Figure 1. And the overview of team communication is illustrated in Figure 2. The network consists of networked islands of ground sensor networks, several swarms and backbone nodes which connect swarms. A ground sensor network consists of a number of small sensors (e.g., environment monitoring sensors, chemical detecting sensors, motion detecting sensors, etc). In our target system, each UV is able to communicate with a ground sensor node by deploying the same radio. As illustrated in Figure 1, we assume that radio connectivity between the swarms is provided by some form of mobile wireless backbone network infrastructure. This infrastructure may consist of rovers on the ground, or low orbiting satellites, or UAVs (unmanned airborne Vehicles). Different frequencies are generally used in the backbone (among backbone nodes) and in the local access networks (swarm to backbone, inter-swarm and intra-swarm) to improve performance and to simplify scheduling. Moreover, backbone links may be implemented with phase array, beam forming antennas to increase the range and bandwidth. In the paper, we assume that there is at least one virtual link between two swarms that perform the same mission, i.e., belong to the same multicast group. A swarm S is a connected un-directed graph with the maximum distance D from a node i to j (i and $j \in$ S). A link (i, j) implies a direct connection between i and j. A swarm S is defined as a set of nodes satisfying following conditions:

- **Same mobility pattern**: Our affinity team model, where nodes in the same team move together allowing the limited individual mobility of each node, assumes that the mobility of a swarm can be specified as a vector with velocity and direction as shown in Figure 2.
- **Performing the common mission (common multicast group membership**
- **D-hop reachability**: Any two nodes in the team are mutually reachable within D-hop distance.

This system requires four communication types as follows:
(a) Communications between swarms: as multiple swarms perform the complex common mission (e.g., find the hot spot where hazardous chemicals are distributed in a large terrain), data transmissions between swarms are necessary to co-work.
(b) Communications within a swarm: a UV sends collected data/events to the leader of the swarm. The leader, then, aggregates events from all UVs in the swarm and transmits to other swarms or the command post.
(c) Communications with the command post: the command post delivers the command. And the leaders of swarms report collected information back to the command post.
(d) Communications with sensor fields deployed in the environment (e.g., ground or sea): each UV communicates with ground sensors to collect the information/events.

In the paper, we focus on communications (a), (b) and (c), called *team multicast*. The communication type (d) is out of the scope of this paper.



Node2	Node2
Node3	Node3

subnetAddr1	LM1	node2
subnetAddr2	LM2	node3
subnetAddr3	LM3	node3

Figure 3. A overview of LANMAR protocol

3. Team Multicast

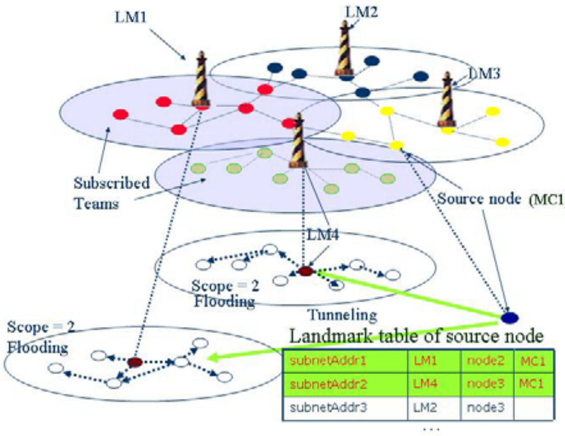
The brute force approach to team multicast among swarms is to break up the teams into individual UVs and to attack the problem as a conventional multicast problem in a multicast group consisting of potentially hundreds of mobile UVs. The advantage of this approach is the ability to use existing ad hoc multicast schemes (e.g., ODMRP [8], MAODV [10], etc). A key limitation of these methods, though, is their limited scalability to network and membership size, and to mobility. In our targeting applications (e.g., battlefield, terrorist attack, search and rescue, disaster relief, etc), large network size and high UV mobility are important factors.

In this section, we introduce a scalable approach called M-LANMAR (Multicast-Enabled LANMAR) to team multicast. The foundation of M-LANMAR is Landmark Ad hoc Routing (LANMAR) protocol [4, 5]. LANMAR provides an efficient proactive routing platform which efficiently exploits affinity team mobility.

3.1. LANMAR Protocol

LANMAR (Landmark Ad Hoc Routing) protocol is a proactive routing [4, 5]. It uses the notion of *landmarks* to keep track of logical subnets. Such a logical subnet consists of nodes that have a common interest and move together as a “group”. A representative of the subnet, i.e., a “landmark"” node, is dynamically elected in each subnet. LANMAR protocol is supported by two complementary, cooperating routing schemes: (a) a local, “myopic” proactive routing scheme using fisheye [6] operating within a limited scope centered at each node and exchanging route information about nodes up to only a few hops; and (b) a "long haul" distance vector routing scheme that propagates the elected landmark of each subnet and the path to it into the whole network. As a result, each node maintains two routing tables: local routing table and landmark table which maintain direct routes to near by destinations and routes to all the landmarks from all the subnets respectively (see Figure 3). With local routing table, nodes will learn how many nodes in the same subnet are reachable from this node (i.e., can find route entry in local routing table). A node who knows more than threshold (say *N*) such neighbor nodes proclaims as a landmark for this subnet and broadcasts to the neighbors. When more than one nodes declare themselves as landmarks in the same group, the node with the largest number of group members wins the election. In case of tie, the lowest ID rules the election.

To send or relay a given packet, a node first queries a route (i.e., next hop ID) to the destination in its local routing table. With any available path, the packet will be directly forwarded to the next hop. Otherwise, this packet will be instead routed towards the publicized landmark in the same logical subnet to the destination. Note that the subnet address of the destination is carried in the packet header.



subnetAddr1	LM1	node2	MC1
subnetAddr2	LM4	node3	MC1
subnetAddr3	LM2	node3	

...

Figure 4. A overview of M-LANMAR

3.2. M-LANMAR Protocol

M-LANMAR (Multicast-enabled Landmark Ad hoc Routing) [7] extends LANMAR enabling multicast function. M-LNAMR protocol is a proactive scheme, where group membership and multicast routes are updated proactively. With the aid of LANMAR, the sources maintain the multicast routes to only landmarks of joined teams instead of individual paths to each member.

Join Multicast Group

In LANMAR, each node keeps fresh routes to all landmarks in the network by periodic landmark updates. Using the landmark updates, a team maintains its membership to multicast group(s). A landmark of a team that wishes to join the multicast group(s) *implicitly* advertises "Join Request" to the sources by piggybacking the targeting multicast group ID(s) (address(es)) on landmark broadcast packet. Upon receiving the "implied" Join Request, each node in the network updates respective landmark entry with the subscribed multicast group IDs. Thus, the Join Request will be propagated into the sources in a few landmark table exchanges. Membership is constantly refreshed, as each landmark includes subscribed multicast addresses to all outgoing landmark update packets.

Leave Multicast Group

When a team who is a part of multicast group wants to leave, the landmark removes the ID of that multicast group from its subscribed multicast groups list. Thus, the landmark will stop advertising the group. The landmark's entry at other nodes will be updated accordingly.

Data Propagation

The source nodes look up their landmark table to find the landmark addresses of the subscribed teams. For each landmark that subscribes to this multicast group, the source creates a "virtual link", i.e., a tunnel, to the landmark and sends encapsulated multicast data. Upon reception of the encapsulated data, each landmark initiates flooding within the subnet so that each member can receive the data (see Figure 4). With an assumption of restricted size of the subnet ("x" hops from the landmark to all nodes), we use local flooding with initial TTL "x+1" (in our simulation x =2). Each node in the team accepts incoming multicast data.

4. Simulation Study

We have designed a series of simulation experiments to evaluate the M-LANMAR solution in the swarm multicast scenario and to compare it to traditional "flat" (i.e., non team aware) schemes. In our simulation study, we consider only the network of UV teams to evaluate the performance of M-LANMAR. Throughout our experiments we use the QualNet [3] simulation tool developed by Scalable Network Technologies. In this study we use the IEEE 802.11 DCF MAC layer protocol, with channel speed set to 2Mbps. The channel propagation model is a two-ray path-loss model.

We begin by comparing M-LANMAR with ODMRP and with flooding (the latter being the most reliable scheme in a lightly loaded, mobile network). In the network, 1000 UVs are uniformly placed within 6000 x 6000 meter square field and grouped into 36 swarms. The transmit radio is 376 meters such that on average each UV has 10 neighbors. The average distance of each swarm is 2. In order to maintain connectivity in the face of mobility, ODMRP is refreshed every 2 seconds; M-LANMAR uses 1 second intervals for landmark updates and 2.3 seconds intervals for local routing table exchanges.

Our first experiment is based on a mobile network scenario. Namely, the nodes in the same team move following the "Reference Point Group Mobility" model [11] with speed 2m/s with 10s pause time. According to this model, each team collectively moves following a reference vector (speed, direction) randomly reselected after each pause. Moreover, each node in the team randomly moves around the reference vector. For this experiment, three randomly selected teams form each multicast group. The number of multicast groups ranges from 1 to 9. The performance measure is the delivery ratio (i.e., ratio of delivered packets/ total expected packets) and normalized control overhead (i.e., the average number of generated control packets to deliver one data packet) of the three protocols. Each multicast group has a source with data rate 2048bytes/sec.

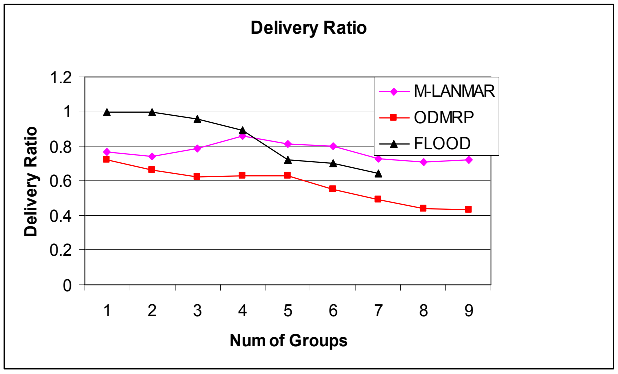


Figure 5. Delivery ratio

Figure 4 shows that in presence of mobility M-LANMAR consistently performs better than ODMRP. Remarkably, for all scenarios, M-LANMAR shows a nearly constant delivery ratio regardless of the given offered load. ODMRP on the other hand suffers from heavy contention and collision due to the control packets illustrated in Figure 6 (the control overhead is much higher in this experiment because of mobility). Flooding also takes a hit in the mobile environment as the number of group increases. Our analysis shows that the flooding delivery ratio eventually drops below M-LANMAR due to too much overhead and congestion. We could not even complete the flooding experiments with more than 8 multicast groups due to the high memory requirements.

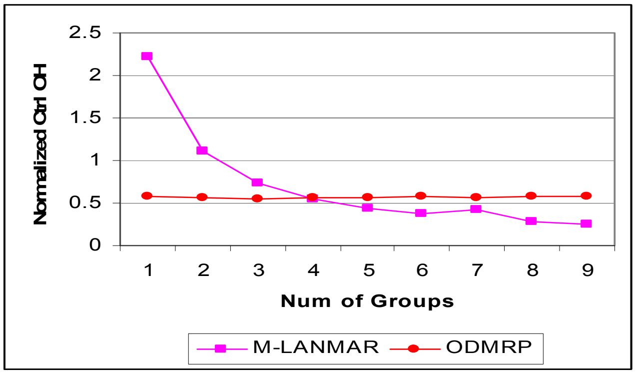


Figure 6. Normalized Control Overhead

So far, we show that the hierarchical structure of M-LANMAR manages to keep congestion in check for increasing number of multicast groups (and therefore offered load) even in the mobility scenario. We now address the issue of **reliable** multicast. Reliable delivery is essential in some of the swarm based applications, for example, the remote control of swarms via commands/controls from a manned command post; the coordination of path plans among swarms in the battle theater, etc. Naturally, reliable multicast subsumes congestion control (as reliable delivery can be achieved only in a congestion free network). However, we now demand 100% delivery ratio as opposed to being content with 80% delivery ratio (as exhibited in Figure 5 above).

In the next set of experiments, we consider the same (mobile) scenario as in Figure 5. In the new set of experiments, we first run M-LANMAR as before, with a simple UDP transport protocol on top. Then, we repeat the experiments with the reliable "multicast transport" protocol RALM [9]. RALM mimics the TCP functionality, with suppression of NACKs (to avoid implosion). Congestion is controlled by using stop and go operation during loss recovery [9]. The results are shown in Figure 7. When RALM is introduced, the M-LANMAR delivery ratio performance jumps up to 100%. RALM suppression of NACKs works well with the Landmark construct.

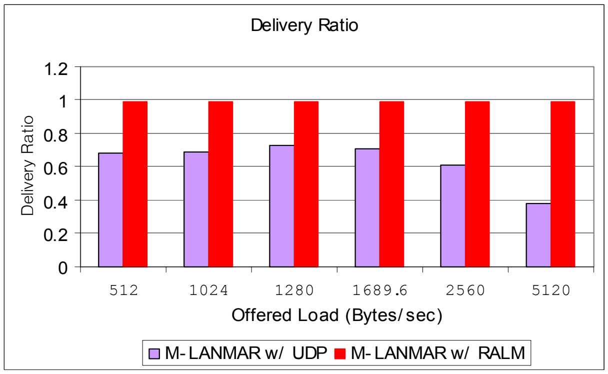


Figure 7. Delivery ratio of M-LANMAR and ODMRP with RALM

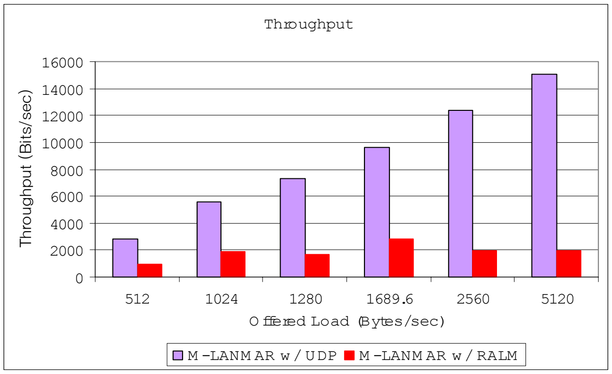


Figure 8. The throughput of M-LANMAR and ODMRP with RALM

There is a price, however, to be paid when 100% delivery ratio is requested – namely, throughput efficiency. In Figure 8 we report overall throughput. We note that the throughput rate achieved by RALM is much lower than the rate measured under UDP. This is expected as the congestion controlled operation of RALM considerably slows down throughput.

5. Conclusion

We have shown that the team multicast model is very appropriate for managing the communication requirements among teams of moving agents equipped with a variety of different sensors. The proposed M-LANMAR solution is efficient and provides low latency and overhead. We have demonstrated the performance advantages of M-LANMAR over a traditional multicast protocol (ODMRP) in representative network and traffic environments. Furthermore, we applied a reliable multicast protocol, RALM, to M-LANMAR as a simple extension of M-LANMAR. Future work will address more aggressive mobility scenarios and will study specific traffic patterns and performance constraints derived from representative swarm applications.

Reference

[1] Mario Gerla, Kaixin Xu, and Allen Moshfegh "Minuteman: Forward Projection of Unmanned Agents Using the Airborne Internet", *IEEE Aerospace Conference 2002, Big Sky, MT, Mar. 2002*

[2] The First IEEE Workshop on Disaster Recovery Networks, DIREN 2002

[3] Scalable Solution Networks, QUALNET, http://www.qualnet.com

[4] M. Gerla, X. Hong, G. Pei, "Landmark Routing for Large Ad Hoc Wireless Networks", *Proceedings of IEEE GLOBECOM 2000, San Francisco, CA, Nov. 2000.*

[5] G. Pei, M. Gerla and X. Hong , "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility", *Proceedings of IEEE/ACM MobiHOC 2000, Boston, MA, Aug. 2000.*

[6] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks", *Proceedings of ICC 2000, New Orleans, LA, Jun. 2000.*

[7] Yunjung Yi, Xiaoyan Hong and Mario Gerla, "Scalable Team Multicast in Wireless Ad hoc Networks Exploiting Coordinated Motion", *NGC 2002*

[8] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks", *Internet Draft, draft-ietf-manet-odmrp-02.txt, Jan. 2000.*

[9] Yunjung Yi, Ken Tang, Mario Gerla, Sung-Ju Lee, Venkatesh Rajendran, and Katia Obraczka, "Achieving Reliable Multicast in Ad Hoc Networks with Congestion Control and Local Recovery", *UCLR Report*

[10] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," *Proceedings of ACM/IEEE MOBICOM, 1999*

[11] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, A Group Mobility Model for Ad Hoc Wireless Networks", *Proceedings of ACM/IEEE MSWiM'99,* Seattle, WA, Aug. 1999.