

# OGHAM : On-demand Global Hosts for Ad-hoc Multicast using Minimum Distance Facility Location

Chia-cheng Hu

Department of Computer Science and  
Information Engineering  
National Taiwan University  
Taipei, Taiwan  
jjhwu@inrg.csie.ntu.edu.tw

Eric Hsiao-kuang Wu

Department of Computer Science and  
Information Engineering  
National Central University  
ChungLi, Taiwan  
hsiao@csie.ncu.edu.tw

Gen-Huey Chen

Department of Computer Science and  
Information Engineering  
National Taiwan University  
Taipei, Taiwan  
ghchen@csie.ntu.edu.tw

**Abstract**—Recent routing protocols and multicast protocols in ad-hoc networks adopt two-tier architecture to accommodate the effectiveness of the flooding scheme and the efficiency of the tree-based scheme. Some hosts with maximum neighbor degree are chosen as RPs (rendezvous point) to forward data. However, these hosts will have higher possibility to be the traffic concentration and bottleneck of the network, and RPs will spend more time forwarding data due to maximum neighbor degree. In this paper, we propose a multicast protocol for Ad-hoc network, called OGHAM, with shorter relay and less concentration via selecting RPs from the hosts with minimum hop distance between them and from other normal hosts to these selected RPs rather than the hosts with maximum neighbor degree.

**Keywords**—ad-hoc; multicast; on-demand; global; linear programming; minimum distance facility location

## I. Introduction

In ad-hoc network, hosts might work in groups to carry out a common task. Several multicast protocols have been recently proposed for ad-hoc networks. Proposed multicast routing can be classified into tree-based protocols and mesh-based protocols. LAM[1], CBT-FR[2] are tree-based protocols. These tree-based protocols suffer from disadvantages of traffic concentration and vulnerability of the core. ODMRP[3] and CAMP[4] are mesh-based multicast protocols that provide richer connectivity among multicast members. Tree-based protocols are generally more efficient in terms of data transmission than mesh-based protocols, but they are less robust against topology changes. Mesh-based multicast protocols flood data packets over all links and provide alternative paths. To the extreme, a multicast packet will be forwarded to every node as long as the network is not partitioned. Furthermore, to join a group for a member involves transmitting a join request message to the existing multicast group tree. This process usually ends up with flooding a large portion of or even the entire network.

Recently, two-tier architecture is adopted to avoid this problem by introducing the notion of rendezvous points (RP) or cores to manage the requests (such as join requests) to be sent toward the core instead of being flooded to the entire network. In ADB [6], it is based on two-tier architecture with the help of an underlying virtual backbone infrastructure.

ADB relaxes the dominating set property by allowing a host to attach to the backbone from a local group. MZR[11], for each host, proposes a pre-configured zone radius which will be fixed for the rest of the network operations, while the other existing proposals, based on the dominating set property, always require each non-backbone host to be at most one hop away from the nearest backbone host. We observe the following facts:

1. Existing two-tier multicast protocols [4][6][11] select some hosts with maximum neighbor degree as RPs to forward data. However, these hosts will have higher possibility to be the traffic concentrations and bottlenecks of the network, and individual RPs will suffer more time from forwarding data due to maximum neighbor degree.
2. The amount of transmitting data among RPs is much more than the data between normal hosts and RPs (due to the data forwarding and the topology maintenance).

Therefore, we propose a multicast protocol, called OGHAM, for ad-hoc network. OGHAM also adopts two-tier architecture to manage the requests. We on-demand select hosts with minimum hop distance between them and from other normal hosts (not RP) to these selected hosts rather than hosts with maximum neighbor. The objectives of the paper are not only to accommodate the flooding effectiveness and tree-based path efficiency, but also to increase robustness in dynamic environment. Our approach is based on linear program rounding to solve this problem called Minimum Distance Facility Location (MDFL), and a novel rounding method is used to obtain the solution.

## II. Minimum distance facility location problem

The recent trend for existing routing protocols and multicast protocols in ad-hoc networks adopts two-tier architecture to control the requests to be sent toward RPs instead of being flooded to the entire network. They aim to extract a dominating set out of the network. However, we observe the new challenge: The quantity of transmitting data between RPs is significantly more than the quantity of transmitting data between normal hosts and RPs for data forwarding and topology maintenance. Therefore, this paper addresses the following challenges:

1. To avoid flooding to the entire network,
2. To lower the traffic concentrations and bottlenecks of the network
3. To minimize multicast relay (the hop distance from sender to receiver)

The proposed strategy is to on-demand select some hosts with minimum hop distance between them as RPs to forward data, but each normal host (not RP) in this region can attach to a RP in  $2r$  hop distance. We denote this problem as MDFL problem.

#### A. Previous work

The MDFL problem is similar to the facility location problem[8], K-MEDIAN[8] problem and K-CENTER [9] problem to find some facilities to provide services for users. They are classic NP-complete optimization problems [7]. However, known techniques for these problems fail to satisfy the challenge of our problem.

#### B. Problem statement

Given an undirected graph  $G=(V, E)$  with a hop distance function  $D: E \rightarrow R^+$  on the edges.

$D(i, j)$ : indicates the hop distance from host  $V_i$  to  $V_j$ , and the distance is symmetric  $D(i, j)=D(j, i)$

$F(i)$ : indicates whether host  $V_i$  is opened as RP

$X(i, j)$ : indicates whether  $V_i$  attaches to RP  $V_j$

$N$ : indicates the number of hosts

$D_i = \sum_j D(i, j)$ : sum the distance from host  $V_i$  to each  $V_j$

We choose a particular subset of all hosts to be as RPs, and each normal host is assigned to corresponding RPs within a hop distance  $r$  ( $2r < R$ ), and the total hop distance (from RPs to RPs and from normal hosts to RPs) is minimized.

$$\sum_{F(i)=1} \sum_{F(j)=1} \frac{1}{2} (F(i) + F(j)) * D(i, j) \quad (1)$$

$$\sum_{F(i)=1} \sum_{F(j)=0} (F(i) + F(j)) * D(i, j) \quad (2)$$

The equation (1) is to sum the distance between RPs, and the equation (2) is to sum the distance from normal hosts to RPs. Our objective function by adding (1) and (2) as following:

$$\sum_{F(i)=0} \sum_{F(j)=1} \frac{1}{2} (F(i) + F(j)) * D(i, j) + \sum_{F(i)=1} \sum_{F(j)=0} F(i) * D(i, j) \quad (3)$$

Since  $D(i, j)$  is symmetric. Therefore equation (2) becomes

$$(2) = \sum_{F(i)=0} \sum_{F(j)=1} \frac{1}{2} (F(i) + F(j)) * D(i, j) + \sum_{F(i)=1} \sum_{F(j)=0} \frac{1}{2} (F(i) + F(j)) * D(i, j) \quad (4)$$

Further, we can sum the distance between normal hosts  $i$  ( $F(i)=0$ ) as following

$$\sum_{F(i)=0} \sum_{F(j)=0} \frac{1}{2} (F(i) + F(j)) * D(i, j) = 0 \quad (5)$$

Using (4)(5), we can write our objective function (3) as

$$(3) = \sum_{F(i)=1} \sum_{F(j)=1} \frac{1}{2} (F(i) + F(j)) * D(i, j) + \sum_{F(i)=0} \sum_{F(j)=1} \frac{1}{2} (F(i) + F(j)) * D(i, j) + \sum_{F(i)=1} \sum_{F(j)=0} \frac{1}{2} (F(i) + F(j)) * D(i, j) + \sum_{F(i)=0} \sum_{F(j)=0} \frac{1}{2} (F(i) + F(j)) * D(i, j)$$

$$= \sum_i \sum_j \frac{1}{2} (F(i) + F(j)) * D(i, j) = \frac{1}{2} \sum_i F(i) \sum_j D(i, j) + \frac{1}{2} \sum_j F(j) \sum_i D(i, j) \quad (6)$$

Finally, we can derive our objective function by replacing  $\sum_j D(i, j)$  with  $D_i$  and  $\sum_i D(i, j)$  with  $D_j$ .

$$(6) = \frac{1}{2} \sum_i F(i) D_i + \frac{1}{2} \sum_j F(j) D_j = \sum_i F(i) D_i \quad (7)$$

The minimum problem is expressed by the following integer linear program (ILP).

*Objective function:*

$$\text{Minimize } \sum_i F(i) D_i \quad (8)$$

*Constraints:*

$$F(i) - X(u, i) \geq 0, 1 \leq u, i \leq N \quad (9)$$

$$\sum_i X(u, i) = 1, 1 \leq u \leq N \quad (10)$$

$$X(u, i) = 0 \text{ if } (D(u, i) > r) \quad (11)$$

$$X(u, i) \in \{0, 1\}, 1 \leq u, i \leq N \quad (12)$$

$$F(i) \in \{0, 1\}, 1 \leq i \leq N \quad (13)$$

In this problem, the constraint (9) ensures that each normal host is assigned to an open RP, and the constraint (10) ensures that each normal host is assigned to one RP. We approach to minimize the total hop distance by solving the above ILP to get the indicator variables  $F(i)$ ,  $X(i, j)$ .

The linear relaxation of this program replaces the last two constraints with  $0 \leq X(i, j) \leq 1$  and  $0 \leq F(j) \leq 1$ .

#### C. Approximation algorithm

We use the following algorithm to produce an approximation solution.

1. Solve the Linear Program relaxation
2. Set  $I = \{i: 0 < F(i) < 1\}$ .  $I$  is the set of fractional values for individual RPs.
3. Select  $i \in I$  which minimizes  $(1 - F(i)) * D_i$
4. Set  $J = \{j: 0 < F(j) < 1, D(i, j) \leq r\}$ ,  $J$  is the set of fractional values for individual RPs near  $i$
5. Compute  $\Delta_2 = \sum_{j \in J} F(j) * D_j$ , set  $F(j) = 0$  for all  $j \in J$ , and set  $X(u, i) = X(u, i) + X(u, j)$ , then set  $X(u, j) = 0$  for all host  $u$
6. Compute  $\Delta_1 = (1 - F(i)) * D_i$ , set  $F(i) = 1$
7. If  $\Delta_1 - \Delta_2 > 0$  and  $I - \{i\}$  is not NULL, find a fractional number  $t$ ,  $F(k) = t * F(k)$  for all  $k \in I - \{i\}$  and the total increasing distance  $\Delta = \Delta_1 - \Delta_2$
8. If there still exist fractional  $F(j)$ , return to step 2
9. For each host  $u$ , select minimum  $D(u, i)$  where  $F(i) = 1$  and  $X(u, i) > 0$ , set  $X(u, i) = 1$  and  $X(u, j) = 0$  for other RP  $i \neq j$
10. For all RP  $i$ , set  $F(i) = 0$  if  $\sum_{u=1}^N X(u, i) = 0$

We solve the linear program relaxation to get a fractional minimum solution in step 1 by the LP-duality theorem using complementary slackness conditions [10]. In step 3, we select a RP  $i$  from set  $I$  with minimum increasing distance (the increasing amount of objective function by setting  $F(i) = 1$ ). The selected RP  $i$  is assigned to be a real RP by setting  $F(i) = 1$  and those nearby RPs  $j$  with fractional values in set  $J$  are to be normal hosts by setting  $F(j) = 0$ . Furthermore, the hosts

originally assigned to nearby RP  $j$  will be re-assigned to RP  $i$ . In step 5 and 6, if  $\Delta_1 - \Delta_2 > 0$ , the total distance will increase. In this case, we just use a fractional number  $t$  small enough to decrease  $F(j)$  if  $F(j) < 1$ , such that the total distance is still smaller than the minimum solution.

We can solve a linear program in polynomial time and the number of loop iterations is linear with the number of possible RPs. In each iteration, we set a RP with fractional  $F(j)$  to be 1 and some RPs with fractional  $F(j)$  to be 0. The integer solution is still obtained within polynomial time.

#### D. Analysis

Clearly when the algorithm terminates, all variables will be integers. We first prove that the hop distance bound is approximately maintained; no host is assigned to a RP in at most  $2r$  hop distance.

**Theorem 1:** The host in multicast region can attach to a RP in at most  $2r$  hop distance.

**Proof:** We claim that at every stage of the algorithm, if  $F(i) < 1$  and  $X(u,i) > 0$  then  $D(u,i) \leq r$ . This is clearly true for the fractional linear program solution. Suppose at some point it becomes untrue. This restricts us to consider stages 5 and 9. In stage 5, we increase the value of  $X(u,i)$ ; but the equal value for other RP  $j$  in set  $J$  also decrease. In stage 9, each host  $i$  selects a RP  $j$  ( $F(j)=1, D(i,j) \leq r$ ). The claim still holds. On the other hand, we may set some host  $u$  must have  $D(u,i) > r$  in stage 6. However, any such  $u$  must have  $X(u,j) > 0$  for some nearly candidate RP  $j$  ( $0 < F(j) < 1$ ) in  $J$ . Since  $j$  in  $J$ , we know that  $F(j) < 1$  and therefore  $D(u,j) \leq r$ . We also know that  $D(i,j) \leq r$ . It follows that  $D(u,i) \leq 2r$  which proves the claim.

**Theorem 2:** The total distance after rounding is at most the minimum distance + max  $D_i$

**Proof:** In stage 3, the total distance may increase due to set some  $F(i)=1$ . In stage 7, we find a fractional number  $t$  to reduce the total distance for keeping the minimum solution. The fractional value  $k$  is small enough such that the objective function is at least smaller than the original minimum solution. When in the last iteration, there is a candidate RP  $i$  with fractional value  $F(i)$ . In the worst case,  $F(i)$  is close to 0. The increasing distance for objective function is  $(1-F(i))*D_i = D_i$  by setting  $F(i)=1$ . It follows that the total distance after rounding is at most the minimum distance + max  $D_i$ .

In this section, we select some hosts as RPs to control flooding and to lower the concentrations. According to theorem 1 and 2, the total hop distance is minimum and each host can attach to a RP in at most  $2r$  hop distance. This leads to the minimum multicast relay.

### III. OGHAM

#### A. OGHAM protocol

##### 1) Construction of a multicast region

When a host  $u$  can't find a RP in  $2r$  hop distance, it will construct a multicast region to select some hosts within radius  $R$  hop distance region. The algorithm consists of the following steps:

1. Host  $u$  broadcasts to all hosts  $v$  within radius  $R$  hop distance region
2. All hosts  $v$  return its neighbor information  $N_R(v)$  to host  $u$ .
3. Host  $u$  solves the minimum distance location problem (as in section II), then broadcasts the result to these hosts in this region
4. Each RP saves the result on its local copy.

We illustrate these steps on the example in Figure 1.  $S_i$  is the sender of group  $i$ , and  $R_{ij}$  is the receiver of group  $i$ , and  $j$  is the sequence number to join group  $i$ . Receiver hosts  $R_{11}$ ,  $R_{12}$ ,  $R_{13}$  join group 1 from their nearest RPs after multicast REGION 1 creation by sender host  $S_1$ .

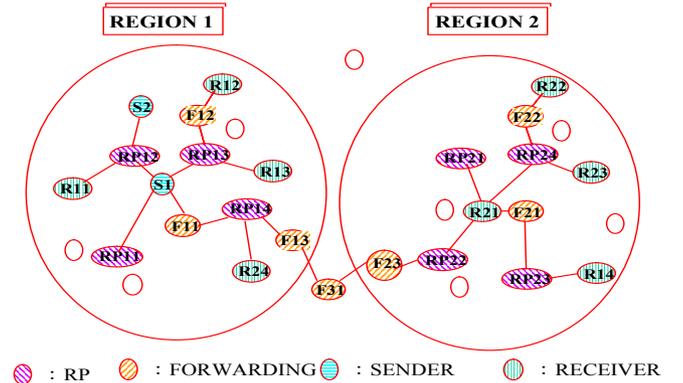


Figure 1: Construction and gluing of multicast regions

##### 2) Gluing of multiple multicast regions

Whenever a receiver can not find the sender from the nearest RP, this RP broadcasts a message (RPs information and group information) to other region's RPs for querying the sender information. If some RP in other region receives the message, then it returns its RPs information and group information to the original RP. These regions could be glued together by exchanging their relative information. The original RP may receive multiple return messages from multiple RPs in the same region, but selects the shortest path to the target RP attached by sender and saves all return paths on its local copy. It also broadcasts the received messages to all RPs in local region. In figure 1, RP14 receives message from RP21 and RP22, selects RP22, and broadcasts to RP11, RP12, and RP13. Then, R14 can join group 1 after gluing region 1 and region 2.

It is unavoidable for a RP to find another RPs in other separate multicast regions by broadcasting a message. But, OGHAM is aimed to avoid flooding inefficiency by just one-time broadcast to glue regions. The receivers in glued regions is unnecessary to broadcast again, they can get sender's information by querying their RPs.

##### 3) Construction of a multicast path

The forwarding hosts keeping the upstream and downstream information are the interior hosts to forward data between active RPs and active hosts (senders and receivers). In figure 1, F11, F12, F13, F21, F22, F23, and F31 are forwarding hosts. A multicast path from sender to receiver is constructed as follows:

1. Receiver joins a group by attaching to the nearest RP  $r$ .

The RP  $r$  selects the shortest path to the RP  $s$  attached by server from its local copy

2. The RP  $s$  selects the shortest path to the RP  $r$  for forwarding multicast data from sender to receiver.

In figure 1, the path from  $S1$  to  $R12$  is  $S1$ - $RP13$ - $F12$ - $R12$ , the path from  $S2$  to  $R21$  is  $S2$ - $RP12$ - $S1$ - $F11$ - $RP14$ - $F13$ - $F31$ - $F23$ - $RP22$ - $R21$ .

#### 4) Finding of sender

Each RP maintains the following information:

1. The path to other connected RPs. (including gluing RPs)
2. The group information (group ID and RPs attached by sender)

When sender is located in local region, receiver can join the group from the nearest RP. The nearest RP will select the shortest path to the RP attached by sender. If sender is located in remote region, the receiver will join the group after regions gluing.

#### 5) Handling of moving hosts

1. Sender: If RP or forwarding host detects the path to sender is broken, it prunes off downstream link and notifies all connected RPs. When the sender rejoins the topology, the new RP attached by the sender notifies other connected RPs to prune old paths and create new paths.
2. Receiver: If a RP or a forwarding host detects the path to the receiver is broken, it prunes off the upstream link. When the receiver rejoins the topology, the new RP creates a new path to sender.
3. New coming receiver: If a new coming receiver can find a RP in  $2r$  hop distance, it can join group by attaching to the nearest RP. Otherwise, it broadcasts the message to  $R$  hop distance to create a new multicast region.
4. RP and forwarding host: if a sender or a receiver detects the multicast path is broken, its algorithm is the same as new coming host to re-attach to a RP or to create a new multicast region.

#### 6) Robustness and Maintenance

If RP or forwarding host detects a broken active multicast path, it notifies the source RP. The source RP selects another path as following: If the target RP is also in region, the source RP selects another path from its local copy. By the above scheme, we have high possibility to find another path to replace the broken path without flooding again. When all possible paths are all broken, we assume the multicast regions are partitioned, we run the glue processing. If we still can't find the target RP, we conclude the network is partitioned.

Due to the movement of hosts, the topology will change dynamically and the efficiency of the selected RPs will degrade gradually. The distance between selected RPs and from normal hosts to relative RPs is not optimal. This problem will be solved when a host can't find a RP in  $2r$  hop distance. There are two conditions that a host can't attach to a RP in  $2r$  hop distance.

1. The host is located at an area without multicast region topology, and it can create a new one.
2. The topology of the multicast region for the host is changed, and it can re-submit the construction protocol to re-optimize the topology in its radius  $R$  hop distance

region.

#### 7) Handover latency of re-joining

When a normal host disconnects with its RP, it will have high possibility to re-join the group by attaching to another RP in  $2r$  hop distance. Otherwise, it will create a new multicast region. New RP will select a new path for this host.

#### 8) Relocation of RP

When a new region is created, it may overlap with other existing regions. The hosts overlapped by multiple regions prune off the old RPs and attaches to the new RP with the shortest path away from it.

## IV. Analysis and comparison

### A. Performance analysis

We discuss the performance of the proposed protocol.

Theorem 3: To construct a multicast region has message complexity  $O(3n-2lh-1)$  and time complexity  $O((3n-2lh-1)*D)$ , where  $D$  is the maximum neighbor degree,  $n$  is the number of hosts in this region,  $lh$  is the number of leaf hosts in this region.

Proof: According to section III.C.1, the number of broadcast messages is  $n-lh$  in step 1. In step 2, the number is  $n-1$ . In step 3, the number is  $n-lh$ . Thus the total number of broadcast messages is  $(n-lh)+(n-1)+(n-lh)=3n-2lh-1=O(n)$ .

Each broadcast takes time  $O(D)$ . We assume that the running time is dominated by message passing instead of CPU processing. Thus the time complexity is  $O((3n-2lh-1)*D)$ .

Theorem 4: To attach to a RP has message complexity at most  $O(3n-2lh-1+2r)$  or  $O(2r)$  and time complexity  $O((3n-2lh-1+2r)*D)$  or  $O(2r*D)$ .

Proof:

Case 1: The host is not located in a multicast region.

According to theorem 1 and 3, the host attach to a RP has message complexity  $O(3n-2lh-1+2r)$  and time complexity  $O((3n-2lh-1+2r)*D)$ .

Case 2: The host is located in a multicast region.

The host can attach to a RP in at most  $2r$  hop distance. To attach to a RP has message complexity  $O(2r)$  and time complexity  $O(2r*D)$ .

### B. Comparison

Furthermore, we compare OGHAM with MZR and ADB. The detail is depicted in table 1.

#### 1) MZR

In the MZR, each host in the network defines a zone around itself with zone radius hops. The zone routing table is kept current by running a proactive protocol inside each zone. The difference from pure proactive protocols is that the routing updates are restricted to the zone, and are not sent to the entire network, while reactive route queries are done at zone border hosts on demand, resulting in a much smaller number of hosts participating in the global flooding searching.

OGHAM is similar to bound flooding. In the MZR, each host must maintenance the zone information around itself by periodically broadcasting to all hosts in zone. When a host wants to join a group, it broadcasts the request to zones over a large portion of or even the entire network. But in the OGHAM, the members of all group applications in connected region can join group in  $2r$  hop distance by attaching to the nearest RP, and only the RPs are responsible to maintenance the topology. Besides, the topology is constructed on-demand.

## 2) ADB

ADB creates a forest of varying-depth trees, each of which is rooted at a backbone host. The hop distance from each host to its backbone host must not exceed the parameter HOP\_THRESHOLD. Similar to other tree-based topology, the backbone will be the bottleneck and the performance will degrade when the number of the group or the number of the group members is increasing. Besides, the paths are non optimal and the backbone may be off-center. But OGHAM will obtain shorter relay and less concentration rather than ADB. Furthermore, the path between RPs is shortest in the initial stage. Even though the path is broken, the RP will select a sub-shortest path from its local copy.

## V. Conclusion and future work

The proposed protocol has following distinct advantages:

1. The region is constructed on-demand and global.
2. OGHAM obtains shorter relay and less concentration.
3. Our algorithm provides the total distance between RPs after rounding is at most the minimum distance + max  $D_i$ .
4. When a host disconnects with its RP due to moving, it will have high possibility to re-attach to another RP. That leads to short handover latency.
5. OGHAM increases the robustness by selecting another path to re-forward data without flooding again.

In future, we plan to build simulation models for

quantitative performance evaluation and then compare the performance of our protocol with others.

## Acknowledgment

This research was supported by the *Communications Software Technology* project of *Institute for Information Industry* and sponsored by *MOEA, R.O.C.*

## References

- [1] L. Ji, M.S. Corson, A Lightweight Adaptive Multicast Algorithm, Proceeding of IEEE GLOBECOM'98, Sydney, Australia, Nov. 1998, pp. 1036-1042.
- [2] Sandeep K. S. Gupta, Pradip K. Srimani, Core-Based Tree with Forwarding Regions (CBT-FR): A Protocol for Reliable Multicasting in Mobile Ad Hoc Networks, Journal of Parallel and Distributed Computing 2001.
- [3] Sung-Ju Lee, William Su, Mario Gerla, On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks, ACM/Kluwer Mobile Networks and Applications, special issue on Multipoint Communications in Wireless Mobile Networks, 2002.
- [4] J.J. Garcia-Luna-Aceves, E.L. Madruga, The Core-Assisted Mesh Protocol, IEEE Journal on Selected Areas in Communications, vol. 17, Aug. 1999, pp. 1380-1394.
- [5] T. Ballardie, P. Francis, J. Crowcroft, Core-Based Tree (CBT): An Architecture for Scalable Inter-Domain Multicast Routing, Communications, architectures, protocols, and applications (San Francisco, C.A,USA), Sep. 13-17, 1993
- [6] Chaiporn Jaikao, Chien-Chung Shen, Adaptive Backbone-Based Multicast for Ad Hoc Networks, IEEE International Conference on Communications (ICC), New York City, April 28-May 2, 2002
- [7] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York, 1979
- [8] M. Charikar, S. Guha, Improved Combinatorial Algorithm for Facility Location and K-media Problems. Proceedings of 40<sup>th</sup> IEEE FOCS, 1999
- [9] Dorit Hochbaum, David Shmoys, A Best Possible Heuristic for the K-center Problem, Math of Operations Research, 10(2):180-184, 1985.
- [10] Vijay V. Vazirani, Approximation Algorithms, Springer-Verlag Berlin Heidelberg 2001.
- [11] V. Devarapalli, D. Sidhu, MZR: A Multicast Protocol for Mobile Ad Hoc Networks, IEEE International Conference on Communications (ICC), Helsinki, Finland, Jun 2001

Table 1: The comparison of OGHAM with MZR and ADB

		OGHAM		MZR		ADB	
Bound Flooding		R		R		HOP_THRESHOLD	
Communication Overhead	Buffer the multicast routing information	RP: 1. Routing path to each connected RP. 2. Group information. Normal Host: Path to the nearest RP		All hosts: 1. The path to all zone hosts 2. The path to target		Cores: Group information. Normal Host: the path to the core	
	The hosts to route	RPs		All hosts		Cores	
	The message and time complexity to construct multicast region	construct a new region: message $O(3n-2lh-1)$ , time $O((3n-2lh-1)*D)$ The host within a multicast region: message 0, time 0		message $O((n-1)*T1)$ time $O((n-1)*T1*D)$ $T1$ : the times to execute broadcast		message $O((n-1)*T2)$ time $O((n-1)*T2*D)$ $T2$ : the times to execute broadcast	
	When to broadcast	On-demand		Period		Period	
	Who store the Group information	All RPs		All Zone hosts		All cores	
Multicast Application	The hop distance to find server	$\leq 2r$		$\leq R$		$\leq$ HOP_THRESHOLD	
Maintenance due to host movement		Exist a region and server information in nearest RP	The hop distance is $\leq 2r$	Server routing information in zone	The hop distance is $\leq R$	exist a nearest core and server information in nearest core	The hop distance is $\leq$ HOP_THRESHOLD
		Exist a region and server information not in nearest RP	To glue region	Other	Query from border hosts	Not exist a nearest core	Select a core and join the tree
		Other	Create and glue regions				