Efficient Flooding in Ad hoc Networks using On-Demand (Passive) Cluster Formation

Abstract—**Many ad hoc network protocols (e.g., routing, service discovery, etc.) use *flooding* as the basic mechanism to propagate control messages. In flooding, a node transmits a message to all of its neighbors. The neighbors in turn transmit to their neighbors and so on until the message has been propagated to the entire network. Typically, only a subset of the neighbors is required to forward the message in order to guarantee complete flooding to the entire network. If the node geographic density (i.e., the number of neighbors within a node's radio reach) is much higher than what is strictly required to maintain connectivity, one can easily see that flooding becomes inefficient because of redundant, "superfluous" forwarding. In fact, superfluous flooding increases link O/H and wireless medium congestion. In a large network, with heavy load, this extra O/H can have severe impact on performance and should be eliminated.**

Clustering and, more generally, route aggregation techniques have been proposed in the past to reduce the flooding O/H. These techniques operate in a proactive, background mode. They use explicit control packets to elect a small set of nodes (cluster-heads, gateways or, more generally, flood forwarding nodes) and restrict to such set the flood forwarding function. The problem of such proactive schemes is the constant traffic O/H introduced in the network.

In this paper, we propose a new flooding mechanism based on passive, on-demand clustering. This mechanism reduces flooding overhead without loss of network performance. Passive clustering dynamically partitions the network in clusters interconnected by gateways. Passive clustering is an "on demand" protocol. It executes only when there is user data traffic; it exploits data packets for cluster formation. Passive clustering has many advantages compared with "active" clustering and route aggregation techniques: (1) it eliminates cluster set up latency and extra control overhead (by exploiting on-going packets); (2) it uses a novel, efficient gateway selection heuristic to elect the minimum number of forwarding nodes (thus reducing superfluous flooding); (3) it reduces node power consumption by eliminating the periodic, background control packet exchange.

Simulation results show that passive clustering can reduce redundant flooding by up to 70% with negligible extra protocol overhead. Moreover, we show that passive clustering can be applied to several reactive, on-demand routing protocols (e.g., AODV, DSR and ODMRP) with substantial performance gains.

I. INTRODUCTION

Multi-hop ad hoc networks (MANETs) have recently been the subject of active research because of their unique advantages. MANETs are self-creating, self-organizing and self-administrating without deploying any kind of infrastructure. They offer special benefits and versatility for wide applications in military (e.g., battlefields, sensor networks etc.), commercial (e.g., distributed mobile computing, disaster discovery systems, etc.), and educational environments (e.g., conferences, conventions, etc.), where fixed infrastructure is not easily acquired. With the absence of pre-established infrastructure (e.g., no router, no access point, etc.), two nodes communicate with one another in a peer-to-peer fashion. Two nodes communicate directly if they are in the transmission range of each other. Otherwise, nodes can communicate via a multi-hop route with the cooperation of other nodes. To find such a multi-hop path to another nodes, each MANET node widely use *flooding* or *broadcast (e.g., hello messages)*. Many ad hoc routing protocols [10] [12] [13] [27] [28], multicast schemes [25], or service discovery programs depend on massive flooding.

In flooding, a node transmits a message to all of its neighbors. The neighbors in turn relay to their neighbors and so on until the message has been propagated to the entire network. In this paper, we call such flooding as *blind flooding*. As one can easily see, the performance of blind flooding is closely related to the average number of neighbors (neighbor degree) in the CSMA/CA network. As the neighbor degree gets higher, the blind flooding suffers from the increases of (1) redundant and superfluous packets, (2) the probability of collision, and (3) congestion of wireless medium [1]. Performance of blind flooding is severely impaired especially in large and dense networks [2].

When topology or neighborhood information is available, only a subset of neighbors is required to participate in flooding to guarantee the complete flooding. We call such flooding *efficient flooding*. The characteristics of MANETs (e.g., node mobility, the limited bandwidth and resource), however, make collecting topological information very difficult. It generally needs huge extra overhead due to the periodic message exchanges or event driven updates with optional deployment of GPS (Global Positioning System)-like system. For that reason many on-demand ad hoc routing schemes and service discovery protocols simply use blind flooding [10] [12] [25]. With periodic route table exchanges, proactive ad hoc routing schemes, unlike on-demand routing methods, can gather topological information without big extra overhead (through piggybacking topology information or learning neighbors). Thus, a few proactive ad hoc routing mechanisms proposed route aggregation methods so that the route information is propagated by only a subset of nodes in the network [27] [28].

In this paper, we propose mechanism for efficient flooding suitable for on-demand protocols based on passive clustering. We require neither the deployment of GPS-like system nor explicit periodic control messages. Our scheme has several contributions compared with previous efficient flooding mechanisms (such as multipoint relay, neighbor-coverage, etc). (1) It does not need any periodic messages. Instead, passive clustering exploits existing traffic to piggyback its small control messages. Based on passive clustering technique, it is very resource-efficient regardless of the degree of neighbor nodes or the size of network. To our knowledge, passive clustering is the only scheme that provides scalability and practicality for choosing the minimal number of forwarding nodes in the presence of dynamic topology changes. Therefore, it can be easily applied to on-demand routing schemes to improve the performance and scalability. (2) It does not have any setup latency, and it saves energy with no traffic. (3) Its maintenance is well adaptive to dynamic topology and resource availability changes.

The organization of the rest part of the paper is as follows. We will present brief related works and motivations of our work in Chapter II, and describe the detailed algorithm in chapter III. Thereafter, we demonstrate the contributions of our work through analysis and extensive simulation studies in Chapter IV and Chapter V. Finally, we conclude the paper in Chapter VI.

II. RELATED WORKS AND MOTIVATIONS

Several recent papers [1] [6] [7] [8] have addressed the problem of blind flooding and proposed solutions to provide efficient flooding. However, the problem of finding a subset of dominant forwarding nodes in MANETs is NP-complete [1]. Thus, all the work about efficient flooding has been focusing on developing efficient heuristics that select a sub-optimal dominant set with low forwarding overhead.

[1] [6] proposed several heuristics to reduce rebroadcasts. In their idea, upon receiving a flooding packet, a node decides whether it relays the packet to its neighbor or not using one of following heuristics: (1) probabilistic scheme where this node rebroadcast the packet with the randomly chosen probability; (2) counter-based scheme where this node rebroadcast if the number of received duplicate packets is less than a threshold; (3) distance-based scheme that uses the relative distance between hosts to make the decision; (4) location-based scheme based on pre-acquired location information of neighbors; (5) cluster-based scheme where only cluster heads and gateways forward again. Our approach, passive clustering, is different from those ideas in that it provides a platform of efficient flooding based on locally collected information (e.g., neighbor information, cluster states, etc.). Each node participates in flooding based on the role or state in the cluster structure instead of heuristics used in [1] [6].

Another approach of efficient flooding is to exploit topological information [6] [8] [7] [24]. With the node mobility and the absence of pre-existing infrastructure in the ad hoc network, all works, as far as we know, use the periodic *hello message* exchange method to collect topological information. Our approach is diverging from those works because passive clustering does not require periodic control messages to collect topological information. Passive clustering, instead, exploits on-going data packets to exchange cluster-related information.

[8] suggested two schemes called *self-pruning* and *dominant-pruning*. *Self-pruning* is similar to the *neighbor-coverage* scheme in [6]. With *self-pruning* scheme, each forwarding node piggybacks the list of neighbors of itself on outgoing packet. A node rebroadcasts (becomes a forwarding node) only when this node has neighbors not covered by forwarding nodes.

While *self-pruning* heuristic utilizes information of directly connected neighbors only, the *dominant-pruning* scheme extends the range of neighbor information to two-hop away neighbors. The *dominant-pruning* scheme is similar to *Multipoint Relay* scheme [7]. In *Multipoint Relay* scheme (MPR), a node periodically exchanges the list of adjacent nodes with its neighbors so that each node can collect the information of two-hop away neighbors. Each node, based on the gathered information, selects the minimal subset of forwarding neighbors, which covers all neighbors within two-hop away. Each sender piggybacks its chosen forwarding nodes (MPRNs) on the outgoing broadcast packet.

Moreover, based on topological information, many schemes have proposed the scheme to choose a dominant set [21] [22] [23]. They, still, depend on the periodic hello messages to collect topological information.

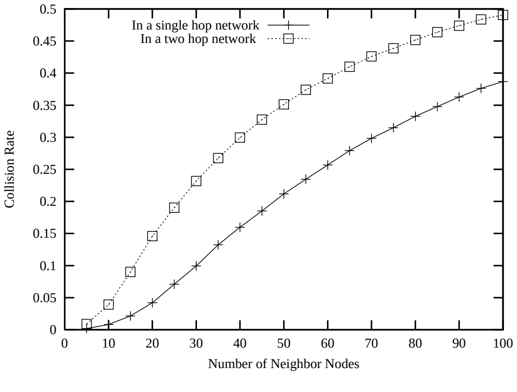

Fig. 1. The collision rate of broadcast

The extra hello messages, however, consume resources and drop the network throughput in MANETs [14]. The extra traffic brings about congestion and collision as geographic density increases [1].
Figure 1 depicts the collision probability of hello messages in a single hop and a two hop network as the number of neighbors increases. This result clearly shows that the neighbor degree increases the collision probability of broadcast (note, the collision probability is more than 0.1 with more than 15 neighbors) and hidden terminal aggravates the collision in the multihop network.
Note that Figure 1 assumes no other traffic except for hello messages in the network. With user-data packets, the collision probability of hello messages will be increased. Thus, it is very hard to collect the complete neighbor topology using hello messages.

The aforementioned schemes (e.g., neighbor-coverage, MPR, etc.), consequently, are not scalable to offered load and number of neighbors.

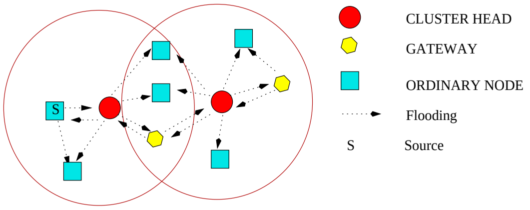

Fig. 2. An Example of Efficient Flooding with Clustering. Only cluster heads and gateways rebroadcast and ordinary nodes stop forwarding.

A. *Brief Overview of Clustering*

Clustering is another method to select forwarding nodes as addressed in [1]. Clustering in this paper can be described as *grouping nodes into clusters*. A representative of each group (cluster) is named as a *cluster head* and a node belonging to more than two clusters at the same time is called a *gateway*. Other members are called *ordinary nodes*. The transmission area of the cluster head defines a cluster. We use a 2-hop clustering where any node in a cluster can reach any other node in the same cluster with at most 2 hops as defined in [9]. With clustering, non-ordinary nodes can be the dominant forwarding nodes as in Figure 2.

Figure 3 illustrates the difference between a clustering and the MPR scheme. The clustering partitions the network into several groups based on the radio range of a cluster head. The network topology, therefore, does not have a serious impact on the clustering performance. MPR, on the other hand, chooses the dominant set using topological information so that the performance of MPR is closely related to the network topology.

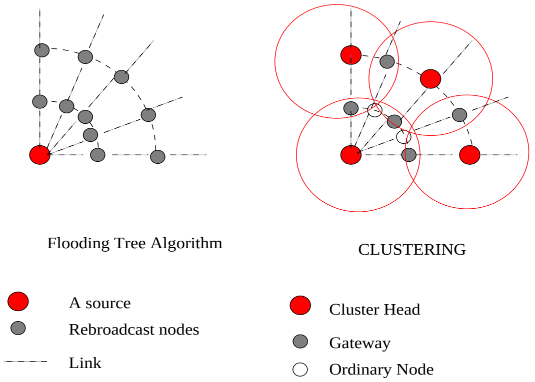

Fig. 3. In flooding tree algorithms, every neighbor of a source has to rebroadcasts since each neighbor is at most one adjacent node of some node. In clustering, however ordinary nodes are not forwarding nodes.

Clustering in ad hoc networks has been extensively studied for hierarchical routing schemes [9] [5] [3], the master election algorithms [4], power control [3], reliable broadcast [20] and efficient broadcast [1] [16]. However, to our knowledge, the cluster architecture has not been commonly used for efficient flooding in spite of the potential benefits. First of all, previous clustering schemes are based on the complete knowledge of neighbors. However, the complete knowledge of neighbor information in ad hoc networks is hard to collect and requires huge control overhead caused by periodic exchanges of hello messages. Secondly, none of the clustering algorithms has proposed a gateway reduction mechanism to select the minimal number of gateways. Thus, the clustering suffers from the large number of gateways in the dense network. Lastly, the previous clustering requires huge maintenance cost in high mobility.

We can summarize three important observations as follows.

1. The selection mechanism to choose the dominant set should be efficient and dynamic. Otherwise, the scheme cannot be used effectively and practically.
2. In a MANET, collecting accurate topological information is very hard and carries the huge overhead.
3. Clustering scheme is independent of the network topology unlike route aggregation protocols (e.g., MPR [7], SPAN [22], GAF [23]).

Those facts motivate our new cluster formation protocol called on-demand (passive) clustering. With keeping advantages of clustering, our scheme eliminates the main control overhead.

III. PASSIVE CLUSTERING

A. Overview of Passive Clustering

Passive clustering is an "on demand" protocol. It constructs and maintains the cluster architecture only when there are on-going data packets that piggyback "cluster-related information" (e.g., the state of a node in a cluster, the IP address of the node). Each node collects neighbor information through promiscuous packet receptions. Passive clustering, therefore, eliminates setup latency and major control overhead of clustering protocols.

Passive clustering has two innovative mechanisms for the cluster formation: *First Declaration Wins* rule and *Gateway Selection Heuristic*. With the *First Declaration Wins* rule, a node that first claims to be a *cluster head* "rules" the rest of nodes in its clustered area (radio coverage). There is no waiting period (to make sure all the neighbors have been checked) unlike that in all the weight-driven clustering mechanisms [3] [5]. Also, the *Gateway Selection Heuristic* (Section III.C) provides a procedure to elect the minimal number of gateways (including distributed gateways) required to maintain the connectivity in a distributed manner.

Passive clustering maintains clusters using implicit timeout. A node assumes that some nodes are out of locality if they have not sent any data longer than timeout duration. With reasonable offered load, a node can catch dynamic topology changes.

B. Construction and Maintenance

When a node joins network, it sets the cluster state to INITIAL. Moreover, the state of a floating node (a node does not belong to a cluster yet) also sets to INITIAL. Because passive clustering exploits on-going packets, the implementation of passive clustering resides between layer 3 and 4.
The IP option field for cluster information is as follows:

- Node ID: The IP address of the sender node. This is different to the source address of the IP packet.
- State of cluster: The cluster state of the sender node
- If a sender node is a gateway, then it tags two IP addresses of cluster heads (CHs) which are reachable from the gateway.

We summarize the passive clustering algorithm as follows:

- Cluster states
  There are 6 possible states; INITIAL, CLUSTER_HEAD, ORDINARY_NODE, GATEWAY, CH_READY, GW_READY and DIST_GW.

- The packet handling
  Upon sending a packet, each node piggybacks cluster-related information. Upon a promiscuous packet reception, each node extracts cluster-related information of neighbors and updates neighbor information table.

- A *cluster head* (CH) declaration
  A node in INITIAL state changes its state to CH_READY (a candidate cluster head) when a packet arrives from another node that is not a cluster head. With outgoing packet, a CH_READY node can declare as a cluster head (CH). This helps the connectivity because this reduces isolated clusters.

- Becoming a member
  A node becomes a member of a cluster once it has heard or overheard a message from any cluster head. A member node can serve as a gateway or an ordinary node depending on the collected neighbor information. A member node can settle as an ordinary node only after it has learned enough neighbor gateways. In passive clustering, however, the existence of a gateway can be found only through overhearing a packet from that gateway. Thus, we define another internal state, GW_READY, for a candidate gateway node that has not yet discovered enough neighbor gateways. Recall that we develop a gateway selection mechanism to reduce the total gateways in the network. The detailed mechanism will be shown in the next Section. A candidate gateway finalizes its role as a gateway upon sending a packet (announcing the gateway's role). Note that a candidate gateway node can become an ordinary node any time with the detection of enough gateways.

C. Gateway Selection Heuristic

A *gateway* is a bridge node that connects two adjacent clusters. Thus, a node that belongs to more than two clusters at the same time is eligible to be a *gateway*. One can easily conclude that only one gateway is needed for the each pair of two adjacent clusters. With this observation, we invent gateway selection mechanism that eventually allows only one gateway for each pair of two neighboring cluster heads. However, it is possible that there is no potential gateway between two adjacent clusters. In other words, two cluster heads are not mutually reachable via a two-hop route. If there is a three-hop route between two nodes, then the clustering scheme should select those intermediate nodes as distributed gateways. Without the knowledge of complete two-hop neighbors' information, choosing minimal but enough number of distributed gateways seems to be very difficult. As we examined in the previous chapter, the topological knowledge carries huge overhead and works inefficiently. Therefore, we propose a counter-based distributed gateway selection mechanism instead.

In summary, our gateway selection mechanism can be summarized as follows.

- Gateway
  A node belongs to more than two clusters at the same time is a candidate gateway. Upon sending a packet, a potential gateway decides two cluster heads among known cluster heads. This node will serve as an intermediate node between those chosen cluster heads. This node cannot be an intermediate node of two cluster heads that have announced by another neighbor gateway node. If the node finds two cluster heads, then it finalizes its role as a gateway and announces two cluster heads (CHs) to neighbors.
  If a gateway has received a packet from another gateway which has announced the same pair of CHs, then this node compares the node ID of itself with that of the sender. If this node has the lower ID, it keeps its role as the gateway. Otherwise, it changes the pair of CHs or changes its state. If this node can find another pair of neighbor CHs that is not announced by any other gateway, then it keeps its state as GATEWAY for the new pair of CHs, otherwise it changes its state to ORDINARY_NODE.

- Distributed gateway
  Passive clustering allows one distributed gateway for each cluster head and each node. A node that belongs to only one cluster C can be an ordinary node when at least two (distributed) gateways are known to this node. Otherwise, it keeps the candidate gateway state. A candidate gateway node can be a distributed gateway if there is no neighbor distributed gateway that also belongs to the same cluster C. If an ordinary node has received a packet from a distributed gateway and no gateway is a neighbor node of that node, then this node changes to a distributed gateway.

Figure 4 shows an example of cluster architecture developed by passive clustering. With moderate on-going traffic, passive clustering allows only one gateway for each pair of clusters and enough distributed gateway nodes.

IV. ANALYSIS OF PASSIVE CLUSTERING

In this section, we analyze the overhead and flooding efficiency of passive clustering. For the message overhead, passive clustering just adds 8 bytes or 16 bytes on each outgoing packet. In the analysis we focus on control message O/H, as the number of messages is more important than the size of each packet in ad hoc networks with IEEE 802.11 DCF protocol [14].

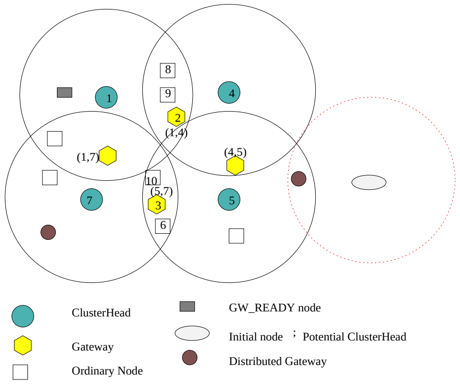

Fig. 4. An Example of Gateway Selection Heuristic. There is at most one gateway between any pair of two *cluster heads*. A gateway can survive only when this node is the only one gateway for announced pair of *cluster heads* or this node has the lowest ID among contention gateways (who announced the same pair of *cluster heads*).

A. Computational Overhead

Passive clustering mechanism is more efficient than distributed tree algorithms in the aspect of processing overhead. The computational overhead of passive clustering is $O(Avg\_Neighbor)$ where $Avg\_Neighbor$ denotes the number of active neighbors. Upon receiving a packet, each node updates its neighbor table and changes its state if necessary. A cluster head only updates its neighbor table. A member node, in addition, should adjust its state based on gateway selection heuristic. The processing overhead for checking enough gateways is max $\{(Num\_CH)(Num\_CH$ -1)(Num_GW)/2, 0 $\}$ when $Num\_CH \geq 2$ and max $\{(Num\_GW), 0\}$ when Num_CH = 1 where Num_CH and Num_GW are respectively the number of cluster heads and gateways known to this node. Therefore, the processing overhead of passive clustering is $Avg\_Neighbor$ + max $\{\frac{(Num\_CH)(Num\_CH-1)(Num\_GW)}{2}, Num\_GW\}$. The maximal number of cluster heads known to one node cannot exceed a certain number (i.e., one node theoretically belongs to less than six clusters at the same time). Thus, $(Num\_CH)(Num\_CH - 1)$ can be bounded by constant C. Finally, the processing overhead will be $O(Avg\_Neighbor)$ since $Avg\_Neighbor$ + C * $Num\_GW \leq Avg\_Neighbor + C * Avg\_Neighbor = O(Avg\_Neighbor)$.

Therefore, each node computes with $O(Avg\_Neighbor)$ computational complexity upon receiving a packet. With outgoing packet, each node simply piggybacks cluster-related information. The complexity is $O(1)$.

B. Reduction of Rebroadcast with Passive Clustering

Passive clustering divides nodes into several groups based on the transmission range of the representative node (cluster head). Thus, the number of forwarding nodes is stable regardless of the geographical density of the network. In other words, the reduction rate improves in proportion to the geographical density.

Figure 5 illustrates the most dense and average case of cluster construction with the assumption that there are infinite number of nodes placed randomly, and the network size is (Tx x Ty) where Tx is the horizontal size and Ty is the vertical size of the network area. Given R of the transmission range of node "A", the number of cluster heads and gateways, in the most dense cluster case, is

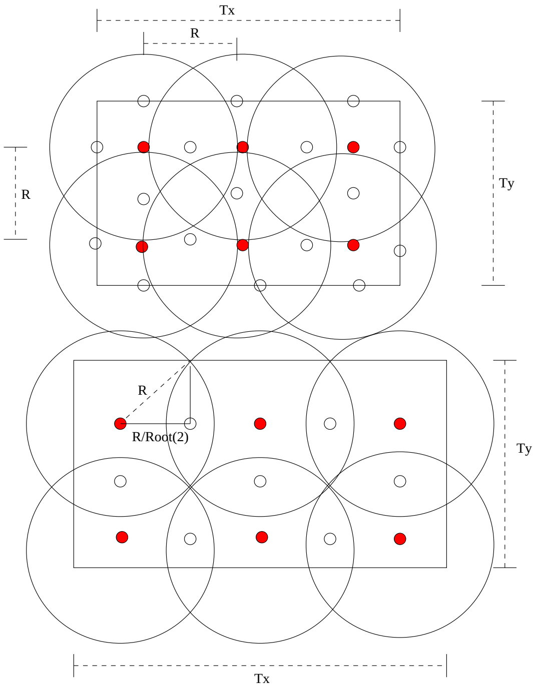

Fig. 5. The average and most dense case of a cluster architecture

$$(\frac{2Tx}{R}+1)(\frac{Ty}{R})+(\frac{Tx}{R})(\frac{Ty}{R}+1). \tag{1}$$

The average number of cluster heads and gateways is

$$(\frac{\sqrt{2}Tx}{R}-1)\frac{\sqrt{2}Ty}{R}+(\frac{\sqrt{2}Tx}{2R})(\frac{\sqrt{2}Ty}{R}-1). \tag{2}$$

For example, given the roaming space 1000x1000 m^2 (Tx = 1000, Ty = 1000), transmission range 250 (R = 250), the number of forwarding nodes will be 21 in the average and 56 in the worst case. If there are totally 1000 nodes in the network, we can save more than 94.4% of rebroadcast in the average.

The reduction rate of passive clustering, consequently, is 1- $\frac{Eq1}{Total_N}$ in average case and 1 - $\frac{Eq2}{Total_N}$ in the worst case such that $Total_N$ denotes the total number of nodes in the network.

V. SIMULATION STUDIES

We simulate passive clustering using Global Mobile Simulation (GloMoSim) library [11], which is a scalable simulation environment for wireless networks using Parsec [15]. First, we illustrate flooding efficiency with passive clustering. We employ a new flooding application where sources send flooding packets to the whole network with constant bit rate. Second, we apply passive clustering to representative reactive ad hoc routing protocols (AODV, DSR and ODMRP), and show the benefits in routing O/H reduction and throughput.

For simulation, we use UDP(User Data Protocol), IEEE 802.11 DCF and two-ray propagation model. The radio propagation of each node reaches up to 250 meters and channel capacity is 2 Mbits/second. The random-way point model is used for node mobility.
Each simulation runs for 600 seconds (10 minutes). The results are averaged over 20 randomly generated node topologies.

A. Flooding Experiments

We analyze flooding efficiency with passive clustering in terms of the flooding reduction rate and the delivery ratio. Each metric is computed as follows.

- *TNP (The Total Number of Packets sent for one broadcast)*: The total number of packets sent from all nodes is divided by the total number of issued broadcast packets from the source. The total packets include the number of rebroadcast packets and the control overhead of the protocol (such as hello or clustering messages in the case of active schemes).
- *NDB (the Number of nodes Delivered the Broadcast)* : The average number of nodes to which the broadcast packet has been delivered. If NDB is equal to the total number of nodes in the network ($Total_N$), then the delivery ratio of broadcast (RDB) is 1. (Note, we exclude the source node)

We demonstrate the superiority of passive clustering by comparison with one of the most efficient flooding tree schemes:*multipoint relay (MPR)* and one of the most popular active clustering protocols:*Lowest ID (LID)* [5]. For reference, we also simulate blind flooding as well. Note that in blind flooding, each node broadcasts at most once the same packet.

We refine *Lowest ID (LID)* algorithm to be applied for efficient flooding as follows: First, we add *UN_DECIDED* state. This state is used for floating nodes that have not decided their final cluster state yet. Those floating nodes also participate in rebroadcast with cluster heads and gateways. Second, LID re-constructs clusters whenever a *cluster head* detects that any member of this cluster has moved out this node's locality. Such maintenance is very poor over the high mobility environment due to excessive overhead. Thus, we modify maintenance to restrict re-construction of clusters only after exchanging of hello messages. Lastly, to improve the flooding efficiency, we develop and add a *gateway reduction method* to LID algorithm. A solution to find the optimal gateways and distributed gateways in a distributed system is NP-complete (*set cover* problem) [19]. Thus, we use the heuristic of MPR scheme. A *cluster head* chooses the list of gateways and sends that list when it broadcasts the cluster information. A *cluster head* chooses a subset of nodes among neighbors which covers up all of nodes within two hop away. A *cluster head* broadcasts the list of *gateways* by piggybacking the chosen set of nodes on the clustering broadcast packet. We can easily prove that those selected gateways are enough to guarantee the complete coverage with assumption of the reliable packet delivery. Like MPR scheme, each node piggybacks the neighbor list on hello messages to exchange two hop neighbors' information.

In summary, four flooding schemes are run as follows: BF (Blind Flooding), MPR-F (Flooding with MPR scheme), AC_LID-F (Flooding with active clustering with Lowest Id Algorithm (AC_LID)) and PC_LID-F (Flooding with passive clustering).

At the beginning of each run, one broadcast source is chosen randomly. After a setup time (10 seconds), the source starts broadcast (flooding) data packets at the rate of 1 packet/second or 4 packets/second. Note that passive clustering does not require setup latency, but other schemes need warm-up time to exchange several hello messages to collect neighbor information. The packet size of each broadcast is 100 bytes.

MPR and AC_LID send hello messages in every 2 seconds following [7]. PC_LID uses 2 seconds cluster timeout. In other words, all entries must be removed from the neighbor list if they are not updated for 2 seconds. MPR and AC_LID use 5 seconds timeout to allow the 1.5 packets loss per each hello message.

Through this simulation, we aim to show that passive clustering is working successfully with scenarios: node mobility and scale. Thus, we first fix the network size and vary node mobility. Then, we test static networks (i.e., no node mobility) of increasing density.

A.1 Fixed Network Size with Node Mobility

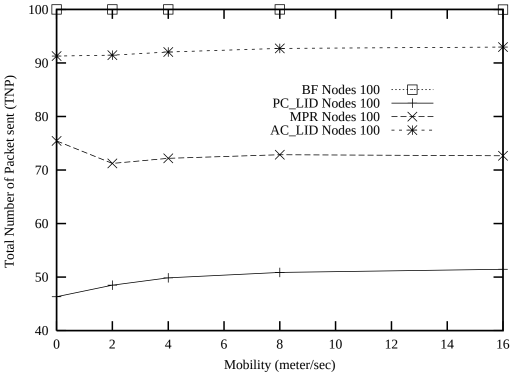

Fig. 6. The TNP of Each Protocol with a single source and the data rate 1pkt/sec. The 100 nodes are place randomly over 1000 x 1000 m^2. The reachable range of each radio power is 250 m.

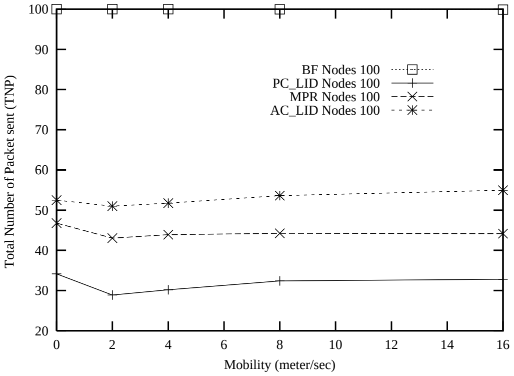

Fig. 7. The TNP of Each Protocol with the packet rate of 4 pkts/sec. The 100 nodes are place randomly over 1000 x 1000 m^2.

We simulate 100 mobile nodes placed randomly within 1000 x 1000 m^2. With these network sizes, the average neighbors will be 8 nodes. We increase node mobility from 0 m/s to 16 m/s with 100 seconds pause time. Figure 6 and 7 indicate the total number of packets required to finish one flooding at different speeds. In those experiments, three remarkable facts are observed. First, flooding efficiency with passive clustering is far better than with the other schemes. This is mainly because passive clustering chooses the sub-optimal dominant forwarding nodes like the other two schemes but it does not require extra control overhead. The other schemes also improve the efficiency of flooding. They, however, are suffering from control overhead due to hello messages or protocol messages. Note that, in spite of extra control overhead and the low data rate (1pkt/sec), each scheme still provides performance gain in terms of TNP metric with deploying an efficient flooding mechanism. Secondly, AC_LID-F generally generates more packets than MPR-F due to many *floating nodes*. Recall that LID algorithm assumes reliable packet delivery. Thus, a control packet loss can block other nodes to complete cluster formation mechanism so that they remain as undecided nodes (floating nodes). Floating nodes should serve as forwarding nodes. As

TABLE I

THE NUMBER OF FLOATING NODES OF AC_LID-F WITH SINGLE SOURCE AND DIFFERENT DATA RATES.

Mobility	0	2	4	8	16
AC_LID-F (1pkt/sec)	6.9	14.1	14.0	14	15.3
AC_LID-F (4pkts/sec)	13.2	14.8	14.8	16.1	17.6

TABLE II

THE NUMBER OF NODES FLOATING NODES OF PC_LID-F WITH SINGLE SOURCE AND DIFFERENT DATA RATES.

Mobility	0	2	4	8	16
PC_LID-F (1pkt/sec)	16.7	23	25	25	25
PC_LID-F (4pkts/sec)	3.7	3.7	4.3	4.9	5.2

(*Floating nodes are the nodes whose cluster state is GW_READY or INITIAL.)

a result, AC_LID tends to have more forwarding nodes than MPR. Table I clearly shows that the number of floating nodes is proportional to the offered load because the increase in offered load reduces the packet delivery ratio. AC_LID-F, in addition to floating nodes, generates about twice control overhead than MPR-F since a node broadcasts another packet to propagate the cluster state to neighbors after exchanging hello messages. Third, the important observation is the difference of flooding efficiency with passive clustering between Figure 6 and 7. The difference shows that flooding efficiency with passive clustering is improved as the user offered load becomes heavy. The reason is that the more frequent user data is generated, the faster passive clustering converges.

The results of delivery ratio in Figure 8 and 9 also show a few interesting facts. First of all, passive clustering clearly provides a robust and efficient platform for flooding. Passive clustering builds strong mesh topology with cluster heads and gateways. While MPR-F and AC_LID-F suffer performance degradations due to incomplete neighbor knowledge. Another outstanding outcome in the results is that the performance of passive clustering is not significantly affected by mobility different from other schemes. This observation verifies that passive clustering maintains clusters dynamically with topology changes. Furthermore, the MPR-F suffer considerable performance damage with highly offered load as Figure 9. This is mainly caused by heavy contention with the high data rate. With AC_LID-F, the increasing number of floating nodes (Table I)improves the delivery fraction in Figure 9.

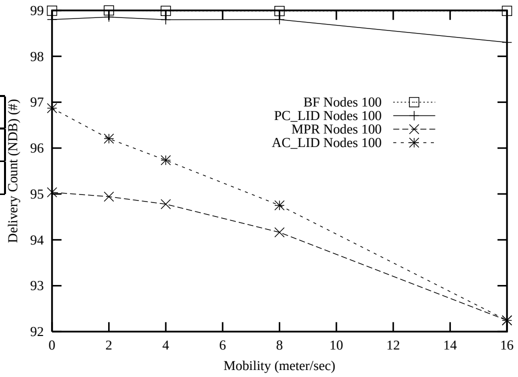

Fig. 8. The NDB of Each Protocol with single source and the data rate 1pkt/sec. The 100 nodes are place randomly over 1000 x 1000 m^2. Clearly, the performance of PC_LID-F is better than AC_LID-F and MPR-F. And clustering schemes outperform MPR protocol.

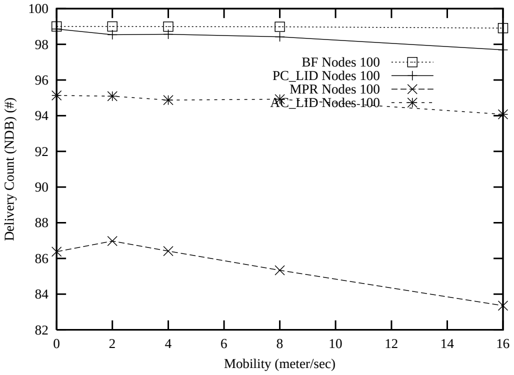

Fig. 9. The NDB of Each Protocol with single source and the packet rate of 4 pkts/sec. The 100 nodes are place randomly over 1000 x 1000 m^2.

A.2 No Mobility with Various Network Size

For the second set of experiments, we use the static network and increase the geographic density by reducing physical network size. In this experiment, 100 nodes are placed randomly over ”x” x 1000 m^2 terrain where ”x” states the horizontal range. We fix the vertical range of the network to 1000 meter and change the horizontal range from 250 to 1500 meter. Figure 10 and 11 show the NDB performance of each protocol following a function of “x”. The MPR performs worse than clustering schemes in sparse networks (i.e., large “x”). This is because inaccurate neighbor topology due to the lost hello messages in MPR has more severe impact on the performance as the network becomes sparse. Moreover, MPR constructs a distributed tree structure and non-leaf nodes forward packets so that a leaf node is likely to have the critical path from the source. While, with clustering algorithm, each node has a few paths from the source because clustering provides a mesh topology instead of a tree structure.

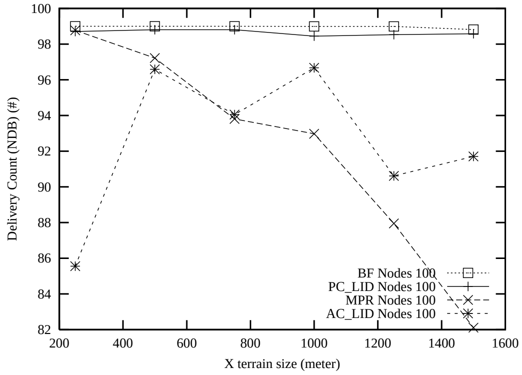

Fig. 10. The NDB of Each Protocol with single source and the data rate 4pkt/sec . The 100 nodes are place randomly over "x" x 1000 m^2.

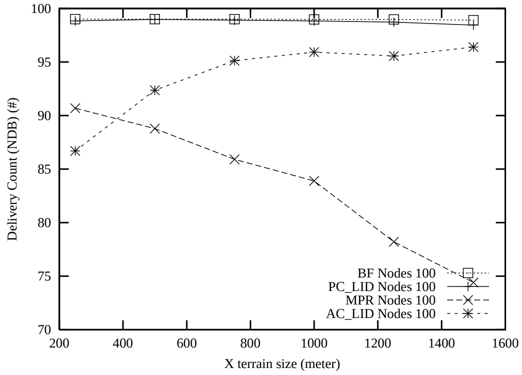

Fig. 11. The NDB of Each Protocol with single source and the data rate 4pkt/sec . The 100 nodes are place randomly over "x" x 1000 m^2.

As in Figure 9, AC_LID-F shows high delivery ratio because of the large number of floating nodes in the network.

Passive clustering provides a fully connected topology regardless of the geographic density of the network.

B. On-demand Routing

We show that passive clustering provides a scalable and effective flooding. Now, we apply passive clustering to reactive routing protocols that depend on flooding. We present this applicability using two prominent on-demand unicast routing schemes: AODV [10] and DSR [12] and a reactive multicast scheme: On-demand Multicast Routing Protocol (ODMRP) [25].

In this experiment, we limit ourselves to passive clustering. MPR and AC_LID require periodic maintenance and control packet exchange and thus are not a good match for reactive routing protocols.

We use the following metrics to show the performance gain with passive clustering.

- *Delivery Ratio*: The total number of packet delivered to destinations is divided by the total number of sent packet from sources.
- *CtrlOH (Normalized Control Overhead)*: The total number of control packet is divided by the total number of delivered packets to destinations.

B.1 Unicast Routing

We use CBR (Constant Bit Rate) sources. We simulate 100 nodes placed randomly within a 1500 x 500 m^2 terrain. Nodes are moving randomly with minimum speed 2 m/s, maximum speed 20 m/s and 100 seconds pause time. We increase the offered load using the number of CBR sessions from 10 to 50. Each CBR source starts a session randomly with the data ratio 2 packets/second and 512 bytes payload size. Note that we include the *noise accumulation* feature in GloMoSim [11] for this experiment. Namely, each node accumulates the power of signals below "receive threshold" as noise.

AODV

We apply passive clustering to our implementation of AODV [10]. AODV has two phases to set up a route: Route Request and Route Reply. The major control overhead of AODV is caused by route queries (RREQ). Therefore, we change the route request phase to apply efficient flooding with passive clustering. In cluster architecture, each node rebroadcasts a new RREQ only when this node is not an *ordinary node*. Consequently, *ordinary node*s are excluded from intermediate nodes for a route. Note that each node could rebroadcast only when the TTL (Time To Live) field of the packet is valid.

Figure 12 and 13 (AODV-PC_LID denotes the combination of AODV and PC_LID-F) demonstrate the performance gain with passive clustering. Passive clustering significantly reduces the flooding overhead and improves the delivery ratio (N $\geq$ 20). As the offered load becomes heavy, the control overhead of AODV grows sharply. It is known that reactive routing protocols tend to generate excessive volume of route queries including re-issuing route queries in heavy offered load [17]. With passive clustering, AODV can improve the performance and scalability since passive clustering mitigates the scalability problem of AODV with efficient flooding.

DSR

DSR has two mechanisms: Route Discovery and Route Maintenance [18]. Route discovery mechanism has two phases: Route Request and Route Reply.

As in AODV, DSR protocol reduces the number of route request packets (RREQ) using aggressive caching

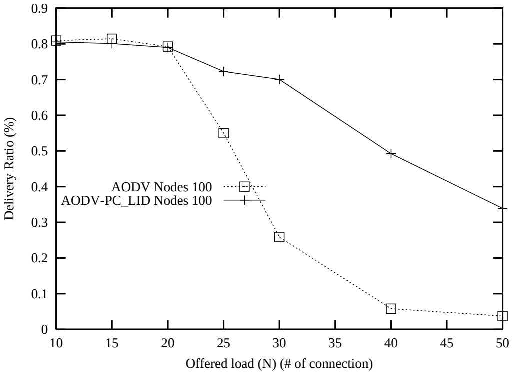

Fig. 12. The Delivery Ratio of AODV with Passive Clustering and without Passive Clustering (100 nodes placed randomly within 1500 x 500 m^2)

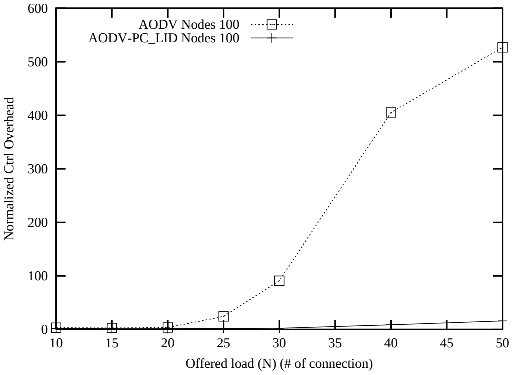

Fig. 13. The Control Overhead of AODV with Passive Clustering and without Passive Clustering (100 nodes place randomly within 1500 x 500 m^2)

of routes. To cache the routes, DSR generates more route replies and errors. Therefore, we apply efficient flooding platforms to the route reply phase as well as the route request phase. Same to AODV, only non-ordinary nodes can forward route queries (RREQ) for the route request phase. For the route reply phase, we change the DSR protocol as follows:

- Route Reply phase: In conventional DSR, a node can initiate a route reply when it receives a new RREQ if it has cached routes to the destination. But with a cluster architecture, only non-*ordinary node*s can initiate this route reply.
- Gratuitous Route Reply [18]: Each node in DSR protocol sends *gratuitous route replies* when it has found a shorter path through this node than the source route in the IP packet. We restrict this feature to only non-*ordinary node*s in a cluster architecture.

Figure 14 and 15 show that passive clustering improves the scalability of DSR by reducing routing overhead ($N \geq 25$). Passive clustering incurs delivery ratio degradation with low offered load ($N \leq 20$). The main reason is that passive clustering restricts route optimization and caching. Thus, the average hop count tends to increase and the route queries are triggered more frequently with passive clustering than original DSR.

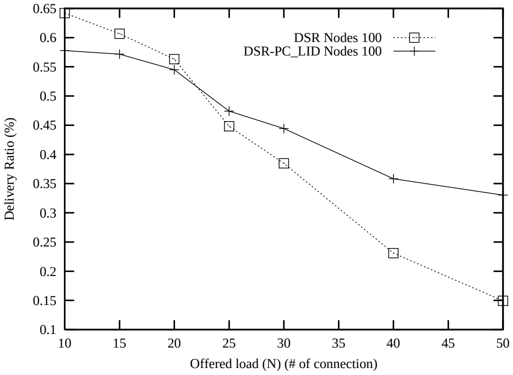

Fig. 14. The Delivery Ratio of Each Protocol of DSR with Passive Clustering and without Passive Clustering (100 nodes placed randomly within 1500 x 500 m^2)

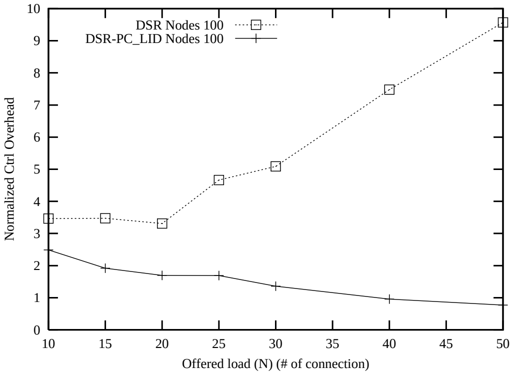

Fig. 15. The Control Overhead of DSR with Passive Clustering and without Passive Clustering (100 nodes placed randomly within 1500 x 1500 m^2)

B.2 Multicast Routing

We simulate 50 nodes placed randomly within a 1500 x 300 m^2 terrain. The node mobility is increased from 0 m/s (i.e., no node mobility) to 16 m/s with 10 seconds pause time. Nodes are moving based on randomway point model. 5 sources are multicasting data packet with 1024 bytes/second data rate and 16 nodes are the total members of 5 sources.

We apply passive clustering to our implementation of ODMRP [25]. ODMRP has two phases to set up a multicast: Join Query and Join Reply. A source floods Join Query packet periodically to find the members of this multicast group. Thus, we use passive clustering to reduce the flooding overhead of Join Query. Only non-*ordinary node*s are forwarding Join Query packet upon receiving the Join Query packet. Consequently, *ordinary nodes* are excluded from Forwarding Groups for any multicast group.

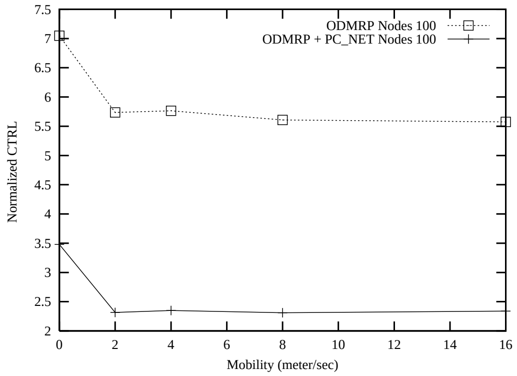

Fig. 16. The normalized control overhead of ODMRP with Passive Clustering and without Passive Clustering. 50 nodes placed randomly over 1500 x 300 m^2. The radio range is 250 m

Figure 16 illustrates the reduction of control overhead for Join Query Packet. The throughput performance is about the same with and without passive clustering.

We can conclude that passive clustering can be applied to several reactive unicast or multicast routing protocols to reduce control overhead and improve the performance and scalability.

VI. CONCLUSION

In this paper, we have introduced an enhanced version of the Passive Clustering protocol. We have applied it to flooding and have showed that it performs as well as (if not better than) existing flood rebroadcast control schemes. Finally, we have applied it to on demand uni and multicast routing.

This paper includes several contributions. First, we improve the clustering scheme with an effective gateway selection heuristic. Our gateway reduction mechanism permits the use of the cluster architecture as a robust and efficient flooding platform over dense, large mobile networks.

Secondly, we investigate the problem of efficient flooding based on topological information. To collect neighbor topology, the network incurs a heavy control overhead penalty- it is very costly to collect accurate topology information with node mobility and dynamic resources. The aforementioned topology-based schemes, in consequence, have the limiting factor of scalability and performance due to the burden of message and processing overhead. Based on those observations, we have conceived a new flooding scheme based on passive clustering. Our flooding scheme is efficient, scalable and robust.

Finally, we show the applicability of passive clustering to a few reactive routing protocols (such as AODV, DSR and ODMRP). Passive clustering reduces the flooding overhead and improves the performance and scalability.

REFERENCES

[1] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan C hen, Jang-Ping Sheu, *The broadcast storm problem in a mobile ad hoc network*, MobiCom, 1999.
[2] Jinyang Li, Charles Blake, Douglas S.J., De Couto, Hu Imm Lee, Robert Morris, *Capacity of Ad Hoc Wireless Networks*, Mobicom, 2001.
[3] C.R. Lin and M. Gerla, *Adaptive Clustering for Mobile Wireless Networks*, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, Sep. 1997, pp. 1265-1275.
[4] Dennis J. Baker and Anthony Ephremides, *The Architectural organization of a mobile radio network via a distributed algorithm*, IEEE Transaction on communications, Vol. com-29, No. 11, Nov. 1981
[5] Gerla, M and Tasi, J., *A Multicluster, mobile, multimedia radio network*, ACM-Baltzer Journal of Wireless Networks, Vol. 1 No. 3, 1995
[6] Yu-Chee Tseng, Sze-Yao Ni, En-Yu Shih, *Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network*, Infocom, 2001.
[7] Amir Qayyum, Laurent Viennot, Anis Laouiti, *Multipoint relaying: An efficient technique for flooding in mobile wireless networks*, INRIA report, March 2000.
[8] H.Lim and C. Kim, *Flooding in wireless ad hoc networks*, IEEE computer communications 2000.
[9] Krishna, P., Vaidya, N.H., Chatterjee, M., Pradhan, D.K., *A cluster-based approach for routing in dynamic networks*, Computer Communication Review, vol.27, (no.2), ACM, April 1997.
[10] C. Perkins, E. Royer, S. Das, *Ad hoc on demand distance vector (AODV) routing*, http://www.ietf.org/internet-drafts/draft-ieft-manet-aodv-03.txt, 1999
[11] UCLA, *Glomosim: A scalable simulation environment for Wireless and Wired Network systems*, http://pcl.cs.ucla.edu-/projects/glomosim.
[12] D.B.Johnson and D.A. Maltz, *Dynamic source routing in ad hoc wireless networks*, Mobile Computing, Academic Publishers, 1996. pp. 153-181.
[13] Z.J.Haas, *A new routing protocol for the reconfigurable wireless networks*, ICUPC '97 pp.562-566
[14] Giuseppe Bianchi, *Performance analysis of the IEEE 802.11 distributed coordination function*, IEEE Journal on selected areas in communications, vol. 18. March 2000.
[15] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, H. Song, *Parsec: A parallel simulation environment for complex systems*, Computer, Vol. 31(10), October 1998, pp. 77-85.
[16] M. Jiang, J. Li, Y. C. Tay, *Cluster based routing protocol (CBR) functional specification*, IETF draft 1998.
[17] Yunjung Yi, Taek Jin Kwon, Mario Gerla, *A load aware routing (LWR) based on local information*, PIMRC 2001 accepted.
[18] J. Broch, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, *A performance comparison of multi-hop wireless ad hoc network routing protocols*
[19] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley Publication. 1974.
[20] E. Pagani, G.P. Rossi, *Reliable broadcast in mobile multihop packet networks*, Mobicom 97 (1997)
[21] J. Wu and H. Li, *On calculating connected dominating set for efficient routing in ad hoc networks*, DIAL M. 1999
[22] B. Chen, K. H. Jamieson, and R. Morris *An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks*, Mobicom, 2001
[23] Y. Xu, J. Heidemann and D. Estrin *Geography-informed Energy Conservation for Ad Hoc Routing* Mobicom, 2001

[24] M. Seddigh, J. Solano and I. Stojmenovic, *Internal nodes based broadcasting algorithms in wireless networks*, Proceedings of the 34th Annual HICSS 2001

[25] 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.

[26] Yunjung Yi, Taek Jin Kwon and Mario Gerla *Passive Clustering (PC) in Ad Hoc Networks*, Internet Draft, draft-ietf-yi-manet-pac-00.txt, Nov. 2001.

[27] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot *Optimized Link State Routing Protocol*, Internet Draft, draft-ietf-manet-olsr-05.txt, Nov. 2001.

[28] R. G. Ogier, F. L. Templin, B. Bellur, M. G. Lewis *Topology Broadcast Based on Reverse-Path Forwarding (TBRPF)*, Internet Draft, draft-ietf-manet-tbrpf-03.txt, Nov. 2001.