# EFFICIENT FLOODING IN AD HOC NETWORKS USING ON-DEMAND (PASSIVE) CLUSTER FORMATION

Yunjung Yi, Mario Gerla and Taek-Jin Kwon*
Computer Science Department
University of California, Los Angeles, CA 90095
e-mail: {yjyi, gerla}@cs.ucla.edu
Telcordia Technologies
Red Bank, NJ
e-mail: tkwon@research.telcordina.com

*Abstract*—
**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 offers several advantages compared with "active" clustering and route aggregation techniques. In particular, 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 within transmission range of each other. Otherwise, nodes must communicate via a multihop route. To find such a multi-hop route, MANETs commonly employ on demand routing algorithms that use *flooding* or *broadcast* 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 will refer to 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, blind flooding suffers from the increase of (1) redundant and superfluous packets, (2) probability of collision, and (3) congestion of wireless medium [1]. Performance of blind flooding is severely impaired especially in large and dense networks [2].

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

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 the periodic collection of topology information difficult and costly (in terms of overhead). For that reason many on-demand ad hoc routing schemes and service discovery protocols simply use blind flooding [10] [12] [25]. In contrast with on-demand routing methods, the proactive ad hoc routing schemes by virtue of periodic route table exchange, can gather topological information without much extra overhead. Thus, the leading MANET proactive ad hoc routing schemes use route aggregation methods to forward routing packets through only a subset of the neighbors [27] [28].

In this paper, we focus on on-demand routing protocols and propose mechanism for efficient flooding based on passive clustering. We require neither the deployment of GPS-like systems nor explicit periodic control messages to identify the subset of forwarding neighbors. Our scheme makes the following contributions compared with previous efficient flooding schemes (such as multipoint relay, neighbor-coverage, etc): (1) it does not need any periodic messages. Instead, it exploits existing traffic to piggyback its small control messages; (2) 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; (3) it does not introduce any startup latency; (4) it saves energy if there is no traffic; (5) it easily adapts to topology and available resource changes.

The organization of the paper is as follows. We present related work in Chapter II, describe the detailed algorithm in chapter III, report simulation results in Chapter IV. Finally, we conclude the paper in Chapter V.

## II. Related Work

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

In [1] [6], the authors propose several heuristics to reduce rebroadcasts. More specifically, upon receiving a flood packet, a node decides whether to relay it or not based on one of the following heuristics: (1) rebroadcast with given probability; (2) rebroadcast if the number of received duplicate packets is less than a threshold; (3) distance-based scheme where the relative distance between hosts determines the rebroadcast decision; (4) location-based scheme where the decision is based on pre-acquired neighbor location information; (5) cluster-based scheme where only pre-computed cluster heads and gateways rebroadcast. Our approach, passive clustering, differs from the above schemes in that it provides a more systematic method based on locally collected information (e.g., neighbor information, cluster states, etc.). Each node participates in flooding based on its role or state in the dynamically constructed cluster architecture instead of depending on local heuristics or on pre-computed clusters.

Another approach to efficient flooding is to exploit topological information [6] [8] [7] [24]. In the absence of preexisting infrastructure, all the above shemes use a periodic *hello message* exchange method to collect topological information. Our approach does not require periodic control messages. Rather, it exploits on-going data packets to exchange cluster-related information. The authors of [8] suggest 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 its neighbors on outgoing packet. A node rebroadcasts (becomes a forwarding node) only when it has neighbors that are not covered by its forwarding nodes. While the *self-pruning* heuristic utilizes information of directly connected neighbors only, the *dominant-pruning* heuristic extends the propagation of neighbor information two-hop away. The *dominant-pruning* scheme is actually 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, then selects the minimal subset of forwarding neighbors, which cover all nodes within two-hops. Each sender piggybacks its chosen forwarding nodes (MPRNs) on the outgoing broadcast packet.

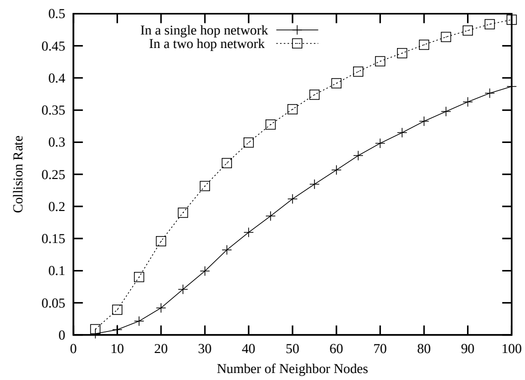


Fig. 1. The collision rate of broadcast

Along the same lines, several other schemes have proposed the selection of a dominant set based on topology [21] [22] [23]. All of these schemes, however, again depend on periodic hello messages to collect topological information.

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 (all nodes hear each other) and a two hop network (with distance at most two hops) as the number of neighbors increases. This result clearly shows that the neighbor degree causes the broadcast collision probability to increase (note, the collision probability is more than 0.1 with more than 15 neighbors). Moreover, the hidden terminal condition aggravates collisions in the two hop network. Note that Figure 1 assumes no data traffic - only hello messages. With user-data packets, the collision probability of hello messages will dramatically increase. Thus, it will be hard to collect complete neighbor topology information using hello messages. As a consequence, the aforementioned schemes (e.g., neighbor-coverage, MPR, etc.) are not scalable to offered load and number of neighbors.

Lastly, we consider clustering. Clustering can be described as *grouping nodes into clusters*. A representative of each group (cluster) is dynamically elected to the role of *cluster head* based on some criterion (e.g., lowest ID). Nodes within one hop of a clusterhead become associated to its cluster. A node belonging to two or more clusters at the same time is called a *gateway*. Other members

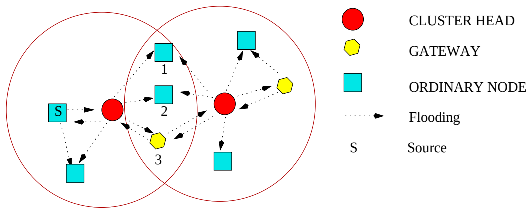


Fig. 2. An Example of Efficient Flooding with Clustering. Only cluster heads and gateways rebroadcast.

are called *ordinary nodes*. Various distributed computation techniques can be used to dynamically create clusters. In the active clustering lowest ID technique [5] each node attempts to become clusterhead by broadcasting its ID to neighbors. It will give up only if it hears a lower ID neighbor. In the sequel, we will discuss other cluster formation techniques in more detail. Based on the above definition, any two nodes in a cluster are at most 2 hops away [9]. With the clustering scheme, the dominant forwarding nodes are the clusterheads and the gateways, as shown in Figure 2.

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 rarely been used for efficient flooding for a number of reasons. First, 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 introduces substantial control overhead caused by periodic exchange of hello messages. Secondly, none of the proposed clustering algorithms includes 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 existing clustering schemes pose high maintenance costs in high mobility.

These limitations motivated our investigation of a new cluster formation protocol called on-demand (passive) clustering. While retaining the advantages of clustering, our scheme eliminates much of the 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 for 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 the nodes it had previously heard from have died or are out of its locality if they have not sent any data within timeout duration. With a reasonable offered load, a node can easily keep track of dynamic topology changes by virtue of this timeout.

### B. Construction and Maintenance

A node can be in different states during the clustering process, namely:

**node states in a Cluster**: There are 6 possible node states; INITIAL, CLUSTER_HEAD, ORDINARY_NODE, GATEWAY, CH_READY, GW_READY and DISTR_GW.

When a node joins the network, it sets its cluster state to INITIAL. Moreover, the state of a floating node (a node does not belong to a cluster yet) also is set to INITIAL. Because passive clustering exploits data packets, the implementation of passive clustering resides between layer 3 and 4. An additional field in the header (the cluster information field) is carried by each packet. This field contains the following entries:

- Node ID: The IP address of the sender node. Not to be confused with the source address of the IP packet.
- State of cluster: The cluster state of the sender node
- Two cluster heads addresses: If a sender node is a gateway, then there is another field with the two IP addresses of the cluster heads (CHs) which are reachable from the gateway.

Below, we provide a summary description of the passive clustering algorithm. See also Figure 3 for an illustration of the procedure.

- The packet handling:

Upon sending a packet, each node piggybacks cluster-related information in the cluster info field. 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 the next outgoing packet, a CH_READY node can declare itself as a cluster head (CH).

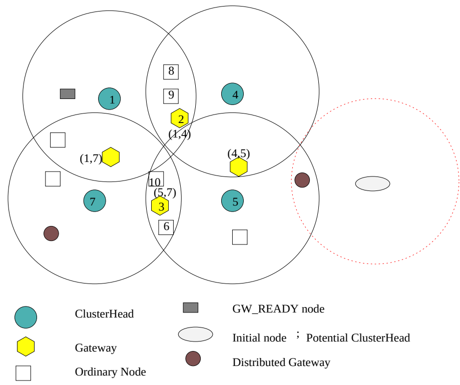


Fig. 3. An Example of Gateway Selection Heuristic. There is at most one gateway between any pair of two *cluster head*s. A gateway can survive as such only when this node is the only gateway for the announced pair of *cluster head*s or this node has the lowest ID among contending gateways (who announced the same pair of *cluster heads*).

• Becoming a member (Gateway or Ordinary node)
A node becomes a member of a cluster once it has heard or overheard a message from any cluster head. A member node will serve as a gateway or an ordinary node depending on the information collected from neighbors. Specifically, a member node settles as an ordinary node only after it has learned (i.e., has heard from) enough neighbor gateways. In passive clustering, 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 (more detail in the next Section). A candidate gateway finalizes its role as a gateway upon sending a packet (announcing its gateway's role). Note that a candidate gateway node can be downgraded to ordinary node at any time after 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 two or more clusters at the same time is eligible to be a *gateway*. One can easily see that only one gateway is needed for the each pair of adjacent clusters. Following this observation, we have developed a gateway selection mechanism that eventually allows only one gateway for each pair of neighboring cluster heads. However, it is possible that there is no potential gateway between two communicating clusters. For instance, suppose that two cluster heads are mutually reachable not by a two-hop but a three-hop route. Then the clustering scheme should select the two intermediate nodes as distributed gateways (DISTR_GW).

The gateway selection mechanism can be summarized as follows.

• Gateway
A node that belongs to two or more clusters at the same time is a candidate gateway. Upon sending a packet, a potential gateway selects two cluster heads among the known cluster heads. This node will serve as an intermediate node between those chosen cluster heads. It cannot be the intermediate gateway for any two other cluster heads that have already been announced by another neighbor gateway node. If the node finds an unique pair of cluster heads, then it finalizes its role as a gateway and announces the pair of cluster heads 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 selects another pair of CHs (that it has heard from) or changes its state to ordinary node.

• Distributed gateway
Passive clustering uses distributed gateway to provide connectivity among clusterheads 3 hops away. Moreover, distributed gateways are common at the boundary of the cluster structure (as shown in Fig 4). A node that belongs to only one cluster $C$ (i.e., has heard only from cluster $C$) can be an ordinary node only if at least two (distributed) gateways are known to this node. Otherwise, it keeps the candidate gateway state. A candidate gateway node becomes a distributed gateway if it has not heard from any other neighboring distributed gateway belonging to the same cluster $C$. If an ordinary node has received a packet from a distributed gateway and no other gateway is a neighbor node of that node, then this node changes to a distributed gateway.

Figure 3 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. Flood packets are forwarded by all nodes types but ORDINARY nodes.

## IV. Simulation Studies

We simulate passive clustering using the Global Mobile Simulation (GloMoSim) library [11], which is a scalable simulation environment for wireless networks based on the Parsec language[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 overhead 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 have refined the *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 the LID algorithm as well. 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 the neighbor gateways which covers up all of the 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. One can easily see that those selected gateways are enough to guarantee the complete coverage under the assumption of reliable packet delivery. Like in the 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 either 1 packet/second or 4 packets/second (two separate cases). 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 broadcast packet size is 100 bytes. MPR and AC_LID send hello messages 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 for a 1.5 packet loss per each hello message. Throughout this simulation, we aim to show that passive clustering is working successfully with node mobility and large scale scenarios. 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

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 4 and 5 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 when used for efficient flooding. 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 prevent other nodes from forming/joining completed clusters so that they remain as undecided nodes (floating nodes). Floating nodes should serve as forwarding nodes. As a result, AC_LID tends to have more forwarding

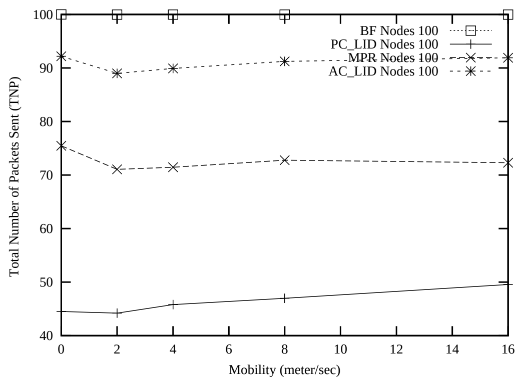


Fig. 4. The TNP of Each Protocol with a single source and the data rate 1pkt/sec. The 100 nodes are place randomly over 1000 x 1000 $m^2$. The reachable range of each radio power is 250 $m$.

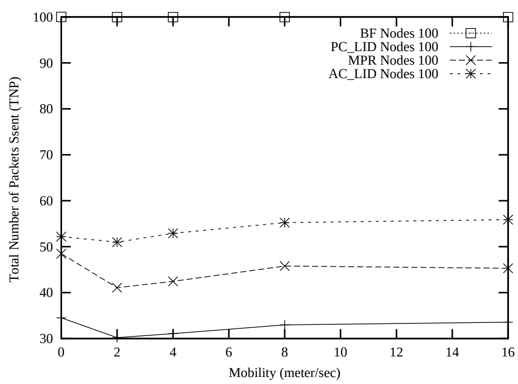


Fig. 5. The TNP of Each Protocol with the packet rate of 4 pkts/sec. The 100 nodes are place randomly over 1000 x 1000 $m^2$.

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 |

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 increases collisions and 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 4 and 5. The difference shows that flooding efficiency with passive clustering is improved as the user offered load becomes heavy. The reason is that the more frequently user data is generated, the faster passive clustering converges.

The results of delivery ratio in Figure 6 and 7 also show a few interesting facts. First, passive clustering clearly provides a robust and efficient platform for flooding. While MPR-F and AC_LID-F suffer performance degradations due to incomplete neighbor knowledge. Also, the performance of passive clustering is not significantly affected by mobility. This observation confirms that passive clustering maintains clusters efficiently, tracking the topology changes. Furthermore, the MPR-F suffer considerable performance damage with highly offered load as shown in Figure 7. This is mainly caused by heavy contention due to the

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.)

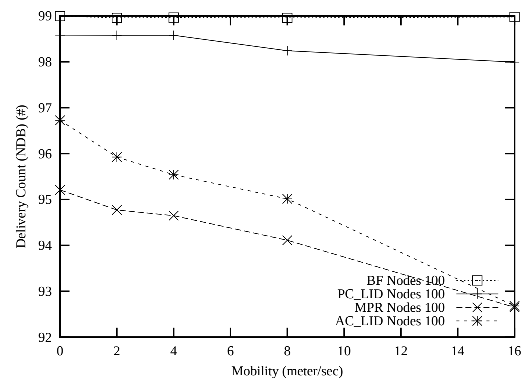


Fig. 6. The NDB of Each Protocol with single source and the data rate 1pkt/sec. The 100 nodes are place randomly over 1000 x 1000 $m^2$. Clearly, the performance of PC_LID-F is better than AC_LID-F and MPR-F. And clustering schemes outperform MPR protocol.

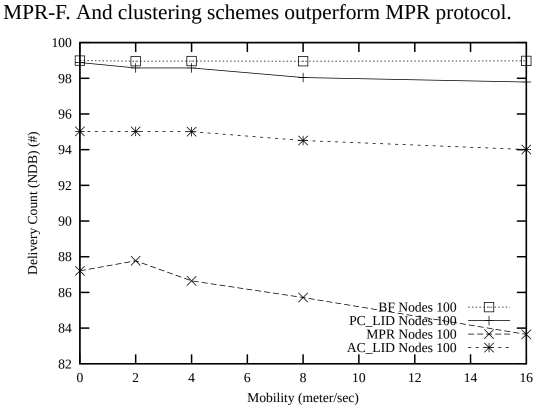


Fig. 7. The NDB of Each Protocol with single source and the packet rate of 4 pkts/sec. The 100 nodes are place randomly over 1000 x 1000 $m^2$.

high data rate. With AC_LID-F, the increasing number of floating nodes (Table I) improves the delivery fraction in Figure 7.

### 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 "H x 1000 $m^2$ terrain where "H 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 8 and 9 show the performance of each protocol following a function of "H". The MPR performs worse than clustering schemes in sparse networks (i.e., large "H"). 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. 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.

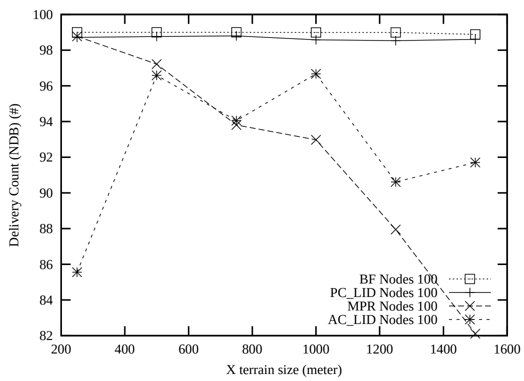


Fig. 8. The NDB of Each Protocol with single source and the data rate 1pkt/sec . The 100 nodes are place randomly over "H x 1000 $m^2$.

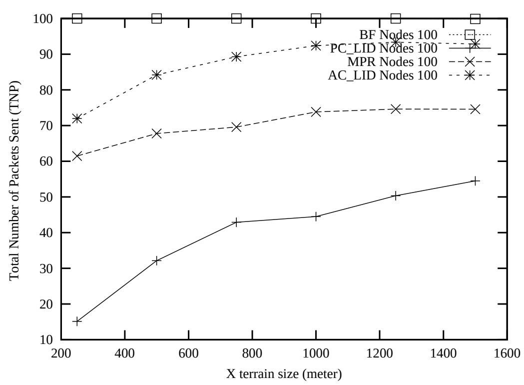


Fig. 9. The TNP of Each Protocol with single source and the data rate 1pkt/sec . The 100 nodes are place randomly over "H x 1000 $m^2$.

Figure 9 illustrates that, as the network becomes denser, the flooding overhead of PC_LID-F becomes considerably smaller. While other schemes do not show significant performance gain with the increase of network density.

## B. *On-demand Routing*

We have shown 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 application using two prominent on-demand unicast routing schemes: AODV [10] and DSR [12] and; a popular 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 packets 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 1000 x 1000 $m^2$ terrain for AODV and 1500 x 500 terrain $m^2$ for DSR. 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 5 to 40 for AODV and from 10 to 50 for DSR. 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 the flood of route queries (RREQ). Therefore, we apply efficient flooding with passive clustering to RREQ. Namely, 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 rebroadcasts only when the TTL (Time To Live) field of the packet is valid.

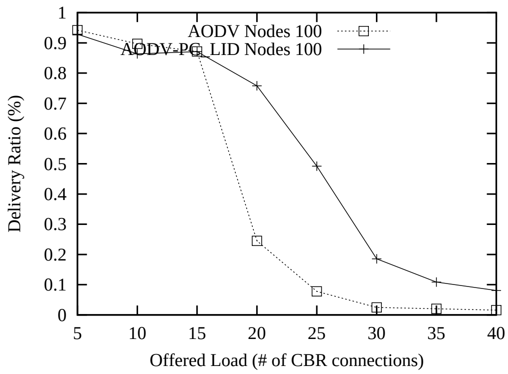


Fig. 10. The Delivery Ratio of AODV with Passive Clustering and without Passive Clustering (100 nodes placed randomly within 1000 x 1000 $m^2$. Each CBR source starts a session randomly with the data rate of 4 packets/second and 512 bytes payload size.)

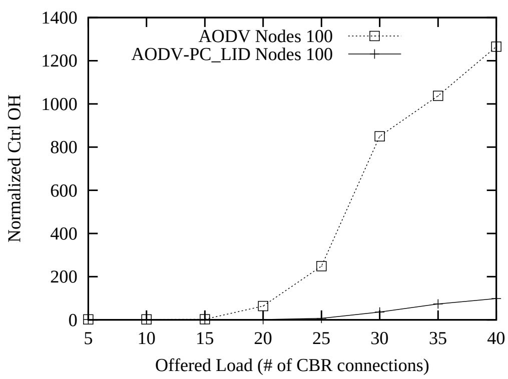


Fig. 11. The Control Overhead of AODV with Passive Clustering and without Passive Clustering (100 nodes place randomly within 1000 x 1000 $m^2$)

Figure 10 and 11 (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 $\geq$ 15). 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 [12] 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 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.

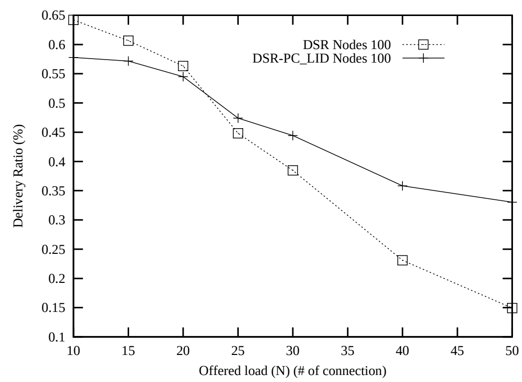


Fig. 12. The Delivery Ratio of Each Protocol of DSR with Passive Clustering and without Passive Clustering (100 nodes placed randomly within 1500 x 500 $m^2$. Each CBR source starts a session randomly with the data rate of 2 packets/second and 512 bytes payload size.)

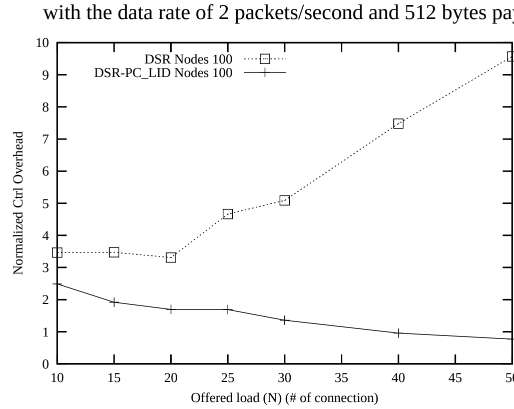


Fig. 13. The Control Overhead of DSR with Passive Clustering and without Passive Clustering (100 nodes placed randomly within 1500 x 1500 $m^2$)

Figure 12 and 13 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.

### 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 random-way 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 packets 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 node*s are excluded from Forwarding Groups for any multicast group.

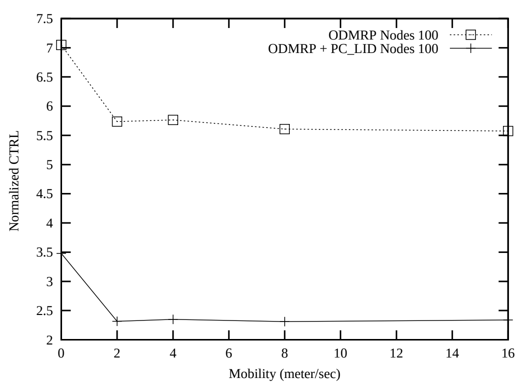


Fig. 14. The normalized control overhead of ODMRP with Passive Clustering and without Passive Clustering. 50 nodes placed randomly over 1500 x 300 $m^2$. The radio range is 250 $m$

Figure 14 illustrates the reduction of control overhead for Join Query Packet.

The above experiments clearly show that passive clustering can be successfully applied to several reactive unicast or multicast routing protocols to reduce control overhead and improve the performance and scalability.

## V. CONCLUSION

In this paper, we have introduced the Passive Clustering protocol and have applied it to flooding, showing that it performs as well as (if not better than) existing flood re-broadcast 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 dynamically changing resources. The aforementioned topology-based schemes, in consequence, are limiting in scalability and performance due to the burden of message and processing overhead. We show that a flooding scheme based on passive clustering removes such limitations. In fact, our proposed flooding scheme was proven to be efficient, scalable and robust.

Naturally, passive clustering finds its home in the domain of on-demand protocols. As such protocols come alive only when there is user data to send, and moreover they almost universally require some form of flooding; they are the ideal candidates for passive clustering enhancement. Clearly, no proactive flood enhancement scheme would make sense in this environment, as it would introduce undesirable, periodic background traffic. The key issue here is to evaluate the enhancement introduced by passive clustering versus the original protocol version. We have applied passive clustering to the most popular reactive routing protocols (AODV, DSR and ODMRP). Passive clustering consistently 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.