Efficient Flooding with Passive Clustering (PC) in Ad Hoc Networks

Taek Jin Kwon
Telcordia Technologies
331 Newman Springs Rd.
Red Bank, NJ 07701, USA
tkwon@research.telcordia.com

Mario Gerla
University of California at Los Angeles
405 Hilgard Ave.
Los Angeles, CA 90095-1596, USA
gerla@cs.ucla.edu

ABSTRACT

An ad hoc network is a fast deployable self-configuring wireless network characterized by node mobility, dynamic topology structure, unreliable media and limited power supply. Nodes in an ad hoc network must cooperate and carry out a distributed routing protocol in order to make multi-hop communications possible. On Demand Routing is one of the most popular routing styles in ad hoc networks. In On Demand Routing, "flooding" is used to find a feasible route from source to destination. The function of flooding is to deliver a packet from one source to every other node in the system. Conventional flooding can be very costly in On Demand networks in terms of network throughput efficiency as well as node energy consumption. The main reason is that the same packet is rebroadcast unnecessarily several times (redundant rebroadcast). Indeed, the penalty of redundant rebroadcast increases when the size of network grows and the density of network increases. In this paper we introduce a novel clustering scheme, call Passive Clustering that can reduce the redundant rebroadcast effect in flooding. We demonstrate the efficiency of the proposed scheme in the AODV (Ad hoc, On demand Distance Vector) routing scheme.

1. FLOODING IN AD HOC NETWORKS

Flooding is a packet dissemination procedure by which every incoming packet at a node is sent out on every outgoing link except the one it arrived on. In a wireless environment, the physical exclusion of the arriving link is impossible. Since the media is broadcast, a single relay of a flooding packet fulfils the task if the broadcasting is successful, i.e., all neighbors receive the packet.

Unfortunately, some of the neighbors may not receive the packet due to many reasons including noise, receivers' status, mobility, collision etc. Since every neighbor that has received the packet will rebroadcast it, flooding can generate an infinite number of duplicate packets if there is no control mechanism. One of the mechanisms for prohibiting infinite duplication is tracking flooding packets. Duplicates are detected (from a unique source identifier and a sequence number, for example) by each receiving node and are immediately discarded in order to avoid endless looping. Another control mechanism is Time-to-Live (TTL). A flooding packet carries a TTL field which represents the maximum hop that the packet can traverse. Upon reception of a flooding packet, the receiving node checks the TTL field and determines whether the packet will be re-broadcasted (after decreasing TTL) or dropped. Path logging in a flooding packet can also be a controlling mechanism. By carrying a list of nodes that a flooding packet has visited, a node can easily avoid looping by examining its ID in the list. If there is a match, the node drops the "returning" packet. In spite of the control mechanisms listed above, flooding generates replicated packet arrivals to each node; namely, one replica for each neighbor. Thus, flooding overhead corresponding to replicated, redundant packets increases with connectivity. Flood search is the capstone of all on-demand routing and multicast protocols. These protocols need to find a path on demand. Since one generally assumes that there is no underlying routing or relative geographical positioning infrastructure that can guide the packet to destination, a path search query must be flooded to the entire network, or at least

through a certain section (scope) of it. Once the path search query packet reaches a destination by flooding, the destination can report a path to the source as a reverse path through which the search packet came. Or the destination can report the path to the source with another flooding in case there are asymmetric links. AODV (Ad hoc On-demand Distance Vector routing [1]), for example, uses scoped flooding to find a route. By tagging "Time To Live (TTL)" on each Route-Request flooding packet, a source gradually enlarges flood search diameters. On the other hand, DSR [2] depends on complete flooding to the entire network if a source cannot find a path to destination in a single hop. If the communication patterns are "local", scoped flooding is effective. On the other hand, if destinations typically many hops away, it would be wasteful to run the incremental scoped flooding.

2. EFFICIENT FLOODING

Generally speaking, flooding in ad hoc networks is used to find a feasible route to a destination or to advertise routing information. If the network is dense, it is not necessary for every node to relay the flood search packet. In fact, it may suffice to use only a subset of nodes as relays. There are many ways to reduce the number of forwarding participants. All of the approaches concern selecting the dominant set, i.e., a minimal subset of forwarding nodes which is sufficient to deliver the flooding packet to every other node in the system. There are two basic approaches for selecting the dominant set: without and with a clustering structure.

The first approach (no clustering) includes the building of a source tree with the maximal number of leaf nodes [3,4,5] and the building of a well-covered mesh [6,7]. By excluding leaf nodes from forwarding participation, the method can improve flooding efficiency. To build such source tree, two-hop connectivity information is necessary. To collect the required information, at least two complete floodings from a source are necessary. The first flooding (which can be replaced with well-coordinated hello messages) is to learn the one-hop neighbors. The second flooding is to report the direct (one-hop) neighbor lists. By collecting the complete neighbor lists of all of its neighbors, a node can construct the two-hop connectivity, i.e., the list of nodes that are two hops away. From this list, each node selects the minimum set of one-hop neighbors which cover all the downstream two-hop neighbors. This problem can be reduced to the well-known "set-cover" problem (NP-complete). Starting from a source and applying this procedure recursively one generates the non-leaf nodes of a minimal flooding tree. Span [6] and GAF [7] build their dominant set as a well-covered mesh. Span selects nodes that are potentially on critical paths as coordinators, i.e. members of a dominant set. GAF partitions the region with a grid such that any nodes in neighboring cells can communicate each other; one node per cell is selected to form the dominant set. The complexity of the selection algorithm in this category is dependent on the number of neighbors (except for GAF which requires GPS information instead). In other words, complete neighbor list knowledge is always the assumption. Note that the neighbor-learning procedure is not trivial in ad hoc networks and it involves substantial overhead with high node density and mobility.

The second approach is based on a two-hop clustering structure. To illustrate this concept, let us consider the n node example in Figure 1. Let r be a transmission range, and the size of the roaming space be $\frac{k}{\sqrt{2}}r \times \frac{k}{\sqrt{2}}r$ where k is an even number (Figure 1 depicts the case of $k = 6$). There are n nodes in the square, but in the figure we only show the nodes at coordinates $(\frac{a}{\sqrt{2}}r, \frac{b}{\sqrt{2}}r)$ where either a or b is an integer smaller than k. This "selection" of nodes is known as "two hop clustering.", ie, any two nodes in a cluster are separated by at most two hops. The nodes at the center of the circles are "cluster heads" and the light-shaded nodes in between are "gateways." Clearly, such nodes represent a connected set. They are in fact the dominant set required to forward the flood packets. Without the cluster overlay shown in Figure 1, each flood packet is relayed exactly n-1 times, as each node must rebroadcast the packet once. On the other hand, $(k-1)\times\frac{k}{2}+\frac{k}{2}\times(\frac{k}{2}-1)=\frac{k(3k-4)}{4}$

broadcasts suffice if only clusterheads and gateways forward the packet. Note that in the cluster restricted forwarding, ALL nodes still receive the flood packet. The flooding reduction is thus $\frac{\frac{k(3k-4)}{4}}{n-1}$.

In a case of n = 100 and k = 6, the number of broadcasts required in the cluster is 21 instead of 99. In other words, 78.8% of transmissions can be saved. This is not even a very dense network (each node has about 12 neighbors). As we increase the number of nodes in the system (and therefore the density), the clustering structure and thus the broadcast remains the same. As a result, the saving increases with the node density.

3. CLUSTERING IN AD HOC NETWORKS

In the previous section we showed that clustering is one of the key approaches to flood overhead reduction. In this section, we elaborate on this important concept. Clustering in wireless ad hoc networks has been investigated in the past in order to enhance network manageability, channel efficiency [8 , 9], and energy economy [10]. Moreover, clustering is indispensable for hierarchical routing or multicasting [11]. However, the clustering schemes proposed so far in the literature are "active". They require a constant refresh rate of cluster-dependent information, and therefore introduce significant background control overhead even if there is no data to send in the network. In some applications, for example, covert military operations and sensor networks, this periodic control traffic is highly undesirable. The penalty introduced by the control traffic (eg, exposure to enemy interception, power consumption, etc) may offset the benefits offered by clustering.

Clustering in ad hoc networks can be informally defined as *grouping of nodes into a manageable set*. Many prior research efforts carried out clustering in different ways. Such efforts started with the DARPA packet-radio network [12]. As a result, the network was dynamically organized into clusters similar to the cluster structure shown in Figure 1.

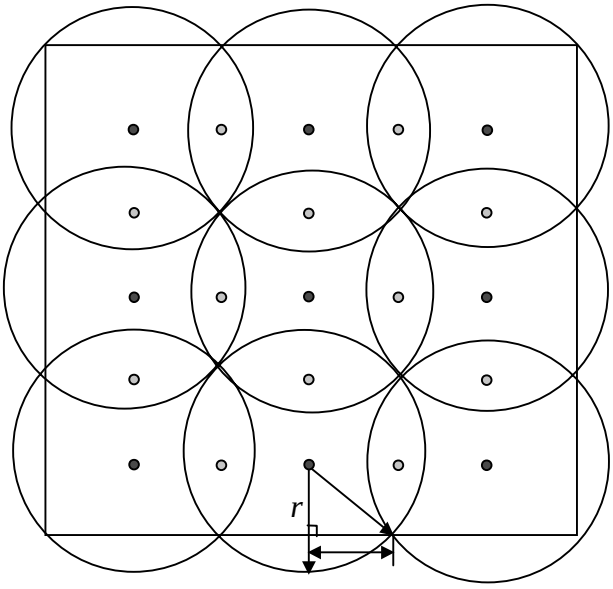

Figure 1. Selective Gateway Flooding Scenario

Several clustering mechanisms have been proposed in the literature [8,9,11,13,14,15]. The schemes reported in [8,11,13] all lead to similar structures (overlapping two-hop clusters with clusterheads). The schemes in [9,14,15] partition the network into disjoint sets of clusters. All of these clustering mechanisms assume prior knowledge of the full neighbor list, or they periodically monitor neighbor information by exchanging explicit control packets. This topology learning overhead is significant if the number of neighbors is large and the topology is dynamic. None of these schemes will work properly with only partial neighbor information.

An important subclass [8,10,11,13] implements two-hop clustering. Two-hop clustering requires that every node in a cluster be reached from another node in the same cluster with at most two hops. Two-hop cluster is a natural clustering structure in ad hoc networks. It only requires direct neighbor information and is easy to construct. The cluster structure in Figure 1 is an example of a two-hop cluster structure.

Two-hop clustering has the following properties:

- There is a clusterhead at the center of a cluster, and the clusterhead can communicate with any node in the cluster with a single hop.
- No clusterheads are directly linked.

- Any two nodes in a cluster are at most two hops away.

Two-hop clustering ends up with a structure similar to the cellular system. There are clusterheads at the center of each cluster (a useful by-product). Nodes belonging to more than one cluster are gateways. The rest of the nodes are ordinary nodes.

3.1 Limitations of Existing (Active) Clustering Schemes

Most clustering algorithms in the past have been studied via simulation and have used the complete neighborhood information. Unlike the simulation environment, accurate global information regarding node locations and adjacency relations is hard to collect in an actual wireless ad hoc network implementation, especially when the node density is high. The major difficulties stem from unreliable and limited link capacity, and from node mobility. Node locations and neighborhood information are key for clustering; unfortunately, they do vary in time. Without the help of a special node - say "oracle" which can listen or talk to all the nodes at the same time - adjacency (neighborhood) information can only be collected by exchanging beacons or hello messages. In this neighbor-learning process, no mobility is generally assumed.

To ensure the correct collection of neighborhood information, existing clustering solutions rely on periodic broadcast of the neighbor list. In the period of neighbor learning and initial clustering, it is essential that there is no mobility for proper convergence. The quasi-stationary assumption must hold during the adjacency information-collecting period, initial clustering, and the re-clustering or clustering maintenance period. If there is motion, we may have to deal with stale neighborhood information during the neighbor-learning period. Moreover, mobility causes adjacency relations to change, which in turn may trigger re-clustering throughout the network. Other drawbacks including isolation (structural disconnection), etc., are listed and explained in [16].

4. PASSIVE CLUSTERING

In this section, we introduce a new cluster formation protocol that is free from the periodic overhead and other limitations discussed in the previous section. This novel approach not only overcomes many limitations of existing clustering mechanisms, but also improves performance and yields new features. Here, we present the concept of passive clustering and illustrate its operation by example. The proof of its correct operation and the detailed description can be found in [16].

4.1 Protocol Overview

Passive Clustering is a cluster formation protocol that does not use dedicated protocol-specific control packets or signals. Conventional clustering algorithms, as earlier discussed, require all of the participating network nodes to advertise cluster-dependent information repeatedly. Moreover, most of the existing clustering schemes require the execution of a separate clustering phase prior to any network layer activity (e.g., routing).

With passive clustering, we avoid all the above limitations. By monitoring user data packets that piggyback some predefined cluster information, we can build impromptu "soft state" clusters for mobile wireless networks. Thus, the cluster infrastructure can be constructed as a by-product of user traffic, without any dependency on the routing protocol, for example.

In passive clustering, each node collects neighbor information from the MAC sender address carried by the incoming packets, and can construct clusters even without collecting the complete neighbor list. This is an innovative approach to clustering which virtually eliminates major cluster overheads - the time latency for initial clustering construction as well as the communication overhead for neighbor information exchanges. Instead of using protocol specific signals or packets, cluster status information (2 bits for four states: Initial, Clusterhead, Gateway, and Ordinary-node states) of a sender is stamped in a reserved field in the packet header. Sender ID (another key piece of information for clustering) is carried by all the existing MAC protocols and can be retrieved from the MAC header. Since in

flooding the MAC packets are transmitted in broadcast (instead of unicast) mode, every node receives and reads the packets (in a promiscuous way), and thus participates in passive clustering.

Note: you cannot perform flooding at the MAC layer because you need to detect duplicates (reading, for example, flood originator ID number which is stored in the packet, not MAC, header). Since passive clustering relies on flooding packets, it may as well be done at the packet layer.

Surprisingly, simulation results show that passive clustering can form better clusters than conventional clustering schemes based on weight (i.e., ID, degree, etc.) information [16]. This is because passive clustering (as used in the support of ad hoc routing schemes) uses network traffic that emanates from sources (i.e., the source in search of a path). If a cluster structure is constructed by a flooding from a single source, the resulting structure is completely immune from logical isolation and lack of connectivity.

Clustering stability and fast convergence time are other important properties required of clustering algorithms. To improve clustering stability and speed up convergence, and most importantly, to avoid the "stationarity" requirement during the neighbor-learning and clustering phase, we developed a new clusterhead election rule which does not require any weight information. We call this rule "*first declaration wins*."

With the *first declaration wins* rule, a node which first claims to be the clusterhead remains the clusterhead and "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 in all the weight-driven clustering mechanisms.

4.2 Operational Description

When a node is ready to become a clusterhead and has packets to send, it declares that it is a clusterhead by stamping its clustering state claim in the packets. Since *passive clustering* does not support explicit control packets or signals of its own, a clusterhead-ready node must postpone its claim until it has outgoing "application" packet-level traffic, for example, flood search packet traffic. After a successful transmission from an aspiring clusterhead, every node within radio coverage learns the presence of the clusterhead by monitoring the "cluster" state of the received packets. At this point, the neighbors of the clusterhead record the clusterhead information (clusterhead ID and the most recent transaction time-timestamp) and change their clustering states as discussed below.

The readiness of being a clusterhead is determined by network activities as well as by the node's clustering state. After a period of inactivity (i.e., no incoming or outgoing traffic for longer than the cluster timeout period), all the nodes revert to the INITIAL state. Only nodes in INITIAL state can be clusterhead candidates – in other words, two hop is the minimum distance between any two clusterheads since all neighbors of a declared clusterhead exit the INITIAL state. After a clusterhead successfully asserts its state, it functions as a clusterhead. Clusterheads collect neighbor information by monitoring the network traffic. They are responsible for relaying intra-cluster packets.

A node that hears more than one clusterhead becomes a GATEWAY. It reverts to ORDINARY node if it does not hear from more than one clusterhead for a given period. In the next section we will describe a slightly modified procedure (selective gateway) in which a part of gateways in this definition also reverts to ordinary node upon hearing a certain number of other gateways. A node that is neither a clusterhead nor a gateway is an ordinary node. The ordinary node does not forward flooding packets. It is precisely this forward-suppression mechanism that reduces flood overhead. Gateway nodes and clusterheads, on the other hand, will keep forwarding the flood packets. Because of the passive nature of the collection mechanism, neighbor information is kept in soft state and is possibly incomplete. Note here again that complete neighbor information is no longer necessary to form the structure. By using timestamps for neighbor information, we preserve the freshness of the information. Ordinary nodes and gateways keep a list of their clusterhead(s) in soft states. The timeout period has to be carefully

chosen based on node mobility and communication pattern. Non-clusterhead nodes can collect their own clusterhead(s) information in a passive way. If a received packet is from a clusterhead (after checking the status information in the packet), non-clusterhead nodes compare the sender ID of the packet with their own clusterhead list and add or refresh accordingly.

5. SELECTIVE GATEWAY PASSIVE CLUSTERING

In typical examples implementing the above basic scheme, one quickly discovers that the number of gateways is quite significant and is typically larger than that of ordinary nodes. Clearly, there is quite a bit of redundancy here, and not all of the gateways have to relay the flooding packets. It is mandatory to reduce the number of gateways in order to achieve efficient flood search packet suppression. Careful gateway selection is the natural solution to improving flooding efficiency.

To select the strictly minimal set of gateways, we would need to collect the clusterhead list for each gateway, and then choose one gateway for each pair of clusterheads. This is another set-cover problem and introduces extra communication and computation overhead since the procedure requires clusterhead list exchanges between gateways. In order to avoid the communication and computation complexity, we introduce a heuristic solution to this problem in the following section.

5.1 Gateway Selection Heuristic

Instead of selecting a single gateway between adjacent clusterheads (two-hops away), we developed a heuristic algorithm that enables a limited number of gateways, and at the same time, preserves adequate connectivity within the resulting cluster structure. The selection algorithm provides many advantages including on-the-fly flooding improvement, redundant connectivity, and higher overall flooding efficiency. The heuristics also allows "distributed gateway" implementations [12].

Every non-clusterhead node monitors and keeps track of the number of clusterheads (NC) and the number of gateways (NG) within range. *Whenever a non-clusterhead node hears a packet from a clusterhead or a gateway*, the node becomes a gateway if $\alpha \cdot NC + \beta > NG$, where α is a coefficient properly chosen based on the desired degree of gateway redundancy ($\alpha \geq 0$) and β is a gateway redundancy factor ($\beta \geq 0$). Otherwise, the non-clusterhead node becomes an ordinary node. The larger the number of clusterheads that a node can hear, the higher the chance to become a gateway. By manipulating α, β, we can control the number of gateways in the system. The larger the number of gateways, the lower the gain in forwarding overhead reduction. On the other hand, if there are too few gateways, connectivity may be impaired leading to a poor network performance. In this paper, α and β are global system parameters and are both set to 1. The values of α and β should be chosen based on considerations including channel quality, noise level, as well as traffic pattern. For that reason, α and β can be local parameters, ie, they can be locally adjusted to provide better adaptability and flexibility. In dense networks where packet collisions abound, higher values of those parameters lead to more gateways and better network performance by distributing network traffic over more gateways. Conversely, in low density we suggest to keep the parameters low to discourage multiple gateway creation. By introducing these heuristics, passive clustering strikes a good balance between clusterheads and gateways and retains only a handful of forwarding nodes for flood search no matter how high the node density is. The gateway selection procedure is fully distributed, and requires only local information. No clusterhead list exchange is required.

5.2 Flooding Improvement On the Fly

Let us consider the example of single-source flooding from a cold start. Every node is in the Initial state, and a source broadcasts a RouteRequest packet. The immediate neighbors of the source receive the packet, and change their state to Clusterhead-Ready. When one of the neighbors is ready to forward the packet, it changes its state to Clusterhead, and broadcasts the RouteRequest packet with the Clusterhead state assertion. This time, all the nodes including the

source that receive the relayed RouteRequest packet from that newly proclaimed clusterhead are eligible to become gateways since they have heard from one clusterhead, and from no gateways (for simplicity, in this case we assume α = 1 and β = 0.) Now, one of the gateways except the source may relay the flood search packet. This relay does not switch any gateways back to ordinary nodes because they still have the number of clusterheads (= 1) which is equal to or smaller than the number of gateways (0 or 1). Let us say that a second gateway within range of the first declared gateway relays the flooding packet. Thereafter, none of nodes in the intersection area of those two gateways can become a gateway – they turn into ordinary nodes after they receive the second flooding packet – their head count for clusterhead equals 1 but they have already 2 gateways. One may notice that there is a chance of critical path loss with these heuristics. However, extensive simulation experiments have shown that the risk of flood delivery failure to certain areas of the network is negligible, even with moderate node density, if the coefficient α and the redundancy factor β are properly chosen [16]. With additional assistance from the routing protocol, we can completely eliminate such "block out" areas.

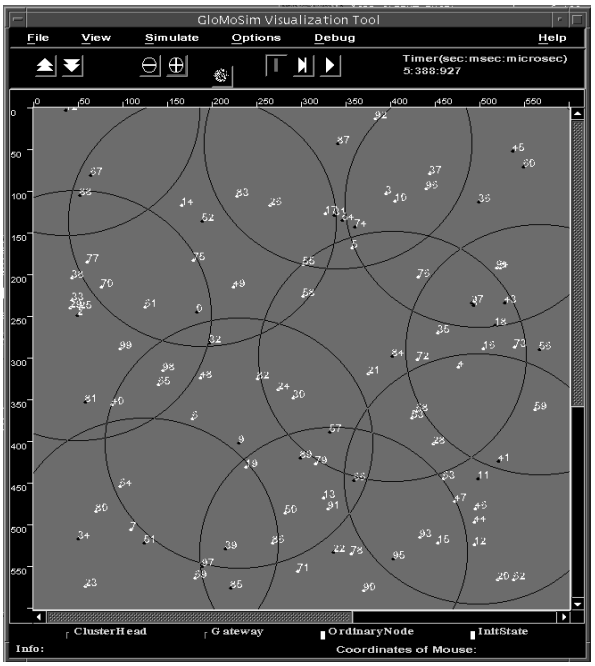

Figure 2. A snap shot of the selective gateway PC

As we have just shown, every node in the intersection between two declared gateways immediately becomes an ordinary node, thus improving flooding efficiency on the fly. In conventional, active clustering approaches, such improvement was possible only after most of the clusters are constructed. Even better, we only allow a few of the gateways in the intersection area. Figure 2 shows a working example of selective gateway passive clustering. This snapshot was taken from an actual simulation experiment with randomly placed 100 nodes in a 600x600 roaming space with 150m of transmission range and ten CBR (Constant Bit Rate) communication pairs which introduce one packet per second each. More detailed simulation environment will be covered in the following section. All the nodes in the system turned out to be well connected by the cluster structure. There were 33 flooding participants (out of 100 nodes). These are the dark nodes in the figure.

5.3 Properties of the Passive Cluster solution

It is appropriate at this point to compare and contrast passive clustering with traditional, lowest ID active clustering. We have already discussed the impact of the background updating procedure and the neighbor list broadcast requirements on the control traffic overhead caused by active clustering. Here we focus on the structure of the solutions. Typically, one finds that the two solutions are comparable (in terms of number and layout of clusters).

Major differences are:

(a) the fact that active clustering is carried out independently, in the background and in parallel across all nodes in the network, while passive clustering is "on-demand" and is initiated by a single "source", namely the first source that needs to send data. Thus, active clustering tends to lead to disconnected islands (which require the "distributed gateway" feature – ie gateway to gateway links - to reestablish connectivity). Passive clustering does not suffer from this problem (albeit it can also be extended to support distributed gateways)

(b) the fact that passive clustering features the "selective gateway" provision. Popular active clustering schemes do not include such feature
(c) the lowest ID feature tends to make the active clustering more sensitive to mobility – the clusterhead can be more easily challenged by newcomers with lower ID

Another important issue is the suitability of Passive Clustering for Low Energy operations, as in battlefield scenarios or sensor network applications. Repeated selection of the same subset of clusters and gateways can be detrimental to low power operation in that it creates uneven energy consumption. In this respect, passive clustering is beneficial. In fact it favors even distribution since at each new cluster formation round (caused by the arrival of a new user data session, say), new clusters and gateways are selected as the source changes and/or, even in the case of same source, the random timers cause different cluster-heads and gateways to assert their role first. In the case of "permanent" traffic pattern where the cluster structure tends to persist, a possible remedy is to associate the cluster-head and gateway status with a minimum energy level requirement. When energy drops below this threshold, the role is given up triggering a new election.

6. SIMULATION STUDY

The simulation models used for the performance evaluation were implemented in the GloMoSim library [17]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation language called PARSEC [18]. The distributed coordination function (DCF) of IEEE 802.11 [19] is used as the MAC layer in our experiments. The radio model uses characteristics similar to a commercial radio interface (e.g., Lucent's WaveLAN). The radio propagation range for each node is 150 meters and channel capacity is 2 Mbits/sec. The roaming space is 600x600 meters square. Each simulation is executed for 10 minutes of simulation time.

Traffic sources are CBR. The source-destination pairs are totally randomized. Data packets are all 512 bytes long. Control packet length is 32 bytes. The random waypoint model [2] was used for node mobility. In this model, a node selects a random target destination within the roaming area and moves towards the destination at a predefined speed. Once the node reaches the destination, it pauses for ten seconds and repeats the process.

We use AODV (Ad hoc On-demand Distance Vector routing) [1] because AODV is one of the most flooding-dependent routing protocols. The only modification we made to AODV was to limit the flood search forwarding function to "non-ordinary nodes." To test the path-finding performance, we ran the following simulations. We first deployed 100 nodes in the simulation space at random. Without node mobility, we chose 2400 random source and destination pairs and ran the selective gateway flood search from cold start one by one. Only one data packet is sent from each source to each destination. There is only one source and destination in a given period (which is much larger than cluster time out (=2 seconds)) to ensure that no residual clustering structure remains after the single transmission. The source finds the destination with the "scoped" selective gateway flood search. A short data packet will be pumped through a path if the path is successfully found. One hundred percent of packet delivery was observed with the experiment.

We also ran a batch of simulations that use a randomly chosen communication pair with varying nodes speeds of 0,2,4,6,8 meters per second. The source sends out a packet every 15 seconds to the destination for 100 minutes (400 tries). The slow packet rate (1/15 packet per second) is to ensure that the cluster structure built by the previous packet delivery dissolves after cluster timeout (2 seconds). With the speed of 0,2 and 4 meters per second, we observed 100 percent packet delivery. 99.25 and 98.25 percent of packet delivery ratio were observed in the case of 6 m/sec and 8 m/sec, respectively. Such packet drop was investigated and traced to route breakage; after a source finds a path, the source cannot deliver the packet because the path was broken in the interim due to the motion.

Passive clustering with selective gateway heuristics can be truly called "on-demand" clustering; if there is no network usage, there is no clustering. As the user starts transmitting data, this protocol deploys clusters as fast as the destination-finding process finds the routes, and thus makes the destination search process more efficient. To validate all the above claims above, we design and run three sets of experiments: (a) node mobility in a moderate node density (100 nodes in the given simulation space), (b) high node density (400 nodes), and (c) without mobility.

6.1 Node Mobility – Low Density

We first compared AODV performance in the usual way. We ran both AODV and the improved AODV with the selective gateway passive clustering (AODV-PC).

We present the following comparisons. The average end-to-end delay, throughput and the average hop counts are visualized with offered loads. The throughput of AODV (see Figure 3 and 4) drops fast after network saturation because the available bandwidth is used mostly by route search. AODV-PC postpones the drop. The more route request packets are flooded, the better the performance of AODV-PC. In light traffic situations (100,200 Kbit/sec), AODV performs slightly better. This is because there is enough capacity for route request flooding, and AODV can find shorter paths (see Figure 5 and 6). Note that the hop count of AODV-PC does not blow up as conventional AODV for higher loads. This means that the path finding mechanism is still working fine, even with a heavy traffic situation. In other words, a source can find a path to a destination even though it cannot send data packets on that path because of saturation. The reason is that the MAC layer keeps a two-level priority queue. Control packets like RouteRequest have higher priority – i.e. no matter how big the data packet queue is, a control packet will always be transmitted.

AODV with selective gateway passive clustering introduces only a fraction of RouteRequest packets into the network. This is the reason why the route-search mechanism worked fine in the simulation. The control traffic was not high enough to saturate the network. This is very useful for live reports in many applications, and also for network management. Short control frame flooding is still available to manage the network.

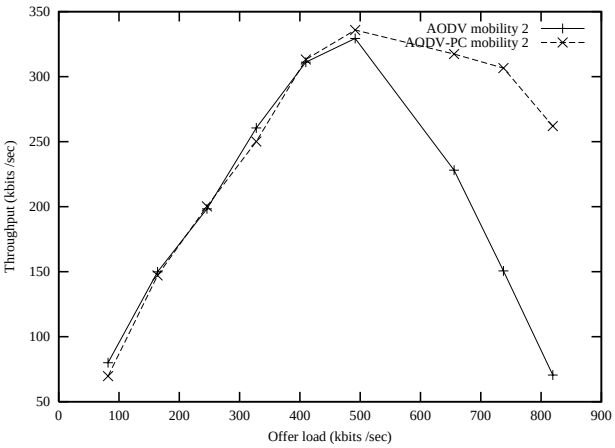

Figure3. Throughput Comparison (Mobility 2 m/sec)

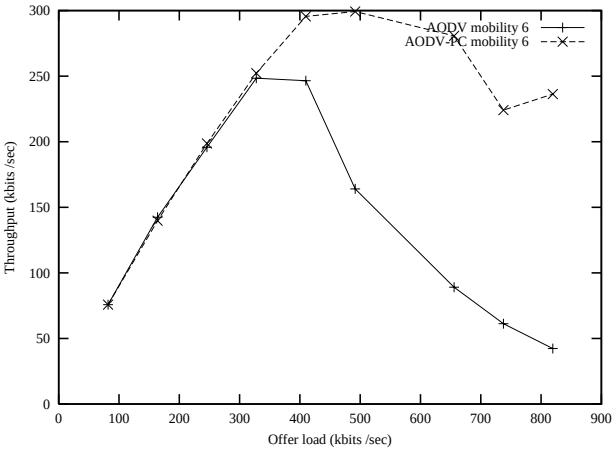

Figure 4. Throughput comparison (mobility 6 m/sec)

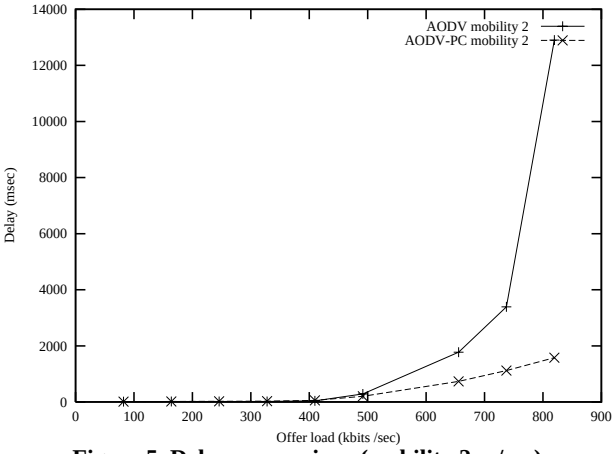

Figure 5. Delay comparison (mobility 2 m/sec)

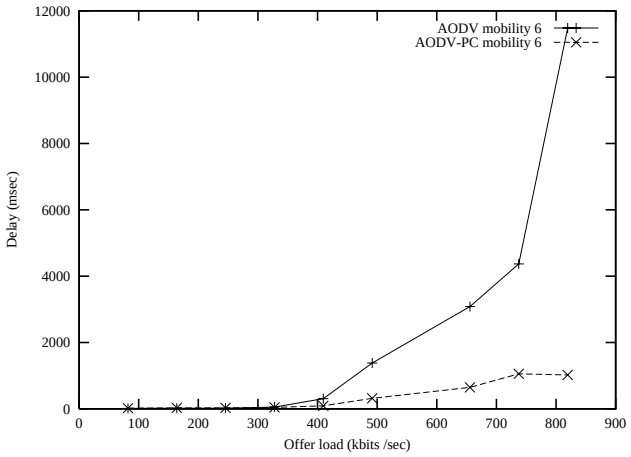

Figure 6. Delay comparison (mobility 6 m/sec)

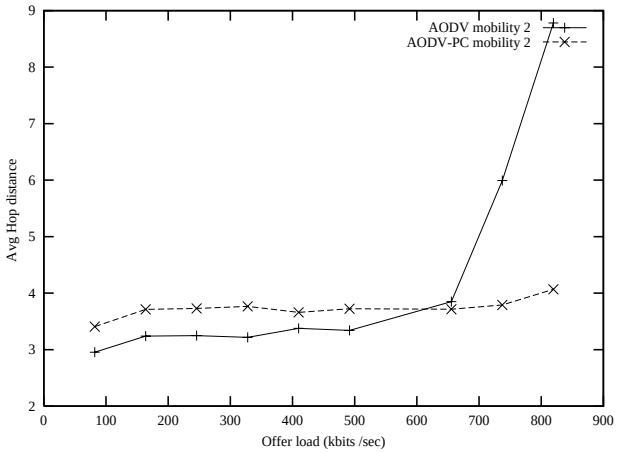

Figure 7. Hop count comparison (mobility 2 m/sec)

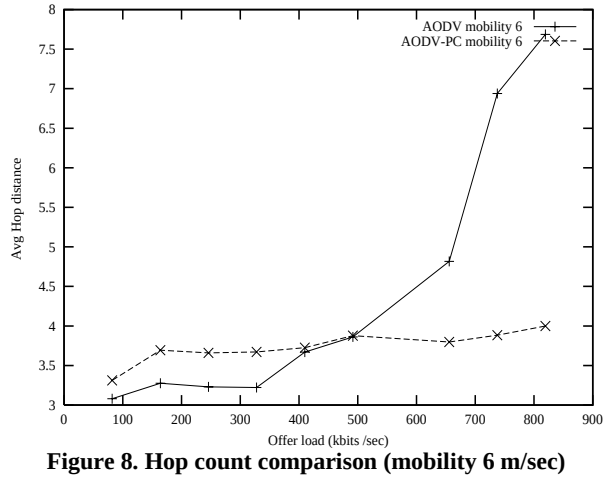

Figure 8. Hop count comparison (mobility 6 m/sec)

6.2 Node Mobility – High Density

In this section, we add 300 more nodes (ie, 400 nodes in total) in the same environment described in Section 6.1. This time we also report the system-wise energy consumption caused by transmission. In the experiments, we assume the following energy consumption rates which are modeled from a Wavelan product.

Radio Mode	SLEEP	RECEIVE	TRANSMIT
(mJ)	0.18	1.48	3.0

Table 1. Power consumption in radio status

We ran simulations at various speeds (0,2,4,6,8 m/sec). In a static environment (mobility 0), the network performance is dependent on the pair selection. Even in the static case, clustering demonstrates better throughput. But the clustered approach experiences slightly higher hop counts, which result in slightly larger power consumption. Note that there is no more flood search once the route is found.

When mobility is introduced, we note that AODV-PC performs far better than AODV. Moreover it improves with higher node mobility. We report here only the results with mobility of 6 m/sec since all the mobility ranges except zero motion show very similar trends. A full set of results is available in [20].

In the dense network, the advantage of clustering is thus obvious. This is because passive clustering is not affected by node density.

Next, (Figure 12) we compare the number of RoutingRequest packet relays to see how good the selective gateway heuristic is. The mobility for the comparison is set to 6 meters per second, and we collect the number of RouteRequest packet relays. With the result, we can see how much of flooding overhead is saved by the selective gateway heuristic.

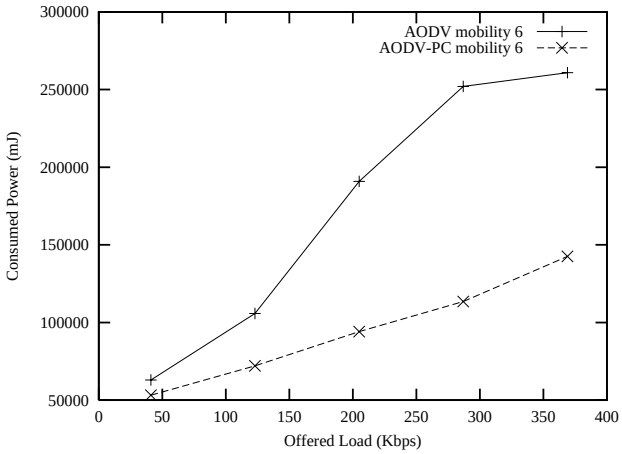

Figure 9. Consumed Power

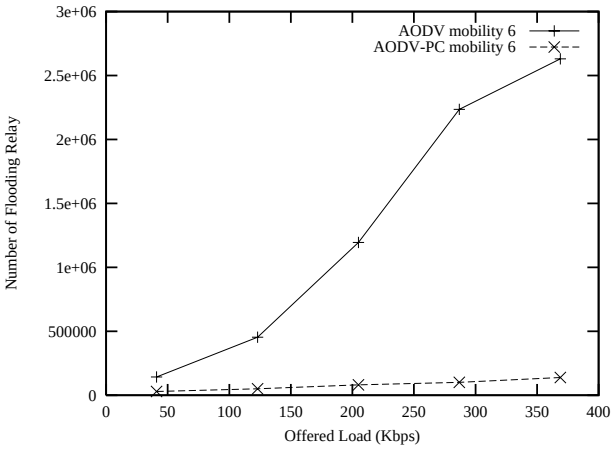

Figure 12. Number of Flooding Relay

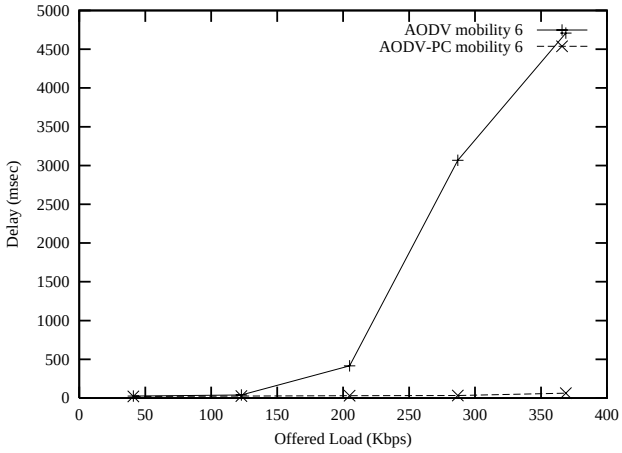

Figure 10. Delay

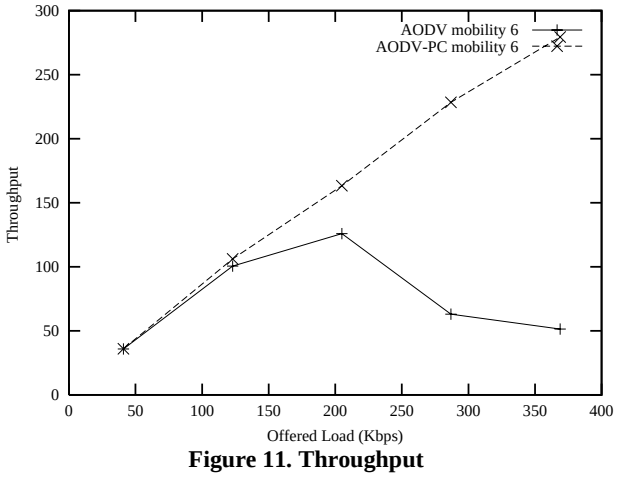

Figure 11. Throughput

The selective gateway heuristic keeps relatively slow increase in the number of flood packet relays. We cannot observe network saturation in the case of AODV-PC in the range of given offered loads. From the simulation results, we can conclude that selective gateway passive clustering can save significant amounts of flooding control overhead.

6.3 No Mobility - Dynamic Traffic

In this experiment, we freeze the node positions, and inject short sessions with bursty traffic. The packet rate is 0.4 packet per second, 3 packets per session. A given number of new source and destination pairs are selected to participate in such bursty communication every 3 seconds with randomized starting times.

This simulation tests the path-finding capability of both AODV and AODV with selective gateway passive clustering in various network load situations. Because there is no mobility, there is no packet delivery loss due to a path break. This scenario is very similar to that of a sensor network where all the nodes are fixed and the communication patterns are short and bursty.

Figure 13 shows packet delivery ratio as a function of number of communication pairs. AODV with the selective gateway passive clustering out-performs conventional AODV in the whole range of the simulation window.

This does demonstrate the effectiveness of the cluster structure in reducing flood redundancy. The selective gateway passive clustering finds paths well, and at the same time, it reduces interference of multiple flooding searches by limiting flood packet relays.

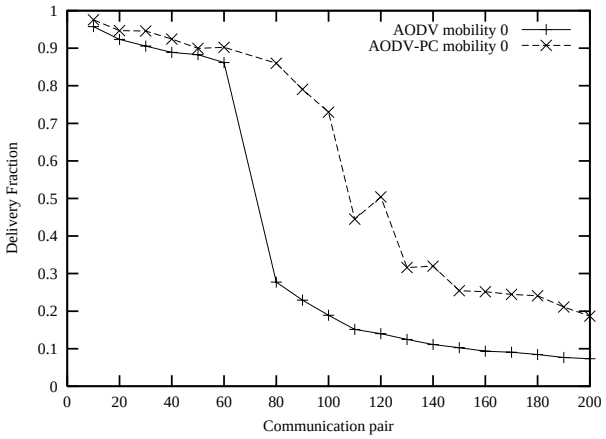

Figure 13. Delivery ratio

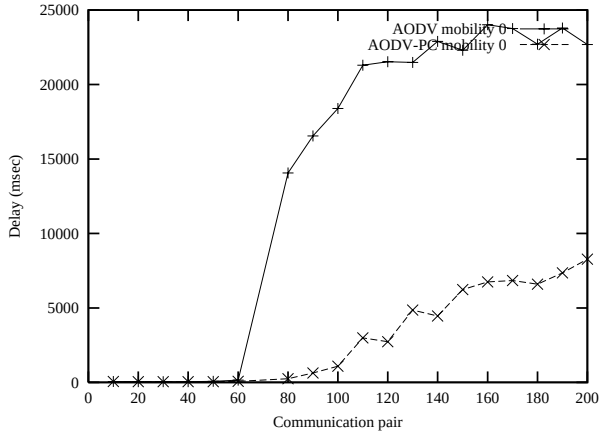

Figure 14. Delay comparion

7. CONCLUSION

We have presented a new clustering algorithm for efficient flooding in ad hoc networks. For efficient flooding, we propose to superimpose an on-demand cluster structure which can be quickly deployed in the "*unstructured*" ad hoc network, and let only non-ordinary nodes (clusterheads, gateways, "initial state" nodes) participate in the flooding process. Due to its passive nature, passive clustering does not introduce any control packets dedicated to the protocol. In other words, it is "control overhead free". Thus, it can reduce the cost of flood search significantly without introducing any line overhead. Even better, there is no preparation time or overhead for selecting dominant sets. As the results, the number of flooding relays can be significantly reduced even during the first flooding. This is the unique feature and strongest advantage of the proposed mechanism. It is especially useful for ad hoc networks with high mobility. The gateway selection scheme is density-adaptive. Its efficiency increases linearly with the number of neighbors, ie, with node density. Beside assisting with flood reduction, the clustering structure offers several other side benefits. In particular, it can be beneficial to routing scalability, reliability and QoS support. Passive clustering is a self-sufficient clustering scheme. The protocol collects all the necessary information itself and does not require costly information like global topology knowledge from the lower layer. The resulting cluster structure is superior to any existing clustering algorithm in terms of stability, mobility robustness and connectivity. Passive clustering can build the cluster structure with partial neighbor information which, in most cases, is the only possible information available in an ad hoc network. In many areas including military applications (e.g. SensIT), this feature has merit since it permits to build the clusters without releasing network topology details to eventual evesdroppers.

REFERENCES

[1] Das, S.R.; Perkins, C.E. and Royer, E. M., *Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks*, In Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, Mar. 2000.

[2] Johnson, D. B., *Routing in Ad Hoc networks of mobile hosts*, Proc. Of Workshop on Mobile Computing and Applications, Dec. 1997

[3]Qayyum, A.; Viennot, L. and Laouiti, A.. *Multipoint relaying: An efficient technique for flooding in mobile wireless networks*. INRIA research report RR-3898, 2000

[4] Lee, S. and Kim, C. *Neighbor supporting ad hoc multicast routing protocol.* Proceedings of First Annual Workshop on Mobile Ad Hoc Networking Computing. Piscataway, NJ, USA: IEEE, 2000. p.37-44.

[5] Lim, H. and Kim, C., *Flooding in wireless ad hoc networks,* Computer Communications, vol.24, (no.3-4), 2000.

[6] Chen, B., Jamieson, K., Balakrishnan, H. and Morris, R., *Span: An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks*, In Proceedings of ACM/IEEE MOBICOM 2001, Rome, Italy, 2001.

[7] Xu, Y., Heidemann, J. and Estrin, D., *Geography-informaed Energy Conservation for Ad Hoc Routing*.. In Proceedings of ACM/IEEE MOBICOM 2001, Rome, Italy, 2001.

[8] Gerla, M and Tsai, J., *Multicluster, mobile, multimedia radio network*, ACM-Baltzer Journal of Wireless Networks, Vol.1, No.3, pp.255-265(1995)

[9] Lin, C.R. and Gerla, M., *Adaptive Clustering for Mobile Wireless Networks*, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, Sep. 1997, pp.1265-1275.

[10] Kwon, T.J. and Gerla, M., *Clustering with Power Control*. Proceedings of MILCOM 1999, Atlantic City, NJ, Oct. 1999.

[11] Chiang, C.-C.; Gerla, M. and Zhang, L. ,*Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks*, ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, *vol. 1, no. 2, 1998*

[12] A.Ephremides, J.; E. Wieselthier and D.J. Baker, *A design concept for reliable mobile radio networks with frequency hopping signaling*, Proc. IEEE 75(1) (1987), pp.56-73

[13] Basagni, S. *Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks*.. VTC, Proceedings of IEEE VTS 50th Vehicular Technology Conference Piscataway, NJ, 1999. p.889-93 vol.2.

[14] Krishnan, R.;. Ramanathan, R. and Steenstrup, M., *Optimization algorithms for large self-structuring networks***.** Proceedings of IEEE INFOCOM '99, Piscataway, NJ (21-25 March 1999.), p.71-8 vol.1.

[15] McDonald, A.B. and Znati, T.F., *A mobility-based framework for adaptive clustering in wireless ad hoc networks.* IEEE Journal on Selected Areas in Communications, Aug. 1999. p.1466-87. vol.17, (no.8)

[16] Kwon, T.J., *Energy Efficient Clustering in Ad Hoc Networks*, Ph.D. Thesis, Department of Computer Science in UCLA, 2000.

[17]Takai, M.; Bajaj, L.; Ahuja, R.; Bagrodia, R. and Gerla, M., *GloMoSim: A Scalable Network Simulation Environment,* Technical report 990027, UCLA, Computer Science Department, 1999.

[18] Bagrodia, R.; Meyer, R.; Takai, M.; Chen,Y.; Zeng, X.; Martin, J. and Song, H.Y. *PARSEC: A Parallel Simulation Environment for Complex Systems*, IEEE Computer, vol. 31, no. 10, Oct. 1998, pp.77-85.

[19] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[20] Kwon, T.J.; Yi, Y.J. and Gerla, M., *Experiments on Passive Clustering in high node density*, Technical Report 200039, UCLA CSD 2001.