Energy and Mobility Aware Clustering Technique for Multicast Routing Protocols in Wireless Ad Hoc Networks

E. Astier¹, A. Hafid¹, A. Benslimane²
¹ Network Research Lab, University of Montreal, Canada
{eastier, ahafid}@iro.umontreal.ca
² Laboratoire d’Informatique d’Avignon, Université d’Avignon, France
benslimane@lia.univ-avignon.fr

Abstract: A number of key issues arise in the implementation of scalable multicast protocols for wireless mobile ad hoc networks (MANETs), namely energy consumption and data delivery over unstable/mobile nodes. To improve scalability of these protocols, clustering has been proposed. Clustering allows reducing the number of mobile nodes participating in multicast routing algorithms, which in turn significantly reduces the routing-related control overhead. In this paper, we propose a clustering algorithm, called RSIDS (Restful Stability based Insomnious Distributed Sensors), which considers both stability and residual energy of neighboring nodes when selecting critical nodes (i.e. cluster heads and gateways). RSIDS uses Passive Clustering (in opposition to active clustering) to form the clustering structure. The critical nodes selection enables the selection of most stable nodes with high residual energy as critical nodes; the goal is to minimize re-clustering (and thus re-branching for multicast protocols) that may generate considerable overhead and packet losses and increase the lifespan of the network. We show, via simulations, that RSIDS outperforms existing clustering schemes, in terms of packet delivery ratio and network lifetime, when used with the MAODV (Multicast Ad hoc On demand Distance Vector) routing protocol.

Key Words: MANETs, Multicast Routing Protocols, Mobility, Cluster, Network Lifetime, Energy Aware, Stability. MAODV.

I. INTRODUCTION

Ad-hoc networks are infrastructure-less, dynamically reconfigurable wireless networks that consist of nodes that act as routers and have different power constraints. In such an environment, we are facing the problem of providing a multicast routing protocol capable of handling host mobility and the various power restrictions of the nodes.

During the last few years, several approaches have been proposed to improve multicast communication in mobile environments. Depending on how the routes connect the multicast members with each other, we can basically distinguish four categories of protocols [3]; namely Meshed-based, Tree-based, Hybrid and Stateless multicast approaches. Some protocols allow data packets to be transmitted over more than one link by creating a mesh covering all group members to increase robustness with the price of putting more redundancy in data transmission. On the other hand, tree-based approaches offer efficiency aiming at reducing the network load along with the overhead of duplicated packets and their ensuing collisions. Source or shared-tree based methods, however, lack robustness in dynamic environments. It must be acknowledged that this efficiency/robustness tradeoff raises key issues in ad hoc multicasting. Hybrid solutions aim to achieve better performance by combining the advantages of both tree and meshed-based approaches.

Nevertheless, all these flat routing schemes have been shown to have limited scalability, due to their route discovery and maintenance procedures [1, 7]. Stateless multicast approaches focus on small multicast groups only.

To improve scalability of these protocols, clustering has been proposed. Clustering allows reducing the number of mobile nodes participating in routing algorithms (including multicast routing), which in turn significantly reduces the routing-related control overhead. Indeed, only cluster heads and gateways (called critical nodes) forward traffic, and therefore are part of the forwarding routes which allow nodes to reach each other. We combine clustering with tree-based multicast routing protocols like MAODV [2] by allowing all nodes to be at the originating point or receiving end of a multicast tree, but only allowing the critical nodes of the clusters to make up the routing nodes connecting them. Mobility aware clustering algorithms like MOBIC [5], form stable clusters using critical nodes with low relative speed to each other in order to minimize the probability of re-clustering. However, these critical nodes perform extra work and can easily become single points of failure as they die early because of excessive energy consumption. This may cause network partition and communication interruption. Hence, it is also important to balance the energy consumption among the mobile nodes. In energy aware clustering algorithms like GRIDS [8], energy is balanced by alternating the status of critical nodes that perform extra work and non-critical nodes that have more energy left. However, this may cause additional re-clustering which increases overhead.

The usefulness of multicasting for group-oriented applications can be compromised in MANETs if we do not envision the use of a clustering scheme (a hierarchical routing algorithm) considering both nodes’ mobility and residual power. To the best of our knowledge, there is no passive clustering scheme, in the open literature, which combines both metrics to overcome MANET limitations. This paper addresses the problem of designing energy and mobility aware clustering algorithm. Our motivation comes from the fact that an energy-balancing clustering algorithm is promising, when applied to a tree-based multicast routing protocol, only if we take into account the robustness (i.e. stability) of the routes. We make use of passive clustering [4] to eliminate clustering overhead (in opposition to active clustering), opportunistic rest periods for critical nodes to eliminate the “early die” problem, and a stability metric to reduce re-clustering and thus re-branching of the multicast tree structure. We evaluate the proposed clustering scheme when used with MAODV [2] to support multicast and compare it with 4 other schemes.
The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 describes details of the proposed clustering scheme. Section 4 demonstrates the effectiveness of the scheme via simulations. Section 5 concludes the paper.

II. RELATED WORK

Mobility is a prominent characteristic of MANETs and is the main factor affecting topology changes and routes’ invalidation. Thus, it is important to take the mobility metric into account in the construction of clusters in order to form a stable cluster structure. Mobility-aware clustering indicates that the cluster structure is computed based on the mobility behavior of network nodes. The basic idea is that by grouping mobile nodes with low relative speeds into the same cluster, the intra-cluster links become more tightly connected and thus the re-clustering rate naturally decreases.

In the cluster formation phase of MOBIC [5], each mobile node sends two consecutive messages to each of its direct neighbors to help that neighbors compute their relative speeds. Then, each mobile node calculates its own aggregate local mobility and broadcasts this information to its neighbors. Also, since MOBIC has an overlapping cluster structure, a mobile node may broadcast more than one cluster-related message (cluster-related status) during the cluster formation procedure. The downside is the need for extra explicit message exchanges among mobile nodes for maintaining the cluster structure. When network topology changes frequently, it results in frequent cluster topology updates, and the control overhead for cluster maintenance increases drastically. This maintenance may consume a large portion of the network bandwidth, drain mobile nodes’ energy quickly, provoke collisions and congestions, and override its improvement of the network scalability and performance. Hence, it is important to reduce the communication overhead caused by cluster maintenance.

Passive Clustering (PC) [4] is a clustering protocol that does not use dedicated clustering-protocol-specific control packets; it constructs and maintains cluster architecture based on data traffic forwarding. PC is suitable for a dense network with high mobility, where mobile nodes’ continuous movement greatly affects the cluster topology. This is because the cluster maintenance of PC is traffic-dependent and immune from increased control overhead caused by frequent changes of cluster structures. PC does not make use of mobility or energy metrics which leads to critical nodes using more energy shortening the network lifetime; indeed, PC suffers from the “early die problem”. GRIDS (Geographically Repulsive Insomnious Distributed Sensors) [8] builds upon Passive Clustering and extends the lifespan of the network by using an efficient selection mechanism of critical (or not) nodes. GRIDS enables balanced energy consumption among the network nodes. Each node determines being insomnious or not based on its residual energy and the number of neighbouring insomnious nodes and their energy level.

In GRIDS, an energy abundant node can challenge cluster head and usurps the role. GRIDS uses the number of Cluster Head neighbors and the number of gateway neighbors to determine the next status of a node when a status change condition is met. However, the frequent status changes of critical nodes forces forwarding routes (and thus multicast tree structures when used with a multicast protocol) to be recreated.

In [9] the authors propose a heuristic (called ILBH and a derivative 3-ILBH) that can be used with PC in order to balance energy consumption of the network nodes; the goal is to balance energy among nodes and thus to extend the network lifetime. Two thresholds $\alpha$ (battery_capacity) and $\beta$ (battery_capacity) ($0 < \alpha < \beta < 1$) are defined so that when a node reaches the first threshold, it changes its state to “sleep” until it reaches the second threshold. During this time interval, the node decreases its listening and reception time to balance energy consumption and becomes an Ordinary Node. This compulsory status change will force the reconstruction of clusters and thus of multicast trees.

MOBIC uses active clustering, which creates additional control packets and consumes greater amount of energy than PC. However, in PC, Cluster Heads and Gateways work more and lose power faster and therefore die earlier than other nodes causing an “early die” problem. GRIDS eliminates this problem by using nodes with the most residual energy as the critical nodes. It changes the status of nodes based on energy but does not take into account the stability of the nodes in the cluster. 3-ILBH helps balancing energy consumption but can also force forwarding routes to be reconstructed when CHs and/or GWs are forced to sleep. In this paper, we propose an algorithm that addresses the problems of existing clustering schemes by making use of passive clustering to eliminate additional control packets, opportunistic rest periods for critical nodes to eliminate the “early die” problem, and a stability metric to reduce the recreation of forwarding routes. When used with MAODV it reduces re-branching the multicast tree structure required to deliver multicast data packets. We also make use of a derived version of 3-ILBH in order to avoid unnatural status changes and preserve established routes and thus multicast trees.

III. RSIDS PROTOCOL DESCRIPTION

In this Section, we present the details of the proposed clustering scheme called RSIDS (Restful Stability based Insomnious Distributed Sensors). A node can be in one of the following 5 states: Initial, Cluster Head (CH), Cluster Head Ready (CHR), GateWay (GW), or Ordinary Node (ON). A CH is a node that is the center of a cluster of nodes with a radius the length of the farthest node that can still receive packets from it. A GW is a node that can communicate with multiple CHs. ON is a node within the cluster that is not a CH or a GW. An Initial node is a node that has not heard from any neighboring nodes. A CHR node is a node that has not heard from any CHs and is ready to send a message. These last two states are temporary.

CHs and GWs can both forward packets, while ONs do not. This leads to CHs and GWs using more energy than ONs due to their increase use as forwarding route nodes. It is important to reduce the amount of energy spent in order to prolong the life of the network nodes and the ability of the network to communicate. To achieve a lower level of energy consumption, we use passive clustering instead of active clustering to create the network clusters. This eliminates the
need for additional control packets and completely eliminates the maintenance phase of active clustering which produces additional hello packets.

A. Stability Metric

To compute the stability of a node \( nd \), we start by finding out whether a neighbor \( nb \) is coming closer or moving away from \( nd \). The distance \( d \) to \( nb \) can be estimated as

\[
\frac{1}{\sqrt{(RxPr/TxPr)}}
\]

where \( RxPr \) and \( TxPr \) correspond to the receiving (by \( nd \)) signal strength and transmission signal strength respectively. An exact calculation of the distance may not be possible due to the difficulties measuring the transmission signal strength (involves accurate channel modeling). However, the ratio of \( RxPr \) from two successive packet transmissions can determine whether \( nb \) is moving closer or farther to \( nd \). The relative mobility of \( nb \) relative to \( nd \) is defined as

\[
M_{nb}^{rel}(nd) = 10 \log_{10} \frac{RxPr_{new}}{RxPr_{old}},
\]

where \( RxPr_{old} \) (\( RxPr_{new} \)) is the strength of the first (second) signal received by \( nd \) from \( nb \); a negative value indicates \( nd \) and \( nb \) are moving apart, and a positive value indicates \( nd \) and \( nb \) are moving closer to each other.

Stability \( S \) is defined as the variance to zero \( E \) of the set of relative mobility values of all the neighbors of \( nd \):

\[
S = E_{(for\ all\ nb)} [(M_{nb}^{rel}(nd))^2] [5].
\]

The lower the value, the greater the stability of a node. The objective is to select critical nodes that have neighbours that remain close to them or are moving toward them.

B. Cluster Formation

Table 1 shows the transitions of the possible status changes of the nodes. The goal is to select high energy CHs and stable GWs. All nodes maintain a soft-state (i.e. expires) list of CHs and GWs that they can overhear. A Node starts Initial and becomes CH Ready if they hear from a Node that is not a CH; otherwise, it becomes a GW. In the initial cluster creation a CH may be surrounded by GWs, but this period does not last long as a GW becomes an ON if it hears from a node with greater stability than its own; this allows demoting less stable nodes to ONs and promoting more stable nodes to GWs. A GW may also change to Initial if it does not hear from a CH for a period of time (Cluster Head Timeout).

A CH Ready node becomes a CH upon sending successfully a packet before hearing from any CH with greater residual energy. Otherwise, a CH Ready node becomes a GW if the stability of the CH it heard from is less than its own, or an ON if it is not. When two CHs get within range of each other, CH contention occurs. The CH with the greatest amount of residual energy maintains its status and the other CH becomes an ON unless it is highly stable and becomes a GW. If a CH loses its status in a CH contention and becomes an ON it remains an ON until its residual energy is greater than all of its neighbors: it is “resting”.

This helps balancing energy consumption and allows an ex CH to rest opportunistically and not be able to become a CH or a GW immediately like in the Passive Clustering or GRIDS algorithms.

\[
\]

If (Node Energy <= 0)
Then Node State => Dead

If (Node is Resting and Node Energy >= Max Neighbor Energy)
Then Node Stops Resting

Switch (Node State):

Case Initial:
If (Incoming Neighbor Node State != Cluster Head)
Then Node State => Cluster Head Ready
Else Node State => Gateway

Case Cluster Head Ready:
If (Incoming Neighbor Node State == Cluster Head &&
Incoming Neighbor Node Energy > Node Energy)
Then If (Node Stability < Incoming Neighbor Node Stability)
Then Node State => Ordinary
Else Node State => Gateway

Case Cluster Head:
If (Incoming Neighbor Node State == Cluster Head &&
Incoming Neighbor Node Energy > Node Energy)
Then If (Node Stability < Incoming Neighbor Node Stability)
Then Node State => Ordinary
Else Node State => Gateway

Case Ordinary:
If (Node Stability == Incoming Neighbor Node Stability and Node is Not Resting)
Then Node State => Gateway

Case Gateway:
If (Node Stability < Incoming Neighbor Node Stability) Then
Node State => Gateway

End Switch

If (Node Sends Packet && Node State == Cluster Head Ready)
Then Node State => Cluster Head

If (Cluster Head Timeout)
Then Node State => Initial

Table 1: Pseudo Code

An ON becomes a GW if it is not resting and it hears from a node with smaller than or equal stability to its own; this allows highly stable nodes to become critical nodes. An ON may also change to Initial if it does not hear from a CH for a period of time defined by a Cluster Head Timeout.

To implement the proposed cluster formation procedure, we modified the routing protocol, under consideration, PDU to include two new fields: residual energy and stability values of the sending node.

C. Energy Balancing

To further balance the energy consumption among the nodes we integrated a modified version of the 3-ILBH heuristics. We consider 3-ILBH that uses 6 thresholds (\( \alpha_1:0.2, \beta_1:0.3, \alpha_2:0.4, \beta_2:0.5, \alpha_3:0.6, \beta_3:0.7 \)) [9] and thus allows balancing energy consumption over 3 intervals ([\( \alpha_1, \beta_1 \]), [\( \alpha_2, \beta_2 \]), and [\( \alpha_3, \beta_3 \)]). A node that reaches a consumption of \( \alpha_i^{\ast} \) battery_capacity (e.g., consumes 20% of its total capacity), is forced to a “sleep” state until its energy consumption reaches \( \beta_i^{\ast} \)battery_capacity (e.g., consumes 30% of its total capacity); then, its state can change to CH, GW or ON following the PC normal operation.

In this paper, we modify 3-ILBH as follows: if a CH or GW reaches a consumption of \( \alpha_i^{\ast} \) battery_capacity, it is not forced to sleep/rest; it will be put in “sleep” state until (a) it gives up its role of GW or CH (based on the operation of...
RSIDS); or (2) it reaches a consumption of $\beta_i \cdot \text{battery_capacity}$. In this case, the rest period starts from the time, $T$, when one of these conditions is satisfied and ends when the node consumes $(\beta_i - \alpha_i) \cdot \text{battery_capacity}$ starting from $T$.

The proposed 3-ILHB modification prevents forcing CHs and GWs to give up their roles to enter “sleep” state at fixed intervals; changing from CH/GW to “sleep” state causes re-clustering/re-branching and thus overhead and data losses. Our proposal allows CHs/GWs some flexibility in when they enter the rest period and allows the critical nodes to rest opportunistically instead of at fixed intervals. Indeed, it helps reducing transitions from CHs/GWs to “sleep” state.

D. Clustering Example

Figure 1 shows a clustering structure computed by the network nodes when running GRIDS. We can see that two pairs of joining clusters (C1 and C2, C3 and C4) have the choice between two possible GWs each (N1 and N2, N3 and N4). GRIDS would choose the nodes with more energy (N2 and N4), ignoring the mobility factor. This structure is clearly not stable as nodes N2 and N4 are moving fast and in different direction than the clusters. In GRIDS, the cluster structure is only governed by energy; stability is not taken into account. Figure 2 shows how RSIDS would have clustered this scenario. The chosen GW nodes (N1 and N3) have less energy but greater Stability and will remain with the cluster longer reducing the need to recreate the forwarding routes, to reconfigure the tree and therefore reducing control packets and collisions.

IV. SIMULATION MODEL AND METHODOLOGY

In this Section, we present the simulation results evaluating the proposed scheme, and other clustering schemes (MOBIC, GRIDS, PC, 3-ILBH), when used by MAODV for multicast routing. We start by briefly describing MAODV and then present the evaluation analysis.

A. MAODV

In MAODV, three tables are needed for routing: a Unicast Routing Table, a Multicast Routing Table and a Group Leader Table; more details on these tables can be found in [10]. When a node in MAODV creates unicast routes it sends Route Request (RREQ) packets which are responded to with Route Reply (RREP) packets when a route is found to create the forwarding route. When a node wants to join a multicast group it broadcasts a Route Request Join packet which are responded to with Route Reply Join packets when a route is found to create the forwarding route and populate the Unicast Routing Table. When a node joins a multicast group it broadcasts a Route Request Join packet which are responded to with Route Reply Join packets when a route is found to create the forwarding route and populate the Unicast Routing Table. When a node joins a multicast group it broadcasts a Route Request Join packet which are responded to with Route Reply Join packets when a route is found to create the forwarding route and populate the Unicast Routing Table.

B. Simulation environment

We used NS-2 (version 2.33) as the simulation platform. Our simulation models a MANET of 100 mobile nodes placed randomly within a 600m x 600m area. As the simulation starts each node randomly picks a new destination and travels toward it. Upon reaching the destination it randomly picks a new destination and travels toward it. Two scenarios are presented. In the first scenario, the nodes travel at a speed of 6m/s; each of 5 sender nodes sends 2 packets per second from time 30 seconds until they either run out of energy or reach the end of the simulation at 1800 seconds. The packets are Multicast CBR traffic of 512k to a multicast group of 20 receiving nodes. In the second scenario, the nodes travel at a speed of 10m/s; each of 10 Sender nodes sends to a multicast group of 10 receiving nodes. All Nodes start with 100J energy. Hellos are sent 0.75 to 1.25 seconds apart and the Cluster Head Timeout was set to 3.75 seconds, for a maximum of 3 missed hellos.

C. Metrics

In our simulations, we consider the following metrics:

- Packet Reception Ratio: The ratio of the total number of data packets actually received versus the total number of data packets supposed to be received. This number presents the effectiveness of a protocol.
- Periodic Packet Reception Ratio: The ratio of the number of data packets actually received versus the number of
data packets that were supposed to be received at the end of each 15 second interval. This number presents the effectiveness of a protocol for each interval.

- Residual Energy: The average amount of energy the nodes have at different times during the simulation. We use it to compare the energy efficiency of various protocols.
- Dead Nodes: The number of dead nodes at different times during the simulation. We use it to compare network lifetime using various schemes.

Figures 3-4 show the running total of data packet reception measured at the end of every 15 seconds over the lifetime of the simulation compared to the theoretical maximum of 200 packets/s. (2 packets/s. X 5 sending node X 20 receiving nodes.) We start at the end of 45 seconds since no data packets are transmitted for the first 30 seconds of the simulation to allow MOBIC time to cluster. We measure the number of data packets that were received per interval. It is worth noting that packets sent during interval \( I_t \) may be not counted, when computing the reception ratio, if they are received during the next interval \( I_{t+1} \); this explains smaller delivery ratio values at the beginning of the simulations (Figs 3-4). At the end of the first simulation RSIDS delivers 28% more than MOBIC, 26% more than PC, 23% more than GRIDS, and 19% more than 3-ILBH.

Figures 5-6 show the periodic packet reception ratio at the end of 15 second intervals. The figures show large drops in reception for all algorithms due to the performance deterioration of the MAODV protocol when used in conjunction with multiple senders and mobile nodes [2]. In our first scenario if one sending node cannot broadcast (e.g., out of range) to the rest of the network, it can impact up to 20% of the packet reception (we have 5 senders), if one receiving node is not reachable by the rest of the network (e.g., out of range) it can affect up to 5% of the packet delivery (we have 20 receivers). When periodic reception drops off sharply toward zero, it means nodes are dying (i.e., running out of energy), which does not start for RSIDS until around 990 seconds. The last packets received for RSIDS are at 1380 seconds compared to 945 for 3-ILBH, 885 for GRIDS, 585 for PC and 495 for MOBIC. PC and MOBIC stop transmitting early because they do not take energy into account when choosing their critical nodes. Data Packet reception ends when either all the multicast source nodes/sink nodes die or when sufficient number of nodes (not belonging to the multicast group) die and no routes are available to forward data packets. RSIDS is able to maintain data reception above 80%, 11% longer than GRIDS, 13% longer than 3-ILBH, 40% longer than PC, and 52% longer than MOBIC.

Figures 7-8 show the average remaining energy of the nodes at 15 second intervals throughout the simulation. RSIDS was able to deliver more packets because its energy was conserved at a higher rate than the others. From the graph, the rebalancing period of 3-ILBH can be seen (if we take the derivative of the 3-ILBH curve, there are three periods where the rate of changes becomes lower; i.e. the curve is not sharply decreasing, the nodes are saving energy). We cannot observe the same tendency in RSIDS because nodes enter this period staggered by their cluster status rather than at fixed periods.

In Table 2, the first node death signifies the start of the decay in packet reception and the beginning of network death, 0% packet rate reception signifies total network...
death. RSIDS outlasted all others in both metrics. We use these metrics instead of absolute depletion of network node energy because once the majority of the sending and receiving nodes die, the rest of the nodes spend little energy and can survive until the end of the simulation as no data is being transmitted and their radios only send and receive occasional HELLO packets.

Figure 7 - Remaining Energy (5 to 20, 6m)

Figure 8 - Remaining Energy (10 to 20, 10m)

Table 2: Node Deaths and Network Death

<table>
<thead>
<tr>
<th></th>
<th>5-20, 6m/s</th>
<th>10-20, 10m/s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st Node Death</td>
<td>0% Reception</td>
</tr>
<tr>
<td>PC</td>
<td>525</td>
<td>585</td>
</tr>
<tr>
<td>Grids</td>
<td>810</td>
<td>885</td>
</tr>
<tr>
<td>Mobic</td>
<td>450</td>
<td>495</td>
</tr>
<tr>
<td>3-ILBH</td>
<td>840</td>
<td>945</td>
</tr>
<tr>
<td>Rsids</td>
<td>990</td>
<td>1380</td>
</tr>
</tbody>
</table>

Figure 9 shows the ratio of overhead packets (MAODV) to data packets from 30 (the start of data transmission) to the end of 435 seconds (before any of the nodes in the algorithms start dying). Using stable routes allows RSIDS to maintain a lower ratio of overhead packets to data packets. When we see a drop in the periodic packet reception ratio, it is usually correlated with an increase in the overhead ratio. This is due to the reduced amount of data packets received and the increased overhead required to find alternate routes to repair the broken multicast tree. MOBIC shows an extra use of control packets throughout due to its active clustering, and the other algorithms show higher levels than Rsids as forwarding routes break and are rebuilt. In average, RSIDS generates 18% overhead compared to 27% for PC, 29% for 3-ILBH, 30% for GRIDS, and 47% for MOBIC.

Figure 9 - Overhead/Data Ratio (5 to 20, 6m)

V. CONCLUSION

We propose RSIDS, a new energy and stability aware passive clustering technique that reduces overhead by maintaining stable multicast routes. As the number of overhead packets decreases, it results in a network having less redundant/superfluous packets, having a lower probability of collisions and a less congested wireless medium. All these advantages combined with appropriate resting periods permit an increase in the network lifespan. Simulations show that RSIDS, when used to support multicast communications, outperforms existing clustering schemes. RSIDS can still be further improved; for example, since nodes know their neighbors’ residual energy, they can predict when a critical node will enter a resting period/die; this will allow a suitable node to take over the critical node role at the right moment. Other possible improvements are being investigated.

REFERENCES