Mobility Based Distributed Domain Generation Algorithm for MANETs

Kyriakos Manousakis, John Baras
Institute for Systems Research, University of Maryland, College Park, MD, USA

Abstract - Ad hoc networks are very important for scenarios where there is not fixed network infrastructure. These scenarios may appear both in the military and the commercial world. Even though there is much advancement in the area of these networks, they do not scale well. This is due to the inability of the existing protocols (e.g., MAC, routing, security) to tolerate the dynamics of these networks when their size becomes large. A remedy is to apply these protocols in a hierarchical manner. The hierarchy generation in these dynamic environments can be advantageous since the network appears to be smaller due to the aggregation and abstraction resulted from the grouping of nodes. The numerous topological changes can be tolerated easier and the applied protocols can perform better. This division greatly reduces overall overhead (e.g., routing overhead with n nodes goes from $O(n^2)$ to $O(n\log n)$) and allows protocols to be tuned to more homogenous conditions). On the other hand, hierarchy has to be generated carefully in order to be beneficial for the network otherwise it may harm it, due to the imposed maintenance overhead. Towards this objective we have to take into consideration the network environment and design appropriately the hierarchy generation algorithms. In this paper we present a mobility based domain generation algorithm (DGA), which in a distributed manner attempts to exploit similarities in the mobility patterns of the participating nodes by utilizing a set of mobility metrics. By grouping together nodes with similar mobility characteristics, the mobility based DGA targets the generation of robust to mobility hierarchical structures. The algorithm presents better scalability and robustness characteristics from well known existing distributed domain generarion algorithms.

1 Introduction

The dynamic nature of ad hoc networks and the lack of a fixed network infrastructure are the main characteristics that prevent these networks from scaling to a large number of nodes. Significant progress has been accomplished for the advancement of ad hoc technology through the design of several protocols (e.g. MAC, routing, security). The weakness of these protocols is their poor scalability characteristics when the underlying networks become mobile since they are incapable to cope with the network dynamics. The performance of these protocols significantly decreases when network topology changes happen, due to the instability of the links or the failure of participating nodes. Even though new protocols are introduced the scalability problem persists.

Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance (CTA) Program, Cooperative Agreement DAAD19-2-01-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The scalability issues of the existing ad hoc protocols and the inherent difficulty of the protocols to handle the dynamics of large mobile networks do not look promising on the development of scalable flat MANETs. A remedy to these inherent problems could be the application of hierarchy. In a hierarchical environment due to the aggregation and abstraction provided by the grouping of nodes, the protocols have to deal with a limited number of nodes, as opposed to the entire network. For the dynamic establishment of hierarchical structures, many domain generation algorithms have been proposed in the literature [1][2][3][4]. However, the weakness of their majority is that they do not take into consideration the characteristics of the network environment. In many cases, these algorithms harm network performance instead of improving it because of the imposed maintenance overhead.

Having understood the significance of the maintenance overhead for the success of a hierarchy establishment scheme, we propose a new mobility based domain generation algorithm (DGA). The main objective of this algorithm is the generation of stable hierarchical structures by grouping together the nodes that present similar mobility characteristics. Doing so, it is expected that the nodes of the same group will remain connected for long periods of time, reducing significantly the membership changes and the resulted maintenance overhead. If the maintenance overhead is reduced, the performance of the network will improve, benefiting from the hierarchical application of the various networking protocols onto a stable hierarchical structure.

Consider the following example, which indicates the significance of our approach. Assume the network environment of figure 1, where the nodes 1-7 are static sensor nodes. The nodes 8-11 are mobile nodes, which are moving as a group around the sensor field. In this case, if we attempt to generate a hierarchical structure based on the proximity of the nodes or by applying any of the existing domain generation methods (lower ID, highest degree) [1][2] then we may end up with frequent re-clustering of the border nodes. The latter will result in maintenance overhead which will negatively affect the performance of the applied protocols.

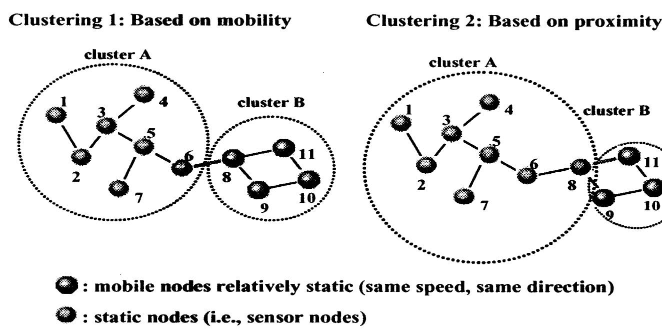


Figure.1. Domain generation motivation example

If we cluster with respect to the mobility of the nodes then in this scenario we eliminate the maintenance overhead, since the two groups that will be generated will remain the same throughout the lifetime of the network. By using the mobility characteristics of the nodes as criterion, a more stable and robust hierarchy is obtained, resulting in better network performance.

The proposed mobility based DGA follows the philosophy of the above example utilizing mobility metrics for the grouping of nodes. The algorithm is based on one-hop information exchange and presents $O(n)$ communication complexity in a network of n nodes. The generated clusters appear to be more robust compared to some well known existing distributed clustering algorithms. Furthermore, the algorithm presents very promising performance characteristics even in cases of large and highly mobile networks, where the existing distributed algorithms fail. The ability of the algorithm to cope with the dynamics of the network emerges from its inherent functionality to group the nodes with respect to these dynamics.

In the next section we present related work that has been done in the area of clustering. In section 3 we define the mobility metrics we apply for the generation of stable hierarchical structures. In section 4 we present the mobility based DGA. Section 5 presents some indicative performance evaluation results and section 6 concludes the paper.

2 Related Work

The idea of generating hierarchy for improving the scalability and robustness of the network has existed for many years. In wireless ad hoc networks the idea of clustering emerged when the packet radio networks were introduced, which are the ancestors of ad hoc networks. In this area, Baker et al [2] introduced the idea of clustering through the concept of a distributed linked cluster architecture. The clustering objective of this work was the hierarchical application of routing in a more robust to topological changes environment. The idea of clustering in ad hoc networks was revisited recently in the context of mobile multimedia wireless networks [5] [6]. One of the most popular clustering schemes among the existing ones is the Lower-ID scheme. This scheme, used in [5], is the point of reference and of comparison for many recently introduced clustering schemes. In [5] Gerla et al., proposed a simple distributed algorithm that yields clusters that are at most two hops in diameter. In each cluster the node with the lowest ID among its one hop neighbors becomes the clusterhead and maintains the cluster memberships of the other nodes in the cluster. An algorithm based on the degree (e.g. number of 1-hop neighbors) of the nodes was proposed in [5]. The nodes having the highest degree among their 1-hop neighbors were selected to be the clusterheads. This algorithm, called also the highest degree clustering algorithm performed much worse than the Lower-ID in terms of the robustness of the clusters. The robustness was measured by the average number of membership changes per unit of time. Due to the popularity of the Lower-ID algorithm there was an attempt for its generalization. DCA (Distributed Clustering Algorithm) [7][4] and WCA (Weighted Clustering Algorithm) [3] generalized the Lower-ID clustering to a weight based clustering by assigning weights to nodes and selecting the clusterheads with respect to these weights instead of the node IDs. All of the above algorithms are mainly concerned with the selection of clusterheads and generate clusters of 2 hops in diameter. They do not take into consideration the network environment for reducing the membership changes and the related overhead as compared to our work. The only work that differs in spirit from the works above and can be classified as a work that takes into consideration the network environment for the clustering of nodes is described in [8]. In [8] a framework for dynamically organizing nodes into clusters in an ad hoc network has been proposed. The authors attempt to bound the probability of path availability. A mobility model was developed and used to derive analytical expressions for the probability of the path availability with respect to time. Even though this is the first attempt of clustering nodes based on the characteristics of the network instead of just selecting clusterheads and generating 2 hop in diameter clusters, this probability based model fails to capture the real mobility model of nodes with respect to neighbors and therefore it is not a sufficient metric for clustering in an environment that cannot be described from the assumed probabilistic mobility model.

We attempt to propose algorithms that take into consideration the network environment and specifically the mobility characteristics (speed, direction) of the nodes but we do not link the clustering decisions with metrics related to any mobility model. The algorithms we propose utilize the node mobility characteristics, which are collected and processed dynamically for the clustering decisions to be obtained.

3 Mobility Metrics

The objective of the proposed algorithm is to generate domains that are robust to the dynamics of the network. The accomplishment of this objective requires the exchange of one hop information between the participating nodes. This information is related with the mobility and topological characteristics of the nodes. A set of such characteristics that each node acquires for itself and exchange them with the neighboring nodes for the domain generation are:

- Geographic coordinates of node i : (x_i, y_i)
- Direction of the node i : θ_i
- Speed of the node i : S_i

Each node can obtain this information about itself from a GPS device or utilizing less accurate methods which are based on communication with neighboring nodes (i.e. received power, time and power difference between subsequent receptions, etc.). When the nodes obtain this information, then they can exchange it with their neighboring nodes for estimating the decision making metric. Since the objective of the proposed algorithm is the generation of a robust to mobility hierarchical structure, the decision making metric is related to the relative mobility characteristics among neighboring nodes. There are many ways to define such a metric with respect to the mobility characteristics mentioned above. Some representative definitions of the metric are:

- Relative direction $\theta_{r_{ij}}$
- Relative velocity $S_{r_{ij}}$

- Link expiration time LET_{ij}

The values of these metrics can be computed locally to each node after the exchange of their mobility characteristics with their neighboring nodes. The definitions of the metrics are:

Relative Direction ($\theta_{r_{ij}}$): The relative direction between two nodes i, j is defined with respect to their individual direction of movement. If i is moving with direction θ_i and j with direction θ_j then their relative direction $\theta_{r_{ij}}$ is defines as:

$$\theta_{r_{i,j}} = \min\left(\left|\theta_i - \theta_j\right|, 360 - \left|\theta_i - \theta_j\right|\right), \quad (1)$$
$$\theta_i, \theta_j \in \left[0^o, 360^o\right),\ \theta_{r_{i,j}} \in \left[0^o, 180^o\right]$$

A graphical example demonstrating the computation of the relative direction of two nodes follows:

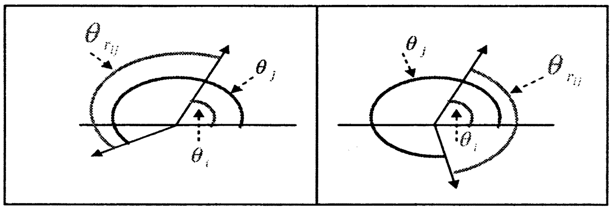


Figure 2. Relative Direction: Computation Approach

Relative Velocity ($U_{r_{ij}}$): The relative velocity of two nodes is the velocity with which a node approaches or recedes from another node, whether both are moving or only one. The mathematical definition of this metric is:

$$U_{r_{i,j}} = \sqrt{U_{X_{i,j}}^2 + U_{Y_{i,j}}^2} \quad (2)$$
$$U_{X_{i,j}} = S_X \cos\theta_i - S_X \cos\theta_j$$
$$U_{Y_{i,j}} = S_Y \sin\theta_i - S_Y \sin\theta_j$$

Link Expiration Time (LET_{ij}): The Link Expiration Time is defined as the estimated lifetime of the link that connects two nodes, and is computed with respect to the mobility characteristics of the corresponding nodes (e.g., direction, speed and their geographical positions). The mathematical definition of this metric is:

$$LET(j,k) = D_{t_i(j \leftrightarrow k)} =$$

$$\begin{cases} \dfrac{-(ab+cd)+\sqrt{(a^2+b^2)r^2-(ad-bc)^2}}{a^2+c^2} & \text{,nodes } j,k \text{ are in range} \\ 0 & \text{,nodes } j,k \text{ are not in range} \\ \infty & \text{,nodes } j,k \text{ are relatively static} \end{cases} \quad (3)$$

$a = S_j \cos\theta_j - S_k \cos\theta_k$

$b = x_j - x_k$

$c = S_j \sin\theta_j - S_k \sin\theta_k$

$d = y_j - y_k$

$r = TxRange_{j,k}$ (*TxRange* in this case is assumed the same for every node)

The values of the above metrics indicate the similarity of the mobility between two neighboring nodes. Specifically, the lower the value of the relative direction and relative velocity, the more similar the mobility. The opposite holds for the link expiration time, where the larger the value the longer is expected the link to survive. The next section presents the functions performed by the domain generation algorithm and how the metrics presented here are utilized for the construction of robust hierarchical structures.

4 Overview of the mobility based DGA

In this section we describe the principal operation of the proposed mobility based domain generation algorithm. An example is also given, which demonstrates the algorithmic steps followed from DGA for generating a robust to mobility hierarchical structure.

4.1 Mobility based DGA

The mobility based DGA is based on one-hop information exchange. The information is related to the mobility metrics we introduced in section 3. We assume that each node characterized from a unique ID, can obtain information about its speed, direction and position (e.g., only in the case where the metric of interest is the Link Expiration Time (*LET*)). Also, the set of their one-hop neighbors can be obtained from the exchange of link state information.

In the proposed algorithm each node joins a cluster after having gone through three phases. The objective is that the generated clusters consist of similar nodes with respect to their mobility characteristics. The three phases are:

Phase I – Neighbor Selection

In this phase each node broadcasts its ID and information (direction, speed, position) related to the decision making metric (relative direction, relative velocity, LET) to its one-hop neighbors. Each node, after the collection of the appropriate information obtains the value of the mobility metric of interest for each one of its neighbors. The set of these values determines the neighbor that the corresponding node will select to join in the same cluster. For example a node will select the neighbor with the lowest metric value in the case where the metric of interest is the relative direction or the relative speed and vice versa when LET is taken into consideration. In the case of a tie, a tie-brake rule is used (ID of the neighbors or random selection).

Phase II – *InfoExchange* List Composition

After the nodes have decided on which is the most appropriate neighbor to form a cluster with, they inform the selected node for this decision. The recipient nodes collect the related messages and record the IDs of the nodes who have selected them. After the completion of this process, each of the nodes generates the *InfoExchange* list, which is the union of the node IDs collected in Phase II, the ID of the selected neighbor and their own ID. The *InfoExchange* list is sorted in ascending order.

Phase III – Cluster Formation

For the cluster formation, each node does not have to communicate with every neighbor but only with those in the *Infoexchange* list. In the distributed environment the sorted

Infoexchange list is utilized for synchronization among the nodes. A node has to wait for the nodes with lower ID in the *Infoexchange* list to decide on the cluster to join and then has to communicate its selection. If the ID of the node is the lowest in the list, then forms a cluster with this ID and communicates it to its neighbors that exist in the list. By the completion of this phase each node belongs to a cluster characterized by a unique cluster ID.
After the high level overview of the various phases, the pseudo-algorithm provided below reveals the detailed functionality of the proposed algorithm as it is performed from each one of the participating nodes:

Phase I

Step I: (Communication)
Broadcast to 1-hop neighbors a TYPE I message:
(*myID*, *Information*)
The information values can be the speed, direction or position of the node and depend on the metric of interest (e.g. relative direction, relative speed, LET)
Step II: (Processing)
Collect the (TYPE I) messages from 1-hop neighbors.
Based on the information collected, evaluate the metric of interest for each one of the neighbors. With respect to the metric values obtained determine which one of the neighbors is the most appropriate for grouping with in the cluster formation.

Phase II

Step III: (Communication)
Broadcast to the selected neighbor (e.g. neighbor with the best metric value) a TYPE II message:
(*myID*, *neighborID*)
Step IV: (Processing)
Collect all TYPE II messages that are referred to my ID.
Generate the *SelectedFromList* list which contains the *neighborIDs* that have selected *myID* as the preferred neighbor to be clustered with. Then generate the *InfoExchangeList*:

$$InfoExchangeList =$$
$$SelectedFromList \cup neighborID \cup myID$$

Sort in ascending order the *InfoExchangeList* with respect to the node IDs.

Phase III

Step V: (Communication)
if myID = Head(InfoExchangeList) then
{ *myCID = CID*
Send to every node with
$nodeID \in InfoExchangeList$
a TYPE III message:
(*myID*, CID=*myID*)
}
else{
Until the reception of a TYPE III message from the nodes with:
$$nodeID \in InfoExchangeList \wedge nodeID < myID$$
{ Upon the reception of TYPE III message {
if $myCID = \varnothing$ *then*
myCID = CID
else
if myCID > CID then
myCID=CID
}
My turn to transmit:
Send *myCID* to every node with
$nodeID \in InfoExchangeList$
Until the reception of TYPE III message from all the nodes with:
$$nodeID \in InfoExchangeList \wedge nodeID > myID$$
{
Upon the reception of a TYPE III message {
if myCID > CID {
myCID=CID
send to all nodes with
$nodeID \in InfoExchangeList$
a message revealing my cluster selection
(*myID*, *myCID*)
}}}}

4.2 Example

In this section we complete the description of the mobility based DGA by providing an example to demonstrate the domain generation algorithm and its various phases for hierarchy construction. Assume that we have the network of figure 3, consisting of 7 nodes. Furthermore, assume that the metric of interest is the relative velocity. In Phase I the nodes broadcast their node IDs, their direction and speed to their 1-hop neighbors. Also, at the same time they collect the analogous information (node ID, direction and speed) from their 1-hop neighbors.

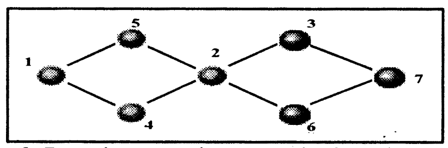


Figure 3. Domain generation example: Sample Network

The nodes, after having collected the appropriate information, they compute their relative velocity with each one of their 1-hop neighbors. Even though the relative velocity is computed locally to each node, the value computed from a pair of nodes (*i*,*j*) is the same, independently of whether is computed at node *i* or node *j*. Assume that for this example the relative velocities for each pair of 1-hop neighbors are provided from the following figure:

Relative Velocity
$U_{r_{ij}} = U_{r_{ji}}$

nbID \ node ID	1	2	3	4	5	6
2	-					
3	-	2				
4	0.2	0.3	-			
5	0	0.1	-	-		
6	-	2.5	-	-	-	
7	-	-	0.1	-	-	0

Figure 4. Relative velocity values as computed pairwise from the neighboring nodes

Based on the above values of relative velocity, each node selects the best match from its 1-hop neighbors (e.g. the lower the relative velocity, the more similar the mobility). Following this rule the selections of the nodes by the end of Phase I are:

$$1 \to (5) \quad 2 \to (5) \quad 3 \to (7) \quad 4 \to (1)$$
$$5 \to (1) \quad 6 \to (7) \quad 7 \to (6)$$

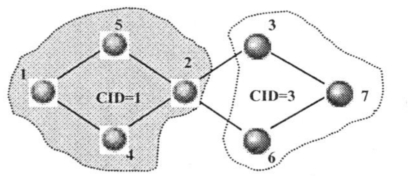
By entering Phase II, the nodes inform their selected 1-hop neighbors. Each node collects and records the IDs of the nodes from which they have been selected into the *SelectedFromList*. The recorded information from each of the nodes is:

$$1 \leftarrow (4,5) \quad 2 \leftarrow (\ ) \quad 3 \leftarrow (\ ) \quad 4 \leftarrow (\ )$$
$$5 \leftarrow (1,2) \quad 6 \leftarrow (7) \quad 7 \leftarrow (3,6)$$

By combining the above information with their selections from Phase I each one of the nodes generates an *InfoExchange* list. The *InfoExchange* lists of the nodes by the end of Phase II are:

$$InfoExchange\ Lists$$
$$1:(1,4,5) \quad 2:(2,5) \quad 3:(3,7) \quad 4:(1,4)$$
$$5:(1,2,5) \quad 6:(6,7) \quad 7:(3,6,7)$$

Phase III completes the generation of the clustering map. Each node will utilize the *InfoExchange* list in order to select the cluster to join. The nodes are listening to the clustering selections of the lower ID 1-hop neighbors that belong to their *InfoExchange* list until their turn comes to decide on the cluster to join. After they decide, they wait for the rest of the 1-hop neighbors (e.g., nodes with higher IDs) in their *InfoExchange* list to decide. For the specific network of figure 3, the generated clustering map after the completion of Phase III looks like:



$$CID = 1:\{1,2,4,5\}$$
$$CID = 2:\{3,6,7\}$$

Figure 5. The clustering map established from the DDC algorithm

If we carefully observe, the values of the relative velocities of the nodes assigned to the same cluster are much lower compared to the values of the relative velocities of the nodes assigned to different clusters. The mobility based domain generation algorithm behaves in accordance to our clustering objectives by grouping together nodes with similar mobility characteristics. By doing so, we aim on the generation of robust to mobility clusters. The nodes in the same cluster are expected to remain connected for longer periods of time compared to the nodes that do not belong into the same groups. As we are going to show later in the performance evaluation section, the generated clusters are more robust to topology changes compared to other distributed clustering algorithms that do not take into consideration the dynamics of the network.

5 Performance Evaluation

This section elaborates on the ability of the proposed domain generation algorithm to establish a hierarchical structure that is robust to mobility. The effectiveness of the approach is evaluated by comparing it with a well known distributed domain generation algorithm (lowest ID – LID) which forms domains without taking into consideration the dynamics of the network environment.

5.1 Robustness of the mobility based DGA

The robustness of the mobility based DGA is measured from the stability of the domains' membership with respect to the mobility of the participating nodes. In order to highlight the effectiveness of the proposed algorithm we are comparing it with the lower ID (LID) algorithm [1], which utilizes metrics unrelated to the network environment for the establishment of hierarchical structures.

LID selects cluster heads (CHs) among the participating nodes based on their IDs. For the domain formation, the remaining nodes (non-clusterhead nodes) are assigned to the CH node with the lowest ID among the CH nodes which are at most 1-hop away. The LID algorithm does not take into consideration the dynamics of the network for the domain formation since the selection metrics (proximity, nodes' IDs) are unrelated to them.

We compared the membership stability of the domains obtained from the mobility based DGA with the corresponding stability of the domains obtained from the lower ID (LID) algorithm. To evaluate the robustness of the hierarchical structures generated by the mobility based DGA compared to LID we measured the average membership changes and the average number of generated clusters. We applied both algorithms in network environments of various sizes and mobility levels. Namely, we generated networks of 100 to 1000 nodes and we applied the Random Waypoint Mobility (RWPM) model with pause time 0.

In the RWPM model each node selects a destination in the limits of the pre-specified area. This destination is approached with constant speed selected from the node at random. When the destination is reached the node selects new destination and new speed and the process is repeated. We investigated several scenarios corresponding to different maximum allowable speeds (between 1m/s and 10m/s) in order to evaluate the robustness of the generated hierarchies in various levels of mobility. We ran the algorithms (mobility based DGA, LID) for 1000s of network time. The statistics were sampled every 1s of network time.

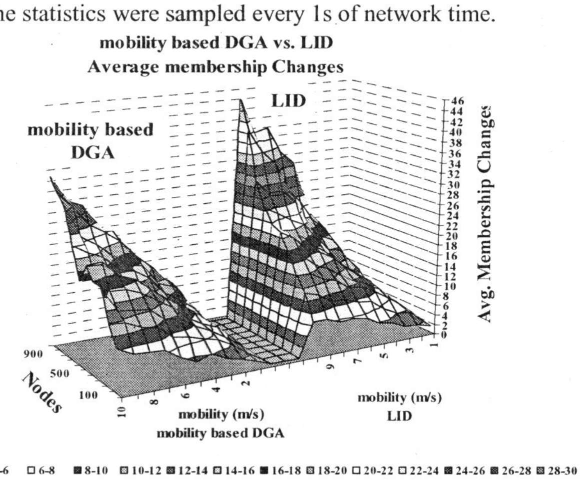


Figure 6. Average membership changes (LID vs. mobility based DGA) with respect to network size and mobility level.

Figure 6 represents the average membership changes for networks of size 100 to 1000 nodes and maximum allowable node speeds of 1m/s to 10m/s. The left part of the graph represents the average membership changes for the mobility based DGA algorithm and the right part represents the average membership changes for the LID algorithm, respectively. The higher the mobility and the larger the size of the network, the better the performance of mobility-based DGA algorithm compared to the performance of the LID algorithm. For example for 1000 nodes and 10m/s maximum speed, mobility-based DGA requires on average 32 membership changes per second and LID requires on average 44 membership changes per second. For 1000 seconds of network time, the mobility based DGA requires on average 12000 less membership changes than LID algorithm, which is an improvement of 27.2%.

Apart from the membership changes, a metric that indirectly characterizes the robustness of the proposed algorithm, is the average number of generated clusters. The smaller the number of generated clusters, the more tolerant is the hierarchical structure to the topological changes due to mobility (e.g. the larger the cluster, the higher the probability of a node, whose connectivity changes, to remain connected to its original cluster).

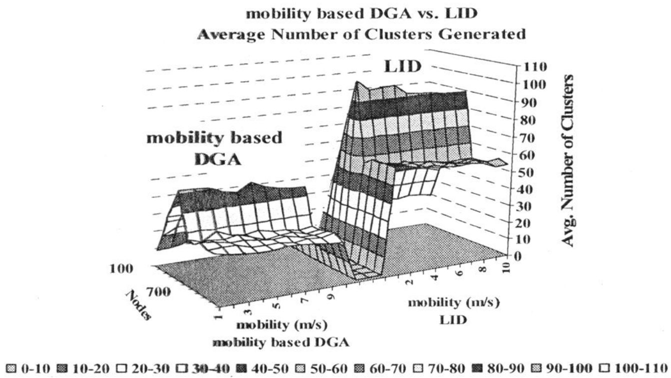


Figure 7. Average number of clusters generated from LID and mobility based DGA algorithms for various network sizes and mobility levels

Figure 7 demonstrates the average number of clusters generated from each one of the algorithms. The left part of the graph represents the average number of clusters generated from the mobility based DGA and the right part represents the average number of clusters generated from the LID algorithm, respectively. The general observation is that the number of clusters that LID generates is more than double the number of clusters generated from the proposed algorithm. This observation suggests that the average cluster size of the mobility based DGA is more than double the average cluster size of LID. The latter can also be explained from the fact that LID does not generate clusters with diameter larger than 2-hops as opposed to mobility DGA, which does not have such restrictions.

Because LID generates clusters without taking into consideration the network environment, it is more possible to harm the performance of the network. The larger the number of membership changes, the larger the introduced overhead to the applied protocols. The stability of the hierarchy generated from the mobility based DGA targets the minimization of this overhead, so that the overall performance of the network is improved.

Even though the superiority of the proposed algorithm has been presented with respect to the RWPM model, the stability of the hierarchical structures established from the mobility based DGA is expected to be even better in scenarios where group mobility is exploited. If we evaluate the proposed algorithm with respect to a group mobility model (Reference Point Group Mobility model), it is expected to establish more robust hierarchical structures compared to RWPM. The mobility based DGA was designed to identify and group together nodes with similar mobility characteristics, so in a network environment, where distinct mobility groups exist, the algorithm tends to eliminate the overhead due to the membership changes by accurately grouping together the nodes that present similar mobility characteristics.

5 Conclusions

We evaluated the mobility based DGA with respect to the RWPM mobility model for various network sizes and mobility levels and we compared the average membership changes and the average number of generated clusters with the corresponding performance metrics of the LID algorithm. The results are indicative of the stability presented from the hierarchical structures obtained from the mobility based DGA, even though the mobility model assumed does not favor group mobility. Namely, mobility based DGA requires 27.2% less membership changes on average than LID, in the case of 1000 nodes moving with respect to RWPM at the maximum speed of 10m/s.

References

[1] R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," IEEE Journal on Selected Areas in Communications, pages 1265-1275, September 1997

[2] D. Baker, A. Ephremides, and J. Flynn "The design and simulation of a mobile radio network with distributed control," IEEE Journal on Selected Areas in Communications, SAC-2(1):226--237, 1984

[3] M. Chatterjee, S. K. Das, D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks, " Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, April 2002, pp. 193-204

[4] S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," Proceedings of Vehicular Technology Conference, VTC 1999-Fall, Vol. 2, pp. 889-893

[5] M. Gerla, J. T-C. Tsai, "Multicluster, Mobile Multimedia Radio Netwroks," Wireless Netwroks 1, pp. 255-265, 1995

[6] R. Ramanathan, M. Steenstrup,"Hierarchically-Organized Multihop Mobile Networks for Quality-of-Service Support," Mobile Networks and Applications, Vol. 3, No. 2, August 1998

[7] S. Basagni, "Distributed Clustering for Ad Hoc Networks," Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Netwroks (I-SPAN'99), Perth/Fremantle, Australia, June 1999, pp. 310-315

[8] B. McDonald, T.F. Znati,"A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks," IEEE JSAC, Vol. 17, No. 8, August 1999

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government