Limiting the Impact of Mobility on Ad Hoc Clustering

Rituparna Ghosh, and Stefano Basagni
Dept. of Electrical and Computer Engineering, Northeastern University
Boston, MA, U.S.A.
rghosh@ece.neu.edu, basagni@ece.neu.edu

ABSTRACT

This paper explores the impact of node mobility on *DMAC*, a typical clustering protocol for mobile ad hoc networks. Several protocols for clustering have been proposed, which are quite similar in operations and performance. We selected one and evaluate the cost of maintaining the clustering structures when the nodes move according to three different mobility models, namely, the random way point model, the Brownian motion and the Manhattan mobility model. Via ns2-based simulations we have observed that the mobility models have different impact on protocol performance. The general trend, however, appears to be the same for networks of increasing size. The second contribution of this paper concerns mitigating the impact of mobility over the clustering structure and hence on the overall network performance. We consider a generalization of DMAC (GDMAC) where rules are established to decrease the number of cluster updates. Via simulation we have observed that GDMAC is effective in reducing the clustering overhead imposed by mobility, and hence its maintenance cost.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless communication.

General Terms: Design, Measurement, Performance.

Keywords: Ad Hoc Networks, Clustering Protocols, Mobility Models.

1. INTRODUCTION

A mobile ad hoc network is an infrastructure-less multi-hop wireless network where each node acts as a router, thus relaying packets of other nodes. These kinds of network are suitable when an instant communication infrastructure is needed and it is not viable to build *ex novo*, or repair, a wired communication system. Limited bandwidth, scarcity of resources (e.g., battery power), and unpredictable and rapid node movements are some of the characteristics of ad hoc network. Therefore, among the major challenges in designing ad hoc protocols there is the requirement of taking into account the dynamic nature of the network topology, where links between neighboring nodes (i.e., nodes that are in each other transmission range) are added and removed all the time. Another challenge posed by the ad hoc architecture is the need to find scalable solutions. As for wired networks, hierarchical approaches have been proposed for routing that make possible to "simplify" the overall network topology now seen as a set of *clusters*. Nodes are partitioned into groups, each with a *clusterhead* that coordinates the cluster formation process and some *ordinary nodes* which rely on the clusterhead for inter-cluster communications. The idea common to basically all clustering protocols is to cluster together nodes that are in physical proximity, thereby providing the network with a hierarchical organization which is smaller in scale, and hence simpler to manage. Clustering for ad hoc networks has been widely investigated and a host of solutions are available. A thorough survey of ad hoc clustering protocols, as well as a performance-based comparison among some of the most representative solutions, has been presented in [8]. The vast majority of clustering protocols deal with static or quasi-static networks, or assume no node mobility while clustering is taking place. Mobile clustering is still largely uncharted territory. With this paper we provide a first account of the effect of different models of mobility on a typical ad hoc clustering organization. In particular, we consider a basic clustering algorithm, the Distributed and Mobility Adaptive Clustering (DMAC) [7] protocol, and we evaluate its performance when the network nodes move according to the random way point mobility model, the random walk mobility model (Brownian motion) [12] and the Manhattan mobility model [20]. Among the variety of ad hoc clustering protocols, DMAC is chosen for its simplicity and efficiency in building the clusters and in maintaining the cluster structure in the face of node mobility.

The contribution of this paper goes beyond the assessment of the impact of mobility on an ad hoc clustering organization. Based on our observation, we have proceeded to defined methods for limiting the effect of mobility on clustering. In particular, we have enhanced DMAC with primitives that decrease the number of the nodes that are elected to be clusterheads (elections) and of the nodes that switch cluster (re-affiliations). We termed the resulting protocol *Generalized DMAC* (GDMAC).

As expected, our observations confirm that the protocol GDMAC is effective in limiting the detrimental effect of mobility on the DMAC-based clustering organization. In particular, and independently of the speed of the nodes, GDMAC outperforms DMAC in all investigated metrics,

PE-WASUN'05, October 10–13, 2005, Montreal, Quebec, Canada.

hence being able to reduce the time the network is "blocked for re-clustering" and at the same time the cost associated to the re-clustering itself.

The aim of this paper is not a comparative study or a performance evaluation of different clustering techniques with respect to mobility. Rather, we aim at evaluating how much mobility affects the hierarchical organization of ad hoc networks and we propose simple and efficient techniques for limiting the impact. In our investigation we take into account realistic network conditions such as packet loss (due to collisions), increasing network density, unpredictable link failures, etc. All the performed experiments confirm the intuitions and expectations behind the basic concept that was proposed in [6]. The GDMAC protocol is indeed able to stabilize DMAC cluster structures under different network dynamics. Our ns2-based implementation takes into account all the realistic physical and MAC layer characteristics of ad hoc networks. The metrics we considered in our study concern the part of clustering performance that is mostly affected by the mobility of the nodes, which include the following averages.

A. Cluster density, i.e., the number of clusters.

B. Cluster stability, measured in terms of cluster lifetime (how long a node is a clusterhead), affiliation time of an ordinary node to a particular cluster (residence time), and number of status changes per node (election to clusterhead and re-affiliation to another clusterhead).

C. Message complexity per node. This is the total number of clustering-related messages sent by a node.

The remainder of the paper is organized as follows. In the next section we describe related work in the area of *mobile* ad hoc clustering. Section 3 introduces DMAC and GDMAC in details. Section 4 shows the simulation results for DMAC and GDMAC in networks where nodes move at different speeds according to the three selected mobility models. Finally, Section 5 concludes the paper.

2. RELATED WORK

Clustering protocols for ad hoc networks are roughly divisible into two main classes, *node-centric* and *cluster-centric*. In node-centric solutions clusters are created around the clusterheads. The clusterheads form a *dominating set* of the network nodes, and are responsible for providing basic network function of behalf of and for their ordinary nodes. Typical functions are controlling channel access, performing power measurements, guaranteeing bandwidth for real time traffic and general coordination of intra and inter-cluster communications. This usually induces on the clusterhead increased overhead with respect to the overhead imposed on the ordinary nodes. A generalized load balancing solution for resolving this asymmetry is given in [1], where a circular queue is maintained that distributes the responsibility of acting as clusterheads evenly among all the cluster nodes (clusterhead rotation). A number of algorithms have been proposed for dealing with cluster formation in ad hoc networks which belong to node centric clustering. Most of the algorithms that have been proposed are based on two fundamental algorithms, lowest ID [4] and highest degree [22]. A generalization of this idea, where each node is at most d hops from a clusterhead is proposed in [2], where a protocol is given that efficiently builds disjoint clusters in which each node is at most $d \geq 1$ hops away from its clusterhead. The network is clustered in a number of rounds which is proportional to d, which favorably compares to most of the previous solutions when d is small. Lin et al. [19] present a *node-ID* based algorithm which is adapted from the early LCA algorithm of [4], as well as on the degree based algorithm from [17]. In [19] the node with the lowest ID is selected as a clusterhead, and the cluster is formed by that node and all its neighbors. The same procedure is repeated among the remaining nodes until each node is assigned a cluster. In [17], the same procedure is performed where instead of considering a node ID a clustered is selected considering nodal degree (the number of a node's neighbors). Basagni [6] proposed the weight based algorithms DMAC and GDMAC (details of which are given in the next section). These algorithms assume only local (1-hop) information and are quite fast and message efficient compared to those requiring global topology information [13]. In ARC [10] a cluster change only occurs when one cluster becomes a subset of another which helps in improving the stability of the clusters. [14] uses mobile agents for this purpose. Mobile agents are used for cluster maintenance and distribution of routing information at each node. Mobile agents also help in cluster size adjustment, re-clustering and continuous cluster state monitoring. In other solutions [3,9,15] the mobility pattern of the nodes is considered for forming the clusters. In [3] the mobile nodes are organized in variable-sized non overlapping clusters, while in [15] the clusters are of variable diameter. The algorithm proposed in [3] uses a combination of both physical and logical partitions of the network for the clustering, while similar moving nodes are grouped together in [15]. Basu et al. [9], similarly to the lowest ID solution, is a weight-based clustering algorithm where *aggregated local mobility* (ALM) is used to elect a clusterhead. The ratio between received power levels of successive transmissions between a pair of nodes is used to compute the relative mobility between neighboring nodes.

Different from the node-centric solutions in ad hoc networks are the cluster-centric ones, also know as *k-clustering*. A k-cluster is made up a group of nodes that are mutually reachable by a path of length $k \geq 1$. In this approach the network is decomposed into clusters of nodes with no specific node designated as clusterhead. In [18], an algorithm for achieving *clique* partitioning (where $k = 1$) and maintenance of the cliques in the face of various network occurrences is presented. The (α, t)-cluster approach is proposed in [21], where dynamically organizing mobile nodes group themselves into clusters in such a way that the probability of path availability between the nodes of the clusters can be bounded. Clustering solutions for unit disk graphs (UDG, one of the model used for representing ad hoc networks mathematically) are presented in [16]. The two phase distributed approximation solution has polynomial time and message complexity. In the first phase a spanning tree of the network is constructed. In the second phase this spanning tree is partitioned into subtrees with bounded diameter. One main problem with this approach is that the nodes are unrealistically assumed to remain stationary during clustering. There are also some other algorithms that have implicit constraints on the cluster diameter [5,21,23]. In [5] a distributed implementation of a centralized clustering algorithm is proposed. It requires constructing a global spanning tree to generate clusters that satisfy certain constraints on the number of nodes in each cluster and the number of hierarchical levels. The priority is to create clusters of size between k and $2k-1$, for a given k. The algorithm first creates a rooted spanning tree which covers the entire network. The cluster formation is then run bottom-up, where subtrees are made into clusters that meet the size requirements. [25] discusses the problem of cluster maintenance at length. The protocol is based on the properties of diameter-2 graphs and makes use of a spanning tree maintained at some nodes. For further references on ad hoc clustering, the reader is referred to [8].

3. DMAC AND GDMAC

In this section we describe the operations of the Distributed and Mobility Adaptive Clustering (DMAC) protocol. We further generalize DMAC to GDMAC by giving the idea of how GDMAC helps in improving the performance of DMAC by limiting the impact of mobility. Upon starting the protocol, a DMAC node computes its *weight*, i.e., a real number > 0 which indicates how good that node is for serving as a clusterhead. For instance, the weight could be computed based on the node's residual energy or on its current velocity. This also implies that a node's weight changes in time, reflecting the changes in the node's status. The node then acquires knowledge of its neighbors' identity and weights and depending on the weights it decides whether to be a clusterhead or not. This process is performed by having each node periodically send out "hello" packets, which carry the node's identity, its current weight and status (either a clusterhead or an ordinary node; clearly at the start of the algorithm the status of a node is undecided). In particular, the node with the bigger weight in its (one hop) neighborhood will declare itself a clusterhead. Consequently, all its neighbors become ordinary nodes. This prevents two clusterhead to become neighbors, thus obtaining a well-spread set of clusterheads throughout the network. According to DMAC specifications [7], an ordinary node always affiliates with the neighboring clusterhead with the biggest weight. If in time a "bigger weighted" clusterhead comes along, the node switches to the new one (re-affiliation). Another instance of re-affiliation happens when two clusterheads come into the hearing range of each other, only the "heavier" (in terms of weight) one stays clusterhead, while the other one becomes its ordinary node. The ordinary node that resigned from the role of clusterhead must affiliate with the biggest neighboring clusterhead. If none is in range, the node becomes a clusterhead itself (election). In this way DMAC is able to accommodate network dynamics due to node mobility and to the arrival of new nodes at a later time. This feature is made possible by the continuous "monitoring" of a node's surroundings to determine the presence of new nodes: As soon as a new node is detected, relevant informations (identity, weight, etc.) are exchanged among the nodes, and suitable procedures are triggered to re-organize the clusters to include the new node.

It is clear that, dependently on the degree of mobility in the network and on the particular mobility pattern followed by the nodes, there can be frequent cluster re-organizations in the form of elections and re-affiliations. There are cases when even a single topological change might trigger numerous role changes among nodes. This "ripple effect" of protocol DMAC-like has been studied in [11] which describes how the introduction of a single new node can cause a change of role reversals among the other nodes along a path of a tree when certain conditions exist. Such undesirable phenomena are detrimental for network performance, as they produce extra protocol overhead, decrease the data throughput and increase data latency (when a cluster is re-organizing, data cannot be efficiently sent).

A possible solution for decreasing the number of mobility-dependent changes (election and re-affiliations) involves relaxing the quite strict DMAC requirements that forbid neighboring clusterheads and that force an ordinary node to always affiliate with the bigger among its neighboring clusterheads. The Generalized DMAC protocol, GDMAC, was defined with this purpose in mind. GDMAC operations are based on the value of two parameters H and K which helps in mitigating the DMAC "chain reaction." Up to $K \geq 0$ clusterheads are now allowed to be neighbors. Furthermore, an ordinary node switches to a newly arrived clusterhead only when the weight of the new clusterhead exceeds the weight of its current clusterhead by a quantity H.

It is clear that GDMAC is indeed a generalization of DMAC. The DMAC protocol is obtained when both K and H are set equal to 0, which can be meaningful in static or quasi-static networks. The higher is the value of H, the less likely a node will switch to a new neighboring clusterhead. The parameter H implements the idea that a cluster re-organization is needed only when the new clusterhead is really better than the current one. Parameter K controls the spatial density of the clusterheads. Having $K = 0$ ensures that no two clusterheads can be neighbors (DMAC). Setting $K > 0$ helps in mitigating cluster re-organization since now a clusterhead is not forced to give up its leadership when up to $K-1$ clusterhead with bigger weights become its neighbors.

4. SIMULATION RESULTS

We have investigated via simulation the impact of mobility over both DMAC and GDMAC. The two protocols have been implemented in the VINT project network simulator ns2 [24].

Our study has been concerned with the following metrics (all averages).

1. Cluster density, i.e., the number of clusters formed during the simulation time.

2. Cluster stability, which we define as how much a cluster remain the same ("stable") while the nodes move. The stability of a cluster is measured with respect to: 1) Cluster lifetime, i.e., how long a node is a clusterhead; 2) Residence time, which is the affiliation time of an ordinary node to a particular cluster, and 3) status changes, i.e., the number of status changes per node (election and re-affiliations).

3. Message complexity per node. This is the total number of CH, JOIN and RESIGN messages sent by a node.

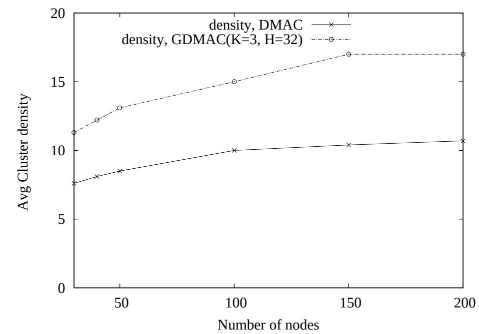
Our ns2 implementation is based on the CMU wireless extension, i.e., on the use of the IEEE 802.11 MAC with the DCF [24]. Our simulations refer to mobile scenarios where n wireless nodes with maximum transmission radius of 250m move in a geographic square area of side L. The mobility models we considered in this paper are the *random way point* (RWP) model, the *random walk* (RW) model and the *Manhattan* (MAN) model. The RWP model is used in many prominent simulation studies of ad hoc network protocols. According to this model, each node begins by pausing in one location for a certain period of time. After this time, a node chooses a random destination and a speed that is uniformly distributed between [minspeed,maxspeed]. The node then travels toward the newly chosen destination in a straight line with the selected speed. Upon arrival, the node pauses for a specified time period before starting the process again. The random walk mobility model mimics erratic movement. In this model a node moves from its current location to a new location by randomly choosing direction and speed. The new speed and direction are both chosen from predefined ranges [speedmin,speedmax] and $[0, 360]$. The Manhattan model, first introduced in [20], describes a realistic city mobility model (i.e., it depicts how a node might move in a city surrounding). This model requires the simulation area to be divided in horizontal and vertical blocks (here 10) and the nodes are forced to move along predefined paths between the blocks with predetermined speed. At the end of a block a node pause (for 5s) and then it decides whether to turn or continue to travel in the same direction with some probability (here 0.5).

We make the common assumption that two nodes are neighbors if and only if their Euclidean distance is ≤ 250m. The simulations have realistic MAC and physical layer characteristics where packets might get lost due to the underlying channel conditions and due to possible collisions and need to be re-transmitted. The nodes beacon their presence periodically (here, every half a second) and the drifting in of a new node is realized when the nodes hear the beacons of a new neighbor. Similarly when a node does not hear beacons from its neighbor within a certain amount of time (here 3s), it assumes the neighbor to be either "dead" or out of range due to mobility. In our simulations the number of nodes n has been assigned the values 30, 40, 50, 100, 150 and 200, while L has been set to 1000m. This allows us to test the protocols on sparse topologies (30 and 40 nodes) and on increasingly dense ones.

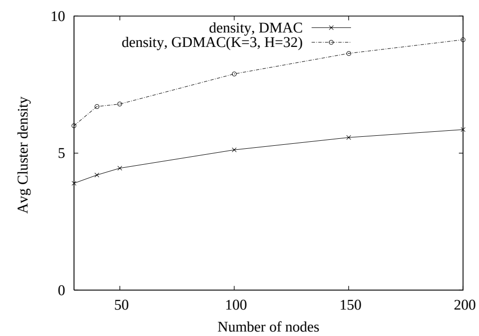
All results are obtained on 100 topologies and the simulation time is 1000 seconds. All nodes have been assigned random weights between 0 and 80. The GDMAC values for K and H are set to 1, 3 and 4, and to 16, 32, 64, and 80, respectively. We have observed that for values of $H \geq 32$ and fixed constant K, there is no significant improvement on the performance of GDMAC. For this reason, we also have considered smaller values of H. In this paper we show results obtained for the two values $K = 3$ and $H = 32$. Simulations are performed where the speed of the nodes is 2, 5 and 10m/sec.

4.1 Cluster density

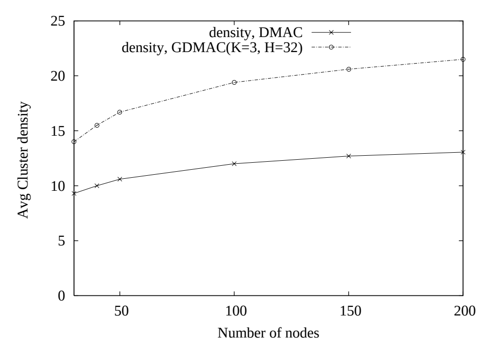
Our first metric concerns the average number of clusters formed during the simulation time. Cluster forming and updating due to election and re-affiliation is the major source of clustering overhead, and thus this metric gives us an idea on how much the two protocols creates clusters while the nodes move. The average cluster density for DMAC and GDMAC ($H = 32$, $K = 3$) with node mobility of 2m/sec are shown in Figure 1(a), Figure 1(b) and Figure 1(c). With increasing n, when the nodes move according to the RWP model the average number of CHs "converges" to 17 in case of GDMAC. For the RW model the average number of cluster is a little less than 10, and for the Manhattan model the average value is 22. For DMAC ($K = 0$), the average number of clusters levels at 10 (RWP), 5 (RW) and 14 (MAN) when n increases. This is clearly because GDMAC allows clusterheads to be neighbors.



(a) RWP



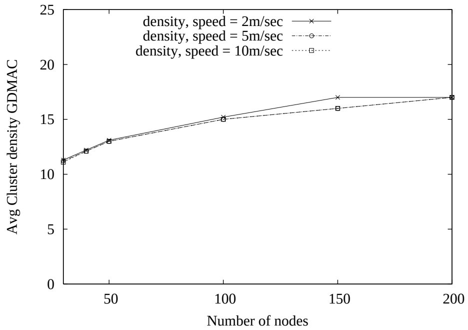
(b) RW



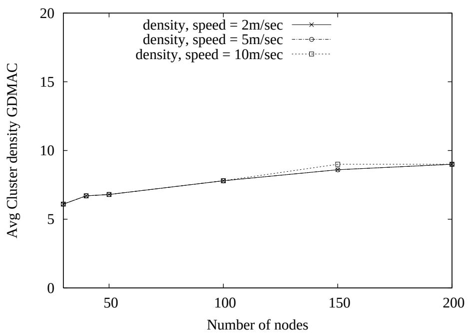
(c) MAN

Figure 1: Average cluster density

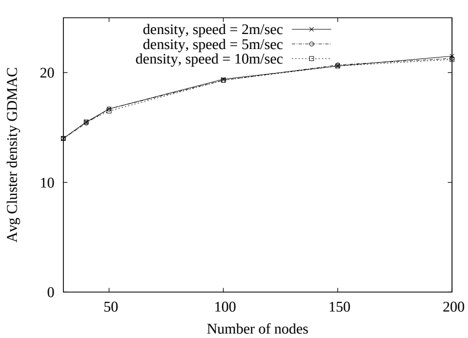
Figure 2(a), Figure 2(b), and Figure 2(c) show the effect of mobility on the GDMAC average cluster density. For greater values of n, mobility has little effect. With increasing n, the density increases until the average density of 17 CHs (RWP), 9 CHs (RW) and 22 CHs (MAN) is achieved.



(a) RWP



(b) RW

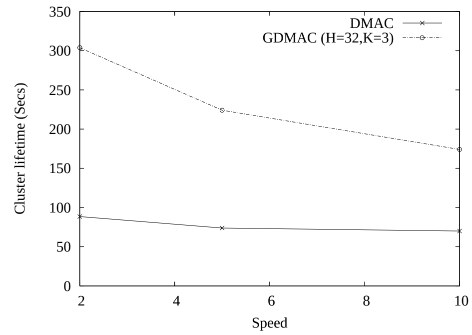


(c) MAN

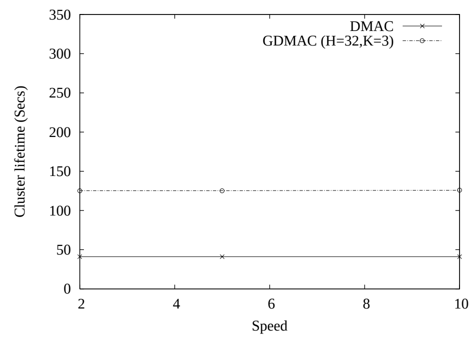
Figure 2: Impact of mobility in GDMAC

4.2 Cluster stability

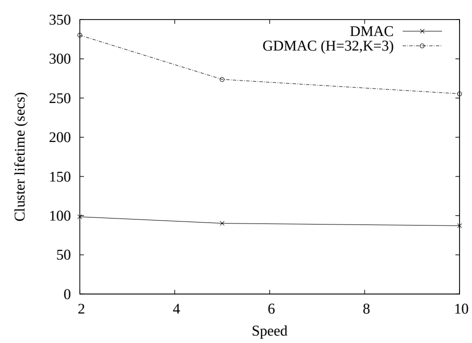
Figure 3(a), Figure 3(b) and Figure 3(c) show cluster lifetime (in secs) over the total simulation time. For the RWP and MAN models, lifetime for both DMAC and GDMAC decreases with increasing speed as nodes tend to drift away faster with higher velocity. Greater lifetime is achieved for GDMAC with $K = 3$ as less CHs are forced to resign when they become neighbors thus increasing the cluster lifetime over DMAC clusters' lifetime. However for the RW model speed does not have too much of an impact on the cluster lifetime since the nodes tend to stay in the center of the simulation area most of the time, which results in static/quasi-static motion. As expected, the protocol GDMAC tends to be more stable than its specific variation DMAC.



(a) RWP



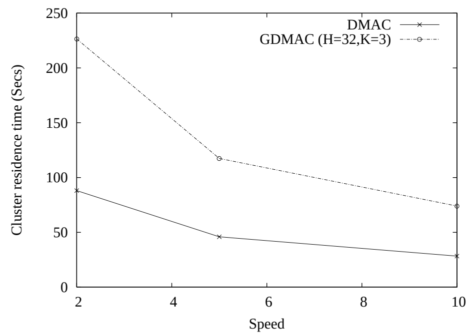
(b) RW



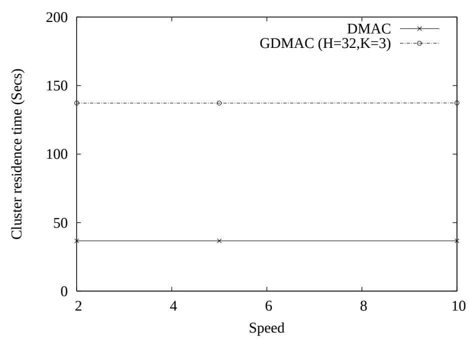
(c) MAN

Figure 3: Cluster lifetime for varying speeds

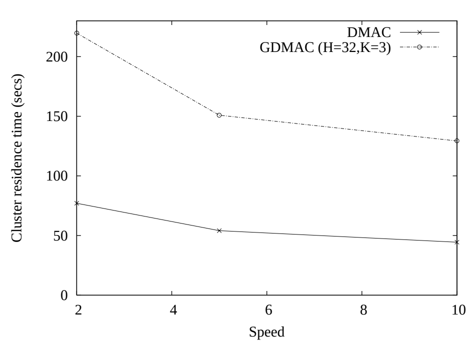
Figure 4 shows the residence time (in secs) of the ordinary nodes for DMAC and GDMAC. For the RWP and MAN models we see that with increasing velocity the clusters become less stable forcing both the ordinary nodes and the CHs to move away more rapidly, thereby forcing the average residence time of ordinary nodes in any particular cluster to drop. In GDMAC this pattern is also observed but the introduction of H and K makes GDMAC more stable than DMAC. For the random walk model though velocity has no major impact, GDMAC is more stable than DMAC.



(a) RWP



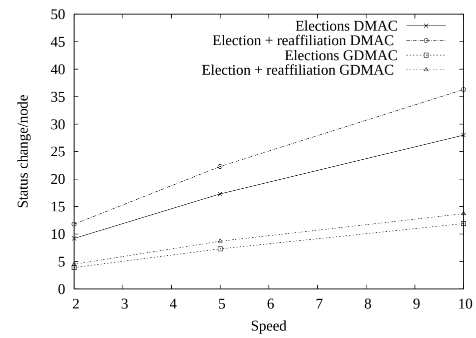
(b) RW



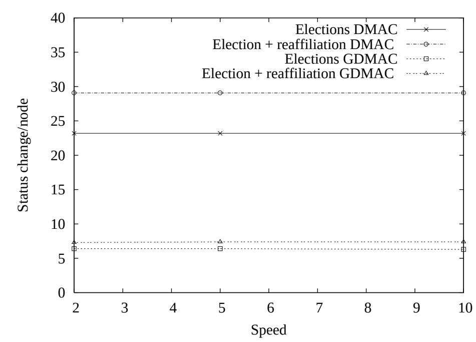
(c) MAN

Figure 4: Residence time for varying speeds

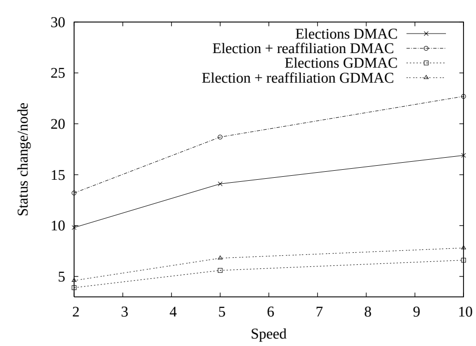
Figure 5(a), Figure 5(b) and Figure 5(c) show the effect of speed on elections and re-affiliations between DMAC and GDMAC ($K = 3$ and $H = 32$). Since an increasing K forces less clusterheads to resign and increasing values of H induce less ordinary nodes switching among clusterheads, GDMAC has a smaller number of elections and re-affiliations than DMAC. For increasing speed the clusters become less stable and hence such status changes increase with speed for RWP and MAN models.



(a) RWP



(b) RW



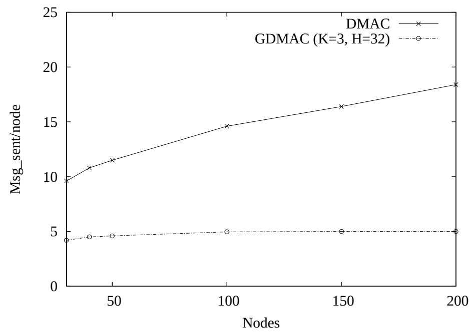
(c) MAN

Figure 5: Elections and re-affiliations

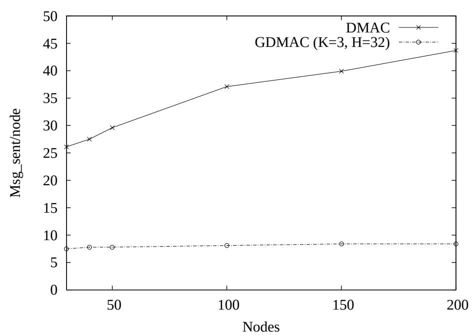
4.3 Message complexity

Figures 6(a), 6(b) and 6(c) show the average number of messages sent by a node. This gives us a measure of the overhead created for maintaining the clusters, which is also an indication on how much this costs in terms of energy per node. The number of messages sent increases with larger values of n until it levels off. This is because with increasing

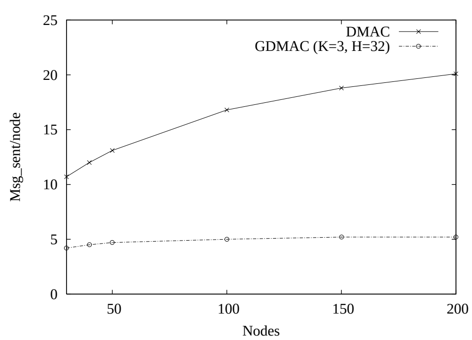
n there is a greater connectivity among the nodes. This helps in keeping a bound on the number of the messages sent. Again, being GDMAC more stable, the number of messages sent is less than those send by DMAC.



(a) RWP



(b) RW



(c) MAN

Figure 6: Number of messages sent

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper we investigated the impact of different kinds of mobility on a typical clustering organization for a mobile ad hoc network. The DMAC protocol has been demonstrated via simulations with respect to metrics relevant for assessing clustering performance, especially in the face of nodes mobility. Once we determined the impact of different mobility models on the selected clustering, we have proposed and studied a simple variation to the basic DMAC. The resulting protocol, GDMAC, has been proven outperforming DMAC with respect to all the considered metrics. In particular, we observed that GDMAC results in good performance independently of the mobility model considered.

Based on the current results, we will keep investigating clustering and mobility into two main directions. First of all, we intend to explore some other metrics, such as energy cost (in case the nodes are energy constrained, as in wireless sensor networks), and those metrics related to constructing and maintaining a backbone of the clusterheads (and hence several routing related metrics). Finally, we will be considering some other clustering solutions for mobile ad hoc networks, and we will be comparing it to the protocol GDMAC investigated in this paper.

6. REFERENCES

[1] Amis, A. D., and Prakash, R. Load-balancing clusters in wireless ad hoc networks. In *Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, ASSET 2000* (Washington, DC, 2000), p. 25.

[2] Amis, A. D., Prakash, R., Vuong, T. H. P., and Huynh, D. T. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of IEEE Infocom 2000* (Tel Aviv, Israel, March 26–30 2000), vol. 1, pp. 32–41.

[3] An, B., and Papavassiliou, S. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad hoc wireless networks. *International Journal of Network Management 11*, 6 (2001), 387–395.

[4] Baker, D. J., and Ephremides, A. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications COM-29*, 11 (November 1981), 1694–1701.

[5] Banerjee, S., and Khuller, S. A clustering scheme for hierarchical control in multi-hop wireless networks. In *Proceedings of the 20th IEEE Infocom 2001* (Anchorage, AK, April 22–26 2001), vol. 2, pp. 1028–1037.

[6] Basagni, S. Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. In *Proceedings of the IEEE 50th International Vehicular Technology Conference, VTC 1999-Fall* (Amsterdam, The Netherlands, September 19–22 1999), vol. 2, pp. 889–893.

[7] Basagni, S. Distributed clustering for ad hoc networks. In *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)* (Perth/Fremantle, Australia, June 23–25 1999), A. Y. Zomaya, D. F. Hsu, O. Ibarra, S. Origuchi, D. Nassimi, and M. Palis, Eds., IEEE Computer Society, pp. 310–315.

[8] Basagni, S., Mastrogiovanni, M., Panconesi, A., and Petrioli, C. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks (S. Olariu, D. Simplot-Ryl, and I. Stojmenovic, editors)* (2006). To appear.

[9] Basu, P., Khan, N., and Little, T. A mobility based metric for clustering in mobile ad hoc networks, 2001.

[10] Belding-Royer, E. M. Hierarchical routing in ad hoc mobile networks. *Wireless Communication & Mobile Computing 2*, 5 (2002), 515–532.

[11] Bettstetter, C., and Friedrich, B. Time and message complexity of the generalized distributed mobility adaptive clustering (GDMAC) algorithm in wireless multihop

networks. In *Proceeding of the 57th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Spring* (Jeju, Korea, April 22–25 2003), vol. 1, pp. 176–180.

[12] Camp, T., Boleng, J., and Davies, V. A survey of mobility models for ad hoc network research. *Wiley Interscience's Wireless Communications & Mobile Computing, Special Issue on Mobile Ad Hoc Networking: Research Trend and Application, S. Basagni and S.J. Lee, eds. 2*, 5 (August 2002), 483–502.

[13] Chatterjee, M., Das, S. K., and Turgut, D. WCA: A weighted clustering algorithm for mobile ad hoc networks. *Journal of Cluster Computing, Special Issue on Mobile Ad Hoc Networks 5*, 2 (April 2002), 193–204.

[14] Denko, M. K. The use of mobile agents for clustering in mobile ad hoc networks. In *SAICSIT '03: Proceedings of the 2003 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology* (Republic of South Africa, 2003), pp. 241–247.

[15] Er, I. I., and Seah, W. K. G. Mobility-based d-hop clustering algorithm for mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference* (Atlanta, Georgia, USA, March 21-25 2004).

[16] Fernandess, Y., and Malkhi, D. *K*-clustering in wireless ad hoc networks. In *POMC '02: Proceedings of the Second ACM International Workshop on Principles of Mobile Computing* (New York, NY, USA, 2002), pp. 31–37.

[17] Gerla, M., and Tsai, J. T.-C. Multicluster, mobile, multimedia radio network. *Wir. Nets 1*, 3 (1995), 255–265.

[18] Krishna, P., Vaidya, N. H., Chatterjee, M., and Pradhan, D. K. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review 27*, 2 (April 1997), 49–64.

[19] Lin, C. R., and Gerla, M. Adaptive clustering for mobile wireless networks. *Journal on Selected Areas in Communications 15*, 7 (September 1997), 1265–1275.

[20] Selection procedures for the choice of radio transmission technologies of the (UMTS) (ts30.03 v3.2.0). (TS 30.03 3GPP), April 1998.

[21] McDonald, A. B., and Znati, T. A mobility-based framework for adaptive clustering in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks 17*, 8 (August 1999), 1466–1487.

[22] Parekh, A. K. Selecting routers in ad hoc wireless networks. In *Proceedings of the SBT/IEEE International Telecommunications Symposium, ITS 1994* (Rio de Janeiro, Brazil, August 1994).

[23] Ramanathan, R., and Steenstrup, M. Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. *Mobile Networks & Applications 3*, 1 (June 1998), 101–119.

[24] The VINT Project. *The ns Manual*. http://www.isi.edu/nsnam/ns/, 2002.

[25] Wang, L., and Olariu, S. A unifying look at clustering in mobile ad hoc networks. *Wireless Communications and Mobile Computing 4*, 6 (September 2004), 623–637.