DISTRIBUTED CLUSTERING ALGORITHMS IN GROUP-BASED AD HOC NETWORKS

R. Massin and C. J. Le Martret

Thales Communications & Security
Waveform Design Department
92622 Gennevilliers, France
Email: raphael-a.massin@thalesgroup.com

P. Ciblat

Institut Mines-Telecom/Telecom ParisTech
Communications and Electronics Department
75013 Paris, France
Email: ciblat@telecom-paristech.fr

ABSTRACT

For dense ad hoc networks, clustering is an appropriate strategy for improving scalability. Moreover, dense networks such as public safety or military networks are also structured through a hierarchical organization via operational groups. This organization usually impacts both the mobility of nodes which move in group, and the data flow since the traffic is mainly intra-group rather than inter-group. In this work, we extend two distributed clustering algorithms, GDMAC and VOTE, by taking into account the group structure. Our simulations of dense ad hoc networks show that our extensions lead to a lower end-to-end communication delay and offer a better stability to mobility.

***Index Terms*—** Ad hoc network, distributed clustering, operational group.

1. INTRODUCTION

Due to the large increase of communication applications and subcribers, wireless ad hoc networks have to handle more users and more data rate. To cope with these difficulties, dense networks are now often deployed. For scalability purpose, it is interesting to decompose a dense network into different clusters since this facilitates i) the radio resource allocation within the clusters due to their smaller number of nodes, and ii) the routing thanks to a decrease of the signaling overhead [1]. In ad hoc networks, there is no central manager and clustering has to be done in a distributed manner.

In some applications of ad hoc networks, such as public safety networks or military networks, the nodes are organized into a hierarchical structure leading to the existence of *operational groups* (e.g. squad, section, etc.). In those networks, nodes may exhibit group mobility behavior. Additionaly, the traffic may be strongly impacted by the network hierarchical organization, being mostly concentrated within operational groups. For both reasons, the clustering solution should take into account operational group information in order to provide more stable networks as well as better end-to-end QoS. For the sake of readability, in the following, *group* refers to *operational group*.

Numerous distributed clustering schemes independent of group structure have been proposed in the litterature. These algorithmic solutions first select Cluster Head (CH) nodes, and then the other nodes (so non-CH nodes) affiliate to CH nodes which leads to the different clusters. Usually a weight is associated to each node and the nodes with the highest weights in a neighborhood are selected to be the CH nodes. The weights can be the node identifier (e.g. MAC address), the node degree, the battery remaining power, metrics related to radio measurments etc., or a combination of them [2–4]. To get the node weight, other solutions rely on the knowledge of nodes' location and speed, typically obtained thanks to a GPS device [5, 6].

In contrast, only a few papers incorporate group information for building the clusters. The authors of [6] define the type-based clustering algorithm (TCA). This clustering scheme associates a stability factor to each node and selects as CH the nodes that have the highest stability factor in a radio neighborhood. The stability factor takes group membership (identified thanks to the IP subnet of each node) into account. A limitation of TCA lies in the fact that two CH nodes may not be neighbors. In dense networks a direct consequence is the formation of big clusters (with a large number of members). This problem is fixed by two distributed clustering algorithms – GDMAC [3] and VOTE [4] – which do not take into account the group structure.

The contribution of the paper is to adapt the powerful distributed clustering algorithms GDMAC and VOTE to the context of group-based ad hoc networks. A rigourous numerical evaluation is also done through the end-to-end network performance metric developed in [7].

The paper is organized as follows. The system model is described in Section 2. The new extensions of the distributed clustering algorithms adapted to group information are provided in Section 3. Section 4 is devoted to the numerical results. Concluding remarks are drawn in Section 5.

2. SYSTEM MODEL

We consider a graph $\mathcal{G}$ defined by its set of nodes $\mathcal{V}$ and its set of edges $\mathcal{E}$. The number of nodes of $\mathcal{G}$ is $N := |\mathcal{V}|$. Two nodes are neighbors if there is an edge between them in $\mathcal{E}$. A clustering solution leads to a partition p of $\mathcal{G}$. A partition contains N_c clusters. The nodes are organized in groups. The set of groups $\mathcal{O}$ is defined as $\{\mathcal{O}_1, \ldots, \mathcal{O}_{N_g}\}$ with N_g the number of groups. Let us note n_k^g the size of group $\mathcal{O}_k$. Each node belongs to only one group.

Similarly to [7], the user traffic depends on the group affiliation: the probability that one node communicates with a node of the same group is equal to $\alpha \in [0, 1]$ and thus the probability that one node communicates with a node in another group is equal to $1 - \alpha$. Since a node of group $\mathcal{O}_k$ can communicate to $n_k^g - 1$ nodes in the same group, the probability to reach one of these nodes is equal to $\alpha/(n_k^g - 1)$. The number of nodes of the other groups with which this node can communicate is equal to $N - n_k^g$ with a corresponding probability of $(1-\alpha)/(N-n_k^g)$. Let us define $\pi_{j|i}$ as the probability that a source node i communicates with a destination node j. If $j = i$, then $\pi_{i|i} := 0$. When $j \neq i$ we have:

$$\pi_{j|i} := \begin{cases} \dfrac{\alpha}{n_k^g - 1} & \text{if } j \in \mathcal{O}_k, \\ \dfrac{1-\alpha}{N - n_k^g} & \text{otherwise,} \end{cases} \tag{1}$$

with $(i, j) \in \mathcal{V}^2$ and $i \in \mathcal{O}_k$.

In clustered networks, inter-cluster communications can be implemented using a medium access control protocol different from the one used for intra-cluster communications. This is justified by the fact that within a cluster a node called resource allocator (RA) can manage the radio resource management (RRM) on behalf of the whole cluster, which allows the RA to optimize it locally. Conversely, inter-cluster RRM is done in a more distributed way (e.g. among the RAs of neighbor clusters) and is thus more difficult to optimize. Therefore, we reasonably assume that the performance (delay, data rate, etc.) of intra-cluster and inter-cluster communications are different. In this context to measure the performance of a clustering solution resulting in a partition p of $\mathcal{G}$, we use the performance metric defined in [7] and denoted by $J(p)$. This performance metric corresponds to the average cost of communications between all pairs of nodes in the network for additive metrics, and its value is for instance comparable to the average end-to-end delay from all nodes to all nodes. It is defined as:

$$J(p) := \frac{1}{N} \sum_{(i,j) \in \mathcal{V}^2} \pi_{j|i} \cdot J_{i,j}(p), \tag{2}$$

where the factor $1/N$ embodies the fact that all nodes i have equal probability to transmit. $J_{i,j}(p)$ is calculated as the cost of communication between source node i and destination node j along a shortest path. This shortest path is calculated taking into account the different costs $\tilde{\gamma}$ for inter-cluster and $\hat{\gamma}$ ($< \tilde{\gamma}$) for intra-cluster communications.

3. NOVEL DISTRIBUTED CLUSTERING ALGORITHMS

We have selected the two distributed clustering protocols GDMAC [3] and VOTE [6] because they allow to adapt the number of clusters to the network node density. In this section, we provide the main contribution of the paper by describing the proposed extensions of GDMAC and VOTE. They have been designed in order to take into account the group structure of the network.

3.1. The extension of GDMAC

The goal of the distributed clustering protocol GDMAC [3] is to build stable clusters in presence of node mobility. In this protocol CH nodes are elected first, and then non-CH nodes affiliate to a neighbor CH, leading to the clusters. To do that, a weight (depending on the context) is associated with each node.

Within a radio neighborhood, nodes with the highest weights are chosen as CH nodes. To increase stability in presence of mobility GDMAC introduces the K parameter, whose value is equal to the number of CH nodes that are allowed to be neighbors of a $(K + 1)$th CH node. A non-CH node affiliates to the CH node within its neighborhood whose weight is highest. In order to obtain stable clusters, a non-CH node remains affiliated to its current CH unless there is a CH node in its neighborhood whose weight exceeds the one of the current CH node by at least a positive lower-bound denoted by H. Note that in the GDMAC paper [3], the weight is allocated randomly. In this article, we prefer to use the node identifier as node weight, as in [2]. We refer to this approach as GDMAC-std.

To extend GDMAC so as to take group membership into account, we propose to calculate the node weights in a different way, and also to modify the way non-CH nodes affiliate to their CH. This leads to two new versions of the GDMAC, denoted by GDMAC-new1 and GDMAC-new2 described hereafter.

GDMAC-new1: the weight used is the so-called stability factor defined in [6]. In that original paper the stability factor has been introduced for clustering group based networks but is associated with a very simple algorithm. Here we thus propose to associate this stability factor with GDMAC. The stability factor of a node is a linear combination of the average of relative speeds with its neighbors, the average of distances with its neighbors, the average number of neighbors, and its remaining energy. The three mentioned averages are weighted averages. In order to take into account the group structure, higher weights in the averages are used for neighbors that are members of the same group as those of the current node. The TCA protocol from [6] can be seen as a particular case of GDMAC-new1 by setting the GDMAC parameter K to zero. We call it TCA-std.

GDMAC-new2: it is an extension of GDMAC-new1 where we modify the non-CH node affiliation strategy. Indeed, a non-CH node affiliates, if possible, to a CH node that is also member of its group instead of choosing it with respect to its weight.

According to the proposed modifications, GDMAC-new1 and GDMAC-new2 are expected to be better suited to the context of group based networks.

3.2. The extension of VOTE

Similarly to GDMAC, VOTE [4] selects CH nodes based on their weight, and then non-CH nodes affiliate to a neighbor CH node. In VOTE, the weight of each node is called its *vote* and is a linear combination of the normalized degree and the battery time remaining. The main difference between GDMAC and VOTE does not lay in the weight definition but in the fact that each CH node manages its cluster size, i.e. the number of nodes (including itself) in its cluster. With VOTE the cluster size is higher bounded by n_{max}. When the number of affilated nodes to a CH node is equal to this value, non-CH nodes refrain to affiliate to this CH. Nevertheless, in the case of simultaneous affiliations a cluster may include more than n_{max} members. Then, the concerned CH randomly rejects as many members as required to satisfy the cluster size constraint.

We here propose to apply the algorithm VOTE (denoted by VOTE-new) by using the stability factor as the weight in order to take into account the group structure. Initial VOTE defined in [4] is hereafter denoted by VOTE-std.

4. SIMULATION RESULTS

4.1. Simulation setup

The simulated network has N nodes deployed in a square area whose side is 750 distance units long. The range of a node is 250 distance units. The nodes are organized in groups of 10 nodes, using a rule derived from the group mobility model defined in [8]. In this model, the nodes of the same group never move farther than a specific distance from the virtual center of the group. In our simulation, this distance has been set to the node range. If the virtual center of a group is closer than this distance from the area boundary, then a new location is drawn for this virtual center to make sure all nodes of the group are always in the area. Simulations have been performed in static and mobile conditions.

As for the traffic parameters, we set $\alpha = 0.9$, meaning that 90% of the traffic is exchanged between members of the same groups. As for the communication costs, we fix $\hat{\gamma} = 1$ and $\tilde{\gamma} = 2$.

The clustering algorithms parameters common to all simulations are now detailed. We consider $H = 10$ for GDMAC-std, and $H = 30$ for GDMAC-new1 and GDMAC-new2. Due to the difficulty to set its value, the values of K will be selected later. To calculate the stability factor, the parameter related to the number of samples to calculate the relative speeds and distances between nodes (denoted by N in [6]) is fixed to five, and the parameter related to the contribution of neighbors of the same group compared to the others nodes (denoted by λ in [6]) is 0.5. Finally, n_{max} is set to 20.

We used a custom simulator which splits time in rounds. During a round, all nodes first run the selected clustering algorithm independently: CH nodes are maintained and non-CH nodes select their CH node. At the end of the round, all nodes update the knowledge of their neighborhood including all information required to run the selected clustering algorithm in the next round.

All simulation results are run over 100 random networks.

4.2. Performance in static networks

Most simulations have been performed for various values of N. As the area is identical, the higher N is, the more dense the network is. For instance, when N goes from 100 to 1000, the average degree grows linearly from a moderate density of about 20 to a high density of about 150. Whatever the value of N, the number of simulation rounds is always chosen large enough in order to ensure the convergence to a stable cluster structure.

4.2.1. Choice of K for GDMAC

The values of cluster size built by GDMAC are overspread since very small clusters coexist with very large clusters. This problem does not occur with VOTE since the cluster size is upper-bounded by n_{max}. In order to have a meaningful comparison between GDMAC and VOTE, K is set for reducing cluster size variability. For each GDMAC extension and each network size, we have determined through simulations the best value of K as indicated in Table 1. The selected value is the lowest one getting an average highest cluster size no greater than n_{max}. GDMAC-std succeeds in filling this criterion only for 100 and 200 nodes. Consequently static networks GDMAC-std simulation results are not provided in Section 4.2.

Nodes	100	200	300	400	500
GDMAC-new1	2	3	5	7	9
GDMAC-new2	1	2	4	6	8
Nodes	600	700	800	900	1000
GDMAC-new1	11	13	17	20	22
GDMAC-new2	11	13	17	19	23

Table 1. K vs. N for proposed GDMAC.

4.2.2. *Cluster size*

In Fig. 1, we plot the average number of members in a cluster with respect to N.

Because TCA-std only allows one CH node in a neighborhood, the cluster size is expected to increase with the number of nodes. This is exactly what can be seen in Fig. 1. This behavior is undesired because in clusters whose size is not upper-bounded, it is not possible to ensure efficient radio resource allocation. Consequently, even if TCA-std is the only reference [6] from the state of the art taking into account group membership, we no longer consider it in the remainder of this paper. Thanks to the appropriate choice of GDMAC K parameter, GDMAC and VOTE based algorithms yield clusters whose maximum size of clusters is almost the same. Yet, their average cluster size are very different.

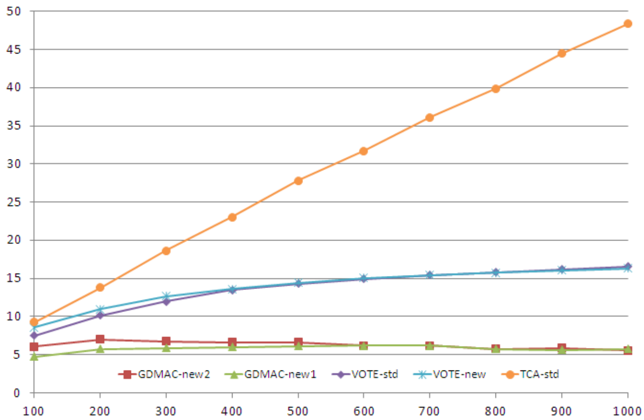

Fig. 1. Average cluster size vs. N in static networks.

The GDMAC based algorithms lead to an average cluster size much smaller than n_{max}, meaning that these protocols build clusters with highly different sizes. The reason of this behavior lies in the number of CH which is fixed for GDMAC. A more detailed analysis shows that GDMAC-new2 builds more balanced clusters. This is due to their different affiliation procedure. With GDMAC-new1, within a neighborhood the CH with the highest weight attracts a lot of nodes in its cluster, and only a few nodes affiliate to CHs with lower weights. GDMAC-new2 ensures that non-CH nodes affiliate to a neighbor CH of the same group, if any. Thanks to this rule, if only one CH of a given group exists in a radio neighborhood, this CH has at least as many cluster members as the number of its neighbors belonging to its group. This explains why GDMAC-new2 builds clusters whose size is more balanced than the ones built by GDMAC-new1. Because the number of CH nodes must always be $K + 1$, the average cluster size is similar for both GDMAC-new1 and GDMAC-new2.

VOTE-std and VOTE-new determine the number of CH dynamically, and build clusters whose size is upper bounded by n_{max}. This leads to clusters of similar size, increasing with the number of nodes in the network. Yet the number of small clusters is still high. Even if VOTE-std and VOTE-new build clusters with higher average size than GDMAC-new1 and GDMA-new2, the cluster size variability is still high.

4.2.3. *Group cluster diversity*

The group cluster diversity (GCD) counts the per group average number of clusters with at least one member of this group. A low GCD value leads to better overall performance since it indicates that group members are gathered in few clusters. This means that a majority of low cost intra-cluster communications and a minority of high cost inter-cluster communications are used during intra-group communications. Let us define $\triangleleft$ a comparison operator between two clustering schemes clu_1 and clu_2 such as $clu_1 \triangleleft clu_2$ if the GCD of clu_1 is greater than the GCD of clu_2.

In Fig. 2, we plot the GCD with respect to N. We have GDMAC-new1 $\triangleleft$ VOTE-std $\triangleleft$ VOTE-new $\triangleleft$ GDMAC-new2. Neither GDMAC-new1 nor VOTE-std nor VOTE-new take group membership during non-CH node affiliation. This leads to a higher GCD. As VOTE-std or VOTE-new result in larger clusters, their GCDs are better than those of GDMAC-new1. Finally, thanks to its new affiliation procedure taking into account the group structure, GDMAC-new2 provides the smallest GCD.

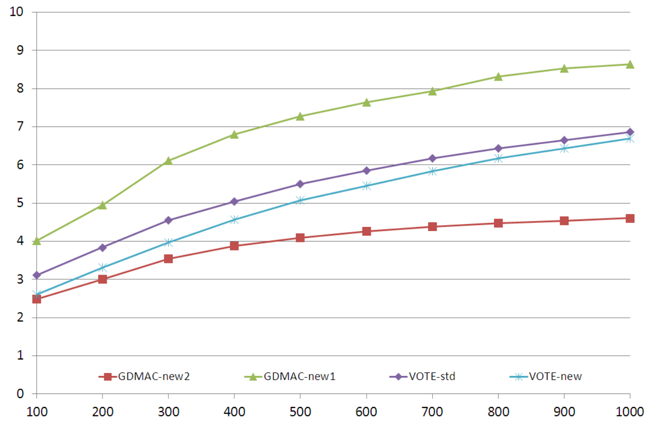

Fig. 2. Group cluster diversity vs. N in static networks.

4.2.4. *Application level performance*

In Fig. 3, we plot the values of the metric J with respect to N. The extra curve, called *singletons*, corresponds to the values of J when the network is only composed of singleton clusters (i.e. each node is its own cluster). This singletons curve provides the upper-bound of the values of J for the considered networks.

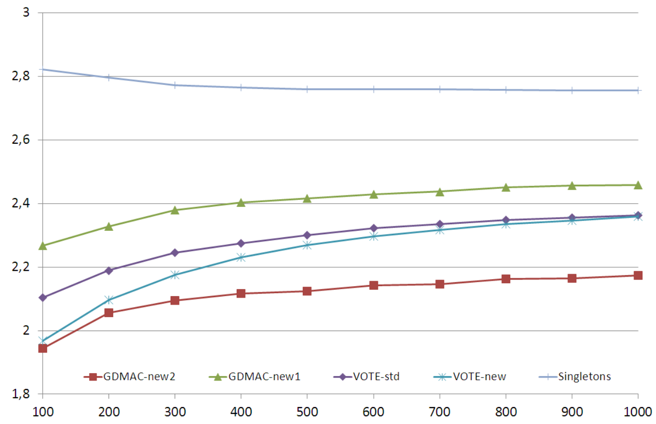

Fig. 3. J vs. N in static networks.

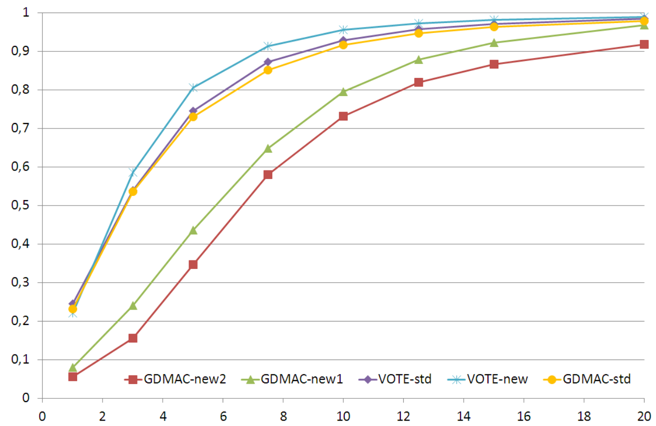

Fig. 4. Mobile network instability vs. maximum speed.

Let us define $\lhd_J$ a comparison operator between two clustering schemes clu_1 and clu_2 such as $clu_1 \lhd_J clu_2$ if the J value achieved by clu_1 is greater than the one of clu_2. We have GDMAC-new1 $\lhd_J$ VOTE-std $\lhd_J$ VOTE-new $\lhd_J$ GDMAC-new2. The good results for GDMAC-new2 can be explained by the small values of GCD: J decreases when the amount of inter-cluster traffic decreases, which is the case when members of the same group are included in a small number of clusters, as achieved by GDMAC-new2.

4.3. Performance in dynamic networks

In this sub-section we simulate mobile networks with $N = 100$ nodes. To assess the stability of the cluster structure built by the different clustering solutions, we define the ratio of the number of simulation rounds when at least one cluster was modified over the simulation time (set to 1000 rounds). Small values of this ratio indicate stable clustering algorithms, and values close to 1 indicate unstable networks. In Fig. 4, we plot this ratio with respect to the maximum node speed.

Thanks to their small cluster size and their use of group information, GDMAC-new1 and GDMAC-new2 lead to the best stability. The other three schemes are highly unstable. VOTE-std and VOTE-new build larger clusters than all GDMAC extensions and have the worst stability. The way VOTE-new uses group information does not help to build clusters that are more stable (than VOTE-std). The initial protocol GDMAC-std does not take into account group membership at all and leads to instable clusters.

5. CONCLUSION

We have proposed and simulated extensions of existing distributed clustering algorithms suited to group-based networks. Our simulations have shown that existing clustering schemes have poor performance in such networks, and that our extensions offer some significant gains. More importantly, we have remarked that the performance are dramatically improved, on the one hand, when the clustering algorithm properly handles the cluster size, on the other hand, when nodes of the same group belong, as much as possible, to the same cluster. We are now working on new distributed clustering algorithms that take into account these two design goals.

REFERENCES

[1] J. Sucec and I. Marsic, "Hierarchical Routing Overhead in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 46–56, Mar. 2004.

[2] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio network," *Journal of Wireless Networks*, vol. 1, no. 3, pp. 255–265, Jul. 1995.

[3] R. Ghosh and S. Basagni, "Mitigating the impact of node mobility on ad hoc clustering," *Wireless Communications and Mobile Computing*, vol. 8, no. 3, pp. 295–308, Mar. 2008.

[4] F. Li, S. Zhang, X. Wang, X. Xue, and H. Shen, "Vote-based clustering algorithm in mobile ad hoc networks," in *Proc. ICON*, Busan, South Korea, Feb. 2004.

[5] Y. Zhang and J. Ng, "A Distributed Group Mobility Adaptive Clustering Algorithm for Mobile Ad Hoc Networks," in *Proc. IEEE ICC*, Beijing, China, May 2008.

[6] H. Wu, Z. Zhong, and L. Hanzo, "A cluster-head selection and update algorithm for ad hoc networks," in *Proc. IEEE GLOBECOM*, Miami, USA, Dec. 2010.

[7] R. Massin, C. J. Le Martret, and P. Ciblat, "A network cost function for clustered ad hoc networks: application to group based systems," in *Proc. IEEE PIMRC*, Washington D.C., USA, Sept. 2014.

[8] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A group mobility model for ad hoc wireless network," in *Proc. ACM MSWiM*, Seattle, USA, Aug. 1999.