Topology Management for Group Oriented Networks

Daniel Câmara, Christian Bonnet, Navid Nikaein
Mobile Communications Department - EURECOM
Sophia-Antipolis, France
Email: {daniel.camara, christian.bonnet, navid.nikaein}@eurecom.fr

***Abstract*—This paper proposes a topology management mechanism for hierarchical group oriented networks. The key innovative aspect of this paper is to consider different interest groups in the clustering formation process. For administrative or performance reasons sometimes it is required to organize nodes, located in the same geographical area, into different groups. This group-oriented topology management provides optimal routing for group members as well as for the last miles communication, where you only need to broadcast the message once to reach the entire interest group. Furthermore, it significantly improves the service provided to upper layers by creating a stable network topology. For example, when establishing a Public Safety Networks (PSN) to handle a disaster scenario the communication from the rescue teams may have no interest, or relation, with the communication hold by the law enforcement teams. In this case it makes sense to divide the network in two subgroups to improve routing and simplifying the medium access control. Here we propose a hierarchical network architecture to enable the organization of nodes into multiple interest groups in the same geographical area. Through simulations, we will show that how this architecture could be adapted to different PSN requirements and what are the key parameters.**

I. Introduction

The greatest part of the topology management algorithms for wireless networks use proximity, signal strength, or combined metrics as the main criteria to organize nodes into clusters. However, in many situations it makes more sense to organize nodes regarding their interests or some common label than taking just proximity into account. To the best of our knowledge, this is the first work to consider interest groups in the cluster formation process. Hui and Crowcroft [7] show that nodes that have a common interest tend to met more often in the network what may simplify the message exchange process. Grouping nodes into groups of interest may also help decrease the whole network amount of traffics because nodes would exchange messages just with nodes inside the same group of interest. In this case common interest messages, such as multicast or group related information would be narrowed to be spread just among the nodes that really have interest on it.

Public Safety Networks (PSNs) is a good example of a kind of environment that could take advantage of grouping nodes according to their role in the rescue efforts. For example, in an earth quake relief effort site, the status and conditions of nearby roads may interest police officers or ambulance drivers, but has little or no importance for the rescue teams digging for survivors. Decreasing the number of irrelevant messages not only save resources by decreasing the amount of traffic in the network but also help the rescue teams to keep their attention focused in their jobs. For instance, if one keeps sending messages that have no interest to the rescue team, really relevant messages may be lost.

We consider here an architecture where nodes are organized into clusters and each node has two interfaces, one for intra cluster communication and other, with higher range and throughput, for inter cluster communication. The first interface could be, for example, WiFi and the second one WiMAX. Figure 1 presents an example of the interest groups setup. We consider that nodes may have a second interface used to create a backbone of communication among the clusters, regardless their specific interests. In this way, the second interface increases the network connectivity and ensures that nodes with specific interests, even when far from other nodes with the same interest are still able to receive their messages.

The heterogeneity in the nodes and in type of networks (i.e. Tetra and WiMax) are something common, mainly in special purpose networks such as military and PSNs, and requires rapid integration of available communication resources in the event of crisis or disaster [11]. For example, the capacity and the autonomy of a radio mounted in a fire truck should be, normally, higher than the radios used by the field agents. And the network used by the fire truck might be different from that of the field agent. This work is inserted in the context of the efforts of the HNPS project (Heterogeneous Network for European Public Safety) [11]. Even though we do not rely on it, our solution also admits, and is able to handle, heterogeneous nodes in the network.

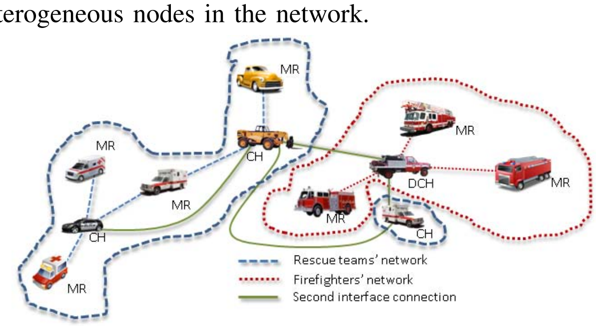

Fig. 1. Interest groups architecture, showing two different interest groups and the second interface links

The remainder of the paper is organized as follows. In Section II we highlight some of the related work in the literature. Section III outlines the objectives of the proposed topology management algorithm. Section IV presents the market based technique used to control the network topology. In Section V we present the interest groups architecture and protocol description, and in Section VI we present the experiments. Finally, we present the conclusions and future work in Section VII.

II. Related Work

The idea of dividing the network nodes into different groups is not new, other researchers have also worked on this field. Among the advantages of this kind of technique Hui and Crowcroft [7] highlight the decreasing in the amount of transmitted traffic. Hui and Crowcroft [7] have make an interesting experiment where they distributed sensor among the participants of IEEE Infocom 2006. From these experiments they divided the users into their academic affiliation and tracked the users mobility pattern. As a result of their experiment they discovered that using labels to indicate the nodes affiliation can reduce the delivery cost, without trading off much against delivery ratio. Nodes of users inside the same group tend to met more often than nodes from different defined communities. Even though this experiment was restricted to a conference environment the same community oriented behavior can be observed in many other real situations.

To control the clustering formation we are using a market based like approach. In [4], Mainland et al. propose the market-based macro-programming paradigm for controlling the behavior of the nodes in a sensor network. Even though the main focus of their work and ours is different, both have the same inspiration. We use the free market economic concept to control the network nodes' behavior and reach stable final configurations. The first welfare theorem states that any free market system will eventually reach Pareto optimality [5]. A Pareto optimal allocation is the one where no one could be made better off without making someone else worse off. In other words, a Pareto allocation is a fair equilibrium point. It is the best allocation one can expect to reach and any change could hurt some of the participants.

Our approach consists of creating a free market environment where nodes can trade the connections freely [9]. We consider that the quality of the service offered by two distinct providers is the same. Each node is free to set its prices, and these vary in accordance to the node load and type; however, among nodes of the same class the basic price is the same. Nodes are free to choose their provider and to change providers, if they have some gain in doing so. In our final setup no node wants to or can change providers without paying more and no provider can increase prices without losing clients. Thus this Market Based Strategy (MBS) reaches an equilibrium that is Pareto optimal.

III. Objectives

The final purpose of this work is to provide auto organization capabilities to the network that will lead to the creation of stable topologies. To accomplish this, the algorithm we propose here has four main objectives.

1) Ensure a stable, or at least as stable as possible, network as fast as possible while respecting the desired architecture. The topology and mechanisms to guarantee connectivity should be stable, trustworthy and rapidly deployable.
2) Creation of homogeneous clusters. Clusters should not only have roughly the same size but it is also important to be able to control and fine tune the network shape and cluster sizes. Cluster heads must be able to optimally handle communication among nodes inside their clusters and exchange key information with neighbor nodes rapidly and efficiently. The optimal number of clusters and elements by cluster vary from one scenario to another so having an easy and standard mechanism to control this size is an important feature.
3) Keep the number of clusters as low as possible, while keeping the clusters with a reasonable size. Having the minimum number of clusters possible not only decreases the number of required RNs but also decreases the number, and size, of control messages in the final network.
4) Ensure that the resulting topology is spanner, meaning that it is connected and longer by a constant factor than that of shortest path. Indeed, having an hierarchy normally imposes some restrictions in the routing, however, we want this impact to be as small as possible in the path lengths.

The technique described here is able to create and maintain well-defined wireless mesh network architectures in a flexible and dynamic way. We are able, by just adjusting a set of parameters, to change the behavior of the whole network without deploying new equipments or protocols. The algorithm is able to provide an easy way to change the network behavior, i.e. number and size of clusters, while respecting the topology constraints. The proposed scheme is general and able be adapted to any wireless mesh network architecture. On the top of that, the algorithm also observes the interest of each node and places them within the appropriated interest group.

IV. Market-based topology management

The Market Based System (MBS) described here intends to create and maintain well-defined wireless mesh network architectures in a flexible and dynamic way. The technique in fact has the power to change the whole behavior of the network by adjusting a small set of parameters, without the need for special equipment or complex protocols.

We base our solution on the economy laws of supply and demand to dynamically organize the network. The first law of supply and demand states that when demand is greater than supply, prices rise and when supply is greater than demand, prices fall. The power of such forces, rise and fall, depends on how great the difference between supply and demand is. The second law of supply and demand, then, states that the greater the difference between supply and demand, the greater is the force on prices. The third law states that prices tend to an equilibrium point, where the supply is equal to the demand [6].

If we align our objectives with the laws of supply and demand we will see that these three laws map perfectly the main requirements of a topology management algorithm. We may map our need to control the number of clusters to the first law of supply and demand. Controlling the prices of each kind of service offered in the network, we can control the number of elements offering such service. The second objective is to have a fast convergence to a stable state. This requirement is met by applying the second law, since the bigger are the differences among supply and demand the faster is the convergence. Finally, recall that our third objective is to maintain a well balanced and as stable as possible network, while respecting the desired architecture. Clusters should not only have roughly the same size but we should have an easy way to control and fine tune that size. These issues are covered by the third law, since the final topology is expected to be a Pareto optimal arrangement [5] and hence it should be stable and fair among all the participants.

The proposed mechanism emulates a free market where nodes are free to change service providers if the cost of the present provider is greater than other nearby. Nodes increase their connection prices also based on the number of nodes they are attending. However, if their prices start to be too high nodes will tend to go use the services of the concurrent servers. With this mechanism the clusters formation is fair and homogeneous.

V. Interest Groups Architecture

The architecture we propose here is a two-level hierarchical one, with the formation of clusters maintained by one cluster head. The architecture admits heterogeneous nodes. To represent this, we will have a set of special nodes that are set as cluster heads by default, i.e. Default Cluster Heads (DCHs). These nodes maintain this status through the whole network's life time. Regular nodes do not need to be close to the DCH. Nodes far from the DCH, or when this is overloaded, should autonomously organize themselves into clusters. We consider that any node may become a Cluster Head (CH) if outside the areas covered by DCHs. The number of interest groups may vary from 1 to N groups. Even though we consider a maximum cluster size, the bigger the number of groups the smaller tends to be the size of the clusters. Our architecture admits a maximum cluster size because this is a requirement for some technologies e.g. Bluetooth.

For ours experiments each interest group is defined in the network startup and must have at least one DCH to represent it. In the experiments only the DCHs have a defined interest group at the beginning of the simulation, and the different groups are attributed evenly to the available DCHs. The interest group of regular nodes is defined by the DCH nearby through the periodic broadcast of connection update messages.

Apart from the CH and DCH nodes no other node receives messages from nodes from different interest groups. Even CH and DCH only consider the received Connection Update messages from nodes in different groups. We do not consider data message, as they are not relevant to the clustering formation purpose. In the regular setup case nodes are supposed to use the second interface also to transmit data.

A. Interests group protocol description

In our approach both CH and DCH send periodically connection update messages announcing their presence, interest groups, and list of connected nodes through two interfaces. The two interfaces have different purposes, the first one, denoted as default interface, is typically used to organize the communication with nodes closer to the CH (WiFi like interface) and the second interface is used to reach farther nodes and with a broader bandwidth capacity (WiMAX like interface). Each connected node, MR, sends also a periodic connection update message only to those nodes it is attached to, i.e. within the same interest group. When arriving via the default interface, it may change the status of the nodes receiving it. If it arrives via the second interface, it is just stored as a way to build the knowledge of the clusters around. Both CH and DCH update messages are sent through two available interfaces.

When a connection update message arrives via the default interface, the associated cost function is compared with that of the current provider. If the cost of this new provider is smaller than the cost of the current provider, a node sends a connection request to this new provider. Note that in addition to a CH/DCH, a MR can be a potential provider in which case it will change its status from MR to CH. A node also sends a connection request if it is an isolated node (IN) or an isolated CH/DCH. This is because the cost associated to an isolated node is set to be higher than that of a connected node, which in turn gives incentives to get connected rather than stay isolated.

When a CH/DCH receives a connection request and it has enough resources to host a new member, it sends a connection response to the node that requested the connection and reserves the resources for this node. If it does not have enough resources, the CH/DCH sends a connection cancel. Case the MR receives a connection cancel in response to a connection request, it forgets the request and waits for a new opportunity to connect to another node or, if it stays as an IN for a long time, three attempts with different bakeoffs intervals, it attempts to become a CH.

When an IN/MR/CH/DCH receives a connection response from a CH/DCH, it releases its resources (its connections), sends a connection confirmation to the CH/DCH that sent the message and registers this new node as the provider. Case the node is an IN/CH it changes state to become a MR. The provider CH/DCH that receives a connection confirmation it updates the information regarding this connection. However, if the node did not send any connection request in the first place, i.e. the message was a mistake, the node sends a connection cancel to the node that just sent the connection response.

If a CH/DCH receives a connection cancel, it releases the resources allocated to this connection.

Cluster splits occur mainly due to the group mobility. The MRs that moved away from the DCH keep sending and receiving connection updates messages. At some point one of the two things will happen with this group. Either the number of attempts to perceive a CH for one of the MR will reach the predefined limit or the cost of one of the MRs will become more attractive than the cost of others. For the first case that MR node will announce itself as a CH and the neighbor nodes, when receiving the connection update, will tend to connect to the new CH. In the second case another node will send a connection request to this node with a better cost. This message than triggers the status change of that MR, which will declare itself as a CH.

Cluster merge can happen when the two clusters belonging to the same interest group are close enough to hear each other via the default interface. A CH/DCH receiving the connection update message from another CH/DCH verifies the cost of this new provider. If the cost is smaller than the cost of its own connection, and the perspective provider has room to accommodate all the present connections (i.e. cluster members), the node sends a connection request to the node that sent the update message, which in turn triggers the connection request of the cluster members to this new provider.

1) Connection Price: Each element in the proposed solution has a connection price. The prices vary among the different nodes, and, normally, the price to pay for a connection to a CH should be higher than the price to pay for a DCH. Standalone CHs/DCHs have also higher connection costs than the ones that already provide connection service to some nodes. The load of the CH/DCH also counts, as the higher is the load of the CH/DCH, the higher is its connection cost. The idea is to have more balanced clusters, however, the costs are attributed in a way that guarantees that all the available resources of an available DCH should be used before a new CH/DCH start to accept connections in the same region. The order of communication costs depends on the type of a provider as follows: DCH providing connection $\leq$ isolated DCH $\leq$ CH providing connection $\leq$ isolated CH $\leq$ MR $\leq$ IN. The costs of the CH/DCH providing connection increases with the number of connections it is handling. This enables a more homogeneous use of the available resources, since less loaded clusters will handle new connections. For example, if an isolated node has two options, a CH providing connection to 5 other nodes, and another CH providing connection to 6 other nodes, it will prefer to connect to the first one. Furthermore, the number of connected nodes to a CH/DCH is upper bounded by the maximum cluster size. To avoid a ping-pong effect between near clusters with close costs changing providers has also a small cost. In this way nodes will change cluster only if the cost of the second cluster is really attractive.

In order to facilitate the notion of cost, we need to map all the related parameters onto a single cost metric, which can be compared and whose minimum can be chosen. Suppose c_p denote the initial cost for each type of provider, k the number of connected nodes, c_i the individual cost for each connected nodes to this provider, s maximum number of connected nodes in a cluster (i.e. size of a cluster) and c_c is the extra cost for changing providers. We consider the cost function of a node n to be:

$$Cost(n) = \begin{cases} \left(c_p + \sum_{i=0}^{k} c_i + c_c\right) \mid n, & k \leq s \\ c_p + \alpha \times s & k = 0 \\ \infty, & k > s \end{cases}$$

where $\mid$ represents the concatenation operation and α the additional weight applied to the isolated nodes. The c_c captures the extra cost a node should pay if it changes its current provider, and it is non-zero when changing the provider and zero elsewhere. To ensure the uniqueness, the cost is concatenated with the unique identity of the node n. In the above formula, the costs of a given provider increases with the number of connections it is handling. This cost increases with the cluster size when a node is isolated, and becomes infinite when the maximum cluster size is reached, i.e. no resource is available. For instance if we assume the maximum cluster size to be 10, the initial cost of a DCH 0, number of connected nodes 3, individual cost 1, and the changing provider cost 0, the cost becomes 3. Note that in the above formula, we can assume s to be the total transport capacity of a CH/DCH and $\sum_{i=0}^{k} c_i$ to be the sum of the capacity portion utilized by each connected node. As a result, the total capacity required by the network scales with the number of CH/DCH, and since CH is a temporary status of a node, this capacity should only scale with DCH.

VI. EXPERIMENTS

For these experiments the DCH are randomly assigned and are static during the whole experiment. All other kind of nodes are spread in a uniform distribution way and move in a random way point fashion in a $1.5km \times 1.5km$ area for one hour simulation time. Each node has two communication interfaces, the communication range for the first interface is $200m$ while for the second interface is $300m$. We varied the network density, connection weights, percentage of DCHs and number of interest groups. We considered here that the maximum allowed size for a cluster is 10. We varied the number of interest groups and percentage of DCHs. We evaluated four different set of basic cost for the nodes. In the Configuration 1 the basic cost to connect to a DCH=0, CH=3, MR=10. The connection costs for Configuration 2 are DCH=0, CH=5, MR=10, for Configuration 3 the costs are DCH=0, CH=10, MR=20 and for Configuration 4 the costs are DCH=50, CH=50, MR=0. The intention with these four different costs was to evaluate if the network responds to the proposed market based method. The third configuration is the standard case, for the experiments that does not mention it. Configuration 3 takes into account the maximum size of the cluster, set to these experiments as 10 nodes. The purpose of this configuration is to instigate nodes to connect to first to DCH, if these are not available, than to connect to CHs and in the last case connect to a MR, that will than became a CH.

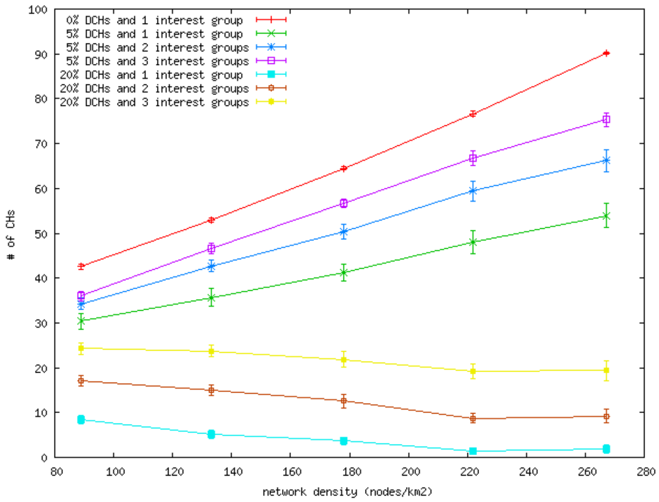

Fig. 2. Variation of the average number of CHs in the network when we increase the percentage of DCHs

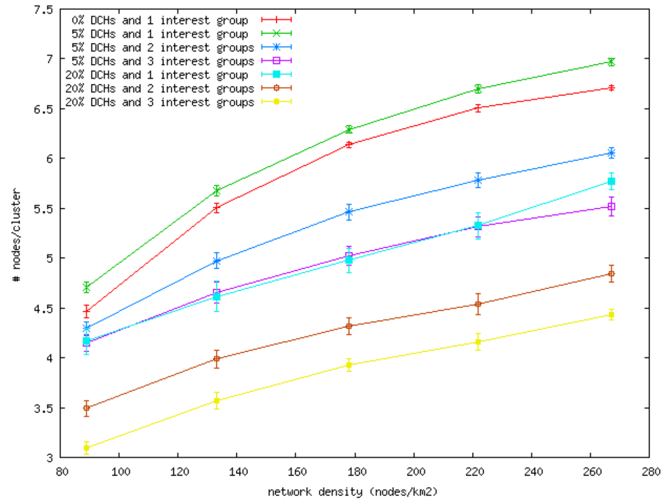

Fig. 3. Average cluster size CHs in the network when we increase the number of interest groups

Configurations 1 and 2 are an attempt to verify if with slight changes in the basic connection costs we are able to control the network behavior. For the fourth configuration we want to evaluate a situation where we incentive the nodes to connect to MR, that will, because of that became a CH.

The graph in Figure 2 shows the average number of CHs as the network density increases (i.e. the number of nodes per km^2) when we vary the percentage of DCHs and number of groups. CHs are created only when nodes are either outside the area of a DCH or when the DCH have not enough resources to grant the node's connection requirements. We can see that the increasing in the percentage of DCHs decreases the number of CHs. When we have a 20% DCH distribution the number of CHs reaches a saturation point and the number of CHs required is stable, even slightly decreasing for more dense networks, where the full power of the DCHs can be better explored. This means that the 20% DCH distribution were enough to supply the network with the needed DCHs. When we increase the number of interest groups the number of CHs also increases. This is expected since when we increase the number of groups is equivalent to split the network, the bigger the number of groups the harder is for a node to find a nearby cluster within the same interest. For the other experiments, where there were either less or no DCH at all, as expected the number of CH required for satisfying the network needs increases with the number of network nodes. It is important to notice that adding only 5% DCHs decreases from 19.5% to 37.2% the number of required CHs in the network.

Figure 3 shows that the average size of clusters decreases with the increasing in the number of DCHs and interest groups. This makes sense since the increase in the number of DCHs increases the attachment options for the nearby nodes. Note also that the average size of cluster increases with the network density. For these experiments, on average, the clusters did not reach the maximum defined cluster size, which is of 10 nodes. The main factors responsible for this are the mobility and the creation of new clusters. When a new cluster has to be created, its size will be 1, only the CH. This decreases the average size of the clusters. However, it is important to say that, during the experiments, when the maximum value was reached the designed cost function could control the nodes behavior and form new clusters.

Figure 4 presents the average percentage of the network that is disconnected during the simulations. We can observe that the increase in the number of DCHs help to stabilize the network and decrease the number of disconnected nodes. In reality the network, with the addition of DCHs is more stable, the number of states changes vary from 20% to 50% in the observed configurations. Great part of the state changes are from CH or MR to IN, what means that the node is disconnected and perceive it is isolated from the rest of the network. Furthermore, increasing in the network density increases the network connectivity. In the results presented here, to be considered as connected, the nodes must have established a link between them and have to be inside the communication range, otherwise their connection is not considered.

Figure 5 shows the average path size for any source and destination versus the percentage of DCH in the network. We consider here only the paths between nodes that are connected; this is why for the 89 nodes per km^2 configuration, a sparse network, the size of the paths is smaller and the values appears to be inverse. The denser the network the more nodes are reachable and we also have a more diverse options of paths. This is why the path size from the 89 nodes per km^2 network to the 133 nodes per km^2 nodes increases and after that it decreases. The increase in the number of DCHs decreases the average size of the paths and the increase in the number of groups increase the average path size. This is expected because the increasing in the number of DCHs helps to stabilize the network while the increasing in the number of groups makes it harder to find nodes in the same group to exchange messages. Note that in our simulation setup, the area is 2.25 km^2 and the transmission range on the second interface is $300m$. Given the simulation setup, theoretically the maximum diameter of the network is 7 hops, and thus the average becomes 3.5 hops. The simulation result shows that on average the path size is 4.5 hops (the maximum diameter is 9 hops). This indicates that the stretch factor of the resulting topology is 1.28 spanner.

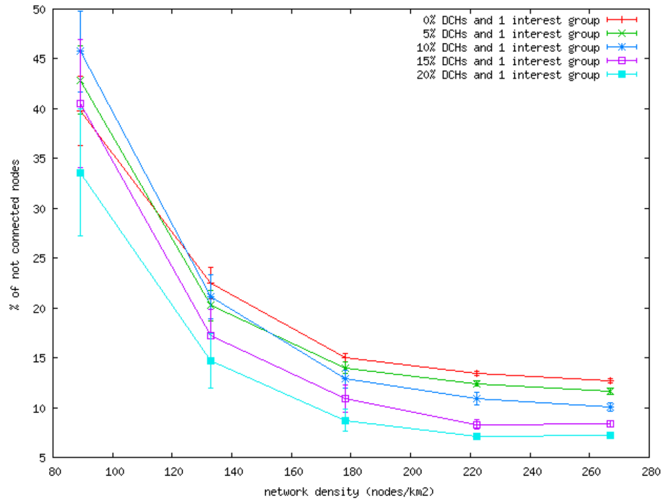

Fig. 4. Percentage of network disconnection

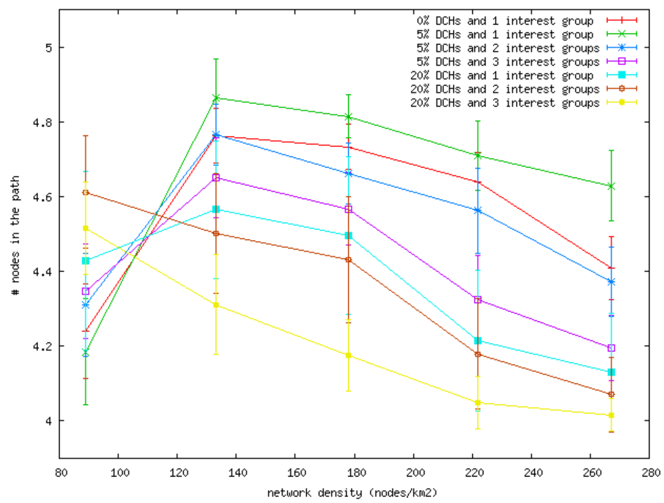

Fig. 5. Average Path Size

This is also expected because, in this case, all communications should pass through the CH/DCH, which in turn add two extra hops to the path from the source and destination nodes to the corresponding CH.

The graph Figure 6 shows the average size of the clusters for the different configurations. We can see that controlling the basic cost of the network elements we can change the behavior of the network as a whole, even in a fine grain manner. The small variations provided by the configurations 1 to 3, affect the network exactly the way they were expected to do. This shows the potential of the technique we are proposing here. It enables, just by the manipulation of a small set of parameters a fine tuning of the network topology.

VII. Conclusions

This paper presented a technique to perform topology management for group based mesh networks. In the best of our knowledge this is the first work to consider interest groups in the cluster formation of wireless mesh networks. Our results show that the proposed technique, by just handling local information, and without the complete final configuration, the proposed method is able to guarantee the correct cluster formation and role attribution to the entire network.

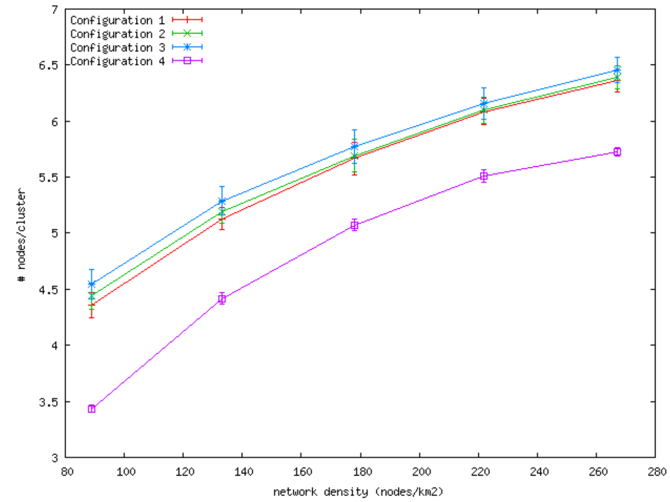

Fig. 6. Average cluster size for different cost configurations

The cost function, responsible for modeling the network shape, can be as simple or as complex as one needs it to be. For the results presented here, we chose to focus on the number of clusters, however, other factors could be taken into account. The important point to consider is that cost function calculation is a flexible way to control the network topology behavior. This flexibility is an interesting asset for networks such as public safety networks where different disaster sites could have different network requirements and the network operation can be shaped as desired. The cluster sizes are homogeneous; the technique enables a load balance among clusters in a dynamic and simple way.

In future, we plan to investigate the impact of the proposed topology management on the performance of the standard routing protocols in a typical PSN application with the group mobility.

References

[1] J. Q. Bao and W. C. Lee, Rapid deployment of wireless ad hoc backbone networks for public safety incident management, in Proc. IEEE Globecom, Washington D.C., USA, November, 2007

[2] Rajmohan Rajaraman, Topology control and routing in ad hoc networks: a survey, ACM SIGACT News, v.33 n.2, June 2002

[3] P. Santi, Topology Control in Wireless Ad Hoc and Sensor Networks, WILEY, July 2005

[4] G. Mainland, L. Kang, Sébastien Lahaie, D. C. Parkes, and M. Welsh, Using virtual markets to program global behavior in sensor networks, ACM SIGOPS European Workshop, Leuven, Belgium, Sept., 2004

[5] Xiaokai Yang, Economics: New Classical Versus Neoclassical Frameworks, Wiley-Blackwell, Jan 2001

[6] Hubert D. Henderson, Supply And Demand, Kessinger Publishing Company, July 2004

[7] P. Hui and J. Crowcroft, How small labels create big improvements, PerCom Workshops'07, New York, USA,Mar., 2007

[8] K. Sun, P. Peng, P. Ning, and C. Wang, Secure Distributed Cluster Formation in Wireless Sensor Networks, In Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC), IEEE, Washington, DC, Dec., 2006

[9] Daniel Câmara, Christian Bonnet, Topology Management for Public Safety Networks, International Workshop on Advanced Topics in Mobile Computing for Emergency Management (MCEM 2009), ACM, Leipzig, Germany, June, 2009

[10] Distributed Computing Group at ETH Zurich, Sinalgo - Simulator for Network Algorithms, Retrieved December 19, 2009, from disco.ethz.ch/projects/sinalgo/

[11] Celtic - Telecommunication Solutions Project, HNPS - Heterogeneous Network for European Public Safety, http://www.hnps.eu/, Jun., 2010