Market Based Strategy

Public Safety Networks are an extreme and challenging environment for topology management protocols. As stated in Section 2.8.1 the main concerns for PSNs are rapid deployment and survivability [12]. These concerns are also important in other networks, but are not normally the main concerns. Moreover, the network requirements for different disaster scenarios may differ greatly. This chapter describes a flexible distributed algorithm to perform network admission control and topology management for public safety wireless networks. The proposed algorithm is anot only able to dynamically adapt to different network requirements, but also to create homogeneous clusters, where the number of mobile routers attached to each cluster is roughly the same. The technique successfully creates and maintains the desired topology relying only on an elegant and customized cost function.

8.1 Introduction

The deployment and the management of nodes for wireless mesh/ad hoc networks are challenging problems and they become even more interesting when we consider them in the context of public safety networks. Not only is this kind of network, by nature, life-critical but they also have strict requirements. Moreover, these requirements may vary significantly for different disaster sites [85]. A stable network structure is crucial for enabling the creation of efficient higher layer algorithms and at the same time enhancing scalability and capacity for large-scale wireless ad hoc networks [91].

PSNs must be reliable and endure even when deployed through rough environments. The algorithms running on this kind of network should take this into account. The network organization is important to guarantee stability and provide communication even during the most severe conditions. Simple structures, such as a planar network, may be easier to deploy and to maintain, but this kind of organization is neither scalable nor appropriate for use in large scale deployments. Structured networks, on the other hand, are more scalable, but the structures must be created and maintained. This work focuses on hierarchical network topologies. Even though the proposed method is general and adapted to any wireless mesh network, we believe that we can benefit if we apply it to highly dynamic and unpredictable networks, as is the case with public safety networks.

8.2 Background

To the best of our knowledge, no other work approaches the topology adaptability problem in the same way we do. In most cases, if the topology requirement changes a completely new algorithm must be designed and deployed.

In [73], 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 both works 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 [112]. 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. 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.

8.3 Objectives

To accomplish the main objective of this part of the thesis, the creation of stable topologies, the algorithm proposed here has three main objectives.

1. Ensure a stable, or at least as stable as possible, network as fast as possible while respecting the desired architecture. As the target application are PSNs, 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 disaster scenario to another.

3. Finally, keep the number of clusters as low as possible, while keeping the clusters of 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.

   The technique described here intends to create and maintain well-defined wireless mesh network architectures in a flexible and dynamic way. We want to be able, by just adjusting a set of parameters, to change the behavior of the whole network without deploying new equipment or protocols. The algorithm must be 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 can be adapted to any wireless mesh network architecture. As a proof of concept, we applied the method to three different hierarchical networks: a simple clustering algorithm, the CHORIST one and a third, more complex organization.

8.4 Market-based topology management

The 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 [57].

If we align our main objectives with the laws of supply and demand we will see that these three laws map perfectly to 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 an 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. Cluster heads must be able to optimally handle the communication among nodes inside their clusters and exchange key information with neighboring nodes fast and efficiently. However, the optimal number of nodes per cluster depends upon many factors, such as number of attendees and agencies involved, kind of disaster and environmental conditions. These issues are covered by the third law, since the final topology is expected to be a Pareto optimal arrangement [112] and hence it should be stable and fair among all the participants. Figure 8.1 presents these relationships schematically.

The basic mechanism of the evaluated protocols is as follows: whenever an IN arrives in the network, it broadcasts a connection request for the nodes nearby. This request is answered by all the MR/RN/CH in the region. The neighboring nodes answer with their status, number of connections and link

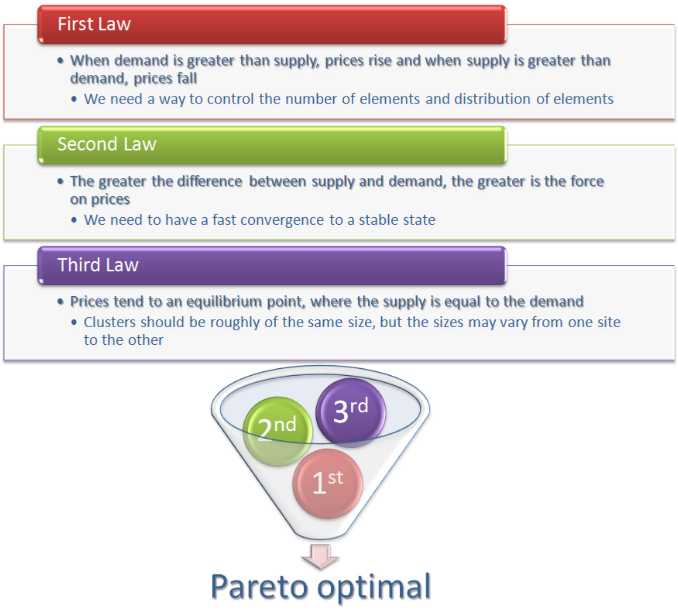

Figure 8.1: Relation of the economic laws of supply and demand and the requirements for PSNs topology management algorithms

status. This information is used to define a connection cost to each one of the possible sponsor nodes. The information in the answer packets and the cost function determine to which node the IN will attach. The cost policy states that, considering all the given data, the lowest cost sponsor should be chosen.

To increase the network stability a node just gives up being a CH or a RN if it moves and loses all its connections, or if it moves and enters in conflict with other well established, lower cost, CH/RN in the region.

A node should always try to attach to the node that presents the lowest attachment cost. To decrease the number of CHs, the chosen basic connection costs should give greater priority to CHs in detriment of the other kind of nodes. Only if there are no CHs around or they are completely overloaded should an IN decide to attach to a MR or a RN and become a new CH. Similarly, to promote a more homogeneous load balance, the cost function guarantees that an IN node will always attach to the least loaded, or the best suited sponsor. Algorithm 4 describes in further details the method when maintaining the CHORIST network architecture.

The cost function can be as simple or as complex as one may need. For

Algorithm 4 - Market Based Strategy CHORIST topology control high level algorithm

```
1: Node Arrives (IN);
2: IN sends a connection request through broadcast;
3: Waits for the responses;
4: if (received any Connection response) then
5:    Weights the costs of the responses;
6:    Sends a connection confirmation to the node with the lower cost
7:    if (connected to a CH) then
8:       Becomes a MR;
9:    else if (connected to a MR or to a RN) then
10:      Becomes a CH;
11:   end if
12: else
13:   if (number of trials less than 3) then
14:      Returns to 2;
15:   else
16:      Becomes a CH;
17:      Sends a connection Update;
18:   end if
19: end if
20: Waits for messages;
21: if (received a Connection Request) then
22:   Responds with a Connection Response informing all its connections;
23: else if (received a Connection Confirmation) then
24:   Registers the connection;
25:   Reevaluates state (may become a RN);
26: else if (received a Connection Response) then
27:   if ( interesting) then
28:      Sends a Connection Confirmation;
29:      Registers Connection;
30:      Reevaluates state (may become a RN);
31:   else
32:      Sends a Connection Cancel;
33:   end if
34: else if (received a Connection Update) then
35:   Registers the Update;
36:   From time to time Evaluate Updates to find not Connected CHs;
37: else if (received a Connection Cancel) then
38:   Removes the connection;
39:   Reevaluates actual state (may become a MR or a IN);
40: end if
41: Returns to 20;
42: From time to time broadcasts a Connection Update ;
```

this work our cost function considers basically the clusters' load. However, other factors could be taken into account as well, e.g. perceived quality of signal, number of blocked nodes and mobility pattern. The used function can be described as:

$$C = \beta_k + \sum_{i=0}^{n} \epsilon_i, \tag{8.1}$$

where C is the connection cost for one specific sponsor candidate and β_k is the basic connection cost for each kind of server. In a free market environment, there is no difference between the services provided by two distinct servers. For this reason the basic connection cost for all servers in the same class k, is the same. A class is a kind of of node, for CHORIST, for example, would be CH, MR or RN, n represents the number of nodes connected to this specific sponsor and ϵ_i represents the individual cost for each one of the already sponsored nodes. For the experiments we set ϵ to be one for each connection the node has, but this value can be gauged according to the topology needs. The last part of the formula provides an adaptive behavior that enables nodes to choose the best servers for their needs, i.e. the less loaded ones; however the formula could be much more complex.

The cost function calculation is a flexible way to control network connections and the topology behavior. By fine-tuning the cost function one can, for example, decrease the number of connections of each CH and increase, or decrease, the size of the clusters. This flexibility is interesting, mainly for PSNs where different disaster sites may have different needs and the network operation can be shaped as desired. By changing and broadcasting a new basic costs vector, one can even change completely the behavior of an already established network without any full software or hardware update.

8.5 Experiments for the MBS topology control

8.5.1 Environment

The evaluations were made using Sinalgo simulator [96] in a $2Km^2$ area. We vary the number of nodes and the communication range of the nodes. All experiments were conducted using Linux Fedora Core release 6 in an Intel Xeon 1.86GHz machine with 16GB of RAM. All graphs are presented with a confidence interval of 99% and each point is the result of the mean of 34 runs with different network configurations. The nodes arrive randomly and are placed uniformly over the observed area. As the experiments observed in

Section 7.4, the centralized implementation works as an oracle: its results are the best possible ones and unachievable with distributed algorithms. However, this offline implementation shows us how far the proposed algorithm is from the theoretical minimal CH optimal solution.

All experiments were conducted for different communication ranges of 50, 100, 150, 200, 250, and 300 meters. However, as the final results for these variations did not present any meaningful difference, we will present only the values obtained for the 200 meters communication range experiments. To evaluate the adaptability capacity of the proposed solution we defined different network configurations and node costs. Considering the implemented cost formula 8.1, if one needs, for example, a network with fewer CHs, it is only a matter of decreasing the basic CH connection cost and increasing the costs for other kind of nodes. In this way nodes will prefer to attach to an existing CH, as it is cheaper than to attach to other nodes to create a new CH. For each different target scenario the cost values should be adapted accordingly to the final desired network shape.

We created six different scenarios with different basic costs for each type of nodes. The basic cost configurations used in the experiments were:

- Configuration 1: favors the creation of clusters, as much as possible. It has high cost to connect to a cluster and low cost for connecting to other nodes. The basic connection cost values (β) are CH=20, MR=5, RN=1.

- Configurations 2 to 5: are variations over the standard configuration, smaller costs for attaching to CHs and larger ones for RNs and MRs. The objective of testing these configurations is to establish whether small variations of costs affect the algorithm behavior. β values are:

  - Configuration 2 CH=0, MR=2, RN=1
  - Configuration 3 CH=0, MR=5, RN=3
  - Configuration 4 CH=0, MR=7, RN=5
  - Configuration 5 CH=0, MR=20, RN=5

- Configuration 6: tries to shape the network as close as possible to the minimum WCIDS, the target configuration of the implemented offline approach. For this case values are: CH=0, MR=50, RN=45.

Configurations 1 and 6 are diametrically opposite in the sense that the first aims to stimulate the creation of CHs while the second aims to keep the number of clusters as small as possible. The differences among the configurations and the desired final network shape are expressed by the histograms in Figure 8.2. These histograms were created from typical runs of the simulation. We can observe that the technique really manages to control the network topology going from the extreme case of a nearly minimum number of CHs to the case where almost all nodes are CHs.

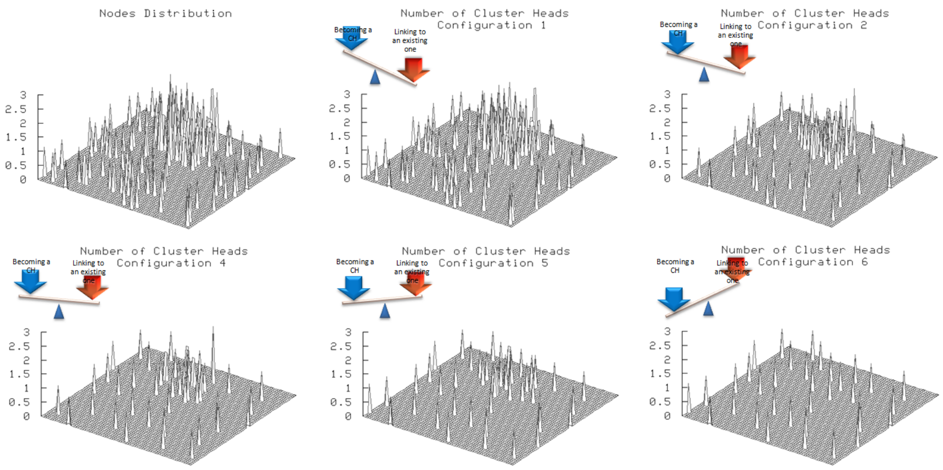

Figure 8.2: Number of cluster heads spread through the network according to the different evaluated configurations, for a 40% concentration network scenario.

To validate the technique we applied it to three different hierarchical network organizations. The three networks are a simple generic cluster, CHORIST and an interest group one. The aim here is to show that the technique is independent of the target architecture and at the same time it can shape the format of the final network topology. We will present the experiments in order of complexity of the topology protocols, from the simplest to the most sophisticated one.

8.5.2 Simple cluster experiments

The generic cluster algorithm is also a two layer one but simpler than the CHORIST architecture. CHs may be connected directly or through MRs, there is no RN role. Figure 8.3 shows an example of the expected behavior of the simple cluster algorithm. The minimum number of CHs for this scenario is also a WCIDS, where the CHs are not in the range of one another and the message exchange occurs through a common MR. Figure 8.4 shows the state machine for the generic cluster algorithm.

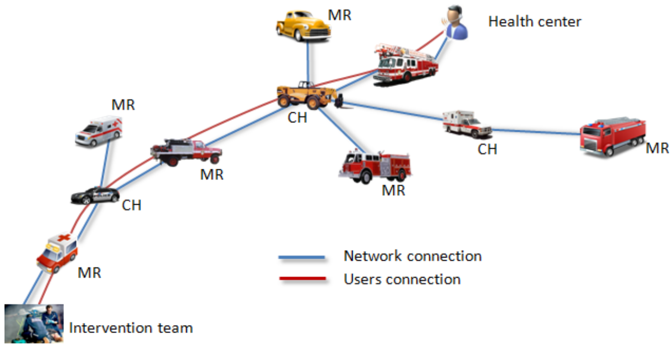

Figure 8.3: Simple cluster architecture, showing an end-to-end users communication

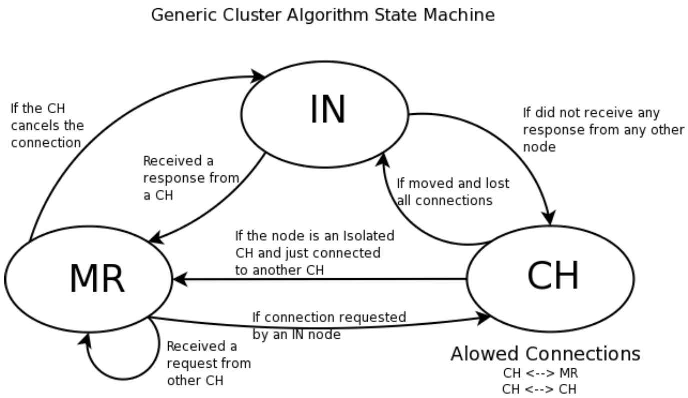

Figure 8.4: State machine for the generic cluster algorithm

The graph in Figure 8.5 shows the number of CHs for different network sizes for the simple clustering algorithm. As we can observe the number of CHs changes in the way it was expected to. The small changes in the cost values also show that using the technique one can even make a fine grain control of the network shape. With regard to the minimum CHs configuration, the values reached by Configuration 6 are quite close to the ones found by the minimum WCIDS algorithm, normally inside the 99% confidence interval range. However, it is worth reminding that the offline implementation, not only has the complete view of the network, but also works using the final configuration, while our approach, MBS, works only with local information, the CHs are assigned dynamically, the algorithm does not need to know the entire topology in advance and nodes join the network at different moments during the simulation time.

Another interesting characteristic we can notice from the graph in Figure 8.5 is the slope of the curves: for Configuration 1, where the CH attachment cost is abusive, the slope is more accentuated, and when the cost to attach to a CH decreases, the slope of the curve is given by the increase in the cost of the attachment to MRs and RNs. The differences between the two graphs are also expected since the evaluated protocols are different and have different elements. So the proportional connection costs are different.

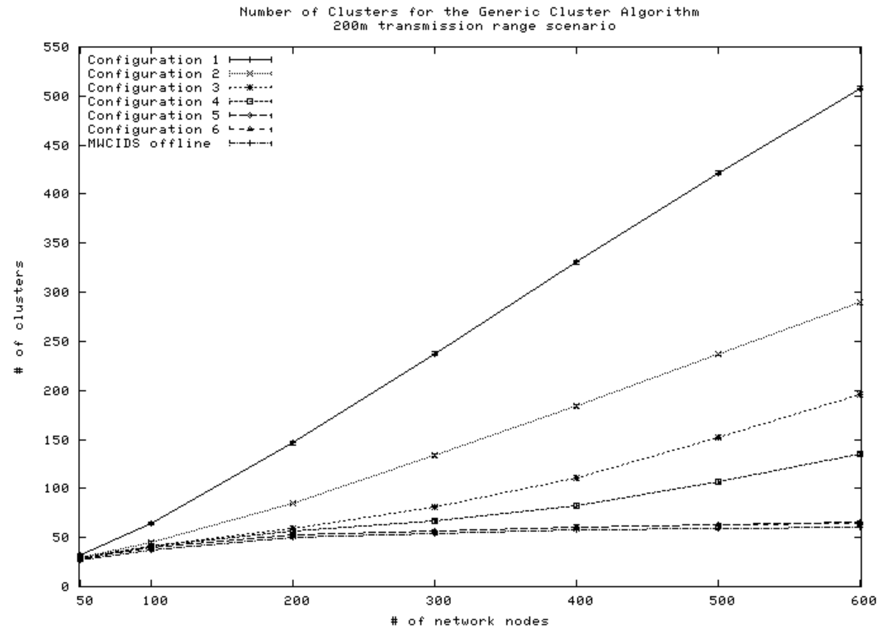

Figure 8.5: Number of Cluster Head nodes for the generic clustering topology

Figure 8.6 presents the average size ratio of clusters when the network size increases. We define cluster size ratio as: $CSR = (nMR+nRN)/nCH$, where CSR is the cluster size ratio, nMR, nRN and nCH are, respectively, the number of mobile routers, relay nodes and cluster heads of the whole network scenario. The average shown is the average over all the evaluated scenarios. From these graphs we can perceive that by fine tuning the costs we can model the clusters' behavior. The offline approach has the biggest cluster size ratio since its main goal was to reach the minimum number of clusters, so the clusters are larger. The standard deviation for the cluster sizes, for all the evaluated configurations, is typically below 0.05, this means the clusters are indeed well balanced, as we first intended. Moreover, we can control the clusters size by changing the cost function. We can perceive from the graphs that the different configurations reach a stable point in the ratio of MR + RN and CHs. Except for the configuration where we intend to increase the number of clusters as much as possible, the average cluster size reaches a saturation point and the size of the clusters stays stable independently of the number of nodes in the network.

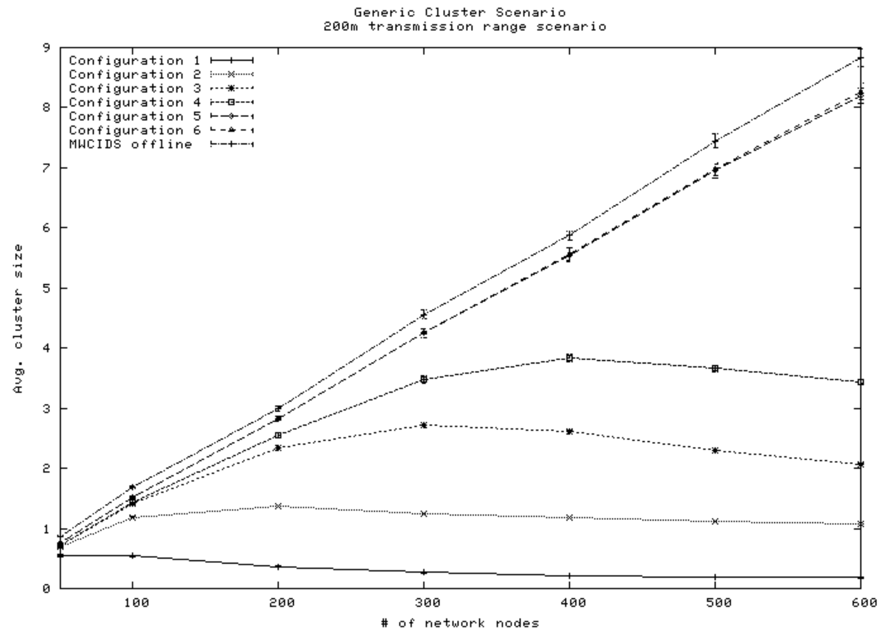

Figure 8.6: Number of nodes per cluster for the generic clustering topology

8.5.3 Relaxed CHORIST experiments

For the CHORIST experiments we have kept all the requirements and constraints described in Chapter 7, we just relaxed two constraints. First for these experiments two CHs may be in the same area, although when searching for the minimum number of clusters the WCIDS is still the target architecture. The second constraint we relaxed was that for these experiments the MR may broadcast connection updates. These two changes are required to allow the network to vary the clusters concentration. However, the state machine observed in Figure 7.2 is still valid and all the transitions are rigorously respected.

The same observations made for Figures 8.5 and 8.6 are valid for Figures 8.7 and 8.8. As expected, even though the protocol and the nodes organization are different, the technique managed to shape both architectures in the same way.

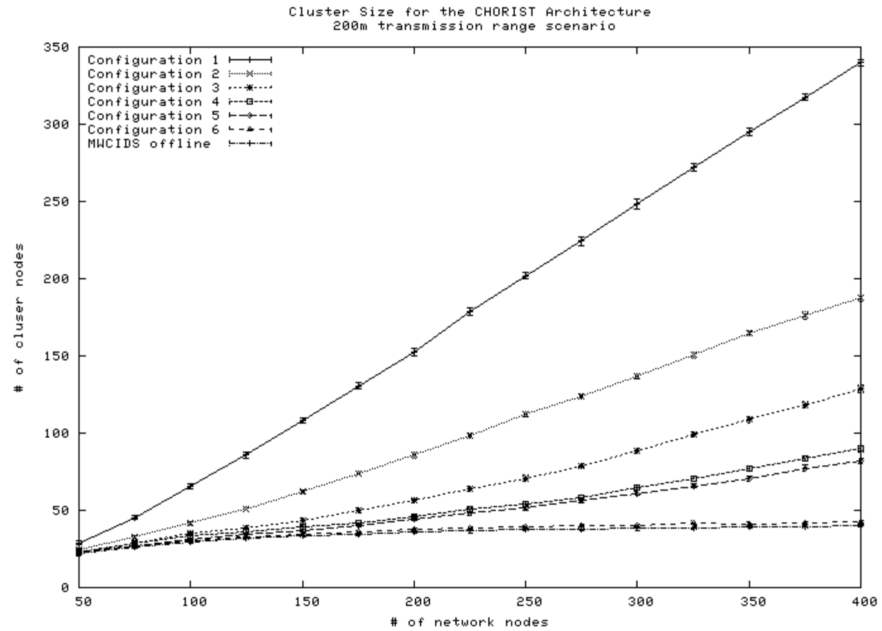

Figure 8.7: Number of Cluster Head nodes for the CHORIST topology

To simulate different disaster scenarios we varied the concentration of the network. We randomly chose a point in the defined area and evaluate different nodes densities within a 300m radius from this point. The observed concentrations were 10%, 20%, 30% 40% 50% 60% 80%. Figure 8.9 presents the Configuration 2 cluster sizes and the cluster distribution, for the different evaluated distributions. We can observe that for Configuration 2, as it was intended, the cost function increases the number of CHs in the more crowded areas while simultaneously keeping the size of the clusters under control.

The graph of Figure 8.10 shows the number of messages sent through the entire network during the simulation time for each one of the defined configurations. As expected the bigger the size of the network the larger the number of messages exchanged among nodes. However, the volume of messages shows only small variations between configurations. Even though the network shape changes considerably, the message cost to generate and

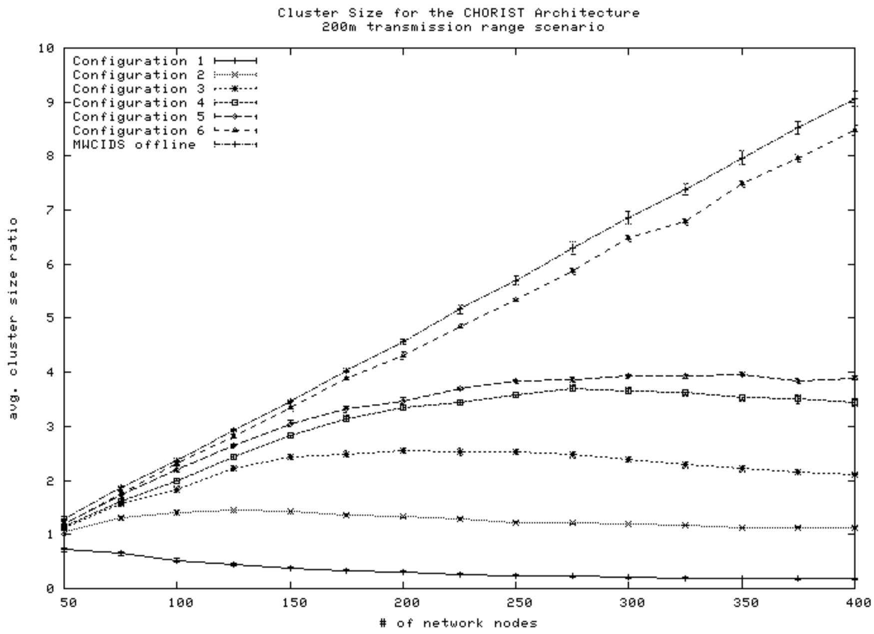

Figure 8.8: Number of Cluster Head nodes for the CHORIST topology

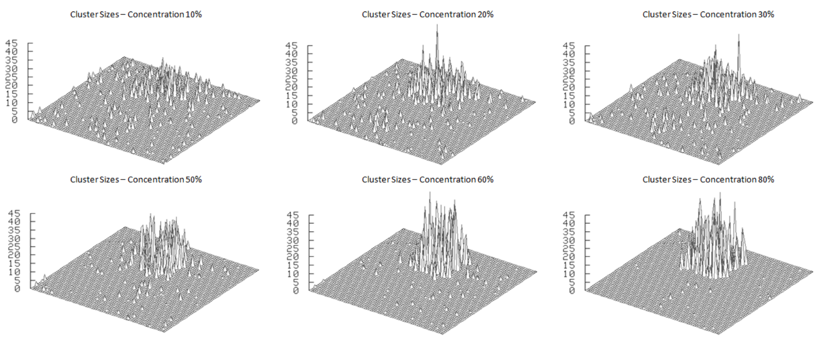

Figure 8.9: Cluster sizes for configuration 2 varying the nodes concentration.

maintain a network, with the minimum and maximum number of CHs, is basically the same. This behavior is the same for all the experiments for the three evaluated network types.

The graph in Figure 8.11 shows the number of clusters a relay usually

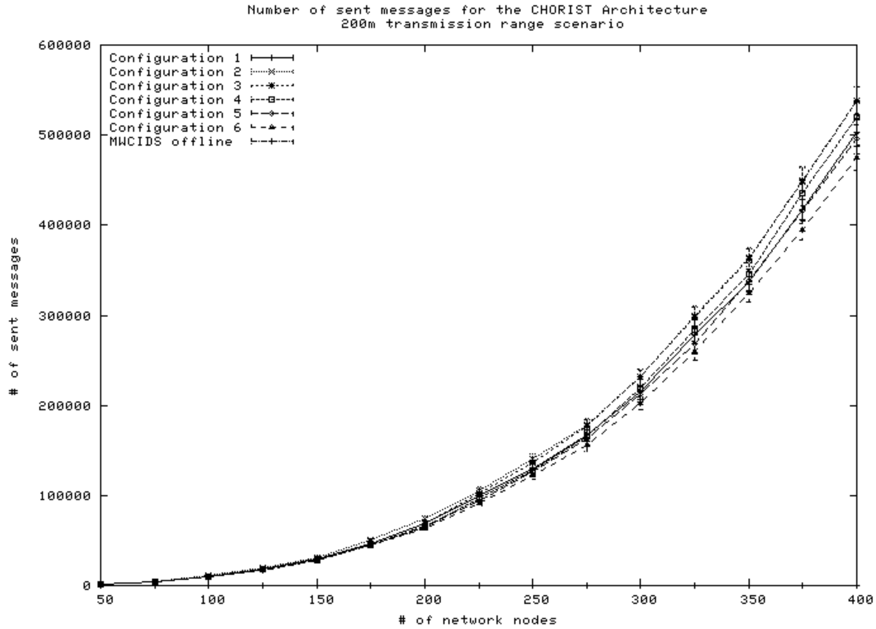

Figure 8.10: Number of sent messages through the network nodes for the CHORIST topology

connects. Comparing the graphs of Figure 8.5 and Figure 8.11 we see that the number of RN connections is directly related to the number of CHs in the network. The bigger the number of clusters the higher the load for the available RNs. When we decrease the number of clusters we also decrease the need for RNs. For the WCIDS offline implementation this value is around two, i.e. on average a RN connects only two clusters.

For all CHORIST evaluated cases, our technique increases the number of relay nodes more than the minimum value, given by the offline implementation. The first reason for this is that, the technique does not have a global view to be able to select the best global RNs. Second, as we create more clusters it is only natural to have more RNs to interconnect them. However, the most important factor is that CH nodes chose their RNs in a selfish manner. They chose the best suited nodes, from their point of view, not that of the network. Consequently, it is possible to have, for example, two different nodes acting as RN between the same two CHs, just because each CH chose their RN in a selfish manner. In this case instead of having one RN acting as a gateway between these two CHs, as it is the case in the offline approach, the network will have two RNs. Each one of them acting as a RN

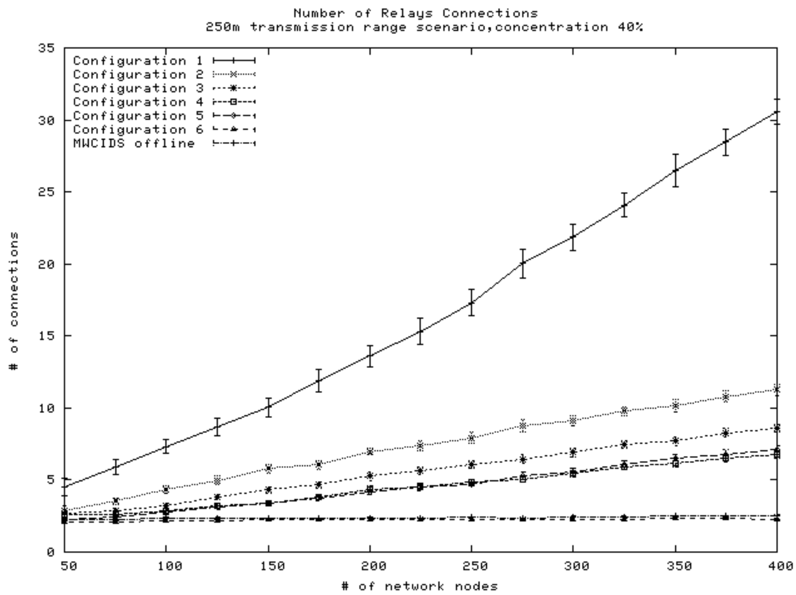

Figure 8.11: Average number of clusters connected by a RN

for one of the CHs involved. However, the increase in the number of RNs has some advantages. For example, the cost function could take into account the channel reliability and, in this case, having two RNs the network stability would increase. Another point to observe is that, as example of what already observed in our standard implementation of CHORIST presented in Chapter 7, the path sizes are smaller when we increase the number of RNs. As expected, the same occurs for this implementation. Increasing the number of RNs, also increases the paths' diversity, enabling the occurrence of smaller routes between nodes.

The graph of Figure 8.12 shows the average size of clusters for the configuration 2 network for different network sizes and concentrations. The network concentration effectively affects the size of the clusters. However, the standard deviation for the cluster sizes, in all the evaluated configurations, is typically around 0.15. This means that, even though the concentration changes, the sizes of the clusters are well balanced. Within the same scenario, the number of nodes per cluster does not present any significant variation, as we first intended.

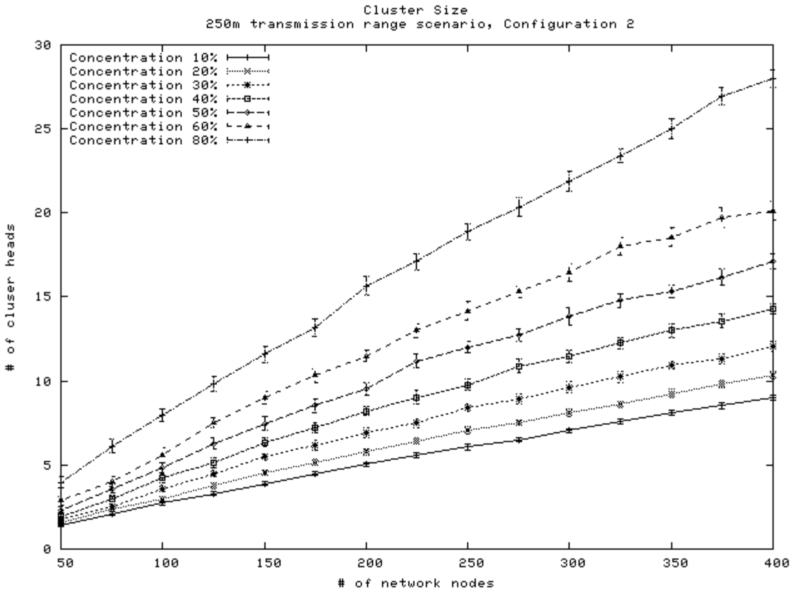

Figure 8.12: Average cluster size for different concentrations of configuration 2

8.5.4 Interest groups experiments

This observed topology management algorithm is also hierarchical, with the formation of clusters maintained by one cluster head. These experiments present mainly two distinctions when compared to the previous ones. The first difference is that here we have a set of special nodes that are declared cluster heads by default, i.e. Default Cluster Heads (DCH). These nodes maintain this status throughout the network's life. Other nodes become cluster heads (CH) only in areas not covered by these DCHs. The second difference is that the method also considers a variable number of interest groups (1 to N groups). Each interest group is defined in the network startup and must have at least one DCH to represent it. The DCH does not necessarily have to be close to all nodes in its group. This kind of behavior may be interesting for PSNs when one want to maintain the different authorities' traffic separate. For example, normally in a disaster scenario the police missions and interest differ from the ones of medical staff. So it makes sense to have different interest groups for these two distinct teams.

Interest groups may also have an important role in decreasing the amount of traffic, as observed by Hui and Crowcroft [58]. Sometimes in PSNs some messages may need to be spread to all nodes in a specific group but may be meaningless for nodes in other groups. For example, in case of an earthquake 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. For simulation purposes, 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 and even CH and DCH only receive Connection Update messages from nodes in different groups. We allow this to increase connectivity and make the CH nodes aware of the number of clusters around.

Each element in the proposed solution has a connection price. The prices vary among the different nodes, and the price to pay for a connection to a CH is 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 loaded the CH/DCH 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 goes like this: DCH providing connection $<$ isolated DCH $<$ CH providing connection $<$ isolated CH. The costs of the DCH/CH providing connection increases with the number of connections it is handling. For example, if an isolated node has two options, an CH with providing connection to 5 other nodes, and another CH providing connection to 6 other nodes, it will prefer to connect to the first one.

Interests group protocol description

Periodically CH and DCH nodes send connection update messages announcing their presence and list of connected nodes. Each connected node, MR, sends also a periodic a connection update, but only to the node it is attached to. CH and DCH updates are sent through two available interfaces. When arriving by the default interface it may change the status of the nodes that received it. If it arrives by the second interface, it is just stored as a way to build the knowledge of the clusters around. The two interfaces have different purposes, the first one could be a WiFi like interface, to organize the communication with nodes closer to the CH and the second interface could be a WiMAX kind of interface, to reach further nodes and with a broader bandwidth capacity. This interface would normally be used to transfer data between the clusters. Figure 8.13 presents an example of the interests groups setup.

If the node that received the update message, by the default interface, is a CH or a DCH, this node verifies the cost of the income message. If the node is not already connected to a DCH, the cost of this new provider is smaller than the cost of the present provider, or the node own connection cost, and the perspective provider has room to accommodate all the present connections, the node sends a connection request to the node that sent the update message. If the node is an IN (isolated node) or an isolated CH/DCH and the node that sent the message has enough space, this node sends a connection request to the node. Anything is better than stay isolated.

When a CH/DCH receives a connection request, and it has enough resources, it sends a connection response to the node that requested the connection and reserves the resources to this node. If it does not have enough resources the CH/DCH sends a connection cancel.

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. However, if the node didn't send any connection request in first place, i.e. the message was a mistake, the node sends a connection cancel to the node that just sent the connection response. Case the MR/CH/DCH receive 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 a IN for a long time, three attempts with different bakeoffs intervals, it may became a CH.

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

For all practical purposes there is no difference from CH and DCH. The differences are in terms of connection costs, lower for DCH. Other difference is that being a CH is a transient state, a CH node may become a MR at any moment if it finds another node that has a lower cost than it has. However, a DCH is always a DCH, no matter what. A DCH continuously broadcast update messages being able even to receive connections, if some other node needs it, and is able to pay for the price. Figure 8.14 presents the state machine for the DCH node.

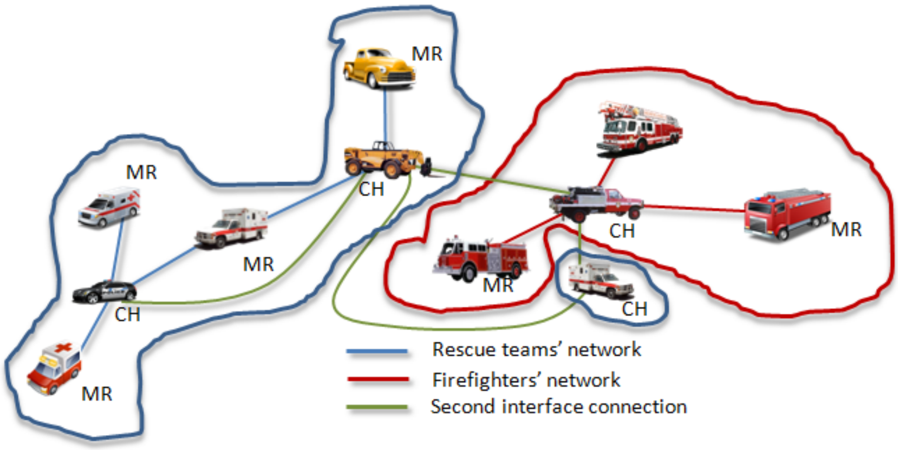

Figure 8.13: Interest groups architecture, showing two different interest groups and the second interface links

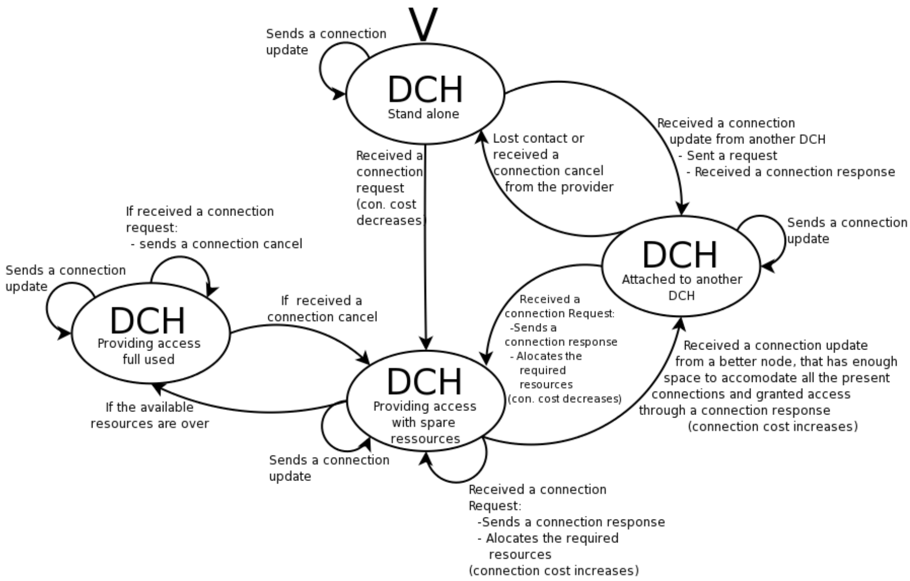

Figure 8.14: Default Cluster Head state machine

Experiments

For these experiments nodes move in a random way point fashion in a $1000 \times 1000m^2$ area for one hour simulation time. The communication range for

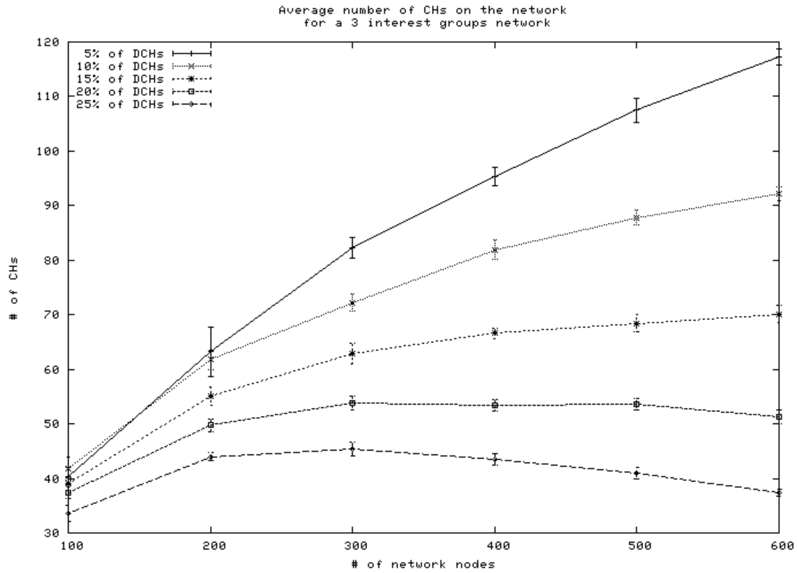

Figure 8.15: Variation of the average number of CHs in the network when we increase the percentage of DCHs

the first interface is $100m$ while for the second interface the communication range is $300m$. The used basic connection weights are: $DCH = 0$, $CH = 8$, $MR = 50$. We considered here that the maximum allowed size for a cluster is 7. We varied the number of interest groups and percentage of DCHs. The DCHs are placed randomly thought the network and the remaining IN nodes are placed in a maximum distance of $130m$ from these DCHS.

The graph in Figure 8.15 shows the average number of CHs on the network when we vary the percentage of DCHs. For this clustering algorithm 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. The number of CHs are more or less stable for networks with more than 300 nodes because the limit of the size of the clusters were not a problem for these experiments, and as the nodes does not have a common movement pattern, the occurrence of CHs have a closer relation to the size of the area than with the occurrence of overpopulated clusters. As the area does not change, the average number of CHs needed to cover the area also does not change significantly.

Figure 8.16 presents the average number of CHs when we vary the percentage of DCH and the number of interest groups. The graph shows the

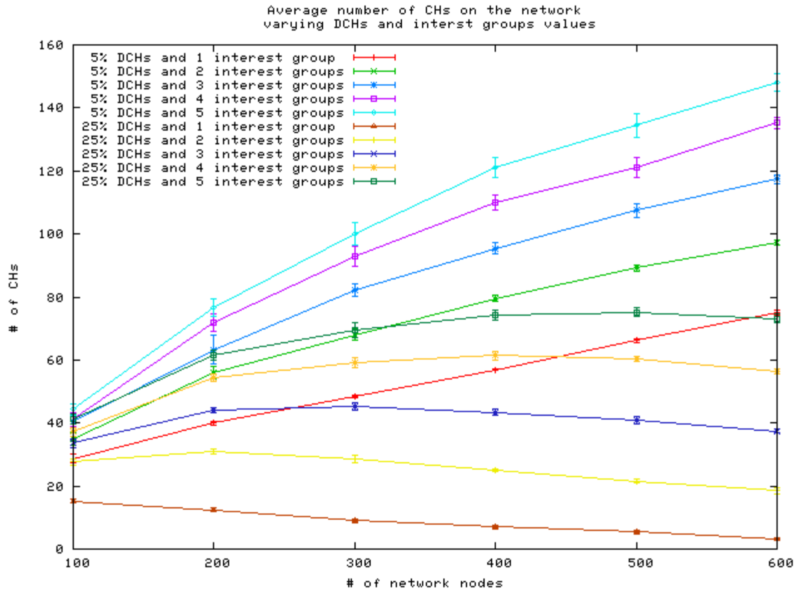

Figure 8.16: Variation of the average number of CHs in the network when we increase the number of interest groups

curves for 5% and 25% of DCHs, the minimum and maximum number of DCHs we evaluated. We can observe that the behavior is consistent for both percentages and that when we increase the number of interest groups we increase also the number of CHs in the network. 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 with the same interest. In this way more nodes start to become CHs. We can also observe that 25% of DCHs on the network is enough to stabilize the number of required CHs over the simulated area. On the other hand, Figure 8.17 shows that the average size of clusters decrease with the increasing in the number of DCHs and interest groups. The average cluster size, for these experiments, on average did not reach the maximum defined cluster size, which is of 7 nodes. However, when the maximum value was reached during the experiments the designed cost function could control the nodes behavior and form new clusters.

One interesting thing of the graph in Figure 8.17 is that we can observe that networks with 5% DCHs and 3, 4 and 5 interest groups have a similar behavior, in terms of numbers of nodes per cluster, than a 25% DCHs network with 2, 3 and 4 interest groups respectively. This is interesting because shows that the impact of the number of DCHs has also a relation with the number

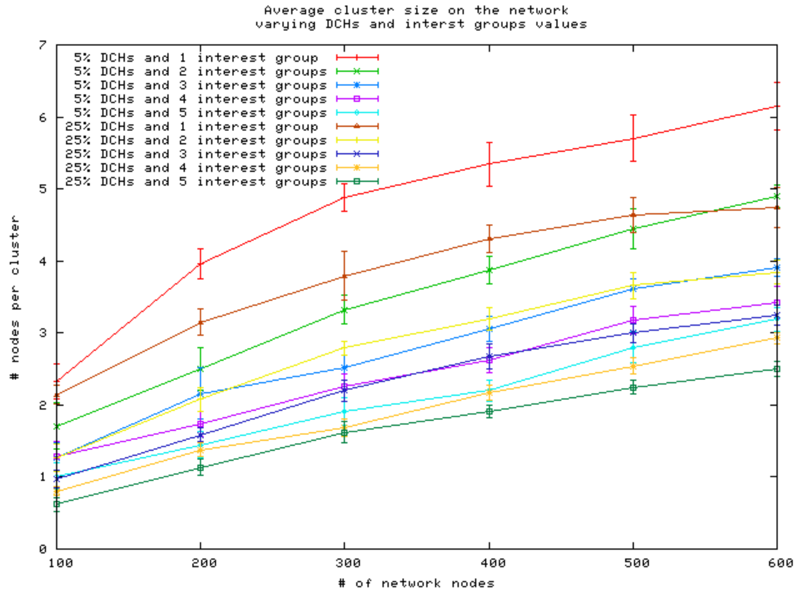

Figure 8.17: Average cluster size CHs in the network when we increase the number of interest groups

of interest groups.

For these experiments we also tracked the number of times a node changed status during the simulation period i.e. CH to MR or MR to CH. Figure 8.18 show that when we increase the number of interest groups we decrease the number of changes. This relation, counter intuitive at first, comes from the fact that when we have more interest groups nodes spend more time to find another cluster with the same interest, so they tend to became CHs and stay as CHs for more time than the nodes in environments with less interest groups. When we have less interest groups the tendency is for nodes that become a CH to find faster another cluster, thus changing states more frequently. Even though the number of changes for the 25% DCH network is smaller, we can observe that the behavior does not change significantly when we change from 5% to 25% the number of DCHs on the network. Not only the shape of the curves is similar, but also the values themselves are close. This means that the number of changes has a small dependency to the percentage of DCHs on the network. It is more related to the mobility, number of interests groups and area coverage than with the number of DCHs in the network.

The graph in Figure 8.19 shows the average number of standalone clusters during the whole simulation time: we consider as standalone clusters

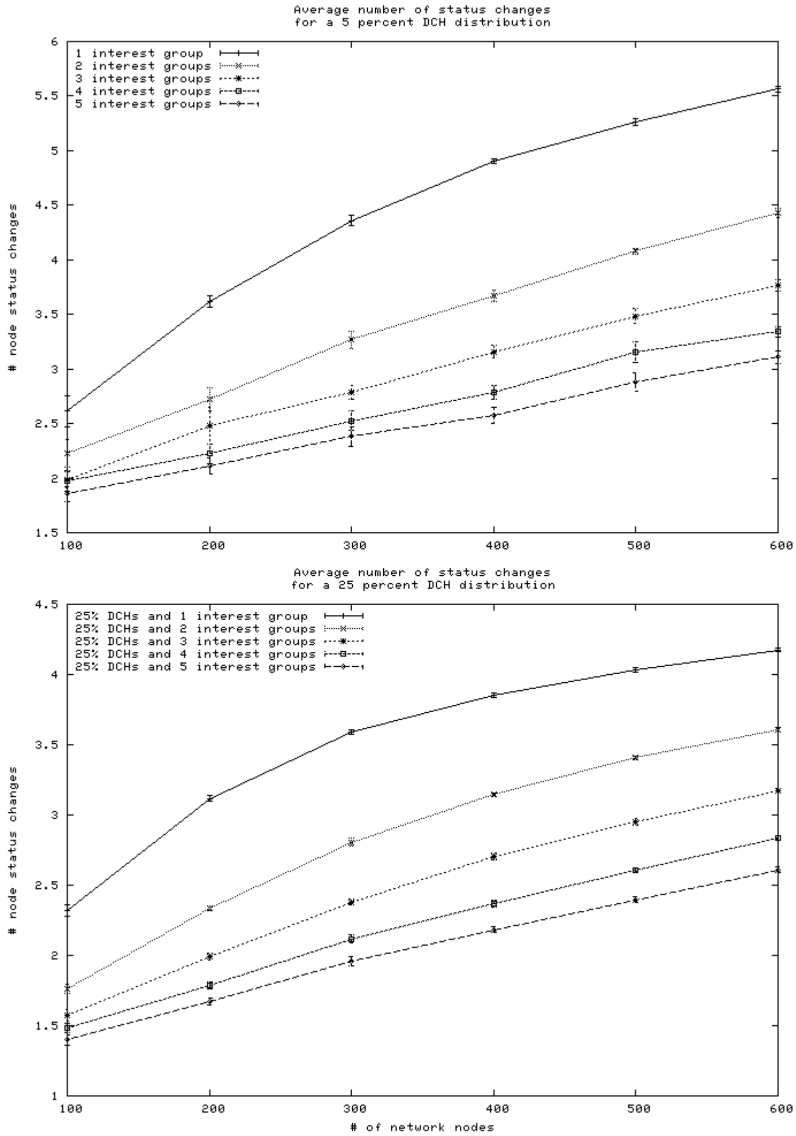

Figure 8.18: Average number of status change for 5% and 25% DCHs networks

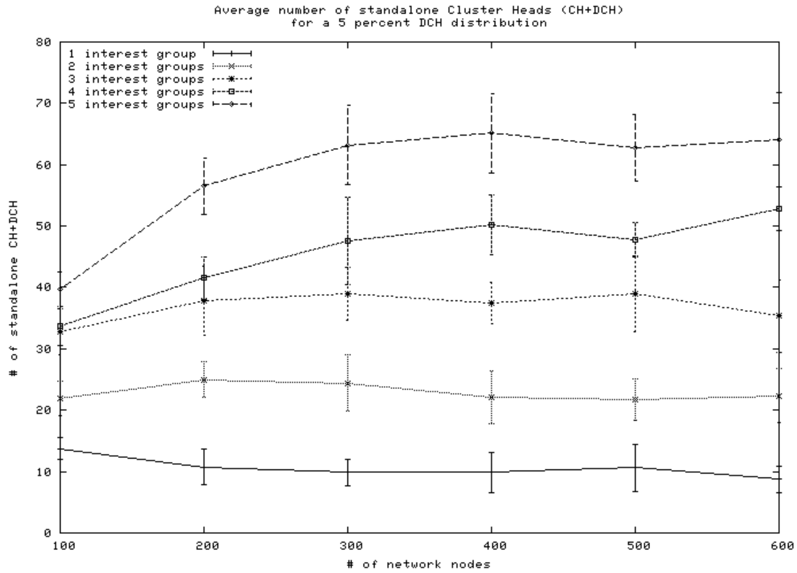

Figure 8.19: Average number of standalone clusters when we increase the number of interest groups

those that just have a single member, either a CH or a DCH. The bigger this value the worse it is for the network. This means that the network is more fragmented and more control messages will be required to maintain the structure, as can be observed in Figure 8.20, i.e. the cost of the network is higher and less bandwidth will be available for data traffic. Another thing we can observe is that the average number of standalone clusters is stable and independent of the number of nodes in the network. Again, the increase in the number of interest groups contributes to the occurrence of standalone clusters.

The higher the number of interest groups the bigger the number of control messages exchanged during the simulation period. Figure 8.20 shows that when we increase the number of interest groups the number of exchanged messages also increases. Even though the difference is relatively small, the graph only considers the messages exchanged to maintain the network. More clusters also means more connectivity changes that can affect other protocols, for example routing or peer discovery processes: In other words, the increase in the number of clusters and the over-segmentation of the network may lead to a "domino effect", where the other layers protocols will also need to exchange more control messages making the difference between the curves larger.

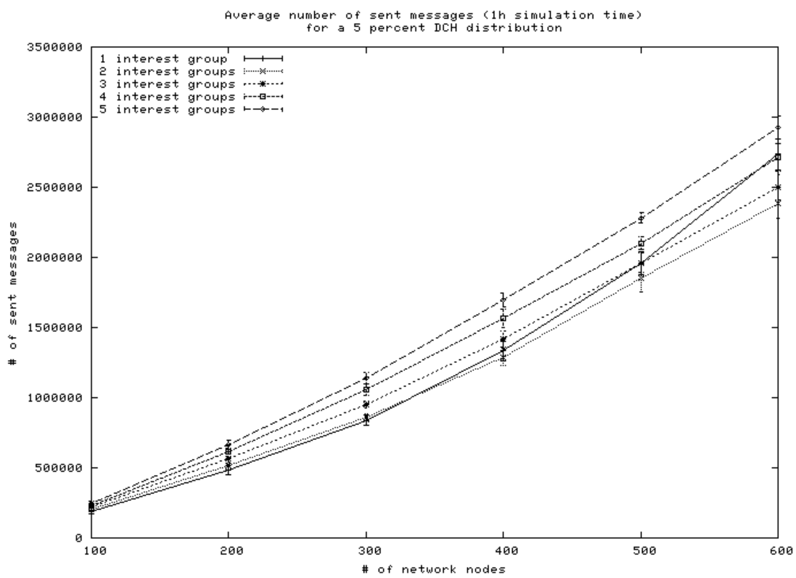

Figure 8.20: Average number of sent control messages when we increase the number of interest groups

8.6 Conclusions

This chapter presented a technique to perform network admission control and topology management for structured mesh networks. The results show that by handling only local information and without the complete final configuration, the proposed method guarantees the correct clustering formation and role attribution to the nodes. The technique is also able to shape fairly distinct final network configurations. For example, just controlling a vector of cost functions one can go, in a distributed way, from a completely clustered network to the one that has the minimum possible number of clusters.

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.