Topology Management

As stated in the previous chapter, one of the premises to enable the full utilization of the WMN potential is self-organization. The network must be able to have mechanisms to control its behavior and adapt to new situations. This chapter introduces the network topology control that is a key aspect on wireless mesh and ad hoc networks. Control and maintain a defined and stable topology simplifies the tasks of the other layers algorithms. These algorithms are expected to work autonomously and adapt themselves to different situations maintaining the defined network structure.

6.1 Introduction

Topology management, or topology control, algorithms are used to reduce the initial topology of the network to save energy, increase the lifetime and improve stability of the network. The main goal is to maintain a desired topology, normally aiming to reduce the number of active nodes, and/or links, save resources and organize the network.

Topology control algorithms select the communication range of a node, and construct and maintain a network topology based on different aspects such as node mobility, routing algorithm and energy conservation [11]. For Santi [92] topology control is also about dynamically changing nodes' transmit power to achieve a specific goal, from the network perspective, while decreasing the power consumption. However Santi highlights that power control alone is not enough to define a topology control algorithm. Power control algorithms normally focus on the best power choice for a single channel, or transmission, while topology control mechanisms have a more systemic view as they aim to optimize the *whole* network.

Wightman and Labrador [109] consider topology control to be composed of two subproblems: topology construction and topology maintenance. Topology construction is the phase where the initial deployed nodes are first organized. In the beginning there is no control over the position of the nodes and their interconnections. Some areas may be over populated or have too many between nodes, while other areas may be poorly covered and connected. The objective of the topology construction phase is to minimize these discrepancies while organizing the network within the specified constraints. Topology maintenance is the process of maintaining the reduced topology with the desired characteristics. This process is required since after some time the established topology may change, for example, nodes may move or run out of battery.

6.2 Topology Formation

6.2.1 Neighbor Discovery

The performance of both ad hoc and mesh networks depend on the interaction among communicating entities in a given neighborhood. Thus, in general, before a node starts communicating, it must discover the set of nodes that are within its direct communication range. Once this information is gathered, the node keeps it in an internal data structure so it can be used in different networking activities such as routing. The behavior of an ad hoc node depends on the behavior of its neighboring nodes since it must sense the medium before it starts transmitting packets to nodes in its interfering range, which can cause collisions at the other nodes.

Node discovery can be achieved with periodic transmission of beacon packets (active discovery) or with promiscuous snooping on the channel to detect the communication activity (passive discovery). In probe-based distributed protocol for knowledge range adjustment (PRADA) [75], a given source node periodically sends a discovery packet to its neighboring nodes, to which the latter reply with a location update packet (that might include, for instance, the node's geographical location). PRADA adjusts dynamically its communication range, called topology knowledge range, so it leads to a faster convergence of its neighboring nodes.

6.2.2 Packet Forwarding Algorithms

An important part of any multi-hop network is the packet forwarding algorithm that chooses which neighboring node is going to be used to forward the data packet. It does so following a forwarding goal, having the shortest average hop distance from source to destination for instance. In this case, the set of potential nodes may include only those within direct communication range of the current node or also all nodes along the route to the destination. The forwarding goal may also include some QoS parameters such as the amount of energy available at each node.

The following forwarding algorithms consider only nodes that are in direct communication range of the node that has a data packet to be forwarded, as depicted in Figure 6.1. The Most Forward within Radius (MFR) forwarding algorithm [86] chooses the node that maximizes the distance from node S to point p. In this case, as depicted in Figure 6.1, it is node 1. On the other hand, the Nearest Forward Progress (NFP) forwarding algorithm [102] chooses the node that minimizes the distance from node S to point q. In this case it is node 2. The Greedy Routing Scheme (GRS) [55] uses the nodes' geographical location to choose the one that is closest to the destination node D. In this case it is node 3. The compass selected routing (COMPASS) algorithm [44] chooses the node that minimizes the angle α but considering the 4 nodes that are closest to node D. In this case it is node 4. The random process forwarding algorithm [81], as the name suggests, chooses a random node that is in direct communication range from S.

The Partial Topology Knowledge Forwarding (PTKF) algorithm [75] chooses a node using a localized shortest path weighted routing where routes are calculated based on the local topological view, taking into consideration the transmission power needed to transmit on that link.

6.3 Classification

Broadly speaking, topology control algorithms for ad hoc networks can be classified as having a hierarchical or clustering organization, or a power-based control organization [11] [110]. Furthermore, these algorithms can be either centralized, distributed, or localized.

6.3.1 Clustering Algorithms

The clustering process consists in defining a cluster-head node and the associated communication backbone, typically using a heuristic. The goal is to

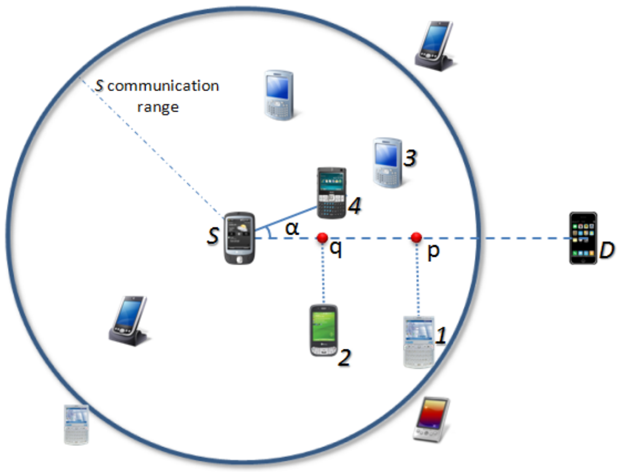

Figure 6.1: Strategies used by some forwarding algorithms

avoid redundant topology information so that the network can work more efficiently. Clustering algorithms are often modeled as graph problems such as the Minimum Connected Dominating Set (MCDS) [49]. This problem asks for the minimum subset of nodes V' in the original graph $G = (V, E)$ such that V' forms a dominating set of G and the resulting sub-graph of the MCDS has the same number of connected components as G. In other words, if G is a connected graph, so is the resulting sub-graph. MCDS is a NP-complete problem [48], and thus, we must look for approximate solutions [11]. In the case of the clustering algorithm, nodes in the dominating set represent the cluster-heads and the other nodes are their neighbors. An inherent characteristic of an ad hoc network, which makes this problem much more difficult, is that its topology is dynamic. The cluster heads can be selected using either deterministic or non-deterministic approaches:

- A deterministic solution is similar to a distributed synchronous algorithm in the sense that it runs in rounds. In this case there is just one round, and after finishing it the cluster-heads are chosen. Suppose we have a node and its neighboring nodes, i.e., its one-hop neighborhood. The lowest ID solution selects the node with the lowest identifier among these neighbors to create the Minimal Dominating Set (MDS) [48], whereas the max degree solution selects the node with the highest degree [65] [97]. The MOBIC solution examines the variations of RSSI (Received Signal Strength Indicator) signal among them to select the cluster-head [13].

- A non-deterministic solution runs multiple incremental steps to avoid variations in the selection process and to minimize conflicts among cluster-heads in their one-hop neighborhood. Examples of this approach are CEDAR [97], SPAN [29], and solutions based on a spanning tree algorithm [49].

6.3.2 Power-Based Control Algorithms

A mobile node in a MANET needs an energy source (typically a battery) to be able to execute all its tasks. Batteries need to be recharged to provide a continuous energy supply for a node. To extend the lifetime of nodes in an ad hoc network, we need algorithms to determine and adaptively adjust the transmission power of each node so as to meet a given minimization goal and, at the same time, maintain a given connectivity constraint. Some possible minimization goals are control the maximum or average power and define a maximum or average connectivity degree. Some connectivity constraints are a simplex communication or

a full-duplex communication (biconnected). Ramanathan and Hain [86] propose a topology control algorithm that dynamically adjusts its transmission power such that the maximum power used is minimized while keeping the network biconnected.

6.4 Further readings

The literature on topology control is vast, eventhough it is mainly devoted to ad hoc and sensor networks. In fact, one can find a wide range of good proposals, to solve specific problems, introductory surveys and books dedicated to this subject that can give a broader and deeper view of the subject. Good surveys are the works of Rajaraman [85] and Santi [92]. Two good books devoted to topology control are the works of Santi [91] and Labrador [70].

Chapter 7

Dynamic Topology Implementation for CHORIST

This chapter presents the implementation and evaluation of a distributed topology management algorithm for implementing the CHORIST architecture. CHORIST is a European Commission project that addresses environmental risk management focusing on natural hazards and industrial accidents [33]. The CHORIST consortium defined the desired topology but this work is the first one to present an implementation of it. The proposed algorithm is able to dynamically adapt to the nodes' mobility while maintaining the desired topology.

7.1 Introduction

The deployment and management of nodes in wireless PSNs is a fundamental and challenging problem. A well-defined and well-maintained network structure is an indispensable step to enable the creation of efficient higher layer algorithms [12]. For this reason topology control becomes a basic functionality to enhance scalability and capacity for large-scale networks [91]. Unlike in other networks the main concerns in public safety networks are rapid deployment and survivability [12].

The main contribution of this chapter is the proposal of a stable and efficient solution for implementing and managing the structure designed by the CHORIST project [33], taking into account the constraints imposed by the communication model. The backbone topology, depicted in Figure 7.1, is composed of Cluster Heads (CHs), Mesh Routers (MRs) and Relay Nodes (RNs). All the nodes' roles must be defined dynamically, using only local information and following the channel model defined by the consortium [84].

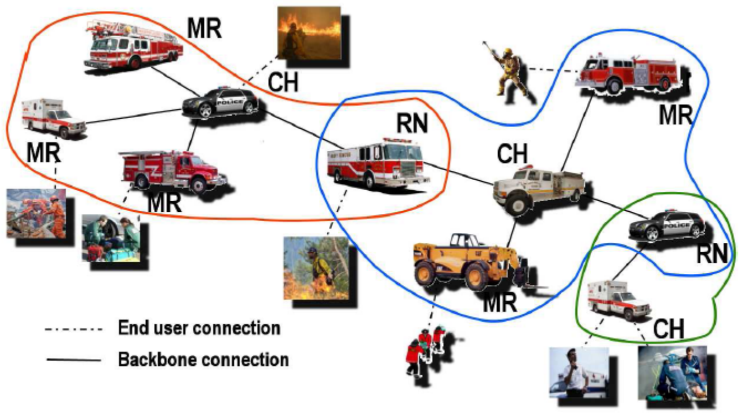

Figure 7.1: CHORIST network description and components.

The CHORIST structure was designed to be efficient and to decrease interference among nodes. Unlike other solutions, hierarchical structures are normally scalable and decrease the overall need for control messages among the nodes [14]. However, creating and maintaining such a structure has a cost. This cost can be measured in terms of bandwidth and delay. We consider that understanding the mechanics of such costs, and the tradeoffs involved, is an important step to enabling the creation of efficient and useful networks. Our proposal builds and efficiently coordinates the proposed CHORIST two-level hierarchical topology.

7.2 Chorist architecture

The core of the CHORIST network is a two-level hierarchical structure. A firefighter, for example, could use any kind of node as an access point, however, inside the proposed structure each node has its specific role. Cluster Heads (CHs) are the nodes responsible for managing the radio resources for their clusters. Relay Nodes (RNs) are the nodes that are part of two, or more, clusters and act as a bridge between them. Mesh Routers (MRs) are the nodes attached to CHs; they obey the CHs scheduling in order to communicate with other nodes. Nodes not yet attached to the network, and those that for some reason have lost their roles, are called Isolated Nodes (IN). If required, an IN may become a CH or a MR. The organization of these elements follows a well-defined and strict composition. Two CHs cannot be directly connected, neither can two RNs. For example, if a CH needs to exchange control data with another CH, the messages must be forwarded through a RN. This is done to avoid two CHs being physically located close to each other and have a more uniform cluster distribution.

The CHORIST backbone follows the channel model defined by the OpenAirInterface [84]. The main architecture of CHORIST, is derived from the OpenAirInterface adopted channel model and frequency reuse pattern. From the topology management point of view the two main constraints of the channel model are: no CH should be in the range of another CH and broadcast channels are reserved for CHs: no other node should broadcast messages. Two neighboring MRs may communicate directly, if previously agreed, but the communication must be direct, not through a broadcast channel. A MR, when inside a CH area, should be attached to it. Figure 7.2 shows all the allowed state changes for the CHORIST node status.

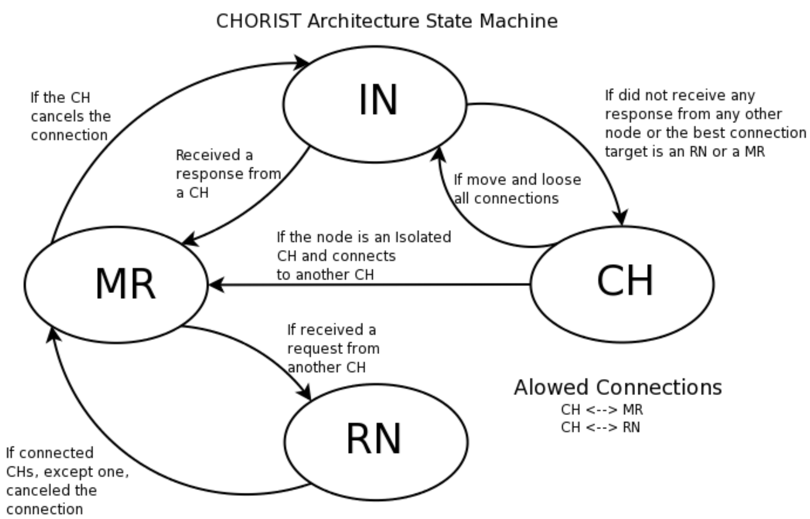

Figure 7.2: CHORIST state machine

7.3 Our implementation of CHORIST

Sometimes it is useful to abstract network problems as graph problems. If we consider the network as a graph, taking nodes as vertices and connections as edges we can reduce the CHORIST architecture to a two steps Weakly Connected Independent Dominating Set (WCIDS) [54].

For a given graph $G = (V, E)$ and a subset S of the set of vertices $V(G)$, S is called a dominating set if any vertex $v \in G$, v is either inside S or v adjacent to a vertex in S. In our case S can represent both the CH and the MR sets. A set S is called connected if S is a dominating set and the subgraph induced by S is connected. In graph theory a set of vertices is called independent if no two elements in it are adjacent, i.e. there is no edge that connects any pair of vertices of the set. In the CHORIST architecture we have exactly the same configuration: the CH set must be a dominating set, since all MRs and RNs should be connected to a CH. Moreover, two CHs should not be in the range of each other. It is important to notice also that the RN set also needs to be a dominating set, regarding the formed CH set. I.e. if we consider the CH set as S, then $V(G)$ would be the whole network. If we consider the MR set as S, then $V(G)$ would be the selected CH set. This makes the problem even more interesting. The minimum independent set is the one with the lowest possible cardinality. The minimum dominating set is desirable since we want to decrease, as much as possible, the number of links and signaling messages exchanged among CH nodes.

Reducing the CHORIST network structure to the solution of the WCIDS problem helps understanding the topology, but does not solve the problem. Unfortunately, both the dominating set and the connected dominating set problems are proven to be NP-Complete [34] [48]. One of the most well-known heuristics for solving the connected dominating set problem is the centralized approach proposed by Guha and Khuller [49]. Although there are distributed implementations of this heuristic [47], the CHORIST topology is not exactly the same and the distributed approach cannot be used directly in this case. Furthermore, we must also consider that, unlike graphs, that are static, a network topology is dynamic: nodes may attach and detach from the network at any time, so the graph changes constantly.

Our protocol assumes a reactive approach; nodes perceive changes in their vicinity through periodic connections update messages sent by the CHs. As a result, the delay to react to changes is linked to the frequency of the update messages. If a mobile node gets out of the CH range it takes a few seconds for the node to realize that it may be in an area uncovered by any other CH and, thus, it is its duty to become a CH. Algorithm 2 presents the protocol in further details. Figure 7.3 shows the message passing diagram for the CH discovery process and creating a RN one, the (1) sign in the diagram indicates that the communication occurs with a single message transfer.

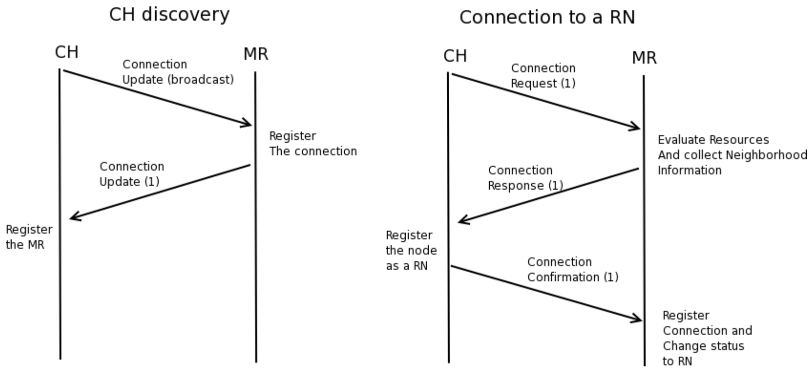

Figure 7.3: CHORIST CH discovery and RN connection request message transfer

The CHORIST hierarchy provides scalability to the network structure. In [14] Royer argues that hierarchical networks present better performance and are more robust. They enable the achievement of higher data throughputs. Another important characteristic of hierarchical networks is the decrease in the number of required links among nodes. This can be perceived from Figure 7.4, which compares the connectivity of the four evaluated methods. The number of links varies considerably among the approaches. The first diagram, WCIDS, shows the result of the application of the Weakly Connected Independent Dominating Set over the network connection graph. The second diagram shows the application of our technique over the same scenario. We can see that even though the clusters are at different positions the number of CHs, represented by bigger squares, is the same. Moreover, the number of generated edges is also nearly the same, even though for our approach they are generated dynamically and only with local information. The next two diagrams show the same nodes distribution connected through planar techniques. The number of created links, for both, is considerably bigger. In the k-nearest neighbor technique, each node connects to at least k neighbor nodes. For the fixed range technique, if two nodes are within the communication range of each other they are connected.

An important characteristic we want to emphasize about the problem is that both, RN and CH sets, should be WCIDS and as small as possible. However, both sets are not independent. The RNs selected to compose the RN WCIDS must be selected among the CH WCIDS nodes' neighbors.

Algorithm 2 - The CHORIST high level algorithm, from the point of view of the node

```
1: Node Arrives (actual status = IN);
2: Waits for Connection Updates;
3: if (received an Update message) then
4:    Sends a Connection Request to the CHs;
5:    Evaluates responses;
6:    Sends a Connection Confirmation to the best option;
7:    Becomes a MR;
8: else if (number of trials less than 3) then
9:    Returns to 2;
10: else
11:    Becomes a CH;
12:    Broadcasts a Connection Update;
13: end if
14: Waits for messages;
15: if (received a Connection Request && node is a MR a RN or a CH) then
16:    Responds with a Connection Response informing all its neighbors;
17: else if (received a Connection Confirmation && node is a CH a MR or a RN) then
18:    Registers the connection;
19:    if (requester is a CH && actual status != CH) then
20:       Becomes a RN;
21:    end if
22: else if (received a Connection Response) then
23:    Sends a Connection Confirmation;
24:    Registers Connection;
25: else if (received a Connection Update) then
26:    Registers the Update;
27:    Registers the Neighbor;
28:    if (actual state == CH && sender == CH) then
29:       There is another CH in the range;
30:       Decides, based on the his and the sender's ranks, whether to give up being a
          CH or not;
31:       Sends an Update Message;
32:       Waits a Random time;
33:    end if
34:    From time to time Evaluate Updates to find not Connected CHs;
35: else if (received a Connection Cancel) then
36:    Removes the connection;
37:    Reevaluates actual state (may become a MR);
38: end if
39: Return to 14;
40: //From time to time evaluates the connections and sends updates
41: if (connection timeout occurred) then
42:    Removes neighbor
43:    Reevaluates state (may became a IN or a MR)
44:    if (is a IN) then
45:       Return to 2;
46:    end if
47: end if
48: From time to time sends a Connection Update for the connected nodes;
```

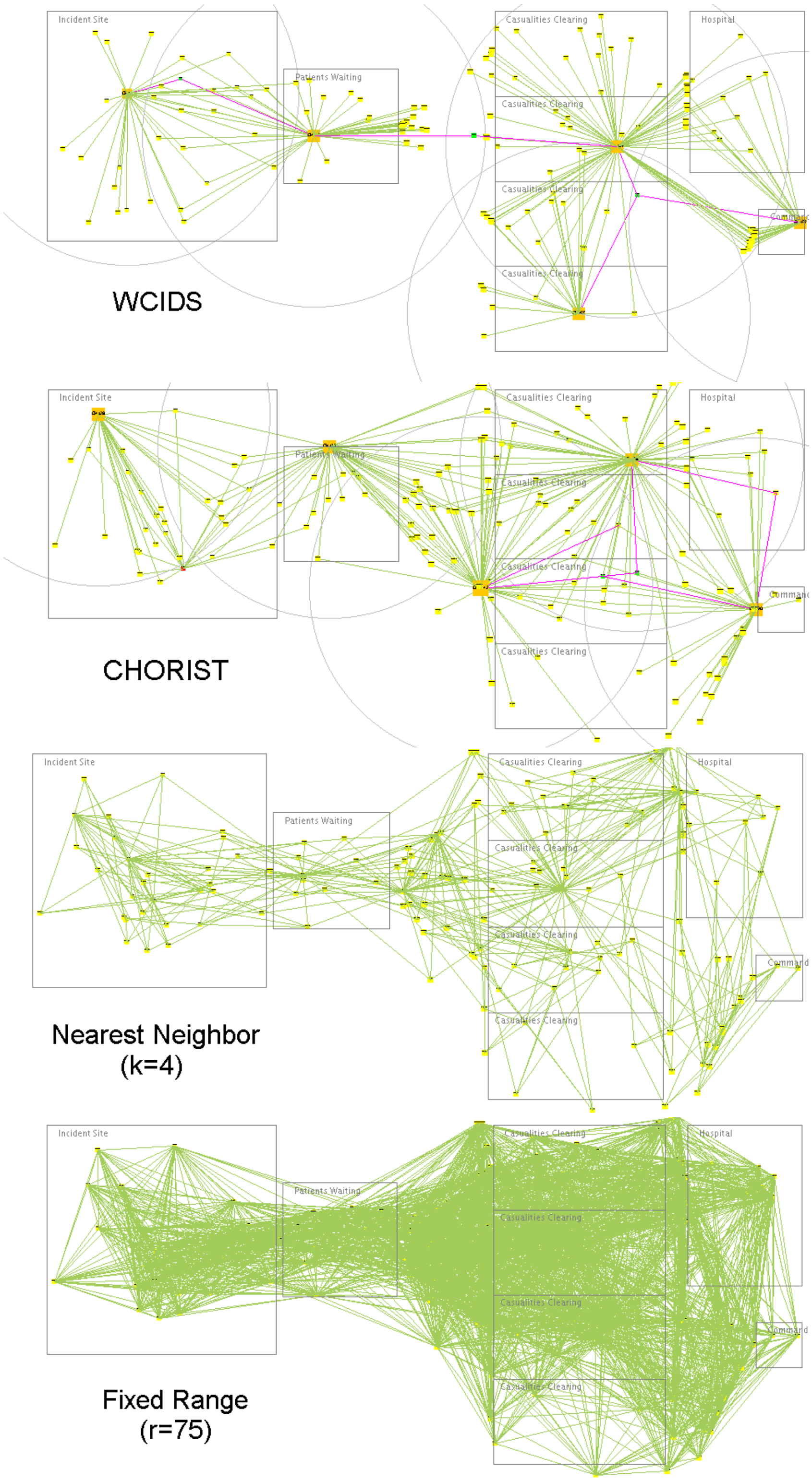

Figure 7.4: The connectivity of the different topology control strategies for the evaluated scenario (300x200m area, 150 nodes, 75m communication range)

Our solution fulfills this and all the other requirements of the CHORIST architecture requirements.

7.4 Experiments for the CHORIST Network Implementation

7.4.1 Environment

The evaluations were made using Sinalgo simulator [96] in a $2000x2000m^2$ area for the WCIDS. When using the Aschenbrucket al. distribution model we used the same area described in [9], $300x200m^2$ area. The area is composed of one incident site, one patient waiting area, four casualties treatment areas, one hospital and one command center. Figure 7.5 presents a typical cumulative histogram of nodes distribution through the space over the simulation time. We can see that the corridors between the defined areas are the places with higher concentration of nodes. This occurs because nodes are constantly moving through these spaces to get patients from one area and move them to another area. The peaks are the entrance ports of the areas, each node should wait for a random interval in this area to get or drop patients, so these places are supposed to have a bigger concentration of nodes.

We vary the number of nodes and their communication range. All experiments were conducted using Linux Fedora Core release 6 on 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 of 3 hours simulation time with different network configurations and 1% of message loss. For the comparisons with the WCIDS algorithm scenarios, nodes arrive randomly and are placed uniformly over the observed area.

The centralized WCIDS implementation is an adaptation of the Guha and Khuller [49] algorithm and works directly over the connection graph. This implementation is an oracle that knows the position of all nodes and uses this information to create the minimum arrangement in an offline manner. The final result is the best possible one and is hardly achievable with distributed algorithms, where nodes have only local information and new nodes arrive at different moments throughout the network lifetime. However, it represents a base of comparison to evaluate how far our implementation is from the theoretical minimal CH/RN optimal solution.

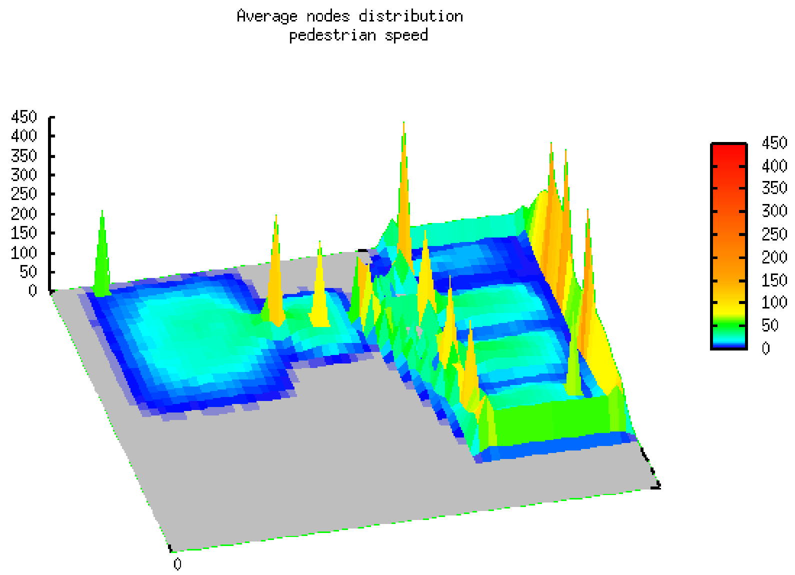

Figure 7.5: Cumulative histogram of nodes positions during a typical one hour simulation time

7.4.2 Minimum WCIDS proximity

Figure 7.6 presents a comparison between CHORIST and the offline WCIDS implementation for different density scenarios. The number of CHs created for both is close, normally with an overlap on the 99% confidence interval. This is the case even though our approach works in a distributed way, nodes just have local information and nodes arrive randomly during the network uptime. We can also perceive that the number of clusters increases sub linearly, relative to the number of nodes in the network. This indicates that for the CHORIST network the number of clusters has a closer relation to the are covered than the number of network nodes. On the other hand, the number of nodes per cluster increases almost linearly with the number of network nodes. Nevertheless the number of cluster nodes for both approaches, CHORIST and WCIDS stays basically the same for all evaluated scenarios. The number of RNs generated by our implementation of CHORIST has, on average 6.75% more RNs than the WCIDS implementation. This occurs mainly because for our approach, CHs chose their RNs in a selfish way. A CH picks the most interesting nodes, for its point of view to become its RNs, although this does not necessarily mean that these are the best nodes from the network point of view. Thus, it could happen that two CHs consider two different nodes to be relays between them, one for each CH, i.e. one RN on each communication direction. However, the increase in the number of RN has a good side since it decreases the size of the communication paths passing through the CHs.

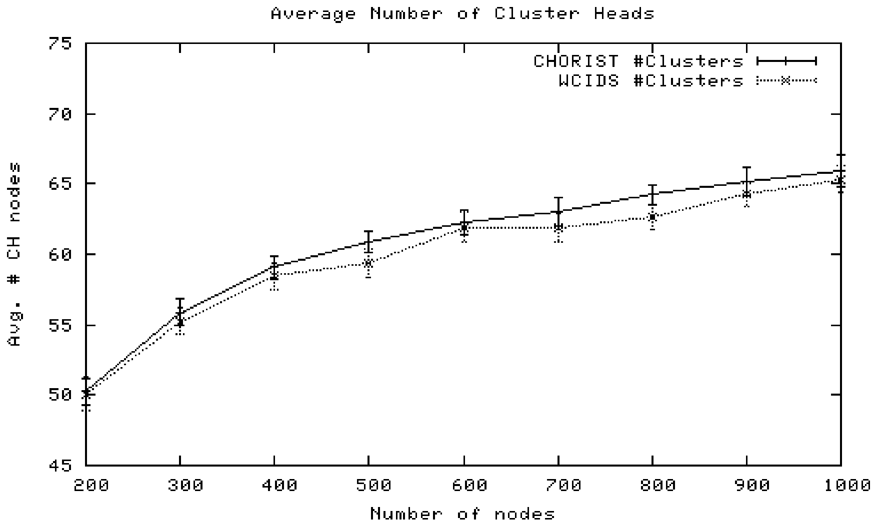

Figure 7.6: Average number of clusters on the network varying the number of nodes on the network and for 200m communication range

Figure 7.7 presents the average path lengths, between CHs, in the network. This measure is important because it reflects the traffic of control messages, e.g. scheduling, topology management, among the CHs. This traffic can be intense, so the shorter the paths the better. As we can see the size of the path for the CHORIST nodes is smaller than the ones for the WCIDS implementation. The variability given by increasing the number of CHs and RNs ensures a more diverse set of path options leading to a smaller average path length.

7.4.3 Mobility resilience

To evaluate the CHORIST network stability and availability we use the same distribution and mobility scenario proposed by Aschenbruck et al. [9]. Aschenbruck et al. propose a distribution model that divides the target area into different purpose specific sub-areas, e.g. incident location, patients waiting for treatment, hospital. Even though Aschenbruck et al. model is far

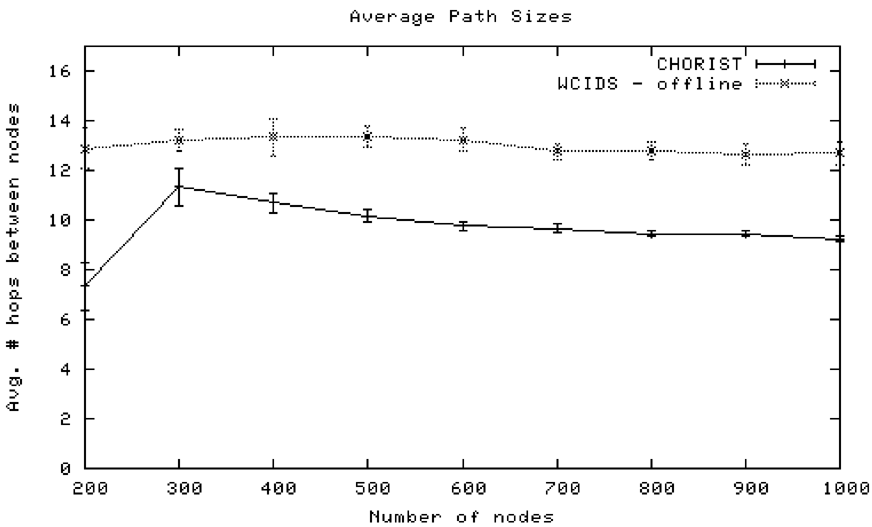

Figure 7.7: Average path size passing only through CHs and RNs, varying the number of nodes on the network and using $200m$ communication range

from covering all the possible mobility and distribution scenarios for PSNs, it is an elegant model based on a real maneuver simulation. However, instead of just evaluating the connectivity, as in [9], we implemented the protocols and compared our results with those of the other algorithms, under the same conditions.

We use the same area size, nodes distribution and organization described in [8] [9]. However, we simulated the network in two distinct situations, the first one when all nodes have pedestrian speed, ($0.5m/s$ on average and variance of $1m/s$) and another scenario where we have a mix of pedestrian and vehicular nodes. For the second scenario nodes inside the defined zones are pedestrian and nodes that travel from one zone to the other have vehicular speed (average of $40Km/h$ and variance of $4Km/h$).

The work of Aschenbruck et al. [9] does not propose a new algorithm, but compares three exiting planar proposals. No hierarchical strategy is evaluated, even though hierarchical networks are more scalable than planar ones.

One typical example of the application of the evaluated methods can be observed in Figure 7.4. In the k nearest neighbor technique, each node connects to at least k other neighbor nodes. However, one node can accept connections to other nodes that have less than k. This increases the network stability but may lead to some nodes having more than k links. The Algorithm 3 describes the technique. For the fixed range technique, if two nodes are within each other's communication range they should be connected. The protocol for both cases is nearly the same, the difference being that the fixed range does not consider the number of connections, if the node received a Connection Update it connects to the node that sent the update.

Algorithm 3 - The k nearest neighbor algorithm

```
Waits for messages;
if (received a Connection Update) then
    if (the number of connections < k) then
        Sends a Connection Request;
    end if
else if (received a Connection Request) then
    Responds with a Connection Response;
    Registers the connection;
else if (received a Connection Response) then
    Registers the connection;
end if
//From time to time evaluates the connections and sends updates
if (connection timeout occurred) then
    Removes connection
end if
From time to time broadcasts a Connection Update;
```

When exposed to higher mobility rates, transmission failures, delays, and lack of information the performance of the planar algorithms were slightly worse than those reported in [9]. Table 7.1 summarizes the obtained results. We can observe that the degrees of the nodes for the CHORIST architecture are the lowest ones, for both pedestrian and vehicular speed experiments. The percentage of nodes disconnected, columns of the table, are measured from the point of view of each node. They represent what percentage of the other nodes in the network are unreachable, from each node, at each time. For example, for an isolated node this value would be 100%, for the others, if all are connected, it would be 0.67%. Two nodes are connected only if the protocol recognizes them as being attached, and if they are indeed inside the communication range.

For all the evaluated protocols the addition of the vehicular speed nodes had a considerable impact. Every communication protocol needs a time to adapt to topology changes. As nodes are mobile, the view a node has of the topology, connectivity and other nodes' position may be outdated. Sometimes a node recognizes other nodes, which moved, as connected and at the same time may fail to recognize nodes within range as reachable. The CHORIST structure is a more sophisticated one, and it takes slightly more time for the nodes to get organized (e.g. recognize new clusters, attach to them). For this reason more nodes fail to recognize connections, when compared with the k-neighborhood algorithms. However with the increase in mobility, the k neighborhood needs considerably more resources, i.e. links, to reach the same results as those presented by the CHORIST structure.

From Table 7.1 we also have the average path length and the average longest shortest path for each node ($\forall i, j \in V(G) : ls = max_{ij} d(i, j)$), both measured in number of hops. Again, CHORIST paths were smaller than the corresponding k-neighborhood ones. The k-neighborhood algorithm needs $k = 8$ or $k = 10$ to present the same path lengths CHORIST does. However, this also means spending more resources to generate and maintain the structure.

Topology Control Strategy	Pedestrian Avg. node degree	Pedestrian + vehicular Avg. node degree	Pedestrian % of nodes disconnected	Pedestrian % of nodes disconnected	Pedestrian Avg. path size	Pedestrian +vehicular Avg. path size	Pedestrian max path size	Pedestrian +vehicular max path size
CHORIST	2.90	3.38	10.22	19.25	2.23	2.13	4.18	3.97
K nearest neighbor (k=3)	3.25	2.29	6.78	48.03	5.35	4.4	10.51	8.85
K nearest neighbor (k=4)	4.27	3.01	2.03	37.62	4.16	3.86	7.91	7.56
K nearest neighbor (k=6)	6.09	4.66	0.41	20.77	3.23	3.36	6.13	6.43
K nearest neighbor (k=8)	7.97	6.49	0.22	8.64	3.08	1.12	5.33	5.89
K nearest neighbor (k=10)	9.74	8.21	0.21	3.84	2.89	1.13	2.88	2.94
Fixed range 100m	75.14	82.14	0.20	0.22	1.12	1.13	2.88	2.94

Table 7.1: Summary of the disaster area scenario results for both pedestrian and pedestrian plus vehicular scenarios

7.5 Conclusions

This chapter presented an implementation and evaluation of the network architecture proposed by the CHORIST project. The problem was reduced to the minimum Weakly Connected Independent Dominating Set. Even though this problem is NP-complete, our solution reaches values close to the theoretical minimum, using only local information and with nodes arriving at the network at different times. From the mobility experiments we can also conclude that implementation of the CHORIST architecture is stable and able to guarantee relatively low percentage of disconnected nodes while simultaneously decreasing the average path lengths and number of links per node. The proposed topology is stable and resilient to nodes mobility.