A Planar Group-Based Architecture to Scale Ad-Hoc and Sensor Networks

Jaime Lloret[1], Miguel Garcia[2], Fernando Boronat[3], Jesus Tomás[4]
Universidad Politecnica de Valencia, Valencia, Spain
[1]jlloret@dcom.upv.es, [2]migarpi@posgrado.upv.es, [3]fboronat@dcom.upv.es, [4]jtomas@dcom.upv.es

***Abstract*—It is known that grouping nodes gives better performance to the group and to the whole system, thereby avoiding unnecessary message forwarding and additional overheads while allows to scale the network considerably. Many routing protocols for ad-hoc networks and sensor networks have been designed, but none of them is based on groups. In this paper, after a review of group based architectures and of neighbor selection strategies, a planar group-based network architecture is proposed. In the proposal, the network is formed by several groups of ad-hoc devices or sensors. Connections between groups are established as a function of the proximity and the neighbor's available capacity (based on the ad-hoc device or sensor's energy). The messages that are needed to the proper operation are shown. It is also simulated how much time is needed to propagate information between groups and it is calculated the diameter for different topologies**

***Index Terms*—Group-Based Topologies, Group-based architectures, Group-based routing algorithm, Ad-Hoc and Sensor Networks.**

I. Introduction

Many routing protocols can be applied to ad-hoc and sensor networks. They can be distinguished depending on many parameters, but the most used is the topology classification. Next paragraphs will give a brief overview on sensors and ad-hoc routing protocols.

On one hand, sensor routing protocols can be classified into two groups [1] [2]. The first group is formed by protocols based on the network topology and it could be broken down into three subgroups:

1- Plane routing. All nodes in the network have the same role and perform the same tasks. Because of the number of nodes in these networks, the use of a global identifier, for every node, is not feasible. It uses a data-centric routing where the base station sends requests to some regions and the nodes from that regions reply. Some of the algorithms in this group are SPIN [3], Direct diffusion [4], Rumour routing [5], MCFA [6], GBR [7], IDSQ [8], CADR [8], COUGAR [9], ACQUIRE [10], and so on.

2- Hierarchical routing. It is very scalable and has an efficient communication. It has been designed for energy saving purposes, because central nodes have unlimited energy, while leaf sensors have limited energy. When the sensor network topology is formed, data can be routed. Some algorithms such as LEACH [11], PEGASIS [12], TEEN [13], APTEEN [14], MECN [15] and Two-Tier Data Dissemination [16] are hierarchical routing algorithms.

3- Position-based routing. All data is routed through the sensors depending on their position. Distances between sensors are known because of neighboring sensors signals. There are other protocols that base node's situation on GPS and, using that information, route the data to the most adequate sensor. These algorithms consume more energy than others because of the need of GPS signal. Some of those algorithms sleep sensors when the network has not any activity. Some examples are GAF [17], GEAR [18], GOAFR [19], POSANT [20], MACGSP [21] and SPAN [22].

The second group does not have into account the structure of the network. It can be broken into five subgroups:

1- Multipath Routing Protocols. The information could reach the destination through different paths. Because sensors have to calculate several paths, they use a main route when they have enough energy; otherwise, they use an alternative path.

2- Query-Based Routing protocols. They are based on a central node that sends a query about an event to the specific area. When the query arrives to that area, it is routed to the destination sensor, and then it will reply. A sensor from an area could be sleeping, saving energy, while there is not any query to that area.

3- Negotiation-Based Routing Protocols. Before data transmission, the sensor has to negotiate the data it has to send, so redundant data could be deleted, and resources will be available while data exchange. SPIN [3] protocols use this type of routing, but they take into account the network structure.

4- QoS protocols. The information is routed to the sensors taking into account quality parameters such as delay, energy and bandwidth. SAR [23] and SPEED [24] protocols are based on QoS algorithms.

5- Data coherent/incoherent processing based protocols. These algorithms use several routing techniques taking into account the data processing of a coherent or incoherent result.

On the other hand, MANET networks [25] are a type of ad-hoc networks much more studied and mature than Wireless Sensor Networks (WSN) [26]. Independently of the medium access method used [27], in the recent years many routing protocols have been developed for MANET networks [28] [29]. The nodes' mobility, the lack of stability of the topology, the lack of a pre-established organization and performing of the wireless communications are reasons for not using the routing protocols developed for fixed networks. There are standardized routing protocols for MANET networks used by different fixed or mobile devices.

Depending on the information type exchanged by the nodes and on the frequency they do it, the routing protocols in ad-hoc networks are divided into three types: proactives, reactives and hybrids.

The proactive protocols update the routing tables of all the nodes periodically, even though no information is being exchanged. When a topology change occurs, the routing table is updated and the routing protocol finds the optimum route to forward the information. A periodical control protocol message exchange allows this, but consumes bandwidth and energy (batteries). OLSR (Optimized Link State Routing Protocol) [30] is a proactive protocol in which each node knows permanently the network state and the number, availability and addresses of the nodes. This performs a faster routing protocol.

The reactive protocols only maintain routing routes in their tables when a node has to communicate with another node in the network. With these protocols, when a communication starts, as the right route is unknown, a route discovering message is sent. When the response is received, the route is included in the routing tables and the communication is now possible. The main disadvantage of these protocols is the latency at the beginning of the communications (route discovery time) but they improve the network and energy resources use. Inside this kind of protocols, the source-based protocols and hop-by-hop protocols are located. AODV (Ad-Hoc On-demand Distance Vector) [31] is a reactive protocol. It has a minimalist behavior because it hardly overloads the ad-hoc network and needs very few memory comparing with other protocols. DSR (Dynamic Source Routing Protocol) [32] is also a reactive protocol developed specifically for ad-hoc networks. It only sends information when it is required, saving bandwidth, energy and battery.

Finally, the hybrid protocols are a combination of the other two types of routing protocols, taking the advantages of both of them. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding, so these protocols divide ad-hoc networks into different zones, where near nodes use proactive routing meanwhile far nodes use reactive routing. An example of a hybrid protocol is TORA (Temporally-Ordered Routing Algorithm routing protocol) [33].

Despite of the type of node is talking about (ad-hoc devices or sensor nodes), from now both of them will be called just nodes.

None of the routing protocols aforementioned are group-based. Our proposal is to divide the network of nodes into several groups and if a node has to send data to other groups, when this data arrives to one node from a group, it is propagated to the rest of nodes in its group.

The rest of the paper is structured as follows. Section 2 describes group-based architectures and gives some examples to distinguish between group-based and non group-based architectures. Then, section 3 will show several neighbor selection strategies that can be used to establish connections between neighboring nodes. The description of our proposal, its analytical model and the equations that will follow the nodes to select their neighbors are shown in section 4. The protocol operation and the messages designed for its proper operation are shown in section 5. All analysis done with the protocol are shown in section 6. It is also shown the messages bandwidth consumption. Finally, section 7 summarizes the results and gives the future research.

II. GROUP-BASED ARCHITECTURES

First, a distinction have to be presented between a groupware architecture, where all nodes collaborate towards the proper operation and the success of the purpose of the network (despite of where they are placed and the relationships between them), and group-based architecture, where the whole network is broken down into groups and each group could perform different operations (but information could be transmitted between groups and there must exist connections between nodes from different groups). In a wireless group-based architecture, a group consists of a set of nodes that are close to each other (in terms of geographical location or in terms of round trip time). These groups could have a link if a node from a group is near a node from another group. The distance between two nodes is given by the network latency or the round trip time.

The main goal in a wireless group-based topology is the network protocol and the group management, that is the design of an efficient algorithm for a new node to find its nearest (or the best) group to join in. Then, some important issues must be designed despite of the routing protocol inside each group (it could be different for each group): (i) How to build neighboring groups, and (ii) a protocol to exchange messages between neighboring groups.

The performance of the group-based network highly depends on the efficiency of this nearby group locating process. All works found in the literature, where nodes are divided into groups and connections are established between nodes from different groups, have been developed to solve specific issues such as distribution service in multimedia networks [34], group-based games over NAT/Firewalls [35], and so on.

Two types of group-based topologies can be distinguished:

1- Planar group-based topologies. All nodes perform the same roles and there is just one layer. However, in some works there is a directory server or a rendezvous point (RP) for content distribution coordination.

2- Layered group-based topologies. Nodes from layered group-based topologies could have several roles (2 roles at least). Depending on which type of role they are running, they will become to a specific layer. All nodes in the same layer will have the same role. There will be connections between nodes from the same layer and from adjacent layers.

There are several differences between both group-based topologies. While layered group-based topologies grow structured organized by upper layers, planar group-based topologies grow unstructured without any organization. In layered group-based topologies anyone can know exactly where each group is and how to reach it. Otherwise planar group-based topologies, every time a node wants to reach other group, the message should travel through many unknown groups due to groups join the network as they appear. Delays between groups in layered group-based topologies could be lower because connections between groups can be established taking into account this parameter, otherwise, in planar group-based topologies connections between groups are established by group's position, their geographical situation or because of their appearance in the network. Layered networks address several complexities because nodes could have several types of roles and fault tolerance for every layer must be designed. On the other hand, planar networks are more simplex because all nodes have the same role. In order to have scalability, layered group-based topologies must add more layers to its logical topology, while planar group-based topologies could grow without any limitation, just the number of hops of the message.

Cluster-based networks are a subset of the group-based networks, because each cluster could be considered as a virtual group. But a group-based network is capable of having any type of topology inside any group, not only clusters. In cluster based architectures each cluster has adjacencies with other clusters and all clusters have the same rules. A cluster can be made up of a Cluster Head node, Cluster Gateways and Cluster Members [36] [37]. The Cluster Head node is the parent node of the cluster, which manages and checks the status of the links in the cluster, and routes the information to the right clusters. The rest of the nodes in a cluster are cluster members. In this kind of network, the Cluster Head nodes have a total control over the cluster and the size of the cluster is usually about 1 or 2 hops from the Cluster Head node. A cluster member does not have inter-cluster links.

A routing protocol based on zones can be found in the literature. It is the Zone Routing Protocol (ZRP) [38] [39]. Each node proactively maintains routing information for a local neighborhood (routing zone), while reactively acquiring routes to destinations beyond the routing zone. ZRP and our proposal have several common features, e.g. they could be applied over any type of routing protocol, they scale well and the information is sent to border nodes in order to reach destinations outside their zones. The main difference between them is that in ZRP each node maintains a zone and the nodes in that zone have different nodes in their zone while in our proposal all the nodes that form a group have the same nodes in their group.

On the other hand, other works of groups systems such as the following will not be considered in this paper:

1- The community based mobility model for ad hoc network research presented in [40], because although the network is organized in groups, and nodes can move from one host to another, there is not any connection between border nodes from different groups.

2- The landmark hierarchy presented in [41] because although there is a node with higher role which has connections with nodes from other groups, its leaf nodes do not have.

3- Another example similar to the last one is the BGP routing protocol architecture [42]. The routers inside the autonomous system do not have connections with routers from other autonomous systems unless they are BGP routers. They are the backbone of the network, so they are in the highest hierarchy level.

4- It is not considered moving groups such as Landmark Routing Protocol (LANMAR [43]), where the set of nodes move as a group, so the group can enlarge or diminish with the motion of the members.

5- Multicast groups for ad-hoc and sensor networks [44], because there are not connections between groups.

III. NEIGHBOUR SELECTION STRATEGIES

In order to obtain a network topology, the nodes have to be interconnected, so there has to be a strategy that nodes must follow to select their neighbors. A neighbor selection algorithm decides which parameters are used, and their values, to select the best neighbor node. This election is given taking into account that the network has to accomplish an objective function. It must take also into account how connections are distributed in the network.

Each topology, each system and each protocol has the development and the improvement of its neighbor selection as its main goal. Basic neighbor selection algorithms depend on the node upstream bandwidth (BW) and/or on the number of connections (n).

A. Based on genetic algorithms

A genetic algorithm (GA) is stochastic and an adaptive heuristic search technique used in computing to find exact or approximate solutions to optimization and search problems which is based on the mechanism of natural selection, genetics, and evolutions. Genetic algorithms use techniques inspired by evolutionary biology. They represent an intelligent exploitation of a random search within a defined search space to solve a problem. It has been proven to be an effective tool to solve complex search problems with large solution spaces. In order to define a typical genetic algorithm, it is required a genetic representation of the solution domain and a fitness function to evaluate the solution domain. Genetic Algorithms start with an initial set of random solutions called population. Each individual in the population, called a chromosome, is an encoded string of symbols representing a solution, which may be feasible or not.

Simon G. M. Koo et al. [44] [45] proposed a genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks, which enhances the decision process performed at the tracker for transfer coordination. It increases content availability to the clients from their immediate neighbors. This neighbor selection strategy is being used for BitTorrent P2P network.

B. Based on Distributed Hash Tables (DHT) structures

Many systems locate nodes in the topology based on key-based graph structures such as CAN [46], Chord [47], Pastry [48] and Tapestry [49]. Their files are associated with a key (produced, for instance, by hashing the file name) and each node in the system is responsible for storing a certain range of keys. When a node joins the network, takes a key as input, and routes a message to the node responsible for that key, in response. Nodes have identifiers which are taken from the same space as the keys (i.e., same number of digits). Each node maintains a routing table consisting of a small subset of nodes in the system (their neighbors). In DHT structures, neighbors are selected as a function of the values of the key of the new node. When a node receives a query for a key for which it is not responsible, the node routes the query to the neighbor node that makes the most closest towards resolving the query.

The number of neighbors differs for each network. In Tapestry and Pastry a node has O(log n) neighbors, while in CAN has O(d) neighbors (d is the dimension of the toroidal space). Chord maintains two sets of neighbors. Each node has a successor list of k nodes that immediately follow it in the key space. There is a finger list of O(log n) nodes spaced exponentially around the key space. These systems do not take care of the underlying network, so a neighbor of a node could be very far (in terms of Round Trip Time, RTT).

C. Based on the Internet underlying network

Trying to achieve the goal of having neighbors close “logically”, several systems have been proposed.

Plethora [50] is based on a two-level overlay architecture. The global overlay serves as the main data repository, and there are several local overlays that serve as caches to improve access time to data items. Local overlays contain nodes which IP are closer. Nodes that are in the same Autonomous System (AS) should be in the same local overlay together with nodes of neighboring ASs. The global overlay is implemented using any prefix-based DHT system. It is used as the main repository and helps direct nodes to local overlays where they belong. The authors of this work adopt Pastry as Plethora’s underlying algorithm to support the local overlay. A node builds a list of ASs over the time that can be presented in its local overlay. Authors also propose to build this list using traceroute to some of the nodes in the node’s state tables. The node can measure the delay to each member of its routing table using the traceroute. If the delay is less than a system parameter, it includes the AS of the probed node in its neighborhood list. In each local overlay there is a node, the local overlay leader, which controls the number of nodes in the local overlay. Each node in a local overlay maintains a pointer to its current leader to determine if the leader has departed or failed.

T-DHT [51] is a scalable and distributed algorithm for the construction of distributed hash tables which are strongly oriented to the underlying network topology. The system is based on a virtual coordinate system. To build it, three (or more) reference nodes are randomly selected. Each node triangulates its position in the virtual coordinate space from these reference nodes. The virtual coordinate space is node’s position in the network topology, but not the physical position. Inspired by the Content Addressable Network (CAN), the authors of the work construct a two-dimensional DHT on the top of the virtual coordinate system. The coordinate space is divided among the participating nodes. Each node maintains a rectangular area around its position in the virtual coordinate system. It also maintains a routing table containing the path to its neighbors in the coordinate system and the area each neighbor maintains. A node may have links to nodes which are not direct neighbors in the hash table. Three steps must be performed to join the T-DHT. First, the node wishing to join must first find a node which is already in the T-DHT. Then, via T-DHT routing, it finds the node maintaining the zone of its position in the virtual coordinate system. This zone is equally split between the two nodes. Finally, the new member informs its neighbors about its presence.

To bootstrap, the first reference node maintains the whole DHT. When the virtual coordinates are assigned, it announces its T-DHT membership to its neighbors. Next, the neighbors join the distributed hash table and themselves announce their membership to their neighbors. The joining node only needs to contact the node maintaining the corresponding area.

D. Based on the RTT to the neighbour

It is important to achieve lower delays to exploit proximity in the underlying network in terms of RTT. Otherwise, each overlay hop has an expected delay equal to the average delay between a pair of random overlay nodes, which stretches route delay by a factor equal to the number of overlay hops and increases the stress in the underlying network links. Proximity neighbor selection (PNS) can be used to achieve low delay routes and low bandwidth usage. It selects routing state entries for each node from among the closest nodes in the underlying topology that satisfy constraints required for overlay routing. On the other hand, finding effective ways to produce proximity information is crucial for overlay networks to route efficiently. The proximity information can be used to partition nodes into clusters or to estimate distances among them.

M. Castro et al. presented in [52] a proximity neighbor selection (PNS) using heuristics approximations (called constrained gossiping (PNS-CG)). It can be used over DHT structures such as Pastry or Tapestry. The flexibility in the choice of nodeIds to fill routing table slots can be exploited to implement PNS effectively. Proximity neighbor selection picks the closest node in the underlying network from among those whose nodeIds have the required prefix. The proximity metric used in the definition of closest is RTT. In PNS-CG, when a new node x with nodeId X joins the overlay, it must contact an existing overlay node which routes a message using X as the key. The new node obtains the nth row of its routing table from the node encountered along the path from the existing overlay node to X whose nodeId matches X in the first n-1 digits. Then, it updates other node's routing tables. When x's resulting routing table is updated, the closest node can be found using a specified algorithm designed by them.

It performs both low delay routes and low bandwidth usage with low overhead than other protocols such as Pastry and Tapestry, although the algorithm uses the routing state maintained by Pastry to locate nearby seed nodes for joining the network.

Others systems, such as the one presented by X. Zhichen in [53], locate new nodes in the topology using landmark clustering, as a preselection process to find nodes that are possibly close to a given node, and then perform RTT measurements to identify the actual closest node. Each node is assigned a landmark number that reflects its physical position in the network. Landmark clustering is based on the intuition that nodes close to each other are likely to have similar distances to a few selected landmark nodes. A node uses its landmark number as the DHT key to access relevant proximity information. To effectively use the proximity information generated, the information of the system is stored as soft-state in the system itself. To guide the placement of proximity information landmark clustering is used. Later, this information is used for nodes to discover other nodes physically near.

E. Based on their geographical location

The type of networks where it is more needed to chose the neighbors geographically because of their features are wireless ad-hoc and sensor networks. In these networks, connections are established only if they are closed, because of their coverage area limitation. In these types of networks, all nodes must know their own positions, either from a GPS device, if outdoors, or through other means and its neighbors' positions. There use to be a location registration and lookup service that maps node addresses to locations. If a node knows its neighbors' positions, the locally optimal choice of next hop is the neighbor geographically closest to the packet's destination. Forwarding in this regime follows closer geographic hops, until the destination is reached.

The self-describing nature of position is the key to geography's usefulness in routing. The position of a packet's destination and positions of the candidate next hops are sufficient to make correct forwarding decisions, without any other topological information. Examples given are the Routing with guaranteed delivery in ad hoc wireless networks presented by P. Bose et al. in [54] (also called GFG algorithm), they described a routing algorithm that guarantees delivery of messages in MANETs, and GPSR geographic routing algorithm presented by Karp, B. in [55] (they transformed GFG algorithm into a protocol). In geographic routing packets are stamped with the positions of their destinations. Their graph is planar, that is, there are enclosed polygonal regions bounded by edges. In planar graphs, there could be loops when the destination is disconnected.

Although grouping nodes give many benefits to the network, we have not found any neighbor selection strategy for group-based wireless sensor networks and none of them tackles the way of establishing connections and discovering neighbors between nodes from different groups. On the other hand, in none of them connections are established using nodes' capacities.

IV. ARCHITECTURE DESCRIPTION

A. Architecture Operation

An architecture, where the structure of nodes is based on the creation of groups and the nodes have the same functionality in the network, is proposed. In each group exists a central node that limits the zone where the node from the same group will be placed, but its functionality is the same that the rest of the nodes in the network. Every node has a node identifier (called *nodeID*) that is unique in its group. The first node in the network acquires a group identifier (called *groupID*) that could be given manually, or using GPS, or using a wireless location system or by other means. New joining nodes will know their group identifier from their new neighbors. Border nodes are, physically, the edge nodes of the group. When there is an event in one node, this event is sent to all the nodes in its group in order to take the appropriate actions. All groups could have the same or different routing protocol inside. All nodes in a group can know all information about their group using the routing protocol (see references [30], [31], [32] or [33] as examples). Border nodes have connections with other border nodes from neighbor groups and are used to send information to other groups or to receive information from other groups and distribute it inside. Because it is wanted a fast routing protocol, OLSR [30] has been chosen to route information inside all groups, but it can be changed by other routing protocol depending on the network's characteristics. When the information is for a node of the same group it is routed using the *nodeID*. Every node runs its link state database locally and selects the best path to a destination based on a metric (described later).

When the information has to be sent to other groups, the information is routed directly to the closest border node to the destination group using the *groupID*. When a node from destination group receives the information, it routes it to all nodes in its group using OLSR protocol. Links between border nodes from different groups are established primarily as a function of their position, but in case of multiple possibilities, neighbors are selected as a function of their capacity. In order to establish the boundaries of the group, two choices can be considered: (i) limiting the diameter of the group to a maximum number of hops (e.g., 30 hops, as the maximum number of hops for a tracer of a route), and (ii) establishing the boundaries of the area that it is wanted to be covered. Figure 1 shows the proposed architecture topology.

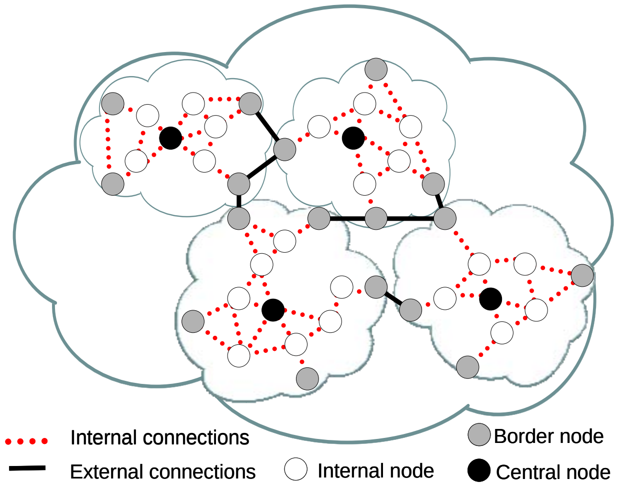

Figure 1. Proposed architecture topology.

TABLE I.
EQUATION OF THE PARAMETERS

Parameter	Equation
λ	$\frac{Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1 - \frac{E^2}{K_1}}$
C	$\frac{T{\cdot}K_3}{\lambda}$
Metric	$\sum_{i=1}^{r} C_i$

B. Analytical Model and Neighbor Selection Algorithm

Every node is characterized by 3 parameters: *nodeID*, *groupID* and *λ*. *nodeID* and *groupID* have been defined before. λ parameter is defined in order to introduce the capacity of the nodes in our architecture. It is used to determine the best node to connect with. The higher the λ parameter, the best node to connect with is. In our proposal, λ parameter depends on the node's number of available links (*Available_Con*) and its maximum number of links (*Max_Con*), because the number of connections will affect to the performance of the node, its % of available load, because the load of the node has a direct implication on its capacity, and its energy consumption, because it is better nodes with higher energy. λ equation is shown in table I, where *L* is the available load and *E* is the energy consumption. Their values vary from 0 to 100. E=0 indicates it is fully charged, so λ parameter is 0 and E=100 indicates it is fully discharged. K_1 defines the minimum value of energy remaining in a node to be suitable for being selected as a neighbor. K_2 gives different λ values from 0 in case of *L*=0 or *Available_Con*=0. K_2 has been considered equal to 100 in order to get λ into desired values.

Figure 2 shows λ parameter values when the maximum number of links for a node is 16 when its available number of links for different available energy values of the node. Node's load is fixed to 50%.

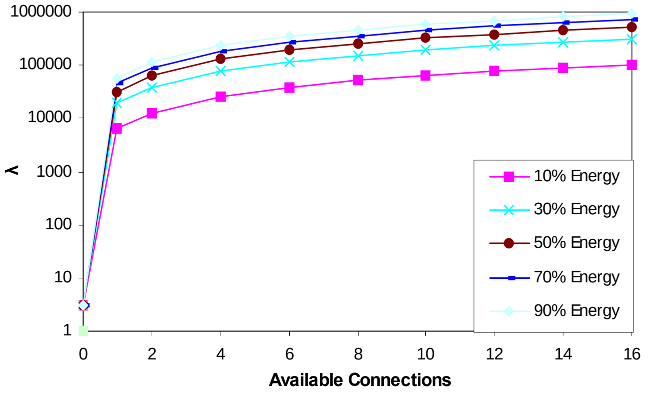

Figure 2. λ values for different available connections and different energy values.

The cost of the i^{th}-Node has been defined as the inverse of the i^{th}-Node λ parameter multiplied by *T* (the delay of its reply in msec). It is shown in table I. K_3=10^3 gives *C* values higher than 1. The *metric* for each route is based on the hops to a destination (r) and on the cost of the nodes (C_i) in the route as it is shown in table I. The metric gives the best path to reach a node.

V. PROTOCOL OPERATION AND MESSAGES

This section describes how the designed protocol operates. Messages designed and their bandwidth consumption are also shown.

A. Group Creation and Maintenance

Let a new node be in the network (it could be the first). Its operation will follow the steps of the flow chart shown in figure 3. First, it will send a hello message (called *helloGroup*) in order to join a group. If there is no response from any node for 3 seconds, there is not any group in the network and the node considers itself as a central node of a group, and it will take the value *groupID*=1 and *nodeID*=1. When the node receives *helloGroup ACK* messages from several candidate neighbors, first it puts a time stamp in their reply and chooses the best nodes to have a link with (this election is taken based on the λ parameter which comes in the *helloGroup ACK* message). The time stamp will be used to calculate *C* parameter. Responses received after 3 seconds will be discarded because they are so far compared with the receive ones. In case of receiving replies from nodes of different groups, it will choose the group which replies have the highest average λ parameter, so it will take into account replies only for that group. Then, the node will send an *okGroup* message to the selected neighbors in the same group, and the neighbors will reply with the *okGroup ACK* message with the assigned *nodeID* and indicating the link has been established. Nodes have a table with all its neighbors of its group. Nodes will broadcast *keepalive* messages periodically to their neighbors. If a node does not receive a *keepalive* message from a neighbor before the dead time, it will remove this entry from its database and will start the group update process. As the *groupID* is in the *helloGroup ACK* message, the new node will know which group has joined. Finally, the neighbor node will send a *newNode* message to the central node, to run the algorithm for changing the central node if it is needed.

When a new node joins the group, the central node of the group could be changed. The procedure designed for changing the central node is as follows. The group diameter (d_{group}) has been defined as the shortest number of hops between the two most remote nodes in the group (in our case, $d_{group} \leq 30$). The central node is in the central place, so its distance from the most remote boundaries of the group is $d_{group}/2$. Let's suppose that there is a distance of d_1 to one side and d_2 to the other side of the most remote sites. Initially $d_1=d_2$, and only when one of the distances 2 hops bigger than the other the central node is changed, otherwise, it keeps being the central node. It will happen for any orientation of the network. When one distance is 2 times bigger than the opposite, the central node is changed to that side so both distances become equal. The old central node will send a *centralrole* message to the new central node in order to indicate this change, and then the new central node will reply with the *centralrole* ACK message in order to acknowledge becoming the new central node.

When there is a change of the central node of a group, all the nodes in the group must be advised. In order to update all nodes in the group, the new central node will send a *changeCentral* message to indicate the new central node and the distance from it to the node processing this control packet. This update is distributed using the OLSR algorithm. Once the network has converged, every node sends *keepalive* messages periodically to its neighbors. Figures 4 and 5 show the messages procedure when the central node changes and when it doesn't.

Links between border nodes from different groups are established based on their replying delay and, in case of draw, on the λ parameter of the replying nodes, but it could be changed by any algorithm using node's position or choosing the neighbor with the lower distance (in number of hops) to the central node. When our proposal is based on the λ parameter, the load of the network between groups is distributed, but when our proposal is based on node's position or choosing the neighbor with the lower distance to the central node the number of nodes in the groups is balanced.

Two choices are proposed to establish the boundaries of the group:

1) When the boundaries of the group are the same of the area that it is wanted to be covered, border nodes are known using GPS.
2) When the boundary of the group is limited by the diameter of the group, the maximum number of hops from the central node must be known. Every time a new node joins a group, it receives the newNode ACK message with the number of hops to the central node. When it achieves the maximum number of hops, the node is marked as a border node, and it will inform to new joining nodes that they must create a new group.

B. Leavings and Fault Tolerance

When a node leaves the group, it will send *nodeDisconnect* message to its neighbor nodes. They must reply with a *nodeDisconnect ACK* message and the neighbor node closest to the central node will send the *nodeDisconnect* message to the central node. The central node distributes the update information using OLSR algorithm. If the neighbor node doesn't have links with other neighbors, it must start a new connection process sending a *helloGroup* message. The procedure is shown in figure 6.

If the leaving node is the central node, it assigns the central node role to the best candidate to be the central node (in case of draw, it will choose the one with higher λ parameter, this node is called backup central node and will be always kept advised by the central node) using the *centralRole* message and when it receives a *centralRole ACK* message from new central node, leaves the group. Then, the new central node will send a *changeCentral* message to the group to inform the change. If the central node fails down without any advertisement, the backup central node will become central node because the lack of keepalive messages from the leaving central node. Figure 7 shows all the steps explained.

When a regular node fails down, its neighbor nodes will know the failure because of the missing of *keepalive* messages. The procedure is the same as when the node leaves the network voluntarily. Neighbor nodes will delete that entry from their database and if a node keeps without any neighbor, it will begin the discovery process as a new node in the network.

C. Connections between different groups

Once a node has joined a group, it is able to send and receive keepalive messages as it has been explained before. Nodes have 2 tables, first one has all its neighbors of its group and the second one has all its neighbors from other groups. When a node receives a broadcast keepalive message from a node that belongs from another group, it checks if it is in its other groups' neighbors table. If it is inside, it resets the keepalive time for that entry to zero, but if it is not inside, it adds that entry to the table. The flow chart shown in figure 8 explains the procedure explained.

D. Message fields

11 messages (see figure 9) have been designed. Several fields for future purposes, such as version, length and options fields, have been included. In order to reserve enough bits for the parameters 32 bits have been used for the λ, the GroupID and the nodeID parameter. These messages have to work together with the routing protocol used in the group. See reference [30] for OLSR protocol, reference [31] for AODV protocol, reference [32] for DSR protocol and reference [33] for TORA protocol.

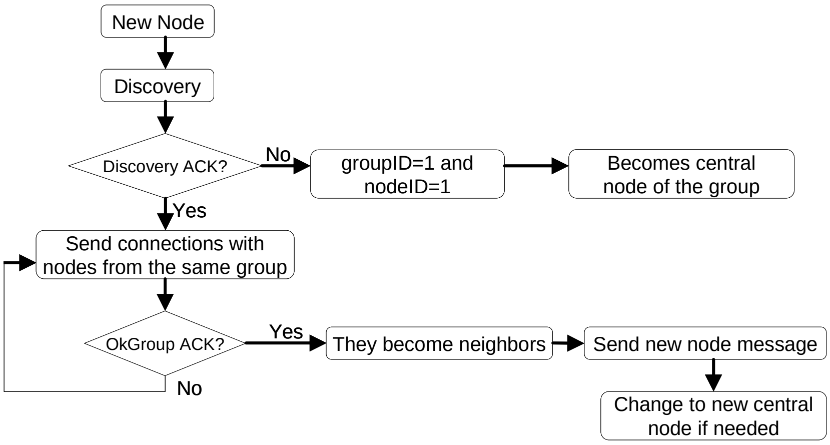

Figure 3. Flowchart of the discovery operation of a new node.

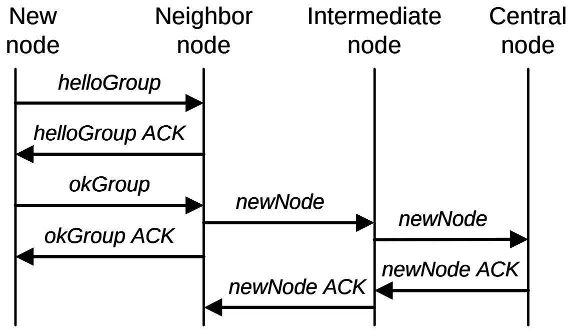

Figure 4. Messages when central node changes.

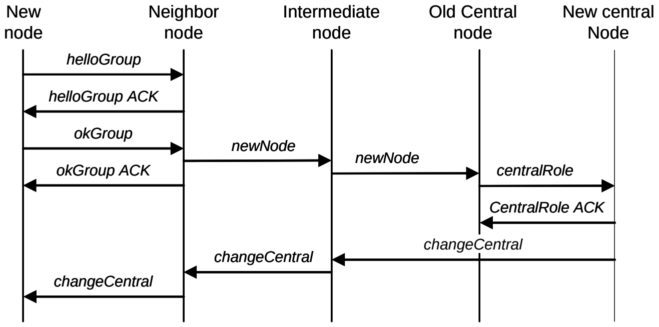

Figure 5. Messages when it doesn't change.

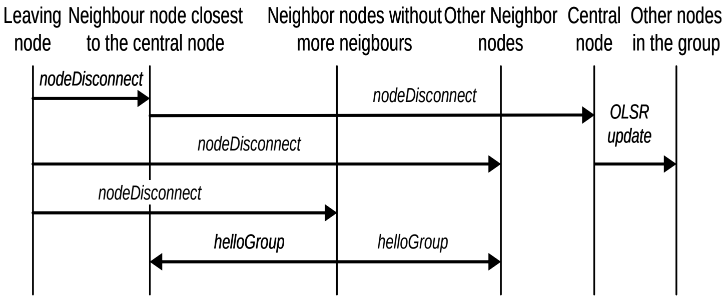

Figure 6. Messages when it doesn't change.

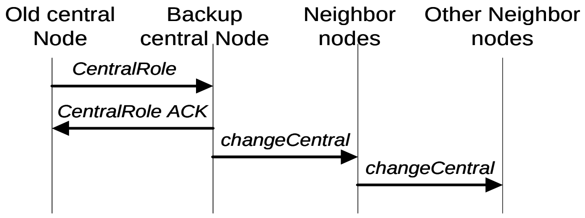

Figure 7. Messages when it doesn't change.

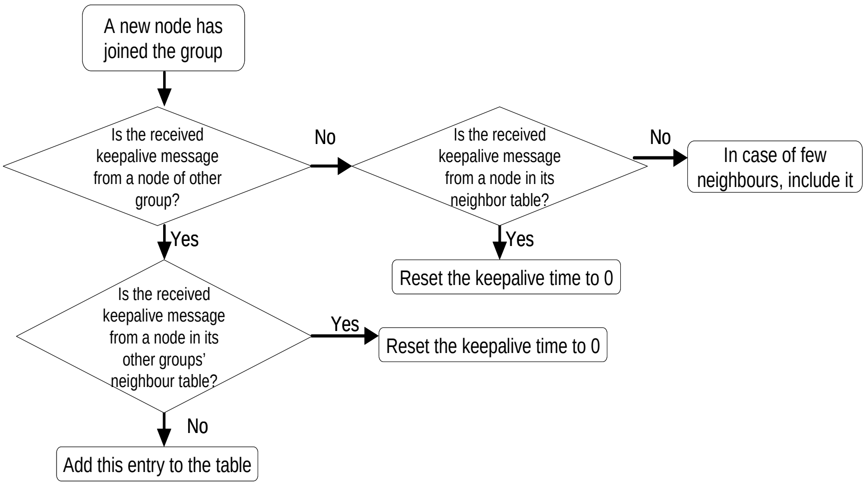
A new node has joined the group
Is the received keepalive message from a node of other group?
No
Is the received keepalive message from a node in its neighbor table?
No
In case of few neighbours, include it
Yes
Yes
Reset the keepalive time to 0
Is the received keepalive message from a node in its other groups' neighbour table?
Yes
Reset the keepalive time to 0
No
Add this entry to the table

Figure 8. Flowchart of the operation to establish links with nodes from other groups.

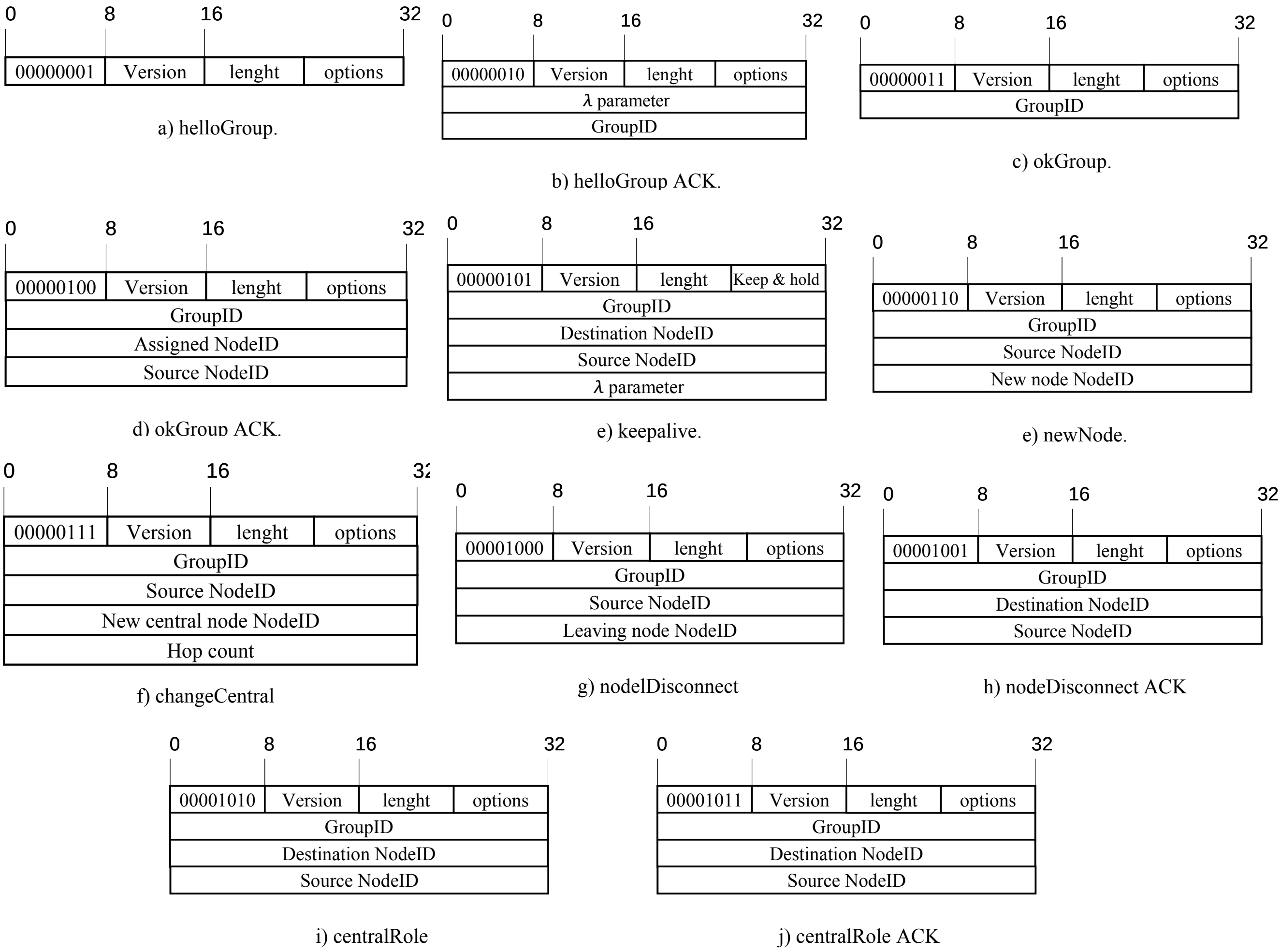

Figure 9. Messages designed

E. Bandwidth consumtion

In order to know the bandwidth consumption of the protocols, the number of bits for every message have been calculated when TCP/IP headers are used. The sum of these headers is 58 bytes. Figure 10 shows the bandwidth for all messages. The messages consume very few bandwidth, so our proposal is a feasible option to enhance the network. The messages that consume more bandwidth are the *keepalive* messages and *changeCentral* messages (218 Bytes taking into account TCP/IP headers). The second ones have been *okGroup ACK*, *newNode*, *nodeDisconnect*, *centralRole* and *centralRole ACK*. The one which consumes less bandwidth has been *helloGroup*.

VI. Analysis

Let T_i be as the time needed by two nodes to communicate each other, and RTT as the mean value of the round trip time between both nodes. So, T_i can be calculated using the expression given in table II. The time needed to communicate a source node with a destination node in a different group is calculated using the expression given for $T_{max\_intergroup}$ in table II. Where n is the number of intermediate groups, t_{source_border} is the time needed to arrive from the source node to the border node in the same group, $t_{max\_intragroup\_i}$ is the time required to go through the i-th group, and $t_{border\_i\text{-}border\_i+1}$ is the time needed to transmit the information from the border node of a group to the border node of another group connected to the previous one.

t_p is defined as the average propagation time for all the message transmissions between two nodes in the architecture. Its expression is shown in table II. Where m represents the number of nodes involved in the path minus one. Taking into account t_p, the time needed to transmit information from the source node to the border node of the same group (T_{source_border}) is defined as it is shown in table II. Where d_{source_border} are the number of hops needed to arrive form the source node to the border node of the same group. The maximum time to cross a group ($T_{max\_intragroup\_i}$) is defined by the expression shown in table II. i indicates the group and the d_i is the number of hops in the group. On the other hand, the number of hops for j groups is shown in table II.

Replacing equations in Table II, equation 4 is obtained.

$$T_{\max\_intergroup} = \left(d_{source\_border} + \sum_{i=1}^{n} d_i + n + 1\right)\cdot t_p \tag{4}$$

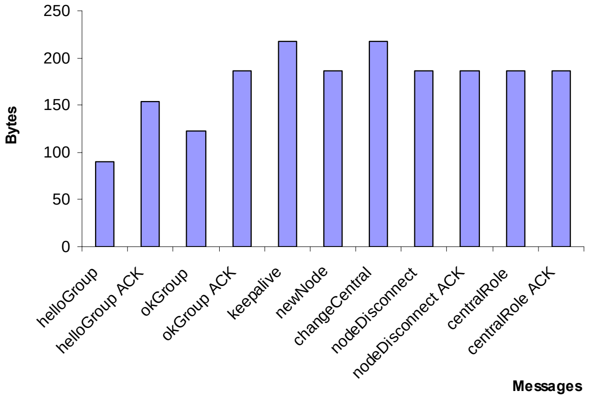

Figure 10. Messages bandwidth.

TABLE II.
EQUATIONS FOR PARAMETERS BETWEEN DIFFERENT NODES

Parameter	Equation
T_i	$\frac{RTT_i}{2}$
$T_{max\_intergroup}$	$t_{source\_border} + \sum_{i=1}^{n} t_{\max\_intragroup\_i} + \sum_{i=1}^{n+1} t_{border\_i-border\_i+1}$
t_p	$\frac{\sum_{i=1}^{m} T_i}{m}$
T_{source_border}	$d_{source\_border} \cdot t_p$
$T_{max\_intragroup\_i}$	$d_i \cdot t_p$
d_i	$\sum_{j=1}^{d_j} d_j$

A. Time variation as a function of the number of hops to the border node when all the groups have the same number of hops

This simulation is done fixing the number of intermediate groups and the number of hops between source node and the border node of its group is varied. Then, it is observed what happens when the number of hops of the intermediate groups increases.

The number of intermediate groups have been fixed as 4. Considering that all the intermediate groups have the same number of hops, it means $d_1=d_2=d_3=d_4=d$, and introducing these values in equation 4, expression 5 is obtained.

$$T_{\max\_intergroup} = \left(d_{source\_border} + 4 \cdot d + 5\right) t_p \qquad (5)$$

When higher values to d_{source_border} for each value of d are given, the maximum inter group time ($T_{max\_intergroup}$) rises lineally.

B. Time variation when the number of hops to cross the groups varies

This section studies what happens when the distance between source node and the border node of the source group is constant and the number of hops of the intermediate groups and for different number of groups vary. The parameter d_{source_border} is fixed to a value of 10. Using equation 4, equation 6 is obtained.

$$T_{\max\_intergroup} = \left(11 + \sum_{i=1}^{n} d_i + n\right) \cdot t_p \qquad (6)$$

Now, d_i is varied in order to observe the time needed to achieve the destination. Results are shown in figure 11. Here, it is deduced that the number of groups in a network doesn't affect the connection time to a large extent when the mean number of hops to go through the groups is low. Nevertheless, when the mean diameter of the groups is big, when the number of intermediate groups is increased, the connection time rises too much. So, the mean diameter of the groups becomes more relevant in the calculation of the final connection time ($T_{max\_intergroup}$) for bigger networks.

In figure 12, how the connection time varies according to the number of groups for different number of hops can be observed. d_{source_border} has been chosen equal to 20, and the number of groups that will be crossed for different mean diameters of the groups have been varied, instead of varying the mean diameter of the groups.

C. Variation of the time for different number of groups and different distances between source and border nodes in the same group

In this section it is analyzed how the maximum inter group time varies when the mean diameter of the group is constant and when the number of groups for different distances between source and border nodes of the same group vary. In order to perform this experiment, the mean diameter of the groups has been equal to 20. Equation 7 shows that the connection time depends on the distance between the source and border nodes in the same group and on the amount of groups in the network.

$$T_{\max\_intergroup} = \left(d_{source\_border} + 21 \cdot n + 1\right) t_p \qquad (7)$$

Figure 13 shows the behavior of the $T_{max\_intergroup}$ as a function of n for several d_{source_border} values. On the other hand, it is observed in figure 14 that the maximum inter group time (t_p) increases when the number of intermediate groups increases. It is happening in all the analyzed cases. Nevertheless, it can be seen that there is not a big difference in the final time when there is large or short distance between source and border node (d_{source_border}). It means, it is more relevant the number of groups than d_{source_border} for having better $T_{max\_intergroup}$. This is an important subject to take into account when designing node networks.

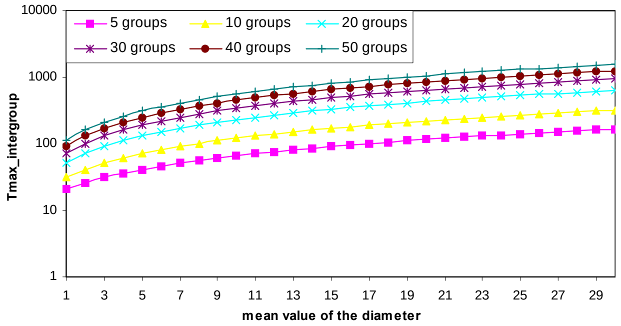

Figure 11. Tmax_intergroup variation according to the mean diameter of the groups.

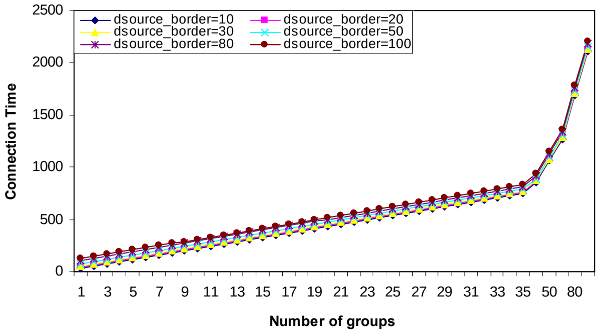

Figure 13. Variation of the connection time according to the number of groups.

TABLE II.
DIAMETERS OF SOME TOPOLOGIES THAT CAN BE USED IN THE GROUP

Topology	Diameter
Tree	$d = 2 \cdot \log_M [n \cdot (M-1)+1] - 2$
2D Grid	$d = 2 \cdot (\sqrt{n} - 1)$
3D Grid	$d = 3 \cdot (\sqrt[3]{n} - 1)$
Zipf Law	$d \approx \log(\log(n))$
Logarithmic Law	$l_n \approx d \approx \ln\left(\frac{n}{2}\right)$
P2P partially decentralized	$d = \frac{n}{16}$

D. Number of nodes as a function of the topology of the group

In order to limit the coverage of the group, let's suppose there are a maximum number of hops between the central node and the most distant border node, using the shortest path. In this option, the geographic coverage of the nodes in a group will depend on the amount of nodes in some specific part of the group topology and on its position. The influence area is defined as all nodes near the central node, with an access time lower than T_{max}.

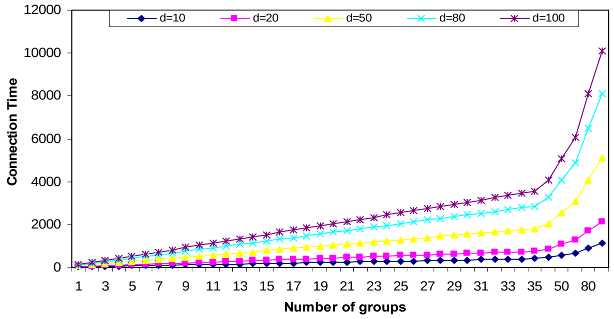

Figure 12. Variation of the connection time according to the number of groups.

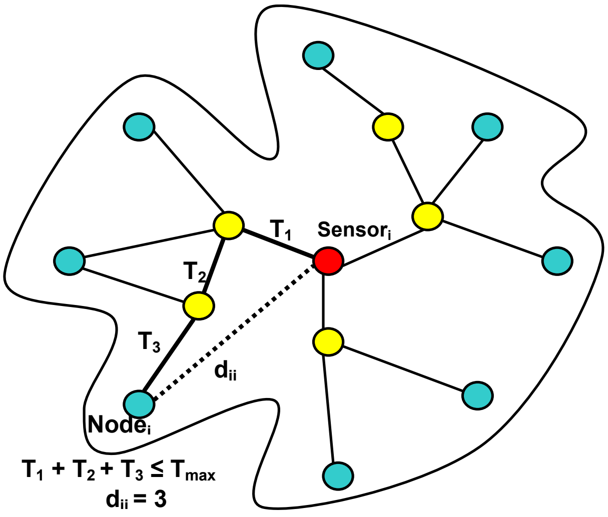

Figure 14. Central node influence area.

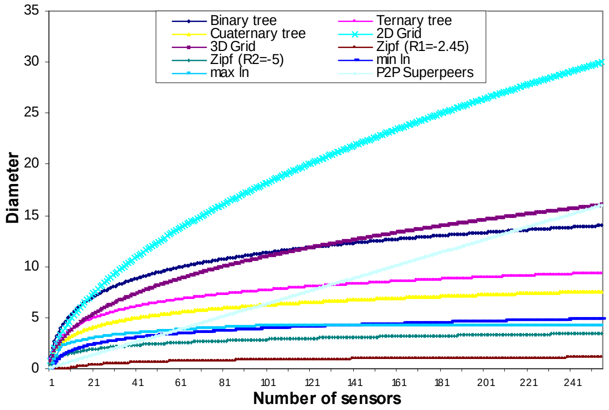

Figure 15. Diameter according to the total number of nodes.

It is supposed that the nodes are in a distance of d_{ij}. It can be observed in figure 14, where T_{max} is specified according to the network and the distance d_{ij}, in number of hops. It will be the minimum distance between both nodes.

First, a maximum diameter of 30 hops has been taken. This indicates that there will have a maximum of 14 nodes between the central node and any border node. The maximum distance can vary depending on what is wanted: larger or smaller groups, but it will depend on the needs of the project in which the protocol is applied. When an event is noticed by a node of the group, it will be forwarded immediately to the entire group. So, a commitment between the number of nodes in the group and the number of groups needed to cover the entire node network, in order to minimize the convergence time, have to be achieved. The diameter depends on the topology of the nodes in the group.

In a tree topology, the number of links is n-1, where n is the number of nodes in the tree. The diameter of the tree topology is shown in table III. *M* is the number of son nodes of a node in a tree topology (2 if binary, 3 ternary, etc.). The diameters of a 2D and of a 3D grid topology, with *n* nodes, are shown in table III. If the network scales freely, without control, following the Zipf law, when the maximum distance (d_{max}) have to be chosen, the cut-off effect shown by R. Cohen et al. in [56] [57] can be used. In networks where routing algorithms cross the network in few hops (such as Internet), *d* is defined as it is in table III. On the other hand, the logarithmic law states that the mean distance between two border nodes is equivalent to the diameter according to the law described in [58]. It can be calculated as it is in table III. Finally, there is a law that follows P2P partially decentralized networks. It can be considered a TTL of 6, and according to a mean of 16 superpeer neighbors in every hop, there will be 96 border nodes for every central node. So, the diameter of the network can be calculated as it is shown in table III. Figure 15 shows the comparative between the studied topologies. The lower diameter is Zipf law with a R coefficient of -2,45. Nevertheless, when there are less than 14 nodes in the network, the worst case is the binary tree topology. Between 15 and 22 nodes, both binary tree and 2D grid topologies are the worst ones, and when there are more than 23 nodes, the worst case is the 2D grid topology.

VII. Conclusions

A group-based architecture provides some benefits for the whole network such as the content availability is increased because it could be replicated to other groups, it provides fault tolerance because other groups could carry out tasks from a failed group and it is very scalable because a new group could be added to the system easily. On the other hand, a group-based network can significantly decrease the communication cost between end-hosts by ensuring that a message reaches its destination with small overhead and highly efficient forwarding. Grouping nodes increases the productivity and the performance of the network with low overhead and low extra network traffic.

In this paper, the proposal of a group-based architecture where links between groups could be established by physical proximity plus the neighbor node capacity, has been shown. Its operation, maintenance and fault tolerance have been detailed. Messages designed to work properly have been shown. All simulations show its viability and how it could be designed to improve its performance. Finally, the designed messages, and the bandwidth they will waste in the network, have been also shown.

The architecture proposed could be used for specific cases or environments such when it is wanted to setup a network where groups appear and join the network or by networks that are wanted to be split into smaller zones to support a large number of nodes. There are many application areas for this proposal such as rural and agricultural environments or even for military purposes. Now, we are programming the protocol for a specific wireless device to test it over a real environment using the OLSR routing protocol.

References

[1] L.M. Feeney, A taxonomy for routing protocols in mobile ad hoc networks, October 1999, SICS Technical Report T99:07.

[2] J. N. Al-Karaki, A. E. Kamal, Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications Volume 11, Issue 6, pp. 6-28, Dec. 2004.

[3] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’99), Seattle, WA, August 1999.

[4] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.

[5] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.

[6] F. Ye, et al., "A Scalable Solution to Minimum Cost Forwarding in Large Scale Sensor Networks", in the Proceedings of International Conference on Computer Communications and Networks(ICCCN), Dallas, TX, October 2001.

[7] C. Schurgers and M.B. Srivastava, “Energy efficient routing in wireless sensor networks,” MILCOM Proc. on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.

[8] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks," The International Journal of High Performance Computing Applications, Vol. 16, No. 3, August 2002.

[9] Y. Yao and J. Gehrke, “The cougar approach to in-network query processing in sensor networks”, in SIGMOD Record, September 2002.

[10] N. Sadagopan et al., “The ACQUIRE mechanism for efficient querying in sensor networks,” in the Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska, May 2003.

[11] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks". Proceeding of the Hawaii International Conference System Sciences, Hawaii, January 2000.

[12] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems," in the Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.

[13] A. Manjeshwar and D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks”. Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[14] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks”. Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing, Ft. Lauderdale, FL, April 2002.

[15] V. Rodoplu and T.H. Ming, "Minimum energy mobile wireless networks," IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, pp. 1333-1344, 1999.

[16] F. Ye et al., “A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks”, in the Proceedings of Mobicom’02, Atlanta, GA, Sep. 2002

[17] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing", in the Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’01), Rome, Italy, July 2001.

[18] Y. Yu, D. Estrin, and R. Govindan, “Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks,” UCLA Computer Science Department Technical Report-01-0023, May 2001.

[19] F. Kuhn, R. Wattenhofer, and A. Zollinger, “Worst-case optimal and average-case efficient geometric ad-hoc routing,” Proc. the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 267–278, Annapolis, MD, Jun. 2003.

[20] Shahab Kamali and Jaroslav Opatrny. A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks. Journal of Networks, Volume: 3 Issue : 4. April 2008. Pp 31-41.

[21] Maria Calle and Joseph Kabara, MAC Protocols for GSP in Wireless Sensor Networks, Journal of Networks, Volume: 3 Issue: 6 June 2008. Pp. 29-35

[22] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", in Wireless Networks, Vol. 8, No. 5, pp. pp: 481-494, 2002.

[23] K. Sohrabi, et al., "Protocols for self-organization of a wireless sensor network,” IEEE Personal Communications, Vol. 7, No. 5, pp. 16-27, October 2000.

[24] T. He et al., “SPEED: A stateless protocol for real-time communication in sensor networks,” in the Proceedings of International Conference on Distributed Computing Systems, Providence, RI, May 2003.

[25] M. Frodigh, P. Johansson, P. Larsson. "Wireless ad hoc networking. The art of networking without a network". Ericsson Review, No. 4, 2000.

[26] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. “A survey on sensor net-works”. Communications Magazine, IEEE. Volume 40, Issue 8. Pp. 102- 114. Aug 2002.

[27] S. Kumar, V. S. Raghavan, J. Deng. "Medium Access Control protocols for ad hoc wireless networks: A survey". Ad Hoc Networks. Vol. 4, Issue 3, Pp. 326-358. May 2006.

[28] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications, April 1999, pp. 46-55.

[29] R. Rajaraman, “Topology control and routing in ad hoc networks: a survey”. ACM SIGACT News. Volume 33, Issue 2, Pp. 60-73. June 2002.

[30] T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol". RFC 3626. Oct. 2003.

[31] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing". RFC 3561. July, 2003.

[32] D. Johnson, Y. Hu, D. Maltz. "The Dynamic Source Routing Protocol (DSR) for Mobile Ad-hoc Networks for IPv4". RFC 4728. February, 2007.

[33] V. Park, S. Corson: Temporally-Ordered Routing Algorithm (TORA) Version 1, Functional Specification, Internet Draft, IETF MANET Working Group, June 2001

[34] Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia 6 (2) (2004).

[35] Wierzbicki, A., Strzelecki, R., Swierczewski, D. and Znojek, M. Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications, in: Second IEEE International Conference on Peer-to-Peer Computing (P2P2002), Linöping, Sweden, 2002.

[36] M. Jiang, J. Li, Y.C. Tay. “Cluster Based Routing Protocol (CBRP)”. Internet-draf, draft-ietf-manet-cbrp-spec-01.txt, National Uni-versity of Singapore, 14 August 1999.

[37] Chiang, C.C., Wu, H.K., Liu, W., Gerla, M., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", The Next Millennium, the IEEE SICON, April 14-17, 1997, Singapore, National University of Singapore, Kent Ridge (Singapore).

[38] Zygmunt J. Hass., Marc R. Pearlman, Prince Samar. “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, IETF, Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[39] Z. J. Haas and M. R. Pearlman, “The zone routing protocol: A hybrid framework for routing in ad hoc networks,” in Ad Hoc Networks, C. E. Perkins, Ed. Reading, MA: Addison-Wesley, 2000.

[40] Mirco Musolesi, Cecilia Mascolo. “A Community Based Mobility Model for Ad Hoc Network Re-search”, Second International Workshop on Mul-ti-hop Ad hoc Networks: from theory to reality REALMAN 2006, 26 May 2006, Florence (Italy).

[41] P. F. Tsuchiya, “The Landmark Hierarchy: a new hierarchy for routing in very large networks,” Computer Communication Review, vol.18 (1988), issue.4, pp. 35-42.

[42] Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4)”, RFC 1771, March 1995.

[43] G. Pei, M. Gerla and X. Hong, “LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility”, in Proceed-ings of IEEE/ACM MobiHOC, August 2000, pp. 11-18, Boston, MA (USA).

[44] Jorge Sá Silva, Tiago Camilo, Pedro Pinto, Ricardo Ruivo, André Rodrigues, Filipa Gaudêncio, and Fernando Boavida. Multicast and IP Multicast Support in Wireless Sensor Networks. Journal of Networks, Volume: 3 Issue: 3 March 2008. Pp. 19-26

[45] S.G.M. Koo, K. Kannan, C.S.G. Lee, A genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks. Proceedings of the 13th IEEE International Conference on Computer Communications and Networks, ICCCN’04, Chicago, IL, October 2004, pp. 469–474.

[46] Simon G.M. Koo, Karthik Kannan and C.S. George Lee, On neighbor-selection strategy in hybrid peer-to-peer networks, Future Generation Computer Systems Volume 22, Issue 7, August 2006, Pages 732-741.

[47] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A Scalable Content-Adressable Network, ACM Sigcomm 2001

[48] I. Stoica, R. Morris, D.Karger, F.Kaashoek, H. Balakrishnan, Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications, ACM Sigcomm 2001

[49] A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), heidelberg, Germany, pages 329-350, Noviembre, 2001

[50] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, J.D. Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, Vol. 22, Issue 1. Pp 41- 53, 2004

[51] R. A. Ferreira, S. Jagannathan, A. Grama. Locality in structured peer-to-peer networks, Journal of Parallel and Distributed Computing, Vol. 66, Issue 2. Pp. 257-273. Feb. 2006.

[52] Olaf Landsiedel, Katharina Anna Lehmann, Klaus Wehrle, T-DHT: Topology-Based Distributed Hash Tables. Fifth IEEE International Conference on Peer-to-Peer Computing. Pp. 143-144. 2005.

[53] M. Castro, P. Druschel, Y. C. Hu and A. Rowstron, Proximity neighbor selection in tree-based structured peer-to-peer overlays, Technical Report MSR-TR-2003-52, Microsoft Research, Microsoft Corporation. 2003.

[54] Xu, Z., Tang, C., Zhang, Z. Building topology-aware overlays using global soft-state. Proceedings of the 23rd Int'l Conference on Distributed Computing Systems. May 2003.

[55] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. Proceedings of ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, USA, August 1999.

[56] Karp, B., Kung, H.T. GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking. Pp. 243–254. New York, USA, 2000.

[57] R. Cohen and S. Havlin. Ultra small world in scale free graphs. Physical Review Letters, 90:058701, 2003.

[58] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdowns. Physical Review Letters, 85:4626–4628, 2000.

[59] Gyorgy Hermann. Mathematical investigations in network properties. Proceedings IEEE Intelligent Engineering Systems, 2005. INES '05. Pp 79-82. September 16-19, 2005.

Jaime Lloret was born in Villajoyosa, Alicante (Spain) November 18, 1972. He received his M.Sc. in Physics in 1997 at University of Valencia and a postgraduate Master in Corporative networks and Systems Integration in 1999. He received his M.Sc. in Electronic Engineering in 2003 at University of Valencia and his Ph.D. in telecommunication engineering at the Polytechnic University of Valencia in 2006. He obtained the first place given by the Spanish Agency for Quality Assessment and Accreditation for the Campus of Excellence in the New Technologies and Applied Sciences area and he was awarded the prize of the best doctoral Student in the Telecommunications area in 2006 according to the Social Council of the Polytechnic University of Valencia.

He is a Cisco Certified Network Professional Instructor. He worked as a network designer and administrator in several companies. He is Associate Professor in Polytechnic University of Valencia (Spain). He has been the editor of several conference proceedings and an associated editor and Guest editor of several international journals. Until 2008, he had more than 55 scientific papers published in national and international conferences, he had more than 25 educational papers and he had more than 25 papers published in international journals (several of them with Journal Citation Report).

Dr. Lloret has been involved in more than 40 Program committees of International conferences and in several Organization and steering committees until 2008. He was the chairman of SENSORCOMM 2007 and UBICOMM 2008 and he is the chairman of ICNS 2009. He is IEEE member, IARIA Fellow member and TCP member of IASTED, WSEAS and IIIS.

Miguel Garcia was born in Benissa, Alicante (Spain) December 29, 1984. He received his M.Sc. in Telecommunications Engineering in 2007 at Polytechnic University of Valencia and a postgraduate "Master en Tecnologías, Sistemas y Redes de Comunicaciones" in 2008. He is currently a Ph.D. student in the Department of Communications of the Polythecnic University of Valencia.

He is a Cisco Certified Network Associate Instructor. He is working as a researcher in IGIC in the Higher Polytechnic School of Gandia, Spain. Until 2008, he had more than 20 scientific papers published in national and international conferences, he had several educational papers and several papers published in international journals (some of them with Journal Citation Report).

Mr. Garcia has been technical committee member in several conferences. He is IEEE graduate student member.

Fernando Boronat was born in Gandia, Valencia (Spain), July 16, 1970. He received his M.Sc. in Telecommunications Engineering in 1994 and his Ph.D. in Telecommunication Engineering in 2004 at Polytechnic University of Valencia and a postgraduate MBA in 1997 in CEREM Business International School.

He worked as R&D engineer in Compañía Telefónica Valenciana, S.L. and in Aeoronáutica y Control, S.L. now, he is Lecturer in Polytechnic University of Valencia (Spain) since 1996. He has several papers (research and educational papers) published in national and international conferences and journals (several of them indexed in SCI JCR): His research interests includes Networks, Multimedia Protocols, Multimedia Synchronization and Sensor Networks.

Dr. Boronat is an IEEE member and is currently involved in several Program Committees of International conferences (EUROMEDIA, SENSORCOMM, UBICOMM...).

Jesús Tomás Gironés received the Master degree in computer science and the Ph.D. degree in pattern recognition and artificial intelligence from the Universidad Politécnica de Valencia (UPV), Valencia, Spain, in 1993 and 2003, respectively.

He joined the academic staff of UPV in 1993. Since 1998, he is an Associated Professor. His current research interests include statistical pattern recognition, machine translation, and text mining. He is a member of the Pattern Recognition and Human Language Technologies Group at the "Instituto de Tecnología Informática". He has participated in several national and international research projects.