A new neighbour selection strategy for group-based wireless sensor networks

Miguel Garcia[1], Diana Bri[2], Fernando Boronat[3], Jaime Lloret[4]
Department of Communications, Polytechnic University of Valencia
[1,2]{migarpi, diabrmo}@ posgrado.upv.es, [3,4]{ fboronat, jlloret}@dcom.upv.es

Abstract

In any type of networks a neighbour selection method is needed to form the topology of the network and to know which node the information has to be sent to reach a destination. Nowadays, several selection strategies exist that are based on different aspects and mainly designed to work in common networks. In this paper we will show our study about those different methods and, then we show the development of a suitable neighbour selection strategy for group-based wireless sensor networks (WSN) that is based on a capacity parameter defined by us and the new neighbour distance. We also present the proposal architecture for WSNs and the protocol when a new node joins a group and has to select its neighbours.

1. Introduction

The number of nodes, the number of connections in the network, the degree of the nodes and the diameter of the network are main parameters used to determine a network topology [1], but there are others, such as the bandwidth, that could be considered.

In order to obtain a network topology, its nodes have to be interconnected, so there has to be a strategy that nodes must follow to choose their neighbours. A node has many ways to choose its neighbours. Each topology, each system and each protocol has the development and the improvement of its neighbour selection as its main goal. Examples given of neighbour discovery mechanisms are reference [2] in pure P2P networks, references [3] and [4] in hybrid P2P networks, reference [5] in unstructured P2P networks, reference [6] in content delivery systems and reference [7] in distributing systems. Both the routing protocols and the interconnection strategy are the main issues taken into account in all type of data networks.

Although grouping nodes give many benefits to the network, we have found very few works about it, but none of them are for WSN and none of them tackles the way of establishing connections and discovering neighbours between nodes from different groups.

The structure of the paper is as follows. The related works are presented in the next Section. Section 3 describes the group-based architecture and the neighbour selection proposal. Finally, in section 4, we conclude the paper giving the benefits of our proposal compared with the other neighbour selection systems.

2. Related works

A neighbour selection algorithm decides which parameters are used, and their values, to select the best neighbour node. This election is given taking into account that it has to accomplish an objective function. It must take also into account how connections are distributed in the network.

Several neighbour selection algorithms exist. Some basic algorithms that depend on the node upstream bandwidth (BW) or on the number of connections (n) are the following ones [2]:

i) Greedy: This algorithm selects the node with the highest value of the neighbour selection parameter (NSP) among all candidates. This parameter is given by expression 1.

$$NSP = \frac{BW}{n+1} \tag{1}$$

ii) Fit: Let b_0 be the downlink bandwidth of the requester. If all neighbouring candidates have a value NSP<b_0, it selects the node with the highest NSP value, otherwise, chooses the one that has minimal positive value of NSP-b_0.

iii) Fastest Link: This algorithm selects the node with highest upstream bandwidth without taking into account the number of connections.

iv) Random: It chooses its neighbour randomly despite its neighbour upstream bandwidth and their number of connections.

Throughout the years more complex types of strategies for neighbour selection have been developed.

2.1. 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 infeasible.

Simon G. M. Koo et al. [3] 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 neighbours. This neighbour selection strategy is being used for BitTorrent P2P network.

2.2. Based on Distributed Hash Tables (DHT) structures

Many systems locate nodes in the topology based on key-based graph structures such as CAN [8], Chord [9], Pastry [10] and Tapestry [11]. 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 neighbours). In DHT structures, neighbours 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 neighbour node that makes the most closest towards resolving the query.

The number of neighbours differs for each network. In Tapestry and Pastry a node has O(log n) neighbours, while in CAN has O(d) neighbours (d is the dimension of the toroidal space). Chord maintains two sets of neighbours. 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 neighbour of a node could be very far (in terms of Round Trip Time, RTT).

2.3. Based on the Internet underlying network

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

Plethora [12] 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 neighbouring 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 [13] 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

neighbours in the coordinate system and the area each neighbour maintains. A node may have links to nodes which are not direct neighbours 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 neighbours 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 neighbours. Next, the neighbours join the distributed hash table and themselves announce their membership to their neighbours. The joining node only needs to contact the node maintaining the corresponding area.

2.4. 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 neighbour 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 [14] a proximity neighbour 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 neighbour 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 n^{th} 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 [15], 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.

2.5. Based on their geographical location

The type of networks where it is more needed to chose the neighbours 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 neighbours' positions. There use to be a location registration and lookup service that maps node addresses to locations. If a node knows its neighbours' positions, the locally optimal choice of next hop is the neighbour geographically closest to the packet's destination. Forwarding in this regime follows successively 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 [16] (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 [17] (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.

Previous mechanisms do not consider peer grouping to structure the network and in none of them connections are established using nodes' capacity.

3. Group-Based Architecture

The proposed architecture is based on a one-level architecture. Each time an event occurs, the update is propagated through the group. Each group is composed by three types of sensors. The central sensor delimits the area of the group (it has the same function than other sensors in the group). The intermediate sensor that is the one responsible for routing the information received from more external sensors to the central sensor of the group (they allow a fast convergence when a change in the network takes place). Finally, the border sensor is the responsible for routing the information from inside its group to other groups or for receiving information from other groups and distributing it inside its own group (they are very important because they give the group's boundary).

A sensor sends information to its group using the Reverse Path Forwarding (RPF) algorithm. This algorithm is also used to send data to the border sensor to reach neighbouring groups. Each group has an RPF database but, when this information has to be sent to a specific group, it is directly routed to the nearest border sensor to this group. When the sensor in the neighbouring group receives the information, it routes it to all the sensors in its group. As the system is based on groups, the information is sent quickly to other groups (through the shortest path from the border sensor). More information about how it works and its application areas can be found in our paper in [18].

Figure 1 shows a 3D vision of the proposed architecture, but just in one group. Border sensors are in light gray, intermediate sensors are in dark gray and the black node is the central sensor.

Nowadays there are numerous types of wireless sensors. Their coverage varies so much. Consequently, we have studied the number of sensors needed to cover a specific area according to their coverage radius. The observed coverage radius in existing networks can be divided in 9 main groups as it is shown in table 1. Figure 2 shows the number of sensors needed for cover different areas according to the coverage radius of the used sensors. It is given for each group in table 1. Despite of the sensor's range, we have defined the area of a group by the diameter of the group, so there will be a maximum number of hops between the most remote sensors. The procedure maintenance is explained later.

3.1. Neighbour selection algorithm

Each sensor has 3 parameters (*sensorID*, *groupID*, λ) that characterize the sensor. Let λ parameter be the sensor capacity that depends on the sensor's upstream and downstream bandwidth (in Kbps), its number of available links (*Available_Con*) and its maximum number of links (*Max_Con*), its % of available load and its energy consumption. It is used to determine the best sensor to connect with. The higher the λ parameter, the best sensor to connect with is. It is defined by equation 2.

$$\lambda = \frac{\left(BW_{up} + BW_{down}\right) \cdot Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1 - \frac{E^2}{K_1}} \qquad (2)$$

Where $0 \leq Available\_Con \leq Max\_Con$. L is the available load and E is the energy consumption. L and E values vary from 0 to 100, according to the state of the sensor. An energy consumption of 0 indicates it is fully charged and when it has a value of 100, indicates it is fully discharged. K_1 defines the minimum value of energy remaining in a sensor to be suitable for being selected as a neighbour. K_2 gives λ values different from 0 in case of L=0 or *Available_Con*=0. The root is out of the division because when the sensor is fully discharged, λ parameter has to be 0. We have considered K_2=100 to get λ into desired values. Figure 3 shows λ parameter values when the maximum number of links for a sensor is 16, for a bandwidth value of 2 Mbps, as a function of its available number of links for different available energy values of the sensor. Sensor's load is fixed to 50%. It shows that higher Energy values give higher λ parameter, so the sensor is more likely to be chosen as a neighbour, in case of less available connections for the same energy, the higher with most available connections is chosen.

Figure 4 shows λ parameter values when the maximum number of links for a sensor is 16 and all have the same available number of links (*Available_con*=6) as a function of the sensor available energy for different bandwidth values. Sensor's load is fixed to 80%. It shows that as the Energy is being consumed, λ parameter is being lower, but when it gets the 80% of consumption, it decreases drastically, so the sensor is more likely to be chosen as a neighbour, in case of more available energy. Figure 4 also shows that a sensor with higher bandwidth is preferred.

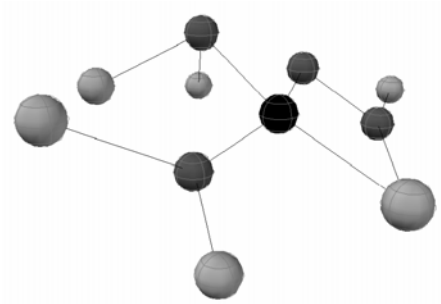

Figure 1. Proposed architecture topology example.

Group	Covarage radius, in meters
Group 1	1
Group 2	2
Group 3	5
Group 4	10
Group 5	20
Group 6	50
Group 7	100
Group 8	200
Group 9	500

Table 1. Groups depending on their coverage area.

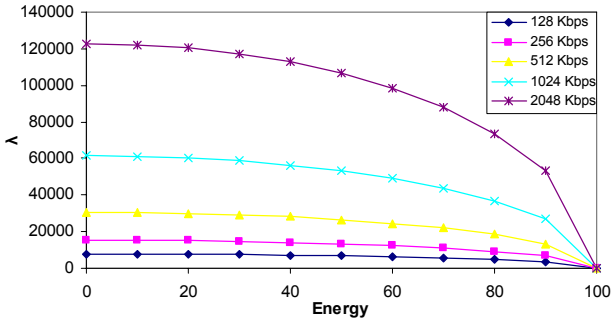

Figure 4. λ values as a function of the Energy of the sensor.

3.2. Protocol

When a new sensor joins the sensor network, it sends a discovery message (called *helloGroup* message) in order to join a group. If there is no response from any sensor for a pre-established time interval (in our case is 3 seconds), the sensor considers itself as a central sensor of a group in the network, and it will take the value *groupID*=1 and *sensorID*=1 (later, if existing sensor groups are joined, they will arrange a new *sensorID* for the group with less number of sensors). When the sensor receives *helloGroup ACK* messages from several candidate neighbours (they could be from different groups), it takes the RTT from each of them and puts a time stamp in their reply. Sensors which are in a distance of 15 hops to the central node will no reply. Then, the new sensor chooses the best sensor from the same group to have a

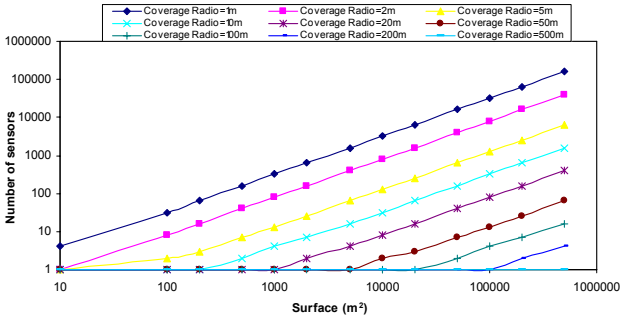

Figure 2. Number of sensor according to the coverage radius and the area to cover.

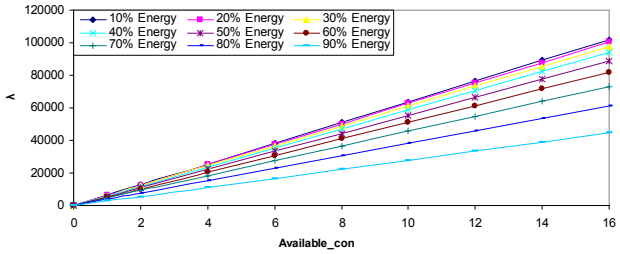

Figure 3. λ parameter values as a function of the available connections of the sensor.

link with (this election is taken according to the λ parameter which is included in the *helloGroup ACK* message and to the RTT). As the *groupID* is in the *helloGroup ACK* message, the new sensor will know which group it has joined (the one with lowest RTT) and it will store this value. If the sensor which sent the *helloGroup ACK* message doesn't receive a reply (*okGroup* message) in a limited period of time, it means that it has not been elected as a neighbour. Finally, new neighbours will reply with the *okGroup ACK* message with the assigned *sensorID* and indicating the link has been established. Sensors will send *keepalive* messages periodically to their neighbours. If a sensor does not receive a *keepalive* message from a neighbour for a dead time, it will remove this entry from its database and will start the group update process. All the process described is shown in figure 5. New sensors and failures in the sensor network have to be known by all the nodes, so, between all the elected neighbours there will be one, called responsible, that will send the information about the new node, using *newSensor* messages, to the central node. Then, central sensor could be changed because the central reference of the group has been moved (this decision is taken using the value of the diameter of the group). When it occurs, the sensors in the group will be advised by a *changeCentral* message. To provide fault tolerance for the central sensor, it calculates which is the best candidate and sends it *keepaliveCentral* messages periodically. In case of changes, updates are distributed using RPF algorithm.

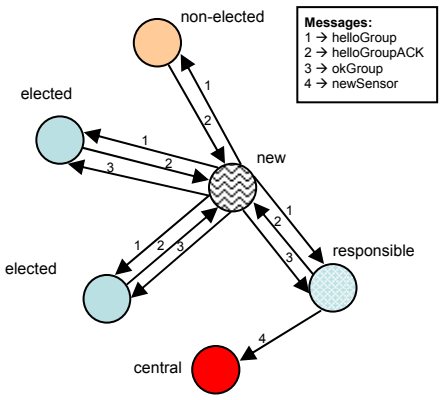

Figure 5. Message Exchange when a node joins the group.

4. Comparative and conclusions

Neighbour selection has been a hot topic for several years in many research topics, but there has not been any research where the nodes, which have to select its neighbours, are placed in different groups. We have shown a state of the art of the main existing neighbour selection algorithms. Some are based on search techniques, there are others based on DHTs, others are based on GPS and, finally, several are based on RTT which give fastest replies. Our proposal is based on RTT and on a λ parameter which combines several parameters such as bandwidth, load, energy, available number of connections and on the maximum number of connections of the sensor. It gives us fastest replies while introduces major parameters when the neighbour selection decision takes place. The number of node's neighbours could be limited only by establishing a maximum in all previous works, whereas in our proposal as the number of connections of a node gets higher, it is less probable to be chosen as a neighbour. In our system, λ parameter allows to distribute the load of the network between groups while it balances the number of sensors in the groups, distributing them uniformly in all the groups of the network.

5. References

[1] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger, "Network Topology Generators: Degree-Based vs. Structural", Proceedings of the ACM SIGCOMM, August 2002.

[2] L. Zou, E. Zegura, M.H. Ammar, The effect of peer selection and buffering strategies on the performance of peer-to-peer file sharing systems. Proceedings of 10th Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecom. Systems, Fort Worth, TX, 2002.

[3] 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.

[4] 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.

[5] D.S. Bernstein, Z. Feng, B.N. Levine, S. Zilberstein, Adaptive peer selection. Proc. of the 2nd Int'l Workshop on Peer-to-Peer Systems, Berkeley, CA, Feb. 2003.

[6] J. Byers, J. Considine, M. Mitzenmacher, S. Rost, Informed content delivery across adaptive overlay networks. ACM SIGCOMM 2002, Pittsburgh, PA, August 2002.

[7] S.G.M. Koo, C.S.G. Lee, K. Kannan, A resource-trading mechanism for efficient distribution of large-volume contents on peer-to-peer networks. Proceedings of the 14th IEEE International Conference on Computer Communications and Networks, pp. 428–433. San Diego, CA, October 2005.

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

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

[10] 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

[11] 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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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,

[18] J. Lloret, M. Garcia and J. Tomas, A Group-Based Architecture for Wireless Sensor Networks, Int'l Conf. on Networking and Services, Athens (Greece), June 2007.