W-GRID: A P2P SELF-ORGANIZING INFRASTRUCTURE FOR ROUTING IN AD-HOC WIRELESS NETWORKS

Gabriele Monti, Gianluca Moro

Dept. of Electronic, Computer Science and Systems

University of Bologna

[gmonti, gmoro]@deis.unibo.it

Large scale wireless ad-hoc networks of computers, sensors, PDAs etc. (i.e. nodes) are revolutionizing connectivity and leading to a paradigm shift from centralized systems to highly distributed systems. A plethora of routing algorithms have been proposed to support path discovery into this class of networks, ranging from broadcasting/flooding approaches to those using global positioning systems (e.g. GPS). W-Grid is a novel infrastructure whose self-organizing characteristic enables message routing among participants without reliance on flooding/broadcasting operations or GPS. The resulting ad-hoc network does not suffer from the dead-end problem, which occurs in geographic-based routing when a node is unable to locate a neighbor closer to the destination than itself.

Extensive performance analysis and experiments have been conducted, and the results compared to GPSR, which is considered to be the most efficient solution not using broadcast operations but exploiting nodes' position. Analyses show that the difference between the performance of our approach and GPSR is minimal and therefore very interesting, since no GPS is used. In this paper we discuss the case of wireless ad-hoc networks where nodes are not inherently mobile but they may disconnect from the network (e.g. failures)

1. Introduction

Recent advances in information communication technology have led to the rapid development of small, powerful, multi-function devices with multi standard radio interfaces including Bluetooth, Wi-Fi and Wi-Max. For example, ad-hoc networks are being designed where devices/nodes can directly communicate within a limited space both indoor, such as a building, and outdoor, such as a metropolitan area, without the need of a fixed pre-configured infrastructure and rigid data/communication protocols. Connectivity in this environment is supported by multi-hop transmission, meaning that intermediate nodes act as signal repeaters according to a given routing strategy. A wide number of routing algorithms for ad-hoc networks have been proposed (see section 2), ranging from those that adopt message broadcast/flooding to those using global position systems (GPS) to discover the routing path towards the destination.

This problem has also been addressed in cases of both total absence and partial availability of geographic location information by generating virtual coordinates to approximate the real ones. Our solution may be classified within this set of approaches in that it also uses virtual coordinates; it is distinctive in that it does not aim to approximate real coordinates and does not rely on GPS.

The W-Grid [17] is related to centralized grid files solution such as R-Tree [5], NIBGF [11] (which has recently been extended to work in wired distributed environments) and G-Grid [12]. It generates, in decentralized manner, virtual coordinates for each network device which reflect its local connectivity with others and uses this information to support message routing.

In this paper we present W-Grid for wireless ad-hoc networks that allows any participant to communicate with every other node without knowing both its own physical position and the position of the recipient. This is achieved by using a simple location service that the W-Grid structure itself implicitly supplies. In the paper we explore the case of networks where, though nodes are not inherently mobile, each device can suddenly disconnect (e.g. failures). The work is organized as follows. Section 2 discusses related works and our contribution. In Section 4 we describe the rules for generating the virtual coordinates and we give details about the routing algorithm and the location service. In Section 4 we present the really interesting results (in term of average path length) which have been returned by the simulations we ran in order to compare our approach with GPSR [7]. Section 5 concludes the paper with open issues and perspective works.

2. Related Works

Routing protocols for ad-hoc networks are typically subdivided into three main categories: Table-driven (also known as proactive), On-Demand (or Reactive) and geographic routing protocols.

Table-driven routing protocols [2,10,15] recall the Internet distance-vector and link-state protocols. Nodes maintain tables that store routing information and any change in network topology triggers propagating updates in order to maintain a consistent view. This can cause heavy overhead affecting bandwidth utilization, throughput and power usage. The advantage of these protocols is that routes to any destination are always available without the latency of a route discovery, but on the other side, they do not perform properly when the number of participating node is high. The main differences in protocols belonging to this category are on the number of tables that nodes store and how they are updated.

On-demand routing protocols [6,14,16] are characterized by a path discovery mechanism that is initiated when a source needs to communicate with a destination that it does not know how to reach. Usually route discovery requires a form of query flood and for this reason on-demand routing incurs in a delay whenever a new path is needed. The differences between the several on-demand protocols are in the implementation of the path discovery mechanism.

A completely different approach is used by geographic routing protocols such as [7,8]. The idea in geographical routing is to use nodes locations as their address, and to forward packets with the goal of reducing as much as possible the physical distance to the destination. Geographic routing achieves good scalability since each node only needs to be aware of neighbors' position and because it does not rely on flooding to explore network topology. However it suffers of dead end problems, especially under low density environment or scenarios with obstacles or holes.

Problems are caused by the inherent greedy nature of the algorithm that can lead to situation in which a packet gets stuck at a local optimal node that appears closer to the destination than any of its known neighbors. In order to solve this flaw, correction methods such as perimeter routing, that tries to exploit the well-known right hand rule, have been implemented. However, some packet losses still remain and furthermore using perimeter routing causes loss of efficiency in terms of average path length.

Another limitation of geographic routing is that it needs nodes to know their physical position. Usually authors assume that they embed GPS but it must be said that GPS receivers are expensive and energy inefficient compared to the devices that could participate in ad-hoc networks. Besides, GPS reception might be obstructed by climatic conditions and doesn't work indoor.

Recently, virtual coordinates were proposed to exploit the advantages of geographic routing in absence of location information [1,3,9]. The motivation is that in many applications it is not necessary to know the exact coordinates but is often sufficient to have virtual coordinates that approximate real ones. Unfortunately virtual coordinate systems suffer the same dead end problem of standard geographic routing.

W-Grid employs virtual coordinates like these last algorithms but it is based on a different approach which does not approximate real coordinates and therefore eliminates the risk of dead-ends.

3. W-Grid main features

W-Grid virtual coordinates are generated on a one-dimensional space and the devices can be absolutely unaware of their physical location. The main idea is to map nodes on a binary tree so that the resulting coordinate space reflects the underlying connectivity among them. Basically we aim to set parent-child relationships to the nodes which can sense each other, in this way we are always able to route messages, in the worst cases simply following the paths indicated by the tree structure. Using virtual coordinates that do not try to approximate node's geographic position we eliminate any risk of dead-ends.

In a W-Grid network nodes behave like peers, meaning that each one is responsible both for addressing requests of communication (location service) than for routing them. For this reason from now on we will refer to them as nodes or peers indistinctly.

3.1. Virtual Coordinate Selection

Each peer must be assigned one or more virtual coordinates c_i, each one represented by a binary string. Thus, when a node n turns on for the first time it starts a wireless channel scan (beaconing) searching for any existing W-Grid network to join (namely any neighbor device that already holds W-Grid virtual coordinates). If none W-Grid network is discovered n creates a brand new virtual space coordinate and elects itself as root of it by getting the virtual coordinate "*"[1]. On the contrary, if beaconing returns one or more devices which already have a W-Grid coordinate, n will join the existing network by getting one or more virtual coordinates.

[1] It is conventional to label "*" the root node

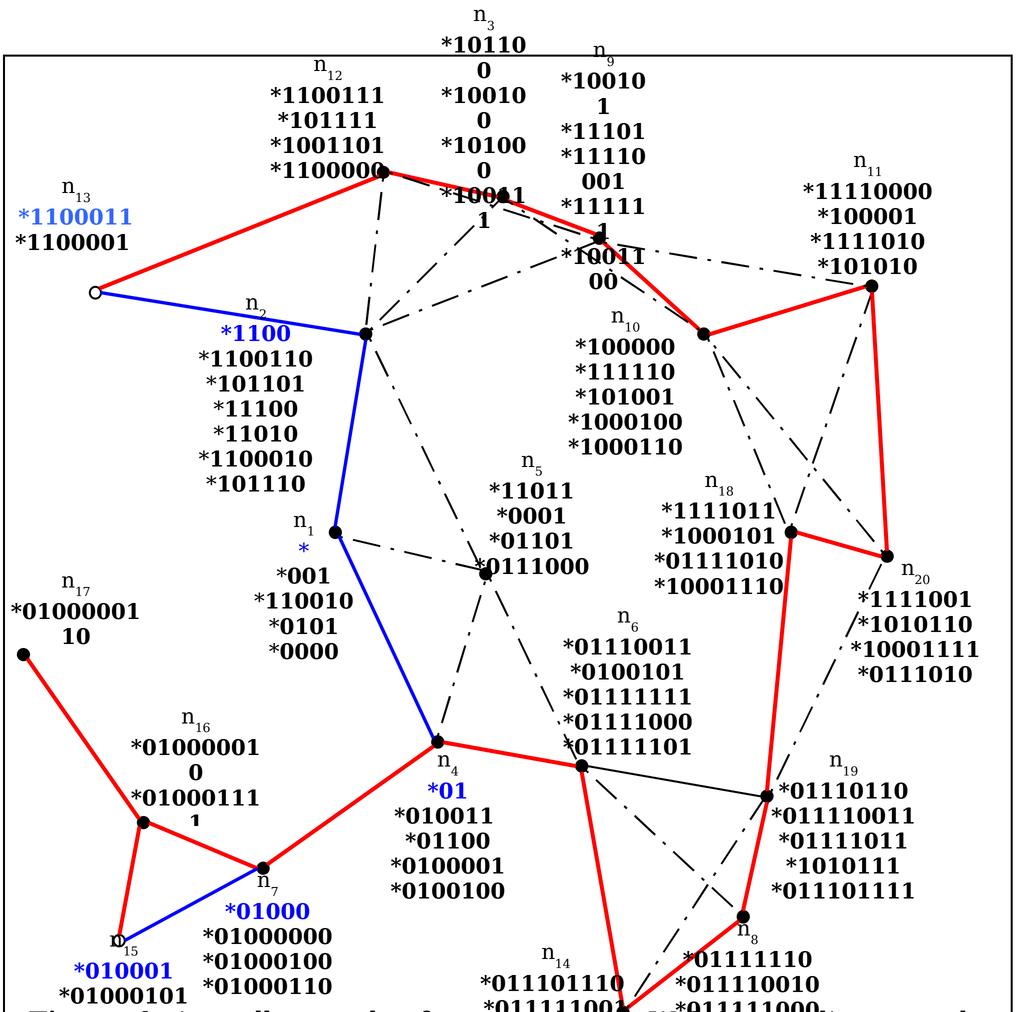

Figure 1. A small example of a network with W-Grid coordinates and routing of a message with W-Grid (blue) and GPSR (red).

Extensive experiments have showed that assigning different coordinates per node improves routing efficiency, in fact having more than one coordinate means that a node is placed in different positions of the tree structure and this has two positive effects on the system.

Firstly, the probability that two nodes physically close have very different virtual coordinates, which may happen when a multi-dimensional space (in which nodes are spread) is mapped into a mono-dimensional space, is highly reduced. Besides, this implies that for each couple of nodes there will be several different paths that allow packet routing, improving network robustness against unexpected failures of nodes (see section 3.3).

The joining operation involves two different tasks: the selection of the coordinate to split[2] (if more neighbors are available) and the split operation.

Coordinate Setup. Splits occur whenever a node needs a W-Grid coordinate. The actors in a split operation are a node n_g with one of its coordinates, let us say c_g and the joining node n_j, we say that c_g is split when adding a bit (a "*0*" or a "*1*") to it. For instance the result of a split to c_g will be $c_{g'} = c_g$ + "*1*" and $c_{g''} = c_g$ + "*0*". After the split one out of the two generated coordinates is assigned to the joining node, while the other one is kept by the giving one that will then hold two different coordinates.

Obviously no more splits can be performed on the original coordinate c_g since this would generate duplicates. In order to avoid that two (or more) nodes get the same coordinate, any choice that the asking node(s) make must be confirmed by the giving node n_g. Thus if two nodes ask for the same coordinate to split only one request will succeed and the other one will be canceled.

Coordinate Selection. During the coordinate setup, if the number of neighbors holding virtual coordinates is more than one, let us say *k*, n_j must choose one node among $n_1, .., n_k$ and ask for a coordinate. The selection strategy we adopt is to choose the shortest coordinate[3] (in terms of number of bits).

If two or more strings have the same lengths the nodes will choose the one that is more distant[4] from all the other candidates. The choice of the shortest possible *c* aims to reduce as much as possible the length of the coordinates in the system. In Figure 1 it is possible to see a small example of a W-Grid network. In order to improve legibility for each node we show only the coordinates that have not be split but actually also split coordinates are stored at nodes, for instance node n_1, which is the node that started the network, has also coordinates ***, **0* and **00*. Through multiple splits of root coordinate * we obtained **001*. The only exceptions are coordinates printed in blue that represent the various intermediate hops toward the destination (n_{13}).

3.2. Routing algorithm

As we stated in the previous subsection, the coordinate creation algorithm of W-Grid generates an order among the nodes (a binary tree). The main benefit of such organization is that messages in the

---

[2] We use the term split to recall the relation of W-Grid with the older NIBGF data structure

[3] Among the ones that still can be split, see Coordinate Setup

[4] According to W-Grid concept of distance explained in section 3.2

worst case travel across the network by following parent-child relationship, guaranteeing the message delivery in any case.

The routing of a message is based on the concept of distance among coordinates. We define the distance between two coordinates c_i and c_j as the sum of the number of bits of c_i and c_j which are not part of their common prefix. For instance:

$$d(*\mathbf{0}011,*\mathbf{0}11) = 5$$

Basically, given a message and a target binary string c_t each node n forwards it to the neighbor that present the shortest distance to c_t according to the described W-Grid metric. In Figure 1 blue and red paths shows a W-Grid and a GPSR packet routing respectively, in this particular case our system perform better than GPSR. This is due to the perimeter mode that GPSR needs to enter when greedy routing fails.

3.3. Nodes Failure

In ad-hoc networks nodes usually have scarce resource and they especially suffer of power constraints. This can lead to nodes failures that could affect routing efficiency. In W-Grid routing robustness is guaranteed by multiple coordinates that peers are given at the time they join the network. By having different coordinates we allow routing through different paths and therefore if a broken path is discovered the packet can change direction (e.g. next hop) and follow another path.

In Figure 2 we present the case of a packet that must be routed to coordinate **11*. During the routing a dead end occurs, node n_5 cannot find any neighbor that improves its distance from the destination. The solution to this problem is that whenever a node discovers a broken path it deletes the coordinates that causes the dead end and than recalculates the distances of its neighbors from the packet destination, at this point there will surely be a neighbor that improves the distance from the destination (even the one that forwarded the packet to the current node). By deleting the coordinate we ensure that the same packet (and the future packets with the same destination) will not be forwarded to it again.

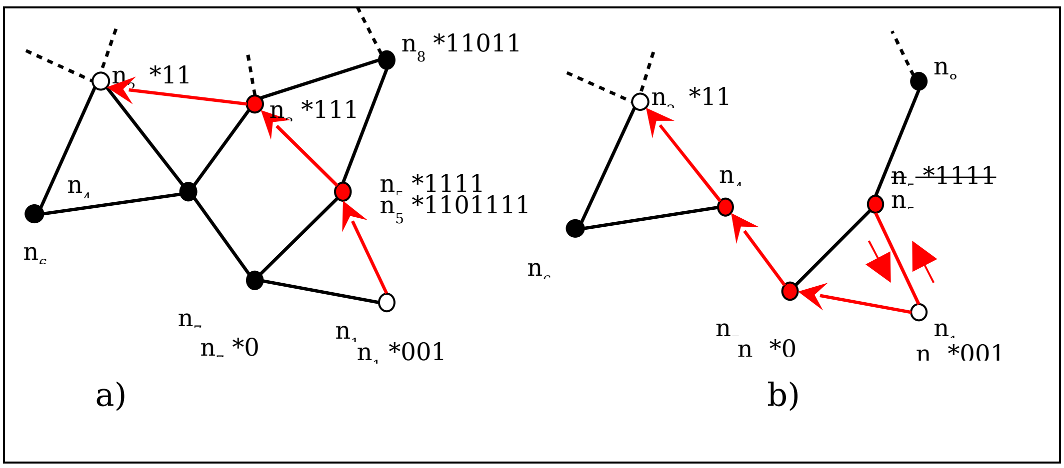

Figure 2. Effect of node failure (n_3) during routing of a packet from node n_1 to n_2

Due to the limited space available we invite readers to refer to W-Grid technical report [4] for details about nodes failure recovery.

3.4. Location service in W-Grid

Supposing that each node *n* that composes the network is univocally identified by a public *ID* (such as the e-mail address, the MAC Address or any other unique value) we can obtain a location service in W-Grid by hashing each *ID* into a binary string (*h(ID)*) and storing information about node's W-Grid coordinates at the peer whose coordinate has the longest common prefix with *h(ID)*.

In this way, a node n_s that need to communicate with another node nr will simply search the network for the $h(ID_r)$ discovering n_r W-Grid virtual coordinates. After this, the message will be delivered to the recipient by the W-Grid routing algorithm.

Examples about defining a domain *Dom(ID)* for the various ID_i and a possible hashing function that allow to translate *ID*s into binary strings of arbitrary length can be found in [4] and [13].

4. Experimental Results

To evaluate the performances of W-Grid algorithm we implemented a Network Simulator in Java. We simulated network deployment upon areas with different dimensions generating nodes in random positions avoiding partitions in the network.

Table 1. Simulation results for different area dimensions (in meters, 50 runs for each configuration)

Simulation area dimensions	APL W-Grid	MSE W-Grid	APL GPSR	MSE GPSR	% dropped packets GPSR	MSE dropped GPSR	Nodes number
800x800	6,13	3,11	7,49	8,45	10,01%	321,14	~120
1000x1000	8,05	4,46	9,02	13,00	9,24%	578,74	~205
1200x1200	9,75	4,48	9,65	12,75	8,64%	224,68	~295
1400x1400	11,54	4,99	10,87	14,52	5,42%	247,45	~400
1600x1600	13,97	5,87	13,72	14,99	7,36%	308,31	~520
1800x1800	14,82	6,41	14,14	12,15	9,17%	335,91	~660
2000x2000	17,44	8,44	15,57	13,20	8,14%	405,15	~820

Then we generated random messages among nodes in order to compare the Average Path Length (APL, measured in hops) covered by messages sent using W-Grid and GPSR.

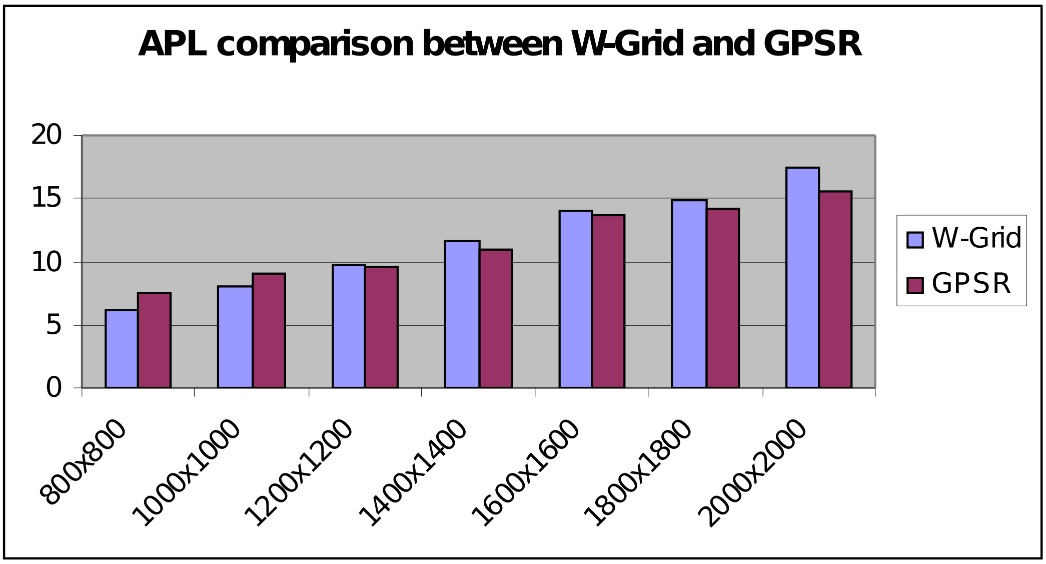

Figure 2. Performances comparison between W-Grid and GPSR.

Coordinate creation is gradual, the simulation randomly chooses one node that beacons first and elects itself as root of a new virtual coordinate space. Then, as described in Section 3 we let that periodic beaconing builds the W-Grid network. Each node gets as many coordinates as the number of its neighbors making the system scalable (the number of neighbors is fixed and independent from the network size) while, as can be seen from Table 1, performances are very good.

Once that every node had got its coordinates the simulator started the generation of 50000 messages between randomly chosen couples of sender/recipient nodes. Each message was routed according to our algorithm, following the virtual coordinates, and at the same time it was routed using GPSR algorithm (exploiting [x,y] physical positions of nodes).

Even if the comparison appears prohibitive, since GPSR can stay very close to the ideal routing algorithm also because it uses physical position of nodes, W-Grid returns amazing performances, especially considering that it doesn't require any kind of information about geographic position of nodes. This means not only a vaster and heterogeneous space of application, not limited only by GPS (or any other position estimation equipment) embedded devices, but also an easier deployment in every condition and everywhere.

Figure 3 shows that the number of hops (APL) is almost the same in W-Grid and GPSR, but if we consider the natural advantage of GPSR that knows physical positions of the nodes we can say that the results are very good since, in some configurations our algorithm presents better performances, due to the perimeter issue of GPSR that can cause longest paths.

Besides, it is important to say that W-Grid doesn't fail any message delivery and it performances are almost the same in the different runs per area (see W-Grid MSE) showing that it is not affected by network topology. On the other side GPSR presents a notable percentage of routing failures and its performances are variable and dependent from nodes positions.

5. Conclusions and future work

In this paper we presented W-Grid, a virtual coordinates system and a routing algorithm that can be employed even when nodes are unaware of their physical location. The main idea is to map nodes on an index structure and make the resulting coordinates space reflecting the underlying connectivity among them. Parent-child relationship are set among peers which can sense each other allowing them to always route requests, in the worst case by simply following the paths indicated by the tree structure.

The system doesn't suffer of the dead end problem since it uses virtual coordinates that don't approximate real ones and since it doesn't use GPS it works both indoor and outdoor.

Future works will concern the introduction of multiple virtual spaces (namely multiple roots) among which nodes can choose at routing time the next hop according to the space which better reduces the distance to the destination. We are also studying the possibility of introducing a path learning capability at nodes in order to improve the W-Grid APL.

6. References

[1] R. Bischoff and R. Wattenhofer. *Analyzing connectivity-based multi-hop ad-hoc positioning*, 2004.
[2] W. L. C. Chiang, H. Wu and M. Gerla. *Routing in clustered multi-hop, mobile wireless networks*. In *Proc. IEEE SICON'97*, pages 197-211, April 1997.
[3] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. *Geographic routing without location information*. In MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking, pages 96-108. ACM Press, 2003.
[4] G. Moro, G. Monti, and A. Ouksel. *W-grid: a p2p self-organizing infrastructure for routing and data management in ad-hoc wireless networks*. Technical Report DEIS-LIA-06-001, University of Bologna (Italy), 2006. LIA Series no. 75.
http://p2p.ingce.unibo.it/»gmoro/papers/DEIS-LIA-06-001.pdf.
[5] A. Guttman. *R-trees: A dynamic index structure for spatial searching*. In B. Yormark, editor, *SIGMOD Conference*, pages 47-57. ACM Press, 1984.
[6] D. B. Johnson, D. A. Maltz, and J. Broch. *DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks*. Ad hoc networking, pages 139-172, 2001.

[7] B. Karp and H. T. Kung. *GPSR: Greedy perimeter stateless routing for wireless networks*. In MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, pages 243-254. ACM Press, 2000.
[8] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. *Geometric ad-hoc routing: of theory and practice*. In PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing, pages 63-72. ACM Press, 2003.
[9] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, *Virtual coordinates for ad hoc and sensor networks*. In Proceedings of 2nd Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), 2004.
[10] S. Murthy and J. J. Garcia-Luna-Aceves. *An efficient routing protocol for wireless networks*. Mobile Networks and Applications, pages 183-197, 1996.
[11] A. Ouksel and O. Mayer. A robust and efficient spatial data structure: *The nested interpolation-based grid file*. Acta Informatica, 353-373, 1992.
[12] A. Ouksel and G. Moro. *G-grid: A class of scalable and self-organizing data structures for multi-dimensional querying and content routing in p2p networks*. In Agents and Peer-to-Peer Computing. Springer-Verlag, 2004.
[13] A. M. Ouksel, G. Moro, and W. Litwin. *GGF: A generalized grid file for distributed environments*. Technical report, DEIS Univ. of Bologna, Univ. of Illinois at Chicago, 2002.
[14] V. D. Park and M. S. Corson. *A highly adaptive distributed routing algorithm for mobile wireless networks*. In INFOCOM(3), pages 1405-413, 1997.
[15] C. Perkins and P. Bhagwat. *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*. In ACM SIGCOMM'94, pages 234-244, 1994.
[16] C. E. Perkins and E. M. Royer. *Ad-hoc on-demand distance vector routing*. In WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, page 90. IEEE Computer Society, 1999.
[17] G. Moro, G. Monti, and A. Ouksel. *W-grid: a p2p self-organizing infrastructure for routing and data management in ad-hoc wireless networks.* In ICPS 2006: International Conference on Pervasive Services, Lyon, France, 26-29 June 2006.