Routing and Localization Services in Self-Organizing Wireless Ad-Hoc and Sensor Networks Using Virtual Coordinates

Gianluca Moro, Gabriele Monti and Aris M. Ouksel

***Abstract*—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 global positioning systems (GPS). W-Grid is a novel infrastructure whose self-organizing routing and localization Services enable message routing and data management 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. As an example of data management we present a location service that allows a device to send messages to any another node without knowing its position. 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. Analysis show that the difference between the performance of our approach and GPSR is minimal and therefore very interesting, since no GPS is used.**

I. 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, where 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 II), 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 generates, in decentralized manner, virtual coordinates for each network device which reflect its local connectivity with other devices and uses this information to support message routing. These virtual coordinates also delineate the data space partition for which a device is assigned management responsibility, meaning that it is possible to distribute across the W-Grid network any kind of data. In order to proof this feature we will give a short description of a location service. Alternatively W-Grid can act as the network layer upon which existing indexing structures can be applied. For instance we think about the ones that were developed in the past for centralized environments (e.g. R-Tree [3] and NIBGF [9], see [2] for an extensive survey) and which have recently been extended to work in distributed environments, especially in wired peer-to-peer networks [8] [12], which need a predefined topology and are therefore used on top of TCP/IP layer.

Due to space limitations in this paper we describe W-Grid for wireless ad-hoc and sensor networks where, though nodes are not inherently mobile, each device may also disconnect from the network (e.g. failures). The work is organized as follows. Section II discusses related works and our contribution. In Section III, we describe the rules for generating the virtual coordinates and the routing algorithm, while data management issues are addressed in Section IV. In Section V 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 [5]. Section VI concludes the paper with open issues and perspective works. Due to space limitation we invite readers to refer to [6] for the extended version of this paper.

II. 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 [10] [1] 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 may cause heavy overhead affecting bandwidth utilization, throughput and

Gianluca Moro and Gabriele Monti are with the Dept. of Electronic, Computer Science and Systems, University of Bologna
Viale Venezia, 52 47023 Cesena, Italy
email: gmoro,gmonti@deis.unibo.it

Aris M. Ouksel is with the Dept. of Information and Decision Sciences, The University of Illinois at Chicago
601 South Morgan, Chicago, USA
email: aris@uic.edu

power usage.

On-demand routing protocols [11] [4] 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.

A completely different approach is used by geographic routing protocols such as [5]. The idea in geographical routing is to use a node's location as its 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 (no flooding is used) but it suffers of dead end problems, especially under low density environment or scenarios with obstacles or holes. Another limitation of geographic routing is that it needs nodes to know their physical position. Usually authors assume to embed GPS receivers that are expensive, energy inefficient and do not work in indoor environments or when weather conditions are not perfectly good. A more detailed description of these protocols can be found in the W-Grid technical report [6].

III. Routing and Virtual Coordinate Space partitioning

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 coordinates space reflects the underlying connectivity among them. Basically we aim to set parent-child relationship to the nodes which can sense each other, in this way we are always able to route requests and 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 n 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.

A. Virtual Coordinate Selection and Space partitioning

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 have already a W-Grid coordinate, n will join the existing network by getting virtual coordinate(s).

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

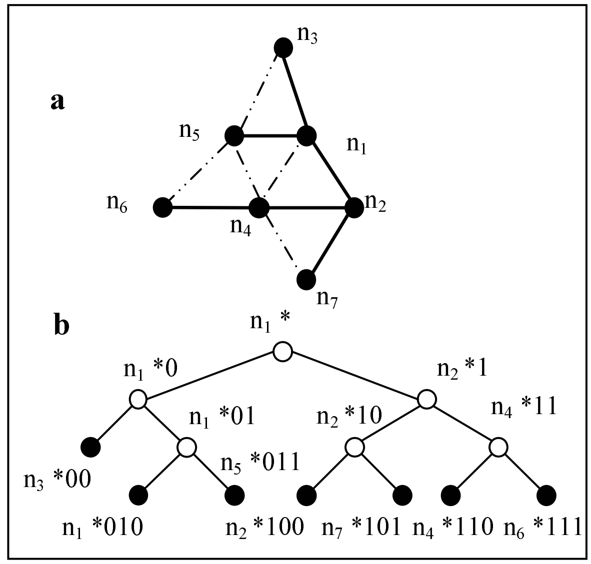

Fig. 1. Physical(a) and logical(b) network. Empty circles represent split coordinates, full black circles are coordinates that can still be split.

Extensive experiments have showed that a higher number of 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. This reduces 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 translated into a mono-dimensional space. The joining operation involves two different tasks: the selection of the coordinate to split (if more neighbors are available) and the split of an existing coordinate.

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 cs is assigned to the joining node, while the other is kept by the giving one that will then hold two cs. Obviously no more splits can be performed on the original coordinate c_g since this would generate duplicates. Surely the term "split" will appear misleading right now, but the reasons why it is called like that will be clear when the data management part of W-Grid will be presented in Section IV.

Coordinate Selection. In the coordination 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 [2](in terms of number of bits) c. If two or more strings have the same lengths the selection is random. In Figure 1 it is possible to see a small example of a W-Grid network. The choice of the shortest possible c allows to reduce as much

[2] among the ones that still can be split, see Coordinate Setup

as possible the length of the coordinates in the system. 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.

B. Routing algorithm

As we stated in the previous subsection, the coordinate creation algorithm of W-Grid generates an order among the nodes and its structure is represented by a binary tree. The main benefit of such organization is that messages in the 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_i forwards it to the neighbor that present the shortest distance to c_t according to the described W-Grid metric.

C. Nodes Failure

In MANETs nodes usually have scarce resource and they especially suffer of power constraints. This may 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.

Due to the limited space available for the description we invite readers to refer to W-Grid technical report [6].

IV. Data Management in W-Grid

W-Grid organizes peers in a tree structure and distributes data (tuple/records with any kind of information) among them by hashing the values of the record attributes into binary strings and storing them at peers whose W-Grid coordinates match the strings.

Since W-Grid c_is are binary strings, we can see from Figure 2 that they correspond to leaf nodes of a binary tree. Therefore a W-Grid network acts directly as a distributed database. Obviously coordinates that have been split(the empty circles in Figures 1 and 2) do not contain any data.

A really innovative aspect of W-Grid, in comparison with [3] [9] [2], is the capability to balance storage load (and consequently routing requests) among nodes.

A. Location service

Supposing that each peer n_i that composes the network is univocally identified by a public ID_i (such as the e-mail address, the MAC Address or any other unique ID) we may think about inserting in the distributed database, implicitly defined by W-Grid, information about peers location (W-Grid coordinates) using as key (both for insertion and search) the peers IDs. In this way, a node (n_s) that need to communicate with another node (n_r) simply searches the network for the ID_r and will discover where n_r can be found. After this, n_s will be able to send a message to the recipient simply using the W-Grid routing algorithm.

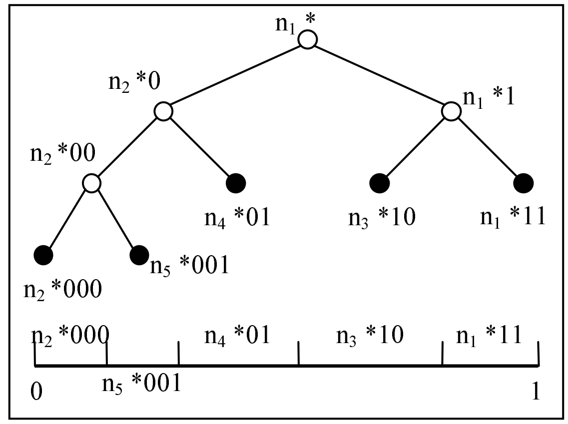

Fig. 2. Correspondence between W-Grid virtual coordinates and data space partitions.

Practically speaking, defining a domain Dom_{ID} for the ID_is we can use a hashing function[3] to translate any ID into a binary string of arbitrary length (see [6] for a more detailed example).

This location service example is just one of the possible data management applications implementable in W-Grid. In fact, it is possible to manage each kind of one-dimensional and multi-dimensional data by translating them into binary string with the use of hashing algorithms.

V. 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. 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.

Nodes perform periodic beaconing (every 300ms) and generate messages at a parameterizable frequency. The beaconing is asynchronous, namely each peer is independent from the others, as it happens in real networks and we supposed a radio transmission range of 100 meters. Coordinate creation is gradual, the simulation randomly choose one node that beacons first and elects itself as root of a new virtual coordinate space. Then, as described in Section III we let that periodic beaconing builds the W-Grid network.

Once that every node had got its c the simulator (50 simulations per each different area) started the genera-

[3] See [7] for details about any possible hashing algorithm

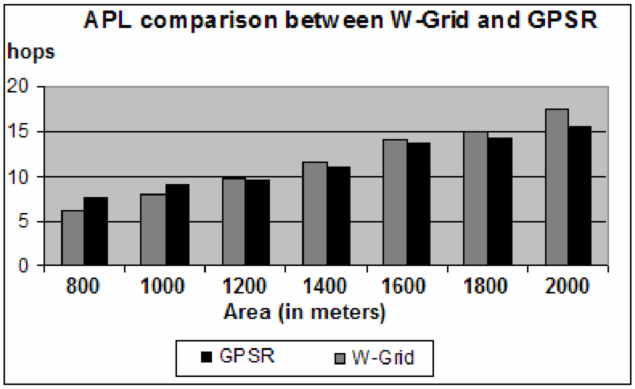

Fig. 3. Average path length comparison between W-Grid and GPSR

tion 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). Obviously the comparison is prohibitive, since GPSR can stay very close to the ideal routing algorithm also because it uses physical position of nodes. But our intention was to prove that W-Grid can return good performances anyway, especially considering that it doesn't require any kind of information about geographic position of nodes. This means not only a vaster and heterogenous 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.

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 may 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 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.

VI. 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. W-Grid virtual coordinates can act as a distributed database without needing neither special implementation nor reorganization and any kind of data can be distributed, stored and managed. 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.

References

[1] W. L. C. Chiang, H. Wu and M. Gerla. Routing in clustered multihop, mobile wireless networks. In *Proc. IEEE SICON'97*, pages 197–211, April 1997.

[2] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[3] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD Conference*, pages 47–57, 1984.

[4] D. Johnson, D. Maltz, and J. Broch. DSR: the dynamic source routing protocol for multihop wireless ad hoc networks. *Ad hoc networking*, pages 139–172, 2001.

[5] B. Karp and H. Kung. GPRS: 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.

[6] G. Moro and G. Monti. 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.

[7] G. Moro, G. Monti, and A. Ouksel. Merging G-Grid P2P systems while preserving their autonomy. In *P2PKM*, 2004.

[8] 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 *in Proceedings of the Second International Workshop on Agents and Peer-to-Peer Computing, Melbourne, Australia, July 2003*, volume 2872, pages 123–137. Springer, 2003.

[9] M. A. Ouksel and O. Mayer. A robust and efficient spatial data structure: the nested interpolation-based grid file. *Acta Inf.*, 29(4):335–373, 1992.

[10] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.

[11] C. Perkins and E. 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.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM Press, 2001.