

VIRTUAL TRELIS ROUTING

How Regular Structures can ease Network Operations

Julien Ridoux, Anne Fladenmuller, Yannis Viniotis

LIP6 - UPMC

8, rue du C. Scott - Paris, France

{julien.ridoux, anne.fladenmuller}@lip6.fr, candice@eos.ncsu.edu

ECE Department - NCSU

Box 7911 - Raleigh, NC 27695, USA

Abstract The mobility characteristic of Ad Hoc and Sensor networks implies that the topological information contained in the traditional IP address can no longer reflect the position of a node in the network. In this paper we propose a new approach to resolve the location-identification coupling contained in the IP address. The approach uses a Virtual Regular Structure (VRS) to describe the addressing space of the entire network; such a structure provides additional desired properties such as robustness and multi-paths. Our approach explores a distributed implementation of the VRS based on a trellis graph description of routing tables instead of the traditional trees. We show that the construction of the optimal structure is an NP-Complete problem; in this paper, we propose a heuristic and evaluate its performance via simulations.

Keywords: Self-Organized Networks, Regular Structures

1. Introduction

Unlike traditional (wired) networks, the Ad Hoc and Sensor networks we target in this study face unique challenges: they do not rely on any routing infrastructure, their topology is flat and not known in advance, their links are not reliable and they have to be tolerant to transient node presence. The main functions of a network layer, namely, building a routing table, performing lookups in this table, taking a routing decision, and efficiently processing packets, have to be performed in the presence of these challenges.

IP addressing, as it is commonly used, does not suit the nature of such Ad Hoc and Sensor networks. Because of the transient presence of nodes, the node identity and location can not be maintained in the same piece of information. The implicit association between identity and location used in wired IP networks eases the process of building, maintaining

and using routing tables. As soon as the nature of the network forces separation of identity and location, a lookup on the identity has to be realized in order to retrieve the location. Indirect routing approaches [6] or geographic approaches [4] are examples of this explicit association retrieval. From a high-level point of view, putting all the information needed by the network layer in a *single* data element would improve the functions of the network layer. Such a perfect situation is, of course, not realistic, and, depending on the nature of the networks studied, several association combinations are possible.

In this paper, we want to introduce a new topological space (i.e., an addressing scheme different from the traditional IP address space and tree-like routing tables) that is suitable for and facilitates network layer operations in the presence of mobility, node faults, network mergers and splits. The precise definition and distributed construction of the addressing scheme used by the network layer is the fundamental and first step to realize. In this paper, we focus our attention to this point only. For a more detailed description of how one can use our new addressing scheme to perform data forwarding, see [3].

The study done by Castro *and al.* in [7] shows that imposing a *structure* in this scheme eases the functions of the network layer and thus improves its overall performance. Since Sensor or Ad Hoc networks cannot rely on a physical structure by themselves, we propose to introduce structure in a *virtual* manner.

The Virtual Regular Structure (VRS) we propose maintains an organized, logical topology, in spite of mobility effects. The basic idea is to define and build the VRS in a *recursive* manner, using the same building block (a trellis graph), in order to ease the network management operations when nodes move or network splits/mergers occur. The construction of the VRS is based on physical neighborhood information only, to reflect the physical topology of the nodes that form the network. The set up of the VRS is “constructive” and realized locally, in order to avoid the need of a global view of the network, difficult to obtain in our context. This mechanism avoids widespread flooding on the network for any management operation and owns, by nature, good properties in terms of scalability.

In section 2, we detail how the topological location space is constructed. We show that the trellis construction problem is NP-Complete and provide a heuristic solution. In section 3, we present simulations showing the feasibility of constructing a VRS based on topological information.

2. VRS based on Trellis Graphs

Our proposal to create the Virtual Regular Structure (VRS) is based on the following hypotheses, reflecting the nature of Ad Hoc and Sensor networks. The network is built from scratch and the physical topology created by the nodes arrival process can not be predicted. We make no assumption on the properties of the topology and suppose that no infrastructure is present to help supporting the addressing or routing mechanisms. We make the hypothesis that each node joining the network possesses a unique Universal Identifier (UI), which could be an IP address. The definition of this identifier is not handled by the system, but reference [2] provides information for such a definition.

Trellis graphs are used here to implement the VRS for Ad Hoc and Sensor networks. To the best of our knowledge, trellis graphs have been used only in [8] at the networking level but with a different purpose from ours. A trellis graph can be generated by a Convolutional Encoder [9].

A trellis is a connected graph, repeating a set of vertices, all of which have the same degree. The number of vertices of a trellis is 2^L , L defining the trellis size. Another representation of such a trellis graph is a Finite State Machine (FSM). An example of trellis graphs is shown in the right part of figure 1. The vertices of a trellis, labeled by binary values (in the example from 00 to 11), represent the different FSM states. The trellis edges represent the FSM transitions. A number of trellis graphs can be generated, with different node degrees, introducing more or less redundancy and so, more or less possible paths between two distinct vertices. In the rest of this section we consider trellis graphs of size 4 for explanation purposes.

2.1 Description of a Single Cluster

In order to set up the VRS in a distributed manner, the FSM defining the trellis is known *a priori* by all the nodes composing the network. The trellis pattern defines the routing table layout that is used by nodes in a cluster. The clusters are formed by associating nodes (labeled with their UI) to vertices of a trellis (entries of the routing table). This association creates a mapping of the physical topology into the virtual one (the trellis graph). The label of the vertex of the trellis associated to a node is called its Local Relative Address (LRA). This mapping introduces the correlation between the identity of the nodes (their UI) and their address (their LRA). Depending on the node arrival sequence (even if resulting in the same physical topology), the construction heuristic we propose possibly produces different mappings.

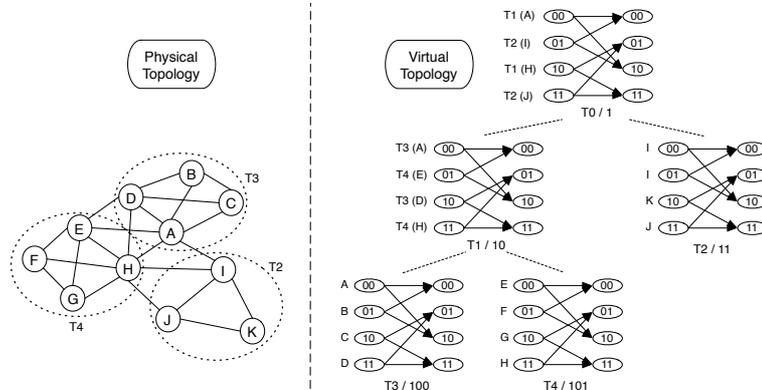


Figure 1. Spanning the Network.

Figure 1 gives an example of such a clustering. The topology in figure 1 contains the four nodes A , B , C and D , which are associated to the vertices of a trellis labeled T_3 . In this case, each node of the physical topology is associated to a unique vertex of the trellis. In trellis T_3 , node A is associated to the vertex labeled 00 and nodes B , C and D are respectively associated to vertices 01, 10 and 11. This mapping operation results in the creation of a cluster of nodes. This cluster represents a subset of the nodes' physical connectivity. Figure 1 shows that a node can be associated to different vertices of a trellis. As an example, node I in trellis T_2 is associated to two vertices, *i.e.*, I is being given two LRA in the virtual structure: 00 and 01. This shows that one or several LRA can be allocated to a physical node.

2.2 Spanning the entire Network

As we presented, a trellis graph defines a cluster of a maximum number of nodes. In order to be able to span the entire network and include each node in the VRS, our mechanism adds clusters (*i.e.*, trellis graphs) in a recursive manner. This recursive construction introduces trellises of trellises, repeating the same regular structure.

Figure 1 shows trellis graphs T_3 , T_4 and T_2 mapping respectively the nodes $\{A, B, C, D\}$, $\{E, F, G, H\}$ and $\{I, J, K\}$. This mapping of nodes creates a first level of clusters. In order to allow all the nodes in the network to communicate, the trellis graphs T_2 , T_3 and T_4 are interconnected recursively. Figure 1 gives an example of a VRS with two levels of recursion. As an example, T_1 associates T_3 to its states 00 and 10,

while it associates T_4 to its states 01 and 11. T_0 and T_1 can then be described as “trellis of trellis graphs”.

Even if trellis graphs are defined in a virtual space, the actual communication between them has to be ensured by the physical nodes interconnecting them. The two nodes chosen to represent their trellis in higher recursive levels are always those whose labels have the smallest and highest values in the trellis. We name these two nodes the Trellis Heads. In figure 1, T_3 is represented in T_1 by its Trellis Heads $A(00_{T_1})$ and $D(10_{T_1})$. In the same manner, T_4 is represented by nodes $E(01_{T_1})$ and $H(11_{T_1})$ in T_1 .

We define a Trellis Prefix (TP) to identify each regular structure. By definition each trellis of size 2^L gathers $2^{(L-1)}$ sub-trellis. Each trellis T attributes a prefix p written as a bit string of size $(L-1)$ to the nodes representing a common sub-trellis. A sub-trellis’ TP is the recursive concatenation of the higher level trellis’ prefix and p . The highest trellis in the recursive hierarchy possesses the label “1”. An example of the resulting TP attribution is illustrated in figure 1, where T_4 ’s TP is 101, the concatenation of T_1 ’s TP (10) and T_4 ’s prefix in T_1 (1).

At each step of the VRS construction, each label of each trellis composing the VRS is associated to a physical node. It is then possible for a source to retrieve the path in the VRS to any destination. This path is represented by the Local Relative Address of the destination in its trellis and the Trellis Prefix of the cluster it belongs to.

In figure 1 example, let suppose that F and J need to reach K . For J , the case is trivial. J and K belong to the same cluster and share the same mapping information. J knows K ’s LRA as 10_{T_2} and can reach it based solely on its knowledge of the trellis connectivity. From F ’s point of view, the path to reach K is represented by the Trellis Prefix of the cluster F belongs to (the trellis T_4 with prefix 101) and the LRA of F in T_4 (01_{T_4}). The forwarding of packets is realized on per cluster basis, by processing the Trellis Prefix of the destination contained in the packet header. When the packet reaches the destination’s cluster, it is forwarded to the destination thanks to its LRA information. Since we introduce redundancy in the structure, several paths can be defined for each cluster the packets go through, giving the multi-path property.

2.3 Construction Complexity and Heuristic

Finding the optimal construction of the recursive clusters (*i.e.*, one that minimizes the number of clusters) is an NP-Complete problem. In the following we formalize the problem in terms of graph theory and show the equivalence with the well-known H-Matching problem [5].

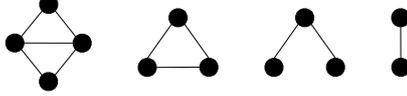


Figure 2. The 4 Derived Graphs of the trellis of size 4.

The cluster resulting from the mapping of nodes' UI to the trellis vertices corresponds to a connected subgraph of the physical network topology. We name this connected subgraph the trellis Derived Graph. The mapping operation producing this Derived Graph can be represented by the following steps.

(1) Let \mathcal{M} be the FSM representation of a trellis with a number s of states s_i . \mathcal{M} can be represented as an edge-labeled, directed graph M . We first make the adjacency matrix of M be symmetric and put the values on the diagonal to 0 (to remove the self loops), resulting in the connected undirected graph M' .

(2) Let W be a set of vertices such that $|W| \in [2, s]$. W represents the nodes of the physical topology belonging to a common cluster. Then, the mapping operation is represented by an application \mathcal{A} such that $\forall i \in [1, |W|], \forall j \in [1, s], v_i \mathcal{A} s_j$ produces an instance of M' .

(3) The presence of a physical node in the cluster, mapped to different labels of the trellis, results in duplicate entries in M' adjacency matrix. By removing these entries we obtain one of the trellis Derived Graphs $D = (W, F)$, whose cardinality corresponds to the number of nodes in the cluster. Figure 2 gives all the possible Derived Graphs of the trellis of size 4 where $|W| \in [2, s]$.

Let $G = (V, E)$ be the undirected, connected graph representing a given network topology. Let $D = (W, F)$ be one of the possible Derived Graphs of a trellis of a given size. Creating the clusters on G that will map the nodes to the trellis is equivalent to an H-Matching of G , where H is the Derived Graph D . Then,

- (i) if $|W| = 2$, the H-Matching of G by D is realized in a polynomial time.
- (ii) if $|W| \geq 3$, the H-Matching of G by D is NP-Complete (the proof of this result can be found in [5]).

The smallest trellis of interest to us produces Derived Graphs of more than two vertices, leading to the definition of clusters along an H-matching where $|W| \geq 3$. An optimal H-Matching in our context consists in placing as many nodes as possible in a trellis, *i.e.*, using the Derived Graphs for which $|W|$ is maximal, and is then NP-Complete.

As said, the complete construction of the VRS is a recursive process. This implies to realize the H-matching described, at each level of recursion, on the Trellis Heads obtained from the previous clustering step. Each step of the construction corresponds to the solution of an NP-Complete problem.

Since the construction of the VRS is an NP-Complete problem, we propose a heuristic to set it up, based on the fact that nodes join the network in an incremental manner. Our heuristic builds the top trellis of the VRS first, and adds lower level trellises as new nodes join. We describe in the following algorithm 1 the key points of the heuristic only. When a node joins a network, it tries to join the existing VRS by being inserted in an existing trellis. If no possible position is available, because of physical connectivity constraint, the arriving node creates a new trellis with its neighbor. This heuristic ensures that any new node joining the network can be configured and adapts the total size of the addressing space to the number of nodes present in the network.

This heuristic shows how our addressing space is built in a distributed but local manner among physical neighbor nodes. The VRS is built without widespread flooding mechanisms, by maintaining locally the common mapping in use in a cluster. Having a regular structure of finite size, and a construction mechanism realized locally gives natural scalability properties to our proposal.

Algorithm 1 Construction Heuristic

```

node  $N$  joins the network
if  $N$  detects some neighbors then
  each neighbor returns the list of trellises it belongs to
   $N$  sorts the trellises by their decreasing level in the VRS and then the decreasing number of
  nodes they contain
  for each trellis do
     $N$  lists the available positions in the trellis
    for each position do
       $N$  checks recursively the position in lower trellis
      if the position is one of the Trellis Head then
         $N$  checks recursively the position in upper trellis
      end if
      if position is valid then
        break
      end if
    end for
  end for
  if No available position has been found then
    for each trellis do
       $N$  selects two neighbors that do not belong to a lower trellis
       $N$  creates a new trellis with these two neighbors
    end for
  end if
else
   $N$  creates a new trellis
end if

```

3. Evaluation and Simulations

With the definition of our heuristic, it is always possible to build a recursive trellis-based VRS containing all the network nodes. Since the complexity of such construction is NP-Complete for an existing topology, it is important to simplify the construction as much as possible. It is also important to maintain performance in terms of computational load on the nodes for the routing, minimization of the information storage required for our VRS or matching of the physical minimum route and the path in the VRS to the destination. Analyzing the structural properties of the VRS produced gives a first estimate of such costs.

3.1 Definition of performance estimates

The computation required to route packets along the VRS mostly depends on the length of the path between nodes. The longer the path in the VRS, the more processing time is required to compute it, since more nodes are involved in forwarding packets. Statistically, it seems intuitive that the *depth* of the VRS will have an impact on the average length of the path between nodes. The depth of the VRS can give us some insight on the performance in terms of average routing processing load for a given topology.

Each trellis stores information on each association (LRA, UI) of its own trellis and of all the trellises of longer prefixes below it. The number of LRA per trellis is fixed, and several LRAs can be associated to one UI. So, in order to optimize the amount of information stored in the VRS, we need to minimize the number of trellises and to create the shortest structure possible. This is clearly what our heuristic tends to do, by limiting the creation of new trellises.

As a result of this analysis, we can deduce that the depth of the virtual structure and the number of trellises has an impact on the performance of our approach. Although our heuristic tends to minimize both metrics, it can not be guaranteed they will both be optimized. One major characteristic of our proposal is that, although we are sure a VRS can be built, we can not predict its resulting shape. Then, for a given topology, several VRS can be built, depending on the node arrivals: the number of trellises as well as the maximum number of recursive trellises (depth of the VRS) can differ. It is thus important to understand which parameter will impact performance the most.

In order to evaluate these structural performance estimates, we generated topologies where nodes are placed randomly in a square geographic area. Each wireless node has a transmission range of 250 meters. The dimension of the square area is defined by the geographical density of

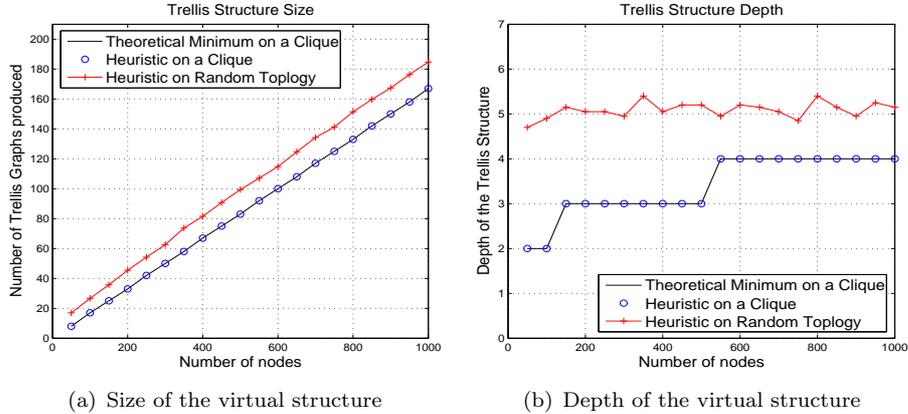


Figure 3. Heuristic Quality Evaluation.

the topology, fixed to 35 nodes per km^2 . The value for a topology size corresponds to an average over 25 runs in the same conditions. In these simulations, we studied the construction heuristic we propose by using a trellis of size 8.

3.2 Heuristic Evaluation

In order to evaluate the proposed heuristic, we computed the theoretical minimum number of trellises and the minimum depth of the VRS for a given number of nodes, respectively shown in figure 3(a) and 3(b). These bounds correspond to a physical topology that can be represented as a clique, *i.e.*, without any physical connectivity constraint. In both figures 3(a) and 3(b) we observe that the proposed heuristic leads to the theoretical minimum bounds on a clique topology. This clearly shows that the heuristic achieves its goal on a topology where the physical constraints does not have to be taken into account.

On the random topology we produced, figure 3(a) shows that our heuristic remains close to the optimum number of trellis. Even if our heuristic does not clearly lead to the optimum because of the nodes arrival process and the physical connectivity constraint, we observe that the number of trellis graphs produced does not increase exponentially with the size of the topology.

Figure 3(b) shows that the depth of the structure on a random topology can be more than twice the optimal depth computed. For topologies of smaller sizes, the connectivity of the network does not allow new nodes to be inserted in trellises of higher level in the structure. This leads to the construction of new sub-trellises that increase the overall depth of

the structure. Having a VRS of higher depth while the total number of trellises stays close to the optimum indicates that the VRS is not well balanced. Nevertheless, we again observe that the depth of the VRS does not increase exponentially with the number of nodes.

4. Conclusions

In this paper we have proposed the use of virtual Trellis structures to (a) organize the addressing space of a self-organized Ad Hoc and Sensor networks in a structured fashion, and, (b) to perform data routing in such an environment. Thanks to the redundancy and recursion introduced in the trellis-based VRS, we provide a distributed dynamic addressing scheme, with the following advantages: localized operations, no size limit in the addressing space, a built-in multi-path routing structure, computed locally, and robustness to various kinds of mobility. The construction of the optimal trellises was proven to be an NP-Complete problem. A heuristic was provided to build the VRS and evaluated via simulations. Future work will study the impact of group mobility on our scheme and compare the performance of our approach to other schemes.

References

- [1] C.E. Perkins (1996). IP Mobility Support. RFC 2002, IETF.
- [2] I.Stoica, R.Morris, D.Karger, M.F.Kaashoek, H.Balakrishnan (2001). Chord: A Scalable P2P Lookup Service for Internet Applications. In *Proc. of ACM SIGCOMM*.
- [3] J.Ridoux, A.Fladenmuller and Y.Viniotis (2004). Beyond the Tree Structure: a new way to configure nodes in SONs. Technical report. http://www-rp.lip6.fr/~ridoux/Publications/TR_BeyondTrellis.pdf.
- [4] J.Li, J.Jannotti, D.De Couto, D.Karger and R.Morris (2000). A Scalable Location Service for Geographic Ad Hoc Routing. In *Proc. of ACM MOBICOM*, pages 120–130.
- [5] Kirkpatrick, D. G. and Hell, P. (1978). On the Completeness of a generalized Matching Problem. In *Proceedings of the tenth annual ACM symposium on Theory of computing, San Diego, CA*.
- [6] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. P. Hubaux and J. Y. Le Boudec (2001). Self-Organization in Mobile Ad Hoc Networks: the Approach of Terminodes. *IEEE Communications Mag.*, 39(6).
- [7] M.Castro, M.Costa and A.Rowstron (2003). Should we build Gnutella on a Structured Overlay? In *HotNets-II. Cambridge, MA, USA*.
- [8] S.D.Nikolopoulos, A.Pitsillides, and D.Tipper (1997). Addressing Network Survivability issues by finding the K-best paths through a Trellis Graph. In *Proc. of IEEE INFOCOM*.
- [9] S.Lin, D.J.Costello (1983). *Error Control Coding: Fundamental and Applications*. Electrical Engineering Series. Prentice-Hall.