Easily-Managed and Topology-Independent Location Service for Self-Organizing Networks

Aline C. Viana
Marcelo D. de Amorim
Serge Fdida
LIP6/CNRS
University of Paris VI
{viana,amorim,sf}@rp.lip6.fr

Yannis Viniotis
Dept. of ECE
NC State University
candice@ncsu.edu

José F. de Rezende
GTA/COPPE
Federal Univ. of Rio de Janeiro
rezende@gta.ufrj.br

ABSTRACT

The need for efficient location mechanisms is an important issue in scalable self-organizing networks. Existing solutions are inherently dependent on the spatial distribution of nodes in the topology. This leads to limitations that go against the principles of self-organization. In this paper, we propose Twins, an easily-managed location service for self-organizing networks. Twins defines a logical multidimensional space that is a strict mathematical representation of the network geographic space. This representation is obtained through Hilbert space-filling curves. The geographic space is used for addressing and routing, while localization is based on the curve. Control messages are routed based on the logical structure while data packets are routed in a hop-by-hop basis with greedy next-hop choice. In this paper, we evaluate the Twins management operations in terms of fairness of space sharing and logical/geographic distances between nodes and their location servers. Our results show that Twins assures a fair distribution of control overhead and scales well with the number of nodes.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Distributed networks; C.2.2 [**Network Protocols**]: Routing protocols

General Terms

Design, Management

Keywords

Location service, self-organizing networks, network topology, wireless communication, protocol architecture, routing protocols.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.
Copyright 2005 ACM 1-59593-004-3/05/0005 ...$5.00.

1. INTRODUCTION

Recent advances in communication technologies are opening new ways for mobile users to get connected to each other. In addition to the traditional wired infrastructure, which is characterized by a static and relatively centralized management model, users will have the possibility to spontaneously establish *self-organizing networks* (SONs). Ad hoc and wireless sensor networks are examples of such networks.[1]

The main characteristics of a SON that set it apart from a traditional network are the lack of a management infrastructure (*e.g.*, no centralized addressing scheme is present) and the dynamics of the network. The two main characteristics impose two fundamental requirements on the design/operation of a SON: (a) all nodes in the SON should assume the same management responsibilities, and (b) any network operation (*e.g.*, management of the addressing space) should be inherently distributed. Additional peculiarities include the possible lack of geographic positioning infrastructure, the limited and variable capacity of wireless links, and the energy-constrained nature of some nodes.

Self-organization introduces new, radically different technical requirements. We discuss design requirements in Section 2. In this paper, we focus only on routing-related issues, in both the data and the management planes. Because of the different requirements they confronted, traditional routing protocols were not designed to efficiently address all the constraints a SON imposes. Routing in SONs requires then revisiting important components of the routing architecture/protocol suite, such as: (a) establishing and managing an addressing scheme, (b) providing a location service (when required), and (c) defining a packet forwarding mechanism.

Much research has been done in all three components. In the traditional Internet model, routing information is embedded into the nodes-positioning-dependent node address, *i.e.*, IP addresses have been defined for both *identifying* and *locating* a node in the network. This does not work well in mobile networks, because permanent node addresses cannot include dynamic location information, which invalidates topology information. More recently, a number of flooding-based protocols have been used to address this problem in

[1]Other types of networks also fall within the classification of SONs (*e.g.*, P2P networks). In this paper, we refer to SONs to represent non-overlay networks.

the specific case of ad hoc networks. Nevertheless, it has been observed that these architectures do not scale well [1]. For instance, in wireless sensor networks, where the potential number of addressable nodes may be in the order of thousands, current solutions cannot be used. Geographic (or position-based) routing has proven to be an efficient solution to address the new requirements introduced by SONs. Because of its simple forwarding decisions and no need for maintaining explicit routes, this kind of routing algorithm is scalable and robust to dynamic networks [2]. Position-based routing protocols, however, are heavily dependent on the existence of an efficient location service, since a source must know the destination's position before sending a message. Existing scalable location services are inherently dependent on the spatial distribution of nodes in the topology. As we will see in this paper, this leads to a number of limitations that go against the principles of self-organization.

In this paper, we propose *Twins*, a novel architecture to perform position-based routing supported by an efficient and easily-managed location service in SONs.[2] Twins defines a logical multidimensional space that is a strict mathematical representation of the network geographic space. The logical space is completely partitioned among the nodes and routing is performed based on the partitions assigned to these nodes. In this way, Twins defines a multi- to one-dimensional mapping of the logical space, which allows the location service to be performed in an easily-managed one-dimensional structure. The need for an efficient location mechanism is even more important where topologies become larger and more complex in their addressing structure. Multidimensional cartesian spaces are robust to dynamic networks but introduce complexity to the management of the space partitioning. This is because they are completely dependent on the network geometry, which is dynamic and unpredictable in the case of SONs. Our goal is to obtain a compromise between robustness and complexity. Twins allows geographic routing to be performed in a logical space, while a topology-independent location service is implemented over a one-dimensional (and thus much easier to manage) addressing structure.

Twins introduces a hierarchical addressing architecture that is based on the mathematical concept of Hilbert space-filling curves. In summary, with respect to the design criteria presented in Section 2, the main technical advantages of our proposed architecture and associated management operations are scalability, physical connectivity-based routing, simplicity of management, and fairness. Management operations are simplified thanks to (a) separation between node identifier and node address, and (b) separation between multidimensional addressing/routing architecture and one-dimensional location structure. These separations, respectively, decouple the data and control plane operations of the network, and assure the topology-independence of the location service. The good clustering property of Hilbert curve assures that the locality between nodes in the multidimensional architecture is better preserved in the linear structure than the others space-filling curves [3]. Fair distribution of the control overhead among all present nodes is achieved via a management protocol that strives to allocate (almost) equal partitions of the addressing space to each node. Data forwarding is done using the standard notions of rendezvous abstraction and path selection based on geographic routing [4, 5, 6].

[2]The term Twins is not an acronym; it makes reference to two distinct though similar architectural representations of the network.

Hilbert space-filling curves have been used in a number of domains. The work conducted by Xu *et al.* [7], for instance, solves the dimensionality mismatch between their proposed landmark space and a peer-to-peer overlay network. Their approach describes the construction of an auxiliary network for any distributed hash-table (DHT) based overlay to take advantage of the nodes physical proximity. Nevertheless, the auxiliary network must run over static environments (as Internet-like topology), which is clearly not applicable to dynamic SONs. Moreover, their proposal falls under the umbrella of techniques to exploit topology information in overlay routing.

Twins is not an overlay network. Overlays rely upon the support of a routing protocol that guarantees the connectivity between nodes. Twins implements a location service at the routing layer. The benefits of DHT-based location mechanisms [8] are directly incorporated into the network layer to distribute node location information and are completely independent of any peer-to-peer system implemented at the application level. Furthermore, Twins takes advantage of the Hilbert space-filling curve's properties to implement a topology-independent location service, which is adaptable to highly dynamic environments.

We evaluate the performance of Twins through analysis and simulations. We show that the amount of work each node performs does not rise quickly as a function of the total number of nodes, and thus, we show how well Twins scales. We firstly focus on the fairness of the proposed address management algorithm. We show via a theoretical and simulation study that the used management scheme results in fair distribution of control overhead. We evaluate then the space partitioning-based path length over the geographic path length. Our results show this cost scales well with the number of nodes. We also provide some results that illustrate the importance of correctly dimensioning the space-filling curve, and its effects on Twins's fairness.

The paper is organized as follows. In Section 2, we discuss design requirements for establishing an addressing space and routing (data forwarding and management) in a SON. In Section 3, we contextualize our proposition and review existing works. In Section 4, we present the concept of Hilbert curves and discuss how it can be used to establish a structured addressing space for a SON. In Section 5, we present how the location service can be provided; we discuss in detail how the forwarding operations of the model can be implemented with the new concept of Hilbert addresses. In Section 6, we describe the required operations in the management plane. Finally, in Section 7, we present the evaluation of Twins's performance.

2. SON REQUIREMENTS

When required, a scalable location service for SONs must be designed in such a way to respect a number of requirements that directly affect the routing architecture. The most important requirements are:

- *Infrastructure-free and non-authority capabilities:* Nodes must be autonomous and decisions must be taken in a local scope through simple neighborhood consensus.
- *Distributed nature:* information and management re-

sponsibilities should be completely distributed among the nodes in the network.

- *Flexibility in route selection and dynamic-network management:* the addressing structure should offer flexibility in route selection. This issue has an impact on the dynamic-network management and affects the performance in terms of path length, traffic concentration, and resilience to failures.
- *Scalability and low control message overhead:* lookup operations should avoid heavy-overhead solutions like flooding the entire network to locate a node. Related requirements include: simple forwarding decisions, low communication cost, and forwarding tables that are independent of the total number of nodes in the network.
- *Manageable complexity:* the addressing structure should be as flexible as possible when handling the addressing space allocation as nodes join/leave/move.
- *Fair distribution:* fair allocation of routing overhead to nodes is required in order to better distribute the management responsibilities throughout the nodes. Fair allocation is strongly related to the design of the location service.
- *Easy path computation:* paths must be easily determined, independently of the complexity of the addressing structure.

In general, when a source wants to communicate with a destination, the only information it has is the destination's *identifier*. The location service is responsible for translating this identifier into an *address*. An example is the DNS in the Internet case, which receives a name (*e.g.* a URL) and gives the corresponding IP address. Nevertheless, DNS relies upon a centralized architecture, which is clearly not implementable in a SON.

A SON requires a dynamic association between identification and location of a node and the specification of an architecture to manage this association. Recent attempts to implement such a scheme rely on the concept of *indirect routing* [4, 5, 6, 9]. The idea consists of completely distributing node location information in location servers throughout the topology. We focus on the main issues that must be addressed in SONs and in indirect routing.

In the indirect routing strategy, a destination node's address and identifier have different meanings. By implementing the indirect routing approach, a location service can be used by a source to obtain the destination's address at the time preceding the communication. Distributed hash tables (DHTs) have been adopted as a scalable framework to implement indirect routing, and consequently an efficient location service, upon which a variety of self-organizing systems have been built [4, 5, 6]. Furthermore, geographic or position-based routing has proven to be an efficient solution to address the new requirements introduced by SONs [2].

Twins implements a DHT-based location service completely independent of the physical location of the nodes. The topology-independence of the location service is based on the mapping between nodes' identifiers into *rendezvous points* (RP), and on the association of these rendezvous points with *rendezvous nodes* (RN). Each node has an identifier, which is hashed into a RP of an addressing space. Partitions of this addressing space are assigned to nodes in the network. A node whose partition contains a RP is the RN responsible for storing the current location information of the node associated with that RP. The basic operation in our location service is `Lookup`(RP), which returns the node controlling the partition of the space containing that point. Let C be a node in the network and $l(C)$ be its current location. The C's RP is given by hashing C's ID, *i.e.* RP(C)= $h(\mathrm{ID}_C)$, where $h(\cdot)$ is a hash function known *a priori* by all the nodes. The partition of the logical addressing space under the responsibility of node i is noted $p(i)$. Node i will be the RN of node C if and only if $h(\mathrm{ID}_C) \in p(i)$.

3. TWINS CONTEXTUALIZATION

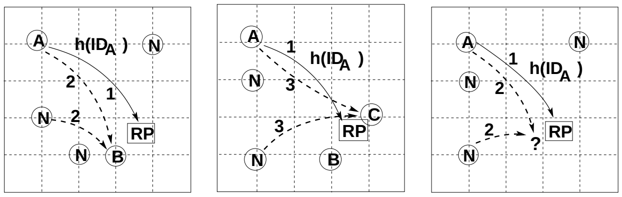
The design of a self-organized network architecture requires an efficient combination of robustness and complexity. The resilience of existent proposals and, consequently, the performance of the routing protocols are strongly related to the complexity of the deployed addressing structure. On the one hand, tree-like structures (*e.g.* L+ [10] and Peernet [11]) lead to simple manageable spaces. Nevertheless, they have low route selection flexibility, which results in low routing performance and poor resilience to failures/dynamic networks. Their low complexity is obtained at the cost of some loss of robustness. On the other hand, more complex structures, like multidimensional cartesian spaces, improve the resilience and routing performance due to the flexibility in route selection. Nevertheless, the associated addressing and location models become more complex and require a tight association between the logical and physical planes.

Multidimensional cartesian spaces have been used by a number of research proposals [4, 5, 6] to the implementation of robust indirect routing using geographic coordinates. An intuitive manner to implement indirect routing using geographic coordinates is to assign zones of the entire geographic space to each one of the nodes. Each node is responsible for managing location information associated with its zone. Such a technique has been successfully applied in peer-to-peer overlays. Nevertheless, neighbors in an overlay can be physically distant, which is not true in a SON.

Due to the complexity involved with a dynamic multidimensional partitioning and management, existing approaches only associate nodes to RPs (*i.e.*, to geographic positions) but not space partitions to nodes. This requires the presence of at least one node near the RP to play the role of the RN. This cannot be guaranteed in a SON because of its inherent variability, in time and in space (see Fig. 1). Topology changes imply that the network is periodically checked in order to accordingly update the associated RNs. The selection of RNs is strongly correlated to their physical coordinates, which introduces some loss of freedom in the indirect location service and extra overhead. This may be prohibitive in many dynamic networks.

3.1 Related works

Terminodes [4], GLS [5], and DLM [6] are examples of interesting DHT-based geographic routing systems. In Terminodes, each node advertises its current position to location servers placed in a geographic region called Virtual Home Region (VHR) of radius R. The center of the region is obtained by hashing the node's ID into a position in the geographic space. The radius R is increased or decreased if the number of nodes in the VHR is respectively below or above a predefined threshold. This dependency on density forces the protocol to check the entire region periodically and to



(a) (b) (c)

Figure 1: Lookup phase (arrow 1) in a topology-dependent location service. (a) $h(ID_A)$ result in a geographic position $RP(A)$, where the closest node is B. Lookup messages are then forwarded to node B (arrow 2). (b) With topology change, the closest node is now the node C (arrow 3). The periodical scan of the network will then update A with its new RN information. (c) Nodes are geographically far from RP(A). It is then difficult to determine and manage which node must play the role of RN(A).

change location servers accordingly. Furthermore, nodes in sparse regions need to scan and advertise larger regions in order to avoid empty VHRs.

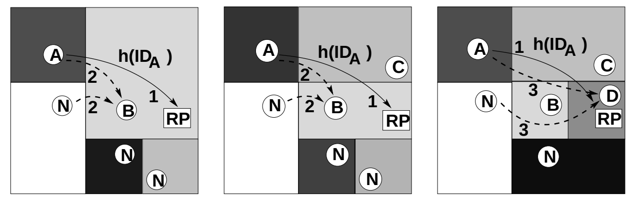
In GLS, the geographic topology is defined as a hierarchy of grids. For each hierarchy, each node determines a location server whose ID is the closest to its own ID. As in the case of Terminodes, each GLS node needs to scan each hierarchy of regions to find out which node will serve as a location server. Furthermore, due to topology changes, this scan needs to be performed periodically in order to keep location servers in accordance with the current status of the network.

DLM is similar to GLS, but it uses a different strategy to select location servers. In its approach, a hash function maps a node's ID into a set of physical regions in the network – selected nodes in those regions act as the node's location servers. Nevertheless, in order to find an appropriate location server, a node may need to scan several regions if there is no node in the previous computed one.

Observe that these approaches require the presence of at least one node around the RP to properly function. In Twins, a node does not need to scan the network to determine its new RN each time topology changes happen. By consequence, the heart of our approach is centered on the management operations, in order to correctly and consistently partition the addressing space among nodes in the network.

3.2 Logical/geographic addressing separation

Based on the abovementioned observations, we propose to completely decouple the geographic plane, used for routing, from the localization plane, used by the location service. Observe that this does not correspond to the establishment of an overlay network. Our system offers the freedom of geographic forwarding for data packets and a complete topology-independent location service. By mathematically representing the geographic distribution of nodes in a logical addressing space, we dynamically assign (1) slices of this addressing space to RN, and (2) nodes to points/addresses in those slices. As we will see in the following, Twins assures that there exists at least one node in the network that owns the slice of space containing the RP. Furthermore, the associated management plane assures the correct and dynamic space partitioning, as well as the accurate forwarding of messages toward the corresponding RN. In the case of topology changes, the space partitioning is automatically updated by the neighbors on the curve (local-scope operation with cost $O(2)$) without requiring any intervention of the nodes whose RP fall within the slice being recovered (see Fig. 2).



(a) (b) (c)

Figure 2: Lookup phase (arrow 1) in a topology-independent location service. $h(ID_A)$ results in a RP managed by the node B. Lookup messages to RP(A) are then forwarded to node B (arrow 2). (b) With a topology change, node B is still the node managing RP(A), though C is the closest node to this point. Messages are correctly forwarded to B without generating any overhead/update in A. (c) If the topology changes and node B is no more RN(A), messages are correctly forwarded to the new node D (arrow 3).

In the following sections, we present our logical addressing system and explain how our mathematical representation allows the specification of a logical structure where geographic routing can be exploited while a topology-independent location service is performed in a simple way.

4. TWINS ADDRESSING STRUCTURE

The solution we propose is to decouple the location service from the geographic coordinates of the nodes. We will show that this can be obtained through the definition of a logical addressing space that mathematically represents the distribution of nodes under some criteria to be defined in Subsection 4.2. We will also see that the use of this logical structure is the key for obtaining both scalability and low management complexity of the addressing space.

4.1 The logical addressing space

Let $\mathcal{G}$ and $\mathcal{L}$ be, respectively, the geographic and logical spaces of the SON. The first step is to define a function that maps node coordinates in $\mathcal{G}$ into node addresses in $\mathcal{L}$. The objective is to use the resulting logical distribution of nodes to implement the location service, instead of the geographic coordinates. In this section, we focus on the mapping function. The way we use this logical space for identifying and managing RNs will be subject of Section 6.

A simple mapping function can be obtained if we define the logical space with the same mathematical structure of the physical topology. Without loss of generality, we assume in this paper that the physical topology is a two-dimensional plane of the network geographic area, with (x, y) coordinates for any location. With this assumption, a logical space can be dimensioned to exactly match the geographic area.

Although the use of multidimensional spaces (in this discussion a plane) increases the robustness of the forwarding mechanism, it also increases the complexity of the location system, *i.e.,* the management of RPs. In the ideal case, we should use a multidimensional space for forwarding, a one-dimensional space for locating, and a simple mapping between addresses in these spaces. In Twins, we use a one-dimensional representation of the logical space, through a mathematical technique called *space-filling curves*. Space-filling curves are commonly used to reduce a multidimensional problem to a one-dimensional problem. The curve is essentially a linear traversal of the discrete multidimensional space. We find interesting examples of their use in: data structures, storage and retrieval of multidimensional data, data compression, and optimization of routing systems [12, 13, 14].

We are not aware of any existing approach based on space-filling curves for the implementation of easily-managed location service at the network layer, prior to our own. Xu *et al.* [7] use Hilbert space-filling curves (*cf.* Section 4.2) to map coordinates in a landmark space (relatively high dimension) to IDs/keys in the overlay network (2-dimension). The authors describe generic techniques to construct an auxiliary expressway network for any distributed hash-table based overlay to take advantage of the inherent node heterogeneity that exists in the physical network. Such systems are classified as techniques to exploit topology information in peer-to-peer overlay routing. Their objective is to give to applications explicit knowledge about, and control over, data locality at a coarse-grain. To address these issues, instead of randomly generate node IDs and document keys, the idea is to promote the use of location-based node IDs and keys to encode physical topology and improve routing. Their topology-aware expressway uses Internet-like environments, like AS-level topology derived from BGP reports, and chooses only relatively stable nodes to act as expressway nodes. Moreover, periodical route advertisements using a variant of the distance vector algorithm are necessary among expressway nodes, which are significantly influenced by the dynamics in node membership. In this case, the benefits of an auxiliary network is reduced, being the default overlay used to publish information about the created topology, which incurs several physical hops for every logical hop. Instead, Twins's performance is completely independent of any application or distributed hash-table based overlay. Our primary goal is to address the constraints a SON imposes on routing-related issues and still assuring its requirements. In this way, Hilbert curves are used as a technique to implement an efficient location service while the position-based routing protocol execution is assured.

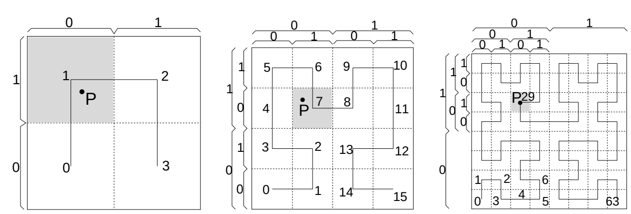
We describe in the following the mapping technique we adopt in this paper, namely *Hilbert Space-Filling Curve* [15]. The choice of Hilbert space-filling curves is due to its better linear mapping, which outperforms other curves (*e.g.*, z-order curve, snake curves, gray-coded curves) in preserving locality [3]. Its clustering properties allow us to assure that near nodes in the geographic space will represent near points on the curve.

4.2 Hilbert space-filling curves

Hilbert curves pass through every point in a n-dimensional space once and only once in some particular order. The idea is to graphically express a mapping between one-dimensional values and coordinates of points. To illustrate the main concept of Hilbert curves we consider in this section a two dimensional space. Without loss of generality, we assume a mapping between the points of a square and a finite line segment as the limit of an infinite sequence of nested intervals whose length tends to zero. Thus, a point in two dimensions is defined as the limit of an infinite sequence of nested squares whose area tends to zero.

Fig. 3 reproduces the figures shown in Hilbert's original paper, except that we number the points from 0 rather than from 1. Fig. 3(a) shows the initial square and line, each one divided in four pieces. The numbers show the correspondence between sub-squares and line intervals, established so that adjacent line intervals always correspond to adjacent sub-squares. The line connecting the centers of the sub-squares is a filling curve and defines an ordered sequence of sub-squares. Fig. 3(b) shows the next step in which each square and its corresponding line interval have been further subdivided with re-orientations of the sub-square sequences for the first and last sub-squares of Fig. 3(a). The re-orientations ensure that the adjacency property is preserved everywhere. Fig. 3(c) indicates the third step in the sequence.

The number of squares defines the *order* of the Hilbert curve. In practical applications, the process can be terminated after k steps to produce an approximation of a space-filling curve of order k. This curve passes through 2^{kn} sub-squares (for n dimensions) whose center-points belong to a space of finite granularity. The transformation of a curve of order $k-1$ into a curve of order k can be viewed as a replacement of each point on the former with a first order curve. The Hilbert Curve manifests a useful property in which consecutively ordered points are adjacent to each other in space. Although we have presented here the two-dimensional case, the Hilbert curve concept can be easily extended to any number of dimensions.



(a) 1st-order (b) 2nd-order (c) 3rd-order

Figure 3: The line connecting the centers of the sub-squares in each sub-figure is a filling curve and defines an ordered sequence of sub-squares in 2 dimensions.

Points in a Hilbert curve can be identified by either their coordinates in the multidimensional space or by their ordinal positions on the line. The two identifiers are defined here as (1) *derived-key*, which represents the ordinal position of a point on a curve of any order, and (2) *n-point*, which is the concatenated coordinates of a point on a first order curve; the strict definition of an *n-point* identifies the position of a point in the hierarchy of squares of the space. As an example, consider the point P in Fig. 3(c), whose *derived-key* is 29. This point on the curve locates P in the hierarchy of squares of *n-points*: 01, 10, and 01. The n-points identify the position of P in each square of the space defined by the 1st (Fig. 3(a)), 2nd (Fig. 3(b)), and 3rd-order curves (Fig. 3(c)). For example, 01 is the n-point of the high-lighted square in Fig. 3(a).

4.3 Structure considerations

Twins covers the network's geographic area with a space-filling curve, so that any specification of (x, y) coordinates can be equivalently given by the relative position along the curve. The mapping between geographic coordinates to logical addresses on the curve is straightforward; it is obtained by locally applying the Hilbert algorithm for associating n-points and derived-keys to geographic coordinates [16]. The only requirement is to respect the geographic positions of nodes when placing them on the addressing structure. n-points and derived-keys can be then obtained from geographic coordinates, and nodes can be addressed by either one of these values. By using the bijective mapping defined by the Hilbert Space-Filling Curve, our proposal addresses each node with a n-point – a point in the defined hierarchy of squares of the logical space – and with its corresponding derived-key – a point in the highest order Hilbert curve. We call these two addresses *Hilbert addresses*.

Twins assumes that nodes know their geographic position, which can be determined by any absolute or relative positioning mechanism [17, 18]. It is important to note here that, in practice, a single Hilbert point can be assigned to multiple nodes: the ones whose geographic coordinates fall inside the same Hilbert sub-square. As we will see in Section 6, Twins assigns segments of the Hilbert curve to nodes. We call such a segment the *control region* of a node. Nodes in the same Hilbert sub-square then manage the same partition of the addressing space, which allows inherent replication of location information in the architecture and increased robustness to failures. On the Hilbert curve, two nodes play particular roles. They are the *predecessor* and the *successor* nodes. They are formally identified through their corresponding Hilbert addresses and are responsible for control regions spanning the segments of the Hilbert curve immediately before and after the control region of node i.

The use of Hilbert sub-squares to address nodes in the Twins structure provides a dampening factor in reducing the effects of mobility. Local movements of nodes, *i.e.*, movements within their Hilbert sub-squares, do not have any impact on the Twins addressing structure. Moreover, nodes in sub-squares containing two or more nodes can freely move to other regions without generating much overhead. We will see in Section 6.3 that leave management operations are only necessary when nodes move out of their regions and cause the regions to become empty.

5. LOCATION SERVICE AND FORWARDING

In the following, we detail the register and lookup phases, and describe the forwarding decisions for data and management-related packets.

5.1 Location service

We use distributed hash tables (DHTs) to implement our location service. A standard linear congruential hash function, known by all nodes, is applied to the node identifier. Let us assume w.l.g. that an identifier is an m-bit number that uniquely identifies a node. In this way, a new node maps its identifier into a RP, *i.e.* a *derived-key* in the curve.[3] Observe that the new node does not know the actual identity of its RN. It simply knows that the management operations of the location service assure that there is a node whose control region holds the resulting hashed ID or *derived-key*. Then, using that derived-key as a destination's address, the new node sends a `Register` message indicating its location information. The way our location structure is constructed and managed guarantees the `Register` message will be correctly forwarded to the responsible RN (Section 6). The location information should remain unchanged unless the RN or the joint node moves.

The same hash operation applied to a destination's identifier is used by a source that does not know a destination's location. A `Lookup` message containing the hashed destination's ID, then, is sent to the resulting RP. The message will be correctly routed to the corresponding RN. Once having received the `Lookup` answer from the contacted RN, the source can directly communicate to the destination using the obtained Hilbert address, *i.e.,* n-point and derived-key, and/or the equivalent geographic coordinate.

A RN can manage multiple location identifiers, depending on its control region's size. How segments of the curve are distributed among nodes is the subject of the Section 6.

5.2 Forwarding table

When a node is assigned a Hilbert address and a control region, it also obtains information concerning its immediate neighbors in the logical space. This neighborhood information will compose its forwarding table. Every node in the system sends to its immediate one-hop neighbors a `Hello` message followed by periodic refreshes. `Hello` messages contain: (1) the node's current Hilbert address and/or the equivalent geographic coordinate, (2) the node's control region, and (3) the identification of the node's successor and predecessor on the Hilbert curve (*cf.* Section 6.2).

As a property of the position-based routing, the number of entries in the forwarding table is limited to $O($*number of single-hop radio neighbors*$)$ and the routing communication overhead to $O(1)$.

5.3 Forwarding operation

Forwarding in Twins can be implemented in two ways, depending on the type of the destination of the message. Data packets, for instance, are addressed to well-defined nodes (whose geographic coordinates are known), after the lookup phase.

5.3.1 Forwarding toward geographically-defined nodes

Destination of `Data` packets should be precisely defined, *i.e.* they should have a well defined position in the topology. Thus, the header of any `Data` packet contains the destination's `ID` as well as its Hilbert address/geographic coordinate. The next hop is determined based on the hierarchy of

[3]Note that since both the identifier and the hash function are invariant, the resulting RP does not change.

squares where the destination is located. The destination's Hilbert address in its *n-point* form identifies the destination location in each hierarchy of squares of the mathematical addressing space. The greedy forwarding algorithm is then applied by each hop between the source and the destination.

5.3.2 Forwarding toward a RP/RN

Using geographic routing to send messages to RPs is not adequate because RPs are logical entities, and may not be occupied by a node. Recall that messages addressed to a RP must be received by the node whose control region contains RP, *i.e.* the corresponding RN. In such a situation, forwarding must be implemented based on the control regions of the nodes.

Twins defines as control packets the `Register`, `Lookup`, and `Discover` (defined later in Section 6.2) messages, which should be region-based forwarded. In such a method, next-hops are determined based on the control region information of physical one-hop neighbors and on their identification of successor and predecessor on the curve, instead of their position in the geographic space. Note that forwarding is not performed following the curve. Packets are routed following segments of the Hilbert curve owned by one-hop neighbors in the physical network. In this way, region-based forwarding also exploits the advantages of the geographic network area. Thus, dead-ends can be addressed in the same way as in case of forwarding toward geographically-defined nodes.

At each hop, a node looks for an immediate neighbor (1) whose control region contains the target RP address or (2) that gets the message closer to the segment of the Hilbert curve that contains the target RP. The RN of the routed message will be the node whose control region holds (1) the RP of the new node, in case of `Register` messages, (2) the RP of the destination, in case of `Lookup` messages, or (3) the *derived-key* of the new node, in case of `Discover` messages.

We take advantage of the superior clustering property of Hilbert curve [3] in preserving the locality between points in the multidimensional space and in the linear space. Packets will then follow segments of close physical nodes. Section 7 shows how the distance that control packets travel compares with the geographic distance in hops between the source and the destination.

6. MANAGEMENT OPERATIONS

One of the biggest challenges imposed by our addressing system is how to easily manage it, while assuring the SONs' design requirements discussed in Section 2. More specifically, the challenge is to consistently partition the defined logical space in order to (a) associate potential RPs to nodes, and, consequently, (b) delegate their function as RNs. We manage to solve both issues using the Hilbert space-filling curve. Recognizing the complexity involved with a multidimensional partitioning, a bijective mapping between points in our hierarchical addressing space and points on a curve allows the application of simple one-dimensional location method.

Nodes joining the network control segments of the curve and only maintain information about the nodes that control the segments immediately before and after their own one. With this information, at the node departure and independently of the physical position of the nodes, the location service assures a simple re-assignment of the abandoned segment.

The following subsections describe the management operations when nodes join or leave the network. The described operations will be only performed by the first node (in the case of join operation) and by the last node (in the case of leave operation) of a Hilbert sub-square. Otherwise, nodes (1) that are joining the network get their control regions and the associated databases by communicating to neighbors in the same Hilbert sub-square, since they all share the same control region, or, (2) that are leaving only discard the previous managed databases, since these databases are replicated in all the remainder nodes within the same sub-square.

Twins succeeds to put the following features together:

- Decoupled logical and geographic addressing structures; this allows for robust multidimensional forwarding and easy network partitioning/management of the addressing space.
- Two forwarding modes that are robust to dynamic changes in the network.
- Easily-managed and simple location service, due to the use of a line segment (the designed Hilbert curve), instead of a region of a space.
- Scalable, consistent, and topology-independent location service, where RPs selection is completely independent of the physical location of the nodes.
- Completely distributed partitioning of the logical space, and the guarantee that each partition of the space is always managed by at least one node.
- Low message exchange overhead for guaranteeing the consistency of the space partitioning; only two nodes, the predecessor and successor nodes on the curve, must be contacted, as we will shortly describe in Section 6.2.
- Fair distribution of control overhead among nodes in the network (see Section 7).

The following subsections present in more detail our location service mechanism's management operations.

6.1 Definitions

The segment under the control of node i is called the control region of node i, noted $\mathcal{R}_i$. Let us call the *predecessor* and the *successor* of node i the p_i and s_i respectively. Let $\mathbf{N}$ be the set of nodes in the network, k be the order of the Hilbert curve, n be the dimensionality of the logical space, $\mathcal{H}$ be the set of points in the highest order Hilbert curve, and H_j be the Hilbert address of node j in its derived-key form. The predecessor and the successor of node i are defined formally via:

$$H_{p_i} = \max_{H_j < H_i} H_j, \quad \forall j \in \mathbf{N},$$

$$H_{s_i} = \min_{H_j > H_i} H_j, \quad \forall j \in \mathbf{N},$$

subject to $p_i = null$ if $H_i = 0$, and $s_i = null$ if $H_i = 2^{kn} - 1$.

6.2 Management operations when nodes join

For the sake of generality, we consider in this paper the case where any node in the topology can play the role of a RN. Thus, when a new node joins the network and it is also the first node of a Hilbert sub-square, besides registering itself in its RN, it must also be assigned a control region, *i.e.,* a segment of the Hilbert curve.

Node i obtains a control region from its predecessor and successor (and only from them). Thus, the first step for i is to determine which nodes are its predecessor and successor. It simply sends a `Discover` message with its own derived-key as the destination address. This message is forwarded through the region-based forwarding algorithm until it is received by the node whose control region contains the corresponding derived-key. This node is clearly either the successor or the predecessor of i. Once one of them identified, the other is automatically obtained.

Let $\mathcal{R}_i = [X_i, Y_i]$ be the region to be assigned to i, and p_i and s_i be its respective predecessor and successor nodes.

The lower and upper limits $[X_i, Y_i]$ are defined as

$$\begin{aligned} X_i = \quad & H_{p_i} \quad + \quad \left\lceil \frac{H_i - H_{p_i}}{2} \right\rceil \quad + \quad 1, \\ Y_i = \quad & H_i \quad + \quad \left\lceil \frac{H_{s_i} - H_i}{2} \right\rceil . \end{aligned} \tag{1}$$

The regions under the control of p_i and s_i, as well as their respective predecessor and successor nodes must also be updated. The new values are: $Y_{p_i} = X_i - 1$, $X_{s_i} = Y_i + 1$, $p_{s_i} = s_{p_i} = i$.

Once the control region assignment finished, the new node knows the exact identities of its predecessor and successor. This information is useful for maintaining the consistency of the space sharing. Observe that when a node joins the network, only those two other nodes are affected (*i.e.,* management-related packets are exchanged with), independent of the total number of nodes in the network.

By construction, the region assignment mechanism has three important properties. First, at any time, the highest order Hilbert space-filling curve is completely divided among the nodes, *i.e.,*

$$\mathcal{H} = \bigcup_{k \in \mathbf{N}} \mathcal{R}_k. \tag{2}$$

Second, if each derived-key in the space-filling curve is assigned to exactly one node, their control regions are mutual exclusive:

$$\mathcal{R}_i \bigcap \mathcal{R}_j = 0, \qquad \forall i, \forall j, i \neq j. \tag{3}$$

Finally, the control region of a node always holds its own Hilbert address, independently of how control regions are assigned to nodes:

$$H_i \in \mathcal{R}_i, \qquad \forall i. \tag{4}$$

6.3 Management operations when nodes leave

When the only occupant of a sub-square leaves, the control region it managed must be taken over by another node. A node explicitly hands over its control region and the associated database to its predecessor and/or to its successor node. Any criterion can be applied to select the node(s) which take over an abandoned control region. In this paper, we apply the One-sided Merging Criterion (OMC), where the entire abandoned control region is assigned to the predecessor or the successor (the one that has the smallest control region). More specifically, let t_n denote the time of the n-th join or leave event, $V_k(n)$ denote the volume of the control region controlled by node k at time t_n, $\forall k \in \mathbf{N}$. Then, the assignment is done as follows:

$$\min\{V_{p_i}(n), V_{s_i}(n)\} += \quad V_i(n). \tag{5}$$

The merging of control regions guarantees the continuity among segments of the Hilbert curve assigned to nodes. In this way, the predecessor and the successor nodes should update their information about the nodes responsible for regions spanning the segments immediately after and before their own control region. The neighbors of these predecessor and successor nodes, as well as the neighbors of the leaving node, also update their forwarding table accordingly.

7. PERFORMANCE EVALUATION

In this section we evaluate some fundamental properties related to Twins's management operations specified in Section 6, both analytically and by simulation. We present simulation results for Twins that show how well it scales. Good scaling means that the amount of work each node performs does not rise quickly as a function of the total number of nodes. The metrics we use are the number of location database entries each node must store and the number of control packets each node receives or generates in order to deal with a given workload. These metrics are dependent on how fair the curve partitioning among nodes is. The simulations show for various scenarios the fairness of our protocol.

Recall that region-based forwarding is based on the segments of the Hilbert curve assigned to the neighborhood of a node. We evaluate then the region-based path length over the geographic path length. Our results show that this cost scales well with the number of nodes. We also provide some results that illustrate the importance of correctly dimensioning the space-filling curve, and its effects on the fairness of the Twins protocol.

7.1 Why fairness?

As we have seen in Section 6, a generic node i is at all times assigned a control region. For each node under its control (*i.e.*, for each node in the Hilbert curve segment under its management,) node i will receive, from time to time, one of the following types of management-related packets:

- One `Register` and one `Discover` message every time a node joins its control region.
- One `Register` and one `Discover` message every time a node moves and needs to be assigned a new control region and to update its new location information.
- A random number of `Lookup` messages from any node present in the system, that has data to send to a destination whose RP is in the control region of node i. Such messages are sent every time a node executes, if necessary, the locating part of the data transfer protocol described in Section 5.1.

In this way, the *volume* of the control region assigned to a node represents a good measure of the overall control overhead that the node incurs. Note that the overhead associated with each type of management-related packets increases (in the stochastic sense), as the volume of the control region increases.

Assuming that our approach uses a consistent hashing[4] [19] to assign `IDs` to RPs, the guarantee of fair distribution of control overhead among nodes infers the guarantee of the fairness property of our management operations. Nevertheless, due to the uncontrolled randomness in the nodes' join/leave behavior, fairness guarantee is not evident.

Based on these observations, in the following subsections, we evaluate the fairness property of our proposal by theoretical analysis and simulation experiments.

7.2 Fairness analysis

As discussed above, for our fairness analysis we will focus on the volume of the control region. More precisely, let $t_1, t_2, \ldots, t_n, \ldots$ denote the sequence of node join and leave time instants. The volume of the control region, $V_p(n)$, controlled by node p at time t_n, is measured in number of nodes. Based on the notation introduced in Section 6, we can write $V_p(n) = Y_p - X_p$, where, abusing notation a little, Y_p and X_p denote the upper and lower values of the Hilbert curve segment under the control of node p, calculated at time t_n, via (1) or (5).

For the remainder of this subsection, we focus on two generic nodes, p_1 and p_2. We assume for simplicity that these nodes remain in operation for ever. Let $V_{p_1}(n)$ (respectively $V_{p_2}(n)$) denote the volume of the control region controlled by node p_1 (respectively p_2), at time t_n. In this subsection we will investigate whether the control region assignment algorithm defined by the One-sided Merging Criterion (OMC) in Section 6.3 results in a "fair allocation" of the control overhead among nodes p_1 and p_2. The next subsection shows the fairness property obtained when the OMC is applied.

At time t_{n+1}, one of the following events will occur:

$$\begin{aligned}
BL &\triangleq \{\text{a neighbor of both } p_1, p_2 \text{ leaves}\}\\
BJ &\triangleq \{\text{a neighbor of both } p_1, p_2 \text{ joins}\}\\
P1L &\triangleq \{\text{a neighbor of only } p_1 \text{ leaves}\}\\
P1J &\triangleq \{\text{a neighbor of only } p_1 \text{ joins}\}\\
P2L &\triangleq \{\text{a neighbor of only } p_2 \text{ leaves}\}\\
P2J &\triangleq \{\text{a neighbor of only } p_2 \text{ joins}\}\\
O &\triangleq \{\text{a node joins or leaves but is not a neighbor of } p_1 \text{ or } p_2\}
\end{aligned}$$

Let $I(n)$ and $I'(n)$ be random variables that denote the volume of the control region of a node that leaves at time t_n in the case of a BL and $P1L$ or $P2L$ event, respectively. Let $D(n), D'(n)$ denote two generic random variables such that (in the almost sure sense): $D(n) < V_{p_1}(n)$ and $D'(n) < V_{p_1}(n)$.

The control region assignment operations we have specified in (1) and (5) give rise to a sequence of random variables $\{V_{p_1}(n)\}_{n=1}^{\infty}$, that satisfies the following relationship:

$V_{p_1}(n+1) =$	*Condition*
$V_{p_1}(n) + I(n)$,	if event was BL and $V_{p_1}(n) \le V_{p_2}(n)$
$V_{p_1}(n)$,	if event was BL and $V_{p_1}(n) > V_{p_2}(n)$
$V_{p_1}(n) - D(n)$,	if event was BJ and p_1 assigns it a volume $D(n)$ of its control region
$V_{p_1}(n) + I'(n)$,	if event was $P1L$ and p_1 receives its control region
$V_{p_1}(n) - D'(n)$,	if event was $P1J$ and p_1 assigns it a volume $D'(n)$ of its control region
$V_{p_1}(n)$,	all other events

A similar expression can be written for node p_2. The first equality in the above described sequence says that, under the specified control region assignment operation, when a node that is a common neighbor (in the Hilbert curve sense) of both nodes p_1 and p_2 leaves, its control region is allocated to node p_1, and not p_2, since node p_2 carries at the time a higher volume. The remaining expressions can be explained in a similar fashion.

Note that the random variables $I(n)$, $I'(n)$, $D(n)$, $D'(n)$ depend on the network dynamics assumptions and $V_{p_1}(n)$, $V_{p_2}(n)$ only. Suppose, for simplicity of presentation, that nodes join and leave according to a stochastic process that is independent of the "history" of the processes $\{V_{p_1}(n)\}_{n=1}^{\infty}$ and $\{V_{p_2}(n)\}_{n=1}^{\infty}$. From the defined sequence of random variables, we can then easily see that the sequence $\{V_{p_1}(n)\}_{n=1}^{\infty}$ is a Markov Chain (MC). We will use in this section a drift analysis of this chain to intuitively define and explain the fairness properties of our control region assignment algorithm.

The volumes $V_{p_1}(n), V_{p_2}(n)$ can be represented as a point in the two-dimensional plane, shown in Fig. 4. As n varies, such a point moves randomly inside the triangular region bounded by the horizontal axis, the vertical axis and the line described by the equation $\sum_{i\in\mathbf{N}} V_{p_i}(n) = |\mathbf{N}|$. On this plane, the straight line $V_{p_1}(n) = V_{p_2}(n)$ is called the "fairness line", since all points on this line describe control region allocations with exactly equal volumes. The triangular region above the fairness line is "unfair" to node p_1, since $V_{p_1}(n) > V_{p_2}(n)$.

Suppose that at the time instant, t_n, of the n-th event, the volumes are described by point C in the figure. Let us determine some MC drifts at this point.

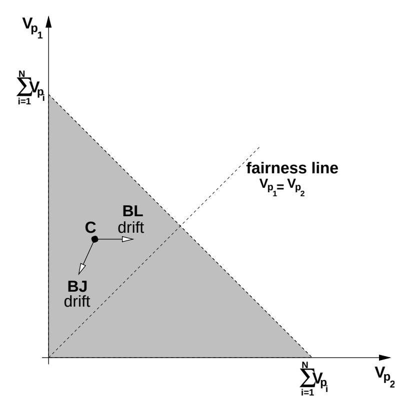


Figure 4: Drift analysis and fairness properties for the BJ and BL events.

When the event is BL, since $V_{p_1}(n) > V_{p_2}(n)$, node p_2 will be assigned the control region of the leaving node; thus, $V_{p_1}(n+1)$ remains unchanged and $V_{p_2}(n+1) = V_{p_2}(n) + I(n)$.

Therefore, for such an event, the point $(V_{p_1}(n+1), V_{p_2}(n+$

[4]A technique that, with high probability, balances load among nodes.

1)) will thus move (*i.e.*, drift) toward the fairness line. The drift for the event BJ is also indicated in Fig. 4 and can be explained in the similar fashion. The key observation is that, on average, the unfairness to node p_1 has been improved, *i.e.*

$$\lim_{n\to\infty} EV_{p_1}(n) = \lim_{n\to\infty} EV_{p_2}(n). \quad (6)$$

7.3 Simulation results

We have conducted a number of simulation experiments using the OPNET Modeler v10.5 simulator. All the nodes have a transmission range of 300 meters and run the IEEE 802.11 MAC protocol in the ad hoc mode. Nodes are uniformly distributed in a square universe. The size of the universe is chosen such that the average node density is about 150 nodes per square kilometer. The square universe is partitioned into a grid hierarchy defined by the Hilbert curve.

We are interested in the mobility model described by T. Henderson *et. al* in [20]. The authors show that in many situations (*e.g.*, campus wireless networks), users spend almost all their time in their home location, or remain at a single location for long periods of time. Moreover, they observed that laptop-equipped users follow a "ping-pong" model, where they arrive in a region of the campus, stop, open their laptops, connect to the network, and then, when they have finished, disconnect and leave. Following these mobility characteristics, network dynamics in our experiments is modeled by two parameters: the probability of nodes being selected to leave the network (noted α) and the rejoin time interval (noted τ). In this way, nodes are uniformly selected to leave the network, and once selected, they may wait for a time interval before rejoining at a new random location in the network. We use the OMC merging criterion when nodes rejoin (*cf.* Section 6.3).

7.3.1 Fairness results

This section analyzes join/leave operations and the resulting fairness aspects of our proposal. Fig. 5 shows the average volume of the control region as a function of the 16000 simulation seconds, in a 1000-node network and a 4096-position Hilbert curve. The results show the variation of the average volume for 10 nodes that have joined the network at different and distributed simulation times. In these experiments, the fairness is evaluated in scenarios under different degrees of α and variable node density in the network.

In the first graph of Fig. 5, nodes have low probability of leaving the network ($\alpha = 5\%$), where node density is maintained constant. In other words, nodes rejoin the network immediately after the hand over of their control region ($\tau = 0$). The second and third graphs show the results obtained with varying nodes' density and degree of nodes' dynamics. With high probability of leaving the network ($\alpha = 60\%$), nodes selected to leave wait 25 seconds outside the network before rejoining it.

The graphs verify that Twins achieves fair distribution of control region's volume among the nodes. In the beginning, when only a few nodes are present in the network, volumes are "unfairly" distributed. As the network size grows with time, the volumes converge to equal and thus fair values, in all scenarios shown in Fig. 5. We also observe that larger values of α increases the rate of convergence (of course, there is no convergence for $\alpha = 0$). The same behavior of convergence was observed for all nodes we do not show in the figures, but that are implicitly represented in the histogram

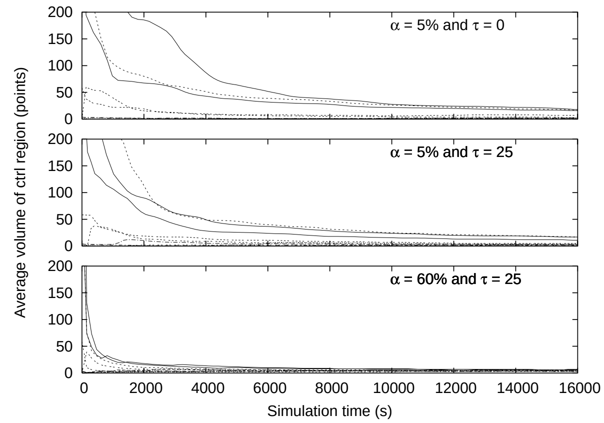


Figure 5: Average volume of the control region as a function of the simulation time, for: 1000 nodes, variable degrees of network dynamics, and variable node density.

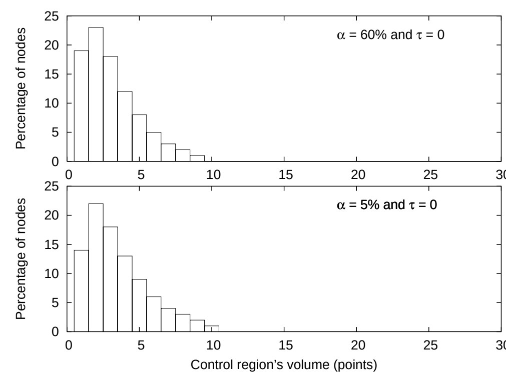


Figure 6: Histogram of the Hilbert curve partitioning among nodes.

shown in Fig. 6. The graphs show a large concentration of nodes with the same control region's volume, which proves that our results agree with the theoretical fairness analysis.

7.3.2 Curve dimensioning

In this section, we evaluate the effect that the dimensioning of the Hilbert curve has on the Twins performance. Our simulations show the influence of the occupation degree of the Hilbert curve on the Twins's fairness guarantee. Fig. 7 shows the effect of the degree of the occupation degree on the fairness convergence of Twins for 30000 simulated seconds. The experiments were performed in a universe where the probability for a node to leave the network is $\alpha = 20\%$ and a fixed node density. For 10 selected nodes, the results show a faster fairness convergence when the ratio of $\frac{1024}{450}$ of nodes' occupation is used.

The histogram of the Fig. 8 shows in more details the effects of the degree of nodes' occupation in the Hilbert curve on the Twins fairness guarantee. The results were obtained under the same conditions that produced Fig. 7. Once again, the smaller the ratio between the size of the Hilbert curve and the total number of nodes, the larger is the concentration of nodes with the same control region's volume.

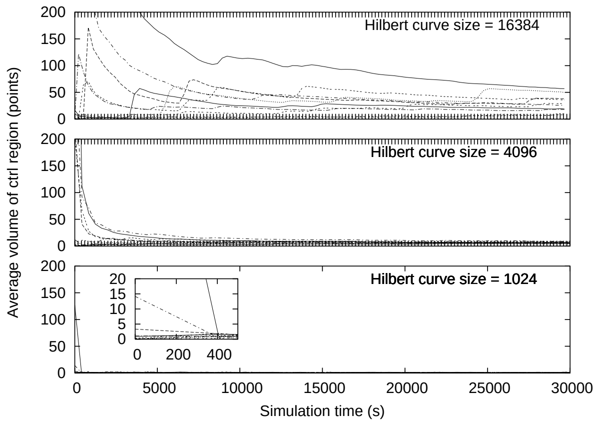


Figure 7: The effect of the occupation degree on the fairness convergence of Twins's management operations.

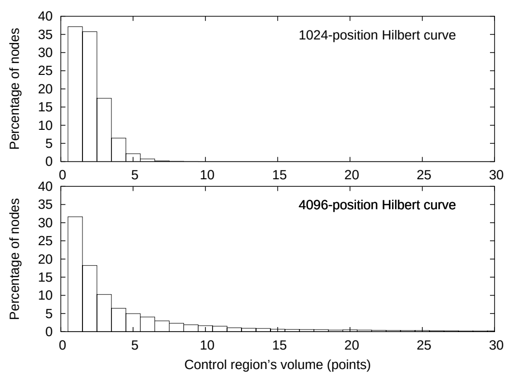


Figure 8: Histogram of the Hilbert curve partitioning among nodes for different occupation degrees.

7.3.3 *Region-based forwarding vs. geographic forwarding*

We evaluate now the distance that control packets travel using the region-based forwarding algorithm compared to the geographic distance for the same source-destination pair. For each node, we record the total number of region-based forwarding hops that each `Register` message traverses. We average the hop lengths for all `Register` messages sent during the simulation time.[5] Fig. 9 shows the average path stretch as a function of the total number of nodes. The path stretch is defined as the average region-based path length over the average geographic path length. The results were obtained for a 256-position Hilbert curve over a 2000-second simulation. The results show that Twins's region-based forwarding is very close to the geographic-based one, independently of the network size.

Since the `Lookup` and `Discover` messages are also routed based on the control region of the neighbors, the comparison results obtained for `Register` messages are also valid for them. In particular, `Lookup` packets are forwarded using the region-based forwarding, while their replies are sent directly to the source using geographic forwarding.

[5]Recall that nodes send a `Register` message each time their positions change.

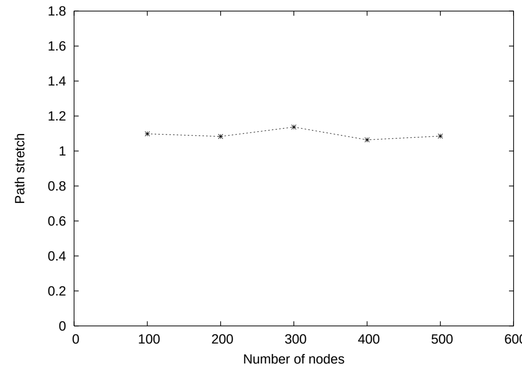


Figure 9: Path stretch *vs.* network size.

It is important to stress that the region-based forwarding is based on the segments of the Hilbert curve assigned to the geographic immediate neighbors and not on neighbors on the one-dimensional localization plane. This latter limits the neighborhood of a node to two and the path length to $O(N)$, where N is the number of nodes in the network. Fig. 10 shows the average path length obtained for the two Twins's forwarding mechanisms as well as for the curve-based forwarding mechanism (used here only for comparison purposes). For each node, we show the average number of forwarding hops that each `Register` message traverses during the simulation time. The simulation involves 400 nodes in a square universe partitioned into a grid hierarchy defined by 256-position Hilbert curve. Observe that region-based forwarding is not performed following the curve (shown as curve-based forwarding on the figure). The graph shows that the distance traveled by a control message using the Twins region-based forwarding is very close to the bound given by geographic forwarding.

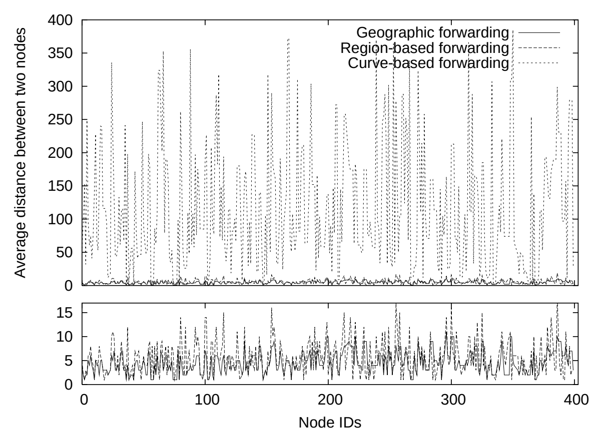


Figure 10: Average path length for a 400-node network, $\alpha = 15\%$, and $\tau = 0$. For the sake of clarity, the small graph at the bottom of the figure represents a zoom of the first graph's y-axis (between 0 and 15).

8. CONCLUSIONS AND FUTURE WORK

The design of SONs requires an efficient tradeoff between robustness and complexity. Focusing on this observation, we proposed: (a) two separate representations of the addressing space (logical and geographic) in a SON, and (b) a scalable location service that is independent of the spatial distribution of nodes over the topology and that is in accordance with the principles of SONs.

The theory of Hilbert curves has been around for a while; their application in SONs is novel. The presented architecture, Twins, decouples the data and control plane operations of the network, and assures the topology-independence of our easily-managed location service. With respect to the design requirements for SONs, the main strengths of Twins are: routing that is robust to dynamic networks, simplified management operations, simple forwarding decisions, low communication cost, scalability, distributed management overhead, and fair allocation of routing overhead. Moreover, Twins allows inherent replication of location information and increased robustness to failures, without generating any additional overhead to the system. These characteristics are due to placing physical neighbor nodes in the same Hilbert region and the direct communication among them.

In this paper, we provided an evaluation of the fairness of the control overhead distribution via theoretical and simulation analysis. We examined the effects of various system parameters, such as occupation degree of the Hilbert curve, network density, and network variability. Our results show that Twins exhibits strong fairness properties, even though the rate of convergence to the fair state is variable. We also evaluated the performance of our region-based forwarding algorithm. We observed that our proposal scales well and does not incur extra overhead when compared with the bounds obtained by the geographic-based approach.

Future work will include: (a) a more detailed evaluation of the Twins protocol overhead and the influence of the curve dimensioning, (b) an evaluation of the location path performance, and, (c) an evaluation of mechanisms to deal with merge and split operations of the entire network.

9. REFERENCES

[1] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of ACM Mobicom*, (Dallas, TX), Oct. 1998.

[2] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of ACM Mobicom*, (Boston, MA), Aug. 2000.

[3] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the hilbert space-filling curve," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 124–141, Jan. 2001.

[4] J. P. Hubaux, T. Gross, J. Y. L. Boudec, and M. Vetterli, "Towards self-organized mobile ad hoc networks: the terminodes project," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 118–124, Jan. 2001.

[5] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of ACM Mobicom*, (Boston, MA), Aug. 2000.

[6] Y. Xue, B. Li, and K. Nahrstedt, "A scalable location management scheme in mobile ad-hoc networks," in *Proceedings of IEEE Conference on Local Computer Networks (LCN)*, (Tampa, FL), Nov. 2001.

[7] Z. Xu, M. Mahalingam, and M. Karlsson, "Turning heterogeneity into an advantage in overlay routing," in *Proceedings of IEEE Infocom*, (San Francisco, CA), Mar. 2003.

[8] A. C. Viana, M. D. Amorim, S. Fdida, and J. F. Rezende, "Self-organization in spontaneous networks: the approach of DHT-based routing protocols," *To appear in Ad Hoc Networks Journal*, 2005.

[9] J. Ridoux, A. Fladenmuller, Y. Viniotis, and K. Salamatian, "Trellis-based virtual regular addressing structures in self-organized networks," in *Proceedings of IFIP Networking*, (Waterloo, Canada), pp. 511–522, May 2005.

[10] B. Chen and R. Morris, "L+: Scalable landmark routing and address lookup for multi-hop wireless networks," tech. rep., Massachusetts Institute of Technology, Cambridge, Massachusetts - MIT LCS Technical Report 837 (MIT-LCS-TR-837), Mar. 2002.

[11] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Scalable ad hoc routing: The case for dynamic addressing," in *Proceedings of IEEE Infocom*, (Hong Kong), Mar. 2004.

[12] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmayer, "Space-filling curves and their use in the design of geometric data structures," *Theoretical Computer Science*, vol. 181, no. 1, pp. 3–15, July 1997.

[13] B. Moghaddam, K. J. Hintz, and C. V. Steward, "Space-filling curves for image compression," in *Automatic Object Recognition*, 1991.

[14] L. K. Platzman and J. J. B. III, "Space filling curves and the planar travelling salesman problem," *Journal of the Association for Computing Machinery*, vol. 36, no. 4, no. 4, pp. 719–737, 1995.

[15] D. Hilbert, *Ueber Stetige Abbildung Einer Linie auf ein Flachenstuck.* Mathematische Annalenn, 38:459-460., 1891.

[16] G. Breinholt and C. Schierz, "Algorithm 781: Generating hilbert's space-filling curve by recursion," *ACM Transactions on Mathematical Software*, vol. 24, no. 2, pp. 184–189, June 1998.

[17] E. D. Kaplan, *Understanding GPS: Principles and Applications.* Artech House Telecommunications Library, 1996.

[18] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," in *Proceedings of the 34th HICSS*, Jan. 2001.

[19] D. R. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web," in *Proceedings of ACM Symposium on Theory of Computing*, (El Paso, TX), May 1997.

[20] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proceedings of ACM Mobicom*, (Philadelphia, PA), Oct. 2004.