Twins: Easy-Managed Location in Self-Organizing Networks Using Hilbert Space-Filling Curves

Aline C. Viana[1,3], Marcelo D. de Amorim[1], Yannis Viniotis[2], Serge Fdida[1], and José F. de Rezende[3]

[1] Laboratoire LIP6/CNRS – Université Pierre et Marie Curie – Paris, France
[2] Dept. of ECE – North Carolina State University – Raleigh, NC, USA
[3] COPPE/Poli – Universidade Federal do Rio de Janeiro – Rio de Janeiro, Brasil

Abstract

The need for efficient location mechanisms is an important issue in self-organizing networks where topologies become larger and more complex in their addressing structures. In this paper, we propose Twins, an easy-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 logical space is used for addressing and routing, while location is performed in an easy-managed one-dimensional structure obtained by the use of Hilbert space-filling curves. The Twins location service follows a rendezvous-based communication abstraction, where node location information is distributed throughout the topology. Routing is performed in a hop-by-hop basis with greedy next-hop choice, which requires only a limited amount of signaling information. In this paper, we focus on fairness evaluation of various Twins space management operations. Our results show that Twins assures a fair distribution of control overhead among nodes.

Index Terms

System design, location service, self-organizing networks.

I. 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, sensor, and wireless mesh networks are examples of such networks.

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 mobility of nodes. Additional peculiarities include the possible lack of geographical positioning infrastructure, the limited and variable capacity of wireless links, and the energy-constrained nature of some nodes. The two main characteristics impose two fundamental requirements on the design/operation of a SON: (a) all nodes in the SON must assume the same management responsibilities, and, (b) any network operation, *e.g.*, management of the address space, should be inherently distributed.

Self-organization introduces new, radically different technical requirements. We present detailed design requirements in Section II. 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, and, (c) defining a packet forwarding mechanism.

This work has been supported by CAPES, COFECUB, CNRS, and EuronetLab. Contact email: Aline.Viana@lip6.fr.

Much research work has been done in all three components. The proposed solutions meet the design requirements mentioned in Section II to varying degrees. In the traditional Internet model, routing information is embedded into the topological-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 (even if they are not SONs), 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 the specific case of ad hoc networks. Nevertheless, it has been observed that these architectures do not scale well beyond a few hundred nodes [1], [2]. For instance, in sensor or wireless mesh 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 mobility [3], [4]. 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.

In this paper, we propose *Twins*, a novel architecture to perform easy-managed location in SONs. 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 adjacencies between these partitions. Twins defines then a multi- to one-dimensional mapping of this logical space, which allows the location service to be performed in an easy-managed one-dimensional structure. The need for efficient location mechanisms is even more important in an environment where topologies become larger and more complex in their addressing structure. Multidimensional Cartesian spaces are robust to mobility but introduce complexity to the management of the space partitioning. This is because they are completely dependent on the network topology, which is dynamic and unpredictable in the case of SONs. In this way, our goal is to obtain a compromise between robustness and complexity. Twins allows geographic routing to be performed in a logical space, while a physical-position-independent location service is implemented over a one-dimensional and thus much easier to manage address 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 II, the main technical advantages of our proposed architecture and associated management operations are scalability, simplicity of management and fairness. Our approach draws its scalability advantage from the fact that it is instantiated as a rendezvous communication abstraction where nodes' location is completely distributed throughout the network. 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 physical-position-independence of the location service. 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 geographical routing.

The paper is organized as follows: in Section II we present in more detail the specific design requirements for establishing an addressing space and routing (data forwarding and management) in a SON. In Section III we present the concept of Hilbert curves and discuss how it can be used to establish, in a distributed way, a structured addressing space for a SON. In Section IV we present how the location service can be provided, using the indirect model and the rendezvous abstraction; we discuss in detail how the data forwarding operations of the model can be implemented with the new concept of Hilbert addresses. In Section V we discuss the required operations in the management plane, which ensure that the mobility of SON nodes does not disrupt network operations. Finally, in Section VI we present an initial evaluation of Twins. We focus on the fairness of the proposed address management algorithm; we study three different management schemes and show via a theoretical and simulation study that all of them result in fair distribution of control overhead.

II. SONs' DESIGN REQUIREMENTS

Due to their spontaneous and dynamic characteristics, SONs require the support of a scalable and easy-managed addressing/location service. 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* [5], [6], [7]. The idea consists of completely distributing node location information in location servers throughout the topology. In the following, we focus on the main issues that must be addressed in SONs and in indirect routing.

A. Requirements

- *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 responsibilities should be completely distributed among the nodes in the network.
- *Flexibility in route selection and mobility management:* the addressing structure should offer flexibility in route selection. This issue has an impact on the mobility 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 address 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.

B. Indirect routing and the role of the addressing space

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 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 [5], [6], [7]. Furthermore, geographic or position-based routing has proven to be an efficient solution to address the new requirements introduced by SONs [3], [4].

Twins system employs a DHT-based location service completely independent of the physical location of the nodes and implements a position-based routing. The physical-position-independence of our location service is based on the mapping between nodes' identifier to *rendezvous points*, and on the association of these rendezvous points to *rendezvous nodes*.

In our proposal, each node has an identifier, which is in turn hashed into a *rendezvous point* in an addressing space. Partitions of this addressing space are assigned to nodes in the network. A node whose partition contains a rendezvous point is called the *rendezvous node* and is responsible for storing the present location information of the node associated with that rendezvous point.

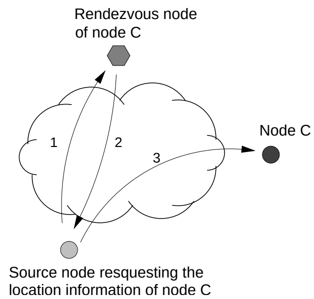


Fig. 1. Lookup (arrows 1 and 2) phase and direct communication (arrow 3) in a DHT-based indirect routing procedure.

The basic operation in our location service is Lookup(*rendezvous point*), which returns the node controlling the partition of the space corresponding to that point. More specifically, let C be a node and $l(C)$ be the present location of C. Let also $p(i)$ be the partition of the logical addressing space under the responsibility of node i. Node i will be the rendezvous node of node C if and only if $h(\mathtt{ID}_C) =$ *rendezvous point*$(C) \in p(i)$, where $h(\cdot)$ is a predefined hash function and $\mathtt{ID}_C$ is the identifier of node C. Nodes that just join the SON or nodes that have just moved into a new region register themselves by sending their location information to a corresponding rendezvous node. Thus, when a source node wants to retrieve C, it must first determine which node stores $l(C)$. The first step is then to locate C's rendezvous node by performing $h(\mathtt{ID}_C)$, which results clearly in the same *rendezvous point*(C). A request message is then routed through the partitions of the logical addressing space until it is received by C's rendezvous node, the node that controls the partition that contains *rendezvous point*(C) (cf. arrow 1 in Fig. 1). The rendezvous node responds with a message containing $l(C)$ (arrow 2), which allows the destination to be directly contacted (arrow 3).

As it can be observed, the addressing space has an important role in the design of the location service, and should be properly partitioned among nodes. In the same way, in order to benefit from the position-based routing's strengths, a mapping between geographical coordinates and points in the designed addressing space should be established. In the following sections, we present in more detail how Twins addresses these issues.

III. Address Spaces based on Hilbert curves

The implementation of indirect routing using geographic coordinates has already been addressed in a number of research proposals [5], [6], [7]. These approaches have interesting characteristics but present a common problem in the way the location service is implemented. In these systems, the logical address space is completely correlated with the geographic space; this introduces some loss of freedom in the location service. Rendezvous points are in fact geographic coordinates in the physical space. This requires the presence of at least one node near this point to play the role of the rendezvous node, which cannot be guaranteed in a SON because of its inherent variability (in time and in space).

The solution we propose is to decouple the location service from the geographic coordinates of the nodes (without, however, creating an overlay network). 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 III-B. We will also see in detail in Section V that the use of this logical structure is the key for obtaining both scalability and low management complexity of the addressing space.

A. The logical addressing space

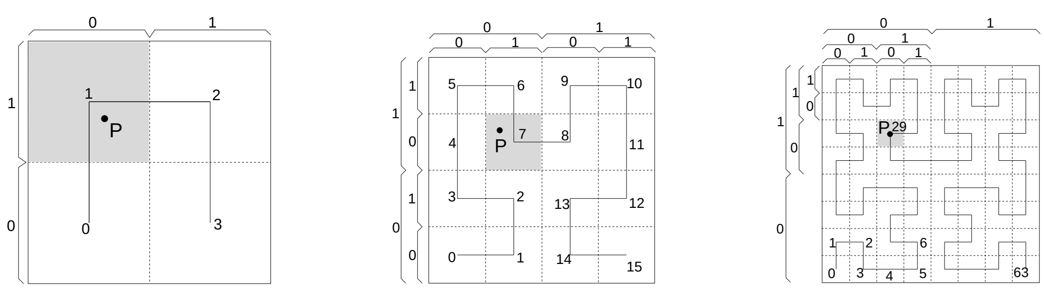
Let $\mathcal{G}$ and $\mathcal{L}$ be, respectively, the geographic and logical spaces of the SON. The first step is to define a function that maps node addresses in $\mathcal{G}$ into node addresses in $\mathcal{L}$. The objective is to use the resulting logical distribution of nodes to implement the 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 rendezvous nodes will be the subject of Section V.

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. With this assumption, the mapping function becomes a simple combination of scaling, rotation, and translation operations (cf. Subsection III-C).

Before detailing our approach, let us argue on the importance of associating logical to geographic spaces. 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 rendezvous points. 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 the Twins solution, we use a one-dimensional representation of the logical space, through a mathematical technique called *space-filling curves*. A space-filling curve defines a bijective mapping between points in a Cartesian product space and points on a line. Many solutions for obtaining a space-filling curve have been proposed in the literature. We describe in the following the mapping we adopt in this paper, namely *Hilbert Space-Filling Curve*.

B. Hilbert space-filling curve

This concept emerged in the 19th century and is originally credited to Peano [8] in 1890. The first graphical or geometrical representation of the space-filling idea has been given by David Hilbert [9]. 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. 2 reproduces the figures shown in Hilbert's original paper, except that we number the points from 0 rather than from 1. Fig. 2(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. 2(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. 2(a). The re-orientations ensure that the adjacency property is preserved everywhere. Fig. 2(c) indicates the third step in the sequence.



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

Fig. 2. 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.

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. Refer to Lawder [10] for the general case algorithm.

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:

- *derived-key*: it represents the ordinal position of a point on a curve of any order.
- *n-point*: 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. 2(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. 2(a)), 2nd (Fig. 2(b)), and 3rd-order curves (Fig. 2(c)). For example, 01 is the n-point of the high-lighted square in Fig. 2(a).

C. Mapping geographic into logical spaces

The maximum number of nodes to be supported in the network will fix the value k of the highest order of the Hilbert curve to be designed. This highest order curve then, defines the mathematical structure to be used as our logical addressing space.

The first two nodes that join the network determine the association between the physical and the logical spaces. Let $\mathbf{N} = \{n_1, n_2, n_3, \ldots\}$ be the set of nodes in the network such that n_i has joined the network before n_j if $i < j$. When n_1 joins the network, the logical space is completely separated from the physical topology. By definition, this node occupies the center of the logical space. The arrival of this node fixes the origin of the logical plane but not the orientation, which is determined by the arrival of node n_2. We have now the orientation of the logical space, which makes the assignment of logical addresses trivial. The only requirement is to respect the relative geographic positions when placing other nodes on the logical structure. This can be achieved by using a GPS [11] or any relative positioning algorithm [12].

We can now obtain n-points from physical coordinates and, consequently, derived-keys from physical coordinates. From this point, every node 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.

IV. Location Service and Forwarding

Our location service is composed by a Register and a Lookup phase. In the following subsection, we detail these phases and describe the forwarding decisions for data and management-related packets.

A. 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 a m-bit number that uniquely identifies a node. In this way, a new node maps its identifier onto a rendezvous point, *i.e.* a *derived-key* in the curve.[1] Observe that the new node does not know the actual identity of its rendezvous node. It simply knows the management operations of the location service assure that there is a node whose control region, *i.e.*, its partition of the space, holds the resulting hashed *derived-key*. Then, using the derived-key as a destination address, the new node sends a `Register` message

[1] Note that since both the identifier and the hash function are invariant, the resulting rendezvous point does not change either.

indicating its location information. The way our location structure is constructed and managed guarantees the `Register` message will be correctly forwarded to the responsible rendezvous node (Section V). The location information should remain unchanged unless the rendezvous node 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 destination's `ID`, then, is sent to the resulting rendezvous point. [2] The message will be correctly routed to the corresponding rendezvous node. Once having received the `Lookup` answer from the contacted rendezvous node, 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 rendezvous node can manage multiple location identifiers. How segments of the curve are distributed among nodes is the subject of the Section V.

B. 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 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 (Section V-B).

We will see in the following that the number of entries in the forwarding table is limited to $O($*number of single-hop radio neighbors*$)$ and the routing communication overheard to $O(1)$.

C. Data and control packet forwarding

The header of any data packet contains the destination's `ID` as well as its Hilbert address/geographic coordinate. When a source node needs to forward a data packet toward destination D, the node consults its forwarding table and chooses one neighbor which is the closest to D. It then forwards the packet to that neighbor, which itself applies the same forwarding algorithm. This procedure is repeated until the destination is reached.

The next hop is determined based on the hierarchy of squares where the destination is located.[3] The destination's Hilbert address in its *n-point* form identifies the destination location in each hierarchy of squares of the mathematical addressing space.

Control packets (*i.e.* `Register` or `Lookup`) are forwarded in a slightly different manner. Next-hops are determined based on the control region information of immediate neighbors, instead of their position in the $\mathcal{G}$ and $\mathcal{L}$ spaces. The target node in that case will be the node whose control region holds the hashed `ID` of the new node – in case of `Register` – or of the destination – in case of `Lookup` messages.

The hashed `ID`, *i.e.* the resulting rendezvous point, is specified in the packet header. Recall that this point does not identify the exact position of the corresponding rendezvous node. At each hop, a node looks for an immediate neighbor whose control region contains the rendezvous point or that gets the message closer to the rendezvous point. Packets are then routed following successive control segments of the Hilbert curve instead of n-points.

V. 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 II. More specifically, the challenge is to

[2] For improved location information availability, one could assign multiple different rendezvous points to each node, each one located in a different grid hierarchy. `Lookup` queries can then be sent to the rendezvous point closest to the source in the hierarchy of grids.

[3] We do not consider in this paper the possibility of dead-ends. We simply suppose that such a situation can be addressed by efficient existing approaches like GPSR [3].

consistently partition the defined logical space in order to (a) associate potential rendezvous points to nodes, and, consequently, (b) delegate their function as rendezvous nodes. 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.

Twins succeeds to put the following features together:

- decoupled logical and geographical addressing structures; this allows routing advantages to be fully exploited;
- routing that is robust to mobility, due to the use of multidimensional addressing space;
- easy-managed and simple location service, due to the use of a line segment, instead of a region of a space;
- scalable, consistent, and physical-position-independent location service, where rendezvous points selection is completely independent of the physical location of the nodes;
- completely distributed partitioning of the logical space, which is independent of the network's density;
- 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 V-B;
- fair distribution of control overhead among nodes in the network (see Section VI).

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

A. Definitions

The segment under the control of node i is called the control region of node i, noted $\mathcal{R}_i$. On the Hilbert curve, two nodes perform a specific and crucial task. These nodes are the *predecessor* and the *successor* of node i. Let us call them p_i and s_i respectively. 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 node i control region's range. More formally, 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:

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

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

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

B. 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 rendezvous node. Thus, when a new node joins the network, besides registering itself in its rendezvous node, it must also be assigned a control region, *i.e.*, a segment of the Hilbert curve.

The segment of region to be under the control of a joining node i is obtained as follows. Let $\mathcal{R}_i = [X_i, Y_i] \subset \mathcal{H}$ be the region to be assigned to i, and p_i and s_i be its respective future predecessor and successor. The lower and upper limits X_i, Y_i are defined as

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

The regions under the control of p_i and s_i, as well as their new predecessor and successor information are updated as follows: $Y_{p_i} = X_i - 1$, $X_{s_i} = Y_i + 1$, $p_{s_i} = s_{p_i} = X_i$.

Every 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.

C. Management operations when nodes depart

When nodes leave the network [4] the control regions they managed are taken over by another node, since the location structure is distributed. A node, then, 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 Section VI-C, three different merging criteria are simulated in order to analyse the fairness of our proposal. The first simulated criterion, called TMC (Two-sided Merging Criterion), assigns a part of the abandoned control region to both the predecessor and the successor. In this case, their new control regions will be updated to:

$$\begin{aligned} Y_{p_i} = &\quad H_{p_i} \quad + \quad \left\lceil \tfrac{H_{s_i} - H_{p_i}}{2} \right\rceil , \\ X_{s_i} = &\quad Y_{p_i} \quad + \quad 1. \end{aligned} \tag{2}$$

In the second criterion, called OMC (One-sided Merging Criterion), the entire abandoned control region is assigned to the predecessor or the successor 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:

$$\min\{V_{p_i}(n), V_{s_i}(n)\} += \quad V_i(n). \tag{3}$$

Finally, the last criterion, called AMC (Average Merging Criterion), selects the node that controls the smallest average volume to be assigned the abandoned control region as follows:

$$\min\{\tfrac{1}{k}\textstyle\sum_{n=1}^{k} V_{p_i}(n), \tfrac{1}{k}\sum_{n=1}^{k} V_{s_i}(n)\} += \quad V_i(n). \tag{4}$$

The different merging criteria have been proposed in order to evaluate the one that achieves a more uniform partitioning of the space-filling curve over all the nodes in the network. Furthermore, sending the control region to a predecessor and/or a successor node also guarantees the continuity among segments of the Hilbert curve assigned to nodes.

Finally, the predecessor and the successor nodes 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 (Section IV-B).

Once a mobile node arrives at a new neighborhood, it restarts the address allocation and region assignment procedures. The node's `ID` and, consequently, the rendezvous point resulting from the hashing `ID`'s operation, remain unchanged. However, its received Hilbert address is dependent of its new location. This new location information is updated with a `Register` message at the correspondent rendezvous node, whose identification also remains unchanged.

[4]We assume the existence of some mechanism that allows a node to determine when it is leaving its location. Giving details about this mechanism is not our focus here.

D. Properties of the region assignment mechanism

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.*,

$$\mathcal{H} = \bigcup_{k \in \mathbf{N}} \mathcal{R}_k. \tag{5}$$

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

$$\mathcal{R}_i \bigcap \mathcal{R}_j = 0, \quad \forall i, \forall j, i \neq j. \tag{6}$$

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

$$H_i \in \mathcal{R}_i \quad , \quad \forall i. \tag{7}$$

E. Comparison to other approaches

Terminodes [5], [13], GLS [6], and DLM [7] are examples in the literature of DHT-based geographic routing systems. In Terminodes, each node advertises its current position to location servers placed in a computed geographical region (Virtual Home Region - VHR) of radius R. The parameter R is increased or decreased if the number of nodes in the VHR is not adequate. This density dependency forces the protocol to check the entire region periodically and changes location servers accordingly. Besides that, nodes in a low density network will need to scan and advertise large regions in order to avoid empty VHRs.

In the GLS proposal, for each hierarchy of regions, each node employs location servers with `ID`s closest to its own `ID`. Each GLS node, then, needs to potentially scan the entire region to find out which node has the closest `ID`. Due to topology changes, this scan needs to be periodically performed in order to accordingly change the location servers.

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` to a set of physical regions in the network; selected nodes at those regions act as the location servers of the node. 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.

In Twins system, rendezvous nodes are selected based on the hashed rendezvous point and can be any node in the network. Its management operations assure the consistency of the system and, consequently, the correct forwarding of messages toward the corresponding rendezvous node. Thus, a node does not need to have any information about the identity or location of the searched rendezvous node, when sending `Register` or `Lookup` messages.

Changes in the topology (*i.e.*, mobility of the nodes) generate changes in the space's partitioning among the nodes, and consequently in the identity of their rendezvous nodes. [5] Twins guarantees the correct partitioning update among the nodes and the new rendezvous nodes selection in an adaptive and automatic manner. The new rendezvous node(s) of a node is automatically determined based on the new updated partitioning, without any intervention or processing at the node. Thus, a node does not need to scan the network to determine its new rendezvous node 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.

[5] The hashed rendezvous point of a node `ID` never changes, but the segment of the curve that contains it may be assigned to different nodes due to topology changes.

VI. Performance evaluation

In this section we evaluate some fundamental properties related to Twins' management operations specified in Section V, both analytically and by simulation. We focus on fairness of the curve partitioning among nodes.

A. Why fairness?

As we have seen in Section V, 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 three types of management-related packets:

- One `Register` message every time a node joins its control region.
- One `Register` message every time a node moves and needs 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 rendezvous point is in the control region of node i. Such messages are sent every time a node executes the locating part of the data transfer protocol described in Section IV-A.

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* [6] [14] to assign IDs to rendezvous points, 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.

B. 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 V, 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), (2), (3), or (4).

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. Due to a lack of space, in this subsection we will only investigate whether the control region assignment algorithm defined by the One-sided Merging Criterion (OMC) in Section V-C results in a "fair allocation" of the control overhead among nodes p_1 and p_2. The next subsection, however, shows the fairness property obtained when the three described merging criteria are applied.

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

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

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

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 (3) 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 $V_{p_1}(n) \leq V_{p_2}(n)$
$V_{p_1}(n),$	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 mobility 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. 3. 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's determine some MC drifts at this point.

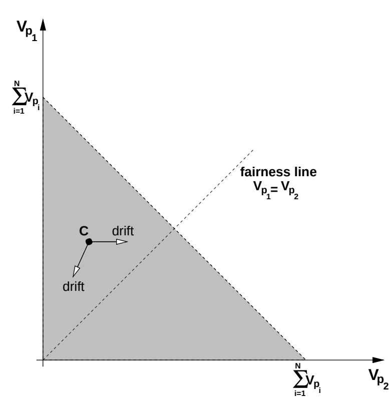


Fig. 3. Drift analysis and fairness properties.

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

$$V_{p_2}(n+1) = V_{p_2}(n) + I(n). \tag{8}$$

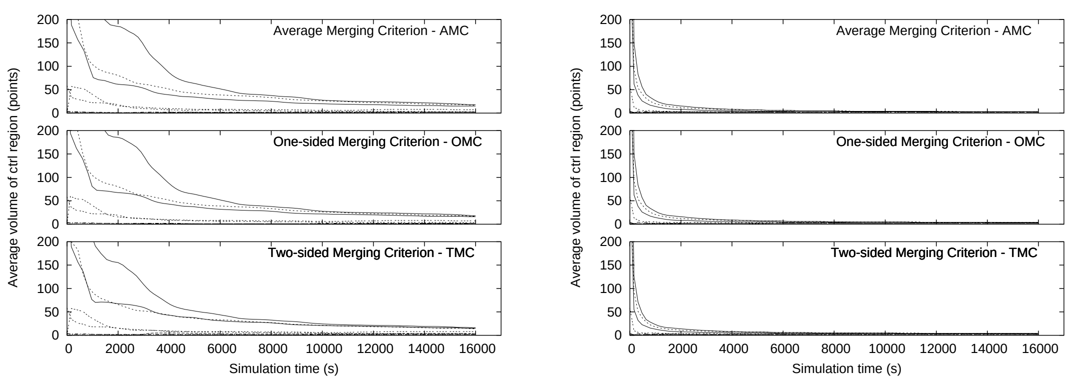
Therefore, for such an event, the point $(V_{p_1}(n+1), V_{p_2}(n+1))$ will thus move (*i.e.*, drift) towards the fairness line. The drift for the event BJ is also indicated in Fig. 3 and can be explained in the similar fashion. The key observation is that, on average, the unfairness to node p_1 has been improved. For a more rigorous analysis, see [15], where we show that

$$\lim_{n\to\infty} EV_{p_1}(n) = \lim_{n\to\infty} EV_{p_2}(n). \tag{9}$$

C. Simulation results

We have conducted some preliminary simulation experiments applying our management operations, under relatively ideal scenarios. Using the OPNET Modeler v10.5 simulator, each node's transmission range has a 300 meters radius. All nodes run the IEEE 802.11 MAC protocol in the ad hoc mode, i.e. no central base station is required. Each simulation runs for 30.000 simulated seconds. The nodes are placed at uniformly random locations in a square universe of 6400 meters on side. This square universe is partitioned into the grid hierarchy defined by a 6th order Hilbert Curve with squares of 100 meters on a side.

This work studies only the join and leave operations and resulting fairness aspects of our proposal, without any data traffic. During the simulation time and based on a specified probability, nodes are uniformly selected to leave the network. Simulations show the results obtained when the selected nodes apply one of the three proposed merging criteria (Section V-C). After handing over its control region and the associated database, each node may wait a time interval before joining a new random location in the network. Thus, nodes' mobility in our experiments is modeled by two parameters: the probability of nodes being selected to leave the network (called α) and the rejoin time interval (called τ).



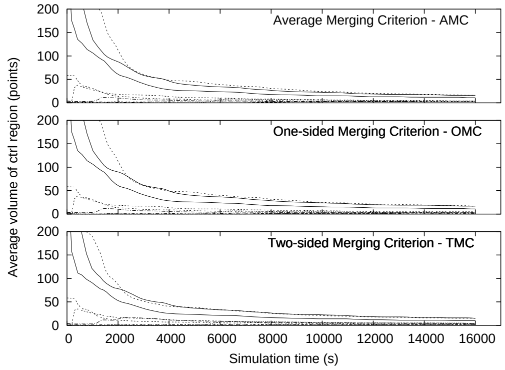
(a) $\alpha = 5\%$ and $\tau = 0$.

(b) $\alpha = 60\%$ and $\tau = 0$.

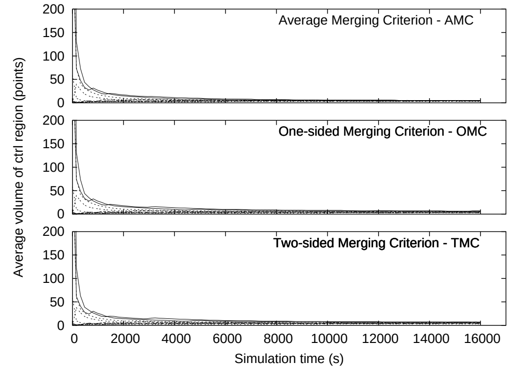
Fig. 4. Average volume of the control region as a function of the simulation time, for: 1000 nodes, variable degrees of nodes' mobility, and constant nodes' density.

For each proposed merging criterion, Fig. 4 and Fig. 5 show the average volume of the control region as a function of the simulation time, in a 1000-node network and 4096-position Hilbert curve. The results show for all simulation times, the variation of the average volume for 10 nodes that have joined the network at different and distributed simulation times. The same 10 nodes are selected in all simulated experiments shown in Fig. 4 and Fig. 5. In these experiments, the fairness is evaluated in scenarios under different degrees of α (Fig. 4) and variable node density in the network (Fig. 5).

In Fig. 4(a), nodes have low probability of leaving the network ($\alpha = 5\%$) and nodes' density is maintained constant and equal to 1000 nodes. In other words, nodes rejoin the network immediately after the hand over of their control region ($\tau = 0$). Maintaining a constant network density, Fig. 4(b) shows the results obtained when $\alpha = 60\%$.



(a) $\alpha = 5\%$ and $\tau = 25$.



(b) $\alpha = 60\%$ and $\tau = 25$.

Fig. 5. Average volume of the control region as a function of the simulation time, for: 1000 nodes, variable degree of nodes' mobility, and variable nodes' density.

Fig. 5 shows the same experiments performed in Fig. 4, however, with variable nodes' density. Nodes selected to leave wait 25 seconds outside the network before rejoining it. Thus, the network density in each instant of time t is constantly changed with the departures and the rejoins of nodes.

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 scenario shown in Fig. 4 and 5. Each individual graph shows the fairness convergence of the control region's volume for 10 random nodes. The three evaluated merging criteria present a very similar behavior in each simulated scenario. We observe, however, that for constant or variable density of nodes in the network, the increase of α, increases the rate of convergence for all merging criteria. The same behavior of convergence was observed for all nodes we do not show in the figures.

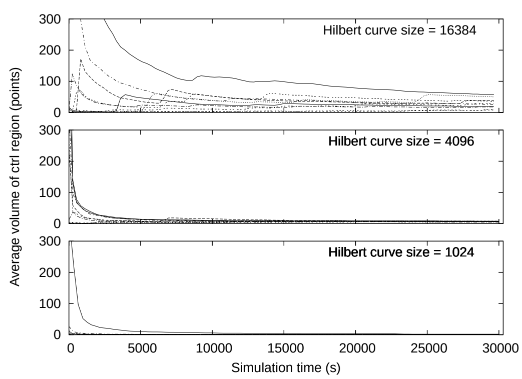


Fig. 6. The effect of the degree of nodes' occupation on the fairness convergence of Twins' management operations.

Another important parameter that exerts strong influence on our fairness guarantee is the degree of nodes' occupation in the Hilbert curve. The smaller the ratio between the size of the highest Hilbert curve and the total number of nodes, the faster the fairness convergence. Fig. 6 shows the effect of the degree of nodes' occupation on the fairness convergence of Twins. The experiments were performed in a 450-node network with: $\alpha = 20\%$, $\tau = 0$, under AMC criterion. For 10 selected nodes, the results show a faster fairness convergence when the ratio of $\frac{1024}{450}$ of nodes' occupation is used. On the graph for Hilbert curve size = 1024, the solid line describes the result for a node that has the biggest segment of the curve, when compared to the other 9 selected nodes. This node controls a boundary segment of the curve. This is why this node starts with an "unfair" initial volume compared to the other 9.

VII. Conclusions and Future Work

The design of SON architecture requires an efficient tradeoff between robustness and complexity. Focusing on this observation, we have proposed the use of (a) two separate representations of the addressing space (logical and geographical) in a SON, and, (b) Hilbert curves to map multidimensional to one-dimensional address space. 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 physical-position-independence of our easy-managed location service. With respect to the design requirements for SONs, the main strengths of Twins are: routing that is robust to mobility, simplified management operations, simple forwarding decisions, low communication cost, scalability, distributed management overhead, and fair allocation of routing overhead.

In this paper, we have provided an evaluation of the fairness of the control overhead distribution via theoretical and simulation analysis. We have examined the effects of various system parameters, such as, degree of nodes' occupation, network density, and network variability. Our results show that all three merging mechanisms exhibit strong fairness properties, even though the rate of convergence to the fair state is variable.

Future work will include (a) a more detailed comparison to other approaches, (b) a more detailed evaluation of control overhead (*e.g.,* size of routing tables, number of control messages exchanged), (c) an evaluation of the data path performance (*e.g.,* average data delay achieved under geographic routing), and, (d) an evaluation of the location path performance (*e.g.,* average management-related packet delay achieved under the Twins architecture).

Appendix

A. Modelling assumptions

For our fairness analysis, we will focus on the volume of the control region only. 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 Hilbert addresses under the control of node p. Based on the notation introduced in Section V, 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), (2), (3), or (4).

For the remainder of this section, 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 section we will investigate whether the control region assignment algorithm defined by the One-sided Merging Criterion (OMC) in Section V-C results in a "fair allocation" of the control overhead among nodes p_1 and p_2.

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

$$BL \triangleq \{a\ neighbor\ of\ both\ p_1, p_2\ leaves\} \quad (10)$$

$$BJ \triangleq \{a\ neighbor\ of\ both\ p_1, p_2\ joins\} \quad (11)$$

$$P1L \triangleq \{a\ neighbor\ of\ only\ p_1\ leaves\} \quad (12)$$

$$P1J \triangleq \{a\ neighbor\ of\ only\ p_1\ joins\} \quad (13)$$

$$P2L \triangleq \{a\ neighbor\ of\ only\ p_2\ leaves\} \quad (14)$$

$$P2J \triangleq \{a\ neighbor\ of\ only\ p_2\ joins\} \quad (15)$$

$$O \triangleq \{a\ node\ joins\ or\ leaves\ but\ is\ not\ a\ neighbor\ of\ p_1\ or\ p_2\} \quad (16)$$

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 (3) give rise to a sequence of random variables $\{V_{p_1}(n)\}_{n=1}^{\infty}$, that satisfies the following relationship:

$$V_{p_1}(n+1) = \begin{cases} V_{p_1}(n) + I(n), & \text{if event was } BL \text{ and } V_{p_1}(n) < V_{p_2}(n) \\ V_{p_1}(n) + I(n), & \text{if event was } BL \text{ and } V_{p_1}(n) = V_{p_2}(n) \text{ and a fair coin shows heads} \\ V_{p_1}(n), & \text{if event was } BL \text{ and } V_{p_1}(n) > V_{p_2}(n) \\ V_{p_1}(n) - D(n), & \text{if event was } BJ \text{ and } p_1 \text{ assigns a volume } D(n) \text{ to the neighbor} \\ V_{p_1}(n) + I'(n), & \text{if event was } P1L \text{ and } p_1 \text{ receives the neighbor's control region} \\ V_{p_1}(n) - D'(n), & \text{if event was } P1J \text{ and } p_1 \text{ assigns a volume } D'(n) \text{ to the neighbor} \\ V_{p_1}(n), & \text{all other events} \end{cases} \tag{17}$$

A similar expression can be written for node p_2. The first equality in Equation 17 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 a higher volume at the time. The second equality says that, when volumes are equal, node p_1 will get the allocation of the leaving node with probability 0.5. 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 mobility assumptions and $V_{p_1}(n), V_{p_2}(n)$ (but not $V_{p_1}(k), V_{p_2}(k)$, for $k \neq n$).

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 $\{V_{p_2}(n)\}_{n=1}^{\infty}$. A very simple example of such a process would be generated by iid arrivals (not necessarily Poisson) and iid sequence of times nodes spend in the system before they leave (not necessarily exponential).

From Equation 17, we can then easily see that the sequence of vectors $\{V_{p_1}(n), V_{p_2}(n)\}_{n=1}^{\infty}$ is a Markov Chain (MC). We will evaluate the properties of the MC in the next sections.

B. The state space of the MC

The volumes $V_{p_1}(n), V_{p_2}(n)$ can be represented as a point in the two-dimensional integer lattice, shown in Fig. 7. More specifically, the state space of the MC is the set S of integer-valued pairs (a, b) such that:

$$S \stackrel{\triangle}{=} \{(a, b) : 1 \leq a, 1 \leq b, a + b \leq |\mathcal{H}|\}$$

where the inequalities $1 \leq a, 1 \leq b$ are due to our assumption that nodes p_1 and p_2 are always present and hence their control region is never empty.

C. The fairness line

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 $V_{p_1}(n) > V_{p_2}(n)$.

Note that, because of the uncontrolled randomness in the nodes' join/leave behavior, it is not possible to force, via any control algorithm, the point $(V_{p_1}(n), V_{p_2}(n))$ to stay on the fairness line for ever.

D. Transitions of the MC

It is possible to move from any state of the MC to any other state, i.e., the MC has only one communicating class. Figures 8 through 11 depict the possible transitions in more detail, based on the event types.

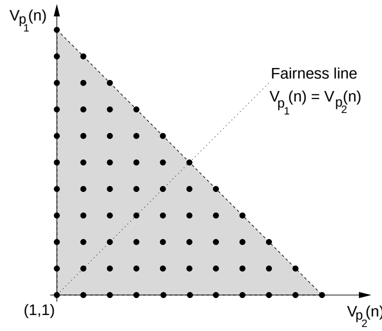


Fig. 7. The state space of the MC.

1) Transitions on the events $P1L$, $P1J$, $P2L$, $P2J$: Consider a generic state (a, b).

When a neighbor of only node p_1 leaves (i.e., event $P1L$), (part of) its control region will be assigned to node p_1, according to Equation 17, resulting in a "vertical, upward" transition shown in figure 8.

When a neighbor of only node p_1 joins (i.e., event $P1J$), node p_1 will assign part of its control region to the new neighbor, according to Equation 1, resulting in a "vertical, downward" transition shown in figure 8.

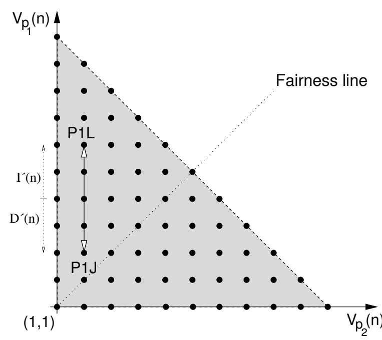


Fig. 8. Transitions on the events $P1L$, $P1J$.

Transitions due to the events $P2L$, $P2J$ result in entirely analogous, but "horizontal" transitions, as shown in figure 9.

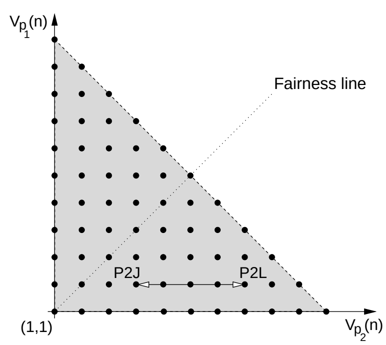


Fig. 9. Transitions on the events $P2L$, $P2J$.

2) Transitions on the event BL: Consider a generic state A, as shown in figure 10, in which $V_{p_1}(n) > V_{p_2}(n)$.

When a neighbor of both nodes p_1 and p_2 leaves (i.e., event BL), all of its control region will be assigned to node p_2, according to Equation 17, resulting in the "horizontal, right-ward" transition shown in figure 10.

Consider next a generic state B, as shown in figure 10, in which $V_{p_1}(n) < V_{p_2}(n)$. When a neighbor of both nodes p_1 and p_2 leaves, all of its control region will be assigned to node p_1, resulting in the "vertical, upward" transitions shown in figure 10.

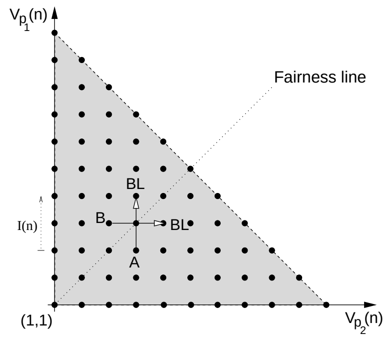


Fig. 10. Transitions on the event BL.

3) Transitions on the event BJ: Consider a generic state A, as shown in figure 11.

When a neighbor of both nodes p_1 and p_2 joins, both node p_1 and node p_2 will assign part of their control region to the new neighbor, according to Equation 1, resulting in a "diagonal, downward" transition shown in figure 11.

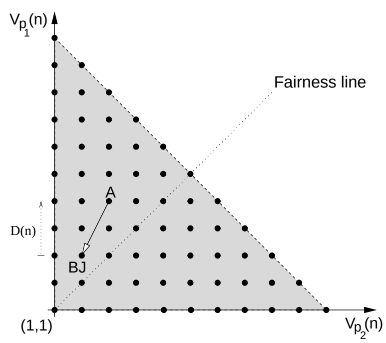


Fig. 11. Transitions on the event BJ.

In summary, any state can be visited from any other state, i.e., the MC has only one communicating class. This fact will be useful in the next section.

E. Fairness analysis

The main result of this paper is the following:

Theorem 1: Consider a Self-Organizing Network in which node joins and leaves occur according to iid processes. Suppose that the OMC (One-Sided Merging Criterion) in Equation 3 is in use. Then control overhead is distributed among nodes in a fashion that is fair in the long-term average sense, i.e.,

$$\lim_{n\to\infty} EV_{p_1}(n) = \lim_{n\to\infty} EV_{p_2}(n). \tag{18}$$

Proof: Proving Equation 18 involves the following two steps.

- **Step 1** We first show that the "steady-state" limits $\lim_{n\to\infty} EV_{p_1}(n)$ and $\lim_{n\to\infty} EV_{p_2}(n)$ exist. Note that, the OMC criterion describes a "dynamic and adaptive" algorithm, with state-dependent actions. As with all such algorithms, it is not clear, a priori, that such limits will exist. We use the fact that the MC describing the system is ergodic, to establish existence of the limits.

- **Step 2** We next show that the difference $\lim_{n\to\infty} E[V_{p_1}(n) - V_{p_2}(n)] = 0$. We use the fact that the random variable

$$\lim_{n\to\infty} X(n) \stackrel{\triangle}{=} \lim_{n\to\infty} V_{p_1}(n) - \lim_{n\to\infty} V_{p_2}(n) \tag{19}$$

has "symmetric" probabilities, in the sense that

$$P[\lim_{n\to\infty} X(n) = k] = P[\lim_{n\to\infty} X(n) = -k],$$

for all values of k, and, therefore, $\lim_{n\to\infty} EX(n) = \lim_{n\to\infty} E[V_{p_1}(n) - V_{p_2}(n)] = 0$.

Step 1. In section B, we have seen that the state space of the MC $(V_{p_1}(n), V_{p_2}(n))$ is finite; in section D we have shown that the chain has only one communicating class. Therefore, the chain is ergodic; the limits

$$\begin{aligned} EV_{p_1} &\stackrel{\triangle}{=} \lim_{n\to\infty} EV_{p_1}(n) \\ EV_{p_2} &\stackrel{\triangle}{=} \lim_{n\to\infty} EV_{p_2}(n) \end{aligned}$$

exist. Moreover, with probability 1, they are equal to the time averages:

$$\begin{aligned} EV_{p_1} &= \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} V_{p_1}(n) \\ EV_{p_2} &= \lim_{k\to\infty} \frac{1}{k} \sum_{n=1}^{k} V_{p_2}(n) \end{aligned}$$

Step 2. Consider next the random variable $X(n)$ in Equation 19. The range of $X(n)$ is the set S_X defined by:

$$S_X \stackrel{\triangle}{=} \{-|\mathcal{H}| + 2, -|\mathcal{H}| + 3, \ldots, -1, 0, 1, \ldots, |\mathcal{H}| - 3, |\mathcal{H}| - 2\}.$$

Note that the sequence $\{X(n)\}_{n=1}^{\infty}$ is also a Markov Chain. In order to see this, consider the events

$$\{X(n) = k_n\}, \quad \{X(n) = k_n, X(n-1) = k_{n-1}, \ldots, X(0) = k_0\}$$

Suppose that $k_n > 0$ (the other cases are similar). Since $V_{p_1}(n) > V_{p_2}(n)$, Equation 17 becomes

$$V_{p_1}(n+1) = \begin{cases} V_{p_1}(n) + I(n), & \text{if event was } BL \\ V_{p_1}(n) - D(n), & \text{if event was } BJ \text{ and } p_1 \text{ assigns a volume } D(n) \text{ to the neighbor} \\ V_{p_1}(n) + I'(n), & \text{if event was } P1L \text{ and } p_1 \text{ receives the neighbor's control region} \\ V_{p_1}(n) - D'(n), & \text{if event was } P1J \text{ and } p_1 \text{ assigns a volume } D'(n) \text{ to the neighbor} \\ V_{p_1}(n), & \text{all other events} \end{cases} \tag{20}$$

Of course,

$$V_{p_2}(n+1) = V_{p_2}(n) + I(n) \tag{21}$$

and thus

$$X(n+1) = X(n) - I(n).$$

Note that the random variable $I(n)$ depends on $X(n)$ only (not $X(n-1)$, $X(n-2)$, etc.). Therefore, given $X(n)$, the "history" of the process $\{X(n)\}_{n=1}^{\infty}$ up to time $n-1$ has no effect on the random variable $X(n+1)$.

The transition probabilities $P[X(n+1)=k|X(n)=l]$ have an interesting "symmetry" property, that is key to our analysis. More specifically, they satisfy the equation:

$$P_{k,l} \triangleq P[X(n+1)=k|X(n)=l] = P[X(n+1)=-k|X(n)=-l] = P_{-k,-l},\ \forall k,l \in S_X. \tag{22}$$

For example, when S_X has 5 elements, the transition probability matrix has the structure shown in Equation 23. Note the "symmetry" in the third row, which represents transitions out of state 0.

$$\begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & l & k \\ j & i & h & g & f \\ e & d & c & b & a \end{bmatrix} \tag{23}$$

Intuitively, this property is a consequence of the "symmetry" of the rules specified in equation 17. Let's verify the property in equation 22 more precisely, in two specific examples.

Example 1. Suppose that $l > 0, k = l+1$. Therefore, due to the event that happened at time instant t_{n+1}, the difference, $X(n+1)$, has increased by one.

Among the events listed in equation 10, the only possible ones to happen at time t_{n+1} are $P1L$, $P2J$ and BJ. Therefore, we can write

$$P[X(n+1)=l+1|X(n)=l] = P[P1L \cup P2J \cup BJ|X(n)=l], l > 0 \tag{24}$$

Consider next the case $l < 0, k = -(l+1)$. Therefore, due to the event that happened at time instant t_{n+1}, the difference, $X(n+1)$, has decreased by one. Among the events listed in Equation 10, the only possible ones to happen at t_{n+1} are again $P1L$, $P2J$ and BJ. Therefore, we can write

$$P[X(n+1)=-(l+1)|X(n)=-l] = P[P1L \cup P2J \cup BJ|X(n)=-l], l > 0 \tag{25}$$

The conditional probabilities in the right-hand sides of Equations 24 and 25 are equal. Indeed, the events $\{X(n)=l\}$ and $\{X(n)=-l\}$ are unions of MC states along two diagonals parallel to the fairness line. Consider two generic states A and B, shown in figure 12, that are "symmetric" across the fairness line. Since the sum of the volumes $V_{p_1}(n)$ and $V_{p_2}(n)$ is the same in both states A and B, the event $P1L \cup P2J \cup BJ$ occurs with the same probability. (Note that this probability depends on the state A. For example, in states C and D, the events $P2J$ or BJ cannot occur, when all nodes are present in the system.) $\triangle$

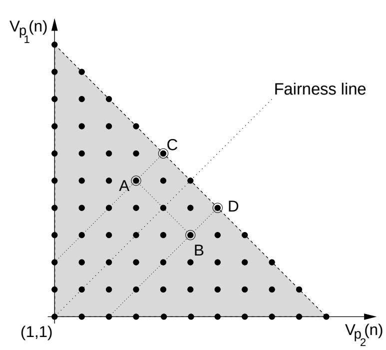


Fig. 12. Equality of conditional probabilities.

Example 2. Suppose that $l > 0, k = l - 1$. Therefore, due to the event that happened at time instant t_{n+1}, the difference, $X(n+1)$, has decreased by one.

Among the events listed in Equation 10, the only possible ones to happen at t_{n+1} are BL, $P1J$, $P2L$ and BJ. Therefore, we can write

$$P[X(n+1) = l - 1 | X(n) = l] = P[BL \cup P1J \cup P2L \cup BJ | X(n) = l], l > 0 \tag{26}$$

Consider next the case $l < 0, k = -(l - 1)$. Therefore, due to the event that happened at time instant t_{n+1}, the difference, $X(n+1)$, has increased by one. Among the events listed in Equation 10, the only possible ones to happen at t_{n+1} are again BL, $P1J$, $P2L$ and BJ. Therefore, we can write

$$P[X(n+1) = -(l - 1) | X(n) = -l] = P[BL \cup P1J \cup P2L \cup BJ | X(n) = -l], l > 0 \tag{27}$$

We can show that the conditional probabilities in the right-hand sides of Equations 26 and 27 are equal, using an entirely analogous argument as in Example 1. $\triangle$

We investigate next the limiting behavior of this MC. Let X denote a random variable that represents the steady state of the MC. The steady-state transition probabilities are, of course, unique and satisfy the equations

$$P[X = k] = \sum_l P_{k,l} P[X = l] \tag{28}$$

(and, of course, the equation $\sum_k P[X = k] = 1$). We show next that a *symmetric* distribution, i.e., one with the property

$$P[X = k] = P[X = -k], \ \forall k \in S_X,$$

satisfies equation 28. Using equation 22, we can write

$$\begin{aligned} P[X = k] &= \sum_l P_{k,l} P[X = l] \\ &= \sum_l P_{-k,-l} P[X = l] \\ &= \sum_l P_{-k,-l} P[X = -l] \\ &= \sum_l P_{-k,l} P[X = l] \\ &= P[X = -k] \end{aligned} \tag{29}$$

verifying the claim. Note that, unless the initial distribution $P[X(0) = l]$ is itself symmetric, the transient probabilities $P[X(n) = l]$ are not, in general, symmetric.

This concludes the proof that the difference is a symmetric random variable. Its expected value, will, of course be 0, concluding the proof of the theorem. $\triangle$

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*, Oct. 1998.

[2] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information wireless ad hoc networks," *IEEE Personal Communications*, pp. 48–57, Feb. 2001.

[3] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of ACM Mobicom*, Aug. 2000.

[4] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Tech. Rep., SI Res. Rep. ISI/RR-87-180, Univ. Southern California, Los Angeles, Mar. 1987.

[5] L. Blazevic, S. Giordano, and J. Y. Boudec, "Self organized terminode routing," *Journal of Cluster Computing*, vol. 5, no. 2, Apr. 2002.

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

[7] 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, USA, Nov. 2001.

[8] Giuseppe Peano, *Sur une coube, qui remplit toute une aire plane (On a curve which completely fills a planar region).*, Mathematische Annalenn, 38:157-160., 1890.

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

[10] Jonathan Lawder, "The applications of Space-filling Curves to the storage and retrieval of multi-dimenstional data," Tech. Rep. JL/1/99, Birkbeck College, University of London, 1999.

[11] Elliott D. Kaplan, *Understanding Gps: Principles and Applications*, Artech House Telecommunications Library, 1996.

[12] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proceedings of ACM Mobicom*, Apr. 2003.

[13] J. P. Hubaux, Th. Gross, J. Y. Le 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.

[14] 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, Texas, United States, May 1997, pp. 654–663.

[15] A. C. Viana, M. D. Amorim, Y. Viniotis, S. Fdida, and J. F. Rezende, "Fairness analysis of the Twins control overhead," Tech. Rep., CACC, North Carolina State University, Raleigh, USA, July 2004.