A Survey on Network Coordinates Systems, Design, and Security

Benoit Donnet†, Bamba Gueye*, Mohamed Ali Kaafar‡
† Université catholique de Louvain, CSE Department – Louvain-la-Neuve, Belgium
∗ Université de Liège – Liège, Belgium
‡ INRIA Rhone-Alpes – Grenoble, France

***Abstract*— During the last decade, a new class of large-scale globally-distributed network services and applications have emerged. Those systems are flexible in the sense that they can select their communication path among a set of available ones. However, ceaselessly gathering network information such as latency to select a path is infeasible due to the large amount of measurement traffic it would generate. To overcome this issue, Network Coordinates Systems (NCS) have been proposed. An NCS allows hosts to predict latencies without performing direct measurements and, consequently, reduce the network resources consumption. During these last years, NCS opened new research fields in which the networking community has produced an impressive amount of work. We believe it is now time to stop and take stock of what has been achieved so far. In this paper, we survey the various NCS proposed as well as their intrinsic limits. In particular, we focus on security issues and solutions proposed to fix them. We also discuss potential future NCS developments, in particular how to use NCS for predicting bandwidth.**

I. Introduction

As innovative ways are being developed to harvest the enormous potential of Internet infrastructure, a new class of large-scale globally-distributed network services and applications such as distributed overlay network multicast [1], [2], content addressable overlay networks [3], [4], and peer-to-peer file sharing such as Gnutella [5], OceanStore [6], BitTorrent [7], [8], etc. have emerged. To achieve network topology-awareness, most, if not all, of these overlays rely on the notion of proximity, usually defined in terms of network delays or round-trip times (RTTs), for optimal neighbor selection during overlay construction and maintenance.

Because these systems have a lot of flexibility in choosing their communication paths, they can greatly benefit from intelligent path selection based on network performance. Collecting up-to-date performance measurements between nodes in an overlay network would be very beneficial for those applications. Especially, in a wide-area network, communication performances have a significant impact on the overall execution time of operations.

For instance, in a peer-to-peer file sharing application, a client ideally wants to know the available bandwidth between itself and all the peers that have the desired file. Proximity-aware distributed hash tables would use latency measurements to reduce the delay stretch of lookups [9]. Content distribution systems would construct network-aware trees to minimize dissemination times [10]. Decentralized web caches need latency information to map clients to cache locations, or to guide the selection of a download server from multiple replicas. And finally, a topology knowledge would allow the construction of efficient multicast delivery trees.

Nevertheless, path performance measurements require to inject probes in the network, burdening so the network and leading to an inadmissible measurement cost: one measurement per pair of nodes and the number of pairs is a quadratic function of the number of nodes. For example, re-directing clients to the nearest data centers would require *Google* to maintain latency from virtually every Web client in the Internet to each of its data centers [11]. Moreover, obtaining the information can exceed the cost of the effective process [12], [13], [14]. In other words, performance measurement is not scalable.

It is important for the new applications presented above to limit the resources consumption and particularly the number of on-demand measurements. In such a context, *Network Coordinates Systems* (NCS) have been proposed to allow hosts to estimate delays without performing direct measurements and thus, reduce the consumption of network resources. The key idea of an NCS is to model the Internet as a geometric space and characterize the position of any node in the Internet by a position (i.e., a *coordinate*) in this space. The network distance between any two nodes is then predicted as the geometric distance between their coordinates. Explicit measurements are, therefore, not anymore required.

Content distribution and file sharing systems can benefit from network coordinates in order to select a number of replicated servers to fetch data from. Azureus [15] (now called *Vuze*), for instance, was the first large-scale real world application to use a coordinates system. In addition to choosing the closest replicated server, reducing the overall length of client to server network paths allow one to localize the communication, leading to lower backbone and inter-ISP link utilization. Similar benefits can be achieved for other large scale distributed applications such as peer-to-peer overlays or online gaming platforms. Oasis [16], a distributed anycast system, is shared across several application services and makes use of network coordinates to amortize deployment and network measurement costs.

This paper, in which we review the different coordinates-based embedding techniques that have been featured in literature so far, is intended to be a single point of reference for researchers interested in NCS.

In this paper, we begin by describing a few proposed works

that do provide network proximity or location estimates, but do not rely on "virtual" coordinates embedded into geometric spaces (Sec. II). Such an introduction is intended to underline the inconvenient usage of a so-called "direct measurement systems" and hence to present the benefits of NCS usage (Sec. III). Then, we concentrate on describing network coordinates systems that fit within the class of landmark-based approaches or more generally centralized systems (Sec. IV-A). We present different distributed coordinates-based systems for network positioning (Sec. IV-B). We also discuss the limitations inherent to NCS and explain how they might be overtaken (Sec. V) and focus in particular on security issues (Sec. VI). We then present potential future directions for NCS, in particular how they can be used to predict bandwidth (Sec. VII). We finally conclude this paper by reminding its main contributions (Sec. VIII).

II. Localization Techniques

Several approaches in the literature provide network proximity or location estimates using either direct pair-wise measurements, or by supplying directly applications with network distances estimates. In contrast to network coordinates systems, these approaches do not attempt to globally model Internet hosts positions using absolute coordinates, but rather, most of them try to contribute in specific application needs, such as special peer lookups, clustering, etc. In the following, we underline the most known approaches that have been proposed for locating network nodes. In Sec. II-A, we focus on the Global Positioning System, a positioning system based on satellites. Sec. II-B addresses geolocation techniques, i.e., determining the physical location of an Internet host . Finally, Sec. II-C describes Meridian, a framework for finding nearest peers in overlay networks.

A. Global Positioning System

The *Global Positioning System* (GPS) [17] is a positioning system based on satellites. Basically, the GPS performs the localization through the computation of the distance separating a GPS receiver and several satellites.

The receiver uses the arrival time of each message to measure the distance to each satellite, from which it determines the position of the receiver (conceptually the intersection of spheres). The resulting coordinates are converted to more user-friendly forms such as latitude and longitude, or location on a map, then displayed to the user.

Since the space has three dimensions, one might think that using three satellites would be enough to calculate the position of a receiver. However, this would require the time to be very accurate (i.e., on a nanosecond scale), which is very difficult to achieve outside a laboratory. Using four or more satellites allows one to get rid of this clock accuracy need.

B. Geolocation approaches

Many works have been proposed for inferring the geographical location of network nodes, rather than the Internet positions (e.g., in a latency space). Geolocation approaches [18] are intended to provide where the host is in the real world, whereas the network coordinates systems intend to provide relative positions between host in terms of network distances (e.g., latency). In other words, for geolocating hosts, distances refer to actual geographic distances between hosts. In contrast, distance in the context of network coordinates systems refers to the network delay between a pair of Internet hosts.

Throughout the years, several techniques have been proposed for locating Internet hosts. They can be roughly classified into four groups: *database*, *DNS*, *clustering*, and *delay measurements*.

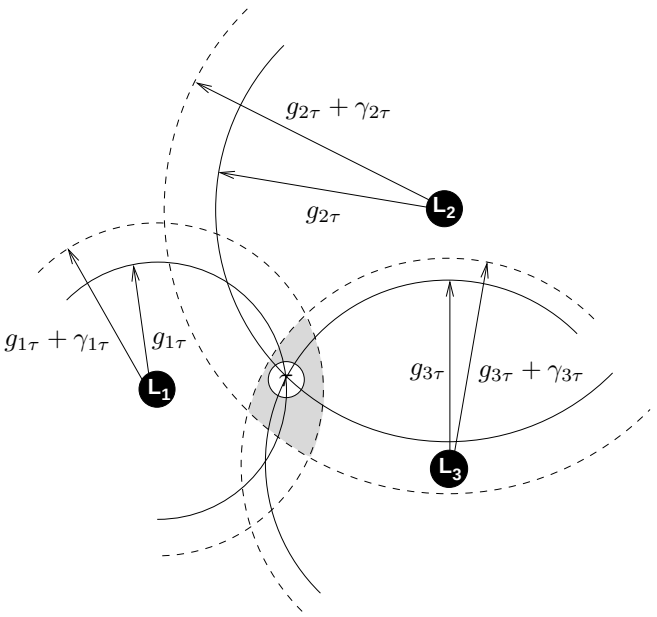


Fig. 1. Multilateration with geographic distance constraints

Database techniques, such as *IP2LL* [19], make use of Whois data to infer geographic location information. DNS techniques, such as *GeoTrack* [20], infer an host location from names provided by the targeted host DNS or the routers close to it. It assumes that the DNS name provides location information at various granularity levels (city, state, or country). Clustering techniques (see, for instance, *GeoCluster* [20]) are based on the notion of *cluster*, i.e., a group of clients that are topologically close and under the same administrative control. Hosts belonging to the same cluster are said co-located. If one knows the geographic location of a few hosts within the cluster, one might infer the location of the whole cluster. Finally, delay measurement techniques, such as *GeoPing* [20] or *Constraint-Based Geolocation* (CBG) [21], try to exploit a correlation between the delay and the geographic distance. For instance, CBG infers the geographical location of network nodes using *multilateration*. Multilateration refers to the estimation of a point position using a sufficient number of distances (geographical distances in our case) to some fixed points whose positions are known. Geographical distances to the landmarks are deduced from the correspondent delay distances (obtained by direct probing between the landmarks and the target host) by relying on the assumption that digital information travels along fiber optic cables at almost exactly

$2/3$ the speed of light in a vacuum. Basically, given the geographical locations of the landmarks and their geographical distances to a given target host, an estimation of the location of the target host is achieved using multilateration.

An example of multilateration is shown on Fig. 1. Plain circles depict the actual geographical distance, while dashed circles refer to the distance obtained when transforming the RTT into geographical distance. There is a distance overestimation, leading to the creation of confidence zone in which the host will be found.

C. Meridian Approach

Wong et al. proposed a framework, called *Meridian* [22], for hosts to lookup their nearest peers in an overlay network. The key idea in Meridian is to construct a multi resolution ring structure that guides requests sent through this structure to nodes that are closer and closer to the sender. Basically, each Meridian node keeps track of a small, fixed number of other hosts that are organized and maintained in a ring structure with exponentially growing ring radii. When a node sends a query, for its nearest peer, such a query is forwarded along the ring structure, which exponentially reduces the distance to the target at each query hop.

In contrast to coordinates-based systems, Meridian acts as an on-demand nearest node look-up service, and focuses more on individual nodes requests, rather than building a global coordinates service, that would allow for multiple distances estimations.

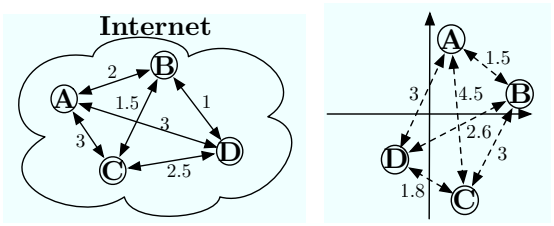
The following section discusses the major challenges of using direct measurements services.

D. Main drawbacks

The different approaches we discussed above try to solve either distance prediction or topology-aware routing problems with direct measurements. Although dynamic network performance characteristics such as available bandwidth and latency are the most relevant to applications and can be accurately measured on demand, the huge number of wide-area-spanning end-to-end paths that need to be considered in these distributed systems makes performing on-demand network measurements impractical because it is too costly and time-consuming.

Proximity measurements, based on repeated pair-wise distance measurements between nodes, can prove to be very onerous in terms of measurement overheads. Indeed, the existence of several overlays simultaneously can result in significant bandwidth consumption by proximity measurements (i.e., ping storms) carried out by individual overlay nodes [23]. Also, measuring and tracking proximity within a rapidly changing group requires high frequency measurements. We can also consider as an example the case of the Stribling's service [24], collecting RTT data between the PlanetLab nodes, that ceased its activity since the measurements overhead induced by this service becomes unmanageable for the PlanetLab infrastructure.

As a summary, most of the systems we introduced above are dedicated to overlay construction and lookups, rather than distance prediction at the Internet scale in a timely fashion.



(a) The Internet (b) The virtual space

Fig. 2. Correspondence between the Internet and the virtual space

In order to predict distances between any pair of nodes in the Internet, these non coordinate-based systems still need to perform costly measurements. In contrast, the embedding techniques do not require a full mesh of RTT measurements, to predict distance between any pair of nodes in the system.

III. Network Coordinates System Basics

To establish a consensus between the performance optimization needs introduced by the overlay networks and the scalability constraints imposed by underlying IP networks, several coordinates-based approaches aiming to estimate network distances have been proposed. The key idea of such systems is to model the Internet as a geometric space and characterize the *position* of any node in the Internet by a position in this space. The network distance between any two nodes is then predicted as the geometric distance between their coordinates without explicit measurements. In other words, if a node x learns the coordinates of a node y, x does not have to perform an explicit measurement to determine the distance to y; instead, the distance between x and y in the coordinates space is an accurate predictor of the network distance. It means that, as long as a reasonably accurate position for a node can be obtained with little effort, much of the distance measurement sampling cost can be eliminated and the remaining overhead amortized over many distance predictions.

Fig. 2 depicts the matching between the Internet and the virtual space. On Fig. 2(a), distances between four Internet hosts are represented. This distance can be, for instance, the round-trip time (RTT). Fig. 2(b) presents the estimated distances into a virtual space. Most NCS map Internet hosts to a virtual geometric space to estimate distances. In such a space, the estimated distance is evaluated using the classical distance function in a geometric space.

Predicting distances through coordinates makes sense if and only if a certain level of accuracy is guaranteed. It would be a matter of concern if the estimated distances do not reflect the reality. Therefore, when computing coordinates, an NCS aims at minimizing a relative error function. Typically, such a function will be built so that a zero value means a perfect prediction, while a positive value indicates that the predicted distance is too large [25]. This is given by:

$$\frac{d_{AB} - \hat{d}_{AB}}{min(\hat{d}_{AB}, d_{AB})} \ . \tag{1}$$

where d_{AB} is the measured distance between nodes A and

B and $\hat{d}_{AB}$ is the predicted distance. The absolute value of the directional relative error is called the *relative error* (Eqn. 2) [1].

$$\frac{\mid d_{AB} - \hat{d}_{AB} \mid}{min(\hat{d}_{AB}, d_{AB})} \ . \tag{2}$$

Most NCS map Internet hosts to a virtual Euclidean space to estimate distances. In such a space, the estimated distance $\hat{d}_{AB}$ is evaluated using the classical distance function:

$$\hat{d}_{AB} = \sqrt{\sum_{i=1}^{d} (\vec{x}_A - \vec{x}_B)_i^2} \ . \tag{3}$$

where $\vec{x}_A$ represents the coordinates vector of host A and $\vec{x}_B$ the coordinates vector of host B coordinates.

While the relative error is a good indicator of the accuracy of a given coordinate system, in terms of distance estimation, in many cases, applications only need to identify the nearest nodes among a set of candidate nodes. To answer how well a coordinate system can identify those closest nodes to a given one, a new metric called *relative rank loss* (*rrl*) has been proposed in [26]. For a given source node, and a randomly selected pair of nodes, this metric orders the actual distances and the estimated distances of each node of the chosen pair towards the source node. It is important that the relative rankings of distances is not lost. The rrl at a source node C can be computed according to the following formula:

$$rrl(\phi, C) = \frac{\{(A,B) \mid A \neq B \ and \ swapped \ (C,A,B)\}}{\frac{(|N|-1)(|N-2|)}{2}} \ . \tag{4}$$

where ϕ is a metric space, N is the set of nodes, (A, B) are elements of $N \times N$ (with N being the set of nodes in the system), and swapped (C, A, B) is true when the C's relative relationship to A and B is different in the two rankings, i.e., the original and the mapping (embedded) spaces. Note that, the rrl takes values between 0 (for no loss of relative order) and 1 (for a complete reversal of order). In other words, this metric quantifies the probability of incorrect rankings.

NCS offer many advantages, among them:

- *Easy and practical support to P2P applications*. Since most of current P2P applications would benefit from nodes' locations in the Internet, NCS seem to be of great benefit to these applications, in particular P2P nodes can easily maintain coordinates that would allow them to characterize proximity among them.
- *Scalability*. NCS have been designed to offer scalability properties to applications using them. In essence, coordinates computed locally and shared among all nodes in the network would allow for network distances estimations with very low overhead. The measurement overhead produced by each node positioning can be amortized over many un-measured distance predictions.
- *Acceptable accuracy*. Even though the mapping between actual network distances and geometric distances in the virtual spaces as constructed by current NCS is not perfect, a reasonably accurate positioning of nodes can be achieved. Network positioning errors achieved by today's NCS are often acceptable for a majority of applications, that would rely on a local appreciation of proximity between nodes, rather than on a complete knowledge of inter-nodes actual distances.

[1] In some literatures, instead of $min(\hat{d}_{AB}, d_{AB})$, d_{AB} is used. This usually produces smaller relative errors.

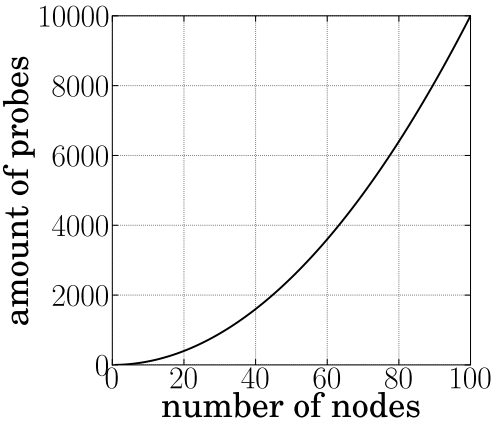


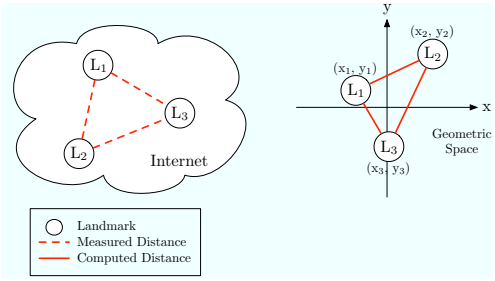
Fig. 3. Measurement overhead versus number of nodes

Fig. 3 shows the measurement overhead (i.e., the amount of probes injected in the network) as a function of the number of nodes in the system. Without any scalable measurement technique, the overhead is a quadratic function of the number of nodes involved in the system. In a system such as Azureus [15], such a measurement campaign would not be scalable. To offer the scalability property, any NCS should have an overhead much lower than the curve presented in Fig. 3.

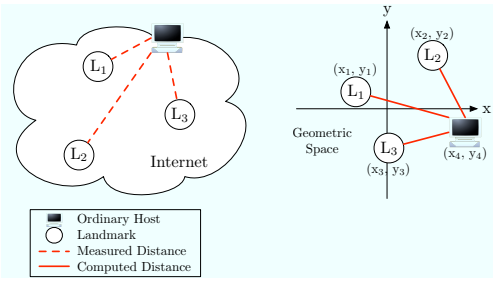
Finally, note that there are two families of NCS. *Landmarks-based* coordinates systems, where a fixed set of well-known trusted nodes are used to compute coordinates for all other nodes in the system. And, *decentralized* coordinates systems, where any node might be used to compute the coordinates of any other. In Sec. IV, we will present various NCS, some of them being landmarks-based, others being decentralized.

IV. Existing Network Coordinates System

Internet Distance Map Service (IDMaps) [12] is the first complete system that aims at predicting Internet distance and might be seen as the predecessor of landmark-based coordinate systems. IDMaps is an infrastructural service in which special *HOPS* servers maintain a virtual topology map of the Internet consisting of end hosts and special hosts called *Tracers*. This virtual topology map is used to predict Internet distance. For example, the distance between hosts x and y is estimated as the distance between x and its nearest Tracer T_1, plus the distance between y and its nearest Tracer T_2, plus the shortest path distance from T_1 to T_2 over the Tracer virtual topology. As the number of Tracers grow, the prediction accuracy of IDMaps tends to improve. Designed as a client-server architecture solution, end hosts can query HOPS servers to obtain network distance predictions.

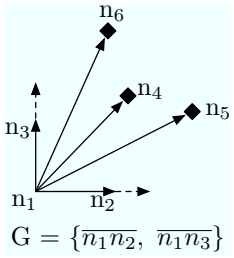


(a) Landmarks operations

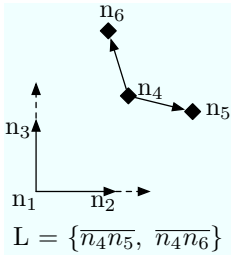


(b) Ordinary host operations

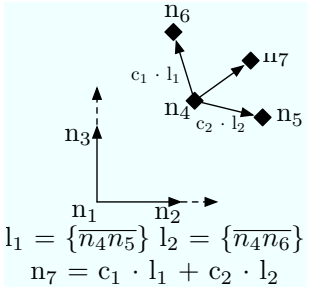
Fig. 4. Operations in a landmark-based approach



(a) Finding lighthouses



(b) Local basis



(c) Transition

Fig. 5. Lighthouses behavior

Compared to the IDMaps, coordinate-based systems are different in that nodes are able to use their own resources to compute their positions in the Internet. Moreover, these systems do not directly interact with any applications. It is up to the applications running on end hosts to decide how to use the computed locations (coordinates).

IDMaps sets then the basis for coordinate-based approaches. Indeed, such a service was driven by the main principle of predicting some Internet distances from an a priori partial knowledge of the topology rather than systematically measuring it.

A. *Landmark-Based Approaches*

Typically, landmark-based approaches are a two part architecture made of *landmarks* and *ordinary hosts*. A landmark refers to a well known node computing its own coordinates while an ordinary host evaluates its coordinates based on landmarks ones. This architecture is illustrated in Fig. 4.

In such an approach, only the landmarks need to perform all-pairs latency measurements, as shown in Fig. 4(a), and then map themselves into the geometric space.

An ordinary host desiring to position itself in the geometric space first performs measurement towards the landmarks. Next, based on those measurements and the landmarks coordinates, it computes its own coordinates. This process is shown in Fig. 4(b).

In the following, we investigate different NCS that are based on landmarks for coordinates computation.

Global Network Positioning (GNP) [13] is the implementation of a standard landmark-based NCS, as presented above. It is the first system to propose modeling the Internet as an n-dimensional geometric space. Given such a space, GNP approximates the latency between pair of hosts as the Euclidean distance between their corresponding coordinates in that space.

With GNP, K landmarks are required, K being at least $n+1$ in an n-dimensional geometric space, otherwise it is impossible to compute an host coordinates. This constraint is explained by the uniqueness of coordinates required for every host. In an n-dimensional space, $n+1$ landmarks to achieve a multilateration towards the target host to localize are indeed necessary.

As already explained, GNP starts by instructing the landmarks to measure the inter-landmark latencies. Based on these latencies, GNP calculates all the landmark coordinates so that the distance between any pair of these coordinates is as close as possible to the actual measured latency between the corresponding pair of the landmarks. The discrepancy between the geometric distance and their corresponding latencies is minimized using the *Simplex DownHill* method [27], a non-linear optimization algorithm.

Given the K landmarks coordinates, GNP can next compute the coordinates of any node A based on the measured latencies between A and each of the landmarks. Host A computes its own coordinates so that the distance between these coordinates and the coordinates of each landmark is as close as possible to its corresponding measured latency. This is again achieved by means of the Simplex DownHill method. Note that all systems discussed below are GNP variations.

Lighthouses [28] is a GNP extension seeking to overcome the limitations generated by the use of landmarks. Indeed, the measurement traffic arriving at each landmark grows in proportion to the number of target hosts as the system scales. To address this, Lighthouses uses multiple landmark sets, with each ordinary host measuring distances to only one landmark set. It is built above the concept of multiple *local basis* with a *transition matrix*.

Lighthouses, like GNP, has a special set of landmark nodes called *global landmarks*. Node A that joins Lighthouses does

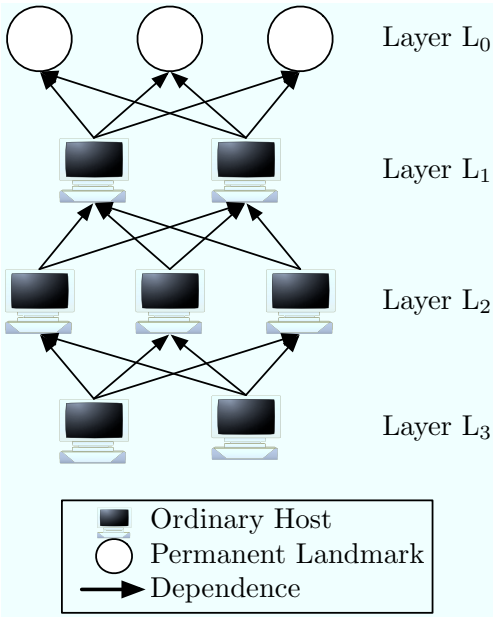


Fig. 6. The hierarchy in NPS

not have to query those global landmarks. Indeed, it first contacts any node, say B, that is already in the system. Node B provides to A the list of nodes that can act as A's landmarks. In a n-dimensional geometric space, the joining node selects $n+1$ nodes at random among those in this list. It should be noted that if node A cannot find $n+1$ landmarks, it constructs a local basis with the ordinary nodes already in the system. Every ordinary node has its own basis, also called the local basis, and computes its coordinates using this basis while in GNP the same basis was used by all the ordinary nodes in the system. Therefore, node A constructs a local basis $L = \{\boldsymbol{l_1}, \boldsymbol{l_2}, \ldots, \boldsymbol{l_n}\}$, where each vector $\boldsymbol{l_i}$ is a pair of landmarks, by applying the *Gram-Schmidt* process [29]. Simply said, the Gram-Schmidt process is a method for orthogonalizing a set of vectors in an inner product space, most commonly the Euclidean space.

To compare the coordinates of two nodes, the position must be expressed accordingly to the same basis. The transition matrix permits to express the coordinates of a node A in the local basis of another node B. As a result, node A computes a transition matrix between its local basis and the global basis. It is worth to notice that this process does not require any additional distance measurements.

Fig. 5 illustrates Lighthouses behavior in a two dimensional environment. Let us consider six nodes already present in the system (n_1 to n_6 - Fig. 5(a)) and a seventh node, n_7 wants to join the system. The first step for n_7 is to contact a node in the system, for instance n_4. n_4 replies with a list of nodes that can act as landmarks for n_7, say $\{n_4, n_5, n_6\}$. n_7 can then start measuring its distance between itself and the lighthouses. n_7 computes next its local basis, using the Gram-Schmidt process (Fig. 5(b)). Finally, Fig. 5(c) shows the computation of n_7 transition matrix.

The *PCoord* scheme [30] proposes another set of landmark selection algorithms. It is similar to lighthouse in that they both do not require each node to measure distances to all the predetermined landmarks. Its best landmark selection algorithm uses gossip protocol [31], [32], [33] to get informed about other nodes so that it can select a well-spread set of landmarks. Landmarks, in PCoord, are only used for bootsrapping while coordinates are calculted in the fahsion of Lighthouses.

The *Network Positioning System* (NPS) [25] extends GNP into a hierarchical coordinate system, where all nodes could serve as landmarks for other nodes. It aims to recover "gracefully" from either landmark failures, or situations where these special entities of the system and their network access links become performance bottlenecks. The main departure from GNP is that any node that has determined its position can be chosen by a membership server to be a landmark for other nodes. The role of the membership server is to provide essentially initial configuration parameters (e.g., identify the landmarks, the maximum number of layers in the system, the geometric space used for embedding, etc.) to ordinary nodes in the system.

Actually, the membership server randomly chooses eligible nodes to become landmarks when the permanent landmarks are too heavily loaded or unavailable. To ensure consistency, NPS imposes a hierarchical position dependency among the nodes (see Figure 6).

Given a set of nodes, NPS partitions them into different layers. A set of 20 landmarks is placed in layer-0 (or L_0), the top layer of the hierarchy (these permanent landmarks are the fixed infrastructure used to define the bases of the geometric space model), and an 8-dimension Euclidean space is used for embedding. Each node in layer L_i randomly picks some nodes in layer L_{i-1} as its landmarks.

Internet Coordinate System (ICS) [34] shares the similarity with GNP and Lighthouses in that it also represents location of ordinary hosts in a Cartesian coordinate system. Nevertheless, ICS provides a unique mapping from the distance mapping to the Cartesian coordinate system. Further, any ordinary host does not have to measure its distance to all the landmarks (called *beacon node* in ICS), but rather to a subset of beacon nodes and obtains a n-dimensional distance vector d_i where n is a number of chosen beacon nodes among the m available. The location of the beacon node is then calculated by multiplying the distance vector with a transformation matrix. This transformation is based on *principal component analysis* (PCA) [35], also called the Karhunen-Lòeve Transform. This transformation projects the distance data space into a new coordinate system. The purpose of PCA is to reduce the dimensionality of a data set (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample information. By information, we mean the variation present in the sample given by the correlations between the original variables. The new variables, called *principal components*, are uncorrelated, and are ordered by the fraction of the total information each one retains.

The transformation matrix is obtained by applying *singular value decomposition* (SVD) on the distance matrix, say D, formed by the delay measured between beacon nodes. Indeed, an *administrative node* is elected among beacon nodes, aggre-

gates the distance vector of all the beacon nodes, obtains the distance matrix D, and applies PCA to obtain the transformation matrix. The role of the administrative node is also to determine the dimension of the coordinate system.

In order to enhance ICS, a clustering scheme for beacon nodes and a partial measurement approach, where only a limited number of beacon nodes are used by an ordinary node A, are also proposed. In such a case, the administrative node groups beacon nodes that are close to each other into clusters, selects for each cluster a median beacon node, and then sends a list of median beacon nodes to a node willing to join the ICS architecture. Nevertheless, with respect to the clustering approach, the beacon nodes need to be placed and well distributed a priori. The obtained results show that, when the median node of each cluster is chosen as beacon node, the estimation errors are smaller than those where beacon nodes are randomly selected. This implies that a partial measurement approach method benefits from choosing most representative beacon nodes.

Tang et al. [36] also applied PCA method to project distance measurements into a Cartesian coordinate system with smaller dimensions. We call this technique *virtual landmarks*. They considered the coordinate of a host in the coordinate system as the distances to virtual landmarks while the coordinate in the distance data space represents the distances to actual landmarks (beacon nodes). Indeed, Tang et al. propose the use of the Lipschitz embedding in order to embed distances between nodes in a low dimensional space obtained by compressing the full delay matrix using the PCA method. The Lipschitz embedding is the basis for a number of important theoretical results on minimum-distortion embedding [37], [38]. For network latency estimation, it has the advantage of being simple to formulate and fast to compute.

The basic idea of the Lipschitz embedding is to use network distances themselves as coordinates. To find the coordinate vector $\overrightarrow{x}_i$, for node i, one sets the j^{th} component of $\overrightarrow{x}_i$ to the measured distance between node i and landmark j, for $j = 1, ..., n$.

The Lipschitz embedding can be accurate because two entities that are close to each other in a metric space typically have similar distances to many other entities. Thus two nearby points in the original metric space may have very similar coordinate vectors, and so may map to nearby points under the Lipschitz embedding.

This study also explores methods to reduce the number m of Landmarks that need to be probed without adversely affecting the accuracy.

By applying the PCA method to an $m \times n$ matrix A in which row i is the initial n-dimensional coordinate vector $\vec{x}_i$ for node i, we can map each $\vec{x}_i$ to a new $\vec{y}_i$ in a lower dimensional space, while approximately preserving distances.

The mapping from $\vec{x}_i$ to $\vec{y}_i$ obtained via PCA is a linear one. That is, $\vec{y}_i = M\vec{x}_i$ for some M (where M is an $r \times n$ matrix). Final coordinate of node i (the components of $\vec{y}_i$) can be seen as distances to virtual landmarks. The distance to a virtual landmark is defined as a linear combination of distances to actual landmarks.

Tang et al.'s most important findings is that the network distances can generally be described as the linear combination of a small number of orthogonal vectors - typically 7 to 9. In such a case, Tang et al. suggest that an embedding in an Euclidean space of 7 to 9 dimensions is likely to be sufficient for reasonable accuracy.

Internet Distance Estimation Service (IDES) [39], [40] operates as standard landmark-based approaches: landmarks measure distances between them and report them to a centralized server (named *information server*). Ordinary hosts measure their distance to and from landmarks. The difference with other approaches described in this section stands in the mapping calculation. Mao et al. provide two learning algorithms allowing a linear dimensionnality reduction applied to matrixes: *Singular Value Decomposition* (SVD) [41] and *Non-Negative Matrix Factorization* (NMF) [42].

B. Distributed Approaches

This class of approaches extends the embedding concept, either by generalizing the role of landmarks to any node existing in the system, or by eliminating the landmark infrastructure. Decentralized Internet coordinate systems can be seen as peer-to-peer network positioning systems.

Practical Internet Coordinates (PIC) [43], [44] does not require explicitly designated landmarks. In PIC, the joining node can pick any node whose coordinates have already been computed to be a landmark. This is similar to GNP [13] but GNP uses a fixed set of landmarks for all the nodes that join the system. On the contrary, a PIC node probes the network distance to each element of a set of landmarks, $\mathcal{L}$, having at least $n+1$ members, n being the chosen geometric dimensional space. It uses an active node discovery protocol to find a set of nearby nodes to use for computing coordinates. Different strategies such as random nodes, closest nodes, and a hybrid of both, are proposed. Then it obtains the coordinates of each landmark and uses the Simplex DownHill method to compute its coordinates such that the errors in the $|\mathcal{L}|$ predicted distances between the node and each node in $\mathcal{L}$ are minimized.

The intuition behind the different strategies to choose the nodes that acts as landmarks is to overcome the inherent Simplex DownHill method limitations (cfr. Sec. V-A). The closest strategy (resp. random strategy) should provide the Simplex DownHill method with better information to position the joining node correctly in the Euclidean space relative to nearby nodes (resp. distant nodes) in the network. Therefore, the closest strategy should achieve lower relative errors when predicting short distances whereas the random strategy should achieve lower relative errors when predicting long distances. The hybrid strategy should achieve something in the middle.

Shavitt and Tankel discover that the accuracy of IDMaps depends on the positions between hosts and Tracers [14]. Worst, IDMaps is only able to find the closest node in 85% of the cases [14].

To overcome this, Shavitt and Tankel introduce *Big-Bang Simulation* (BBS) [14], a way to model the network nodes as a set of particles. Each particle is a node image in a geometric space. Particles are traveling in the space under the effect of potential force field. The name 'Big-Bang Simulation" comes

from the fact that particles are initially placed at the origin of the space.

The field force is derived from the potential energy error which is equal to the total embedding error. The field force reduces the potential energy of particles and particles pull or repulse each others depending on the distance error between them [45]. During each calculation phase, the particles are traveling in trajectories tending to reduce the potential energy of the whole system. At the end of each phase, the system approximately achieves an equilibrium point where the potential energy is minimized [14].

A calculation phase consists of several iterations which moves the particles in discrete time intervals. The particles position and velocity at time $t + \delta t$ are calculated by applying Newton's laws of motion and the new potential energy is calculated at the end of the iteration. Note that, the particles positions and velocities calculated in the current iteration are the input to the next iteration. Increasing the timestep δ provides greater numerical efficiency. On the contrary, decreasing the timestep permits to attract particles to a global minimum potential energy. A good introduction to Newton's laws of motion can be found in [45].

If the particles were only under the force field effect, they would move away too fast and oscillate forever with constant velocity. To fix this, a *friction* force is added.[2] With this force, a part of the energy is lost due to friction and, after a while, the system stabilizes. The friction force depends on the the particle normal force. The moving particles are assigned a friction coefficient μ_k and the static particles are under the effect of the μ_s friction coefficient.

The magnitude f_{ij} of the field force determines how much the induced force pulls or repulses the two particles i and j. A positive value means that the force pulls the two particles together. On the contrary, for a negative value, the two particles are repulsed. Shavitt and Tankel showed that this induced force is given by the derivative of the prediction error.

The previous NCS we have seen use conventional gradient minimization schemes, i.e., the Simplex DownHill method. When such a method is used, the minimization can be caught by a local minimum but a local minimum is not necessarily the global one. Thus, while traditional coordinates systems running the Simplex DownHill method are very sensitive to the initial coordinates, BBS does not care about initial coordinates. This BBS quality is the result of the kinetic energy accumulated by the moving particles, permitting them to escape a local minimum.

Vivaldi [46] is probably the most successful NCS that has been proposed so far. It does not require any fixed network infrastructure and makes no distinctions between nodes. A Vivaldi node collects distance information for a set of neighbors and computes its new coordinates with the collected measurements. The idea is that node A is represented as a unitary mass connected to each neighbor B by a spring with the rest length set to the measured RTT (i.e., d_{AB}). The actual length of the spring is the distance predicted in the coordinates space (i.e., $\hat{d}_{AB}$). A spring always tries to have an actual length equals to its rest length. Thus if $\hat{d}_{AB}$ is smaller than the measured RTT, the spring pushes the two masses attached to it. On the contrary, if the spring is too long, it pulls the masses and reduces its actual length.

The Vivaldi procedure uses each RTT sample to update its coordinates. An identical Vivaldi procedure runs on every node. Each sample provides information allowing a node to update its coordinate. The algorithm handles high error nodes by computing weights for each received sample. The sample used by each node A is based on measurement to a node, B, its coordinates x_B and the estimated error e_B being reported by B. A relative error of this sample is then computed with respect to d_{AB} and $\hat{d}_{AB}$. The node then computes the sample weight, balancing so local and remote error. The local (resp. remote) error represents node A confidence in its own coordinate (resp. node B). This sample weight is used to update an adaptive timestep, δ, defining the fraction of the way the node is allowed to move toward the perfect position for the current sample. Thus, the coordinates are updated by moving a small step towards the position that best reflects the RTT measured. The size of the modification depends on the weight of the sample, and on the difference between the measured (d_{AB}) and the predicted RTTs ($\hat{d}_{AB}$). The Vivaldi algorithm quickly converges towards a solution when latencies satisfy the triangle inequality.

Vivaldi also proposes a variant of Euclidean Coordinates to better model Internet latencies, and introduces the notion of *height* [46]. A height space consists in an Euclidean coordinate augmented with a height vector. This vector models the latency penalty of network access links, such as queuing delay, DSL lines, or cable modems. With height, the distance between nodes is measured as their Euclidean distance plus the height represented by a positive value of the height vector.

It is worth to notice that extensions to Vivaldi have been provided in order to position nodes in an hyperbolic space [47], [48].

V. Limitations and Discussion

In Sec. IV, we discussed several NCS techniques. However most of these systems, if not all, suffer from different limitations. In this section, we discuss such limitations (Sec. V-A and V-B). Note that Sec. VI will focus on a particular limitation: security.

It has been shown in previous sections that an NCS allows for an easy and practical latency prediction on the Internet. However, one could criticize them for requiring expensive maintenance and having more or less accurate prediction. At the very least, *triangle inequality violations* (TIV) could be a major barrier for the accuracy of such systems [49], [50]. Note that the matrix factorization introduced by Mao et al. in IDES [39], [40] allows a representation of distances violating TIVs and asymmetric distances. Further, Lee et al. show that better accuracy can be reached when considering lower dimensional system (i.e., a 2-dimensional Euclidean coordinate system) [50], [51].

Lua et al. observe that absolute relative error may not be the major indicator of the quality of an embedding as

[2]Friction is the force resisting the relative motion of two surfaces in contact or a surface in contact with a fluid (e.g., air on an aircraft).

experienced by a user [26]. They demonstrate that, using other accuracy metrics that attempt to quantify various aspects of user-oriented quality (such as Relative Rank Loss or Closest Neighbors Loss), the quality of the coordinates-based systems is not as high as suggested by the absolute relative error.

Moreover, choosing the suitable geometric space for coordinates embedding, and more generally, to model the Internet has received much attention from the research community and has been shown to be a challenging task. Basically, coordinates systems have concentrated on pure Euclidean spaces or other simple geometric spaces like the surfaces of spheres and tori. Shavitt and Tankel introduce a new coordinates space that places nodes some distance "above" a Euclidean space (height model) [52]. Shavitt and Tankel propose using a hyperbolic coordinates space to model the Internet. The hyperbolic model may address a shortcoming of the Vivaldi's height model that implicitly assumes that each node is behind its own access link. If two nodes are behind the same high-latency access link, the height model will incorrectly predict a large latency between the two nodes: the distance down to the plane and back up.

In addition to those "common" limitations, a few disadvantages which are specific either using landmark-based approaches, or distributed approaches may exist.

A. Landmark-Based Approaches

Obviously, the main drawback of the landmark-based approaches is the need of a dedicated landmarks deployment. In fact, the landmarks number and placement affect the RTT predictions accuracy. Furthermore, landmarks failures and overloading also affect latencies which can be measured with high inaccuracies. Landmark systems do not take advantage of all exchange between nodes (as in Vivaldi [46] for instance): only measurements to landmarks are helpful in updating coordinates. Also, the measurement traffic to the landmarks increases in proportion to the number of nodes participating in the system as well the inter-landmark measurements, moderating so the overall system scalability.

To calculate coordinates, GNP, Lighthouses, and NPS formulate a multidimensional global optimization problem that minimizes the difference between the measured network distance and the Euclidean distance in a Cartesian coordinates system. The Simplex DownHill method is then applied to solve the minimization problem. However, such method only gives a local minimum that is close to the starting value and does not guarantee that the resulting coordinates are unique. This leads to the eventual assignment of different coordinates for the same node depending on the minimization process.

Finally, the problem of using this method is its slow convergence. As for virtual landmarks, it uses the Lipschitz embedding assuming that network distances obey the triangle inequality. It has been demonstrated that Internet traffic does not always follow the shortest possible path [26], [49] and that there is potential violation of the triangle inequality due to routing policies.

B. Distributed Approaches

This class of approaches extends the embedding concept, either by generalizing the role of landmarks to any node existing in the system, or by eliminating the landmark infrastructure. Although, distributed approaches have attractive properties, in particular those of scalability and the "no need" of dedicated infrastructure, one could criticize them for being more vulnerable to security threats, as we will discuss it in Sec. VI, and for having worse prediction accuracy than landmark-based approaches.

Considering PIC as the first system that aimed at introducing a security mechanism against malicious behaviors, we notice that this security mechanism, based on the fact that the triangle inequality systematically holds, might degrade the system performance and accuracy. We will discuss this aspect further in Sec. VI-A. In addition, PIC also uses the Simplex DownHill method whose main drawbacks were already enumerated in Sec. V-A.

As Vivaldi simulates a physical spring system, obviously if the triangle inequality is violated, Vivaldi cannot find perfect positions for the nodes and is stuck in endless oscillations. The nodes never converge towards stable coordinates. This is explained by the fact that Vivaldi uses a moving average of recent relative errors. It has been demonstrated that, in presence of TIVs in the delay space, this local error estimate can oscillate and prevent the system from finding a good solution [53], [54], [55], [56].

Nevertheless, current live implementations and deployments of Internet coordinates systems in the "wild" show that using such distributed NCS is beneficial for P2P applications and overlays ([15], [57], [58]) relying on the notion of network topology-awareness. Using the Azureus BitTorrent network as a testbed, Ledlie et al. show that even if, live, large-scale NCS behave differently than the experimentally tested coordinates system on PlanetLab, Azureus' coordinates achieve the major goal they were designed for: deliver a reasonable accurate positions of nodes, allowing for an acceptable approximation of nodes proximity, and by inference optimization of overlay routing [56] [3]. Ledlie et al. show that incorporating Vivaldi's coordinates in a one million node Azureus network improves the Azureus efficiency. However, this is achievable by implementing specific techniques in Azureus in order to support coordinates in an effective way. Basically, to improve the accuracy and stability of coordinates-based systems, several works propose different techniques:

- *latency filters and application-specific coordinates updates*, in order to make the distinction between constantly evolving "system-level" coordinates and "useful application-level" coordinates that should be stable [56]. It should also be noticed that *SVivaldi*, proposed by De Launois et al. [53], proposes a different method for stabilizing coordinates by asymptotically dampening the effect of each new Vivaldi measurement. SVivaldi allows also coordinates to be more accurate. While this factor does mitigate oscillations in a fixed network, it

[3] Azureus [15] is currently one of the most popular clients for BitTorrent, a file sharing protocol [7]

	Landmarks	TIVs Sensitivity	Forgeable
1. GNP	•	•	•
2. Lighthouses	○	•	•
3. NPS	•	•	○
4. ICS	○	•	•
5. Virtual Landmarks	○	•	•
6. PCoord	○	•	•
7. IDES	•	○	•
6. PIC		•	○
7. Vivaldi		•	•
8. BBS		•	•

TABLE I
LIMITATIONS OF NCS TECHNIQUES.

prevents the algorithm from adapting to changing network conditions.

- *gossip-based coordinates update*, rather than piggybacked coordinates on to application-level messages. This technique has been shown to expand the size of the Vivaldi working set, expanding the set of neighbors for each node, and then improving its accuracy [56].
- *TIVs exclusion or awareness*: Inspired by the removal of a small percentage of the nodes with the largest triangle inequality violations from the Azureus latency matrix. Removing 0.5% of nodes leads to 20% improvement in global accuracy [56]. These observations confirm a theoretical work that shows how to decrease embedding distortion by sacrificing a small fraction of distances to be arbitrarily distorted [59]. These results mainly demonstrate that if a mechanism could prevent a small percentage of nodes (Triangle inequality violators) from affecting the rest of the system, it would improve overall accuracy. Kaafar et al. also show that an hierarchical Vivaldi system where TIVs are less severe, will be more accurate in predicting intra-cluster distances [55]. In addition, Chen et al. propose Pharos [60], [61], a hierarchical approach based on the clustering of nodes, to mitigate the impact of TIVs on distance predictions. Each node uses two set of coordinates in Pharos. Therefore, coordinates computed at the lower (resp. higher) level of clusters are called local coordinates (resp. global coordinates). Within their cluster, nodes use more accurate local coordinates to predict intra-cluster distances, and keep using global coordinates when predicting longer distances towards nodes belonging to foreign clusters.

C. Summary

Table I summarizes limitations of individual NCS techniques described in Sec. IV. We focus on three key aspects: landmarks, sensitivity to TIVs and forgeable. To clarify some of our terminology (Table I): the first column, labeled *landmarks* indicates NCS methods (rows 1-5) that are landmark-based approaches and how these methods are limited by the use of landmarks. The second column, named *TIVs Sensitivity* illustrates the detrimental effect of TIVs on NCS. Finally, the last column, *Forgeable*, denotes coordinates or measurements that may be deliberately invalid. "○" denotes a partial limitation, e.g., for Lighthouses any node that is already in the system can act as landmark. Therefore, the drawbacks generated by the use of landmarks are reduced. "•" indicates an important limitation, e.g., for Vivaldi in the presence of TIVs, nodes stick in endless oscillations leading to inaccurate coordinates.

Lighthouses, ICS, and Virtual landmarks which are based on a linear matrix transformation are less subject to the damage caused by a landmark-based coordinates system. In contrast, the choice of landmarks significantly affects the accuracy of GNP's RTT predictions. Despite NPS includes a hierarchical system for reducing load on the landmark nodes, it is nevertheless dependent on the landmarks positions. Fortunately, distributed approaches such as PIC, Vivaldi and BBS overcome those limitations by eliminating the landmark infrastructure.

Most of NCS techniques assume that triangle inequality holds in Internet. Bullets in the second column exhibit some of the problems for which network coordinates are frequently criticized, i.e., inaccuracy and fragility in the presence of TIVs. In fact, network delays do not necessarily satisfy the triangle inequality due to routing policies. The different coordinates-based embedding techniques reviewed in this paper suffer from TIVs. Therefore, when faced with these TIVs, coordinates systems resolve them by forcing edges to shrink or to stretch in the embedding space; this intuitively results in oscillations of the embedded coordinates, and thus leads to large distance prediction errors.

Unfortunately, NCS are vulnerable to even a small number of malicious nodes lying about their coordinates or measurements. Some of them, NPS and PIC include a strategy for mitigating the effects of simple malicious attacks. For instance hollow bullets in column C (Table I) show that PIC and NPS are less vulnerable to a potential malicious nodes compared to other NCS techniques. Indeed, malicious nodes could potentially lie about their positions and/or inflate network distances by holding onto probe packets. The basic idea in NPS is to eliminate a reference point if it fits poorly in the Euclidean space compared to the other reference points. Nevertheless, the NPS security mechanism can be defeated very simply. Basically, the attacker can delay the measurement probe so that the measured distance will appear to be much greater than the actual distance. At the same time, the attacker lies about its coordinates in a way that the resulting estimated distance is roughly within the measured distance. In addition, the PIC detection mechanism, based on the observation that the triangle inequality holds, is affected by potential inequality violations which often occur in the Internet.

Although network coordinates have attractive properties for latency prediction on the Internet, they have a potential limitation in practice because Internet routing policies cause too many triangle inequality violations. To avoid the pitfall caused by TIVs, it is mandatory to build systems that are TIV-aware.

In conclusion, while developing coordinates-based systems with perfect accuracy is a long-term challenge, current approaches are already sufficiently accurate for most applications and allow trade-offs between accuracy and measurement overhead for dynamic topology-aware overlays.

But it should also be noticed that these come at the expense of slow convergence times ranging from tens of seconds to several minutes [56]. This is several orders of magnitude slower than what is achievable with direct 'on-demand' distance measurements between nodes and is often unacceptable for topology-aware applications whose aim is to quickly identify "best nodes".

We therefore contend that coordinates-based positioning systems are an attractive proposition if they are deployed as a service: every node could run a coordinates system daemon at boot time which would then be capable of providing accurate coordinates estimates to applications and their overlays on request. In essence, the coordinates system could then be seen as a component of a "virtual infrastructure" that supports a wide range of overlays and applications.

But a system providing an "always-on and large scale coordinates service" would also likely be a prime target for attacks, as already introduced above. This disruption could result in the mis-functioning or the collapse of very many applications and overlays. Indeed, as the use of overlays and applications relying on coordinates increases, one could imagine the release of worms and other malware whose purpose is to attack the virtual infrastructure as a whole.

Put simply, regardless of the accuracy of these Internet coordinates systems, securing them is a necessary condition to their deployment. Security in NCS is one of the most relevant limitations that these systems are facing today, especially if we know that most, if not all, of current proposals for coordinates systems assume that the nodes partaking in the system cooperate fully and honestly with each other, that is that the information reported by probed nodes is correct. This makes them vulnerable to malicious attacks. In particular, insider attacks executed by (potentially colluding) legitimate users or nodes infiltrating the system could prove very effective, as shown by Kaafar et al. [62]. In the next section, we discuss different security issues that NCS are facing, detailing the different types of attacks that can be harmful for them. We also present the proposed approaches that deal with these security issues in an attempt to secure the NCS.

VI. Security

Different approaches have been proposed to secure NCS. First, two of the systems described in this paper propose their own specific mechanisms to defend against malicious nodes, namely PIC [43], [44] and NPS [25]. Recently, it has been shown that these mechanisms are rather primitive, still in their infancy, and definitely cannot defend against all types of attacks. So, in a second step, more generic and robust approaches have been proposed. In the following, we discuss the PIC and NPS security mechanisms (Sec. VI-A and VI-B), and present attacks that have been identified and shown to drastically degrade the NCS performance (Sec. VI-C). Finally, we present an overview of the generic defense mechanisms that have been proposed as countermeasures against the NCS attacks (Sec. VI-D).

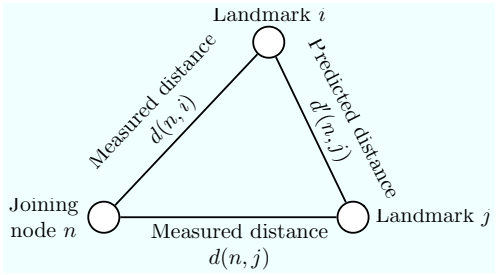


Fig. 7. Triangle inequality with measured and predicted distances in PIC

A. PIC Security

PIC aims at defending the security of its coordinates system against independent malicious participants using a test based on the triangle inequality. Basically, the test relies on the observation that the triangle inequality holds for most triples of nodes in the Internet. Therefore, PIC assumes that for most triples of nodes a, b and c , $d(a,b) + d(b,c) \geq d(a,c)$, where $d(i,j)$ denotes either the measured network distance between nodes i and j or the virtual distance in the geometric space.

The intuition behind the security test of PIC is as follows. An attacker that lies about its coordinates or its distance to the joining node is likely to violate triangle inequality. The joining node uses the distances it measured to each landmark node and the coordinates of the landmarks to check for TIVs. It then removes from its proper set of landmarks used for positioning the nodes that most violate the triangle inequality.

This is illustrated in Fig. 7 where n denotes a new node joining the PIC infrastructure. Landmarks i and j are both used by n to calculate its coordinates.

For each landmark used for coordinates computation, the security test checks whether the upper bounds and lower bounds defined by each landmark j are satisfied by i and computes the $upper_i$ and $lower_i$ metrics.

$upper_i$ is the sum of the deviations above the upper bounds while $lower_i$ is the sum of the deviations below the lower bounds. The security test computes the maximum value of both metrics for all landmarks used by n and removes the landmark which measurements are deviating from the computed upper and lower bounds. Then, the joining node uses the Simplex DownHill method to compute its coordinates with the remaining landmarks. This process is repeated a fixed number of times.

Costa et al. show that such security test can deal with up to 20% of malicious nodes existing in the system [43], [44]. However, subsequent works indicate that network RTTs commonly and persistently violate the triangle inequality [26], [49]. A security mechanism based on the fact that the triangle inequality systematically holds, may degrade the performance of a clean system, i.e., a system without malicious nodes inside.

B. NPS Security

NPS includes a strategy for mitigating the effects of simple malicious attacks. Indeed, malicious nodes could potentially lie about their positions and/or inflate network distances by

holding onto probe packets. The basic idea is to eliminate a landmark (by not considering it as so) if it fits poorly in the Euclidean space compared to the other landmarks. Each node, when computing its coordinates, based on measurements from different landmarks, would reject the reference that provides a relative error significantly larger than the median error of all other reference nodes. Specifically, assume there are N landmarks L_i, at positions P_{Li}, and the network distances from a node H to these are D_{Li}. After H has calculated a position P_H based on these reference points, for each L_i, it computes the fitting error E_{Li} as $\frac{|distance(P_H,P_{Li})-D_{Li}|}{D_{Li}}$. Then the requesting node, H, decides whether to eliminate the landmark with the largest E_{Li}. The criterion used by NPS is that if:

$$\max_i E_{Li} > 0.01 \tag{5}$$

and,

$$\max_i E_{Li} > C \times median_i(E_{Li}), \tag{6}$$

where C is a sensitivity constant. Then, the landmark with $max_i E_{Li}$ is filtered (i.e. H tries to replace it by another landmark for future repositioning).

Unfortunately, it has been shown that such a security mechanism is vulnerable to various attacks [62]. Basically, it consists, from an attacker point of view, in interfering with the constraints 5 and 6 by lying about its coordinates and/or tampering with measurement probes. This leads to a discrepancy between measured and estimated latencies.

In the following, we will present the different classes of identified attacks on NCS, and describe examples of attacks belonging to such classes.

C. Internal Attacks

Noticing that current NCS proposals assume fully cooperation and honesty among nodes, for their coordinates embedding, Kaafar et al. show that NCS are vulnerable to malicious attacks [62], [63]. In particular *internal attacks* executed by (potentially colluding) nodes infiltrating the system could prove being very effective. An internal attack refers to a class of attacks in which malicious nodes have access to the same data as legitimate users, often called *Insiders*. This means that participants are not completely trusted entities, or that malicious nodes have the ability to bypass any authentication mechanism. In essence, malicious nodes are able to send misleading information when probed, or send manipulated information after receiving a request or affect some metrics observed by chosen targets. Based on these assumptions, Kaafar et al. were the first to identify threats on NCS and classify attacks into four families [62]:

- *Isolation* aims at isolating nodes in the virtual coordinates space.
- *Repulsion* tries to alleviate a malicious node's or victim's attractiveness.
- *Disorder* aims at introducing chaos in the coordinates as a form of denial-of-service (DoS) attack.
- *System control* tries to take the control of a node that influences the coordinates of many other nodes or to become such a node.

We can easily illustrate those classes of attacks through four concrete examples that can be executed on the Vivaldi system: `Random` *attack*, `Independent Isolate` *attack*, `Repulse` *attack*, and `Colluding Isolate` *attack*.

A `Random` attack, is an example of the Disorder class attacks on the Vivaldi system, where each time a malicious node is contacted and requested to provide its coordinates, it replies with randomly generated coordinates and a low constant value of its local error.

An `Independent Isolate` attack, is an example of the Isolation class attacks on the Vivaldi system, where the malicious node delays the measured RTT such that it is consistent with the random coordinates it claimed for the victim it chose at the beginning. The malicious nodes aims at moving the victim to force its coordinates to be far away from all other nodes in the system. In other words, the malicious node will consistently and systematically direct the victim towards a designated coordinates aiming at isolating it.

A `Repulse` attack is an example of the Repulsion class attacks on the Vivaldi system. A malicious node claims a position that is far away from the actual coordinates, possibly far away from the origin, and then delays each measurement it receives in a way consistent with such far away position. In this way, the malicious node fools other honest nodes that it is really away from all other nodes in the system.

Finally, a `Colluding Isolate` attack, is an example of the system control class attacks on the NPS system. In this example, the malicious nodes cooperate with each other and behave in a correct and honest way until enough of them become landmarks at the same layer in the NPS architecture. Once at least a minimum number of malicious landmarks has been reached, these attackers identify a common set of victims. The goal of this attack is to push the victims into a remote location at the "opposite" of where the attackers pretend to be, thus isolating the victims from all the other nodes (in the coordinates space).

For all these attacks, Kaafar et al. have shown that larger systems are consistently more resilient than smaller ones (e.g., a system of 100 nodes is more sensitive to an attack performed by 10 nodes than a system with 100,000 nodes with 10,000 attackers) [63]. Hence, it seems to be a compelling case for large-scale coordinates systems to be built as a virtual infrastructure service component. The paradox is of course that always-on, large-scale systems supporting many different applications will always attract more attacks than systems with a smaller reach, while the large size of the system itself would act as a particularly good terrain to create especially virulent propagation of the attack. Kaafar et al. have also shown that there is an intrinsic trade-off to be made between accuracy and vulnerability. Indeed, it has been shown that the more accurate the system for a given system size, the more susceptible it was to a same proportionate level of attack.

D. Generic Security Mechanisms

Guided by the understanding of attack mechanisms and of their consequences on different NCS, some generic security mechanisms for coordinates-based systems have been proposed.

Kaafar et al. propose the use of a *Surveyor Infrastructure* to track normal behavior of nodes positioning using a *Kalman filter* [64], and then share such knowledge with normal nodes [65]. Surveyor nodes (or Surveyors in short) form a group of trusted (honest) nodes, scattered across the network, which use each other exclusively to position themselves in the coordinate space. Typically, during their embedding process, nodes use the Kalman filter parameters of a nearby Surveyor as a representation of normal, clean system behavior to detect and filter out abnormal or malicious activity.

Saucez et al. introduce a formal reputation model to detect misbehaving nodes [66]. The key idea is to associate a reputation to each node. This reputation informs on the reliability of a node: a high reputation refers to an honest node while a low reputation suggests a non-reliable node. The reputation of a node is based on how the node behaved in the past and how old it is in the system. To evaluate the reputation, two new entities are added in the system: the *Reputation Computation Agent* (RCA), a certificating agent, and the *Surveyors*, already introduced by Kaafar et al., that evaluate the nodes trust. Saucez et al. apply this model to Vivaldi, leading to an extension named *RVivaldi*. This solution, however, has the drawback of introducing a single point of failure, the RCA.

Wang et al. propose to secure network coordinates in two phases [67]. Firstly, it tries to protect the computation of coordinates by a customized Byzantine fault detection algorithm. The second phase is based on a TIV phenomena heuristic, and tries to secure the measurements from being delayed by malicious nodes. The said heuristic relies on the observation that authors have made, and that consists in noticing that delaying measurements is likely to make triangle inequality violations. Given the fact that edges that cause severe violations of triangle inequality are often under estimated in a malicious nodes-free coordinates system [54], Wang et al. propose to detect delay measurements using such an heuristic.

Sherr et al. propose another fully decentralized service, called *Veracity*, for securing coordinates systems [68]. Each node is associated with a set of verifying nodes and the node's coordinates are tested based on its error to the verifying nodes. If a node's coordinates have large error in predicting the delays to most of the verifying nodes, the node is considered as malicious and its coordinates will not be used by other nodes.

Zage et al. in [69] have explored the performance of network coordinates systems in adversarial environments. They present a solution based on outliers detection of coordinates and then use statistical detection mechanisms to differentiate good nodes and malicious nodes.

E. Summary

Following notations introduced by Table I in Sec. V-C, Table II summarizes the various security techniques discussed in this section.

	Generic	TIVs Sensitivity	Overhead
PIC Security	●	●	●
NPS Security	○	●	●
Surveyors-Kalman filter	●	●	○
RVivaldi	○	●	●
Byzantine fault detection	○	●	●
Veracity	○	●	●

TABLE II
COMPARISON OF SECURITY SOLUTIONS

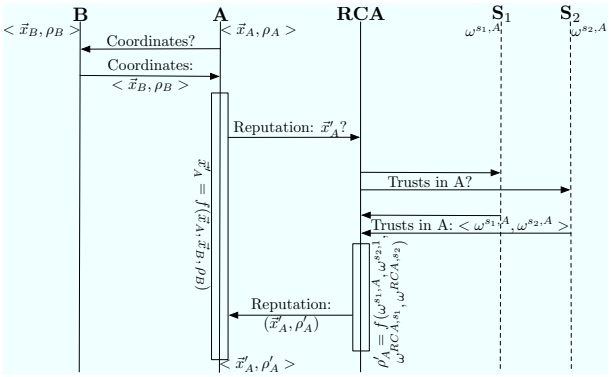


Fig. 8. Interactions between nodes in a reputation-based embedded system

The first column, labeled *Generic*, determines whether the solution is generic (i.e., it might be applied to any NCS) or not. The second column, named *TIVs Sensitivity*, shows if the security introduced in the NCS allows to get rid of TIVs. Finally, the last column (*Overhead*) discusses the overhead introduced by the security add-on. "○" denotes an absence of limitation while ●" indicates the presence of a limitation.

Despite the numerous improvements to NCS covered in this section, we are forced to observe that TIVs sensitivity remains a common feature. Further, most of the improvements introduce an overhead when calculating a node's coordinates. For instance, the reputation model introduced within Vivaldi (i.e., RVivaldi) requires a strong overhead in coordinates calculations, as illustrated by Fig. VI-E. Indeed, in a reputation-based approach, the new coordinates also depend on the reputation of the neighbors. When A updates its coordinates based on measurements with node B, it first contacts B to retrieve its coordinates ($\vec{x}_B$) and reputation (ρ_B). A then calculates its new coordinates as a function of its own coordinates, B's coordinates and B's reputation. Then, A contacts the RCA, to update its own reputation. The RCA aims at constructing a reliable reputation for any node in the embedded system. For this, it requires the Surveyors (S_1 and S_2 on Fig. VI-E). Surveyors are well-chosen nodes performing experience measurements and trust estimation on other nodes. The RCA also calculates its own trust to A's surveyors. Finally, the RCA evaluates the new reputation of A with all these parameters (i.e., the trust surveyors have in A and the trust the RCA has in A's surveyor). Further, as already mentioned, the introduction of the RCA leads to the single point failure risk, which is not suitable for an always-on service.

VII. Next Steps

Up to now, all the NCS discussed in this paper focus on predicting the network latency. It should be noted however, that if latency is the primary network metric that has been embedded in coordinates spaces, there are at least two approaches to including other network characteristics, such as bandwidth and jitter.

First, these could be made as additional dimensions in existing latency space. For instance, Oppenheimer et al. as well as Lee et al. have investigated the inverse correlation between latency and bandwidth [70], [71]. The correlation Oppenheimer found implies that network-aware decisions made in the latency space may result in good bandwidth characteristics.

The second approach is to embed additional performance indicator in their own metric-space, as it has already been performed with delays, and the existing NCS approaches. This is certainly a challenging, but useful task. Bandwidth, for instance, is an important performance metric that has already emerged as a candidate for network coordinates embedding. A set of applications, such as online media streaming or movie downloads require to select servers based on bandwidth in addition to latency. To the best of our knowledge, *Sequoia* [72] is the first NCS proposed to fill this gap. The key idea beyond Sequoia is that, under certain circumstances, the bandwidth might be seen as a *tree metric*. A set of measures is a tree metric if it can be derived from distances on a tree, that is, embedded on a tree [72]. For instance, the bandwidth is a tree metric when it primarily depends on the last-mile access link. Based on this observation, Ramasubramanian et al. derives the notion of *prediction trees*, where end hosts at the leaf connected via a network of virtual inner nodes with assigned link weights model latency or bandwidth. Sequoia maintains a collection of virtual trees between the participants and provides so latency or bandwidth predictions.

Sequoia mostly differs from various NCS surveyed in this paper as it does not aim at mapping any participant into a geometric space.

VIII. Conclusion

The last decade has seen the rising of a new class of large-scale globally distributed network services and applications. Those systems are characterized by the fact they can choose the path to use among a set of available ones. This selection might be done based on the path performance, such as the latency.

However, performing large-scale measurements is inefficient and not network friendly as injected probes consume undue network resources. To make network measurements more scalable, a new range of applications, called Network Coordinate System (NCS), has been developed and extensively studied those last years.

In this paper, we surveyed the various NCS proposed by the networking research community. We provide information on the general behavior of an NCS, that is modeling the Internet as a geometric space and characterize the position of any node in the Internet by a position (i.e., coordinates) in this space. We described NCS that are landmarks-based or fully distributed. We also discussed their limitations and open issues that still needed to be addressed in NCS. In particular, we focus on an important drawback: the security. We reviewed several potential attacks and explained how NCS might be improved in order to be more secure.

All NCS described in this paper focused on latency predictions. However, more and more applications require to measure or at least have an estimation of other network metrics, such as jitter and bandwidth. In this paper, we also explained the first solutions developed by the research community for predicting bandwidth.

Acknowledgments

This work is partially supported by the European Commission-funded 223936 ECODE project.

Mr. Donnet is funded by the Fonds National de la Recherche Scientifique (FNRS – Rue d'Egmont 5, 1000 Brussels – Belgium).

Mr. Gueye is supported by the EU under the ANA FET project (FP6-IST-27489).

References

[1] J. Liebeherr and M. Nahas, "Application-layer multicast with Delaunay triangulations," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, November 2001.

[2] Y. H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Proc ACM SIGMETRICS*, June 2000.

[3] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kasshoek, F. Dabek, and H. Balakirshnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, February 2003.

[4] P. Ratnasamy, M. Francis, M. Handley, R. Kerp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM*, August 2001.

[5] Gnutella, "A distributed peer-to-peer data-sharing system," http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[6] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "OceanStore: An architecture for global-scale persistent storage," in *Proc. International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, November 2000.

[7] "Bittorrent specification wiki," October 2008, http://wiki.theory.org/BitTorrentSpecification.

[8] B. Cohen, "Incentives build robustness in bittorrent," in *Proc. ACM SIGCOMM Workshop on Economics of P2P Systems (P2PECON)*, June 2003.

[9] A. Rowstron and P. Drusche, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM International Conference on Distributed Systems Platforms (ICDSP)*, November 2001.

[10] A. Rowstron, A.-M. Kermarrec, M. CAstro, and P. Druschel, "SCRIBE: The design of a large-scale event notification infrastructure," in *Proc. 3rd International COST264 Workshop on Network Group Communication (NGC)*, November 2001.

[11] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, April 1998.

[12] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gruniewicz, and Y. Jin, "An architecture for a global Internet host distance estimator service," in *Proc. IEEE INFOCOM*, March 1999.

[13] T. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *Proc. IEEE INFOCOM*, June 2002.

[14] Y. Shavitt and T. Tankel, "Big-bang simulation for embedding network distances in euclidean space," in *Proc. IEEE INFOCOM*, March 2003.

[15] Azureus, September 2008, http://azureus.sourceforge.net/index.php.

[16] M. Freedman, K. Laskhminarayanan, and D. Maziéres, "OASIS: Anycast for any service," in *Proc. USENIX Symposium on Networked Design and Implementation (NSDI)*, May 2006.

[17] P. Enge and P. Misra, *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamura Pr., 2001.

[18] J. A. Muir and P. C. Van Oorschot, "Internet geolocation: Evasion and counter-evasion," *ACM Computing Surveys*, 2008, to appear.

[19] University of Illinois at Urbana-Champaign, "IP address to latitude/longitude," 2001, see http://thegestalt.org/simon/ip2ll/.

[20] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for Internet hosts," in *Proc. ACM SIGCOMM*, August 2001.

[21] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of Internet hosts," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1219–1232, December 2006.

[22] B. Wong, A. Slivkins, and E. G. Sirer, "Meridian: a lightweight network location service without virtual coordinates," in *Proc. ACM SIGCOMM*, August 2005.

[23] S. Rewaskar and J. Kaur, "Testing the scalability of overlay routing infrastructures," in *Proc. Passive and Active Measurement Workshop (PAM)*, April 2005.

[24] J. Stribling, "Planetlab all pairs pings," Year, available from http://www.pdos.lcs.mit.edu/~strib/pl_app/.

[25] T. S. E. Ng and H. Zhang, "A network positioning system for the Internet," in *Proc. USENIX Annual Technical Conference*, June 2004.

[26] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft, "On the accuracy of embeddings for Internet coordinate systems," in *Proc. USENIX Internet Measurement Conference (IMC)*, October 2005.

[27] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, January 1965.

[28] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, "Lighthouses for scalable distributed location," in *Proc. 2nd International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2003.

[29] L. J. Corwin and R. H. Szczarba, *Calculus in Vector Spaces (Pure and Applied Mathematics): 2nd edition*. Marcel Dekker Inc., dec 1994.

[30] L. Lehman and S. Lerman, "PCoord: Network position estimation using peer-to-peer measurements," in *Proc. IEEE International Symposium on Network Computing and Applications (NCA)*, August 2004.

[31] L. Alvisi, J. Doumen, R. Guerraoui, B. Loldehofe, H. Li, R. Van Renesse, and G. Tredan, "How robust are gossip-based communication protocols?" *ACM SIGOPS Operating Systems Review*, vol. 41, no. 13, pp. 43–50, October 2007.

[32] A.-M. Kermarrec and M. Van Steen, "Gossiping in distributed systems," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 13, pp. 2–7, October 2007.

[33] K. Birman, "The promise, and limitations of gossip protocols," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 13, pp. 8–13, October 2007.

[34] H. Lim, J. C. Hou, and C.-H. Choi, "Constructing Internet coordinate system based on delay measurement," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 513–525, June 2005.

[35] I. T. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. New York, NY: Springer-Verlag, October 2002.

[36] L. Tang and M. Crovella, "Virtual landmarks for the Internet," in *Proc. ACM SIGCOMM Internet Measurement Conference (IMC)*, October 2003.

[37] W. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert," *Amer. Math. Soc.*, pp. 189–206, 1984.

[38] J. Bourgain, "On Lipschitz embedding of finit metric spaces in Hilbert space," *Israel Journal of Mathematics*, vol. 53, pp. 46–52, March 1985.

[39] Y. Mao and L. Saul, "Modeling distances in large-scale networks by matrix factorization," in *Proc. ACM SIGCOMM Internet Measurement Conference (IMC)*, October 2004.

[40] Y. Mao, L. Saul, and J. M. Smith, "IDES: An Internet distance estimation service for large network," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, no. 12, pp. 2273–2284, December 2006.

[41] D. Lay, *Linear Algebra and Its Applications*. Addison-Wesley, 1996.

[42] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, December 2000.

[43] M. Costa, M. Castro, R. Rowstron, and P. Key, "PIC: Practical Internet coordinates for distance estimation," in *Proc. 24th International Conference on Distributed Computing Systems*, March 2004.

[44] M. Szymaniak, D. Presotto, G. Pierre, and M. van Steen, "Partical large-scale latency estimation," *Computer Networks*, vol. 52, no. 7, pp. 1343–1364, May 2008.

[45] H. Young, R. Freedman, T. Sandin, and A. Ford, *Sears and Zemansky's University Physics with Modern Physics*. Addison-Wesley, 2000.

[46] F. Dabek, R. Cox, K. Kaashoek, and R. Morris, "Vivaldi, a decentralized network coordinated system," in *Proc. ACM SIGCOMM*, August 2004.

[47] C. Lumezanu and N. Spring, "Playing vivaldi in hyperbolic space," University of Maryland, UMD-CS-TR 4843, November 2006.

[48] C. Lumezanu and N. Spring, "Measurement manipulation and space selection in network coordinates," in *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*, June 2008.

[49] H. Zheng, E. K. Lua, M. Pias, and T. G. Griffin, "Internet routing policies and round-trip times," in *Proc. Passive and Active Measurement Workshop (PAM)*, April 2005.

[50] S. Lee, Z. Zhang, S. Sahu, and D. Saha, "On suitability of euclidean embedding of Internet hosts," in *Proc. ACM SIGMETRICS*, June 2006.

[51] S. Lee, L. Zhang, S. Sahu, D. Saha, and M. Srinivasan, "Fundamental effects of clustering on the euclidean embedding of Internet hosts," in *Proc. IFIP Networking*, May 2007.

[52] Y. Shavitt and T. Tankel, "The curvature of the Internet and its usage for overlay construction and distance estimation," in *Proc. IEEE INFOCOM*, March 2004.

[53] C. de Launois, S. Uhlig, and O. Bonaventure, "Scalable route selection for IPv6 multihomed sites," in *Proc. IFIP Networking*, May 2005.

[54] G. Wang, B. Zhang, and T. S. E. Ng, "Towards network triangle inequality violation aware distributed systems," in *Proc. ACM/USENIX Internet Measurement Conference (IMC)*, October 2007.

[55] M. A. Kaafar, B. Gueye, F. Cantin, G. Leduc, and L. Mathy, "Towards a two-tier Internet coordinate system to mitigate the impact of triangle inequality violation," in *Proc. IFIP Networking Conference*, May 2008.

[56] J. Ledlie, P. Gardner, and M. I. Seltzer, "Network coordinates in the wild," in *Proc. USENIX Symposium on Networked System Design and Implementation (NSDI)*, April 2007.

[57] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A blueprint for introducing disruptive technology into the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 59–64, January 2003.

[58] J. Ledlie, P. Pietzuch, and M. I. Seltzer, "Stable and accurate network coordinates," in *Proc. International Conference on Distributed Computing Systems*, July 2006.

[59] Y. Bartal, N. Linial, M. Mendel, and A. Naor, "On metric ramsey-type phenomena," *Annals of Mathematics*, vol. 162, no. 2, pp. 643–709, 2005.

[60] Y. Chen, Y. Xiong, X. Shi, B. Deng, and X. Li, "Pharos: A decentralized and hierarchical network coordinate system for Internet distance prediction," in *Proc. Global Telecommunications Conference (GLOBECOM)*, November 2007.

[61] Y. Chen, Y. Xiong, X. Shi, J. Zhu, B. Deng, and X. Li, "Pharos: Accurate and decentralized network coordinate system," *IET Communications*, vol. 3, no. 4, pp. 539–548, April 2009.

[62] M. Kaafar, L. Mathy, T. Turletti, and W. Dabbous, "Virtual networks under attack: Disruption Internet coordinate systems," in *Proc. ACM CoNEXT*, December 2006.

[63] M. A. Kaafar, L. Mathy, T. Turletti, and W. Dabbous, "Real attacks on virtual networks: Vivaldi out of tune," in *Proc. ACM SIGCOMM Workshop on Large-Scale Attack Defense (LSAD)*, August 2006.

[64] R. E. Kalman, "A new approach to linea filtering and prediction problems," *Transactions of the ASME – Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[65] M. A. Kaafar, L. Mathy, C. Barakat, K. Salamatian, T. Turletti, and W. Dabbous, "Securing internet coordinate embedding systems," in *Proc. ACM SIGCOMM*, August 2007.

[66] D. Saucez, B. Donnet, and O. Bonaventure, "A reputation-based approach for securing Vivaldi embedding system," in *Proc. 13th EUNICE Workshop*, July 2007.

[67] G. Wang and T. S. E. Ng, "Distributed algorithms for stable and secure network coordinates," in *Proc. ACM/USENIX Internet Measurement Conference (IMC)*, October 2008.

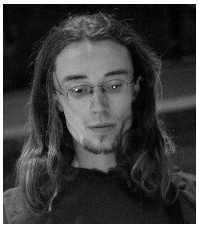
[68] M. Sherr, B. T. Loo, and M. Blaze, "Veracity: A fully decentralized service for securing network coordinate systems," in *Proc. 7th International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2008.

[69] D. Zage and C. Nita-Rotaru, "On the accuracy of decentralized network coordinate systems in adversarial networks," in *Proc. ACM Conference on Computer and Communications Security (CCS)*, November 2007.

[70] S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, and R. Fonseca, "Measuring bandwidth between planetlab nodes," in *Proc. Passive and Active Measurement Workshop (PAM)*, April 2005.

[71] D. Oppenheimer, D. A. Patterson, and A. Vahdat, "A case for informed service placement on planetlab," PlanetLab Consortium, Tech. Rep. PDN–04–025, December 2004.

[72] V. Ramasubramanian, D. Malhki, F. Kuhn, I. Abraham, M. Balakrishnan, A. Gupta, and A. Akella, "A unified network coordinate system for bandwidth and latency," Microsoft Research, Technical Report MSR-TR-2008-124, September 2008.



Benoit Donnet received his MS degree in computer science from the Institut d'Informatique of the Facultés Universitaires Notre Dame De La Paix (Namur - Belgium) in 2003. Mr. Donnet received his Ph.D. degree in computer science form the Université Pierre et Marie Curie in 2006. He is currently FNRS fellow at the Université catholique de Louvain in the IP Networking Lab (`http://inl.info.ucl.ac.be`). His research interests are in Internet measurements, focusing on scalable measurement techniques, Bloom filters, and traffic engineering techniques.

Bamba Gueye received the B.Sc. in Computer Science from the University Cheikh Anta Diop de Dakar, Sénégal. He received the M.Sc. degree in Networking in 2003 and the Ph.D. degree in computer science in 2006, both from the University Pierre et Marie Curie, France. He is currently a Post doc researcher at the University of Liège in the Research Unit in Networking group (`http://www.run.montefiore.ulg.ac.be`). His research interests include Internet measurements characterizing the Internet and measurement-based geolocation.

Mohamed Ali Kaafar received his Eng. degree and MS degree in computer science and Comp. Networking from the Ecole Nationale des Sciences Informatique in Tunisia in 2003 and 2004. Dr. Kaafar received his Ph.D. degree (performed at INRIA, Fr) in computer science form the Université Nice Sophia Antipolis in 2007. He is currently a permanent research scientist at INRIA Rhone-Alpes Grenoble (`http://planete.inrialpes.fr`). His research interests include Internet Security, Anomaly detection and Internet Measurements.