Bell Labs Technical Report
ITD-07-47566C

Coordinate-Based Routing for Overlay Networks

Hilt,Volker; Hofmann,Markus; Vasileious Pappas

03/12/2007

Coordinate-Based Routing for Overlay Networks

Vasileios Pappas [*]
UCLA, Computer Science
vpappas@cs.ucla.edu

Volker Hilt
Bell Labs/Lucent Technologies
volkerh@bell-labs.com

Markus Hofmann
Bell Labs/Lucent Technologies
hofmann@bell-labs.com

Abstract

Traditionally overlay networks perform routing in a way that mimics the underlying IP routing protocols. In this paper we propose a new approach to overlay routing that is based on network coordinates. The core idea is that routing is performed entirely within the coordinate space of a network coordinate system. The main benefits of this approach are that it is scalable to a large number of overlay nodes with a message complexity of O(N) while providing routing decisions that are close to optimal in terms of path delay and error resilience. Furthermore, coordinate based routing allows the realization of many different overlay routing schemes and this flexibility makes its suitable for the implementation of a large variety of overlay networks over a single infrastructure.

1 Introduction

Overlay networks enable the implementation of various Internet services, which the current network infrastructure cannot easily support. For example, application level multicast [8, 2, 31, 7] can successfully overcome the hurdles that the deployment of network level multicast faces. For similar reasons, overlay networks have been proposed in the past to provide quality of service [29] and protection against denial of service attacks [18]. Furthermore, apart from supporting new types of services, overlay networks can considerably improve the performance and the reliability of the current Internet paths [24, 5, 15, 6].

Even though overlay networks are capable of supporting all these new types of services, little improvements have been made on the underlying principles of routing in overlay networks. Most of these networks employ routing schemes that simply mimic the routing algorithms of the underlying IP network. For example, both RON [5] and ESM [8] use a link state routing protocol to build their overlay unicast and multicast forwarding paths respectively. This approach of routing within an overlay network comes with a considerable cost: these systems cannot scale to a large number of overlay nodes N, given that each node has to measure its distance (network delay) to all other nodes. This leads to a network-wide message complexity of $O(N^2)$. Other approaches of building overlay networks abandon the goal of achieving the best overall performance (in terms of delay) in favor of being more scalable. For example, one-hop source routing [15] is scalable but it cannot construct overlay paths that minimize the end-to-end delay. Similarly, Yoid [2] and HMTP [31] compromise the performance of the overlay multicast paths in favor of scalability. Fundamentally, overlay networks currently rely on routing protocols that trade scalability for performance, and vice versa.

Furthermore, current overlay networks lack a common routing framework that enables the deployment of different types of overlay networks over a common infrastructure. This means that a service provider cannot easily use a single overlay network infrastructure to run multiple types of overlay networks or even multiple instances of the same overlay network. For example, in order to offer two types of service, multicast and improved end-to-end reliability, based on ESM and RON respectively, a service provider has to maintain one instance of ESM for each multicast group, and at least one instance of RON. Clearly, these multiple instances come with additional message and maintenance overhead. In contrast, a common routing framework for all types of overlay networks can ease their deployment over the same infrastructure.

In this paper, we propose the use of network coordinate systems for overlay routing in order to overcome the above limitations. Network coordinate systems [21, 10, 9] map the Internet topology to a synthetic coordinate system, based on the round trip times between participant nodes. The distance between the coordinates of two hosts is a prediction of the actual round trip time between them in the Internet. This is different from geographical coordinate systems, which are based on the physical location of a host. Network coordinate systems are typically used today in order to select a nearby server out of a set of replicated servers, or in peer-to-peer networks [11] for the construction of efficient network structures. In this paper we further extend the use of network coordinate systems and we advocate that they can provide a common routing

[*] This work was done while working as an intern at Bell-Labs, Lucent Technologies

framework for overlay networks. In essence, we propose to use network coordinates at the overlay network layer just like network address are used at the IP layer.

Our proposed coordinate-based routing scheme for overlay networks has the following three advantages compared to the current routing schemes:

- *Performance*: Our framework allows the formation of efficient overlay network structures. Our performance evaluation shows that we can achieve a close to optimal path delay and error resilience.

- *Scalability*: Our framework scales well to a large number of overlay nodes and has a network-wide message complexity of $O(N)$. At the same time, the performance of the overlay paths stays independent of the system size.

- *Flexibility*: Network coordinates allow the implementation of various overlay routing protocols. For instance, by using the same framework one can implement an ESM [8] and a RON [5] type of service on top of the same overlay infrastructure.

In this paper, we introduce the overall concept of coordinate-based overlay routing and then focus on one specific application of coordinate-based routing: the implementation of an overlay routing scheme that improves the availability of end-to-end paths. This routing scheme, the one-hop coordinate-based overlay routing, uses just one overlay hop in order to overcome failures that appear at the IP layer. It enables, for example, a VoIP flow to use an alternate path through the overlay network in case of a link failure. We also provide the basic ideas on how to implement two other schemes based on network coordinates: multi-hop and multicast overlay routing.

The rest of the paper is structured as follows: In Section 2 we introduce the fundamental ideas of coordinate-based overlay routing. In Section 3 we present the one-hop coordinate-based overlay routing scheme and in Section 4 we evaluate its performance compared to plain IP and to two other overlay systems [5, 15]. In Section 5 we extend the idea of coordinate-based overlay routing for two other purposes: a multi-hop and a multicast routing system. In Section 6 we present the related work and we close with our summary in Section 7.

2 Coordinate-Based Routing

The core idea of coordinate-based overlay routing is to execute routing decisions within a coordinate space. In this routing architecture all overlay nodes are assigned to certain coordinates, based on their distance to other nodes, and packets are forwarded from node to node by following a well-defined trajectory in the coordinate space. Each type of application is allowed to define the exact shape of the trajectory based on its specific needs, i.e. the goals that it tries to achieve.

This approach of using network coordinates is fundamentally different from the use of coordinates in current applications. So far, network coordinates have been used to identify the closest node among a large number of nodes in a scalable fashion. In this paper we advocate that coordinate systems can provide more services than closest-node selection. They can implement a full-fledged overlay routing system, where the coordinates of nodes are used to determine the forwarding path through the overlay network.

2.1 Applications

There has been a wide variety of overlay applications proposed in the past, many of which can alternatively be implemented by using a coordinate-based routing architecture. In the following paragraphs, we first present three generic types of applications that can be built on top of a coordinate-based overlay system, and then we explicitly provide the types of applications that cannot benefit from such a system.

Applications seeking to improve e2e connectivity

Real-time applications, such as VoIP and media streaming, require high connectivity between participating nodes. Overlay networks have been proposed in the past in order to improve the Internet end-to-end connectivity [24, 5, 15, 6], by increasing path reliability and by minimizing path delay. The common idea behind all these systems is the use of overlay nodes as a way to route around failures that appear on the direct IP path. These systems strive to achieve one or more of the following goals: *i)* improve the end-to-end connectivity, *ii)* minimize the end-to-end delay, and *iii)* scale with the number of overlay nodes. Unfortunately, none of the currently proposed systems achieves all three goals. Indeed, while the first goal is achieved by all of them, the rest of the goals are only partially fulfilled. For example Detour [24] and RON [5] do not scale for a large number of nodes, while MONET [6] and one-hop source routing [15] cannot minimize the end-to-end delay. In contrast, we show that applications can achieve all three goals when utilizing our proposed coordinate-based routing system.

Applications relying on services from middle-boxes

Other applications make use of overlay nodes as middle-boxes that implement composable services. For example, applications on handheld devices may use middle-boxes in order to adapt content available on the Internet to the capabilities of the device [14]. Similarly, VoIP applications with different codecs may use an audio transcoding middle-box to set up a communication.

Yet another example is streaming databases [3, 16] that use one or more middle-boxes in order to deliver results from the data sources to the application. All these types of applications try to achieve the following two goals when selecting a middle-box: *i)* minimize the delay on the path between the server, the middle-box and the client, *ii)* identify the least loaded middle-box. In many cases, an application wants to balance the trade-off between the two goals: for instance a middle-box that provides a reasonably short path and is not overloaded. Coordinate-based routing provides applications with the means of achieving the first goal, i.e. minimize the path delay, in a very scalable way. To achieve the second goal, an application can implement additional mechanisms (e.g. to query the load of a small set of middle-boxes that provide short path delay).

Applications implementing end system multicast

End system multicast applications [8, 2, 7, 31] are another set of applications that can be implemented on top of a coordinate-based routing architecture. Application level multicast seeks to construct an efficient multicast tree that minimizes the end-to-end delay between participating nodes, by only using end-hosts as relaying nodes. Very roughly, each node makes peer connections with other nodes in close distance (in terms of network delay). The final outcome is a multicast overlay network with either a flat [8, 2, 31] or a hierarchical structure [7]. Clearly, a coordinate-based routing architectures can assist nodes in identifying the closest peers. But most significantly, it provides the means for those type of applications to construct overlay multicast trees in a flexible manner, without being bound to a certain overlay multicast protocol. Each application can implement its own multicast routing scheme and choose the most suitable multicast tree construction algorithm.

Applications not supported by coordinate routing

There is set of overlay applications that cannot directly benefit from a coordinate-based routing architecture. These are applications whose primary routing goal can not be mapped to end-to-end delay. For example there are overlay applications that try to utilize the path with the maximum available bandwidth or the path that provides the minimum packet loss. Clearly, a coordinate-based system cannot assist these applications in achieving their primary goal. Note that it may be possible to support applications which try to achieve more than one goal at the same time, with one being a short end-to-end delay. However, in this paper we do not consider these cases, in order to keep the whole system design simple.

Finally, a coordinate-based routing system is not suitable for applications that seek to achieve an end-to-end delay that is shorter than the delay provided by the direct IP path. As we show later in the paper (Section 4.2), our selection algorithm cannot always identify the overlay paths that are shorter than the direct IP path, given that these paths violate the triangle inequality [10]. The reason is that nodes, which violate the triangle inequality, cannot be accurately embedded in the coordinate space. The coordinate-system therefore cannot reliably identify overlay paths shorter than the direct IP path. However, this tension, between Internet triangle inequality violations and accurate embedding of nodes into the coordinate space, does not affect the ability of our selection algorithm to identify overlay paths that have network distances very close to the direct IP paths.

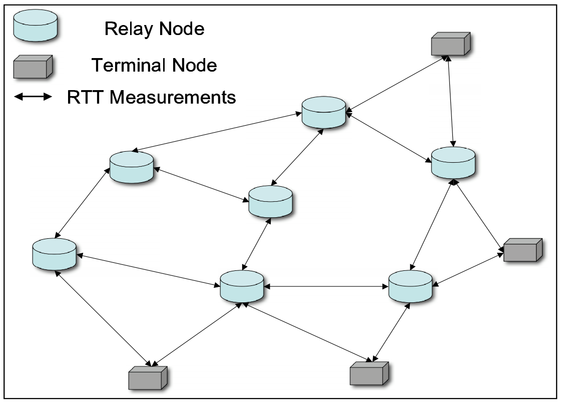

Figure 1: Two types of nodes participate in the overlay network: *relay* and *terminal* nodes. All nodes compute their coordinates based on the RTT measurements.

2.2 System Architecture

In the next paragraphs we describe the main properties of the coordinate-based routing system that are application agnostic. In later sections, we give the system details for each specific type of the above applications.

2.2.1 Types of Overlay Nodes

In our coordinate-based routing architecture (shown in Figure 1) we consider two types of nodes that participate in the overlay network:

- *Terminal Nodes:* These are all nodes which host applications utilizing the overlay network, and which do not forward packets for other nodes.
- *Relay Nodes:* These are all nodes whose main functionality is to forward packets, and which do not generate any application-specific traffic.

A node can, of course, have a dual role and can be a terminal and relay node at the same time. However, the distinction between the logical roles of terminal and relay nodes makes the design clearer. It also enables the creation of different configurations for a coordinate-based overlay network. In one type of configuration, all nodes implement both the terminal and relay node functionalities. This corresponds to a peer-to-peer network in which each node can use all other nodes as relay points to reach a destination. The opposite configuration is an overlay network with dedicated relay nodes distributed in the core of the Internet and terminal nodes at the endpoints. This configuration may be deployed, for example, by a service provider that seeks to build a common infrastructure for the support of various overlay applications. This infrastructure can then be utilized by a diverse set of customers. For instance, an overlay service provider running a coordinate-based routing system can offer both an application level multicast service to a video streaming provider and a reliable end-to-end path service to a VoIP provider.

A major difference between relay and terminal nodes is that the former need to make their coordinates known to the system. Terminal nodes on the other hand do not need to make their coordinates public, given that they are not used by others for forwarding purposes. This characteristic enables terminal nodes to join and leave the system with a low overhead. Coordinate-based overlay networks are therefore relatively insensitive to high join and leave rates of terminal nodes. Relay nodes on the other hand need to publish their coordinates when joining and invalidate them on leave. This requires explicit join and leave operations. A configuration that optimizes for tolerance to high node churn therefore separates terminal and relay nodes. Relay nodes are deployed in the core of the network, e.g. by an overlay network service provider, since these nodes typically have more stable characteristics, such as long system uptime, or low variability of network delays. Terminal nodes on the other hand are usually more volatile and benefit from the low join and leave overhead. Furthermore, the number of terminal nodes can potentially be orders of magnitude higher than the number of relay nodes.

Routing decisions can either be made by the terminal nodes, or by the relay nodes, or even by separate entities, which we call overlay router nodes. If terminal nodes determine the overlay routes, they need to learn the coordinates of relevant relay nodes. This process may add a significant overhead to their join operation. However, it provides the flexibility to the terminal node to implement its own overlay routing algorithm. Making routing decisions by the relay or the overlay router nodes simplifies the joining process for terminal nodes and therefore provides better tolerance to high node churn.

2.2.2 Coordinate System Management

All participating relay and terminal nodes need to determine their coordinates in the network coordinate system. This requires each node to communicate periodically with a small and fixed number of other nodes participating in the coordinate system. Our system uses the Vivaldi [10] algorithm, which computes the network coordinates in a fully distributed manner. However, our system design is not bound to just one system of network coordinates, and thus other systems, such as GNP [21] or PIC [9], can be used instead of Vivaldi. It is interesting to note that relay nodes can take over the task of computing the coordinates for any terminal node that does not have this capability. Thus, currently deployed applications can take advantage of this overlay routing architecture without any modifications. For example, one can use this system in order to improve the end-to-end connectivity of a streaming application, by making the relay nodes compute the coordinates of the streaming server and client, and by providing the best relay node to the client (e.g. via SIP [23] or RTSP [25] signaling).

To make a routing decision, a node needs to know the coordinates of the source node, the destination node and the potential relay nodes. Given this information, the node can identify the best forwarding paths for each application-specific trajectory. The coordinates of the source node are known. The coordinates of the destination node can be retrieved either by directly contacting the destination or with the use of a coordinate lookup service. In many cases, the exchange of coordinates between source and destination can be piggy-backed on an application-level protocol. For example, coordinates can be exchanged during the SIP signaling for VoIP applications. Finally, the node making the routing decision needs to know the coordinates of all relevant relay nodes in order to be able to identify overlay forwarding paths. Again, the way this is achieved is not specific to design of the coordinate-based routing system. For example, one option is to use multicast for the propagation of the network coordinates, while another option is to use a gossip type of flooding protocol. Yet another option is a centralized lookup service that maintains these coordinates.

Network coordinates may change due to the variation of network delays. All relay nodes need to be updated with the new coordinates of the other nodes. The frequency of these updates depends on the frequency and the degree of changes in the network coordinates, and the message overhead that one is willing to accept. For example, if we assume a network of 5000 relay nodes, a coordinate system with 10 dimensions, an update frequency of one hour for each node, and a traffic multiplier factor of two due to protocol overhead, then each relay node learns the coordinates of all the other nodes by receiving a traffic of 9Mbytes per day or 111 bytes/sec, which is negligible. The update frequency only needs to be high enough to capture changes in the network. Failed relay nodes can be detected (and omitted) by terminal nodes. Note that this traffic only increases linearly as we increase the number of relay nodes or the frequency of the updates. Finally we should point out that the coordinates of terminal nodes are conveyed periodically only during an active session and only to nodes (relay and terminal ones) that are associated with the session.

In summary, relay and terminal nodes follow a different approach in computing and conveying their coordinates. This design choice is justified by the reduced message overhead associated with the management of coordinates, and is enabled by the functional separation of relay and terminal nodes.

2.2.3 Forwarding Path Setup

Finally, a terminal node needs to set up the overlay forwarding path that the coordinate-based routing protocol determines. This can be achieved with any of the current protocols that supports source routing. For example one can use either source IP routing, or IP tunneling [26], or the TURN protocol [22]. Another system that can provide the forwarding mechanisms for our coordinate-based routing system is I3 [28]. This system comes with the advantage of supporting arbitrary types of communication between Internet hosts, such as proxy, multicast and anycast forwarding.

3 One-Hop Routing

In this section we introduce the one-hop coordinate-based routing scheme, which is one specific application of coordinate-based overlay routing. One-hop routing denotes a routing scheme that uses at most one relay node to route traffic from a source to a destination terminal node, when the direct IP path between them is not available. One-hop routing is frequently used in overlay networking due to its simplicity. Furthermore, an overlay forwarding path that utilizes multiple relay nodes often provides little additional benefits compared to one-hop routing. It has been shown that the performance and resilience of end-to-end Internet paths can rarely improve by using more than one overlay hops: Andersen *et al* [5] found that in 98% of the cases the overlay path with the shortest delay had just one hop. In addition, Gummadi *at el* [15] showed that using a randomly chosen overlay path of one hop is usually good enough for the identification of working overlay paths, when the direct IP are not available.

The above results suggest that one-hop overlay routing is suitable for applications that seek to improve the connectivity of end-to-end paths, by utilizing an overlay path when the direct IP path is not available. Moreover, applications that require the service of middle-boxes can benefit from one-hop routing, whenever they seek to identify a short end-to-end path. Furthermore, this routing scheme can be used as a basic building block for multi-hop routing. Therefore, in this paper we mainly focus on the design and evaluation of one-hop routing schemes and later (Section 5) we extend this idea to coordinate-based multi-hop and multicast routing schemes.

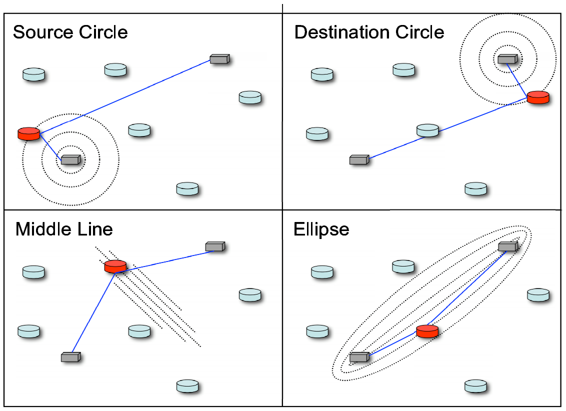

Figure 2: Examples of coordinate based routing policies for one-hop routing in a two-dimensional space. Each policy searches a certain area within the coordinate space.

Next, we first present the basic scheme for one-hop routing, named resilient shortest path, that improves the availability of end-to-end paths. Then, we provide an extension to that scheme, named advanced resilient shortest path, that improves over the selections of short overlay paths.

3.1 Resilient Shortest Path

The design goals of the one-hop overlay routing scheme are threefold. *First*, it should be scalable with the number of participating nodes, both relay and terminal ones. *Second*, it should be able to provide working overlay paths, when the direct IP paths between terminal nodes do not work. *Third*, it should be able to identify overlay paths with delays close to the direct IP paths.

The fist goal is achieved by utilizing the network coordinates, which leads to a network-wide message complexity of $O(N)$. The other two goals are met by utilizing two basic mechanisms. Application-specific events trigger these mechanisms. For example, applications may engage them either at the beginning of a new session, or periodically during a session, or whenever there is a disconnection at the direct IP path, etc. Next, we describe these two mechanisms.

3.1.1 Identifying Short Overlay Paths

The first mechanism used by the on-hop routing is for the selection of the relay node capable of providing a short overlay path, i.e. one that provides network delay between the source and the destination terminal node close to the delay of the direct IP path. This is achieved by utilizing only information about the coordinates of relay nodes, as well as the coordinates of the source and destination node.

One can implement a variety of policies for the selection of the most suitable relay node. The following are general examples for selection policies. They do not necessarily all provide a path with minimized delay and may achieve other routing goals. However, they illustrate that a overlay network developer can pick the most suitable policy out of a set possible policies.

- *Source Circle:* This routing policy selects as a relay (R) node the one that is at the closest distance to the source (S) node:
  $S^1(S) = \{ \quad R \quad | \quad \forall i \quad \|S - R\| \leq \|S - R_i\| \quad \}$
- *Destination Circle:* This policy selects as a relay (R) nodes the one that is at the closest distance to the destination (D) node:
  $S^2(D) = \{ \quad R \quad | \quad \forall i \quad \|D - R\| \leq \|D - R_i\| \quad \}$
- *Middle Circle:* This policy selects as a relay (R) node the one that is in the closest distance to the middle point between the source (S) and the destination (D) node:
  $S^3(S, D) = \{ \quad R, \quad M = (S + D)/2 \quad | \quad \forall i$
  $\|M - R\| \leq \|M - R_i\| \quad \}$
- *Middle Line:* This policy selects as a relay (R) nodes the one that is in almost at equal distance both from the source (S) and the destination (D) node:
  $S^4(S, D) = \{ \quad R \quad | \quad \forall i$
  $|\|S - R\| - \|D - R\|| \leq |\|S - R_i\| - \|D - R_i\|| \quad \}$
- *Ellipse:* This policy selects as a relay (R) node the one that minimizes the sum of its distances to the source (S) and destination (D) node:
  $S^5(S, D) = \{ \quad R \quad | \quad \forall i$
  $\|S - R\| + \|D - R\| \leq \|S - R_i\| + \|D - R_i\| \quad \}$

Each of this policies defines a certain area in the coordinate space. For example, if we assume a two dimensional space, then the first policy will select a relay node by searching within concentric circles centered at the source node. Figure 2 shows the shapes of the areas for the above policies in a two dimensional space. Note that the picture does not show the middle circle policy, given that its area looks like the area of the source or the destination circle policies, with the difference of having the center of the concentric circles at the middle point between source and destination node.

By definition, ellipse is the policy that seeks to minimize the delay of the overlay path, and thus it is expected to perform the best compared to any other coordinate-based routing policy for one-hop routing. In a later section (Section 4.2) we verify this assertion through simulation. Based on the above we use the ellipse routing policy for resilient shortest path overlay routing.

3.1.2 Identifying Working Overlay Paths

The second mechanism implemented by the one-hop overlay routing is used for the selection of working paths. After identifying the best relay node by utilizing one of the above policies, the source node tests if the selected overlay path is functional, i.e. if the overlay path between the source node and the relay node, as well as the path between the relay node and the destination node are functional. This can be done by simply trying to transmit data along the path or by active probing. In case that the overlay path does not work, the source node selects the second best node based on the routing policy, and repeats the same procedure. This procedure returns successfully with the first relay node that can provide a working overlay path.

It is interesting to point out that there is a possible conflict between identifying a working overlay path and identifying a short overlay path when the direct IP path does not work. Indeed, it is very possible that the direct IP path and a short overlay path traverse through the same parts of the network, and thus both of them can be prone to the same set of failures. Thus, one may need to probe more than one relay node in order to identify a working path. Furthermore, randomly selected relay nodes can be more useful in this case. Our evaluation (Section 4.2) shows that selecting the first-5 relay nodes based on the ellipse routing policy yields almost the same resiliency to failures as randomly selecting five relay nodes.

3.2 Advanced Resilient Shortest Path

There are two main issues when using the resilient shortest path routing, described in the previous section. *First*, due to the inherent errors in the embedding of nodes into the coordinate space, it is possible that the first node, selected based on one of the ellipse routing policy, may not be the best node that achieves the application objective. *Second*, in the basic mechanism for identifying a working overlay paths we have assumed that the paths are tried consecutively, i.e. if the first path does not work the second path is probed, and so on. This strategy adds additional delay to the routing process which may not be tolerable for some applications.

We therefore provide a simple extension to the resilient shortest path routing, with the dual goal of improving the

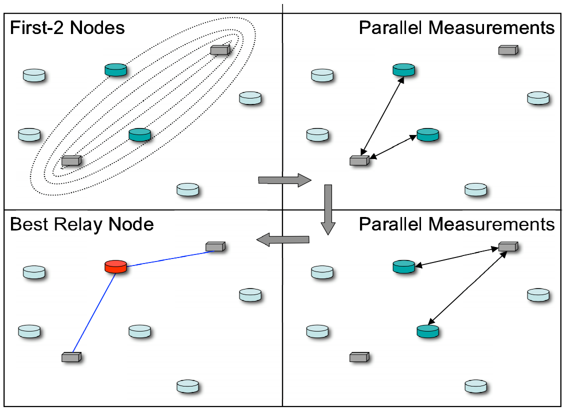

Figure 3: The advanced resilient shortest path routing improves the quality of the overlay path with parallel measurements at the first-k relay nodes selected by the routing policy.

quality of the selected overlay paths and of minimizing the response time in identifying working paths. The main idea is to perform parallel measurements to the first-k relay nodes that are selected by the routing policy. In this way, we are able to identify the best overlay path out of the k paths selected. Furthermore, given that we probe all of them at the same, we can minimize the time needed to identify a working path. The exact number k of parallel measurements is specific to the application needs.

Figure 3 gives an example that shows how the advanced scheme for one-hop routing works. We assume that there are two parallel measurements and that ellipse (S^5) is the policy used in this example. First the source terminal node identifies the best two relay nodes based on their coordinates. Then it probes both of them in parallel and identifies the one that provides a working overlay path that has the shortest delay. Note that in this example, the best node proves to be the one that was selected second by the ellipse routing policy.

4 Evaluation

In this section we evaluate the effectiveness of the one-hop overlay routing in selecting good overlay paths. First, we evaluate its ability to identify relay nodes that can provide short overlay paths. Second, we evaluate its ability to identify alternative working paths in the case that the direct IP path is not functional. Furthermore, we compare the performance of the one-hop coordinate-based overlay routing against the following other three routing schemes:

- *Optimal*: This is an overlay system that can provide the optimal working paths (in terms of the round-trip time). One possible implementation of such a system can be the RON overlay network [5].
- *Random*: This is an overlay network that selects the best relay node out of k randomly selected nodes. The one-hop source routing overlay network [15] is one possible implementation of such a system.
- *Plain IP*: This is just the IP network, i.e. it does not use any overlay nodes in order to improve the round-trip times and to overcome network failures.

The goal of this evaluation is to answer questions such as the following: Are the paths selected by the coordinate based system close to the optimal ones? How much better is the coordinate based system against the random scheme? How is the effectiveness of the coordinate based routing affected by the number of relay nodes and the number of terminal nodes?

4.1 Simulation Settings

In order to answer questions such the above we built a flow-level simulator. For each simulation scenario we select N relay nodes and M terminal nodes. Both types of nodes are randomly placed in a network topology. We use two types of topologies: *A)* IPS-level topologies that we constructed by using the Rocketfuel data [27], and *B)* AS-level topologies that where synthetically generated by using the BRITE topology generator [1]. The sizes of the ISP-level topologies are in the order of 100 nodes while the AS-level topologies are of 18000 nodes. However, most of the results that we present in this section come from simulations on AS-level topologies. We show only these results because they represent the worst-case scenario. The reason is that ISP-level topologies very rarely violate the triangular inequality (i.e. contain paths that connect two nodes through a third node with a shorter delay than the direct IP path) and thus the quality of the overlay paths selected by using the coordinates is as good as using the optimal routing scheme.

We simulate application-generated traffic as sessions between terminal nodes. The initiation of a new session follows a Poisson distribution and the duration of a session follows an exponential distribution with a mean of 3.5 minutes. These specific settings simulate VoIP calls [4]. However, we do not expect that our conclusions are affected by the exact distribution of the session duration. While the absolute numbers about the effectiveness of the one-hop coordinate-based routing shown may change, the relative numbers should be the same when compared to the optimal, the random and the plain IP routing schemes.

In addition, we simulate network failures caused by failures on links that are randomly selected. The duration of each link failure follows an exponential distribution, with a mean of 10 minutes. Again, note that the distribution of duration of the link failures does not affect our conclusions. In the case of a network failure, we assume that the routing tables are not updated instantly and that there is a period during which all application sessions that go through the failed link are not functional. The delay of routing updates follows a distribution similar to the distribution of routing update delays observed in the BGP system [19].

4.2 Simulation Results

In the rest of this section we provide simulation results with the goal of answering a set of questions, related with the effectiveness and the scalability of the coordinate-based routing scheme.

4.2.1 Which coordinate-based routing policy performs the best?

In Section 3.1.1 we presented a number of candidate routing policies for the selections of the shortest overlay path, and we argued that the ellipse is the one that can provide the best paths. We now provide results that support this claim (see Figure 4). These results are based on a simulation with 100 relay and 1000 terminal nodes randomly placed on AS-level topology. It shows the cumulative distribution function (CDF) of the delay on the overlay path that utilizes the fist selected relay node, for each routing policy. Note that the figure shows the actual delays of the paths as they are measured in the topology, and not the delay taken from the coordinate system. It also shows the CDF for the delay of the overlay paths selected by the optimal and the random routing scheme.

These results, as well as the results of other simulations on different topologies and overlay network sizes, suggest that the best coordinate based routing policy for the one-hop routing is the ellipse. Intuitively this can be expected given that this policy tries to minimize both the distance to the source and destination. Thus, if the coordinate system can predict the nodes' distances without errors, then ellipse routing policy can perform as good as the optimal scheme. Indeed, simulation results on the ISP-level topologies show that the ellipse performs as good as the optimal scheme. Unfortunately, for an overlay network that spans the Internet, the best node determined based on network coordinates may not always be the best node in the real topology, due to possible mapping inaccuracies of the coordinate system. Applications that seek to further improve the quality of the selected overlay paths can use the advanced resilient shortest path scheme and perform parallel measurements to selected relay nodes.

4.2.2 How many relay nodes are required for parallel measurements?

By using the same simulation setting as before (100 relay nodes, 1000 terminal nodes and AS-level topology) we compute the probability of identifying the shortest overlay path when we selected the first-k relay nodes. Table 1 shows the probability that the overlay node providing the shortest path available is among the overlay nodes selected based on the one-hop routing as well as the random scheme. The table shows that, for example, selecting four nodes is enough in order to identify the shortest path in more than 50% of the cases. In contrast the random selection performs poorly, given that it cannot identify the closest node even in 5% of the cases and for the same number of selected nodes.

Figure 6 shows how the number of relay nodes that were initially selected for the parallel measurements affects the quality of the selected overlay paths. It provides the average latency for a large number of sessions initiated between random terminal nodes, which use the shortest possible overlay path. The optimal scheme always finds the best overlay path, while the random and the coordinate-based find the best overlay path that is provided by the set of k relay nodes. For reference, we also plot the average delay of the sessions when they use plain IP routing. For this particular simulation setting, we see that the coordinate based scheme can find paths as good as the optimal scheme with only five parallel measurements. Interestingly, the random scheme of 10 parallel measurements finds paths worse than the coordinate based scheme with two parallel measurements.

Given that the average latency may be misleading sometimes when the variance ranges a lot, we also present the latency distribution. Figure 5 shows the cumulative distribution function (CDF) for the delays on the selected overlay paths for the different routing schemes. We see that with five selected nodes the coordinate based routing performs almost as good as the optimal scheme. Moreover, we see that the random selection performs quite poorly in identifying the shortest paths. For example with 5 selected nodes, the random scheme can identify paths that are on average around 25 msec longer than the paths selected by the coordinate based routing scheme. Interestingly, if we use the one-hop coordinate-based routing with just one node we can achieve shorter delays than the random scheme of five nodes.

4.2.3 How does the number of relay and terminal nodes affect the quality of selected paths?

Intuitively, by increasing the number of relay nodes in the network there is higher chance that the shortest overlay path between two nodes becomes even shorter. On the other hand, it is not clear if the different routing schemes can capitalize on this fact. Clearly, the optimal routing scheme by definition is able to do so, given that it always selects the shortest overlay path. Similarly, the coordinate based should be able to find shorter paths, under the

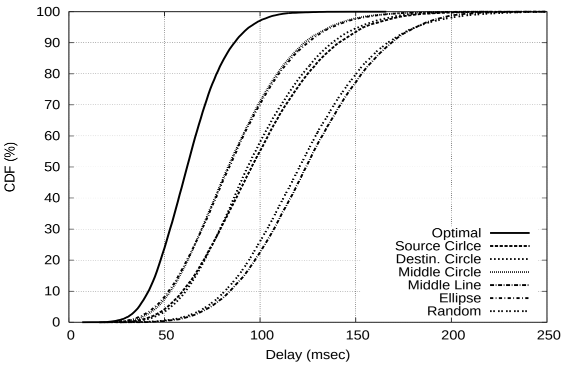

Figure 4: Network distances for the first relay node selected, compared to the shortest and a random overlay path.

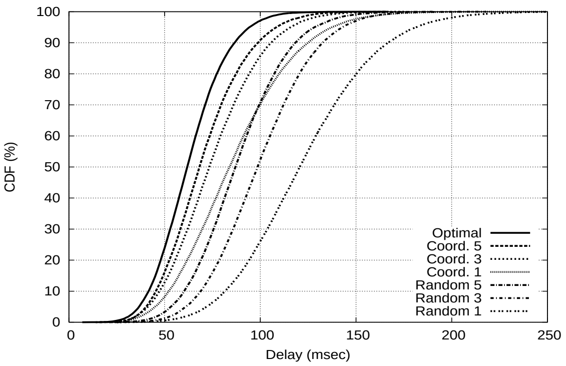

Figure 5: Network distances of the best overlay path selected out of the first-k relay nodes, when k is 1, 3 and 5.

Nodes	1	2	3	4	5	6	7	8	9	10
Coord.	18.30	31.80	46.60	59.80	65.20	70.80	75.00	77.50	80.50	83.10
Random	1.03	1.98	2.99	3.96	4.94	6.05	6.93	7.93	9.07	9.89

Table 1: The probability (%) of identifying the best available path, both for coordinate based routing and for random selection, when the number of selected overlay nodes ranges from 1 to 10. The number of relay nodes is 100.

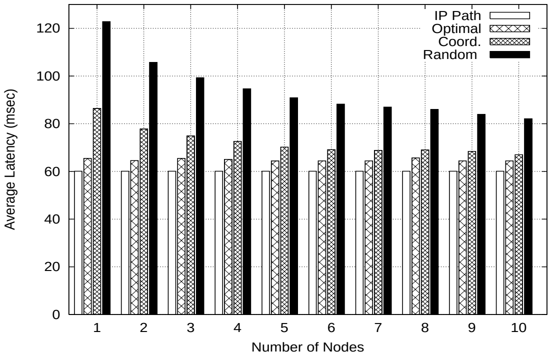

Figure 6: Average latency for the different number of parallel measurements.

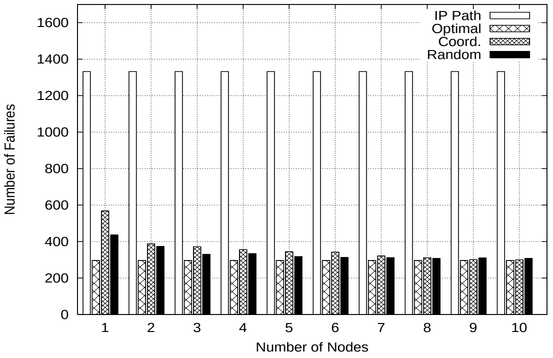

Figure 7: Number of failed flows, when the number of selected relay nodes changes.

assumption that the prediction errors stay the same. In contrast, the random scheme should not be able to take advantage of the additional short overlay paths, given that increasing the number of relay nodes also increases the number of paths with longer latencies.

Figure 8 shows the average latency for the shortest overlay paths, under the different routing schemes. The shortest IP path is also plotted, just for reference. As expected, the optimal overlay routing can identify shorter paths as the number of relay nodes increases. Interestingly, with 100 relay the average latency of the overlay paths is higher than the IP paths, but it becomes lower for 300 or more relay nodes. As expected, the figure shows that the random routing scheme cannot take advantage of the fact that there are more short paths when the number of relay nodes increases, because there are more long paths also. Surprisingly, the coordinate-based routing scheme cannot capitalize on this fact also, but for a different reason: Adding more relay nodes increases the probability of introducing a violation of the triangle inequality, which leads to less accurate predictions based on the coordinates, and consequently poorer selection of relay nodes. Fortunately, the figure shows that the average latency for the coordinate-based routing scheme remains the same, independently of the number of relay nodes, which means that the worst prediction and the larger number of short paths even out each other.

Figure 9 shows how the different routing schemes scale with the number of terminal nodes. These results show that the performance of the different routing schemes is independent of the number of terminal nodes. Indeed, the quality of the overlay path between two terminal nodes should not depend on the presence of other terminal nodes, but only by the presence of the relay nodes. This is true also for the coordinate-based routing, even though one may expect that adding more terminal nodes can also decrease the quality of the prediction. The reason that the coordinate-based routing is not affected is that terminal nodes are never used for relaying sessions, and thus adding more violations of the triangle inequality, by adding more terminal nodes, affects only the prediction errors for selecting terminal nodes rather than the prediction errors for selecting relay nodes.

4.2.4 Can the coordinate-based routing identify working paths when the IP path does not work?

In this section, we examine if the coordinate-based routing scheme can provide the same resilience for network failures as the two other schemes. Intuitively, one may expect that it will perform worse under the following argument: overlay paths with short delay are likely to share many links with the direct IP path, and thus when the direct path is not available there is a high probability that the shortest overlay paths are also not available. The following simulation results show under which situations this may happen.

Figure 7 shows the number of failed sessions that appear in a simulation with 1000 terminal nodes and 100 relay nodes, placed randomly on an AS level topology, when the number of relay nodes that are used to perform parallel measurements changes. A session is considered to have failed when it cannot be initiated or if it is disrupted (no end-to-end connectivity) for more than 10 seconds. The figure shows the failures for the three overlay routing schemes, in reference with the failures that appear by only using IP routing. Clearly, when only one overlay node is selected (i.e. there are no parallel measurements) the coordinate-based scheme does not perform as well as the optimal or the random scheme. As explained previously, this result intuitively is expected. On the other hand, when the number of parallel measurement increases, for instance to five, the coordinate-based scheme can identify working paths that are as good as the random scheme, and almost as good as the optimal scheme (which gives the minimum possible number of failed sessions). Clearly, the figure shows that the coordinate-based scheme can perform as good as the other too schemes. Next, we verify if this is true under different simulation settings.

Figure 10 gives the number of failed flows, when on average 25, 50 and 100 flows per second are created. The number of link failures on average was set to 3 failures per link per year. The figure shows that all schemes perform almost equally well, independent of the number of generated sessions. They can reduce the number of failed flows by two thirds. Similarly, Figure 11 gives the number of failed flows, when varying the average number of link failures per year. We consider the following three cases: each link fails on average 3, 6 and 12 per year. Also the number of flows created is 100 per second. This picture shows a similar pattern as the previous one. In conclusion, the above results show that the coordinate routing scheme provides almost the same resilience as the other two schemes, RON and one-hop source routing.

4.2.5 How does the number of relay and terminal nodes affect the selection of working paths?

Finally, we would like to answer the question of whether the effectiveness of the coordinate-based scheme to identify working paths scales with the number of relay and terminal nodes.

Figures 12 and 13 show the number of failed sessions, when the number of relay nodes and terminal nodes changes. Both figures show that the ability of the coordinate-based scheme, as well as the optimal and random one, to identify working paths when the direct IP path

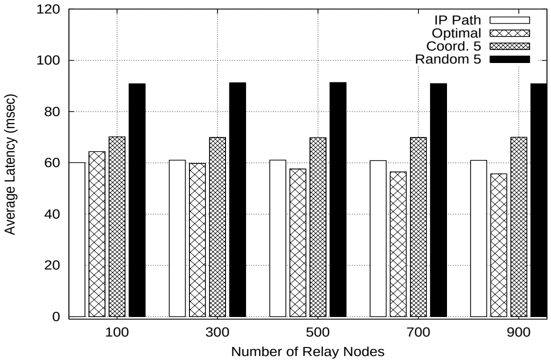

Figure 8: Average latency of the shortest overlay paths, when the number of relay nodes changes.

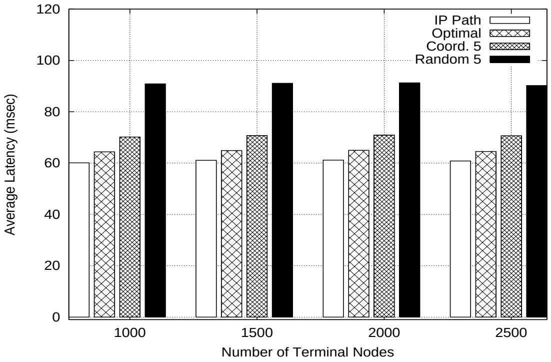

Figure 9: Average latency of the shortest overlay paths, when the number of terminal nodes changes.

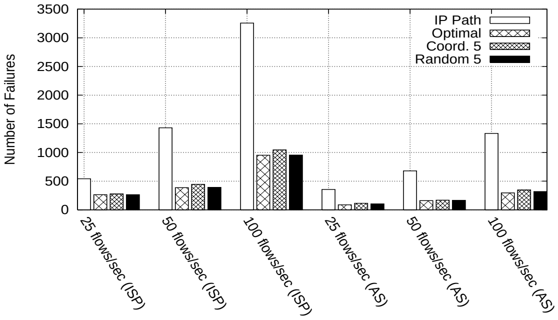

Figure 10: Number of failed flows, when the number of generated flows per second changes.

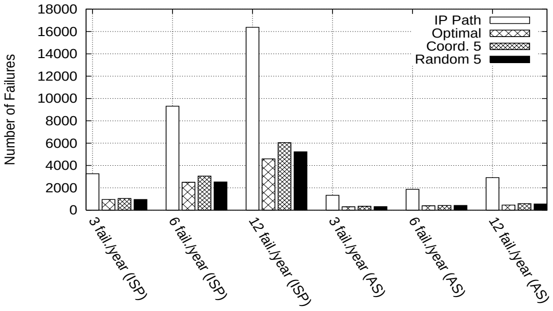

Figure 11: Number of failed flows, when the number of link failures per year changes.

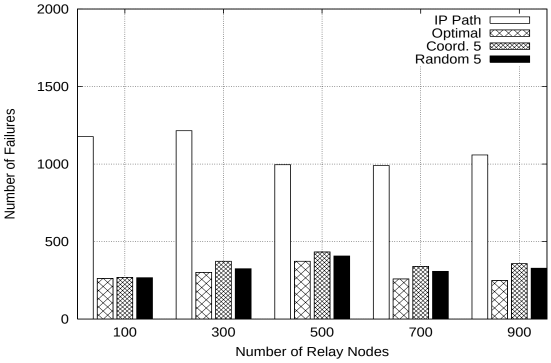

Figure 12: Number of failed flows, when the number of relay nodes changes.

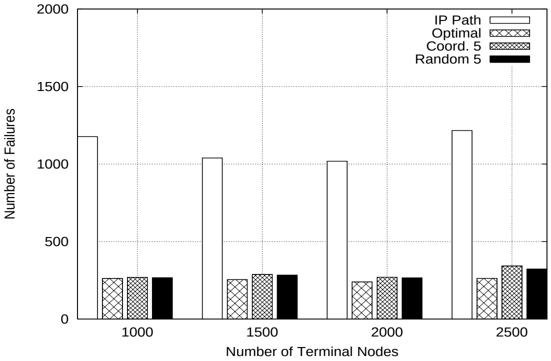

Figure 13: Number of failed flows, when the number of terminal nodes changes.

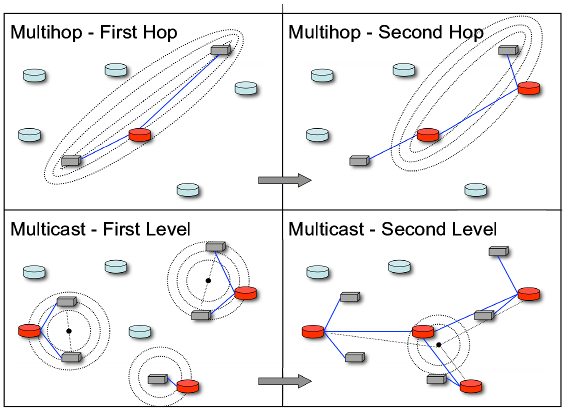

Figure 14: Examples on multi-hop and multicast coordinate-based overlay routing.

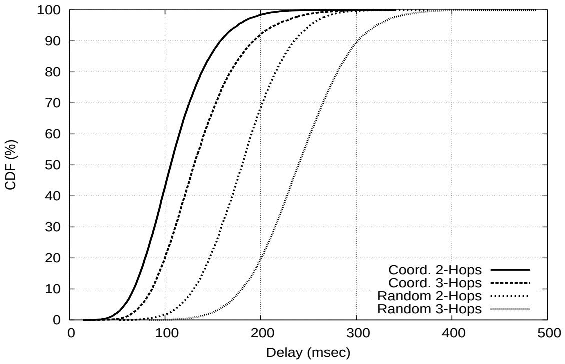

Figure 15: Latencies of overlay paths with two and three relay nodes.

does not work is independent of the size of the overlay network. Intuitively, one may expect that adding more relay nodes will increase the probability of finding a working path. On the other hand, given that even the optimal scheme doesn't improve with a larger number of relay nodes, we come to the conclusion that the remaining failures are not recoverable, i.e. are last hop failures.

5 Beyond One-Hop Routing

One can extend the idea of coordinate-based routing beyond the one-hop. The next two sections discuss how to implement two other routing schemes: a multi-hop and a multicast coordinate-based routing scheme.

5.1 Multi-Hop Routing

One-hop overlay routing is sufficient to overcome most failures and to improve the performance of end-to-end paths. On the other hand, there are applications that can benefit by utilizing more than one hop. For example overlay networks for anonymous communications [12, 17] require more than one hop in order to implement their functionality. Thus, a multi-hop overlay routing scheme that seeks to minimize the network delay of end-to-end flows would be beneficial for such applications[1]. In general, applications that require the use of more than one middlebox can improve their performance by utilizing this multi-hop coordinate-based routing scheme.

The basic idea is to recursively use the one-hop coordinate routing scheme, presented in the previous section, until the required number of overlay hops is met. We consider the following example. The source node S needs to communicate with the destination node D through an overlay path of three hops. It first runs the one-hop routing scheme between itself and the destination, which yields the relay node $R1$, and an overlay path with one hop ($S - R1 - D$). Then it applies the same procedure between itself and node $R1$, as well as node $R1$ and the destination node D, which yields relay nodes $R2$ and $R3$ respectively. At this point the source can set up an overlay path of three hops ($A - R2 - R1 - R3 - B$). In case that more nodes are needed it can apply the same procedure in any subsection of the current overlay path. The upper half part of Figure 14 shows graphically how the above procedure works for an overlay path with two relay nodes.

Figure 15 gives the delay on the overlay paths when using the above multi-hop coordinate based routing scheme, with two and three relay hops, and it compares it with a random multi-hop routing scheme, where each hop is randomly selected. It shows, that the coordinate based routing provides considerable improvements over a routing scheme that randomly selects relay nodes, given that it can identify overlay paths with two and three relay nodes that are shorter by 75 and 100 msec respectively. While a thorough evaluation is required, never the less the above results suggest that the multi-hop scheme shares the main advantages of the one-hop scheme: it is scalable and it can identify short overlay paths.

5.2 Multicast Routing

The coordinate-based overlay multicast routing works as follows: We assume that we have a set of terminal nodes that want to construct an overlay multicast tree, by utilizing the relay nodes. By applying a well-known clustering algorithm [13], such as k-means or hierarchical clustering, each nodes is assigned to a certain group. The main property of the clustering is that nodes that belong to the same group are close to each other, based on the distances derived from their coordinates. Furthermore, every group

[1] Note, that the anonymity of the sender is not necessarily compromised by routing on an overlay path defined by coordinates, given that that receiver or a relay node can only identify the rough direction of the sender.

is allowed to have a maximum number of members, in order to limit the number of connections originating from the relay node that is going to support the group. Then, each group computes the coordinates of its centroid and identifies the relay node that is closest to the centroid (within the coordinate space). This relay node is elected as the cluster-head of the group. Consequently, all terminal nodes in each group connect to the cluster-head of the group.

At this point, all terminal nodes are connected to a certain relay node but the relay nodes are not connected with each other. Thus, the above procedure is repeated but only for the cluster-heads, i.e. cluster-heads are assigned to different groups, the centroids for the new groups are computed, and so on. This procedure is repeated until all participating nodes, relay and terminal ones, are connected. Note that this algorithm can naturally be modified for the construction of multi-layer multicast trees [7], by creating a fully connected meshes within each cluster and allowing only cluster-heads to forward traffic to other cluster-heads.

The lower half of Figure 14 shows an example case of an overlay multicast network construction. There are five terminal nodes that want to set up a multicast session. Furthermore, we assume that each relay node can support at most three connections for this session. This restriction implies that the size of each group cannot exceed three nodes (one relay and two terminal nodes). Thus, the clustering yields 3 groups: two with two terminal nodes and one with just one node. Then, based on the position of the centroids, three relay nodes are selected and are connected with the terminal nodes. By repeating the same procedure for these three relay nodes, we identify a forth relay node, which connects with them. At this point the construction of the multicast tree is over. Note that under a different restriction for the maximum number of connected nodes we could have had a different multicast tree. For instance, if the maximum number of connections allowed by each relay for this session was five then all the terminal nodes could have been connected to just one relay node, actually the one that was selected last in the previous case.

6 Related Work

Detour [24] and RON [5] are overlay systems that can improve the performance and the reliability of end-to-end paths. Both systems implement routing by utilizing a link-state like protocol at the application layer. By following this approach any participating node constantly receives updates by all the other nodes, which leads to a network wide message complexity of $O(N^2)$. One-hop source routing [15] seeks only to improve the availability of end-to-end paths by following a different approach. When the IP path to a destination is not functional, the source randomly selects an overlay node that forwards all the traffic to the destination. In most cases at least one out of four randomly chosen overlay nodes can forward the packets (based on Planet-Lab experiments). Thus, one-hop source routing can be scalable with the number of participating nodes, but it cannot identify short overlay paths. In contrast, our scheme has a complexity of $O(N)$ and can identify short overlay paths at the same time.

Application level multicast [8, 2, 31, 7] has been proposed in the past in order to overcome the hurdles of network level multicast. Again, these overlay systems trade scalability for performance. For example ESM [8] constructs the optimal multicast tree in terms of network delay, but it cannot scale to a large number of participants, for the same reason that Detour and RON cannot scale. In contrast, Yoid [31] or HMTP [31] scale to large groups, but they cannot compute efficient multicast trees. Finally, NICE [7] seeks a compromise between scalability and performance by following a hybrid approach. At the higher level multicast nodes are connected like in ESM system, while at the lower level nodes are connected like in Yoid or HMTP. In contrast, our coordinate-based multicast routing system can implement any of the above three types of overlay multicast systems without compromising scalability or performance.

Network coordinate systems [21, 10, 9] provide an easy and scalable way to predict distances between hosts in the Internet. Coordinate systems are in particular useful to determine the distance between a potentially large number of hosts where measuring the round trip times between all hosts would involve prohibitively high costs. The basic idea of coordinate systems is that each host is assigned to a certain coordinate in a multidimensional Euclidean space such that their coordinates map the network distance between any two hosts in the Internet. While our coordinate-based routing uses network coordinates it differs from the previous work on coordinate systems in the sense that it extends their application beyond the closest node selection.

Meridian [30] is a system that provides a set of services useful for the construction of distributed applications. It achieves that without using network coordinates, which makes it more accurate in terms of selecting the most suitable node compared to a coordinate-based system. On the other hand, Meridian employs active probing between a moderate number of nodes. Thus, each Meridian lookup takes an additional delay (in the order of 200msec) that may be prohibitive for some real-time applications. Most importantly though, all services provided by Meridian are restricted by just one geometric shape, i.e. a circle centered in a specific point. Thus, the one-hop or multi-hop routing based on the ellipse routing policy cannot be implemented with Meridian. In contrast our coordinate-based routing system can support any geometric shape.

I3 [28] is an overlay system that supports a range of primitive functionalities for distributed applications. These include but are not limited to multicast, anycast and mobility support. I3 achieves that by offering a rendezvous-based communication abstraction. The I3 system can provide the mechanisms for the implementation of the forwarding paths in our coordinate-based routing system. P2 [20] is a new programming paradigm, that uses a declarative logic language for the implementation of different types of overlay applications. Both I3 and P2 are orthogonal to our system, but all of them have one goal in common: they seek to implement a range of overlay systems by providing a common framework. I3 provides the framework for the construction of forwarding paths, P2 provides the framework for the programming of overlay systems and our coordinate-based routing provides the framework for routing in overlay networks.

7 Conclusion

In this paper, we have proposed a fundamentally new approach to routing in overlay networks. This approach is based on the use of network coordinate systems. The main idea is to execute overlay routing decisions within the coordinate space of a network coordinate system. The routing decisions are governed by a routing scheme that determines how the system should identify relay nodes in the coordinate space. Many different routing schemes can be implemented depending on the specific goals of a overlay network. We have presented an example routing scheme, the one-hop coordinate-based scheme, that provides error resilience and determines the shortest path through an overlay network.

Our proposed coordinate-based overlay routing scheme comes with the following three main benefits compared to existing approaches to overlay routing: i) it provides a performance that is close to optimal and does not trade efficiency against other goals (e.g. scalability), ii) it has a message complexity of O(N) and scales very well over a large number of overlay nodes and iii) it enables the implementation of various routing schemes on the same overlay network infrastructure allowing service providers to offer different overlay networks to multiple customers.

With our new approach to overlay routing we attain a scalable, efficient and flexible platform that can easily host various overlay networks. It greatly simplifies the development of new overlay networks since it only requires the definition of the appropriate routing scheme that achieves the goals of the overlay network.

References

[1] BRITE. http://www.cs.bu.edu/brite/, 2006.

[2] YOID. http://www.icir.org/yoid/, 2006.

[3] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: A New Model and Architecture for Data Stream Management. In *Proc. of VLDB*, 2003.

[4] A.Markopoulou, F.Tobagi, and M.Karam. Assessment of VoIP Quality over Internet Backbones. In *Proc. of ACM INFOCOM*, 2002.

[5] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. of ACM SOSP*, 2001.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Rao. Improving Web Availability for Clients with MONET. In *Proc. of USENIX NSDI*, 2005.

[7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *Proc. of ACM SIGCOMM*, 2002.

[8] Y. Chu, S. Rao, and H. Zhang. A Case for End System Multicast. In *Proc. of ACM SIGMETRICS*, 2000.

[9] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet Coordinates for Distance Estimation. In *Proc. of IEEE ICDCS*, 2004.

[10] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proc. of ACM SIGCOMM*, 2004.

[11] F. Dabek, J. Li, E. Sit, J. Robertson, F. Kaashoek, and R. Morris. Designing a DHT for Low Latency and High Throughput. In *Proc. of USENIX NSDI*, 2004.

[12] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proc. od USENIX Security Symposium*, 2004.

[13] P. Drineas, R. Kannan, A. F. asd S. Vempala, and V. Vinay. Clustering in Large Graphs and Matrices. In *Proc. of ACM SODA*, 1999.

[14] A. Fox, S. Gribblea, Y. Chawathe, and E. Brewer. Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. In *Proc. of IEEE Personal Communications*, 1998.

[15] K. Gummadi, H. Madhyastha, S. Gribble, H. Levy, and D. Wetherall. Improving the Reliability of Internet Paths with One-hop Source Routing. In *Proc. of USENIX OSDI*, 2004.

[16] R. Huebscha, J. Hellerstein, N. Lanham, S. S. B. Loo, and I. Stoica. Querying the Internet with PIER. In *Proc. of VLDB*, 2003.

[17] S. Katti, D. Katabi, and K. Puchala. Slicing the Onion: Anonymous Routing Without PKI. In *Proc. of ACM HotNets*, 2005.

[18] A. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. In *Proc. of ACM SIGCOMM*, 2002.

[19] G. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proc. of ACM SIGCOMM*, 2000.

[20] B. Loo, T. Condie, J. Hellerstein, P. Maniatis, T. Roscoe, and I. Stoica. Implementing Declerative Overlays. In *Proc. of ACM SOSP*, 2005.

[21] T. E. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-Based Approaches. In *Proc. of IEEE INFOCOM*, 2002.

[22] J. Rosenberg, R. Mahy, and C. Huitema. Traversal Using Relay NAT (TURN). Internet Draft, 2005.

[23] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, 2002.

[24] S. Savage, T. Anderson, A. Aggarawl, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: A Case for Informed Routing and Transport. In *Proc. of IEEE Micro*, 1999.

[25] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Stream ing Protocol (RTSP). RFC 2326, 1998.

[26] W. Simpson. IP in IP Tunneling. RFC 1853, 1995.

[27] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. of ACM SIGCOMM*, 2003.

[28] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proc. of ACM SIGCOMM*, 2002.

[29] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. OverQoS: Offering QoS using Overlays. In *Proc. of ACM HotNets*, 2002.

[30] B. Wong, A. Slivkins, and E. Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In *Proc. of ACM SIGCOMM*, 2005.

[31] B. Zhang, S. Jamin, and L. Zhang. Host Multicast: A Framework for Delivering Multicast to End Users. In *Proc. of IEEE INFOCOM*, 2002.