Meridian-based Grouping in Overlay Networks

Meridian-basierte Gruppierung in Overlay-Netzwerken

Matthias Scheidegger, Torsten Braun, University of Bern (Switzerland)

Summary The performance of peer-to-peer and overlay networks depends to a large extent on their awareness of the underlying network's properties. Several schemes for estimating end-to-end network distances have been proposed to simplify this task. The mOverlay framework identifies groups of nodes that are near to each other in the network topology. Instead of distances between nodes mOverlay measures distances between groups. However, mOverlay's locating procedure has a number of drawbacks. We propose an alternate method for identifying groups using Meridian's closest node search. Simulation results based on PlanetLab measurements indicate that the Meridian-based approach is able to outperform mOverlay in terms of joining delay, the size of the identified groups, and their suitability for a distance estimation service. This alternate method for identifying groups is our main contribution. ►►► **Zusammenfassung** Die Leistungsfähigkeit von Peer-to-Peer und Overlay-Netzen hängt weitgehend von der Kenntnis der Eigenschaften des unterliegenden Netzes ab. Verschiedene Mechanismen zur Distanzschätzung wurden hierzu in der Vergangenheit vorgeschlagen. Das Verfahren mOverlay identifiziert Gruppen von Knoten, welche in einer Netztopologie nahe beieinander liegen. Statt Distanzen zwischen einzelnen Knoten zu bestimmen, werden Distanzen zwischen Gruppen ermittelt. Das mOverlay-Verfahren hat jedoch gewisse Nachteile. Daher wird in diesem Artikel ein alternatives, auf Meridian basierendes Verfahren zum Ermitteln von Gruppen vorgeschlagen. Meridian unterstützt die Suche von nächsten Knoten zu einem gegebenen Knoten. Der Hauptbeitrag dieses Artikels besteht in diesem alternativen Verfahren. Simulationen, die auf Messungen in PlanetLab beruhen, zeigen, dass das alternative Verfahren hinsichtlich Gruppenbeitrittsverzögerungen, der Größe identifizierter Gruppen sowie der Qualität der Distanzschätzungen Verbesserungen erzielt.

KEYWORDS C.2.4 [Computer Systems Organization:Computer-Communication Networks: Distributed Systems], C.2.1 [Computer Systems Organization: Computer-Communication Networks: Network Architecture and Design] distributed applications, peer-to-peer, overlay networks

1 Introduction

Peer-to-peer and overlay networks use logical topologies rather than the physical topology of the underlying network. This allows them to achieve many desirable aims like high scalability or high resilience to node failures. A single logical link connecting two nodes may in fact span many links on the physical network. This property is commonly referred to as the stretch of an overlay topology. Overlay networks usually perform better, if neighbors in the overlay topology are also close to each other in the physical network. On the other hand, logical links with high stretch can make an overlay topology more resilient to node failures.

In recent work [5] we have proposed an overlay distance measurement service using local groups similar to those in mOverlay. In addition, our approach is also able to detect whether or not remote nodes are close together. This is achieved by analyzing time series of distance measurements to remote hosts (obtained, e. g., using `ping`). Similarities in any two time series indicate that the respective remote nodes are close to each other. This enables two improvements. First, the service can be deployed easily, because remote hosts only need to respond to standard tools like `ping` and do not have to run any special software. Second, when looking at the network from a given location, far away groups are often indistinguishable from each other and can be viewed as a single entity. In such cases the local group only needs to store a single distance record for a set of remote groups, which improves the scalability of the service.

Initially, we planned to use mOverlay as a mechanism to identify local groups. However, we have found that replacing a part of mOverlay's locating algorithm with Meridian's closest node search improves its accuracy and reduces the time it takes for a node to join the overlay network. Our contribution in this paper is the modified locating algorithm, which we compare to the dynamic landmark procedure originally proposed for mOverlay in [2].

Although our work mainly aimed at optimizing the performance of our distance measurement service, the resulting algorithm is useful for more than this particular purpose. The grouping structure generated by the algorithm can also be helpful in constructing other kinds of overlay topologies, especially those concerned with optimizing transmission latency. Its distinguishing feature is that groups can be used as approximate equivalence classes with respect to transmission latency. Therefore, instead of computing an optimal topology connecting all nodes in a overlay network, we can solve the smaller problem to find the optimal topology connecting these groups.

The grouping structure may also influence the search algorithms used in peer-to-peer networks. For example, when a peer tries to find a specific document, it may be efficient to send the query to its local group first and only send it to other groups, if the initial request is not successful. Moreover, the query may be sent differently to members of the same group (e. g., using flooding) than to members of other groups. For peer-to-peer networks supporting replicas, the distance information between groups can also help finding the closest replica, which again optimizes the network load of the system. This also applies to latency-sensitive applications such as conferencing tools or online games.

2 Related Work

2.1 mOverlay

The mOverlay [2] framework uses a two tier overlay structure. At tier one, nodes that are close to each other form groups and communicate directly with other members of the same group. At tier two, groups select a number of nearby groups as their neighbors. The groups are chosen such that they can be used as equivalence classes concerning the distance metric. This reduces the endpoint-to-endpoint distance estimation problem to the much smaller problem of estimating distances between groups. Moreover, this structure can serve as a basis for constructing efficient overlay topologies, because it distinguishes between efficient short distance links inside the groups and potentially inefficient long distance links between groups. Using short links is usually more efficient since it reduces the traffic load on the network. Moreover, short links often have more capacity than long ones. Nonetheless, optimal overlay topologies can only be created using a mix of short and long links [9].

In order to decide whether or not a joining node belongs to a given group the following *grouping criterion* is used [2]: "When the distance between a new host *Q* and group *A*'s neighbor groups is the same as the distance between group *A* and group *A*'s neighbor groups, then host *Q* should belong to group *A*."

New nodes iteratively search for a group that meets this grouping criterion. When a node joins the overlay network it first contacts a rendezvous point and obtains contact information for a set of randomly chosen boot hosts. For each boot host it starts a locating process, which tries to find a suitable group for the node. Using several locating processes increases the robustness of the approach, because it reduces the probability that the algorithm finds a sub-optimal group or even fails to find a suitable group to join. The algorithm starts by contacting a boot host, which returns a set of distances between the boot host's own group and its neighbors. The joining node then measures and compares its own distances to these neighbor groups. If the grouping criterion is met, the process terminates and the node joins the group of the boot host. Otherwise, the algorithm chooses the neighbor group that is nearest to the new node and repeats the process. After a predefined number of unsuccessful iterations, or if all available groups have been visited, the new node creates its own group. When a node creates a new group it selects its neighbors from the closest groups it has seen during the locating process, and their neighbors. It then contacts each of the selected neighbors in order to allow them to adjust their own neighbor tables if needed.

We argue that mOverlay's structure, which is based on maintaining links to the nearest groups, is not optimal. This affects the performance of the locating algorithm and may make mOverlay less robust. In Section 3, we give more details on these problems, and we present an optimized, Meridian-based algorithm.

2.2 Meridian

Meridian [3] is a "framework for performing network positioning without embedding nodes into a global virtual coordinate space". It has another focus than mOverlay. Its three main functions are closest node discovery, central leader election, and multi-constraint search. For our work we use Meridian's closest node discovery. Meridian nodes form a loosely connected overlay network. They exchange information about other overlay nodes using a gossiping protocol and keep track of a fixed number of peer nodes. These nodes are sorted into non-overlapping, concentric rings of exponentially growing width around the Meridian node. The ith ring contains nodes with latencies between αs^{i-1} and αs^{i} from the center, and the outermost ring contains nodes with latencies αs^{i^*} and more (i^* designates the number of rings). Within each ring, the nodes are selected to maximize diversity. This is done by constructing k-polytopes (i. e., generalized polyhedra) based on the distances between the nodes. The algorithm selects the k candidate nodes that form the polytope with the largest hyper-volume.

A closest node search aims to identify the Meridian node that is closest to a given end system E in the network. To start the procedure we send a request to an arbitrary Meridian node. This node measures its latency to E and selects the nodes from its cache to which it has similar latency. It then contacts each of these nodes and asks them to measure and report their respective latency to E. The node with the smallest latency to E becomes the next hop, and the procedure repeats. When the next hop is only insignificantly closer than the current one the closest node search terminates and the current node is selected.

2.3 Distance Estimation

A considerable amount of work on network distance estimation has been published in recent years. One of the earliest designs, IDMaps [11], is a distance estimation service that relies on tracers placed at key locations throughout the network. The distance between two clients is estimated by the sum of the distances between the nodes and their respective nearest tracers, plus the distance between those tracers. Dynamic Distance Maps [12] uses a similar way to estimate distances, but uses the tracers to hierarchically decompose the Internet into regions. The main weakness of these approaches is that they need a large number of servers deployed at strategic locations throughout the Internet in order to be effective. This is an obstacle for the acceptance of these approaches, since it requires a considerable investment as well as coordination between Internet service providers. Nevertheless, they have the advantage that they could provide a global distance estimation service without necessitating changes to the end systems.

An important part of the work on network distance estimation focuses on coordinates-based approaches, which normally embed measured network distances in n-dimensional Euclidean space such that the Euclidean distance between two nodes is a good estimate of their distance in the network. GNP [13] is a prominent member of this family. Clients measure their distance to a fixed set of landmark nodes with known coordinates and compute their own coordinates using simplex downhill minimization. It has been argued that its fixed set of landmarks impairs GNP's scalability and makes it vulnerable to attacks and node failures. Consequently, more robust approaches like Lighthouse [14] have been proposed. Here, a varying subset of overlay nodes may be used as landmarks (called lighthouses). Vivaldi [15] does not use any landmarks. Instead, it passively monitors network traffic to obtain distance measurements and applies a distributed algorithm to iteratively adjust the coordinates of the nodes. Coordinates-based approaches have a number of advantages. They require little additional infrastructure in the network (GNP), or none at all (Lighthouse, Vivaldi). Furthermore, coordinates can be included in peer-to-peer messages. This enables nodes in a peer-to-peer network to estimate their distance to other nodes without having to issue a request to the service. Peer-to-peer topologies can also be built based on coordinates using geometric methods. A common disadvantage of the known coordinates-based approaches is that they can only estimate distances between nodes actively participating in the system. A global, coordinates-based distance estimation service would therefore require every end system on the Internet to run specific software, which is hard to achieve.

2.4 Grouping

The concept of grouping end systems has been used in various related works, either to reduce the complexity of overlay topologies or to achieve locality-awareness. IDMaps [11] creates groups of end systems with similar IP addresses, called *address prefixes*. The idea behind this form of grouping is that end systems with the same IP address prefixes tend to be close to each other since routing in the Internet is often hierarchical. However, there are many exceptions to this rule. For example, two end systems with adjacent IP addresses may be connected to the Internet using different technologies, such as ethernet or modem links, resulting in very different latency observed by nodes in the same group. M-coop [16] uses a similar approach but also considers the autonomous system topology of BGP to obtain more detailed information about the routing of IP addresses. Grouping based on IP addresses is used to construct end system multicast topologies in MULTI+ [10]. Similar receiver addresses tend to be assigned to the same subtree, leading to a multicast tree that approximately follows the locality of the receivers.

Dynamic Distance Maps [12] creates a hierarchical clustering of the Internet based on the measured latencies between a set of measurement servers. Groups are formed by assigning end systems to their respective nearest measurement servers.

Binning [17] is a concept useful for adding topology awareness to structured peer-to-peer networks. Nodes are assigned to *bins* based on their distance to a small set of landmarks. The scheme maintains a d-dimensional Cartesian coordinate space partitioned into zones. Each peer measures its distance to the landmarks and sorts the results in ascending order. Nodes are assigned to the same bin if they have the same landmark ordering. The effect of this scheme is that nodes from the same bin tend to be close to each other on the underlying network.

3 Meridian-based Locating

A problem of mOverlay is its topology. Because the groups choose neighbors from their close proximity, the logical links between the groups are very short. This affects the performance of the locating algorithm, since the algorithm follows the topology and thus can only make small steps towards the target node. If the target node is far away, taking bigger steps would be more efficient. Another problem is that mOverlay's topology is prone to so-called network-splits – partitionings of the system into two or more independent parts due to node or link failures.

In order to overcome these problems we have defined an alternative group locating algorithm based on both mOverlay and Meridian. We take the group concept from mOverlay but change the overlay structure. The groups no longer have neighbors. Instead, the group leaders become Meridian nodes. When a new node wants to join, it goes through the following procedure: First, the new node locates a boot node (i. e., a group leader) by sending a request to the rendezvous point. It then asks this boot node to start a Meridian closest node search with itself as target. The search returns the address of the closest group leader to the joining node. At this point, the new node checks the grouping criterion to find out whether or not to join this group. If the criterion is met the new node joins the group. Otherwise, it creates a new group and becomes a Meridian node itself.

Unfortunately, we cannot directly use mOverlay's grouping criterion, because in our Meridian-based approach groups do not have neighbors. We solve this problem using Meridian's node cache. The group leader found by Meridian's closest node search selects a randomly chosen set of *verification nodes* from its node table and creates a list of addresses and latencies to these nodes. The new node receives this list and in turn measures its latency to each of the verification nodes. This provides us with two comparable sets. Furthermore, because mOverlay's grouping criterion is formulated in general terms we also need to specify exactly when two distances can be considered "the same". We say distances x and y are the same if

$$(x \geq y) \wedge ((1-g) \cdot x \leq y)$$

or

$$(x < y) \wedge ((1-g) \cdot y \leq x)$$

holds for a *grouping threshold* $g \in [0, 1)$. The test checks the relative difference between two distances. For example, with a grouping threshold of 0.05 we consider two distances the same if they are within ±5% of each other. A new node joins a group, if the test above succeeds for every verification node.

We believe that this combined approach to grouping nodes solves the problems discussed above. Meridian's closest node search makes the search more efficient. The approach is also less prone to network-splits than mOverlay because Meridian nodes maintain a more diverse set of peer nodes. The loose overlay structure also makes the system more resilient to node failures. A possible drawback of our algorithm is that it only checks the grouping criterion for a single group, which bears the danger that the algorithm might skip over the optimal one. Fortunately, the results in Section 5, more precisely the good quality of the identified groups as shown in Fig. 1 and 3, suggest that this kind of error is rare.

4 Implementation of Simulators

In order to compare the performance of mOverlay's locating algorithm to our Meridian-based algorithm we have implemented simulators for the two approaches. Both simulators are based on a black box network model given by a matrix of the end-to-end latencies between each pair of endpoints in the simulation. For our experiments we use a matrix derived from all-sites ping data measured on PlanetLab [6]. Due to the nature of PlanetLab, we cannot assume that these measurements are fully representative for all settings, in which the approach might be applied (more detailed considerations can be found in [8]). Nevertheless, we believe that they are realistic enough to allow for meaningful analysis of the approach. In both simulators the nodes join the overlay network one after the other, in pseudo-random order (given by the seed value). For each node we record the time that expires until it joins a group or creates its own group. When the simulation ends, we examine the resulting groups according to several criteria, which we discuss in Section 5.

We simulate mOverlay with a simple message-based approach, where each message fits into a single packet and the message processing at a node does not take any time. Thus, a request-response message exchange takes exactly one round-trip time to complete, which is a lower bound for any real implementation of the framework. Furthermore, we skip mOverlay's initial request to the rendezvous point because the performance of this step depends heavily on the implementation of the mechanism (e. g., a well-known address, a DNS-based approach, etc.) and possibly on the placement of the rendezvous point. In the simulator, the locating processes of a joining node run in parallel and stop, when one of them finds a group that meets the grouping criterion. A locating process also stops, if its next hop would be a group it has already visited. If none of the locating processes are successful, the joining node gives up and creates a new group. Locating processes keep a list of visited groups. When a new group is created its neighbors are selected from the lists of all its locating processes. The first two joining nodes are special cases. They automatically create new groups because the grouping criterion cannot be evaluated without further nodes. As mentioned in Section 3, mOverlay does not define how to test, if two distances are the same. However, we need to test this to check the grouping criterion. We have used the test from Section 3 also for the mOverlay simulation, because it is a natural choice.

In contrast to the mOverlay simulator, where we implemented all necessary messages, we did not implement the Meridian approach ourselves. Instead, we have used the official Meridian C++ implementation [4]. We have written wrapper code to redirect any messages to a simulation back-end instead of the network, and we have changed Meridian's time-keeping code to use the simulation time instead of the system clock. Each Meridian node is now a C++ object in the simulator rather than a physical node on the network. When it sends a packet the simulator determines the appropriate transmission latency using the underlying network model and schedules the packet arrival at the destination node accordingly. The wrapper objects also evaluate the grouping criterion at the end of a joining procedure and create a new group if necessary. The simulation back-end is event-based. There are three kinds of events: one for inserting a new node into the scenario, one for triggering Meridian's periodic gossip protocol, and one for packet arrivals at a Meridian node. We start the simulation by scheduling node join events every seven seconds (which corresponds to Meridian's default gossip interval). When a node joins it starts by sending a closest node query to a Meridian node. This search is handled entirely by the original code. When the query returns, the joining node contacts the identified closest node to retrieve a list of verification nodes, which the wrapper code extracts from the Meridian object's latency cache. In the simulator we use a maximum of five verification nodes.

5 Evaluation

5.1 Simulation Scenario

For the simulations we have used a matrix of round-trip times between 77 PlanetLab nodes, based on all-sites ping data from PlanetLab [6]. The simulator estimates the one-way delay between two endpoints by dividing the appropriate round-trip time by two. At the time of writing, 694 machines hosted by 335 sites were part of PlanetLab [7]. This means that each site hosts only slightly more than two machines on average. Consequently, we can expect to find groups of only a few nodes each in our scenario, especially since the 77 nodes in the network model were randomly selected from the available PlanetLab nodes. Using the same data set [6], we have also studied how each of the elements in our matrix changes during the course of one day following the initial measurement. These additional observations were made in intervals of 15 minutes. The resulting time series are used for evaluation, as described in Section 5.2.

5.2 Evaluation Criteria

While the comparison of the joining delays is straightforward, quantifying the quality of the identified groups is not. Grouping can exhibit two kinds of errors, false positives and false negatives. If a node erroneously joins a group, this is considered a false positive and increases the error of grouping. A false negative occurs, if a node erroneously does not join a group and creates a new one instead. This results in too many groups and impairs the efficiency and scalability of the overlay network. Unfortunately, due to the black box nature of our network model, we cannot say a priori, whether a node should join a group or not and are thus unable to directly identify false negatives. Nevertheless, we can define three criteria for the quality of the identified groups.

- Members of a group should be close to each other. Accordingly, we compute the mean round-trip time between members of the same group. Groups with only one node are ignored in this case.
- Bigger groups are preferable because they reduce the complexity of the overlay network. We use the average number of nodes per group as the second criterion.
- One important assumption in mOverlay is that, if two nodes A and B are in the same group, the distances $\overline{AC}$ and $\overline{BC}$ to a node C outside the group are nearly the same. This property enables significantly better scalability of the service. However, it must also hold over time. Otherwise, we would have to permanently reorganize the groups. We define the third criterion accordingly: If A and B are in the same group, $\overline{AC}_t$ should be a good prediction for $\overline{BC}_t$, even if both values vary with the time of measurement t.

We verify this using the time series of round-trip times between the two nodes. Two measurements $\overline{AC}_t$ and $\overline{BC}_t$ are out-of-band of each other if

$$\overline{AC}_t > (1-b)\cdot\overline{AC}_t > \overline{BC}_t$$

or

$$\overline{BC}_t > (1-b)\cdot\overline{BC}_t > \overline{AC}_t$$

holds for a relative margin (or band) $b \in [0,1)$. The *out-of-band ratio* between two nodes is the ratio of out-of-band measurements in the respective time series. In this paper we use a margin of 10% ($b = 0.1$).

5.3 Group Quality

As a first comparison we look at the quality of the groups identified by mOverlay and our Meridian-based approach. For both we use parameters that we have found to produce near optimal results considering the above criteria for group quality. We set the maximum number of neighbors for mOverlay groups to eight and the number of parallel locating processes to five. For the Meridian-based approach we set the maximum number of verification nodes to five.

Fig. 1 shows the mean round-trip times between group members for the grouping thresholds 1%, 2%, 5%, 10%, 20%, and 50% (using a logarithmic scale for better readability). The graphs use a dot-and-whisker format showing the mean with a 90% confidence interval, obtained by running the simulation with 100 different seeds. We have also slightly staggered the graphs along the horizontal axis to improve readability. As Fig. 1 shows, a lower threshold also leads to smaller distances between group members for both approaches. The effect is much bigger for mOverlay because the grouping threshold affects every iteration of the locating process, while the Meridian-based locating algorithm only uses the grouping threshold for its final step. mOverlay's locating process checks the grouping criterion in every step (see Section 2.1) and thus has a chance to settle with a suboptimal group if the threshold is large. Nevertheless, the round-trip times between group members of mOverlay are always bigger on the average than those of the Meridian-based approach. Moreover, the confidence intervals for mOverlay are bigger. We conclude that the Meridian-based approach performs better than mOverlay with respect to the first criterion.

The second aspect we examine is the average number of nodes per group. Fig. 2 shows the group size for the same grouping thresholds as Fig. 1. The groups identified by the Meridian-based approach are bigger for grouping thresholds up to 10%. In contrast, mOverlay identifies much bigger groups with grouping thresholds above 10%, but this comes at the price of much higher round-trip times between group members. As expected, group sizes are rather small because of the wide distribution of the nodes.

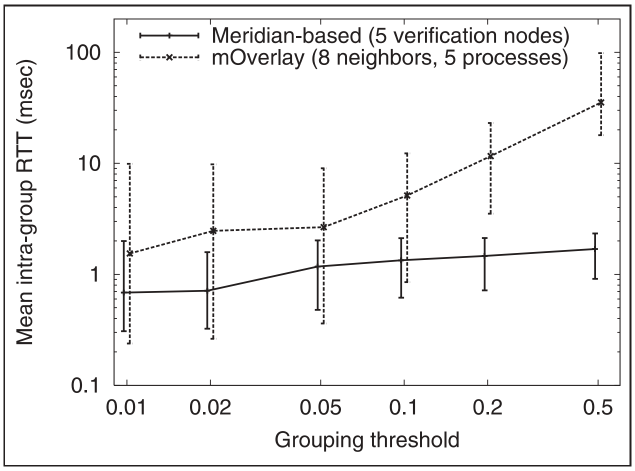

Figure 1 Mean intra group distance.

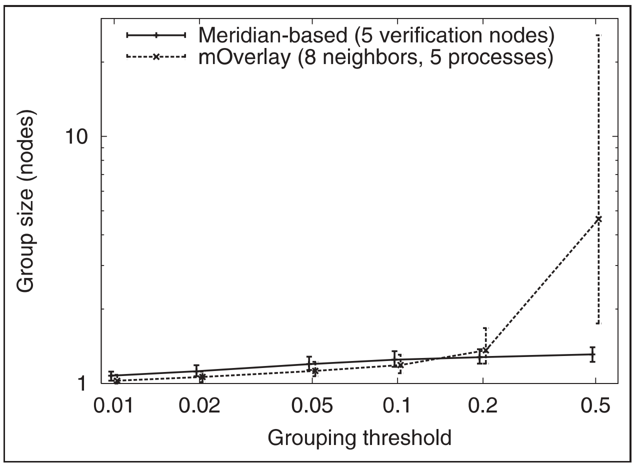

Figure 2 Average nodes per identified group.

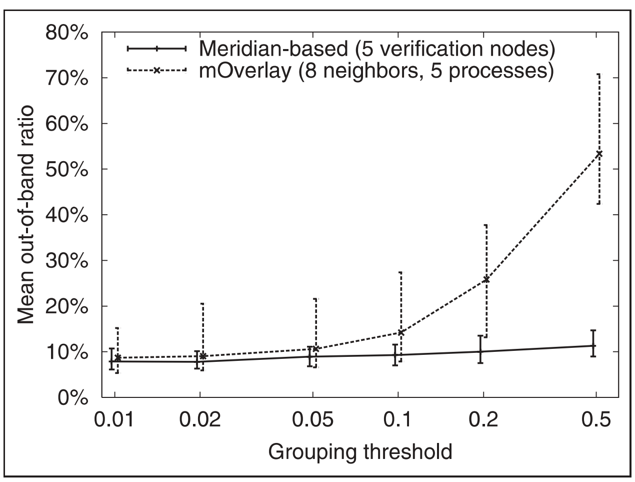

Figure 3 Mean out-of-band ratio using a 10% band.

If the identified groups shall be used as a basis for a distance estimation service they must also have a low out-of-band ratio. We look at this aspect using again the same parameters for grouping threshold and a 10% band for the out-of-band test. The results can be seen in Fig. 3.

The Meridian-based approach has a smaller out-of-band ratio than mOverlay for all grouping thresholds, and it shows less variance. Again, mOverlay shows high sensitivity towards the grouping threshold while the out-of-band ratio of the Meridian-based approach only increases slightly with growing grouping threshold.

5.4 Joining Delay

In addition to a good quality of the identified groups it is also desirable to find the groups in the shortest time possible. We compare the two approaches using the same parameters as in Section 5.3.

Fig. 4 shows the joining delay per node for several grouping thresholds. Again, mOverlay proves to be much more sensitive towards the grouping threshold than the Meridian-based approach. Moreover, unless the grouping threshold is extremely high the Meridian-based algorithm finds the local group much faster than mOverlay.

The joining delay of mOverlay nodes is not only sensitive to the choice of grouping threshold. Reference [1] discusses the influence of the maximum number of neighbors per group and the number of parallel locating processes.

5.5 Effects on Overlay Topology Construction

As mentioned in Section 1, the groups identified by our algorithm can be utilized to simplify the creation of optimal overlay topologies by reducing the problem's size. In order to illustrate this with relation to the simulation results presented above, we consider the construction of an optimal overlay topology with

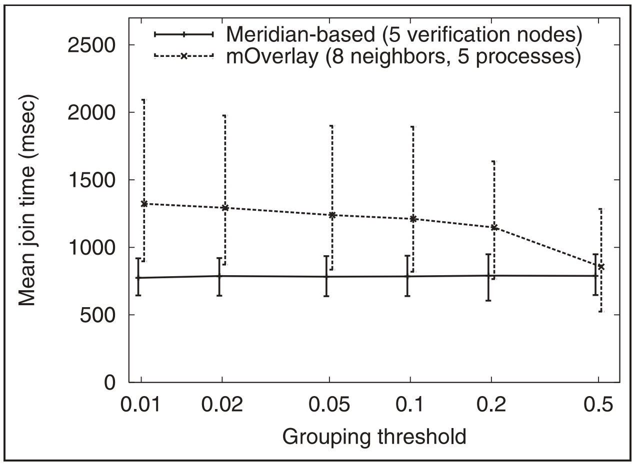

Figure 4 Mean joining delay per node.

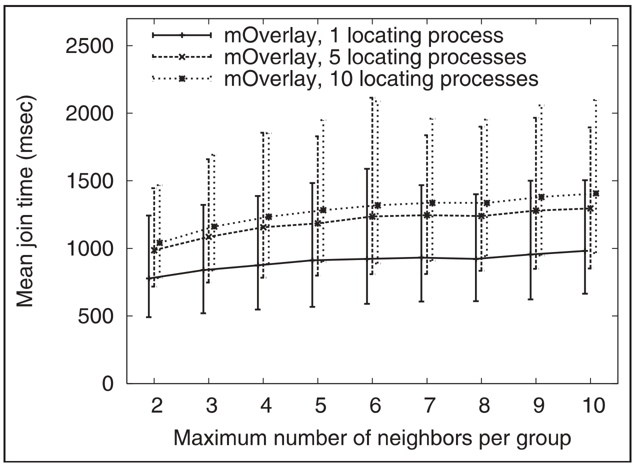

Figure 5 Mean joining delay in mOverlay.

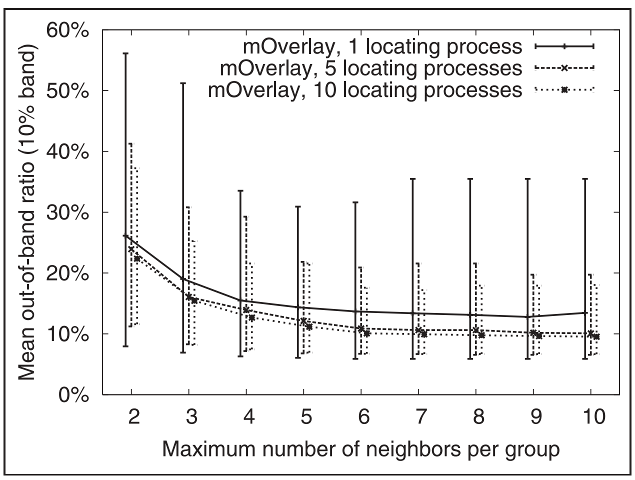

Figure 6 Mean out-of-band ratio in mOverlay.

n nodes and d directed edges leaving each node. Such a topology is optimal if the mean distance (i. e., round-trip time in our case) between each possible pair of nodes is minimal.

We can simplify the problem by considering each identified group

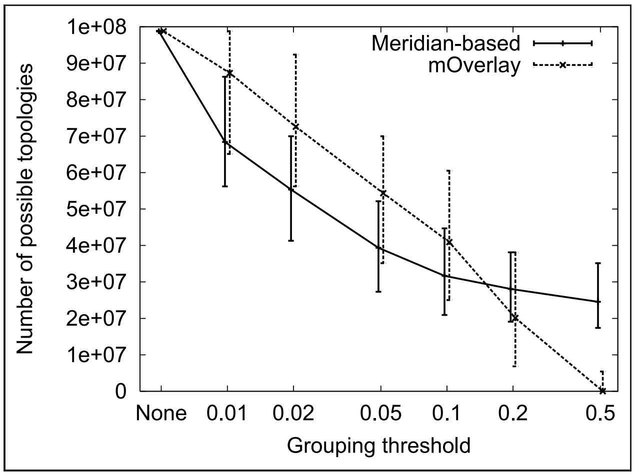

Figure 7 The effect of grouping on the complexity of finding optimal topologies.

a single node in the topology. The topology inside the groups is largely irrelevant for the optimization, since the round-trip times between members of the same group are generally very small. Thus, we can reduce the problem to only optimizing the part of the topology where significant round-trip times can be observed.

An upper bound of this optimization problem's complexity is the number of possible topologies $n\binom{n-1}{d}$, given a number n of nodes and a fixed number d of outgoing edges of a node. Fig. 7 shows the number of possible topologies an optimization algorithm must consider in our scenario of 77 end systems, for the mOverlay algorithm and our Meridian-based algorithm. The leftmost value shows the number of possibilities without any grouping. The other values are derived from the average group sizes given varying thresholds, including their confidence intervals (see Fig. 2). The formula used is $\frac{n}{a}\binom{n/a-1}{d}$, where a is the average group size.

We can see that the complexity of the optimization problem is significantly reduced even for small grouping thresholds. With the Meridian-based approach, a threshold of already 0.05 reduces the number of topologies that need to be considered by more than 50%, while the mean intra-group distance and the mean out-of-band ratio remain at acceptable levels (see Fig. 1 and 3). In contrast, the optimization achieved using the mOverlay method is significantly worse unless we choose a grouping threshold of 0.2 or above, in which case the group identification is quite bad.

6 Conclusions

The locating algorithm of mOverlay has a number of drawbacks, such as suboptimal results and long running times. We have presented an alternate algorithm that combines Meridian's closest node search with mOverlay's grouping criterion.

From the simulation results we conclude that the Meridian-based locating algorithm is faster in most cases. It also identifies larger groups (within a certain parameter range), and the nodes inside the groups are closer together than the nodes in mOverlay groups. Moreover, the groups identified with the alternate algorithm also have a smaller out-of-band ratio, which indicates better suitability for a distance estimation service.

In addition to its original use in a distance measurement service, the proposed algorithm can also help to reduce the complexity of optimizing overlay topologies with respect to transmission latency. We have illustrated this by demonstrating the effect based on the simulation results.

References

[1] M. Scheidegger and T. Braun, “Improved Locality-Aware Grouping in Overlay Networks”, KiVS 2007, Bern, Switzerland, Feb/Mar 2007.

[2] X. Y. Zhang et al., “A construction of locality-aware overlay network: mOverlay and its performance”, *IEEE JSAC*, vol. 22, no. 1, pp. 18–28, Jan 2004.

[3] B. Wong, A. Slivkins, and E. G. Sirer, “Meridian: A lightweight network location service without virtual coordinates”, ACM SIGCOMM’05, Philadelphia, Pennsylvania, USA, Aug 2005.

[4] Meridian C++ code, http://www.cs.cornell.edu/People/egs/meridian/code.php.

[5] M. Scheidegger, T. Braun, and F. Baumgartner, “Endpoint cluster identification for end-to-end distance estimation”, ICC’06, Istanbul, Turkey, June 2006, ISBN 1-4244-0355-3.

[6] J. Stribling, “All-pairs-pings for PlanetLab”, http://pdos.csail.mit.edu/~strib/pl_app.

[7] PlanetLab home page, http://www.planet-lab.org.

[8] L. Peterson et al., “Using PlanetLab for Network Research: Myths, Realities, and Best Practices”, WORLDS’05, Vol. 2, San Francisco, California, USA, Dec 2005.

[9] S. Ratnasamy, M. Handley, R. Karp and S. Shenker, “Topologically aware overlay construction and server selection”, IEEE Infocom’02, New York, NY, June 2002.

[10] L. Garcés-Erice, E. W. Biersack, and P. A. Felber, “MULTI+: Building topology-aware overlay multicast trees”, in QofIS’04, Sep 2004.

[11] P. Francis, S. Jamin, J. Cheng, Y. Jin, D. Raz, and Y. Shavitt, “IDMaps: A global Internet host distance estimation service”, IEEE/ACM ToN, Vol. 9, No. 5, pp. 525–540, Oct 2001.

[12] W. Theilmann and K. Rothermel, “Dynamic distance maps of the Internet”, IEEE Infocom 2000.

[13] T. S. E. Ng and H. Zhang, “Predicting Internet network distance with coordinates-based approaches”, ACM IMC 2003.

[14] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, “Lighthouses

for scalable distributed location”, IPTPS 2003.

[15] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: A decentralized network coordinate system”, ACM SIGCOMM 2004.

[16] S. Srinivasan and E. Zegura, “M-coop: A scalable infrastructure for network measurement,” Third IEEE Workshop on Internet Applications (WIAPP‘03), June 2003.

[17] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, “Topologically-aware overlay construction and server selection”, Infocom 02, 2002.


1


2

1 **Matthias Scheidegger** got his master and Ph.D. degrees in computer science from the University of Bern, Switzerland, in 2001 and 2007, respectively. Until recently, he has been a member of the Computer Networks and Distributed Systems Research Group of the University of Bern. He is now a software engineer at Google Labs Zürich.
Address: Google Switzerland GmbH, Freigutstraße 12, 8002 Zürich, Switzerland,
E-Mail: mbs@google.com and mscheid@iam.unibe.ch

2 **Torsten Braun** got his diploma and Ph.D. degrees from the University of Karlsruhe, Germany, in 1990 and 1993, respectively. From 1994 to 1995 he was a guest scientist with INRIA Sophia Antipolis. From 1995 to 1997 he worked as a project leader and senior consultant at the IBM European Networking Center, Heidelberg, Germany. Since 1998 he has been a full professor of computer science at the Institute of Computer Science and Applied Mathematics (University of Bern, Switzerland), heading the Computer Networks and Distributed Systems Research Group. He has been a board member of SWITCH (Swiss Education and Research Network) since 2000. During his sabbatical in 2004, he has been working as visiting scientist at the Swedish Institute of Computer Science (Kista, Sweden) on transport, routing and medium access control issues in wireless sensor networks. He received best paper awards from the 26th IEEE Conference on Local Computer Networks in 2001 and the 5th International Conference on Wired/wireless Internet Communications in 2007.
Address: IAM, Universität Bern, Neubrückstraße 10, 3012 Bern, Switzerland,
E-Mail: braun@iam.unibe.ch