Modelling the Internet Delay Space Based on Geographical Locations

Sebastian Kaune*, Konstantin Pussep*, Christof Leng*, Aleksandra Kovacevic*,
Gareth Tyson†, and Ralf Steinmetz*
* *Technische Universität Darmstadt, Germany*
† *Lancaster University, United Kingdom*

Abstract

Existing approaches for modelling the Internet delay space predict end-to-end delays between two arbitrary hosts as static values. Further, they do not capture the characteristics caused by geographical constraints. Peer-to-peer (P2P) systems are, however, often very sensitive to the underlying delay characteristics of the Internet, since these characteristics directly influence system performance.

This work proposes a model to predict lifelike delays between a given pair of end hosts. In addition to its low delay computation time, it has only linear memory costs which allows large scale P2P simulations to be performed. The model includes realistic delay jitter, subject to the geographical position of the sender and the receiver. Our analysis, using existing Internet measurement studies reveals that our approach seems to be an optimal tradeoff between a number of conflicting properties of existing approaches.

1. Introduction

Peer-to-peer (P2P) systems have gained significant research attention in recent years. Due to their large-scale, simulation is often the most appropriate evaluative method. Internet properties, and especially their delay characteristics, often directly influence P2P system performance. In delay-optimized overlays, for instance, proximity neighbor selection (PNS) algorithms select the closest node in the underlying network from among those that are considered equivalent by the routing table. The definition of closeness is typically round-trip time (RTT). In addition, having realistic delay cluster properties is equally important when analyzing caching strategies and server placement polices. Therefore, in order to obtain accurate results, simulations must include an adequate model of the Internet delay space.

The main challenges in creating such a model can be summarized as follows: (*i*) The model must be able to predict lifelike delays between a given pair of end-hosts. (*ii*) The model must include realistic delay jitter. Normally, the propagation of messages is affected by the processing and queuing delays of intermediate routers, something which varies over time. Therefore, end-to-end delays vary between two arbitrary nodes and are therefore non-static.

We argue that both requirements are subject to the geographical position of the sender and the receiver. First, the minimal delay is limited by the propagation speed of signals in the involved links which increases proportionally with the link length. Second, the Internet infrastructure is very different in different countries. As long-term measurement studies (cf. Sec. 3.1) reveal, jitter and packet loss rates are heavily influenced by the location of participating nodes. For example, the routers in a developing country are more likely to suffer from overload than those in a more economically advanced country. Such observations are typical for real-world measurement data but are absent in artificial delay models that ignore geographical locations.

The main contribution of this paper is the provision of a predictive model of the Internet delay space that fulfils the above stated requirements. To do so, we use rich data from two measurement projects as input. The resulting delays are non-static, and realistically reflect the delay characteristics caused by the geographical constraints. Additionally, we compare our model against the existing approaches of obtaining end-to-end delays. To this end, we show that our calculated distance of non-measured links is also a suitable representation of delays occurring in the Internet. The complexity of our model is $O(n)$ which is acceptable for the inherent memory constraints of large scale simulations.

In Section 2 we give an overview of the state-of-the art Internet delay models. Section 3 describes the data sets we use and the assumptions we make. Finally, our model is presented in Section 4 and evaluated in Section 5. Concluding remarks are given in Section 6.

2. Related Work

Currently, there are three different approaches to obtaining a delay model for P2P-related simulations. The first approach uses the King tool [1] to compute the all-pair end-to-end delays among a large number of globally distributed DNS servers. In more detail, each server is typically located in a distinct domain, and the measured delays therefore represent the static Internet delay space among the edge networks. Due to the quadratic time requirement for collecting this data, the size of the resulting delay matrix is often limited. For example, [1] provides a delay matrix with 1740 rows/columns. This is a non-trivial amount of measurement data to obtain. Delay synthesizers use this statistical data as an input to produce Internet delay spaces at a large scale [2].

The second approach is based on using artificial link delays assigned by topology generators such as Inet [3] or GT-ITM [4]. Within this approach, a topology file is initially generated for a pre-defined number of nodes. The end-to-end delay is then computed on the fly by determining the topology's all-pair shortest-path; a process which requires high computational power. Alternatively, pre-computation of an all-pair delay matrix squares the memory overhead to $O(n^2)$.

The third approach is to start with the data of Internet measurement projects, e.g. Surveyor [5], CAIDA [6], and AMP [7]. These projects typically perform active probing to up to a million destination hosts, derived from a small number of globally distributed monitor hosts. Prior work uses this data as an input to generate realistic delay by embedding hosts into a low dimensional Euclidean space [8].

This work follows the third approach of obtaining end-to-end delays. However, our approach differs to recent work in the following two major points. First, none of the aforementioned approaches considers realistic delay jitter. That is, recent approaches aim to predict static delays, either the average or minimum delay between two hosts. Second, most of prior work does not accurately reflect delay characteristics caused by the different geographical regions of the world. This issue can, however, highly influence the performance of P2P systems, as we will see later on.

3. Data Collection and Assumptions

This section provides background information on the measured Internet delay data we use in this work. Finally, the assumptions on which our delay model is based on are given.

3.1. Data from two Internet measurement projects

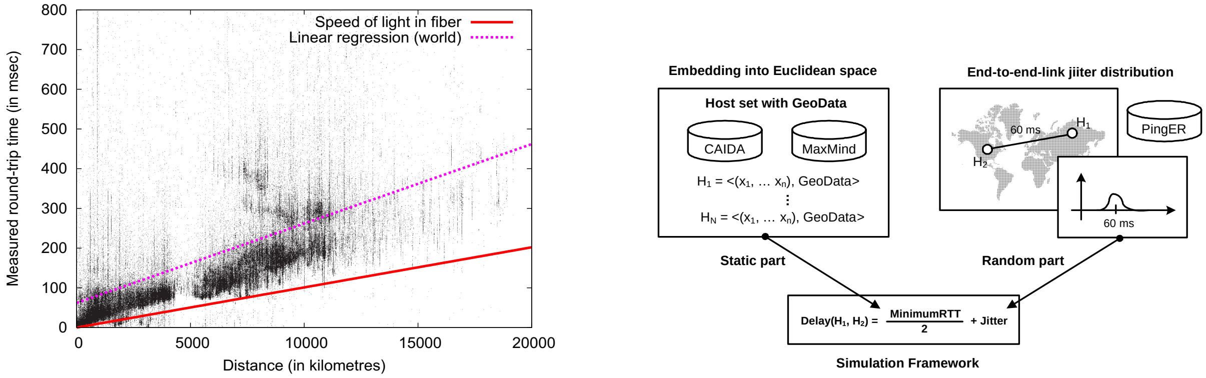
We use two kinds of data for this work. Firstly, we utilise the measurement data of the CAIDA's macroscopic topology probing project from August 2007 [6]. This data contains a large volume of RTT measurements taken between 20 globally distributed monitor hosts and nearly 400,000 destination hosts. Within this project, each monitor actively probes every host stored in the so-called destination list by sending ICMP [9] echo-requests. This lists account for 313,471 hosts covering the routable IPv4 space, alongside 58,312 DNS clients. Each monitor-to-destination link is measured 5-10 times a month, resulting in an overall amount of 40 GB of measurement data. Fig. 1(a) plots this data in relation to the geographical distance between each monitor host and its destinations. Both, the geographical locations of the monitors and the destination hosts are determined by MaxMind GeoIP service[1] [10]. It can be observed that there is a proportionality of the RTT to the length of the transmission medium. Also, the figure clearly shows the impact of the queuing/processing delays caused by intermediate routers. The 'islands' at 8000 - 12000 km and 300 - 400 ms RTT arises from countries in Africa and South Asia.

To study the changes of delay over time, we additionally incorporate the data of the *PingER* project [11]. This project currently has more than 40 monitoring sites in 20 countries and about 670 destination sites in 150 countries in most regions of the world. Compared to the CAIDA project, the number of monitor sites is significantly higher and globally more distributed whereas the amount of remote sites is by order of magnitudes smaller. Nevertheless, the RTT for one monitor-to-destination link is measured up to 960 times a day, in contrast to 5-10 times per month by the CAIDA project. As seen later on, this allows us to accurately predict the inter-packet delay variation between any two hosts located in different regions, countries or continents.

3.2. Assumptions

As already stated before, the CAIDA project measures RTTs containing the end-to-end delay of

[1] The obviously impossible RTT values below the propagation time of the speed of light in fiber can be explained by a false positioning through MaxMind.



(a)

(b)

Figure 1. The measured round-trip times in relation to the geographical distance in August 2007 (left). Overview about our delay space modeling techniques (right).

monitor-to-destination links, but also the transmission and processing delay of the destination host when receiving the echo-request and sending the echo-reply. We assume for the remainder of this paper, that the end-to-end delay can be calculated by dividing the measured RTT by two whilst neglecting the latter two delays. This is a reasonable assumption since the size of the echo packet is only 52 bytes. Also, the processing delay on the destination host is usually in the range of nanoseconds, and is therefore negligible.

4. Model

This section details our model that aims to realistically predict end-to-end delays between two arbitrary hosts chosen from a predefined host set. These delays are non-static, and consider the geographical location of both the source and destination host. Further, the model properties in terms of computation and memory overhead are given.

4.1. Overview

We split up the modelling of delay into a two-part architecture. The first part computes the *minimum delay* between any two hosts based on the measured round-trip time samples of CAIDA, and is therefore static. The second part, on the other hand, is variable and determines the inter-packet delay variation of this minimum delay, also known as *jitter*.

Thus, the end-to-end delay between two hosts $\mathcal{H}_1$ and $\mathcal{H}_2$ is then given by

$$delay(\mathcal{H}_1, \mathcal{H}_2) = \frac{minimumRTT}{2} + jitter \quad (1)$$

Fig. 1(b) gives an overview of our two part delay space modelling techniques. The first part (top left) generates a set of hosts from which the simulation framework can choose a subset from. More precisely, this set is composed of the destination list of the CAIDA measurement project. Using the MaxMind GeoIP database, we are able to look up the IP addresses of these hosts and find out their geographic position, i.e., continent, country, region, and ISP. In order to calculate the minimum delay between any two hosts, the Internet is modelled as a multidimensional Euclidean space $\mathcal{S}$. Each host is then characterized by a point in this space so that the minimum round-trip time between any two nodes can be predicted by a well-defined distance function.

The second part (top right), on the other hand, determines the inter-packet delay variation of this minimum delay; it uses the rich data of the PingER project to reproduce end-to-end link jitter distributions. These distributions can then be used to calculate random jitter values at simulation runtime.

Basically, both parts of our architecture require an offline computation phase to prepare the data needed for the simulation framework. Our overall goal is then to have a very compact and scalable presentation of the underlay at simulation runtime without introducing a significant computational overhead. In the following, we describe each part of the architecture in detail.

4.2. Part I: Embedding CAIDA hosts into the Euclidean Space

The main challenge of the first part is to place the set of destination hosts into a multidimensional Euclidean space, so that the computed minimum round-trip times approximate the measured distance as accurately as possible. To do so, we follow the approach of [12] and apply the technique of global network positioning in a slightly different way. However, this results in an optimization problem of minimizing the sum of the error between the measured and the calculated distances.

In the following, we denote the coordinate of a host $\mathcal{H}$ in a $\mathcal{D}$-dimensional coordinate space as $c^{\mathcal{S}}_{\mathcal{H}} = (c^{\mathcal{S}}_{\mathcal{H},1}, ..., c^{\mathcal{S}}_{\mathcal{H},D})$. The measured round-trip time between the hosts $\mathcal{H}_1$ and $\mathcal{H}_2$ is given by $d_{\mathcal{H}_1\mathcal{H}_2}$ whilst the computed distance $\hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}$ is defined by a distance function that operates on those coordinates:

$$\hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2} = \sqrt{(c^{\mathcal{S}}_{\mathcal{H}_1,1} - c^{\mathcal{S}}_{\mathcal{H}_2,1})^2 + ... + (c^{\mathcal{S}}_{\mathcal{H}_1,D} - c^{\mathcal{S}}_{\mathcal{H}_2,D})^2} \tag{2}$$

As needed for the minimization problems described below, we introduce a weighted error function $\varepsilon(\cdot)$ to measure the quality of each performed embedding:

$$\varepsilon(d_{\mathcal{H}_1\mathcal{H}_2}, \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}) = \left(\frac{d_{\mathcal{H}_1\mathcal{H}_2} - \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}}{d_{\mathcal{H}_1\mathcal{H}_2}}\right)^2 \tag{3}$$

At first, we calculate the coordinates $c^{\mathcal{S}}_{\mathcal{L}_1}, ..., c^{\mathcal{S}}_{\mathcal{L}_N}$ of a small sample of N hosts, also known as *landmarks* $\mathcal{L}_1$ to $\mathcal{L}_N$. These coordinates then serve as reference points with which the position of any destination host can be oriented in $\mathcal{S}$. A precondition for the selected landmarks is the existence of measured round-trip times to each other. In our approach, these landmarks are chosen from the set of measurement monitors from the CAIDA project, since these monitors fulfil this precondition. To this end, we seek to minimize the following objective function f_{obj1}:

$$f_{obj1}(c^{\mathcal{S}}_{\mathcal{L}_1}, ..., c^{\mathcal{S}}_{\mathcal{L}_N}) = \sum_{\mathcal{L}_i, \mathcal{L}_j \in \{\mathcal{L}_1, ..., \mathcal{L}_N\} | i > j} \varepsilon(d_{\mathcal{L}_i\mathcal{L}_j}, \hat{d}^{\mathcal{S}}_{\mathcal{L}_i\mathcal{L}_j}) \tag{4}$$

There are many approaches with different computational costs that can be applied [12][13]. Previous work has shown that a five dimensional Euclidean embedding approximates the Internet delay space very well [14]. Therefore, we select N(=6) monitors out of all available monitors using the maximum separation method [8]. This method determines the subset of N monitors out of all available monitors which produces the maximum sum for all inter-monitor round-trip times[2]. Due to the fact that the overall number of monitors within the CAIDA project is 20, these monitors can easily be obtained through iteration across all combinations.

In the second step, each destination host is individually (one after the other) embedded into the Euclidean space. To do this, round-trip time measurements to all N monitor hosts must be available. Similarly to the previous step, we take the minimum value across the monitor-to-host RTT samples. While positioning the destination hosts coordinate into $\mathcal{S}$, we aim to minimize the overall error between the predicted and measured monitor-to-host RTT by solving the following minimization problem f_{obj2}:

$$f_{obj2}(c^{\mathcal{S}}_{\mathcal{H}}) = \sum_{\mathcal{L}_i \in \{\mathcal{L}_1, ..., \mathcal{L}_N\}} \varepsilon(d_{\mathcal{H}\mathcal{L}_i}, \hat{d}^{\mathcal{S}}_{\mathcal{H}\mathcal{L}_i}) \tag{5}$$

An exact solution of this non-linear optimization problem is very complex and computationally intensive; we therefore apply an approximate solution that can be found by applying the generic *downhill simplex algorithm* of Nelder and Mead [15]. Finally, Fig. 2(b) depicts all embedded monitor hosts (red points) and destination hosts, and their determined geographical location on earth.

Once all host coordinates are computed, we use the directional relative error r to quantify the quality of the overall embedding. This metric describes the relative deviation of the calculated value to the measured distance, and is defined as

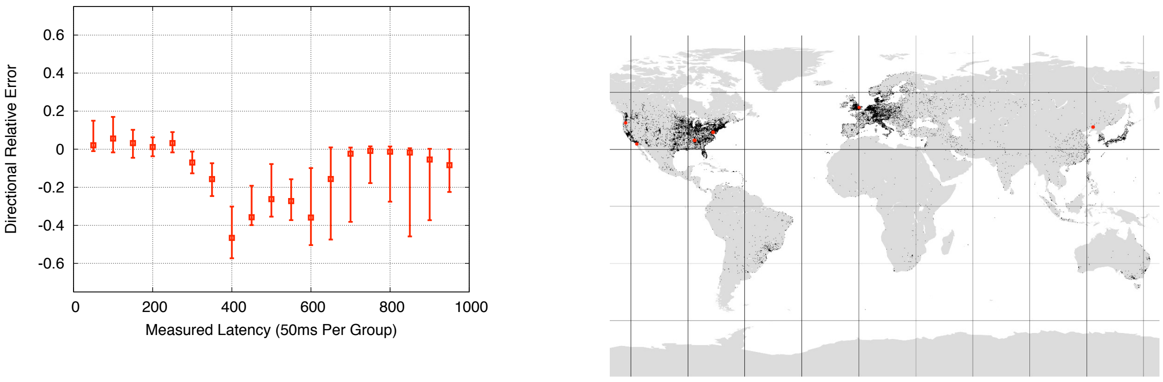
$$r(d_{\mathcal{H}_1\mathcal{H}_2}, \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}) = \frac{d_{\mathcal{H}_1\mathcal{H}_2} - \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2}}{\min(d_{\mathcal{H}_1\mathcal{H}_2}, \hat{d}^{\mathcal{S}}_{\mathcal{H}_1\mathcal{H}_2})} \tag{6}$$

A directional relative error of 0 means that the measured and calculated distances are equal; a value of 1 indicates that the calculated distance is the half of the measured one whilst an error of -1 means that the calculated distance is double of the measured one.

Fig. 2(a) depicts this error after the positioning of all the hosts. For this illustration, we consider only the measured minimum round-trip times used for our embedding as stated above. We then classify them into distinct intervals, each of 50 ms. Afterwards, we calculate the directional relative error for each interval. The squares depict the interval's median error, whereas the vertical lines indicate the 25th percentile and the 75th percentile. It can be observed that the embedding effectively predicts round-trip times in intervals between (0ms, 300ms]. The quality of the embedding decreases, however, for round-trip times in ranges of $300\text{ms}, 600\text{ms}$. That is, our embedding tends to over predict the measured data in these intervals. Nevertheless, we note that the round-trip times in these intervals only account for 5.8% of the measured data.

In this regard, we have also experimented with embeddings in higher dimensional Euclidean spaces reaching up to 11 dimensions. However, we observe only a very negligible improvement of the quality of the modelled delay space; the results for 6, 9 and 12 selected monitor hosts, creating a 5, 8, and 11 dimensional embedding respectively, were nearly identical.

[2] Note that we consider only the minimum value across the samples of inter-monitor RTT measurements



(a)

(b)

Figure 2. The directional relative error of global network positioning with 6 monitors (left). The distribution of embedded peers on earth (right).

4.3. Part II: Calculation of Jitter

Since the jitter constitutes the variable part of the delay, a distribution function is needed that covers its lifelike characteristics. Inspection of the measurement data from the PingER project shows that this deviation clearly depends on the geographical region of both end-hosts. Table 1 depicts an excerpt of the two way-jitter variations of end-to-end links between hosts located in different places in the world. These variations can be monthly accessed on a regional-, country-, and continental level. We note that these values specify the *interquartile range* (iqr) of the jitter for each end-to-end link constellation. This range is defined by the difference between the upper (or third) quartile Q_3 and the lower (or first) quartile $Q1$ of all measured samples within one month. The remarkably high iqr-values between Africa and the rest of the world are explained by the insufficient stage of development of the public infrastructure.

Table 1. End-to-end link inter-packet delay variation (iqr) in msec (January 2008) [11].

	Europe	Africa	N. America	Asia
Europe	1.53	137.14	1.29	1.19
Africa	26.91	78.17	31.79	1.11
S. America	14.17	69.66	10.78	14.16
N. America	2.02	73.95	0.96	1.33
Oceania	4.91	86.28	1.31	2.03
Balkans	1.83	158.89	1.43	1.25
E. Asia	1.84	114.55	1.38	0.87
Russia	2.29	161.34	2.53	1.59
S. Asia	7.96	99.36	16.48	7.46
S.E. Asia	0.86	83.34	13.36	1.27
Middle East	9.04	120.23	10.87	10.20

To obtain random jitter values based on the geographical position of hosts, for each end-to-end link constellation we generate a log-normal distribution with the following probability distribution function:

$$\mathrm{f}(x;\mu,\sigma) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{1}{2}\left(\frac{\ln x - \mu}{\sigma}\right)^2\right) & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The main challenge is then to identify the parameters μ (mean) and σ (standard deviation) by incorporating the measurement data mentioned above. Unfortunately, both values cannot be obtained directly. That is, we are in fact able to determine the expectation value of each constellation, which is given by the difference between the average RTT and the minimum RTT. Both values are also measured by the PingER project, and are available in the monthly summary reports, too. The variance or standard deviation is, however, missing.

For this reason, we formulate an optimization problem that seeks to find a parameter configuration for μ and σ having two different goals in mind. First, the chosen configuration should minimize the error between the measured inter quartile range $\mathrm{iqr_m}$ and $\mathrm{iqr(X)}$ which is generated by the log-normal distribution. Second, it should also minimize the error between the measured and generated expectation, $\mathrm{E_m}$ and $\mathrm{E(X)}$ respectively. Formally, this optimization problem is given by

$$f_{error} = \left(\frac{\mathrm{E}(X) - \mathrm{E_m}}{\mathrm{E_m}}\right)^2 + \left(\frac{\mathrm{iqr}(X) - \mathrm{iqr_m}}{\mathrm{iqr_m}}\right)^2 \tag{8}$$

where $E(X) = e^{\mu+\sigma^2/2}$ and $iqr(X) = Q_3 - Q_1$ as described above. To solve this, we apply the downhill simplex algorithm [15]. Observation of measurement data shows that the iqr-values are usually in the range of 0 to 20 milliseconds[3]. With respect to this, the three initial solutions are set to $(\mu = 0.1, \sigma = 0.1)$, $(\mu = 0.1, \sigma = 5)$, and $(\mu = 5, \sigma = 0.1)$, because these parameters generate random jitter values fitting this range exactly. The minimization procedure iterates then only 100 times to obtain accurate results.

We note that the obtained values for μ and σ describe the distribution of the *two-way jitter* for a specific end-to-end link constellation. In other words, it specifies the deviation of the RTT for all links falling into this geographical category. The one-way jitter is then obtained by dividing the randomly generated values by two. This assumption implicitly follows the basic principle of dividing the measured minimum RTT by two, in order to obtain the minimum delay between two hosts.

4.4. Algorithm and Memory Overhead

Here, we briefly describe the properties of our model in terms of computational costs and storage overhead. These are of major importance since these properties significantly influence the applicability of the model in large scale simulations.

First of all, the embedding of hosts into a $\mathcal{D}$-dimensional Euclidean space has a scalable representation of $O(n)$ while it adequately preserves the properties of the data measured by the CAIDA project. Since the process involved in obtaining this representation is complex and computationally expensive, it is typically done once. Thus, the resulting data can be reused for each simulation run, e.g., in terms of an XML file. In order to obtain the minimum delay between any two hosts in this embedding, the evaluation of the distance function takes then $O(\mathcal{D})$ time which is negligible.

The calculation of the jitter parameters of μ and σ for each possible end-to-end link constellation is also done once, either before the simulation starts or offline. Thus, similar to the pre-computation of the host coordinates, this process does not introduce any computational overhead into the actual simulation process. Nevertheless, the storage of the both parameters μ and σ takes at first sight a quadratic overhead of $O(n^2)$. Due to the fact that the amount of regions, countries and continents is limited, the required amount of memory is, however, negligible. For example, the processing of the data provided in the PingEr summary report of January 2008 result in 1525 distinct link constellations. For each of them, the two parameters μ and σ must be precomputed and stored resulting in a overall storage overhead of $(1525 \times 2) \times 4\,\text{bytes} \approx 12\text{kB}$.

5. Evaluation

This section describes the design of our experiments, and any metrics we think significantly influence the performance of P2P systems. We perform a comparative study against three existing approaches for obtaining end-to-end delays: (i) the King method, (ii) topology generators and (iii) analytical function. Our aim is to show that our model realistically reflects the properties of the Internet delay space. To this end, we show that the calculated delay between non-measured end-to-end links is also a suitable presumption compared to the delays that occur in the Internet.

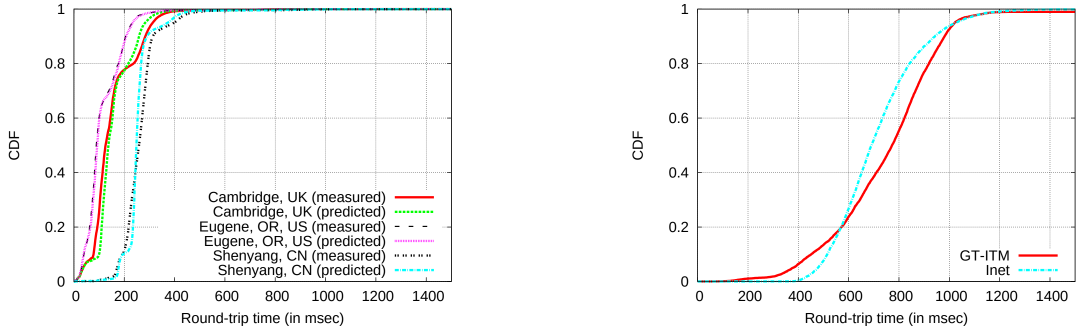
5.1. Experimental Design

We begin by providing background information about the approaches we use in our study later on. The first approach, namely the King method, serves as a reference point in our analysis because it provides measured Internet delay data among a large number of globally distributed DNS servers. We use the measurement data of [2] collected in October 2005. This matrix contains 3997 rows/columns representing the all-pair delays between IP hosts located in North America, Europe and Asia.

The second approach represents the category of topology generators. We are especially interested in the GT-ITM and Inet generators because they are often used in P2P simulations. For GT-ITM, we create a 9090 node transit-stub topology. For Inet, we create a topology for a network size of 10000 nodes. We use the default settings of placing nodes on a 10000 by 10000 plane with 30% of total nodes as degree-one nodes.

Finally, the simplest approach to determining end-to-end delays is applying an analytical function that uses as an input the distance between any two hosts. As seen in Section 3.1, there is a correlation between the measured RTTs and the geographical distance of peers. To obtain a function that reflects this correlation, we perform a *least squares analysis* so that the sum of the squared differences between the calculated and the measured RTT is minimized. Applying linear regression with this least squares method on the measurement data of 40 GB is, however, hardly possible. Therefore, we classify this data into intervals of 200 km each (e.g. $(0\text{km}, 200\text{km}], (200\text{km}, 400\text{km}]$ ...), and calculate the median round-trip time of each interval. Finally, linear regression gives us the following estimation for the RTT in milliseconds:

[3] Africa constitutes a special case. For this, we use another initial configuration as input for the downhill simplex algorithm.



(a)

(b)

Figure 3. The measured and predicted round-trip time (RTT) distribution as seen from different locations in the world (left). The RTT distribution as seen from a typical node generated by topology generators (right).

$$f_{world}(d_{a,b}) = 62 + 0.02 * d_{a,b} \qquad (9)$$

whereas $d_{a,b}$ is the distance between two hosts in kilometres. The delay is then given by $f(d_{a,b})$ divided by two. Fig. 4(a) illustrates this function and the calculated median RTT times of each interval.

5.2. Metrics

To benchmark the different approaches on their ability to realistically reflect Internet delay characteristics, we apply a set of metrics that are known to significantly influence the performance of P2P systems [2]:

- *Cutoff delay clustering* – In the area of P2P content distribution networks, topologically aware clustering is a very important issue. Nodes are often grouped into clusters based on their delay characteristics, in order to provide higher bandwidth and to speed up access [16]. The underlying delay model must therefore accurately reflect the Internet's clustering properties. Otherwise, analysis of system performance might lead to wrong conclusions.

To quantify this, we use a clustering algorithm which iteratively merges two distinct clusters into a larger one until a cutoff delay value is reached. In more detail, at first each host is treated as a singleton cluster. The algorithm then determines the two closest clusters to merge. The notion of closeness between two clusters is defined as the average delay between all nodes contained in both cluster. The merging process stops if the delay of the two closest clusters exceeds the predefined cutoff value. Afterwards, we calculate the fraction of hosts contained in the largest cluster compared to the entire host set under study.

- *Spatial growth metric* – In many application areas of P2P systems, such as in mobile P2P overlays, the cost of accessing a data object grows as the number of hops to the object increases. Therefore, it is often advantageous to locate the 'closest' copy of a data object to lower operating costs and reduce response times. Efficient distributed nearest neighbor selection algorithms have been proposed to tackle this issue for growth-restricted metric spaces [17]. In this metric space, the number of nodes contained in the radius of delay r around node p, increases at most by a constant factor c when doubling this delay radius. Formally, let $B_p(r)$ denote the number of nodes contained in a delay radius r, then $B_p(r) \leq c \cdot B_p(2r)$. The function $B_p(r)/B_p(2r)$ can therefore be used to determine the spatial growth c of a delay space.

- *Proximity metric* – In structured P2P overlays which apply proximity neighbor selection (PNS), overlay neighbors are selected by locating nearby underlay nodes [18]. Thus, these systems are very sensitive to the underlying network topology, and especially to its delay characteristics. An insufficient model of the Internet delay space would result in routing table entries that do not occur in reality. This would in turn directly influence the routing performance and conclusions might then be misleading. To reflect the neighborhood from the point of view of each host, we use the $D(k)$-metric. This metric is defined by $D(k) = \frac{1}{|N|}\sum_{p \in N} d(p,k)$, whereas $d(p,k)$ is the average delay from node p to its k-closest neighbors in the underlying network [19].

5.3. Analysis with measured CAIDA data

Before we compare our system against existing approaches, we briefly show that our delay model produces lifelike delays even though their calculation is divided into two distinct parts.

As an illustration of our results, Fig. 3(a) depicts the measured RTT distribution for the Internet as seen from CAIDA monitors in three different geographical locations, as well as the RTTs predicted by our model. We note that these distributions now contain all available samples to each distinct host, as opposed to the previous section where we only considered the minimum RTT.

First, we observe that our predicted RTT distribution accurately matches the measured distribution of each monitor host. Second, the RTT distribution varies substantially in different locations of the world. For example, the measured path latencies from China to end-hosts spread across the world have a median RTT more than double that of the median RTT measured in Europe, and even triple that of the median RTT measured in the US. Additionally, there is a noticeable commonality between all these monitors regarding to the fact that the curves rise sharply in a certain RTT interval, before they abruptly flatten out. The former fact indicates a very high latency distribution within these intervals, whereas the latter shows that a significant fraction of the real-world RTTs are in the order of 200 ms and above.

In contrast to this, Fig. 3(b) shows the RTT distribution as seen from a typical node of the network when using the topologies generated by Inet and GT-ITM as stated before. When comparing Fig. 3(a) and Fig. 3(b), it can be observed that the real-world RTT distributions significantly differ from the RTT distributions created by the topology generators. In particular, around 10-20% of the real-world latencies are more than double than their median RTT. This holds especially true for the monitor hosts located in Europe and in the US (see Fig. 3(a)). Topology generators do not reflect this characteristic. Additionally, our experiments showed that in the generated topologies, the RTT distribution seen by different nodes does not significantly vary, even though they are placed in different autonomous subsystems and/or router levels. Thus, topology generators do not accurately reflect the geographical position of peers, something which heavily influences the node's latency distribution for the Internet.

5.4. Comparison to Prior Work

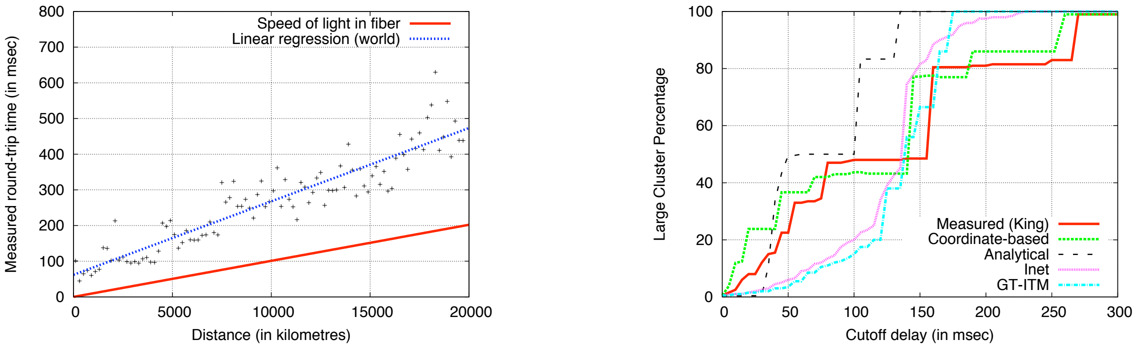
We compare our model against existing approaches for obtaining end-to-end delays using the metrics presented before. The reference point for each metric is the all-pair delay matrix received by the King method. We use this because the data is directly derived from the Internet. However, we are aware that this data only represents the delay space among the edge networks. To enable a fair comparison, we select, from our final host set, all hosts that are marked as DNS servers in CAIDA's destination list. We only utilize those that are located in Europe, Northern America or Asia. These nodes form the host pool for our coordinate-based model, and the analytical function, from which we chose random sub-samples later on. For the generated GT-ITM topology, we select only stub routers for our experiments to obtain the delays among the edge networks. For the Inet topology, we repeat this procedure for all degree-1 nodes. To this end, we scale the delays derived from both topologies such that their average delays matches the average delay of our reference model. While this process does not affect delay distribution's properties, it alleviates the direct comparison of results.

The results presented in the following are the averages over 10 random sub-samples of each host pool whereas the sample size for each run amounts to 3000 nodes[4].

We begin to analyse the cluster properties of the delay spaces produced by each individual approach. Fig. 4(b) illustrates our results after applying the clustering algorithm with varying cutoff values. It can be observed that for the reference model, our approach (coordinate-based), and the distance function, the curves rise sharply at three different cutoff values. This indicates the existence of three major clusters. By inspecting the geographical origin of the cluster members of the latter two models, we find that these clusters exactly constitute the following three regions: Europe, Asia and North America. Further, the three cutoff values of the analytical function are highly shifted to the left, compared to the values of the reference model. Nevertheless, the basic cluster properties are preserved. The curve of our delay model most accurately follows the one of the reference model, but it is still shifted by 10-20 ms to the left. Finally, both topology generated delays do not feature any clear clustering property. This confirms the findings that have already been observed in [2].

To analyse the growth properties of each delay space, we performed several experiments each time incrementing the radius r by one millisecond. Fig. 5(a) depicts our results. The x-axis illustrates the variation of the delay radius r whereas the y-axis shows the median of all obtained $B_p(2r) \,/\, B_p(r)$ samples for each specific value of r. Regarding the reference

[4] It is shown in [2] that the properties we are going to ascertain by our metrics are independent of the sample size. Thus, it does not matter if we set it to 500 or 3000 nodes.



(a)

(b)

Figure 4. Results of linear regression with least square analysis on CAIDA measurement data (left). Simulation results for cutoff delay clustering (right).

model, it can be seen that the curves oscillates two times having a peak at delay radius values 20 ms and 102 ms. Also, our coordinate-based approach and the analytical function produces these two characteristic peaks at 26 ms and 80 ms, and 31 ms and 76 ms respectively[5].

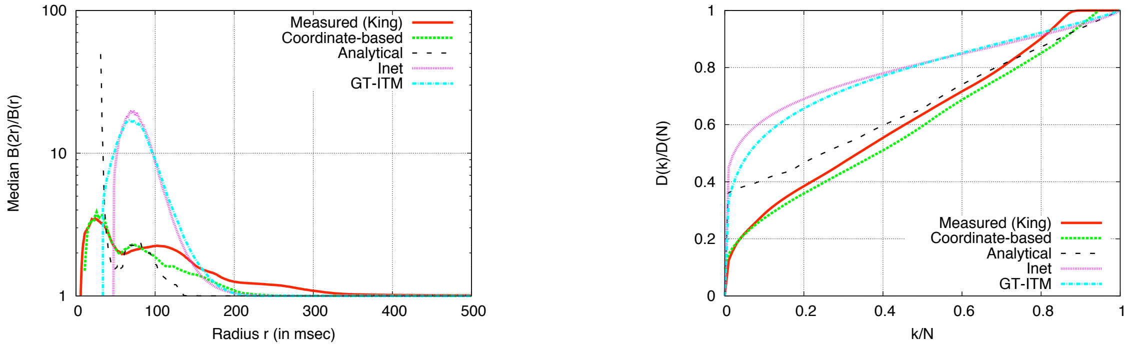
In all of the three mentioned delay spaces, the increase of the delay radius firstly covers most of the nodes located in each of the three major clusters. Afterwards, the spatial growth decreases as long as r is high enough to cover nodes located in another major cluster. Then, it increases again until all nodes are covered, and the curves flatten out. The derived growth constant for this first peak of the analytical function is, however, an order of magnitude higher than the constants of the others. This is clearly a consequence of our approximation through linear regression. Since this function only represents an average view on the global RTTs, it cannot predict lifelike delays with regard to the geographical location of peers. Nevertheless, this function performs better than both topology generated delay spaces. More precisely, none of both reflect the growth properties observed by our reference delay space.

The experiments with the $D(k)$-metric confirm the trend of our previous findings. The predicted delays of our coordinate-based model accurately matches the measured delays of the reference model. Fig. 5(b) illustrates the simulation results. While varying the number of k (x-axis), we plot the delay derived by the $D(k)$-function over the average to all-node delay. Whilst especially the measured delays and the one predicted by our model show the noticeable characteristic that there are a few nodes whose delay are significantly smaller than the overall average, the topology generated delays do not resemble this. As a consequence, it is likely that the application of PNS mechanisms in reality will lead to highly different results when compared to the ones forecasted with GT-ITM or Inet topologies. The analytical function, on the other hand, performs significantly better than the topology generators, even though there is also a noticeable difference in the results obtained by former two delay spaces.

6. Conclusion

Simulation is probably the most important tool for the validation and performance evaluation of p2p systems. The performance of overlay networks depends strongly on a realistic Internet model. Several different models for the simulation of link delays have been proposed in the past, which all have severe shortcomings. Most approaches do not incorporate the properties of the geographic region of the host. Hosts in a generated topology thus have overly uniform delay properties. The analytical approach, on the other hand, does not provide a jitter modell that reflects the different regions and the absolute delays differ from more realistic approaches. Only the King model yields results similar to our coordinate-based system. This proves the realism of our approach because King is directly derived from real-world measurements. The major drawback of King is its limited scalability. It requires memory proportional to n^2 and available datasets are limited to 3997 measured hosts. Statistical scaling of this data allows us to preserve delay properties, but produces solely static delay values [2].

[5] The minimum delay produced by the analytical function is 31ms, no matter the distance. This is why there are no values for the first 30 ms of r.



(a)

(b)

Figure 5. Simulation results for spatial growth of the modelled delay spaces (left), and the $D(k)$-function as proximity metric (right).

Our approach has only linear memory costs and provides a much larger dataset of several hundred thousand hosts. Compared to topology generators the delay computation time is low. In summary, coordinate-based delay models seem to be an optimal tradeoff between many conflicting properties. The key to their success lies in the incorporation of real-world measurement data comprising the heterogeneity of the world's regions. Our model is based on embedding the rich data from the CAIDA project into a multi-dimensional Euclidean space. Compared to prior work on coordinate-based Internet delay models [8], we add a realistic jitter model derived from the PingER monthly reports.

Acknowledgement

The collaboration on this paper has been funded through the European Network of Excellence CONTENT, FP6-0384239.

References

[1] K. P. Gummadi, S. Saroiu, S. D. Gribble, "Estimating latency between arbitrary internet end hosts," in *SIGCOMM IMW '02*, 2002.

[2] B. Zhang, T. S. Eugene Ng, A. Nandi, R. Riedi, P. Druschel, G. Wang, "Measurement-based analysis, modeling, and synthesis of the internet delay space," in *IMC '06*, 2006.

[3] J. Winick, S. Jamin, "Inet-3.0: Internet topology generator," University of Michigan, Tech. Rep., 2002.

[4] E.W. Zegura, K.L. Calvert, S. Bhattacharjee, "How to model an internetwork," in *INFOCOM '96*, 1996.

[5] Surveyor, http://www.advance.org/csg-ippm/.

[6] CAIDA. Macroscopic Topology Project, http://www.caida.org/analysis/topology/macroscopic/.

[7] Active measurement project, http://watt.nlanr.net.

[8] G. Kunzmann, R. Nagel, T. Hossfeld, A. Binzenhofer, K. Eger, "Efficient simulation of large-Scale p2p networks: modeling network transmission times," in *MSOP2P '07*, 2007.

[9] J. Postel, "Internet Control Message Protocol," RFC 792 (Standard), 1981, updated by RFCs 950, 4884.

[10] MaxMind Geolocation Technology, http://www.maxmind.com/.

[11] The PingER Project, http://www-iepm.slac.stanford.edu/pinger/.

[12] T. S. Eugene Ng, H. Zhang, "Towards global network positioning," in *SIGCOMM IMW '01*, 2001.

[13] L.Tang and M. Crovella, "Geometric exploration of the landmark selection problem," in *PAM '04*, 2004.

[14] S. Lee, Z. Zhang, S. Sahu, D. Saha, "On suitability of euclidean embedding of internet hosts," in *SIGMETRICS '06*, 2006.

[15] J. A. Nelder, R. Mead, "A simplex method for function minimization," *Computer Journal*, 1965.

[16] G. Tyson, A. Mauthe, "A topology aware clustering mechanism," in *PGNet '07*, 2007.

[17] D. R. Karger, M. Ruhl, "Finding nearest neighbors in growth restricted metrics," in *SoTC '02*, 2002.

[18] K. Gummadi et al., "The impact of dht routing geometry on resilience and proximity," in *SIGCOMM '03*, 2003.

[19] M. Castro, P. Druschel, Y. C. Hu, A. Rowstron, "Proximity neighbor selection in tree-based structured p2p overlays," Microsoft Research, Tech. Rep., 2003.