



Transport Information Collection Protocol with clustering of information sources

Mohamed Karim Sbai, Chadi Barakat

► To cite this version:

Mohamed Karim Sbai, Chadi Barakat. Transport Information Collection Protocol with clustering of information sources. NTMS 2007, May 2007, Paris, France. inria-00121586v1

HAL Id: inria-00121586

<https://inria.hal.science/inria-00121586v1>

Submitted on 21 Dec 2006 (v1), last revised 24 Jan 2007 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Transport Information Collection Protocol with clustering of information sources

Mohamed Karim SBAI

Projet Planete, INRIA Sophia Antipolis, France
National School of Computer Science (ENSI), Tunisia
Email: Mohamed_Karim.Sbai@sophia.inria.fr

Chadi BARAKAT

Projet Planete, INRIA Sophia Antipolis, France
Email: Chadi.Barakat@sophia.inria.fr

Abstract—We improve and validate TICIP, a TCP-friendly reliable transport protocol to collect information from a large number of sources spread over the Internet [1]. A collector machine sends probes to information sources, which respond by sending back report packets containing their information. TICIP adapts the rate of probes in a way to avoid implosion at the collector and congestion in the network. To ensure smooth variation of the congestion control parameters and to probe sources behind the same bottleneck at the same time, we add to TICIP a mechanism that clusters information sources. This mechanism is based upon the Global Network Positioning (GNP) Internet coordinate system. By running simulations in ns-2 over realistic network topologies, we prove that TICIP with clustering of information sources has shorter collect session duration and causes less packet losses than the initial version that probes sources independently of their location.

I. INTRODUCTION

Nowadays, collecting information from a large number of network entities has more and more applications. Among others, it allows to control transmission quality in multicast sessions, to gather measurements from sensors distributed in a network, to assure census, to collect results of votes, etc. Hence, the conception of efficient protocols supporting such applications is becoming a necessity.

This paper describes how we have improved the Transport Information Collection Protocol (TICIP) [1]. This transport protocol collects information entirely from a set of sources spread over the Internet: a collector machine sends probes to information sources, which in their turn send back report packets containing their information. However, some difficulties come into play:

- There is a risk of congestion in the network due to the limitation of the bandwidth and the large of sources. Furthermore, all sources do not lie behind a sole bottleneck which makes difficult the congestion control.
- The collection traffic can be aggressive towards traffic generated by other applications. In particular, it must not penalise concurrent TCP traffic.
- The loss of probes or reports lengthens the collect session duration, which urges for an efficient retransmission scheme.

For these reasons, TICIP does not only adapt the rate of sending probes as a function of network conditions, but also try to minimize the collect session duration. Moreover, It

shares fairly network resources with concurrent traffic namely TCP one by adapting its probing rate in a way similar to how TCP does (TCP friendliness).

The collector in the former version of TICIP [1] probes information sources in a random order. We show in this paper that this choice causes many problems when moving into large networks like increasing the collect session duration, having high loss rates and generating a traffic out of control. To ensure a smooth variation of the congestion control parameters and to probe sources behind the same bottleneck at the same time, we add to TICIP a mechanism to gather information sources into clusters. This mechanism is based on the modeling of the Internet by an euclidean space and its decomposition into clusters. We use for this the Global Network Positioning system (GNP) [4] which provides Internet host coordinates. The new mechanism makes it possible to traverse the sources from the closest to the farthest from the collector in terms of Round Trip Time. This way sources located behind the same bottleneck are probed together which improves the efficiency of the congestion control. To evaluate the performances of the protocol thus obtained, we have run simulations with the NS-2 simulator [6]. These simulations have shown that TICIP with the new mechanism of clustering information sources has better performances. Indeed, there is a reduction of the loss rate and a decrease in the collect session duration.

In the first section of this paper, we describe the main functionalities of TICIP. We show in the second section that the former version of TICIP has many problems and that the solution is to cluster information sources. In the third section, we explain our approach of clustering information sources. The last section discusses simulations results.

II. TCP-FRIENDLY INFORMATION COLLECTION PROTOCOL

TICIP [1] is a reliable transport information collection protocol implementing diverse functionalities. We focus here on those related to error recovery and network congestion control.

A. Error recovery

The TICIP collector has a list of all information sources. Every source is distinguished by an identifier. The sources whose reports are lost are probed again. To ensure this retransmission mechanism, the collect session is a succession of rounds. In the first round, the collector sends request packets to all sources following their ranking in the list. In a second round, the collector sends requests to sources whose reports were not received in the previous round. The collector continues in rounds until it receives all reports. This behavior in rounds is meant to wait for transitory network congestions to disappear from one round to another and to absorb the excessive delay that some reports may experience.

B. Congestion control

To control the rate of requests and reports across the network, TICIP is based on a report-clocked window based congestion control similar to the TCP one [2]. The collector maintains one variable $cwnd$ indicating the congestion window size in number of requests/reports. New requests are transmitted only when the number of expected reports $pipe$ is less than $cwnd$. TICIP adapts $cwnd$ to the observed loss rate of reports. It proposes two algorithms to do so: Slow start and Congestion Avoidance.

1) *Slow start*: The collector starts a collect session by setting $cwnd$ to RS (protocol parameter) and sending RS request packets. After some time, reports start to arrive. Some of these reports come in time, others are delayed. A timely report indicates that the network is not congested and that the collector can continue increasing its congestion window: $cwnd = cwnd + 1$. The window grows in this way until the network becomes congested. At this point, the collector divides its congestion window by two and enter in the congestion avoidance phase. The protocol comes back to slow start mode whenever a severe congestion appears.

2) *Congestion avoidance*: The congestion avoidance phase represents the steady state of TICIP. During this phase, the collector increases slowly $cwnd$ in order to probe the network for more capacity. Upon each timely report, the congestion window is increased by the following amount: $cwnd = cwnd + \frac{RS}{cwnd}$. When congestion is detected, $cwnd$ is divided by two and a new congestion avoidance phase is started.

C. Congestion detection mechanism

TICIP uses the congestion detection mechanism to compute report loss rates and to decide whether a report is in time, delayed or lost. This mechanism is based upon a timer TO scheduled at the beginning of the session and rescheduled again every time it expires.

1) *Round-trip time estimator*: TICIP sets the timer of the mechanism to an estimate of the round-trip time, using the samples of round-trip time seen so far. The value of the timer is computed using estimates of the average round-trip time and of its variance. Let $srtt$ and $rttvar$ be the estimates of the average and the mean deviation of the round-trip time. Let rtt be the measured round-trip time when a report arrives. The

collector updates the values of the estimates and the timer (TO) in the following way :

$$\begin{aligned} rttvar &= \frac{3}{4}.rttvar + \frac{1}{4}.|srtt - rtt| \\ srtt &= \frac{7}{8}.srtt + \frac{1}{8}.rtt \\ TO &= srtt + 4.rttvar \end{aligned}$$

2) *Detecting network congestion*: TICIP computes the report loss rate during a time window equal to TO . When the timer is scheduled, the collector saves in the variable $torecv$ the number of reports to be received before the expiration of the timer. Let $recv$ be the number of timely reports received between the scheduling of the timer and its expiration. The collector considers then that $torecv - recv$ reports were lost in the network. It estimates the loss rate to $1 - \frac{recv}{torecv}$.

The network is considered as congested if the loss rate exceeds the Congestion Threshold (CT) and severely congested if it exceeds a higher threshold $SCT > CT$ (Severe Congestion Threshold). CT and SCT are two parameters of the protocol. TICIP set them as follows:

$$\begin{aligned} CT &= \min(0.1, \frac{RS}{cwnd}) \\ SCT &= \max(0.9, \frac{cwnd - RS}{cwnd}) \end{aligned}$$

3) *Delayed and timely reports*: A timely report is a report received before its deadline. The deadline of a report is given by the timer. A report not received before its deadline is assumed to be lost. If it arrives later than the deadline, it is considered to be delayed.

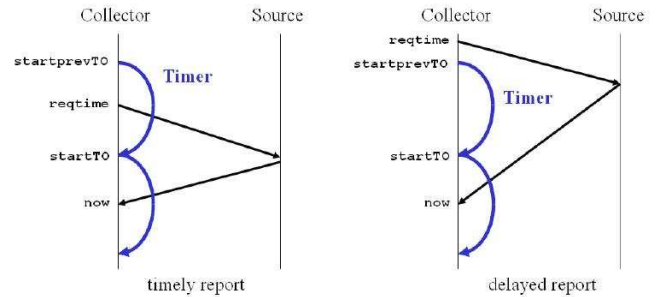


Fig. 1. The two types of reports

The figure 1 explains how the deadline of a report is set. Let $startTO$ be the scheduling time of the timer. Let $startprevTO$ be the previous scheduling time of the timer. When a report is received, the collector extracts from its header the timestamp $reqtime$ indicating the time by which the corresponding probe has been sent. The report is received on time if and only if $startprevTO < reqtime$. The report is a delayed one in the opposite case.

III. PROBLEMS WITH TICP AND NEED FOR CLUSTERING OF INFORMATION SOURCES

In this section, we present the drawbacks of the former version of TICP that motivated our present work.

As we described earlier, the collector has a complete list of information sources' identifiers. A collect session is a succession of rounds. In each round, the collector probes the sources whose reports were not yet received. In a given round, the collector begins by sending a request to the source head of the list, then the following one and so on till the end of the list.

We point out that the ranking of sources in this list has been so far done randomly and independently of any topology information. In reality, sources are more or less far from the collector. The random order results in variable RTTs during the collect session. Since the estimation of round trip time at a given instant depends on its previously measured values, which in our case are unrelated, this estimation seldom gives an idea on the RTT of the next couple probe/report.

An overvaluation of RTT results in a delay in the detection of network congestion; the collector waits more than necessary for already lost reports. This delay means a waste of time and an aggravation of network congestion since the probing rate will not be reduced on time. This is in contradiction with the goals of TICP.

On the other hand, an undervaluation of RTT can cause errors in the computation of the report loss rate since the timer expires prematurely. Thus, some reports are declared lost while they are not. In this case, we reduce unnecessarily the size of the congestion window (*cwnd*) and hence, we increase the collect session duration.

Furthermore with random ranking of sources, packets generated during the collect session can circulate everywhere in the network. At a given moment, this traffic can participate in the congestion of many bottlenecks at the same time. Since it is difficult to adapt the congestion window size to network conditions on all paths from sources to collector, the internet is considered in TICP as a sole bottleneck. Thus TICP does not ensure fairness with concurrent traffic and its mechanism of congestion control is not efficient in case of large networks.

All the drawbacks described above are due to the random order of sending probes to information sources. It is then important to cluster information sources so that those close to each other are probed simultaneously. Also it is important to rank clusters from the nearest to the most distant of the collector so that to ensure that the network conditions vary smoothly and hence TICP congestion control can track them efficiently. The contribution of the current work is to add to TICP such a clustering and ranking mechanism.

A cluster is a group of sources located in the same neighborhood. These sources are such that the latency to

join another node of the cluster is generally lower than the latency to join any source not belonging to it. The idea is that the more the sources are close to each other the more their reports meet the same network conditions on their paths to the collector and the more probable they are located behind the same bottleneck. In this case, the loss of reports indicates that the common bottleneck is congested, so the collector can handle efficiently this congestion by decreasing the rate of sending probes.

The collector probes clusters from the nearest to the farthest. This ensures a smooth variation of the congestion control parameters of TICP, for instance the rate of sending probes and the estimation of round trip time. This again results in an efficient network congestion control.

IV. CLUSTERING OF INFORMATION SOURCES

In this section, we describe our approach to cluster information sources. For this, we use the Global Network Positioning (GNP) system to model the internet by a 2-dimensional euclidean space [3]. A network node in this space is represented by a point. The mathematical distance function gives an approximative value of the RTT between any 2 nodes. To ensure this, a small set of hosts called landmarks distributed across the Internet first compute their own coordinates in this geometric space. These coordinates are then disseminated to any ordinary host who wants to compute its own coordinates relative to the coordinates of the landmarks [4].

The collector and information sources participate in GNP as ordinary hosts. At the end of the GNP operations, each source has a couple of coordinates $H(x_H, y_H)$ and the collector has also its own coordinates $C(x_C, y_C)$.

-2, 2	-1, 2	0, 2	1, 2	2, 2
-2, 1	-1, 1	0, 1	1, 1	2, 1
-2, 0	-1, 0	0, 0	1, 0	2, 0
-2, -1	-1, -1	0, -1	1, -1	2, -1
-2, -2	-1, -2	0, -2	1, -2	2, -2

Fig. 2. Clusters

We define a cluster as being a set of information sources whose representing points are located in a square area. The side of the square is denoted a , which is a parameter of the protocol. The central cluster is the square of side length a whose center is the point representing the collector $C(x_C, y_C)$. A cluster is completely defined by a couple of coordinates (X,Y). These coordinates are those of the center of the

correspondant square relative to the collector coordinates and then normalised by a . The figure 2 illustrates how we cluster information sources.

An information source whose coordinates equals to $H(x_H, y_H)$ belongs to the cluster (X, Y) given by:

- $X = \text{round}(\frac{x_H - x_C}{a})$
- $Y = \text{round}(\frac{y_H - y_C}{a})$

In order to probe information sources from the nearest to the farthest, the collector begins with the central cluster and then follows a spiral trajectory. The figure 3 gives an idea of this trajectory.

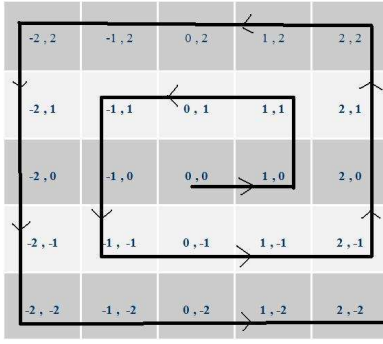


Fig. 3. Order of probing information sources

One can with a simple algorithm find the coordinates of the next cluster during the collection knowing the coordinates of the current cluster.

V. SIMULATION RESULTS

In this section, we discuss the results of our simulations. We have run these simulations in ns-2 [6] in order to evaluate the performance of TICP in its two versions, with and without clustering of information sources. That is why we have implemented GNP and TICP into the simulator ns-2. We generate realistic network topologies for simulations using GT-ITM (Georgia Tech-Internet Topology Modeling) [7][8]. We choose to work on transit-stub topologies which show the complexity and the hierarchical structure of the real internet topology.

The following results are those of simulations run on a transit-stub topology with the following characteristics:

TABLE I
TRANSIT-STUB MODEL PARAMETERS

Parameter	Signification	Scenario
T	Number of transit domains	50
Nt	Average Number of nodes / transit domain	70
K	Number of stub domains / transit node	80
Ns	Average number of nodes / stub domain	70

The average link delay over this network is set to its default value in GT-ITM, which is 18 ms. In each simulation, we choose randomly 500 sources of information and a collector among the nodes of the transit-stub topology. The parameters of TICP are set as in Table II.

TABLE II
TICP PARAMETERS

Parameter	Value
RS	10
probe size	100 Kb
report size	1500 Kb
a	50 ms

A. Network Congestion

We compare between the both versions of TICP with and without clustering of information sources. The comparison criterion is network congestion. Figure 4 represents the evolution of the congestion window size ($cwnd$) as a function of simulation time for TICP without clustering of sources.

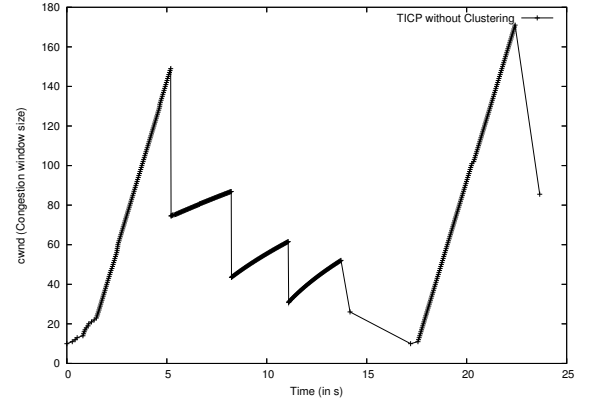


Fig. 4. Cwnd as a function of time for TICP without clustering

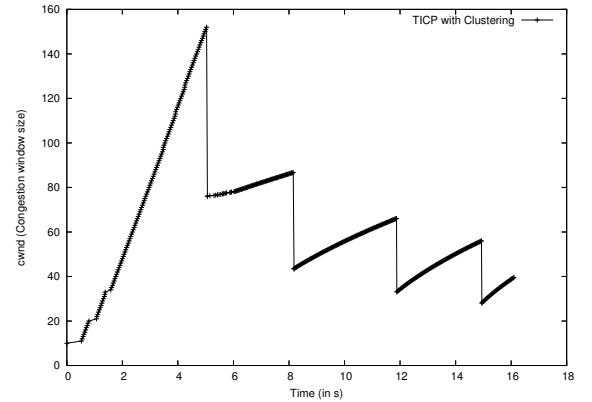


Fig. 5. Cwnd as a function of time for TICP with clustering

We notice that at time 17s there was a reset of $cwnd$ to RS following a severe network congestion (loss rate $> SCT$). TICIP could not adapt the probes sending rate to the available bandwidth at the diverse network bottlenecks. The network is seen by TICIP as a unique bottleneck.

Figure 5 plots the same result but this time for TICIP with clustering of sources. In this case, TICIP remains in the congestion avoidance phase and the severe congestion does not appear. This illustrates that TICIP with clustering adapts the window size to the right network available resources. Indeed, TICIP with clustering of information sources treats bottlenecks one by one rather than at once.

Figure 6 represents the loss rate recorded by TICIP during the collect session for both versions during one simulation.

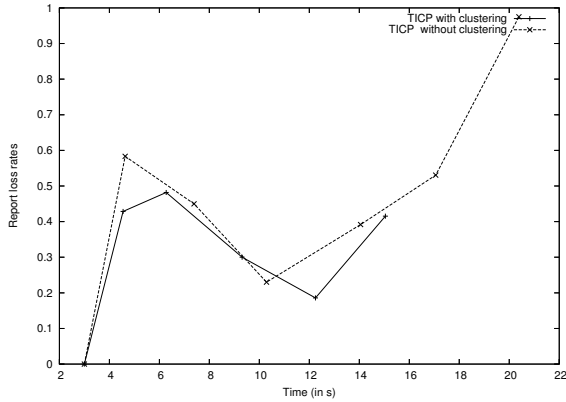


Fig. 6. Loss rate as a function of time

This figure shows that the loss rate for TICIP with clustering of information sources never exceeds 0.5 whereas TICIP without clustering has a loss rate greater than 0.4 in 50% of the time and sometimes it converges to 1. The new version of TICIP avoids reaching high loss rates by adapting the congestion control parameters to network conditions.

B. Collect session duration

In this subsection, we compare between the two versions of TICIP. This time we concentrate on the collect session duration. The figure 7 shows the collect session duration for several simulations of TICIP without clustering of information sources. In each simulation, the order of information sources in the list of the collector is different that is why, we obtain each time a different collect session duration.

For TICIP with clustering, the result is the same since the topology does not change. TICIP with clustering finds the good order of information sources and has a short collect session duration. We save on average 30% of the collect session duration by moving from the TICIP without clustering to TICIP with clustering.

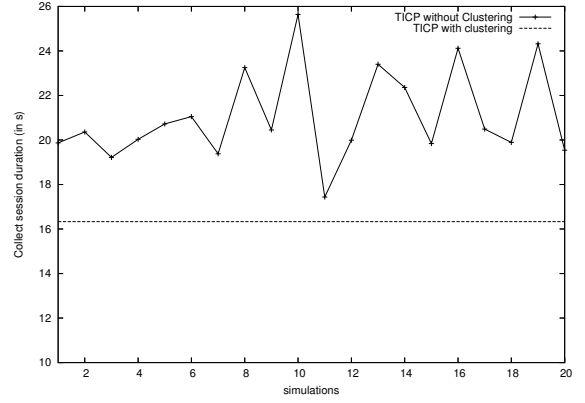


Fig. 7. Collect session duration for different ordering of sources

To evaluate the optimality of TICIP, we implement in ns-2 an information collection protocol having a constant congestion window size. At each simulation, we choose a different window size and we record the minimum of the collect session duration over simulations. Figure 8 presents the evolution of this duration as a function of $cwnd$.

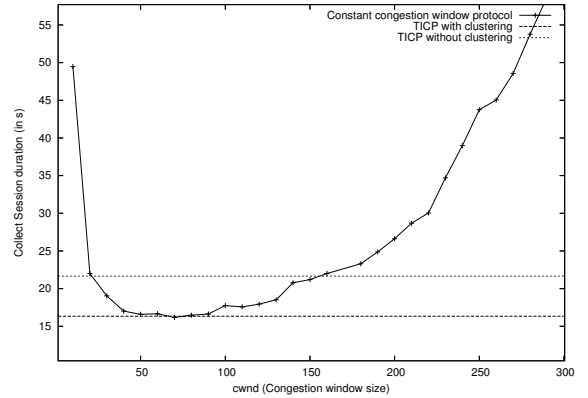


Fig. 8. Optimality of the protocol

The curve has a parabolic pace: for the small congestion window sizes, the collect session duration are long because we have a low probing rate. For large sizes, the network is congested and this lengthens the collect session duration. The role of TICIP is to find dichotomically the good congestion window size that minimizes the collect session duration. We notice in Figure 8 how TICIP with clustering manages to reach the optimal unlike TICIP without clustering.

C. Impact of cluster size

We vary the cluster size and we study its impact on the collect session duration. Taking a very large a is equivalent to TICIP without clustering since sources will be probed independently of their location within the large cluster. Taking a very small results in clusters empty or with a few number of sources which is not efficient since there will be no clustering of sources behind common bottlenecks. There

should be some average a that provides the best performance. Figure 9 validates this intuition where we can see that in network topologies we considered, a value of a around 50ms is the optimal. Each point in the curve of Figure 9 is the average of 5 simulations run over different network realizations satisfying characteristics in Table I. The number of sources is taken equal to 500.

In a future research, we will be studying how the number of sources and the network characteristics impact the choice of the optimal a . For now, we only wanted to justify our choice of a for the network topologies we have considered in our work.

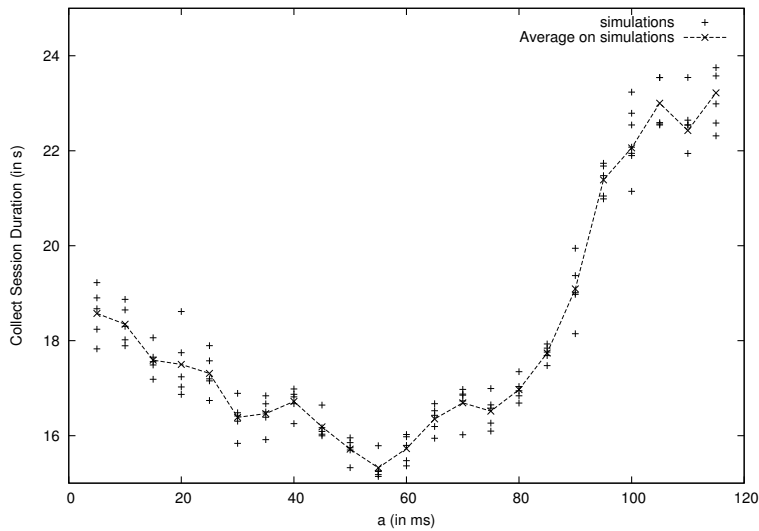


Fig. 9. Impact of cluster size

VI. CONCLUSION AND PERSPECTIVES

TICP is a transport protocol to collect information from a large number of network entities. It aims to control the congestion of the network and to minimize the collect session duration. To ensure a smooth variation of the congestion control parameters, we have added to TICP a mechanism to cluster information sources. The simulation results showed that this mechanism ameliorates the performances of the protocol. In fact, it reduces the loss rate and has short collect session duration. However, the work on TICP is not yet achieved. Our current research focuses on the implementation of the protocol and on its extension to account for sources of large amounts of data.

REFERENCES

- [1] Chadi Barakat, Mohamed Malli, Noamichi Nonaka, "TICP: Transport Information Collection Protocol", in *Annals of Telecommunications*, vol.61, no. 1-2, pp. 167-192, January-February, 2006.
- [2] M.Allman, V.Paxson, W.Stevens, "TCP Congestion Control", RFC 2581, April.1999.
- [3] V.Paxson, M.Allman, "Computing TCP's Retransmission Timer", Internet Draft, April 2000.

- [4] T.S Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches", INFOCOM'02, New York, NY, June 2002.
- [5] T. S. Eugene Ng and Hui Zhang, "Towards Global Network Positioning", Extended Abstract, ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, November 2001.
- [6] The Network Simulator ns-2, <http://www.isi.edu/nsnam/ns/>
- [7] Ellen W.Zegura, Ken Calvert and S. Bhattacharjee, "How to Model an Internetwork", Proceedings of IEEE Infocom '96, San Francisco, CA.
- [8] Ken Calvert, Matt Doar and Ellen W. Zegura, "Modeling Internet Topology", IEEE Communications Magazine, June 1997.