

A Non-cooperative Active Measurement Technique for Estimating the Average and Variance of the One-way Delay ^{*}

Antonio A. de A. Rocha, Rosa M. M. Leão, and Edmundo de Souza e Silva

COPPE/PESC and Computer Science Department
Federal University of Rio de Janeiro(UFRJ)
Cx.P. 68511 Rio de Janeiro, RJ 21945-970 Brazil
{`arocha,rosam,edmundo`}@`land.ufrj.br`

Abstract. Active measurements are a useful tool for obtaining a variety of Internet metrics. One-way metrics, in general, require the execution of processes at the remote machine and/or machines with synchronized clocks. This work proposes a new algorithm to estimate the first two moments of the one-way delay random variable without the need to access a target machine and to have the machine clocks synchronized. The technique uses the IPID field information and can be easily implemented using ICMP Echo request and reply messages.

Key Words: Active network measurement, One-way delay, IPID field

1 Introduction

Active Internet network measurements are an important tool for aiding the modeling and analysis process, helping understanding the characteristics of such complex system as the Internet, and ultimately improve the performance of applications. There are many algorithms that have been proposed in the literature to estimate variables such as packet delays, jitter, loss rate, bottleneck capacity and tools that implement such algorithms. Some performance network metrics, such as round trip packet loss and delay, can be easily obtained. Existing tools like PING use the ICMP protocol and machines over the Internet are usually configured to send an *echo reply* message in response to an *echo request*.

Although round trip measures are relatively easy to compute, it is much harder to obtain one way measures since, in most cases, we cannot simply assume that the forward and reverse paths are symmetric. Forward and reverse paths may have different bandwidths and the set of routers in each path may also be distinct. Even if packets moving through the forward and reverse paths go through the same routers and links, the one-way performance metrics may drastically differ because of traffic asymmetries in each direction.

Current techniques to estimate the one-way delay (or loss rate) require the execution of processes at the remote machine to collect the arriving probes and to

^{*} This work is supported in part by grants from CAPES, CNPq and FAPERJ.

perform a set of operations. For instance, information about the received probes such as sending/receiving times must be collected and evaluated. Therefore, most active measurement tools for estimating this one-way metrics require that tool processes execute at the probe receiver machine. An issue is to obtain the metrics when remote access is not possible. Another difficulty that arises in computing one-way delays is the lack of clock synchronization between end point machines in the source-destination path if synchronization devices such as GPS equipments are not used. Techniques exist to deal with this problem (see [1] and references therein.)

The problem of computing one-way metrics when remote access is impossible has been addressed recently, exploiting information contained in the identification field of the IP header (IPID). Among the metrics obtained by these non-cooperative techniques are the one-way loss rate [2, 3]; out of order arrivals [2, 4]; and the *difference* between the one-way delay from two machines that are sources of probes to a single target machine [5]. In this last case no access is granted to the measurement tool.

In this work we present a new algorithm to estimate the first two moments of the one-way delay (OWD) random variable between two points, where one of them, the target machine, does not run any process of the associated measurement tool. That is, the access is permitted only at the machines that generate probes and no execution privileges are necessary at the target receiver. The technique uses IPID information and requires that at least two source machines generate probes to a target. The technique can be easily implemented using ICMP Echo request messages and ICMP Echo reply. No special clock synchronization equipment or protocol is required at either the sources or the target machine.

The remainder of this paper is organized as follows. Section 2 briefly surveys methods that employ the IPID field to compute measures of interest. Section 3 describes the problem we solve and present the proposed technique for the case that the two probe source machines have their clocks synchronized. We extend the technique in section 4 to relax the requirement that the source machine clocks are synchronized. In section 5 we evaluate the efficacy of the technique both through simulation and experimentation using the PlanetLAB environment. Section 6 summarizes our main contributions.

2 Related measurement techniques using IPID

The IPID is a 16 bits IP datagram header field [6]. It is used by the IP layer to fragment and reassemble a datagram. The algorithm used to calculate the IPID value depends on the operating system. Several of them, such as Windows, FreeBSD, and Linux (up to kernel version 2.2) implement the IPID as a simple global counter.

Recent works in the literature exploit the IPID field values for estimating network metrics. The authors of [5] present a survey of previous work and classify the existing techniques into three categories used for: estimating traffic inten-

sity [7]; identifying the clustering of sources [8, 7]; and identifying packet losses, duplication, and arrival order [2, 4].

The existing techniques proposed to exploit the IPID field have to deal with the wrap around problem. A packet sent by a host later in time may have smaller IPID value than the IPID carried by another packet generated earlier from this same host. The reason for that is the limited size of the IPID field (16 bits) which forces the IPID value to return to zero after reaching 2^{16} . This problem was addressed before in [2, 4, 5]. In [5] methods are described to deal with the wrap around problem and to correct the IPID sequence.

In [5] three new measurement techniques were proposed based on the IPID field: (i) the first is used to infer the amount of network internal traffic generated by a server, from a single passive measurement point; (ii) another serves to identify the number of load-balancing servers behind a single IP address; (iii) the last is an active measurement technique to infer the one-way path delays difference from two distinct sources A and B to a target D.

Our work builds upon technique (iii) of [5] which assumes that the clocks of the source machines are synchronized by GPS and also that the target machine implements a global counter for the IPID field. In that work, the source machines A and B generate probes to the target machine D, at constant intervals of value δ_a and δ_b , respectively, and D sends back to the sources ICMP echo reply packets. From the IPID values, one can identify when a probe generated by A arrives at D between two consecutive probes from B. Let n_A (n_B) be the number of probes that are generated from A (B) starting at instant τ_A (τ_B) such that the (n_A)-th probe arrives in D between the (n_B)-th and ($n_B + 1$)-th probe. Then (see [5]), $\tau_B + d_{BD} + n_B\delta_B \leq \tau_A + d_{AD} + n_A\delta_A \leq \tau_B + d_{BD} + (n_B + 1)\delta_B$. Since δ_B is assumed small, the one-way delay difference $d_{AD} - d_{BD}$ can be estimated by:

$$d_{AD} - d_{BD} \approx \tau_B - \tau_A + n_B\delta_B - n_A\delta_A. \quad (1)$$

3 Technique to estimate the first two moments of the OWD

Similarly to the technique of [5] we assume that probes are generated from two (or more) sources to a single target machine. The target implements a global counter for the IPID field and one does not have any access to that machine. In order to facilitate the explanation of the algorithm and to validate the technique we initially assume the clocks of the source machines are synchronized. However, this assumption is relaxed later, in Section 4.

Our goal is to estimate d_{AD} and d_{BD} , i.e., the one-way delays to a target D from sources A and B, respectively. Suppose that machines A and B generate probes to the target machine D and, during some interval Δ , two of these probes, one from each machine, arrive at D *in sequence*. The corresponding ICMP packets generated by D that are sent back to the source machines will have IPID values that differ by a small amount (assuming δ_B is small). In this scenario, we can establish the following system of equations:

$$\begin{cases} d_{AD} + d_{DA} = RTT_{ADA} \\ d_{BD} + d_{DB} = RTT_{BDB} \\ d_{AD} - d_{BD} = \Psi_{AD-BD} \\ d_{DA} - d_{DB} = \Psi_{DA-DB} \end{cases} \quad (2)$$

where, Ψ_{ID-JD} is the *estimated* OWD difference $d_{ID} - d_{JD}$ and RTT_{ADA} and RTT_{BDB} are the *estimated* round trip times ADA and BDB , respectively.

These equations are linearly dependent and so we need extra information to obtain a unique solution. In what follows we address this issue.

The OWD of a probe from source A , d_{AD} , is equal to the sum of four terms: the overall propagation delay from A to D , T_{AD}^{prop} ; the sum of queueing times at the routers in path AD , T_{AD}^{queue} ; the sum of the transmission times at the links, T_{AD}^{tx} ; the overall processing time, T_{AD}^{proc} . Therefore, assuming that the processing times are negligible,

$$d_{AD} = T_{AD}^{tx} + T_{AD}^{prop} + T_{AD}^{queue}. \quad (3)$$

We further assume that the propagation times in the forward and reverse paths (AD and DA) are identical, however, the capacities and queue times in the forward and reverse paths can differ. Note that the technique does not assume symmetric paths, that is, although $T_{AD}^{prop} = T_{DA}^{prop}$, T_{AD}^{tx} and T_{AD}^{queue} may differ from T_{DA}^{tx} and T_{DA}^{queue} .

Our approach to estimate the transmission and propagation times is based on the generation of probes with two distinct sizes following the three step procedure described below. First, n probes with identical sizes l are sent from A to D . Consequently, the ICMP protocol send ICMP packets back to A with the same size l . Second, the same procedure is repeated but using probes sizes one order of magnitude greater than that used in the first step, that is $10l$. Finally, n probes of size $10l$ are sent from A to D . However, this time we would like to get a return reply of size l .

Sending ICMP request probes and receiving ICMP replies of the same size is trivial, since request/replies have always identical sizes. In the ICMP protocol specification [9], if a machine receives an ICMP Echo request message, it must send back an ICMP Echo reply. The receiving machine will change only the header of the Echo request message, and will send an Echo reply with the same payload. However, the ICMP protocol specification does not allow the Echo request sender to control the size of Echo reply message. To overcome this limitation, we generate packet pairs to emulate the effect of sending a probe Echo request of size X and receiving an Echo reply of size $Y < X$. The method consists of sending two back to back probes(packet pair): the first is an ICMP Echo reply message of size equal to $10l$ bytes and the second is an ICMP Echo request message of size l . (Note that, in the method, an ICMP Echo reply message is generated by the source machine spontaneously, without receiving an Echo request packet.) Both probes cross the same forward path until they reach the destination. (It is a common and reasonable assumption when using packet pair techniques to consider that both packets of the pair follow the same route from

the source to the destination host [10].) In this scenario, the second probe is delayed at each hop by the transmission time of the first probe, since the first is a packet 10 times greater than the second. When the first probe (ICMP Echo reply of size $10l$) arrives at the target machine, it is discarded by the ICMP protocol. On the other hand, the second probe is an ICMP Echo request of size l , and the target machine immediately sends back an ICMP Echo reply of size l . From the three steps above, we can estimate the RTT of a packet with the same size in the forward and reverse directions and the RTT of a packet with size $10l$ in the forward direction and l in the reverse direction. From these estimates we obtain extra equations to solve our problem.

In our work we choose $l = 50$ bytes. Let $RTT_{m,ADA}^{X-Y}$, be the minimum round trip delays obtained when probes of size X are sent to D and the return has size $Y \leq X$. For $RTT_{m,ADA}^{X-Y}$, it is common to assume that the queue time is negligible ([11–14]). Since, from our assumptions, $T_{AD}^{prop} \approx T_{DA}^{prop}$, we have:

$$\begin{cases} T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} = RTT_{m,ADA}^{50-50} \\ 10T_{AD}^{tx} + 10T_{DA}^{tx} + 2T_{AD}^{prop} = RTT_{m,ADA}^{500-500} \\ 10T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} = RTT_{m,ADA}^{500-50} \end{cases} \quad (4)$$

The equations above are linearly independent and so T_{AD}^{tx} , T_{DA}^{tx} , and T_{AD}^{prop} can be estimated from the RTTs. A similar system has to be solved to obtain the values of the transmission and propagation times between B and D . To estimate the OWD, we have still to compute the overall queueing times.

Let $\psi_{AD-BD}^{queue} = T_{AD}^{queue} - T_{BD}^{queue}$, and $\psi_{DA-DB}^{queue} = T_{DA}^{queue} - T_{DB}^{queue}$. The values of ψ_{AD-BD}^{queue} and ψ_{DA-DB}^{queue} can be easily estimated from the transmission and propagation times in each path:

$$\begin{cases} \Psi_{AD-BD}^{queue} = \Psi_{AD-BD} - (T_{AD}^{tx} + T_{AD}^{prop} + T_{BD}^{tx} + T_{BD}^{prop}) \\ \Psi_{DA-DB}^{queue} = \Psi_{DA-DB} - (T_{DA}^{tx} + T_{DA}^{prop} + T_{DB}^{tx} + T_{DB}^{prop}) \end{cases} \quad (5)$$

Since we are able to calculate the difference between the queue time in the forward and reverse paths both from sources A and B , we can rewrite (2) considering only the queueing time in each path:

$$\begin{cases} T_{AD}^{queue} + T_{DA}^{queue} = T_{ADA}^{queue} \\ T_{BD}^{queue} + T_{DB}^{queue} = T_{BDB}^{queue} \\ T_{AD}^{queue} - T_{BD}^{queue} = \Psi_{AD-BD}^{queue} \\ T_{DA}^{queue} - T_{DB}^{queue} = \Psi_{DA-DB}^{queue} \end{cases} \quad (6)$$

where, T_{ADA}^{queue} and T_{BDB}^{queue} are the probe queueing times along the round trip paths ADA and BDB , respectively.

When T_{ADA}^{queue} or T_{BDB}^{queue} is equal to zero, the equations above are linearly independent. If $T_{BDB}^{queue} = 0$ then $T_{BD}^{queue} = T_{DB}^{queue} = 0$. From (6), $T_{AD}^{queue} = \Psi_{AD-BD}^{queue}$ and $T_{DA}^{queue} = \Psi_{DA-DB}^{queue}$. We have then obtained all the necessary quantities to estimate d_{AD} from (3) (and identically d_{DA}).

The details of the procedure are summarized as follows.

Algorithm 1

Step 1: Generate n_A and n_B probes from machines A and B to D . Compute samples for RTT_{ADA}^{X-Y} , RTT_{BDB}^{X-Y} .

Step 2: From the RTT samples, obtain the minimum values of RTT for each source, $RTT_{m,ADA}^{X-Y}$, $RTT_{m,BDB}^{X-Y}$, and then T_{AD}^{tx} , T_{DA}^{tx} , T_{AD}^{prop} , T_{BD}^{tx} , T_{DB}^{tx} , T_{BD}^{prop} using equations (4).

Step 3: Select a sub-set \mathcal{K} of k probe pairs (p_A, p_B) , where p_A and p_B are probes sent from A and B , respectively. A pair (p_A, p_B) is selected if the corresponding replies arriving from D have consecutive IPID values. Obtain samples for Ψ_{AD-BD} and Ψ_{DA-DB} from the sub-set \mathcal{K} . Compute Ψ_{AD-BD}^{queue} and Ψ_{DA-DB}^{queue} using equations (5).

Step 4: Select a pair of the sub-set \mathcal{K} if the RTT_{BDB} value is within the interval $[RTT_{m,BDB}, 1.01RTT_{m,BDB}]$. Call this sub-set \mathcal{L} and suppose that it has m_A pairs. Select a pair of the sub-set \mathcal{K} if the RTT_{ADA} value is within the interval $[RTT_{m,ADA}, 1.01RTT_{m,ADA}]$. Call this sub-set \mathcal{M} and suppose that it has m_B pairs. (Recall that when RTT has minimum value, the queueing times in both ways are negligible.)

Step 5: From each pair of the sub-set \mathcal{L} estimate one sample for T_{AD}^{queue} and T_{DA}^{queue} , and from each pair of the sub-set \mathcal{M} estimate one sample for T_{BD}^{queue} and T_{DB}^{queue} , using equations (6).

Step 6: Use equation (3) to compute m_A samples of d_{AD} and d_{DA} , and m_B samples of d_{BD} and d_{DB} .

Step 7: The average and variance of the OWD can be calculated from:

$$\bar{d}_{path} = \frac{1}{m_j} \sum_{n=1}^{m_j} d_{path}(n)$$

$$Var(d_{path}) = \frac{1}{m_j - 1} \sum_{n=1}^{m_j} (d_{path}(n) - \bar{d}_{path})^2$$

where, for $j = A$ ($j = B$) the *path* index is replaced either by AD or DA (respectively by BD or DB).

4 Extension for non-synchronized sources

In the previous section we assumed that the probe sources had their clocks synchronized. In what follows, we show that this assumption can be relaxed.

The main problem for estimating the OWD if the probe sources have their clocks not synchronized is the clock *Offset* and *Skew*. Solutions for removing the *Offset* and *Skew* to calculate the one-way delay between machines have been discussed in the literature and solutions proposed [11, 15, 12–14, 1]. In these techniques, if one wants to measure, for instance, the OWD, d_{AD} , from machine A to D , probes are generated from A to D and both A and D must run processes of the measurement tool. However, in the technique described in Section 3 to calculate d_{AD} , probes are generated from machines A and B to a target machine D . Thus, we can not immediately use the methods in the literature.

We adapt the algorithm of [14] to remove the *Skew*. In [14] the *Skew* estimation requires the computation of the lower bound of the convex hull of a sequence

of points (i, j) where i is the sending time of a probe from the source machine and j is the OWD computed at the destination. In our extension, consider the probes of size l generated by machines A and B to estimate the delay differences Ψ_{AD-BD} and Ψ_{DA-DB} . Let $\Omega = [(\tau_{BD}(i), d_{BD-DA}(i)) : i = 1, \dots, k]$ be a sequence of k points, taken from the probe pairs of set \mathcal{K} defined in previous section. (Recall that probe pairs in this set are those from machines A and B that arrived approximately at the same time in D .) Take the i -th pair in \mathcal{K} . $\tau_{BD}(i)$ is the sending time of the probe from B to D in this pair, and $d_{BD-DA}(i)$ is the arrival time of the Echo reply of probe from A in this pair minus the sending time of the probe from B in this pair. Intuitively, since the arrival times at D from the probes in the pair are approximately the same, $d_{BD-DA}(i)$ is identical to the time difference that one would obtain if B sends a probe directly to A in a path that passes through D . From the Ω sequence, we can remove the *Skew* in the same manner as in [14].

To estimate and remove the *Offset*, the algorithm presented in [13] could be used, if additional probes are generated from machine A directly to B , and vice versa. To avoid the generation of extra probes, we propose to estimate the *Offset* based on the minimum values of RTT_{BDB} and d_{BD-DA} . Let $d_{m,BD-DA}^s$ be the minimum value of d_{BD-DA}^s in the sequence obtained from Ω after the *Skew* is removed. Let $RTT_{m,BDB}$ be the minimum round trip delay estimated from B to D . It is reasonable to assume that these values are obtained when the corresponding probes see no queuing delay at the routers. Then we have:

$$RTT_{m,BDB} = T_{BD}^{tx} + T_{BD}^{prop} + T_{DB}^{tx} + T_{DB}^{prop}$$

$$d_{m,BD-DA}^s = T_{BD}^{tx} + T_{BD}^{prop} + T_{DA}^{tx} + T_{DA}^{prop} - O_{AB}$$

where O_{AB} is the offset between A and B . From the equations above the *Offset* can be immediately estimated.

5 Experiments and validation

The proposed technique was evaluated through simulation and experiments over the Internet. The main goal of the simulation model was to analyze the technique for different values of bandwidth utilization and when the clock machines are not synchronized. (All the results below are in microseconds.)

The simulation model developed using the TANGRAM-II Modeling environment [1] is illustrated in Figure 1. Objects *Host_A*, *Host_B* and *Host_Target* represent the two sources and target machine, respectively. The routers represented in the model have different bandwidths. A global IPID counter is simulated at the target machine, as well as the Echo reply packets. Besides probe packets, we generate cross traffic using a set of On-Off sources as suggested in [16]. The residence time in the state On and Off follows a Pareto distribution with parameter $\alpha < 2$. We validate the technique by comparing a trace with the probe arrival times collected at the *Host_Target* object with the values estimated by the proposed technique. We also compute the relative error of the average and variance of the OWD.

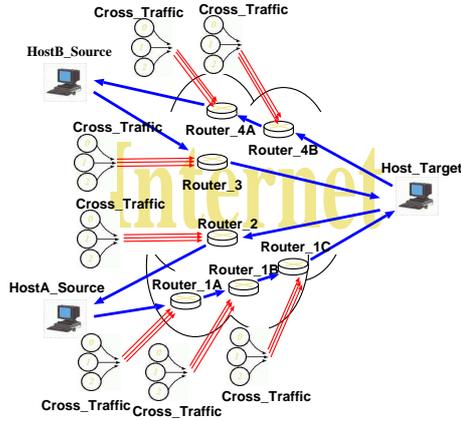


Fig. 1. Simulation model.

We show the results for two scenarios. In the first scenario, link utilizations vary between 30% and 50% and the source machines clocks are synchronized.

Figure 2 shows the results for path *DB* as a function of the simulation time considering the first scenario. When the simulation time is smaller than 20 seconds, the estimated values are inaccurate. This occurs because the number of samples is too small to obtain an accurate OWD estimation. However, after 40s, the accuracy is very good.

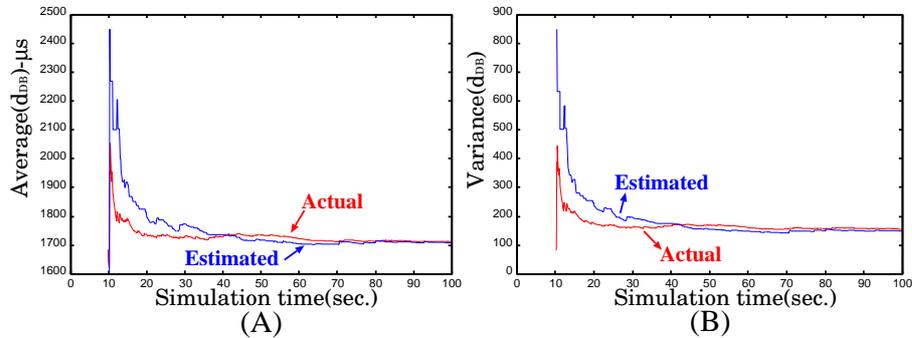


Fig. 2. Average and variance of OWD path DB (link utilization between 30 and 50%).

Figure 3 shows the results for path *AD* when the second scenario is considered. The utilization varies between 65% and 80%, and clocks are not synchronized. In this case longer simulation times are needed as compared to the first scenario. This is expected since, for a given time interval t , the higher the utilization, the smaller the number of samples that can be obtained to estimate the

measures. However, even for high utilization values, the estimation procedure converges fast. Table 1 presents the relative error of the average and variance of the OWD for both scenarios. The relative errors are less than 2% (average) and 13% (variance) when the utilizations are low to moderate. In the second scenario with higher utilizations the relative errors are less than 8% and 29%, respectively. However, one can obtain smaller relative errors by increasing the measurement time.

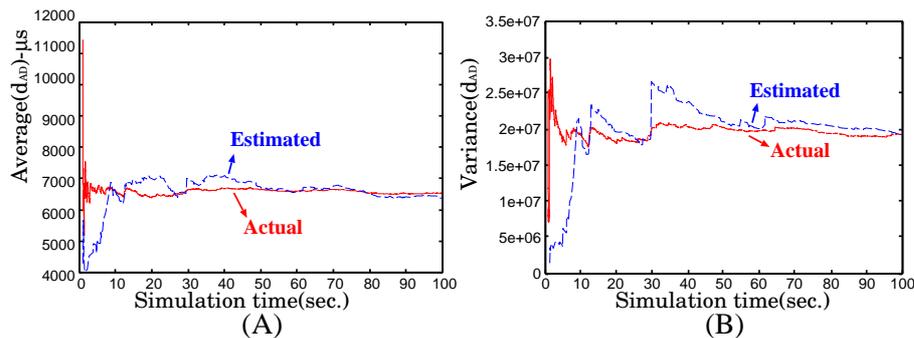


Fig. 3. Average and variance of OWD path AD (link utilization between 65 and 80%).

Table 1. Relative error - simulation.

Path	Scenario 1	Scenario 2
	Average / variance	Average / variance
AD	0.020 / 0.058	0.025 / 0.001
DA	0.013 / 0.011	0.082 / 0.290
BD	0.013 / 0.132	0.057 / 0.220
DB	0.002 / 0.033	0.062 / 0.078

In what follows we show the results of experiments over the Internet. In order to generate probes according to our algorithm we adapted the TANGRAM-II Traffic Generator [1].

In all experiments the probe generation rates for each source are 1,000 packets/s and 100 packets/s. We used machines that are synchronized by GPS to be able to estimate the actual delay values. (Therefore, there is no need to remove the *Skew* and *Offset*.) Considering that most packets have size $l = 50$ bytes, the overload introduced in the network are respectively 400 kbps and 40 kbps which can not be considered an intrusive traffic for the actual network rates. In the first set of experiments three machines were employed: one at UFRJ (Brazil), other at UNIFACS (Brazil) and the third at UMass (USA). Experiments with 30

minutes duration each were executed. The target machine was varied for each. A sample of all results is shown in Table 2, and the OWD relative error was less than 2%.

Table 2. Relative error - experiments UFRJ, Unifacs and UMass.

Path	Relative Error		Path	Relative Error	
	Average / variance			Average / variance	
UFRJ-UMass	0.004 / 0.626		UFRJ-Unifacs	0.009 / 0.152	
UMass-UFRJ	0.005 / 0.022		Unifacs-UFRJ	0.009 / 0.038	
Unifacs-UMass	0.016 / 0.710		Umass-Unifacs	0.001 / 0.015	
UMass-Unifacs	0.015 / 0.087		Unifacs-Umass	0.001 / 0.099	

The second set of experiments was performed using three PlanetLAB machines and during different times of the day. Machines at Berkeley and U.K. generate probes to a target machine in Hong Kong during the 5 first minutes of each hour for 24 hours. Figure 4 illustrates the results obtained for the average and variance of the path from Hong Kong to Berkeley. The figure clearly shows that the technique was able to capture very accurately the behavior of the measures during several hours.

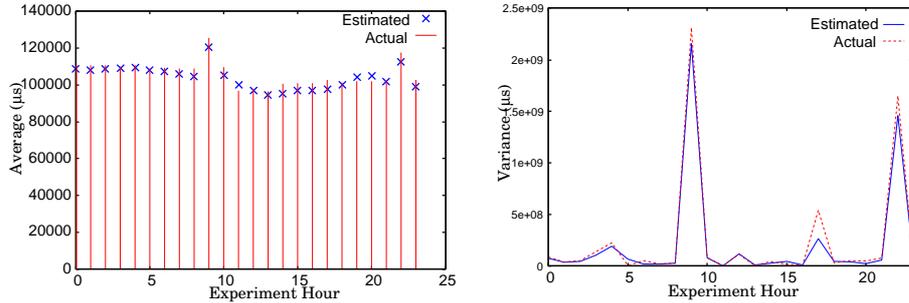


Fig. 4. Average and Variance of the OWD in path from Hong Kong to Berkeley - during 24 hours experiment.

In the third set of experiments (also using PlanetLAB) the source machines were at Seattle and Texas and the target machine was in Korea. Probes were generated in the first minute of each hour, for 10 hours (between 5am to 3pm GMT). Each one minute session was divided into 6, 10 seconds duration sub-sessions. For each sub-session we estimate one sample of the average OWD. Using these 6 samples, we compute the sample average and the confidence interval of the OWD for one session. We consider a 95% level of significance. Figure 5 shows the confidence intervals for both the values estimated by our algorithm

and actual measures of the average OWD, for the Korea-Seattle path. The figure confirms the good accuracy of our approach.

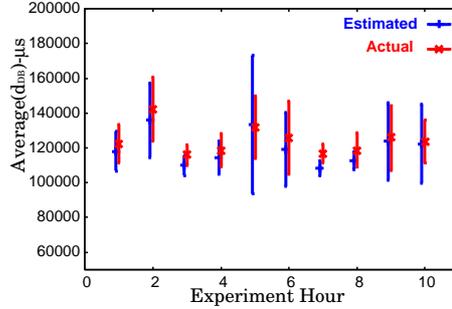


Fig. 5. Confidence interval for the average OWD - path Korea-Seattle.

The last experiment using PlanetLAB hosts involved several machines. Machines at Texas, Stanford, Berkeley, Unifacs, Kaist, France, Israel, U.K. and Hong Kong generated probes simultaneously to a target machine at UMass. The main purpose of this experiment was to investigate the OWD of several paths from different sources machines to a single target. Table 3 illustrates the results. This experiment shows that the technique could be employed, for instance, by an application to choose the “best” path (i.e., with the minimum value of the OWD and/or variance) to serve a request from a client machine (in this example UMass).

Table 3. Experiment from several sources to UMass.

Path	Average			Variance		
	Estimated / Actual / Relative Error			Estimated / Actual / Relative Error		
Texas-UMass	26091 / 25852 / 0.009			150976 / 227899 / 0.509		
Stanford-UMass	35097 / 35562 / 0.013			256008 / 261035 / 0.019		
U.K.-UMass	43777 / 43948 / 0.003			140461 / 199699 / 0.421		
Berkeley-UMass	40680 / 40602 / 0.001			19321457 / 20427774 / 0.057		
Hong Kong-UMass	19321 / 20427 / 0.057			178582 / 263283 / 0.474		
Israel-UMass	85975 / 85607 / 0.004			570297 / 653080 / 0.145		
Kaist-UMass	107122 / 106971 / 0.001			219852 / 292970 / 0.332		
Unifacs-UMass	86716 / 86425 / 0.003			982904 / 227814 / 0.768		
France-UMass	48513 / 48338 / 0.003			207378 / 260358 / 0.255		

6 Main contributions

In this work we propose a novel technique to estimate the average and the variance of the one-way delay. Several experiments using the PlanetLAB infrastructure were performed and the results obtained show that the average and variance of OWD can be accurately estimated.

An important characteristic of the proposal is that it does not require to run any process at the remote machines. Furthermore, it can be used even if the clocks of the source machines are not synchronized. Therefore it is a valuable tool to estimate OWDs from machines one has access to run processes to different machines were no access is granted, provided that the targets run an OS which implements a global IPID counter (such as Windows machines).

References

1. de Souza e Silva, E., Leao, R., Rocha, A., Duarte, F., Silva, A., Filho, F.S., Jaime, G., Muntz, R.: Modeling, Analysis, Measurement and Experimentation with the Tangram II Integrated Environment. In: Int. Conf. on Performance Evaluation Methodologies and Tools. (2006)
2. Mahajan, R., Spring, N., Wetherall, D., Anderson, T.: User-level internet path diagnosis. In: 19th ACM SOSP. (2003) 106–119
3. Savage, S.: Sting: a TCP-based Network Measurement Tool. In: USENIX Symposium on Internet Technologies and Systems. (1999) 71–79
4. Bellardo, J., Savage, S.: Measuring Packet Reordering. In: 2nd ACM SIGCOMM IMW. (2002)
5. Chen, W., Huang, Y., Ribeiro, B., Suh, K., Zhang, H., de Souza e Silva, E., Kurose, J., Towsley, D.: Exploiting the IPID Field to Infer Network Path and End-System Characteristics. In: Lecture Notes in Computer Science. Volume 3431. (2005) 108–120
6. Postel, J.: Internet Protocol (1981) IETF RFC 791.
7. Insecure.org: Idle Scanning and related IPID games (1997) <http://www.insecure.org/nmap/idlescan.html>.
8. Bellovin, S.: A Technique for Counting NATed Hosts. In: 2nd ACM SIGCOMM IMW. (2002) 267–272
9. Postel, J.: Internet Control Message Protocol (1981) IETF RFC 792.
10. Dovrolis, C., Ramanathan, P., Moore, D.: What do Packet Dispersion Techniques Measure? In: IEEE Infocom. Volume 1. (2001) 905–914
11. Loung, D., Biro, J.: Needed Services for Network Performance Evaluation. In: IFIP Workshop on Performance Modeling and Evaluation of ATM Networks. (2000)
12. Paxson, V.: On Calibrating Measurements of Packet Transit Times. In: ACM Sigmetrics. (1998) 11–21
13. Tsuru, M., Takine, T., Oie, Y.: Estimation of Clock Offset from One-way Delay Measurement on Asymmetric Paths. In: SAINT International Symposium on Applications and the Internet. (2002) 126–133
14. Zhang, L., Liu, Z., Xia, C.: Clock Synchronization Algorithms for Network Measurements. In: IEEE Infocom. (2002) 160–169
15. Moon, S., Skelly, P., Towsley, D.: Estimation and Removal of Clock Skew for Network Delay Measurements. In: IEEE Infocom. (1999) 227–234
16. Taqqu, M.S., Willinger, W., Sherman, R.: Proof of a Fundamental Result in Self-Similar Traffic Modeling. In: ACM Computer Communications Review. (1997) 5–23