One-way Delay Measurement and Characterization

Ana Hernandez, Eduardo Magaña
Universidad Pública de Navarra, Pamplona, Spain
{ana.hernandez, eduardo.magana}@unavarra.es

Abstract

In network characterization and network tomography, delay is one of the meaningful parameters to consider. Inside end-to-end delay, one-way delay provides much more information than typical round-trip time. Measuring one-way delay is complex and several components can be distinguished. Measurement errors committed by conventional monitoring hardware are presented. A high-precision European measurement infrastructure has been used to run our experiments and to check general behavior for one-way delay. Maximum likelihood estimation is used to provide generic distribution fitting for one-way delay. The information of one-way delay is also applied to available bandwidth estimation and topology changes during extensive periods of time.

Keywords: one-way delay, high-precision measurement, bandwidth estimation.

1 Introduction

The end-to-end delay of packets is an interesting characteristic of Internet. It provides extensive information about state of networks and application performance. For network characterization, we can get values of transmission and propagation delays looking at the minimum end-to-end delay [1]. The variations of delay are related with queuing effects [2]. Even we can infer topology information, congestion state or route changes from end-to-end delay measurements [3]. From an application performance point of view, large delays will make difficult to obtain a sustainable high-bandwidth flow because of TCP dependency on RTT (Round Trip Time) [4]. The responsible are the slow-start mechanism and the effect of multiple losses over TCP. Besides, irregular variations in delay (jitter) are hard conditions for real-time applications such as videoconference or Voice-over-IP [5]. In this kind of applications, buffers cannot be dimensioned as needed because there are requirements of maximum delay to get interactivity. Finally, end-to-end delay is one of the main parameters to consider in quality of service systems.

For example, knowing the delay distribution along certain path we can deduce if the service is going to receive the requested maximum delay or even we can dimension the buffer size for interactive applications. It is clear then the importance of end-to-end delay for multiple applications and the interest of a depth understanding and characterization of end-to-end delay.

End-to-end delay can be divided into several components [1]:

- Transmission delay: the time needed to put all bits of a packet over a data link.
- Propagation delay: the time needed to propagate a bit though the data link.
- Processing delay: the time needed to process a packet in each router.
- Queueing delay: the time needed to wait in a router queue before transmission.

Processing and queueing delays are stochastic random variables due to variability in processing tasks in routers and in network conditions respectively [1]. However, nowadays processing time should be almost constant except on rare occasions because modern IP routers use hardware assisted forwarding with wire speed. Transmission and propagation delays are almost constant for a certain path, because they depend on link capacity and distance respectively [9]. For our study, we will consider the complete end-to-end delay. Propagation and transmission delays will give a good approximation to the minimum end-to-end delay and this minimum will have to be a constant value.

Normally, end-to-end delay is approximated by RTT. Those measurements are very easy to obtain

This work has been supported by Integrated Project Evergrow (Contract No 001935) funded by the FP6/IST/FET Programme of the European Commission and by Spanish Ministry of Education and Science (Project PINTA TEC2004-06437-C05-03).

using ICMP Request/Reply packets through the *ping* tool and controlling only one of the end nodes. However, asymmetry of paths makes this approximation invalid because delay can be different for both directions of a path [6]. Another procedure is measuring one-way delay (OWD) which is a more complex measurement. It requires expensive hardware but it provides a better characterization of the parameter [7][8].

Several measurements of OWD have been made before, but in very closed environments, with low-precision hardware or using short time intervals [1][9]. Usually, active techniques are used that require injecting test packets in the network (intrusive), although passive techniques can be used too, observing the timing of normal traffic between both ends. Passive techniques are harder to control and therefore they provide less interesting results.

In this paper we have used the European Traffic Observatory Measurement InfrastruCture (ETOMIC) [10][11] to make high precision one-way delay active measurements. This is an active/passive monitoring infrastructure with GPS-synchronized monitoring nodes and special networking cards (DAG Endace 3.6GE). This tool allows high-precision measurements because timestamp is inserted in the sending node by the network card and recovered in the receiving node by the network card again, avoiding effects of general purpose operating systems present in the nodes (Linux OS). Nowadays there are 16 nodes distributed along Europe as show in Fig.1.

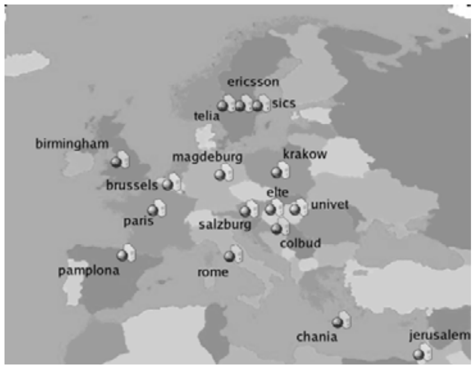

Fig.1. ETOMIC nodes along Europe

ETOMIC allows us to obtain one-way delay measurements with accuracy in the order of hundreds of nanoseconds, much better than any similar platform as RIPE NCC infrastructure [1]. A burst composed by several packets can be transmitted with precise inter-packet timing (nanoseconds) because this burst is programmed in the network card itself.

These measurements will allow us to model delay distributions, study correlation effects between paths and estimate the available bandwidth in our end-to-end path. We will also have information from these measurements about jitter, losses and packet reordering.

This paper is organized as follow. In section 2, we compare OWD measured with conventional and specific network cards. Next section will provide a model for delay distribution and the effect of congested networks. Section 4 will show the long-range evolution of OWD. Finally, conclusions will be presented.

2 One-way delay with high-precision hardware

ETOMIC nodes are PCs equipped with 2 network interface cards:

- Conventional (eth0): low cost network card, used mainly for management functionalities but available also for monitoring. The PC box is GPS-synchronized (by means of the pulse-per-second signal, PPS, captured by the Linux kernel).
- High-precision (dag): Endace DAG 3.6GE with its own processor for advanced functionalities and GPS-synchronized (by PPS again).

The high-precision card has the GPS directly connected to the card. Instead of having the kernel timestamp the arriving/sending packets, timestamping is performed as soon as the packet arrives/leaves by the card itself. As a result, no kernel induced jitter is present in the packet timestamp. We will compare this card with a conventional one in which the timestamping is provided at kernel or user level.

Besides, the high-precision card uses a shared memory as a mean to relay packets to the analysis program running in user space, in such a way that interrupts and packet copies are avoided.

First, we measure the OWD of 2 bursts composed by 12,000 UDP packets of 46 bytes and 10 ms interpacket time from chania to magdeburg nodes. One burst uses conventional (eth0) card and the other uses the high-precision (dag) card. Both bursts find the same network conditions because we send and receive alternatively packets for each burst. The histogram of OWD is shown in Fig. 2. We observe different distribution measured with eth0 or dag, and, what is more important, the mean for both curves is also different (42.19 and 42.45 ms for dag and eth0 respectively). This clearly demonstrates the different

OWD results between both interfaces dag and eth0, in the order of hundreds of microseconds.

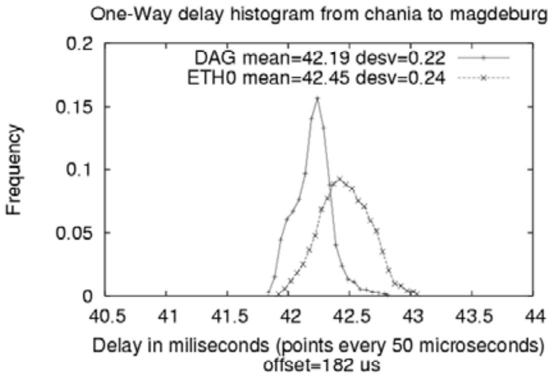

Fig. 2. OWD with dag and eth0, consecutive experiments

The explanation can be found in Fig. 3 that presents the packet transversal though Linux kernel and the points where the packet is timestamped: t1' for transmission (inserted as a new header field in the packet) and t2' for reception. Timestamp t1 is marked when the first bit of the packet is put into the network and t2 is marked when the first bit is received. These both t1 and t2 would be the ideal, however we get different t1' and t2'. For dag card, the timestamping is done at card level, much near to real t1 and t2. However, for eth0 card t1' is done by the application in the sending process and t2' by the kernel in the reception process. The extra OWD offset between dag and eth0 is due to the variability introduced by the kernel and user application, increasing the OWD mean in the order of 182 μs and enlarging the distribution width.

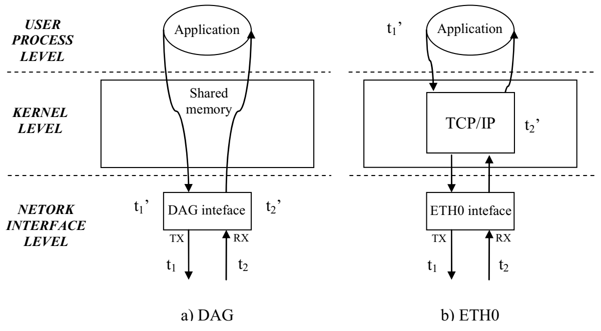

Fig. 3. Timestamping with dag and eth0

Although with the eth0 card the PC is GPS synchronized, the timestamping is done far away from the network cable. In Fig. 2, the histogram of OWD for eth0 card is wider and less concentrated than for dag card.

The effect of late timestamping in eth0 makes the measurement packet size dependent. The OWD offset between dag and eth0 cards is shown in Fig. 4 for different packet sizes. Increasing offset is observed for larger packet sizes because of the extra cost of larger memory copies between network card and kernel, and between kernel and user space.

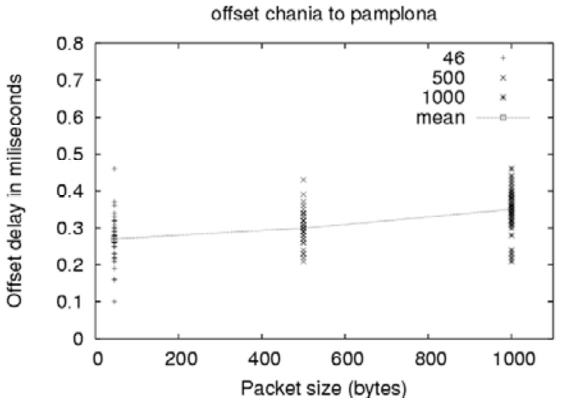

Fig. 4. OWD offset between dag and eth0 for different packet sizes

As a result, it is clear the importance of a high-precision hardware platform for OWD measurements, not only obtaining a real mean value for OWD, but getting a more realistic sample distribution. We are also avoiding secondary effects as memory copies and randomness of a general purpose operating system in packet processing.

3 One-way delay model and available bandwidth estimator

Several papers have studied possible models of OWD, some of them in controlled environments [12], others for limited topologies [9][1] and almost always in short periods of time [1]. An interesting and novel effect in OWD distribution is the sending rate: how many packets per second of certain size we are sending to the destination.

It is predictable that OWD changes with sending rate, the question is how. In Fig. 5 we can see the OWD distribution for different sending rates from ericsson to colbud nodes. In this experiment, we send bursts composed by 12,000 UDP packets of 46 bytes and variable interpacket time. The OWD of those packets is modified by links capacity, buffer sizes and competitive traffic in the network. The fitting curves have been made using maximum likelihood estimation (MLE) [13]. At low speed, the OWD is very concentrated around the mean 17.04ms, following a gamma distribution as shown in Fig. 5.a. As sending rate increases, in Fig. 5.b, OWD distribution is modeled as Pareto, with a longer tail that increases mean and variance of the distribution.

As we get closer to the available bandwidth, the distribution is wider, indicating that the router of the bottleneck is near to congestion. There are more packets that find more delay in the queue of this router, and the loss rate is increasing as shown in Table 1. This is because we are stressing the network as we will see later because of the little packet size (46 bytes at IP level). In Fig. 5.c we find that the distribution of OWD can be approximated by a lognormal, or even by a uniform distribution. The variation in OWD is caused by the queueing delay component.

The parameters of the fitted distributions in Fig. 5 are the following:

a) Gamma distribution: a=78878.2, b=0.0002
b) Pareto: k=0.0816, σ=0.1164
c) Lognormal: μ=2.8447, σ=0.0088

Table 1. Packet losses for different sending rates.

Sending rate (Mbps)	Packets losses	Losses percentage (%)
2.1	243	2
2.6	149	1,2
3.5	719	6

These distributions are concentrated in the order of few hundreds of microseconds, demonstrating the necessity of high-precision in the measurement system. Our measurement error is in the order of few microseconds or less [10], so the measurement error does not affect the distributions under study.

As soon as we get all the available bandwidth between source and destination, the router of the bottleneck will be in congestion. This situation is reached around 5 Mbps (with 46 bytes packet size at IP level plus 14 bytes of Ethernet header, 10,416 packets/sec) and the times series of OWD are shown in Fig. 6. The first part until 490ms shows a slope behavior related with the queuing delay in the congested router at 10,416 packets/sec. A packet is queued in the router until all the previous queued packets were transmitted. Each packet will find more packets in queue so its queuing delay will be larger, showing that linear increasing of OWD in Fig. 6. In the last part, from 490 to 610ms, the OWD is constant, keeping 5 Mbps (10,416 packets/sec) input data rate. In this part, we can see how the congested router can not queue more packets, because its buffer is full. The router starts losing packets because it receives packets faster than it can send them to the next router.

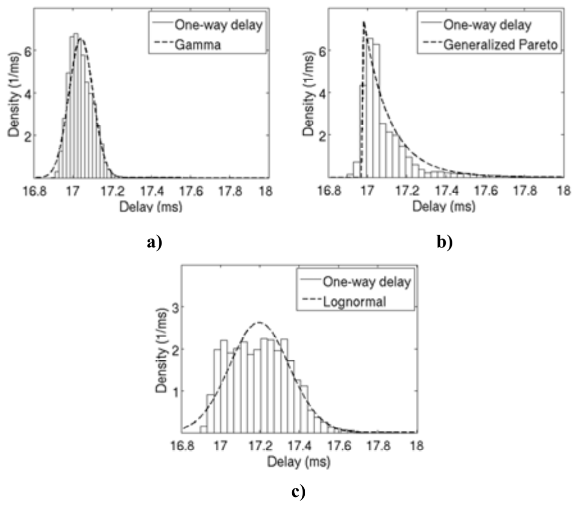

Fig. 5. OWD distribution adjustment for a) 2.1 Mbps b) 2.6 Mbps and c) 3.5 Mbps

Here the question is if the congestion is in the output link (bandwidth bottleneck, 5Mbps would be too low) or inside the router CPU (packets/sec limitation). If we send two bursts, one with 46 bytes sized packets and the other with 100 bytes sized packets, we obtain similar situation of congestion when the burst with 100 bytes sized packets is sent at double speed than for 46 bytes sized packets. For both, the number of packets per second would be similar. So, for this case, we have checked that the congestion is caused by the packets per second (packet rate) instead of the bit rate. The router can not process as many packets as it receives and then the congestion is located in the backplane of the router. It is not a traffic shaper because that limitation in packets per second is shared between all flows in any direction through that router. This means that there is not limitation per flow, the limitation is for the sum of the flows through the router coming from any interface. An easy experiment was enough to confirm that assumption. This makes clear that the backplane/CPU is shared for all input interfaces of the router.

In Fig. 7 we can see the profile of packet losses for the experiment in Fig. 6. The router starts losing packets as soon as it gets more packets in queue. Finally, an increase of packet losses is observed when the buffer of the router is full (last part of Fig. 7).

The slope of the increasing OWD over time in Fig. 6 can be used to estimate the available bandwidth that a user can see for a packet size of 46 bytes. We have the following parameters, choosing one point of the slope:

- C_{in}: sending rate. 5Mbps.
- T: time interval between the first and the selected point of the slope. For our case it is 308ms (base of the triangle formed by the sloped line).
- D: extra delay of the selected point, compared with the delay before the congestion. For our case 106ms (height of the triangle formed by the sloped line).

From these parameters we can infer the value of the bottleneck bandwidth taking into account the extra delay assumed by the selected packet of the slope (for a packet size of 46 bytes). The available bandwidth (bottleneck bandwidth), *Ca*, will be:

$$C_a = \frac{C_{in}\,T}{T+D} = \frac{C_{in}}{1+\frac{D}{T}} \qquad \textbf{(1)}$$

The resulting available bandwidth is 3.72 Mbps for Fig. 6 and for 46 bytes packet size (exactly 7,750 packets/sec, our real router CPU limitation). The advantages of a high-precision measurement platform are present in this figure where for 8,250 packets sent, the OWD is almost a thin line without noise at microseconds scale. In the expression for the available bandwidth *Ca, D/T* is the slope of the line. This slope can be obtained easily from Fig. 6.

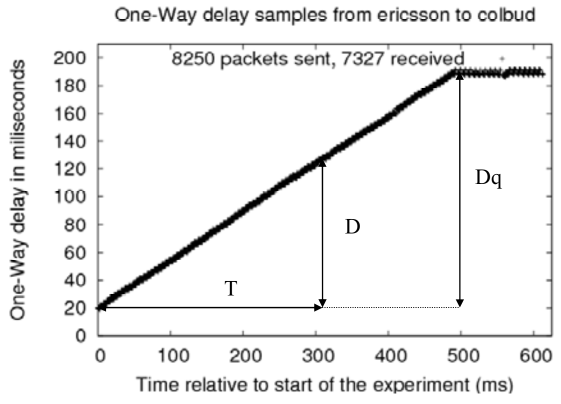

Fig. 6. Time series of OWD.

Another interesting measurement could be the queue size of the congested router. For that case, we would need to find the point where the sloped line could not grow any more because new incoming packets were being lost. If *Dq* would be the extra delay of last no lost packet compared with the delay before the congestion, the queue size, *Qsize*, would be:

$$Q_{size} = C_{in}\ D_q \qquad \textbf{(2)}$$

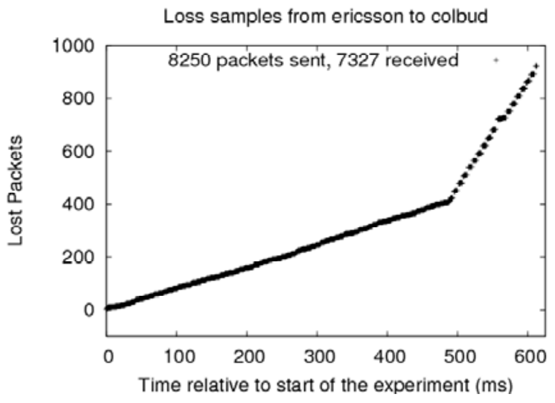

Fig. 7. Time series of packet losses for Fig. 6.

For Fig. 6, we can estimate the queue size with the last packet before the router start to lose packets because the buffer is full and can not queue more packets. So, we get a queue size of 107 KB. In this case, *Dq* would be the height of the triangle formed by the sloped line, and its value is 172 ms in Fig. 6.

4 Long-range evolution of one-way delay

We have been characterizing OWD on short time intervals where that delay parameter can be considered stationary. Another interesting point of view will be to consider extensive periods of time, and the evolution of OWD over those intervals.

In Fig. 8 we show the OWD from magdeburg, pamplona, paris and colbud to jerusalem node over a period of time of approximately five months with two measurements per day. In each measurement, we send bursts of 300 UDP packets of 46 bytes and 1s interpacket time. For most part of the time, the OWD has low variability (less than 5% considering the mean in the interval) and this kind of variability is located in short time intervals as modeled in previous section. Appreciable changes in OWD are basically due to route changes as can be seen around 28 February, 2006 in Fig. 8. These route changes do not follow a specific pattern and they are not usual. The OWD profile is similar for all these four source nodes because, for this case, the route changes are located nearer to destination node, so some final hops are shared for the paths from any source. Any change in those final hops affects similarly to the OWD from any source node. Looking at OWD between any other pair of nodes, it is hard to find situations in which several sources have path hops in common, so the OWD behavior is less related.

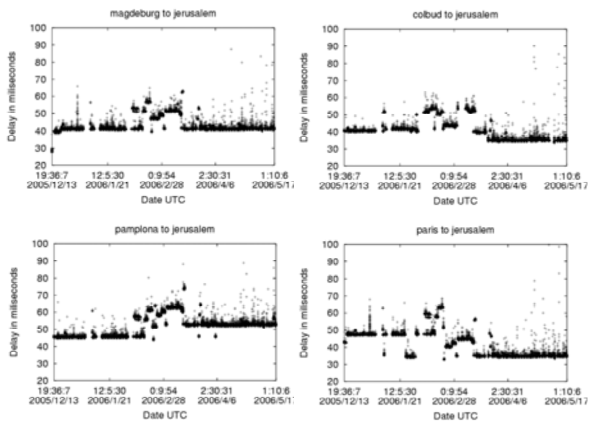

Fig. 8. OWD time series over 5 months.

5 Conclusions

One-way delay is a fundamental parameter in network characterization and quality of service provisioning. The measurement of one-way delay has its own challenges, which can be solved with a high-resolution measurement platform as ETOMIC. The effects of specific network cards (Endace DAG) and conventional network cards have been revised, demonstrating the benefits of the first ones, even in the measurement of mean values for OWD.

The measurements have been adjusted to known distributions (gamma, pareto and lognormal), and the dependency with congestion situations has given interest results. Another application of OWD is as available bandwidth estimator when we congest the path to destination. In that case, the queuing delay could be used to make this estimation. A very low speed path has been found in the OWD study, limited by the processing power of a router and not by links capacity.

Finally, the dependency of OWD with route changes over extensive periods of time has been used to infer correlation in route paths from different sources.

References

[1] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijtervaal, and P. Van Mieghem. Analysis of end-to-end delay measurements in Internet. Passive and Active Measurement Conference (PAM 2002), March 2002.

[2] I. Csabai, P. Hága, P. Mátray, G. Simon, J. Stéger and G. Vattay. Results of Large-Scale Queueing Delay Tomography Performed in the ETOMIC Infrastructure. 9th IEEE Globecom 2006, 28-29, Barcelona, Spain, April 2006.

[3] N. Hu and P. Steenkiste Quantifying Internet End-to-End Route Similarity. Passive and Active Measurement Conference (PAM 2006), March 2006.

[4] R. Wang, G. Pau, K. Yamada, M. Y. Sanadidi and M. Gerla. TCP Startup Performance in Large Bandwidth Delay Networks. INFOCOM 2004, Hong Kong, March 2004.

[5] M. Narbutt and L. Murphy. VoIP Playout Buffer Adjustment Using Adaptive Estimation of Network Delays. 18th Int'l Teletraffic Congress (ITC-18), Elsevier, pp. 1171–1180, 2003.

[6] M. Gerla, S.S. Lee, and G. Pau. TCP Westwood Simulation Studies in Multiple-Path Cases. International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), San Diego, CA, USA, July, 2002.

[7] M. Tsuru, T. Takine and Yuji Oie. Estimation of Clock offset from one-way delay measurement on asymmetric paths. Symposium on Applications and the Internet (SAINT 2002), Los Alamitos, CA, USA, 2002.

[8] T. Iwama, A. Kaneko, A. Machizawa and H. Toriyama. Real-Time Measurement of One-Way Delay in the Internet Environment. The Institute of Electronics, Information and Communication Engineers, Vol.2004, No.B-16-1, pp.386, 2004.

[9] N. M. Piratla, A. P. Jayasumana and H. Smith. Overcoming the Effects of Correlation in Delay Measurements using Inter-Packet Gaps. Proceedings 12th IEEE International Conference on Networks (ICON 2004), pp.233-238, Singapore, Nov 2004.

[10] E. Magana, D. Morato, M. Izal, J. Aracil, F. Naranjo, F. Astiz, U. Alonso, I. Csabai, P. Haga, G. Simon, J. Steger and G. Batía. The European Traffic Observatory Measurement Infraestructure (ETOMIC). Proceedings of IEEE International Workshop on IP Operations & Management (IPOM 2004), pp.165-169, Beijing, China, October 2004.

[11] European Traffic Observatory Measurement Infraestructure (ETOMIC) web page: http://www.etomic.org

[12] D. Constantinescu and A. Popescu. Modeling of One-Way Transit Time in IP Routers. Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services AICT-ICIW '06, February 2006.

[13] J. Myung. Tutorial on maximum likelihood estimation. Journal of Mathematical Psychology archive, Volume 47, Issue 1, pp.90-100, February 2003.