Exploiting the IPID field to infer network path and end-system characteristics *

Weifeng Chen[1],Yong Huang[2], Bruno F. Ribeiro[1], Kyoungwon Suh[1], Honggang Zhang[1], Edmundo de Souza e Silva[3], Jim Kurose[1], Don Towsley[1]

[1] Department of Computer Sciences
University of Massachusetts
Amherst, MA 01003
{*chenwf, ribeiro, kwsuh, honggang, kurose, towsley*}*@cs.umass.edu*

[2]Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003
yhuang@ecs.umass.edu

[3]COPPE and Computer Science Department
Federal University of Rio de Janeiro
Rio de Janeiro, RJ 21945-970 Brazil
edmundo@land.ufrj.br

Technical Report 2004-92

Abstract

In both active and passive network Internet measurements, the IP packet has a number of important header fields that have played key roles in past measurement efforts, e.g., IP source/destination address, protocol, TTL, port, and sequence number/acknowledgment. The 16-bit identification field (IPID) has only recently been studied to determine what information it might yield for network measurement and performance characterization purposes. We explore several new uses of the IPID field, including how it can be used to infer: (a) the amount of internal (local) traffic generated by a server; (b) the number of servers in a large-scale, load-balanced server complex and; (c) the difference between one-way delays of two machines to a target computer. We illustrate and validate the use of these techniques through empirical measurement studies.

Key words: IPID field, one-way delay difference, traffic activity, load-balanced server counting, estimation

*This research has been supported in part by the NSF under grant awards EIA-0131886, EIA-0080119, ANI-0085848, ANI-0070067, ANI-0319871, and ANI-0240487 and by the ARO under DAAD19-01-1-0610 and by CAPES (Brazil). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

In both active and passive network Internet measurements, the fundamental unit of measurement - the IP packet - includes a number of important header fields that have played key roles in past measurement efforts: IP source/destination address, protocol, TTL, port, and sequence number/acknowledgment. The 16-bit identification field (referred to here as the IPID field) has only recently been used to determine what information it might yield for network measurement and performance characterization purposes [1, 4, 7, 9, 5]. In this paper, we explore several new uses of the IPID field, including how it can be used to infer: (a) the amount of internal (local) traffic generated by a server; (b) the number of servers in a large-scale, load-balanced server complex and; (c) the difference between one-way delays of two machines to a target computer. We illustrate and validate the use of these techniques through empirical measurement studies.

The remainder of this paper is structured as follows. In the following section we classify and discuss past work that has examined the use of the IPID field, and place our current work in this context. In Section 3, we describe a technique to infer the amount of a host's traffic that remains internal to its local network, and the complement amount of traffic that passes through a measured gateway link. In Section 4, we describe a technique to identify the number of load-balancing servers behind a single IP address. In Section 5, we introduce a technique to infer the difference between one-way delays. Section 6 concludes this paper with a discussion of future work.

2 Uses of the IPID field

We begin with a brief description of the IPID field and the generation of IPID values, and then classify previous measurement work, as well as our current efforts, into three categories based on their use of the IPID field.

The 16-bit IPID field carries a copy of the current value of a counter in a host's IP stack. Many commercial operating systems (including various versions of Windows and Linux versions 2.2 and earlier) implement this counter as a global counter. That is, the host maintains a *single* IPID counter that is incremented (modulo 2^{16}) whenever a new IP packet is generated and sent. Other operating systems implement the IPID counter as a per-flow counter (as is done in the current version of Linux), as a random number, or as a constant, e.g., with a value of 0 ([1]).

2.1 Global IPID

In this paper, we only consider hosts that use a single *global* counter to determine the IPID value in a packet. To infer whether a host implements a global IPID counter, we probe the host from two different machines by sending http requests. IPID values in the packets returned from the host can be obtained by running tcpdump on the two probing machines separately. If the host uses a global IPID counter, these replying IPID values will belong to a unique sequence. By synchronizing the two probing machines, we are able to compare the replying IPID values, as presented in Figure 1. This figure clearly shows that the IPID values of the packets returned to the two probing machines belong to a unique sequence, and consequently, we can infer that this host uses a global IPID counter.

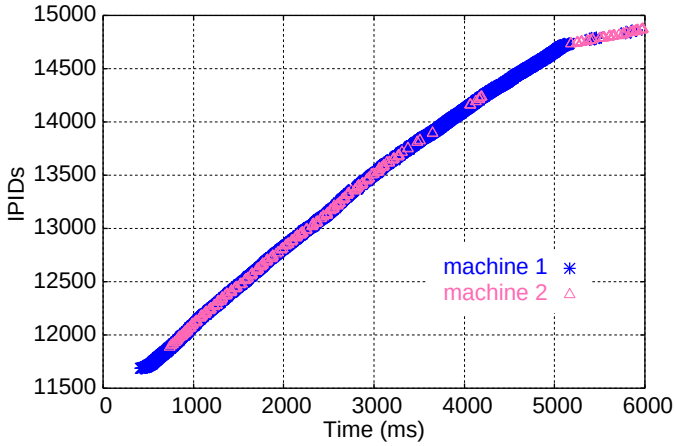

Figure 1: IPID values returned from a global-IPID host.

Instead, if the host does not implement a global IPID counter we obtain a different result. Figure 2 shows the result of a host implementing the IPID counter as a per-flow counter. The IPID values of the replying packets to the two different probing machines consist of two independent sequences, each corresponding to one probing machine. Note that the slopes of these two sequences are different because of the different speeds of the probing packets sent from the probing machines.

Using this process, we probed the web-servers of the top 50 companies ranked by *Fortune* magazine [**?**] and found that 18 (36%) of them have a global IPID counter. Among the top 101 web sites ranked by *PC Magazine* [**?**], 40 of these web sites were found to have a global IPID counter.

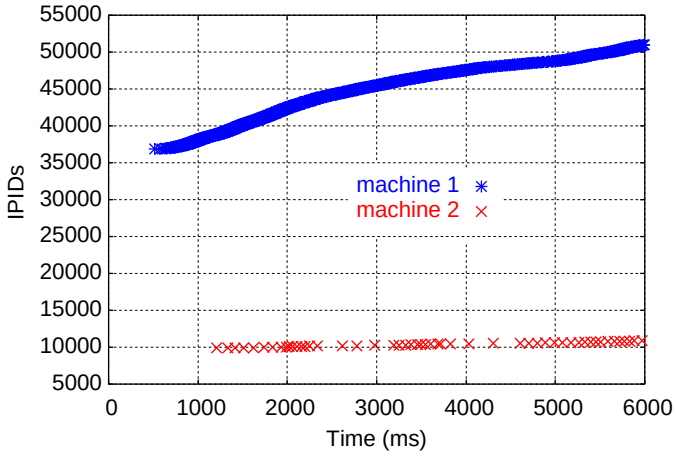

Figure 2: IPID values returned from a non-global-IPID host.

2.2 Classifications of using IPID fields

We can broadly classify previous efforts, as well as our current efforts, using IPID sequences into three categories:

Application 1: Measuring traffic activity. Suppose that we observe a subset of the packets generated by a server, and consider the $(i-1)$-st and i-th observed packets. Let $T(i)$ denote the timestamp of the i-th packet and ΔIPID(i) the difference between the IPID values of the $(i-1)$-st and i-th packets[1]. In this case, $\sum_{i=1}^{n} \Delta\text{IPID}(i)$ represents the number of packets sent by this server in the interval $(T(1), T(n))$. The use of IPID values to infer the total amount of outgoing server traffic is noted in [5]. We additionally note that, for stub networks with a single outbound connection, this also allows us to infer the relative amount of traffic sent to destinations within the network, and to destinations outside of the network. From this single measurement point, we can thus infer one aspect (local/remote) of the spatial distribution of traffic destinations. We consider this approach in Section 3.

Application 2: Clustering of sources. These applications make use of the fact that different hosts have independent (and thus generally different) IPID values, and that IPID values are incremented for each outgoing IP packet sent by a host. We denote the difference in the values of the IPID field of two successively observed packets as ΔIPID. Thus, if we observe two packets generated by the same host within a "short" interval of time, we will generally observe a small ΔIPID value. By identifying small ΔIPID values among

[1] We may obtain a negative value for ΔIPID(i) due to wrap-around. We address this problem later in this paper.

a set of IP packets that were generated within a short interval of time from multiple sources, it is then often possible to identify packets coming from the same source. It is important to note that IPID-based source-identification is thus possible without actually examining the source IP address, which itself may have been aliased. Router alias detection [9], host alias detection and load-balanced multiplexed-server counting [5], and NATed host counting [1] all exploit this observation. Our work in Section 4 builds on initial suggestions in [5] by considering a specific algorithm for identifying the number of servers behind a load-balancer using only observed IPID values.

Application 3: Identifying packet loss, duplication and arrival order. Since a packet generated later in time by a host will carry a larger IPID (modulo 2^{16}) than a packet generated earlier in time by that host, it is possible (after solving the wrap-around problem) to determine the order in which packets are generated by a host. Previous work on detecting packet reordering and loss between a probing host and a router [7] and duplicate packet-detection and re-ordering at a passive monitor [6] exploit this observation. In Section 5, we use the fact that the IPID value of a packet generated in response to a received packet indicates the order in which received packets arrived to develop a new approach for inferring the absolute differences in one-way delays between a set of machines and a target host.

Several technical challenges must be met when using IPIDs in measurement studies. The most important regards wrap-around between two consecutively observed packets from the same source. Correction is easy if we know that only a single wrap-around has occurred. With active probing techniques (where the measurement point sends active probes to a host and observes the IPID of the returned packet), multiple wrap-arounds can be avoided by choosing an appropriate probing interval. In a passive monitoring framework, a more sophisticated method is needed to deal with multiple wrap-arounds, as discussed in the following section.

3 Outbound traffic from a server

In this section, we present a simple technique for measuring the outbound traffic from a server (i.e., the number of packets sent by a server) by passively observing the IPIDs of packets generated by that server at a gateway. The use of active probes to infer the total amount of outgoing server traffic was suggested in [5]. Passive measurement avoids the overhead of active probing, and the attention that active probing may bring (indeed, several of our active probing experiments resulted in our measurement machines being black-

listed at a number of sites!). We will see shortly, however that it is valuable to augment passive probing by occasionally sending active probes in order to handle IPID wrap-around.

Suppose that, at a gateway, we observe a subset of the packets generated by a server, and consider the $(i-1)$-st and i-th packets observed. Let $T(i)$ denote the timestamp of i-th packet and ΔIPID(i) denote the difference between the IPID values of the $(i-1)$-st and i-th packets. In this case, $\sum_{i=1}^{n} \Delta\text{IPID}(i)$ represents the total number of packets sent by this server during the interval $(T(1), T(n))$. Furthermore, if the server accesses the larger Internet only through this gateway, we know that all other packets generated between the $(i-1)$-st and i-th observed packets must have been sent to destinations within the network - providing an easy means to determine the amount of network-internal traffic being generated by a server.

We performed experiments on several popular web servers in our campus. One result is plotted in Figure 3. Since we could not instrument the server, we validated our measurements using periodic active probes. As shown in Figure 3, this result is consistent with that obtained using active probes. Figure 4 shows the amount of network-internal traffic from the server as determined by our proposed passive approach.

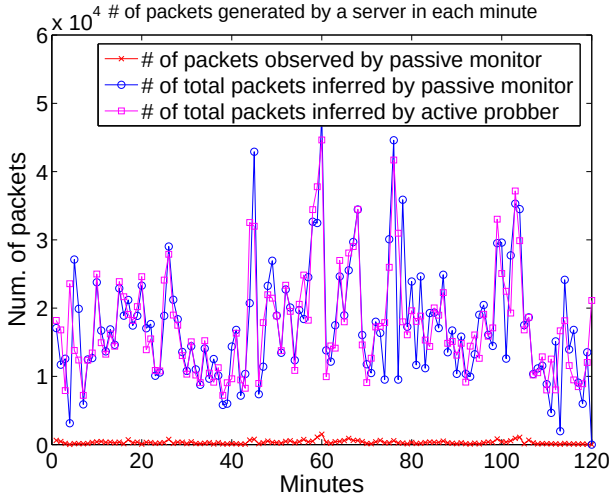

Figure 3: Comparison between passively measured and actively measured outbound traffic from a server

With a purely passive approach to measuring server activity, it can be difficult to detect IPID wrap-around if the amount of traffic observed at the monitor point is very small compared to the amount of network-internal traffic generated by the server. Indeed, in our experiments, we observed popular web servers in our campus that did not serve clients outside of our campus for long periods of time. To solve this problem, we adopt a *hybrid* approach in which adaptively-activated active measurement is used to supplement passive measurement. Specifically, we use an Exponential Weighted Moving Average (EWMA)

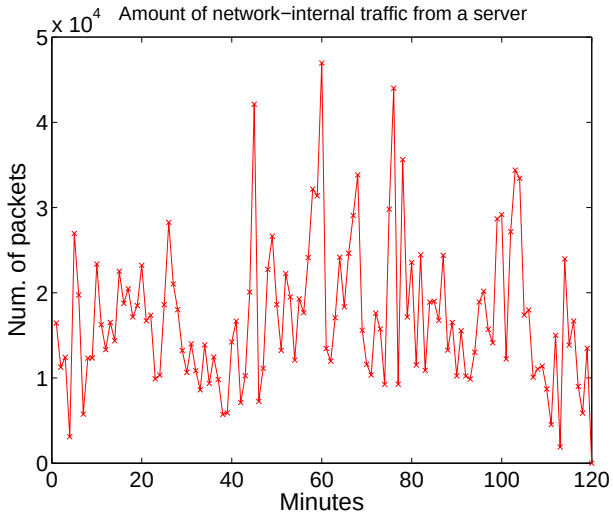

Figure 4: Amount of network-internal traffic from a server

to estimate the rate of IPID increase. Using this estimate, we can then estimate the next IPID wrap-around time, T^*(msec), and start a timer with that value. Whenever we observe a new packet before this timer expires, we reset the timer based on the current estimated IPID rate. If the timer expires, we launch an active probe and reset the timer. We are currently performing additional work to evaluate this hybrid approach.

4 Inferring number of load-balancing servers

If each load-balancing server behind a single IP address has an independent global IPID counter, packets generated by one server have a sequence of IPID values that differs from those generated by a different server. As discussed below, using these observed IPID values, we can classify the packets into distinct sequences, with the number of distinct sequences being an estimate for the number of servers. Figure 5 shows the observed IPID values of the packets generated from a large commercial web server in response to the 5000 probing packets we sent to the server.

We next describe an algorithm to classify the packet IPID sequences. Let $\{I_1, I_2, \ldots, I_{5000}\}$ be the set of IPIDs shown in Figure 5 and $\mathcal{S}$ the set of distinct sequences. Initially, $\mathcal{S} = \emptyset$. The first IPID I_1 is appended to sequence S_1 (namely, $S_1 = \{I_1\}$) and $\mathcal{S} = \mathcal{S} \cup \{S_1\}$. For each following IPID I_j $(2 \leq j \leq 5000)$, I_j is compared to the tail element of all sequences in $\mathcal{S}$. If the difference between I_j and all of the tail elements is larger than a threshold T, a new sequence $S_{|\mathcal{S}|+1}$ is created and $S_{|\mathcal{S}|+1} = \{I_j\}$. Additionally, $\mathcal{S} = \mathcal{S} \cup \{S_{|\mathcal{S}|+1}\}$. Otherwise, I_j is appended to the sequence whose tail element has the smallest difference

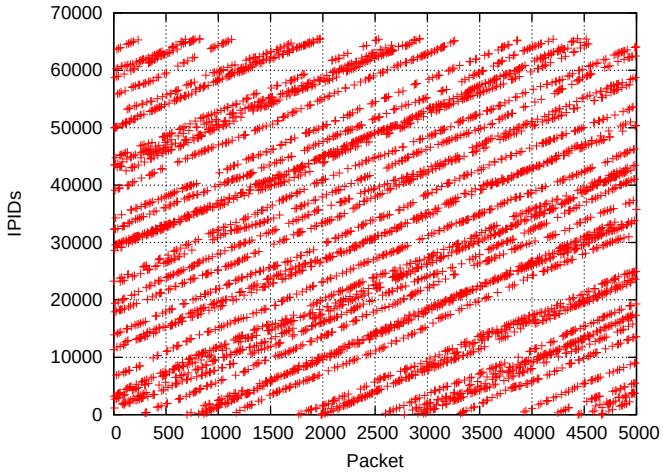

Figure 5: IPIDs of the packets returned from the web server

with I_j. Given T, the algorithm returns the number of sequences, i.e., $|\mathcal{S}|$, and the corresponding sequence sizes, i.e., the number of packets in each sequence.

Our algorithm will return a different number of sequences of different sizes for different values of T. Ideally, the sequence sizes should be equal, with probing packets being forwarded at equal rates to the servers. In practice, however, these rates are close but not equal, due to the mixing of probing packets with other traffic. For this experiment the interval between two successive probing packets was set to 3ms to minimize these effects.

To determine an appropriate T, we introduce two parameters: load balancing factor (LBF) and coefficient of variation (CV) of sequence size. We define LBF_T as $\text{LBF}_T = P_{\min}/P_{\max}$, where $P_{\min}$ (resp. $P_{\max}$) is the number of packets in the smallest (resp. largest) sequence returned by the algorithm with a given T. Ideally, for a well balanced server, an appropriate T should produce a LBF_T very close to 1. The second parameter, CV_T, is defined as $\text{CV}_T = \sigma_T/\mu_T$, where σ_T and μ_T are the standard deviation and the mean of the sequence sizes respectively for a given T. Intuitively, an appropriate T results in a small CV_T.

Figure 6 shows LBF_T, CV_T and the number of sequences as a function of T. A T is *appropriate* when LBF_T achieves the maximum and CV_T achieves the minimum. The figure indicates that a $T \approx 4000$ is appropriate, resulting in 30 sequences. That is, we estimate that the web server has 30 load-balancing servers. Table 1 shows the numbers of packets in these 30 sequences.

Based on this value of T, the algorithm described above divides the IPID values shown in Figure 5 into 30 sequences. We plot these 30 sequences in different colors in Figure 7 where wraparounds of each

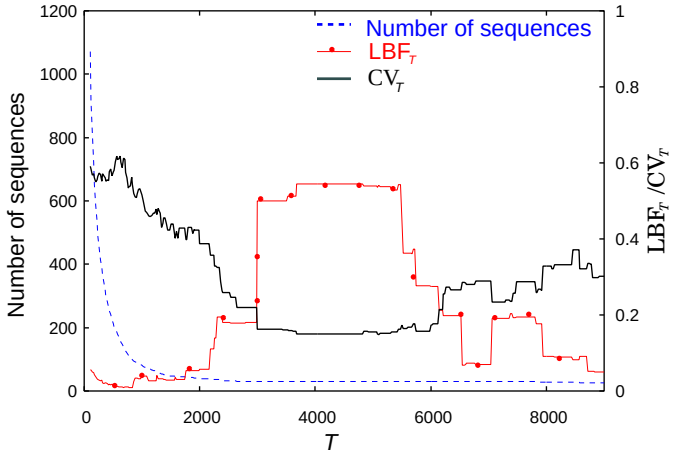

Figure 6: Number of sequences, LBF_T and CV_T vs. T

162, 165, 180, 155, 156, 131, 188, 136, 178, 186
170, 162, 167, 228, 208, 193, 158, 177, 144, 169
145, 145, 168, 192, 177, 124, 173, 129, 202, 132

Table 1: Number of packets in classified sequences.

sequence were removed by adding 64K to the values so that every sequence is always monotonically increasing. We observe that the slopes of all of these 30 sequences are almost the same, which suggests that each load-balancing server receives a comparative number of probing packets.

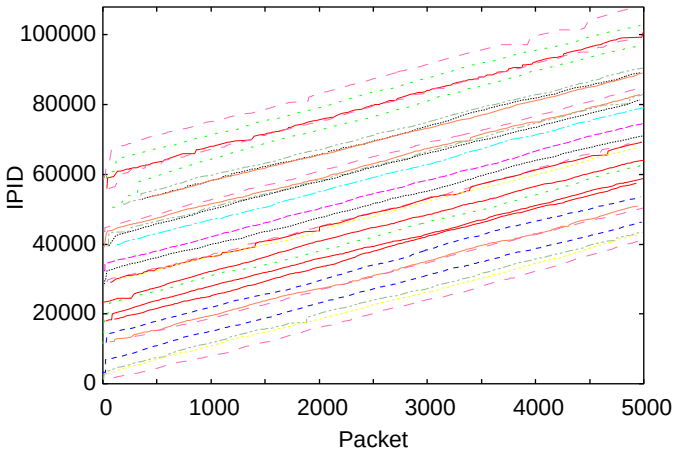

Figure 7: IPID sequences of the load-balancing servers of a commercial web-server.

5 Inferring one-way delay differences

In this section, we present a simple technique that uses the IPID field to infer the differences in one-way delays from a set of GPS-synchronized probing sources to an "IPID order capable" destination target. By "IPID order capable" we mean that the destination has a global IPID counter and that wrap-arounds can be detected. Importantly, we do *not* require the destination to be GPS-synchronized. Such delay differences can be used to infer shared path segments using recently developed network tomograpic techniques [8, 2, 3]. In addition, if one of the sources is able to determine (or accurately estimate) the absolute magnitude of its one-way delay to the destination, then all other nodes can determine the absolute values of their one-way delays as well. Knowledge of one-way delay can be valuable in many circumstances.

[8] presents a methodology for estimating one-way delay differences from non GPS-synchronized (or coarsely synchronized) sources to common destinations using a semi-randomized probing strategy and the packet arrival ordering collected at destinations. Given GPS-synchronized source clocks, the deterministic probing strategy we study is considerably simpler. As in [8], the key idea is for sources to send probes (e.g., ICMP echo packets) to a remote host, and use the observed arrival ordering to infer path characteristics. Our approach differs from [8] in the way we obtain arrival order information. In [8] all destination machines must be instrumented. Using IPID, we are able to obtain the packet arrival orders without instrumenting any destination machine. In the following, we consider only two source nodes; the approach easily generalizes to the case of additional source nodes.

Our goal is to infer the one-way delay difference from two GPS-synchronized sources A and B to a destination D, i.e., the difference between path delays d_{AD} and d_{BD}. Consider two packets p_1 and p_2 sent from A and B to D at the same time. If p_1 arrives before p_2, the IPID, I_1, of the packet returned by D in response to p_1 will be smaller (modulo 2^{16}) than the IPID, I_2, of the packet responding to p_2.

We exploit this ordering of returned IPID values as follows. As illustrated in Figure 8, A and B begin simultaneously probing D using different probing intervals δ_A and δ_B, respectively. The n_A-th packet sent from A arrives at D between the $(n_B - 1)$-st packet and the n_B-th packet sent from B. If the delay does not change significantly during the measurement interval, we have:

$$\begin{aligned} & d_{BD} + (n_B - 1)\delta_B \leq d_{AD} + n_A\delta_A \leq d_{BD} + n_B\delta_B \\ \Rightarrow \quad & (n_B - 1)\delta_B - n_A\delta_A \leq d_{AD} - d_{BD} \leq n_B\delta_B - n_A\delta_A \end{aligned}$$

Note that the difference between the upper- and lower-bounds depends on δ_B. Thus by reducing δ_B, we can

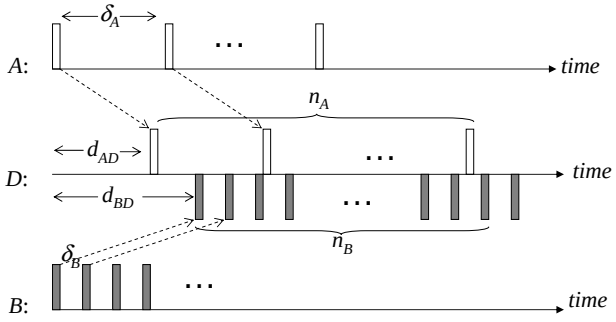

Figure 8: Arriving orders of packets

improve the accuracy of the inferred delay difference $d_{AD} - d_{BD}$. We conjecture that we can extend these techniques to handle the case of varying delays during the measurement interval as well.

We have validated the approach in a simple test scenario. In our experiments we send *ICMP echo* packets from source machine A (at Unifacs, a university in Brazil) and B (a machine at the University of Minnesota) to a destination machine D at the University of Massachusetts. Machine A sends one packet per second and machine B sends one packet every 3ms. Our measurements indicate that the IPID-inferred delay difference, namely, $d_{AD} - d_{BD}$, is around 230ms. We also send probes from A and B to a GPS-equipped machine, D', at the University of Massachusetts that was close to D. Based on the recorded data on D', we can measure $d_{AD} - d_{BD}$. Figure 9 shows the difference of the measured values and the IPID-inferred values as a function of time. From the figure, one can see that the inferred values are very close to the measured values. Furthermore, it should be noticed that most of the differences are within 3ms for $\delta_B = 3$ms. Figure 10 shows the relative error of the IPID-inferred values (I) to the measured values (M), where the relative error is defined as $(I - M)/M$.

Table 2: Statistical results for several one-way delay inference experiments.

Experiment ID	Time of day (EST)[1]	Mean of error[2] (ms)	Standard deviation of error, σ (ms)	Mean of measured values, $\bar{M}$ (ms)	$\sigma/\bar{M}$
1	2004-10-14 11pm	2.02	9.73	232.17	0.042
2	2005-01-14 9pm	3.37	18.57	354.49	0.052
3	2005-01-15 6am	3.53	22.27	361.14	0.062
4	2005-01-17 2pm	11.44	23.16	356.24	0.065
5	2005-01-17 5pm	9.53	20.38	361.11	0.056

[1]:The beginning time when an experiment was conducted.
[2]: error=$|I - M|$.

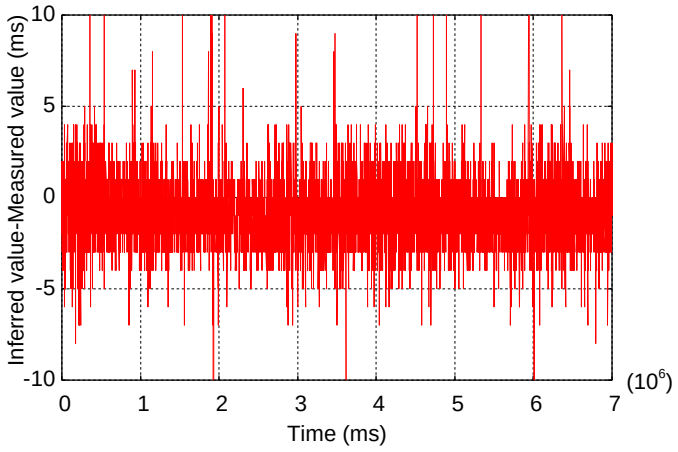

Figure 9: Difference of IPID inferred $d_{AD} - d_{BD}$ and measured $d_{AD} - d_{BD}$.

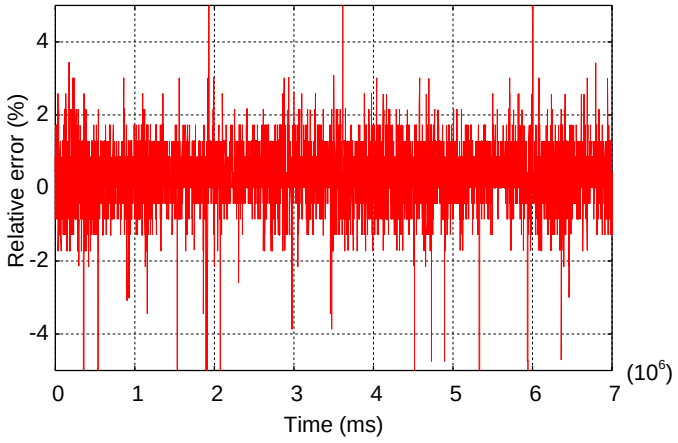

Figure 10: Difference of IPID inferred $d_{AD} - d_{BD}$ and measured $d_{AD} - d_{BD}$.

Statistical results from the experiment shown in Figure 9 are presented in the first line of Table 2, which includes the results of four other experiments. This table contains a broad set of experimental scenarios. We run experiment 2 and 3 with $\delta_B = 4$ms (Figure 8). Experiments 4 and 5 also used $\delta_B = 4$ms but during the busiest traffic hours. From the results we can see that during the busiest hours the one-way delay inference error becomes larger. This effect is due to the higher inter-packet jitter in our measures. Figure 11 depicts how higher jitters affect inter-arrival times of A and B packets at D (Figure 8). A packet from A is more likely to arrive between packets B_1 and B_2 than between packets B_2 and B_3. Thus the probability that packets from A will arrive between two given consecutive B packets increases with the jitter delay value. This problem could be ameliorated by sending two A packets with smaller sending intervals, i.e., a pair of A packets. These pairs can eliminate samples where both packets in a pair arrived between two identical B

packets (Figure 11(b)). Samples where two packets of a pair interleaved with two B packets (Figure 11(c)) will produce more accurate inferences.

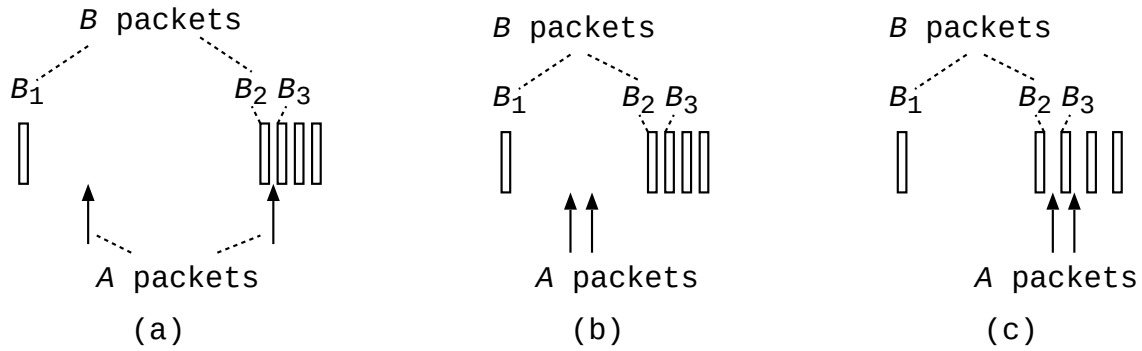

Figure 11: Jitter examples.

6 Conclusions

In this paper, we explored several uses of the IPID field for inferring network path and end-system characteristics. We classified previous IPID-related measurement efforts into three general application areas, and showed that, by using the IPID field, it is possible to infer: (a) the amount of internal (local) traffic generated by a server; (b) the number of servers in a large-scale, load-balanced server complex and; (c) the difference between one-way delays of two machines to a target computer. We illustrated and validated the use of these techniques through empirical measurement studies.

As with previous measurement techniques exploiting other packet header fields, header fields (such as the TTL and IPID fields) can be exploited for measurement purposes not initially envisioned in the design of IP. We hope that our work will add to the toolkit of network measurement techniques. We also hope that future measurement studies can build on this work, and that additional clever ways will be found to exploit the IPID field for measurement purposes.

References

[1] S. Bellovin. A technique for counting NATed hosts. In *Proc. ACM Internet Measurement Workshop(IMW)*, November 2002.

[2] M. Coates and R. Nowak. Network tomography for internal delay estimation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2001.

[3] F. Lo Presti, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. *IEEE/ACM Trans. Networking*, 10:761–775, 2002.

[4] A. Hussain, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks. In *Proc. ACM SIGCOMM*, August 2003.

[5] Insecure.org. Idle scanning and related IPID games. http://www.insecure.org/nmap/idlescan.html.

[6] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Measurement and classification of out-of-sequence packets in a tier-1 IP backbone. In *Proc. IEEE INFOCOM*, April 2003.

[7] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level internet path diagnosis. In *Proc. ACM Symp. on Operating Systems Principles (SOSP)*, October 2003.

[8] M. Rabbat, M. Coates, and R. Nowak. Multiple source, multiple destination network tomography. In *Proc. IEEE INFOCOM*, March 2004.

[9] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, August 2002.