Validation and Improvement of the Lossy Difference Aggregator to Measure Packet Delays

Josep Sanjuàs-Cuxart, Pere Barlet-Ros, and Josep Solé-Pareta

Department of Computer Architecture
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
{jsanjuas,pbarlet,pareta}@ac.upc.edu

Abstract. One-way packet delay is an important network performance metric. Recently, a new data structure called Lossy Difference Aggregator (LDA) has been proposed to estimate this metric more efficiently than with the classical approaches of sending individual packet timestamps or probe traffic. This work presents an independent validation of the LDA algorithm and provides an improved analysis that results in a 20% increase in the number of packet delay samples collected by the algorithm. We also extend the analysis by relating the number of collected samples to the accuracy of the LDA and provide additional insight on how to parametrize it. Finally, we extend the algorithm to overcome some of its practical limitations and validate our analysis using real network traffic.

1 Introduction

Packet delay is one of the main indicators of network performance, together with throughput, jitter and packet loss. This metric is becoming increasingly important with the rise of applications like voice-over-IP, video conferencing or online gaming. Moreover, in certain environments, it is an extremely critical network performance metric; for example, in high-performance computing or automated trading, networks are expected to provide latencies in the order of few microseconds [1].

Two main approaches have been used to measure packet delays. Active schemes send traffic probes between two nodes in the network and use inference techniques to determine the state of the network (e.g., [2–5]). Passive schemes are, instead, based on traffic analysis in two of the points of a network. They are, in principle, less intrusive to the network under study, since they do not inject probes. However, they have been often disregarded, since they require collecting, transmitting and comparing packet timestamps at both network measurement points, thus incurring large overheads in practice [6]. For example, [7] proposes delaying computations to periods of low network utilization if measurement information has to be transmitted over the network under study.

The Lossy Difference Aggregator (LDA) is a data structure that has been recently proposed in [1] to enable fine-grain measurement of one-way packet delays using a passive measurement approach with low overhead. The data structure is extremely lightweight in comparison with the traditional approaches, and can collect a number of samples that easily outperforms active measurement techniques, where traffic probes interfering with the normal operation of a network can be a concern.

The main intuition behind this measurement technique is to sum all packet timestamps in the first and second measurement nodes, and infer the average packet delay by subtracting these values and dividing over the total number of packets. The LDA, though, maintains several separate counters and uses coordinated, hash-based traffic sampling [8] in order to protect against packet losses, which would invalidate the intuitive approach. The complete scheme is presented in Sect. 2.

This work constitutes an independent validation of the work presented in [1]. Section 3 revisits the analysis of the algorithm. In particular, Sect. 3.1 investigates the effective number of samples that the algorithm can collect under certain packet loss ratios. This work improves the original analysis, and finds that doubling the sampling rates suggested in [1] maximizes the expectation of the number of samples collected by the algorithm. In Sect. 3.2, we contribute an analysis that relates the effective sample size with the accuracy that the method can obtain, while Sect. 3.3 compares the network overhead of the LDA with pre-existing techniques.

For the case when packet loss ratios are unknown, the original work proposed and compared three reference configurations of the LDA in multiple memory banks to target a range of loss ratios. In Sects. 3.4 and 3.5 we extend our improved analysis to the case of unknown packet loss, and we (*i*) find that such reference configurations are almost equivalent in practice, and (*ii*) provide improved guidelines on how to dimension the multi-bank LDA.

Sect. 4 validates our analysis through simulations, with similar parameters to [1], for the sake of comparability. Finally, in Sect. 5 we deploy the LDA on a real network scenario. The deployment of the LDA in a real setting presents a series of challenges that stem from the assumptions behind the algorithm as presented in [1]. We propose a simple extension of the algorithm that overcomes some of the practical limitations of the original proposal.

At the time of this writing, another analysis of the Lossy Difference Aggregator already exists in the form of a public draft [9]. The authors provide a parallel analysis of the expectation for the sample size collected by the LDA and, coherently with ours, suggest doubling the sampling rates compared to [1]. For the case where packet loss ratios are unknown beforehand, their analysis studies how to dimension the multi-bank LDA to maximize the expectation for the sample size. Optimal sampling rates are determined that maximize sample sizes for tight ranges of expected packet loss ratios. Our analysis differs in that we relate sample size with accuracy, and focus on maximizing accuracy rather than sample size. Additionally, our study includes an analytic overhead comparison with traditional techniques, presents the first real world deployment of the LDA and proposes a simple extension to overcome some of its practical limitations.

2 Background

The Lossy Difference Aggregator (LDA) [1] is a data structure that can be used to calculate the average one-way packet delay between two network points, as well as its standard deviation. We refer to these points as the sender and the receiver, but they need not be the source or the destination of the packets being transmitted, but merely two network viewpoints along their path.

The LDA operates under three assumptions. First, packets are transmitted strictly in FIFO order. Second, the clocks of the sender and the receiver are synchronized. Third, the set of packets observed by the receiver is identical to the one observed by the sender, or a subset of it when there is packet loss. That is, the original packets are not diverted, and no extra traffic is introduced that reaches the receiver.

A classic algorithm to calculate the average packet delays in such a scenario would proceed as follows. In both the sender and the receiver, the timestamps of the packets are recorded. After a certain measurement interval, the recorded packet delays (or, possibly, a subset of them) are transmitted from the sender to the receiver, which can then compare the timestamps and compute the average delay. Such an approach is impractical, since it involves storing and transmitting large amounts of information.

The basic idea behind the LDA is to maintain a pair of accumulators that sum all packet timestamps in the sender and the receiver separately, as well as the total count of packets. When the measurement interval ends, the sender transmits the value of its accumulator to the receiver, which can then compute the average packet delay by subtracting the values and dividing over the total number of packets.

The LDA requires the set of packets processed by the sender and the receiver to be identical, since the total packet counts in the sender and the receiver must agree. Thus, it is extremely sensitive to packet loss. In order to protect against it, the LDA partitions the traffic into b separate streams, and aggregates timestamps for each one separately in both the sender and the receiver. Additionally, for each of the sub-streams, it maintains a packet count. Thus, it can detect packet losses and invalidate the data collected in the corresponding accumulators. When the measurement interval ends, the sender transmits all of the values of the accumulators and counters to the receiver. Then, the receiver discards the accumulators where packet counts disagree, and computes an estimate for the average sampling delay using the remainder.

Each of the accumulators must aggregate the timestamps from the same set of packets in the sender and the receiver, i.e., both nodes must partition the traffic using the same criteria. In order to achieve this effect, the same pre-arranged, pseudo-random hash function is used in both nodes, and the hash of a packet identifier is used to determine its associated position in the LDA.

As packet losses grow high, though, the number of accumulators that are invalidated increases rapidly. As an additional measure against packet loss, the LDA samples the incoming packet stream. In the most simple setting, all of the accumulators apply an equal sampling rate p to the incoming packet stream.

Again, sender and receiver sample incoming packets coordinately using a pre-arranged pseudo-random hash function [8].

As an added benefit, the LDA data structure can also be mined to estimate the standard deviation of packet delays using a known mathematical trick [10]. We omit this aspect of the LDA in this work, but the improvements we propose will also increase the accuracy in the estimation of the standard deviation of packet delays.

3 Improved Analysis

The LDA is a randomized algorithm that depends on the correct setting of the sampling rates to gather the largest possible number of packet delay samples. The sampling rate p presents a classical tradeoff. The more packets are sampled, the more data the LDA can collect, but the more it is affected by packet loss. Conversely, lower sampling rates provide more protection against loss, but limit the amount of information collected by the accumulators.

This section improves the original analysis of the Lossy Difference Aggregator (LDA) introduced in [1] in several ways. First, it improves the analysis of the expected number of packet delay samples it can collect, which leads to the conclusion that sampling rates should be twice the ones proposed in [1]. Second, it relates the number of samples with the accuracy in an easy to understand way that makes it obvious that undersampling is preferable to sampling too many packets. Third, it compares its network overhead with pre-existing passive measurement techniques. Fourth, it provides a better understanding and provides guidelines to dimension multi-bank LDAs.

3.1 Effective Sample Size

In order to protect against packet loss, the LDA uses packet hashes to distribute timestamps across several accumulators, so that losses only invalidate the samples collected by the involved memory positions. Table 1 summarizes the notation used in this section. Given n packets, b buckets (accumulator-counter pairs) and packet loss probability r, the probability of a bucket of staying useful corresponds to the probability that no lost packet hashes to the bucket in the *receiver* node, which can be computed as $(1-r/b)^n \approx e^{-n\,r/b}$ (according to the law of rare events).

Then, the expectation for the number of usable samples, which we call the effective sample size, can be approximated to $E\left[S\right] \approx \frac{(1-r)\,n}{e^{n\,r/b}}$. In order to provide additional protection against packet losses, the LDA also samples the incoming packets; we can adapt the previous formulation to account for packet sampling as follows:

$$E\left[S\right] \approx \frac{(1-r)\,p\,n}{e^{n\,r\,p/b}} \tag{1}$$

Reference [9] shows that this approximation is extremely accurate for large values of n. The approximation is best as n becomes larger and the probability of

Table 1. Notation

name	variable	name	variable
n	#pkts	b	#buckets
r	packet loss ratio	μ	average packet delay
p	sampling rate	$\hat{\mu}$	estimate of the avg. delay

sampling a packet loss stays low. Note that this holds in practice; otherwise, the buckets would too often be invalidated. For example, when the absolute number of sampled packet losses is in the order of the number of buckets b, it obtains relative errors around 5×10^{-4} for as few as $n = 1000$ packets. Note however that this formula only accounts for a situation where all buckets use an equal fixed sampling rate p, i.e., a single bank LDA. Section 3.5 extends this analysis to the multi-bank LDA, while Sect. 4 provides an experimental validation of this formula.

Reference [1] provides a less precise approximation for the expected effective sample size. When operating under a sampling rate $p = \alpha\, b/(L+1)$, a lower bound $E[S] >= \alpha\,(1-\alpha)\, R\, b/(L+1)$ is provided, where R corresponds to the number of received packets and L to the number of lost packets; in our notation, $R = n\,(1-r)$ and $L = n\,r$. Trivially, this bound is maximized when $\alpha = 0.5$. Therefore, it is concluded that the best sampling rate p that can be chosen is $p = 0.5\,\frac{b}{n\,r+1}$.

However, our improved analysis leads to a different value for p by maximizing (1). The optimal sampling rate p that maximizes the effective sample size for any known loss ratio r can be obtained by solving $\frac{\partial E[S]}{\partial p} = 0$, which leads to $p = \frac{b}{n\,r}$ (in practice, we set $p = \min(b/n\,r, 1)$). Thus, our analysis approximately doubles the sampling rate compared to [1], i.e., sets $\alpha = 1$ in their notation, which yields an improvement in the effective sample size of around 20% at no cost. The conclusions of this improved analysis are coherent with the parallel analysis of [9], which also shows that the same conclusions are reached without the approximation in (1).

Assuming a known loss ratio and the optimal setting of the sampling rate $p = \frac{b}{n\,r}$, then, the expectation of the effective sample size is (by substitution of p in (1)):

$$E[S_{opt}] = \frac{1-r}{r\,e}\, b \qquad (2)$$

In other words, given a known number of incoming packets and a known loss ratio, setting p optimally maximizes the expectation of the sample size at $\frac{1-r}{r\,e}$ samples per bucket. Figure 1 shows how the number of samples that can be collected by the LDA quickly degrades when facing increasing packet loss ratios. Therefore, in a high packet loss ratio scenario, the LDA will require large amounts of memory to preserve the sample size. As an example, in order to sustain the same sample size of a 0.1% loss scenario, the LDA must grow around

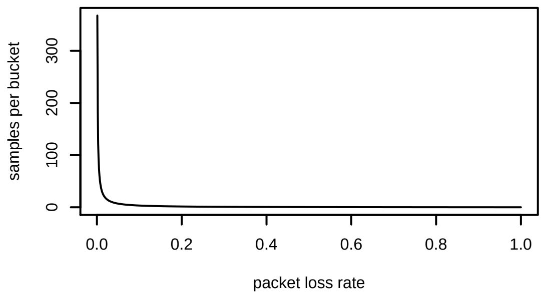

Fig. 1. Expected number of samples collected per bucket under varying packet loss ratios, assuming an ideal LDA that can apply, for each packet loss ratio, the optimal sampling rate

50 times larger on 5% packet loss, and by a factor of around 250 in the case of 20% packet loss.

Recall that this analysis assumes that the packet loss ratios are known beforehand, so that the sampling rate can be tuned optimally. When facing unknown loss ratios, the problem becomes harder, since it is not possible to configure $p = \frac{b}{n\,r}$, given that both parameters are unknown. However, this analysis does provide an upper bound on the performance of this algorithm. In any configuration of the LDA, including in multiple banks, the expectation of the effective sample size will be at most $\frac{1-r}{r\,e}\,b$.

3.2 Accuracy

It is apparent from the previous subsection that increasing packet loss ratios have a severe impact on the effective sample size that the LDA can obtain. However, the LDA is empirically shown to obtain reasonable accuracy up to 20% packet loss in [1]. How can we accommodate these two facts? The resolution of this apparent contradiction lies in the fact that the accuracy of the LDA does not depend linearly on the sample size but, instead, the gains in terms of accuracy of the larger sample sizes are small.

The LDA algorithm estimates the average delay μ from a sample of the overall population of packet delays. According to the central limit theorem, the sample mean is a random variable that converges to a normal distribution as the sample size (S in our notation) grows [11]. The rate of convergence towards normality depends on the distribution of the sampled random variable (in this case, packet delays).

If the arbitrary distribution of the packet delays has mean μ and variance σ^2, assuming that the sample size S obtained by the LDA is large enough for the normal approximation to be accurate, the sample mean can be considered

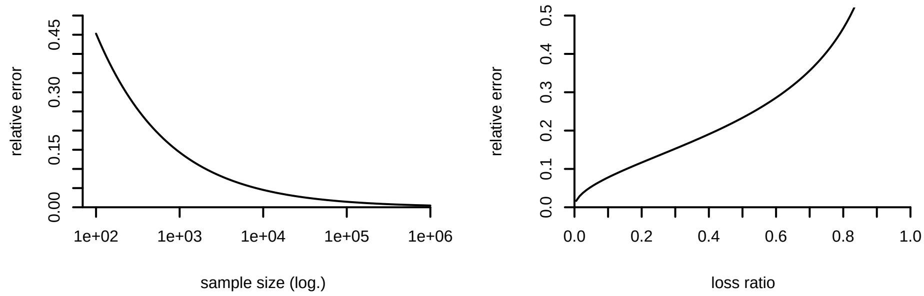

Fig. 2. 99% confidence bound on the relative error of the estimation of the average delay as a function of the obtained sample size (left) and as a function of the packet loss ratio (right), assuming a 1024 bucket ideal LDA, 5×10^6 packets and Weibull ($\alpha = 0.133$, $\beta = 0.6$) distributed packet delays

to be normally distributed, with mean μ and variance σ^2/S, which implies that, with 99% confidence, the estimate of the average delay $\hat{\mu}$ as the sample average will be within $\mu \pm 2.576 \frac{\sigma}{\sqrt{S}}$ and, thus, the relative error will be below $\frac{2.576\,\sigma}{\mu\,\sqrt{S}}$.

An observation to be made is that the relative error of the LDA is proportional to $\frac{1}{\sqrt{S}}$, that is, halving the relative error requires 4 times as many samples. A point is reached where the return of obtaining additional samples has a negligible practical impact on the relative error.

As stated, the accuracy of the LDA depends on the distribution of the packet delays, which are shown to be accurately modeled by a Weibull distribution in [6], and this distribution is used in [1] to evaluate the LDA. Figure 2 plots, as an example, the accuracy as a function of the sample size (left) and as a function of the loss ratio (right) when packet delays are Weibull distributed with scale parameter $\alpha = 0.133$ and shape $\beta = 0.6$, and 5×10^6 packets per measurement interval (these parameters have been chosen consistently with [1] for comparability). It can be observed that, in practice, small sample sizes obtain satisfactory accuracies. In this particular case, 2000 samples bound the relative error to around 10%, 8000 lower the bound to 5%, and 25 times as many, to 1%.

3.3 Overhead

Ref. [1] presents an experimental comparison of the LDA with active probing. In this section, we compare the overhead of the LDA with that of a passive measurement approach based on trajectory sampling [8] that sends a packet identifier and a timestamp for each sampled packet. As a basis for comparison, we compute the network overhead for each method per collected sample. Note that, for equal sample sizes, the accuracy of both methods is expected to match, since samples are collected randomly.

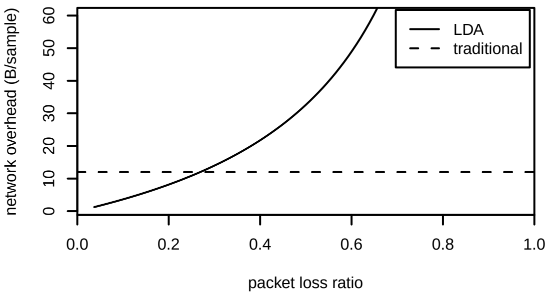

Fig. 3. Communication overhead of the LDA relative to a traditional trajectory sampling approach, assuming 12 byte per bucket and per sample transmission costs

Traditional techniques incur an overhead directly proportional to the collected number of samples. For example, an active probe will send a packet to obtain each sample. The overhead of a trajectory sampling based technique is also a constant α bytes/sample. For example, a 32 bit hash of a packet plus a 64 bit timestamp set $\alpha = 12$.

However, as discussed in the previous section, the sample size collected by the LDA depends on the packet loss ratio. Assuming a single-bank, optimally dimensioned LDA, it requires sending $b \times \beta$ bytes (where β denotes the size of a bucket) to gather $\frac{1-r}{r\,e}\,b$ samples. Thus, the overhead of the LDA is $\frac{\beta\,r\,e}{1-r}$ B/sample, and using 64 bit timestamp accumulators and 32 bit counters yields $\beta = 12$.

The LDA is preferable as long as it has lower overhead, i.e., $\frac{\beta\,r\,e}{1-r} < \alpha$ and, thus, $r < \frac{\alpha}{\beta\,e+\alpha}$. The values of α and β will vary in real deployments (e.g., timestamps can be compressed in both methods). In the example, where $\alpha = \beta = 12$, the LDA is preferable as long as $r < \frac{1}{e+1} \approx 0.27$. Figure 3 compares the overheads of both techniques in such a scenario, and shows the superiority of the LDA for the lowest packet loss ratios and its undesirability for the highest.

3.4 Unknown Packet Loss Ratio

It has already been established that the optimal choice of the LDA sampling rate is $p = \frac{b}{n\,r}$, which obtains $\frac{1-r}{r\,e}\,b$ samples. However, in practice, both n and r are unknown a priori, since they depend on the network conditions, which are generally unpredictable. Thus, setting p beforehand implies that, inevitably, its choice will be suboptimal.

What is the impact of over and under-sampling, i.e., setting a conservatively low or an optimistically high sampling rate on the LDA algorithm? We find that undersampling is preferable to oversampling. As explained, the relative error of the algorithm is proportional to $1/\sqrt{S}$. Thus, oversampling leads to collecting

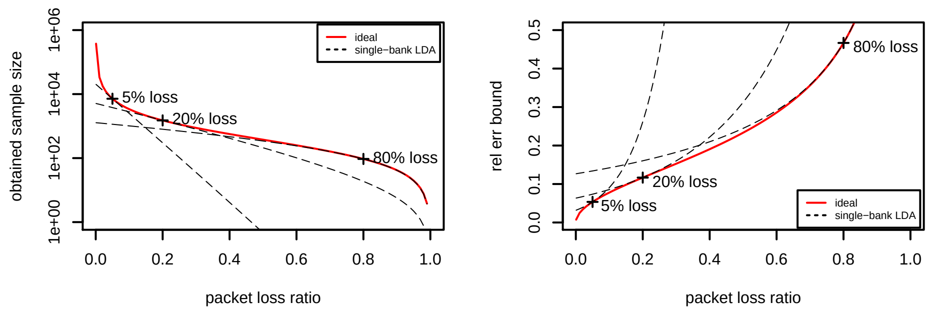

Fig. 4. Impact on the sample size (left) and expected relative error (right) of selecting a sub-optimal sampling rate

a high number of samples on low packet loss ratios, and slightly increases the accuracy on such circumstances, but leads to a high percentage of buckets being invalidated on high loss, thus incurring large errors. Conversely, undersampling preserves the sample size on high loss, thus obtaining reasonable accuracy, at the cost of missing the opportunity to collect a much larger sample on when losses are low, which, however, has a comparatively lower impact on the accuracy.

Figure 4 provides a graphical example of this analysis. In this example we consider, again analogously to [1], Weibull ($\alpha = 0.133$, $\beta = 0.6$) distributed packet delays. We compare the sample sizes and accuracy bounds obtained by different configurations of the LDA using a value of p targeted at loss ratios of 5%, 20% and 80%. All LDA configurations use $b = 1024$ accumulators. It can be observed that, in terms of sample size, the conservative setting of p for 80% loss underperforms, in terms of sample size, under the lowest packet loss ratios, but this loss does not imply an extreme degradation in terms of measurement accuracy. On the contrary, the more optimistic sampling rate settings achieve better accuracy under low loss, but incur extreme accuracy degradation as the loss ratio grows.

3.5 The Multi-Bank LDA

So far, the analysis of the LDA has assumed all buckets have a common sampling rate p. However, as exposed in [1], when packet loss ratios are unknown, it is interesting to divide the LDA in multiple banks. A bank is a section of the LDA for which all the buckets use the same sampling rate. Each of the banks can be tuned to a particular sampling rate, so that, intuitively, the LDA is resistant to a range of packet loss ratios.

Reference [1] tests three different configurations of the multi-bank LDA, always using equal (or almost) sized banks. No systematic analysis is performed on the appropriate bank sizing nor on the appropriate sampling rate for each of the banks; each LDA configuration is somewhat arbitrary and based on intuition.

We extend our analysis to the most general multi-bank LDA, where each bucket i independently samples the full packet stream at rate p_i (i.e., our analysis supports all combinations of bank configurations and sampling rates). We adapt (1) accordingly:

$$E[S] \approx \sum_{i=1}^{b} \frac{(1-r)\, p_i\, n}{e^{n\, r\, p_i}} \tag{3}$$

When every bucket uses the same sampling rate, the two equations are equivalent with $p_i = p/b$ (each bucket receives $1/b$ of the traffic and samples packets at rate p). As for the error bound, the analysis from Sect. 3.2 still holds.

We have evaluated the three alternative multi-bank LDA configurations proposed in [1], using the same configuration parameters and distribution of packet delays. Figure 5 compares the accuracy obtained by the three configurations. The figure assumes, again, a Weibull distribution for packet losses, with shape parameter $\beta = 0.6$ and scale $\alpha = 0.133$, and a number of packets $n = 5 \times 10^6$. All configurations use $b = 1024$ buckets. The first uses two banks, each targeted to 0.005 and 0.1 loss; the second, three banks that target 0.001, 0.01 and 0.1 loss; the third, four banks that target 0.001, 0.01, 0.05 and 0.1 loss. The figure shows that, in practice, the three approaches (*lda-2*, *lda-3* and *lda-4* in the figure) proposed in [1] perform very similarly, which motivates further discussion on how to dimension multi-bank LDAs. The figure also provides, as a reference, the accuracy obtained by an *ideal* LDA that, for every packet loss ratio, obtains the best possible accuracy (from (2)).

We argue that, consistently with the discussion of subsection 3.4, in order to support a range of packet loss ratios, the LDA should be primarily targeted towards maintaining accuracy over the worst-case target packet loss ratio. Using this conservative approach has two benefits. First, it guarantees that a target accuracy can be maintained in the worst-case packet loss scenario. Second, it is guaranteed that its accuracy over the smaller packet loss ratios is at least as good.

However, this rather simplistic approach has an evident flaw in that it does not provide significantly higher performance gains in the lowest packet loss scenarios, where a small number of high packet sampling ratio provisioned buckets would easily gather a huge number of samples. Based on this intuition, as a rule of thumb, 90% of the LDA could be targeted to a worst-case sampling ratio, using the rest of the buckets to increase the accuracy in low packet loss scenarios.

A more sophisticated approach to dimensioning a multi-bank LDA is to determine the vector of sampling rates $< p_1, p_2, \ldots, p_b >$ that performs closest to optimal across a range of sampling rates. We have used numerical optimization to search for a vector of sampling rates such that it *minimizes the maximum difference* between the accuracies of the multi-bank LDA and the ideal LDA across a *range* of packet loss ratios. Additionally, we have restricted the search space to sampling rates that are powers of two for performance reasons [1, 9].

Table 2. Per-bucket sampling rates respective to the full packet stream of the numerically optimized LDA for the given scenario. Overall sampling rate is around 0.47%

sampling rate	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}
#buckets	2	74	6	189	7	27	717	2

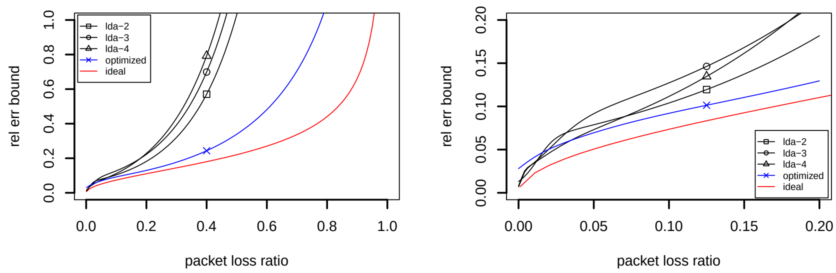

Fig. 5. Error bounds for several configurations of multi-bank LDA in the 0-1 packet loss ratio range (left) and in the 0-0.2 range (right)

We have obtained a multi-bank LDA that targets a range of loss rates between 0.1% and 20% for the given scenario: 5 million packets, Weibull distributed delays, and 1024 buckets. The best solution that our numerical optimizator has found is, coherently with the previous discussion, targeted primarily to the highest loss ratios. Table 2 summarizes the resulting multi-bank LDA. Most notably, a majority (70%) of the buckets use $p_i = 2^{-20}$, i.e., are targeted to a packet loss ratio of 20%, while fewer (around 20%) use $p_i = 2^{-17}$, i.e., are optimized for around 2.6% loss. All buckets combined sample around 0.47% of the packets.

Figure 5 shows the result of this approach (line *optimized*) when targeting a range of loss rates between 0.1% and 20% for 5 million packets with the mentioned Weibull distribution of delays. The solution our optimizer found has the desirable property of always staying within below 3% higher relative error than the best possible, for any given loss ratio within the target range. These results suggest that there is little room for improvement in the multi-bank LDA parametrization problem.

In the parallel analysis of [9], numerical optimization is also mentioned as an alternative to maximize the effective sample size when facing unknown packet loss. Optimal configurations are derived using competitive analysis for some particular cases of tight ranges of target packet loss ratios $[l_1, l_2]$. In particular, it is found that both for $l_2/l_1 \leq 2$, and for $l_2/l_1 \leq 5.5$ and a maximum of 2 banks, the optimal configuration is a single bank LDA with $p = \frac{\ln l_2 - \ln l_1}{l_2 - l_1}$. We believe that our approach is more practical in that it supports arbitrary packet loss ratios and it focuses on preserving the accuracy, rather than sample size.

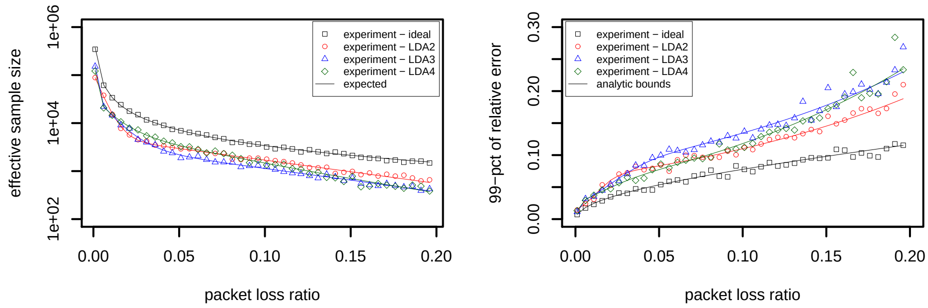

Fig. 6. Effective sample size (left) and 99 percentile of the relative error (right) obtained from simulations of the LDA algorithm using 5 million packets per measurement interval, and Weibull distributed packet delays

4 Validation

In the previous section, we derived formulas for the expected effective sample size of the LDA when operating under various sampling rates, and provided bounds for the expected relative error under typical distributions of the network delays. In this section, we validate the analytical results through simulation.

We have chosen the same configuration parameters as in the evaluation of [1]. Thus, this section not only validates our analysis of the LDA algorithm, but also shows consistency with the previous results of [1]. The simulation parameters are as follows: we assume 5 million packets per measurement interval, and Weibull ($\alpha = 0.133$, $\beta = 0.6$) distributed packet delays. In our simulation, losses are uniformly distributed. Note however that, as stated in [1], the LDA is completely agnostic to the packet loss distribution, but only sensitive to the overall packet loss ratio. Thus, other packet loss models (e.g., in bursts [12]) are supported by the algorithm without requiring any changes.

Figure 6 (left) compares the expected sample sizes with the actual results from the simulations. The figure includes the three multi-bank LDA configurations introduced in [1], with expected sample size calculated using (3), and the *ideal* LDA that achieves the best possible accuracy under each packet loss ratio, obtained from (2). This figure validates our analysis of the algorithm, since effective sample sizes are always around their expected value (while in [1], only a noticeably pessimistic lower bound is presented).

On the other hand, Figure 6 (right) plots the 99 percentile of the relative error obtained after 500 simulations for each loss ratio, and compares it to the 99% bound on the error derived from the analysis of Sect. 3.2. The figures confirm the correctness of our analysis for both the effective sample size and the 99% confidence bound on the relative error.

5 Experiments

5.1 Scenario

In the previous section, a simulation based validation of our analysis of the LDA has been presented that reproduces that of [1]. In this section we evaluate the algorithm using real network traffic. To the best of our knowledge, this is the first work to evaluate the algorithm in a real scenario.

Our scenario consists of two measurement points: one of the links that connect the Catalan academic network (also known as Scientific Ring) to the rest of the Internet, and the access link of the Technical University of Catalonia (UPC). In the first measurement point, a commodity PC equipped with a 10 Gb/s Endace DAG card [13] obtains a copy the of the inbound traffic via an optical splitter, and filters for incoming packets with destination address belonging to the UPC network. In the second measurement point, a commodity PC equipped with a 1 Gb/s Endace DAG card [13] analyzes a copy of the traffic that enters UPC, obtained from a port mirror from a switch.

5.2 Deployment Challenges

The deployment of the LDA in a real world scenario presents important challenges. The design of the LDA is built upon several assumptions. First, as stated in [1], the clocks in the two measurement points must be synchronized. We achieve this effect by synchronizing the internal DAG clocks prior to trace collection. Second, packets in the network follow strict FIFO ordering, and the monitors can inject control packets in the network (by running in the routers themselves) which also observe this strict FIFO ordering, and are used to signal measurement intervals. In our setting, packets are not forwarded in a strict FIFO ordering due to different queueing policies being applied to certain traffic. Moreover, injecting traffic to signal the intervals is unfeasible, since the monitors are isolated from the network under study.

Third, in the original proposal, the complete set of packets observed in the second monitor (*receiver*) must have also been observed in the first (*sender*). In [1], the LDA algorithm is proposed to be applied in network hardware in a hop-by-hop fashion. However, this assumption severely limits the applicability of the proposal; for example, as is, it cannot be used in our scenario, since *receiver* observes packets that have been routed through a link to a commercial network that *sender* does not monitor (we refer to these packets as *third party traffic*). This limitation could be addressed by using appropriate traffic filters to discern whether each packet comes from *receiver* (e.g., source MAC address, or source IP address), but in the most general case, this is not possible. In particular, in our network, we lack routing information, and traffic engineering policies make it likely that the same IP networks are routed differently.

The problem lies in that the LDA counters might match by chance when, in *receiver*, packet losses are compensated by extra packets from the third party traffic. The LDA would assume that the affected buckets are usable, and introduce severe error. We work around this by introducing a simple extension to the data structure: we attach to each LDA bucket an additional memory position that stores an XOR of all the hashes of the packets aggregated in the corresponding accumulator. Thus, *receiver* can trivially confirm that the set of packets of each position matches the set of packets aggregated in *sender* by checking this XOR. From a practical standpoint, using this approach makes third party traffic count as losses. We use 64 bit hashes and, thus, the probability of the XORs matching by chance is negligible[1].

5.3 Experimental Results

We have simultaneously collected a trace in each of the measurement points in the described scenario, and wrote two CoMo [14] modules to process the traces offline: one that implements the LDA, and another that computes the average packet delays exactly. The traces have a duration of 30 minutes. We have configured 10 seconds measurement intervals, so that the average number of packets per measurement interval is in the order of 6×10^5.

We have tested 16 different single-bank configurations of the LDA with $b = 1024$ buckets and sampling rates ranging from 2^0 to 2^{-15}. Also, we have used our numerical optimizator to obtain a multi-bank LDA configuration that tolerates up to 80% loss in our scenario. Figure 7 summarizes our results.

As noted in the previous discussion, third party traffic that is not seen in *sender* is viewed as packet losses in *receiver*. Therefore, our LDAs operate at an average loss rate of around 10%, which roughly corresponds to the fraction of packets arriving from a commercial network link that *sender* does not monitor.

Hence, the highest packet sampling ratios are over-optimistic and collect too much traffic. It can be observed in Fig. 7 (right) that sampling ratios from 2^0 to 2^{-4} lose an intolerable amount of measurement intervals because all LDA buckets become unusable. Lower sampling rates, though, are totally resistant to the third party traffic.

Figure 7 (left) plots the results in terms of accuracy, but only including measurement intervals not lost. It can be observed that 2^{-6} and 2^{-7} are the best settings. This is consistent with the analysis of Sect. 3, that suggests using $p = \frac{b}{n\,r} \approx 0.17 \approx 2^{-6}$. The figure also includes the performance of our numerically optimized LDA, portrayed as a horizontal line (the multi-bank LDA is a hybrid of the other sampling rates). It performs very similarly to the best static sampling rates. However, it is important to note that this configuration will consistently perform close to optimal when losses (or third party traffic) grow to 80%, obtaining errors below 50%, while the error bound for the less flexible single bank LDA reaches 400%.

[1] The XORs of the hashes have to be transmitted from *sender* to *receiver*, causing extra network overhead. Choosing the smallest hash size that still guarantees high accuracy is left for future work.

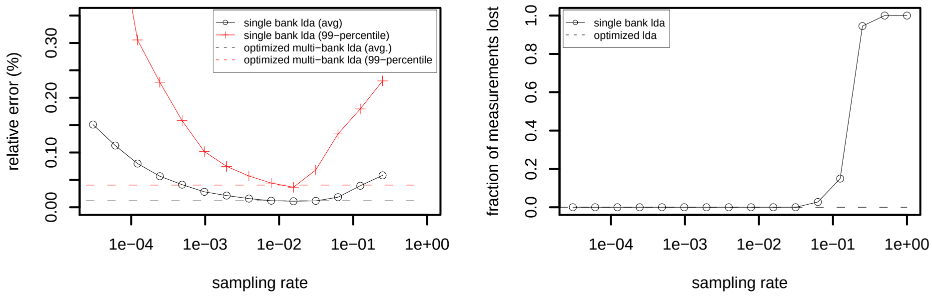

Fig. 7. Experimental results

On average, for each measurement interval, the optimized LDA collected around 3478 samples, while transmitting 1024×20 bytes (8 for the timestamp accumulators and the XOR field plus 4 for the counter for each bucket), resulting in 5.8 B/sample of network overhead. A traditional technique based on sampling and transmitting packet timestamps would cause a higher overhead, e.g., if using 8 byte timestamps and 4 byte packet IDs, it would transmit 12 B/sample. Thus, in this scenario, the LDA reduced the communication overhead in over 50%.

6 Conclusions

We have performed a validation on the Lossy Difference Aggregator (LDA) algorithm originally presented in [1]. We have improved the theoretical analysis of the algorithm by providing a formula for the expected sample size collected by the LDA, while in [1] only a pessimistic lower bound was presented. Our analysis finds that the sampling rates originally proposed must be doubled.

Only three configurations of the more complex multi-bank LDA were evaluated in [1]. We have extended our analysis to multi-bank configurations, and explored how to properly parametrize them, obtaining a procedure to numerically search for multi-bank LDA configurations that maximize accuracy over an arbitrary range of packet losses. Our results show that there is little room for additional improvement in the problem of multi-bank LDA configuration.

We have validated our analysis through simulation and using traffic from a monitoring system deployed over a large academic network. The deployment of the LDA on a real network presented a number of challenges related to the assumptions behind the original proposal of the LDA algorithm, that does not tolerate packet insertion/diversion and depends on strict FIFO packet forwarding. We propose a simple extension that overcomes such limitations.

We have compared the network overhead of the LDA with pre-existing techniques, and observed that it is preferable under zero to moderate loss or addition/diversion of packets (up to ∼25% combined). However, the extra overhead of pre-existing techniques can be justified in some scenarios, since they can provide further information on the packet delay distribution (e.g., percentiles), than just the average and standard deviation that are provided by the LDA.

Acknowledgments

We thank the anonymous reviewers and Fabio Ricciato for their comments, which led to significant improvements in this work. This research has been partially funded by the *Comissionat per a Universitats i Recerca del DIUE de la Generalitat de Catalunya* (ref. 2009SGR-1140).

References

1. Kompella, R., Levchenko, K., Snoeren, A., Varghese, G.: Every microsecond counts: tracking fine-grain latencies with a lossy difference aggregator. In: Proc. of ACM SIGCOMM Conf. (2009)
2. Bolot, J.: Characterizing end-to-end packet delay and loss in the internet. Journal of High Speed Networks **2**(3) (1993)
3. Paxson, V.: Measurements and analysis of end-to-end Internet dynamics. University of California at Berkeley, Berkeley, CA (1998)
4. Choi, B., Moon, S., Cruz, R., Zhang, Z., Diot, C.: Practical delay monitoring for ISPs. In: Proc. of ACM Conf. on Emerging network experiment and tech. (2005)
5. Sommers, J., Barford, P., Duffield, N., Ron, A.: Accurate and efficient SLA compliance monitoring. ACM SIGCOMM Computer Communication Review **37**(4) (2007)
6. Papagiannaki, K., Moon, S., Fraleigh, C., Thiran, P., Diot, C.: Measurement and analysis of single-hop delay on an IP backbone network. IEEE Journal on Selected Areas in Communications **21**(6) (2003)
7. Zseby, T.: Deployment of sampling methods for SLA validation with non-intrusive measurements. In: Proc. of Passive and Active Measurement Workshop (2002)
8. Duffield, N., Grossglauser, M.: Trajectory sampling for direct traffic observation. IEEE/ACM Transactions on Networking **9**(3) (2001)
9. Finucane, H., Mitzenmacher, M.: An improved analysis of the lossy difference aggregator. (public draft) http://www.eecs.harvard.edu/~michaelm/postscripts/LDApre.pdf.
10. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. Journal of Computer and system sciences **58**(1) (1999)
11. Rohatgi, V.: Statistical inference. Dover Pubns (2003)
12. Sommers, J., Barford, P., Duffield, N., Ron, A.: Improving accuracy in end-to-end packet loss measurement. ACM SIGCOMM Computer Communication Review **35**(4) (2005)
13. Endace: DAG network monitoring cards http://www.endace.com.
14. Barlet-Ros, P., Iannaccone, G., Sanjuàs-Cuxart, J., Amores-López, D., Solé-Pareta, J.: Load shedding in network monitoring applications. In: Proc. of USENIX Annual Technical Conf. (2007)