Approximating the end-to-end delay using local measurements: a preliminary study based on conditional expectation

Huu-Nghi Nguyen, Thomas Begin, Anthony Busson, Isabelle Guérin-Lassous

Approximating the end-to-end delay using local measurements: a preliminary study based on conditional expectation

Huu-Nghi Nguyen, Thomas Begin, Anthony Busson and Isabelle Guérin Lassous
Université Lyon, UCB Lyon 1, CNRS, ENS de Lyon, Inria, LIP UMR 5668
15 parvis René Descartes, F-69342, Lyon, FRANCE

***Abstract*—With Software-Defined Networking (SDN), computer networks will gain a better control and management on their physical resources and on flows requiring a specific type of QoS. To fulfil these objectives, networks must be able to characterize the end-to-end performance of flows, which are usually unknown and not directly measurable. Instead, networks have typically at their disposal only local measurements, collected at each node.**

In this paper, we propose a new method to evaluate the variance of the end-to-end delay based only on measurements collected on nodes. The core of our solution is to link samples of waiting times at different nodes in order to approximate the corresponding covariance terms, needed to refine the approximate value of the sought variance. We evaluate the accuracy of our solution using two different scenarios. The obtained results show that the method is generally good, with a relative error on the estimated standard deviation of the end-to-end delay usually close to 5%, though it may exceed 10% when the network is facing high levels of load and experiencing packet losses.

I. Introduction

In recent years, Software-Defined Networking (SDN) has emerged as a promising paradigm for computer networks. In SDN, the network comprise switches and controllers. SDN switches are mostly dedicated to the forwarding of packets, and to statistic computation on the conveyed traffic. On the other hand, one or several controllers are in charge of managing the network. This includes tasks such as route computations as well as more elaborated tasks (e.g., load sharing, resource allocation, QoS provisioning, admission control). Therefore, within the SDN paradigm, the control plane is centralized, and all decisions are taken by the controllers. This approach starkly differs from the classical distributed approach of the Internet. SDN networks are expected to be more flexible, manageable, and prone to efficient optimization policies than their classical IP counterparts. This new technological framework also offers the opportunity to implement services, typically QoS, using a global knowledge of the network performance.

Thus, SDN controllers will be able to make their decisions (e.g., routing, resource allocation, admission control) based on end-to-end performances of flows. This ability appears as a huge potential asset for guaranteeing an end-to-end quality of service to certain flows. However, end-to-end performance measurements are typically hard to collect. A crucial question in this context is then: is it possible to infer end-to-end performance based only on local measurements collected on the SDN switches? In this paper, we tackle this issue for the end-to-end delay. Although each SDN switch can measure the delays spent by packets kept waiting before transmission, the end-to-end delays of packets cannot be directly derived from these local measurements. Indeed, correlations occur between these local delays, and the SDN controllers must handle them properly to obtain relevant indicators of the performance: variance, or distribution of the end-to-end delay. We propose a method to estimate the variance of the end-to-end delay experienced by packets within SDN networks.

This paper is organized as follows. In the next section, we present related work and the motivation of this work. In section III, we present the considered scenario, and the notation. We also decompose the different terms involved in the variance of the end-to-end delay. Most of these terms are measurable or computable locally by the SDN switches. We show that the correlation mainly lies in the successive waiting times (time spent by a packet in a buffer). Section IV describes our proposed method to evaluate this correlation. Numerical results are shown in section V. Section VI concludes this paper.

II. Problem statement and related work

In this paper, we focus our efforts on the one-way end-to-end delay of packets. Note that this quantity differs from the round-trip delay measurement, also know as RTT [1], [2]. Note that the former is not simply half of the latter because network paths are known to be asymmetric [3]. Also, depending on the application or type of traffic (e.g. FTP or VoIP/Video), the relevancy of each type of delays differs.

Besides, measurement schemes may either be intrusive [4] or non-intrusive [5]. Intrusive schemes rely on probe packets to assess the latency between two arbitrary nodes of the network. Because they operate without any knowledge on the network (high level of privacy ensured), many intrusive schemes were proposed [6]–[10]. Despite considerable efforts, intrusive measurement schemes still suffer from a somewhat lack of understanding (e.g. effect of probe packet size and rate), and furthermore, the probe traffic may itself cause buffer

978–1–5090–0284–9/16/$31.00

overflows thereby representing a bias in the outcome [11]–[13].

Note that the vast majority, if not all, of existing measurement schemes (intrusive and non-intrusive altogether) heavily relies on clock synchronization between the nodes [14]–[16]. This can be viewed as a major drawback since this typically requires to equip nodes with a highly-specialized and costly GPS device.

Instead, we propose to infer the empirical distribution of the one-way end-to-end delay of packets, and a fortiori its two first moments, without any need of clock synchronization. The proposed method belongs to non-intrusive measurement schemes. It only resorts to measurements that can be collected independently by each node (e.g. packet sizes, inter-arrival times, source and destination nodes). In a previous work [17], we introduced a method to approximate the variance of the end-to-end delay in the case of a single flow with no competing traffic. Although this work represented an important milestone, we had to significantly revise our approach to deal with the case of multiple flows. We present the corresponding extension in this paper.

III. Scenarios, notations and variance decomposition

A. Notation

Without loss of generality, we study only the end-to-end delay of a given flow. Let f_1 denote that flow. We refer to this flow as the primary flow, and to its path along the SDN network as the primary path. The random variable P^{f_1} describes the packet size of the primary flow f_1.

We label the nodes of the primary path from 1 to N. Links connecting two successive nodes k and $k+1$ are characterized by their transmission capacity $\mathcal{C}_{k,k+1}$ (bps), and their propagation delay $\mathcal{R}_{k,k+1}$ (msec). Each node maintains a built-in buffer which is ruled by a First-Come First-Served discipline.

In addition to the primary flow, other flows may enter and leave the path at intermediate nodes of the primary path. We denote by $f_{k,l}$ the flow entering the primary path at node k, and leaving it at node l after hopping through $l-k$ along the path. Note that if a flow leaves the primary path, but then returns on a subsequent node, we will consider it as two different flows. Figure 1 illustrates such an example of a primary path of 4 nodes with a primary flow f_1 that is competing with one flow that leaves and re-enters into the primary path. In our model, this flow is split into two flows.

B. Assumptions

Throughout our work, we assume that nodes can measure for each packet they process:

- the size,
- the departure time (time at which a packet leaves the node based on a non-synchronized local clock),
- the destination address,
- the source address,
- the waiting time (time spent in the buffer waiting for transmission).

Thanks to the combination of the source and destination addresses, nodes are able to distinguish the different flows. Finally, we assume that each node k knows the propagation delay and the capacity of subsequent links on the primary path, i.e., $R_{l,l+1}$ and $C_{l,l+1}$ for $l \geq k$.

C. End-to-end delay decomposition

The time spent by any packet of a given flow f to go through a node k can de decomposed as a sum of random variables as follows:

(i) the processing delay, which is the necessary time to handle the packet header and to commute the packet to the proper network interface of the node - it is considered as negligible (it is also close to constant and thus its variance is null);
(ii) the transmission delay $\mathcal{S}_k^f$ that corresponds to the time to transmit physically the packet on the link - it depends on the packet size and the link capacity;
(iii) the waiting time $\mathcal{W}_k^f$ that amounts to the time spent by the packet kept in the buffer before its transmission;
(iv) the propagation delay $\mathcal{R}_{k,k+1}$ which is the time to propagate one bit from node k to node $k+1$ - it is supposed constant and its variance is consequently null.

It follows that the local delay for a packet of flow f_1 at node k, i.e. the time between its arrival at node k and its arrival at the next node $k+1$, can be expressed as:

$$D_k^{f_1} = \left(\mathcal{W}_k^{f_1} + \mathcal{S}_k^{f_1} + \mathcal{R}_k\right). \tag{1}$$

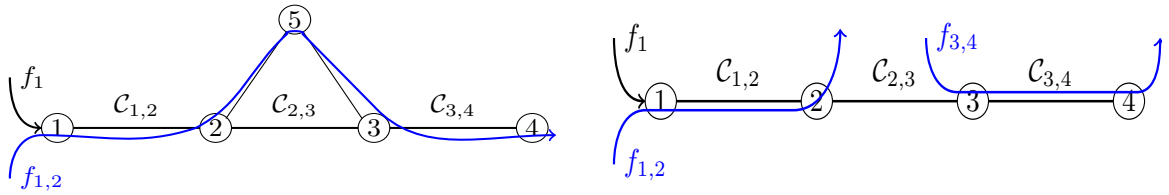
Note that this local delay, $D_k^{f_1}$, can be measured at node k. Therefore, we can formulate the end-to-end delay of the primary flow D^{f_1} as:

$$D^{f_1} = \sum_{k=1}^{N-1} \left(\mathcal{W}_k^{f_1} + \mathcal{S}_k^{f_1} + \mathcal{R}_k\right). \tag{2}$$

From (2), the mean of the end-to-end delay can be straightforwardly derived. It suffices that each node along the path sends an estimate of the mean local delay $D_k^{f_1}$ to the SDN controller. Then, the SDN controller simply computes the sum. However, computing the variance requires a more complex algorithm as the correlations between the local delays have to be taken into account. Using the properties of the variance on (2), and noting that $\mathcal{R}_{k,k+1}$ is a constant, we obtain:

$$\begin{aligned} \mathbb{V}[D^{f_1}] &= \mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{W}_k^{f_1}\right] + \mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{S}_k^{f_1}\right] \\ &+ 2 \times \sum_{1 \leq k < l \leq N-1} \mathrm{Cov}(\mathcal{W}_k^{f_1}, \mathcal{S}_l^{f_1}). \end{aligned} \tag{3}$$

$\mathcal{S}_k^{f_1}$ is a linear function of P^{f_1}, namely $\mathcal{S}_k^{f_1} = \frac{\mathrm{P}^{f_1}}{C_{k,k+1}}$. It follows that $\mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{S}_k^{f_1}\right]$ can be easily computed given the packet size distribution. Typically, this computation can be



(a) Real scenario.

(b) Modeled scenario.

Figure 1: A specific scenario where one flow leaves and re-enters into the primary path. In the realistic scenario, flows $f_{1,2}$ leaves at node 2 and re-enters at node 3. In the modeled scenario, flows $f_{1,2}$ is split into 2 flows $f_{1,2}$ and $f_{3,4}$.

carried out on the first node of the primary path. For each incoming packet, the node computes the transmission time over each subsequent link of the primary path, and sum them up to obtain a sample of $\sum_{k=1}^{N-1} \mathcal{S}_k^{f_1}$. The empirical variance is then obtained from this set of samples. As for the term $\mathrm{Cov}(\mathcal{W}_k^{f_1}, \mathcal{S}_l^{f_1})$, it can be computed in an analogous way. Indeed, each node k measures the waiting time $\mathcal{W}_k^{f_1}$, and then compute $\mathcal{S}_l^{f_1}$ based on the actual packet size and the link capacity $C_{l,l+1}$. By doing so, we obtain an estimation of the empirical covariance term in (3). Therefore, only the first term in (3) is left unknown.

We expand this variance term of the waiting times as follows:

$$\mathbb{V}\left[\sum_{k=1}^{N-1} \mathcal{W}_k^{f_1}\right] = \sum_{k=1}^{N-1} \mathbb{V}\left[\mathcal{W}_k^{f_1}\right] + 2 \times \sum_{1 \le k < l \le N-1} \mathrm{Cov}\left(\mathcal{W}_k^{f_1}, \mathcal{W}_l^{f_1}\right). \quad (4)$$

Note that in (4), the sum of $\mathbb{V}\left[\mathcal{W}_k^{f_1}\right]$ can be estimated locally at each node. Thus, using (3) and (4), our issue of estimating the variance of the end-to-end delay boils down to evaluating the correlation between the waiting times: $\mathrm{Cov}(\mathcal{W}_k^{f_1}, \mathcal{W}_l^{f_1})$. We propose a technique to evaluate this quantity in the next section.

IV. Estimating correlations between the successive waiting times

Intuitively, the correlation between waiting times of successive nodes should be more important than between nodes that are distant of several hops. Consequently, we evaluate only $\mathrm{Cov}(\mathcal{W}_k^{f_1}, \mathcal{W}_{k+1}^{f_1})$ for $1 \le k \le N-1$ and neglect the other terms. Simulation results will show that this assumption does not introduce significant errors.

The rationale of the proposed method is to allow the SDN controller to link samples of the waiting times on the two nodes k and $k+1$ to estimate their covariances. This link is made through a recurrent function that can be computed/implemented on both nodes k and $k+1$. This function infers the waiting time at node $k+1$ from local measures at node k. Nodes k and $k+1$ computes conditional expectation of their local waiting times according to this function. The controller is then able to map the conditional expectations according to the value of the recurrent function. To help the reader to understand which quantity is measured and which one is inferred, all quantities/samples that are inferred have a hat ($\widehat{w}$ for instance), and the ones that are measured by the nodes do not have it.

Our method is based on the following conditional expectation of $\mathbb{E}[\mathcal{W}_k^{f_1}\mathcal{W}_{k+1}^{f_1}]$:

$$\mathbb{E}\left[\mathcal{W}_k^{f_1}\mathcal{W}_{k+1}^{f_1}\right] = \mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_k^{f_1} \,\middle|\, \mathcal{W}_{k+1}^{f_1}\right] \mathcal{W}_{k+1}^{f_1}\right]. \quad (5)$$

To obtain an estimation of Equation 5 we propose the following approach. It is possible for a node to infer the waiting time at the next node. Indeed, it can measure the departure time of the leaving packets. These times correspond to the arrival time at the next node. Let us denote p^m the packet size of the m-th packet ($m \ge 1$). We also consider the variable $\widehat{\delta}_{k,k+1}^m$ that describes the inter-arrival time between packets $m-1$ and m at node $k+1$ inferred at node k. s_k^m is the transmission time of packet m at node k (it is locally measured or directly deduced from the packet length). Assuming that the server discipline is FIFO (First In-First Out), the waiting time and inter-arrival time at the next queue can be inferred through the following recurrent function:

$$\begin{cases} \widehat{w}_{k,k+1}^m = \max\{0, \widehat{w}_{k,k+1}^{m-1} + s_k^{m-1} - \widehat{\delta}_{k,k+1}^{m-1}\} \\ \widehat{\delta}_{k,k+1}^m = s_k^m + \max\{0, \widehat{\delta}_{k,k+1}^{m-1} - \widehat{w}_{k,k+1}^{m-1} - s_k^{m-1}\}, m > 1. \end{cases} \quad (6)$$

where $\widehat{w}_{k,k+1}^m$ is thus the waiting time of the packet m at node $k+1$ inferred by node k. We denote $\widehat{\mathcal{W}}_{k,k+1}^{f_1}$ the random variable which corresponds to these samples. Node $k+1$ will use the same recurrent function to obtain the values inferred by node k. The corresponding random variable is denoted $\widehat{\mathcal{W}}_{k+1,k+1}^{f_1}$. Local waiting times measured on the two nodes k and $k+1$ can then be mapped through the inferred values of w_{k+1}^m. Equation 5 is then approximated as:

$$\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_k^{f_1} \,\middle|\, \mathcal{W}_{k+1}^{f_1}\right] \mathcal{W}_{k+1}^{f_1}\right] \approx \mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_k^{f_1} \,\middle|\, \widehat{\mathcal{W}}_{k,k+1}^{f_1}\right] \mathbb{E}\left[\mathcal{W}_{k+1}^{f_1} \,\middle|\, \widehat{\mathcal{W}}_{k+1,k+1}^{f_1}\right]\right]. \quad (7)$$

The right hand side of Equation (7) is computed by the use of two histograms evaluated at node k. Node k divides the range of the computed values of $\widehat{\mathcal{W}}_{k,k+1}^{f_1}$ into L disjoint

intervals. The i^{th} of these intervals is denoted Δ_i. Node k measures the waiting time of packet m, denoted w_k^m and uses the recurrent function given by Equation 6 to obtain the corresponding sample $\widehat{w}_{k,k+1}^m$ of the inferred waiting time at the next queue (node $k+1$). It obtains a set of couples $(w_k^m, \widehat{w}_{k,k+1}^m)$. In order to approximate conditional expectations in Equation 7, in the interval Δ_i, we set:

$$\widehat{\mathbb{E}}\left[\mathcal{W}_k^{f_1} \,\middle|\, \widehat{\mathcal{W}}_{k,k+1}^{f_1}\right](i) = \frac{1}{c_i} \sum_{m=1}^{M} w_k^m \mathbb{1}_{\widehat{w}_{k,k+1}^m \in \Delta_i}. \quad (8)$$

where c_i is the number of samples $\widehat{w}_{k,k+1}^m$ lying in the interval Δ_i, and M the total number of samples.

Node $k+1$ is able to apply exactly the same recurrent function to infer the values of $\widehat{w}_{k,k+1}^j$ computed by node k. From the inter-arrival times (that are measured on node $k+1$), it applies recurrently the first equation of the equation system 6. These samples are denoted $\widehat{w}_{k+1,k+1}^m$: waiting time on node $k+1$ of packet m inferred at node $k+1$. They correspond to the previously defined random variable $\widehat{\mathcal{W}}_{k+1,k+1}^{f_1}$. These samples can be different on the one computed by node k as some packets can be lost. We set:

$$\widehat{\mathbb{E}}\left[\mathcal{W}_{k+1}^{f_1} \,\middle|\, \widehat{\mathcal{W}}_{k+1,k+1}^{f_1}\right](i) = \frac{1}{c_i^{'}} \sum_{m=1}^{M^{'}} w_{k+1}^m \mathbb{1}_{\widehat{w}_{k+1,k+1}^m \in \Delta_i}. \quad (9)$$

where $c_i^{'}$ is the number of samples $\widehat{w}_{k+1,k+1}^m$ in Δ_i, and $M^{'}$ the total number of samples.

Our estimator of Equation 7 is then given by:

$$\widehat{\mathbb{E}}[\,\mathbb{E}[\mathcal{W}_k^{f_1} \,|\, \widehat{\mathcal{W}}_{k,k+1}^{f_1}]\mathbb{E}[\mathcal{W}_{k+1}^{f_1} \,|\, \widehat{\mathcal{W}}_{k+1,k+1}^{f_1}]] = \sum_{i=1}^{L} \frac{c_i}{M} \widehat{\mathbb{E}}\left[\mathcal{W}_k^{f_1} \,\middle|\, \widehat{\mathcal{W}}_{k,k+1}^{f_1}\right](i) \cdot \widehat{\mathbb{E}}\left[\mathcal{W}_{k+1}^{f_1} \,\middle|\, \widehat{\mathcal{W}}_{k+1,k+1}^{f_1}\right](i) \quad (10)$$

V. Numerical results

We now investigate the general accuracy of our proposed solution. To simulate the real behavior of a network, and thus getting the exact value for the variance of the end-to-end delays, we use the discrete-event simulator NS-3 [18]. This allows us to consider different network configurations (in terms of nodes, links, and buffers). The submitted traffic, namely the primary flow and the other flows, is replayed from a real trace that was collected from a campus at the university of Stuttgart (Germany) [19]. The trace contains around 44 million events, and corresponds to 4 hours of communications between 6 and 10 P.M. The packet size spreads a large range of values, from 64 to 1500 bytes.

Once the simulator has been setup and run, we compute the standard deviation of the end-to-end delay as found by our solution. The standard deviation is statistically equivalent to the variance ($\sigma = \sqrt{\mathbb{V}}$), and has the same unit as the end-to-end delay (in seconds). The computation is carried out on 100 consecutive packets. We repeat 1440 replications of the same experience using various parts of the trace in order to evaluate the statistical behavior of each solution. Note that, in order to evaluate the performance of our solution, we also run an Oracle solution. This latter collects the end-to-end delays of each of the 100 probe packets, and then derives the corresponding variance using the classical variance estimator. Note that this can be easily done in the simulator since the synchronization clock inside NS-3 is no longer an issue.

In addition to the results brought by our solution, we include those delivered by a trivial technique. The trivial method simply approximates the end-to-end variance by summing together the variance of the sojourn delays (waiting and transmission) at each node along the primary path. By nature, the trivial method neglects the covariance terms.

We consider two scenarios. Our first scenario, scenario A, deals with a network composed of 4 nodes and a total of two flows. Figure 2a depicts the corresponding topology. We set the buffer at each node to a length of 15,000 bytes. The transmission capacity of the links are fixed to $\mathcal{C}_{1,2}$=8, $\mathcal{C}_{2,3}$=5, $\mathcal{C}_{3,4}$=3 Mbps. The primary path is shared by a secondary flow that enters at node 2 and leaves at node 3. The secondary flow sends its packets at a mean rate of 2 Mbps. As for the primary flow, f_1, we consider three levels of sending rate, namely 0.5, 1 and 1.5 Mbps.

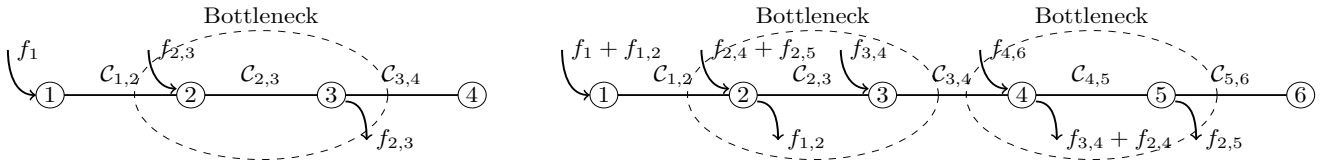
The results for the scenario A are presented in Figures 3a, 3b, 3c. It displays the cumulative distribution function of the relative error (in percents) committed by our solution and by the trivial solution. When the rate of the primary flow f_1 is set to 0.5 Mbps, Figure 3a shows that, in more than 95% of replications, our solution leads to a relative error for the variance of the end-to-end delay less than 10%. Using the trivial solution in these same conditions, only around 15% of replications have an error below 10%. Figures 3b and 3c report the corresponding results when the primary flow f_1 is increased to 1 and 1.5 Mbps, respectively. Although the values slightly differ, the overall behavior of the proposed solution and the trivial remains roughly the same.

We now turn to scenario B, illustrated by Figure 2b. Here, the network comprises 6 nodes and a larger number of flows. The transmission capacities of the links are as follows: $\mathcal{C}_{1,2}$=6, $\mathcal{C}_{2,3}$=4, $\mathcal{C}_{3,4}$=7, $\mathcal{C}_{4,5}$=3, $\mathcal{C}_{5,6}$=5 Mbps. As for the multiple secondary flows, the secondary flows $f_{1,2}, f_{2,4}, f_{2,5}, f_{3,4}, f_{4,6}$ have sending rate of 1, 0.8, 0.2, 2 and 0.2 Mbps, respectively. Note that, under these conditions, we ensure that the incoming rate at each link remains below 70% of its capacity. Again, we consider three levels of sending rate for the primary flow, namely 0.5, 1 and 1.5 Mbps.

Figures 3d, 3e, 3f show the results found for scenario B. Broadly speaking, the results are generally poorer than in scenario A. For example, with a primary flow of rate 1 Mbps, 80% (compared to 95%) of replications lead to a relative error on the variance of the end-to-end delay less than 10%. Nonetheless, the proposed solution still significantly outperforms the trivial solution. Also, we notice that as the intensity of the sending rate of f_1 increases, the accuracy of our solution decreases. We believe that stems from the

Table I: Overall distribution of the relative error on the standard deviation of the end-to-end delay.

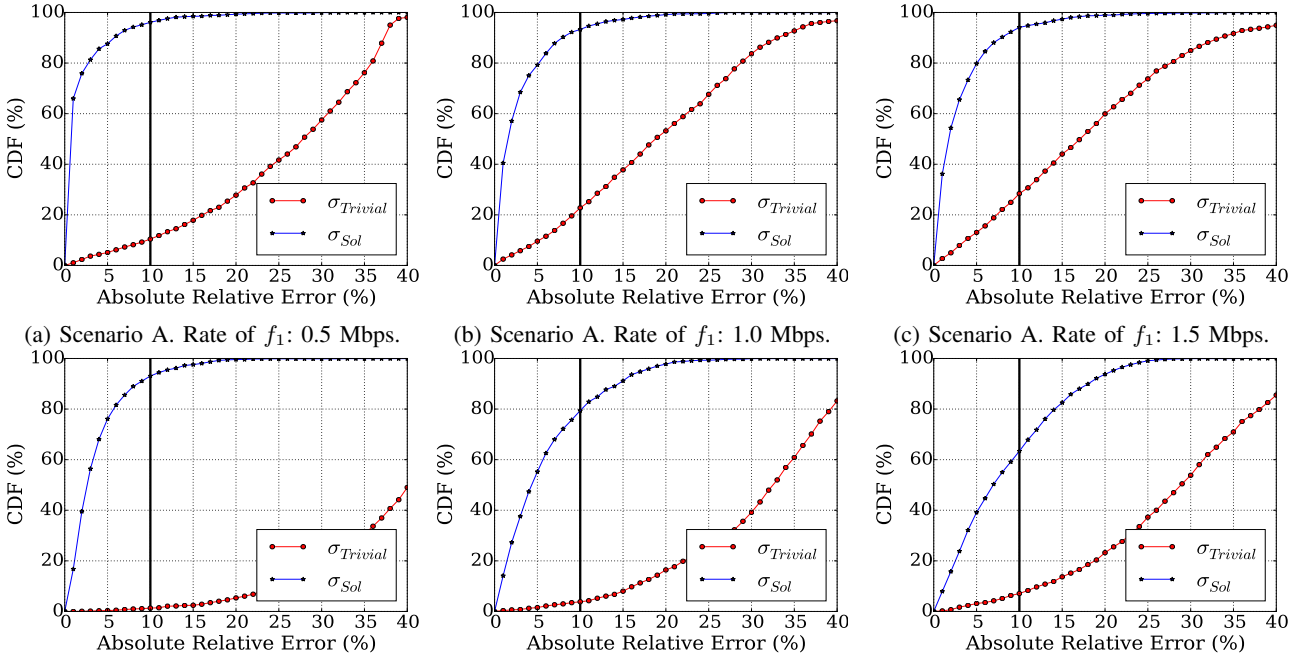
	Rate of f_1: 0.5 Mbps				Rate of f_1: 1 Mbps				Rate of f_1: 1.5 Mbps			
Error	$\leq 10\%$	$\leq 20\%$	$> 30\%$	Mean	$\leq 10\%$	$\leq 20\%$	$> 30\%$	Mean	$\leq 10\%$	$\leq 20\%$	$> 30\%$	Mean
$\sigma_{sol-scen-A}$	**96.**11	**99.**31	**0.**07	**1.**84	**93.**26	**99.**17	**0.**07	**3.**05	**94.**01	**98.**96	**0.**21	**3.**19
$\sigma_{sol-scen-B}$	**93.**06	**99.**58	**0.**14	**3.**80	**79.**40	**97.**77	**0.**07	**6.**02	**63.**37	**93.**75	**0.**07	**8.**47



(a) Small study case with one bottleneck.

(b) Bigger study case with two bottlenecks.

Figure 2: The primary flow f_1 workload is set to 0.5, 1.0 and 1.5 (Mbps). In Figure 2a, the utilization rates at the first bottleneck are 50%, 60% and 70%. In Figure 2b, the utilization rates at the first bottleneck are 37%, 50% and 63%, at the second bottleneck are 32%, 50% and 66%.



(a) Scenario A. Rate of f_1: 0.5 Mbps.

(b) Scenario A. Rate of f_1: 1.0 Mbps.

(c) Scenario A. Rate of f_1: 1.5 Mbps.

(d) Scenario B. Rate of f_1: 0.5 Mbps.

(e) Scenario B. Rate of f_1: 1.0 Mbps.

(f) Scenario B. Rate of f_1: 1.5 Mbps.

Figure 3: Cumulative distribution function of the relative errors for the standard deviation of the end-to-end delay for different workloads of the primary flow f_1.

increasing number of packet losses that disrupt our method. Interestingly, the trivial methods tends to enhance as the rate of f_1 grows, but stays far less accurate than the proposed solution. We believe that this improvement occurs because at higher level of loads, the queuing delays become less variable and so their correlation tends to decrease.

Finally, Table I summarizes the results obtained in these two scenarios by our solution. We observe that in the many studied examples, regardless of the actual value of sending rate for f_1, the relative errors on the estimation of the variance of the end-to-end delay remains below 10% in over 90% of the replications explored. The mean relative error of our method tends to peak when the traffic load is high (with a corresponding relative error around 10%), and in general, delivers accurate results with a mean relative error less than or close to 5% in most cases.

VI. Conclusion

With the development of SDN, computer networks aim at gaining a better control on the management of their resources. This may lead to cost savings and to a better handling of flows requiring a specific type of QoS. However, to fulfil these objectives, networks must be able to characterize the end-to-end performance of flows. Unfortunately, computer networks are usually not equipped with tools that allow direct end-to-end measurements. Instead, networks have typically at their

disposal only local measurements, collected at each node. Although evaluating the mean value of a path metric is usually rather simple, things are becoming much more complex for higher-order moments.

In this paper, we propose a new method to evaluate the variance of the end-to-end delay based only on measurements collected on nodes. The core of our solution is to link samples of waiting times at different nodes in order to approximate the corresponding covariance terms, needed to refine the approximate value of the sought variance. We evaluate the accuracy of our solution using two different scenarios. The obtained results show that the method is generally good, with a relative error on the estimated standard deviation of the end-to-end delay usually close to 5%, though it may exceed 10% when the network is facing high levels of load and experiencing packet losses.

ACKNOWLEDGMENT

This work is funded by the French National Research Agency ANR INFRA DISCO under the "ANR-13-INFR-013" project.

REFERENCES

[1] G. Almes, S. Kalidindi, and M. Zekauskas, "A Round-trip Delay Metric for IPPM - RFC 2681," Tech. Rep., Sep. 1999.

[2] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP) - RFC 5357," IETF, Tech. Rep., Oct. 2008.

[3] V. Paxson, "End-to-end routing behavior in the internet," *Networking, IEEE/ACM Transactions on*, vol. 5, no. 5, pp. 601–615, Oct 1997.

[4] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP) - RFC 4656," IETF, Tech. Rep., Sep. 2006.

[5] L. Mark, G. Pohl, T. Zseby, and K. Sugauchi, "Passive One-way Delay Measurement - RFC 2026," IETF, Tech. Rep., Jun. 2006.

[6] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM, 2002, pp. 5–18.

[7] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Van Mieghem, "Analysis of end-to-end delay measurements in internet," in *Proc. of the Passive and Active Measurement Workshop-PAM*, vol. 2002, 2002.

[8] M. J. Luckie, A. J. McGregor, and H.-W. Braun, "Towards improving packet probing techniques," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. ACM, 2001, pp. 145–150.

[9] O. Gurewitz, I. Cidon, and M. Sidi, "One-way delay estimation using network-wide measurements," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2710–2724, 2006.

[10] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end internet path performance," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 375–386, 2006.

[11] A. Pasztor and D. Veitch, "The packet size dependence of packet pair like methods," in *Quality of Service, 2002. Tenth IEEE International Workshop on*, 2002, pp. 204–213.

[12] V. J. Ribeiro, J. Navratil, R. H. Riedi, R. G. Baraniuk, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Presented at*, no. SLAC-PUB-9732, 2003.

[13] A. Johnsson and M. Björkman, "Measuring the Impact of Active Probing on TCP," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2006.

[14] S. B. Moon, "Measurement and Analysis of End-to-end Delay and Loss in the Internet," Ph.D. dissertation, University of Massachusetts at Amherst, 2000.

[15] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Delay Metric for IPPM," IETF, RFC 2679, 1999.

[16] J. Wang, M. Zhou, and Y. Li, "Survey on the End-to-End internet Delay Measurements," in *HSNMC*, 2004.

[17] N. Nguyen, T. Begin, A. Busson, and I. Guerin-Lassous, "Towards a passive measurement-based estimator for the standard deviation of the end-to-end delay," in *NOMS 2016 ()*, Istambul, Turkey, apr 2015.

[18] T. R. Henderson *et al.*, "Network Simulations with the ns-3 Simulator," ACM SIGCOMM Demos, 2008, code available: http://www.nsnam.org/releases/ns-3.1.tar.bz2.

[19] D. Sass, "The dormitory network "Selfnet" of the University of Stuttgart," Oct. 2004.