

Measurement-Based Network Link Dimensioning

Ricardo de O. Schmidt¹, Hans van den Berg^{1,2} and Aiko Pras¹

¹ University of Twente, Enschede, The Netherlands

² TNO Information and Communication Technology, Delft, The Netherlands

Email: {r.schmidt, j.l.vandenberg, a.pras}@utwente.nl

Abstract—The ever increasing traffic demands and the current trend of network and services virtualization calls for effective approaches for optimal use of network resources. In the future Internet multiple virtual networks will coexist on top of the same physical infrastructure, and these will compete for bandwidth resources. Link dimensioning can support fair share and allocation of bandwidth. Current approaches however, are ineffective at smaller timescales or require traffic measurements that are not easy to obtain. In this thesis we focused on easy to deploy and accurate link dimensioning approaches for the future Internet. The start point of our work is a dimensioning formula, proposed in 2006, built upon the assumption of Gaussian traffic. This formula is able to accurately estimate required capacity at very small timescales. To do so it requires traffic statistics that can be obtained from packet captures. The contribution of this thesis is threefold. First, we prove that the assumption of Gaussian traffic holds for current Internet traffic and, hence, the dimensioning formula can still be applied. Second, instead of relying on costly packet captures, we develop and validate link dimensioning approaches that estimate the needed traffic statistics from measurement data obtained via technologies that are largely found in today’s networks (namely, sFlow and NetFlow/IPFIX). Our approaches are able to accurately estimate required capacity at timescales as low as 1ms. Last, we propose a link dimensioning approach that uses measured data from the recent and already widely available OpenFlow. We also investigate the quality of flow-level measurements in current implementations of OpenFlow, and demonstrate that these are not yet accurate enough for link dimensioning purposes.

Index Terms—Link dimensioning, bandwidth estimation, Gaussian traffic, NetFlow, IPFIX, sFlow, OpenFlow.

I. INTRODUCTION

A significant increase on traffic demands has been observed in the past decade¹. Although the amount of bandwidth resources will likely not become a problem in the future Internet, the current trend on virtualizing services and networks will add complexity to the management of such resources. This will ultimately call for more sophisticated approaches to fairly share and allocate available bandwidth resources.

In the future Internet we expect scenarios in which network operators will still own and control most of the physical infrastructure, but end users will be directly connected to companies that control essential services and retain users’ content. These companies are often called the Internet *big players* [12]. Virtual networks will enable transparent and seamless connection between end users and big players. The

added complexity on managing bandwidth resources will rise from the coexistence of many virtual networks on top of a single physical infrastructure. Efficient and accurate link dimensioning approaches can certainly make the difference on the proper management of bandwidth resources. Such approaches can (i) support operators on the optimal allocation of their bandwidth resources, while (ii) helping to meet the Quality of Service (QoS) metrics agreed with the big players, ultimately (iii) providing end users with good levels of Quality of Experience (QoE).

A. Link dimensioning background

To accommodate increasing traffic demands, operators typically over-provision their networks. A common approach to do so is to read interface counters via SNMP every 5 to 15 minutes and calculate the average bandwidth utilization [13]. A safety margin is then added to the bandwidth utilization. Usually, as a rule of thumb, this margin is defined by a percentage of the bandwidth utilization [14]. The main drawback of this whole approach is that traffic fluctuations that happen at much smaller timescales (*e.g.*, seconds or fractions of seconds) are averaged within too large time bins, and ultimately, network performance and user experience degrades due to overlooked short-term traffic bursts.

Aiming at higher accuracy at shorter timescales, alternative approaches have been proposed to properly dimension network links. However, the higher accuracy of these link dimensioning approaches often comes at the cost of higher demands on efforts for measuring traffic. For example, the work in [15] proposes a link dimensioning formula that requires traffic statistics (*i.e.*, traffic variance) usually calculated from packet-level measurements. Although very accurate, approaches that required packet captures are not adopted by network operators mostly because today’s traffic rates make packet capturing operationally and financially (almost) unfeasible. Therefore, operators stick to the easy-to-use, though not reliable, SNMP-based rules of thumb.

The main goal of this PhD thesis [1] was to develop approaches for link dimensioning that are – almost – as easy-to-use as SNMP-based rules of thumb, and – almost – as accurate as packet-based approaches. Next, we briefly introduce the dimensioning formula we use in our approaches and then we set out the contributions of this thesis.

¹<https://ams-ix.net/technical/statistics/>

B. Link dimensioning formula

The starting point of our work is the dimensioning approach proposed in [15] and further validated in [16], [17]. Aiming at link transparency, this approach aims at assuring that the provided link capacity C satisfies $\mathbb{P}\{A(T) \geq CT\} \leq \varepsilon$, where $A(T)$ denotes the total amount of traffic arriving in intervals of length T , and ε indicates the probability that the traffic rate $A(T)/T$ is exceeding C at the timescale T . In [15] a dimensioning formula is provided that requires that traffic aggregates are Gaussian (*i.e.*, $A(T)$ are normally distributed) and stationary. The link capacity $C(T, \varepsilon)$ needed to satisfy the condition above can be calculated by

$$C(T, \varepsilon) = \rho + \frac{1}{T} \sqrt{-2 \log(\varepsilon) \cdot v(T)}, \quad (1)$$

i.e., the mean traffic rate ρ is added by a safety margin that depends on the variance $v(T)$ of $A(T)$. Relying on the variance $v(T)$ this dimensioning formula is able to take into account the impact of possible traffic bursts on the required link capacity. In addition, it is very flexible: network operators can choose T and ε according to the QoS level they want to provide to their customers. Although accurate, the dimensioning approach from [15] requires continuous packet capture to calculate ρ and $v(T)$. To eliminate the need of packet capture, we developed approaches to estimate these statistics from measurement data largely found at today's networks, namely sampled packets and flows. However, such data only provides a summary or an aggregated view of the actual traffic. One of the challenges in this thesis was, therefore, to properly estimate ρ and $v(T)$ from coarser measurement data than continuous packet capture.

C. Contributions

The contribution of this thesis [1] can be divided in three parts. In our first contribution we extensively assessed the Gaussian character of current Internet traffic [2], [3]. Previous works assessing traffic Gaussianity relied on old datasets, and we believe that the advent and widely adoption of recent online services (*e.g.*, social networking, video streaming and online storage) have changed the behavior of Internet users and potentially reshaped important traffic characteristics. Among other important findings, in this study we proved current traffic is Gaussian and, hence, the dimensioning formula above can still be applied.

Given the inaccuracy of the SNMP-based rule of thumb approach, and the requirement for packet capture of the packet-based approach from [15], in the second contribution of this thesis we aimed at developing link dimensioning approaches using largely available measurement technologies. In particular, we developed and validated approaches to estimate ρ and $v(T)$ for Eq. (1) using sampled packets from sFlow [4] and flow-level data from NetFlow/IPFIX [5], [6]. These approaches estimate required capacity with much higher accuracy than SNMP-based rules of thumb, even at timescales as low as 1ms. In addition, our approaches use measurement data widely

TABLE I
MEASUREMENT DATASET

abbr.	length	# of hosts	link capacity	avg. use
A	24h	6.5k	2×1 Gb/s	15%
B	6h	886k	10 Gb/s	10%
C	84h45min	10.5k	155 and 40 Mb/s	19%
D	4h	1.8M	2×10 Gb/s	8%
E	5h	3M	2×10 Gb/s	10%
F	13h15min	4M	n/a	n/a

available at operators, what makes them more easy-to-use than packet-based approaches such as [15].

Nowadays, OpenFlow is gaining lots of interest as being the best known enabler of SDN (Software-Defined Networking) architectures. Although OpenFlow primary task is packet forwarding, in theory it can also measure traffic at the flow level (NetFlow/IPFIX style). Given the increasing number of OpenFlow-enabled network devices, in the third contribution of this thesis we proposed an approach that uses the OpenFlow protocol to retrieve flow data measured at OpenFlow switches [7]. This data can later be applied to one of the flow-based approaches for link dimensioning we proposed. However, in practice the flow data from OpenFlow lacks accuracy and might not be reliable. Therefore, we have also assessed the quality of per-flow data obtained from current implementations of the OpenFlow protocol. We show that, right now, data inaccuracies prevent its use for link dimensioning purposes.

D. Organization

The remainder of this paper is organized as follows. In Section II we describe the measurement dataset used to validate our proposed approaches, and we present results of an extensive assessment of the Gaussian fit of current network traffic. In Section III we present our proposed link dimensioning approaches, as well as a quantitative assessment of these approaches. In Section IV, we describe our OpenFlow-based approach to retrieve flow data from switches, and show pitfalls we have found on the quality of the measured data by OpenFlow. Finally, in Section V we summarize our work.

II. NETWORK TRAFFIC DATASET

A. Description of Dataset

Our measurement dataset, summarized in Table I, consists of 548 15-minute packet traces, captured between 2011 to 2012 at six different locations around the globe. The trace duration of 15 minutes has been chosen in accordance with [16], [17]. Longer time periods are generally not stationary due to the diurnal pattern. The packet traces allowed for the reproducibility of experiments and for the comparison of different link dimensioning approaches with the exact same input traffic. Traffic from locations *A*, *B* and *C* were collected by us, from a link between a university building to the university's gateway (*A*), and from the gateway of two universities (*B* and *C*). The other three locations, *D*, *E*, and *F*, comprise traces from ISP backbone links available at the public repositories

of CAIDA [18] (*D* and *E*) and MAWI [19] (*F*). For a more detailed description of traffic characteristics from each location, please refer to [2].

B. Traffic Gaussianity

The advent of many online services, *e.g.*, Facebook, Drop-Box, YouTube and Netflix, has changed users behavior, what potentially reshaped characteristics of Internet traffic. Important to us is the Gaussian fit of traffic, since it is a major requirement from Eq. (1). Traffic Gaussian fit was addressed by few works in the past, such as [20], [21]. These works relied on traffic data measured relatively long ago, even before the aforementioned applications and services became highly popular in the Internet. Therefore, it was important to once again assess the traffic Gaussian fit, validating the use of the dimensioning formula of Eq. (1) with current traffic. We comprehensively studied the Gaussian fit for all traces in our dataset, and our results were published in [2], [3].

We wanted to know if $A(T) \sim \text{Norm}(\rho, v(T))$, where ρ is the mean traffic and $v(T)$ the traffic variance at timescale T . To assess traffic Gaussianity for all traces in our dataset we used the *linear correlation coefficient* [22] defined by

$$\gamma(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (2)$$

where x is the inverse of the normal cumulative distribution function of the sample, and y is the ordered sample, *i.e.*, $A(T)$. A $\gamma \geq 0.9$ supports the hypothesis that the underlying distribution is normal, which corresponds to a Kolmogorov-Smirnov test for normality at significance 0.05 [21].

Fig. 1 shows the CDF of γ for all traces in our dataset at different timescales. For $T = 1s$, around 84% of all traces are at least “fairly Gaussian”, *i.e.*, $\gamma > 0.9$. Location *A* is a 24-hour measurement and around 50% of its traces have $\gamma < 0.9$, from which most measured overnight when less hosts are active in the network, resulting in a lower traffic aggregate and, consequently, lower Gaussian fit. Around 90% of traces from the other 5 locations have $\gamma \geq 0.9$. At locations with larger aggregates, such as *D*, (almost) all the traces have $\gamma \geq 0.9$. Note that the only significant difference at $T = 100ms$ is for traces from *C*. In [2] we proved that the Gaussian fit persists at timescales from 1ms to 30s. At very small timescales, however, T approximates the packets transmission intervals, resulting in a binary-like behavior (have or not packet), which is not Gaussian. But also at very large T , few bins might average traffic from nearby bursts, resulting in a much higher rate than the trace average rate, which ultimately disturbs Gaussian fit.

We also demonstrate that for high-speed links it is safer to relate Gaussian fit to average rate than to the number of simultaneously active hosts, as previously suggested in [20], [21]. Fig. 2 compares the γ for each trace from location *A* at $T = 1s$, with the respective trace average rate and average number of simultaneously active hosts per second. Traces from 23:00 to 00:15 have a good Gaussian fit even though the number of active hosts remains roughly the same as for the non-Gaussian traces in the overnight. The average rate of

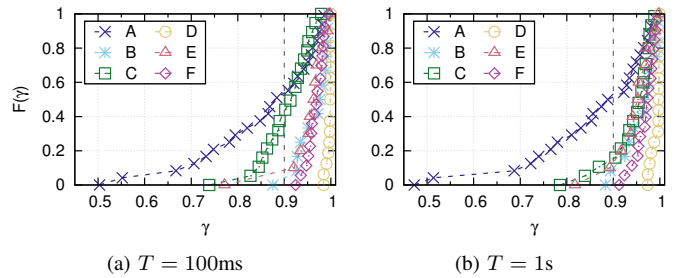


Fig. 1. CDF of the gaussianity fit γ for all traces in our dataset.

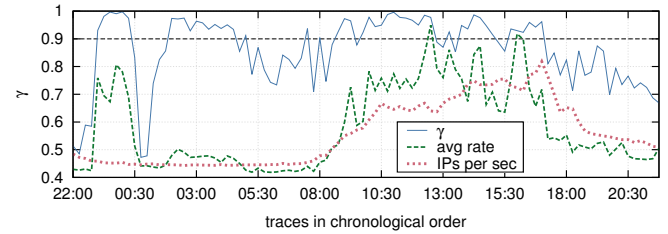


Fig. 2. Gaussianity fit γ per trace compared to the trace average traffic rate and the average number of simultaneously active IPs per second. X-axis shows all 96 traces from location *A* chronologically ordered. Notice that γ -axis shows γ value only.

the Gaussian traces is, however, much higher than the one of the other traces. That is because the behavior of very few hosts, using a handful of applications, can significantly affect Gaussian fit, as we proved in the follow-up work [3].

III. MEASUREMENT-BASED LINK DIMENSIONING

In this section we describe the developed link dimensioning approaches and present results of their quantitative assessment.

A. sFlow-based Approach

Sampled packets provide a partial view of the actual traffic in a link. Our developed approaches compensate for the missing information when estimating the statistics needed by the Eq. (1). Average rate ρ can be easily estimated by multiplying the measured traffic rate from sampled packets by the sampling rate. Estimating traffic variance $v(T)$, however, is not straightforward due to additional variance introduced by the sampling process, which might lead to undesired results [4]. In [8] we developed approaches to estimate $v(T)$ from sampled data obtained using one of the three sampling methods: Bernoulli or n -in- N , which are defined in [23], or the specific sampling method implemented by the widely available sFlow [24]. We validated these proposed approaches with the traces from our dataset sampled at various rates. The sampling algorithms implemented by us are available in [9].

B. NetFlow/IPFIX-based Approaches

Flow data, such as NetFlow [25] or IPFIX [26] flows (or equivalents such as J-Flow [27]) give an aggregate view of the actual traffic. With flow data we miss information on individual packets that are essential for calculating $v(T)$ needed by Eq. (1). For example, we know duration and number

of packets and bytes of a flow. We do not know, however, the size and time of individual packets and how packets are distributed throughout the flow duration. To overcome this problem we developed two approaches based on NetFlow v5, *i.e.*, flows defined with the 5-tuple key: source and destination IP addresses, source and destination ports and transport protocol. NetFlow v5 is widely available in network devices and we can obtain the same flow data from newer versions of NetFlow or from IPFIX-based probes.

In [5] we developed a pure flow-based link dimensioning approach that solely uses flow data to estimate required capacity. Under the optimistic assumption that packets are of constant size and uniformly distributed within their respective flows, this approach builds a flow-level time series from the flow data. Traffic variance is then calculated from this time series and finally applied to Eq. (1). This straightforward approach is able to accurately estimate required capacity from flows mostly at second timescales but not at millisecond timescales. The algorithm that implements the creation of the flow-level time series is available in [10].

In [6] we developed a flow-based approach able to accurately estimate required capacity at millisecond timescales. This is a hybrid approach that combines flow data with mathematical models that model the behavior of individual packets within flows. The accuracy at very small timescales of this approach comes at the cost of requiring occasional and short packet captures for parameters tuning. However, we showed that the tuned parameters remain valid for very long periods (up to months), making the hybrid approach measurement-wise lightweight as compared to the packet-based approach from [15].

C. Quantitative Assessment

In this section we present a summary of the results from the PhD thesis [1] on the validation of the proposed approaches. For the following experiments we converted the packet traces from our dataset into sampled packets and flows. We sampled all original traces using sFlow method with a rate 1:10, and we also converted all original traces into flows (NetFlow v5) using YAF [28] with active and inactive timeouts set to 60s and 20s resp. In the following, C_{sFlow} refers to estimations using the sFlow-based approach, $C_{pureFlow}$ refers to the pure flow-based approach, and C_{hybrid} refers to the hybrid flow-based approach. In the dimensioning formula we always set $\varepsilon = 0.01$ and T from 1ms to 1s. These values also comply with previous works [16], [17]. In [1] we present results with variations on the parameters of sampling methods, flows creation and the dimensioning formula.

We validated the proposed approaches against an empirically defined ground-truth $C_{emp}(T, \varepsilon)$, which is the $(1 - \varepsilon)$ -quantile of the empirical CDF of the aggregated data rate calculated from the packet trace. That is, $C_{emp}(T, \varepsilon)$ is the minimum capacity such that the fraction of time intervals of size T in which rate is higher than this capacity is exactly ε :

$$C_{emp}(T, \varepsilon) := \min \{C : \#\{A_i(T) \mid A_i(T) > CT\}/n \leq \varepsilon\}, \quad (3)$$

where $A_1(T), \dots, A_n(T)$ are the n empirical traffic aggregates on timescale T , and ε is the bandwidth exceedance probability. Fig. 3 shows the data rate time series at various T for an example trace from our dataset, and the estimations of required capacity using the different approaches. This figure clearly shows that at very small T the pure flow-based approach is inaccurate. From this figure it also becomes clear how the mathematical models supported the hybrid flow-based approach on estimations at smaller T . The sFlow-based approach did not underestimate required capacity at any T for this example trace. At large T all approaches succeeded in estimating required capacity as compared to C_{emp} .

To verify the accuracy of the estimated required capacity for the whole dataset, we calculate for each trace the fraction of measured intervals in which the traffic aggregate $A_i(T)$ exceeds $C(T, \varepsilon)$:

$$\hat{\varepsilon} := \#\{A_i(T) \mid A_i(T) > C(T, \varepsilon)T\}/n. \quad (4)$$

Note that $\hat{\varepsilon} \leq \varepsilon$ is equivalent to $C(T, \varepsilon) \geq C_{emp}(T, \varepsilon)$.

Fig. 4 shows the average and standard deviation (error bars) of $\hat{\varepsilon}$ for all traces per location in our dataset. The dashed line at $\hat{\varepsilon} = 0.01$ in the plots of this figure represent the optimal situation, in which required capacity was neither underestimated nor excessively overestimated. From the plots in this figure, it is clear how the sFlow-based approach (Fig. 4a) is more stable than the other two flow-based approaches through timescales ranging from 1ms to 1s. Although using sampled packets as input, these are still more granular data than flows. Once again the difference between the two flow-based approaches is very clear from the obtained results at different values of T . While for the pure flow-based approach (Fig. 4b) estimations of required capacity at smaller T result in high underestimation (*i.e.*, $\hat{\varepsilon} > \varepsilon$), the estimations from the hybrid flow-based approach (Fig. 4c) result in very low (if any) underestimation. The traffic variance at larger T depends more on flow dynamics than on packet dynamics. This way, results for C_{hybrid} are very similar to those for $C_{pureFlow}$ at $T = 1s$ (note the different scale of y-axis in plots of Fig. 4b and 4c). On having NetFlow-like data, one can use the simpler pure flow-based approach for link dimensioning at larger T while the more complex hybrid approach at smaller T .

From $\hat{\varepsilon}$ we can pinpoint cases of underestimation of required capacity. However, when $\hat{\varepsilon}$ is too low, *e.g.*, location A at 1ms in Fig. 4c, it might be that the link dimensioning approach excessively overestimated the actual required capacity. To quantify a possible excessive overestimation of the required capacity, we calculate the relative error, in percentage, between the estimated required capacity $C(T, \varepsilon)$ – using any of the proposed approaches – and $C_{emp}(T, \varepsilon)$. The relative error RE is, therefore, given by

$$RE = \frac{C(T, \varepsilon) - C_{emp}(T, \varepsilon)}{C_{emp}(T, \varepsilon)} \cdot 100\%. \quad (5)$$

Fig. 5 shows the obtained RE for all traces per location in our dataset for estimations of required capacity at $T = 1s$. Once again it is clear how the sFlow-based approach (Fig. 5a)

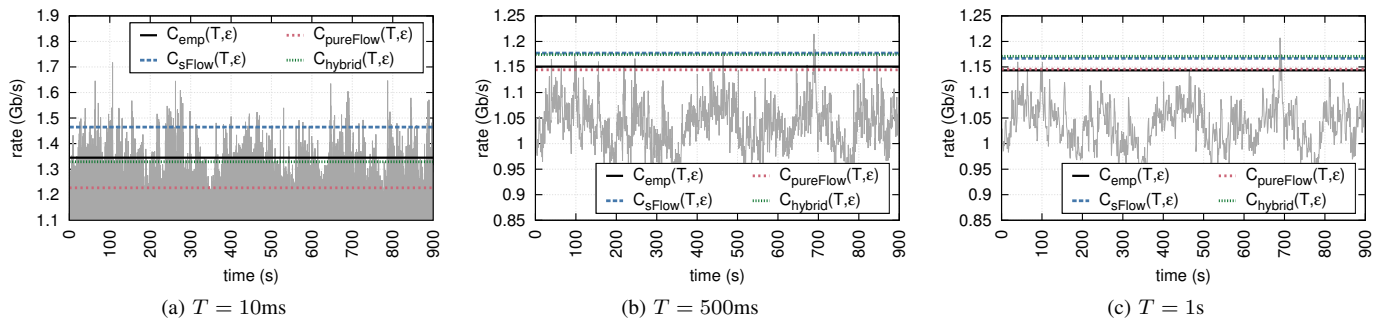


Fig. 3. Estimation of required capacity for each of the proposed approaches, at different T , using a sample traffic trace from our dataset. At any of the considered values of T , this sample trace has $\gamma > 0.9$ (*i.e.*, the traffic is sufficiently Gaussian).

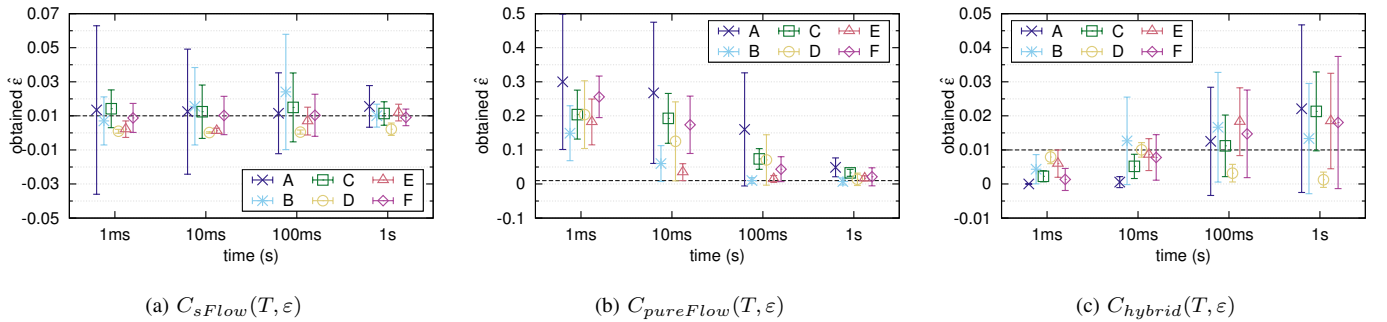


Fig. 4. Average and standard deviation (error bars) of $\hat{\varepsilon}$ for all traces in our dataset (Table I) at various T for each proposed approach for link dimensioning. Notice the different scale of the y-axis for the three plots.

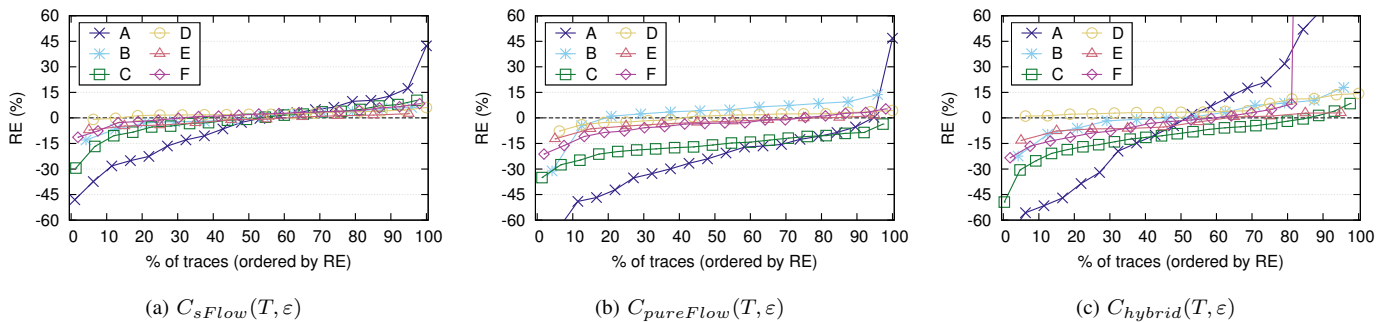


Fig. 5. RE for all traces in our dataset using each of the proposed link dimensioning approaches at $T = 1s$; ε set to 0.01.

gives more stable results. With this approach, for most traces $-15\% \leq RE \leq 15\%$. Also for the pure flow-based approach (Fig. 5b), most traces are within the same RE limits but traces from A and C . The worst results on RE observed for both sFlow-based and pure flow-based approaches in Fig. 5a and 5b, respectively, are mainly due to the smaller traffic aggregates of traces from locations A and C . For the sFlow-based approach, the smaller the traffic aggregate, the smaller the amount of sampled data from which the estimation of required capacity is calculated. This means that for better estimations for such traces using the sFlow-based approach, one must sample traffic with higher sampling rates than 1:10 (*e.g.*, 1:5). For the pure flow-based approach, the excessive under and overestimation can be eased off by reducing the values of timeouts in the flow metering/exporting process (*i.e.*, active and inactive timeouts). As demonstrated by us in [1], [5], although demanding more

measurement effort, shorter timeouts help the approach to better reconstruct short-term traffic fluctuations, ultimately yielding more accurate estimations even at $T < 1s$. Once again one can see in Fig. 5c that the hybrid flow-based approach has a similar performance than the pure flow-based one at larger T . Cases of excessive overestimation in the hybrid approach might happen when, *e.g.*, the model's parameters are fitted using traces with non-Gaussian traffic. Gaussianity is an important requirement for all three proposed approaches, which is actually inherited from the dimensioning formula we use. In the hybrid approach, however, once the model's parameters are fitted using a non-Gaussian trace, the estimation of required capacity for all consecutive traces is compromised. The fact that locations A and C have many non-Gaussian traces (Fig. 1) explains the wider range of RE for these two locations in Fig. 5.

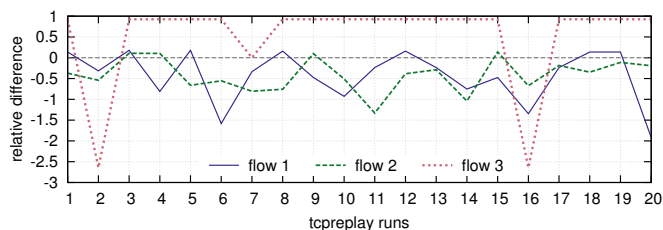


Fig. 6. Number of measured packets by OVS each time the same flow was sent through the switch in the virtual experimental setup.

IV. OPENFLOW-BASED APPROACH

Motivated by the increasing interest from industry and academia on OpenFlow [29], as the third contribution of this thesis we investigated the feasibility of using measurement data from OpenFlow for link dimensioning. Since its earliest specifications, OpenFlow allows for measuring traffic in a NetFlow/IPFIX fashion, *i.e.*, flow level. In this section we describe our proposed approach that uses OpenFlow to retrieve flow data measured at the switch, and we show results of the quality assessment of the measured data from current OpenFlow implementations.

In [7] we propose an OpenFlow-based approach that runs on top of the OpenFlow controller to retrieve flow data from the OpenFlow switch. Since we defined OpenFlow flows as the same of NetFlow/IPFIX, we do not propose new ways to calculate statistics needed by Eq. (1). Instead, we propose to use OpenFlow flows as input to one of the flow-based approaches from Section III-B. Our OpenFlow-based approach implements a mix of passive and active operations and it solely uses messages defined within the OpenFlow protocol to retrieve flow data (*e.g.*, duration and packet/bytes counters). The passive portion of the approach asks the OpenFlow switch to report statistics of terminated flows, due to timeout expiration, using *flow removed* messages. The received data is stored until a new estimation is to be calculated. To ensure that all measured data for the current period is considered when estimating required capacity, the active portion of our approach uses *stats request* messages to “force” the switch to report on currently active flows.

Although definitions of OpenFlow [29] allow for our proposed approach to work, we have identified several pitfalls on current OpenFlow implementations that affect the quality of measured data. In [1] we present results from experiments using a virtual and a physical OpenFlow-enabled network. We implemented the proposed approach on top of Ryu OpenFlow controller [30] using OpenFlow version 1.3 (implementation available in [11]), and we ran experiments to assess the quality of measured data in: (a) a virtual setup running Open vSwitch 2.1.2 (OVS) [31], likely the most popular OpenFlow implementation; and (b) a physical setup using Pica8 P3592 OpenFlow switch, running PicOS 2.3, which is based on OVS. In both cases we observed serious inaccuracies on the measured data that actually invalidate its use by applications that rely on flow data, including link dimensioning.

For instance, in one of our experiments we sent three different flow 20 times (*i.e.*, many experiment runs) through the OpenFlow switch and retrieved the accounted number of packets of each run. Fig. 6 shows the relative different between the number of sent and measured packets by OVS in the 20 experiment runs. In this figure, if the relative difference is zero, OVS correctly measured the number of packets. If it is negative, OVS measured less packets, and if positive more packets than actually sent. Although the number of forwarded packets by the switch was correct in all runs, as observed in the sink machine to which packets were routed, OVS surprisingly reported more or less measured packets than the actual number of sent packets. For the case of PicOS this problem is even more serious. Besides inaccurate number of measured bytes, PicOS does not implement an actual packet counter. Instead, it reports the total of measured bytes divided by a constant 100 as being the number of measured bytes. This ultimately yields completely unrealistic numbers, preventing the measured data to be used by any application. Note that results in Fig. 6 reported results of OpenFlow measurements for a single flow. As expected, when submitting OVS and PicOS to much higher traffic aggregates, consisting of several thousands of flows, inaccuracies become even worse. Our findings led us to conclude that these inaccuracies mostly result from implementation decisions and from the low priority given to measurement operations at the switch.

V. SUMMARY

In the future Internet link dimensioning will support fair share and allocation of link resources. In this thesis we developed link dimensioning approaches that are (almost) as easy-to-use as SNMP-based rules of thumb for over-provisioning, and (almost) as accurate as packet-based approaches.

In this thesis we performed an extensive study on the Gaussian character of current network traffic, from which we conclude that the dimensioning formula from Eq. (1) [15] can still be applied to today’s traffic. We also demonstrate that it is safer to relate Gaussianity assumption to the measured traffic rates than to the number of active hosts in a network, as suggested by previous works. In addition, we show that the behavior of few hosts can compromise Gaussianity fit of an aggregate comprising traffic of several thousands of hosts. We developed and validated three fully operational link dimensioning approaches. These approaches are able to estimate required capacity at timescales as low as 1ms from measurement data obtained via sFlow and NetFlow/IPFIX. It is important to mention that both sFlow and NetFlow/IPFIX are largely found nowadays at network operators infrastructure and, hence, our proposed approaches are ready to use. We have also proposed to use OpenFlow to retrieve flow-level measurement data from switches for link dimensioning purposes. We demonstrate, however, that current implementations of OpenFlow provide measurements of poor quality that, as for now, prevent their use for link dimensioning.

ACKNOWLEDGEMENTS

The authors would like to thank Ramin Sadre and Anna Sperotto for their valuable contributions to this thesis. This PhD thesis was partially supported by EU FP7 UniverSelf (257513), EU FP7 FLAMINGO NoE (ICT-318488), EU FP7 MCN (318109) and SURFnet Gigaport3 projects.

RELEVANT PUBLICATIONS

- [1] R. de O. Schmidt, "Measurement-based Link Dimensioning for the Future Internet," Ph.D. dissertation, University of Twente, 2014.
- [2] R. de O. Schmidt, R. Sadre, and A. Pras, "Gaussian Traffic Revisited," in *Proceedings of the IFIP Networking Conference*, 2013, pp. 1–9.
- [3] R. de O. Schmidt, R. Sadre, N. Melnikov, J. Schönwälder, and A. Pras, "Linking Network Usage Patterns to Traffic Gaussianity Fit," in *Proceedings of the IFIP Networking Conference*, 2014, pp. 1–9.
- [4] R. de O. Schmidt, R. Sadre, A. Sperotto, and A. Pras, "Lightweight Link Dimensioning using sFlow Sampling," in *Proceedings of the 9th International Conference on Network and Services Management (CNSM)*, 2013, pp. 152–155.
- [5] R. de O. Schmidt, A. Sperotto, R. Sadre, and A. Pras, "Towards Bandwidth Estimation using Flow Measurements," in *Proceedings of the 6th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security (AIMS)*, 2012, pp. 127–138.
- [6] R. de O. Schmidt, R. Sadre, A. Sperotto, H. van den Berg, and A. Pras, "A Hybrid Procedure for Efficient Link Dimensioning," *Elsevier Computer Networks*, vol. 67, pp. 252–269, 2014.
- [7] R. de O. Schmidt, L. Hendriks, R. van de Pol, and A. Pras, "OpenFlow-based Link Dimensioning," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2014.
- [8] R. de O. Schmidt, R. Sadre, A. Sperotto, and A. Pras, "Impact of Packet Sampling on Link Dimensioning," *IEEE Transactions on Network and Service Management*. Submitted and under review, 2014.
- [9] R. de O. Schmidt, "Sampling algorithms," <https://github.com/ricardoschmidt/sampling/>, online, accessed Oct. 2014.
- [10] —, "Algorithm to create flow-level time series," <https://github.com/ricardoschmidt/flow-ts/>, online, accessed Oct. 2014.
- [11] —, "Source code of OpenFlow controller," <https://github.com/ricardoschmidt/openflow/>, online, accessed Oct. 2014.

REFERENCES

- [12] V. Gehlen, A. Finamore, M. Mellia, and M. M. Munafò, "Uncovering the Big Players of the Web," in *Proceedings of the 4th International Workshop on Traffic Monitoring and Analysis*, ser. TMA'12, 2012, pp. 15–28.
- [13] Cisco Systems Inc., "How To Calculate Bandwidth Utilization Using SNMP," http://www.cisco.com/image/gif/paws/8141/calculate_bandwidth_snmp.pdf, 2005, online, accessed Apr. 2014.

- [14] —, "Best Practices in Core Network Capacity Planning," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/quantum/white_paper_c11-728551.pdf, 2013, online, accessed Aug. 2014.
- [15] R. van de Meent, "Network Link Dimensioning: A measurement & modeling based approach," Ph.D. dissertation, University of Twente, 2006.
- [16] H. van den Berg, M. Mandjes, R. van de Meent, A. Pras, F. Roijers, and P. Venemans, "QoS-aware bandwidth provisioning for IP network links," *Elsevier Computer Networks*, vol. 50, no. 5, pp. 631–647, 2006.
- [17] A. Pras, L. Nieuwenhuis, R. van de Meent, and M. Mandjes, "Dimensioning Network Links: A New Look at Equivalent Bandwidth," *IEEE Network*, vol. 23, no. 2, pp. 5–10, 2009.
- [18] Center for Applied Internet Data Analysis (CAIDA), <http://www.caida.org/data/overview/>, online, accessed Oct. 2014.
- [19] Measurement and Analysis of the WIDE Internet (MAWI), "MAWI Working Group Traffic Archive," <http://mawi.wide.ad.jp/mawi/>, online, accessed Oct. 2014.
- [20] J. Kilpi and I. Norros, "Testing the Gaussian approximation of aggregate traffic," in *Proceedings of the 2nd ACM SIGCOMM Internet Measurement Workshop (IMW)*, 2002, pp. 49–61.
- [21] R. van de Meent, M. Mandjes, and A. Pras, "Gaussian Traffic Everywhere?" in *Proceedings of the of the IEEE International Conference in Communications*, 2006, pp. 573–578.
- [22] B. M. Brown and T. P. Hettmansperger, "Normal Scores, Normal Plots and Tests for Normality," *Journal of the American Statistical Association*, vol. 91, no. 436, pp. 1668–1675, 1996.
- [23] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection," RFC 5475, 2009.
- [24] P. Phaal, S. Panchen, and N. McKee, "InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks," RFC 3176, 2001.
- [25] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954, 2004.
- [26] B. Claise, B. Trammell, and P. Aitken, "Specifications of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011, 2013.
- [27] Juniper Networks, "Juniper Flow Monitoring," <http://www.juniper.net/us/en/local/pdf/app-notes/3500204-en.pdf>, 2011, online, accessed Jun. 2014.
- [28] C. M. Inacio and B. Trammell, "YAF: Yet Another Flowmeter," in *Proceedings of the 24th Large Installation System Administration Conference*, ser. LISA'10, 2010, pp. 1–12.
- [29] Open Networking Foundation, "OpenFlow Switch Specification," <http://www.opennetworking.org/sdn-resources/onf-specifications/openflow>, 2013, online, accessed Oct. 2014.
- [30] "Ryu SDN Framework," <http://osrg.github.io/ryu/>, online, accessed Oct. 2014.
- [31] "Open vSwitch (OVS)," <http://openvswitch.org/>, online, Accessed Oct. 2014.