



A hybrid procedure for efficient link dimensioning



R. de O. Schmidt^{a,*}, R. Sadre^b, A. Sperotto^a, H. van den Berg^{a,c}, A. Pras^a

^a University of Twente, Enschede, The Netherlands

^b Aalborg University, Aalborg, Denmark

^c TNO Information and Communication Technology, Delft, The Netherlands

ARTICLE INFO

Article history:

Received 15 October 2013

Received in revised form 6 March 2014

Accepted 8 April 2014

Available online 18 April 2014

Keywords:

Link dimensioning

Bandwidth estimation

Flows

NetFlow

IPFIX

ABSTRACT

An important task for network operators is to properly dimension the capacity of their links. Often, this is done by simple rules of thumb based on coarse traffic measurements provided, e.g., by SNMP. More accurate estimations of the required link capacity typically require packet-level measurements, which are hard to implement in today's high-speed networks. The challenge is, therefore, to accurately estimate the traffic statistics needed for estimating the required link capacity with minimal traffic measurement effort. This paper proposes a novel, hybrid procedure for link dimensioning that combines flow-level measurements, minimal efforts on packet captures, and an analytical traffic model. The result is an efficient and robust method to estimate required link capacities. Although the idea of estimating required capacities from flows is not new, the novelty of this paper is that it proposes a complete, efficient and deployable procedure. The proposed procedure has been extensively validated using real-world traffic captures dating from 2011 to 2012. Results show that, with minimal measurement effort, we are able to efficiently estimate the required bandwidth at timescales as low as 1 ms.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

An important task for network operators is to properly provision the capacity of their links. Under-provisioned links might result in immediate decrease in network performance, which can even be perceived by end users. Aiming at adequate QoS (*Quality of Service*), operators continuously monitor link usage. A commonly adopted approach is to read interface counters via SNMP (*Simple Network Management Protocol*) and use obtained values to roughly estimate required capacity for current traffic. Performing these measurements is relatively easy because such protocol is already implemented in most devices. However, the estimation of required capacity might lack accuracy since short-term traffic fluctuations are hard to

capture via SNMP. Therefore, network operators tend to over-provision their links by using a *rule of thumb*: adding “large-enough” safety margins on top of the traffic averages obtained from SNMP counters. Over-provisioning, however, can lead to waste of link resources. Aiming at more efficient provisioning, in the recent past the research community has proposed several more accurate procedures for estimation of the required link capacity. Instead of relying on SNMP, these procedures often require traffic measurements solely at the packet level. However, continuous packet-level measurements in today's high-speed networks, with traffic rates of 10 Gb/s and more, are hard to deploy because they demand dedicated and mostly expensive devices.

Backbone links capacity provisioning is not the only possible application for link dimensioning approaches. These can also be used for a variety of related network management and configuration operations. Efficient

* Corresponding author. Tel.: +31 534898041.

E-mail address: r.schmidt@utwente.nl (R. de O. Schmidt).

estimations of required capacity enable operators to know the residual capacity of their links (i.e., unused capacity). This information can be used, for example, to efficiently reallocate traffic in operations of load balancing and also towards energy efficiency. Furthermore, in a dynamic on-demand bandwidth service, link dimensioning can be applied on allocation of requests for resources, supporting QoS provisioning.

Contribution. This paper presents an efficient and practical link dimensioning procedure. Aiming at minimal measurement effort, this procedure uses flow-level traffic measurements (NetFlow/IPFIX-like measurements) combined with sporadic packet captures and an analytical model to efficiently describe short-term traffic fluctuations. The traffic model proposed in this paper extends the original model in [1] and allows us to predict traffic variance from flows at arbitrary timescales. This variance is then used in the dimensioning formula from [1–3]. Although the idea of using flow measurements for estimation of required link capacity is not new, the novelty of this paper is that we propose a complete and deployable procedure for link dimensioning. Our procedure has been extensively validated using real-world traffic measurements captured on universities routers and operators backbone links around the globe in 2011 and 2012. Our results show that we are able to efficiently estimate the required link capacity with minimal measurement effort at timescale as low as 1 ms.

Organization. The remainder of this paper is structured as follows. Related work on link dimensioning is described in Section 2. Flow-level network traffic monitoring is introduced in Section 3. In Sections 4 and 5 we detail the background on which we base our contributions, and also present the proposed flow-based link dimensioning procedure. Then, a complete overview of the proposed procedure is given in Section 6. The measurements dataset used in this paper is presented in Section 7. The validation of the proposed procedure and results discussion are done in Section 8. In Section 9 we provide a discussion on the parameters of the proposed solution and provide directions on how to set them in real deployments. Finally, in Section 10, we draw our conclusions.

2. Related work

The problem of bandwidth provisioning has been extensively studied. Several of the proposed solutions are technology-specific. For example, recently, Anjum et al. [4] proposed a bandwidth allocation procedure for delay sensitive applications along a path of point-to-point MPLS (*Multiprotocol Label Switching*) connection. More general solutions, such as [5,3,6], have also been proposed, in which intelligent over-provisioning of backbone links is presented as an attractive alternative for QoS achievement; Fraleigh et al. [5] focuses on packet delay, while [3,6] on link rate exceedance. However, because they require traffic measurements at the packet level, such solutions are hard to deploy since packet monitoring in high-speed networks requires powerful and expensive technologies. Pras et al. [3] and Mandjes and van de Meent [6] also propose an indirect

method towards link dimensioning, in which traffic statistics are computed from samples of the router's buffer content. Although this approach does not need on-link traffic measurements, it requires additional complexity to be implemented in the routers.

In [9] the authors propose a bandwidth estimator based on a $M/G/\infty$ model. The main limitation of this work is, however, that it requires continuous packet-level measurements to observe packet arrivals and sizes. In addition, the model is further divided in four different sets of equations, and the selection on which one to use will depend on the timescale the operator wishes to dimension a given link. Our proposed solution differs by the fact that the timescale is already modeled within the adopted dimensioning formula (originally from [2,3]). This allows flexibility to the operator without the need to readapt the dimensioning procedure if timescale is changed.

In [7], the authors propose a traffic model based on Poisson flow arrivals and i.i.d. flow rates able to predict bandwidth consumption for non-congested backbone links. Our contribution differs in the proposed way of computing the traffic variance, since in our case no assumption on the evolution over time of the traffic in a single flow is needed. In [8], the authors provide dimensioning formulas for IP access networks, and the QoS is measured by useful per-flow throughput. In such work only elastic data traffic (TCP connections) was considered, while we do not put any constraint on the nature of the traffic.

The work in [1] proposes a provisioning procedure requiring minimal measurement effort, using minimal model assumptions, and with QoS constraints expressed in link rate exceedance. However, this work focuses on traffic variation that are solely due to fluctuations at the flow level, and the proposed bandwidth provisioning method is only valid for relatively large timescales, e.g., 1 s. We build upon this modeling approach by proposing an extended version of the model. In short, we propose a flow-based formula and additional packet-based correction factors that together enable better estimations of required capacity at smaller timescales. Finally, in [10] we propose a purely flow-based approach to estimate traffic variance from flow-based time series, which proved to work at timescales as low as 1 s. The modeling approach in the present paper, however, lowers this boundary to 1 ms.

3. Flow-based traffic monitoring

In this section we provide a brief introduction to the concept of flow-based monitoring. In [11], a *flow* is defined as a set of packets that share common properties passing an observation point in the network. A commonly-used flow definition is based on a 5-tuple key consisting of source and destination IP addresses, source and destination ports, and transport layer protocol.

A flow-monitoring probe exports information on the observed flows by means of *flow records*. Flow records are usually generated on the basis of timers with configurable timeouts, namely *active* and *inactive* timeout. These are defined as:

- The **inactive timeout** defines how long the monitoring device keeps a flow record in its internal memory before exporting it after the last packet of the flow has been observed. Consequently, flows with packet inter-arrival times larger than the inactive timeout are split into multiple flow records.
- The **active timeout** tells the monitoring probe to export a flow record after a given time interval, even if the flow is still active. That is, the active timeout defines a maximum duration for an active flow record and, hence, it causes flows with durations longer than the active timeout to be exported as multiple flow records.

Nowadays, the majority of network devices is flow-enabled, such as, among others, Cisco routers with embedded NetFlow [12] and IPFIX-based monitoring probes. Consequently, flow-based approaches are easy to deploy in large infrastructures with minimal effort. The downside is that data aggregation performed by the flow probe comes at the cost of information loss. Typically, a flow record does not contain information on individual packets, such as the packet arrival times or the packet size. This has a direct impact on the problem of link provisioning since these are important information to compute essential traffic characteristics, such as traffic variance. The relationship between link dimensioning and traffic variance is detailed in Section 4.

It is important to understand the trade-off between short and long timeouts. By using longer timeouts, measured data is more aggregated, which might avoid excessive measurement-related traffic. That's mainly an issue in distributed monitoring scenarios, where exporting and collector processes are not located in the same physical device [11]. Longer timeouts, however, may require more buffer resources in the metering process to keep record of many long flows. On the other hand, short timeouts generate more flow records and may increase measurement-related traffic. The advantage of short timeouts is that short-term traffic fluctuations can be better reconstructed [10]. Further discussions on the impact of flow timeouts on traffic monitoring can be found in Section 7 and on link dimensioning in Section 9.

4. Models definition

In this section we first briefly introduce previous work on which our procedure is based, namely, the link dimensioning formula from [1–3] and the flow-level traffic model from [1]. We also present novel contributions that are extensions to the flow model from [1] and an important part of our proposed procedure.

4.1. Link dimensioning formula

The work in this paper is based on the link dimensioning formula for Gaussian traffic proposed in [1–3], where a statistical approach to the problem of link dimensioning is provided, structured around the goal of “link transparency” (Gaussianity of traffic has been extensively assessed in previous works [17–19]). With this, the authors indicate the situation in which users should almost never perceive

performance degradations due to lack of bandwidth. Link transparency is statistically guaranteed when the provided link capacity C satisfies:

$$\mathbb{P}\{A(T) \geq CT\} \leq \varepsilon, \quad (1)$$

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 .

The authors of [1–3] provide a bandwidth provisioning formula applicable under the assumption that the traffic aggregates $A(T)$ at timescale T are *normally distributed* and *stationary*. They show that the link capacity $C(T, \varepsilon)$ needed to satisfy Eq. (1) can be computed by:

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

where ρ is the mean traffic rate and the second term can be seen as a “safety margin” depending on the variance $v(T)$ of $A(T)$ and chosen exceedance probability ε . The formula is therefore able to take into account the impact of possible traffic bursts on the link capacity. In addition, it is very flexible: network operators can choose T and ε according to the QoS that they want to provide to their customers. For example, while larger T (i.e., around 1 s) would be enough to provide good quality of experience to users on elastic services, shorter T should be chosen when real time applications are predominant in the network. ε should be chosen in accordance to the desired QoS.

When using Eq. (2) to calculate the bandwidth requirement of empirical network traffic, the main challenge generally consists in estimating $v(T)$ from the measurement data, especially if T is small. The goal of this paper is to minimize or eliminate the need for packet measurements for link dimensioning. To do so, we provide a procedure to estimate $v(T)$ from *flow-level* measurements, supported by an analytical representation of traffic characteristics at the packet level.

4.2. Flow-based model

The authors of [1] present an M/G/ ∞ model to estimate $v(T)$ at the flow-level. In its simplest form, the model assumes that traffic flows are created according to a Poisson process with rate λ and have i.i.d. duration D . Furthermore, it assumes that all flows have an identical and constant traffic rate r . The mean throughput is then $\rho = \lambda \delta r$ with $\delta = \mathbb{E}[D]$ and the amount of traffic in a period of time T is $A(T) = r \int_0^T N(t) dt$ with $N(t)$ being the number of active flows at time t .

The basic idea of the model is that $N(t)$ is identical to the number of busy servers in a M/G/ ∞ queueing station with arrival rate λ and service time distribution F_D . Using this assumption, the variance $v_{flow}(T)$ of $A(T)$ is found to be given by:

$$v_{flow}(T) = \lambda r^2 \left(2T \int_0^T x(1 - F_D(x)) dx - \delta \int_0^T x^2 f_{D'}(x) dx + \delta T^2 (1 - F_D(T)) \right), \quad (3)$$

where D' is the residual distribution of D , i.e., $1 - F_D(x) = \delta f_{D'}(x)$ [1]. As usual, f_x and F_x denote, respec-

tively, the density and distribution function of a random variable X . Knowing the variance, Eq. (2) is used to compute the bandwidth requirement $C(T, \varepsilon)$.

It should be noted that, differently from the research in this paper, the model in [1] was not directly applied to empirical flow measurements, but rather was used for a mathematical analysis of traffic behavior.

4.3. Flow-level traffic variance

The authors of [1] also give explicit expressions for the variance in case of negative exponentially and Pareto distributed flow durations. For the former the variance becomes [1]:

$$v_{exp}(T) = 2\rho\delta^2r(e^{-T/\delta} - 1 + T/\delta). \quad (4)$$

By examining empirical data we have found out that the distribution of flow duration is long-tailed and fits better to Pareto- or Weibull-like distributions. Nonetheless, as the authors point out in [1] and also shown in Section 8 of this paper, the choice of the duration distribution does not affect much the resulting estimated variance. Therefore, one might even consider to use a simple model, where flows are assumed to have a constant duration δ (further motivations for such a choice will be given in Section 5). Assuming a deterministic distribution F_D , Eq. (3) simplifies to

$$v_{const}(T) = \begin{cases} \rho r \left(T^2 - \frac{T^3}{3\delta} \right), & \text{if } T < \delta, \\ \rho r \left(T\delta - \frac{\delta^2}{3} \right), & \text{if } T \geq \delta. \end{cases} \quad (5)$$

4.4. Packet-level modeling of flows

The basic model in [1] assumes that the traffic rate inside a flow is constant. In general this is not true because IP traffic is transported in form of discrete packets with non-constant inter-arrival times. As a consequence, the basic model underestimates the traffic variance due to possible bursts of packets in a flow. In [1], the authors also proposed an extension of the model aiming at modeling also the packet details within flows. Assuming that flows consist of packets of constant size s arriving according to a Poisson process, the estimation of the variance becomes (called corrected variance in the following):

$$v_{corr}(T) = v_{flow}(T) + \phi, \quad (6)$$

with the correction term ϕ given by:

$$\phi_1 = \rho s T, \quad (7)$$

accounting for quantized nature of the traffic.

However, our experiments with empirical data reveal that the corrected variance also underestimates the real variance. Therefore, we propose two further extensions of the model by relaxing the assumptions of Poisson arrivals and constant packet sizes. Next, these extensions are detailed.

4.4.1. Poisson arrival and non-constant packet size

It is clear that IP packets are not constant in size. Under the assumption that packet arrivals inside a flow are Poisson distributed with i.i.d. non-constant packet sizes S , the correction term ϕ in Eq. (6) becomes:

$$\phi_2 = \rho T \chi, \quad (8)$$

where $\chi = \frac{\mathbb{E}[S^2]}{\mathbb{E}[S]}$ with $\mathbb{E}[S]$ and $\mathbb{E}[S^2]$ being the first resp. second moment of the packet size. A proof for Eq. (8) is given in Appendix A. Note that Eq. (7) immediately follows from Eq. (8) for a deterministic packet size distribution.

4.4.2. Bursty arrival and non-constant packet size

Similar to the previous extension, we assume that the packet size S is not constant. In addition, we assume that packets arrive in bursts of P packets and the time between bursts is i.i.d. and exponentially distributed (batch Poisson process), where P is geometrically distributed with success probability p , i.e., $\mathbb{P}[P = i] = (1 - p)^{i-1}p$. Hence, the packet inter-arrival time IA is hyper-exponentially distributed with squared coefficient of variation $c_{IA}^2 = \frac{2-p}{p}$, which suggests that p can be estimated from an empirically measured squared coefficient of variation by

$$p = \frac{2}{1 + c_{IA}^2}. \quad (9)$$

Remarkably, a packet burst of P packets can simply be modeled as a huge “super-packet” of byte size $S' = \sum_{i=1}^P S_i$, where S_i is the size of the i th packet in the burst, i.i.d. like S . Since P and S_i are independent, we obtain $\mathbb{E}[S'] = \mathbb{E}[S]/p$ and $\mathbb{E}[S'^2] = \frac{p\mathbb{E}[S^2] + 2(1-p)\mathbb{E}[S]^2}{p^2}$ (see Appendix B). Applying this result to Eq. (8) with $\chi = \frac{\mathbb{E}[S'^2]}{\mathbb{E}[S']}$, the correction term ϕ in Eq. (6) becomes:

$$\phi_3 = \rho T \frac{p\mathbb{E}[S^2] + 2(1-p)\mathbb{E}[S]^2}{p\mathbb{E}[S]}. \quad (10)$$

5. Flow classification

From the work in [1], we learn that flow rate plays an important role on the calculation of the traffic variance. In addition, from testing the model with empirical data we have observed that using a single set of model parameters for all flows in a measurement period does not provide satisfying results. One reason is the fact that different applications may result in distinct flow characteristics. Another reason is that we are working with flow records, which introduces an artificial upper limit to the flow duration. In order to better account for this behavior, we group flows according to their rate and duration. Ultimately, the traffic variance that goes into the formula of Eq. (2) is obtained by simply adding up the individual variances of all classes.

Fig. 1 illustrates how flow records related to each other by their respective rate and duration. This figure shows the positioning of flows in a scatter plot by their rate and duration. We can also clearly see the upper limit introduced to the flow duration given the use of flow records. In this 2-dimensional classification, we divide the rate-duration

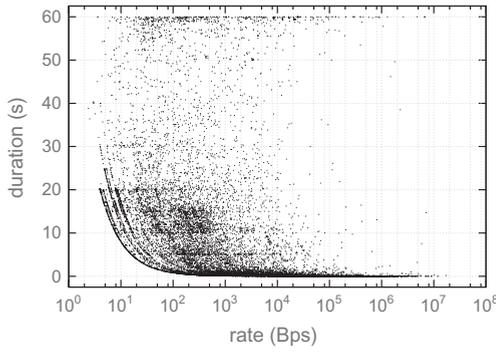


Fig. 1. Example of flow records relationship by rate and duration; duration upper-bound is defined by active timeout of 60 s; for clarity points are sampled every 100.

space into cells of size $\theta \times \eta$ and assign all flow records in the measurement period to flow classes $\Gamma_{ij}, i, j \in \mathbb{N}$, where Γ_{ij} contains all flow records with a traffic rate in the interval $[i\theta, (i+1)\theta[$ and a duration in the interval $[j\eta, (j+1)\eta[$. As done for the classification per rate, for each class Γ_{ij} we determine the flow arrival rate λ_{ij} , the flow traffic rate r_{ij} , the average flow duration δ_{ij} , and the average packet size s_{ij} . On defining a small η as compared to the average duration of flow records, we can assume constant duration δ_{ij} within classes. In this case, δ_{ij} is set to the average duration of flow records in the class Γ_{ij} and Eq. (5) is used to calculate the flow-level traffic variance for each individual class.

6. Overview of the proposed procedure

The complete procedure to calculate $C(T, \varepsilon)$ from flow record measurements for a given timescale T and bandwidth exceeding probability ε is summarized in Fig. 2. In this section we describe the procedure using the flows classification per rate θ and duration η . The exact same procedure can be used for classification only by rate. To do so, the value of η should be set to ∞ .

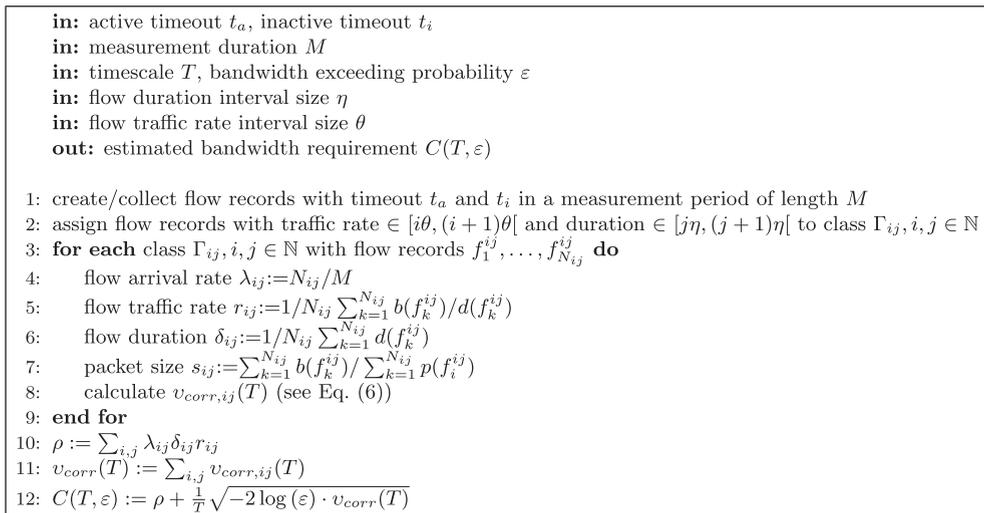


Fig. 2. Procedure for the estimation of the bandwidth requirement from flow records.

The first step (line 1) consists of collecting flow record data for a desired duration M . As explained in Section 3, the records depend on the active timeout t_a and the inactive timeout t_i . We will discuss the effects of the timeouts in the experiments in Section 8.

In line 2, we assign the flow records to classes according to their traffic rate and their duration. The granularity of the classes depends on the parameters η for the flow duration and θ for the traffic rate. We will study various values for η and θ in the experiments in Section 8. Once all flow records have been assigned, the model parameters are determined for each class (lines 4–7) and the variance $v_{corr,ij}(T)$ is computed using Eq. (6). As already explained in Section 4.3, the calculation of the variance $v_{flow}(T)$ can be adapted if a different flow duration distribution is considered. The calculation of the packet correction factor ϕ can also be adapted according to the operators requirements (see Section 4.4).

Finally, the overall traffic rate ρ and variance $v_{corr}(T)$ are computed in lines 10 and 11 and the formula of Eq. (2) is used to calculate the required capacity $C(T, \varepsilon)$ (line 12). Based on the results of our experiments, the selection of values for the parameters $t_a, t_i, M, \eta, \theta, T$, and ε will be discussed in Section 9.

It is important to mention that we disregard flow records with a duration of 0 s, which are mostly composed by single packets, because their traffic rate is undefined. Depending on timeouts configuration, these records may account for more than half of all flow records. However, the impact of removing such flows on the proposed link dimensioning procedure is negligible because they typically only carry around 1% of all transferred bytes.

This section concludes the theoretical part of this paper. In the following we first describe the measurements dataset and then we present the validation of the proposed procedure. Before concluding the paper, we provide a general discussion on parameters setting and their implications on deployment.

Table 1
Summary of measurements.

Abbr.	Description	Year	Length	# Of hosts	Link capacity	Avg. use
A	Link from university's building to core router	2011	24 h	6.5k	2 × 1 Gb/s	15%
B	Core router of university in The Netherlands	2012	6 h	886k	10 Gb/s	10%
C	Core router of university in Brazil	2012	84 h 45 min	10.5k	155 and 40 Mb/s	19%
D	Backbone links connecting Chicago and Seattle	2011	4 h	1.8M	2 × 10 Gb/s	8%
E	Backbone links connecting San Jose and Los Angeles	2011–2012	5 h	3M	2 × 10 Gb/s	10%
F	Trans-Pacific backbone link	2012	13 h 15 min	4M	n/a	n/a

7. Measurements data set

In this section we first present the measurements data set used in the validation of our proposed procedure for estimating required bandwidth. It is important to highlight that the entire data set is composed by packet captures, which allows us to validate the proposed procedure against a ground truth (i.e., required capacity empirically found – details in Section 8.1). We also describe the procedure of creating flows out of the packet measurements and present some statistics of the traffic at the flow level.

7.1. Measurement locations

In this section we describe the measurement data set used throughout this paper. The entire dataset comprises 548 15-min traces totaling 137 h of captures. The trace duration of 15 min has been chosen in accordance with [18]. Longer time periods are generally not stationary due to the diurnal pattern. These traces come from different locations around the globe and account for a total of more than 13.3 billion packets. Traffic captures were done at the IP packet level, using tools such as `tcpdump`. Table 1 presents a summary of the data obtained from the six measurement locations. Note that the column “length” gives the total duration of the, not necessarily successive, 15-min traces, i.e., a length of 1 h corresponds to four traces.

Location A. In this location an aggregate link 2 × 1 Gb/s was measured. This link connects a building to the gateway of a university. Most traffic in this link is actually internal to the university. Due to the small number of active hosts in the link, single activities, such as an overnight automatic backup, can completely reshape the traffic for a period. This measurement took place in a week day of September 2011, and lasted for 24 successive hours.

Location B. It was measured a 10 Gb/s link, comprising all the incoming and outgoing traffic in the gateway of a university. The traffic was captured during the first 15 min of every full hour, during 24 h. This measurement took place in December 2012. Most traffic is web browsing and email.

Location C. It was measured an aggregate link (155 Mb/s and 40 Mb/s) also at the gateway of a university. This measurement took place from September 2012 to December 2012. Traces consist of first 15 min of every full hour. Measurements happened from 08:00 to 22:00. Most traffic is web browsing and email.

Locations D and E. Traces from this location are from CAIDA's public repository [13,14]. Four unidirectional

backbone links of 10 Gb/s, interconnecting four cities, were measured (i.e., two in each location). Traces of *D* are from May and June 2011, and traces from *E* are from December 2011 and January and February 2012.

Location F. Traces of this location come from MAWI's public repository [15] and they consist of captures in a trans-pacific link. No additional information on link capacity and usage is provided by MAWI.¹ Traces of this location are from November and December 2012.

For measurements directly performed by us (i.e., locations *A, B* and *C*), no packet losses were observed. From CAIDA's website we know that, for one link of the location *D*'s pair, packet losses are likely to happen. For traces from location *F*, no information on packet loss is provided in MAWI's repository.

7.2. Flow data

We used YAF (*Yet Another Flowmeter*) [16] to create flow records out of the packet traces. YAF is an IPFIX-based software flow probe. We generated three different sets of flow measurements, based on three different combinations of active and inactive timeouts, namely, 5 and 2 s (henceforward referred as *a5i2*), 60 and 20 s (*a60i20*) and 120 and 30 s (*a120i30*). Such values allowed us to judge the impact of short, medium and long timeouts on the procedure (for the definition of timeouts, see Section 3).

Fig. 3a shows the average number of flow records per 15-min trace for the different measurement locations and timeouts. As expected, considering the link capacity and utilization, traces from *D* and *E* generated two orders of magnitude more flow records than, for example, traces from *A*, for the *a5i2* timeouts. The small difference between the number of flow records for any combination of timeouts indicates that most of the flows have a duration lower than 5 s. For *a5i2* flows, we can observe a slight increase in the number of flows for locations *A* and *C*. This means that for these locations few more flows last longer than 5 s.

Fig. 3b shows the average number of (simultaneously) active flow records per trace. Traces from *A* have an average of 542 active *a5i2* flow records per second and around 1.6k active *a120i30* records per second. Traces from *D* and *E* have an average of, respectively, 30.9k and 48.7k active *a5i2* flow records per second, and around 68.8k and 136.9k active *a120i30* records per second. The longer the timeouts the longer flow records take to be exported by the flow exporter. This explains why the number of

¹ The information on the link capacity given on the MAWI website is not consistent with the throughput observed in the traces.

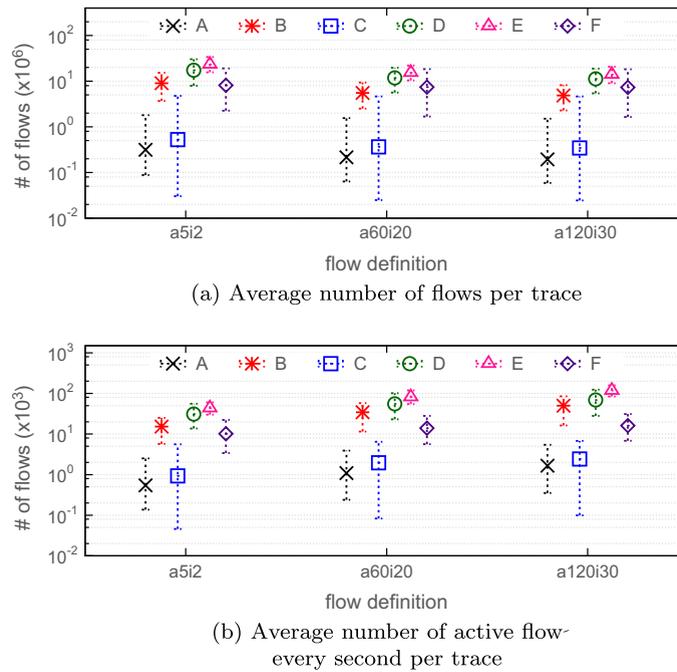


Fig. 3. Flow statistics for all traces per location.

simultaneous active flows increases for longer timeouts, as observed in Fig. 3b. Therefore, although resulting in a smaller number of flow records to be further processed, longer timeouts might demand more resources from the measurement device. Operational considerations on the choice of the timeouts are given in Section 9.

7.3. Traffic properties

7.3.1. Link usage

Although Table 1 presents the average link use for each location, such value is generally not constant over the measurement period. In fact, for some locations it varies substantially. Fig. 4a shows the average traffic rate per

15-min for each location. The figure also shows the minimum and maximum values of mean rate per trace. As one can see, traffic from locations with lower-capacity links and lower averages are the ones that also vary most. For example, for traces of location C, the mean rate reaches values that are $32\times$ smaller than the average, while for traces of E mean rate varies at most ± 1.3 times. Moreover, in particular for locations A, B and C, low averages are most likely to be due to the overnight period, while high averages to the day.

7.3.2. Traffic Gaussianity

The dimensioning formula from Eq. (2) requires input traffic to be Gaussian. Therefore, it is important that such

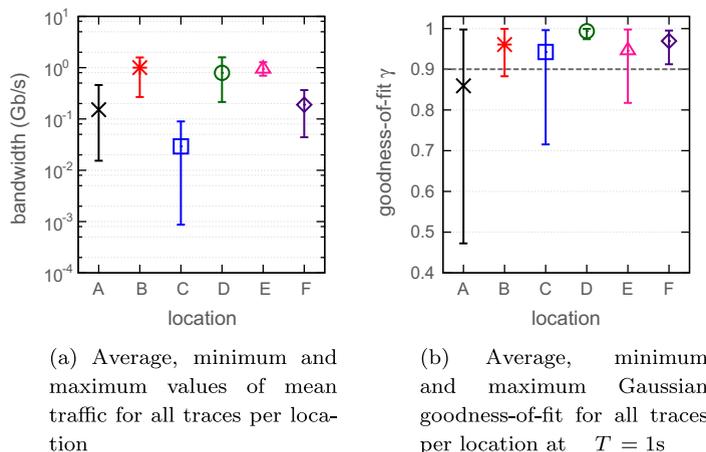


Fig. 4. Traffic properties for the entire dataset.

property of traffic is studied within our dataset. This will further allow us to study the relation between the accuracy of the estimation of required bandwidth and the degree of Gaussianity of the traffic in Section 8. In short, considering a traffic aggregate $A(T)$, at timescale T (for any $T > 0$), we want 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 comply with previous works [17–19], among the many available procedures to quantify Gaussianity *goodness-of-fit*, we have chosen to use the *linear correlation coefficient* [20], which is defined by:

$$\gamma(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (11)$$

where x is the inverse of the normal cumulative distribution function of the sample and y is the ordered sample (i.e., $A(T)$). Fig. 4b shows the average, minimum and maximum values of γ for all traces per location at $T = 1$ s. A value $\gamma \geq 0.9$ supports the hypothesis that the underlying distribution is normal. The main take away of Fig. 4b is that most of our traces have $\gamma \geq 0.9$, i.e., around 83% of all traces in our dataset are at least “fairly” Gaussian. Location A is a 24-h measurement and around 50% of its traces have $\gamma < 0.9$. Most of these traces are measurements of the overnight period in which less users are active in the network, resulting in lower traffic aggregate and, hence, reduced Gaussian character. At locations with larger aggregates, such as D, all the traces are above the limit 0.9. In this section we only show the Gaussian fit of traffic at $T = 1$ s. However, we have studied the Gaussian goodness-of-fit of our entire dataset for T ranging from 1 ms to 30 s and results in all timescales are persistent. In [19] one can find a thorough study on Gaussian properties of our dataset.

8. Experiments and validation

In this section we present and discuss results of experiments with the proposed flow-based procedure. In Section 8.1 we introduce the methodology used for the validation of the procedure. In Section 8.2 we show the impact of flow duration distribution on link dimensioning. The importance of the packet correction factor on the estimation of required capacity at shorter timescales is shown in Section 8.3, as well as how the packet-level parameters can be fitted. In Section 8.4 we show the persistence of fitted packet-level parameters for long term use on link dimensioning. Finally, in Section 8.6 we show results of the extensive validation of the proposed procedure using the entire measurements dataset.

8.1. Methodology

To validate the performance of the flow-based procedure, we apply it to the flow records generated from the 15-min packet traces and compare the estimated required bandwidth with the empirical one, namely the 99th-percentile of the empirical CDF distribution of the throughput. This value represents the minimum capacity that should be allocated so that in only a predefined fraction of time

intervals of size T (i.e., ε) the traffic rate $A(T)/T$ will be above this capacity. Thus, the empirical estimation is defined as:

$$C_{emp}(T, \varepsilon) := \min \{C : \#\{A_i | A_i > CT\} / n \leq \varepsilon\}, \quad (12)$$

where A_1, \dots, A_n are the empirical traffic aggregates on timescale T and ε is the bandwidth exceedance probability.

To verify the accuracy of the estimated required capacity C , for a particular trace, we calculate the number of measured intervals in which the traffic aggregate A_i exceeds C :

$$\hat{\varepsilon} := \#\{A_i | A_i > CT\} / n. \quad (13)$$

From $\hat{\varepsilon}$ we are able to assess whether the estimated required capacity is sufficient or not for a given trace. Clearly, if $\hat{\varepsilon} \leq \varepsilon$ the procedure did not underestimate the required capacity. However, it is also important to check whether the procedure excessively overestimate the required capacity. To quantify the overshooting of the link dimensioning procedure, if any, we calculate the relative error, in percentage, between the estimation and the empirical value (for any T and ε). The relative error is given by:

$$RE = \frac{C - C_{emp}}{C_{emp}} \cdot 100\%. \quad (14)$$

In the following experiments we have always grouped flow records into classes defined by $\theta = 1000$ bytes/s and $\eta = 100$ ms according to the procedure described in Section 5. Flow records were created using active timeout of 60 s and inactive timeout of 20 s. Discussion on how the definition of classes and flow timeouts might impact on the accuracy of required capacity estimations is given in Section 9. In addition, to comply with previous works [1–3], in the following experiments we always set $\varepsilon = 1\%$ in Eq. (2).

8.2. Choice of flow duration distribution

As explained in Section 4.3, to calculate the flow-level traffic variance one may choose a formula according to the distribution of flow duration. From real flow measurements we have observed that the duration of flow records tend to follow a long-tailed distribution, hence, justifying the selection for a Pareto- or Weibull-based variance formula. However, [1] also shows that flow duration do not play an important role in the final estimation of required capacity in the variance. Therefore, difference between estimations using different variance formulas should be negligible. Nonetheless, since in this work we use flow records, which implies a upper-bound for duration, and also by the fact that we classify flow records according to their properties, it is important to revalidate the importance played by duration on variance formulas.

Fig. 5 compares the estimation of required capacity computed using exponential- (Eq. (4)) and constant-based (Eq. (5)) variance formulas at various timescales T . These estimations are represented by C_{exp} and C_{const} , respectively. It also plots the estimation curve of empirical capacity C_{emp} to illustrate the cases in which the flow-based estimation

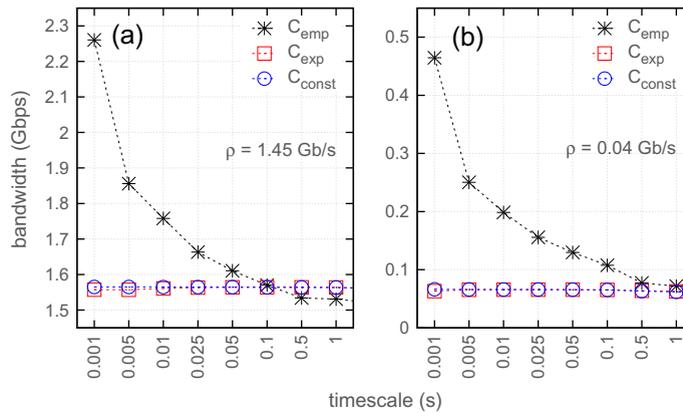


Fig. 5. Estimation of required capacity at various T ; example traces: (a) from D and (b) from C .

is successful. This example shows that the difference between results from both formulas is indeed insignificant and that we can use the simpler constant-based model. Note that in this example we do not implement the packet correction factor ϕ . That is, the flow-based procedure solely gives us a baseline estimation that suffices required capacity at larger T . The packet-level correction factor is, therefore, needed so that the increasing demand as observed for C_{emp} at smaller T is met. The packet correction factor is validated in the following sections.

8.3. Packet correction factor

The packet correction factor ϕ helps us to capture packet-level details within flows, ultimately, aiming at better estimations of required capacity at small timescales. Fig. 6 provides an example of estimation of required capacity C_{flow} using the flow-based model and each one of the three packet correction factors from Section 4.4. In this example, all parameters for the packet correction factor formulas were computed out of the measurements. In Fig. 6, C_{const} is computed using Eq. (2) with variance $v_{const}(T)$ from Eq. (5). For C_{flow} , however, the traffic variance is calculated using Eq. (6) where $v_{flow}(T)$ comes from Eq. (5).

The packet-level correction ϕ_1 , from [1], assumes Poisson packet arrivals and deterministic packet sizes within the flow records. Although better than the purely flow-based method, as shown in Fig. 6, ϕ_1 is clearly still too optimistic and leads to an underestimation of the required link capacity mainly at small timescales. In ϕ_2 we take into account the influence of the packet size distributions appearing in the formula of Eq. (8) through the ratio of its second and first moments. The measured values of the first two moments of packet size distribution slightly increases the estimated required capacity, but still leads to an underestimation. The main take away of this analysis is that the Poisson packet arrival process within flows is apparently too “friendly”. Therefore, in ϕ_3 , in addition to the packet size, we explicitly take into account the burstiness of the packet arrival process. This is done by the assumption that the packets arrive according to a compound Poisson process with geometrically distributed batch sizes and then fit (the first and second order statistics of) this process to measurements on the real arrival process. The assumption of a compound Poisson packet arrival process is, however, very conservative (i.e., “too bursty”), which explains the (strong) overestimation of the required bandwidth by ϕ_3 , as observed in Fig. 6.

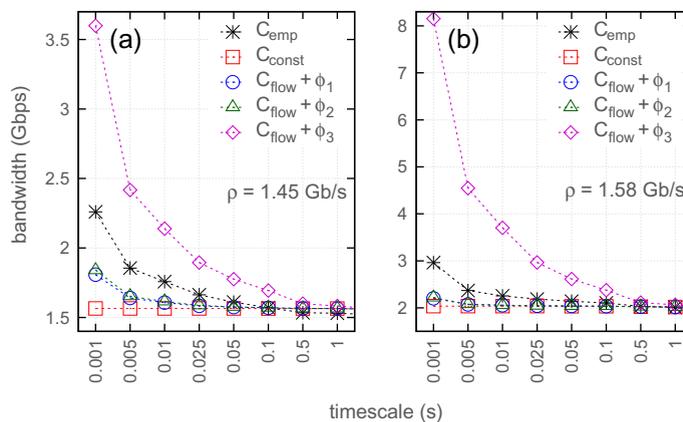


Fig. 6. Estimation of required capacity using different packet correction factors; example traces from locations (a) D and (b) B .

Since parameters for ϕ_2 and ϕ_3 computed from traffic measurements were not sufficient to provide an accurate estimation of required capacity, we propose such values to be fitted against empirically observed data. It is valid to observe that the fitting procedure does not substitute the model because neither χ nor p depend on other important parameters such as T and ε . Considering how the flow model and the packet correction factor were built, the fitting of a single value of χ or p is done for a specific ε and for any T . Therefore, only one “universal” value of χ or p is obtained for the given trace.

8.3.1. Fitting procedure

The amount of traffic $A(T)$, obtained from packet-level measurements, allows us to compute the ground truth $C_{emp}(T, \varepsilon)$ (see Eq. (12)). A value for χ or p is chosen such that the resulting estimation of required capacity C_{flow} satisfies the condition $\hat{\varepsilon} \leq \varepsilon'$ at any T . ε' is the acceptable exceedance probability for the fitting procedure only, i.e., the stopping condition for fitting. The value of ε' should be chosen at most equal to ε so that the fitted values of χ and p would ultimately yield $C_{flow} \geq C_{emp}$ for all considered T .

Fig. 7 shows the estimation curves from the flow-based procedure, using fitted χ for ϕ_2 and fitted p for ϕ_3 . The main take away of this figure is that results from the flow-based procedure supported by the packet correction factor are accurate with fitted χ or p since there is no underestimation. However, such accuracy is questionable at T where excessively overestimation happens, e.g., from 1 ms to 10 ms for Fig. 7b. Such overestimation happens in situations where $\chi_{T=100\text{ms}} > \chi_{T=1\text{ms}}$, i.e., the estimation of the required capacity requires greater χ at larger T than at shorter ones. We have also observed for the example traces of Fig. 7 that the packet level correction is needless at $T > 500$ ms. That is, at such timescales $\chi = 0$ and $p = 1$ cancel out the packet correction factors ϕ_2 and ϕ_3 , respectively.

Operators might be interested in a single T or a reduced set of T . In such cases, the fitting procedure can be performed to those specific T only. This would both reduce the execution time of the fitting algorithm and increase

the accuracy of the fitted χ or p . The later would help to avoid situations as shown in Fig. 7b, where the required χ or p differs too much for small and large values of T . In this case, fitted χ for large T is too high, or p is too low and, hence, they are not an optimal value for the whole range of T .

Now the question is whether a fitted χ or p will remain valid for further successive estimations of required capacity for the same link. Since the fitting process involves packet-level measurements, it is important to minimize such cost as much as possible. That is, if the fitted χ or p can be reused for a long period of time, one will hardly ever need to perform packet measurements for the fitting procedure. The persistence of fitted χ and p is presented in the next section.

8.4. Consistency of fitted χ and p

In the previous section we have shown that fitting χ or p provide us better results at any timescale. However, the drawback is that the fitting process requires packet-level traffic captures to compare the flow-based estimation against an empirical one. The ideal situation would be that the fitted values for χ or p remain valid for a long period of time, providing accurate estimations of required capacity. In this section we show the consistency of fitting χ or p for successive estimations of required bandwidth for the same location. The results in this section used flow records classification by rate and duration and $\varepsilon = 1\%$.

Fig. 8 shows the estimation curves for T ranging from 1 ms to 1 s. In this figure, the estimations of required capacity for four successive traces from location D are depicted. Fig. 8a shows the estimations with fitted χ , and Fig. 8b with fitted p . For both cases, the fitting procedure was performed only for trace 1 of the four traces and the fitted values reused for successive estimations of required capacity. For each trace, the estimation C_{flow} is compared to the trace's respective empirical estimation C_{emp} . The main take away of Fig. 8 is that C_{flow} is never significantly below the respective empirical C_{emp} . This means that the fitted values of χ and p for the first trace were successfully

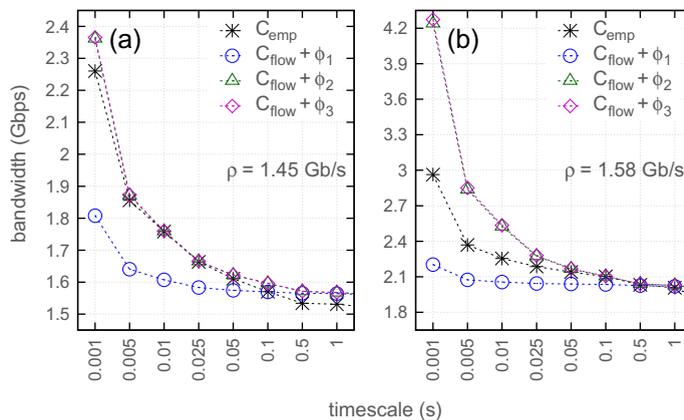


Fig. 7. Estimation of required capacity using packet correction factor ϕ with fitted values of χ and p ; example traces: (a) from D and (b) from B .

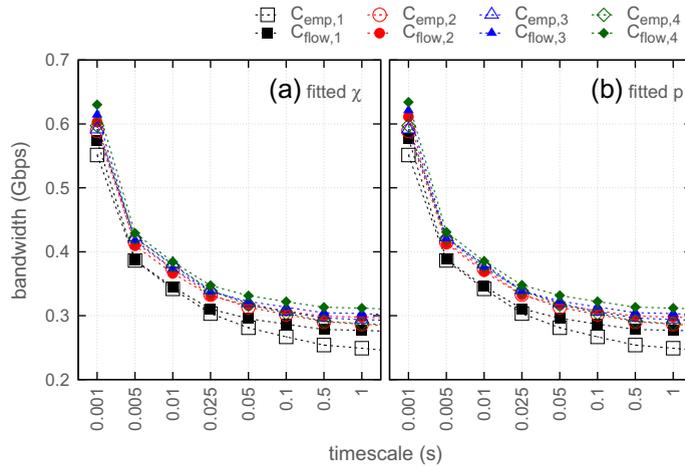


Fig. 8. Estimation of required capacity for four successive traces from location *D* with (a) χ and (b) p fitted only for trace 1.

reapplied in further successive estimations for traces from location *D*.

To extend the example illustrated in Fig. 8, we assessed the validity of fitted χ and p for a larger sequence of traces from locations *D* and *E*. Fig. 9a and b shows the relative difference between C_{emp} and C_{flow} for eight traces from location *D* with fitted χ and p , respectively. Fig. 9c and d shows the same results for eight 15-min traces from location *E*. The first 4 traces (traces 1–4) were captured roughly two months before the last 4 traces (traces 5–8).

Fig. 9 shows the difference in percentage of the calculated C_{flow} using ϕ_2 or ϕ_3 and C_{emp} . That is, y-axis represent how much the obtained C_{flow} , using fitted χ from trace 1,

underestimates or overestimates the empirical required capacity C_{emp} at different T . Clearly, due to the fitting procedure, for trace 1 $|C_{flow} - C_{emp}| \geq 0$ (i.e., no underestimation). However, one can see that the overestimation at short T is not very high, at most around 10%. It means that the obtained exceedance probability $\hat{\epsilon}$ for such cases is less than the defined 1% for ϵ . There are also cases of underestimation, but these are not less than 5%. This means that the obtained error is probably not much higher than the defined ϵ .

For the same set of traces, Fig. 9 shows the difference in percentage of the calculated C_{flow} using ϕ_3 and C_{emp} . The same behavior as for fitted χ can also be observed in this

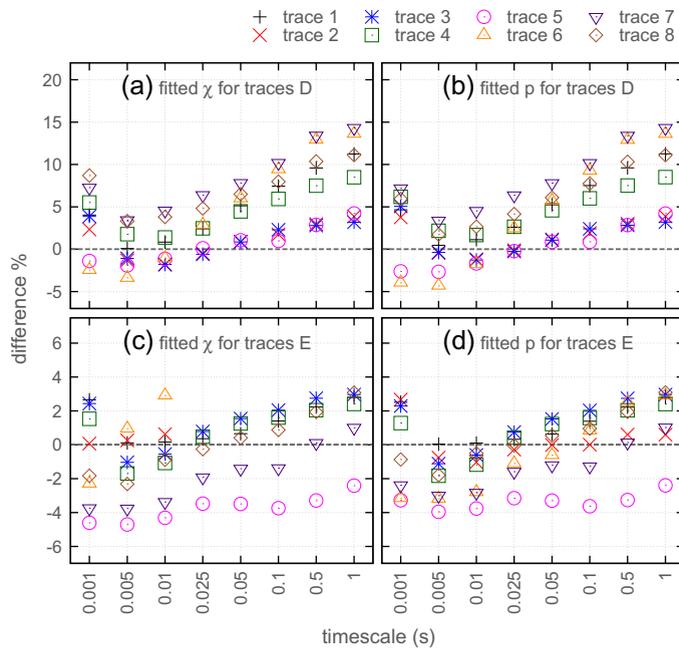


Fig. 9. Relative difference between C_{flow} and C_{emp} for eight successive traces from two different locations with χ (a and c) and p (b and d) fitted only for trace 1.

example, but few differences are noticeable. For example, at $T = 1$ ms, underestimation is slightly higher for traces 5 and 6 and, for all other traces, overestimation is more centralized around 5%.

The main conclusion of Fig. 9 is that the fitted value of χ or p for a single trace remained valid for several successive traces, supporting accurate estimation of required capacity and keeping differences between estimations very small, specially at shorter T . The fitting procedure inherits from the dimensioning formula of Eq. (2) the dependency on Gaussian traffic. Therefore, fitting with non-Gaussian traffic may not yield expected results. This problem is better detailed in the next section.

8.5. Fitting with non-Gaussian traces

One of the key requirements of the link dimensioning formula of Eq. (2) is that input traffic is Gaussian (i.e., normal-distributed). Obviously, such requirement also extends to the fitting procedure, since the dimensioning formula is used. Attempting to fit χ or p using non-Gaussian traffic might result in unexpected behavior of the fitting procedure. In this section we use an example trace from location E that is non-Gaussian at larger timescales. This is an unexpected characteristic since traffic is presumably less Gaussian at shorter T (for more details see [17–19]).

In the example trace used in this section, several traffic bursts of millisecond-precision occurred close to each other in time, as one can see in Fig. 10a. However, since traffic bursts were not excessive high at shorter T , the distribution of traffic throughput was still “enough” Gaussian for the dimensioning formula. However, by increasing the size of the bins in the time series, i.e. larger T , the close-by bursts were averaged together resulting in long-lasting traffic peaks with much higher throughput than the other averages in the time series (see Fig. 10b). These long-lasting peaks compromised the Gaussianity fit of the trace at larger T , as one can see in Fig. 11.

Fig. 11 illustrates the Gaussianity goodness-of-fit of the traffic averages at $T = 10$ ms and $T = 10$ s by means of quantile–quantile (Q–Q) plots. These plots are created by plotting the inverse of the normal cumulative distribution function and the ordered sample – i.e., these are the same of the pairs used for Eq. (11). In Q–Q plots, the more the points fall in a perfect diagonal line, the more the underlying distribution is Gaussian. Therefore, by visually analyzing Fig. 11, one could claim that traffic is not Gaussian at both T . However, as one can see, $\gamma_{T=10\text{ms}} > 0.9$, which supports the hypothesis of Gaussian traffic, but $\gamma_{T=10\text{s}} < 0.9$ and, therefore, traffic is not Gaussian at $T = 10$ s. Although one may still argue that at both T many points fall far from the diagonal line, at $T = 10$ ms many more points are sufficiently close to the perfect diagonal as compared to $T = 10$ s and, hence, these balance the resulting Gaussian fit.

When executing the fitting procedure with the example trace of this section, the yield values for χ and p do not make sense at larger timescales. The proposed packet correction factor is intended for helping the flow-based model to estimate required capacity at shorter T . Therefore, the larger the T , the more we expect that $\chi \sim 0$ and $p \sim 1$. That is, the packet correction factor is cancelled since it is not needed at larger T (see Figs. 6 and 7). However, in this example trace, the fitted values of χ and p are more conservative at large T (i.e., estimations of required capacity are much higher than actually needed). Note that we have already mentioned few times that the packet correction factor is not intended for large T , but for clarity of the example used in this section we have opted for showing results with $T = 10$ s.

Fig. 10a shows the time series for the example trace of location E at $T = 10$ ms. Considering that it is a 15-min-long trace, at $T = 10$ ms there are 90k bins of size T . The fitting stop condition is set to $\varepsilon' = 1\%$, which means that we allow for 900 of these bins to have values above the estimated $C_{\text{flow}}(T, \varepsilon)$. Under these parameters, we obtain $\chi = 12,700$. When defining $T = 10$ s, near-by traffic peaks

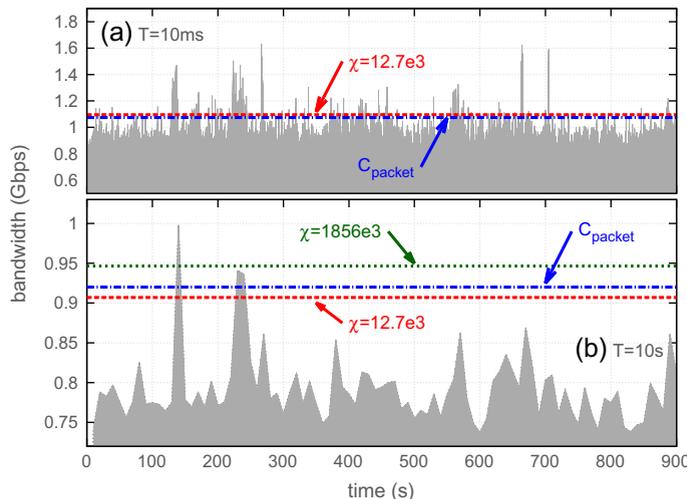


Fig. 10. Time series and estimations of required capacity using fitted χ for example trace from location E .

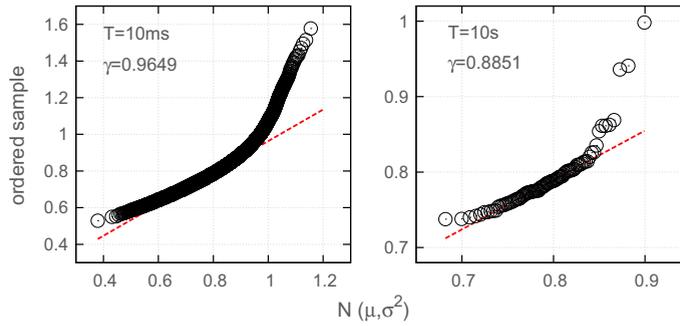


Fig. 11. Q–Q plots for example trace from location E.

between 10 s and 30 s are averaged together resulting in three main huge peaks. Such peaks directly impact on the fitting procedure. The defined $\varepsilon' = 1\%$ at $T = 10$ s means that only 0.9 bins can be above the estimated $C_{flow}(T, \varepsilon)$. This results in $\chi > 5M$. Even if we consider the interpolated value between the 99th and 100th percentiles of the empirical traffic averages, the fitting procedure yields a unreasonable $\chi > 1.8M$.

The main take away of this analysis is that the resulting traffic peaks at $T = 10$ s demand disproportionately high values of χ so that the fitting condition of $\varepsilon' = 1\%$ is met. Considering practical deployment of link dimensioning, $\chi_{T=10s} = 12700$ would suffice, since the operator would be interested in finding a long lasting χ that potentially takes care of regular traffic bursts and disregards unusual peaks – i.e., focusing on customary network behavior and not on exceptions.

It is important to mention that, at $T = 10$ s, as one can see in Fig. 10, even the packet-based dimensioning approach, as proposed in [2,3], fails on estimate the required capacity of traces that present the same behavior as the one studied in this section. In the next section we present results of a thorough validation of the proposed flow-based procedure for link dimensioning using all measurements from our dataset.

8.6. Extensive validation and overall results

In this section we validate the proposed flow-based procedure for link dimensioning by estimating the required capacity for all traces in our dataset. We present results for both packet correction factors ϕ_2 , from Eq. (8), and ϕ_3 , from Eq. (10). A single fitting of χ and p is done for each location using the very first trace in chronological order. Then, the obtained values for χ and p are reapplied to all successive traces for each location. Our conclusions on the quality of estimations are drawn based on the obtained exceedance probability $\hat{\varepsilon}$ given by Eq. (13). As well as in previous sections, we used *a60i20* flows, i.e., created with active timeout of 60 s and inactive timeout of 20 s. Resulting flow records were classified by their respective rate and duration, following the procedure detailed in Section 5. The parameters used for the classification were $\theta = 1000$ bytes/s and $\eta = 100$ ms. To comply with previous works, the exceedance probability was set to $\varepsilon = 1\%$ and T varied from 1 ms to 1 s.

The summary of results is shown in Fig. 12, where for each location the average and standard deviation of $\hat{\varepsilon}$ at various T are plotted. One can see the small difference on results between the approach using ϕ_2 , in Fig. 12b, or the one using ϕ_3 , Fig. 12c. Nonetheless, the approach using ϕ_3 is slightly more conservative. In addition, for comparison purposes, Fig. 12a shows the average and standard deviation of $\hat{\varepsilon}$ for the purely packet-based approach as proposed in [1–3]. That is, mean traffic rate and variance were calculated directly from packets. The main take away of this comparison is that our proposed approaches, helped by the packet correction factors, manage to achieve more conservative estimations at short T , but they demonstrate to be more unstable at large T . The purely packet-based approach was successful (i.e., $\hat{\varepsilon} \leq \varepsilon$) in about 22% of all traces at $T = 10$ ms and 48% at $T = 1$ s. Due to conservative estimations, our procedure using ϕ_2 correctly estimated required capacity for 64% of traces at $T = 10$ ms and for 22% at $T = 1$ s. Furthermore, using ϕ_3 , success was improved to 87% of traces at $T = 10$ ms and 28% at $T = 1$ s.

For the proposed procedure, the worse estimations at $T \geq 100$ ms can be related to fitting of parameters χ and p using non-Gaussian traces. As explained in Section 8.5 this would result in better estimations of required capacity for shorter T , but traffic bursts at larger T would result in higher $\hat{\varepsilon}$. Therefore, underestimation problems at larger T could be alleviated by assuring χ and p to be fitted using Gaussian traffic. Nonetheless, if one considers all estimations of required capacity that resulted in a not too high $\hat{\varepsilon}$, let's say less than 2%, results become more expressive for our procedure. In such case, for the proposed procedure using ϕ_2 , 95% and 44% of traces had $\hat{\varepsilon} \leq 2\%$ at $T = 10$ ms and $T = 1$ s, respectively. For our procedure using ϕ_3 , 99% and 59% of traces had $\hat{\varepsilon} \leq 2\%$ at $T = 10$ ms and $T = 1$ s, respectively.

Although $\hat{\varepsilon} \leq \varepsilon$ is desirable, excessive overestimation is not. If overestimation happens it should be between reasonable boundaries, i.e., not overly higher than the empirical capacity C_{emp} for any T and ε . For example, from the plots of Fig. 12b and c one can see that at very small time-scales $\hat{\varepsilon} = 0$ for location A and the standard deviation is insignificant. The reason for this becomes clear when computing the relative error RE , from Eq. (14). Fig. 13 shows the normalized RE for all traces in our dataset. Note that, since there are different number of traces per location, the x-axis in Fig. 13 shows the percentage of traces per location sorted from left to right by their respective RE .

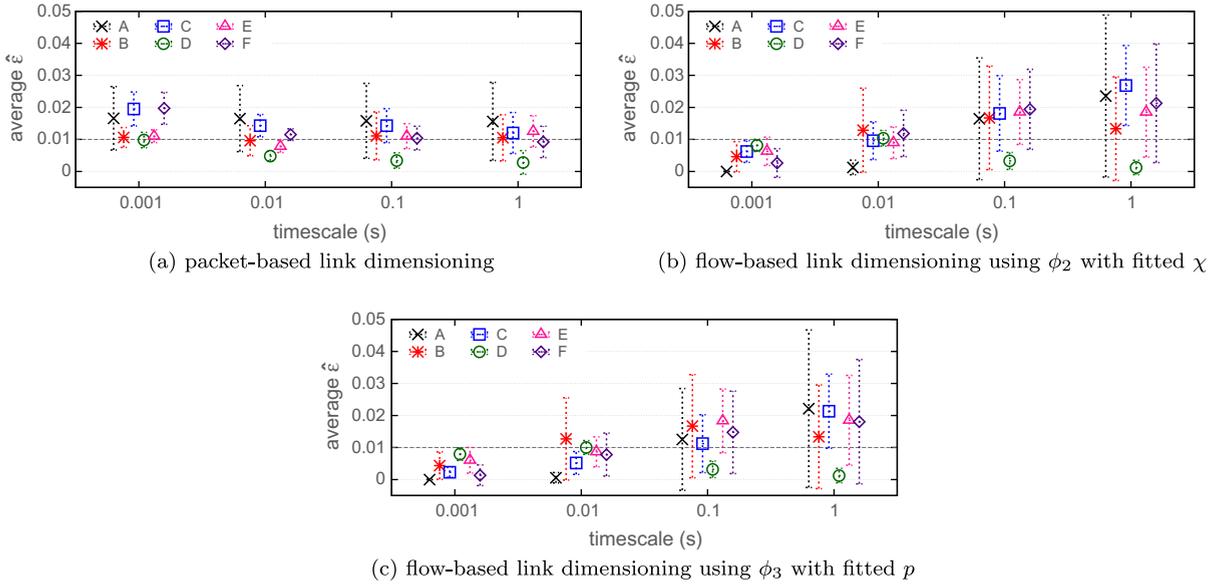


Fig. 12. Average and standard deviation of $\hat{\epsilon}$ per location for all traces in our dataset.

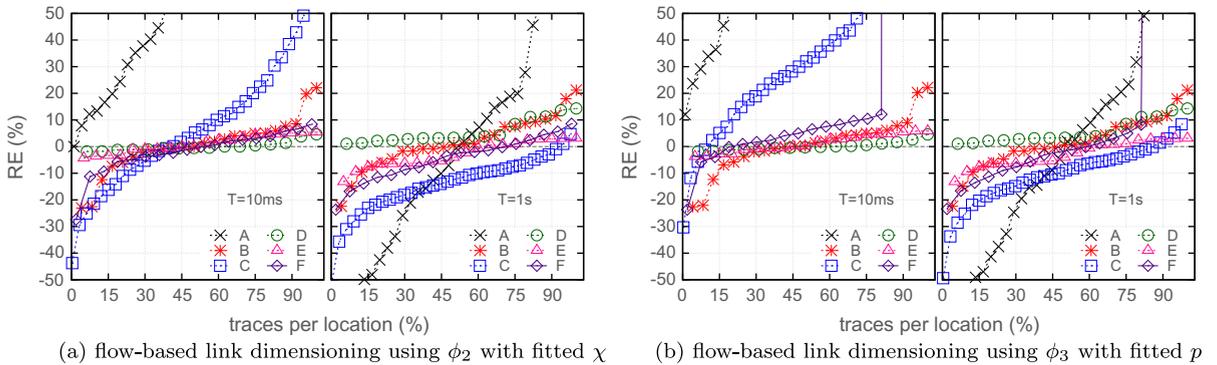


Fig. 13. Relative error for all traces per location; y-axis is limited to $[-50 \dots 50]$ for visualization reasons.

In Fig. 13 one can see that the overestimation is actually quite high for most traces from A at small T . Using ϕ_3 with fitted p at $T = 10$ ms (Fig. 13b), for only about 15% of traces from A the $C(T, \epsilon)$ is less than 50% more the $C_{emp}(T, \epsilon)$. This problem, although with less intensity, can also be observed for traces from location C. As previously mentioned, in our experiments we fitted χ and p only once. Locations A and C illustrate what happens when the shape of the traffic in the measured link constantly varies. Since the measured link in these locations carries traffic of a small number of users, it only takes few users to change traffic properties and invalidate previously fitted χ and p . Besides, these measurements also capture differences in traffic due to day and night patterns. By fitting parameters only once, the “bad fitting” was never fixed and for the other remaining traces the fitted values of χ and p were not the correct ones and, ultimately, yielded mostly very conservative results. For such networks, a system implementing the proposed link dimensioning procedure would better to also implement a checking process to, e.g., decide whether to run the fitting

of parameters again once exorbitant values of estimated required capacity were obtained (i.e., the fitting process should be performed again aiming at having proper values for χ or p). Another idea would be to use different values of χ and p fitted at different times of the day.

Considering only traces from C, at $T = 1$ s and using ϕ_2 with single fitting of χ in around 76% of traces the estimated $C(T, \epsilon)$ was kept in between 20% for more or less the empirical estimation $C_{emp}(T, \epsilon)$. At the same timescale and using ϕ_3 with fitted p , around 84% of traces had estimated capacity within this range. For most of the traces of the other locations (with a larger and regular number of active users throughout the measured period) the estimated required capacity $C(T, \epsilon)$ remained between reasonable bounds, i.e., between 20% for more or less the empirical estimation. For example, at $T = 10$ ms, excluding those from locations A and C, using ϕ_2 around 96% of all traces had estimated $C(T, \epsilon)$ within the range of 20% for more or less the empirical estimation and, using ϕ_3 , it was more than 87% of all traces. In the latter case, for

few and not necessarily consecutive traces from F , the fitted value of p was not appropriate, leading to excessive overestimation of required capacity.

9. Operational considerations and selection of parameters

The proposed flow-based procedure relies in a number of parameters. This section is dedicated to discuss the parameters that were not presented in previous sections and their respective impacts on the accuracy of the proposed link dimensioning procedure.

9.1. Measurement duration

In this paper we have only used 15-min long traces, hence, simulating traffic being monitored every 15 min. The measurement duration should be reasonably chosen such that the traffic during the measurement can be considered stationary, as required by the dimensioning formula of Eq. (2). Longer periods might capture undesired periodic changes on traffic behavior hurting its stationarity character. However, that's not true to assume that traffic will always be stationary when measured in periods of 15 min. It will depend on the traffic nature and network users behavior. The measurement period of 15 min used in this paper was chosen to comply with previous works [1–3].

9.2. Flow timeouts

The active timeout t_a and inactive timeout t_i are set on the flow exporter and they define the length of a flow record and, consequently, the level of aggregation of traffic information. The chosen timeouts will depend on the purposes of traffic monitoring at the network operator. The analysis of previous sections were presented using flow records created with $t_a = 60$ s and $t_i = 20$. However, we have tested our proposed flow-based link dimensioning procedure using many other combinations of timeouts, varying t_a from 5 s to 120 s and t_i from 2 s to 30 s (always obeying the condition $t_a > t_i$). We have not observed any significant difference between results obtained with different timeouts and, therefore, we assume that, for the tested range of values, the timeouts combination does not impact on the accuracy of the estimated required capacity. It is important to know, however, that the amount of processed flow records is the most dominating factor in the computation time in the proposed flow-based procedure.

9.3. Flow records classification

In the previous sections, we presented results obtained with flow records classified by rate $\theta = 1000$ bytes/s and duration $\eta = 100$ ms. Since the definition of these parameters depends on the traffic nature, the network operator would also be responsible for such task. By testing the proposed flow-based procedure we have observed that the smaller the θ and η parameters are defined, the more accurate is the estimation of required capacity. However, the

smaller are such parameters the more classes will be created and, consequently, the more time the procedure may take to compute the required capacity. From the results presented above we can conclude that the settings used in this paper are enough for providing satisfactory accuracy on estimations of required capacity. It should be emphasized that the proposed link dimensioning procedure is very lightweight and even a standard computer can perform the computations for 20 K flow classes in few seconds.

9.4. Exceedance probability ε

To comply with previous works [2,3], in this paper we have always set $\varepsilon = 1\%$. Clearly, it does not make sense to choose smaller ε at large T when the measurement duration is no longer than 15 min (as in the case of this paper). For example, setting $\varepsilon = 1\%$ at $T = 10$ s means that the dimensioning formula should return an estimated required capacity so that under-provisioning happens in only 0.9 out of 90 time bins. Consequently, the link dimensioning procedure may result in excessive overestimation so that over-provisioning happens for all time bins. In addition, network operators must take into consideration the length of the time bin defined by T . That's because the larger T the more traffic is aggregated within a single time bin. This means that, depending on the link load, a single under-provisioned time bin at $T = 1$ s might result in much bigger problems of performance than a under-provisioned time bin at $T = 10$ ms. Therefore, ε must be chosen to avoid underestimation but also avoiding unnecessary overestimation.

9.5. Fitting of χ and p

The crucial point of the fitting procedure for the packet corrections ϕ_2 and ϕ_3 is that ε' should be chosen such that $C_{flow} \geq C_{emp}$. The chosen value for ε' should be enough to avoid underestimation but also not too conservative so that overestimation is not excessively high. For example, $\varepsilon'_1 = \varepsilon$ will result in $C_{flow,1} = C_{emp,1}$ and $\hat{\varepsilon} = \varepsilon$. If the fitted value of χ or p is subsequently used for estimating required bandwidth of the next 15-min measurement period, and $C_{flow,2} < C_{emp,2}$, the end result may be the undesired $\hat{\varepsilon}_2 > \varepsilon$. Therefore, ε' should be wisely chosen obeying $\varepsilon'_2 < \varepsilon$. This way $C_{flow,2} > C_{emp,2}$ for the fitted trace, and a safety margin is kept in order to assure $\hat{\varepsilon}_2 \leq \varepsilon$ for successive traces using the previously fitted χ or p . To play safe, the network operator may choose $\varepsilon' = \varepsilon$, as done in the experiments in this paper. To further reduce risks of having many underestimation cases for successive traces, χ and p should be fitted only using Gaussian traces.

9.6. Choice between ϕ_2 and ϕ_3

Concerning the packet correction factors ϕ_2 and ϕ_3 , we have showed that the latter provided better results than the former. However, this small gain comes at a cost. The trade-off is that ϕ_3 requires the second moment of packets size $\mathbb{E}[S^2]$ (see Eq. (10)). A simple modification in the flow exporter is needed so that the sum of squares of packets

size is also exported within the flow record. Therefore, ϕ_3 deployment is limited to cases in which network operator is able and willing to modify the flow exporter. Modifications can easily be made if the operator uses an open-source flow monitoring tool. For example, we have implemented such modifications in the open source exporter YAF [16].

9.7. Procedure execution performance

Regarding the performance of the whole link dimensioning procedure, one can divide it in two parts: the traffic measurements and the dimensioning calculations. Although our proposed procedure requires sporadic packet-level traffic measurements for the fitting of parameters, these captures do not need to happen for long periods and the main basis of the procedure relies solely on flow-level measurements. It remains, therefore, a lightweight procedure in terms of traffic measurements. Concerning execution time of the calculations for estimating the required capacity, we have observed that even for the largest traces (i.e., those from location *D, E* and *F*) the whole procedure usually took less than a minute to complete. For example, for a large 15-min trace from location *D*, our procedure classified more than 5.6 million flows (defined as *a60i20*) by their respective rate and duration into almost 14,000 classes (defined by $\theta = 1000$ bytes/s and $\eta = 100$ ms, which is the most granular classification tested by us). For each class a variance was computed using Eq. (6). These variances were summed up and, with the trace average rate ρ , applied into the link dimensioning formula. Ultimately, the estimation of required capacity $C(T, \varepsilon)$ was obtained for the same range of timescales used throughout this paper (i.e., 1 ms to 1 s). The overall procedure took around 50 s to complete. Nonetheless, the most costly operation in the whole proposed procedure is the fitting process of parameters for the packet correction factor. This process mainly depends on the range of timescales of interest. The larger the range, the longer the fitting process takes to fit the parameters such that the condition $\hat{\varepsilon} \leq \varepsilon$ is satisfied for all the considered timescales. Using the same example trace from *D*, it took around 1 min 45 s for fitting p (used in ϕ_3) for the same range of timescales from 1 ms to 1 s. Note that these time measurements come from a prototype brute-force implementation. System performance was not the focus of this paper. However, one can certainly expect significantly lower run times with a proper production-ready implementation.

9.8. Link dimensioning in practice

It is inevitable that network operators, even having good estimations of required capacity for their links, will eventually add safety margins on the top of these estimations. As mentioned in the beginning of this paper, this is already adopted practice. However, nowadays operators add margins on top of traffic averages obtained from reading SNMP counters at very coarse time resolutions, such as 5-min averages. The procedure proposed in this work comes to add more reliability on link dimensioning by providing a well founded baseline estimation. Independently

of adding or not a safety margin on top of the estimations, our procedure proved to be, at finer time resolutions, as much efficient as a packet-based approach. Nonetheless, by adding a safety margin, problems of underestimation of required capacity due to, e.g., fitting of parameters with non-Gaussian traces, can be alleviated. For instance, in cases of Fig. 9 around 5% extra capacity (i.e., on top of the estimated one) would already be sufficient for all considered traces to have $\hat{\varepsilon} \leq 1\%$.

10. Conclusions

In this paper we propose a practical link dimensioning procedure aiming at minimal traffic measurement efforts. Our procedure extends the work from [1] by adding a method to capture packet-level details besides the flow-level ones. At the same time our procedure remains lightweight and efficient being able to estimate the required bandwidth within seconds even when several thousands of flows are measured.

The proposed procedure provides a well founded baseline estimation of required capacity for network traffic streams. By using measurements at the flow level, and seldom requiring packet captures, our proposed procedure is – almost – as easy to deploy as SNMP-based approaches and with the advantage that it allows to gather information about traffic fluctuations at finer time resolutions. The main advantage of our procedure is that by integrating analytical modeling with measurement data, estimations of required capacity are as accurate as fully packet-based approaches without the overhead of performing continuous packet captures. Although required by our procedure, packet-level measurements do not need to be longer than 15 min and values obtained from them remain valid for very long periods of time.

Our findings showed that our procedure is able to accurately estimate the required capacity for a range of time resolutions as low as 1 ms. For most applications, e.g., web browsing, end users usually experience QoS at the timescale of 1 s. Having accurate estimations of required capacity for shorter timescales can now also help ensuring QoS for delay sensitive applications, like VoIP and real time video streaming.

Acknowledgements

This work has been funded by FP7 Universal Self (#257513) and FP7 Flamingo (ICT-318488).

Appendix A. Flow model with Poisson packet arrivals

In this section we prove Eq. (8) under the assumption that packet arrivals inside a flow are Poisson with rate μ and have arbitrary sizes i.i.d. like S .

Let T be the aggregation timescale. Obviously, if we observe the active traffic flows during an interval of length T , we will mostly only see fragments of those flows because most flows start or end outside that interval. Let L be the total length of flow fragments observed in that interval. Since packet arrivals are Poisson, the p.m.f. of the number

of packets K_l arriving during that interval for $L = l$ is (neglecting packet transmission times)

$$\mathbb{P}[K_l = k] = \frac{(\mu l)^k}{k!} e^{-\mu l}.$$

Let S_i be the size of the i th packet, i.i.d. like S . The total number of bits A_l arriving in that arrival is then

$$A_l = \sum_{i=1}^{K_l} S_i,$$

with Laplace–Stieltjes transform (LST)

$$\begin{aligned} L_{A_l}(q) &= \mathbb{E} \left[e^{-q \sum_{i=1}^{K_l} S_i} \right] = \sum_{k=0}^{\infty} \frac{(\mu l)^k}{k!} e^{-\mu l} (\mathbb{E}[e^{-qS}])^k \\ &= e^{-\mu l(1 - \mathbb{E}[e^{-qS}])}. \end{aligned}$$

Let $g(l)$ be the p.d.f. of L , with LST $G(\cdot)$. The LST of the total amount of traffic $A(T)$ arriving in that interval is then

$$\begin{aligned} L_{A(T)}(q) &= \int_{l=0}^{\infty} g(l) L_{A_l}(q) dl = \int_{l=0}^{\infty} g(l) e^{-\mu l(1 - \mathbb{E}[e^{-qS}])} dl \\ &= G(-\mu(1 - \mathbb{E}[e^{-qS}])). \end{aligned}$$

Hence, the first and second moment of $A(T)$ are given by

$$\mathbb{E}[A(T)] = -\frac{d}{dq} G(-\mu(1 - \mathbb{E}[e^{-qS}])) \Big|_{q=0}, \quad (\text{A.1})$$

$$\mathbb{E}[A(T)^2] = \frac{d^2}{dq^2} G(-\mu(1 - \mathbb{E}[e^{-qS}])) \Big|_{q=0}. \quad (\text{A.2})$$

By applying basic differentiation rules to Eq. (A.1) and using the identities $-\frac{d}{dx} G(x) \Big|_{x=0} = \mathbb{E}[L]$ and $-\frac{d}{dx} \mathbb{E}[e^{-qS}] \Big|_{x=0} = \mathbb{E}[S]$ we obtain

$$\mathbb{E}[A(T)] = \mu \mathbb{E}[S] \mathbb{E}[L].$$

Similarly, Eq. (A.2) gives

$$\mathbb{E}[A(T)^2] = \mu \mathbb{E}[S^2] \mathbb{E}[L] + \mu^2 \mathbb{E}[S]^2 \mathbb{E}[L^2].$$

Hence, the variance of $A(T)$ is

$$\text{Var}[A(T)] = \mu \mathbb{E}[S^2] \mathbb{E}[L] + \mu^2 \mathbb{E}[S]^2 \mathbb{E}[L^2] - \mu^2 \mathbb{E}[S]^2 \mathbb{E}[L]^2.$$

Noting that the difference $\mu^2 \mathbb{E}[S]^2 \mathbb{E}[L^2] - \mu^2 \mathbb{E}[S]^2 \mathbb{E}[L]^2$ is simply the variance $v_{\text{flow}}(T)$ of the traffic in the constant-traffic-rate model in Eq. (3), introduced in Section 4.2, it holds

$$\text{Var}[A(T)] = v_{\text{flow}}(T) + \mu \mathbb{E}[S^2] \mathbb{E}[L].$$

Using $\mu \mathbb{E}[S] \mathbb{E}[L] = \rho T$, where ρ is the mean of the total traffic throughput, we finally obtain Eq. (8).

Appendix B. Flow model with bursty packet arrivals

Let P the number of packets in a burst. Since $\mathbb{P}[P = i] = (1-p)^{i-1} p$, the mean and variance of P are $\mathbb{E}[P] = 1/p$ and $\text{Var}[P] = (1-p)/p^2$, respectively. The byte size of the burst can be expressed as a sum of a random number of random variables: $S' = \sum_{i=1}^P S_i$. Under the assumption of S_i i.i.d. like S and the independence of S_i and P , it is known for such sums that

$$\mathbb{E}[S'] = \mathbb{E}[P] \mathbb{E}[S],$$

$$\text{Var}[S'] = \mathbb{E}[P] \text{Var}[S] + \mathbb{E}[S]^2 \text{Var}[P],$$

which allows to calculate $\mathbb{E}[S'^2]$.

References

- [1] H. van den Berg, M. Mandjes, R. van de Meent, A. Pras, F. Roijers, P. Venemans, QoS-aware bandwidth provisioning for IP network links, *Comput. Networks* 50 (5) (2006) 631–647.
- [2] R. van de Meent, Network Link Dimensioning – A Measurement & Modeling-based Approach, Ph.D. Thesis, Univ. of Twente, 2006.
- [3] A. Pras, R. van de Meent, M. Mandjes, Dimensioning network links: a new look at equivalent bandwidth, *IEEE Network* 23 (2) (2009) 5–10.
- [4] B. Anjum, H. Perros, X. Mountrouidou, K. Kontovasilis, Bandwidth allocation under end-to-end percentile delay bounds, *Int. J. Network Manage.* 21 (2001) 536–547.
- [5] C. Fraleigh, F. Tobagi, C. Diot, Provisioning IP backbone networks to support latency sensitive traffic, in: Proc. of the IEEE INFOCOM 2003, 2003, pp. 375–385.
- [6] M. Mandjes, R. van de Meent, Resource dimensioning through buffer sampling, *IEEE/ACM Trans. Network.* 17 (5) (2009) 1631–1644.
- [7] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, P. Owezarski, Modeling internet backbone traffic at the flow level, *IEEE Trans. Signal Process.* 51 (8) (2003) 2111–2124.
- [8] T. Bonald, P. Olivier, J. Roberts, Dimensioning high speed IP access networks, in: Proc. of the 18th International Teletraffic Congress (ITC), 2003.
- [9] W. Lautenschlaeger, F. Feller, Light-weight traffic parameter estimation for on-line bandwidth provisioning, in: Proc. of the 24th International Teletraffic Congress (ITC), 2012.
- [10] R. de O. Schmidt, A. Sperotto, R. Sadre, A. Pras, Towards bandwidth estimation using flow measurements, in: Proc. of the 6th International Conference on Autonomous Infrastructure, Management and Security (AIMS), 2012.
- [11] J. Quittek, T. Zseby, B. Claise, S. Zander, Requirements for IP Flow Information Export (IPFIX), RFC 3917, 2004.
- [12] Cisco Systems Inc., Cisco IOS Flexible NetFlow Configuration Guide, Release 12.4T, 2008. <<http://www.cisco.com/>>.
- [13] K.C. Claffy, D. Andersen, P. Hick, The CAIDA Anonymized 2011 Internet Traces. <http://www.caida.org/data/passive/passive_2011_dataset.xml>.
- [14] K.C. Claffy, D. Andersen, P. Hick, The CAIDA Anonymized 2012 Internet Traces. <http://www.caida.org/data/passive/passive_2012_dataset.xml>.
- [15] MAWI Working Group Traffic Archive. <<http://mawi.wide.ad.jp/>>.
- [16] C.M. Inacio, B. Trammell, YAF: yet another flowmeter, in: Proc. of the 24th Large Installation System Administration Conference (LISA), 2010.
- [17] J. Kilpi, I. Norros, Testing the Gaussian approximation of aggregate traffic, in: Proc. of the 2nd ACM SIGCOMM Internet Measurement Workshop (IMW), 2002, pp. 49–61.
- [18] R. van de Meent, M. Mandjes, A. Pras, Gaussian traffic everywhere? in: Proc. of the IEEE International Conference in Communications (ICC), vol. 2, 2006, pp. 573–578.
- [19] R. de O. Schmidt, R. Sadre, A. Pras, Gaussian traffic revisited, in: Proc. of the 12th IFIP Networking, 2013.
- [20] B.M. Brown, T.P. Hettmansperger, Normal Scores, Normal plots and tests for normality, *J. Am. Stat. Assoc.* 91 (436) (1996) 1668–1675.



Ricardo de Oliveira Schmidt is a Ph.D. candidate at the Design and Analysis of Communication Systems Group (DACs) of the University of Twente, The Netherlands. He received a M.Sc. degree in Computer Science from the Federal University of Pernambuco (UFPE), Brazil, and a B.Sc. degree in Computer Science from the University of Passo Fundo (UPF), Brazil. His research interests include network management, traffic measurements, routing and addressing.



Ramin Sadre is an Assistant Professor at the Distributed and Embedded Systems group of the Aalborg University, Denmark. He received a Ph.D. degree from the University of Twente for his thesis titled “Decomposition Based Analysis of Queuing Networks”. His research interests include traffic modeling, the design and analytical performance evaluation of communication systems, and the design of network intrusion detection systems.



Anna Sperotto is a Postdoctoral researcher at the Design and Analysis of Communication Systems Group (DACS) of the University of Twente, The Netherlands. She received a M.Sc. degree in Computer Science from the Ca’Foscari University, Venice, Italy, in 2006 and a Ph.D. degree from the University of Twente, in 2010. Her main topics of interest include intrusion detection and traffic monitoring and modeling.



Hans van den Berg received the M.Sc. and Ph.D. degree in mathematics (stochastic operations research) from the University of Utrecht, The Netherlands, in 1986 and 1990 respectively. From 1986, he worked at the Centre for Mathematics and Computer Science (CWI), Amsterdam. In 1990, he joined KPN Research (now TNO Information and Communication Technology, since January 2003). He is particularly working on performance analysis, traffic management and QoS provisioning for wired and wireless multi-service communication networks (ATM, IP, UMTS, WLAN, ad hoc networks). Currently, he is a Senior Research Member and leader of the QoS group within TNO ICT, Delft, The Netherlands. Since July 2003 Hans van den Berg has a part-time position as full professor within the faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente.



Aiko Pras is an Associate Professor in the Departments of Electrical Engineering and Computer Science at the University of Twente, the Netherlands where he is leading the Design and Analysis of Communication Systems Group. He received a Ph.D. degree for his thesis titled “Network Management Architectures”. His research interests include network management technologies, network monitoring, measurements and security. He is a steering committee member of several conferences, including IM/NOMS and CNSM, and series/associate editor of ComMag and IJNM.