Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 00 (2020) 1–17

On the Dynamics of Valley Times and its Application to Bulk-Transfer Scheduling

David Muelas[a], José Luis García-Dorado[a,*], Sergio Albandea[a], Jorge E. López de Vergara[a,b], Javier Aracil[a,b]

[a]*Department of Electronics and Communications Technologies, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain.*
[b]*Naudit High Performance Computing and Networking, S.L., Spain*

Abstract

Periods of low load have been used for the scheduling of non-interactive tasks since the early stages of computing. Nowadays, the scheduling of bulk transfers—i.e., large-volume transfers without precise timing, such as database distribution, resources replication or backups—stands out among such tasks, given its direct effect on both the performance and billing of networks. Through visual inspection of traffic-demand curves of diverse points of presence (PoP), either a network, link, Internet service provider or Internet exchange point, it becomes apparent that low-use periods of bandwidth demands occur at early morning, showing a noticeable convex shape. Such observation led us to study and model the time when such demands reach their minimum, on what we have named valley time of a PoP, as an approximation to the ideal moment to carry out bulk transfers. After studying and modeling single-PoP scenarios both temporally and spatially seeking homogeneity in the phenomenon, as well as its extension to multi-PoP scenarios or paths—a meta-PoP constructed as the aggregation of several single PoPs—, we propose a final predictor system for the valley time. This tool works as an oracle for scheduling bulk transfers, with different versions according to time scales and the desired trade-off between precision and complexity. The evaluation of the system, named *VTP*, has proven its usefulness with errors below an hour on estimating the occurrence of valley times, as well as errors around 10% in terms of bandwidth between the prediction and actual valley traffic.

Keywords: Network planning, bulk transfers, valley times, demand-curve modeling.

1. Introduction

As soon as computing was born, periods of relative low load have been used for scheduling non-interactive tasks—typically referred to as batch-processing [1]—such as database consolidation or banks' payment processing on mainframes. With the advent of the Internet era, these tasks gradually evolved to run in a distributed fashion—often geodistributed [2, 3]—to an extent that both the virtualization of Internet infrastructures and cloud services are common nowadays. Thus, resource replication, security backups or virtual machine cloning have become regular tasks, where large data volumes are exchanged without a specific or precise timing [4]. These processes are usually termed as massive or bulk transfers [5], and their scheduling is critical for both cost optimization and efficiency.

On the one hand, in terms of cost, a smart scheduling will be imperative for those Internet actors that pay for bandwidth according to volumes. This is particularly critical when Internet Service Providers (ISP) charge for transit following burstable billing [5], typically built on the 95th percentile of bandwidth usage [6]. An example of this situation are Tier 2 ISPs—networks with national or regional scope—that are connected to and charged by larger Tier 1 ISPs based on peak utilization [7]. A similar scenario is often found by cloud service providers and content distribution networks (CDN), as they interconnect their data centers [8] and upstream/downstream traffic to clients by means of Tier 1 ISPs, again, following burstable rates [9]. This way of charging customers opens up the opportunity to trans-

*Corresponding Author.
Email addresses: dav.muelas@uam.es (David Muelas), jl.garcia@uam.es (José Luis García-Dorado), sergio.albandea@aol.com (Sergio Albandea), jorge.lopez_vergara@uam.es (Jorge E. López de Vergara), javier.aracil@uam.es (Javier Aracil)

Received: 3rd June 2020. Revised 20th August 2020. Accepted: 23rd September 2020.

The final publication is available at Elsevier via doi:10.1016/j.comcom.2020.09.015, to be published in *Computer Communications*.

mit bulk transfers on low-use periods without modifying the 95th percentile, hence keeping the expenditures constant.

On the other hand, in terms of efficiency, a network manager may try to minimize the interferences of bulk transfers with regular traffic without deploying new resources or altering the existing topologies. This scenario spans different scales. At the top, ISPs that disseminate contents in different points of presence (PoP) or concentrate logs and traffic measurements in their main premises. As an example, National Research and Education Networks (NRENs) often centralize monitoring tasks in a specific PoP [10]. In the middle, virtualized/software-defined networks that are operated by a single owner/manager, although the infrastructure lies on top of the hardware of a provider [11]. At the bottom, any customer that uses a dedicated link regardless the underlying technology—e.g., GMPLS. For example, a bank that leases links between its branch offices. The common factor of all these examples is the interest to accomplish a smart planning of bulk transfers as they interact with regular traffic.

Netflix's approach to this problem is to team up with ISPs to deploy embedded Open Connect Appliance (OCA) servers—machines which store the most popular contents closer to final users, seeking to reduce latency. According to Netflix public sheets, they generally implement these updates from 2 a.m. to 2 p.m.[1] As a further step in this line, the authors in [12] designed NetStitcher, a store-and-forward system designed to exploit low-cost transmission windows. They defined such periods to be fixed between 3-6 a.m. local time and equal for different places in the U.S.

However, this type of traffic should be exchanged when the infrastructure is, in fact, experiencing a low usage, and this time varies—e.g., weekday vs. weekend. The study of these ideal intervals both longitudinally and spatially is one of the main goals of this work. Interestingly, traffic-demand profiles typically show a convex shape during off-peak periods with an abrupt minimum at early morning—let us say a valley shape.

We pursue to shed light on how the moment of occurrence of this valley—hereinafter, valley time—varies among days, over time and in different places. Specifically, we approach the characterization of valley times with the daily valley-hour metric—the 60 minutes of a day with minimum aggregated traffic, in line with the busy-hour idea [13] to characterize periods of heavy use. In such a way, the valley-hour softens the bursty nature of bandwidth time series, while it remains useful to state the moment with lowest network load and simplifies comparisons and modeling of diverse PoPs.

To this end, we have measurements for years on several PoPs in the Spanish NREN (RedIRIS [14]). We found that valley-time occurrence can be modeled by a Gaussian process after applying simple transformations [15], which allows us to easily compare PoPs' behavior and explain the differences by PoP and other intrinsic factors—such as the month or the day of the week/month when measurements were gathered [16].

Extending the scope of [15], we now focus not only on single-point's occupation, but also on the demands shown by multi-hop paths. That is, given a transfer traversing more than one place, we wonder when the resulting curve of bandwidth demands has a low usage. Smart scheduling in these scenarios require certain collaboration between the operators in charge of each point. This matter calls for solutions that minimize both sharing of critical information—such as detailed performance measurements—and complexity of operation and coordination between actors. Additionally, apart from the different single-point patterns, it becomes apparent that large networks spanning more than a time zone will behave differently than nationwide topologies. Our proposal tackles these issues with estimations using minimal information from performance baselines: specifically, expected location and load during the lowest and busiest daily periods.

Finally, all the resulting findings and conclusions are translated into a prediction system, called *VTP*, which stands for Valley Time Predictor. *VTP* predicts valley times and answers the ideal temporal frame to carry out a bulk transfer—actually, potentially, any batch task—given a PoP or a set of PoPs in the case of paths. The results of applying *VTP* to predict valley times in diverse scenarios have been proven useful, as errors below an hour in terms of time occurrence and around 10% in terms of bandwidth were found. Hence, these results highlight the applicability of the system to schedule future transfers in both single and multi-hop scenarios.

The rest of the paper is organized as follows. In Section 2, preliminaries are presented, including the related work, an overview of daily traffic patterns in a variety of networks, and the specific dataset considered in our study. Then, Section 3 describes the architecture and operation of *VTP*, to provide a general perspective of our solution. After that, Section 4 and Section 5 develop the models of valley times in single and multi-PoP scenarios, respectively. Section 6 includes the experimental evaluation of a proof-of-concept of *VTP*, and Section 7 concludes this work and draws some future lines of work.

2. Preliminaries

In this section, we first detail how the research community has paid attention to the scheduling of bulk transfers; then, we illustrate how traffic demands vary through the day; and, finally, we explore one relevant case to motivate our proposal.

2.1. Related Work

The scheduling of bulk transfers has gained relevance because of its impact on the performance of regular traffic, as well as due to the significant cost in volume-based pricing scenarios—even more given the last reports of the humongous volumes that data centers exchange [17].

One approach to this problem is to exploit the fact that carriers charge customers following 95th-percentile metering [6, 18]. Specifically, the exchanged traffic is typically measured in 5-minute slots and the 95th-percentile sample is used for determining the charges. This observation prompted the research community to propose to carry out bulk transfers when the network use is below such percentile, as this will not generate an

[1] `https://openconnect.netflix.com/`

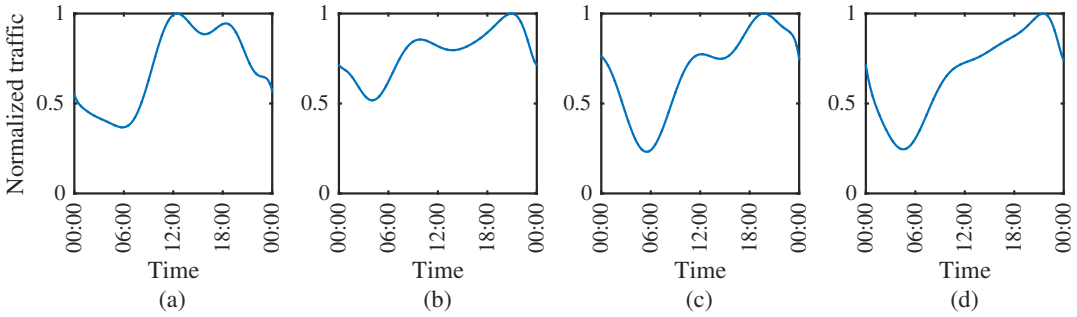

Fig. 1. Normalized traffic examples of daily patterns: (a) academic link, (b) national ISP, (c) international ISP, and (d) international IXP.

extra charge [12, 19, 20]. Note that an alteration of the traffic pattern—i.e., a flatter or uneven shape—, without modifying the percentile is irrelevant for billing.

An alternative approach is to change the routing [9, 21, 22] in such a way that traffic is forwarded to less-used network paths or cheaper data centers. Indeed, Amazon has recently studied physical solutions to move massive data to the cloud—e.g., a hard-drive-bag designed to transfer up to 50 TB of data, or even a secure data truck that stores up to 100 PB, seeking to move exabytes to Amazon Web Services in a matter of weeks [23, 24].

Finally, another option is to identify those periods when a network is in a low-use state, for example at early mornings. As already introduced, the authors in [12, 25] identified the interval between 3 and 6 a.m. as appropriate for undertaking backups. Similarly, Netflix suggests the interval from 2 a.m. to 2 p.m. to perform content-distribution tasks. However, early-morning ranges may differ from one PoP to another, from a country to another, etc. Consequently, these ranges cannot be assumed as a fixed and homogeneous interval. Then, our first aim is to provide a more formal description of the early-morning lapse.

Regarding how to handle transfers involving more than a PoP—multi-PoP paths—the authors in [12] proposed to equip the PoPs of a given topology with storage capacities so that transfers can wait up to a subsequent low-use period in its transit from source to destination. They named such proposal as NetStitcher, generically referred to as a store-and-forward system [26], which was later improved for making both multiple concurrent transfers [27, 28] and backups in optical networks [29] as well as exploiting software-defined network architectures [30].

However, this approximation still demands to know when low-use periods take place. In addition, such lapses have been arbitrary considered fixed, which is unlikely to be optimal in multi-PoP transfers. Therefore, these store-and-forward systems can benefit from our empirical study of the dynamics of low-use period of networks. Additionally, these approaches present two main limitations. First, the potentially long time that data is stored waiting for a proper interval, likely the next day for data traveling from west to east [12]. Second, they require storage equipment on PoPs, with significant grades of synchronism and investment needs.

Alternatively, other studies have focused on cooperative approaches in which different Internet actors interact to reduce cost and give better service. Such cooperation may involve CDNs and ISPs [31] or even P2P users [32]. Our proposal is in line with the latter works as it aims to find a trade-off interval between the set of PoPs involved.

2.2. Traffic activity through the day: Low-use periods

The research community has devoted significant effort to the characterization of the Internet dynamics [33]. Specifically, the authors in [34] studied the evolution of users' traffic demands during the course of a day. However, while busy periods have received significant attention ([16, 35]) likely given its immediate application in network planning and provisioning of bandwidth capacity, their counterpart, the low-used periods, do not—despite its potential applicability to the scheduling of large non-time-critical tasks.

Network traffic-demand shapes during a day have been typically broken down into two main groups: those generated by enterprise or academic users, who access the network within their workplaces, and domestic users, who use the network from their residences. Figure 1 illustrates the daily traffic shape for different scenarios during working days with significant examples—averaged and normalized to strengthen trends. Specifically, examples from an academic network, national and international ISPs, and from an international Internet Exchange Points (IXP) [36].

Several characteristics arise. The most immediate one is the identification of two phases in the daily traffic profile, one of low use and another of high, or busy, activity. However, depending on the specific scenarios, the duration and occurrence of such periods vary. Moreover, we find diverse intra-day amplitudes—i.e., the difference between the bandwidth reached at high and low intervals. This fact may be explained by daily routines of the mix of users each network serves: while enterprise and academic ones experience the peak before lunchtime—the time when the busiest activity is achieved within the day—in domestic environments, peak times usually occur in the evening, when most people return home.

Nevertheless, there is a noteworthy nexus among their respective less-use moments: all of them occur at early morning with a clear convex shape. In other words, at a certain time at night, the aggregate-traffic demands decrease strictly up to a minimum when demands start to increase abruptly. We refer to it hereinafter as valley time of a PoP, given its typical convex shape. We believe that bulk transfers must be scheduled around

Table 1
Summary of measurements for the set of PoPs under study

Point of presence	Average bandwidth (Mb/s)	Valley-hour bandwidth (Mb/s)	Valley-hour average start (a.m.)	Busy-hour bandwidth (Mb/s)	Busy-hour average start (p.m.)
PoP1	194	46	06:02	415	01:03
PoP2	215	87	05:52	379	01:28
PoP3	107	30	06:32	222	01:48
PoP4	1180	510	05:54	1879	02:07
PoP5	248	53	06:05	549	01:17
PoP6	103	23	06:00	228	01:58
PoP7	98	27	05:39	207	01:52
PoP8	117	24	06:11	243	01:49
PoP9	1071	485	06:18	1500	02:28
PoP10	730	438	06:37	991	02:06

this point. On the other hand, the busy periods present patterns that are flattened or, even, irregular, with two peaks. In addition, the busy period tends to be longer than the period of low use, again likely related to human-life dynamics.

Another consideration to underline is the different decreasing and increasing slopes before and after the valley moment. In Figure 1, both in the academic and ISP examples, the traffic decline at nighttime tend to be slower than the growth underwent after the valley time. This behavior is not so clear in the IXP taken as an example, where the slopes around the valleys are far more symmetric—in fact, this behavior is shared by many other IXPs [36].

How these demand curves are altered when considering end-to-end paths, that is, paths involving more than a PoP, has not received significant attention by the Internet community. However, note that this is not an abnormal scenario as, for example, Internet paths typically traverse several different ASs [37].

2.3. Inspection of a national-wide network: RedIRIS

RedIRIS, the Spanish NREN, comprises more than 350 institutions, basically universities, research centers and hospitals. It features nationwide Internet exchange points with Espanix and Catnix, and international connections to the European Research and Education Network, GÉANT, as well as to Level 3 and Cogent. As of 2005, its topology was formed by 18 PoPs and further updated to 40 by 2014, ten of which become part of this study for reasons of availability and reliability of measurements.

During 6 years, from January 2008 to December 2013, all Netflow records generated in such ten PoPs were gathered and processed. On the one hand, Netflow summarizes a set of consecutive packets sharing quintuple description—IP addresses and ports from both source and destination, and protocol—into a simple record with timestamps and number of bytes/packets. On the other hand, Netflow features sampling capacities, which alleviates the short inter-packet times in high-speed networks. Particularly, in our case NetFlow suffered from a packet-sampling rate between 1/100 and 1/200. By elaborating on such data, bandwidth time series were estimated and, subsequently, statistics such as busy hours and valley times were calculated.

The daily traffic in academic networks previously introduced is in line with the features found in RedIRIS. However, we remark that we have found significant diversity. Some characteristics of each PoP are summarized in Table 1, and Figure 2 shows the median of daily bandwidth time series for all the PoPs with the maximum and minimum amplitudes.

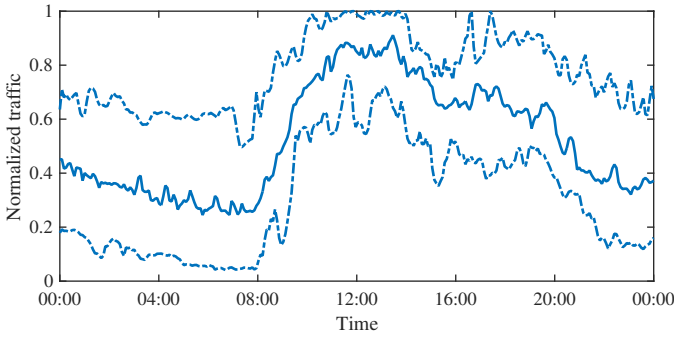

Fig. 2. Summary of daily traffic patterns in RedIRIS. Median is presented with solid line, minimum and maximum amplitudes with dashed lines.

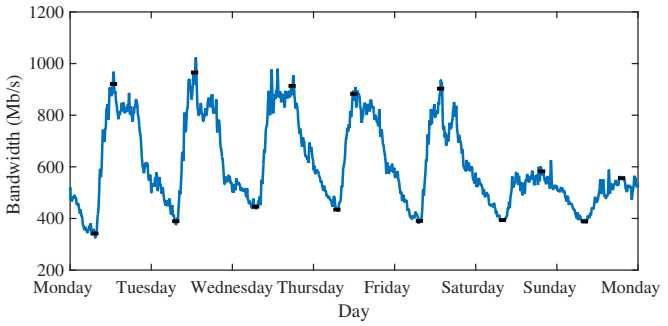

Fig. 3. Example of traffic during a week in RedIRIS. Horizontal black segments show busy and valley times.

Remarkably, the amplitude—ratio valley-hour/busy-hour—ranges from 0.1 to 0.6, so this characteristic is not shared by the set of PoPs under study. The rationale behind this heterogeneity is that the proportion and nature of background traffic differs across PoPs, which makes the traffic at night to not decline to the same extent [38].

Intuitively, daily traffic may be strongly influenced by the day of the week. In particular, it is intuitive to conjecture a disparity between the pattern from Mondays to Fridays—or weekdays—and weekends. For example, the traffic during weekends at academic networks is considerably reduced on the grounds of an appreciable drop in the number of users. Figure 3 depicts a typical pattern for the weekly traffic of a given PoP. It stands out that there is similarity in the patterns observed during the weekday and that it is discontinued on weekends. This fact is also aligned with other works [39] and other free available monitors even for commercial links—e.g., LAIIX IXP[2].

Both weekdays and weekend still present a clear valley time, but with a different starting time, duration, and traffic amplitude. For example, on weekends such amplitude can be about 1/3 with respect to weekdays. Likely because background traffic remains nearly constant whereas active traffic dips [38]. Moreover, the figure depicts the daily busy and valley hours,

[2]Los Angeles International Internet eXchange: `http://www.laiix.net/mrtg/sum.html`

and it becomes apparent that heterogeneity between weekdays and weekends, as well as variation between themselves.

3. *VTP* foundations and architecture

3.1. *VTP* overview

The previous observations motivate to go further in the characterization of valley times with the final aim of improving the scheduling of bulk transfers. Specifically, we propose to accomplish such scheduling by:

- Predicting the interval of lowest network load—load can be, for example, bandwidth but not necessarily—, what we refer to as the valley time. In particular, we are considering intervals of 60 minutes and consequently, the valley time of a day is the consecutive 60 minutes whose aggregated traffic is minimum. In other words, the reverse to the well-known term busy-hour [13] typically used to characterize peak times.
- Selecting a suitable bitrate for the transfer to control its effects on bandwidth usage.
- Adjusting the transfer to be centered at the predicted valley time. In this case, we apply a straightforward strategy that takes the particular size of a bulk transfer and the desired transmission rate—typically constant bitrate (CBR)—as parameters, and tries to optimize the moment in which the transfer starts.

Our approach seems to be the natural one, if we assume a convex and fairly symmetric shape for low-use periods—which is very common, as we have already shown. However, valley shapes are not perfectly symmetric, and interactions in multi-PoP scenarios can attenuate the convexity of the curves of demand. Despite these apparent shortcomings of this strategy, we assess their quantitative significance to justify the advantages of our solution. With this, our solution tries to improve the scheduling of non-time-critical tasks selecting the moment with the lowest global affection to the infrastructure—instead of assuming fixed periods as commented in other cases such as Netflix and NetStitcher.

3.2. Modeling of valley times

VTP follows a multi-level modeling strategy to infer valley times. First, it obtains a local model related to the single-PoP dynamics—e.g., a leased link—, where the key is the correct estimation of the valley time occurrence. Section 4 is devoted to this. Then, *VTP* builds a multi-PoP model—e.g., a set of PoPs comprising a path—, where several single-PoP behaviors are aggregated to provide an idea of the moment with the lowest global load. In this case, the model must take into account that more than a type of network traffic profile and time zones may simultaneously affect data transmissions. Coherently, the predicted valley time for a specific path—that is, the central moment of the scheduled transfer—will correspond to a compromise solution for the involved PoPs.

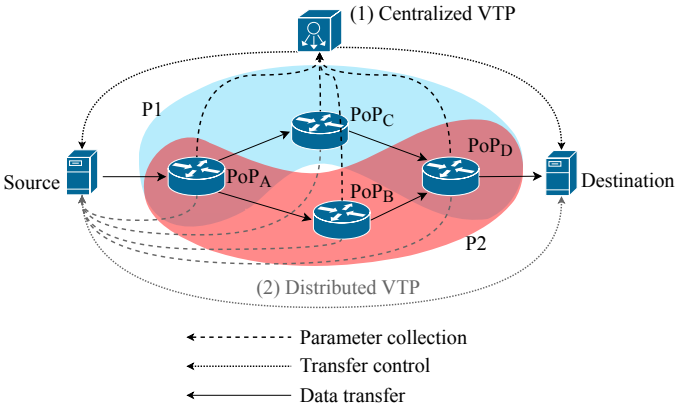

Fig. 4. *VTP* architecture and operation.

More formally, we consider the global load of a given path $P = \{PoP_i\}$ as the load of a meta-PoP which aggregates those of all the $\{PoP_i\}$ PoPs in that path. We illustrate this idea with the example in Figure 4. Here, we depict a scenario where there are two different paths between *Source* and *Destination*, $P1$ and $P2$. Each path encompasses three single PoPs—specifically, $P1 = \{PoP_A, PoP_C, PoP_D\}$ and $P2 = \{PoP_A, PoP_B, PoP_D\}$. We then say that their load is $S_{P1}(t) = Agg(\{PoP_A(t), PoP_C(t), PoP_D(t)\}), S_{P2}(t) = Agg(\{PoP_A(t), PoP_B(t), PoP_D(t)\})$, where $Agg(\cdot)$ is a suitable aggregation function—e.g., sum or average—and t represents the time of day. Therefore, we define the valley time of a path as the moment in which its load reaches its minimum value:

$$VT_{Path} = \text{argmin}_t\{S_{P_{Path}}(t)\} \quad (1)$$

By load—or interchangeably, traffic demands—(S), we can refer, simply, the bandwidth-usage time series in absolute terms—i.e., Mb/s—or normalized by capacity to make equally relevant each PoP of a path—i.e., in percentage of use. Alternatively, such traffic-demand time series can be the complementary of the available bandwidth per PoP—i.e., the difference between capacity and used bandwidth [40, 41]. In such a way, the valley time of two PoPs would be that moment when more aggregate bandwidth is available instead of the moment with less bandwidth in use. In any case, the problem remains equivalent as the difference is a transformation of the data inputs whereas the methodology will be the same. In particular, we will consider bandwidth in absolute terms, as the capacity of some of the links analyzed in this work is unknown.

To solve the optimization problem in (1), we propose two different solutions:

First, a high dimensional model that reconstructs network-load time series per single PoP based on its valley and busy times and the load during these times. Then, the set of per-PoP time series in the path are aggregated, and the valley time calculated as it would be a single-PoP case. This general framework gains signification for multi-time-zones paths. We note that this smart scheduling scheme will depend on the collaboration among different operators in many cases—as it relies on integrating management information gathered from several PoPs. This fact motivates the reduction of the measurements

that are shared among actors, as they may disclose internal or critical information. To solve this potential issue, our algorithm reduces its input to only the values of the two characterizing moments of days that stood out during the previous inspection of diverse network dynamics—namely, both the moments with the lowest and highest load.

Second, using an analytical simplification of the high dimensional model, we propose to average the set of times when valleys occur for each single PoP of a path. This heuristic model will be less imprecise for paths traversing near PoPs, ergo located in close time zones and, intuitively, more homogeneous population. Section 5 delves into both solutions.

3.3. VTP operation

To conduct our experiments, we have developed a proof-of-concept of all these methods, denoted as *VTP*. The aforementioned characteristics of our solution allow *VTP* system to operate either in centralized or distributed fashions as presented in Figure 4.

In the former, it works as an oracle running in a specific PoP—e.g., a controller to whom the rest of PoPs may ask for scheduling. Alternatively, an instance of *VTP* runs in each PoP and by the time it requires scheduling, it takes the role of the scheduler.

The information that *VTP* needs are past values of the PoP valley hours (if it is operating in single-PoP scenario), valley hours and valley bandwidths (low dimensional model), and both valley and busy metrics (high dimensional model). In these two last cases, the measurements encompass all the PoPs involved in the request—i.e., the members of a path. If there is a PoP working such as an oracle, the information of each PoP of the topology can be retrieved and stored periodically, whereas, in the distributed operation the information is retrieved upon request.

With the required data, the scheduler node can carry out the estimation of the ideal moment for the transfer upon request. To do so, it predicts valley times, averages several valley times or aggregates traffic demands time series according to the particular scenario and model dimensionality (Section 3.2). Finally, *VTP* responds to the PoP that sent the request, and the transfer can be scheduled at the desired transmission rate.

As the prediction system input consists of only values for two prominent moments of days, we remark that each PoP only needs to track such values and be able to share with the oracle or the rest of PoP of the topology, which limits the cooperation to a minimum. Additionally, complexity reduces to the computation of a simple formula—instead of requiring the assertion and sharing of combinations of multiple time series in real-time, or using a vast amount of precomputed paths that represent the diversity of the Internet. Regarding the data volumes that are needed to build a predictor for *VTP*, we note that they may be flexibly adjusted according to to the availability of measurements and its periodicity. Here, and given that we have data for years, we used one year of daily measurements as default to fit the models and evaluate the system. However, lower data depths may be used with possible reductions in statistical power depending on the variability of the underlying measurements and the size of effects. In those cases, *VTP*'s predictor may converge to the fixed hour predictor.

3.4. VTP advantages

VTP's approach offers two main advantages when compared to other alternatives based on both machine and deep learning approaches [42, 43]. First, the typical periodicity of traffic profiles—explored in the previous sections—simplifies the study of complete-daily time series to the location of busy and valley periods. Second, the constructive philosophy behind our modeling offers an easy-to-interpret linkage between the input features and output model [44].

In this sense, we are more interested in the typical shapes of the demand curves than in the low-grain description of bandwidth/usage time series, which has been extensively studied in other works—such as [42, 43, 45], where the authors focused on the prediction of traffic in the scale of a few minutes, with applications to cellular communications using neural networks and exploiting the number, distribution and class of users in a base station or the saturation of the access and mobility management functions in 5G networks, respectively. Moreover, we have also tried to minimize assumptions and information needs to reduce dependencies on the availability of information apart from traffic time series. This distinguishes our solution from others such as [46], where the authors presented a predictive model for network traffic using flow classification by means of C4.5 trees and naïve-Bayes discretization.

While fine-grained models may have richer predictive capabilities, they also tend to be far more sophisticated and require much more measurements (i.e., data for training), input features which may not be available, computational capacity and, sometimes, offer results which may be difficult to interpret [47]. With this in mind, we believe that *VTP* can offer a good trade-off between predictive power and complexity, therefore reducing model complexity and, consequently, the risk of overfitting and computational cost of the learning and estimation processes.

4. Prediction of valley times in single-PoP scenarios

This section studies the dynamics of valley times, searching for diversities and homogeneities both in temporal and spatial sense—i.e., significant time over different PoPs. Then, we explore how to exploit the conclusions as a prediction tool for future valley times. To this end, we have comprehensively analyzed the set of measurements from RedIRIS that we presented before.

4.1. Patterns over time

While we have previously delved into daily and weekly patterns of traffic, we now consider valley time occurrence over time—i.e, time series of daily valley occurrences in local time. Figure 5(a) depicts an illustrative example corresponding to one of the PoPs under study.

As a starting point, we wonder if the daily occurrence of valley times has changed over a 6-year period in RedIRIS. To

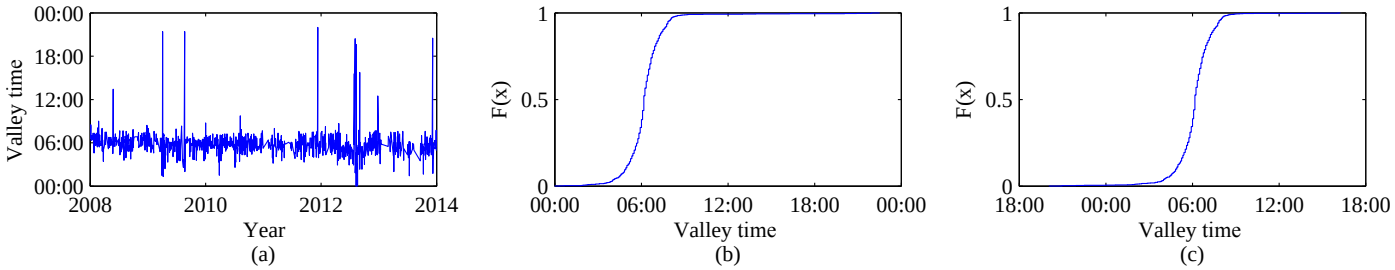

Fig. 5. Example for the daily valley time of a given PoP depicted as: (a) a time series, (b) a CDF and (c) CDF after translation.

Table 2
Linear correlation coefficient for Gaussian fit before and after time translation (→6 p.m.)

PoP	Before translation	After translation	PoP	Before translation	After translation
PoP1	0.71	0.90	PoP6	0.76	0.92
PoP2	0.76	0.96	PoP7	0.80	0.94
PoP3	0.83	0.95	PoP8	0.70	0.92
PoP4	0.75	0.92	PoP9	0.80	0.90
PoP5	0.68	0.86	PoP10	0.73	0.89

this end, we conducted a Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [48], which seeks to state whether the null hypothesis that an observable series is stationary around a trend can be rejected against the alternative of non-stationary.

Results show that stationarity of valley time occurrence cannot be rejected for a confidence level of 99% on all PoPs but PoP2 and PoP7. Moreover, all confidence intervals are roughly symmetric around 0. This suggests that most of the PoPs show a trend but that likely this trend is 0—i.e., a flat curve with no significant changes over time. Paying extra attention to the PoPs where the null hypothesis of stationary trend was rejected, we found that both presented occasional periods of instability related to spurious changes in their routing tables. Once such periods were ignored, PoP2 and PoP7 behave equivalently to others.

4.2. Modeling valley times

Summarizing the previous insights, we can state that (i) valley times tend to occur at a well-delimited temporal frame (early morning); (ii) they do not present significant trends over long periods; and (iii) the process exhibits intra-week and intra-day variations and significant heterogeneity between PoPs.

To model and capture this latter variability, we assess if the time series of valley-times occurrence can be considered as Gaussian processes. For several decades, some standard tests have been conducted to certify the best-fit model of a variable—e.g., Kolmogorov-Smirnov or Lilliefors. However, alternative tests to such standard ones have recently gained relevance on long-term networking data as being more adequate for such type of data [49]. Specifically, we use the correlation-test [50], where the linear correlation coefficient—ρ—between the quantile-quantile (Q-Q) plot and the order statistics of the sample should be a high value—namely, 0.9—for the acceptance of the null hypothesis that states that data come from a specific distribution.

The application of the test to valley-times occurrence time series are in Table 2 (second and fifth columns). It becomes apparent that outputs are not high enough to assert normality. Essentially, variable distributions are far from being symmetric. That is, a day has 1440 minutes; hence, the point of symmetry may be located to around 720 minutes. However, as previous sections show valley times tend to appear at early morning—i.e., in formal terms, a distribution with positive skew. That is because we have arbitrarily decided to conduct the study under the natural day schedule for humans—days start and end at midnight. However, there is not any reason to maintain such restriction in our analysis, so we can assume arbitrary day starts and ends to center valley times in the range and make their distribution symmetric.

Specifically, in our case we consider a day to start at 6 p.m., while this figure can vary according to different types of networks as explained in the previous sections. Figure 5 illustrates the process for a given PoP since data is captured, valley-hours calculated, and finally translated to a fairly symmetric-shaped process. The results of normality tests after translation are also available in Table 2. These results lead us to conclude that valley times are fairly Gaussian, as most of the figures are either close or exceed 0.9.

4.3. Factor analysis

Once valley times are modeled as a Gaussian process, we focus on how to explain its variance with measurable factors. This problem becomes more tractable as the factorial analysis of Gaussian processes has been analytically studied for a century although only recently exploited in the Internet area [51, 52]. Specifically, ANOVA is a statistical technique to analyze and explain measures from several simultaneous effects to decide which are significant and quantify their impact [53].

The factors that are accessible and are considered relevant in this study are the day of the week, weekday/weekend, day of the month, month, working/bank-holiday day and PoP—i.e., a 6-way model. However, there are several intuitive factors that are hardly available—e.g., the relevance of current-scheduled background traffic, number of users connected to each PoP, network adjustments over time, changes in workplaces or universities schedules or specific network outages, among others. These uncontrolled features will account for the unexplained variance, often named experimental error. Consequently, a significant experimental error is expected, however the interesting point is if, even so, the model is useful in practical terms as the following section evaluates.

Table 3
7-way ANOVA table including main effects and interactions between factors with valley times as dependent variable

Factor	Sum of Squares	df	Mean Square	F	p-value	Factors (%)	Num. levels	% Factors / Num. levels
Corrected Model	6113392	782	7818	2.40	0.00			
μ	337985852	1	337985852	103554.89	0.00			
Weekday (or weekend)	1556607	1	1556607	476.93	0.00	25.46	2	12.73
Bank holiday	8916	1	8916	2.73	0.10	0.15	2	0.07
Weekday * Bank holiday	0	0				0.00	4	0.00
Day of the week	58800	5	11760	3.60	0.00	0.96	7	0.14
PoP	982456	9	109162	33.45	0.00	16.07	10	1.61
Month	102894	11	9354	2.87	0.00	1.68	12	0.14
Day of the week * Bank holiday	5358	4	1340	0.41	0.80	0.09	14	0.01
Day of the week * Weekday	0	0				0.00	14	0.00
PoP * Weekday	215248	9	23916	7.33	0.00	3.52	20	0.18
PoP * Bank holiday	110649	9	12294	3.77	0.00	1.81	20	0.09
Month * Weekday	100129	11	9103	2.79	0.00	1.64	24	0.07
Month * Bank holiday	24916	5	4983	1.53	0.18	0.41	24	0.02
Day	119097	30	3970	1.22	0.20	1.95	31	0.06
Day * Weekday	163655	30	5455	1.67	0.01	2.68	62	0.04
Day * Bank holiday	68994	29	2379	0.73	0.85	1.13	62	0.02
PoP * Day of the week	125410	45	2787	0.85	0.74	2.05	70	0.03
Month * Day of the week	156684	43	3644	1.12	0.28	2.56	84	0.03
PoP * Month	698978	99	7060	2.16	0.00	11.43	120	0.10
Day * Day of the week	409316	108	3790	1.16	0.13	6.70	217	0.03
PoP * Day	887661	270	3288	1.01	0.46	14.52	310	0.05
Day * Month	317623	63	5042	1.54	0.00	5.20	372	0.01
Error	5463656	1674	3264					
Total	349562900	2457						
Corrected Total	11577048	2456						

R^2=0.528

We opt for an ANOVA iterative approach—often named as ANOVA Type I—, whereby the most general or simple factors—i.e., with the fewest number of levels—are firstly used to explain variance, and the most specific ones are considered progressively to explain the remaining variance. This approach tries to bound model complexity, as it dramatically grows when the number of levels increases. Moreover, more levels demand further data description, which jeopardizes scalability. For example, while the factor day of the week includes 7 possible levels—Sunday, Monday, $\cdots$, Saturday—, the factor weekday/weekend only has two possible levels—it is a weekday or not. This way, having only two model parameters—weekday and weekends—is preferable to a 7-parameter one—one per day of the week—assuming that the explained variance remains qualitatively equivalent. According to the above, we have applied ANOVA to data, including main effects and interaction between factors, as Table 3 shows, ordering factors and interactions from the lowest to highest number of levels, which is indicated in the eighth column.

The first six columns account for the typical ANOVA output. The sum of squares, in second column, gives an intuition of how important a factor is, and the sixth column states the p-value for the null hypothesis. Given a standard significance value of 0.05 or 0.1, below that level the null hypothesis is rejected—significant factor—, and accepted otherwise—non-significant factor. Additionally, the seventh column gives the percentage of variance explained by each term excluding the error. Finally, the last column shows a score of relevance, defined as the percentage of explained variance with respect to the number of levels.

We prominently find that weekday/weekend is both quantitatively and qualitatively the factor which has more effect in the variance, followed by PoP. If we pay attention to the percentage of variance explained, we notice that Day * Month and PoP * Day interactions have quantitative importance. However, a more detailed inspection revealed the reasons. They are a set of specific dates—e.g., New Year's Eve and Day, or Labor Day—, which behave so differently that affected the study. Similarly, PoP * Day reflects some very specific dates such as regional holidays. However, note that we are not interested in exceptional days, but in generality. Once such particular days were ignored, the factors become roughly non-significant. As a conclusion, by turning attention to the relevance score, only weekday/weekend and PoP show significant relevance. The former proves what we had already suspected from the visual review of previous sections, while the latter points to significant differences among PoPs' behavior, albeit being part of the same network.

4.4. Translation of factor analysis into a prediction tool

Given the results of the ANOVA analysis, we propose that *VTP* bases its estimation of valley times (VT) on the sum of parameter estimates:

$$\widehat{VT} = \mu + PoP + weekday + \epsilon \tag{2}$$

where μ represents a constant for all cases, the term PoP accounts for the particularity of each PoP, a third term that modifies valley time for weekdays/weekends and finally a term for the unexplained variance—assumed to be normally distributed with zero mean and constant variance.

We evaluate the accuracy of the system by estimating the ANOVA parameters of the sample for one year, using them to predict the valley time for the following year and, finally, contrasting them with the real values measured—i.e., how accurate the model is as a predictor for the following year. We note that

other time intervals for the training phase can be considered, both longer or shorter ones so being more precise or simpler, respectively. Actually, optimal duration for training can be carried out with machine-learning techniques [43]. However, we will show in the following that the key to provide an useful prediction tool is not in the exact precision of this phase. Anyhow, the results showed that for 50% of the samples, the error obtained is less than 40 minutes; for the 80th percentile, less than 70 minutes; and for 90% of samples, less than 100 minutes. If we compare these results with the null model, i.e., the total mean of the data aggregate, our factorial system reduces errors by 5% in the 50th percentile; by 27% for the 80th percentile; and 13% in the 90th percentile.

5. Prediction of Valley Times in multi-PoP scenarios

This section details how *VTP* copes with the estimation of valley times of multi-PoP scenarios. Remarkably, two significant differences arise from the single-PoP study.

First, the valley time of multi-PoP scenarios may depend both on the individual shapes and on the specific measurement of traffic volumes. For example, given a path involving two PoPs, one with some Gb/s bandwidth on average and another with marginal traffic, the impact of both in the resulting meta-PoP valley extremely differs—the resulting aggregate valley time will occur according to the first PoP. Similarly, if relevance is measured as link utilization instead of absolute throughput, a high-utilized link will dominate the resulting aggregated time series, and therefore the valley time occurrence.

Second, when paths traverse different time zones, both national—in countries with more than one time zone—or international scenarios, the typical busy and valley times can be dramatically different. In this case, we are going to use the RedIRIS' measurements, as they follow actual bandwidth time series patterns, but shifted to represent time series of other parts of the world—so addressing the problem not only in a nationwide scale but closer to global Internet.

Let us first provide a visual inspection of these two issues, and then, detail our two different approaches to them.

5.1. Visual inspection of the problem

Figure 6 depicts two PoPs bandwidth times series and its resulting aggregation (the darkest color), as well as the valley times for each of them as vertical lines, and also the global valley for a path traversing both PoPs (as dashed lines). Progressively, one of the bandwidth time series is shifted to illustrate different time zones—from 2 to 10 hours. For the sake of visualization, both time series are scaled representations of the same PoP.

Interestingly, when the time offset is not significant, the aggregated shape and valley times are only barely altered. However, as the time shift increases, individual valley times and the aggregated one start to separate from each other, and the resulting curve shape tends to flatten as a result of the overlap of individual traffic shapes, making it difficult to detect a clear, sole convex zone as the valley time. In fact, in these cases several modes and antimodes may appear—i.e., two or more dips. Moreover, the steepness of the slopes before and after the valley time is also altered.

This example assumes as input two PoPs of equal relevance, although marked imbalance among nodes is expectable in the Internet. This behavior appears in traffic statistics of large IXPs that span several time zones—e.g., Japan and the U.S.[3], where one of the time zones dominates the shape of the meta-PoP.

As a conclusion, the resulting aggregated shape will be an interaction of the particular bandwidth, shapes, and time zones involved in each PoP being part of a path. While the time zone is a simple data and the bandwidth on valley and busy periods are measured by our tool *VTP*, in the multi-PoP approach, the particular traffic shape of each PoP needs to be modeled to span all the phenomenon interactions.

5.2. General high dimensional model

Bearing in mind the results of the previous exploratory analysis of multi-PoP scenarios, we define a method to represent their variety with a high dimensional model. We recall that, while both the busy and valley hours of single PoPs can be easily characterized by the moments of highest and lowest activity, this is not the case for multi-PoP scenarios—as stated before, the latter presents complex behaviors depending on interactions among the individual activity profiles.

To overcome this matter, we follow a constructive approach in which the global behavior emerges from the single PoP ones. Our solution can fit a wider range of situations, hence simplifying the detection of the period of minimum global activity. Remarkably, this approach overcomes the problems that arise during the definition of low activity periods in scenarios where PoPs in different time zones are involved. As previously stated, this is a relevant factor for scheduling in operational networks, as human behavior is tied to the hour.

Our model needs that the following hypothesis hold:

- There exists a usual activity pattern for each of the PoP involved in the multi-PoP path.
- Such pattern presents both global maximum and minimum instants, and high and low activity periods.

The first hypothesis is required to ensure the definition of a usual activity pattern for the multi-PoP path. The second one is extracted from the previously presented findings and defines the constraints for the shape of the high-dimensional baseline that we derive below. We use a functional model [54], as it is an approach capable of capturing very detailed behavioral patterns, to obtain such a high-dimensional estimation of the PoP behavior with respect to time. We also consider a smoothed version of such behavior, as we are not interested in the detection of short periods of low activity.

Our objective is to obtain a parameterizable function, $F(t)$, $t \in \mathbb{T}$ where $\mathbb{T}$ represents the time of day, which fairly fits the daily behavior of the PoP while requiring a minimal

[3] JPNAP Osaka Service `http://www.jpnap.net/english/jpnap-osaka/traffic.html`

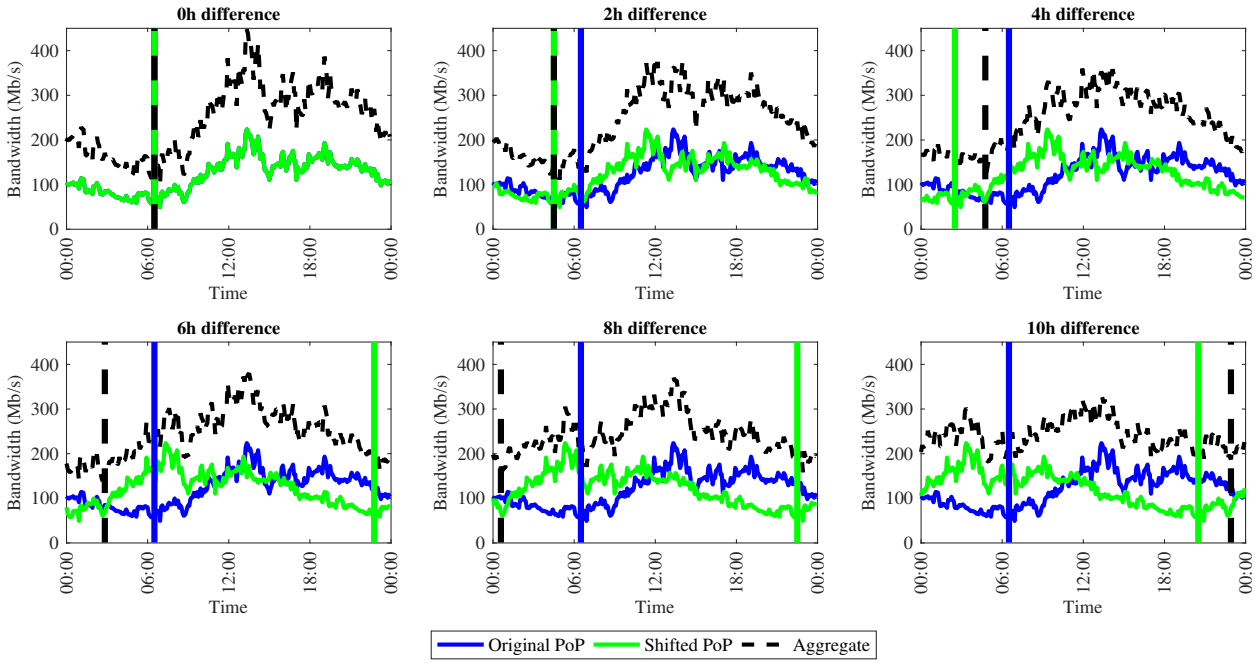

Fig. 6. Comparison of time zones and their effect on time series and valley moment of a path comprehending two identical PoPs.

parameter space. Additionally, $F(T)$ must reflect the characteristic transient between low and high activity regions and fulfill the previous hypothesis.

With this in mind, we define the constraints in (3) to obtain a smooth function $F(t)$ with maximum and minimum in BT and VT—i.e., the busy and valley times, respectively.

$$\begin{cases} BW_B = \max(F(t),\ t \in \mathbb{T}) = F(BT) \\ BW_v = \min(F(t),\ t \in \mathbb{T}) = F(VT) \end{cases} \tag{3}$$

We note that these constraints imply that:

$$\frac{dF(t)}{dt} = 0,\ \text{when } t = BT,\ VT \tag{4}$$

On the one hand, we can rescale $F(\cdot)$ with a linear transformation to obtain $\hat{F}(\cdot)$ such as its values are bounded in the interval $[-1, 1]$ following (5):

$$\hat{F}(t) = \frac{2F(t) - (BW_B + BW_v)}{(BW_B - BW_v)} \tag{5}$$

On the other hand, and without loss of generality, we consider a transformed time domain $H : t \in \mathbb{T} \rightarrow \tau \in [0, 1]$, with the properties in (6) to simplify the model adjustment:

$$\begin{cases} H(BT) = \tau_B = 1 \equiv 0 \\ H(VT) = \tau_v \end{cases} \tag{6}$$

This formulation implies that the description of its behavior can be summarized with the 4-tuple (BT, VT, BW_B, BW_v). Specifically, (BW_B, BW_v) are required to define the scaling $\hat{F} \rightarrow F(t)$, while (BT, VT) determine the transformation H.

Taking into account these properties and the oscillatory behavior of observed network loads, we accomplish a constructive definition of $\hat{F}(t) = (\hat{F} \circ H^{-1})(\tau)$ using a sine/cosine restricted to a bounded interval as basic function. Such a model represents the PoP state as:

$$\hat{F}(t) = (\hat{F} \circ H^{-1})(\tau) = \cos(\omega(\tau)\tau),\ \tau \in [0, 1] \tag{7}$$

This function—i.e., a basic cosine function with a time-dependent frequency—improves the fitting of the model to the observations—e.g., the variation of the frequency with respect time makes possible the introduction of asymmetry in the low activity period, as previous observations suggest.

As we are considering only two parameters to adjust the model (namely, τ_B and τ_v) we use a linear function to represent $\omega(\tau)$. We define a convex combination between two extreme values, $\{\omega_i\}_{i=1,2}$ following (8):

$$\omega(\tau) = \omega_1\tau + \omega_2(1 - \tau),\ \tau \in [0, 1] \tag{8}$$

Now, we recall that:

$$\begin{cases} \hat{F}(\tau_B) = \hat{F}(1) = \cos(\omega_1) = 1 \\ \hat{F}(\tau_v) = \cos(\omega(\tau_v) \cdot \tau_v) = -1 \end{cases} \tag{9}$$

Applying the arccos function to both expressions, we obtain a family of parametric solutions given by:

$$\begin{cases} \omega_1 = (2k_1)\pi,\ k_1 \in \mathbb{Z} \\ \omega(\tau_v) = \dfrac{(2k_2 + 1)\pi}{\tau_v},\ k_2 \in \mathbb{Z} \end{cases} \tag{10}$$

Hence, we may select $\{k_1, k_2\}$ for each specific value of τ_v so that the fitting is optimal. To do so, we can consider the behavior of $\hat{F}(t)$ and its first derivative, to fix the number of relative maximums and minimums to three —i.e., the busy and valley periods given the periodicity of the model. A grid optimization

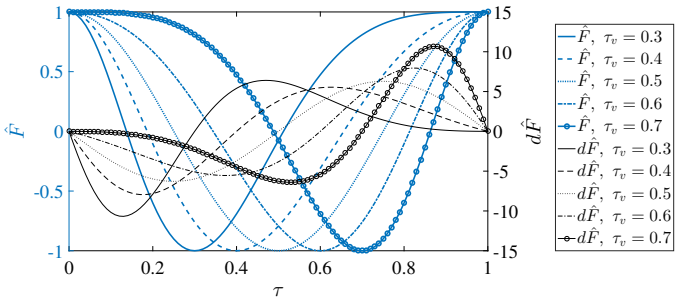

Fig. 7. Behavior of model and derivative, for different values of τ_v.

Algorithm 1 Location of values for $\{k_1, k_2\}$.

Input: τ_v
Output: $\{k_1, k_2\}$
Initialization:
$k_1 = 0$
Grid optimization:
repeat
  if $(\tau_v > 0.5)$ **then**
    $k_1 = k_1 - 1$
    $k_2 = \text{floor}(k_1 \cdot \tau_v + (1 - \tau_v) \cdot \tau_v - \frac{1}{2})$
  else
    $k_1 = k_1 + 1$
    $k_2 = \text{ceil}(k_1 \cdot \tau_v - (1 - \tau_v) \cdot \tau_v - \frac{1}{2})$
  end if
  $\omega_1 = 2 \cdot \pi \cdot k_1$
  $\omega_2 = \frac{((2 \cdot k_2 + 1) - 2 \cdot k_1 \cdot \tau_v^2)}{((1 - \tau_v) \cdot \tau_v)}\pi$
  $\omega(\tau) = \omega_1 \cdot \tau + \omega_2 \cdot (1 - \tau)$
  $(\hat{F} \circ H^{-1})(\tau) = \cos(\omega(\tau) \cdot \tau)$
  $\frac{d}{d\tau}(\hat{F} \circ H^{-1})(\tau) = -\sin(\omega(\tau) \cdot \tau) \cdot \frac{d}{d\tau}(\omega(\tau) \cdot \tau)$
until $(|\{\tau : \frac{d}{d\tau}(\hat{F} \circ H^{-1})(\tau) = 0\}| \leq 3)$
return $\{k_1, k_2\}$

formulation of such an approach is presented in Algorithm 1, where k_2 is adjusted from the value given to k_1. In this case, the algorithm explores different values of k_1 starting from 0, to facilitate the update of parameters depending on the position of τ_v with respect to 0.5. We note that, in some extreme cases, this algorithm may not be able to adjust $\{k_1, k_2\}$ to suitable values. Computational evaluation supports the application in all the cases observed in Section 2, where there are more than six hours between the valley and busy times; and provided good results even when $min(\tau_v, 1 - \tau_v) > 0.2$ if some tolerance to derivative cancellation is added. This is a consequence of the linear changes of the frequency, which is not adequate for fast transitions between the low and high activity regions—requiring approximately four hours with our formulation. This is illustrated in Figure 7, where we plot the behavior of $\hat{F}(t)$ and $\frac{d\hat{F}(t)}{dt}$ for different values of τ_v. Note that the amplitude of $\frac{d\hat{F}(t)}{dt}$ increases when τ_v lies far from 0.5, which explains the update rule in the algorithm.

In the following, and with illustrative purpose, we select the next specific values for simplicity:

$$\begin{cases} \omega_1 = 2\pi \\ \omega(\tau_v) = (\omega_1 \tau_v + \omega_2(1 - \tau_v)) = \frac{\pi}{\tau_v} \end{cases} \tag{11}$$

We remark that this is the solution that we used during the experimental performance assessment of the model because it fitted all the PoPs in our dataset.

Substituting the value of ω_1, we can get the value of ω_2:

$$\omega_2 = \frac{(1 - 2\tau_v^2)\pi}{(1 - \tau_v)\tau_v} \tag{12}$$

With the expressions in (5), (7) and (8), the final model expression is given in (13):

$$\begin{cases} \hat{F}(t) = & (\hat{F} \circ H^{-1})(\tau) \\ & = \cos((2\pi\tau + \frac{(1 - 2\tau_v^2)\pi}{(1 - \tau_v)\tau_v}(1 - \tau))\tau) \\ F(t) = & \frac{(1 + \hat{F}(t))(BW_B - BW_v)}{2} \\ & + BW_v \end{cases} \tag{13}$$

Figure 8 presents an example of the resulting curve adjusted to the parameters of a randomly chosen daily observation of the traffic in one of the PoP under study. This illustrates how our model fairly represents the behavior extracted from the findings in the previous sections.

Once we have derived a function to estimate the load in single-PoP scenarios, we define the *set of parameters for a multi-PoP path, $\mathcal{P}$,* as the set of 4-tuples corresponding to the PoPs that conform the path:

$$\begin{aligned} \mathcal{P} = & \{P_i\}_{i=1,\dots,N} \\ = & \{(t_{P_i}, t_{v_i}, BW_{B_i}, BW_{v_i})\}_{i=1,\dots,N} \end{aligned} \tag{14}$$

As each element P_i is linked to a specific PoP, $\mathcal{P}$ is implicitly describing the *set of activity functions for the multi-PoP path*, $\mathcal{F} = \{F_{P_i}\}_{i=1,\dots,N}$. We note that this approach allows mixing data from different time zones if the time domain of the model, τ, which encapsulates each specific time transformation H_i for $\{F_{P_i}\}$, is reverted into a shared time space, t', taking into account the relation among the original domains. With this in mind, we abuse of notation and consider this latter shared time domain in our formulation—omitting the corresponding transformation for the sake of readability.

Following the abstraction of a multi-PoP path as a meta-PoP by aggregation, we then add the individual loads as in (15):

$$S_{Path}(t') = \sum_{i=1}^{N} F_{P_i}(t'), t' \in \mathbb{T} \tag{15}$$

which is a particular case of the aforementioned general expression, and aims to reconstruct the cumulative throughput for all the PoPs in the path. With this formulation, the prediction of the valley time for the path is the argument that minimizes the value of $S_{Path}(t')$.

Remarkably, $S_{Path}(t')$ provides an approximation that requires as input parameters only the set $\mathcal{P}$. Therefore, this

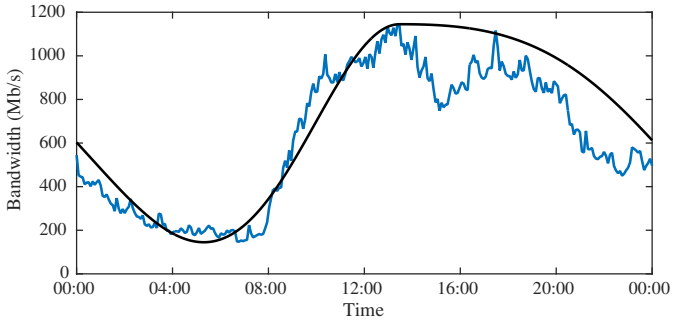

Fig. 8. Example of the high dimensional model, adjusted to the behavior of one of the trajectories.

model is able to estimate the global behavior of a path with minimal information, consisting of two relevant points for each node. Moreover, this description provides a straightforward error function to evaluate different estimations of the valley time for a path. Specifically, let VT_{Path} be the actual valley time and $\widehat{VT}_{Path}$ an estimation of that value. We can define an absolute error function $C(\cdot)$ as shown in (16):

$$C(\widehat{VT}_{Path}) = S_{Path}(\widehat{VT}_{Path}) - S_{Path}(VT_{Path}) \tag{16}$$

We remark that this error function implicitly defines an equivalence relation for estimations, as displayed in (17):

$$\begin{aligned} &\widehat{VT}^1_{Path} \equiv \widehat{VT}^2_{Path} \iff \\ &C(\widehat{VT}^1_{Path}) = C(\widehat{VT}^2_{Path}) \end{aligned} \tag{17}$$

where $\{\widehat{VT}^1_{Path}, \widehat{VT}^2_{Path}\}$ are two different estimations of the valley time of the path. Additionally, (16) can be expanded to obtain an error bound for estimations using (5), as shown in (18):

$$\begin{aligned} C(\widehat{VT}_{Path}) = & \sum_{i=1}^{N} F_{P_i}(\widehat{VT}_{Path}) - \sum_{i=1}^{N} F_{P_i}(VT_{Path}) \\ = & \sum_{i=1}^{N} [\frac{(BW_{B_i} - BW_{v_i})}{2} \\ & (\hat{F}_{P_i}(\widehat{VT}_{Path}) - \hat{F}_{P_i}(VT_{Path}))] \end{aligned} \tag{18}$$

The relation in (17) and error bound in (18) entail that two very dissimilar estimations may provide fairly similar results. The evaluation of *VTP* takes into account this matter, as we discuss in the following section.

5.3. Low-dimensional heuristic for PoPs with similar valley times

Once we have provided a general formulation to solve the problem, we focus on a particular and relevant case in which our method can be reduced to a low-dimensional and simpler estimation: scenarios in which the valley of all the PoPs are near the aggregated valley.

Equation (18) shows that the error for estimations of the path's valley time can be expressed in terms of the differences of the values of $\hat{F}_{P_i}(\cdot)$. With this in mind, we now focus on cases where the valley times of every PoP in the path (VT_i) are in a neighborhood of VT_{Path} and $\widehat{VT}_{Path}$. In such cases, we may consider the Taylor series of $\hat{F}_{P_i}(\cdot)$ centered in VT_i, given in (19):

$$\begin{aligned} \hat{F}_{P_i}(t') = & \hat{F}_{P_i}(VT_i) + \\ & \sum_{j=1}^{\infty} \frac{1}{j!} \frac{d^j \hat{F}_{P_i}(VT_i)}{dt^j} (t' - VT_i)^j \\ = & -1 + \sum_{j=1}^{\infty} \delta_{i,j} (t' - VT_i)^j \end{aligned} \tag{19}$$

Hence, if (19) holds, we may rewrite $C(\widehat{VT}_{Path})$ obtaining the expression in (20):

$$\begin{aligned} C(\widehat{VT}_{Path}) = & \sum_{i=1}^{N} \{ \frac{(BW_{B_i} - BW_{v_i})}{2} \\ & \sum_{j=1}^{\infty} \delta_{i,j} [(\widehat{VT}_{Path} - VT_i)^j - (VT_{Path} - VT_i)^j] \} \\ = & \sum_{i=1}^{N} \{ \frac{(K_i - 1)(BW_{v_i})}{2} \\ & \sum_{j=1}^{\infty} \delta_{i,j} [(\widehat{VT}_{Path} - VT_i)^j - (VT_{Path} - VT_i)^j] \} \end{aligned} \tag{20}$$

where $K_i = \frac{BW_{B_i}}{BW_{v_i}}$. With this expression we designed a simplified heuristic: *if the effects of* $(K_i - 1)$ *and* $\delta_{i,j}$ *are not considered, the higher* BW_{v_i} *is, the nearer* $\widehat{VT}_{Path}$ *should be to* VT_i *to minimize* $C(\widehat{VT}_{Path})$. In such manner, the model can omit the local variations with time ($\delta_{i,j}$) and the value in the peak hour (K_i), reducing its input to the intrinsic characteristics of the valley moment—i.e., amplitude, BW_{v_i}, and position.

Therefore, if every individual valley time $VT_i, i = 1, \ldots, N$ is in a neighborhood of VT_{Path}, we can apply a reduced version of the prediction model that provides an estimation $\widehat{VT}_{Path}$:

- Which is in a neighborhood of every VT_i.
- Whose distance to VT_i is inversely proportional to BW_{v_i}.

The first condition holds if $\widehat{VT}_{Path}$ is in the convex hull of the individual valley times, and taking into account the second one, we obtain the estimation in (21):

$$\widehat{VT}_{Path} = \frac{\sum_{i=1}^{N} (VT_i \cdot BW_{v_i})}{\sum_{i=1}^{N} BW_{v_i}} \tag{21}$$

where VT_i and BW_{v_i} are the valley time and average bandwidth during the valley-hour for each PoP i, respectively.

This reduction is also supported by the inspection of our measurements. Paying attention to Figure 6, it turns out that the aggregate valley time usually occurs between—or at least close if more than two PoPs are involved—the single valley times of each of the PoPs when time scales are not much different. In addition, a second characteristic that we can infer is that the path's valley time bias towards the single valley time of one or other PoP depending on bandwidth. This approach can be a good approximation for paths with few intermediate nodes and,

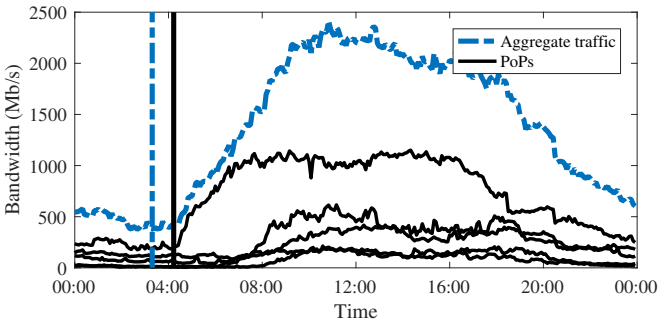

Fig. 9. Effect on the valley times for a path of five PoPs with limited time-zone differences.

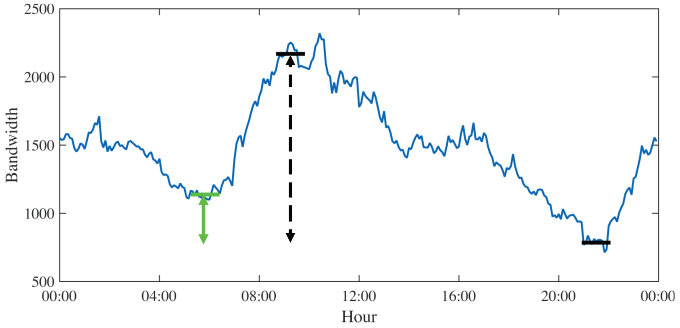

Fig. 10. Example of errors in terms of bandwidth and time for a given PoP. The actual valley time is shown as an horizontal line in dark color and the estimation as an horizontal line in light color. The amplitude bandwidth is depicted with a dashed arrow and the prediction error as a continuous arrow.

especially, when the time difference between them is limited. In these cases, individual peaks are not crucial because they introduce few variations in the global valley, since they are located far from the valleys.

As a first visual example of a prediction using this latter model, Figure 9 shows its application for the case of a 5-PoP path. The vertical lines indicate valley times: as a dashed line the aggregate ideal, and, as a continuous line, the time predicted by the low-dimensional method. The difference, in minutes, between them two is about half an hour. More importantly, note that the difference in terms of bandwidth between the two moments is marginal—which is the most relevant aspect to schedule a transfer. The rationale behind this is that such time shift is small enough that does not make prediction fall outside of the off-peak interval of the aggregate curve. In this way, this approximation gains relevance in cases such as nationwide networks—i.e., close time zones and similar patterns.

6. Evaluation of *VTP*

Once we have presented the analytical foundations of *VTP*, we pursue to measure to what extent it can result useful to network managers. To do so, we have conducted an experimental evaluation with a twofold orientation. First, we focus on the system's capacity to accurately estimate the period of minimal activity and its improvements when compared to other scheduling strategies which are based on fixed-times. Second, we analyze how the scheduling of bulk transfers would affect charges in the common 95th percentile-based billing scheme.

6.1. Accuracy of valley time prediction

As the previous sections have explained, the bandwidth shape of a multi-PoP node may show a diverse range of behaviors. From the typical shape with clear valley and busy times to shapes with more than one valley including, even, flat shapes. In this latter scenario, it is worth remarking that a prediction of a valley time deviated for many hours from the actual valley time can still be very precise in bandwidth terms—and so being a good moment to carry out bulk transfers as it is a period of low use.

For example, in a strictly flat shape, any prediction of the valley time is equally good. In almost-flat shapes, although a specific moment would be the ideal valley time, the impact of errors on the prediction is negligible—likely, only a few Mb/s.

Note that the same does not apply to a single PoP, where clear valleys tend to appear in the shape of bandwidth time series, and, then, a shift in the estimation of several hours implies large error in terms of Mb/s. Given that, evaluation should not consider differences between predicted and actual valley times, but how much throughput such difference entails.

Specifically, we define the prediction error as the difference between the average bandwidth during the estimated and actual valley times. Then, the prediction error ratio emerges, after its normalization by the actual amplitude bandwidth (busy minus valley), as a mechanism to make comparable the error between PoPs with different traffic aggregates—e.g, 100 Mb/s and 10 Gb/s links. That results in the expression:

$$Prediction\ error\ ratio = \frac{BW_{\widehat{VT}_{Path}} - BW_{VT_{Path}}}{BW_B - BW_{VT_{Path}}} \quad (22)$$

This way, a valley time prediction just on the busiest time will output an error of 1, and, conversely, when the actual and predicted valley times overlap the error is zero. In between these figures, intermediate scenarios with low error values will point out to good predictions and those with high errors will suggest unsuccessful operation of *VTP*.

Figure 10 illustrates such measurements where the predicted valley time was about 6 a.m. and the actual time was 9 p.m. The continuous vertical arrow shows the error of the prediction in terms of bandwidth, and the amplitude of the curve is shown as a vertical dashed arrow. It becomes apparent that although the prediction can fail by hours—about 15 hours in this example—, the error in terms of Mb/s between the actual valley and the estimated one is not that significant. In other words, the key is not in being strictly close to the moment the valley time occurs, but to be close in Mb/s to the bandwidth during the valley time.

To provide an evaluation for diverse scenarios, we create 1000 random paths that span a diverse number of hops, time zones, traffic shapes and bandwidth amplitudes. We used RedIRIS time series as basic data, and apply transformations—i.e., time shifts and linear scaling—with randomly selected transformations to obtain a controlled but rich dataset for the experimental evaluation. It is worth remarking that this setup

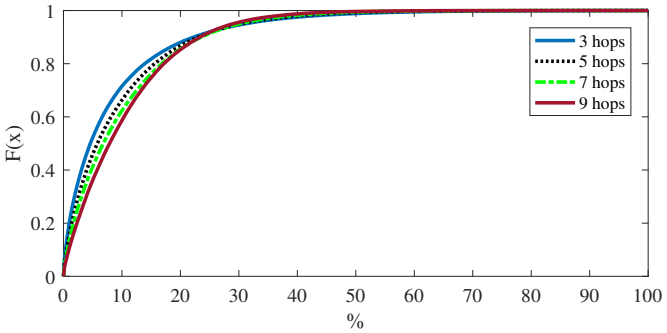

Fig. 11. Cumulative distribution of the prediction error rate for different number of hops.

defines a challenging scenario for *VTP*. Specifically, other types of networks—e.g., IXP o enterprise ones—typically present deeper and more symmetric valley times than those from academic networks, as shown in Section 2.2.

In more detail, each path is created with a number of hops between 3 and 9 hops—according to [37] where hops at autonomous system scale are measured; a random time-zone location to every PoP between 0h and +/-12h UTC—to span all the globe; and for the base shapes, we used one year of aggregate time series from RedIRIS. Finally, the actual valley times are calculated for this evaluation set of paths and the *VTP* system, based on both the high and the low dimensional model, applied.

Figure 11 shows the prediction error for different number of hops with the high-dimensional model. It becomes apparent that accuracy is only marginally sensitive to the number of hops, obtaining a 50^{th} percentile of the error of nearly 10% to the ideal moment, and 90^{th} percentile of error of about 25%.

On the other hand, Figure 12 depicts the errors for different time shifts for both methods with respect to the actual valley time. For comparison purposes, the figure also shows the prediction of applying a fixed-time method. This latter method represents the trivial and common approach of transmitting during the low-use period of the source PoP, ignoring the rest of the path. In particular, we consider that this method knows perfectly the valley time of the PoP that is the source in the path—but not of others—and uses this period.

For a topology deployed within the same time zone, both the general model and the simplified methods are equivalent in quantitative terms, and even the fixed-time approach is still valid. Nonetheless, as we extend the time separation between PoPs, the low-dimensional approach progressively worsens, and the fixed-time counterpart achieves dramatic errors for each time boundary—in many cases, similar to randomly scheduled transfers.

These results suggest that 5h-shifts between PoPs may be considered the threshold for the validity of the low-dimensional approach. Beyond that, the increase of the relative error of such method clearly tips the scales in favor of the high-dimensional model. Given these figures, the weighted-average method is still suitable for links such as those from Europe or Africa to East Coast of South America, between PoPs of Asia and Oceania, or US coast-to-coast paths and, in general, intra-continental links. For longer distances, the general model is much more adequate.

6.2. Impact on the daily 95^{th} percentile

Finally, we also evaluate *VTP*'s impact on the daily 95^{th} percentile, as those users charged by this mechanism may incur in additional costs when carrying out bulk transfers. Essentially, our procedure is to take per-PoP traffic time series, add the traffic of a bulk transfer according to *VTP* scheduling, and assess how 95^{th} percentiles vary. Transfers will be centered in the predicted valley time, regardless of their volume.

We set out different scenarios where PoPs generate traffic volumes between 10% and 30% of their daily traffic exchanged as additional bulk transfers—exceeding worst-case scenarios reported in many data centers [38]. Within our measurement dataset, these percentages correspond to volumes in a range from 30 GB to 500 GB. We assume CBR bulk transfers at an intermediate level between the traffic on busy and valley times, specifically the median of the traffic-demands curve, so avoiding strong synchronization between PoPs.

Then, we generate 1000 random paths with 5 PoPs, estimate their aggregate valley times, choose the transmission rate as the median throughput for the PoP with lower traffic intensity in each path, append the new bandwidth to the times series of each PoP involved, and, finally, measure changes in the five PoPs' 95^{th} percentiles separately. Then, we add the new 95^{th}-percentile samples obtained—the traffic volume that will represent a cost—and divide the outcome by the sum of the original 95^{th} percentiles. This results in a ratio that expresses the fraction of traffic that will entail an extra expense with respect to the original scenario—i.e., without bulk transfers.

The outputs of this experiment showed that *VTP* increases the 95^{th}-percentile aggregate only between 1% and 4%, which proves its effectiveness. To put these figures into perspective, we compare them with the ratios resulting of applying the fixed-time scheduling for the source PoP of each path, as in the previous section. Interestingly, the fixed-time approach performs between 2 and 4 times worse than *VTP*.

7. Conclusions

Throughout this paper, we have shed light on the dynamics of valley time phenomenon—those periods of the lowest bandwidth demand in a network. We have applied this knowledge to the design of *VTP*, a prediction system that suggests a time to carry out bulk transfers—although, potentially, other batch-processing task may also exploit it. *VTP* constructs demand curves based on the busy and valley times observing the particularities of the shape of traffic demands observed in this paper. Specifically, a convex shape for periods of low intensity and a concave counterpart for busy periods, but with different durations, and even different slopes before and after valley times among other aspects to match the typical behaviors of real networks. Our solution makes use of a multi-level modeling approach, which starts with the explanation of dynamics in single-PoPs. Then, it builds an estimation of the behavior of the ag-

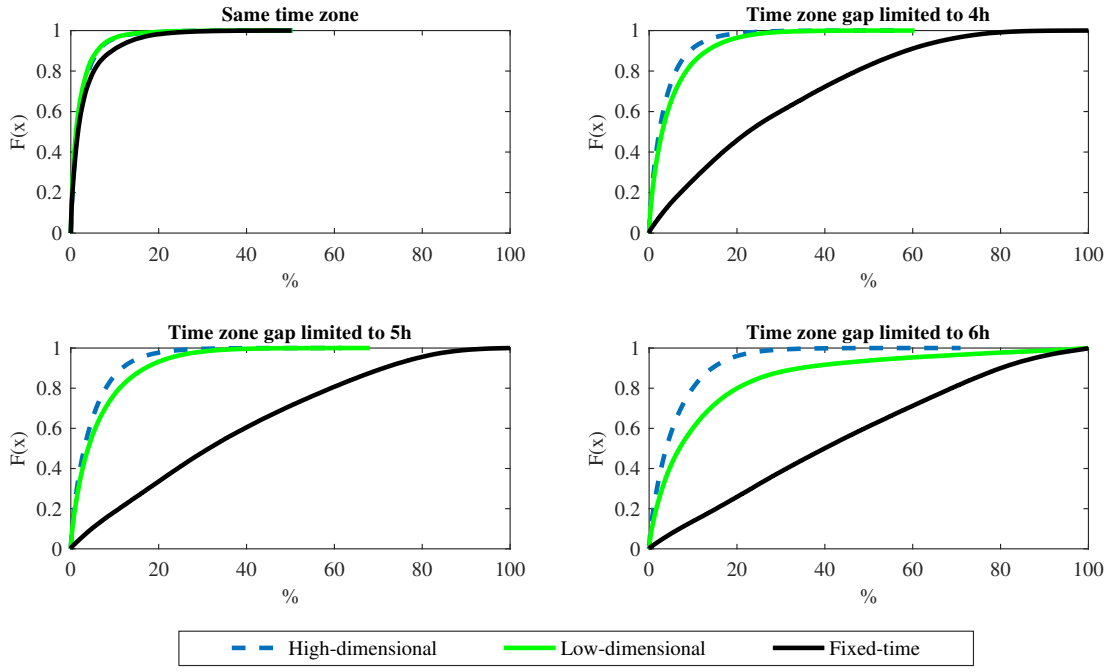

Fig. 12. Cumulative distribution of the prediction error ratio for different approaches.

gregation of many PoPs to approximate the behavior of multi-PoP scenarios. We rely on the collaboration among PoPs in the search of a trade-off solution that enables scheduling transfers in a suitable moment in terms of the global behavior of aggregated traffic.

Regarding single-PoP scenarios, we have found that valley times over time can be described as a Gaussian process, which makes it possible to extract factors that describe the phenomenon and predict the valley-time occurrence.

When modeling multi-PoP scenarios, we have found relevant both when the valley time occurs and the specific interaction of the volumes of bandwidth measured in each PoP. By observing this, we formulated a method to construct curves that fairly overlap daily measurements. Then, our proposal estimates the moment of minimum global utilization with the aggregation of such curves. Moreover, and as a step towards simplification, we found that when the valley times of PoPs in a path are near, the model can be reduced to a weighted average of the individual valley times.

The results of *VTP* for single-PoP scenarios show errors below an hour. For multi-PoP scenarios based on a diverse set of RedIRIS time series—adjusted with different time scales and shifts to provide diversity—divergences with respect to the optimal moment are typically below 10%, whereas alternative approaches provided results close to randomness. Our tests also show a noticeable improvement regarding costs—between 2 and 4 times better than a fixed-time approach. Therefore, *VTP* arises as a promising mechanism to help network managers in their tasks of scheduling bulk transfers.

We plan to assess the impact of assuming a non-symmetric convex shape—e.g., a certain degree of skewness—for valleys, and using variable rates for the bulk transfers instead of CBR. Certainly, transmissions centered on valley times showed significant results and this characteristic would add complexity to the model; however, finding some point of compromise may result in a more precise version of *VTP*.

Acknowledgments

This work has been partially supported by the European Commission under the project H2020 METRO-HAUL (Project ID: 761727).

References

[1] I. B. M. z/OS Basic Skills Information Center, Mainframe concepts, White paper, 2008.
[2] Z. Wu, H. V. Madhyastha, Understanding the latency benefits of multi-cloud webservice deployments, ACM SIGCOMM Computer Communication Review 43 (2) (2013) 13–20.
[3] J. Yao, P. Lu, L. Gong, Z. Zhu, On fast and coordinated data backup in geo-distributed optical inter-datacenter networks, Journal of Lightwave Technology 33 (14) (2015) 3005–3015.
[4] N. Laoutaris, G. Smaragdakis, R. Stanojevic, P. Rodriguez, R. Sundaram, Delay-tolerant bulk data transfers on the Internet, IEEE/ACM Transactions on Networking 21 (6) (2013) 1852–1865.
[5] Z. Feng, W. Sun, F. Li, W. Hu, Efficient elastic bulky traffic transfer with a new pricing scheme based on number of flows, Computer Communications 67 (2015) 45–55.
[6] X. Dimitropoulos, P. Hurley, A. Kind, M. P. Stoecklin, On the 95-percentile billing method, in: Conference on Passive and Active Network Measurement, 2009, pp. 207–216.
[7] G. Tselentis, J. Domingue, A. Galis, A. Gavras, D. Hausheer, Towards the future Internet: A European research perspective, IOS Press, 2009.
[8] J. L. García-Dorado, Bandwidth measurements within the cloud: Characterizing regular behaviors and correlating downtimes, ACM Transaction Internet Technology 17 (4) (2017) 39:1–39:25.
[9] M. Marcon, N. Santos, K. P. Gummandi, Netex: Cost-effective bulk data transfers for cloud computing, Tech. rep., Tech. rep., Max Planck Institute for Software Systems (2012).

[10] J. L. García-Dorado, J. A. Hernandez, J. Aracil, J. E. Lopez de Vergara, F. J. Monserrat, E. Robles, T. P. de Miguel, On the duration and spatial characteristics of Internet traffic measurement experiments, IEEE Communications Magazine 46 (11) (2008) 148–155.
[11] M. K. Chowdhury, R. Boutaba, A survey of network virtualization, Computer Networks 54 (5) (2010) 862–876.
[12] N. Laoutaris, M. Sirivianos, X. Yang, P. Rodriguez, Inter-datacenter bulk transfers with NetStitcher, in: ACM SIGCOMM, 2011, pp. 74–85.
[13] V. Gupta, What is network planning?, IEEE Communications Magazine 23 (10) (1985) 10–16.
[14] RedIRIS, the Spanish Research and Education Network, https://www.rediris.es/index.php.en.
[15] S. Albandea, J. L. García-Dorado, D. Muelas, J. E. López de Vergara, J. Aracil, Valley times in the Spanish academic network, in: IFIP/IEEE Symposium on Integrated Network and Service Management, 2017, pp. 564–567.
[16] J. L. García-Dorado, J. A. Hernández, J. Aracil, J. E. López de Vergara, S. Lopez-Buedo, Characterization of the busy-hour traffic of IP networks based on their intrinsic features, Computer Networks 55 (9) (2011) 2111–2125.
[17] Forrester Research, The future of data center wide-area networking, http://www.forrester.com (2016).
[18] R. Stanojevic, N. Laoutaris, P. Rodriguez, On economic heavy hitters: Shapley value analysis of 95th-percentile pricing, in: ACM SIGCOMM Conference on Internet Measurement, 2010, pp. 75–80.
[19] Y. Feng, B. Li, B. Li, Jetway: minimizing costs on inter-datacenter video traffic, in: ACM Multimedia Conference, 2012, pp. 259–268.
[20] T. Nandagopal, K. P. N. Puttaswamy, Lowering inter-datacenter bandwidth costs via bulk data scheduling, in: Symposium on Cluster, Cloud and Grid Computing, 2012, pp. 244–251.
[21] J. L. Garcia-Dorado, S. Rao, Cost-aware multi data-center bulk transfers in the cloud from a customer-side perspective, IEEE Transactions on Cloud Computing 7 (1) (2019) 34–47.
[22] X. Xie, Q. Ling, P. Lu, W. Xu, Z. Zhu, Evacuate before too late: Distributed backup in inter-DC networks with progressive disasters, IEEE Transactions on Parallel and Distributed Systems 29 (5) (2018) 1058–1074.
[23] Amazon WS news blog, https://aws.amazon.com/es/blogs/aws/aws-importexport-snowball-transfer-1-petabyte-per-week-using-amazon-owned-storage-appliances.
[24] Amazon WS news blog, https://aws.amazon.com/es/blogs/aws/aws-snowmobile-move-exabytes-of-data-to-the-cloud-in-weeks.
[25] E. Varki, Gpsonflow: Geographic positioning of storage for optimal nice flow, ACM Transactions on Modeling and Performance Evaluation of Computing Systems 3 (3) (2018) 12:1–12:29.
[26] J. H. Wang, J. Wang, C. An, Q. Zhang, A survey on resource scheduling for data transfers in inter-datacenter WANs, Computer Networks 161 (2019) 115–137.
[27] Y. Wang, S. Su, A. X. Liu, Z. Zhang, Multiple bulk data transfers scheduling among datacenters, Computer Networks 68 (2014) 123–137.
[28] Y. Feng, B. Li, B. Li, Postcard: Minimizing costs on inter-datacenter traffic with store-and-forward, in: Conference on Distributed Computing Systems Workshops, 2012, pp. 43–50.
[29] P. Lu, L. Zhang, X. Liu, J. Yao, Z. Zhu, Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks, IEEE Network 29 (5) (2015) 36–42.
[30] Y. Wu, Z. Zhang, C. Wu, C. Guo, Z. Li, F. C. Lau, Orchestrating bulk data transfers across geo-distributed datacenters, IEEE Transactions on Cloud Computing 5 (1) (2017) 112–125.
[31] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, R. Weber, Pushing CDN-ISP collaboration to the limit, ACM Computer Communication Review 43 (3) (2013) 34–44.
[32] H. Zhuang, I. Filali, R. Rahman, K. Aberer, Coshare: A cost-effective data sharing system for data center networks, in: IEEE Conference on Collaboration and Internet Computing, 2015, pp. 11–18.
[33] S. Floyd, V. Paxson, Difficulties in simulating the Internet, IEEE/ACM Transactions on Networking 9 (4) (2001) 392–403.
[34] F. Mata, P. Žuraniewski, M. Mandjes, M. Mellia, Anomaly detection in VoIP traffic with trends, in: International Teletraffic Congress, 2012, pp. 2:1–8.
[35] R. van de Meent, M. Mandjes, A. Pras, Smart dimensioning of IP network links, in: IFIP/IEEE Workshop on Distributed Systems: Operations and Management, 2007, pp. 86–97.
[36] Data Center map, Internet Exchange Points, http://www.datacentermap.com/ixps.html (2019).
[37] B. Huffaker, M. Fomenkov, D. Plummer, D. Moore, K. Claffy, Distance metrics in the Internet, in: IEEE International Telecommunications Symposium, 2002, pp. 200–202.
[38] Y. Chen, S. Jain, V. K. Adhikari, Z. L. Zhang, K. Xu, A first look at inter-data center traffic characteristics via Yahoo! datasets, in: IEEE INFOCOM, 2011, pp. 1620–1628.
[39] P. Velan, J. Medková, T. Jirsík, P. Čeleda, Network traffic characterisation using flow-based statistics, in: IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 907–912.
[40] M. Jain, C. Dovrolis, End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput, IEEE/ACM Transactions on Networking 11 (4) (2003) 537–549.
[41] S. K. Khangura, M. Fidler, B. Rosenhahn, Machine learning for measurement-based bandwidth estimation, Computer Communications 144 (2019) 18–30.
[42] C. Zhang, H. Zhang, J. Qiao, D. Yuan, M. Zhang, Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data, IEEE Journal on Selected Areas in Communications 37 (6) (2019) 1389–1401.
[43] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, P. Bertin, Improving traffic forecasting for 5G core network scalability: A machine learning approach, IEEE Network 32 (6) (2018) 42–49.
[44] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Interpretable machine learning: definitions, methods, and applications, arXiv preprint arXiv:1901.04592.
[45] U. Rathnayake, M. Iftikhar, M. Ott, A. Seneviratne, Realistic data transfer scheduling with uncertainty, Computer Communications 34 (9) (2011) 1055–1065.
[46] L. Wang, X. Wang, M. Tornatore, K. J. Kim, S. M. Kim, D. Kim, K. Han, B. Mukherjee, Scheduling with machine-learning-based flow detection for packet-switched optical data center networks, IEEE/OSA Journal of Optical Communications and Networking 10 (4) (2018) 365–375.
[47] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (5) (2019) 206–215.
[48] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, Y. Shin, Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?, Journal of Econometrics 54 (1-3) (1992) 159–178.
[49] J. Kilpi, I. Norros, Testing the Gaussian approximation of aggregate traffic, in: ACM SIGCOMM Workshop on Internet Measurement, 2002, pp. 49–61.
[50] R. Van De Meent, M. Mandjes, A. Pras, Gaussian traffic everywhere?, in: IEEE International Conference on Communications, 2006, pp. 573–578.
[51] S. Urmela, M. Nandhini, A framework for distributed data mining heterogeneous classifier, Computer Communications 147 (2019) 58–75.
[52] J. L. García-Dorado, J. Aracil, Flow-concurrence and bandwidth ratio on the Internet, Computer Communications 136 (2019) 43–52.
[53] O. J. Dunn, V. A. Clark, Applied statistics: analysis of variance and regression, Wiley, 1974.
[54] D. Muelas, J. E. López de Vergara, J. R. Berrendero, J. Ramos, J. Aracil, Facing network management challenges with functional data analysis: Techniques & opportunities, Mobile Networks and Applications 22 (2017) 1124–1136.

David MUELAS is currently a data scientist at BBVA Data & Analytics. Previously, he was researcher at Universidad Autónoma de Madrid (Spain), with interest in network traffic analysis, SDN and applied mathematics. He received the B.Sc. degrees in Mathematics and Computer Science (2013), M.Sc. degrees in Mathematics and Applications, and in Information and Communications Technologies (2015), and a Ph.D. in Computer Science and Telecommunication Engineering (2019), all of them from Universidad Autónoma de Madrid.

José Luis GARCÍA-DORADO received his M.Sc. and Ph.D. degrees both in Computer and Telecommunications Engineering from Universidad Autónoma de Madrid (UAM) in 2006 and 2010, respectively. He is a member of the High Performance Computing and Networking (HPCN) research group at UAM since 2005. From then, he was awarded a four-year predoctoral fellowship by the Ministry of Education of Spain (2007), and he was a Visiting Scholar with the Telecommunication Networks Group at Politecnico di Torino, Italy (2010), the Internet Systems Laboratory at Purdue University, USA (2013), and the Faculty of Applied Science at Universidad Técnica del Norte, Ecuador (2014 and 2015). Currently, he is an Associate Professor at UAM whose research interests are in the analysis of Internet traffic: its management, modeling, and evolution.

Sergio ALBANDEA received his B.Sc. in Telecommunication Technologies and Services (2015) and M.Sc. in Telecommunication Engineering (2017) degrees from Universidad Autónoma de Madrid, where he carried out his thesis within the scope of valley times in the Internet. Currently, he is a software engineer at Avaloq whose interests are in the monitoring and analysis of core banking networks on cloud architectures.

Jorge E. LÓPEZ DE VERGARA is an associate professor at Universidad Autónoma de Madrid (Spain) since 2007 and is a partner of Naudit HPCN, which is a spin-off company that was founded in 2009 and is devoted to high-performance traffic monitoring and analysis. He received his M.Sc. and Ph.D. degrees in Telecommunication Engineering from Universidad Politécnica de Madrid (Spain) in 1998 and 2003, respectively, where he also held a 4-year FPU-MEC research grant. During his Ph.D., he stayed for 6 months in 2000 at HP Labs in Bristol. He studies network and service management and monitoring, and has coauthored more than 100 scientific papers on this topic.

Javier ARACIL received his M.Sc. and Ph.D. degrees (Honors) from Technical University of Madrid in 1993 and 1995, both in telecommunications engineering, and his five-year degree in mathematics from UNED in 2009. In 1995 he was awarded a Fulbright scholarship to pursue post-doctoral research at the University of California, Berkeley. In 1998 he was a Research Scholar at the Center for Advanced Telecommunications, Systems and Services of the University of Texas at Dallas. He was an Associate Professor for the University of Cantabria and Public University of Navarra. Currently, he is a full professor at UAM and a founding partner of the spin-off company Naudit HPCN. His research interests are in optical networks and performance evaluation of communication networks. He has authored more than 100 papers in international conferences and journals.