Statistical network monitoring: Methodology and application to carrier-grade NAT

Statistical Network Monitoring: Methodology and Application to Carrier-Grade NAT

Enrico Bocchi[1,2]*, Ali Safari Khatouni[1], Stefano Traverso[1], Alessandro Finamore[3],
Maurizio Munafò[1], Marco Mellia[1], Dario Rossi[2]

1 - Politecnico di Torino, Italy
2 - Telecom ParisTech, France
3- Telefonica Research, Spain

Abstract

When considering to passively collect and then process network traffic traces, the need to analyze raw data at several Gbps and to extract higher level indexes from the stream of packets poses typical BigData-like challenges. In this paper, we engineer a methodology to extract, collect and process passive traffic traces. In particular, we design and implement analytics that, based on a filtering process and on the building of empirical distributions, enable the comparison between two generic collections, e.g., data gathered from two different vantage points, from different populations, or at different times. The ultimate goal is to highlight statistically significant differences that could be useful to flag to incidents for the network manager.

After introducing the methodology, we apply it to assess the impact of Carrier-Grade NAT (CGN), a technology that Internet Service Providers (ISPs) deploy to limit the usage of expensive public IP addresses. Since CGN may introduce connectivity issues and performance degradation, we process a large dataset of passive measurements collected from an ISP using CGN for part of its customers. We first extract detailed per-flow information by processing packets from live links. Then, we derive higher level statistics that are significant for the end-users, e.g., TCP connection setup time, HTTP response time, or BitTorrent average download throughput. At last, we contrast figures of customers being offered public or private addresses, and look for statistically significant differences. Results show that CGN does not impair quality of service in the analyzed ISP deployment. In addition, we use the collected data to derive useful figures for the proper dimensioning of the CGN and the configuration of its parameters in order to avoid impairments on end-users' experience.

Keywords: IP networks; Computer Network Management; Network Address Translation; Big Data; Network Measurements; Performance

1. Introduction and Motivation

Measurements have always played a central role to guide traffic management, to improve network and application design, and, in general, to understand the Internet. As a result, several tools are available for both active and passive measurements. The former let the network administrator run on-demand specific tests at the expense of an increased network load. The latter permit a continuous monitoring by simply observing traffic, a challenging task given the several Gbps currently carried by backbone links. Collected measurements can be gathered to form a BigData-like repository, and later leveraged to extract further knowledge, e.g., to contrast performance before and after an upgrade or to monitor performance of applications being accessed from different parts of the network. Often, the network administrator needs engineering means

*Corresponding author

Email addresses: enrico.bocchi@polito.it (Enrico Bocchi[1,2]), ali.safari@polito.it (Ali Safari Khatouni[1]), stefano.traverso@polito.it (Stefano Traverso[1]), alessandro.finamore@telefonica.com (Alessandro Finamore[3]), maurizio.munafo@polito.it (Maurizio Munafò[1]), mellia@tcl.polito.it (Marco Mellia[1]), dario.rossi@enst.fr (Dario Rossi[2]) "to compare" measurements collected at different times and places, or, generally speaking, different population subsets (e.g., fiber vs copper cable, fixed vs mobile, etc). In case of significant differences in results, additional actions can be taken to identify, and possibly fix, the root cause of these differences.

In this paper, we engineer a methodology that accomplishes the above process, and apply it to a specific use-case, namely the deployment of Carrier-Grade NAT (CGN). We leverage the passive monitoring technologies recently developed by the mPlane project [1], which offers a scalable architecture to deploy, collect and analyze Internet measurements. Referring to Fig. 1, we form a *measurement* layer by instrumenting several Points of Presence (PoPs) of an ISP with Tstat [2], a high-performing passive probe. By observing packets exchanged by end-users, each probe builds detailed logs for TCP and UDP flows in real time. Logs are then moved to a central *repository*, where several gigabytes of raw data are collected every hour. To obtain valuable information from the logs, we design practical *analytics* to extract the subsets of data of interest and compute high level performance indexes. In particular, we focus on engineering a methodology that allows us to detect whether statistically significant differences are present in measurements comparing different user metrics or periods of time. While several metrics [3] allow one to compare two empirical distributions, ingenuity is needed to engineer a robust system capable of conveying simple yet telling differences in a compact way.

While the methodology is generic and would allow the comparison of generic populations (e.g., IPv4 vs IPv6, HTTP/1.1 vs HTTP/2, Android vs iPhone, etc.), in this work we apply it to quantify the impact of CGN the ISP has deployed. Network Address Translation (NAT) techniques have become a viable cheap solution to alleviate public IPv4 exhaustion. In a nutshell, a router implementing NAT functionality remaps the IP address space of a private network into one (or more) public IP address(es). CGN technologies extend this concept by masking a whole ISP network using NAT [4]. In this scenario, customers' home routers are assigned private IP addresses. When communicating with hosts in the public Internet, the CGN router temporary maps the private, edge-facing IP address of the customer to one available public, Internet-facing IP address. This approach enables the ISP to mask part of its network as a large private network, significantly reducing the total amount of public IP addresses to use. Indeed, ISPs are more and more looking into these solutions as the price of a public IP address has now reached 10$/year per IP.[1]

However, NAT and CGN break the end-to-end paradigm of the Internet communication model. On the one hand, NAT-ed hosts cannot be directly addressed from the Internet, which is unsuitable for applications that require reachability from the public Internet. On the other hand, the NAT mapping operations may add delay to packets or cause loss. Despite a large body of work focusing on NAT technologies and NAT traversal techniques (which we overview in Sec. 7), little effort has been devoted to assess CGN impact on actual user experience.

We study both aspects in this work, whose main contributions can be summarized as follows:

- Building on well-known statistical methodologies, we engineer analytics to assess differences between measurement aggregates (Sec. 2).
- We define analytics to assess statistical differences between measurement aggregates, which allow one to either highlight spatial (e.g., different populations) or temporal (e.g., same population at different times) discrepancies (Sec. 2).
- We instantiate these analytics specifically to evaluate the impact of CGN deployment on populations of users with either private or public IP addresses. For the purpose, we define key performance indicators that are relevant for user quality of experience, such as connection setup time for web traffic, average transfer rate for BitTorrent traffic, etc. (Sec. 4).
- We then consider a real CGN from an actual ISP deployment, and process the data collected by monitoring about 17,000 residential customers for one month (Sec. 3). Results show that no statistically significant difference can be observed between the two populations for the considered performance indicators (Sec. 5).
- We leverage actual usage patterns to provide statistical figures that allow the network operator to properly design and dimension the CGN deployment while avoiding impairments to the end-users (Sec. 6).

[1] http://www.ipaddressnews.com/2014/04/07/343

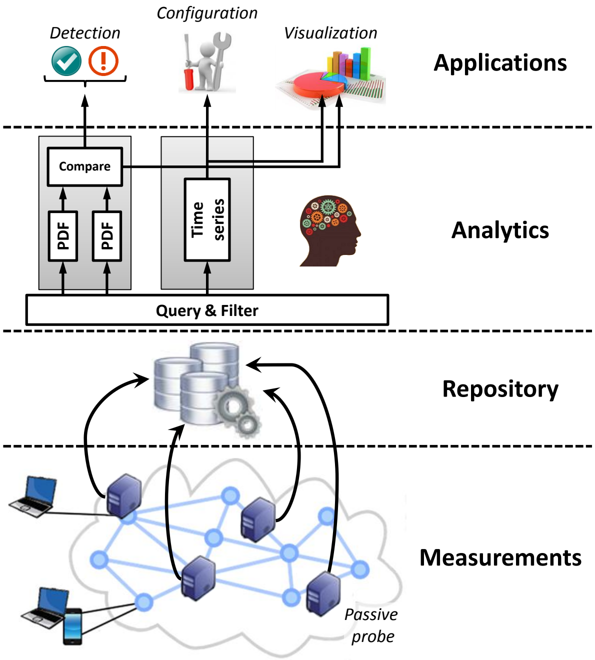


Figure 1: Illustration of the measurement framework.

2. Analytics to highlight and quantify statistical differences

In this work, we focus on analytics to compare measurements referring to different datasets, i.e., that help us to quickly pinpoint eventual performance differences between different populations of users. In this section, we provide an overall view and the necessary elements to understand the potential applications of the framework, deferring detailed statistical considerations and sensitivity analysis to Appendix A.

2.1. Measurement data collection and analysis workflow

Fig. 1 illustrates our workflow. Several layers are visible. From the bottom, the *Measurements* layer consists of passive Tstat probes installed in an operational network. Probes are responsible for extracting traffic summaries based on the continuous observation of packets generated by end-users: at flow completion, Tstat logs more than 100 metrics whose schema is strictly defined. As such, the output is a tabular database, where each row represents a flow and each column is a specific key performance indicator (KPI). More details about KPI definitions are given in Sec. 4. Data is temporarily stored at the probe premises, and then moved asynchronously to the central *Repository* based on Hadoop and Hive, which is located in the BigData Laboratory of Politecnico di Torino.

Once measurements are stored in the repository, a “Query & Filter” engine allows us to easily extract the measurement samples of interest (e.g., select Round Trip Time measurements for TCP connections where application layer protocol is HTTP, server name matches **.google.com*, client IP address is private, and date is any day of October 2014). Thanks to the SQL-like interface offered by Hive, the “Query & Filter” module allows us to easily access a large dataset in a simple, intuitive and scalable way, and quickly output sets of samples.

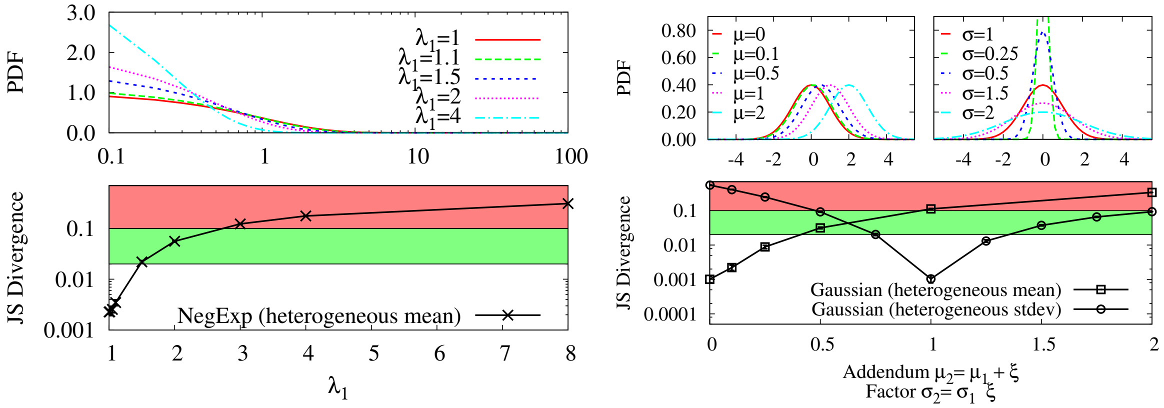
2.2. Empirical distribution estimation

We next estimate the Empirical Probability Density Function (EPDF) and Empirical Cumulative Distribution Function (ECDF) using a simple module that, given the size of bins and support range, computes the frequency of samples falling in each bin, i.e., the probability p_i that the sample takes values in the i-th bin. Given the amount of data to process is typically limited (few millions of samples), and the lack of iterative processing, we opted to implement this module using Python.

2.3. Comparison and quantization functions

While extracting data involves scalability issues successfully solved by the BigData approach, the comparison of EPDFs no longer needs BigData processing, but poses design and practical challenges the analyst has to cope with. For instance, one should ensure having an adequate population of samples, carefully choose the binning size, consider the nature of the performance indicators on which EPDFs are built, etc. We defer the formal discussion of such aspects to the Appendix A. By now, we limit the discussion to the description of a method to compare EPDFs.

To this goal, we build upon well-known statistical approaches to engineer a method to compactly quantify the difference between two distributions. We follow an engineering approach guided by intuitive choices to define a simple heuristic to let the analyst take informed decisions. Generally, the output of the comparison process can be a real value in a continuous range, or a categorical output from a (small) set of possible values (e.g., a boolean).



(a) Negative exponential

(b) Gaussian

Figure 2: Illustrative examples of Jensen-Shannon divergence computed on: (a) negative exponential distributions with heterogeneous mean rates λ_1 versus reference mean $\lambda = 1$; (b) Gaussian distributions with heterogeneous mean $N(\mu, 1)$ or standard deviation $N(0, \sigma)$ versus the reference distribution $N(0, 1)$.

In the first case, comparison tells the analyst how different the statistics are, while in the second case it just tells if they differ or not. We consider the second approach.

In formal terms, the comparison function has the form $F(p, q) : (\mathbb{R}, \mathbb{R}) \to \mathbb{R}$, while the quantization function can be defined as $Q(F(p, q)) : \mathbb{R} \to \mathbb{N}$, where $p = p(x)$ and $q = q(x)$ are two empirical distributions under analysis.

Without loss of generality, in this paper we define a simple quantization function that considers three possible levels, correlating with a no difference state (0), a definitively different state (2), and a possibly different state (1) requiring further investigation. Such quantization function can be written as:

$$Q(F(p,q)) = \begin{cases} 0 & \text{if} \quad F(p,q) < Q^{-} \text{negligible} \\ 1 & \text{if} \quad Q^{-} \leq F(p,q) < Q^{+} \text{noticeable} \\ 2 & \text{if} \quad F(p,q) \geq Q^{+} \text{relevant} \end{cases}$$

with states discriminated by the lower Q^{-} and upper Q^{+} thresholds. Intuitively, the two thresholds have practical relevance, and they relate to different levels of "warning": specifically, Q^{-} discriminates between practically *negligible* and *noticeable* differences, whereas Q^{+} discriminates between practically *noticeable* and *relevant* changes. This three-level quantization serves the purpose of prioritizing attention to the relevant events first, without however loosing track of noticeable events that are still worth investigating with lower priority: in the case of a single threshold $Q^{\star}$, the risk is that ISPs would either be overwhelmed with false positive signals (e.g., $Q^{\star} = Q^{-}$) or fail to notice possible interesting phenomena (e.g., $Q^{\star} = Q^{+}$).

Selecting these thresholds requires ingenuity since (i) there is a dependency between the thresholds Q^{-}, Q^{+} values and the comparison function $F(p, q)$; (ii) the value of $F(p, q)$ can be noisy when distributions p and q are computed over small population samples; (iii) the value of $F(p, q)$ can be affected by class imbalance when population samples of distributions p and q are of different orders of magnitude; and (iv) the value of $F(p, q)$ can be affected by the measurement process (e.g., binning strategy, number of bins, etc.).

While both Q^{-} and Q^{+} thresholds have only practical relevance (and as such can be configured by a human expert), we are implicitly assuming that the comparison function $F(p, q)$ is computed over statistically significant populations (so to ensure that the observed difference is also statistically relevant), and that no random fluctuations arise due to the above mentioned variables. To simplify the discussion, we consider a single statistical distance measure (SDM) for the sake of illustration, and defer to the Appendix A thorough discussion and sensitivity analysis on all related settings (e.g., different metrics, population size, binning, etc.).

2.4. Jensen-Shannon divergence

As representative SDM in this class, we take the Jensen-Shannon divergence (JS_{div}), which is defined as:

$$JS_{div} = \sum_i \Big\{ \frac{1}{2} p_i \ln\Big(\frac{p_i}{\frac{1}{2}p_i + \frac{1}{2}q_i} \Big) + \frac{1}{2} q_i \ln\Big(\frac{q_i}{\frac{1}{2}q_i + \frac{1}{2}p_i} \Big) \Big\}$$

where p_i and q_i are the empirical probabilities of samples taking values in the i-th bin. JS_{div} is a popular statistical measure based on the Kullback-Leibler divergence. JS_{div} adds symmetry, i.e., $JS_{div}(p, q) = JS_{div}(q, p)$, and bounded image, i.e., $JS_{div} \in [0, \ln(2)]$ to the Kullback-Leibler divergence. JS_{div} is equal to 0 if $p = q$, while it saturates to $\ln(2)$ for two completely disjoint distributions.

We focus on understanding how the JS_{div} varies when comparing two synthetic EPDFs, with the aim at defining Q^-, Q^+ thresholds to separate the areas into three states in a generic case. To this extent, we consider (i) negative exponential distributions with different mean; and (ii) Gaussian distributions with different mean and/or different standard deviation. These EPDFs are representative of diverse properties that may appear in network data; e.g., packet inter-arrival times and packet size in VoIP calls can be approximated by Gaussian distributions [5]; requests generated by user-activities are well approximated by Poisson processes and have as such negative exponential inter-arrival times [6, 7]. For the sake of the example, we now quantify differences between controlled EPDFs that are representative of synthetic, yet plausible, processes.

Such analysis is useful both to visually tie the JS_{div} behavior to some well-known distributions, and to identify quantization thresholds that discriminate among significant (2), noticeable (1) and negligible (0) differences between EPDFs. The purpose is to illustrate the methodology we followed in setting the quantization function $Q(F(p, q))$. A more in-depth analysis is given in Appendix A.

We consider a simple case where samples are extracted from p, q. We take care of avoiding any impact in the JS_{div} scores that can be tied to population size, imbalance or binning strategy (see Appendix A). Let us consider first the comparison of p, q which are both negative exponential distributions $NegExp(x, \lambda)$ of parameter λ.

We consider $p = NegExp(x, \lambda_0)$ as a reference, and choose λ_0=1, while $q = NegExp(x, \lambda_1)$ is instead shaped according to a distribution of parameter λ_1, with $\lambda_1 \in [1, 8]$ in our experiments. From both distributions, we extract 10^6 samples, obtain the empirical EPDFs using 1000 bins in a $[0, 100)$ support. This leads to bins of size $\Delta b = 100/1000 = 0.1$. For each bin i, we estimate p_i and q_i as the ratio between the number of samples falling in the i-th bin, i.e., $[i\Delta b, (i+1)\Delta b)$, and the total number of samples.

Negative exponential EPDFs p and q are depicted in the top portion of Fig. 2(a), whereas the bottom plot reports the JS_{div} versus λ_1. Without loss of generality, we select thresholds $Q^- = 2/100$ and $Q^+ = 1/10$, so that a clearly visible change in the distribution space (top) is visible in the JS_{div} space (bottom) as well. Intuitively, when $JS_{div} \in [Q^+, \ln(2)]$, the difference between the two EPDFs is significant (red area). When $JS_{div} \in [Q^-, Q^+)$ the difference is noticeable (green area), and negligible if $JS_{div} \in [0, Q^-)$ (white area).

We repeat the experiment this time considering Gaussian distributions $N(x, \mu, \sigma)$ of average μ and standard deviation σ. As before, we generate a reference sample p corresponding to $(\mu, \sigma) = (0, 1)$, and samples q with different (μ, σ) parameters. The upper-left plot of Fig. 2(b) shows EPDFs of q with parameters $\mu \in \{0, 0.1, 0.5, 1, 2\}$ and σ=1, while the upper-right plot shows EPDFs when μ=0 and $\sigma \in \{0.25, 0.5, 1, 1.5, 2\}$. Fig. 2(b) lower plot reports the JS_{div} values when comparing the abovementioned distributions against p.

The previous threshold selection proves to be effective also in the case of Gaussian distributions: visible differences in the upper plots of Fig. 2(b) appear to be separated by the $Q^- = 2/100$ and $Q^+ = 1/10$ thresholds.

In real cases, the domain knowledge can be used to set thresholds, and especially Q^+ as per our previous discussion. In general, any threshold choice results arbitrary, which applies to any SDM of choice (and possibly being even more complicated for those SDMs with infinite support). We point out that the framework we propose is not limited to the use of the JS_{div} measure. Rather, in Appendix A we consider a large set of SDMs and identify a set of those that are equivalent in this respect, as they share a number of desirable properties symmetry and boundedness being the most desirable ones. In our context, we specifically look for SDMs with bounded support as it makes the comparison of the difference between distributions more practical. More importantly, the symmetry property is required as it makes the SDM invariant to choice of the distribution considered as reference. While asymmetric metrics can be used to contrast a suspect population against a well-behaving one, we have no a priori knowledge on which population should be considered the reference. While we

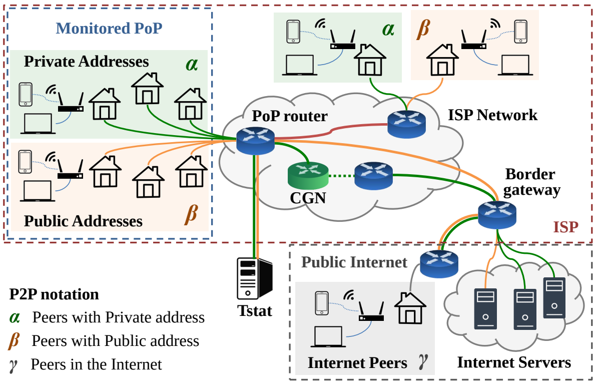


Figure 3: The monitoring scenario we consider in this study.

discuss these issues further in Appendix A, the information provided in this section allows us to understand the application of the general framework to the CGN use-case we focus on in the reminder of this paper.

3. Monitoring Scenario and Dataset

To characterize the implications of CGN, we rely on passive measurements obtained by instrumenting a monitoring probe in the operational network of an European country-wide ISP. Each customer device accesses the Internet via an ADSL home router. The ISP assigns either a public or private IP address to each home router according to the customer's subscription type. Traffic directed to the Internet and coming from home routers with a public IP address (public home routers) is routed directly to the final destination, while traffic from home routers with a private IP address (private home routers) has to cross the CGN device first.

The CGN used by the monitored ISP is based on the NAT444 standard [8], which relies on *sessions* to translate the private, edge-facing IP address of a home router into a public, Internet-facing one. When the CGN receives the first packet from a private home router, it starts a new session, temporarily mapping the private address to the first available public address in a pool. It then converts the address of all subsequent packets according to the same mapping.[2] After a given inactivity time during which no packets are observed, the session expires and the public address is put back in the pool of available addresses.

	Private	Public
# of TCP flows	990M	767M
# of UDP flows[4]	2,676M	1,941M
# of failed-TCP flows	301M	347M
Traffic Volume	168TB	105TB

Table 1: Statistics for home routers with private and public IP addresses.

3.1. Monitoring Setup

Fig. 3 depicts the monitoring scenario in which we operate. Three regions are identified: (i) a monitored PoP; (ii) the ISP network; and (iii) the public Internet. We deploy a passive probe inside the PoP and we instrument it to process the packets flowing through the PoP router. This router forwards the traffic coming both from private and public home routers, thus we are in the condition to monitor the traffic produced by the two populations of users.

In the case of web traffic, private and public home routers have to reach servers located in the public Internet. Therefore, the traffic they produce has to cross the PoP router, the CGN if the traffic is generated by private home routers, and the ISP border gateway. In the case of BitTorrent traffic, peers can be located both inside the ISP network (α, β in Fig. 3) and in the public Internet (γ). The ISP assures end-to-end connectivity among customers within its own network, independently on the type of IP address assigned to each home router.

3.2. Dataset description

We leverage a dataset collected during the month of October 2014. It consists of TCP, UDP and failed-TCP[3] logs carrying 1,757M, 4,617M and 648M records respectively, for a total of more than 273TB of network traffic. We split each of our logs in two subsets according to the IP address type of the customer's home router.

[2] The amount of public addresses available at the NAT is smaller than the number of customers provided with a private IP address. Consequently, the pool size of public addresses must be carefully set to minimize allocation costs, while guaranteeing satisfactory connectivity. See Sec. 6 for a thorough discussion.

[3] Tstat labels as failed TCP connections for which the Three-Way Handshake is not completed (e.g., when the sole SYN message is observed).

[4] In Tstat, a UDP flow starts at the first seen packet exchanged between two endpoints and ends 65 seconds after the last seen packet.

Tab. 1 provides statistics about the traces we consider, separately for private and public home routers. In total, we find more than 17,000 active home routers. Out of these 60% (40%) are assigned a private (public) IP address.[5]

Unless stated otherwise, in the reminder of the paper we present the results obtained by focusing on October 2014. Analysis conducted on other periods show very similar results. Overall, the dataset available to us is large enough to avoid random effects due to small population size or imbalance between classes.

3.3. Web traffic extraction

For this work we consider as web traffic all flows carrying HTTP transactions, excluding those containing BitTorrent metadata (details in the next section). Particularly, we isolate in our dataset *active* web users (i.e., IP addresses belonging to the monitored ISP) as those generating at least one flow carrying some piece of HTTP information and at least one flow carrying some HTTPS information. This is enough to filter out VoIP and traffic automatically generated by/to non-human entities such as smart TVs. These are traffic classes which are not of interest for this study.

3.4. BitTorrent traffic extraction

For this work we consider only BitTorrent traffic as it is the most used P2P application in our dataset. To isolate *active* BitTorrent peers in our dataset, we consider only those generating at least one flow which Tstat classifies as BitTorrent and transporting at least 1 MB of data.[6]

According to the type of IP address at the home router and to the location of the peer in the network (α, β, γ in Fig. 3), peers can experience different reachability conditions. To assess this, we check if peers are able to receive incoming connections from their counterparts. We define a peer as *reachable* if its home router is properly configured and ports are forwarded to the BitTorrent application. In case the home router is not properly configured and the BitTorrent application is non reachable, we define a peer as *unreachable*.

Four classes of peers emerge:

		Private	**Public**
TCP	Reachable	631 (35%)	496 (50%)
	Unreachable	1188 (65%)	499 (50%)
	Total	1819	995
UDP	Reachable	891 (77%)	591 (95%)
	Unreachable	262 (23%)	33 (5%)
	Total	1153	624

Table 2: Number of active BitTorrent peers classified according to their reachability condition.

- *Private–Unreachable*: any peer behind private home routers that does not receive incoming connections;
- *Private–Reachable*: any peer behind private home routers that receives incoming connections from other peers in the ISP network (α and β in Fig. 3). Reachability from peers in the Internet is not guaranteed as the CGN limits incoming connections;
- *Public–Unreachable*: any peer behind public home routers that does not receive incoming connections;
- *Public–Reachable*: any peer behind public home routers that receives incoming connections from both the ISP and the Internet. This is the only class of peers that is reachable by everyone.

Tab. 2 characterizes the number of active peers over TCP and UDP according to their reachability condition. Notice that only the 35% (50%) of peers with a private (public) IP have their home router properly configured and are thus reachable over TCP. This potentially is due to the scarce success of NAT traversal techniques for TCP flows.

In the case of UDP, instead, the ratio of reachable peers is higher both for private (77%) and public (95%) home routers. This is due to the fact that NAT traversal techniques like STUN [10, 11] are more effective over UDP, and enable peers to receive incoming connections.

4. Key performance indicators

In this section we define the performance indicators we are interested in when considering the CGN impact. At a high level, we consider both objective Quality of Service (QoS) metrics that are broadly available and independent from the applications, as

[5] The home router IP address can be considered as an identifier of the household. It may hide several devices connected to the Internet.

[6] Tstat employs both DPI techniques and behavioral classifiers to identify flows carrying BitTorrent information. For details, refer to [9].

well as objective metrics related to user Quality of Experience (QoE) and that are instead application specific. In what follows we describe the Layer-3 and Layer-4 QoS metrics, as well as the Layer-7 QoE metrics in more details.

Among the many measurements provided by Tstat, we consider for each traffic flow: (i) the TCP Round-Trip-Time (RTT) between the probe and server; (ii) the Time-To-Live (TTL) seen at the probe of packets sent by the server; (iii) the total per-flow amount of bytes sent and received by the client; (iv) the application layer protocol (e.g., HTTP, HTTPS, BitTorrent); and (v) the timestamps of packets that are instrumental to obtain further indices.[7] Finally, we use the Fully Qualified Domain Name (FQDN) [12] of the server to split traffic according to the service generating it.

Fig. 4 shows Tstat observing a HTTP transaction. In a nutshell, Tstat correlates TCP data segments and acknowledgments, and records timestamps of significant packets. For instance, by correlating times of a data segment with the corresponding acknowledgment, it computes a sample of the RTT. Average and standard deviation of RTT is then obtained by considering all samples in the TCP flow. We defer the interested reader to [2, 9] to obtain the detailed description of how performance indicators are extracted from packet traces.

Since CGN may impact both network, transport and application layer performance, we detail how we combine basic metrics provided by Tstat to build higher level measurements that we use to contrast the impact of CGN at different layers.

4.1. Network layer metrics

- *Hop count from Server to PoP (#Hops).*

The minimum number of hops being traversed by packets transmitted from the server to the client. The operating system of the server sets the initial value of the TTL, with power of 2 values being the typical choice.[8] Each router along the path then decreases the TTL. The value observed at the probe is thus an indication of the number of hops on the path from the server to the probe located in the PoP. In more detail, given a flow, we take the maximum server-to-client TTL observed by Tstat. We then choose x as the exponent minimizing $\#Hops = 2^x - TTL$, $\#Hops > 0$. The resulting $\widehat{\#Hops}$ is the minimum number of hops that packets in the considered flow have traversed before reaching the probe. In our scenario we expect packets received by private home routers to traverse a possibly larger number of hops due to the presence of the CGN (one or more hops).

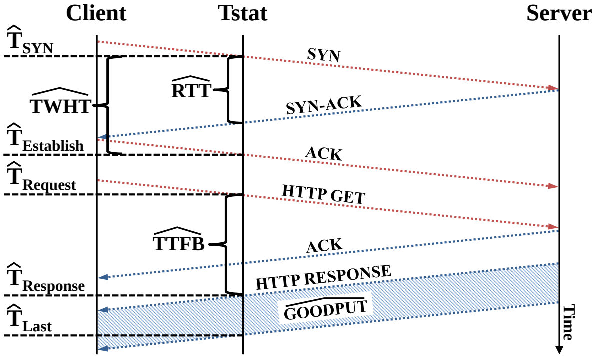


Figure 4: An example of HTTP transaction with metrics considered for our analysis.

- *PoP to Server Round Trip Time (RTT).*

The average RTT Tstat measures in a flow ($\widehat{RTT}$) on packets transmitted from the client to the server. Referring to Fig. 4, we consider only the RTT from the probe in the PoP to the server and backward, thus including only the backbone part of the path and ignoring the access portion. Estimating the RTT is complicated by the presence of mechanisms such as packet retransmission and delayed acknowledgments. The latter in particular can lead to bloated RTT estimations. We refer the reader to [9] for details. We expect packets transmitted by private home routers to experience a higher latency because of the CGN packet processing.

4.2. Transport layer metrics

- *TCP Three-Way Handshake Time (TWHT).*

The amount of time measured by Tstat ($\widehat{TWHT}$) required to successfully establish a TCP connection using the Three-Way Handshake (TWH). Referring to the upper part of Fig. 4, let $\hat{T}_{SYN}$ be the timestamp of the SYN packet sent by the client to start the connection establishment procedure, and let $\hat{T}_{Establish}$ be the timestamp of the packet carrying the ACK message ending the TWH. We define

[7] Notice that the probe measures the timestamps at a vantage point close to the customers. Therefore, for some metric X we can only gauge its estimated measure $\widehat{X}$.

[8] `http://subinsb.com/default-device-ttl-values`

the $\widehat{TWHT}$ as

$$\widehat{TWHT} = \hat{T}_{Establish} - \hat{T}_{SYN}$$

In our scenario we expect the $\widehat{TWHT}$ to be higher for private home routers due to the time needed by the CGN to allocate the resources for the new communication session.

For the sake of completeness, we also consider some advanced specific TCP metrics that are directly computed by Tstat [9]: (i) The number of SYN messages observed during connection setup, SYN; (ii) the number of out of sequence segments, OoS; (iii) the number of duplicated segments Dup. These are measurements that we expect to be altered in case of connectivity issues introduced by the CGN. A large value of SYN, for instance, indicates that the client experienced difficulties in establishing the connection due to, e.g., exhaustion of NAT resources.

4.3. Application layer metrics

Except for few traffic classes (such as e.g., VoIP which is quite well understood), measuring Quality of Experience is still a heavily debated subject[13]. In this paper, we measure QoE by proxy of some relatively simple metrics that are however tied to the application expectations: namely, we express user desire for interactive Web pages via latency measure (e.g., Time to the First Byte); similarly, we consider the throughput for bulk BitTorrent download, which is inversely proportional to the completion time, the main user satisfaction metric for BitTorrent.

- *Time to first byte (TTFB).*

Referring to Fig. 4, the amount of time that elapses between the first segment containing the HTTP request sent by the client ($\hat{T}_{Request}$) to the first segment with payload sent by the server ($\hat{T}_{Response}$). We define the $\widehat{TTFB}$ as

$$\widehat{TTFB} = \hat{T}_{Response} - \hat{T}_{Request}$$

In HTTP flows, it represents a measure of the time span between the application request issued by the client and the consequent response by the server. Also in this case, we expect the CGN to eventually delay the response time due to NAT operations.

- *Per-connection Goodput (G).*

The average rate at which the server delivers information to the client. This is the paramount performance index for download services. Let $\hat{T}_{Response}$ and $\hat{T}_{Last}$ (see Fig. 4) be the timestamps of the first and the last data packet sent by the server, and let D_{down} be the size of the application payload sent by the server. We define the average download goodput as

$$\hat{G}_{down} = \frac{D_{down}}{\hat{T}_{Last} - \hat{T}_{Response}}$$

It is similarly possible to evaluate the average goodput in the upload direction by considering the amount of bytes sent by the client to the server (D_{up}) and referring to the timestamps relative to the client traffic. To have a good estimation of the goodput, we evaluate $\hat{G}_{down}$ only on flows for which $D_{down} \geq 1$ MB, and $\hat{G}_{up}$ for flows where $D_{up} \geq 500$ kB, i.e., we avoid computing the goodput for short-lived flows.

- *Average Throughput (Thru).*

$\hat{G}_{down}$ is a representative measure of performance when the download of a content is done using a single flow, e.g., when downloading some software from the web. For P2P applications however, the speed at which a peer downloads a content is more complicated to compute since multiple parallel connections are used by the application. For instance, BitTorrent typically downloads content from 5 to 10 peers at the same time, using both TCP or UDP at the transport layer. To measure the overall performance of a peer, we compute the average download (upload) throughput $Thru$ considering all data received (sent) by a client in a time interval of duration $\Delta T = 10$ mins. Only flows classified as BitTorrent are considered. Formally, given time interval i, we consider all TCP and UDP flows that Tstat classifies as BitTorrent, and terminated in the time interval, $F(i) = \{f|\hat{T}_{Last}(f) \in i\Delta T\}$. Let $D_{tot}(i) = \sum_{k \in F(i)} D(k)$ the total amount of data those flows carried. Then

$$\widehat{Thru}(i) = \frac{D_{tot}(i)}{\Delta T}$$

5. Impact of CGN on users' traffic

The goal of this section is to check whether one of the two classes of customers experience worse performance than the other due to the type of IP address they have at their home router. To do so, we split flows into two subsets, based on if they are coming from private or public home routers. For each subset, we then compute the empirical EPDF for each metric, and we finally evaluate the

Metric	Web Traffic				BitTorrent	
	All Flows	www.google.com	TOP-50 Google	phobos.apple.com	Reachable	Unreachable
$\widehat{\#Hops}$	**0.223**	**0.666**	**0.682**	**0.689**	**0.184**	**0.162**
$\widehat{RTT}$	0.001	0.006	0.007	0.007	0.055	0.002
$\widehat{TWHT}$	0.002	0.010	0.011	0.016	0.029	0.008
$\#SYN$	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
OoS	<0.001	–	–	–	–	<0.001
Dup	0.001	0.001	0.001	<0.001	<0.001	<0.001
$\widehat{TTFB}$	0.002	0.006	0.008	0.006	0.031	0.005

Table 3: Jensen-Shannon divergence for considered metrics and different Internet services.

JS_{div} among the two EPDFs. For the alarm thresholds Q^- and Q^+ of the JS_{div}, we will consider the choices identified in Sec. 2.4: $Q^- = 0.02$, and $Q^+ = 0.1$. In particular, we will concentrate on all changes labeled as *practically significant*, i.e., exceeding Q^+.

5.1. Impact on network and TCP layer metrics

We start our analysis by gauging the impact of CGN on network- and transport- layer metrics described in Sec. 4.1 and Sec. 4.2, respectively. We report the collected results in Tab. 3. We focus on the Web traffic first, as reported on the left-hand side of the table. We show the result of experiments considering flows directed to (i) any remote server ("all flows"); (ii) "www.google.com" servers (i.e., *Google Search*); (iii) TOP-50 most used IP addresses of Google servers ("TOP-50 Google"); and (iv) "phobos.apple.com" servers providing *iTunes Store* contents.[9]

As shown, the only metric that consistently exceeds the alarm threshold Q^+ for both web traffic and BitTorrent is the number of hops, $\widehat{\#Hops}$, which is highlighted in bold in the table. To validate the above finding, we directly compare the distributions of $\widehat{\#Hops}$ in Fig. 5. For the ease of visualization, we report the CDF of some services, as results are similar for any service. A clear offset between the $\widehat{\#Hops}$ of private and public home routers appears, showing that private ones have to traverse more hops to reach the Internet. Such offset is present for all services. We verified this outcome with the ISP network administrators, who confirmed that the difference is due to some extra routers that packets sent/received by private home routers have to go through to reach the CGN. However, such routers are well dimensioned and not congested, with little to no implication on the performance, as testified by other metrics in Tab. 3.

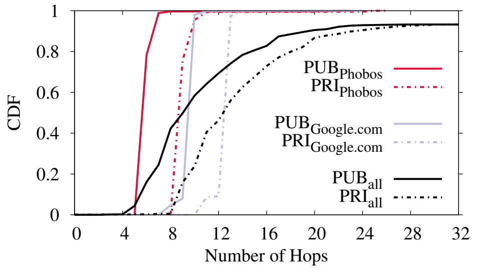


Figure 5: CDFs of the hop count ($\widehat{\#Hops}$) from the server to the client for private and public home routers against different web services. Clear differences are visible.

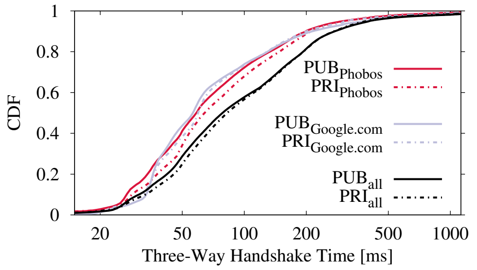


Figure 6: CDFs of time needed to complete the Three-Way Handshake ($\widehat{TWHT}$) for private and public home routers against different web services. No significant differences are visible.

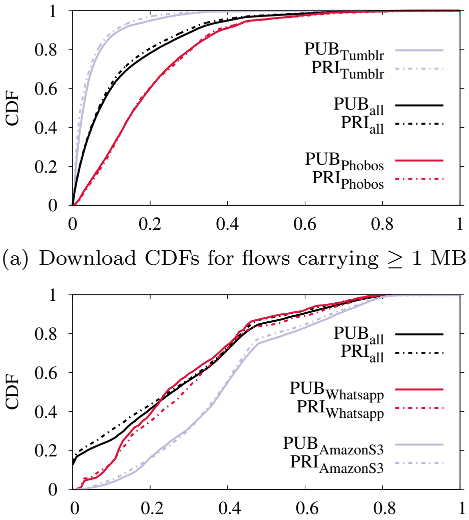
In summary, the JS_{div} values for web traffic are below Q^-, meaning that the CGN configuration of our scenario does not induce any significant impact on performance.

Let us focus on the time needed to establish a

[9] We focus on this selection of services as they appear to be popular in the monitored network, and the amount of TCP flows for each of them satisfies the requirements for a proper use of the JS_{div}.

	Service	FQDN	JS_{div}
Download	All	*	0.001
	Facebook Video	fbcdn-video-*.akamaihd.net	0.004
	Tumblr	media.tumblr.com	0.021
	Phobos	phobos.apple.com	0.022
Upload	All	*	0.004
	Amazon S3	eu-irl-*.s3.amazonaws.com	0.007
	Whatsapp	mm*.whatsapp.net	0.033
	Dropbox	dl-*.dropbox.com	0.046

Table 4: Jensen-Shannon divergence for average goodput distributions in download and upload directions.



(a) Download CDFs for flows carrying ≥ 1 MB.

(b) Upload CDFs for flows carrying ≥ 500 kB.

Figure 7: Normalized average goodput CDFs for flows carrying Web traffic.

TCP connection $\widehat{TWHT}$. This is a typical metric one would expect to be affected by additional delay introduced by the CGN when private home routers try to establish new connections. Indeed, the CGN may require some time to initiate the session and translate addresses. Also in this case JS_{div} is very small for Web traffic. Fig. 6 shows details distributions for private and public home routers with respect to the same Internet services. Differences are practically negligible.

5.2. Impact on application layer metrics

We complement the above findings by applying the JS_{div} on the indices presented in Sec. 4.3. The last row of Tab. 3 shows the JS_{div} of the Time to First Byte, $\widehat{TTFB}$. Results for web traffic indicate that this metric is again not affected by the presence of the CGN, and that users accessing the Internet from private or public home routers face similar delays.

	Reachable	Unreachable
Download	0.005	0.004
Upload	0.004	0.003

Table 5: Jensen-Shannon divergence for average throughput distributions in download and upload directions.

Next, we perform the same analysis for the web traffic average goodput $\hat{G}$. We consider several popular services that exchange a large amount of data, and for which $\hat{G}_{down}$ is thus relevant, i.e., Facebook Video, Tumblr and Phobos. For $\hat{G}_{up}$ we selected Amazon S3, Whatsapp and Dropbox. We report the results in Tab. 5.2, and draw the CDFs in Fig. 7.

Observe that the JS_{div} never overcomes the Q^{+} threshold, meaning that the CGN does not significantly harm the download/upload speed of private home routers. However, the JS_{div} values for Whatsapp and Dropbox in the upload direction, and for Tumblr and Phobos in the download direction, are higher than the Q^{-} threshold. Fig. 7(a) details the distribution of $\hat{G}_{down}$ (we omit Facebook Video to ease the visualization).[10] The curves referring to private and public home routers show indeed very similar trends, justifying small JS_{div} values, as confirmed by Tab. 5.2. Fig. 7(b) reports results for $\hat{G}_{up}$. Also in this case the curves show very similar CDFs with the only exception of Whatsapp. In this latter case, the difference between the two distributions is confirmed by the JS_{div} =0.033.

Interestingly, a relatively large amount of flows (13.98%) in Fig. 7(b) show almost zero throughput. By double-checking, we realize that those are long-lived flows with a duration higher than 10 min, and showing a number of uploaded bytes that slightly exceeds the 500 kB threshold. For some services, indeed, clients establish a single TCP connection with the remote server and keep sending tiny portions of data intermittently, de facto zeroing the upload throughput.

At last, we focus on the JS_{div} for BitTorrent traffic, distinguishing between (i) reachable peers (i.e., those who have port forwarding properly configured at their home router); and (ii) unreachable peers, as defined in Sec. 3.

[10] We normalize the measured throughput to not show the actual bandwidth provided by the monitored ISP.

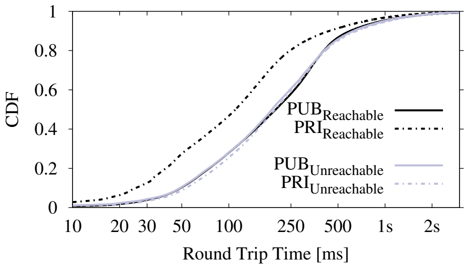


Figure 8: Round Trip Time for BitTorrent traffic according to peers' reachability conditions.

We consider the average download throughput. Several works in the literature (see Sec. 7 for a detailed discussion) show how multiple factors impact BitTorrent performance: content popularity, content availability, type of peers (e.g., seeders, leechers) involved in the transfer, peers cooperation techniques (e.g., tit-for-tat), etc.

Tab. 5 shows the JS_{div} values: for both classes of peers, the computed values are one-order of magnitude below the Q^- threshold, proving that the CGN does not affect throughput for BitTorrent. In a nutshell, peers obtain the same performance, no matter whether they have private or public addresses.

5.3. Discussion on BitTorrent implications

Let us get back to the right-hand side of Tab. 3 which shows the JS_{div} values for BitTorrent traffic. JS_{div} values are below Q^- for *unreachable* peers, but they fall in the noticeable range ($[Q^-, Q^+]$) for *reachable* peers. Let us consider first the results for $\widehat{RTT}$. Fig. 8 shows the $\widehat{RTT}$ of BitTorrent connections.[11] It is evident that the RTT for reachable peers with a private IP address appears to be lower than the RTT measured for all the other peers. To better understand this aspect, we characterize the reachability condition of peers inside the ISP, and of the counterparts with which they establish a connection.

To exemplify reachability conditions, refer to Fig. 3. It depicts peers inside the ISP (α, β) and peers outside (γ). Consider a *reachable* peer with private IP (α): it can receive incoming connections from all the peers inside the ISP network (β), but not from peers in the Internet (γ). On the other hand, a *reachable* peer with public IP (β) can receive incoming connections from both the peers inside the ISP (α) and from peers in the Internet (γ).

	Reachable	**Unreachable**
Private	58% ISP – 42% Internet	7% ISP – 93% Internet
Public	10% ISP – 90% Internet	6% ISP – 94% Internet

Table 6: Distribution of contacted peers

Such reachability conditions have implications also on the distribution of contacted peers, as shown by the Table 6 which reports the percentage of contacted peers by *reachable* and *unreachable* peers. Reachable peers with a private IP establish more connections with other peers inside the ISP (52%) than in the Internet (42%). All other classes of peers are more prone to connect to peers in the Internet (>90%). This is due to peers in the ISP that contacted private but reachable peers, like α.

As a consequence, private reachable peers experience a lower RTT since, they contact peers inside the ISP network, which are closer in space and exhibit a lower RTT. This is reflected in the other metrics, $\widehat{TWHT}$ and $\widehat{TTFB}$, apparently showing noticeable differences, those metrics being strictly related to the RTT (cfr. Fig. 4). This behavior is expected and is a direct consequence of the lower RTT experienced by private reachable peers. Despite this difference, no evident impact observed in download throughput, cfr. Table. 5.

6. Resource saving for different NAT policies

In this section we aim at providing some practical guidelines for the configuration of CGNs. In particular, we analyze different NATing policies and their saving in terms of public IP addresses to be used to offer connectivity to the ISP customers. We consider two different cases: (i) a simple NAT policy according to which a customer is given a public IP address for the period of time she is active. (ii) a NAT and Port Address Translation (PAT) policy for which a customer is given a block of ports on a given public IP address, for the time she is active. The *activity period* of a customer (i.e., of an IP address) starts at her first packet arrival at the probe and ends after T_{out}min after her last packet arrival at the probe. After this time the resource is returned to the pool of available IP addresses of the CGN.

[11] The measured $\widehat{RTT}$ is inflated by the queuing delay of packets stacked in the upload queue of home routers. All measurements are equally biased by this phenomenon and it does not harm the reliability of the metric.

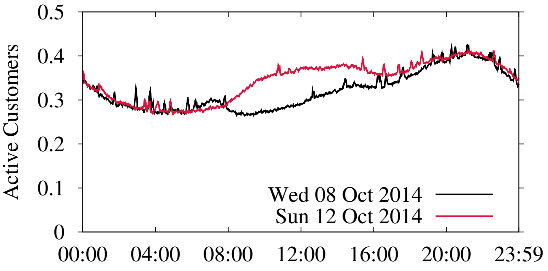


Figure 9: Fraction of active customers in different days.

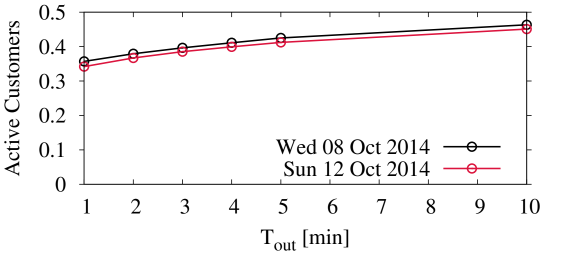


Figure 10: Maximum fraction of active customers observed in different days vs. NAT's T_{out}.

To conduct our analysis, we first must determine the number of active customers, and observe how their activity varies over the day. The analysis is conducted using the dataset described in Sec. 3.2, and thus refers to the population of the monitored PoP. However, the analysis can be easily extended to the entire ISP customer population. We focus on the traffic directed to destinations outside the ISP network, ignoring the traffic internal to the ISP, which is not subject to the CGN. We consider TCP and UDP traffic. In particular for TCP, we take into account both successfully completed and failed connections, as in both cases the CGN has to allocate a public IP address and/or a block of port. We suppose that any TCP and UDP connection requires a dedicated (IP address, port) pair on the NAT, and this association must be maintained for the whole connection lifetime. The (IP address, port) pair will be released, freeing the resource, only after T_{out} minutes have passed.

For the experiments in this section we pick a workday (Wednesday) and an off day (Sunday), so to consider different activity patterns.

6.1. NAT based on simple address mapping

We first emulate the resource usage in the simple NAT scenario. We expect that in the worst case, i.e., when all customers are active, the ISP would need as many public IP addresses as the overall customer population inside the network.

From the NAT perspective, varying T_{out} influences the number of active customers in the network. We suppose T_{out} is 5 min. Fig. 9 shows the evolution, over 1-day, of the active customers, for both the weekday and the off day. The result shows that a simple NAT policy would turn into considering active approximately 40% of customers that are active at the same time, and hence the ISP would save roughly up to 60% of the public IP addresses. Finally, observe that the user activity is rather regular and similar for different days of the week. We conduct the same experiment on different days, and we observe very similar results (omitted for briefness).

For completeness, we check the impact of T_{out} on the estimation of active customers. In fact, the larger T_{out}, the longer the customer appears as active to the NAT, and the longer the time the NAT has to wait before redeeming the public IP address. To this end, we show in Fig. 10 how the maximum fraction of active customers measured in the day (typically reached at the evening) changes when the T_{out} varies between 1 and 10 min.[12] As shown, the fraction of customers which have to be considered as active increases by 10% only when increasing T_{out} up to 10 min. Observe also that there is no substantial difference between different week days.

6.2. CGN based on PAT policy

While the above NATing technique might reduce the pool of public IP addresses to use, the actual savings are still limited, as the number of concurrent active customers is considerably large. Therefore, we investigate the resource requirements when NAT and PAT policy is in place, i.e., each active customer is given a block of ports on an public IP address.

For this policy, it is crucial to dimension the size of the block of ports the CGN shall allocate per customer. Hence, we have to count the per customer number of concurrent connections. We expect the CGN to assign continuous bulks of ports

[12] Notice that RFCs suggest to set T_{out} to 2 min for UDP [14] and 2 h for TCP [15]. However, the suggested thresholds have been shown to be too long, and they lead suboptimal retention policies [16]. For this reason, we explore a threshold space closer to the order of tens of minutes.

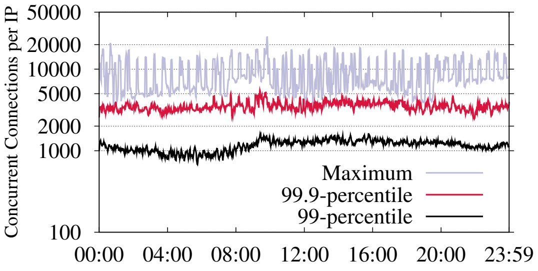


Figure 11: Maximum, 99.9- and 99.9-percentile of the number of per-customer concurrent active connections (computed as the maximum between the numbers of UDP and TCP connections).

to each customer. The sizing of the block of ports should be based on the transport protocol, i.e., TCP or UDP, employing the largest number of ports. For instance, let p_{TCP} and p_{UDP} be the number of concurrent active TCP and UDP connections, respectively. The block size must then be larger than $\max(p_{TCP}, p_{UDP})$. We proceed as follows. We choose $T_{out} = 5$ min, and we consider as concurrent the connections observed in 1 min long time bin. For each customer, we count the numbers of concurrent TCP and UDP connections. We then pick the maximum between the two, and use the result to build a per-minute distributions. In Fig. 11 we report the maximum, the 99.9- and the 99-percentiles obtained from the per-minute distributions and their evolution over time. As shown, the number of per-customer parallel connections rarely overcomes 20,000. In fact, we observe that customers employing a so wide number of ports are mostly users running P2P applications which open many parallel UDP flows. We see that 99% of customers never use more than 2,000 concurrent connections. Allocating a bulk of 2,000 ports for each customer would allow the ISP to use one public IP address for 32 customers. Considering a more conservative approach, i.e., adopting the 99.9-percentile as a reference, we observe roughly 6,000 concurrent connections, leading in this case to allocate about 6 customers per public IP address.

7. Related work

In this section we position our paper with respect to related studies. This section mostly discusses how our study complements the body of work present in the literature about measurement frameworks for anomaly detection and NAT technologies.

7.1. Methodologies for Anomaly Detection

Despite the goal of this study being determined the performance discrepancies between different user classes, our work is close to the body of studies focusing on anomaly detection, for which [17, 18] offer good surveys. In particular, our work resembles studies which target the problem of performing anomaly detection in large scale operational networks. For instance, [19, 20] are notable examples of methodologies which leverage data from passive probes, topology information, routing tables, etc. to match predictions to actual measurements to pinpoint deviations. The works that more than others resemble ours are [21] and [22]. Similar in spirit to our work,[21] designs a methodology for the comparison of empirical distributions obtained from traffic data, but, differently, to this end, the authors propose a new metric derived from the Kullback-Leibler divergence. The comparison with this and other metrics, together with the motivations which lead us to use the Jensen-Shannon divergence, are widely discussed in Appendix A. Instead, [22] designs a measurement framework for the comparison of empirical distributions built at different time bins. In this case, the authors leverage a complex statistical tool, the Generalized Likelihood Ratio Test, to highlight changes. Despite being similar in spirit, our approach is substantially more practical, and can be seen as complementary to the tool presented in [22].

7.2. NAT technologies

In the last years, ISPs have deployed CGN in their networks to limit the utilization of public IPv4 addresses and postpone their final exhaustion day [23, 24]. Given their strategic importance, CGNs have been matter of investigation in a large body of studies conducted by both standardization authorities and academia.

The IETF RFCs [14, 15, 25] standardize the requirements, implementations and behaviors for CGNs. A significant effort has been spent in standardizing mechanisms for NAT traversal, hole punching [26] and Interactive Connectivity Establishment (ICE) [27].

A remarkable amount of work has been dedicated to the task of identifying NAT deployment in residential networks [28, 29, 30, 31]. Similarly, but in a mobile scenario, [32] presents the results of an active measurement campaign to detect the presence of NAT middleboxes deployed in cellular networks.

Another branch of studies has focused on understanding the impact of CGNs on users' QoS and application-level experience [33, 34, 35, 36]. This paper falls in this category. Škoberne et al. [33] report a comprehensive classification of NATed scenarios and speculate about which impairments each of them could introduce. Ohara et al. [34] present a set of results obtained in a testbed. Specifically, they analyze the impact of network delays on the TCP connection establishment with and without CGN. In [35] the authors evaluate the hole punching technique for NAT traversal, and how this impacts the communication establishment in P2P applications. More similar in spirit to our work, [36] describes a case study conducted in controlled testbed where multiple CGN configurations are tested to evaluate their impact on several network applications and services. These include web, video streaming, P2P and gaming. These experiments are based on single sessions and do not consider actual performance testing. The results presented in [36] show that the presence of CGN has no substantial impact on users' browsing, thus confirming our observations. Differently, P2P applications like BitTorrent might be severely impaired. Our results, obtained in a real scenario and from a passive measurement complement these observations. In fact, even if we can not discriminate leechers from seeders as authors of [36] do, we show that NAT444 has a deep impact on the peer selection.

A last family of work focuses on CGN dimensioning aspects, such as port allocation and retention, which we assess in Sec. 6. In [16] the authors collect aggregate traffic traces from a real ISP network to investigate ports allocation and retention strategies in CGNs. The analysis shows that recommended timeout values in [15, 14] might be too long, resulting in suboptimal retention policies. In this paper we revisit the results presented in [16] by using more recent traces (2014 vs. 2009), collected from an actual residential ISP vs. campus network. Our results are different: despite more recent Internet applications have practically doubled the number of concurrent connections, we do not observe the need to decrease the expiry timeout on the NAT for UDP sessions in order to provide connection to the users.

Finally, to the best of our knowledge, this is the first work that specifically targets the problem of quantifying the impact of CGN on end-user experience from passive measurements. This paper extends our previous work, [37], in a number of respects. First, this work designs a systematic and general methodology for the comparison of KPIs from different populations. Second, in this work we use such methodology to evaluate the impact of CG-NAT on BitTorrent traffic too. Third, this extended version provides a thorough discussion (Sec. 6) about the possible resource saving which different CG-NAT policies would guarantee.

8. Conclusions

Network administrators lack effective tools to quickly pinpoint differences among several datasets obtained from traffic summaries. In this work, we aimed at filling this gap, and defined a methodology that builds on the statistical distance measures (such as Jensen-Shannon) divergence to assess statistical discrepancies between empirical distributions obtained from different populations or at different times. As an application example, we employed these analytics to study a large-scale CGN deployment, whereby ISP customers are split in two different populations, i.e., users assigned private IP addresses vs those assigned public IP addresses. For this particular scenario, we delineate several key performance indicators, relevant for users' quality of experience. In particular, we gauge the impact of CGN deployment on the web browsing experience and on BitTorrent traffic.

Our results show that the CGN technology is stable and mature. As with any study based on passive measurement, results in the following are specific to the deployment that is under observation. Conditioned to measurement in our dataset, results suggest that if properly engineered and configured, CGN does not harm users' web browsing experience. Albeit the presence of the CGN has an evident impact on the neighborhood construction of BitTorrent, it does not affect the average transfer rate of peers. We conclude that the ISP we consider in our study may have no actual need to provide users with public IP addresses, when not specifically required. Finally, we analyzed our network traces to quantify the actual saving CGN policies could guarantee. In case of simple NAT policy, the considered ISP could save about 50% of public IP addresses. When NAT and PAT are combined, the saving can exceed one order of magnitude.

Generalizing such results is however an entirely different matter: in the case of CGN, admittedly there exist many different configurations and deployments, and it would be dangerous to project the lessons learned on our dataset across heterogeneous deployments. Of course, gains in terms of IP addresses savings are estimated based on the usage pattern of the ISP under study and cannot be generalized. For what instead concerns the performance implication, we point out that our study is limited to Web and BitTorrent traffic, but does not consider VoIP or Gaming traffic that have more stringent delay and jitter requirements: should this traffic be especially important in another CGN deployment, the impact of CGN may be different in that case. Second, notice that our performance metrics are computed for successfully opened connections: as such, we are not measuring if the deployment of CGN is changing the failure rate of connection attempts, which could possibly harm the performance at a session level. Third, changes in user/applications patterns can also heavily affect the results: e.g., currently browsers open many connections in parallel, which is unfavorable for CGN scenario while HTTP/2 opens one per domain, so that these findings may need to be reassessed over long time periods with longitudinal studies.

In spite of these limitations, which are not peculiar to this work but that naturally arise whenever a specific dataset is considered, we believe that the methodology outlined in this work will surpass the lessons learned by its application to the particular question about CGN we address here – and rather possibly enable such longitudinal investigations.

Acknowledgements

This work has been funded by the *mPlane* project (grant agreement no. 318627) in the 7th European Framework Programme. We thank Valeria Di Gennaro for her initial help on this work, that she carried out in the context of her M.Sc. stage at LINCS `http://www.lincs.fr`.

References

[1] B. Trammell, P. Casas, D. Rossi, A. Bär, Z. Houidi, I. Leontiadis, T. Szemethy, and M. Mellia, “mPlane: an Intelligent Measurement Plane for the Internet,” *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 148–156, May 2014.

[2] A. Finamore, M. Mellia, M. Meo, M. Munafò, and D. Rossi, “Experiences of Internet Traffic Monitoring with Tstat,” *Network, IEEE*, vol. 25, pp. 8–14, 2011.

[3] A. L. Gibbs and F. E. Su, “On Choosing and Bounding Probability Metrics,” *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.

[4] S. Jiang, D. Guo, and B. Carpenter, “RFC 6264 - An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition,” Internet Engineering Task Force (IETF), Tech. Rep., 2011. [Online]. Available: {https://tools.ietf.org/html/rfc6264}

[5] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, “Revealing Skype Traffic: When Randomness Plays with You,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Aug. 2007. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282386

[6] R. Birke, M. Mellia, M. Petracca, and D. Rossi, “Understanding VoIP from backbone measurements,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE.* IEEE, 2007, pp. 2027–2035.

[7] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, “Detailed Analysis of Skype Traffic,” *Multimedia, IEEE Transactions on*, vol. 11, no. 1, pp. 117–127, Jan 2009.

[8] C. Donley, L. Howard, V. Kuarsingh, A. Chandrasekaran, and V. Ganti, “Assessing the Impact of NAT444 on Network Applications,” Internet Engineering Task Force (IETF), Tech. Rep., 2011. [Online]. Available: {https://tools.ietf.org/html/draft-donley-nat444-impacts-01}

[9] M. Mellia, M. Meo, L. Muscariello, and D. Rossi, “Passive Analysis of TCP Anomalies,” *Computer Networks*, vol. 52, no. 14, pp. 2663–2676, 2008.

[10] Z. Hu, “NAT Traversal Techniques and Peer-to-Peer Applications,” in *Peer-to-peer technologies, networks and systems Seminar on Internetworking*, 2005.

[11] J. Rosenberg, R. Mahy, P. Matthews and D. Wing, “RFC 5389 - Session Traversal Utilities for NAT (STUN),” Internet Engineering Task Force (IETF), Tech. Rep., 2008. [Online]. Available: {https://tools.ietf.org/html/rfc5389s}

[12] I. N. Bermudez, M. Mellia, M. M. Munafò, R. Keralapura, and A. Nucci, “DNS to the Rescue: Discerning Content and Services in a Tangled Web,” in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC ’12. New York, NY, USA: ACM, 2012, pp. 413–426. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398819

[13] P. Le Callet, S. Moller, and A. Perkis, “Qualinet White Paper on Definitions of Quality of Experience (2012),” European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland, Version 1.20, Tech. Rep., March 2013.

[14] F. Audet, and C. Jennings, “RFC 4787 - Network Address Translation (NAT) Behavioral Requirements for Unicast UDP,” Internet Engineering Task Force

(IETF), Tech. Rep., 2007. [Online]. Available: {https://tools.ietf.org/html/rfc4787}

[15] S. Guha, K. Biswas, B. Ford, S. Sivakumar, and P. Srisuresh, "RFC 5382 - NAT Behavioral Requirements for TCP," Internet Engineering Task Force (IETF), Tech. Rep., 2008. [Online]. Available: {https://tools.ietf.org/html/rfc5382}

[16] S. Alcock, R. Nelson, and M. David, "Investigating the Impact of Service Provider NAT on Residential Broadband Users," University of Waikato, Tech. Rep., 2010. [Online]. Available: {http://www.wand.net.nz/~salcock/spnat/tech_report.pdf}

[17] A. Patcha and J.-M. Park, "An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2007.02.001

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," in *Computing Surveys.* ACM, 2009, vol. 41, pp. 1 – 58.

[19] H. Yan, A. Flavel, Z. Ge, A. Gerber, D. Massey, C. Papadopoulos, H. Shah, and J. Yates, "Argus: End-to-end Service Anomaly Detection and Localization from an ISP's Point of View," in *IEEE INFOCOM*, 2012.

[20] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using Traffic Feature Distributions," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 217–228, Aug. 2005. [Online]. Available: http://doi.acm.org/10.1145/1090191.1080118

[21] A. D'Alconzo, A. Coluccia, F. Ricciato, and P. Romirer-Maierhofer, "A Distribution-Based Approach to Anomaly Detection and Application to 3G Mobile Traffic," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Nov 2009, pp. 1–8.

[22] A. Coluccia, A. D'Alconzo, and F. Ricciato, "Distribution-based Anomaly Detection via Generalized Likelihood Ratio Test," *Comput. Netw.*, vol. 57, no. 17, pp. 3446–3462, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.07.028

[23] A. Muller, F. Wohlfart, and G. Carle, "Analysis and Topology-based Traversal of Cascaded Large Scale NATs," in *Proceedings of the 2013 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ser. HotMiddlebox '13. New York, NY, USA: ACM, 2013, pp. 43–48. [Online]. Available: http://doi.acm.org/10.1145/2535828.2535833

[24] P. Richter, M. Allman, R. Bush, and V. Paxson, "A Primer on IPv4 Scarcity," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 2, pp. 21–31, Apr. 2015. [Online]. Available: http://doi.acm.org/10.1145/2766330.2766335

[25] S. Perreault, I. Yamagata, S. Miyakawa, A. Nakagawa, and H. Ashida, "RFC 6888 - Common Requirements for Carrier-Grade NATs (CGNs)," Internet Engineering Task Force (IETF), Tech. Rep., 2013. [Online]. Available: {https://tools.ietf.org/html/rfc6888}

[26] P. Srisuresh, B. Ford, and D. Kegel, "RFC 5128 - State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)," Internet Engineering Task Force (IETF), Tech. Rep., 2008. [Online]. Available: {https://tools.ietf.org/html/rfc5128}

[27] J. Rosenberg, A. Keranen, B. B. Lowekamp, and A. B. Roach, "RFC 6544 - TCP Candidates with Interactive Connectivity Establishment (ICE)," Internet Engineering Task Force (IETF), Tech. Rep., 2012. [Online]. Available: {https://tools.ietf.org/html/rfc6544}

[28] L. DiCioccio, R. Teixeira, M. May, and C. Kreibich, "Probe and Pray: Using UPnP for Home Network Measurements," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, N. Taft and F. Ricciato, Eds. Springer Berlin Heidelberg, 2012, pp. 96–105. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28537-0_10

[29] S. M. Bellovin, "A Technique for Counting Natted Hosts," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, ser. IMW '02. New York, NY, USA: ACM, 2002, pp. 267–272. [Online]. Available: http://doi.acm.org/10.1145/637201.637243

[30] G. Maier, F. Schneider, and A. Feldmann, "NAT Usage in Residential Broadband Networks," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, N. Spring and G. Riley, Eds. Springer Berlin Heidelberg, 2011, vol. 6579, pp. 32–41. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19260-9_4

[31] V. Krmicek, J. Vykopal, and R. Krejci, "Netflow Based System for NAT Detection," in *Proceedings of the 5th International Student Workshop on Emerging Networking Experiments and Technologies*, ser. Co-Next Student Workshop '09. New York, NY, USA: ACM, 2009, pp. 23–24. [Online]. Available: http://doi.acm.org/10.1145/1658997.1659010

[32] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang, "An Untold Story of Middleboxes in Cellular Networks," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 374–385. [Online]. Available: http://doi.acm.org/10.1145/2018436.2018479

[33] N. Škoberne, O. Maennel, I. Phillips, R. Bush, J. Zorz, and M. Ciglaric, "IPv4 Address Sharing Mechanism Classification and Tradeoff Analysis," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 391–404, Apr. 2014. [Online]. Available: http://dx.doi.org/10.1109/TNET.2013.2256147

[34] Y. Ohara, K. Nishizuka, K. Chinen, K. Akashi, M. Kohrin, E. Muramoto, and S. Miyakawa, "On the Impact of Mobile Network Delays on Connection Establishment Performance of a Carrier Grade NAT Device," in *Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference*, ser. AINTEC '14. New York, NY, USA: ACM, 2014, pp. 1:1–1:8. [Online]. Available: {http://doi.acm.org/10.1145/2684793.2684794}

[35] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-Peer Communication Across Network Address Translators," in *USENIX Annual Technical Conference*, 2005.

[36] C. Donley, L. Howard, V. Kuarsingh, J. Berg, and J. Doshi, "RFC 7021 - Assessing the Impact of Carrier-Grade NAT on Network Applications," Internet Engineering Task Force (IETF), Tech. Rep., 2013. [Online]. Available: {https://tools.ietf.org/html/rfc7021}

[37] E. Bocchi, A. Khatouni, S. Traverso, A. Finamore, V. Di Gennaro, M. Mellia, M. Munafò, and D. Rossi, "Impact of Carrier-Grade NAT on Web Browsing," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, Aug 2015, pp. 532–537.

Name	Abbrv	Formula	Image	Properties		
				Metric	Bounded	Symmetric
Jensen-Shannon	JS	$JS_{div}(p,q) = \sum_i \left\{ \frac{1}{2} p_i \ln\left(\frac{p_i}{\frac{1}{2}p_i + \frac{1}{2}q_i} \right) + \frac{1}{2} q_i \ln\left(\frac{q_i}{\frac{1}{2}q_i + \frac{1}{2}p_i} \right) \right\}$	$[0, ln(2)]$		✓	✓
Kullback-Leibler	KL	$KL_{div}(p,q) = \sum_i p_i log(\frac{p_i}{q_i})$	$[0, \infty)$			
Chi Square	χ^2	$\chi^2_{dis}(p,q) = \sum_i \frac{(p_i - q_i)^2}{q_i}$	$[0, \infty)$			
Separation	S	$S_{dis}(p,q) = max_i \left(1 - \frac{p_i}{q_i} \right)$	$[0, 1]$		✓	
Total variation	TV	$TV_{dis}(p,q) = \frac{1}{2} \sum_i \lvert p_i - q_i \rvert$	$[0, 1]$		✓	✓
Hellinger	H	$H_{dis}(p,q) = \left[\sum_i (\sqrt{p}_i - \sqrt{q}_i)^2 \right]^{\frac{1}{2}}$	$[0, \sqrt{2}]$		✓	✓
Kolmogorov	K	$K_{met}(P,Q) = \sup_x \lvert P(x) - Q(x) \rvert$	$[0, 1]$	✓	✓	
Wasserstein	W	$W_{met}(P,Q) = \int_{-\infty}^{\infty} \lvert P(x) - Q(x) \rvert dx$	$[0, 1]$	✓	✓	
Discrepancy	D	$D_{met}(P,Q) = \sup_{\text{all closed balls B}} \lvert p(B) - q(B) \rvert$	$[0, diam\Omega]$	✓	✓	
DCF09 [21]	L	$L_{div}(p,q) = \frac{1}{2} \left(\frac{KL_{div}(p,q)}{E_p} + \frac{KL_{div}(q,p)}{E_q} \right)$	$[0, \infty)$			✓

Table A.7: Statistical Distance Measures. In the above formulas, p and q denote two empirical distributions on the measurable space Ω, with p_i and q_i being their samples, and P and Q their cumulative distribution functions. Note that in L, E_x is the entropy of empirical distribution x (we preferred to use E instead of the common H notation to avoid conflicts with H – Hellinger).

Appendix A. Statistical Distance Measures

In this work we selected a specific Statistical Distance Measure (SDM) that we used as $F(p,q)$, namely the Jensen-Shannon divergence (JS_{div}). The purpose of this section is to (i) show the broad set of SDMs from a theoretical viewpoint, indicating the criteria used to narrow down SDMs selection; and (ii) assess robustness of the $F(p,q)$ estimation as function of the p,q population size and binning strategy employed.

Appendix A.1. SDM Comparison

In this work we do not aim at proposing a novel SDM. We instead prefer to collect a set of well-known and established SDM available in literature, analyze their features and choose the most suitable one for our use-case. Gibbs et al. [3] compare a variety of SDMs, shedding light on their properties and on the relationships among them. Without aiming at completeness, we report in Tab. A.7 a list of 9 representative SDMs considered in [3], plus the SDM proposed in [21]. Specifically, for each SDM the table reports its name, abbreviated notation, definition, co-domain and three relevant properties: (i) *Metric*, the SDM is a function defining a metric distance between each pair of elements in a set; (ii) *Bounded*, the SDM co-domain is finite; and (iii) *Symmetric*, the SDM is invariant to which of the two distributions is considered the reference, i.e., $F(p,q) = F(q,p)$.

From Tab. A.7 it is easy to see a rather heterogeneous picture. Most SDMs are divergence measures, with the exclusion of Kolmogorov (K), Wasserstein (W) and Discrepancy (D), which are metrics. With the exception of Kullback-Leibler (KL) and Chi-Square (χ^2), all other SDMs have a bounded co-domain. Finally, only Jensen-Shannon (JS), Total Variation (TV) and Hellinger (H) are symmetric. At last, we explicitly consider the metric proposed in [21] which is symmetric, but not bounded and not a metric. None of the SDMs exhibits all three properties. As we shall see later, these properties play an important role in the SDM selection.

In terms of provenance and use, JS and KL are information theoretic measures. Loosely speaking, KL expresses the amount of information that is required to encode q knowing p, while JS expresses the average amount of information carried by q which is not in p. χ^2, H and K are often used for statistical tests. [21] has been proposed to specifically tackle anomaly detection in network measurements context.

In principle, any of the SDMs in Tab. A.7 can fit the purpose of our framework, so we illustrate here some relevant criteria to narrow down the SDM selection to a small set of equivalent functions.

In general terms, we'd like the measures to be symmetrical (because our framework does not have a reference, well behaving, population), and bounded (to be able to practical define the thresholds Q^- and Q^+). This means that Jensen-Shannon (JS), Total Variation (TV) and Hellinger (H) are all good candidates and are equivalent to our purpose. For practical purposes, we restrict our attention to JS_{div} as reference $F(p,q)$ measure.

Appendix A.2. JS_{div} Sensitivity analysis

We now assess the SDM robustness to factors that may affect the EPDF estimation, as these may induce artificial errors leading to wrong conclusions. Indeed, the whole framework rely on the ability to compute a statistically relevant distance measure $F(p,q)$ between two population samples (represented by their EPDFs p and q). This distance measure $F(p,q)$ is then compared to two empirical thresholds Q^- and Q^+ to discriminate between cases having a practically negligible, practically noticeable or practically relevant significance. Of course, this *practical significance* holds only provided that $F(p,q)$ is also *statistically significant*, as otherwise differences between the population samples that p and q may be actually artifacts tied to a number of random fluctuations. Otherwise stated, the relevance of the framework is conditioned to the statistical significance of the computed metrics, as otherwise it would be possible to raise alarms that are however not statistically significant.

To avoid the above problem, we need not consider the potential source of errors that can indeed affect SDMs, of which the most prominent are: (i) the binning strategy used to compute the samples of p and q distributions; (ii) the imbalance in the population size of p and q; and (iii) the finitude of p and q populations.

Appendix A.2.1. Binning strategy

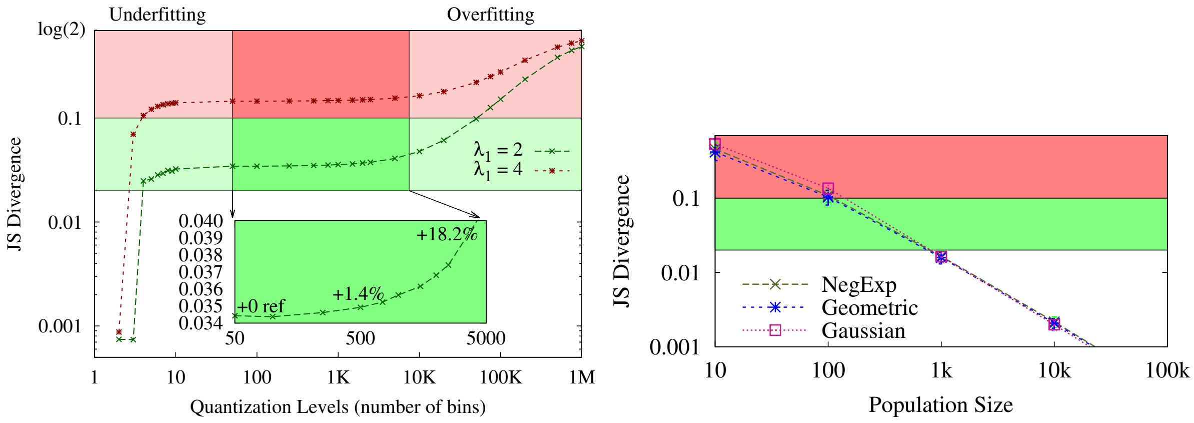
Let us first start from the impact of the binning strategy. Taking JS_{div} as an example, we assess the operating conditions of the framework that ensure proper evaluation of the EPDFs. We expect the binning adopted in estimating the EPDF to play a role for continuous metrics with domain in $\mathbb{R}$: intuitively, coarse bins smooth down differences (JS_{div} decreases, approaching 0 in the limit case where all samples fall in the same single bin). Fine grained bins, in contrary, exacerbate differences (JS_{div} increases and approaches $\ln(2)$ for rational bins of vanishing size, each of which contains a single or few samples).

It is thus important to assess the settings of the uniform binning strategy, i.e., the support and bin size (or equivalently, number of bins). As done previously, we follow an engineering and experimental approach. We consider p and q as negative exponential distributions, with $\lambda_0 = 1$, $\lambda_1 \in \{2,4\}$. Given the previous Q^-, Q^+ thresholds, we expect $q = NegExp(x,2)$ to fall in the intermediate state, while $q = NegExp(x,4)$ to be significantly different from p. To avoid small population noise, we use finite sequences of 10^6 samples for each distributions. We then extract the empirical distributions from the two dataset by considering a number of bins which varies from 2 to 10^6. We limit the support in the $[0,100)$, thus $\Delta b \in [0.001, 50]$. We then compute the JS_{div} to compare p and q. For each value of the bin, we repeat 100 runs.

Fig. A.12(a) show results, where the x-axis reports the number of bins, and the y-axis the corresponding JS_{div} value. Note the logarithmic scales. When the number of bins is smaller than 50, a underfitting phenomenon emerges, so that the JS_{div} artificially drops to small values. Similarly, when the number of bins grows larger than 5,000, an overfitting phenomenon is visible, so that the JS_{div} artificially increases. We see that the JS_{div} is consistent for number of bins in the 50-5,000 range, where the EPDFs are correctly estimated. The inset details the relative error that occurs to JS_{div} with respect to the value obtained when using 50 bins, i.e., the reference. The relative error is below 19%. It follows that quantization oddities are controllable, provided a large number of samples is available, and that the support of the distribution is limited.

In general, it is good practice to select a binning strategy that is tied to the physics of the metric: for example, use an unitary bin size for measurements that takes integer values (e.g., the Number of Hops, of SYN messages, etc.), or relate the bin size to the unit of scale of interest (e.g., a 1 ms accuracy for RTT and time-related metrics, or consider bins of 10 kbps when dealing with throughput). This calls for ingenuity and suggest the involvement of domain expertise.

In presence of heavy-tailed distributions, the choice of logarithmic binning strategies, or of mixed linear-logarithmic ones as suggested in [21], could be considered. By using logarithmic binning one would alleviate the problem of vanishing bins with



(a) Quantization levels

(b) Population size

Figure A.12: Sensitivity analysis of Jensen-Shannon divergence for: (a) varying number of bins, (b) varying population size for two finite realization of the same process.

few samples (which typically would occur in the tail of the distribution), and limit the number of bins. However, this comes at a cost. Notice indeed that engineering questions would arise: How many bins should be used, and how to properly set the switching threshold from linear to logarithmic binning? All these choices have, in our opinion, to be driven by domain knowledge and should be tailored to the application domain.

Appendix A.2.2. Population size

Clearly, a specular question is in place: for a given bin size choice, what is the impact of the number of samples on the estimation of the EPDF? Intuitively, while any finite sequence deviates from quantiles of the theoretic distribution, small population samples tend to exhibit larger deviations.

Taking two finite realizations of the same process, we estimate the empirical EPDFs p and q and compute the JS_{div}. To avoid binning errors, we consider real-valued distributions (i.e., Gaussian, negative exponential) and an integer-valued distribution (Geometric). We then estimate the two (nominally identical) EPDFs using a number of samples that varies from 10 to 10^5 samples. We compute the JS_{div} (which we expect to be close to 0) considering 1,000 bins.

Fig. A.12(b) shows results. Irrespectively of the distribution, JS_{div} is strongly affected by the population size (linear slope in log-log plot). As expected, an excessively small population inflates the JS_{div} value. Specifically, having less than 1,000 (100) samples in the population causes the JS_{div} to exceed the warning threshold for noticeable (significant) differences for all the distributions. It is thus recommended to employ the JS_{div} on population larger than 1,000 samples, assumption verified in our dataset.

However, it is important to mention that artifacts caused by a limited population size may have an impact in case the methodology is used in real-time (e.g., on short time window) scenarios, or to compare the same population over different temporal samples. This possibly mandates a minimum duration of the observation period, especially in off-peak times, so to reach a minimum level of observation samples.

Finally, population imbalance is worth discussing, as it may introduce yet another bias. Yet, we experimentally observe that, as long as the smallest population is statistically significant, then no noticeable bias appears – which is intuitive since EPDFs renormalize the contribution of each population.

Authors

Enrico Bocchi received his M.Sc. Degree in Telecommunications Engineering from Politecnico di Torino, Italy, in 2013. In 2014, he joined the Telecommunication Networks Group of Politecnico di Torino as Ph.D. Candidate. His research interests cover multiple aspects of Internet traffic monitoring, including cloud storage services measurements and benchmarking, network traffic characterization and classification, and applications to security. He is currently with the Laboratory of Information, Networking and Communication Sciences (LINCS), Paris, France, in the frame of a Joint-PhD degree between Politecnico di Torino and Telecom ParisTech, working on performance assessment of novel Internet protocols.

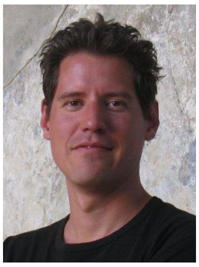
Ali Safari Khatouni received his M.Sc. Degree in Computer and Communication Networks Engineering from Politecnico di Torino, Italy, in 2014. He joined the Telecommunication Networks Group of Politecnico di Torino as Ph.D. Candidate in 2015 and he is currently a participant of the mPlane Integrated Project. His research interests include Internet traffic monitoring, characterization and analysis.

Stefano Traverso Ph.D. His research interests include privacy-preserving systems, network measurements and content delivery networks. During his Ph.D. and Post-doc he has been visiting Telefonica I+D research center (Barcelona, Spain), NEC Laboratories (Heidelberg, Germany) and Alcatel-lucent Bell Labs (Paris, France). He is currently a Post-doc Fellow of the Telecommunication Networks Group group of Politecnico di Torino.

Alessandro Finamore received his Ph.D. in Electronics and Communication Engineering (2012), and M.Sc. (2008) from Politecnico di Torino. He has been an intern at University of Purdue, Lafayette, IL-USA in 2010, Telefonica Research, Barcelona, Spain in 2012, and Narus Inc., Sunnyvale, CA-USA in 2014. He coauthored of more than 30 publications, and participated in the TPC of venues such as Infocom, PAM and TMA. His research interests are in the area of Internet traffic analysis, mobile systems, user quality of experience and mobility, CDNs services, and BigData frameworks. He is currently associate researcher at Telefonica Research in Barcelona.

Maurizio Munafò is Assistant Professor at the Department of Electronics and Telecommunications of Politecnico di Torino. He holds a Dr.Ing. degree in Electronic Engineering since 1991 and a Ph.D. in Telecommunications Engineering since 1994, both from Politecnico di Torino. He has coauthored about 70 journal and conference papers in the area of communication networks and systems. His current research interests are in simulation and performance analysis of communication systems and traffic modeling, measurement, and classification.

Marco Mellia graduated from the Politecnico di Torino with Ph.D. in Electronic and Telecommunication Engineering in 2001, where he holds a position as Associate Professor. He has co-authored over 250 papers published in international journals and presented in leading conferences. He participated to the program committees of several conferences including ACM SIGCOMM, ACM CoNEXT, ACM IMC, IEEE Infocom, IEEE Globecom and IEEE ICC. He is Area Editor of ACM CCR, and ACM/IEEE Transactions on Networking. He is the coordinator of the mPlane Integrated Project, which focuses on building an Intelligent Measurement Plane for Future Network and Application Management.



Dario Rossi is a Professor at Telecom ParisTech (Paris, France) and Ecole Polytechnique (Palaiseau, France). He received his M.Sc. and Ph.D. Degrees from Politecnico di Torino in 2001 and 2005 respectively, and his HDR Degree from Universit'e Pierre et Marie Curie (UPMC) in 2010. During 2003-2004, he held a visiting researcher position in the Computer Science division at University of California, Berkeley. He has coauthored over 9 patents and over 150 papers in leading conferences and journals, that received 3 best paper awards, a Google Faculty Research Award (2015), and an IETF Applied Network Research Prize (2016). He participated in the program committees of over 50 conferences including ACM ICN, ACM CoNEXT, IEEE Infocom of which he was also Distinguished Member (2015, 2016). His current research interests include Internet traffic measurement, Information centric networks and high speed networking. He is a Senior Member of IEEE and ACM.