Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Measuring network throughput in the cloud: The case of Amazon EC2

Valerio Persico[a], Pietro Marchetta[a], Alessio Botta[a,b,*], Antonio Pescapè[a,b]

[a] *Electrical Engineering and Information Technologies, University of Napoli "Federico II", Via Claudio 21, 80125 Napoli, Italy*
[b] *NM2 srl, Italy, Via Fiumiciello, 7, 80142 Napoli, Italy*

ARTICLE INFO

Article history:
Received 25 February 2015
Revised 9 August 2015
Accepted 19 September 2015
Available online xxx

Keywords:
Cloud networking monitoring and measurement
Cloud networking performance
Cloud network throughput

ABSTRACT

Cloud providers employ sophisticated virtualization techniques and strategies for sharing resources among a large number of largely uncoordinated and mutually untrusted customers. The shared networking environment, in particular, dictates the need for mechanisms to partition network resources among virtual machines. At the same time, the performance of applications deployed over these virtual machines may be heavily impacted by the performance of the underlying network, and therefore by such mechanisms. Nevertheless, due to security and commercial reasons, providers rarely provide detailed information on network organization, performance, and mechanisms employed to regulate it. In addition, the scientific literature only provides a blurred image of the network performance inside the cloud. The few available pioneer works marginally focus on this aspect, use different methodologies, operate in few limited scenarios, or report conflicting results.

In this paper, we present a detailed analysis of the performance of the internal network of Amazon EC2, performed by adopting a *non-cooperative* experimental evaluation approach (i.e. not relying on provider support). Our aim is to provide a quantitative assessment of the networking performance as a function of the several variables available, such as geographic region, resource price or size. We propose a detailed methodology to perform this kind of analysis, which we believe is essential in a such complex and dynamic environment. During this analysis we have discovered and analyzed the limitations enforced by Amazon over customer traffic in terms of maximum throughput allowed. Thanks to our work it is possible to understand how the complex mechanisms enforced by the provider in order to manage its infrastructure impact the performance perceived by the cloud customers and potentially tamper with monitoring and controlling approaches previously proposed in literature. Leveraging our knowledge of the bandwidth-limiting mechanisms, we then present a clear picture of the maximum throughput achievable in Amazon EC2 network, shedding light on when and how such maximum throughput can be achieved and at which cost.

1. Introduction

An increasing number of Internet services as well as private IT infrastructures have now been moved to the cloud, mainly due to several economical and technical benefits (e.g. services on-demand, reduced costs, optimized hardware and software resources utilization, performance flexibility) [12,16,24,32]. The industry more and more critically depends

[*] Corresponding author at: Electrical Engineering and Information Technologies, University of Napoli "Federico II", Via Claudio 21, 80125 Napoli, Italy.
Tel.: +390817683865.
E-mail addresses: valerio.persico@unina.it (V. Persico), pietro.marchetta@unina.it (P. Marchetta), a.botta@unina.it (A. Botta), pescape@unina.it (A. Pescapè).

http://dx.doi.org/10.1016/j.comnet.2015.09.037
1389-1286/© 2015 Elsevier B.V. All rights reserved.

on public cloud infrastructures. But this dependence has grown much faster than our understanding of the performance limits, dynamics, and evolution of these facilities. The scarce comprehension is the consequence ofseveral factors: (i) detailed information about cloud performance, characteristics, settings, and load are considered confidential by cloud providers for security and commercial reasons [26,29]; (ii) virtualization limits customers' understanding of if and how their applications' performance is impacted by other customers [37]; (iii) service level agreements (SLAs) only vaguely describe the performance guarantees, and customers can only refer to incomplete and rough information advertised by the provider [6].

A clear example of this lack of knowledge is related to the intra-cloud, high-performance network (i.e. the network connecting hosts inside the same cloud data center), an essential component of the cloud architecture. All cloud providers grant (high-performance) network connectivity to their customers (commercial users, researchers, etc.) and deploy accurate monitoring tools to continuously check the status of the cloud. However, they seldom make promises about network performance figures achievable, and more importantly, they typically provide only qualitative or coarse-grained information about such performance [6,26]. As a consequence, cloud customers suffer from being mostly unaware of the quality of service their applications may receive from the intra-cloud network, in a context where virtualization and resource sharing may introduce substantial performance penalties for data-intensive or computation-intensive applications [36,37]. For these important reasons, measuring intra-cloud network performance has very recently attracted interests from the research community [24,32], and *non-cooperative* approaches seem the only viable solution for general customers to obtain detailed information in public cloud.

1.1. Motivations

Several scientific works adopted non-cooperative approaches to shed light on the intra-cloud network performance, with the valuable goal of characterizing the network performance, supporting optimized virtual machine (VM) allocation, and comparing different cloud providers. Unfortunately, the overall current picture offered by the literature is blurred, with different methodologies leading to conflicting results that are hard to compare. Often, the performance of the intra-cloud network is only one of the aspects marginally analyzed by these works, and the adopted methodology is not exhaustively described. This further complicates the replication of the analysis in the same or other contexts. In addition, the potential dramatic impact of the virtualization and the advanced network resource allocation strategies have been only rarely taken into account, thus weakening a correct interpretation of the results reported [37]. Finally, the performance of intra-cloud network has been characterized only in few and roughly described scenarios among all the possible ones in which a cloud customer may operate, limiting the representativeness of the documented findings.

This blurred image together with the potential applications of this knowledge motivates thorough analyses of the intra-cloud network performance.

1.2. Amazon EC2: an important case study

In this paper, we aim at improving the comprehension of the intra-cloud network performance by focusing on a specific cloud service: Amazon Elastic Compute Cloud (EC2). We use this provider as a case study to highlight how characterizing the network performance is challenging and how classic methodologies and tools may provide misleading results in the complex cloud environment, especially when virtualization and advanced network allocation strategies are not correctly taken into account. Amazon EC2, the Infrastructure as a Service (IaaS) offered by Amazon, is one of the most popular IaaS services that provide computing resources over the Internet [23].

Amazon reports clear information on the allocation of resources such as memory or CPU to the VMs available through EC2. However, much less information is reported regarding the way the provider allocates and guarantees network resources to such VMs. Networking performance is not advertised by the provider through quantitative metrics, but rather through a qualitative description (e.g., *Low to Moderate*, *Moderate*, and *High*) [3]. Hence, a general customer going to instantiate a cloud resource cannot be aware of the expected networking performance related to a certain cost, while he/she is only provided with basic information (e.g., that a VM of a certain size is expected to have networking performance *better* than another one whose hourly cost is smaller). Coarse grained information is made available by Amazon to customers through *Cloudwatch*, a monitoring service for AWS resources and applications, which reports metrics for CPU utilization, data transfer, and disk usage activities for each VM. Regarding the network performance, however, Cloudwatch only reports the incoming/outgoing volume of traffic to/from each VM with 5 min resolution (up to 1 min resolution with extra fees). Hence, cloud users and researchers are typically forced to adopt non-cooperative network monitoring approaches to derive more detailed statistics about the network performance.

1.3. Challenges

Characterizing intra-cloud network performance with non-cooperative approaches is particularly challenging for a number of reasons we discuss in the following.

The number of possible scenarios is extremely high. Despite the several efforts in literature, exhaustively evaluating the network performance of a cloud provider is practically unfeasible for most researchers. In fact, the number of possible scenarios in which a cloud user may operate is so high that it would be extremely costly (both in terms of money and time) to carry out all the required analyses. As an example, Table 1 reports some of the customizable parameters in Amazon EC2 having a potential impact on the network performance. A researcher aiming to exhaustively evaluate the intra-cloud network performance of Amazon EC2 should normally consider (i) all the availability zones deployed in each geographical region (27) as well as (ii) all the possible combinations of VM types and sizes (28^2). This already yields a number of scenarios to be analyzed that is larger than 20, 000. If we sum up other potentially relevant aspects such as the operating system and scenarios involving more than one

Table 1
List of potential parameters to consider when measuring the network throughput in Amazon EC2.

Parameter	Possible values
VM type	General purpose (T2, M3), compute optimized (C4, C3), memory optimized (R3), storage optimized (I2, HS1)
VM size	micro, small, medium, large, xlarge, 2xlarge, 4xlarge, 8xlarge
VM region (available zones)	2 × EU (5), 1 × US East (5), 3 × US West (6), 3 × Asia Pacific (7), 1 × China (1), 1 × South America (2)
VM operating system (available versions)	Windows (Windows server (27), Amazon Linux, Debian (2), SUSE (5), FreeBSD (2), CentOS (20), Red Hat Enterprise Linux (14), SUSE Linux Enterprise Server (1), Ubuntu (7), other, Linux (1)

availability zone, we obtain a number of combinations that are beyond the possibilities of most researchers. Finally, we should also consider that providing meaningful statistics on the network performance requires repeating measurements over a large observation period, also varying the characteristics of probing traffic. In summary, subsampling the set of possible scenarios is inevitable. But doing so without losing generality is very challenging. We propose a methodology to cope with this complexity and describe in details all the characteristics of the scenarios we considered.

Network resources allocation strategies may heavily impact measurement results. Network virtualization and dynamic allocation strategies transparently employed by the cloud provider may strongly impact network performance measurements, whose results may appear misleading or incorrect. As we detail along the paper, in our experimentation we have observed both (i) traffic shaping policies causing transient fluctuation of the network throughput measured and (ii) limitations on the maximum rate at which the traffic can be delivered to a VM depending on its size. In this paper we provide evidence of such policies and limitations, also showing the impact caused on performance measurements. Moreover, we show the throughput achievable by the VMs as a function of the size and considering all the possible variables under control.

Cloud environments rapidly evolve over time. Cloud providers continuously work to (i) build new facilities, (ii) improve the underlying technologies, and (iii) provide new more efficient services, also possibly replacing the old ones. Nevertheless, they only provide qualitative descriptions for the performance attainable on the intra-cloud network. The continuous evolution complicates the research work aiming at characterizing the performance of cloud environments in general and of cloud networks in particular. We argue that in this context, it is of the utmost importance that scientific works focusing on cloud performance detail as much as possible the environment settings taken into account as well as the methodology adopted to carry out the analysis. In this way, the documented analysis can possibly be replicated and the results can be properly interpreted and considered in longitudinal studies.

1.4. Paper contributions

In order to improve the understanding of intra-cloud network performance, we performed more than 5000 h of experimentation to characterize the network throughput offered by Amazon EC2. We did not rely on the limited and coarse-grained information made available by Amazon and provide, for the first time in literature, the following main contributions:

- We propose and describe in details a methodology (choice of metrics, identification of the observation period, selection of scenarios of interest, etc.) to perform measurements in cloud environments in order to obtain a significant performance characterization;
- We improve the understanding of the complex policies and limitations of the intra-cloud network and characterize and quantify their impact on measurement experiments and network throughput in general;
- We carefully characterize the network throughput in a large set of different scenarios obtained varying parameters including the size of virtual machine, the data center geographical region, the transport protocol, the addressing mechanism, etc.

Compared to the literature, we present a comprehensive view of the network throughput of Amazon EC2, also highlighting the specific conditions in which these performance can be achieved by the customers. We also exhaustively describe the adopted methodology to encourage similar analysis for other cloud providers and to foster longitudinal studies of the intra-cloud network performance.

1.5. Paper organization

The paper is organized as follows. Section 2 provides an overall picture of the related literature and positions the paper accordingly. Section 3 presents a methodology to deal with the several challenges related to the characterization of the intra-cloud network performance, also providing a common ground to interpret the results. In Section 4, we discuss the ability of the VMs to generate network traffic, and how the throughput varies with the sending rate, also deepening the possible causes of the highlighted trends. In Section 5 we then provide a clear picture of the network throughput in Amazon EC2. Finally Section 6 ends the paper with concluding remarks.

2. Related work

Measuring the network throughput. A commonly adopted index of the performance perceived by a network communication is the maximum achievable throughput, which depends on the remaining capacity along the path (i.e. the available bandwidth). In the last decade, many techniques and tools for measuring the end-to-end available bandwidth in a network path have been proposed, evaluated, and compared [9]. Under the assumption that end-hosts are not a bottleneck for the communication, a simple yet effective – although intrusive – approach to estimate the maximum throughput is the injection of synthetic traffic [18]. Several synthetic traffic generators exist (e.g., *iperf* [35], *netperf*

Please cite this article as: V. Persico et al., Measuring network throughput in the cloud: The case of Amazon EC2, Computer Networks (2015), http://dx.doi.org/10.1016/j.comnet.2015.09.037

[19], *D-ITG* [5,8,14], and *nuttcp* [15]). They differ in terms of complexity and features, allowing users to measure the network throughput as well as other performance parameters.

In this paper we adopt synthetic traffic generation for estimating the network throughput. More specifically, we use the tool *nuttcp* which well fits the requirements of our analyses, as described in Section 3.

Network throughput in public cloud. Monitoring cloud performance has recently attracted great interest [4]. Many researchers analyzed network performance not relying on the information advertised by the cloud provider, and used the results to compare different providers and support *network-aware* decisions. Unfortunately, these pioneer works adopted different methodologies and tools, reporting conflicting results that are hard to compare. Seldom the adopted methodology is described in enough details to allow the replication of the analysis. Due to the strong challenges we reported above, very few of the possible scenarios have been tested, which strongly limits the representativeness of the provided results. In the following, we review related works focusing on Amazon EC2.

Table 2 provides the overall picture on the Amazon EC2 intra-cloud network performance provided by the literature. Li *et al.* [24] proposed a non-cooperative approach to benchmark different clouds in terms of cost, VM deployment time, computation, storage, and networking. Regarding the network performance, they focused on both the intra-cloud and the wide-area network, and measured throughput and latency using *iperf* and *ping*. For the Amazon EC2 intra-cloud network, the authors measured a TCP throughput in the range [600–900] Mbps. Due to the cost of the measurements, however, the authors also admitted that their results are achieved in few specific scenarios and cannot be considered general.[1] Wang and Ng [37] focused on the impact of virtualization on networking performance in public clouds and characterized it for EC2. They took advantage of *ping* and *ad hoc* tools to characterize intra-cloud network performance using small and medium VM sizes. The authors measured significant delay variation and throughput instability. According to them, this variability seems not to be related to any explicit rate shaping enforced by the provider. The paper reports maximum network throughput of 700–900 Mbps for medium-sized VMs with both TCP and UDP. The authors performed experiments over space (large number of VMs, short time interval) and time (reduced number of VMs, long time interval). Shad *et al.* [32] carried out a study on the performance unpredictability of AWS. Different benchmarks were proposed to evaluate VM deployment time, CPU, memory and disk I/O performance, storage service access, and network bandwidth. Regarding network performance, the authors used *iperf* to evaluate maximum TCP and UDP throughput. They found that networking performance (available bandwidth intra- and inter-availability zone) ranges from 200 to 800 KB/s (1.6–6.4 Mbps) in US data center and from 400 to 900 KB/s (3.2–7.2 Mbps) in Europe data center. The authors reported that the network performance is 9% higher for instances placed inside the same availability zone. We note that these values are strongly conflicting with those reported in previous studies. Raiciu *et al.* [29] used different tools (*traceroute*, *ping*, and *iperf*) to obtain a blueprint of the EC2 network performance and took advantage of it to properly deploy applications and optimize their performance. They reported evidences of paths between VMs of different lengths and with available bandwidth between 1 and 4 Gbps, depending on VM mutual position. Finally, LaCurts *et al.* [20] described an approach to improve application performance by deploying the applications on the nodes with adequate network performance. Since customers have no direct control of VM placement, authors proposed a system called Choreo: this system enforces application placing after measuring the network performance between VM pairs through UDP packet trains. The measurement study performed by the authors to motivate their system is based on *netperf*. This study showed a large variability of network throughput measured with medium-sized VMs from Amazon EC2. Such parameter varied between 300 and 4400 Mbps, and most of the measurements (80%) reported values between 900 and 1100 Mbps.

Compared to the state of the art, we also adopt a non-cooperative approach to characterize the network throughput in Amazon EC2. However, we firstly use a much larger set of scenarios of interest. Secondly, we thoroughly detail the methodology adopted to foster the replication and validation of our analysis also for other cloud providers and scenarios. As we detail in the following, network resource allocation strategies and their impact have very rarely been considered in literature.

Network resource allocation strategies. The literature describes several possible strategies that can be used by cloud providers to dynamically allocate network resources among customers. Although many models for allocating network resources have been proposed and are publicly known, public-cloud providers typically employ their own customized solutions [24] and no detailed information is disclosed to customers [3]. Such strategies aim at supporting diverse needs and differ in terms of goals. We briefly describe the most common ones in the following. We point the reader to [26] for more details. Common strategies are: enforcement of a global rate-limit on the overall aggregated traffic generated by all the sites of a customer (*distributed rate limiting*) [28]; allocation of congested links between customers according to weights based on specific policies, e.g. based on payment [22]; definition of differentiated service models, guaranteeing bandwidth among specific endpoints and treating traffic as best effort for other endpoints (*pipe model*) [17]; efficient bandwidth allocation (i) to achieve max–min fairness across VMs, sending traffic through congestion-controlled hypervisor-to-hypervisor tunnels [33], or (ii) to provide predictable network performance, giving the illusion of a single, nonblocking switch, connecting all the VMs of a customer, where each VM has a minimum guaranteed bandwidth (*hose model*) [31]. Moreover, the presence of traffic shaping has been widely analyzed in broadband access networks [7,10,13,21,34]. On the other hand, to the best of our knowledge, its adoption and impact in the public clouds have not been properly analyzed.

Taking into account the effects of these strategies, we provide a clear picture of the maximum throughput achievable

[1] Note that the authors used labels instead of names to identify the different providers. We inferred EC2 performance among their results by looking at the different geographical regions of the data centers.

Table 2
The overall picture of the intra-cloud network performance of Amazon EC2 from the literature. NA stands for not available information.

Paper	Year	VM type/size	EC2 regions	Measured throughput [Mbps]	Notes
Li *et al.* [24]	2010	NA/NA	US (North California, North Virginia), EU (Ireland)	[600–900]	- Different throughput variability in different regions - No impact of availability zone
Wang and Ng [37]	2010	NA/small	US (North California), EU (Ireland)	[400–800] (small)	- 10 s long measurements
		NA/medium		[700–900] (medium)	
Shad *et al.* [32]	2010	NA/small	US (North California), EU (Ireland)	[1.6–6.4] (US)	- Higher variability when VMs are placed in different availability zones
				[3.2–7.2] (EU)	- Higher variability in US region
Raiciu *et al.* [29]	2012	NA/medium	NA	[1000–4000]	- Available bandwidth related to mutual position
LaCurts *et al.* [20]	2013	NA/medium	NA	[296–4405]	- 10 s long measurements

on EC2 network and of how such a maximum can be obtained.

3. A methodology to characterize cloud network performance

Characterizing the intra-cloud network throughput is a very challenging task for several reasons including the extremely high number of possible scenarios in which a cloud customer may operate (see Section 1.3). In this section, we describe the choices we made to deal with this complexity. We first define the factors of interest to identify and motivate the scenarios we took into account (Section 3.1). Then, we detail the reference architecture as well as the settings, tools, and metrics we adopt (Section 3.2). Our intent is to ease as much as possible the understanding of the precise conditions in which we perform the analysis and then its replication in other scenarios or for other cloud providers. We believe that this methodology represents an important contribution for the analysis of public cloud networks.

3.1. Sampling the scenarios of interest

Amazon EC2 allows customers to highly customize their environment. The high number of available options translates into a large number of scenarios in which a cloud customer may operate. Sampling the space of possible scenarios is necessary when the goal is to provide meaningful and representative results of the intra-cloud performance while keeping the complexity and cost of the analysis acceptable. In the following, we discuss the factors having, in our opinion, a major potential impact on the intra-cloud network performance as well as our sampling strategy. The scenarios analyzed in this paper are obtained by combining all the sampled values of these factors.

3.1.1. Service model

Although the network can impact the performance of distributed applications in all the three layers defined by NIST [25] (Infrastructure-, Platform-, and Software-as-a-Service, or IaaS, PaaS, and SaaS, respectively), we believe that the best layer to characterize the performance of the intra-cloud network is IaaS. This layer guarantees the level of flexibility needed for this analysis, as also demonstrated by previous works [24,32,37]. In particular, IaaS allows us to deploy and use widely-used network diagnostic and measurement tools in the cloud, making common operating system API available. Moreover, direct access to the VM allowed at this layer provides higher control over the factors of influence for the network performance.

Table 3
Selected sizes for VM of type *m3* (Dec 2014). ⋆*: prices vary across regions.*

VM Size	vCPU	RAM (GB)	Networking performance	Hourly cost (min-max)⋆ (€ /*h*)
Medium	1	3.75	Moderate	0.070–0.098
Large	2	7.5	Moderate	0.140–0.196
Xlarge	4	15	High	0.280–0.392

3.1.2. VM type and size

When instantiating VMs on the cloud, a customer can choose their type and size among the ones made available by the provider. In this way, users can take advantage of different preconfigured settings in terms of storage size, computation capabilities, and network performance. Machine hourly cost changes according to the previously mentioned characteristics.

As reported in Table 1, Amazon EC2 makes VM types optimized for different goals available with the intent of easing the configuration of the cloud environment. Customers can select VMs that are optimized for storage, computation, etc. according to their needs. In our experiments, we focused on *general purpose* VMs, which provide a balance of CPU, memory, and network resources, making them the best choice for many applications (small and medium-sized databases, memory-hungry data processing tasks, and back-end servers for SAP, Microsoft SharePoint, and other enterprise applications [3]). In more detail, in our experimental campaigns we used *paravirtual* (as virtualization type) and *m3* VMs (general purpose, new generation) of three different sizes, namely *m3.medium, m3.large, m3.xlarge*. Hereafter, we respectively refer to them simply as *medium*, *large*, and *xlarge*. As reported on the Amazon EC2 website [3], this type of VM has fixed performance (in terms of CPU), which guarantees the absence of CPU resource-sharing and performance-variability phenomena that could impact the measurement process [37]. Table 3 contains more details on the VMs adopted in our analyses.

Table 4
Selected regions.

Region	Continent	Launched
North Virginia	North America	2006
Ireland	Europe	2007
Singapore	Asia	2010
Sao Paulo	South America	2011

The documentation by Amazon clearly describes the available resources in terms of memory and CPU. On the other hand, it only reports a qualitative description of the network performance expected such as *Low*, *Moderate*, and *High* [3]. As we show in the next sections, although the provider advertises *Moderate* network performance for both medium and large VMs, we have experimentally observed that large VMs obtain much higher performance also from the network point of view.

3.1.3. VM geographical region and availability zone

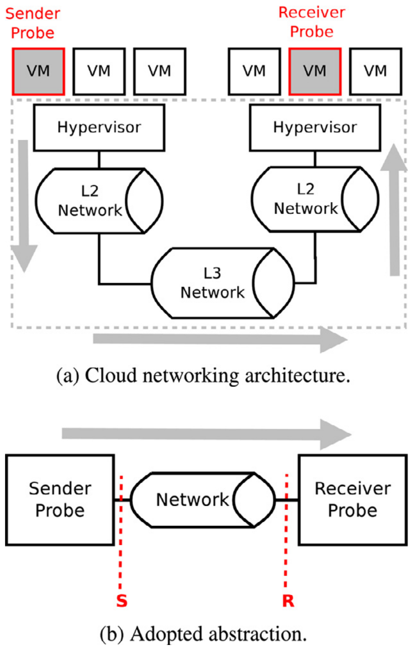
When placing a new VM in the Amazon cloud, the customer can choose among a number of different geographically distributed regions (see Table 1). Customers select different regions to meet their own technical and legal requirements. Each region is associated to different data centers claimed to be completely independent from the others. Since 2006, Amazon deployed data centers in 11 different regions spread world-wide. Works in the literature either omit the region under test [20,29] or report performance variations across regions [24,32].

Our goal is to measure the network throughput between VMs placed inside the same region. Since this operation is costly, we are forced to select a subset of all the possible regions for our experiments. We have selected four regions placed in different continents, to obtain a representative picture of the network performance of the cloud provider. These regions were activated at different times from 2006 to 2011. Hence, they also potentially leverage different technologies such as the processor families [32]. As shown in Table 4, the regions we selected are: *North Virginia (North America)*, *Ireland (EU)*, *Singapore (Asia)*, and *Sao Paulo (South America)*.

Inside each Amazon region, the customer has at his/her disposal multiple *availability zones* (i.e. different locations advertised to be interconnected through low-latency links), opening the possibility of designing robust applications able to overcome potential zone fails [3]. Results in the literature are conflicting about the impact of availability zones on the network performance [24,32]. We have deployed VMs in different availability zones part of the same region, to evaluate the impact of this choice. Results regarding this aspect are reported when relevant.

3.1.4. Communication channel

The VMs on Amazon EC2 can be reached through a public or a private IP address [30]. Private addresses can be used only for communications between VMs deployed in the same data center. Public addresses, instead, allow the VMs to be reachable from the public Internet. Communicating through public addresses, however, comes at an additional cost, depending on the traffic volume. We have measured the achievable throughput when the receiver VM is reached both through its private and public address. We refer to these logical communication channels as *private* and *public channel*, respectively. Note that private and public channels may not correspond to the same physical path.



(a) Cloud networking architecture.
(b) Adopted abstraction.

Fig. 1. Cloud network architecture and its abstraction. Two different monitoring points can be identified for each experiment, able to catch the dynamics of outgoing traffic at the sender (S) and of incoming traffic at the receiver (R).

3.1.5. VM relocation

We have also investigated the impact of VM *relocation* on the network performance, i.e. what happens when VMs are destroyed and created from scratch. The aim is to understand whether the choices operated by the provider when deploying the configuration required by the customer have an impact on the network performance.

3.2. Reference architecture, tools, and metrics

In our experiments we adopt the reference architecture and the settings, tools, and metrics we detail and motivate in the following.

3.2.1. Reference architecture

Fig. 1 reports the conceptual scheme we refer to in our experimental campaigns. We aim at measuring the network throughput between a pair of VMs under the control of the same cloud customer. These VMs are instrumented with a standard operating system and all the necessary network measurement and diagnostic tools we used for estimating the network performance. Hereinafter, we refer to a VM as *probe*. More specifically, we use the term *sender* and *receiver* probe to identify the VM in charge of sending and receiving the network traffic, respectively. In each region, we have used sender and receiver probes of different sizes (i.e. medium, large, and xlarge). As depicted in Fig. 1a, the traffic of the sender probe normally first traverses the hypervisor layer at the sender side. Then it flows through L2/L3 devices and middleboxes composing the high performance network. Finally, the traffic reaches the hypervisor at the receiver side before being delivered to the receiver probe. Note that as a cloud customer, we are not aware of the specific location of the sender and receiver probes, which may also be hosted and managed by the same hypervisor. Furthermore, by adopting a black box approach, we consider the L2/L3 devices as well as the hypervisors as part of the network connecting sender and receiver probe (Fig. 1b). Basically, we consider as *network* all the logical and physical components interposed between the virtual network cards connecting sender and receiver VMs. Note that this choice entirely fits the point of view of the general cloud customer who has no visibility on the cloud physical infrastructure, network internal dynamics, and cloud provider policies, despite their potentially heavy impact on the performance he/she perceives.

3.2.2. Tools and settings

We have used the network diagnostic and measurement tool named *nuttcp* to measure the network throughput between the sender and receiver probe. We have chosen *nuttcp* after an initial experimental campaign in which we have used and compared the most widely used similar tools. This initial campaign has shown that *nuttcp* is able to respect the imposed bitrate more accurately than other tools, which do not generate the full bitrate required, very likely because of the virtualized environment of the cloud. This aspect is very important for our measurements, as detailed in the following. Thanks to *nuttcp*, we have determined the raw TCP or UDP network layer throughput by transferring memory buffers from sender to destination probes. Traffic has been generated either for a specified interval or for a given amount of bytes. Besides the information provided by *nuttcp*, we also take advantage of two additional monitoring points, in order to infer and characterize the effects of the network resource allocation strategy employed by the cloud provider. More specifically, we derive the rate of the traffic being transmitted through the (virtual) network interface of the sender probe and the rate of the traffic arriving at the network interface of the receiver probe. We derive this information monitoring the traffic volume exposed by the Linux operating system API reporting the status of the network interface. As we demonstrate in the next sections, monitoring the network interface of a VM discloses valuable information. Differently from what happens in traditional computing environments, the network interface and link capacity of each VM in the cloud is virtualized and directly controlled by the hypervisor [2]. Here, the cloud provider may potentially implement sophisticated network resource allocation schemes.

In our analysis, we characterize the network throughput by comparing the amount of traffic *nuttcp* is configured to generate (hereinafter referred to as *target traffic*), with the observation at the two monitoring points described above, i.e., the outgoing rate observed at the sender monitoring point (hereinafter referred to as *true sending rate*) and the rate measured at the receiver side (*receiving rate*). Note that the true sending rate represents the rate at which the VM delivers data to the hypervisor which is already considered as part of the network in our methodology.

We have measured the network throughput for both UDP and TCP transport protocols [11] in our experiments. On the one hand, UDP is typically used to analyze the performance of the raw IP traffic. UDP adds no closed loop-control, leaving the complete control on the generated traffic to the probe, no matter what the state of the network is. On the other hand, TCP throughput, which is governed by flow and congestion control, allows probe traffic to be subjected to the status of the network path and provides information on the performance of the numerous TCP-based applications.

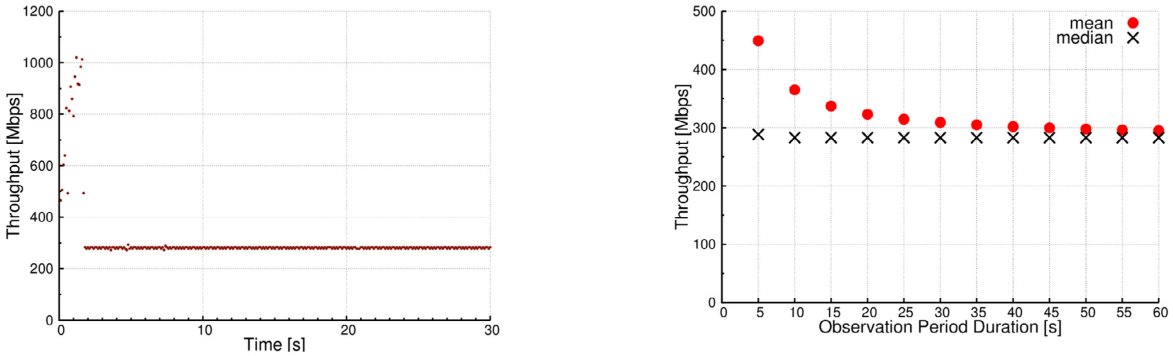
Finally, we have also investigated the impact of the packet size and performed repeated experiments to investigate the presence of daily or weekly patterns in the intra-cloud network performance.

3.2.3. Metric for network throughput

Measuring network throughput in the cloud can be very costly. This operation consumes computation and network resources that are charged by the cloud provider according to the pay-as-you-go paradigm. Furthermore, fast and accurate measurements are highly appreciated to guarantee high responsiveness to those frameworks exploiting network measurements [20,29]. As a consequence, finding a good tradeoff between accuracy and cost is of the utmost importance.

Cost and accuracy depend on the duration of the observation period, and the metric used to evaluate the network throughput may determine misleading results. Tuning the duration of the observation period as well as selecting the right metric for the network throughput is further complicated by the effects of the mechanisms employed by the provider to reach the desired network resource allocation. An example is reported in Fig. 2. An intra-cloud communication in Amazon EC2 typically reaches a higher network throughput during a first transient period, and then settles to a lower yet stable value. Applications using short-lived communications may obtain higher, although unstable network throughput. Fig. 2a reports an example of this phenomenon observed with TCP traffic. Note that this trend cannot be explained with TCP internal dynamics such as slow start or congestion control mechanisms. Indeed, a similar atypical behavior was *always* observed also in all the UDP-based communications we monitored. Hence, we consider this as a clear evidence of traffic shaping policies (e.g. token bucket), employed by the cloud provider as network resource allocation strategy.

The presence of this initial throughput spike cannot be ignored by the researchers willing to provide an accurate view of the network performance. Measurements performed during the initial interval are not representative of the expected performance over longer periods. The initial spike at the beginning of the communication may also explain the different throughput ranges of values reported in literature (see Section 2). Fig. 2b shows how well mean and median values calculated over observation periods of increasing durations properly capture the maximum network throughput in the stable period. We have monitored the network throughput between medium-sized VMs over intervals of different durations. We have then computed the different metrics (i.e. mean and median) by only considering the throughput samples obtained during the first 5 s, first 10 s, and so on. The figure shows that the mean throughput value converges



(a) First 30 seconds of a TCP intra-cloud communication highlighting the presence of an initial transient spike. Similar spikes were observed in all the TCP and UDP communications monitored.

(b) Capturing the stable maximum network throughput achievable with different metrics.

Fig. 2. Measuring network throughput in Amazon EC2. Typically, the network throughput reaches a stable value only after an initial transient period, likely due to the network resource allocation strategy employed by the cloud provider (a). The initial spike impacts the accuracy of the network throughput measurements. The median value captures the stable value of network throughput much sooner than the mean value, requiring a shorter and cheaper observation period (b).

to the stable value much slower than the median one. This finding is consistent across all the experimental campaigns we performed, i.e., for different combinations of VM sizes, in different regions, for different types of traffic, over different channels.

According to these results, we have decided to report the median value of the network throughput observed over observation periods lasting at least 8 min. This metric represents the *stable throughput* achievable in a communication between VMs deployed in the same Amazon region, filtering the noise caused by this initial transitory. In the following we refer to this value simply as the maximum throughput. Note that choosing the median is not *universally* the right choice, but it represents a valid option for the cloud provider under test.

The proposed methodology is general, i.e. it can be easily adapted and applied to all the public cloud providers and for other network performance indexes such as jitter, latency, and packet loss. In more general terms, we believe that the methodology described above clearly identifies all the relevant aspects to carefully consider when the final goal is evaluating the performance of public intra-cloud networks.

4. Throughput trends in Amazon EC2

In this section, we first focus on the VM traffic-generation capabilities representing a potential source of inaccuracy when carrying out this type of analysis. Then, we highlight trends in the network throughput measured at the receiver side. Finally, we dig into a potential root-cause of the highlighted trends. In these analyses, we have relied on UDP since its behavior is not affected by the condition of the network path, differently from TCP (see Section 3.2.2).

4.1. Impact of the sender VM

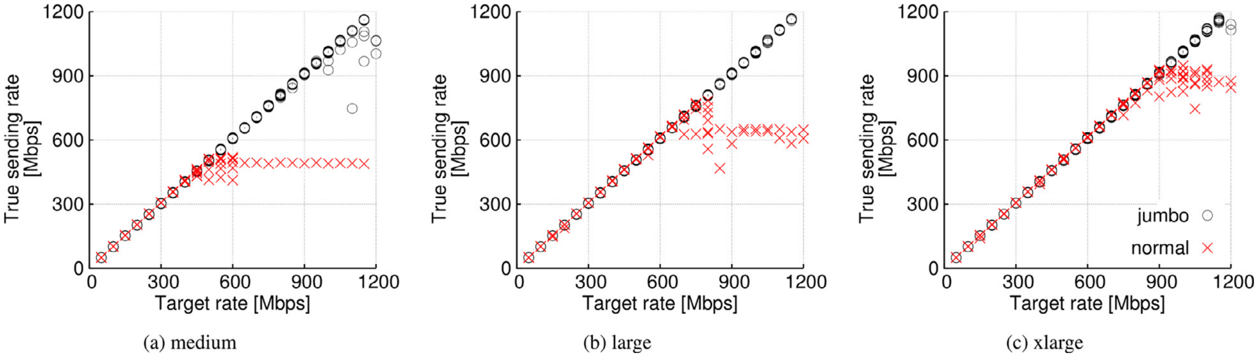
The network throughput can be accurately measured through synthetic traffic generation only when the computation capabilities of the involved end hosts are not a bottleneck for the communication. For instance, if the sender host and the measurement tool are not capable to fulfill the remaining capacity of the network path, the available network throughput is incorrectly underestimated. In the cloud, it is essential to check whether and in which conditions this assumption holds since virtualization proved to (i) introduce significant performance penalties to applications [36], (ii) invalidate measurement outcomes [37], and (iii) compromise the interpretation of typical measurement metrics [38]. Synthetic traffic has been widely adopted in previous works [20,24,29,32,37] but the potential impact of the VM-generation capabilities has been neglected in such literature.

We have instructed *nuttcp* to generate traffic at a given target rate and monitored the true sending rate (i.e. the rate of the traffic actually flowing into the network) to check whether the tool is able to sustain the target rate on a given VM.

We have performed experiments with two different application-level packet sizes: 1024 B and 8192 B (hereafter simply *normal* and *jumbo* UDP packets).[2] We have performed 350 experiments 8 min long for each VM size in different regions, with target rates ranging from 50 to 1200 Mbps. Target rate and true sending rate are compared in Fig. 3: VMs of any size are not able to inject traffic into the network at a rate higher than a given threshold (i.e. a *cap*) when using normal packets. This cap proved to mainly depend on the sender VM and its size. In more details, the cap has proved to be very stable over time for a fixed VM: experimental results have shown that the Coefficient of Variation (CoV, $\frac{\sigma}{|\mu|}$) of the cap is always smaller than 2% for any observation period up to 72 h. However, relocating (i.e. destroying and re-creating) the VM, also in the same region and with the same size, may reveal different cap values. Table 5 reports aggregated statistics on the cap values for all the considered regions: larger VMs can achieve a higher value of the maximum true sending rate with normal packets, which can be explained

[2] Note that TCP protocol does not suit this kind of analysis due to the congestion control mechanism that would force the sending rate to be limited by the bottleneck along the whole end-to-end path (see Section 3.2.2).

Please cite this article as: V. Persico et al., Measuring network throughput in the cloud: The case of Amazon EC2, Computer Networks (2015), http://dx.doi.org/10.1016/j.comnet.2015.09.037



(a) medium

(b) large

(c) xlarge

Fig. 3. Target rate vs. true sending rate for different sending probe sizes. When using normal packets (1024 B), the true sending rate does not overcome a cap value. This limitation is not observed in case of jumbo packets (8192 B).

Table 5
Cap in Mbps on true sending rate observed when using normal packets: mean (± std dev).

Region	medium	large	xlarge
North Virginia	489.1(± 17)	747.3(± 9.0)	944.1(± 19.1)
Ireland	495.5(± 20.0)	731.8(± 10.3)	948.0(± 15.3)
Singapore	485.5(± 3.8)	730.2(± 9.7)	925.1(± 22.8)
Sao Paulo	492.6(± 5.3)	748.1(± 24.5)	1018.3(± 43.8)

with the resource partition enforced by the provider (e.g. higher computation capabilities to larger VMs). On the other hand, we have observed no cap on the true sending rate when using jumbo packets: in this case, the target rate is achieved by imposing a much lower load on the virtual CPU.

Fig. 4 reports the distribution of the cap values for the EU region (Ireland) with VMs relocated several times. In this region, medium, large, and xlarge instances are subjected to a cap imposing a maximum throughput of 495.5, 731.8 and 948.0 Mbps on average, respectively. Interestingly, although Amazon advertises that both medium and large VMs receive *Moderate* networking performance [3], our results clearly show that large instances are allowed to inject traffic into the network at a much higher rate. We have also noticed a higher variability of the cap imposed to xlarge VMs, with 63% of large instances receiving a cap higher than 5% of xlarge VMs.

In summary, synthetic traffic generation capability of the adopted VMs should be carefully taken into account when the final goal is measuring the intra-cloud network performance through non-cooperative approaches. We have observed that the adopted measurement tool on EC2 VMs is not able to generate traffic at the requested target rate when relying on 1024-byte packets. This is not true when using jumbo packets. The obvious conclusion might be the adoption of jumbo packets. However, we will see in the following that this choice can have a detrimental impact on the network throughput at the receiver side.

4.2. Impact of packet size and public/private channel

We have discovered that packet size and communication channel (public or private) have an impact on the network throughput measured. Fig. 5 reports how the network throughput measured at the receiver side changes when the true sending rate increases for all the nine possible combinations of sender and receiver sizes. In this analysis, we have instructed the sender probe to perform 8 min long generations of UDP traffic for each target rate. We have considered target rates ranging from 50 to 1200 Mbps. In each experiment, we have extracted the median value of true sending rate at the sender side and the median value of the network throughput at the receiver side. Overall, we have performed 350 experiments for each region by also relocating the VMs. The results reported in Fig. 5 are related to the EU region (Ireland) but they are quantitatively and qualitatively representative also for the other regions.

The figure shows that the network throughput saturates to a maximum value independently from the packet size. Such value represents the maximum throughput an application deployed on a VM can achieve towards another VM in the same datacenter, through the network slice granted by the cloud provider. Interestingly, the figure also shows that jumbo packets allow generating traffic at higher rate but such higher rate dictates a drastic decrease of the network throughput at the receiver. The figure highlights a common pattern in the network throughput as a function of the true sending rate. The shape of the curves can be modeled through the following equations, describing the network throughput $R(x)$ as a function of the true sending rate x for each packet size:

$$\text{Jumbo packets}: R(x) = \begin{cases} x & x \le \alpha \\ \alpha & \alpha < x \le \beta \\ \Psi(x) & x > \beta \end{cases}$$

$$\text{Normal packets}: \quad R(x) = \begin{cases} x & x \le \alpha \\ \alpha & x < cap \end{cases}$$

Basically, the network throughput increases with the true sending rate up to a first value, which depends on the packet size. We have named this value *flattening edge* (α) because after this point, the throughput trend is typically flat or at least it does not increase, saturating to a constant value. After this point, the two packet sizes show significantly different behaviors. With jumbo packets the network throughput at the receiver side starts to strongly decrease after a certain value of the true sending rate (the phase represented by $\Psi(x)$

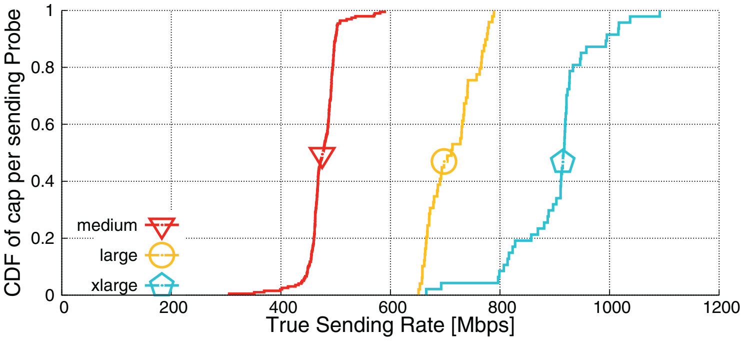
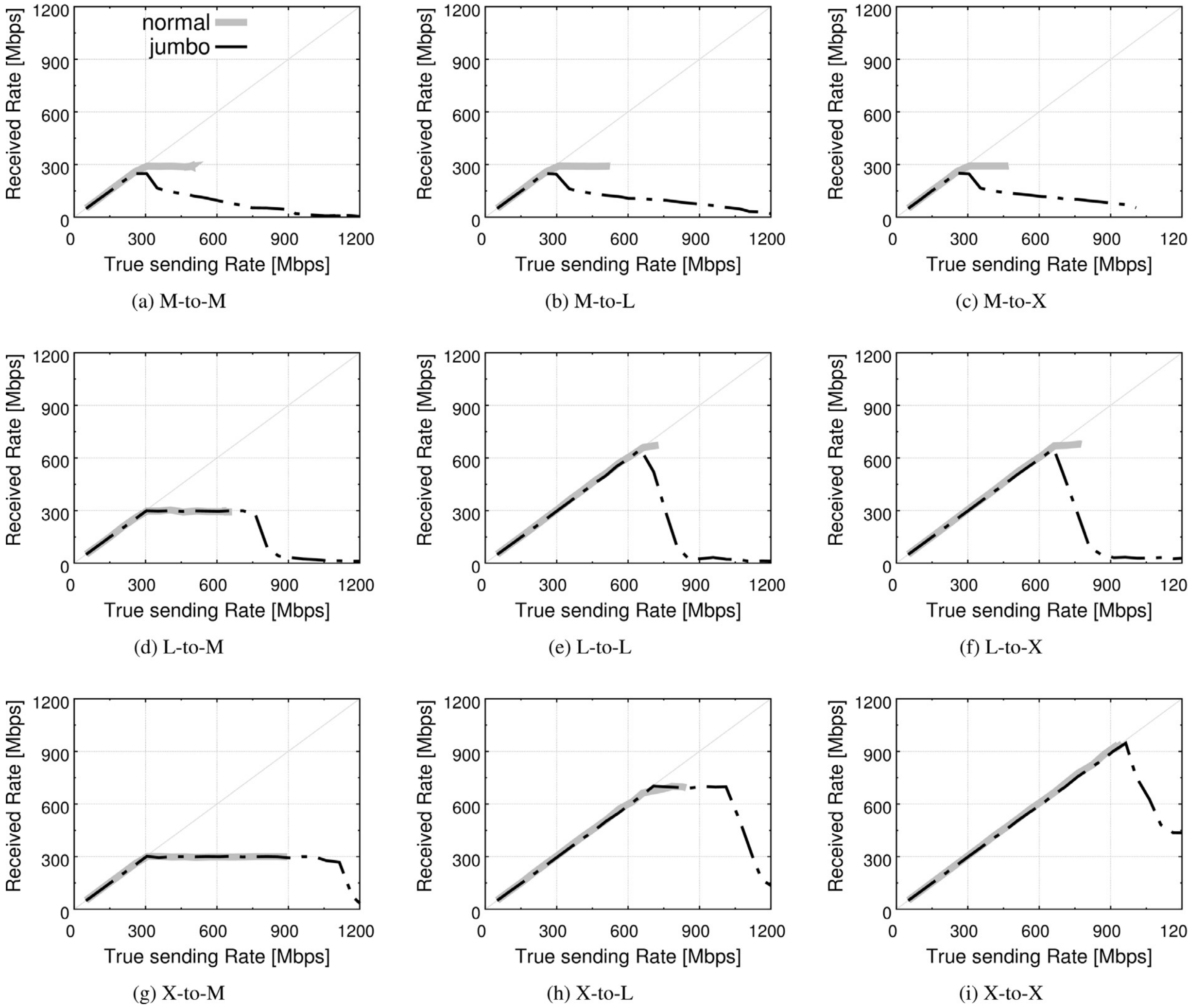


Fig. 4. Cap value distributions for EU Region (Ireland). The values of the cap observed when using normal packets vary with the size of the virtual machine: the larger the VM size, the higher the true sending rate allowed.



(a) M-to-M

(b) M-to-L

(c) M-to-X

(d) L-to-M

(e) L-to-L

(f) L-to-X

(g) X-to-M

(h) X-to-L

(i) X-to-X

Fig. 5. Maximum UDP throughput towards the VM public address. The network throughput at receiver side dramatically decreases at high true sending rates when using jumbo packets (dashed black lines). This behavior is not observed with normal packets (solid gray lines). *M:medium, L:large, X:xlarge.*

Table 6
Estimated values for the (a) flattening and (b) penalty edge. The tables show the average values computed over the experiments performed in different regions. Standard deviation omitted being negligible.

(a) Flattening edge – The network throughput saturates after this value of true sending rate. *N: normal packets, J: jumbo packets.*

		receiver		
		medium	large	xlarge
sender	**medium**	$251.9^{(J)}$ – $307.6^{(N)}$	$251.5^{(J)}$ – $307.7^{(N)}$	$252.0^{(J)}$ – $307.4^{(N)}$
	large	$302.2^{(J)}$ – $306.2^{(N)}$	$656.9^{(J)}$ – $678.8^{(N)}$	$656.8^{(J)}$ – $663.6^{(N)}$
	xlarge	$304.8^{(J)}$ – $308.8^{(N)}$	$710.6^{(J)}$ – $664.2^{(N)}$	$961.5^{(J)}$ – $924.1^{(N)}$

(b) Penalty edge – the network throughput rapidly decreases for true sending rate higher than this value. This happens only with jumbo packets.

		receiver		
		medium	large	xlarge
sender	**medium**	302.9	302.2	303.1
	large	708.1	656.9	656.8
	xlarge	1111.4	1013.9	961.5

in the previous equations). We have named the value of the true sending rate after which the throughput starts decreasing as *penalty edge* (β) because after this value, the network throughput significantly drops. On the other hand, this specific trend is not spotted when adopting normal packets. Indeed, we notice that the network throughput does not decrease after the penalty edge when using normal packets (see Fig. 5a for instance), even if high true sending rates are not achieved when relying on this kind of synthetic traffic (see Section 4.1). In Section 4.3 we provide the results of further analyses in order to explain this trend.

For normal packet traffic, we have experimentally observed that the evolution of the network throughput with the true sending rate is highly predictable. For jumbo packets, this property is experimentally verified up to the penalty edge: with higher true sending rate, we have always noticed a strong decrease of the network throughput but we could not derive an exact trend. The figure also clearly shows how the flattening and penalty edges depend on the size of sender and receiver probes: detailed values are reported in Table 6. The penalty edge values significantly increase for larger sender sizes. Considering the values of the medium-sized sender as a baseline, we have observed a growth for this threshold of more than 2 × (3×) for large (xlarge) sender probes. The flattening edge values, instead, seem to be determined by the smallest between sender and receiver VM size, i.e. the threshold increases only in case of larger size for both sender and receiver. The values of this threshold also depend on the packet size: especially for medium-sized sender probes (Fig. 5, first row), we have noticed higher values for the flattening edge when comparing normal to jumbo packets, while differences are also noticed for xlarge sender probes communicating with either large or xlarge instances. Finally, Table 6 also reveals those sender–receiver combinations for which the flattening edge is equal to the penalty edge: these combinations correspond to the curves in Fig. 5 where the network throughput starts decreasing immediately after the growing trend (Fig. 5e–i). Note that this throughput decreasing trend has not been observed on the private channel. We dig into this phenomenon in the next section.

The impact of these results is twofold. On the one hand, researchers aiming at characterizing the performance of the cloud network may strongly underestimate the maximum throughput. This happens if they rely on the injection of UDP traffic at high target rate, as often suggested by classic methodologies. Indeed, injecting traffic at high rate always determines very low network throughput at destination, when issuing jumbo packets on the public channel. On the other hand, users and applications seeking the highest network throughput at destination have to carefully limit the sending rate.

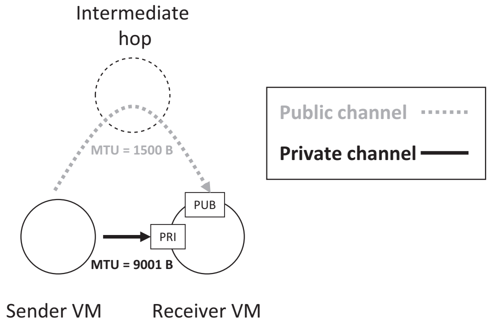


Fig. 6. EC2 intra-cloud paths. Traffic directed to the receiver VM crosses two different paths when directed to the private (PRI) and the public (PUB) IP address. In the latter case an intermediate hop is traversed.

4.3. Deepening the throughput detrimental effect

We experimentally observed the maximum network throughput achieved on the public channel *always* strongly decreases when (i) using jumbo packets, and (ii) the true sending rate overcomes a threshold we named penalty edge. This phenomenon, however, was never observed when the communication was established through the private channel. We performed further analyses looking for differences between private and public channels potentially explaining the causes of the observed phenomenon.

In this analysis, we employed the network diagnostic tool named *tracepath* [1] to infer the characteristics of private and public channels. Tracepath represents an evolution of the classic *traceroute* tool, providing additional path-related information. We took advantage of tracepath and performed multiple experiments (from 5 to 10) for each of the considered scenarios. Interestingly the outcome was the same across them. The results of this analysis are outlined in Fig. 6. We noticed three main differences between public and private channels. First, we discovered that the network traffic flowing through the private channel *always* directly reaches the receiver probe whereas one intermediate network-layer device is *always* traversed on the public channel. This device is likely the middlebox in charge of translating public into

Table 7
Maximum stable throughput for Amazon EC2 across different regions when the receiver VM is reached through the public or private IP address. *M:medium, L:large, X:xlarge.*

Sender	UDP		TCP	
to receiver	Public $\mu \pm dev$	Private $\mu \pm dev$	Public $\mu \pm dev$	Private $\mu \pm dev$
M to M	291 ± 0	298 ± 1	293 ± 1	299 ± 0
M to L	291 ± 0	300 ± 2	293 ± 1	300 ± 0
M to X	291 ± 0	298 ± 2	293 ± 1	298 ± 2
L to M	300 ± 1	300 ± 1	299 ± 0	300 ± 1
L to L	665 ± 13	696 ± 3	684 ± 1	699 ± 1
L to X	670 ± 6	694 ± 4	685 ± 2	700 ± 1
X to M	299 ± 1	300 ± 1	299 ± 2	301 ± 1
X to L	708 ± 22	698 ± 5	699 ± 2	702 ± 0
X to X	897 ± 16	993 ± 8	939 ± 4	996 ± 1

private addresses. Differently from [29], we never observed paths connecting sender and receiver probes involving more than one intermediate hop. Several possibilities may explain this discrepancy including operational changes in the data center such a more efficient VM allocation strategy posing the VMs in the proximity of each other, as well as a change in the internal network infrastructure in terms of devices or configurations. Second, the Maximum Transmission Unit (MTU) is 9 KB on the path of the private channel – thus supporting jumbo frames – while it is only 1.5 KB on the public one. Third, and consequently, injecting jumbo packets on the public channel induces IP packet fragmentation. We experimentally observed that packet fragmentation occurs directly at the sender VM. These results were verified across all the tested regions.

Based on these findings, we can provide a possible explanation of the observed phenomenon. Using jumbo packets allows the sender to easily inject synthetic traffic into the network at the desired rate. However, the injected packets are fragmented on the public channel determining a potentially disruptive impact on the throughput measured at the receiver side. Indeed, each jumbo packet is fragmented in 6 smaller packets: losing even one of these fragments causes an entire jumbo packet to be discarded.

5. Maximum throughput between two VMs

Thanks to the acquired knowledge, we can now provide an overall picture of the maximum throughput between two VMs in Amazon EC2. We have performed 10 min lasting experiments for each considered scenario (i.e. for each of the combinations obtained by varying region, protocol, channel, and VM-size combination) at the maximum target rate achievable. Table 7 reports mean and standard deviation of the median throughput measured across the different geographical regions when the receiver VM is reached through its private and public address. Recall that the throughput reported in Table 7 is very stable over time. Experimental results support the following findings.

- Comparing Tables 5 and 7, we can see that the traffic generation cap on the sender side does not significantly impact the network throughput measured, on average. Hence, the network throughput measured through synthetic traffic generation can be considered reliable since the sender machine proved not to be a bottleneck for the communication.

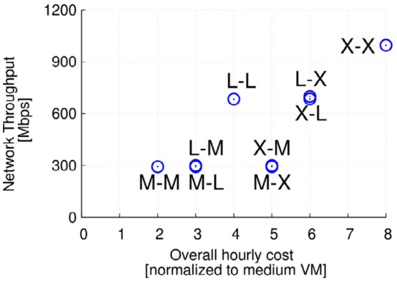


Fig. 7. Maximum unidirectional (sender-to-receiver) throughput vs. hourly cost. When focusing only on network performance, the best option in EC2 is using VMs of the same size. *M:medium, L:large, X:xlarge.*

- The intra-cloud network throughput is very similar across the different regions (see the small standard deviation values): the cloud provider seems to adopt a strategy to guarantee similar network performance to its customers in different regions. This is interesting considering that the hourly cost for a VM varies with the regions (there is a gap of +40% between the least and most expensive regions). Consider the case of a user having a distributed application running on multiple VMs, exchanging data among them, and whose performance depends on the network throughput but not on the location of the data center (e.g. a scientific application). In this case, deploying all the VMs inside the cheapest region seems the best option to obtain the maximum performance at the lowest cost. Note this finding is not valid for other cloud operators such as Microsoft Azure [27], where the intra-cloud network throughput measured in different regions significantly varied. This further highlights how each cloud provider has its own way of organizing the network resources [24].
- Traffic is exchanged at a slightly higher rate along the private channel compared to the public one. Also considering the extra-fee paid to use public channels, cloud customers should always prefer the private channels over the public ones, when possible.
- Also, for almost all the explored combinations of VM size, we have observed equal or higher network throughput for TCP compared to UDP. Our analyses indicate that the cloud provider allows medium, large, and xlarge VMs to deliver UDP (TCP) traffic at maximum 300 (300), 696 (700), 993 (996) Mbps, respectively. Similarly, medium, large, and xlarge VMs are allowed to receive UDP (TCP) traffic at maximum 300 (301), 708 (702), 993 (996) Mbps. Hence, although EC2 documentation reports as *Moderate* the network performance for both medium and large instances, our results show that large VMs definitely receive more network resources than medium instances.
- Finally, when the network throughput is the most important aspect, our results show that the best performance can be obtained with VMs of the same size. Indeed, the maximum throughput is always limited by the minimum size between the sender and receiver. Fig. 7 compares the network performance (i.e. the maximum

sender-to-receiver throughput) and the total hourly cost normalized to the cost of a single medium VM. Using two large VMs seems the best trade-off between cost and network performance. We have achieved the same conclusion when also using other (non network-related) performance indexes instead of the cost, such as the overall number of virtual CPUs or size of the amount of memory assigned to the VM. Note that according to Fig. 7 the network throughput is not a monotonic function of the overall hourly cost.

We can compare Tables 7 and 2 to highlight a few important differences carefully considering that (i) previous works rarely provided details about which specific type of VMs they employed for the experimentation, and (ii) the features offered by the provider to the customer may have changed over time. We have experimentally observed that the size of the VM has a huge impact on the perceived network performance, an aspect underrated in [20,24,32] and only partially considered in [37]. We have measured a much lower network throughput for medium instances (250–300 Mbps) than the one reported in [20] (700–900 Mbps), [29] (1000–4000 Mbps), and [37] (296–4405 Mbps). A first possible explanation for this discrepancy is a change in the operational status of these data centers, potentially caused by the deployment of higher-performance applications or by a variation of the resource allocation strategy. Another possible explanation may be spotted looking at the adopted methodology. These works monitored the network throughput with experiments during only 10 s. As we already described in Section 3.2.3, network throughput in Amazon EC2 is typically much higher and unstable during a first transient period of time. This throughput burst over short observation periods may heavily impact the accuracy of the measurements.

6. Conclusion

In public cloud environments, the performance of data- and computation-intensive applications deployed over multiple VMs can be highly impacted by the performance of the intra-cloud network. Unfortunately, cloud providers do not disclose sufficiently detailed information about such networks. Moreover, the pioneer works in literature focusing on this aspect adopted different methodologies and few limited scenarios, reporting also conflicting results. This blurring image of intra-cloud network performance is further weakened by (i) the extremely large number of possible scenarios a cloud customer may operate in as well as (ii) the advanced network-resource allocation strategies employed by the provider, both having an impact on the network performance perceived.

In this paper, we have identified challenges and pitfalls when characterizing the intra-cloud network performance. We have highlighted them by using as a case study Amazon EC2, one of the most popular IaaS services. For the first time in the literature, to the best of our knowledge, we have proposed and thoroughly detailed a methodology to perform this kind of analysis and to define the precise conditions to contextualize and properly interpret the results obtained. This represents a first advancement beyond the state of the art, which has often underrated the challenges of these analyses. Thanks to 5000 h of experimentation, we have achieved the following main findings. We have experimentally observed short-lived initial transient throughput spikes in the communication between VMs representing a clear evidence of policy enforcement adopted by the cloud provider through network resource allocation mechanisms (e.g. traffic shaping). To avoid ambiguities, we have identified a suitable and cost-effective metric (i.e. the median) to quantify the stable network throughput. We have verified that VMs cannot inject traffic into the network at a rate higher than a threshold depending on the VM size and the adopted packet size. However, these limits do not impact the throughput measured, on average. In contrast, the throughput has proven to be strongly dependent on the smallest size between sender and receiver VMs while other factors, including the geographic region, have only limited influence. We have experimentally noticed that the measured network throughput significantly drops when the traffic consists of jumbo packets and the sending rate exceeds a given threshold. Since the resulting traffic consists of a large amount of fragments, we have identified the disruptive impact of packet loss as the possible cause of this behavior: losing even one fragment causes an entire jumbo packet to be dropped. Finally, although medium and large instances should receive the same network performance (i.e. *Moderate*) according to the official EC2 documentation, large VMs have been proven to receive much more network resources.

As future work, we plan to deepen the intra-cloud network performance for this and other cloud providers by also considering other relevant performance indexes such as latency and jitter. We also plan to extend these analyses to inter-cloud scenarios where the VMs are deployed in different regions or in different cloud environments.

Acknowledgments

This work is partially funded by the MIUR projects: PLATINO (*PON*01_01007), SMART HEALTH (*PON*04*a*2_*C*), S2-MOVE (*PON*04*a*3_00058), SIRIO (*PON*01_02425), and Art. 11 DM 593/2000 for NM2 srl. The experimental work in this paper was also supported by a grant provided by Amazon AWS in Education.

References

[1] Man page for `tracepath`, http://www.unix.com/man-page/linux/8/tracepath/.
[2] Xen Networking, http://wiki.xenproject.org/wiki/Xen_Networking.
[3] Amazon web services website, instance type matrix, instance type matrix, December 2014. http://aws.amazon.com/.
[4] G. Aceto, A. Botta, W. de Donato, A. Pescapè, Cloud monitoring: a survey, Comput. Netw. 57 (9) (2013) 2093–2115, doi:10.1016/j.comnet.2013.04.001.
[5] S. Avallone, A. Pescapè, G. Ventre, Distributed internet traffic generator (D-ITG): analysis and experimentation over heterogeneous networks, in: ICNP 2003 poster Proceedings of the International Conference on Network Protocol, 2003.
[6] H. Ballani, P. Costa, T. Karagiannis, A. Rowstron, Towards predictable datacenter networks, in: Proceedings of the ACM SIGCOMM 2011 Conference, ACM, New York, NY, USA, 2011, pp. 242–253, doi:10.1145/2018436.2018465.
[7] S. Bauer, D. Clark, W. Lehr, Powerboost, in: Proceedings of the Second ACM SIGCOMM Workshop on Home Networks, HomeNets '11, ACM, New York, NY, USA, 2011, pp. 7–12, doi:10.1145/2018567.2018570.
[8] A. Botta, A. Dainotti, A. Pescapè, A tool for the generation of realistic network workload for emerging networking scenarios, Comput. Netw. 56 (15) (2012) 3531–3547.

[9] A. Botta, A. Davy, B. Meskill, G. Aceto, Active techniques for available bandwidth estimation: comparison and application, in: E. Biersack, C. Callegari, M. Matijasevic (Eds.), Data Traffic Monitoring and Analysis, Lecture Notes in Computer Science, 7754, Springer, Berlin, Heidelberg, 2013, pp. 28–43, doi:10.1007/978-3-642-36784-7_2.
[10] A. Botta, A. Pescapè, On the performance of new generation satellite broadband internet services, Commun. Mag. IEEE 52 (6) (2014) 202–209.
[11] A. Botta, A. Pescapè, G. Ventre, Quality of service statistics over heterogeneous networks: analysis and applications, Eur. J. Oper. Res. 191 (3) (2008) 1075–1088.
[12] Y. Chen, S. Jain, V. Adhikari, Z.-L. Zhang, K. Xu, A first look at inter-data center traffic characteristics via Yahoo! datasets, in: Proceedings of the 2011 IEEE International Conference on Computer Communications INFOCOM, 2011, 2011, pp. 1620–1628, doi:10.1109/INFCOM.2011.5934955.
[13] M. Dischinger, A. Haeberlen, K.P. Gummadi, S. Saroiu, Characterizing residential broadband networks, in: Proceedings of the Seventh ACM SIGCOMM Conference on Internet Measurement IMC, 2007, pp. 43–56.
[14] D. Emma, A. Pescapè, G. Ventre, Analysis and experimentation of an open distributed platform for synthetic traffic generation, in: Proceedings of the Tenth IEEE International Workshop on Future Trends of Distributed Computing Systems, FTDCS 2004, IEEE, 2004, pp. 277–283.
[15] B. Fink, R. Scott, nuttcp, http://nuttcp.net/ v6.1.2.
[16] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, The cost of a cloud: research problems in data center networks, SIGCOMM Comput. Commun. Rev. 39 (1) (2008) 68–73, doi:10.1145/1496091.1496103.
[17] C. Guo, G. Lu, H.J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, Y. Zhang, SecondNet: A data center network virtualization architecture with bandwidth guarantees, in: Proceedings of the Sixth International Conference, Co-NEXT '10, ACM, New York, NY, USA, 2010, pp. 15:1–15:12, doi:10.1145/1921168.1921188.
[18] M. Jain, C. Dovrolis, End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput, SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 295–308, doi:10.1145/964725.633054.
[19] R. Jones, et al., Netperf: a network performance benchmark, Information Networks Division, Hewlett-Packard Company, 1996.
[20] K. LaCurts, S. Deng, A. Goyal, H. Balakrishnan, Choreo: network-aware task placement for cloud applications, in: Proceedings of the 2013 Conference on Internet Measurement Conference, IMC'13, ACM, New York, NY, USA, 2013, pp. 191–204, doi:10.1145/2504730.2504744.
[21] K. Lakshminarayanan, V.N. Padmanabhan, Some findings on the network performance of broadband hosts, in: Proceedings of the Third ACM SIGCOMM Conference on Internet Measurement, IMC '03, ACM, New York, NY, USA, 2003, pp. 45–50, doi:10.1145/948205.948212.
[22] T. Lam, S. Radhakrishnan, A. Vahdat, G. Varghese, NetShare: virtualizing data center networks across services, Department of Computer Science and Engineering, University of California, San Diego, 2010.
[23] L. Leong, et al., Magic Quadrant for Cloud Infrastructure as a Service, Worldwide report, Gartner, Inc., 2015.
[24] A. Li, X. Yang, S. Kandula, M. Zhang, Cloudcmp: Comparing public cloud providers, in: Proceedings of the Tenth ACM SIGCOMM Conference on Internet Measurement, IMC '10, ACM, New York, NY, USA, 2010, pp. 1–14, doi:10.1145/1879141.1879143.
[25] P. Mell, T. Grance, The NIST Definition of Cloud Computing, 2011., NIST Special Publication 800 (2011): 145.
[26] J.C. Mogul, L. Popa, What we talk about when we talk about cloud network performance, SIGCOMM Comput. Commun. Rev. 42 (5) (2012) 44–48, doi:10.1145/2378956.2378964.
[27] V. Persico, P. Marchetta, A. Botta, A. Pescapè, On Network Throughput Variability in Microsoft Azure Cloud, in: Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM),, 2015.
[28] B. Raghavan, K. Vishwanath, S. Ramabhadran, K. Yocum, A.C. Snoeren, Cloud control with distributed rate limiting, SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 337–348, doi:10.1145/1282427.1282419.
[29] C. Raiciu, M. Ionescu, D. Niculescu, Opening up black box networks with cloudtalk, in: Proceedings of the Fourth USENIX Conference on Hot Topics in Cloud Computing, 2012, p. 6.
[30] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear, Address Allocation for Private Internets, 1996, (RFC 1918 (Best Current Practice)). Updated by RFC 6761.
[31] H. Rodrigues, J.R. Santos, Y. Turner, P. Soares, D. Guedes, Gatekeeper: supporting bandwidth guarantees for multi-tenant datacenter networks, in: Proceedings of the Third Conference on I/O Virtualization, USENIX Association, Berkeley, CA, USA, 2011, p. 6.WIOV'11
[32] J. Schad, J. Dittrich, J.-A. Quiané-Ruiz, Runtime measurements in the cloud: observing, analyzing, and reducing variance, Proc. VLDB Endow. 3 (1–2) (2010) 460–471, doi:10.14778/1920841.1920902.
[33] A. Shieh, S. Kandula, A.G. Greenberg, C. Kim, B. Saha, Sharing the data center network., in: Proceedings of the Eighth USENIX Conference on Networked Systems Design and Implementation NSDI, 2011.
[34] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, A. Pescapè, Measuring home broadband performance, Commun. ACM 55 (2012) 100–109.
[35] iPerf: The network bandwidth measurement tool, http://iperf.fr, July 2015.
[36] R. Tudoran, A. Costan, G. Antoniu, L. Bougé, A performance evaluation of azure and nimbus clouds for scientific applications, in: Proceedings of the Second International Workshop on Cloud Computing Platforms, CloudCP '12, ACM, New York, NY, USA, 2012, pp. 4:1–4:6, doi:10.1145/2168697.2168701.
[37] G. Wang, T. Ng, The impact of virtualization on network performance of Amazon EC2 data center, in: Proceedings of the 2010 IEEE Conference on Information Communications INFOCOM, 2010, pp. 1–9, doi:10.1109/INFCOM.2010.5461931.
[38] J. Whiteaker, F. Schneider, R. Teixeira, Explaining packet delays under virtualization, ACM SIGCOMM Comput. Commun. Rev. 41 (1) (2011) 38–44.

Valerio Persico is a PhD student at the Department of Electrical Engineering and Information Technology (DIETI) of the University of Napoli Federico II (Italy), where he received his MS degree in 2013, defending a thesis about a novel technique for topology discovery of IP networks. He is a member of the research group called Traffic, working in the area of computer networks and multimedia, and part of the larger COMICS (COMputers for Interaction and CommunicationS), a research group on networking. His research interests fall in the area of networking and of IP measurements; in particular his past and present work focuses on traffic classification, IP topology discovery, IP path tracing, IP alias resolution and cloud monitoring. During his PhD he coauthored several conference publications, being awarded with the Best Student Paper Award for his paper “Don't Trust Traceroute (Completely)” at CoNext 2013. He also served and serves as peer reviewer for international conferences and journals such as: IEEE Globecom, IEEE Consumer Communications and Networking Conference (CCNC), IEEE International Conference on Communication (ICC), IEEE International COnference on Cloud Engineering (IC2E) and Elsevier Simulation Modelling Practice and Theory (SIMPAT).

Pietro Marchetta received his PhD in Computer Engineering at University of Napoli in 2014. Currently, he holds a postdoctoral position at the Department of Electrical Engineering and Information Technology of the University of Napoli Federico II (Italy). His main research activities focus on methodologies, techniques and large-scale distributed platforms for Internet Measurements with a specific focus on Internet routing and performance. These activities have been conducted in the context of international collaborations with leading global research groups.

In 2010, he visited the IP Networking Lab at the University of Louvain-la-neuve (UCL), where he contributed to design, develop and evaluate a novel Internet topology discovery approach based on IGMP.

In 2013, he joined as visiting researcher the Networked Systems Laboratory of the University of Southern California (USC) working on path prediction techniques and performance evaluation approaches. He served and serves a reviewer for a dozen of conferences and journals (e.g. Elsevier's Computer Networks and Future Generation Computer Systems).

For his research, he received some awards including the first place at the ACM Student Research Competition at SIGCOMM 2012 and the Best Student Workshop Paper Award in CoNEXT 2013.

Alessio Botta received the M.S. degree in telecommunications engineering and the Ph.D. degree in computer engineering and systems from the University of Napoli Federico II, Napoli, Italy. He currently holds a post-doctoral position with the Department of Computer Engineering and Systems, University of Napoli Federico II. He has co-authored over 50 international journal (the IEEE COMMUNICATIONS MAGAZINE, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and Elsevier's Computer Networks) and conference [the IEEE Global Communications (Globecom), the IEEE International Conference on Communications (ICC), and the IEEE Symposium on Computers and Communications (ISCC)] publications. His current research interests include networking, and, in particular, network performance measurement and improvement, with a focus on wireless and heterogeneous systems. Dr. Botta has served and serves as an independent reviewer of research and implementation project proposals for the Romanian government. He was a recipient of the Best Local Paper Award at the IEEE ISCC 2010. In the research area of networking, he has chaired international conferences and workshops, served and serves several technical program committees of international conferences (IEEE Globecom and IEEE ICC), and acted as a reviewer for different international conferences (the IEEE Conference on Computer Communications) and journals (the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE NETWORK MAGAZINE, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY).

Antonio Pescapè [SM'09] received the M.S. Laurea Degree in Computer Engineering and the Ph.D. in Computer Engineering and Systems at University of Napoli Federico II, Napoli (Italy). He is currently an Associate Professor at the Department of Electrical Engineering and Information Technology of the University of Napoli Federico II (Italy) where he teaches courses in Computer Networks, Computer Architectures, Programming, and Multimedia and he has also supervised and graduated more than 100 among BS, MS, and PhD students. His research interests are in the networking field with focus on Internet Monitoring, Measurements and Management and on Network Security. Antonio Pescapè has co-authored over 140 journal (IEEE ACM Transaction on Networking, Communications of the ACM, IEEE Communications Magazine, JSAC, IEEE Wireless Communications Magazine, IEEE Networks, etc.) and conference (SIGCOMM, Infocom, Conext, IMC, PAM, Globecom, ICC, etc.) publications and he is co-author of a patent. He has served and serves as workshops and conferences Chair (including IEEE ICC (NGN symposium)) and on more than 90 technical program committees of IEEE and ACM conferences. He serves as Editorial Board Member of Journal of Network and Computer Applications and has served as Editorial Board Member of IEEE Survey and Tutorials (2008-2011) and was guest editor for the special issue of Computer Networks on "Traffic classification and its applications to modern networks" and for the special issue of Journal of Future Generation Computer Systems on "Internet of Things and Cloud Services". For his research activities he has received several awards, comprising a Google Faculty Award, several best paper awards and two IRTF (Internet Research Task Force) ANRP (Applied Networking Research Prize). Antonio Pescapè has served and serves as independent reviewer/evaluator of research and implementation projects and project proposals co-funded by the EU Commission, Sweden government, several Italian local governments, Italian Ministry for University and Research (MIUR) and Italian Ministry of Economic Development (MISE). Antonio Pescapè is a Senior Member of the IEEE.