

Testing a Cloud Provider Network for Hybrid P2P and Cloud Streaming Architectures

Javier Cerviño*, Pedro Rodríguez*, Irena Trajkovska*, Alberto Mozo†, and Joaquín Salvachúa*

*Dpto Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain.

{jcervino,prodriguez,irenatr,jsalvachua}@dit.upm.es

†Dpto Arquitectura y Tecnología de Computadores, Universidad Politécnica de Madrid, 28031 Madrid, Spain.
amozo@dit.upm.es

Abstract—The number of online real-time streaming services deployed over network topologies like P2P or centralized ones has remarkably increased in the recent years. This has revealed the lack of networks that are well prepared to respond to this kind of traffic. A hybrid distribution network can be an efficient solution for real-time streaming services. This paper contains the experimental results of streaming distribution in a hybrid architecture that consist of mixed connections among P2P and Cloud nodes that can interoperate together. We have chosen to represent the P2P nodes as PlanetLab machines over the world and the cloud nodes using a Cloud provider’s network. First we present an experimental validation of the Cloud infrastructure’s ability to distribute streaming sessions with respect to some key streaming QoS parameters: jitter, throughput and packet losses. Next we show the results obtained from different test scenarios, when a hybrid distribution network is used. The scenarios measure the improvement of the multimedia QoS parameters, when nodes in the streaming distribution network (located in different continents) are gradually moved into the Cloud provider infrastructure. The overall conclusion is that the QoS of a streaming service can be efficiently improved, unlike in traditional P2P systems and CDN, by deploying a hybrid streaming architecture. This enhancement can be obtained by strategic placing of certain distribution network nodes into the Cloud provider infrastructure, taking advantage of the reduced packet loss and low latency that exists among its datacenters.

Index Terms—Network, Streaming, Videoconference, P2P, Cloud Computing, QoS.

I. INTRODUCTION

The improvements in Internet access technologies e.g., the availability of high bandwidth and the spread of portable media player devices have opened new opportunities to receive high quality, on-demand, and interactive multimedia applications. In the recent years, the use of both peer-to-peer (P2P) and content delivery networks (CDN) ¹ architectures for real time video streaming have aroused much interest in several fields, such as education, TV industries, research communities, etc.

P2P networking has favourable characteristics, such as high scalability, self-configuration and organization. Many people consider them as suitable infrastructures for supporting real time streaming. However, P2P networks possess dynamic characteristics that can decrease drastically the performance of these real-time applications. Among these characteristics, it is worth taking into consideration the ability of peers to enter

and leave the system without any prior notification. This can however cause service interruptions if the adaptive mechanism does not correctly manage such changes. There exist also problems of making the right decision in routing schemes and bandwidth utilization, or dealing with heterogeneity of terminals, such as TV, Laptops, mobile phones or tablets, that also goes hand in hand with the variety of network connections including ADSL, Cable Modem, UMTS, WiFi, etc.

From other side, private content delivery networks have reached a great success in real time streaming, mainly for IPTV, life casting and live video streaming of events. They even allow users to broadcast from mobile phone applications or websites. The implementation of these networks has one great flaw, that is the need of dedicated infrastructure which makes CDN services much more expensive than P2P based systems, mainly for broadcasters.

In [1], we proposed a hybrid and distributed architecture for multimedia streaming combining P2P and cloud computing, focusing on the QoS requirements to make the architecture commercially usable, and offering some QoS APIs for the cloud. This solution is an alternative to P2P-CDN hybrid architectures because it is not based on CDN principles. On the contrary, it implements P2P systems over public and private Cloud infrastructures.

This paper reports an experimental validation of the previous architecture in terms of quality of service (QoS). For doing this, we have carried out two different tests. First of all by measuring some common network QoS parameters, we have checked whether and up to which level it is reasonable enough to use a Cloud provider network for forwarding live streaming media data. Despite research like [2] on QoS such as timeliness, scalability, availability, trust and security required in cloud SLA agreements, or performance and load balancing issues, we have found a gap of measurements in multimedia applications for QoS such as latency, bandwidth, packet loss and jitter in real scenarios combining clients and provider inside and out of the cloud. The same authors [3] have developed an architecture for mobility support of wireless devices for executing multimedia applications on the cloud, whereas the participants in our architecture are static nodes.

Next, we designed four different mixed P2P-Cloud scenarios that comprise most of the service real use cases. These scenarios represent a streaming distribution network, where the

¹<http://www.akamai.com/>

network nodes could be located in the Internet or inside the cloud infrastructure. Our objective with this was to qualify the hybrid network's ability to transmit multimedia streams and also to analyse the trade off between nodes placed in the cloud and peers placed inside a commercial network. Therefore the idea is not to make any comparison among public, private or community cloud, rather to rely on PlanetLab[4] nodes as particular peers that would interoperate with nodes placed in a cloud infrastructure. The obtained results conclude that some scenarios allow to enhance the overall QoS of the streaming service when Cloud network is used to forward streaming sessions among distant peers. For the scope of the measurements we chose the most significant Cloud provider at this moment - Amazon ², leaving out the rest of the providers for future comparisons of the results. We have chosen PlanetLab nodes as an example of experimental peers, since we could obtain variety of machines in different parts of the world that could serve us to represent the P2P part of the architecture.

Similar work that relays on PlanetLab nodes as edge routers in a cloud proxy network is found in [5]. They use such architecture to examine an optimal data distributed operations as a step towards improvement of cloud applications and cloud-proxy network optimization. As it will be commented later in II the reader can find other studies for available Cloud infrastructures, like [6] or [7], but they do not deal with detailed network QoS, at least not for those that are important for real time streaming services. Due to this we extended this research to know exactly the response of those networks.

This paper demonstrates a complete study for two worlds joined in non-trivial experiments. Instead of simulating the different scenarios with software applications we have implemented all of them in existing infrastructure available at a Cloud provider and in PlanetLab nodes around the world. Thus, the results can be taken as real proof that validates one segment of our previous architecture proposal.

In section II we present a summary of previous studies and real time streaming systems that motivated our research. Section III contains the first measurements of the cloud provider network, that is the first step to know how a Cloud infrastructure can enhance the P2P system responsiveness. Section IV explains the different test scenarios in which we were introducing different Cloud hosted nodes for improving the QoS and it reports the obtained results. Finally in section V we comment some conclusions based on these results tracing the way of our following research steps.

II. RELATED WORK

In the business world, cloud computing has started to migrate with increased rhythm from initially being only a buzzword, to securely conquering the public and private market. In the research community we are witnessing an increased number of attempts to qualify and quantify the network performance of a certain provider. Cloud computing rates today as a common infrastructure for hosting various

applications which in the same time attracts the attention of the research world.

We located several works related to analysis of the network I/O performance for applications in the cloud that focus on searching a way to optimize resource sharing e.g. [8] Mei et al. explain this considering the throughput and resource sharing effectiveness as parameters. Other research [9] conduct performance measurements of the storage management addresses, both for the application and the service providers to achieve increased efficiency in I/O performance, resource provisioning and workload scheduling.

In [10] they perform an analysis of EC2's management and security facilities in the same time considering evaluation measurements of Amazon S3 and SQS, finding EC2 good enough when considering cost-time trade-off.

Deelman et al. [11] study the cost-performance analysis of various execution plans used by the same application example. They show that good choice of both storage and compute resources reduces cost which doesn't affect the application performance. As a work to basically examine the use of cloud computing in science, "Nimbus Science Cloud" ³ has been pointed out as an example cloud provider for scientific means. In [12] authors offer a semantic analysis for modelling and testing applications in the cloud through a cloud graph model, that can be used to dynamically compose cloud computations.

Research as [13], are of high interest for our work because their Split&Merge architecture used for encoding time of high performance video could serve us as a guide for treating the video streams in the cloud i.e., coding/decoding and evaluating the cost of this operation.

Finally, Ang Li et al. [7] have recently designed a tool that offers the possibility of comparing several aspects among various cloud providers by taking into consideration parameters such as, CPU velocity, I/O of the disk and latency, bandwidth and response time. The main objective is relating their performance and compare the cost each of them imports. Although their results in some part are similar with the measurements we have independently collected and performed during the period of three months, we focus on applying the knowledge obtained from our metrics to the area of multimedia streaming. Focusing moreover on exhaustive analysis of the Amazon EC2 cloud would lead to profound elaboration of the possibilities that each of its data centers offer, for deploying multimedia cloud applications and possibly of hybrid models with P2P solutions. Interpreting the results would serve as a base to establish more accurate solutions and investigate the network tolerance when such applications interrelate and communicate among each other.

We want to characterize the network of the cloud providers and examine the reliability of QoS parameters such as latency, bandwidth, response time and jitter. Finally, we are interested in how these specific results can reflect in various applications for multimedia streaming and videoconference like [14].

²<http://aws.amazon.com/ec2/>

III. AMAZON NETWORK MEASUREMENTS

As mentioned in the previous section, there exist related research about Amazon network performance (in terms of bandwidth) and computer processing capacity in different kinds of virtual machines. In this section we want to extend this research in order to evaluate the network QoS of the different datacenters around the world owned by Amazon.

We will start by explaining some network features to be measured in the experiments and we will discuss their importance in video and audio streaming. Afterwards we will present the architecture of the environment, giving some details about each component and the datacenters to be examined. Later will be shown the results of the measurements and we will finish by interpreting the collected data, concluding about the observed QoS of the Amazon cloud and whether the network is able to transport video and audio streams at real time.

A. Important network features

In [6] we can see how Wang measures the network performance over different kinds of virtual machines and shows that using Amazon Small Instances is harmful to the network responsiveness. The metrics they use are always inside a local datacenter, excluding measurements between distant ones. We extended this work and based on their arguments to measure the network performance between different Amazon datacenters.

The work in [7] shows how different Cloud providers can offer varying performance in the execution of applications depending on various aspects, like the variation of the requests sent by the clients and the infrastructures owned by Cloud providers. A conclusion of this work is the importance of the network performance offered by the Cloud provider and the responsiveness in traffic between different datacenters of the same provider. The limit of this study is that it does not take into account various datacenters, so we think that this part of

³<http://workspace.globus.org/clouds/nimbus.html>

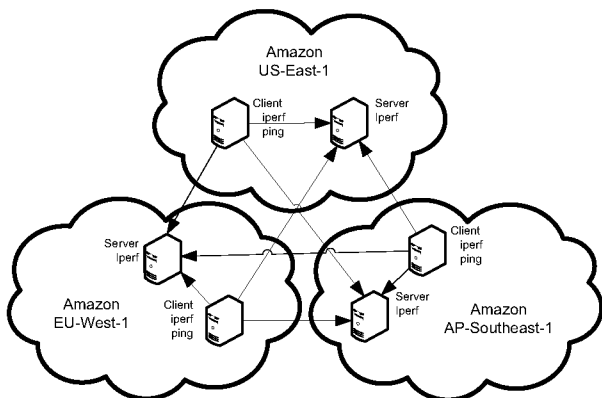


Fig. 1: Architecture of the measuring environment

the study needs to be improved in order to test the hypothesis for our architecture.

For doing this, we considered four very important network parameters for video/audio streaming and for many other real-time services: bandwidth, delay, jitter and packet losses. These are the traditional QoS parameters taken into account in real time communication applications since VoIP ([15] and [16] are two examples).

1) *Bandwidth*: The architecture this study is mainly based on is a hybrid system (P2P and centralized) used for video/audio streaming. Other typical use case, apart from this is a scenario in which end users could see Internet TV channels. These users are supposed to be behind a wide range of access networks such as xDSL, Wifi, etc. Due to these reason a streaming service must allow configuration for different bandwidth. We had to take into account as well, that there is no communication bottleneck in the core network in any of the cases examined. Therefore, in our tests we also had to measure the limits of available Amazon network's bandwidth.

2) *Delay*: We mentioned previously that users can join the service from different access points. We therefore have to take into account possible additional network delays in their connections. They also could access from different places in the world, including regions, countries and even continents. Thus the core network that will be used only as a transport network may introduce inherent propagation delays.

3) *Jitter*: It is the variation in the time between packet arrivals caused by network congestion, timing drift, or route changes. Jitter could manifest in different ways depending on the application that is used. Web browsing is fairly resistant to jitter, but any kind of streaming media (voice, video, music) is quite sensitive to it. As for these reasons jitter is an undesired parameter in any network connection, and according to ETSI [17] it is very important that the amount of jitter is lower than 20 ms in VoIP applications (we want to probe the network performance for videoconference applications too, so we are going to set this limit for the network).

4) *Packet loss*: Packet loss can occur for variety of reasons including link failure, high levels of congestion that lead to buffer overflow in routers, Random Early Detection (RED, a technique used in routers to apply congestion control), Ethernet problems or an occasional misrouted packet. Packet loss causes degradation in voice and video quality. If packet loss concealment (PLC, a technique used to mask the effects of lost or discarded packets) is used then isolated losses may be less noticeable. But for example when packet loss occurs in bursts of 20-30% loss lasting 1-3 seconds, this can mean that the average packet loss rate for a call appears low, although the user reports call quality problems. In our case we want a core transport network with a very low packet loss rate.

B. Architecture of the measuring environment

Figure 1 represents the architecture of the Amazon measurement phase. In this figure we can see three Amazon datacenters in which we have executed several virtual machines. Some of

them were used as servers and others were used as clients. In the clients we run the tool *iperf* and ping applications that took measures about the connections established to the servers. Every virtual machine is a Large Amazon instance that ran Ubuntu Server operating system (Ubuntu Karmic 9.10 AMD64 Server).

The three amazon datacenters used here were from different sessions: US-East-1 in Northern Virginia, EU-West-1 in Ireland and AP-Southeast-1 in Singapur. There is one more region in California that we considered no relevant because of its proximity to the US-East-1. Among the applications we run the *iperf* in server mode for bandwidth, jitter and packet loss measurements with the set up commands as explained below.

1) *Server Bandwidth*: Sends and receives packets using only one TCP connection with read and write buffers size set to 8 KB, maximum segment size equals to 1460 bytes and window size set to 85.3 KByte.

```
iperf -i 10 -s -p 6031 -f k -y C
```

2) *Server Jitter and packet loss*:

```
iperf -i 240 -s -u -p 6032 -f k -y C
```

We also run *iperf* in client mode for bandwidth, jitter and packets loss measuring:

3) *Client Bandwidth*: The configuration is same as in the server.

```
iperf -t 3000 -i 10 -c serverHostname
-p 6031 -r -L 6030 -f k -y C
```

4) *Client Jitter and packet loss*: The read and write buffers size are set to 8 KB.

```
iperf -t 3000 -i 240 -u -c serverHostname
-p 6032 -L 6030 -f k -y C
```

Finally we made measurements of network delay using the *ping* test tool.

C. Measurement results

In the figures 2, 3, 4 we can see the same four graphs respectively: the bandwidth graph represents the cumulative distribution function (CDF) of the tests that measured the amount of the available data bitrate in each connection between two regions, measured in bits per second. Jitter graph represents the CDF of the jitter in these connections, packet loss graph represents the probability density function (PDF) of the packets loss detected between them and finally Round Trip Time (RTT) graph indicates CDF of the round trip time delays between two nodes in different regions.

1) *Measurements between Europe and USA*: Figure 2 represents the measurements between Europe and USA regions. Based on the bandwidth graph the average is over 45 Mbits per second for a single sender thread. Meanwhile jitter results show a jitter bounds between 0.1ms and 1ms, which is enough even for VoIP communications. Regarding the loss rate, the tool proved that it is low enough and appropriate for streaming purposes. Finally RTT is between the margins of 90ms and

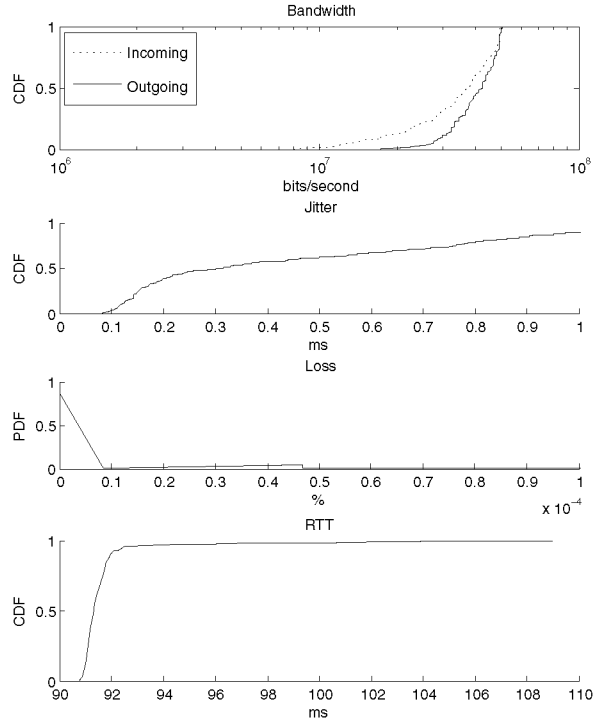


Fig. 2: Measurement results between datacenters EU-US

92ms, that are typical values for connections between distant peers.

2) *Measurements between Europe and Asia*: Figure 3 represents the measurements between Europe and Asia regions. In this case we can see that an increment of the outgoing bandwidth reaches over 70Mbps for a single sender thread. However in this case there are differences between incoming and outgoing traffic because the incoming traffic drops to values between 8 and 20Mbps. Jitter and loss measures are very low again and enough to satisfy streaming service purposes. In the case of RTT we can see an increment in the mean value, that could reach 270ms. It is mainly because the peers are geographically far away from each other.

3) *Measurements between USA and Asia*: Figure 4 represents the measurements between USA and Asia regions. Most of the bandwidth measures show a value of 90Mbps for outgoing data while for the incoming data the bandwidth drops to values over 20Mbps. In this case we can see a short increase in the jitter, that now reaches over 0.3 ms. In spite of this, it stays too low to become a concerning value for streaming services. Once again the loss rate is very low and in this case we see that RTT is over 250ms, that is a typical value for this kind of connections.

As a result of the measurements explained above, we can infer that using a cloud provider such as Amazon to stream multimedia content is highly efficient and meets the QoS

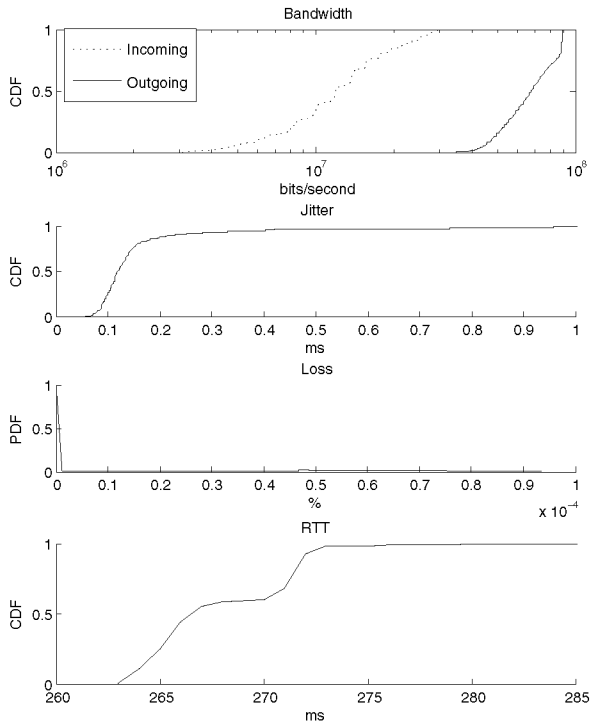


Fig. 3: Measurement results between datacenters EU-AP

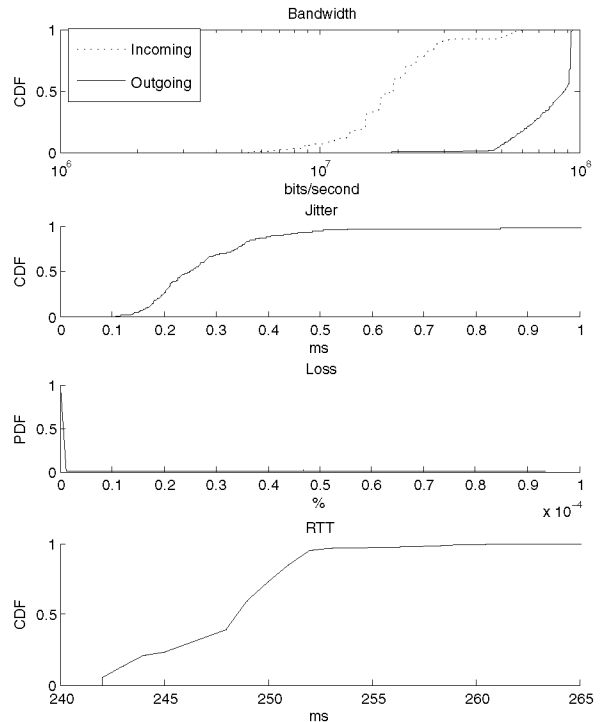


Fig. 4: Measurement results between datacenters US-AP

requirements of a streaming distribution service.

IV. SCENARIOS

In this section we present several scenarios deployed in a tree based streaming distribution network. The nodes of the distribution network are located in different geographical points around the world. Initially, all the nodes are located in the Internet but outside the cloud infrastructure, and gradually, some of them are moved into the cloud infrastructure. The rationale of this experiment is to measure the improvements in terms of quality of service when we gradually move network nodes from Internet to a cloud infrastructure. We use PlanetLab[4] to be able to spread geographically the distribution network nodes in Internet, and Amazon as cloud provider.

To further approximate a real scenario, we used in our tests real video traffic, in order to compare the results to the actual requirements of a multimedia transmission in terms of QoS. In the first subsection we describe the tools used in the measurements as well as the parameters to be measured. After that, we describe the general architecture of the experiment scenarios and the experimental setup. Finally the results are presented.

A. Tools and statistics

Before presenting the architecture itself we will describe the tools used and the way the statistics are obtained. We

need a set of components that allow us to generate media and redistribute it. Furthermore, these parts need to be flexible and easily automatable to make the measuring process as smooth as possible. We have based the tools on a previously developed project toolkit, changing them in accordance to our need and making the necessary adjustments in the environment in order to be able to gather statistics.

The tools we use come from our system Isabel[18], a multipoint group collaboration tool which includes video and audio conferencing. The low level components of Isabel are daemons coded in C/C++ that can be operated via a control port using a set of primitives. The video component is able to encode, decode, transmit, receive and present video in a wide variety of codecs. Furthermore, it can be configured to broadcast video from a file or from a synthetic source of video in different sizes, bit-rates and codecs. Real-time Transport Protocol (RTP) is used to transmit media.

The second component and the most important in terms of characterizing the network is called *irouter* and its main objective is to receive and forward packets from and to other components including other "irouters". In our scenarios the irouter will receive data from the video daemon, present it and then forward it to other irouters. This component was modified to gather data from all incoming packets and to generate the statistics that we will process and present in the results subsection.

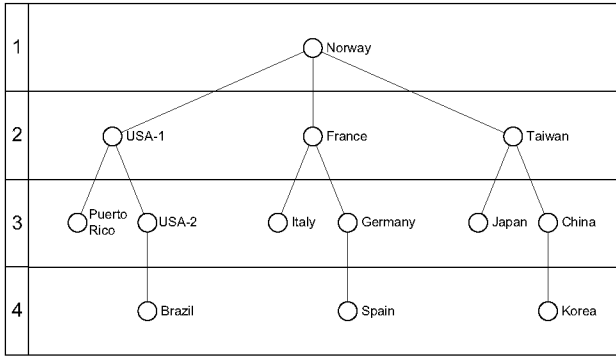


Fig. 5: Node distribution

For practical reasons, the router component dumps the raw data in text files that will later be collected and processed statistically in order to separate the actual measurement process from the calculations. The data accumulated is roughly equivalent to that explained in the Intra-Cloud measurements described in this document. We will now explain the data collected by the router defining each parameter:

- 1) *Packets received*: The total packets received in an interval.
- 2) *Bytes received*: The number of bytes received in an interval.
- 3) *Packets lost*: The amount of packets lost in an interval. A discontinuity in the sequence number of RTP packets is perceived as a loss.
- 4) *Packets disordered*: A packet is considered disordered if it arrives in time to fill a gap in sequence numbers. It is considered to be “in time” if it arrives in the same interval.
- 5) *Packets duplicated*: A packet is considered duplicated if two packets with the same sequence number arrive anytime.
- 6) *Packets recovered*: When forward error correction (FEC) is active, packets recovered via this mechanism are recorded here.
- 7) *Inter-Packet Gaps (IPGs)*: The difference in time in the arrival of two consecutive packets. IPGs are used in order to calculate an average of the jitter introduced by the network. As explained in [16] you can theoretically obtain an accurate approximation to an upper bound of jitter using:

$$\sigma^2 = \frac{1}{N_i - 1} \sum_{j=2}^{N_i} (IPG_j(i) - E[IPG(i)])^2$$

$$J(i) = 3\sigma$$

For a normal distribution of random jitter this value means that the 99.7% packets will arrive with a jitter lower than $J(i)$. But in practice it does not follow a Normal distribution because there are several external causes other than only random noise. So we preferred to calculate directly the 99.7 percentile of values in:

$$|IPG_j(i) - E[IPG(i)]|$$

B. Scenario architecture description

To test our hybrid model, we designed 4 scenarios in 4 levels, to gradually introduce the distribution network nodes from Internet to the cloud infrastructure. The cloud provider we used was Amazon with instances equivalent to those explained in section III. To locate our nodes outside the cloud provider we use PlanetLab that enables us to easily place nodes in different countries.

We also divided geographically the location of the peers to further expose the challenges found when distributing multimedia content among different continents, and to see the advantages of having geography in mind when designing the topology of the distribution. As shown in figure 5 we divided our scenarios into three different zones: Europe, America and Asia and we located 4 peers forming a tree topology. The distribution of the nodes was designed according to the location of Amazon available EC2 locations at the time of planning the tests, that is, roughly West USA, East USA, Europe and Asia. By locating the PlanetLab nodes in those areas we can effectively compare the effect on the performance when distributing content with the Amazon network infrastructure. The typical PlanetLab hardware is 4x Intel cores @ 2.4Ghz with 4GB RAM, well beyond our needs for this measures.

With those assets, we defined four scenarios by progressively locating the levels of the hierarchy in the closest possible location given by the cloud provider. Thus, in the Scenario 1, all the nodes are PlanetLab machines located in the country specified in the figure. In the Scenario 2, the level 1 node (the source node at Norway) is replaced by one node located in the European datacenter of Amazon. In the Scenario 3, the second level nodes USA-1, France and Taiwan are replaced by the corresponding Amazon nodes located in the USA, Europe and Asia datacenters respectively. Finally, Scenario 4 substitutes the level 3 nodes (Puerto-Rico, USA-2, Italy, Germany, Japan and China) by the the corresponding Amazon nodes located in USA, Europe and Asia datacenters.

The type of traffic used for the test aims to reproduce a real video streaming as seen in many internet services. As was mentioned above, RTP was used for transmission, and the transmitted video streaming was a test pattern of 640x480 pixels encoded in MPEG-4 with a bitrate of 500 Kbps. In order to discard outliers, three executions of each experiment scenario were run and average values were calculated. It should be noted that only Scenario 1 generated a high variability in the obtained results. Executions were run from Monday to Friday during business hours when Planetlab nodes were available.

C. Results

The figures 6, 7 and 8 show the values of jitter, bytes received and packet loss, observed by each node in the streaming distribution tree. The formerly described four scenarios are shown consecutively in X axis, and the nodes in each scenario are also grouped in X axis by geographical criterion.

The figure 6 shows the 99.7th percentile of jitter values obtained by running the experiment in the four scenarios. This figure indicates clearly that the jitter values decrease when

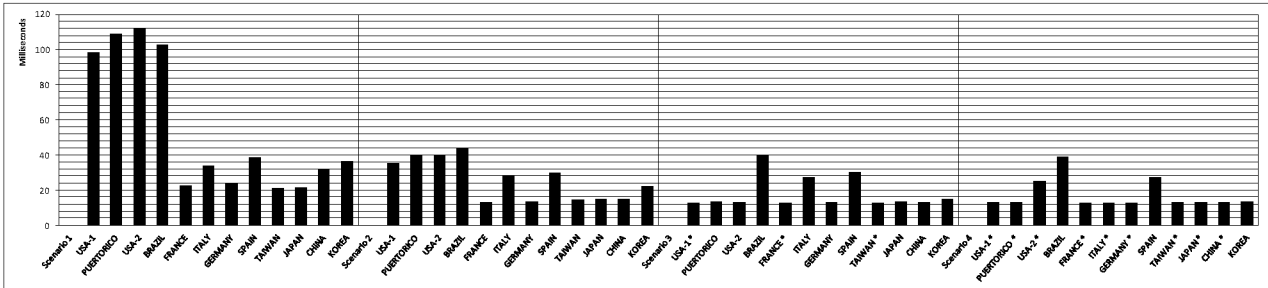


Fig. 6: The 99.7th percentile of jitter in each scenario

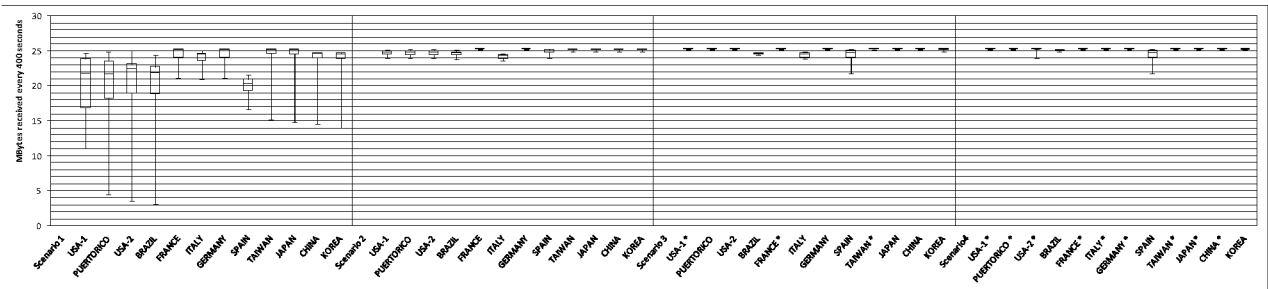


Fig. 7: Bytes received in each scenario

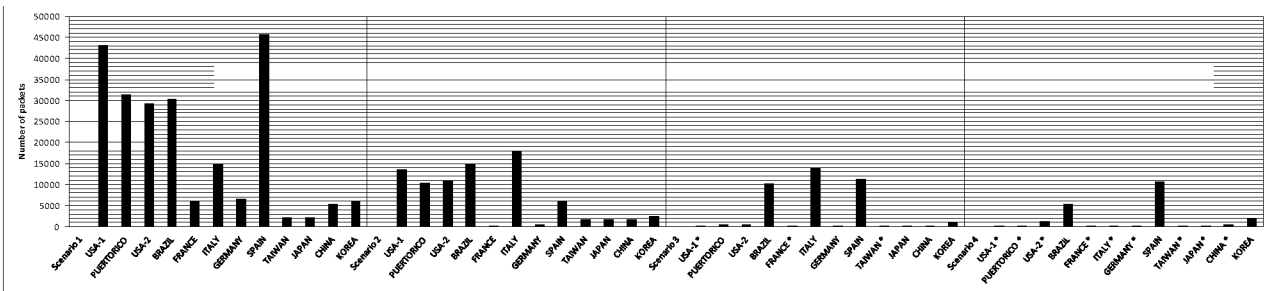


Fig. 8: Packet losses in each scenario

some levels of the distribution tree are moved into the cloud network. In Scenario 1, as expected, with all the communication links outside of the Cloud provider network, significant jitter problems appeared, mainly when the streaming session packets crossed intercontinental links from Europe (Norway) to America (USA-1), and so, the whole subtree (USA-1, Brazil, PuertoRico and USA-2 nodes) suffered from big jitter values. Even when the source node was moved inside the cloud provider (Scenario 2), the jitter values were still bad when the packets jumped outside the Cloud network from Europe to America. However, when the root nodes of each continental sub-tree (USA-1, France, and Taiwan) were moved into the cloud network (Scenarios 3 and 4) and the intercontinental jumps were fully deployed inside the Cloud network, the measured jitter values decreased significantly. Observing the obtained jitter values in Scenario 3 and 4, it is not expected to have severe jitter problems[19] even if a multi-conference service was deployed. Also, it must be observed that jitter

values did not improve significantly when the third level nodes of the distribution network were moved into the cloud. This is because the jitter problems appeared mainly when crossing intercontinental links from the source node to the second level nodes and so, enhancing the connectivity between second and third level nodes did not generate significant improvements in the obtained jitter values.

The figure 7 shows in *box plot* format the distribution of the streaming session throughput for each scenario and for each network node. The number of bytes received in each node is accumulated during periods of 400 seconds, and a summary of five values is shown in each *box plot*: minimum value, lower quartile, median, upper quartile and largest value. Again, Scenario 1 shows a high variability in the number of bytes received (mainly in American nodes). The rest of scenarios exhibit a nearly uniform throughput distribution, and so, we can hold that the throughput problems are mainly found in intercontinental jumps when only Internet links are

used. These problems can be nearly eliminated, moving the source node into the Cloud network (scenarios 2, 3 and 4). In these scenarios, some links of the intercontinental path that the session packets traverse, are internal transmission links of the Cloud network. These links exhibit a better throughput than the corresponding Internet external links that the PlanetLab nodes will use to jump between continents, and so, the overall throughput response of the intercontinental path, where the source is inside the cloud, will improve.

Finally, the figure 8 shows the number of packet losses that appeared during the transmission of the streaming session. We represent for each scenario the observed packet losses in each network node. Again, due to the better quality of the cloud provider network infrastructure, its transmission links generated less packet losses than the corresponding Internet links of PlanetLab nodes. In this way, the packet losses were accumulated for each link of the path of the streaming session. As this value was lower in cloud links, the more nodes are within the cloud, the less packet losses will occur. This situation can be observed clearly in Scenario 4, where only the nodes outside of the cloud (Brazil, Spain and Korea) accumulate a bigger number of packet losses than the other nodes that are inside the cloud in this scenario.

We can conclude in this section: (a) the jitter problems are nearly eliminated if the source and destination node of each intercontinental jump are located both in the cloud, (b) the throughput problems are alleviated moving into the cloud the source node of each intercontinental jump, and (c) the number of losses decreased as we were adding nodes to the cloud.

V. CONCLUSIONS

With the work in this paper we have made validation measurements that will be used as a first step towards realizing the architecture proposed in our previous paper [1]. First we have validated the benefits of a known Cloud provider infrastructure when used as a core of a P2P architecture instead of using normal peers that usually are behind commercial or limited Internet connections. An important conclusion is that the QoS of a streaming service can be efficiently improved as opposed to the traditional P2P and CDN systems by deploying a hybrid architecture of mixed P2P and Cloud streaming networks. This hybrid architecture is obtained placing strategically some distribution network nodes into the Cloud provider infrastructure, taking advantage of the reduced packet loss and low latency that exists among its datacenters. As result we have reduced the jitter in the end points, and we have improved the available bandwidth. We have also minimized the packet loss in each of the peers involved in the streaming scenarios. The differences that appear between Scenario 2 and 3 demonstrate that using connections between distant Cloud datacenters can help to improve the QoS response of streaming even in videoconferencing P2P systems. Therefore, using a Cloud network infrastructure to cross continents has improved the majority of the QoS problems. As for further research we prepare an extension of the experiments with more complex P2P topologies basing on this work. We will focus

on identifying more differences between scenarios 2 and 3 that we consider of a particular interest. Once confirmed the possibilities of establishing multimedia streaming applications on the described hybrid architecture, we will perform further analysis on an application level.

REFERENCES

- [1] I. Trajkovska, J. Salvachua Rodriguez, and A. Mozo Velasco, "A novel p2p and cloud computing hybrid architecture for multimedia streaming with qos cost functions," in *Proceedings of the international conference on Multimedia*, ser. MM '10. ACM, 2010, pp. 1227–1230.
- [2] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, and E. Turrini, "Qos-aware clouds," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 321–328.
- [3] S. Ferretti, V. Ghini, F. Panzieri, and E. Turrini, "Seamless support of multimedia distributed applications through a cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 548–549.
- [4] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: An overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 00–00, July 2003.
- [5] J. Weissman and S. Ramakrishnan, "Using proxies to accelerate cloud applications."
- [6] G. Wang and T. S. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," *2010 Proceedings IEEE INFOCOM*, pp. 1–9, Mar. 2010.
- [7] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10. ACM, 2010, pp. 1–14.
- [8] Y. Mei, L. Liu, X. Pu, and S. Sivathanu, "Performance measurements and analysis of network i/o applications in virtualized cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 59–66.
- [9] S. Sivathanu, L. Liu, M. Yiduo, and X. Pu, "Storage management in virtualized cloud environment," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 204–211.
- [10] S. L. Garfinkel, "An evaluation of amazon's grid computing services: Ec2, s3 and sqs," Center for, Tech. Rep., 2007.
- [11] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, ser. SC '08. Piscataway, NJ, USA: IEEE Press, 2008, pp. 50:1–50:12.
- [12] W. Chan, L. Mei, and Z. Zhang, "Modeling and testing of cloud applications," in *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, 2009, pp. 111–118.
- [13] R. Pereira, M. Azambuja, K. Breittman, and M. Endler, "An architecture for distributed high performance video processing in the cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 482–489.
- [14] P. Rodriguez, D. Gallego, J. Cervino, F. Escribano, J. Quemada, and J. Salvachúa, "Vaas: Videoconference as a service," in *CollaborateCom*, 2009, pp. 1–11.
- [15] B. Klepec and a. Kos, "Performance of VoIP applications in a simple differentiated services network architecture," *EUROCON'2001. International Conference on Trends in Communications. Technical Program, Proceedings (Cat. No. 01EX439)*, pp. 214–217, 2001.
- [16] K. Imran, M. Mellia, and M. Meo, "Measurements of Multicast Television over IP," *2007 15th IEEE Workshop on Local & Metropolitan Area Networks*, no. October, pp. 170–175, Jun. 2007.
- [17] A. McLellan, "White paper. Conflicting messages from the hint at growing resistance." *The Health service journal*, vol. 120, no. 6224, p. 3, Sep. 2010.
- [18] J. Quemada, T. de Miguel, S. Pavon, G. Huecas, T. Robles, J. Salvachua, D. Ortiz, V. Sirvent, F. Escribano, and J. Sedano, "Isabel: an application for real time collaboration with a flexible floor control," in *Collaborative Computing: Networking, Applications and Worksharing, 2005 International Conference on*, 0 2005.
- [19] G. Karlsson, "Asynchronous transfer of video," *Communications Magazine, IEEE*, vol. 34, no. 8, pp. 118–126, Aug. 1996.