

Original Research Paper

Methodology to Asses Internet Protocol Connection Availability: A Prerequisite For Feasible Video Stream Through Content Delivery Networks

Vitor Chaves de Oliveira, Gunnar Bedicks Junior and Cristiano Akamine

Electrical Engineering Faculty, Mackenzie Presbyterian University, São Paulo/SP, CEP 01302-907, Brazil

Received: 11-09-2014

Revised: 25-09-2014

Accepted: 01-10-2014

Corresponding Author:

Vitor Chaves de Oliveira

Electrical Engineering Faculty,
Mackenzie Presbyterian University,
São Paulo/SP, CEP 01302-907,
Brazil

Email: vitor_chaves_oliveira@yahoo.com.br

Abstract: Recent events have shown that content delivery networks as well as Broadband Providers are failing to provide a continuous service, especially on live video stream transmissions for numerous customers. This study aims to present a Methodology to uninterruptedly measure uplink-downlink of a given IP connection. Based on an open-source assemblage of development and data storage platforms it was programmed a software that automatically attends the proposed assessment. The significance of availability is effusively addressed on this article, since it is the first requirement regarding quality of service in any engineered communication. The proposed method relates to the fact that for a video and/or audio web stream to successfully occur, a connection with each end-user device needs to be sustained the entire time, establishing a complex two-way communication. Meanwhile, traditional cable and satellite broadcasts are less stressful one-way connections, demanding only that the end-user device to be placed within range of a radio signal. Given this scenario and adding the substantial demand increase for high quality media content from the internet, emerges an essential need to control the service delivered by CDNs and Broadband Providers. The developed software also creates a reasonable billing mechanism, which can function as a new technical milestone on contracts and/or Service Level Agreements. This tool also imposes a key function to the user's role, once it requires a setup of all seeked tests to be manually inserted, which might limit it for some routines.

Keywords: Control Methodology, Network Software, Media Stream, IP Connection Availability, Broadcast, Content Delivery Network

Introduction

The unceasing challenge to cost-effectively monitor computer networks has recently gained the spotlight after millions of paying customers faced, repetitively, an unavailable service (Sharma and FitzGerald, 2014; Greenfield, 2012; Chai, 2014) and given that the end-user devices do not natively measure and store network-Quality of Service (QoS)-data (Nunes, 2011; Oberortner *et al.*, 2012), these clients cannot objectively know how stable their connection are and

consequentially, are unable to assure the contract attendance of Broadband Providers and content delivery networks. Starting from this necessity, it was developed a software tool to address the availability issue. Such computer program permits a technical automated continuous user-tailored assessment which provides a report that demonstrates how available their IP connections behave over the time. Further than the motivation of protecting domestic and corporate technology consumer, it is revealed the inherent limitations for IP based broadcasting.

On a related matter, it is worth mentioning that emerging along with growth in media flow over the Internet (Ellacott, 2014; McMillan, 2013) is the intensification in entropy due to audio and video with higher definition (Cloonan and Allen, 2011; Shannon, 1948). In other words, the throughput (bits per second) demand is escalating because the streamed files are increasing in quantity and size. Although this progression may seem rapid, some specialists inform that it has only reached a fraction of what it could be in the next years. Such statement is acceptable once it's analyzed that in the last year the average american citizen watched, monthly, 6 h of video on the web and 152 h via traditional TV (Sharma and FitzGerald, 2014; Katel, 2014). In this way, with the video IP stream representing less than 4% of all viewed content and, if at this low level this kind of service demonstrated considerable unavailability (Sharma and FitzGerald, 2014; Greenfield, 2012; Chai, 2014), a great deal of concern on how can a user be protected from being wrongly billed for a poorly delivery connection is justified.

Quality of Service-QoS in IP Networks

In Computer networks, i.e., Internet Protocol networks, Quality of service is an assemble of technical conditions that permeate a qualitative operation for a given application which relies on an IP network to function (Oliveira *et al.*, 2013a; Baldini *et al.*, 2014). These requirements translate into specific QoS parameters in computer networks, such as throughput, jitter, latency and losses (Oliveira *et al.*, 2013b; Khanafer *et al.*, 2014). The first parameter, known as throughput, refers to the amount of data transmitted per time interval, measured in bits per second (bps) or Bytes per second (Bps). The latency, or delay, denotes the time used for the transmission of packets from the origin point (or node) reach the destination point (or node). While jitter corresponds to the time-variation of delay, that is, the difference of time expended by data packets to travel from transmitter to the receiver. As for the losses, it corresponds to the data detriment occurring in the transmission. It impairs the information of segments of bits. Losses can be represented by the bit error rate, or Bit Error Rate (BER), the packet error rate or Packet Error Rate (PER); or the error rate tables, or Frame Error Rate (FER) and that all these rates are measured in percentages (%) (Oliveira, 2012).

All these parameters relate to the concept of availability. Essentially, it precedes them, so that for any of these parameters to even be measured, first the

connection needs to available (Oliveira, 2012). Therefore, it's established that the availability concept ranks as the first concern for even basic quality of service standards. In this way, it becomes preminent to define and quantify it and through a mathematical calculation of probabilities is how this valuable statistic is estimated and only by real experimental measures it can be provided. Initially, it is necessary to understand that to estimate the availability, it's necessary to know that all the devices present in a network have, individually, a probability of failure. This failure depends on two main factors that form what is usually called downtime, the expected equipment failover and the repair time of technicians. The first is normally listed on the data-sheet of each equipment and the second is provided by the company that supports the operation with technical personnel. In this manner, the availability is considered to be the probability, in case of an estimation, that a service, or a device, is available during a period of time. Therefore, by subtracting from 1 (100%) the sum of the Average Expected Downtime of the Device (AEDD) with the Average Time to Repair the Device (ATRD) divided it by the Total Period of Time (TPT). The equation shown below Equation 1 describes the mathematical calculation employed to describe the Availability of a Single Network Equipment (ASNE) given in percentage (Tron, 2014):

$$ASNE(\%) = 1 - \frac{AEDD + ATRD}{TPT} \quad (1)$$

Equation 1 to estimate the availability of a single network device.

Keeping this in mind, it is now possible to consider the complete network availability and it is found by first computing the availability of each single device and them multiplying them from end to end. That is, multiplying the Availability of a Single Network Equipment 1 (ASNE-1: First device) by the Availability of a Single Network Equipment 2 (ASNE-2) and so on (until the Last Device), provides the Full Network Availability (FNA). Such equation is demonstrated as follows Equation 2 (Tron, 2014):

$$FNA(\%) = ASNE - 1 * ASNE - 2 * ... * ASNE - Last Device \quad (2)$$

Equaton 2 to estimate the full network availability.

Exploring both these statistical estimative calculus, it is implied that the communication path needs to be completely known to apply them. This means it's

required the knowledge of the ASNE about every server, switch, router, repeater, cable line. Between the origin and destination to calculate it. But, on further analysis regarding this issue, it is possible to imagine the emerged difficulty of it, because the connection is over the Internet. Besides, it's reminded that web connections are formed dynamically for each data packet channeled usually through Dijkstra's shortest-path algorithm or some derivation of it. Thus the availability estimation cannot be utilized. Particularly since the Transmission Control Protocol/Internet Protocol (TCP/IP) model, which computer networks are founded upon, do not ordinarily transmit any detailed information regarding what sort of equipment (brand, type) they routed data through and given the fact that to even implement such solution would be unfeasible due to costly increase in throughput (Tanenbaum and Wetherall, 2010).

Following this logic, it is acknowledged that a method to evaluate the availability of an Internet connection service requires, by definition, an experimental approach and only in a second moment generate, through mathematical modeling of previously collected data, statistical metrics to delimitate expectations for available service. Nevertheless, it is imperative to assume that such estimation would only be valid to each end user physical link. A proposed experimental arrangement to construct such estimation is achieved by a nonstop collecting and archiving of network monitoring. Along with it, a calculation is equally crucial to produce the pursued availability metric. This Experimentally Measured Full Network Availability (EMFNA) is indicated through the subtraction from 1 (100%) of the sum of the duration referring to All the Measured Failure Times (AMFT) divided by Total Period of Time (TPT), demonstrated as follows Equation 3:

$$EMFNA(\%) = 1 - \frac{AMFT}{TPT} \quad (3)$$

Equation 3 to provide the experimentally measured full network availability.

Another relevant aspect in an experimental setup to watch if a connection is available regards to the fact that the collected data should be time stamped. This additional information would assist on tracing certain periods of the day, week, month or year that the network service is more or less likely to be functioning properly. A reason that enhances the importance of the time stamp is the live stream transmissions from the CDNs, once it could be

scheduled at better availability times. Or, at least the customer could know the probability for his connection to be available in the programmed hours beforehand. Besides, the customer would also be able to discern the best periods, in terms of connection, to watch his on demand stream videos from the CDNs.

Content Delivery Networks Versus Broadband Providers

A Content Delivery Network (CDN) is an implementation of techniques designed to improve the efficiency in providing and distributing media content to end users over the Internet (Rouse, 2011; Octoshape and T-Mobile, 2014). Whereas a Broadband Provider (BP), or Internet Service Provider (ISP), is customarily a telecommunication company that retails bandwidth, degreed in Mbps (Megabits per second), which connects customers to the World Wide Web (ITU, 2003; Christensson, 2014). From these descriptions it is perceived the close dependence relation shared by CDNs, BPs and the availability issue.

CDN's techniques translate into hardware and software arrangement that sets numerous servers is dispersed locations that automatically replicates content based on users preferences acquired through artificial intelligence algorithms (Buyya *et al.*, 2008). The initial idea is to physically place computers that will form a shorter communication route, in number of nodes, with the consumers. In a second moment, these computers will storage, a copy of the content which users can, or are more likely to, access. Thus, the web pathway is reduced, increasing the network availability, because there are fewer devices composing the Full Network Availability (FNA) as seen in Equation 2. Appraising to this gain is the fact that with more sources to provide the service, one can act as cluster of another, the entire system becomes more reliable. A schematics presenting the CDN's mechanism is demonstrated below on Fig. 1. It is shown how the data is transmitted from the Origin Servers, then over to the Cache Servers and finally to its nearest End-users. These end-users are exemplified by different devices: Tablets, laptops, desktops and cell-phones. Completing this illustration is the role of the Internet Service Provider (ISP) which supplies these connections.

From this CDN strategy, two main characteristics are extracted: The focus on content access improvement and the reliance on Broadband Providers (ISPs) at several levels in the system. This latter mentioned, fundamentally circles back to the availability concern. Hence comprehending the BP's role is imperative.

A Broadband Provider, regularly, has a predetermined obligation to assure that a certain quantity of bandwidth is offered to the client across the time. Likewise, it is implied that for a connection service to have any sort of throughput compliance, it needs to be responding accordingly; that is, it must be available. However, a misguided contractual justification might be assumed disregarding the availability question and even display truthful information. Consider, for example, that the averaged flow-rate during a 100 min stream video transmission was 2.5 Mbps and that a BP contractual agreement that states a minimum bandwidth of 2.0 Mbps. Some could affirm that the service was fulfilling. But, now cogitate that the link was offline for 50 min, i.e., transmitting at 0.0 Mbps rate and that for the other 50 min it ran at a constant throughput of 5.0 Mbps. Now the problem of analysis based only on throughput terms arises. Because, in the forged example, once the customer could not view half the video an unsatisfactory service classification is clear, which goes completely against the previous analysis. The plotted chart below in Fig. 2 elucidates such situation, it express how an average is a poor data to be taken

into consideration. This throughput average calculation is presented on Equation 4 following below the chart.

From the graphic chart is possible to visualize the calculated average, for all the referred time, versus the 'real' Measured Throughput. The calculated average graphic line implies a good connection service during the whole one hundred minutes. Whereas, in reality, the stream only occurred for the first fifty minutes and it was disconnected for the second half of it.

Furthermore, it is vital to recollect that a CDN as well as a BP share objective responsibility in proving content for end-users. Since both need to be accessible for a successful network operation and the foremost important content delivery operation in IP networks nowadays is video stream (Technavio, 2014).

$$\text{Throughput Average (Mbps)} = \frac{\text{bits transferred (Megabits)}}{\text{Time (seconds)}} \quad (4)$$

Equation 4 to assess the throughput average in Megabits per second.

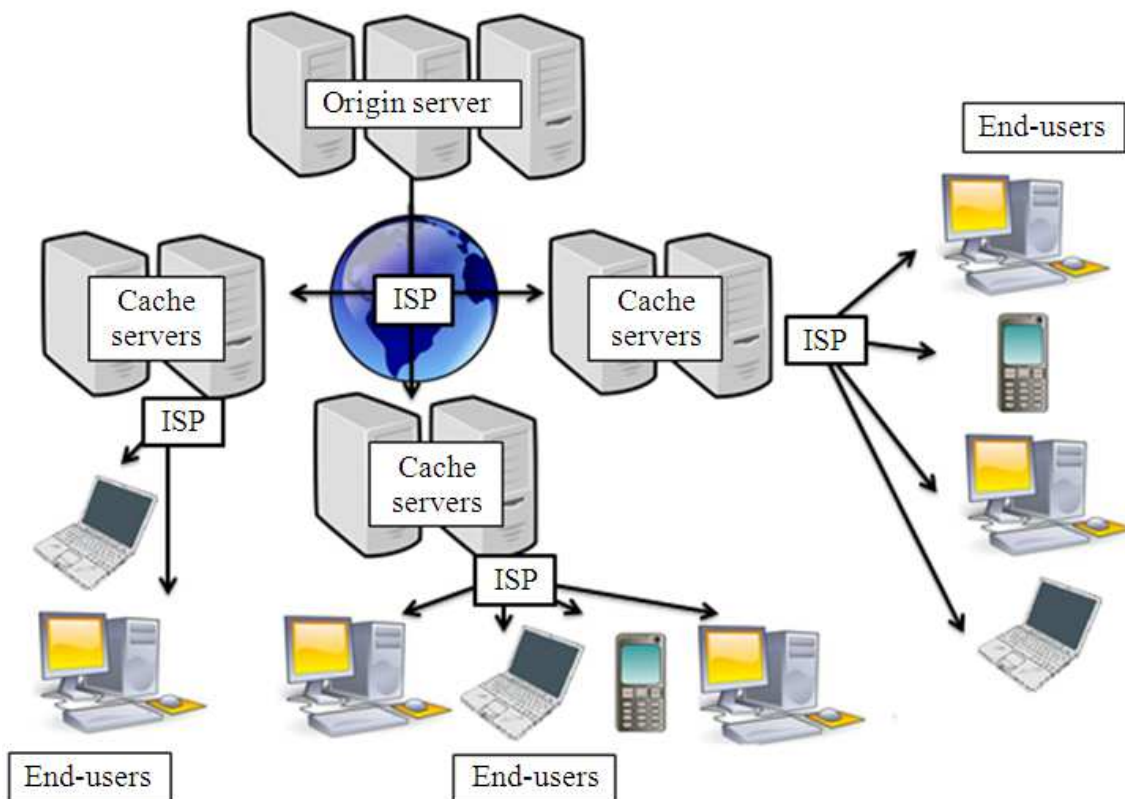


Fig. 1. Schematics of a content delivery network presenting the data flow through an Internet Service Provider (ISP) connection

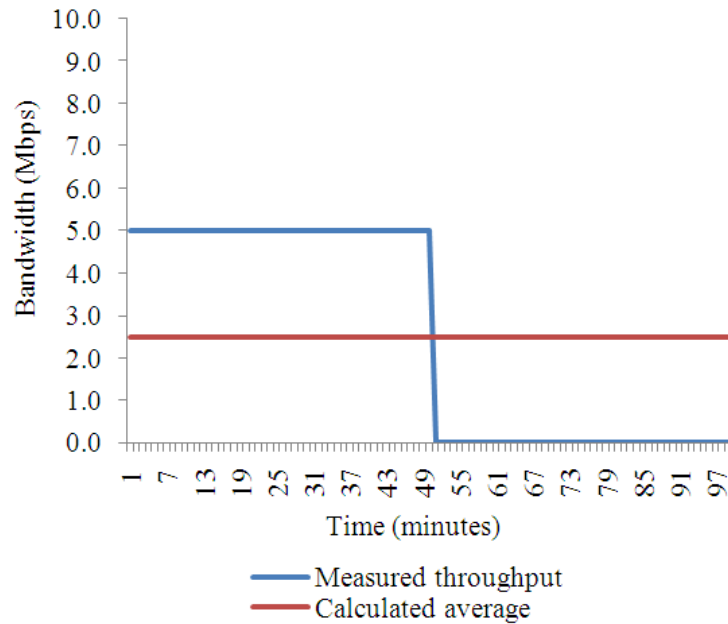


Fig. 2. Network connection evaluation of throughput in a stream video transmission

IP Stream: A Two-Way Connection

In any given computer-network-based media flow transmission a constant communication channel is continued throughout its stream. This relation can be classified as a two-way connection due to several control mechanisms natively attached to it. Ever since computerized communications were first invented, control over data ‘travelling’ on the network has been a necessity and as in time, the network grew more and more complex, it became inevitable (Tanenbaum and Wetherall, 2010).

The TCP/IP model is, by definition, a controlled network by handshakes verification. In general terms, handshakes are check messages sent to confirm successful reception from the destination to the source. Suppose a packet ‘data 1’ is sent to a certain endpoint, the sender waits a determined time for the receiver to send a packet ‘data 1 received send data 2’. In case this time is exceeded and/or the confirmation packet isn’t received, it will re-send ‘data 1’. This procedure assures consistency for the information, but increases throughput. This may does not bring significant negative effect if these packets are of a file, once it would just be transmitted over a best effort process. However, this control takes a major role once on-demand audio and video stream is required (Tanenbaum and Wetherall, 2010).

IP media streaming from the Internet, especially as it is today, is a multifaceted job. First lets picture the entire system’s progression when, as an

exemplification, a user access a website and starts a video exhibition. Once the click is perceived by the server that stores the content, a constant and sequential duplication of the file takes place. Such copy is temporarily held in the memory and then sent to the user’s address. Along with it, compressing and codification techniques are applied. It is noteworthy to highlight it is normal that the available bandwidth varies in time and that this could harm or even stop viewer’s experience. As a way to avoid such jamming, a constant monitoring of the transmission through QoS parameters measures and handshakes checks with the user’s equipment are applied. This commonly happens for two main reasons: Assure a sequential data flow and to adapt the bitrate to the link’s throughput. What it does, is to increase compression level (at the server side), reducing media quality, so that the video can still be watched when the user’s connection deteriorates (Tanenbaum and Wetherall, 2010; Octoshape and T-Mobile, 2014).

By this short description several reasons as to why it is correct to name IP Stream as a costly two-way connection are presented. Moreover, it is possible to understand how it’s more likely to show failure during events for numerous users in comparison to a traditional TV Broadcast. This latter does not need to duplicate content for each user, transmit packets to each customer address, monitor every single consumer connection, adapt its video quality or relay a collection of complex servers architecture closer to viewers as in a CDN strategy. In a few words, it is

valuable to point out that TV Broadcast is a simpler one-way communication. To watch it and have a good quality, users basically require a good Signal-to-Noise Ratio (SNR) and syntonize (tune) into the desired channel's frequency band. A satisfactory SNR means that the signal (wanted information) possesses a significative higher power than noise (unwanted signal or interference) and a frequency band is a portion of the spectrum. To illustrate this with a practical example, it is known that a digital TV channel in Brazil occupies a 6 Mhz frequency length and a satisfying SNR for it is about 22dB; keep in mind that Brazil's digital TV standard is the ISDB-T or 'Integrated Services Digital Broadcasting Terrestrial', this transmission tecnology presents a coverage of 46.8%, considering the country's inhabitants and should be present in the entire country before 2016 (Tanenbaum and Wetherall, 2010; Octoshape and T-Mobile, 2014; Stolfi, 2008; AAT, 2012).

Since the World Wide Web video stream presents serious risks, consumer awareness of the issue in escalating and with it, emerges a necessity to protect themselves from wrongful billing and an answer to this is provided by monitoring the ISPs and CDNs delivered connectivity. This translates into uninterrupted availability verifications. In the subsequent sections, it is described a software tool developed to attend this objective.

Materials and Methods

In order to perform the evaluation at hand, a computer programmed solution was designed and it employed various software-based technologies. Commencing the tool's development, it was listed the objectives it should feature: Use of only open source programs, Web integrated environment, continuous and time stamped log of tests, operational in multiplatform end-user hardware and user controlled and customizable evaluations. All these characteristics were achieved by employing the following materials and methods described.

The software is centered in java programming language applied over the open source 'spring tool suite 3.4.0 release' (PS, 2014). This suite is clutched to the Eclipse Integrated Development Environment (IDE) that offered a consistent set of appropriate instruments for the intended project. In this suite takes place what can be named as the program's core codification and most relevant fragments of such code are outlined in this section. Beyond coding, it is pertinent to indicate the connections with other apparatuses. Two of these were functional associations with a Relational Database Management

System (RDBMS) and a web server. The RDBMS, program utilized to store the performed evaluation's reports, was the 'MySQL 6.0 community edition' (OC, 2014). It is a database system that uses the distinguished 'Structured Query Language (SQL)' for managing information. While the chosen Web Server, working as a pathway for communicating with the internet and assisting to deliver a browser interface was the 'Apache Tomcat 7.0.50' (Tomcat, 2014). This designated trio operates in synchronized coordination and to efficiently comprehend this, it's shown below a step by step explanation of the finalized software through foreground, i.e., user interface and background, i.e., coded algorithms and tables.

The initial interface is present in Fig. 3, it is accessible via any web browser, after the software is installed and the Apache Tomcat Server is fixed as automatic background local service. Then, by entering "http://localhost:8080/ping" or "http://127.0.01:8080/ping" in the address slot. The 'localhost' or '127.0.0.1' indicates it is a local service physically installed at the user's machine. The '8080' is a well-known TCP port for Web Server (Tomcat, 2014) and 'ping' is the address alias specified to the software and also a reminder to the network utility employed in this application (Microsoft, 2014). Also, for the referred example in Fig. 3, it's using 'Mozilla Firefox 29.0' (Mozilla, 2014).

Additionally, the below displayed page offers ten Uniform Resource Locators URLs selections to be verified over several possible time allocations as well as choosing the amount of desired tests. These different URLs options allow testing of either an Internet Protocol (IP) number or a Domain Name System (DNS) address and in computer networks, both IP and DNS respond for the location of a sought resource, that is, data (Tanenbaum and Wetherall, 2010) and noted that, on the World Wide Web, data is typically held at a website, consistently and continuously testing if a URL is available acquires an enhanced implication. Especially when that most accessed websites in the Internet offer media content, i.e., images, music and/or video (Technavio, 2014).

The exemplified application's initial page (Fig. 3) brings all the optional fields already filled with a range of user selected, that is, written, parameters. These options start at the top of the screen across URLs 1 through 10. Then, it is quantified the amount of times this series of addresses will be tested in the 'Number of Repetitions' field. Subsequently lower, there are two selectable temporizations. The first is the time among each complete series of checks, in milliseconds (ms), labeled as 'Interval between sets of

tests (ms)'. The second refers to the milliseconds to be delayed after every single URL examination, entitled 'Interval between each ping (ms)'. After that, a 'Save' button has the function to inscribe the investigations and activate testing. It also provides a confirmation message confirming if the request is effectively queued. Finally, right below is located another button, acting as a link to the program next page. It is colored in blue letters and regarded as 'View Ping Tests and Reports'. Such page is revealed by Fig. 4 and encloses the analysis results sent to be processed on the initial page.

The application screen shot illustrated in Fig. 4 exhibits an informational table that fully reports each tests set state. So, as soon as the save button on Fig. 3 is clicked what is configured in the URLs fields assembled with the other specifications are sent to begin running according to a queue line. It is a priority list provided based on time frame sequencing. This way, the first saved set sent receives an ID numbered as 1 and the next sets get an ID on $n + 1$ basis, i.e., last ID number +1, a mathematical formula ran for each new web addresses series to be checked. This characteristic is on the first column of the first table, designated at the left upper side as ID, where it is readable from bottom to top the eldest to the earliest. Also, a clock imprint of this queuing procedure is captured and placed on the sixth column of the upper table, from left to right. It is named as date/time, describing a day/month/year and a hours: Minutes: Seconds time indication. Continuing on the same table at the top of the next image, there are several columns: ID, Status, Repetitions, Interval between sets of tests (ms), Interval between each ping (ms), date/time and actions. All are previously detailed except status and actions.

About the 'Status' lines, it is valid to explain that it indicates the state of the inscribed tests showed in the interface on Fig. 3. In this manner, it informs if a queued request is on one out of five possible conditions: Processed successfully, aborted, aborting, processing or waiting to start processing. The 'processed successfully' means the evaluations finished without any setback. 'Aborted' is shown when the user terminated the checks before its completion. 'Aborting' is the same but it is displayed while the termination is in course at the moment. The 'Processing' tells that the ping investigations are ongoing, that is, currently occurring in the user's device. To conclude, the 'Waiting to Start Processing' is a prior status from the last one, it shows that the selected checks are on hold, expecting to begin running once others finish or are aborted. Meanwhile, the 'Actions' column features up to two possible link options: List and stop. The 'List' link reveals the

bottom table in Fig. 4, which details the tests results as they occur. The 'Stop' is utilized to cease a particular ongoing assessment, because once clicked, causes the testing's cancellation and, promptly after activated, it is demonstrated an aborting and/or aborted status for it. In addition, there is another link in blue letters to be observed, titled: 'Schedule Tests'. It is a navigational connection to the program's first page (Fig. 3), allowing the user to include more sets of tests, even while another is running.

The contents in the lower table of Fig. 4 are divided in four columns to display the availability checks for every IP or DNS. Beginning on the left to the right side, it is seen 'Repetition N^o', 'URL', 'success' and 'date/time'. The first marks the tests attempt number on each URL. Subsequently to it is the DNS or IP, also branded as the Uniform Resource Locator. Then, a 'Success' column is present, informing if the URL was available or not at that instant for that the specific repetition and such result is given by a Yes or No answer. Right next to this is the time stamp for each examination outcome, it carries the same pattern as the above 'date/time'.

To complete the materials and methods employed in this solution relating the software's central programming is indispensable and since it is acknowledged that the ping procedure is the program's center, its commented encoding as it is typed inside the 'spring tool suite' is in Fig. 5. In it, is viewed a java code containing the designed routine for separately pinging URLs and capturing its outcomes. Specifically, a couple tailored parameters that accompany the network utility tool (ping) are noted. These are the '-n 1' and '-w 2000', representing respectively, the total of one echo requests packets sent and a 2000 milliseconds waiting time for its reply. After the ping occurs on the URL the system captures a true or false attribute that is stored on a binary (1 or 0) pattern in the database, which will be later showed as a Yes or No in the success column (Fig. 3) on the user's exploration page. Besides this, there is another setting that inserts an extra timeslot of 3000 milliseconds nest to the ping's occurrence. This time window is included as a failsafe measure, because with it the system has enough time to process the pings, store the data in MySQL table and placed it on the user's interface. This avoids the requests to pile up and/or jam the software, even when the machine is under heavy use or when the user requires back to back URLs checks with zero time configured in the 'Interval between each ping (ms)' field. Consequently, it also assures that the software won't occupy a significative portion of hardware processing time on the device it is installed. Also guaranteed due to this, is low network bandwidth consumption.

As the ping tests are led to run, two key tables inside a MySQL database system grow populated with its requests and results. These are called 'queue' and 'ping-log' and they are typified by Fig. 6 and 7. The 'queue' table, considered in Fig. 6, inscribes the demanded availability inquiry as soon as the 'Save' button (Fig. 3) is hit. So, major parts of the requests detailed information are carried into a row in this table. That is, both temporizations ('interval-milisecs' and 'interval-ping-milisecs') and the repetitions amount ('repeat-num') are stored in it. Besides this, it holds the moment ('create-timestamp') the requests were sent as well the current state of handling ('status'). Such distinction reflects to the pre-viewed 'Status' column in the Exploration Interface Page (Fig. 4). Also, it is just as pertinent to point out the

attribution of a 'queue-id', which numerically organizes the order of events per row, i.e., queued request.

As for the other emphasized table, named 'ping-log', its most noticeable role is to continuously record the tests results. In Fig. 7 the ping responses are corroborated through the 'successful' column, if it displays 1 means the Uniform Resource Locator was available at that exact moment corresponded via the 'create-timestamp' column and if it is 0, this is an unavailability indication. It also keeps the attempt number ('repeat-id') related to it, the URL ('url') checked and the requested queue's identification ('queue-id'). To conclude, in the upper left of this table there is a 'ping-log-id' providing an arithmetical sequencing beginning in 1, at the start and systematically adding 1 to the next examination row.

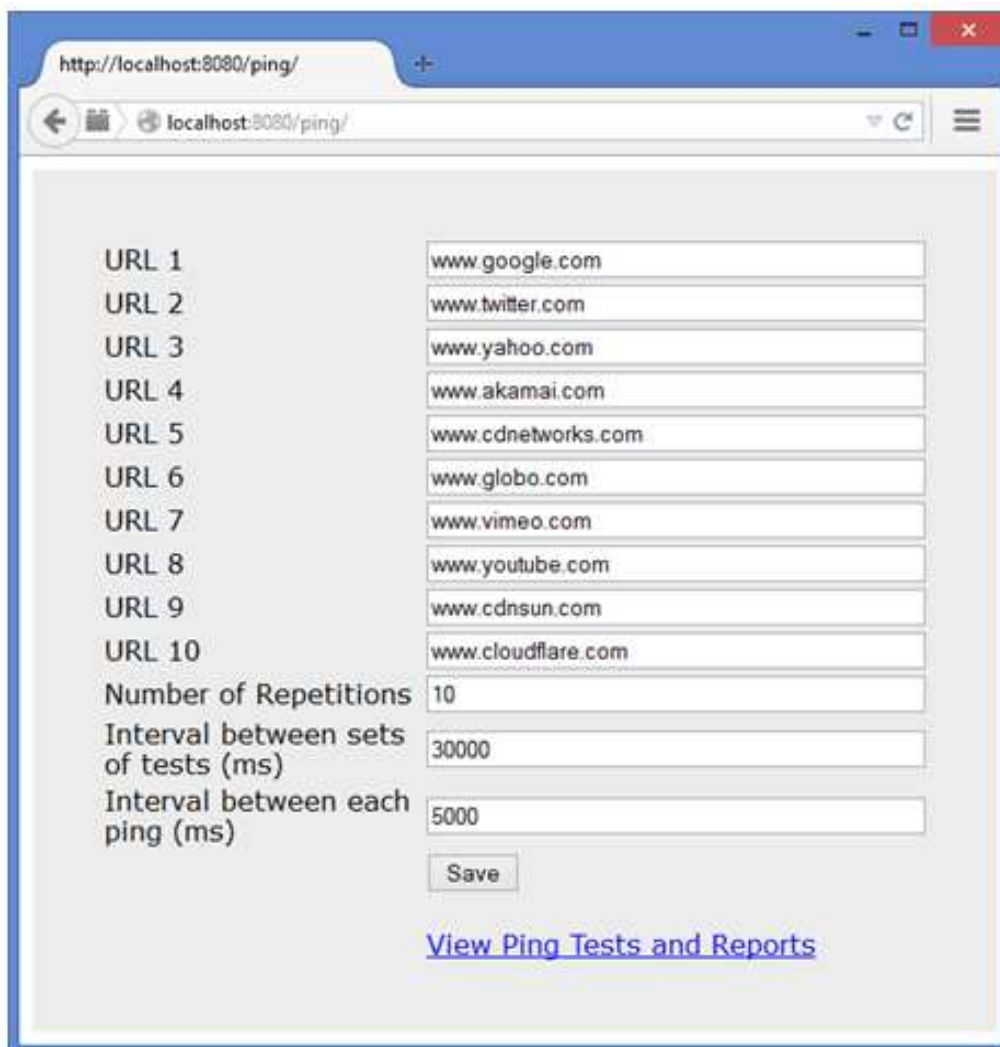


Fig. 3. Availability software initial page filled with websites URLs examples and other configurations via Firefox browser

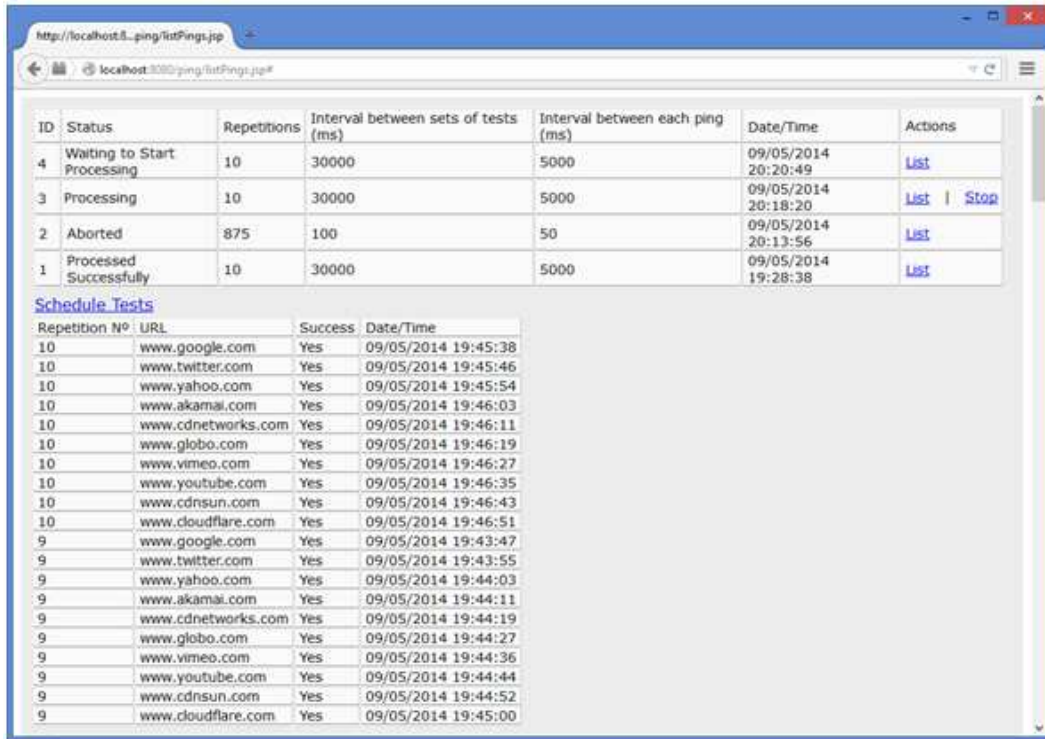


Fig. 4. Availability software exploration page with URLs sample checks results via Firefox browser

```
// Method to ping URLs and capture the results
public static boolean runPing(String ipstr) {
    boolean retv = false;
    try {
        InputStream ins = Runtime.getRuntime()
// Invokes the execution of ping against a URL address (ipstr)
//sending 1 echo request( -n 1) and waiting 2000 milliseconds(-w 2000) for its response
        .exec("ping -n 1 -w 2000 " + ipstr).getInputStream();
// Makes the program wait (sleep) for 3000 milliseconds
        Thread.sleep(3000);
//Captures the response and saves the true or false response
        byte[] prsbuf = new byte[ins.available()];
        ins.read(prsbuf);
        String parsstr = new StringTokenizer(new String(prsbuf), "%")
            .nextToken().trim();
        if (!parsstr.endsWith("100") && !parsstr.endsWith("again."))
            retv = true;
    } catch (Exception e) {
        retv = false;
    }
}
// Returns the result of the URL's ping
//that will later be shown as a Yes or No information in the Success column
return retv;
}
```

Fig. 5. Availability software java encoding on spring tool suite-method to ping URLs and capture the results

queue_id	status	repeat_num	interval_milisecs	interval_ping_milisecs	create_timestamp
1	PROCESSED	10	30000	5000	09/05/2014 19:28:38
2	ABORTED	875	100	50	09/05/2014 20:13:56
3	PROCESSING	10	30000	5000	09/05/2014 20:18:20
4	PENDING	10	30000	5000	09/05/2014 20:20:49

Fig. 6. Availability 'queue' table, adapted screenshot from MySQL workbench

ping_log_id	queue_id	repeat_id	url	successful	create_timestamp
1	1	1	www.google.com	1	09/05/2014 19:29:00
2	1	1	www.twitter.com	1	09/05/2014 19:29:08
3	1	1	www.yahoo.com	1	09/05/2014 19:29:16
4	1	1	www.akamai.com	1	09/05/2014 19:29:24
5	1	1	www.cdnetworks.com	1	09/05/2014 19:29:32
6	1	1	www.globo.com	1	09/05/2014 19:29:40
7	1	1	www.vimeo.com	1	09/05/2014 19:29:48
8	1	1	www.youtube.com	1	09/05/2014 19:29:56
9	1	1	www.cdnsun.com	1	09/05/2014 19:30:04
10	1	1	www.cloudflare.com	1	09/05/2014 19:30:12
11	1	2	www.google.com	1	09/05/2014 19:30:51
12	1	2	www.twitter.com	1	09/05/2014 19:30:59
13	1	2	www.yahoo.com	1	09/05/2014 19:31:07
14	1	2	www.akamai.com	1	09/05/2014 19:31:15
15	1	2	www.cdnetworks.com	1	09/05/2014 19:31:23
16	1	2	www.globo.com	1	09/05/2014 19:31:31
17	1	2	www.vimeo.com	1	09/05/2014 19:31:39
18	1	2	www.youtube.com	1	09/05/2014 19:31:47

Fig. 7. ‘Ping-log’ table, adapted screenshot from MySQL workbench

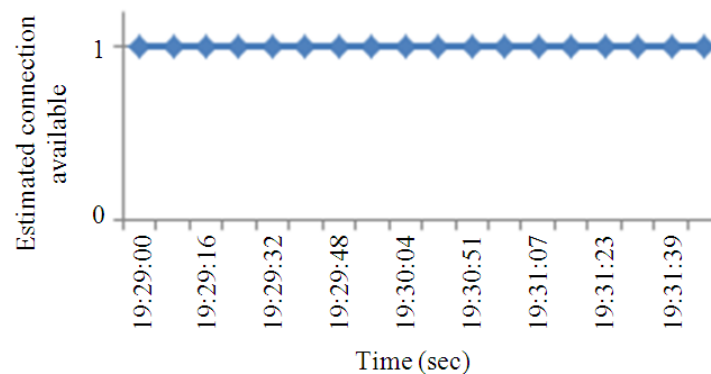


Fig. 8. Network availability graph from ping-log table

Results

Through the data presented in both tables, it is conceivable to generate an availability timeline graph, similar to a Network Connection Evaluation (Fig. 2), as well as to compute an Experimentally Measured Full Network Availability (EMFNA, Equation 3). In this manner, the sought results of the software application appear, that is, the methodology to assess IP connection availability is reached. By plotting the ‘ping-log’ table it is clear to visualize how available the device’s internet connection was during a short two minutes period. So, in Fig. 8, starting at 09/May/2014 19 h 29 m 00 s and lasting until 19 h 31 m 47 s on the same day, all the 18 measures taken via ping checks are emphasized and established over the time, or 167 seconds. In it, is seen a 100% positive availability, zero unavailable seconds, for the assumed

period, which is also calculated by the EMFNA’s formula. The Equation 5 defines such computation for the considered testing:

$$\text{ping_logEMFNA}(\%) = 1 - \frac{0 \text{ seconds}}{167 \text{ seconds}} = 100\% \quad (5)$$

Equation 5 Calculating ping-log’s availability with EMFNA.

The following graph confirms this calculation and displays the two last columns at the right of the ‘ping-log’ table, ‘successful’ and ‘create-timestamp’. This is the reason for the time axis beginning at 19 h 29 m 00 s and also why the estimated connection available axis is on a one or zero basis, i.e., available or unavailable connection.

It is crucial to observe that these results surpasses a general network availability assessment, allowing

the same data analysis to be performed separately or combined, that is, for every URL or for a set of it. Thus the availability responsibility is dividable by CDN provider for instance. However, if most or all URLs under verification start to convey into unsuccessful connection reports and if they are from different providers, like the ones in the exposed table, it is reasonable to consider it is the broadband Providers (ISP) fault.

Ultimately, behind a full availability register, stands the simultaneous creation of an end-user billing mechanism that can only reduce the ISP's or CDN's invoice. Suppose a user sets the software to catalog the network availability information for a month and once the month finishes it computes the consumer's EMFNA result is X%. In case he pays \$Y monthly for a BP's service, he could claim to pay only a value equals to the multiplication of \$Y times X%, as a way to get compensation for the non-delivered connection or request for a non-chargeable connection period equivalent to the time he was left offline. Imagine X is 85%, Y is \$100 and the month was 30 days long. The client could either ask for a \$15 (\$100-85) discount, paying just \$85 (\$100*85%), or request for 4.5 days (30 days*15%) free of charge connection period. For a corporate customer, a certain period could be more significant than another, like business hours. In such contract, hours between 09 to 17 h could have a weight of 75% as the hours from 17 to 09 h corresponded to a 25% weight in a weighted average and this could certainly be reflected in the analysis captured through this software, since all checks are automatically time stamped.

Discussion and Conclusion

Given that the current scenario of global computer networks presents a questionable attendance to consumers increasing demands and as content continuously migrates to IP, its inherent limitations tend to decrease connection availability. Consequently, ways to detect these failures are essential, especially when the monitoring need to be distributed, continuous, stand modifications along the time, consume a minimum of hardware processing, require low network impact and run on different end-user devices.

The proposed methodology presented a viable solution to face these challenges, producing a real-time network assessment associated with a tailoring strategy that is user-managed even during tests. Via this routine, the uplink-downlink interval delivered by a CDN or a BP is exhaustively examined. This allows clients to technically prove if a resource was available

or not, providing an effective contract control over these paid services. This billing at user side is a main achievement of this application, bringing financial advantages and consumer protection from poor QoS. It is imperative to remind that the application was developed entirely on open-source architectures, i.e., without any proprietary costs.

It is also necessary to point out that this tool due to the enhanced user's role in it, requiring a setup of all needed tests to be manually inserted, could be considered to be non-practical or limited to some pre-established environments. The improvement of this constraint could be a new point of study for future works in the field. Perhaps building a hybrid approach, in which the user chooses between a manual and/or an automated testing.

Another point to this study was to reveal the difficult job that is to transfer services that are traditionally broadcasted by TV systems into the web stream. Exposing the fact that a packet based network, such as the Internet, requires an excessive technical effort to take a TV's role, especially when live events for numerous viewers are intended. Thus, questioning if it is worth the monumental investments to make live web transmissions possible. But, as it seems to be an unchangeable future, users need to be alert and demand service as promised in their contracts and in case it is not delivered in the correct fashion, claim their due restitutions through cost-effective technical measuring.

Funding Information

The authors have no support or funding to report.

Author's Contributions

All authors equally contributed in this work.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- AAT, 2012. Coverage of Digital TV on Brazil in Numbers (in portuguese). National Telecommunications Agency. Anatel apud Teleco.
- Baldini, G., S. Karanasios, D. Allen and F. Vergari, 2014. Survey of wireless communication technologies for public safety. IEEE Commun. Surveys Tutorials, 16: 619-641. DOI: 10.1109/SURV.2013.082713.00034

- Buyya, R., M. Pathan and A. Vakali, 2008. Content delivery networks. 1st Edn., Springer Science and Business Media, Springer-Verlag, Berlin, ISBN-10: 354077887X, pp: 434.
- Chai, B., 2014. 'True Detective' Finale Crashes HBO Go. Wall Street J.
- Christensson, P., 2014. ISP Definition. Tech Terms Computer Dictionary.
- Cloonan, T. and J. Allen, 2011. Competitive analysis of adaptive video streaming implementations. ARRIS.
- Ellacott, J., 2014. Video streaming driving impressive growth in video content delivery market: TechNavio trends. Infiniti Research Limited.
- Greenfield, R., 2012. NBC Claims Super Bowl Streaming Victory, But Viewers Cry 'Fail'. The Wire. The Atlantic Monthly Group.
- TU, 2003. The Birth of broadband-media fact sheet. International Telecommunication Union (ITU).
- Katel, P., 2014. Future of TV: Can Traditional Television Thrive in the Internet Age. CQ Press, Washington DC, pp: 24.
- Khanafar, M., M. Guennoun and H. Mouftah, 2014. A survey of beacon-enabled IEEE 802.15.4 MAC protocols in wireless sensor networks. IEEE Commun. Surveys Tutorials, 16: 856-876. DOI: 10.1109/SURV.2013.112613.00094
- McMillan, G., 2013. Viewers are flocking to streaming video content-and so are advertisers. Wired.
- Microsoft, 2014. Microsoft TechNet-PING. Microsoft Inc.
- Mozilla, 2014. Mozilla Firefox. Mozilla.
- Nunes, R.J.F., 2011. Web portal for continuous internet access quality of service measurement. Masters dissertation. Technical Superior Institute. Technical University of Lisboa.
- Oberortner, E., S. Sobernig, U. Zdun and S. Dustdar, 2012. Monitoring performance-related QoS properties in service-oriented systems: A pattern-based architectural decision model. Proceedings of the Proceedings of the 16th European Conference on Pattern Languages of Programs, Jul. 13-17, Irsee, Germany, ACM New York. DOI: 10.1145/2396716.2396729
- OC, 2014. MySQL-The world's most popular open source database. ORACLE. Oracle Corporation.
- Octoshape and T-Mobile, 2014. Transitioning from broadcast TV to broadband TV with livestream perform. Octoshape Inc and T-Mobile Inc.
- Oliveira, V., 2012. Impacts of the variation of applications and of local hardware on the flow rate of computer networks with desktop virtualization (in portuguese). Masters dissertation. Electrical Engineering Faculty. Pontifical Catholic University of Campinas.
- Oliveira, V., A. Mota and L. Mota, 2013a. Impacts of application usage and local hardware on the throughput of computer networks with desktop virtualization. Am. J. Applied Sci., 10: 117-122. DOI: 10.3844/ajassp.2013.117.122
- Oliveira, V., I. Yano, A. Mota and L. Mota, 2013b. Feasibility of desktop virtualization per software services and local hardware based on the network throughput. J. Comput. Sci., 9: 827-837. DOI: 10.3844/jcssp.2013.827.837
- PS, 2014. Spring Tool Suite. SPRING. Pivotal Software.
- Rouse, M., 2011. Definition: Content Delivery Network (CDN). TechTarget. Search AWS.
- Shannon, C., 1948. A mathematical theory of communication. Bell Syst. Technical J., 27: 623-656. DOI: 10.1002/j.1538-7305.1948.tb00917.x
- Sharma, A. and D. FitzGerald, 2014. ABC's oscars streaming outage shows web limitations for TV networks. Traffic overload during academy awards tested live-streaming capabilities. Wall Street J.
- Stolfi, G., 2008. Temporal interleaving through Fourier transform-a robust method for transmission of digital TV signals (in portuguese). Thesis Ph.D., Electrical Engineering Faculty. University of São Paulo.
- Tanenbaum, A.S. and D. Wetherall, 2010. Computer Networks. 5th Edn., Pearson Prentice Hall, Upper Saddle River, ISBN-10: 0132126958, pp: 933.
- Technavio, 2014. Global video CDN market 2014-2018. Technavio. Infiniti Research Limited.
- Tomcat, 2014. Apache Tomcat. Apache Software Foundation.
- Tron, N., 2014. White paper: Network Availability. N-TRON Corporation: The industrial network company.