Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Using latency as a QoS indicator for global cloud computing services

Pedersen, Jens Myrup; Riaz, Tahir; Dubalski, Bozydar ; Ledzinski, Damian ; Celestino Júnior, Joaquim ; Patel, Ahmed

Published in:
Concurrency and Computation: Practice & Experience

DOI (link to publication from Publisher):
10.1002/cpe.3081

Publication date:
2013

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Pedersen, J. M., Riaz, T., Dubalski, B., Ledzinski, D., Celestino Júnior, J., & Patel, A. (2013). Using latency as a QoS indicator for global cloud computing services. *Concurrency and Computation: Practice & Experience*, *25*(18), 2488-2500. https://doi.org/10.1002/cpe.3081

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: April 25, 2024

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2013; **25**:2488–2500
Published online 27 June 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3081

SPECIAL ISSUE PAPER

Using latency as a QoS indicator for global cloud computing services

Jens Myrup Pedersen[1,*], M. Tahir Riaz[1], Bozydar Dubalski[2], Damian Ledzinski[2], Joaquim Celestino Júnior[3] and Ahmed Patel[4]

[1]*Department of Electronic Systems, Aalborg University, Aalborg, Denmark*
[2]*Institute of Telecommunications, University of Technology and Life Sciences in Bydgoszcz, Bydgoszcz, Poland*
[3]*Computer Networks and Security Laboratory, Universidade Estadual do Ceará, Fortaleza, Ceará, Brazil*
[4]*Center for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia*

ABSTRACT

Many globally distributed cloud computing (CC) applications and services running over the Internet, between globally dispersed clients and servers, will require certain levels of QoS in order to deliver and give a sufficiently smooth user experience. This would be essential for real-time streaming multimedia applications such as online gaming and watching movies on a pay as you use basis hosted in a CC environment. However, guaranteeing or even predicting QoS in global and diverse networks that are supporting complex hosting of application services is a very challenging issue that needs a stepwise refinement approach to be solved as the technology of CC matures. In this paper, we investigate if latency in terms of simple ping measurements can be used as an indicator for other QoS parameters such as jitter and throughput. The experiments were carried out on a global scale, between servers placed in universities in Denmark, Poland, Brazil, and Malaysia. The results show the correlation between latency and throughput, and between latency and jitter, even though the results are not completely consistent. As a side result, we were able to monitor the changes in QoS parameters during a number of 24-hour periods. This is also a first step toward defining QoS parameters to be included in service level agreements for CC at the global scale in the foreseeable future. Concurrency and Computation: Practice and Experience, 2013.© 2013 Wiley Periodicals, Inc.

Received 31 May 2013; Accepted 4 June 2013

KEY WORDS: cloud computing; ICT infrastructure; QoS; service level agreements

1. INTRODUCTION

Cloud computing (CC) technology and its supporting services are currently regarded as an important trend toward future distributed and pervasive computing services offered over the global Internet. Several architectures exist for CC, which differ in what kind of computing services are offered, culminating with the advances with Web 4.0 and mobile technologies [1]. Detailed studies of different approaches to CC can be found in [2][3]. To keep it simple, CC can be divided into two domains. The first domain consists of resources for computations and applications access, and their use by users—seen as traditional client server model. The second domain consists of networks or more specifically the Internet, which enables computation for accessing and sharing computation

*Correspondence to: Jens Myrup Pedersen, Department of Electronic Systems, Aalborg University, Aalborg, Denmark.
†E-mail: jens@es.aau.dk

Copyright © 2013 John Wiley & Sons, Ltd.

and data resources for servers and clients in an array of complex arrangements [4]. Although the word 'cloud' is a metaphor for Internet, based on depictions in computer network diagrams to abstract the complex infrastructure it conceals [5], CC is a general term for anything that involves delivering hosted services such as Infrastructure as a Service, Platform as a Service, and Software as a Service [6]. Key factors of CC growth are increase in computation power and broadband network access. Today, the use of CC is growing at an exponential rate [7]. Few well-known examples of these services are remotely distributed and centralized data storage, remote offices such as Google Docs, Office Live, cloud gaming services, virtual desktops, multimedia streaming, and grid computing [6] [8]. It is widely predicted that Web 4.0 will create a myriad of possibilities for developing new applications within clouds needing QoS [1].

In order for clients to smoothly connect to services offered from servers and/or other clients located all over the globe, a specified QoS is often required, sometimes expressed in service level agreements (SLAs). If these servers and clients are just connected through the Internet, it can be challenging to obtain a consistent service: the traffic is dynamically routed through different providers, from end-user access at the edge through distribution networks to national, international, and even transcontinental backbones. This also makes it difficult to impossible to provide guarantees or even predictions of QoS because most often we do not have insight into the provider's networks and routes of global connections, which are likely to change dynamically and continuously. Known behaviors, such as temporal changes in traffic amounts, may also be different in the different networks, making it difficult to come up with a simple prediction model. Moreover, different kinds of traffic may be prioritized on the basis of, for example, packet sizes, protocols, and source/destination addresses, adding to the complexity of modeling and predicting behaviors or even continuously monitoring changes in QoS in a simple manner. Working in this uncontrollable environment also makes it hard to apply existing QoS techniques that focus on providing guarantees based on admission control [9][10].

In general, existing methods for measuring end-to-end QoS can be divided into two main classes: active monitoring, where the measurements are based on traffic/packets/probes injected into the system, and passive monitoring, where measurements are based on observing existing traffic and studying, for example, throughput and response times for receiving acknowledgments (ACKs). The advantage of passive monitoring is that it does not add any communication overhead. It makes good sense for some applications, for example, video streaming [11], where a continuous flow and large amount of packets make it possible to observe the parameters that are important for the application in some feedback loop. For other applications, this is more difficult, for example, multimedia interactive applications (duplex) with burstier traffic patterns and various periods of one-way, two-way, and idle communications. This paper investigates if active monitoring based on Internet ping packets can be used as an indicator for the most important QoS parameters. This takes advantage of the active network monitoring approach while keeping the overhead to a minimum.

The commonly used QoS parameters include delay/latency, jitter, packet loss, and bandwidth. Different applications have different QoS requirements [13]: whereas some applications are tolerant to all parameters, and will do fine with whatever best-effort service is available (for example, simple file transfer, web surfing, and email), others will be critical with respect to one or two parameters (voice and video over internet protocol, game streaming), and others will be demanding in terms of more parameters (remote apps, database access, remote controlling, etc.). This criticalness will depend on the nature of the application, but as CC matures as a technology, more time-critical and safety-critical applications are expected to be developed, such as smart grids, Supervisory Control and Data Acquisition networks, telehealth, teleoperation, and remote monitoring , which are all data intensive.

Although there is no strict relation between these parameters, there is a reason to expect a certain correlation because common problems in the networks such as congestion and router/link failures can be assumed to affect all the parameters in a negative way. On the other hand, absolute correlations cannot be expected. For example, the fact that link capacities are limited does not imply congestion, and so it is obvious that we can experience good delay/jitter/packet loss performance even with a limited bandwidth.

In this paper, we present a practical investigation of the hypothesis: is there a relation between the changes in delay and other QoS parameters between machines in the global Internet? Moreover, we present measurements performed throughout 24 continuous hours between different network points, giving an idea of how the QoS in the current Internet changes within a specified time frame.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

The main contribution of the paper is testing the hypothesis against real-life operations and the practical results, where we measure how different QoS parameters change over time in different long-distance networks. We show that smaller increases in latency often also lead to longer file transfer times, even though the results are not fully consistent, and the opposite relation was not observed in our results. This knowledge is important to take into account when designing cloud services and architectures and when defining SLAs: simply measuring ping throughout a session can, upon observing a sudden change/increase, give a warning that network conditions are worsening or improving, triggering a check against the SLA whether to escalate alerts to the service provider to either optimize the service or seek reimbursements for failure to deliver against the SLA. However, we believe that more refined methods could give more accurate results, which could be used for gradually adjusting or moving (fully or in parts) a service.

Much of the existing research within the field deals with estimating bandwidth while keeping complexity and overheads low, see for example [14], [15], and [16]. A recent overview and comparison of tools for this purpose is provided in [17]. Compared to this, our approach is broader in scope as it attempts to also estimate other parameters, such as delay and jitter. Moreover, the overhead imposed by ping packets is very small. It made it suitably possible to test various network conditions continuously without imposing any significant load to the network that could affect the performance. This distinguishes our approach from otherwise very useful tools for bandwidth estimation, such as Netalyzr [18]. Although it might give a less accurate estimation, changes in the continuous measurements could be used for triggering a more accurate estimate, thus combining the best of both worlds.

The paper is organized as follows. After the introduction, Section 2 presents relevant background and definitions. Section 3 presents the methods and test setup applied, leading to the results in Section 4. Section 5 presents conclusion and further works.

This paper is an extension of our previous conference paper [19].

2. BACKGROUND AND DEFINITIONS

The well specified QoS parameters are bandwidth/throughput, delay, jitter, and packet loss. Because the work in this paper mainly deals with high-capacity connections between universities, the focus is on the following three parameters, all relevant to different real-time and non-real-time CC services:

Throughput: Measured as the average maximum data rate from sender to receiver when transmitting a file. Thus, in our terms, throughput is measured one direction at a time. For the experiments in this paper, it is measured by the time it takes to transmit a file of a certain length.

Delay: Measured as the round-trip time for packets, simply using the standard ping command.

Jitter: Measured on the basis of the variation in delay. This paper bases the measurements on the ping packets as described previously and adopts the definition from RFC 1889 [12]: $J = J' + (|D(i-1,i)| - J')/16$. So the jitter J is calculated continuously every time a ping packet is received, based on the previous jitter value, J', and the value of $|D(i,j)|$, which is the difference in ping times between the ith and jth packets.

Packet loss is not considered during this study, because it is expected to be statistically insignificant (the assumption was actually confirmed during the experiments).

In order to be able to compare trends, it is necessary to smooth the observed values of throughput and delay. This is explained in details in Section 3.

3. METHODS

3.1. Description of experiments

The idea is to measure how the different QoS parameters change over a 24-hour period, between different geographically dispersed clients connected to the Internet. In particular, both inter-European and transcontinental connections are studied. For this reason, a setup is established with a main measurement

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

node at Aalborg University, Denmark (AAU), and other measurement stations at remote universities in the following places: (i) Bydgoszcz, Poland, (ii) Bangi, Malaysia, and (iii) Fortaleza, Brazil, shown in Figure 1. In the figure, the connections from AAU to the rest of the locations are highlighted over a mesh network among all the servers. In future experiments, other locations can be chosen instead of AAU. Between AAU and each of the measurement stations, experiments are conducted over 24-hour periods, where latency (ping packets sent from AAU), jitter (derived from the ping measurements), and throughput (based on FTP transfers from AAU to the measurement stations) were continuously measured. The ping packets were sent in 10 s intervals, whereas the FTP transfer was done by transmitting a 20 MB file from AAU to the measurement stations every 5 min.

The smoothing function mentioned in Section 2 was chosen to avoid small variations (e.g., due to operating system scheduling) destroying the bigger picture, whereas, on the other hand, the smoothening intervals were not so long to either distort or obviate the actual trends being monitored. It was chosen to show ping and throughput in moving non-weighted averages over 10-minute intervals. This means that the throughput for measurement t is given by the average of the measurements $t-1$, t, and $t+1$. For the latency, it is the average of the ping values measured within 5 min before and after the actual measurement time. The jitter is by definition a moving average, and no additional smoothing has been applied.

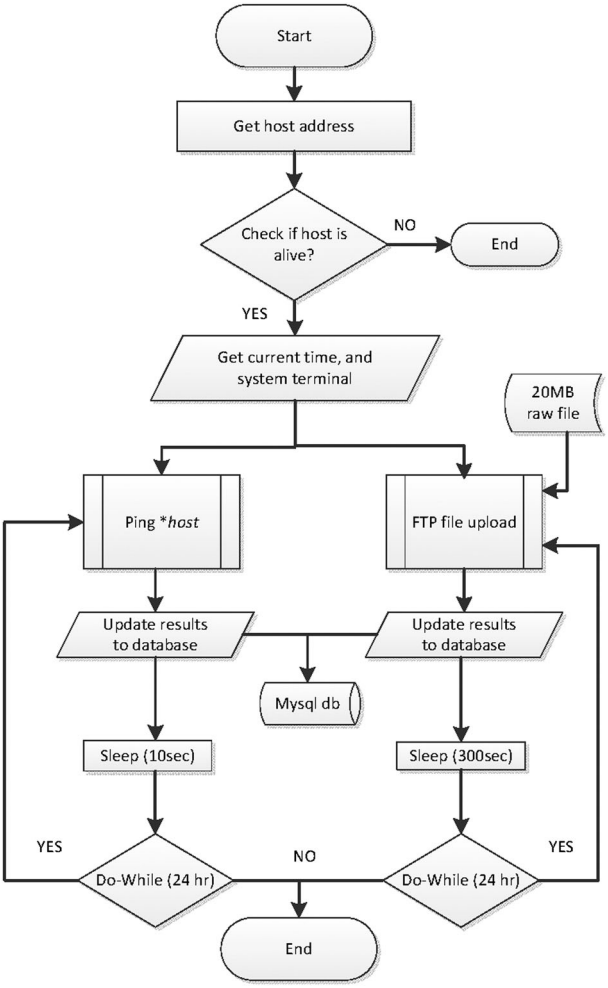

Figure 1. Geographical locations for the setup. The black lines indicate the connections used for our experiments.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

It should be noted that the time intervals between FTP transfers, and the choice of smoothing function for these values, were based on the assumption that we would usually see only smaller variations in file transfer for neighbor observations, that is, that transfer times would be quite stable for most observations. As described later, there were larger local variations than we expected.

For each destination, the measurements were carried out for two 24-hour intervals in order to confirm the validity of the results. Because of technical problems, only one measurement was done for the server in Malaysia.

3.2. Test setup

The test setup consists of four dedicated PCs running Ubuntu operating system, connected to the Internet, and assigned public IP addresses. All the PCs were running FTP servers, and also having ping enabled, and access to the Internet connection was co-shared with a gigabyte connection. Python scripting was used in order to implement the proposed measurement methodology. Besides that, MySQL database was used for storing the results. The AAU location was selected for the primary execution of the scripts. Both ping and FTP were accessed in system terminal through Python's 'sub-process' module. The output from ping and FTP were then parsed using Python's 're' module. In order to avoid any bottleneck and so on, only one simultaneous test to any given remote server was performed. A high level flow chart is shown in Figure 2.

4. RESULTS

First, the possible correlation between latency and throughput is investigated, based on the results shown in Figures 3–7. In order to be able to observe temporal behaviors, the measurement values are arranged from midnight to midnight, even though the actual experiments started at different times as listed in the figure captions.

It is hard to find a consistent correlation between the two parameters, and in many places, the parameters seem to change independently of each other. This is, for example. the case for the 'spike' of increase in latency in Figure 7, shown with more details in Figure 8. The latency increases significantly for a while, but this is not matched by an increase in file transfer times. In the other figures, there appear to be a relationship, where an increase in latency also results in increased file transfer times. This is most visible where the latency is quite stable over time, as small spikes in

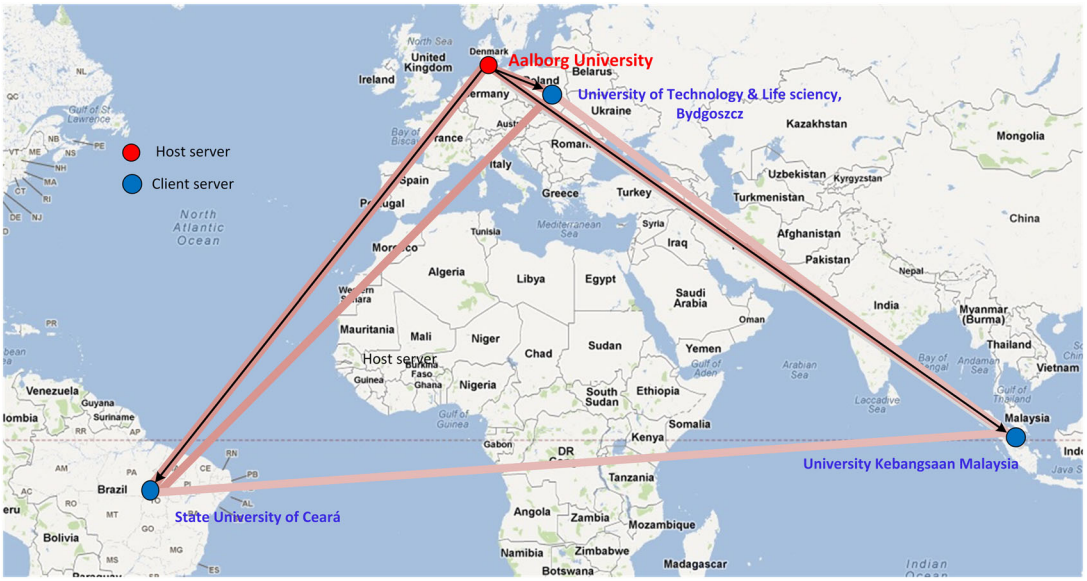

Figure 2. Flow chart of the experimental setup, illustrating how file transfer and latency measurements are carried out concurrently.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

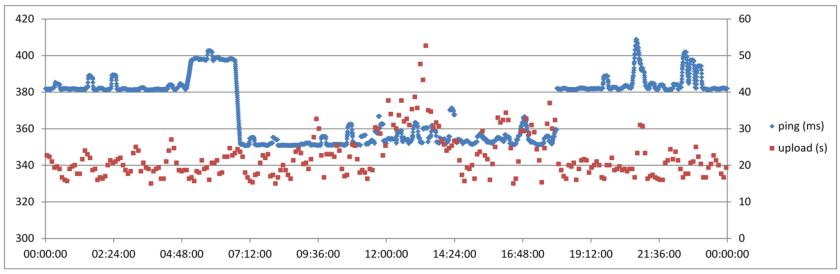

Figure 3. Latency (ping, left scale) and throughput (upload times, right scale) for the first experiment between Aalborg University and the server in Brazil. The experiments were carried out between 18:00 and 18:00 (Danish time, Coordinated Universal Time + 1).

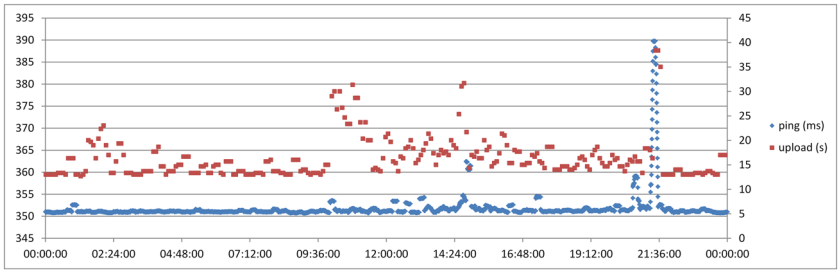

Figure 4. Latency (ping, left scale) and throughput (upload times, right scale) for the second experiment between Aalborg University and the server in Brazil. The experiments were carried out between 10:00 and 10:00 (Danish time, Coordinated Universal Time + 1.

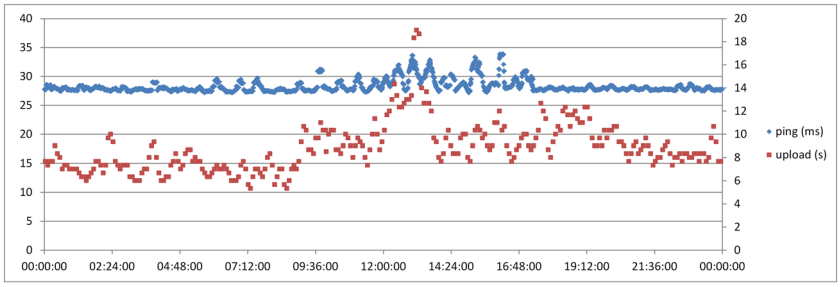

Figure 5. Latency (ping, left scale) and throughput (upload times, right scale) for the first experiment between Aalborg University and the server in Poland. The experiments were carried out between 11:40 and 11:40 (Danish time, Coordinated Universal Time + 1).

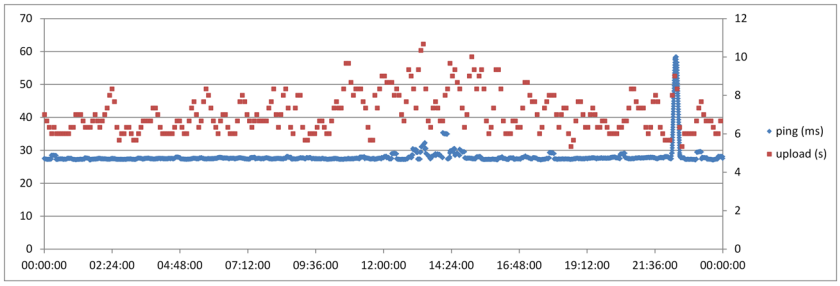

Figure 6. Latency (ping, left scale) and throughput (upload times, right scale) for the second experiment between Aalborg University and the server in Poland. The experiments were carried out between 10:00 and 10:00 (Danish time, Coordinated Universal Time + 1).

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

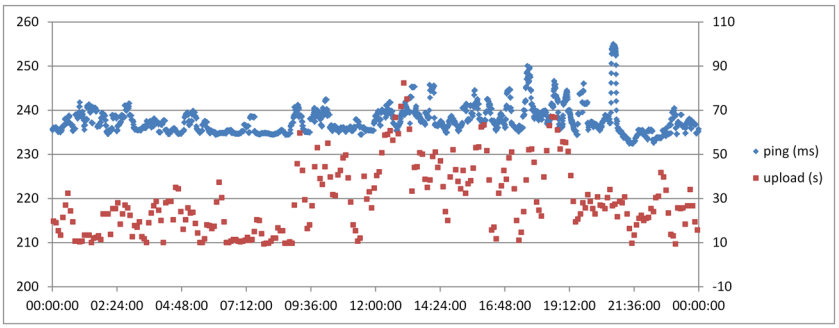

Figure 7. Latency (ping, left scale) and throughput (upload times, right scale) for the experiment between Aalborg University and the server in Malaysia. The experiments were carried out between 11:30 and 11:30 (Danish time, Coordinated Universal Time + 1).

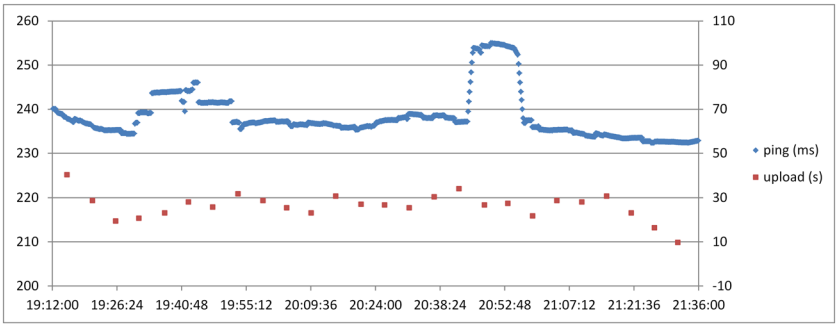

Figure 8. The same results as shown in Figure 7 but showing only the time from 19:12 to 21:36, and thus including both a smaller and larger spike. There seems to be no significant correlation between the ping and upload times in this figure.

latency are matched by spikes in throughput. This is clearly visible in Figures 3, 5, and 9, where the latter shows a more detailed view of the first part of Figure 3 and can also be seen in other figures. The tendency was confirmed also by studying more of the experiments closer. The opposite does not seem to hold: file transfer time seems to vary even when the latency remains constant, and when latency increases, this is not necessarily reflected in the file transfer times.

What can also be observed from these figures is that there are no consistent variations over the 24-hour periods. The variations are generally locally varying over time, with some rather dramatic changes, for which we do not know the reasons. For the Polish results (Figures 5, 6), there could be a relation, where file transfer times and, to some extent, latency increase during working hours, but it is hard to tell, and the patterns are not really similar for the two days.

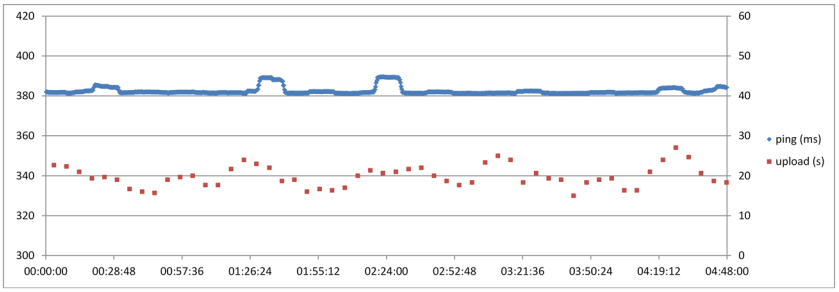

Figure 9. The results from Figure 3, for the first 4:48 h show that the upload times seem to increase during the latency spikes.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

Figures 10–13 demonstrate the smoothing functions of FTP transfer and ping times for the first set of Polish measurements. The measurements with the Brazilian and Malaysian sites show similar trends. Figures 10, 11 show larger variations in FTP transfer times than we expected, implying also larger differences between the actual and smoothed values than expected. From these figures, it is also clear how a single measurement value impacts the overall pictures. Figures 12, 13 show that the ping values are generally smooth, and the smoothing function ensures that a single variation does not significantly influence the overall results.

Calculating correlation coefficients for the relationship between ping and latency did not lead to conclusive results. The strongest correlation was found in the first experiment between AAU and Poland, where the correlation coefficient was 0.49. For the other experiments, the values were 0.22 (Poland), −0.35 and 0.29 (Brazil), and 0.37 (Malaysia).

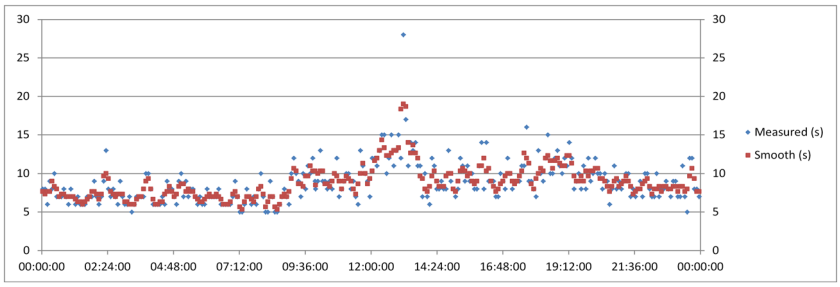

Figure 10. Real and smoothened FTP measurements for the first experiment between Aalborg University and the server in Poland.

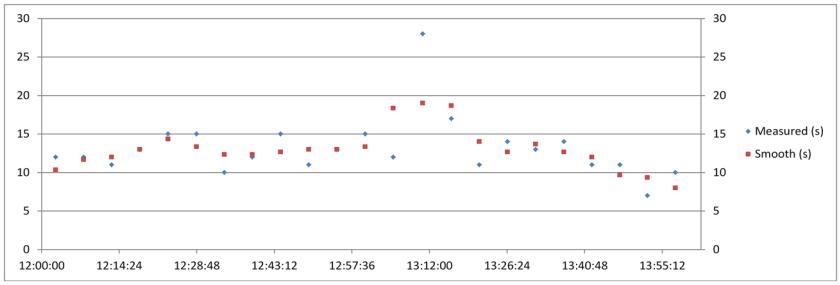

Figure 11. Real and smoothened FTP measurements for the first experiment between Aalborg University and the server in Poland. It is the same results as presented in Figure 10 but presenting a limited time span for better visualization of what is happening.

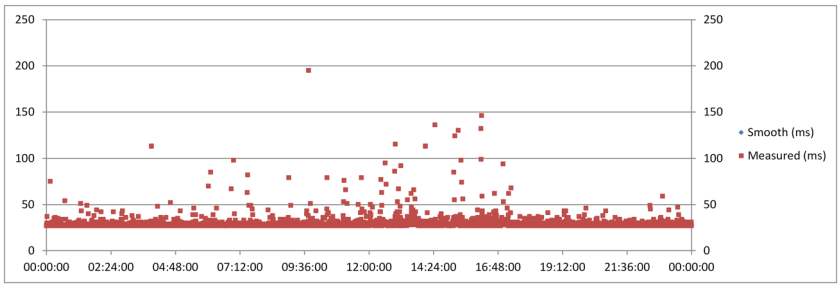

Figure 12. Real and smoothened ping measurements for the first experiment between Aalborg University and the server in Poland.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

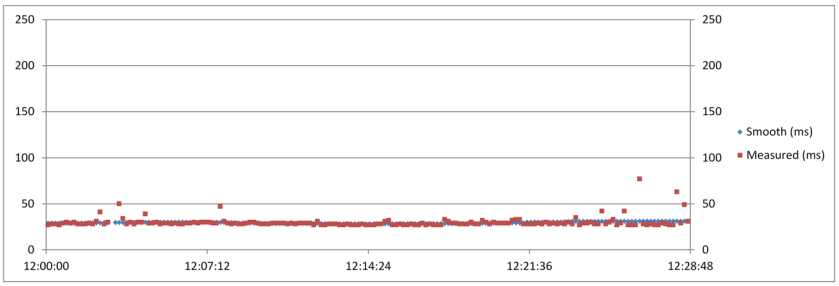

Figure 13. Real and smoothened ping measurements for the first experiment between Aalborg University and the server in Poland. It is the same results as presented in Figure A but presenting a limited time span for better visualization of what is happening.

Next the relationship between latency and jitter is studied. At a first glance, there is a close dependency between latency and jitter, where the spikes in latency are followed also by spikes in jitter. See Figures 14–19. Some relationship was also confirmed by the correlation coefficients, which were 0.69 and 0.33 (Poland), 0.00 and 0.47 (Brazil), and 0.58 (Malaysia).

This is not surprising as the jitter is by definition expressing changes in latency, so the correlation could be a result of the spiky nature of variations. An interesting observation is that when latency increases and stabilizes for a while, the jitter seems to fall back to the previous levels, adding to the difficulty in establishing a clear relationship.

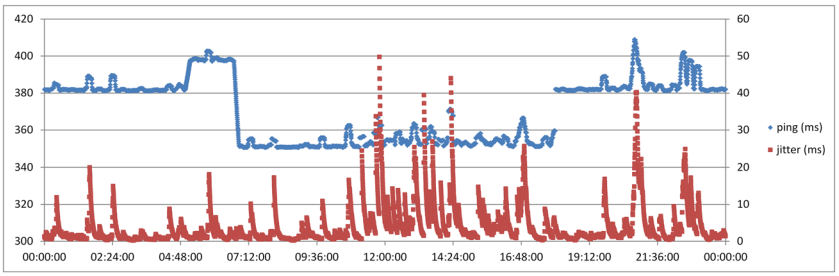

Figure 14. Latency (ping, left scale) and jitter (right scale) for the first experiment between Aalborg University and the server in Brazil. The experiments were carried out between 18:00 and 18:00 (Danish time, Coordinated Universal Time + 1).

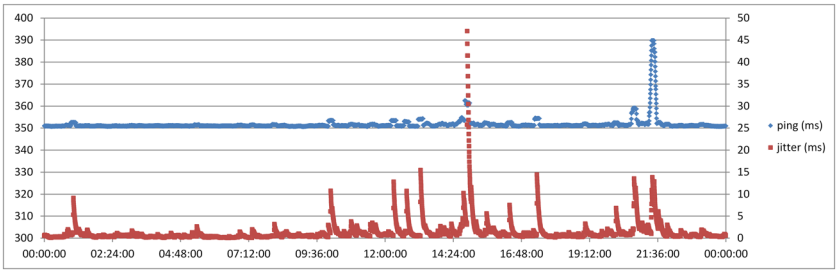

Figure 15. Latency (ping, left scale) and jitter (right scale) for the second experiment between Aalborg University and the server in Brazil. The experiments were carried out between 18:00 and 18:00 (Danish time, Coordinated Universal Time + 1).

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

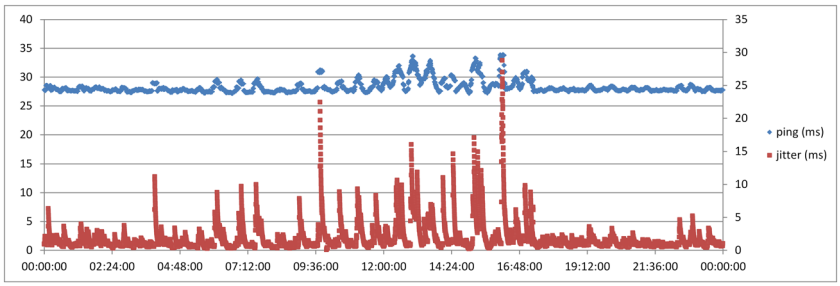

Figure 16. Latency (ping, left scale) and jitter (right scale) for the first experiment between Aalborg University and the server in Poland. The experiments were carried out between 11:40 and 11:40 (Danish time, Coordinated Universal Time + 1).

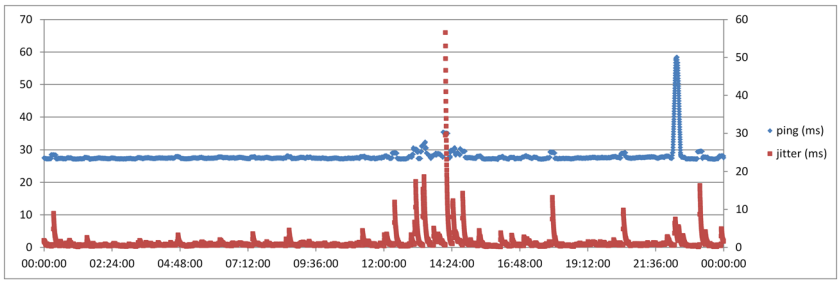

Figure 17. Latency (ping, left scale) and jitter (right scale) for the second experiment between Aalborg University and the server in Poland. The experiments were carried out between 10:00 and 10:00 (Danish time, Coordinated Universal Time + 1).

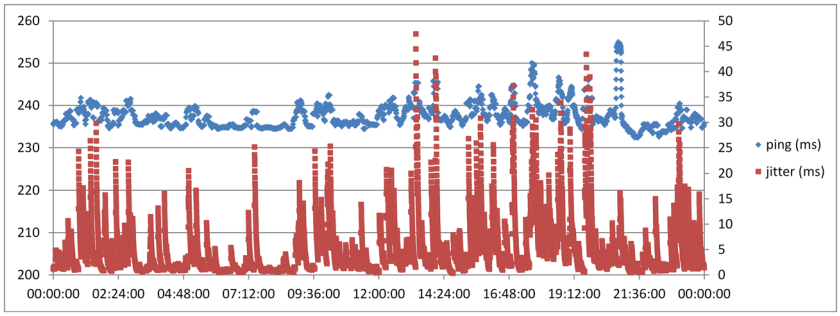

Figure 18. Latency (ping, left scale) and jitter (right scale) for the experiment between Aalborg University and the server in Malaysia. The experiments were carried out between 11:30 and 11:30 (Danish time, Coordinated Universal Time + 1).

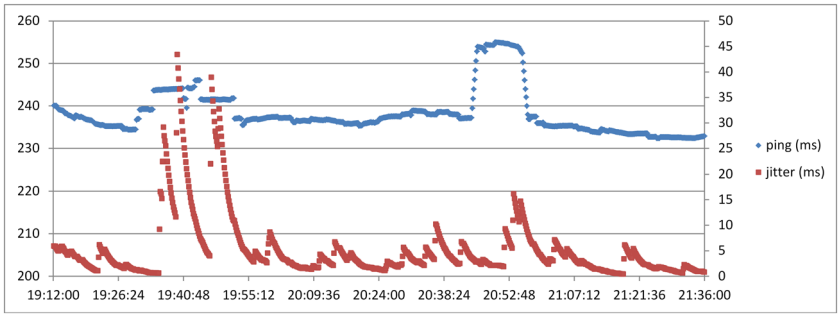

Figure 19. The same results as shown in Figure 14 but showing only the time from 19:12 to 21:36, and thus including both a smaller and larger spike in ping times.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

5. DISCUSSION

Predicting network QoS is essential when choosing how (and where) to run services in the cloud. This paper investigated if it is possible to use latency as an indicator for the other QoS parameters throughput and jitter.

Experiments were carried out between servers in globally spread locations (Denmark, Poland, Malaysia, and Brazil). Based on these, it was not possible to find any fully consistent relationships between the parameters. However, it seems that in many cases, smaller spikes in latency are correlated to spikes also in file transfer times. The results were not fully consistent though, and in several cases, we observed changes in throughput without corresponding changes in latency. Observing the correlation coefficients did not bring fully conclusive results either. All in all, there is probably some correlation between changes in latency and changes in throughput, and to some extent, changes in latency can be used to predict that throughput may also change. Our observations also show some relationships between latency and jitter. This seems to be partly because of the spiky nature of latency: when latency increases for a short while and decreases again, this leads to higher jitter during this peak.

The results obtained are important to keep in mind when designing cloud services and/or cloud architectures, as well as defining QoS requirements and related SLAs, where it might be necessary to constantly monitor all relevant QoS parameters in order to be able to act upon network changes—for example, by adapting or moving services. For other services, it might be sufficient to simply send ping packets with certain intervals, to check if things appear to be stable.

We have not discussed in this paper what action should be taken when there are indications of worsening network conditions, and this might also depend on the specifics of the application—a simple response could be to ,for example, lower the sending rate by choosing a lower-quality codec for video or audio, but this solution does not apply in all cases. For other applications, a controlled shutdown could be initiated, implemented on an application-specific basis. In order to react upon the changes, a strategy could be to have changes in latency measurements trigger a larger set of measurements to make a more precise assessment of the quality of the connection (possibly monitoring this continuously for a certain time to be able to react upon quality dropping below a certain threshold value). These additional measurements could be carried out using existing tools and methods for bandwidth and performance estimation. In all cases, before defining such a policy, it should be clear what information is required in order to adjust the behavior of an application accordingly.

The proposed approach could be complemented in a number of ways, including the previously mentioned refinements of the ping measurements, and an addition of passive monitoring, for example, by checking the response time for Transmission Control Protocol (TCP) ACKs. There is a number of other ways to combine active and passive monitoring that could be relevant to consider as well.

First of all, we considered only round-trip times, that is, latencies in both sending and receiving directions. Whereas packet loss or latency in any direction might affect TCP transfers, a reduction in throughout in a direction where mainly ACKs are sent might not impact the quality of the connection at all, and so measuring the quality in either direction might give a more precise picture than just measuring round-trip times.

Second, we only studied TCP traffic and not User Datagram Protocol (UDP) traffic, despite the fact that UDP is often used for real-time communication such as in gaming, voice and live video streaming. With no ACKs, the previous statement on studying one-way latency (and other quality parameters) is even more relevant for UDP traffic. Within the limits of global time consistency, it is not difficult to measure one-way UDP delays.

Third, more measurements than just changes in latency might be needed in order to assess the quality of a connection—either at certain time intervals or when trigged by the latency measurements, other measurements, or some trigger from the application running. Developing TCP and/or UDP throughput measurements that do not overload or congest the network connection, especially if the sample rates are even higher than that of the experiments conducted in this paper, is a challenging task. This is even more critical in a setting where there are indications of network problems.

The last aspect is detection of where in the network the problem occurs. In the global Internet, where the connections include many different network operators, service providers, and management services, the information might not be of immediate use, but it could be used for statistical purposes

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

and improvement of connections in a longer term. Developing methods for detecting where a change/problem occurs is outside of the scope for this paper, but we would encourage further research to look into this exciting problem.

6. CONCLUSION AND FURTHER WORKS

Constantly monitoring the quality of an Internet connection is valuable for a wide range of upcoming CC services. In this paper, we tested a simple but effective approach of delay measurements by ping packets for indicating changes in other network performance parameters. Compared with many existing approaches, for example, bandwidth estimation, an important advantage is the low overhead, which allows for continuous measurements.

However, as the results in this paper indicate that simple ping measurements do not alone give a precise image of the changes that occur, it is worth discussing what other or additional measurements could be used, while still keeping the overhead to a minimum. Also, future research work could investigate how such a low-intrusion approach could be effectively used for detecting severe changes or paralysis in network conditions in combination with other techniques and QoS parameters that would potentially trigger more sophisticated and accurate intrusive performance measurements or estimations. There are very good opportunities and potential in combining our approach from this paper with such state-of-the-art techniques and complex QoS parameters that have been defined and developed for CC services during the last few years, for example, [20][21]. The ability to assess network performance without over exercising complex measurements or estimations process is a key to obtaining accuracy, which is critical to fine-tune the network and network usage.

For future research, it could be interesting to collect larger amounts of experimental data, which would also make it possible to use a more analytical approach when looking for patterns. It could also be interesting to check whether responsive times for TCP ACKs could be used instead of or in addition to ping, allowing for passive monitoring.

Future research should also investigate the smoothing function(s) and adjust the sampling interval for FTP transfers. As most of the changes in latency seem to be rather short, it might give a more clear correlation between latency and other parameters to smoothen less and/or over shorter time intervals. This would require a higher sampling rate of file transfers, creating more loads on the network during the experiments and potentially affecting latency measurements. The same problem occurs if file sizes are increased to decrease the inaccuracy created by small variations in the time it takes to, for example, initiate each transfer.

Specifically for CC, future research should allow us to come up with a comprehensive list of parameters and supporting criteria in measuring CC environment performance from the bottom basic network infrastructure(s) to the highest level applications that run over or on the cloud. This requires new, innovative ways to do active real-time monitoring, probably through an overlay signaling network that does not interfere with normal CC operation or consumption of resources and bandwidth.

ACKNOWLEDGMENTS

We would like to thank the staff at Universiti Kebangsaan Malaysia Computer Centre and Faculty of Information Science and Technology Technical Support, Malaysia, and Rudy Matela, Universidade Estadual do Ceará, Brazil, for helping with the experiments.

REFERENCES

1. Na L., Patel A., Latih R., Wills C., Zhukur Z., Mulla R. A study of mashup as a software application development technique with examples from an end-user programming perspective. *Journal of Computer Science* 2010; **12**(6): 1406–1415. DOI: 10.3844/jcssp.2010.1406.1415.
2. Foster I., Zhao Y., Raicu I., Lu S. Cloud computing and grid computing 360-degree compared. *Grid Computing Environments Workshop, GCE'08*, 2008; 1–10. DOI: 10.1109/gce.2008.4738445.
3. Zhang Q., Cheng L., Boutab R. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications* 2010; **1**(1): 7–18. DOI: 10.1007/s13174-010-0007-6.
4. Wang G., Eugene Ng T.S. Understanding network communication performance in virtualized cloud. *IEEE Multimedia Communication Technical Committee E-Letter* 2011.

Copyright © 2013 John Wiley & Sons, Ltd.

5. Marinos A., Briscoe G. Community cloud computing. First International Conference Cloud Computing, CloudCom (Lecture Notes in Computer Science volume 5931) 2009; 472–484. DOI: 10.1007/978-3-642-10665-1_43.
6. Patel A., Seyfi A., Tew Y., Jaradat A. Comparative study and review of grid, cloud, utility computing and software as a service for use by libraries. *Library Hi Tech* News 2011; **28**(3): 25–32. DOI: 10.1108/07419051111145145.
7. Rings T., Caryer G., Gallop J, Grabowski J., Kovacikova T., Schilz S., Stokes-Rees I. Grid and cloud computing: opportunities for integration with the next generation network. *Journal of Grid Computing* 2009; (3)**7**:375–393. DOI: 10.1007/s10723-009-9132-5.
8. Armbrust M., Fox A., Griffith R., Joseph A., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I., Zaharia M. Above the Clouds: A Berkeley View of Cloud Computing. University of California, Berkeley, 2009. Available online at http://d1smfj0g31qzek.cloudfront.net/abovetheclouds.pdf.
9. Brewer O.T., Ayyagari A. Comparison and analysis of measurement and parameter based admission control methods for quality of service (QoS) provisioning. *Proceedings of the 2010 Military Communications Conference* 2010. IEEE, 2010; 184–188.
10. Ban S. Y., Choi J. K., Kim H.-S. Efficient end-to-end QoS mechanism using egress node resource predication in NGN network. *In proceedings of International Conference on Advanced Communications Technology, ICACT, 2006*. IEEE, 2006; **1**:483–486. DOI: 10.1109/ICACT.2006.206012.
11. Lehikoinrn L., Räty T. Monitoring end-to-end quality of service in a video streaming system. *Proceedings of the 8th International Conference on Computer and Information Science* 2009. IEEE, 2009; 750–754. DOI: 10.1109/ICIS.2009.167.
12. RFC 1889: RTP: a transport protocol for real-time applications. The Internet Engineering Task Force (IETF), 1996. Availale online at http://www.ietf.org/rfc/rfc1889.txt.
13. Chen K., Wu C., Chang Y, Lei C. Quantifying QoS requirements of network services: a cheat-proof framework. *In Proceedings of the second annual ACM conference on Multimedia systems (MMSys '11)*. ACM,New York, NY, USA, 2011; 81–92. DOI: 10.1145/1943552.1943563.
14. Kola G. and Vernon M.K. Quick probe: available bandwidth estimation in two roundtrips. In *SIGMETRICS Performance Evaluation Review* 2006; 359–360. DOI: 10.1145/1140103.1140319.
15. Xing X., Dang J., Mishra S., Liu X. A highly scalable bandwidth estimation of commercial hotspot access points. *INFOCOM* 2011: 1143–1151. DOI: 10.1109/INFCOM.2011.5934891.
16. Li M., Wu Y.-L., Chang C.-R. Available bandwidth estimation for the network paths with multiple tight links and bursty traffic. *Journal of Networks and Computer Applications January* 2013; **1**(37):353–367. DOI: 10.1016/j.jnca.2012.05.007.
17. Goldoni E., Schivi M. End-to-end available bandwidth estimation tools, *An Experimental Comparison. Traffic Monitoring and Analysis. Lecture Notes in Computer Science Volume* 2010; **6003**:171–182. DOI: 10.1007/978-3-642-12365-8_13.
18. Kreibich C., Weaver N., Nechaev B., Paxson V. Netalyzr: Illuminating the edge network. *IMC'10 Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010; 24–259. DOI: 10.1145/1879141.1879173.
19. Pedersen J., Riaz M., Júnior J., Dubalski B., Ledzinski D., Patel A. Assesing measurements of QoS for global cloud computing services. *Proceedings Of the IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, 2011. IEEE, 2011; 682–689. DOI: 10.1109/DASC.2011.120.
20. Suakanto S., Supangkat S. H., Suhardi, Saragih R. Performance measurements of cloud computing services. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)* April 2012; **2**(2):9–20. DOI : 10.5121/ijccsa.2012.2202.
21. Bardhan S., Milojicic D. A mechanism to measure quality-of-service in a federated cloud environment. In *Proceedings of the 01 workshop on Cloud services, federation, and the 8th open cirrus summit*, 2012; 19–24. DOI: 10.1145/2378975.2378981.

Copyright © 2013 John Wiley & Sons, Ltd.