ICMP-based Third-Party Estimation of Cloud Availability

Maurizio Naldi

***Abstract*—Cloud availability is an important parameter present in a typical Service Level Agreement (SLA). In order to check compliance with SLA commitments, a third party availability measurement is strongly needed. An availability estimation methods is evaluated here, based on the periodic repetition of sequence of probing packets in ICMP. Majority Voting, which declares a cloud to be available only if a majority of probing packets gets an echo from the cloud, appears to provide an accurate estimation even when the packet loss probability is rather high.**

***Keywords*—Cloud, Availability, ICMP, Service Level Agreement (SLA).**

I. Introduction

Cloud availability is a major performance parameter for cloud platforms. When an individual or a company switch to the cloud, they wish the platform to be at least as available as their in-house infrastructure. For that reason, availability is always present among the parameters to be monitored in cloud monitoring systems [1] and to be considered in cloud platform assessment systems [2], [3]. All major commercial cloud platforms typically include it in SLAs [4], [5], and boast of their values: in the survey reported in [6], 15 providers out of 17 declared at least 99.9% availability, with 12 providers declaring 100% availability.

In order to check such performance claims, third-party availability measurements are strongly desired. While a model to predict cloud reliability has been proposed in [7], and the availability of single servers has been analysed in [8], not many efforts are present in the literature to investigate the overall availability actually offered on commercial platforms. For example, availability is not considered in the comparison carried out in [9]. In the description of a major commercial platform, Microsoft Azure, provided in [10], though a high availability is claimed right in the title, no figures are provided for the expected availability. Notable exceptions are represented by [11] and [12], where a probing method based on ICMP (Internet Control Message Protocol), suitable for third-party measurements, is adopted, though exhibiting several criticalities [13]. A different approach relies on reported data rather than actual measurements: data from cloud provider status dashboards and press releases have been collected and analysed in [14] and [15].

In this paper we investigate the accuracy of the ICMP-bsaed approach to measure the availability of a cloud, after the early analysis reported in [16]. Building on the impact of networking failures on the estimation of availability as evaluated in [16], we provide a MonteCarlo simulation-based estimation of the availability achieved under concurrent true cloud outages and networking failures. Three different criteria to output an availability statement are compared: Majority voting, Unanimous Positive voting, and Unanimous Negative voting. We find that Majority Voting provides an accurate availability estimation in a wide range of cases, while the other criteria err on either side, with heavy underestimation when the packet loss probability exceeds 10^{-3}.

The paper is organized as follows. In Section II we describe the testing arrangement based on ICMP, while its performances are analysed in Sections III, IV, and V on three different timescales.

Manuscript received October 30, 2016, revised January 27, 2017.

M. Naldi is with the Department of Computer Science and Civil Engineering, University of Rome Tor Vergata, 00133 Roma, Italy e-mail: maurizio.naldi@uniroma2.it.

This research has been supported by the Italian Ministry of Education, University, and Research (MIUR) under PRIN 2012C4E3KT national research project "AMANDA Algorithmics for MAssive and Networked DAta".

II. Testing Methodology

The way we measure availability has a great impact on the numeric value we get. In this section, we clarify what is meant by availability in this context and how we measure it by using ICMP.

The availability A of a system, which undergoes phases of normal working separated by outages, is defined as the ratio of the uptime and the total time:

$$A = \frac{\text{Uptime}}{\text{Total Time}} = \frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}}. \quad (1)$$

In the context of a cloud, since the service provided by a cloud is to respond to service demands, such as for some content stored in the cloud (cloud storage) or for the result of some processing task carried out on the cloud (cloud computing), we can consider as uptime the time during which the cloud responds, and as downtime the time during which it stops doing so.

It is therefore natural to see the cloud as oscillating between two states: UP and DOWN. This is called the *Dual State model* in [17], where transitions between the two states are triggered by failures and service demands are not satisfied throughout the downtime period. Several real world examples of Service Level Agreements in clouds are amenable to being formulated according to this model, e.g., Amazon EC2, HP Cloud Compute, and Google Apps SLA, again as observed in [17]. It is implicit that this model considers no graceful degradation: either the cloud responds to the service demand or not.

In order to check whether the cloud is up or down, the method we analyse here, proposed in [12], employs the *ping* command of the Internet Control Message Protocol (ICMP)

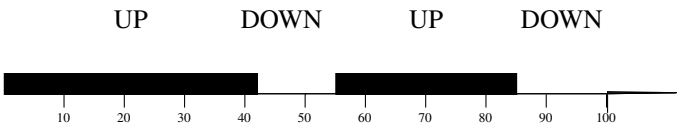


Fig. 1. Sequence of up and down states

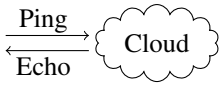


Fig. 2. Testing arrangement

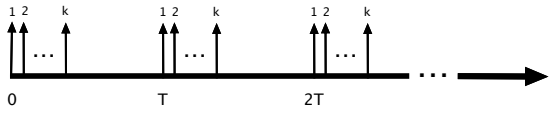


Fig. 3. Test sequence

[18]. A ping-based approach to measure availability had already been employed in [19] to analyse cloud storage facilities and in [20] to analyse the reliability of internet sites. The method operates by sending echo request packets (*pings*) to the target cloud (the front-end server, e.g., the hostname in the URL of the stored object for cloud storage) and counting the echoes as a measure of success. A ping response indicates that the target host is connected to the network, is reachable from the query agent, and is in a sufficiently functional state to respond to the ping packet. After waiting for ICMP echo replies, the protocol reports packet loss and round-trip-time statistics. Though it is recognized that failure to respond is not so informative because it cannot be reliably inferred that the target host is not available, the foremost causes of failed response are related to the network (and are accounted for in our analysis later), excepting the presence of some form of firewall in the end-to-end path that blocks the ICMP packet being delivered [21].

Under this approach, the cloud is seen as a black box (see Fig. 2). Though the resource may be located elsewhere, so that queries are actually served by a server in a different location, the availability of the cloud is embodied by the responsiveness of the front-end server. Therefore, this scheme serves equally well as a model to analyse configurations where contents or processing resources are distributed over several geographical areas, as long as the service access point is one.

Each ping allows therefore to see if the front-end server is up and running. Pings are not shot in isolation, but are sent off in bursts to achieve a better accuracy in the face of temporary glitches. As shown in Fig. 3, the tester sends off a burst of k pings (in [12] a repetition period of 2 seconds was envisaged), and the whole sequence is periodically repeated every T seconds (which, again in [12], was set at 10 or 11 minutes). If the aim is to compare the observed availability against the Service Level Agreement commitments over any period of length C, we consider an observation window of length C for contractual purposes.

At the end of each probing sequence made of k pings, the tester outputs an availability statement concerning the cloud, declaring the cloud service either available or not. That statement is kept as valid till the next probing sequence, when a new availability statement is emitted. The observation window can therefore be considered as made of $B = \lfloor C/T \rfloor$ time blocks, so that we can have a number of availability statements in the $[0, B]$ range, where the lowest value (0) represents a service considered as totally unavailable over the observation window, while the largest value (B) represents a 100% availability. If we use the symbol N_{out} to indicate the number of blocks for which a NOT AVAILABLE statement is output, the availability estimate is:

$$\hat{A} = 1 - \frac{N_{\text{out}}}{B} = 1 - \frac{N_{\text{out}}}{\lfloor C/T \rfloor}. \tag{2}$$

In order to correctly estimate large availability figures, the number of blocks must be correspondingly large, otherwise the granularity due to the blocks will mask short unavailability periods. For example, if $B = 100$, the next largest availability figure to 100% that we can estimate is $99/100 = 0.99$, i.e. just a 2-nine availability. If we wish to measure availability figures as large as four nines ($A = 0.9999$), the minimum number of blocks must be 10000. In order to achieve that number, if we set $T = 10$ minutes, the observation window must be at least $10 \cdot 10000 = 100000$ minutes long, which corresponds to slightly more than 69 days. In general, to measure an availability A, the observation window must be

$$C \geq \frac{T}{1 - A}. \tag{3}$$

In order to arrive at an availability statement for any single sequence of k pings, we consider three alternative criteria, by adding a Unanimous Positive voting to the two listed in [16]:

- Majority voting
- Unanimous Positive voting
- Unanimous Negative voting

In Majority Voting an outage is declared if a majority of pings get no echoes. In Unanimous Positive voting, all echoes must be received to declare the cloud available. In Unanimous Negative voting, an outage is instead declared if no ping gets an echo. What we get in an example for sequences made of $k = 5$ pings is shown in Table I. As can be seen, the Unanimous Positive voting criterion will lead to the most severe availability statements, while the Unanimous Negative voting criterion will output a NOT AVAILABLE statement just when there is a persistent string of missing echoes.

III. Missing Echoes over a Single Ping

Before examining the compliance of cloud services with the committed availability (i.e., over an observation window), we analyse the behaviour of the cloud over a single ping.

The actual outcome of the testing process is impacted by both network and cloud failures, so that an outage may be declared even when the cloud is perfectly running, thus leading to a false outage declaration. In this section, we examine the

TABLE I
OUTCOMES OF CRITERIA FOR AVAILABILITY STATEMENTS

Ping outcome		Statement		
Replies	Missing echoes	Majority	Un. positive	Un. negative
0	5	NOT AVAIL.	NOT AVAIL.	NOT AVAIL.
1	4	NOT AVAIL.	NOT AVAIL.	AVAIL.
2	3	NOT AVAIL.	NOT AVAIL.	AVAIL.
3	2	AVAIL.	NOT AVAIL.	AVAIL.
4	1	AVAIL.	NOT AVAIL.	AVAIL.
5	0	AVAIL.	AVAIL.	AVAIL.

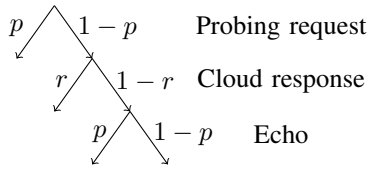


Fig. 4. Sequence of events and pertaining probabilities

testing process in the general case when the cloud may be available, but network failures are also present, and in the two alternative cases when the cloud is either perfectly working (and responding to pings) or not working.

If the cloud is available, we expect to receive a positive echo response for each request, but probing packets routinely get lost due to network failures. If p is the packet loss probability on the testing-station to/from cloud path (we assume that it is the same in either way, and the failures on the two trips are uncorrelated due to time spacing), the resulting sequence for a single probing instance is shown in Fig. 4. There we see how the various path sections contribute to the outcome of the testing process in a single ping, when there is the possibility that the cloud is not working.

If the true cloud outage probability is r, the probability of missing an echo (on a single probing instance) is equal to the sum of the probabilities of the three left branches in the tree of Fig. 4:

$$\begin{aligned} P_{\text{me}} &= p + (1-p)r + (1-p)(1-r)p \\ &= p(2-p) + r(1-p)^2 \simeq 2p + r, \end{aligned} \tag{4}$$

where we see clearly the contribution due to network failures (first term of the sum) and that due to the cloud (second term). In Fig. 5 (showing the true, rather than approximate, P_{me}) we see that the probability of declaring an outage is a linear function of r and biased approximately by a $2p$ term.

The biasing factor P_{me}/r may indeed be very large, as shown in Fig. 6, especially when $p > r$.

This general case may be simplified if we look at the two cases where the cloud is either working or not working.

If we focus on the former case, the response tree reduces to that shown in Fig. 7: two leaves of the binary tree give rise to a negative outcome, and just one (both trips occurring with no packet loss) is reported as successful.

Since the cloud is available, what is reported as a failure is actually a false outage. If we mark by a flag variable X the status of the cloud ($X = 1$ if the cloud is working and 0 otherwise), and by another flag variable Y the outcome of the ping test ($Y = 1$ if we get an echo and 0 otherwise), the

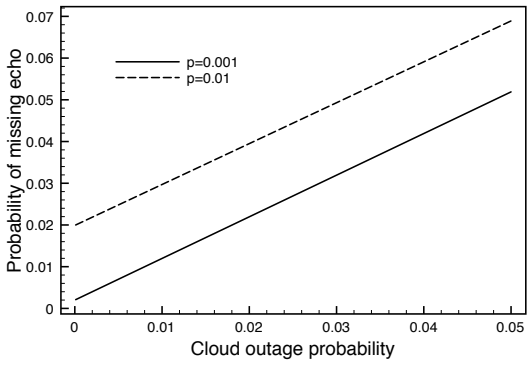


Fig. 5. Probability of missing echo

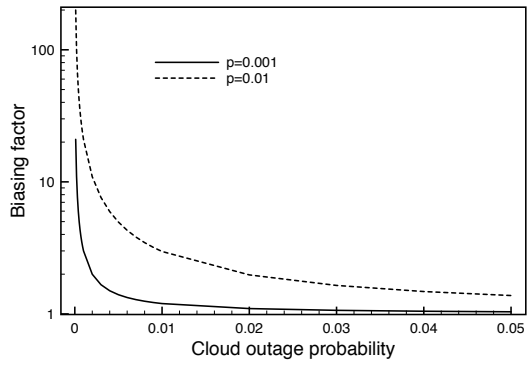


Fig. 6. Biasing factor

probability of a missing echo conditional to the cloud working in a single probing instance is therefore:

$$P_{\text{mecw}} = P[Y = 0|X = 1] = p + (1-p)p = p(2-p) \simeq 2p. \tag{5}$$

In the case that the cloud is not working, of course we will never receive an echo, though the network may be operating correctly.

IV. FALSE OUTAGES OVER A PROBING BURST

If we move from a single ping to a sequence of pings, we have several alternative criteria to declare an outage (a false outage), among which the most relevant have been described in Section II:

- Majority voting;
- Unanimous Positive voting;
- Unanimous Negative voting.

We now evaluate the probability of declaring a false outage under the three criteria when the cloud is perfectly working.

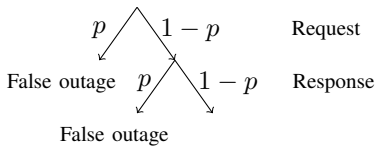


Fig. 7. Sequence of events under no cloud failure

TABLE II
FALSE OUTAGE PROBABILITY OVER A PROBING BURST

Criterion	False outage probability
Majority voting	$\sum_{i=k_{\min}}^{k} \binom{i}{k} P_{\text{mecw}}^{i}(1-P_{\text{mecw}})^{k-i}$
Unanimous Positive voting	$1-(1-P_{\text{mecw}})^{k}$
Unanimous Negative voting	P_{mecw}^{k}

According to Majority Voting, we declare a cloud outage after a sequence of k echo requests when we have at least $k_{\min} = \lceil \frac{k+1}{2} \rceil$ negative responses (no echoes). We assume that the outcomes of successive probing instances in the case of networking failures are uncorrelated. The number of missing echoes in a burst of k requests follows therefore a binomial distribution with parameters p and k. The false outage probability with k probing instances is therefore:

$$P_{\text{fo}}(k) = \sum_{i=k_{\min}}^{k} \binom{i}{k} P_{\text{mecw}}^{i}(1-P_{\text{mecw}})^{k-i} \tag{6}$$

Under the second criterion (Unanimous Positive voting), we must instead receive all echoes in a sequence of k requests to declare the cloud available, or, alternatively, we declare the cloud down in all but one case (all echoes present). Therefore the false outage probability is:

$$P_{\text{fo}}(k) = 1-(1-P_{\text{mecw}})^{k} \tag{7}$$

Finally, the Unanimous Negative voting criterion requires instead that no echoes are received to declare an outage, so that the probability of an outage declaration over k pings is:

$$P_{\text{fo}}(k) = P_{\text{mecw}}^{k} \tag{8}$$

The three cases are summed up in Table II.

The aim of the close repetition of probing instances is anyway to knock down the probability of false outages. In Fig. 8, we see that, for the choice $k = 9$ adopted in [12], the repetition mechanism is highly effective: even when the packet loss probability is quite high ($p = 0.05$), the probability of false outage is as low as $8 \cdot 10^{-10}$ when the Unanimous Negative voting criterion is chosen and $8 \cdot 10^{-4}$ with Majority Voting. However, the Unanimous Positive voting is quite more severe in declaring outages, leading to a high false outage probability (0.6 when the packet loss probability is 0.05). As expected, the Unanimous Negative voting criterion is much more effective than Majority Voting in ruling out false outages, while the Unanimous Positive voting criterion is not at all.

The number of pings in the elementary testing sequence has of course a significant impact. In Fig. 9, plotted for the same packet loss probability $p = 0.01$, we see the exponential fall of the probability of false outages under the Unanimous Negative voting criterion; Majority Voting exhibits a softer descent, whose staircase-like appearance is an artifact due to the majority rule (e.g., when passing from 4 to 5 pings the useful cases are respectively 3 or 4 out of 4, but 3, 4, or 5 out of 5). Instead, the Unanimous Positive voting criterion leads to a false outage probability increasing with the number of pings. This rule proves therefore ineffective in reducing the number of false outages.

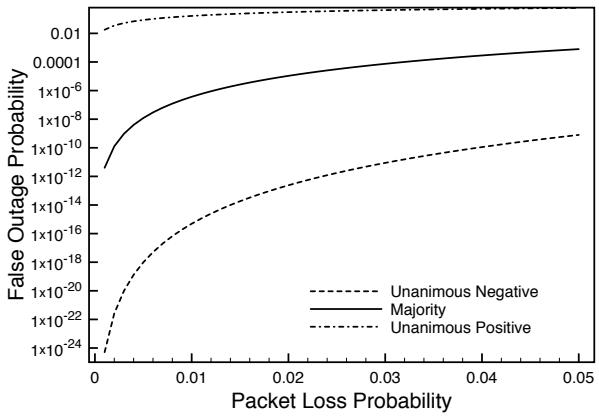


Fig. 8. Probability of false outage (k=9)

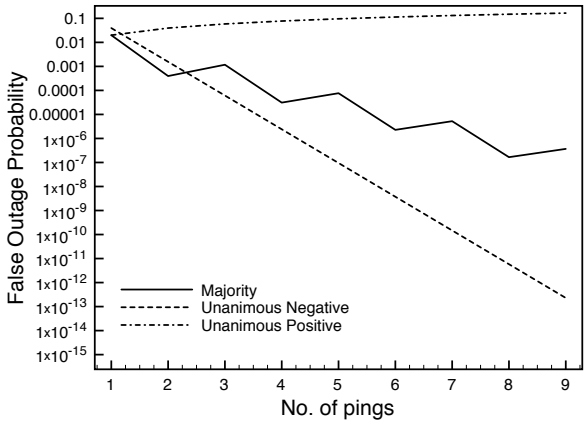


Fig. 9. Impact of the number of retries on false outage probability (p=0.01)

V. AVAILABILITY ESTIMATION OVER A CONTRACT PERIOD

In Sections III and IV we have examined the impact of networking failures on the observed outage probability over two different timescales: a single probing instance and a probing burst. However, the presence of outages is typically assessed to check compliance with Service Level Agreements (SLA). In that case, the check is conducted by comparison with committed availability goals within a (relatively long) observation window . In this section, we examine the availability estimate resulting from the previously described ICMP-based testing approach when an adequately long observation window is considered.

We adopt the testing sequence depicted in Fig. 3. We perform that testing sequence continuously and collect statistics over successive periods of length C, so that C is the observation window prescribed in the SLA. This window must be significantly larger than the testing period T so as to have a large number of testing sequences to perform a statistical estimation of availability. Over any testing period, after sending a burst of k pings, a decision is taken as to the presence of an outage through one of the three criteria described in Section IV. The cloud status resulting from this decision is maintained till the next testing period. If the number

TABLE III
OUTAGE DURATION STATISTICS [MIN]

Company	Average	Standard deviation	CV
Google	506.64	872.58	1.72
Amazon	473.56	1093.18	2.31
Rackspace	607.03	1210.30	1.99
Salesforce	95.2	84.35	0.89
Windows Azure	224.3	165.78	0.74

TABLE IV
ESTIMATED PARAMETERS OF THE GENERALIZED PARETO DISTRIBUTION

Company	scale parameter β	shape parameter ξ
Google	405.29	0.39
Amazon	276.43	-0.12
Rackspace	381.19	0.3
Salesforce	192.47	-0.64
Windows Azure	312.32	-0.35

of testing periods for which an outage has been declared is N_{out}, the availability is estimated as:

$$\hat{A} = 1 - \frac{N_{\text{out}}}{\lfloor C/T \rfloor}, \tag{9}$$

the quantity in the denominator representing the number of testing periods contained in the observation window.

In order to assess the capability of this ICMP-based approach to correctly estimate the availability, we perform a MonteCarlo simulation, where the observed availability is averaged over 1000 simulation instances.

The model of cloud outages is based on the findings of the empirical analysis reported in [15]. The statistics collected over several prominent cloud storage providers are reported in Table III. For each provider a best-fit search was conducted to model the outage durations. It turned out that the outage duration (represented here by the random variable D) is best modelled by a Generalized Pareto distribution, whose cumulative distribution function is [22]:

$$\mathbb{P}[D < x] = G_{\xi,\beta}(x) = \begin{cases} 1 - (1 + \xi x/\beta)^{1/\xi} & \text{if } \xi \neq 0 \\ 1 - e^{-x/\beta} & \text{if } \xi = 0 \end{cases}, \tag{10}$$

where β is the scale parameter and ξ is the shape parameter. The average duration is related to the scale and shape parameters by the expression

$$\overline{D} = \frac{\beta}{1 - \xi}. \tag{11}$$

The optimal values for β and ξ found during that analysis are shown in Table IV.

Here, in order to account for the variety of performances shown in Table IV, we consider a fixed shape parameter $\xi = 0.39$ (the value pertaining to Google) and a scale parameter dictated by the average outage duration $\overline{D}$ through the expression obtained by inverting Equation (11):

$$\beta = \overline{D}(1 - \xi). \tag{12}$$

TABLE V
REFERENCE CASE

Parameter	Value
Observation window C	30 days
Testing Period T	10 minutes
Average Outage Duration $\overline{D}$	500 minutes
No. of pings k	9

In the simulation procedure the durations were generated through the inverse method [23].

The occurrence of outages was instead simulated using a Poisson process with rate λ dictated by the true availability and the average outage duration, following the approach in [24]. By resorting to Wald's identities (see, e.g., Section 34.14.2.11 of [25] or Section 1.7.3 of [26]), we can write the availability as evaluated over the observation window C as:

$$A = 1 - \frac{\mathbb{E}[\sum_{i=1}^{N} D_i]}{C} = 1 - \frac{\overline{N} \cdot \overline{D}}{C} = 1 - \frac{\lambda C \overline{D}}{C} = 1 - \lambda \overline{D}, \tag{13}$$

where N is the number of outages occurring within the period C, and D_i is the duration of the i-th outage. The use of Wald's identities is valid since there is only a weak correlation among the number of summands in the sum in Equation (13) and each of the summands. In fact, the duration of an outage is independent of the number of outages, while the number of outages has a very weak correlation with outage durations, as long as we consider cloud services with high availability. Equation (13) can be easily inverted to find the rate of the Poisson process:

$$\lambda = \frac{1 - A}{\overline{D}}. \tag{14}$$

In order to assess the estimation capabilities of the three decision criteria described in Section IV, we have considered a range of values for the parameters involved in the testing method:

- Observation window length C;
- Testing period length T;
- Number k of pings in each probing burst.

As to the observation window C, we have considered values from 1 to 12 months. A testing period of 10 minutes was adopted in [12], which we mostly maintained, though we have experimented with values from 5 to 15 minutes. As to the number of pings, again in [12] a sequence made of 9 pings was recommended, and we sticked to that choice, though we also considered values as low as 5. However, unless stated otherwise, here we report the results using the reference case described in Table V since we found negligible differences for parameter values outside the reference case.

Finally, the impact of networking failures was accounted for by considering a packet loss probability ranging from 10^{-4} to 10^{-2}.

The range of availability values for which we tested the measurement scheme was [0.95,0.999], which is consistent with what has been reported in several attempts to provide third-party measurements of availability (see, e.g., [15]).

TABLE VI
IMPACT OF THE OBSERVATION WINDOW LENGTH

Observation window [days]	Availability Estimate	
	Majority Voting	Unanimous Positive voting
30	0.9898	0.9981
60	0.9885	0.9982
90	0.9900	0.9982
180	0.9902	0.9982
360	0.9899	0.9982

TABLE VII
IMPACT OF THE TESTING PERIOD

Testing Period [minutes]	Availability Estimate	
	Majority voting	Unanimous Positive voting
5	0.9898	0.9981
10	0.9885	0.9981
15	0.9900	0.9982

We now examine the impact of each of the testing method parameters on the availability estimate by adopting either the Majority Voting criterion or the Unanimous Positive voting one. We rule out the Unanimous Negative voting criterion since in all the experiments it always produced an availability estimate equal to 1, i.e., it never produced an outage statement.

We start with the length of the observation window, for which we observe that it has a negligible influence on the accuracy of the estimation (the accuracy was not tested for lengths shorter than a month). We report in Table VI the results for the reference case (the true availability was set at 0.99): the range of estimates is always below 0.07% of the central value for Majority Voting and 0.01% for Unanimous Positive voting. We note that Majority Voting provides a very good estimate of the true value, while the Unanimous Positive voting criterion always overestimates the actual availability, turning the true 2-nine availability into a nearly 3-nine one.

The impact of the testing period can likewise be considered as negligible, as shown in Table VII, which reports some values obtained for the reference case (where the true availability is 0.99). A similar situation occurred for parameter combinations different from the reference case. Again, we notice the very good estimate delivered by Majority voting, and the overestimate due to Unanimous Positive voting.

This can be considered as valid as long as the testing period does not become comparable with the duration of an outage. If we mark the occurrence of the measurement timepoint preceding the outage as time 0, so that the next measurement takes place at the time T (e.g., 10 minutes as in the reference case), the outage will take place at a random time O such that $0 \leq O \leq T$. If we consider O to be uniformly distributed, the failure will not be detected if the recovery from the outage is achieved before the next measurement interval. If the outage duration is L, that condition can be expressed as $O + L < T$. The probability that the outage goes undetected is then

$$P_{\text{nodet}} = \mathbb{P}[O + L < T] = \mathbb{P}\left[\frac{O}{T} < 1 - \frac{L}{T}\right] = \begin{cases} 0 & \text{if } T \leq L \\ 1 - \frac{L}{T} & \text{if } T > L \end{cases} \quad (15)$$

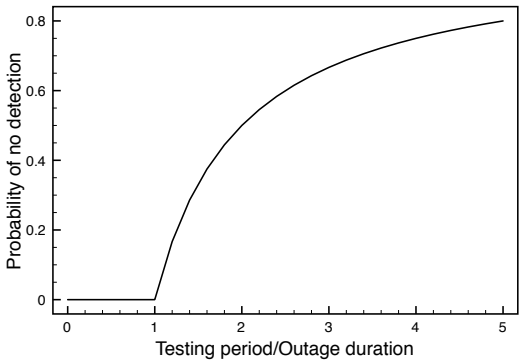


Fig. 10. Probability of an outage going undetected

TABLE VIII
IMPACT OF THE TESTING SEQUENCE LENGTH

No. of pings	Availability Estimate	
	Majority Voting	Unanimous Positive voting
5	0.9893	0.9990
6	0.9902	0.9988
7	0.9905	0.9986
8	0.9899	0.9984
9	0.9898	0.9981

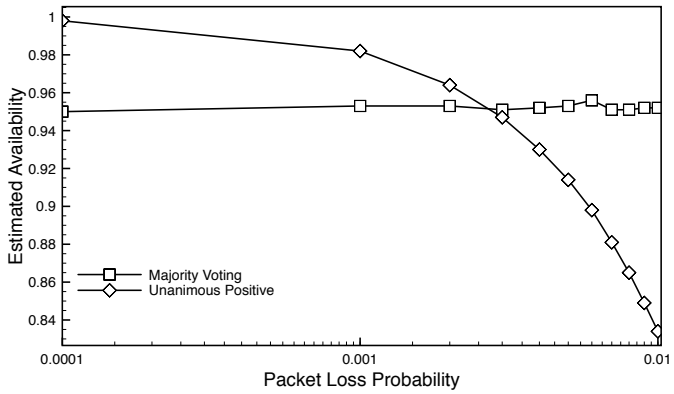
The resulting no-detection probability is shown in Fig. 10.

In that case, it could become difficult to compute the extent of SLA violations and the amount of possible compensations or insurance claims [27][28][29]. In fact, of the three measures (Number of failures; Number of long outages; Cumulative outage duration.) envisaged to be used in an insurance policy for network failures in [24] (but applicable straightforwardly to the case of clouds), just one is considered in [12]. The cumulative outage duration is equal to the overall unavailability, but the number of failures is heavily distorted since short-lived outages may go undetected, and the number of long outages may be underestimated as well, unless the threshold is longer than the measurement interval.

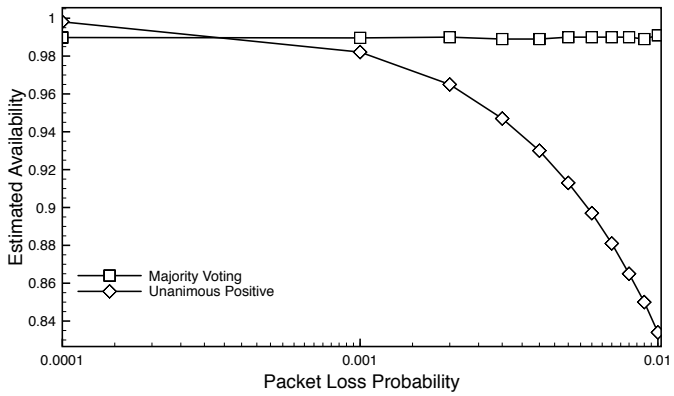
If we now turn to the length of testing sequence, i.e., the number of pings in a probing burst, again we see in Table VIII (which refers to the reference case, with $A = 0.99$, but similar results are obtained for other cases as well) that the impact of the actual number of pings is very small, though for the Unanimous Positive voting criterion the estimate can be seen to decrease as the number of pings grows, providing a better estimate. The choice of $k = 9$ seems therefore the best one in the range examined.

After examining the set of choices for the testing scheme parameters, and having found out that the accuracy is not significantly affected within the range of values considered (with the single exception of the number of pings in a probing burst, for which the highest value is to be preferred), we can now see the impact of networking failures, which is the major source of concern.

In Fig.11, plotted for two values of the true availability ($A = 0.95, 0.99$) and the reference case, we see that the outcome of the Majority Voting criterion is negligibly im-



(a) True Availability = 0.95



(b) True Availability = 0.99

Fig. 11. Availability Estimate

pacted, while the availability estimate is significantly altered under the Unanimous Positive voting criterion. As expected, the situation turns critical as the packet loss probability grows. The turning point can be located around $p = 10^{-3}$, where the estimate starts decreasing, turning from an overestimate into an underestimate as more and more false outages enter the picture. When the packet loss probability is as bad as 10^{-2}, the availability estimate would be so bad as to consider the cloud very unreliable though it actually keeps providing the usual good performance.

The analyses conducted so far, over a wide set of experiments than those reported here, show therefore that significant differences have emerged among the decision criteria we may adopt to declare an outage at each testing period. Precisely, the Majority Voting criterion has always provided a correct estimate, while the Unanimous Positive voting criterion typically errs on the plus side (overestimating the availability) when the packet loss probability is small enough, but instead grossly underestimates the availability when the packet loss probability exceeds 10^{-3}.

VI. Conclusion

We have evaluated the accuracy of a cloud availability estimation method based on the periodic repetition of sequences of probing packets (the pings made available in the ICMP protocol) by MonteCarlo simulation. Three different criteria (Majority voting, Unanimous Positive voting, and Unanimous Negative voting) have been compared to output an availability statement. Majority Voting provides an accurate statement over a wide range of testing parameters and context scenarios, even when the packet loss probability, a major source of false outages, is rather high. Instead, the Unanimous Positive voting criterion, outputting an availability statement just after all the pings in a sequence have received an echo, can lead to gross underestimation when the packet loss probability exceeds 10^{-3}. The Unanimous Negative voting criterion, always outputting an availability statement unless no echoes are received in a probing sequence, is completely unreliable, providing a statement of 100% availability in all the cases examined. The Majority Voting is therefore the criterion of choice in the ICMP-based method to estimate the availability of a cloud.

References

[1] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, "Gmone: A complete approach to cloud monitoring," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2026–2040, 2013.

[2] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a catalogue of metrics for evaluating commercial cloud services," in *Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on*, Sept 2012, pp. 164–173.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[4] A. Cuomo, G. Di Modica, S. Distefano, A. Puliafito, M. Rak, O. Tomarchio, S. Venticinque, and U. Villano, "An sla-based broker for cloud infrastructures," *Journal of Grid Computing*, vol. 11, no. 1, pp. 1–25, 2013.

[5] V. C. Emeakaroha, M. A. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. D. Rose, "Towards autonomic detection of {SLA} violations in cloud infrastructures," *Future Generation Computer Systems*, vol. 28, no. 7, pp. 1017 – 1029, 2012.

[6] E. Casalicchio and L. Silvestri, "Mechanisms for SLA provisioning in cloud-based service providers," *Computer Networks*, vol. 57, no. 3, pp. 795–810, 2013.

[7] Y.-S. Dai, B. Yang, J. Dongarra, and G. Zhang, "Cloud service reliability: Modeling and analysis," in *15th IEEE Pacific Rim International Symposium on Dependable Computing*, 2009.

[8] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *Proceedings of the 1st ACM symposium on Cloud computing SoCC*. ACM, 2010, pp. 193–204.

[9] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 1–14.

[10] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci *et al.*, "Windows azure storage: a highly available cloud storage service with strong consistency," in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. ACM, 2011, pp. 143–157.

[11] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in *9th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2010*, Vancouver, BC, Canada, October 4-6, 2010, pp. 61–74.

[12] Z. Hu, L. Zhu, C. Ardi, E. Katz-Bassett, H. Madhyastha, J. Heidemann, and M. Yu, "The need for end-to-end evaluation of cloud availability," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, M. Faloutsos and A. Kuzmanovic, Eds. Springer International Publishing, 2014, vol. 8362, pp. 119–130.

[13] M. Naldi, "A note on "the need for end-to-end evaluation of cloud availability"," *arXiv CoRR Preprint Series*, vol. abs/1408.0510, 2014.

[14] C. Cérin, C. Coti, P. Delort, F. Diaz, M. Gagnaire, Q. Gaumer, N. Guillaume, J. L. Lous, S. Lubiarz, J.-L. Raffaelli, K. Shiozaki, H. Schauer, J.-P. Smets, L. Séguin, and A. Ville, "Downtime statistics of current cloud solutions," Available at http://iwgcr.org/wp-content/uploads/2013/06/IWGCR-Paris.Ranking-003.2-en.pdf, June 2013.

[15] M. Naldi, "The availability of cloud-based services: Is it living up to its promise?" in *9th International Conference on the Design of Reliable Communication Networks, DRCN 2013, Budapest, Hungary*, March 4-7, 2013, pp. 282–289.

[16] ——, "Accuracy of Third-Party Cloud Availability Estimation through ICMP," in *39th International Conference on Telecommunications and Signal Processing (TSP)*. Vienna: IEEE, 2016.

[17] G. Hogben and A. Pannetrat, "Mutant apples: a critical examination of cloud sla availability definitions," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 1. IEEE, 2013, pp. 379–386.

[18] J. Postel, "Internet control message protocol," ISI Network Working Group Request for Comments 792, 1981.

[19] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems." in *OSDI*, 2010, pp. 61–74.

[20] D. D. Long, J. L. Carroll, and C. Park, "A study of the reliability of internet sites," in *Reliable Distributed Systems, 1991. Proceedings., Tenth Symposium on*. IEEE, 1991, pp. 177–186.

[21] G. Huston, "Measuring ip network performance," *The Internet Protocol Journal*, vol. 6, no. 1, pp. 2–19, 2003.

[22] A. J. McNeil and T. Saladin, "The peaks over thresholds method for estimating high quantiles of loss distributions," in *Proceedings of 28th International ASTIN Colloquium*, 1997, pp. 23–43.

[23] I. Mierlus-Mazilu *et al.*, "On generalized pareto distributions," *Romanian Journal of Economic Forecasting*, p. 107, 2010.

[24] L. Mastroeni and M. Naldi, "Network protection through insurance: Premium computation for the on-off service model," in *8th International Workshop on the Design of Reliable Communication Networks DRCN, Krakow, Poland*, 10-12 October 2011, pp. 46–53.

[25] A. D. Poularikas, *The Handbook of Formulas and Tables for Signal Processing*. CRC Press, 1999, ch. Probability and Stochastic Processes.

[26] F. Beichelt, *Stochastic Processes in Science, Engineering and Finance*. Chapman & Hall/CRC, 2006.

[27] M. Naldi and L. Mastroeni, "Violation of service availability targets in service level agreements," in *Federated Conference on Computer Science and Information Systems - FedCSIS 2011, Szczecin, Poland*, 18-21 September 2011, pp. 537–540.

[28] L. Mastroeni and M. Naldi, "Compensation policies and risk in service level agreements: A value-at-risk approach under the on-off service model," in *Economics of Converged, Internet-Based Networks - 7th International Workshop on Internet Charging and QoS Technologies, ICQT 2011, Paris, France*, ser. Lecture Notes in Computer Science, vol. 6995. Springer, October 24, 2011, pp. 2–13.

[29] P. Cholda, E. L. Følstad, B. E. Helvik, P. Kuusela, M. Naldi, and I. Norros, "Towards risk-aware communications networking," *Rel. Eng. & Sys. Safety*, vol. 109, pp. 160–174, 2013.