Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2010, Article ID 812979, 7 pages
doi:10.1155/2010/812979

Research Article

A Case Study of the Accuracy of SNMP Measurements

Matthew Roughan

School of Mathematical Sciences, University of Adelaide, Adelaide, South Australia 5005, Australia

Correspondence should be addressed to Matthew Roughan, matthew.roughan@adelaide.edu.au

Received 6 November 2009; Accepted 23 February 2010

Academic Editor: Christian Schlegel

For some time it has been known that the standard method for collecting link-traffic measurements in IP networks—the Simple Network Management Protocol or SNMP—is flawed. It has often been noted that SNMP is subject to missing data, and that its measurements contain errors. However, very little work has been aimed at assessing the magnitude of these errors. This paper develops a simple, easily applicable technique for measuring SNMP errors, and uses it in a case study to assess errors in a common SNMP collection tool. The results indicate that most link-load measurement errors are relatively small, but the distribution has a heavy-tail, and that a few measurement errors can be as large as the measurements themselves. The approach also allows us to go some way towards explaining the cause of the errors.

1. Introduction

Calibration of measurements is a key activity in all scientific endeavors, no less so for Internet measurement [1]. Measurements always contain errors. We must estimate their size if we are to have any confidence in the measurements' suitability for a particular task. Precise assessments of errors in measurements are not reported in the Internet measurement literature as often as we might hope (exceptions being [2, 3]), perhaps because it is often difficult to establish ground truth against which to validate.

In this paper, we present a case study considering the accuracy of a set of SNMP (Simple Network Management Protocol) measurements of link loads in the Abilene network [4]. As the name says, SNMP is simple, and so easy to collect, and almost ubiquitous in its use by network management. It allows the collection of data such as the number of bits, and packets to cross an interface over some time intervals, typically every five minutes. In the research literature, it is sometimes seen as a poor source of data compared to others such as flow-level collection. This lack of respect for SNMP arises in part because it can often suffer from artifacts, errors, and missing data. Nevertheless, it is hard to overestimate how useful SNMP data is to network managers, and much work has gone into extending this utility by making it possible to use this data to estimate traffic matrices (e.g., see [5, 6]), and from these detect anomalies (e.g., see [7]). Hence, it is important to consider the accuracy of SNMP measurements.

Most previous consideration of errors in SNMP link-load measurements has been based on arguments unsupported by data. For instance, Zhao et al. [8] argue that SNMP errors are primarily caused by errors in polling timestamps, and present an error model, but data is not used to validate the model or assumption about the cause of errors. Few works have considered the accuracy of the measurements themselves, one exception being [2], which looks at packet loss measurements, not link load which we examine here. Other works assessing SNMP measurements have focussed primarily on testing the efficiency of the protocol in terms of network resources used in measurement. See [9] for a survey.

This paper assesses the accuracy of SNMP link load measurements specifically, and its results indicate that most link-load measurement errors are relatively small, but the distribution has a heavy-tail, and that a few measurement errors can be as large as the measurements themselves. The errors occur in distinct patterns and we explain their cause using this structure.

We do not argue that these results are universally representative, but rather that they show clearly the need for verification of measurements. As noted above it is rare for operators to have ground-truth data against which to compare their SNMP measurements, and hence establish

their accuracy. So the other major contribution of this paper, apart from a publically reported case study of SNMP measurement accuracy, is a methodology that is easy and practical for most network providers to use to establish the accuracy of their measurements. The approach we adopt is to exploit the redundancy available in SNMP measurements to perform self-calibration.

2. Background

In IP networks today, link load measurements are readily available via the Simple Network Management Protocol (SNMP). SNMP is useful because it is supported by most devices in an IP network. The SNMP data that is available on a device is defined in an abstract data structure known as a Management Information Base (MIB). A Network Measurement Station (NMS) periodically requests or *polls* the appropriate SNMP MIB data from a router (or other device). The standard MIBs defined on most routers/switches include a cyclic counter of the number of bytes transmitted and received on each of its interfaces. Hence we can obtain basic traffic statistics for the entire network with little additional infrastructure support—all we need is an SNMP poller that periodically records these counters. However, one should note carefully that SNMP counters (on devices) do not count the number of packets per interval, but only a running total. In order to compute packets per interval, we need to send polls at precise times. A typical polling interval for SNMP is five minutes.

SNMP data has many known limitations. Data may be lost in transit (SNMP uses unreliable UDP transport), or by the NMS, for instance, if the NMS crashes or reboots. Data may be incorrect through poor SNMP agent implementations, or because a counter has wrapped multiple times (this is easier than you might expect as old versions of SNMP used 32-bit counters and these could wrap quite quickly on a high-speed link, e.g., in less than 4 seconds on a 10 Gbps link), counter resets (say after a router reboot), or because the timing of SNMP polls is somewhat hard to control. This "jitter" in poll timing arises because

(i) NMSs must perform polls to many devices, and cannot perform them all concurrently;

(ii) timing on typical commodity hardware is not always very accurate [10];

(iii) SNMP processes on routers and switches are given low-priority and may therefore have a delayed response;

(iv) poll packets may take some time to transit the network.

The net effect is that the time at which we aim to conduct a poll and the actual time of the poll are often offset by some jitter. This problem is compounded in some systems that do not even record when the poll was sent/received at the NMS (let alone the actual time the poll was answered by the network device), but only the intended time of the poll in the polling schedule.

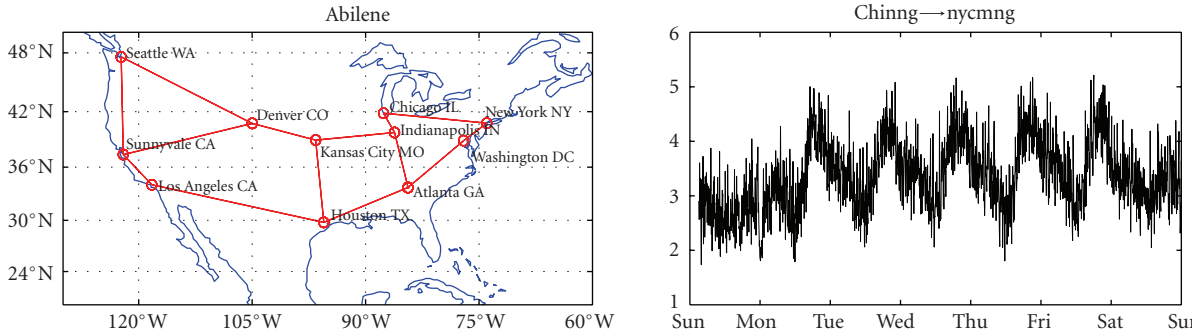
Obviously, the quality of such measurements varies depending on the NMS system, and the SNMP agent implementation on routers or other network devices. Some systems implicitly perform a crude interpolation when reporting the polling times, whereas other systems may make use of proprietary features of certain network devices to improve the accuracy of the timestamps. Other systems attempt to provide reliable transport of polls through retransmission (though this improves reliability at the expense of increasing delays between the desired and actual polling times). However, even where these facilities exist, the question still remains of how accurate the measured timestamps and values are. One should never simply accept that these will be accurate, given the many difficulties of getting timestamps in non-real-time systems [10] without accurate hardware clocks. Moreover, SNMP implementations are often "add-ons," and given little consideration in the original design and architecture of devices, and given low priority in terms of testing and maintenance.

Many network managers assume that the errors in measurements are negligible. However, such assumptions are dangerous because errors can feed into management processes, corrupting the results, resulting in congestion or wasted resources. The size and nature of errors in a set of SNMP measurements will depend on the polling software, the network devices in question, and even the traffic on the network. It is important that ongoing calibration of measurements is a part maintaining quality in a network.

Note that what we propose here is different from compliance testing, such as one might conduct on an SNMP agent [11]. Such compliance testing is necessary, but only shows that an SNMP agent correctly responds to polls, and so forth. An agent can respond "correctly" and still the measurements contain errors such as those due to timing. Likewise, benchmarking and simulation [9] are of little use in this domain because we are interested in the performance of a particular SNMP/NMS system, and the details of a deployment are hard to really capture (e.g., what are the failure rates of the NMS, what are the delays in agent responses for an SNMP agent on a router under realistic traffic and control loads).

The difficulty of calibrating SNMP systems in the field is that the major alternative source of data, flow-level data, is unsuitable for the task because the timing of flows is random (not fixed to the granularity of the SNMP measurements) and hence the datasets are incommensurate. The only (currently) practical source of ground truth data would be a packet trace, and few operators are willing to pay the cost of installation and management of the devices necessary to collect such data from high-speed links.

The alternative proposed here is to use the redundancy already present in many SNMP datasets to self-calibrate the data. More specifically, many operators would collect SNMP data from the interfaces at either end of a substantial set of links in their network. We exploit this redundancy by performing comparisons between measurements from either end of the link to assess errors.



(a) Abilene network. All links are 10 Gbps, except Indianapolis to Atlanta which is 2.5 Gbps

(b) Traffic from Chicago to NY from 5th through 12th of February

Figure 1: Abilene network and example traffic.

3. Data and Methodology

This case study uses SNMP data from the Abilene (Internet2) [4] network over the period Jan 4th-March 28th 2006 (chosen because this is the longest contiguous and public SNMP dataset available to us). During this period Abilene had 14 links (all 10 Gbps except for the link between Atlanta and Indianapolis, which was 2.5 Gbps). The data consists of (nominally) five-minute counts of bytes on each of the 28 interfaces in the network.

The RRD (Round-Robin Database) tool [12] was used to collect and store the data. The RRD tool is a free tool for collecting, storing, and displaying SNMP data. It is anecdotally reported to be used in a very large number of SNMP installations. It has the feature that it collects data in a fixed length window, with only a relatively short period stored at the highest resolution. Longer sequences of averages (with coarse time resolution) are created and stored automatically, allowing for a natural multiresolution view of the data that has proved useful for network operators. For our purposes, we wished to examine the high-resolution data over a long time window, so RRD files were collected at daily intervals, and the five-minute data extracted to create a nearly 3-month long dataset. To be precise, the dataset consists of two $28 \times 23{,}498$ arrays of measurements. The first has 23,498 five-minute interval records measuring the number of bytes transmitted out of each of the 28 interfaces on the network, and the second array measures the number of bytes transmitted into each interface.

Figure 1 shows the Abilene network (during the measurement period) and an example of traffic data over the period from the 5th through 12th of February, 2006. We can see in the traffic a not atypical daily and weekly cycle, often present in traffic, over which are superposed random fluctuations.

We denote the instantaneous traffic rate at time t by $x(t)$, the traffic over the interval $[t, t+s)$ by $y(t, t+s)$, and the observed traffic over the interval by $\hat{y}(t, t+s)$, that is,

$$y(t, t+s) = \int_t^{t+s} x(t)dt,$$
$$\hat{y}(t, t+s) = y(t, t+s) + \epsilon(t, t+s), \tag{1}$$

where $\epsilon(t, t+s)$ is the measurement error. Operators are generally concerned with relative errors, that is, $r = \epsilon/y$.

3.1. Estimation of Errors. We study errors by comparing the data from each end of a link. The total traffic entering a link during some time interval should be almost the same as the traffic departing during the same interval. The difference is the traffic that is *on-the-wire* at the start and end of the interval.

We can calculate the maximum traffic on-the-wire for a link—it is simply the bandwidth-delay product. On the Abilene network, the maximum bandwidth-delay product is 109.8 Mbits on the `Houston` → `Los Angeles` link. Relative to the typical 5-minute traffic on this link, this is only around 1 part in a million. Moreover, the difference resulting from traffic on-the-wire at the start and end of the interval will (in almost all cases) be much smaller than this, firstly because it is the difference of two values (that are likely to be close due to their proximity in time) and secondly because many links are underutilized much of the time.

Therefore, the errors due to on-the-wire traffic are negligible, and we can attribute any significant differences to errors in the SNMP data collection itself. The difference between the traffic as measured at each end of a link will be

$$\hat{y}^{(1)}(t, t+s) - \hat{y}^{(2)}(t, t+s) = \epsilon^{(1)} - \epsilon^{(2)}, \tag{2}$$

where the superscript refers to the two ends of the link. So we observe the difference of two measurement errors. This combination of two errors could combine to make a bigger error, or cancel, but the average (expected) error is given by

$$E\left[\epsilon^{(1)} - \epsilon^{(2)}\right] = E\left[\epsilon^{(1)}\right] - E\left[\epsilon^{(2)}\right]. \tag{3}$$

We have no reason to presuppose that errors are different at any point in the network, so we assume that the average error at each point is constant. Hence, $E[\epsilon^{(1)} - \epsilon^{(2)}] = 0$, that is, the mean of the observed differences will be zero. Therefore, we cannot (easily) observe bias in our measurements, for instance as caused by a constant offset in the timing of polls as compared to estimated times of polls. This issue cannot

be resolved from SNMP data alone. We need either ground-truth traffic data or at least, we need a good clock against which to check timing.

Variances can likewise be estimated from simple statistical properties providing that errors at different locations are independent[1], in which case,

$$\mathrm{Var}\left(\epsilon^{(1)} - \epsilon^{(2)}\right) = \mathrm{Var}\left(\epsilon^{(1)}\right) + \mathrm{Var}\left(\epsilon^{(2)}\right). \tag{4}$$

Under the homogeneity assumption, the variance of the observed errors will be twice the variance of the error at a single interface.

3.2. Source of Errors. It is interesting to consider what causes errors in SNMP measurements. There are two major candidates: firstly some type of instrumentation error, for example, failure to update counters correctly, and secondly unaccounted jitter in poll timing.

We can go some way towards testing which is the major factor by considering how each of these errors appears in our data. It is typical to assume that instrumentation errors are uncorrelated over time. Hence, if we aggregate a series of measurements, we would expect these errors to follow the Central Limit Theorem. More precisely, if we were to aggregate a series of measurements $\hat{y}(t_1, t_2), \hat{y}(t_2, t_3), \ldots, \hat{y}(t_n, t_{n+1})$ over equally spaced intervals (t_i, t_{i+1}) to get $\hat{y}(t_1, t_{n+1})$, then the error in the aggregate would be likewise aggregated and therefore, assuming independence, and stationarity, the variance of the error in the aggregate would be

$$\mathrm{Var}(\epsilon(t_1, t_{n+1})) = \sum_{i=1}^{n} \mathrm{Var}(\epsilon(t_i, t_{i+1})) = n\,\mathrm{Var}(\epsilon(t_1, t_2)). \tag{5}$$

The total traffic over the interval increases roughly in proportion to the length of the interval, so the variance of the relative error $r(t_1, t_{n+1})$ over this interval would decrease by approximately $1/n$, and the standard deviation by $1/\sqrt{n}$.

On the other hand, errors from timing jitter arise because of errors in integral end-points (say δ_1 and δ_2 at the start and end, resp.), so

$$\begin{aligned}\hat{y}(t, t+s) &= \int_{t+\delta_1}^{t+s+\delta_2} x(t)dt \\ &= y(t, t+s) - \int_{t}^{t+\delta_1} x(t)dt + \int_{t+s}^{t+s+\delta_2} x(t)dt.\end{aligned} \tag{6}$$

However, when we aggregate, the errors from the incorrect end-points of the integral concertina so that the final error in $\hat{y}(t_1, t_{n+1})$ is of the same order of magnitude as the individual errors in each measurement. Hence, the variance of relative errors decreases approximately as $1/n^2$, and the standard deviation as $1/n$.

We should be able to see these different behaviors:

(i) $1/\sqrt{n}$ error standard deviation with aggregation level n, for instrumentation errors at the SNMP agent;

(ii) $1/n$ error standard deviation with aggregation level n, for poll-time jitter errors.

and from these determine the major source of the errors in this set of SNMP data.

Table 1: Error sizes: "missing" refers to the total number of missing data points from either end of the link (out of 23,498 entries). M and SD are the mean and standard deviations of the relative error calculated as described in the text.

Source → Destination	Missing	M	SD
Atlanta → Houston	8	−4.9e−05	0.0054
Houston → Atlanta	1	2.9e−05	0.0049
Atlanta → Indianapolis	178	−1.5e−05	0.0065
Indianapolis → Atlanta	173	−3.8e−05	0.0066
Atlanta → Washington	5	−3e−05	0.0054
Washington → Atlanta	1	−3.4e−05	0.0065
Chicago → Indianapolis	3	−6e−05	0.0060
Indianapolis → Chicago	3	2.8e−05	0.0056
Chicago → New York	2	1.6e−05	0.0039
New York → Chicago	7	−3.3e−05	0.0041
Denver → Kansas City	5	−3.9e−05	0.0061
Kansas City → Denver	5	1.8e−05	0.0034
Denver → Sunnyvale	5	−7.9e−06	0.0041
Sunnyvale → Denver	6	−6.1e−06	0.0054
Denver → Seattle	5	6.3e−06	0.0048
Seattle → Denver	5	−2.9e−05	0.0049
Houston → Kansas City	5	−1.3e−05	0.0072
Kansas City → Houston	5	−3.3e−05	0.0063
Houston → Los Angeles	5	7.6e−05	0.0078
Los Angeles → Houston	5	−3.9e−05	0.0053
Indianapolis → Kansas City	4	4.3e−05	0.0048
Kansas City → Indianapolis	5	−3.6e−05	0.0054
Los Angeles → Sunnyvale	5	2.1e−05	0.0040
Sunnyvale → Los Angeles	5	−3.7e−05	0.0043
New York → Washington	6	−4.7e−05	0.0066
Washington → New York	6	4.3e−06	0.0038
Sunnyvale → Seattle	4	−1.1e−06	0.0066
Seattle → Sunnyvale	4	3.1e−05	0.0054

4. Results

The main results of the case study are reported in Table 1. We measure the errors using the difference between traffic estimates at each end of the link, and then take this value relative to the average traffic on the link. Ideally, we would compute the value relative to the true average, but we do not know this. However, as noted above, as we aggregate traffic over longer intervals, we should expect that errors decline, and so the average traffic across the complete set of measurements will have only a small error (at least compared to that in each interval), and so it serves as a proxy for the correct value at each time point. We then compute the variance, and divide by two (to account for errors from both ends of the link). We report the standard deviation (SD) because this value is easier to interpret.

Mathematically, the reported value is

$$\mathrm{SD} = \sqrt{\frac{\mathrm{Var}(r)}{2}}, \tag{7}$$

where we estimate the mean and variance of the relative errors r by the estimators

$$M \simeq \frac{1}{n}\sum_{i=1}^{n} r = \frac{1}{n}\sum_{i=1}^{n} \frac{\hat{y}^{(1)}(t_i, t_{i+1}) - \hat{y}^{(2)}(t_i, t_{i+1})}{\overline{y}},$$

$$\mathrm{Var}[r] \simeq \frac{1}{n-1}\sum_{i=1}^{n}\left[\frac{\hat{y}^{(1)}(t_i, t_{i+1}) - \hat{y}^{(2)}(t_i, t_{i+1})}{\overline{y}} - M\right]^2, \tag{8}$$

where $\hat{y}^{(1)}$ and $\hat{y}^{(2)}$ refer to measurements from either end of the link, and $\overline{y}$ is the average traffic on the link. The mean relative errors M are shown in Table 1, and are almost negligible as expected. The SDs lie between 0.0038 and 0.0078, with an average of 0.0054 representing relative errors around 0.5%. Results are discussed in more detail below.

4.1. Missing Data. The number of missing data from each link is shown in Table 1. The numbers, when compared with the overall number of data points (23,498), are very small (<0.8%). However, it is interesting to consider the structure of these losses—they do not occur completely at random. Figure 2(a) shows a time-event plot with "∘" showing the location of the missing data. There are two prominent features.

(1) *vertical structure:* many of the missing data are correlated (in time) across all, or a substantial subset of the interfaces. This can represent a problem in the NMS, resulting in data loss across multiple sources.

(2) *burstiness:* although the loss rate is low, losses tend to occur in bursts; see particularly the bursts in mid-February on the `Atlanta-Indianapolis` link. The fact that this burst occurs on the only low speed (2.5 Gbps) link in the network is suggestive of the problem being related to capacity constraints on the interface cards, perhaps because the measurements were deprioritised while the link was heavily utilized, or possibly because polls were dropped in transmission across the link when it was congested.

4.2. Error Magnitude. Table 1 shows the SD of the relative errors on each link. It is also interesting to consider the distribution of these errors, which we plot in Figure 2(b) across the whole network. This figure shows the CCDF (Complementary Cumulative Distribution Function) of the absolute value of the relative errors. The distribution is heavy-tailed (though does not follow a simple power-law type distribution). A very large proportion of relative errors are small (approximately 90% are less than 1% in size) but there are some errors as large as 100%. This is a critical insight, though it is not new. Anecdotally, this type of phenomena has been reported before, and may arise from poor implementations of counters, or counter-rewraps/resets. It is one of the underlying motivations for 95th percentile billing [13]. In 95th percentile billing, the goal is to bill based on the peak rate of utilization, but it is assumed that SNMP measurements are flawed, and so the more robust 95th percentile is used as a proxy measure of the peak utilization. Our measurements support this view of the errors, in that a small number or errors are large, certainly large enough to affect billing.

Spike-like errors also have distinct implications for anomaly detection algorithms [7], which are often aimed at detecting sudden changes in traffic. Spikes caused by measurement artifacts such as we observe here will also register as anomalies unless care is taken to eliminate such artifacts from the data.

Figure 2(b) also shows a crude simulation of the errors, created by using a mixture of an exponential distribution (for the smaller errors) and a Pareto distribution (for the larger errors). These were chosen as the simplest distributions that show the required properties in the regimes of interest. The simulated data is generated using a mixture model where the error is drawn from an exponential distribution with mean 0.0035 and probability 0.99882 of selection, and a Pareto distribution with cumulative distribution function

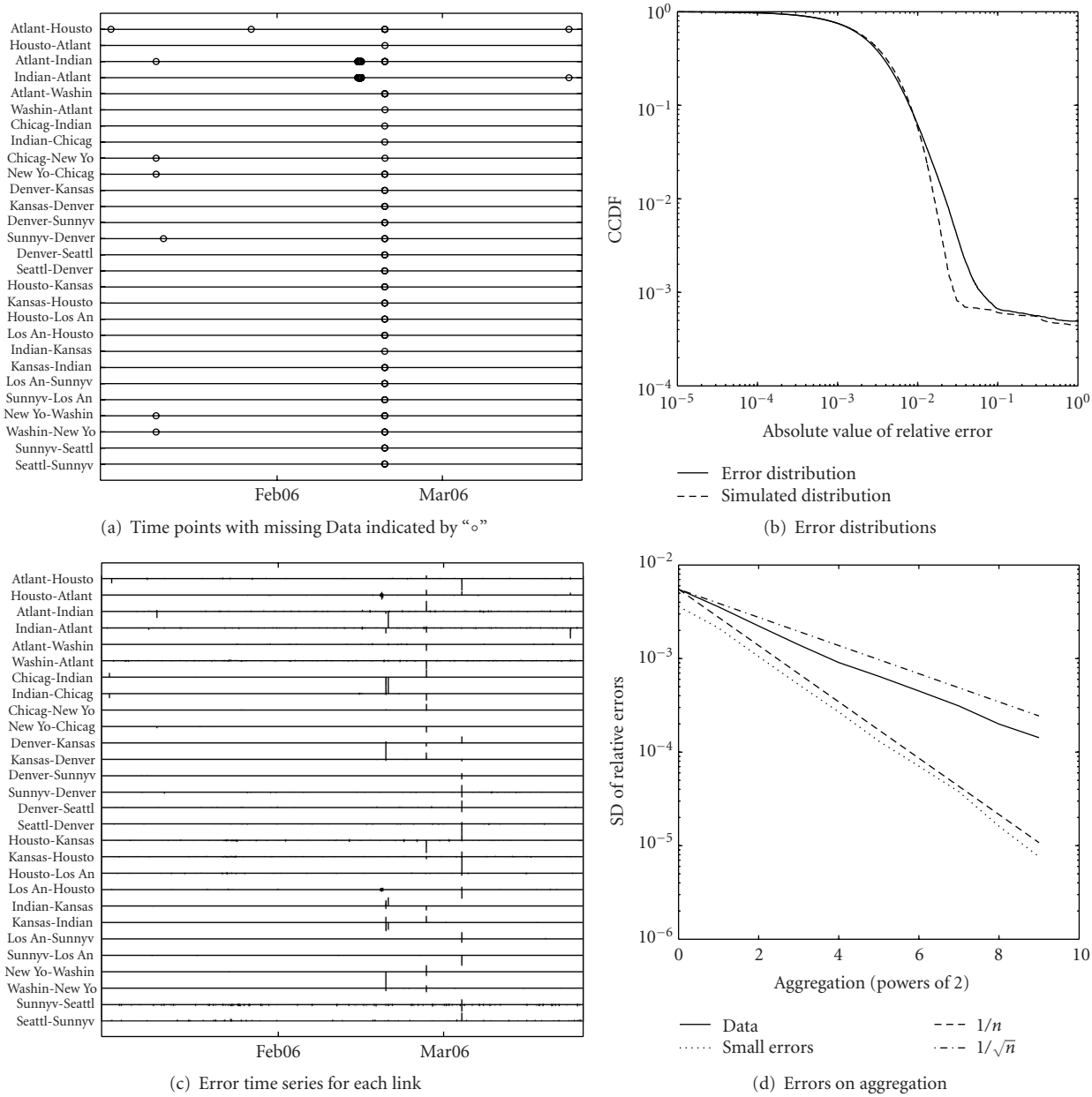
$$F(x) = 1 - \left(\frac{b}{x}\right)^{\alpha}, \tag{9}$$

with probability of selection of 0.00118, and parameters $\alpha = 0.12$ and $b = 0.0005$.

The simulation fits the real distribution at either end quite well, though there is some deviation around the transition, which is too sharp in the simulated data. Perhaps there is a third class of error that occurs in this region, though it is hard to separate these cleanly from the other two types of errors.

In any case, the fitting itself is not the most interesting feature. The feature that is perhaps most important is that the parameters of the Pareto distribution used here fall in the set of cases where the mean is infinite (at least in theory though in reality it would be truncated). In these cases, it is common to observe slow convergence of estimators for means and variances [14], and hence the need in this paper to analyze such a long set of data. Essentially, we need a long enough dataset to see the rare (but very large) error events that occur occasionally. The need for long datasets in this type of analysis is another factor that makes calibrating measurements harder than one might naively expect.

4.3. Source of Errors. Figure 2(c) shows the magnitude of the errors on each link as a function of time. We can see that at least one burst of errors is associated closely with the missing data in mid-February (see Figure 2(a)). When an SNMP poller restarts, it is possible that it loses track of a counter, resulting in a large error in the estimate of traffic in the interval following restart. Even if the SNMP poller maintains its state, some pollers are not smart enough to realize that previous data is missing, and hence miscalculate the average rate over the actual measurement interval (ignoring the



(a) Time points with missing Data indicated by "∘"

(b) Error distributions

(c) Error time series for each link

(d) Errors on aggregation

FIGURE 2: SNMP errors and missing Data.

missing polls). This can explain one errored measurement interval.

When we more closely examine the time series of measurements, we do indeed see errors in intervals following the missing data, but we also observe subsequent errors, which we cannot explain. Likewise, there are other bursts of measurement errors in the data shown in Figure 2(c) that cannot be explained by missing data. However, they could still be the result of a restart of the NMS, which occurs within a single measurement interval, avoiding missing any polls, but still resetting counters. Such resets are detectable in some systems (the value of the counter drops to nearly zero), and a well-designed NMS may elide these data points, though this results in additional missing data.

So what of poll-timing errors? In Figure 2(b), we see that the distribution is apparently a mixture of two main types of errors: small (perhaps exponentially distributed) errors with magnitude less then about 1%, and large (approximately Pareto) errors, with magnitude greater than 10%. We will attempt to determine the major cause of these types of errors through aggregation.

Figure 2(d) shows the effect of aggregation on the data, on a log-log graph, as well as a $1/n$ and $1/\sqrt{n}$ curves. The curve labelled "data" is based on aggregations of the whole time series; whereas the curve labelled "small errors" is based on a subset of data from the middle sixth of the sequence where errors are all small. The pattern we see in the results is obvious. When we use the complete dataset, including

the large errors, the data more closely follows the the $1/\sqrt{n}$ curve, at least asymptotically. As discussed above, this is an indication that jitter is not the major source of error in these measurements. On the other hand, when we restrict our attention to portions of the data with smaller errors, then the curve very closely follows the $1/n$ curve.

The implication is that the larger errors (above 10% say) tend to be caused by singular problems in data collection, which are consistent with Figure 2(c) which shows that these errors are correlated with missing data. On the other hand, the smaller errors (below 1%) are likely to be caused by timestamp jitter. Converting the magnitude of the errors into a crude estimate of the size of the timestamp jitter, we might expect such jitter to be below 3 seconds in size (1% of the time interval).

Obviously, it would be ideal to verify the root cause of the errors using direct "ground-truth" measurements, but a key point of this article is that operators rarely have such measurements. They are expensive to obtain, and require their own calibration experiments before they can be used as a benchmark. In the mean time, the above analysis provides some insight into probable causes.

5. Conclusion

This paper presents a case study of the errors in SNMP link traffic measurements. The results indicate that most measurement errors are small, but the distribution has a heavy (Pareto-like) tail, and that a few measurement errors can be as large as the measurements themselves. Correlations in errors across links suggests that the major cause of these is problems in the NMS, most likely because of counter resets.

The fact that the majority of large errors seem to be related to problems in the NMS suggest that a long-term strategy should be to make this more reliable. This is often achieved through using multiple redundant servers. However, we should note that this can be expensive, complicated to get right, and the current reliability may be sufficient for many tasks. For instance, in the system above, there appear (from Figure 2(c)) to be no more than three significant problems over the course of nearly three months. Many network operators would be sanguine about such a level of error in a measurement system (if not in their network itself).

However, we do not argue that these results are representative, as the type and magnitude of errors in any system depends on its details. The errors could be much larger, but network operators often assume that such errors are negligible, or deal with them through crude rules of thumb. However, calibration of instruments is a basic scientific tenet, and so such studies should be a basic requirement in all NMS installations. The key contribution of this paper is a simple, almost cost-free technique to perform such calibration. It is so easy, it could even be performed continuously to provide an ongoing test of the validatity of link load measurements.

Acknowledgments

This work was supported in part by the Australian Research Council through DP0665427. The author would also like to thank Abilene for making data available, and to Andrew Coyle for valuable discussion of the approach described here.

Endnotes

1. We see that this assumption can be false, but the variance is nevertheless a useful indicator of the relative size of the errors.

References

[1] V. Paxson, "Strategies for sound internet measurement," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC '04)*, pp. 263–271, Taormina, Italy, October 2004.

[2] P. Barford and J. Sommers, "Comparing probe- and router-based packet-loss measurement," *IEEE Internet Computing*, vol. 8, no. 5, pp. 50–56, 2004.

[3] J. Sommers, P. Barford, and W. Willinger, "A proposed framework for calibration of available bandwidth estimation tools," in *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*, pp. 709–718, Cagliari, Italy, June 2006.

[4] Abilene/Internet2, http://www.internet2.edu/.

[5] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 206–217, San Diego, Calif, USA, June 2003.

[6] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of the ACM SIGCOMM Conference on Computer Communications*, pp. 301–312, Karlsruhe, Germany, August 2003.

[7] Y. Zhang, Z. Ge, M. Roughan, and A. Greenberg, "Network anomography," in *Proceedings of the Internet Measurement Conference (IMC '05)*, Berkeley, Calif, USA, October 2005.

[8] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: making use of multiple data sources," *Performance Evaluation Review*, vol. 34, no. 1, pp. 133–144, 2006.

[9] L. Andrey, O. Festor, A. Lahmadi, A. Pras, and J. Schönwälder, "Survey of SNMP performance analysis studies," *International Journal of Network Management*, vol. 19, no. 6, pp. 527–548, 2009.

[10] A. Pásztor and D. Veitch, "PC based precision timing without GPS," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 1–10, Marina Del Rey, Calif, USA, June 2002.

[11] "Silvercreek, the SNMP test suite," http://silvercreek.iwl.com/.

[12] T. Oetiker, "RRDtool," http://oss.oetiker.ch/rrdtool/.

[13] X. Dimitropoulos, P. Hurley, A. Kind, and M. P. Stoecklin, "On the 95-percentile billing method," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement (PAM '09)*, vol. 5448 of *Lecture Notes in Computer Science*, pp. 207–216, Springer, Seoul, South Korea, April 2009.

[14] M. E. Crovella and L. Lipsky, "Long-lasting transient conditions in simulations with heavy-tailed workloads," in *Proceedings of the Winter Simulation Conference*, pp. 1005–1012, Atlanta, Ga, USA, December 1997.