Toward Trusted Time: Remote Server Vetting and the Misfiring Heart of Internet Timing

Yi Cao, Darryl Veitch, *Fellow, IEEE*

***Abstract*—The core of the Internet's timekeeping system are the Stratum-1 timeservers, those connected to reference hardware, that anchor the server hierarchy. It is essential that these root servers are accurate and reliable, and this it is typically taken as a given. We examine this premise through an examination of 102 prominent Stratum-1 servers, using 3 datasets spanning 6 years, collected in reference testbeds with authoritative timestamping. We describe a methodology capable of rigorously removing congestion related variability, allowing server errors to be unambiguously revealed. We use the data and methodology to assess the health of public network timing, and how it varies over time, by reporting on the type, severity, duration, and prevalence of server errors, and how they relate to protocol level information. We present conclusive evidence that the system has problems. We find that errors are widespread, significant, often endemic, consistent over time, and typically come with no warning at the protocol level. Our results highlight the lack of oversight in the current system, and provides the foundation of a server health monitoring capability, necessary to restore and maintain trust in network timing. We describe three specific applications where our results can have an impact. Our data, detailed results and software are publically available.**

***Index Terms*—NTP, clock synchronization, stratum-1 time servers, public infrastructure, clock errors.**

I. Introduction

Timekeeping is a vital service provided by computer operating systems. An operating system's *system clock* is software built on local hardware counters, which is synchronized through timestamp exchange to a reference timeserver over the network. For a high proportion of the global computer population, this means synchronization to a public timeserver accessed by the Network Time Protocol (NTP) [1], [2].

Anchoring network timing are the *Stratum-1* timeservers, so named as they have local access to reference hardware such as GPS or atomic clocks. The Stratum-1 servers are relatively few in number, typically hosted by National Laboratories responsible for reference time, research institutes or universities. Each is a root of a server hierarchy, where *Stratum-*s timeservers, $s>1$, themselves synchronize to *Stratum-*$(s-1)$ servers over the network.

As the foundation of a distributed timing system, the Stratum-1 servers are assumed to be highly reliable and accurate. However, if this assumption failed to hold, how could we know? On the system side, Stratum-1 servers act essentially as independent islands, without any mechanism for cross-validation, nor are there tools or infrastructure available for independent remote monitoring and assessment. From the client side perspective, judging its own server is inherently problematic. Not only is the client's clock designed to trust and follow that of its server, the transported reference timestamps are viewed through the shroud of network latency. This is typically highly variable, and anything from tens of microseconds to hundreds of milliseconds and beyond in size. This may dwarf the errors in a Stratum-1 server, which are *expected* to be tens of microseconds at most.

It follows from the above that, even if one were looking for them, discrepancies between the client and server clocks could easily be attributed to network latency, and/or shortcomings of the client synchronization daemon/algorithm. In fact, suspicion is unlikely to be directed toward a Stratum-1 server unless it is simply unreachable, or the error gross, for examples several seconds, minutes or more.

This paper helps address the many open needs of time server health monitoring and evaluation, necessary to establish trust in network timing. The first contribution is a longitudinal study of timing errors in a set of 102 influential Stratum-1 timeservers over a period of almost 6 years. The study is based on detailed data sets from three experiments, each months long, collected in 2011, 2014-15, and 2016-17 (see Table I) using reference testbeds. We perform a detailed examination of timing errors within these data sets, characterising error frequency, size and type. Our analysis expands on that of our earlier work [3] based on the first two experiments, in particular by including a third data set of much higher resolution, redoing the analysis on all datasets with improved tools and comprehensive reporting, and providing far more complete results.

The second contribution is an improved methodology, building on that from [3], [4], for remote server error measurement, that greatly improves the ability to filter congestion effects from server errors, resulting in more accurate measurement both of errors and underlying path asymmetry, and reduces the reliance on expert manual decision making. We developed a graphical visualization and analysis tool to accelerate the manual localization and confirmation of errors, and to automate, using the above methodology, the actual taking and recording of error measurements in a form that enables use as a labelled dataset. Both the tool, and the labelled dataset, are available at [5]. As one indication of the sensitivity and accuracy of the methodology, Section V-D reports on anomalies which were correctly diagnosed as **not** being due to server error, despite having signatures which differed from those of server errors by only $1\,\mathrm{ms}$, and being located in Europe, across the

Cao is with the National Institute for Applied Statistics Research Australia.
Veitch is with the School of Electrical and Data Engineering, University of Technology Sydney, Australia.

planet from the testbed monitor. This diagnosis was confirmed through communication with network administrators.

The third contribution is to take advantage of the new high resolution dataset, graphical tool and methodology, to examine Stratum-1 server errors in more detail and over new dimensions than previous work. We examine error sizes and durations jointly, and show how they vary both within and across servers. We also examine in detail relevant protocol aspects, such as the occurrence of NTP packets whose Stratum field set to 0, indicating an unsynchronized clock, and examine their relationship to empirical server errors. One of the key findings is that in the majority of cases, server errors are not presaged or flagged by any such protocol signs.

The paper is organised as follows. Section II describes the selection of servers, the testbed, the experiment and its data. Section III outlines the principles, challenges and methodology of our server error analysis. Section IV details and discusses the finding for the longitudinal study. Section V dives into the new data set in more detail, both for server error and protocol aspects. Section VI describes prior work, and we conclude in Section VII with a discussion on implications and applications. This work does not raise any ethical issues.

II. The Data

The data was gathered in three experiments we conducted spaced years apart, summarised in Table I. In each, the main source of the servers used is '**ListOrg**', a list of public IPv4 Stratum-1 (S1) time servers maintained at *ntp.org* by the NTP Project. This is a well known resource that includes many servers from National Laboratories. As an indication of its influence, 50% of the **ListOrg** servers (99 of 197 at Dec. 2016) were included in the NTP Pool Project [6], a widely used load balancing and timeserver selection service incorporating some 4000 S1 and S2 servers supporting millions of clients. In the recent survey of the NTP eco-system [7], it is reported (Figure 6, section 4.4.1) that the *ntp.org* members of the Pool have disproportionate influence, and better performance on average.

In 2011 experiment **Exp1** targeted a server list **List1**, a subset of 89 **ListOrg** servers that responded to queries at that time. In 2014 **Exp2** used **List2**, which updated **List1** by removing servers that were no longer in **ListOrg**, or which no longer responded, augmented by a set, **Au**, of 13 Australian S1 servers. These included all the Australian public S1 servers at that time, as well as 4 servers from the Australia's National Laboratory (the NMI), which are not public. We define **ListLong**(itudinal) as those servers in (**List1** $\cap$ **List2**) $\cup$ **Au** which returned useful data, some 102 servers in total. The data for this list was first studied in [3].

In 2016 experiment **Exp3** targeted **List3**, a much larger server list. The dataset, first studied in [4], is publicly available at [5]. As a superset of **ListLong**, we use it to enable a longitudinal analysis across all three experiments spanning almost 6 years. We define **ListNew** to be the 91 servers, out of **ListLong**'s 102, that responded with useful data in **Exp3**.

Exp	Servers (#)	Start	End	duration
Exp1	**List1** (119)	Mar. 4, 2011	Aug. 3, 2011	151 days
Exp2	**List2** (117)	Dec. 5, 2014	April 2, 2015	124 days
Exp3	**List3** (479)	Nov. 29, 2016	Feb. 2, 2017	64 days

TABLE I

In summary, the results below for **Exp1**, **Exp2**, **Exp3**, cover servers in **ListLong****Au**, **ListLong**, and **ListNew** respectively. To compactly identify the servers, we adopt the nested lexicographical ordering of **ListLong** from [4] based on country code, then url, providing a unique key in the range 1–102. The mapping to server url's is provided in the tables in Section IV.

For an NTP packet i that successfully completes its trip from the host to the server and back, we obtain the ordered 4-tuple $\{T_{a,i}, T_{b,i}, T_{e,i}, T_{f,i}\}$ of timestamps, called a *stamp*. Here $T_{b,i}, T_{e,i}$ are the (incoming and outgoing respectively) UTC timestamps made by the server, extracted from the returning NTP packet header. The timestamps $T_{a,i}, T_{f,i}$ are taken within the testbed. Since the servers are Stratum-1, the roots of the Internet's timing system, an authoritative evaluation demands a reference quality timestamping methodology.

The timestamps $T_{a,i}, T_{f,i}$ are of passively tapped copies of the timing packets that respectively exit and enter the host, taken by a precision capture card. The card's hardware clock is synchronised using a Pulse-Per-Second (PPS) output of an atomic clock, itself locked to the PPS output of a roof mounted GPS receiver. The host timestamps are not used.

Experiments **Exp1** and **Exp2** used (tap, card, atomic, GPS) = (100Mbps hub, Endace DAG3.7GP, SRS PRS10, Trimble Acutime 2000) in a testbed at the University of Melbourne for a timestamping accuracy below 200ns (see [3]). **Exp3** was performed within an improved testbed at the University of Technology Sydney, and used (GigEth NetOptics IX-TP-CU3 tap, Endace DAG 7.5G4, SRS FS725, Trimble Acutime™GG) for a 3-sigma accuracy of 100 ns, and a quantization resolution of 7.5 ns (see [4]).

In each experiment a host launched independent instances of a RADclock [8] NTP request–response exchange daemon, one for each server in its target list in parallel, with all packets timestamped and captured by the DAG card. The generated dataset consists, for each instance/server, of the sequence of well-matched stamps, and for **Exp2** and **Exp3** (but not for **Exp1** due to subsequent hardware data loss) the corresponding two-bit Leap Indicator (LI) and 8-bit server Stratum fields, extracted from the returning NTP response headers.

In **Exp1** and **Exp2** each instance used a polling period of 64 seconds, however, data loss resulted in available data for **Exp1** having a period which is only 256s for most servers. In **Exp3** a value of 1 second was targeted. A prior calibration run determined the smallest period out of $\tau = \{1, 4, 16, 64, 256, 1024\}$ seconds actually accepted by each server, resulting in $\{74, 5, 12, 0, 0, 0\}$ daemons using the above periods respectively for servers in **ListNew**.

The end-2016 leap second [9] occurred within **Exp3**. Our goal is to examine generic server health issues, not those specific to leap seconds, which we have already studied in detail in [4]. From [4], [5] we know, for each server in **List3** $\supset$ **ListNew**, the interval about the leap second, if any, where packets carry leap-affected server timestamps. For the

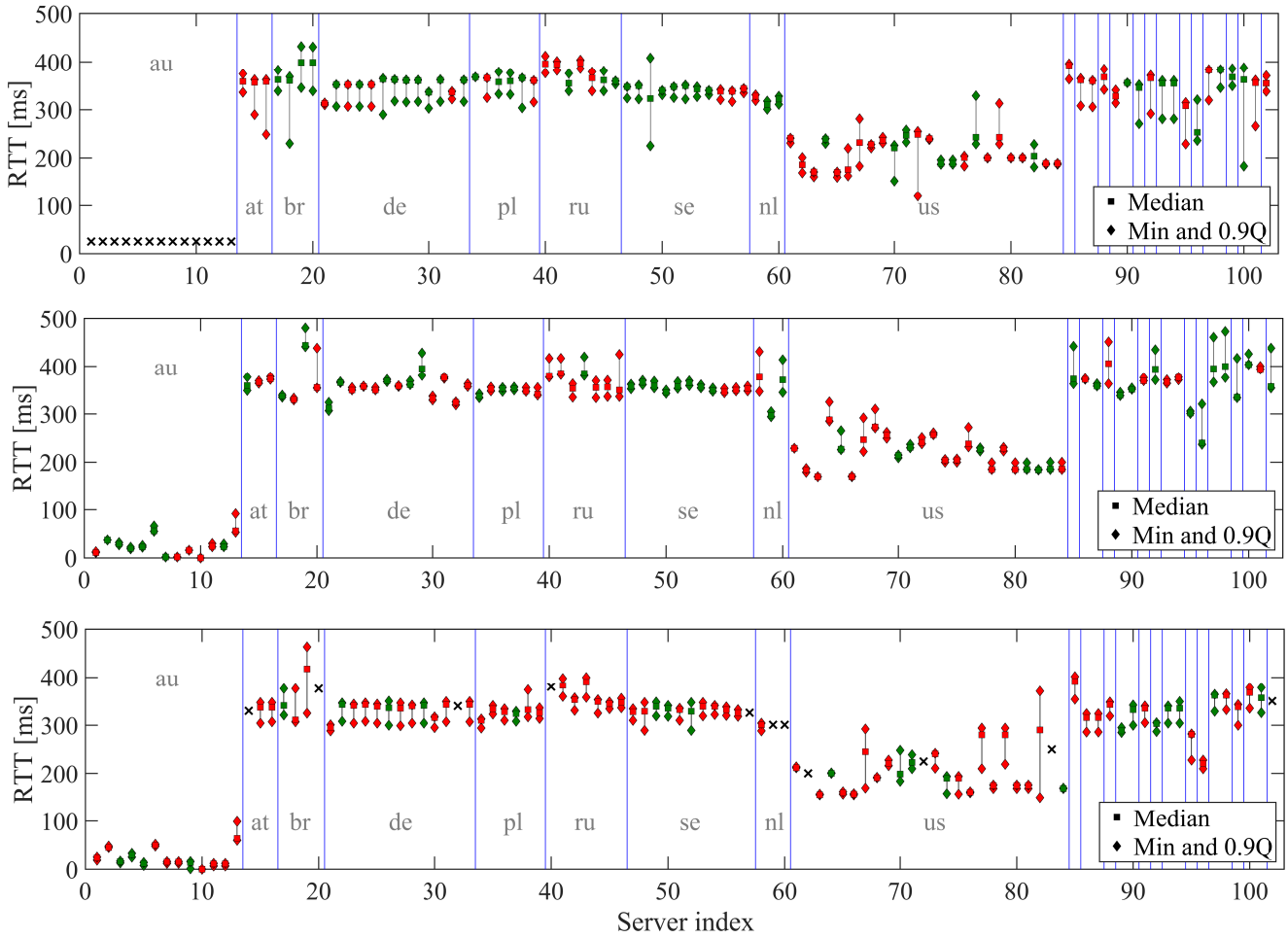

Fig. 1. RTT statistics and classification labels indexed according to the 102 **ListLong** server ordering, from top to bottom **Exp1**, **Exp2** and **Exp3**. In this ordering, countries with 3 or more servers are first, followed by: {am(1),be(2),bg(1),cz(2),fr(1),ie(1),it(2),jp(1),mx(1),ro(2),si(1),es(2),ua(1)}. The color coding follows the server detection classification: Red (Errored), Green (Good). For **Exp1** and **Exp3**, crosses mark servers with no data, namely the au servers for **Exp1**, and servers $\{14, 20, 32, 40, 57, 59, 60, 62, 72, 83, 102\}$ for **Exp3**.

17 servers in **ListNew** that are affected, this interval has been excised from the data to simplify analysis.

One of the most important parameters impacting our experiments is the client–server RTT, and its variability. Figure 1 sets the scene by summarizing the RTTs observed for each server over each dataset. The minimum RTT shown is often significantly lower than the median, a reflection of route change variability as we see below. Whereas the 90% percentiles shown all fall below 500ms, the maximum RTT (not shown) is in many cases of the order of several seconds.

III. Server Error Methodology

In this section we establish the principles through which server errors can be detected, and describe our operational methodology and the basis of server error measurement. Our methodology, which we apply to each of **Exp1**, **Exp2**, and **Exp3** data below, is inspired by, yet distinct from, that of [3]. In particular, the rigorous congestion removal and defined estimators of Section III-B are new, as is the tool described in Section III-C.

A. Principles

From the measured 4-tuples $\{T_{a,i}, T_{b,i}, T_{e,i}, T_{f,i}\}$, five essential estimated path quantities can be defined and modelled:

$$\begin{aligned}
\text{Forward delay} &: \; D_i^{\uparrow} \equiv T_{b,i} - T_{a,i} = \underline{d}^{\uparrow} + q_i^{\uparrow} + \mathbf{e_i} \\
\text{Server delay} &: \; D_i^{\rightarrow} \equiv T_{e,i} - T_{b,i} = \underline{d}^{\rightarrow} + q_i^{\rightarrow} \\
\text{Backward delay} &: \; D_i^{\downarrow} \equiv T_{f,i} - T_{e,i} = \underline{d}^{\downarrow} + q_i^{\downarrow} - \mathbf{e_i} \\
\text{Round Trip Time (RTT)} &: \; R_i \equiv T_{f,i} - T_{a,i} = \underline{r} + q_i \\
\text{Path Asymmetry} &: \; A_i \equiv D_i^{\uparrow} - D_i^{\downarrow} = \underline{a} + a_i + 2\mathbf{e_i} \; .
\end{aligned}$$

The first four time series can be modelled as positive minima $\underline{d}^{\uparrow}$, $\underline{d}^{\rightarrow}$, $\underline{d}^{\downarrow}$, and $\underline{r} = \underline{d}^{\uparrow} + \underline{d}^{\rightarrow} + \underline{d}^{\downarrow} > 0$, to which are added the non-negative terms $q_i^{\uparrow}$, $q_i^{\rightarrow}$, $q_i^{\downarrow}$ and $q_i = q_i^{\uparrow} + q_i^{\rightarrow} + q_i^{\downarrow}$, representing random congestion variability along the path. The term $\mathbf{e_i} \in \mathcal{R}$ represents an error in the server timestamps in stamp i. The final *asymmetry* series A_i has an underlying value $\underline{a} = \underline{d}^{\uparrow} - \underline{d}^{\downarrow} \in (-\underline{r}, \underline{r})$, and a real valued variability $a_i = q_i^{\uparrow} - q_i^{\downarrow}$ due to congestion, and carries an error of $2\mathbf{e_i}$. The values $\underline{d}^{\uparrow}$, $\underline{d}^{\downarrow}$, $\underline{r}$, $\underline{a}$ are not in fact constant, but undergo step changes due to changes in routing. The resulting piecewise-constant functions we call *baselines* in the case of the positive $\underline{d}^{\uparrow}$, $\underline{d}^{\downarrow}$ and $\underline{r}$, and the *underlying asymmetry* in the case of $\underline{a}$.

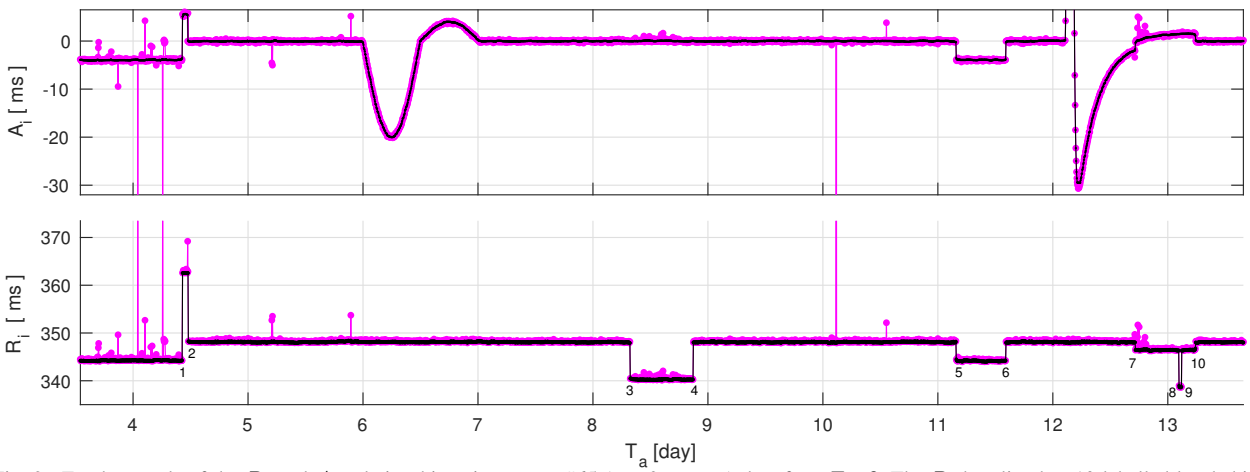

Fig. 2. Fundamentals of the R_i and A_i relationship using server #65 (*time2.stupi.se*) data from **Exp3**. The R_i baseline has 10 labelled level shifts separating 11 constant zones, with corresponding modified shifts in A_i, and error events over $[6, 7]$ (synthetic) and $[12.1, 13.2]$ days. Black curves overlaying the magenta data are the output of non-linear filters aiming to remove congestion and track the baseline $\underline{r}$ and the underlying asymmetry $\underline{a}$.

The difficulty we face in measuring $\mathbf{e_i}$ is that there are three sources of variability which interact: level shifts due to routing, congestion, and the server errors $\mathbf{e_i}$ themselves. As the delays $D_i^{\uparrow}$, $D_i^{\downarrow}$ suffer from all three, they are poorly suited to error measurement. However R_i is *entirely independent of server timestamps*, and therefore of $\mathbf{e_i}$ in the model above. This allows path conditions to be judged independently of server behaviour. In contrast, the asymmetry series A_i carries an amplified server error 'signal' $2\mathbf{e_i}$, while the congestion 'noise' is partially cancelled in a_i. Thus in our methodology A_i provides the basis of error detection, with R_i providing the basis of independent disambiguation.

We illustrate the possible relationships between the R_i and A_i time series in Figure 2. The lower plot, of R_i, shows moderate baseline variability, with 11 constant-$\underline{r}$ zones, and 10 level shifts. The corresponding shifts in $\underline{a}$ and hence A_i in the upper plot can be characterised as follows. Shifts $\{3,4,8,9\}$ are symmetric, meaning that the changes $\Delta\underline{d}^{\uparrow}$ and $\Delta\underline{d}^{\downarrow}$ are equal, and so $\Delta\underline{a} = 0$. Shifts $\{5,6,7,10\}$ are uni-directional, that is they occur in only one of $\underline{d}^{\uparrow}$ and $\underline{d}^{\downarrow}$, and so $|\Delta\underline{a}| = |\Delta\underline{r}|$. Finally, changes $\{1,2\}$ are bi-directional but $\Delta\underline{d}^{\uparrow} \neq \Delta\underline{d}^{\downarrow}$, so partial cancellation occurs and $0 < |\Delta\underline{a}| < |\Delta\underline{r}|$. As for congestion variability, in this case it is very small relative to the baseline change amplitudes. Nonetheless the variability components q_i and a_i, though low, can be seen to vary in character across zones.

The final form of variability is server error. We first provide an example with ground truth by adding controlled errors to the server timestamps. The chosen error function is an oscillation appearing over the interval $[6, 7]$ days, with $[\min, \max]$ values of $[-10, 2]$ ms. As expected, the error function has no impact on R_i, but appears in A_i, with its extremal amplitudes doubled to $[-20, 4]$ ms. Finally, consider the prominent event seen in A_i over $[12.1, 13.2]$ days, with a minimum at $A_i = -31.1$ms, and a maximum at $A_i = 281$ms (beyond plot range). Since the routing shifts and congestion magnitudes over the period are known from R_i to be much smaller, the event can only be due to server error. The alternative explanation, that the true baselines of $d_i^{\uparrow}$ and $d_i^{\downarrow}$ evolve in a precisely equal and opposite way over an extended period, is pathological and we assume that this does not occur. See however the discussion in Section V-D on related phenomena.

It is important to note that a constant error is not in fact detectable, unless it results in a failure of causality, that is $D_i^{\uparrow} < 0$ or $D_i^{\downarrow} < 0$. This follows from the fact that it is $\underline{a} + 2\mathbf{e_i}$, not $\underline{a}$, which is *identifiable* (in principle measurable) from the data. Thus strictly speaking our methodology is for the measurement of *changes* in server error, rather than of absolute error. In what follows we assume (when possible) that the servers are typically not in error, so that when an error change is observed, we interpret and speak of it as a move from an $\mathbf{e_i} = 0$ state, to a state with $\mathbf{e_i} \neq 0$.

B. Error Measurement

In this section we describe a methodology for the measurement of a server error thought to lie within an interval, a *Server Anomaly Zone* (SAZ), which lies inside a larger *Nice Zone* (NZ). An NZ is an interval where R_i is free both of routing events, and of sustained congestion events which would prevent $\underline{r}$ from being reliably measured. For example each of the constant-$\underline{r}$ zones of Figure 2 is a Nice Zone. The error *Context Zone* CZ is defined as CZ = NZ \ SAZ and has left and right components. The selection of NZ and SAZ is discussed in Section III-C, here they are taken as given.

To estimate $\mathbf{e_i}$ we must first know $\underline{r}$ and $\underline{a}$. A natural estimator of $\underline{r}$ over NZ is $\hat{\underline{r}}_{\text{NZ}} = \min_{i \in \text{NZ}} R_i$. Sometimes one-way delays can be exploited to define an estimator $\hat{\underline{r}}_{\text{CZ}}$ over CZ which is often superior (see below). In that case we set our final $\hat{\underline{r}} = \min(\hat{\underline{r}}_{\text{CZ}}, \hat{\underline{r}}_{\text{NZ}})$, otherwise $\hat{\underline{r}} = \hat{\underline{r}}_{\text{NZ}}$.

By definition, $\underline{a}$ is constant over the NZ, but its estimation is more difficult than for $\underline{r}$ because a_i takes either sign, congestion cannot be assumed to be symmetric, and data over SAZ can't be used because of the possibility of server error. We proceed as follows. Since $a_i = q_i^{\uparrow} - q_i^{\downarrow}$, and $q_i = q_i^{\uparrow} + q_i^{\downarrow}$,

we have $|a_i| < q_i$, and so $\underline{a} \in [A_i - q_i, A_i + q_i]$ for all $i \in$ CZ. Hence the intersection of all such intervals yields a feasible interval, $[\max_i(A_i - q_i), \min_i(A_i + q_i)]$, where $\underline{a}$ must lie. We estimate this interval by estimating q_i as $R_i - \hat{\underline{r}}_{\text{NZ}}$, and define $\hat{\underline{a}}$ to be its midpoint:

$$\hat{\underline{a}} = \frac{1}{2}\Big(\max_i(A_i - R_i + \hat{\underline{r}}_{\text{NZ}}) + \min_i(A_i + R_i - \hat{\underline{r}}_{\text{NZ}}) \Big) \ . \quad (1)$$

This estimator is very robust, in fact it is *independent* of any constant error in $\hat{\underline{r}}_{\text{NZ}}$, and even holds if the intersection is null, an impossible event that points to $\hat{\underline{r}}_{\text{NZ}}$ being too high. If this occurs, it implies we must lower the baseline estimate to at least

$$\hat{\underline{r}}_{\text{CZ}} = \hat{\underline{r}}_{\text{NZ}} - \frac{1}{2}\Big(\max_i(A_i - R_i + \hat{\underline{r}}_{\text{NZ}}) - \min_i(A_i + R_i - \hat{\underline{r}}_{\text{NZ}}), \Big) \quad (2)$$

the value that (just) allows the intersection to be non-null.

To measure the $\mathbf{e_i}$ function across the SAZ, we must deal with congestion there. We do this by calculating a conservatively *adjusted asymmetry* as follows. Since $|a_i| < q_i$, for each i we can remove a_i by pushing A_i inward by q_i toward the underlying value of $\underline{a}$, *without overshoot*, as follows:

$$\underline{A}_i(\underline{a}) = \underline{a} + \frac{A_i - \underline{a}}{|A_i - \underline{a}|} \max(0, |A_i - \underline{a}| - q_i) \ . \quad (3)$$

When $\underline{A}_i(\underline{a}) = \underline{a}$, it means either an error was not present at that point, or that it was smaller than $q_i - |a_i|$ in magnitude and so was suppressed by the conservative congestion removal. At stamps where congestion is nearly uni-directional: $|a_i| \approx q_i$, and so $\underline{A}_i \approx \underline{a} + \mathbf{e_i}$, and $\mathbf{e_i}$ is revealed if present.

To capture the global size of the error function over SAZ regardless of its detailed shape, we use the range of the adjusted asymmetry about $\hat{\underline{a}}$, replacing q_i and $\underline{a}$ by their estimates in (3), yielding

$$\hat{E} = \frac{1}{2}\Big(\max\{\hat{\underline{a}}, \max_{i \in \text{SAZ}} \underline{A}_i(\hat{\underline{a}})\} - \min\{\hat{\underline{a}}, \min_{i \in \text{SAZ}} \underline{A}_i(\hat{\underline{a}})\} \Big) \quad (4)$$

as an estimate of the true range $E = \max\{0, \max_{i \in \text{SAZ}} \mathbf{e_i}\} - \min\{0, \min_{i \in \text{SAZ}} \mathbf{e_i}\}$. This estimate is robust as the extremes of the error are least vulnerable to suppression by the congestion bounding, and only a few samples near the extremes with close to uni-directional congestion are needed for good estimation. The estimate is also highly robust as well as conservative to failure of the constant-$\underline{a}$ assumption, or poor-$\underline{a}$ estimation, because A_i values will still be adjusted away from the extremes, leaving the adjusted range unaffected.

In principle any value of $\hat{E}$ above zero represents a lower bound on the error, since the estimate should obey $\hat{E} \le E$ due to the congestion suppression. However this is not guaranteed: when both congestion is low and $\hat{\underline{r}}$ is inaccurate, $\hat{E} > E$ is possible, though unlikely. To help quantify this uncertainty, the following ratio provides a measure of the significance of $\hat{E}$ by measuring it in units corresponding to a bound, $\bar{E}_{BL}$, on the error in $\hat{\underline{r}}$:

$$\mu = \hat{E} / \bar{E}_{BL} \ . \quad (5)$$

We use $\bar{E}_{BL} = \text{median}_{i \in \text{NZ}}(R_i) - \hat{\underline{r}}$ as a loose upper bound on the unknown true error. This choice is, in general, very conservative, and corresponds well with visual appraisals. As a conservative measure, if $\mu > 1$, we can be very confident that the error is unambiguously above measurement errors.

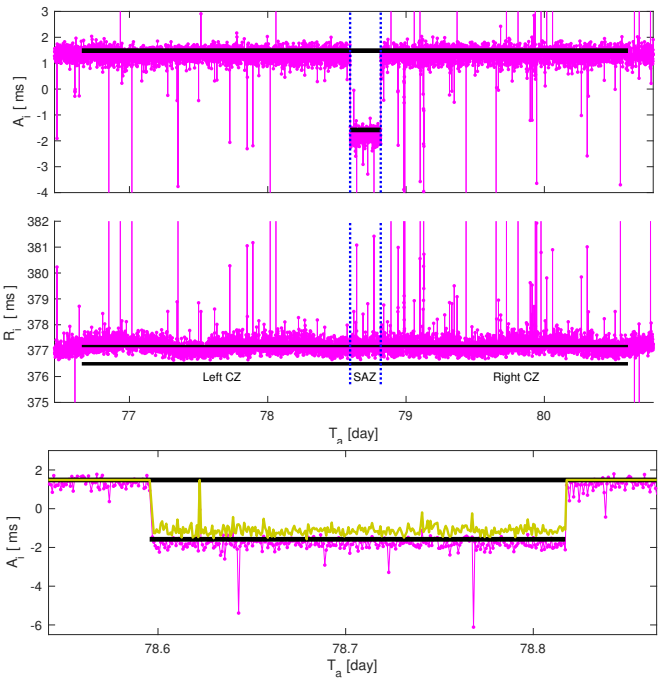

Fig. 3. Measurement of a server error. Middle plot: R_i and its measured minimum $\underline{r}$ (lower black line) and median (thin black line) over a Nice Zone, covering NZ = CZ ∪ SAZ. Upper plot: A_i and its measured underlying $\underline{a}$ (black line over NZ) Lower plot: zoom centered on the SAZ, showing the adjusted asymmetry $\underline{A}_i$ over NZ in green, whose extremes define the error range (black lines). Outside the SAZ $\underline{A}_i = \hat{\underline{a}}$ as expected.

Figure 3 illustrates the analysis above, with $\hat{\underline{r}} = \hat{\underline{r}}_{\text{CZ}} = 376.49 < \hat{\underline{r}}_{\text{NZ}} = 376.66$ ms, $\hat{\underline{a}} = 1.48$ms, and an error bound of $\hat{E} = 1.53$ms in the context of $\bar{E}_{BL} = 0.685$ ms, yielding a ratio of $\mu = 2.2$. Note how the IZ was defined with unequal left/right CZ components to exploit the stamps available for estimation while avoiding the very short-lived baseline shifts in R_i visible to either side. The bottom plot in the figure provides a zoom showing the adjusted asymmetry in action.

To illustrate the effectiveness of the above estimation approach, we provide the following simulation results. The Nice Zone path model is as follows. We set $(\underline{d}^{\rightarrow}, \underline{r}, \underline{a}) = (0.04, 100, 10)$ ms, implying $(\underline{d}^{\uparrow}, \underline{d}^{\downarrow}) = (54.98, 44.98)$ ms. For the variable components, $q_i^{\rightarrow}$ is uniformly distributed over $[0, 50]$ µs, and $q_i^{\uparrow}$, $q_i^{\downarrow}$ are each exponentially distributed with mean (and standard deviation) of $b = 1$ ms, implying that $E_{BL} \equiv \text{median}(R_i) - \underline{r} \approx 1.70$ ms. Using these models, 10,000 independent times series for each of $D_i^{\uparrow}$, $D_i^{\rightarrow}$ and $D_i^{\downarrow}$ were drawn, and series for R_i and A_i generated. The Nice Zone is N samples wide, with the anomaly zone taking up the central $N/3$ of these. The error function over the SAZ consists of an upward level shift of $E/2$, followed by a downward shift of $-E$ in the centre of the SAZ, and finally a shift back up of $E/2$ at the end. We present results for $N = 300$ and 1500.

Figure 4 gives the relative error $(\hat{E} - E)/E$, estimated based on the median of the relative errors observed over the 10,000 independent experiments. For both curves, the straight line of slope -1 in the log-log plot beyond $E/E_{BL} = 0.04$ shows that

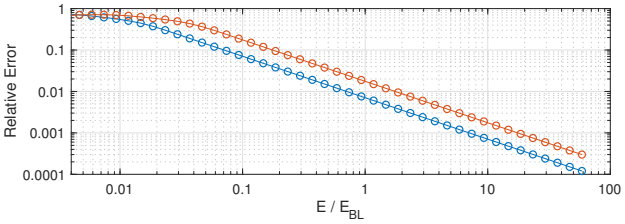

Fig. 4. Relative Error of $\hat{E}$ as a function of the normalized error E/E_{BL}. Top: $N = 300$, Bottom: $N = 1500$.

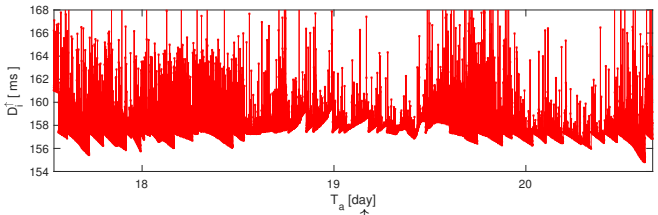

Fig. 5. Example of **H** server #88. The $D_i^{\uparrow}$ baseline is continuously perturbed by errors of S&R type.

estimation error drops as $\sim E^{-1}$. Consider the lower curve where errors are lower since $N = 1500$ is higher. Around $E/E_{BL} = 1$ the relative error is only 0.7%, consistent with the claim above that $\mu = 1$ is conservative.

It can (and does) happen that a server clock change occurs inbetween the server timestamps $T_{b,i}$ and $T_{e,i}$. This results in a different placement of the $\mathbf{e_i}$ terms, modifying the analysis above. For space reasons we omit the details.

C. Zone Selection Methodology

From the examples in Figures 2 and 3, the determination of Nice Zones and Server Anomaly Zones may seem straightforward. Unfortunately, each of baseline variability, congestion, and server error behaviour, can be far more complex. Since we seek a highly reliable methodology free of false positives, we adopt an assisted manual approach capable of managing this complexity, while providing detailed labelled data for future studies (available at [5]).

Server error detection and measurement was performed within a Matlab-based visualization, analysis and tagging tool we developed. The tool is used to assist in manually scanning the $D_i^{\uparrow}$, $D_i^{\rightarrow}$, $D_i^{\uparrow}$, R_i and A_i time series in parallel to detect server anomalies, and provides features which aid in the reliable interpretation of candidate anomalies given their routing and congestion contexts, and to observe their relationship to protocol measurements (Stratum and LI values). A central feature is the incorporation of the rigorous measurement of Section III-B to deal with congestion and evaluate candidates.

When a viable candidate is identified, the SAZ and NZ boundaries are selected, the measurement of Section III-B triggered, and the inputs and results are stored. Additional manually assessed qualifiers are also recorded as part of the tag. The *expertise* qualifier indicates whether the strict conditions of a Nice Zone are satisfied, or not, for example because routing events intersect the SAZ. Such *complex* anomalies are in the minority, and are not included in the results below. The *Anice* qualifier records whether the constant-$\underline{a}$ assumption is valid over CZ, to allow $\hat{\underline{r}}$ to be selected appropriately as described above. This assumption may fail when the server is continuously in error, where $\hat{\underline{r}}_{CZ}$ could be highly inaccurate. Server events severe enough to break causality ($D_i^{\uparrow} < 0$ or $D_i^{\downarrow} < 0$) are certainly errors and so are automatically flagged, but are processed manually like any other candidate. The final manually assessed qualifier, described below, is the anomaly *type*.

Some servers have so many errors that it is necessary to modify the tagging procedure. Instead of isolating each error, which is either impossible because errors are continuous, or impractical because there are so many, large Nice Zones are sought containing representative examples of the error behaviour. A server is deemed to be of this *High Prevalence* (**H**) type when the server exhibits errors over 25% of the time, and when in the error regions either (i) errors are continuous with size consistently above 0.5 ms, or (ii) discrete errors occur with frequency in excess of 0.5 per hour. For such servers we tag the 10 largest Nice Zones within the error zone(s), Additional prominent errors which are distinct from the 'continuous' error pattern are also individually tagged.

The accelerations afforded by the tool notwithstanding, the detailed examination of the three data sets, with a total of around 500 million stamps, is very labour intensive, and the analysis below represents months of effort.

D. Server and Error Classification

To describe qualitatively the overall error status of a server, we define the following detection classes:

E(rrored): at least one detected error
G(ood): no clear sign of a server error

Errored is a high confidence statement that the server has been in error. It is not a mere heuristic classification, but is based on direct measurements of error while controlling (informally but conservatively) for statistical variability. **G**ood is a confident statement that there are no major errors, but does not imply that there are none. For example anomalies smaller than the underlying error in $\underline{r}$ will generally be undetectable.

To describe quantitively the number of errors in **E** servers:

R(are): less than 1 error per week
C(ommon): more than 1 error per week but not **H**igh
H(igh): in error more than 25% of the time (see above).

The gap between **C** and **H** servers is huge, for example in **Exp3 C** servers have on average 30 errors (see Table V), whereas most **H** servers are in error 100% of the time.

Finally, error function shape over a server anomaly zone is characterised broadly as follows:

S&R (Skew and Return): a large class containing at least one sudden shift together with one or more periods of skew (linear change) and/or convergence (levelling out of skew);
LS (Level Shift): one or more level shifts (and nothing else);
D(rift): oscillatory or wandering appearance with no shifts.

The variety of behaviour observed is broad and complex, but the classes above are distinct, and can be mapped to plausible underlying causes. The S&R class (see the example on the right in Figure 2, and Figure 5) is consistent with a synchronization

ID #	URL	(Exp1, Exp2, Exp3) Strata Seen {0, 1, 2, ..., 16}	Detection { G, E }	Preval. { R, C, **H** }	Error Types Seen { D, L=LS, S=S&R }	Size $\hat{E}$ [ms]	Significance Ratio μ
1	augean.eleceng.adelaide.edu.au	{–}; {3}; {0,2,3}	(–, E, E)	(–, **H**, **H**)	–, D, {D, S}	–, 9, 4.1	–, 14, 14
10	server-in-our-lab.au	{–}; {1}; {1}	(–, E, E)	(–, **H**, **H**)	–, D, D	–, 0.19, 0.27	–, 5.9, 35
15	ts1.aco.net	{–}; {1,2}; {0-2}	(E, E, E)	(R, R, **H**)	{L, S}, {L, S}, {D, L, S}	0.74, 2, 1.4	3.2, 3.7, 2.1
16	ts2.aco.net	{–}; {1,2}; {0,1}	(E, E, E)	(R, R, R)	L, L, {L, S}	3 , 1.5 , 95	2.9, 2.8, 13
20	ntps1.pads.ufrj.br	{–}; {1,2}; {–}	(**G**, E, –)	(· , C, –)	· , S, –	· , 4.3, –	· , 10, –
21	ntp.probe-networks.de	{–}; {2}; {2}	(E, **G**, E)	(**H**, · , **H**)	{D, S}, · , {D, S}	8, · , 1.1	5.2, · , 2.4
23	ntp1.fau.de	{–}; {1,2}; {1,2}	(E, E, E)	(C, R, R)	{L, S}, {L, S}, {D, S}	2.5, 2.1, 1.8	4.4, 2.8, 1.0
25	ntp3.fau.de	{–}; {1,2}; {1,2}	(E, E, E)	(R, R, **H**)	{L, S}, {L, S}, D	2.4, 2, 6.3	4.2, 3.5, 0.4
27	ntps1-1.cs.tu-berlin.de	{–}; {1}; {0,1}	(G, E, E)	(· , R, **H**)	· , {L, S}, {D, L}	· , 1.5, 9.7	· , 3.4, 3.3
31	time.fu-berlin.de	{–}; {1,2}; {1,2}	(G, E, E)	(· , R, R)	· , S, {D, S}	· , 73, 1.0	· , $1.8e^2$, 0.60
32	time1.one4vision.de	{–}; {0,1}; {–}	(**E**, E, –)	(C, C, –)	L, L, –	79, $1.8e^2$, –	$2.4e^2$, $3.7e^2$, –
33	zeit.fu-berlin.de	{–}; {1,2,10}; {1,2}	(G, E, E)	(· , R, R)	· , S, {L, S}	· , 76, 2.8	· , $1.4e^2$, 0.75
35	ntp1.net.icm.edu.pl	{–}; {2}; {2}	(E, **E**, E)	(**H**, R, R)	D, S, D	2.9, 1.3, 2.8	9.5, 2.8, 0.15
38	time.coi.pw.edu.pl	{–}; {1}; {0,1,6}	(G, E, E)	(· , R, R)	· , {D, S}, S	· , 11, $3.9e^4$	· , 29, $1.2e^3$
39	vega.cbk.poz-.pl	{–}; {2}; {3}	(E, E, E)	(**H**, **H**, **H**)	{D, L, S}, {D, S}, {D, S}	$1.2e^2$, 18, 4.3	69, 49, 0.35
40	ntp0.ntp-servers.net	{–}; {1,2}; {–}	(**E**, E, –)	(R, R, –)	{L, S}, S, –	14, 10, –	21, 4.7, –
41	ntp1.ntp-servers.net	{–}; {1,2}; {0-2}	(E, E, E)	(**H**, R, **H**)	D , S , {D, L, S}	21, $2.1e^2$, 5.2	23, 40, 4.7
44	ntp2.vniiftri.ru	{–}; {1,12}; {0-2,12}	(**E**, E, E)	(R, R, C)	L, L, {L, S}	88, $5.9e^3$, 9.5	31, $1.3e^4$, 15
55	time1.stupi.se	{–}; {1,2}; {0-2}	(**E**, E, E)	(R, R, R)	{L, S}, S, S	$1.4e^2$, 81, $5.1e^4$	$4.4e^2$, $1.1e^2$, $1.2e^4$
56	time2.stupi.se	{–}; {1,2}; {0-2}	(**E**, E, E)	(R, R, R)	{L, S}, S, {L, S}	10, 1.5e2, $7.8e^4$	16, 94, $1.4e^5$
57	timehost.lysator.liu.se	{–}; {2,3}; {–}	(E, E, –)	(R, **H**, –)	S, {D, S}, –	11, 3, –	35, 7.1, –
58	ntp.remco.org	{–}; {1,2}; {0-2}	(**E**, E, E)	(R, C, R)	{D, S}, {D, S}, L	6.8, 2.6, $6.4e^5$	8.3, 3.2, $3.0e^5$
61	bonehed.lcs.mit.edu	{–}; {1}; {0-2}	(E, E, E)	(C, R, **H**)	{D, L, S}, S, {D, S}	2.1, 8.9, 0.4	9.2, 13, 1.5
62	clock.danplanet.com	{–}; {2,3}; {–}	(**E**, E, –)	(R, **H**, –)	L, {D, S}, –	35, $2.0e^3$, –	23, $3.7e^2$, –
63	clock.isc.org	{–}; {1,2}; {1,2}	(E, E, E)	(**H**, **H**, **H**)	{D, S}, {D, L, S}, {D, S}	3.2, 1.7, 0.99	19, 5.6, 3.5
65	clock.sjc.he.net	{–}; {1}; {0-2,4}	(E, **G**, E)	(C, · , C)	{L, S}, · , S	4.3, · , 0.41	27, · , 0.95
66	clock.via.net	{–}; {2,3}; {2}	(E, E, E)	(R, **H**, **H**)	S, {D, S}, {D, S}	27, 27, 8	88, 52, 14
68	nist.netservicesgroup.com	{–}; {1}; {0,1}	(E, E, E)	(**H**, **H**, **H**)	S, {L, S}, S	10, 20, $4.2e^3$	47, 59, $8.5e^3$
69	ntp.myfloridacity.us	{–}; {1}; {0,1}	(E, **E**, E)	(**H**, R, **H**)	S, {L, S}, {D, L}	$1.1e^2$, $1.0e^3$, 15	31, $3.5e^2$, 4.1
72	ntp2.netwrx1.com	{–}; {1,2}; {–}	(**E**, E, –)	(R, C, –)	{L, S}, S, –	84, 13, –	13, 5.7, –
73	rackety.udel.edu	{–}; {1,2}; {0-4}	(E, E, E)	(R, R, **H**)	L, {D, S}, {D, S}	51, 35, 2.4	19, 38, 4.7
78	time-a.timefreq.bldrdoc.gov	{–}; {1}; {1}	(E, E, E)	(**H**, **H**, **H**)	S, {D, L, S}, {L, S}	64, 2, 23	41, 5.6, 80
79	time-b.nist.gov	{–}; {1}; {1}	(E, **E**, E)	(R, C, C)	L, S, S	2.8, 2.2, 2.1	7.3, 4.6, 2
80	time-b.timefreq.bldrdoc.gov	{–}; {1}; {1}	(E, E, E)	(**H**, **H**, **H**)	{L, S}, {D, S}, {L, S}	32, 8.8, 36	24, 25, 95
83	utcnist.colorado.edu	{–}; {1}; {–}	(E, G, –)	(**H**, · , –)	{D, L, S}, · , –	3.2, · , –	14, · , –
84	utcnist2.colorado.edu	{–}; {1}; {1}	(E, E, G)	(**H**, R, ·)	{L, S}, S, ·	14, 2, ·	36, 5.7, ·
88	ntp.bsdbg.net	{–}; {1-8,10}; {0-3}	(E, E, E)	(**H**, **H**, **H**)	S, {D, S}, {D, L, S}	36, 9.4, 5.6	1.7, 7.2, 19
89	ntp.nic.cz	{–}; {1,2}; {1}	(E, **G**, G)	(R, · , ·)	S, · , ·	2, · , ·	4.8, · , ·
91	canon.inria.fr	{–}; {1}; {0-3}	(**G**, E, E)	(· , R, **H**)	· , S, {D, S}	· , 14, 1.4	· , 67, 3.4
94	ntp2.inrim.it	{–}; {1,2}; {0-2}	(G, E, G)	(· , R, ·)	· , {L, S} ·	· , $5.0e^2$, ·	· , $6.8e^2$, ·

TABLE II

SERVER ERROR CHARACTERISTICS FOR SERVERS THAT WERE **E** IN **EXP1** OR **EXP2** IN [3]. SERVERS IN NATIONAL LABORATORIES HAVE BOLD CYAN URL'S. **LISTNEW** SERVERS (**EXP3**) HAVE COLORED SERVERIDS MATCHING THE COLOURS USED IN SECTION V. DETECTION CLASSIFICATIONS THAT DIFFER FROM THOSE OF [3] ARE MARKED IN BOLD, AND H SERVERS ARE BOLDED FOR EMPHASIS.

algorithm imposing or noticing clock offsets and then attempting to cope with the consequences. Errors in the LS class have the appearance, superficially, of routing events, and include all errors occuring over only a single stamp. They could be caused by a constant being misguidedly added to the clock output outside of the main synchronization algorithm. Finally, the Drift class is indicative of a synchronization daemon which has lost local stability (oscillations), or a reference source or daemon that has died (resulting in temperature dependent drift).

IV. LONGITUDINAL FINDINGS

In this section we present the longitudinal results of the paper, focussing primarily on summary statistics of error.

Since we report on servers with public identities, we have taken pains to ensure that their performance is individually traceable. We are fortunate that the number of servers,102, is just small enough to still allow this level of granularity, and after trying many alternatives, we found a tabular format to the most informative, direct, and ultimately compact way to get the key information across: namely which servers are in error, when and to what extent.

We split our results on errored servers across two tables. Table II covers the 40 servers which, in the analysis of [3], were found to be errored in at least one of **Exp1** and **Exp2**. Table III groups the other 46 servers which we find to be errored (using our new analysis) in any of **Exp1**, **Exp2**, or **Exp3**. Table IV covers the remaining 16 servers, that were not **E** in any experiment. In each table, for the server in the indicated experiment, "–" denotes that the data was unavailable, and "·" indicates that column is not applicable.

In all tables the Strata column gives, for each experiment, the list of all stratum values seen for each server. Similarly, in the errored Tables II and III, the Type column lists all error

types seen, whereas the Size and Ratio columns contain, for each experiment, median values taken over all tagged errors fulfilling the strict Nice Zone requirements. In contrast, [3] provided only a single representative error (and ratio) across **Exp1** and **Exp2** jointly. We emphasize that our results here are based on a new systematic error tagging of all servers, in each of the three datasets, using new methodology: a very significant increase over the work reported in [3].

We organise our observations mined from the tabular data into a number of central messages.

1: Errors are widespread Collectively over all experiments, 86 out of 102 (84%) of servers were found to be errored. Furthermore the errors are often very common, with 25 of these 86 (29%) being classed as **H** in at least one experiment. Overall, the prevalence labels {**R**,**C**,**H**} occur {104,18,43} times, or $\{63, 11, 26\}\%$. Thus typically, individual errored servers have either very few errors, or an enormous number.

2: Errors are significant Over all experiments, 60 (59%) of errored servers have median sizes recorded in excess of 10ms, and 32 (31%) above 100ms. Extreme values, not reflected in the median, go well beyond this. For example in **Exp3** server #69 posted two errors over 200 years in size! Nor are most errors subtle. Significance ratios above a 'blatant' 10 occur in 64 servers, and huge values $\mu \geq 100$ in 35 cases.

3: Our methodology has improved sensitivity We compare against the detection results of [3] for **Exp1** and **Exp2**. Each of the servers in Table II had at least one **E** detection in [3] (called **B** there), *all of which* are also classed as **E** here, and there are an additional 5 **E** detections which were previously **G**. In contrast, none of the servers in Table III were errored in [3], but now 27 servers are found to be. These findings support (i) the claim from [3] that the **E** detections are reliable; (ii) our claim that the new methodology has greater sensitivity. Moreover, in addition to {**G**,**E**}, a third **A**(mbiguous) class was used in [3] when firm conclusions could not be drawn. This proved unnecessary with our new methodology, which reclassified the $\{11, 25, 6\}$ **A** instances within Tables II, III, and IV as #{**G**,**E**}={5,6}; {14,11}; and {6,0} respectively. All $\mathbf{A} \rightarrow \{\mathbf{G}, \mathbf{E}\}$ and $\mathbf{G} \rightarrow \mathbf{E}$ Detection reclassifications are marked in bold in the tables. Of the 42 new **E** detections, 38 (90%) are **R**, that is rare prevalence, more likely to be missed by a less sensitive detector.

4: Server behaviour is consistent over time The frequencies of the Detection symbols $(\mathbf{G}, \mathbf{E}, -)$ are $(45, 44, 13)$, $(48, 54, 0)$, and $(24, 67, 11)$ for **Exp1**, **Exp2**, **Exp3** respectively. This exhibits a clear increase in the $\mathbf{E}/\mathbf{G}$ ratio over time: $0.98 \mapsto 1.1 \mapsto 2.8$, however we believe this to be an artifact of the increased total number of stamps from **Exp1** $\mapsto$ **Exp2**$(\times 3)$ to **Exp2** $\mapsto$ **Exp3**$(\times 10)$, along with increasing sampling rate, which increase the detection probability of rare and short lived anomalies respectively, rather than an actual increase in server error rates. We assess per-server consistency over time by restricting to the 78 servers with data in each experiment, and comparing event probabilities against what the above marginals imply, assuming a model of independent experiments. Detection classification is very consistent: we find $\{\mathbf{EEE}, \mathbf{GGG}\}$ in $\{31,14\}\%$ of cases, well above the $\{19,6.3\}\%$ expected from the model.

5: The link between Stratum and Error is Complex Although the servers in **ListLong** are ostensibly S1, they often display other strata: only $\{15,24,63\}\%$ of servers in Table II, III, IV are always S1. Not surprisingly given its larger stamp population, **Exp3** consistently found greater strata diversity. Also as expected, non-errored servers (Table IV) have 'quieter' strata in general, and servers #$\{1,21,39,62,66\}$, that are never S1, perform poorly. There are however many exceptions: #14 is never S1 but performs well, and #$\{78,80,81\}$ have **H** prevalence and yet announce themselves as S1 in every packet. The relationship between strata and errors requires a closer examination, which we perform in the next section.

6: Many National Laboratory Servers are Errored There are 24 servers run by National Laboratories, highlighted as bolded cyan in the tables. Of these 20 (83%) servers are errored, and of these, 7 were **H** in at least one experiment. Of the 6 (**H**,**H**,**H**) servers, 4 were from National Laboratories.

V. Error Anatomy and Prediction

In this section we take advantage of the greater resolution of **Exp3**, and the detailed tagging, in order to explore the occurrences and nature of server errors in more detail, and to explore their observed relationship to protocol behaviour.

A. Error Diversity

Whereas the tables above provided only a single median value for error size per server, the scatter plots in Figure 6 show both size and duration for all tagged anomalies for all servers, organised according to error prevalence, and colored according to server using the scheme employed for server ID in Tables II and III.

For the 40 **R** servers (left plot), containing a total of 104 error zones, the chief finding is the extraordinary diversity (note the logarithmic scale covering 6 orders of magnitude on both axes!) across anomalies in each of the size and duration dimensions, as well as jointly. Error durations range from under a second to over 50 days, and sizes from sub-ms to 1.6 hours. Across servers, diversity is also apparent: we find no sign of similar behaviour (the alignment around a duration of 1s results from the sampling). In terms of intra-server diversity, the definition of **R** implies a maximum of 9 errors, which is still enough to reveal similarities if present. We observe some diffuse clusters in some servers, but no discernable pattern in others.

For the 7 **C** servers (middle plot, total of 212 errors), some exhibit a degree of intra-server clustering, in particular the 90 errors from NIST server #79 (small crosses), however sometimes this is in one dimension only, and in other cases no clustering is apparent. The diversity is still very impressive in each dimension, though not as extreme as for **R**. Across servers, we again see very different patterns.

For the 20 **H** servers (right plot), as errors are not individually tagged, we replace error duration by *Etime*, the fraction of the trace taken up by the errored zones, and for error size a median value is given of the errors included within tagged

ID #	URL	(Exp1, Exp2, Exp3) Strata Seen {0, 1, 2, ..., 16}	Detection {G, E}	Preval. { R, C, **H** }	Error Types Seen { D, L=LS, S=S&R }	Size $\hat{E}$ [ms]	Significance Ratio μ
2	**ntp.perth.nmi.gov.au**	{–}; {1}; {1,2}	(–, G, E)	(–, ·, R)	–, ·, S	–, ·, 0.65	–, ·, 2.3
6	ntp.waia.asn.au	{–}; {1}; {0,1,3}	(–, G, E)	(–, ·, R)	–, ·, L	–, ·, $6.9e^2$	–, ·, $1.3e^3$
7	ntp1.net.monash.edu.au	{–}; {1}; {1}	(–, G, E)	(–, ·, R)	–, ·, L	–, ·, $1.8e^2$	–, ·, $2.5e^2$
8	ntp10.net.monash.edu.au	{–}; {1}; {1}	(–, **E**, E)	(–, R, C)	–, L, L	–, 44, 18	–, $1.9e^2$, $3.2e^2$
9	syd4gps0.syd.ops.aspac.uu.net	{–}; {1}; {1,16}	(–, **E**, G)	(–, R, ·)	–, S, ·	–, 2, ·	–, 7.8, ·
11	tick.une.edu.au	{–}; {1}; {0,1}	(–, **E**, E)	(–, R, R)	–, L, S	–, 33, 65	–, 26, $1.9e^2$
12	tock.une.edu.au	{–}; {1,12}; {0,1}	(–, G, E)	(–, ·, R)	–, ·, S	–, ·, 41	–, ·, $1.2e^2$
13	vk6hgr.echidna.id.au	{–}; {1-3}; {0-3,11}	(–, **E**, E)	(–, R, **H**)	–, {L, S}, {D, L, S}	–, 99, 22	–, 19, 6.5
14	asynchronos.iiss.at	{–}; {2}; {–}	(**E**, G, –)	(R, ·, –)	S, ·, –	18, ·, –	2.6, ·, –
18	c.st1.ntp.br	{–}; {1,2}; {0,1}	(**G**, **E**, E)	(·, R, R)	·, L, {L, S}	·, 1.7, 1.3	·, 5, 2.1
19	d.st1.ntp.br	{–}; {1,2}; {0-2}	(**G**, **G**, E)	(·, ·, R)	·, ·, S	·, ·, 2.4	·, ·, 6.5
24	ntp2.fau.de	{–}; {1}; {0,1}	(G, **E**, E)	(·, R, C)	·, S, {D, S}	·, 26, 18	·, 37, 29
28	**ptbtime1.ptb.de**	{–}; {1}; {0,1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, $1.1e^3$	·, ·, $2.1e^3$
30	rustime01.rus.uni-stuttgart.de	{–}; {1,13}; {1,2}	(G, **E**, E)	(·, R, R)	·, S, D	·, 82, 4.5	·, 24, 0.45
34	ntp.certum.pl	{–}; {1}; {1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, 6.9	·, ·, 3.4
36	ntp1.tp.pl	{–}; {1,2,8}; {0-3}	(G, G, E)	(·, ·, R)	·, ·, D	·, ·, 2.7	·, ·, 0.45
42	**ntp1.vniiftri.ru**	{–}; {1}; {0-3,12}	(G, **E**, E)	(·, R, R)	·, S, {L, S}	·, $1.8e^2$, 2.3	·, 98, 2.9
43	ntp2.ntp-servers.net	{–}; {1}; {0-2}	(**E**, **G**, E)	(R, ·, R)	L, ·, L	50, ·, $7.0e^3$	28, ·, $4.4e^2$
45	**ntp3.vniiftri.ru**	{–}; {1}; {0-2}	(G, **E**, E)	(·, R, R)	·, S, S	·, 20, 46	·, 24, 55
46	**ntp4.vniiftri.ru**	{–}; {1-3}; {0-2}	(G, **E**, E)	(·, C, R)	·, L, {L, S}	·, $2.4e^2$, $2.1e^2$	·, $7.4e^2$, $2.7e^2$
47	ntp1.gbg.netnod.se	{–}; {1}; {1}	(G, G, E)	(·, ·, R)	·, ·, S	·, ·, $1.0e^3$	·, ·, $2.4e^2$
48	ntp1.mmo.netnod.se	{–}; {1}; {1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, $1.0e^3$	·, ·, $2.7e^2$
51	ntp2.gbg.netnod.se	{–}; {1}; {1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, $1.0e^3$	·, ·, $1.7e^3$
53	ntp2.sp.se	{–}; {1}; {1}	(G, G, E)	(·, ·, R)	·, ·, S	·, ·, 0.75	·, ·, 1.2
54	ntp2.sth.netnod.se	{–}; {1}; {1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, $1.0e^3$	·, ·, $6.5e^2$
64	clock.nyc.he.net	{–}; {1,4}; {0-2,4}	(**G**, **E**, G)	(·, R, ·)	·, L, ·	·, $4.5e^2$, ·	·, $1.1e^3$, ·
67	gps.layer42.net	{–}; {1,2}; {1,2}	(**E**, **E**, E)	(R, R, R)	L, L, {L, S}	62, 97, $4.5e^3$	$1.0e^3$, 23, 55
74	t1.timegps.net	{–}; {1,2}; {1}	(G, **E**, G)	(·, R, ·)	·, S, ·	·, 30, ·	·, 33, ·
75	t2.timegps.net	{–}; {1,2}; {0-2}	(G, **E**, E)	(·, R, R)	·, S, D	·, 30, $3.3e^2$	·, 26, $2.0e^2$
76	time.xmission.com	{–}; {1,2}; {1,2}	(**E**, **E**, E)	(R, R, R)	L, {L, S, D	2.6, 0.94, 0.89	4.9, 3.6, 3.4
77	**time-a.nist.gov**	{–}; {1}; {0,1}	(**G**, **G**, E)	(·, ·, R)	·, ·, S	·, ·, $1.9e^2$	·, ·, $2.6e^2$
81	**time-c.timefreq.bldrdoc.gov**	{–}; {1}; {1}	(**E**, G, E)	(R ·, **H**)	S, ·, {L, S}	18, ·, 9	37, ·, 28
82	timekeeper.isi.edu	{–}; {1}; {0,1}	(**G**, **G**, E)	(·, ·, R)	·, ·, L	·, ·, $2.9e^2$	·, ·, $3.1e^2$
85	ntp.amnic.net	{–}; {1}; {0,1,2}	(**E**, **G**, E)	(C, ·, R)	{L, S}, ·, {D, S}	13, ·, 16	14, ·, 0.75
86	**ntp1.oma.be**	{–}; {1}; {0,1}	(**E**, **E**, E)	(R, R, R)	S, S, {L, S}	$2.3e^2$, 29, 28	$1.3e^2$, 49, 11
87	**ntp2.oma.be**	{–}; {1}; {0,1,16}	(**E**, G, E)	(R, ·, C)	{L, S}, ·, S	91, ·, 4.5	$1.3e^2$, ·, 7.5
92	ntp-galway.hea.net	{–}; {1,2}; {1}	(**E**, **G**, G)	(R, ·, ·)	S, ·, ·	7.3, ·, ·	9.3, ·, ·
93	**ntp1.inrim.it**	{–}; {1,2}; {0-2}	(G, **E**, G)	(·, R, ·)	·, L, ·	·, $1.0e^3$, ·	·, $1.3e^3$, ·
95	clock.nc.fukuoka-u.ac.jp	{–}; {1}; {0-3}	(**E**, G, E)	(R, ·, C)	S, ·, {L, S}	12, ·, 0.26	36, ·, 0.6
96	cronos.cenam.mx	{–}; {1}; {0,1}	(G, G, E)	(·, ·, R)	·, ·, S	·, ·, 2.5	·, ·, 2.9
97	ntp2.usv.ro	{–}; {1}; {0,1}	(**E**, G, G)	(R, ·, ·)	{L, S}, ·, ·	59, ·, ·	$1.3e^2$, ·, ·
98	ntp3.usv.ro	{–}; {1}; {0,1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, $2.4e^3$	·, ·, $5.5e^3$
99	ntp.mostovna.com	{–}; {1}; {0,1}	(G, G, E)	(·, ·, R)	·, ·, L	·, ·, 0.65	·, ·, 2.4
100	**hora.roa.es**	{–}; {1}; {0-2}	(**G**, **G**, E)	(·, ·, R)	·, ·, S	·, ·, $1.2e^2$	·, ·, 22
101	ntp.i2t.ehu.es	{–}; {1}; {1}	(**E**, **E**, G)	(R, R, ·)	S, S, ·	$1.3e^2$, 5.9, ·	$1.2e^2$, 9.6, ·
102	ntp.time.in.ua	{–}; {1}; {–}	(**E**, **G**, –)	(R, ·, –)	L, ·, –	12, ·, –	2.6, ·, –

TABLE III

RESULTS FOR SERVERS NOT IN TABLE II THAT ARE ERRORED IN AT LEAST ONE EXPERIMENT. SERVERS IN NATIONAL LABORATORIES HAVE BOLD URL'S. **LISTNEW** SERVERS HAVE DISTINCT COLORED SERVERIDS MATCHING THE COLOURS USED IN SECTION V.

zones. Of the 20 **H** servers, 14 are constantly in error, with errors ranging from sub-ms to over an hour.

To quantitively summarize errors, we evaluated the total number, #**E**, of error zones for each **R** and **C** server, and for **R**, **C** and **H** servers, the aggregate duration metric *Etime*, representing the average probability of encountering an error. Of the non-**H** servers, #65 is the worst in this respect, its 27 error zones covering 8.2% of the trace, whereas #79, with 90 errors, covers only 0.25%. For completeness, per-server values for #**E**, and all other metrics defined in this section, are given in the appendix in Table V.

B. Protocol Connections

By protocol behaviour of a server, we mean the values of the Stratum and Leap Indicator (LI) fields in NTP response headers. The value LI = 3 signals that the server clock is *unsynchronized*. According to the NTPv4 standard, a system clock should in fact set both LI = 3, and S = 16 when unsynchronized (at startup or at any other time). However in transmitted packets, the standard requires that S = 16 is represented as S = 0. Thus, the expected behaviour is that each response has either $(S = 0, LI = 3)$, or $(S \neq 0, LI \neq 3)$. An exception was introduced in *ntpd* version beginning 4.2.8p3-RC2 (July 2015, [10]) for responses sent in a small interval

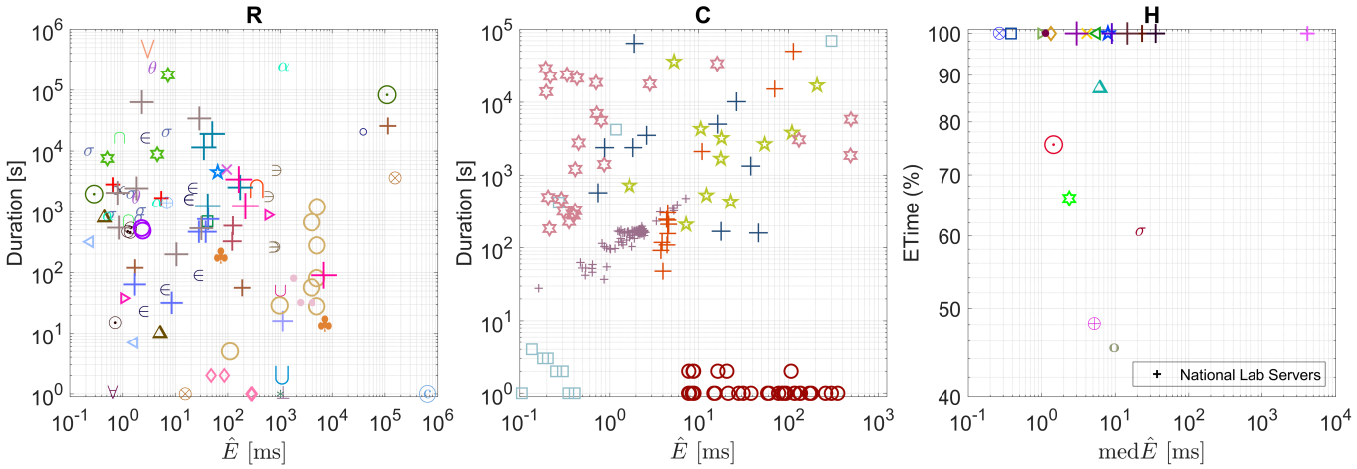

Fig. 6. Individual server anomalies characterized by error duration and size. Servers are grouped according to Prevalence, from left to right: **R**, **C**, **H**. For **H** servers, Etime instead of duration is plotted. The anomalies of each server share a unique symbol, and color. National Laboratory errors all use plotting symbol "+" of varying sizes.

ID #	URL	(Exp1, Exp2, Exp3) Strata Seen {0, 1, 2, ..., 16}	Detection {G}
3	ntp.brisbane.nmi.gov.au	{–}; {1}; {1}	(–, G, G)
4	ntp.melbourne.nmi.gov.au	{–}; {1}; {1}	(–, G, G)
5	ntp.sydney.nmi.gov.au	{–}; {1}; {1}	(–, G, G)
17	a.st1.ntp.br	{–}; {1}; {0,1}	(**G**, **G**, G)
22	ntp0.fau.de	{–}; {1}; {1}	(G, G, G)
26	ntps1-0.cs.tu-berlin.de	{–}; {1}; {0,1,16}	(G, G, G)
29	ptbtime2.ptb.de	{–}; {1}; {0,1}	(G, **G**, G)
37	ntp2.tp.pl	{–}; {1,8}; {0-2}	(G, G, G)
49	ntp1.sp.se	{–}; {1}; {0,1}	(G, G, G)
50	ntp1.sth.netnod.se	{–}; {1}; {1}	(G, G, G)
52	ntp2.mmo.netnod.se	{–}; {1}; {1}	(G, G, G)
59	ntp0.nl.uu.net	{–}; {1}; {–}	(G, **G**, –)
60	ntp4.linocomm.net	{–}; {1}; {–}	(G, **G**, –)
70	ntp.your.org	{–}; {1}; {1}	(**G**, G, G)
71	ntp1.conectiv.com	{–}; {1}; {1}	(G, G, G)
90	time.ufe.cz	{–}; {1,2}; {0,1,11}	(G, G, G)

TABLE IV
NON-ERRORED SERVERS (**G** IN ALL EXPERIMENTS).

.

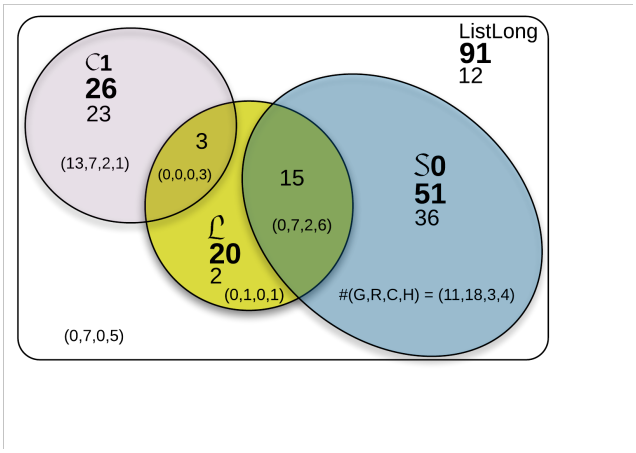

Fig. 7. Breakdown of servers according to protocol exceptions, with server error context. The bold numbers are the server populations of the named sets, and the numbers below them the populations of the immediate subset.

$[-1.5, 1]$ seconds about a leap second. For the purposes of this paper we have excised any such packets.

We find three kinds of exceptions to the expected behaviour above, all of which point to implementation bugs of some kind. First, for a single packet in **R** server #99, we find $(\mathrm{S} = 0, \mathrm{LI} = 0)$. Second, although rare, $\mathrm{S} = 16$ is observed in a small number of servers, namely #{9,26,87}. For simplicity in what follows we map $\mathrm{S} = 16$ to $\mathrm{S} = 0$. Finally, we encounter many instances of $(\mathrm{S} \neq 0, \mathrm{LI} = 3)$, which we call 'excess LI' or *L responses*. We measured for each server, the extent of this via the ratio $\rho_L = Ltime/(Ztime + Ltime) \in [0, 1]$, where $Ztime$ and $Ltime$ are respectively the proportion of responses with $\mathrm{S} = 0$, or *L responses*. Although many servers exhibit the ideal $\rho_L = 0$, just as many have ρ_L close to 1, the worst possible.

Figure 7 provides an overview of the distribution of protocol exceptions over the space of all servers. The sets shown are $\mathcal{S}0$: servers with at least one $\mathrm{S} = 0$ response, $\mathcal{L}$: servers with at least one *L response*, and $\mathcal{C}1$: servers where stratum is a constant $S = 1$ in all responses. In each resulting subset, the actual error behaviour is summarised as server counts for the {**G**,**R**,**C**,**H**} classification. The ideal picture would have $\mathcal{C}1 \cap \mathcal{L} = \emptyset$, $\mathcal{L} = \emptyset$, and $\{\mathbf{G}, \mathbf{R}, \mathbf{C}, \mathbf{H}\} = \{26, 0, 0, 0\}$ within $\mathcal{C}1$. The figure shows that the departures from this are significant, with only 13 servers enjoying both no errors accompanied by the expected zero warnings. We next dig deeper to consider the structure of errors and protocol behaviour both temporally and semantically within each server.

The tagged server errors can be used to partition each time series into errored **E**-zones, alternating with error-free zones. The same can be done with respect to protocol: we define a protocol or **P**-zone as a contiguous sequence of responses that each carries an anomalous protocol 'warning', that is either an L response, or $\mathrm{S} \neq \mathbf{S}\mathrm{N}$, where $\mathbf{S}\mathrm{N}$ is the nominal stratum (that which occurs in over 90% of responses), or both. Typically $\mathbf{S}\mathrm{N} = 1$, however $\mathbf{S}\mathrm{N} = 2$ for servers #{21,35,66,95}, and $\mathbf{S}\mathrm{N} = 3$ for servers #{1,39}. We measure the number, #**P**, of **P**-zones for each server, and the corresponding proportion, Ptime,

of the data they cover. The Ptime for **G** servers, at 0.02% on average, is far smaller than that of errored servers, being $\{0.14, 4.9, 5.2\}\%$ for $\{\mathbf{R}, \mathbf{C}, \mathbf{H}\}$ respectively (see Table V for full details).

We now examine in detail the connection between error and protocol behaviour. Three joint statistics for **R** and **C** servers are central here. First, #**E**&**P** gives the number of **E**-zones which intersect with at least one **P**-zone. Comparing to the total number #**E** of **E**-zones, we find that typically errors are not externally detectable by protocol means: 234 of 316 **E**-zones (74%) occurred with *no protocol indication of any kind*. As for **H**-servers, the Ptime values are orders of magnitude below Etime, and in 4 cases Ptime$= 0$ while Etime$= 100\%$! Speaking generally again, even if there is some protocol warning, it is typically partial, extending over only a fraction $\rho_P < 100\%$ of the error zone duration. For most servers the average of these warning fractions over all error zones with warnings is under 50%.

Next, we consider P$\in$E, which gives the fraction of Ptime-lying within **E**-zones. It is generally high, and reaches 72.6% and 41.8% when averaged over **R** and **C** servers respectively. Consistent with this, a closer inspection of the **P**-zones shows that they are generally both more frequent, and longer, within **E**-zones than outside them. Thus, although most errors are not flagged by the protocol, when protocol warnings do occur they are mainly associated with actual errors. This is all the more significant when one considers the density of warnings within **E**-zones compared to without, since Etime is only a small to very small percentage of the total trace duration for **R** and **C** servers. Simply put, the background rate of warnings not associated to server errors is very low.

C. Are Protocol Exceptions Predictive?

To gain further insight into when protocol warnings may be predictive of server errors, and when they are not, it is necessary to consider the internal structure of **P**-zones. To that end we define a number of protocol pattern types using regular expressions in the symbols $\{\uparrow, \downarrow, L, \oslash\}$, delimited by nominal responses denoted $\mathbb{SN}$ with $(\mathrm{S} = \mathbb{SN}, \mathrm{LI} = 0)$, where $\oslash$ denotes a response with $\mathrm{S} = 0$, and $\uparrow$ (resp. $\downarrow$) denote that $S > \mathbb{SN}$ (resp. $S < \mathbb{SN}$). Note L is disjoint from $\uparrow$ and $\downarrow$ since all L responses have $S = \mathbb{SN}$. We adopt the familar notation whereby $[x \cdots y]+$ denotes one or more symbols, and $[x \cdots y]*$ zero or more symbols, from the set $\{x, \cdots y\}$.

Each pattern *type* is defined by the first non-$\mathbb{SN}$ symbol which appears, and contains one or more sub-types, according to the presence of additional symbols. Patterns matching the definitions below are therefore disjoint by construction.

$\uparrow$-type
$P_\uparrow$: $\mathbb{SN}\ [\uparrow]+\ \mathbb{SN}$
$P_{\uparrow,\oslash}$: $\mathbb{SN}\ [\uparrow]+\ \oslash\ [\uparrow \oslash]*\ \uparrow\ \mathbb{SN}$

$\downarrow$-type
$P_\downarrow$: $\mathbb{SN}\ [\downarrow]+\ \mathbb{SN}$ (seen in server #95 only, $\mathbb{SN} = 2$)

L-type
P_L : $\mathbb{SN}\ [L]+\ \mathbb{SN}$
$P_{L,\uparrow}$: $\mathbb{SN}\ [L]+\ \uparrow\ [L\ \uparrow]*\ \mathbb{SN}$

$\oslash$-type
$P_\oslash$: $\mathbb{SN}\ [\oslash]+\ \mathbb{SN}$
$P_{\oslash,L}$: $\mathbb{SN}\ [\oslash]+\ L\ [\oslash L]*\ \mathbb{SN}$
$P_{\oslash,\uparrow}$: $\mathbb{SN}\ [\oslash]+\ \uparrow\ [\oslash\ \uparrow]*\ \mathbb{SN}$

The patterns are **not** designed to capture all possible words in the symbol alphabet. Rather, they were chosen to efficiently capture all patterns actually present in the data. We consider two statistics for pattern type for each server: *Plist*, which lists all types seen, and *Pvec*, their corresponding populations.

The distribution of pattern types both within **E**-zones (for **R**, **C**, and the error component of **H** servers) and outside **E**-zones (for **R**, **C**, **G**, and non-errored component of **H** servers) was exhaustively examined. The following emerged.

1: Most patterns are $P_\oslash$. The sub-type vector $\{P_\uparrow, P_{\uparrow,\oslash}, P_\downarrow, P_L, P_{L,\uparrow}, P_\oslash, P_{\oslash,L}, P_{\oslash,\uparrow}\}$ has empirical histogram $\{7.2, 0.073, 0.64, 5.6, 0.032, 85.6, 0.61, 0.33\}\%$. Thus the most common type is $\oslash$, dominated by the simple sub-types $P_\oslash$. For each type the simple subtype dominates.

2: The pattern mix outside error zones is different. The sub-types $\{P_\uparrow, P_{\uparrow,\oslash}, P_\downarrow, P_L, P_{L,\uparrow}, P_\oslash, P_{\oslash,L}, P_{\oslash,\uparrow}\}$ have histogram $\{17.7, 0.44, 0.15, 19.7, 0.22, 56.5, 3.4, 1.9\}\%$ within **E**-zones, dominated by $P_\oslash$ but with appreciable $P_\uparrow$ and P_L, compared to $\{5.4, 0.012, 0.72, 3.2, 0, 90.4, 0.15, 0.061\}\%$ without, which is almost exclusively $P_\oslash$. We find no patterns of sub-type $P_{L,\uparrow}$ at all outside of error zones, and almost none of $P_{\uparrow,\oslash}$ or $P_{\oslash,\uparrow}$, however the small number of pattern samples precludes strong conclusions about their use as error predictors.

3: Error zones intersect either no patterns, or just one. In only 13 cases over all servers did an **E**-zone intersect more than one pattern. Otherwise, as reported above, there was a single pattern in 30% of cases (which covered only a portion of the error zone), or none 70% of the time.

4: Patterns are not associated to reboot events. It is natural to consider that some protocol patterns could be explained by server reboot events. For example the extended $P_\oslash$ pattern $\mathbb{SN}\,[M]*\,[\oslash]+\,\mathbb{SN}$ could capture a reboot event where $[M]*$ denotes a *gap*, that is a number of contiguous missing responses due to the server being down, and $[\oslash]+$ a period after rebooting before the server has re-established a lock on its reference. Furthermore, we expect the *refid* header field to carry the value STEP or INIT for such $\oslash$ responses. In fact, out of all patterns only 28.3% were associated with a gap. Almost all of these were $P_\oslash$ patterns, however of these only 2.4% satisfied all the above criteria. We conclude that reboot events can explain at most only a very small percentage of protocol events.

D. Shared Anomalies

We checked systematically across all tagged anomalies to see if any were *shared* across servers. By this we mean anomalies in different servers with precisely the same temporal location (up to sampling limitations), the same size up to baseline uncertainty, and the same temporal shape, suggesting a causal link. There are two mechanisms which could give rise to such: (i) a shared reference timing infrastructure

exhibiting anomalies, (ii) pathological network switching or routing generating precisely equal and opposite changes in delay, synchronously, in the two directions.

No shared anomalies were found. However, we uncovered a total of 10 events with a signature approximating that of a server error. Each was shared over the same set of 14 European servers, namely #{(15,16), (23,24), (31,33), (26,27), (28,29), 85, (97,98), 101} and each had exactly the same LS-like signature, being nearly antisymmetric abrupt level shifts in one-way delays: $\Delta\underline{d}^{\uparrow} = -8\,\mathrm{ms}$, $\Delta\underline{d}^{\downarrow} = 9\,\mathrm{ms}$, resulting in $\Delta\underline{a} = 17\,\mathrm{ms}$, and $\Delta\underline{r} = -1\,\mathrm{ms}$, followed 1 or 2 minutes later (in one case 40min) by a return to the original values.

To independently confirm the network origin of these events, network administrators were contacted. All those who replied confirmed that their servers made use of stand-alone, unshared reference sources. Moreover, TTL values extracted from the response headers showed a change in hop count precisely in those packets forming the anomalies. Being confirmed as routing in origin, these anomalies are **not** included in our results on server errors above.

Without a careful methodology, such pathological routing anomalies could be mistaken for server errors. The following general comments can be made: 1) A sufficient condition to conclude that an anomaly is due to a server is that $\underline{a} \notin [-\underline{r}, \underline{r}]$, since no network effect can break this causality bound, 2) Any anomaly which is not of a simple LS type is even more pathological, suggesting the origin is on the server side.

VI. Prior Work

Much of the extensive literature on network timing relates to clock synchronisation algorithms in various contexts, for example [11], [8], [12], [13], [14], [15], [16]. There are a number of surveys of the NTP eco-system [17], [18], [19], [20], [21], [22], as well as less formal implementation and operational accounts [23], [24], however, extreme configuration issues aside, the accuracy of server timestamps themselves is typically not questioned, and rarely studied in any detail. For example the recent work [7], the most comprehensive eco-system survey by far, simply states that servers are taken as reputable if they are of Stratum-1 or 2. One of the few works which utilises a client-side reference infrastructure is [25], which examines data from 63 (resp. 361) S1 servers with polling period 90s (resp. 90min.) over 24hrs (resp. 115hrs), and states that only 10% – 20% of S1 servers are accurate to sub–ms, however this is not based on any methodology enabling congestion, routing and server effects to be disambiguated. The only works we are aware of which have both the reference data, and the techniques, to make robust conclusions on deployed servers, in particular S1 servers, are [3] and [4]. We re-examine the data from these works, improve upon the techniques used, significantly extend the depth and breadth of the analysis, and make the raw data, server error labelled data, and tagging tool available. A complementary work is our paper [26], which mines the data from **Exp3** with a different aim: the discovery of the best S1 servers from the perspective of several metrics, including server errors, server availability, and performance around leap seconds.

As a separate point, we note that the impact on clients of errors in its server will depend strongly on many factors including the clock synchronization algorithm in use, availability of back-up servers, the size and duration of server errors, and the path characteristics. Potential errors can range from negligible ($< 10\,\mu\mathrm{s}$) and short-term (few seconds) at one extreme, to permanent and extreme (10's of ms to seconds or well beyond plus high variability) at the other. The onus on the Stratum-1 server is to show near perfect behaviour to anchor the performance of its serving tree.

VII. Discussion

The message from the Longitudinal study of Section IV, and the detailed analysis of **Exp3** in Section V, is that the core of the Internet's timekeeping infrastructure has significant and ongoing problems. We have been able to show conclusively that the clock errors in Stratum-1 servers are widespread and diverse, are often of significant size, can last from hours to years, and are typically not flagged by the protocol. Our earlier work on these data sets ([3], [4]), demonstrated that errors were often present, but were not able to survey them thoroughly as we do here. In particular, our methodology has produced 'tag files', recording the output of this survey in detail, which opens up the following opportunities.

Root cause analysis and remediation Remote root cause analysis is likely to yield incomplete and non-definitive results. Server administrators are far better placed in this regard. We have contacted those responsible for errored servers, making available not only our findings, but the detailed tagging files. The latter provides precise and exhaustive error locations together with characterization, which can be checked against internal logs and other local metadata to track down the underlying factors efficiently, helping administrators locate the cause(s) for themselves more quickly.

We have been disappointed in the response from server administrators in terms of them reporting the specific causes of, and fixes for, the anomalies we have observed and reported. However, root cause analysis is not central here. Consider the problem we are faced with in more general terms. Remote root cause analysis may help to remediate today's errors more quickly and promote best practice, but it does not remove the longer term need for health monitoring. Our work is dedicated to advancing the latter, which can be used not only in longitudinal studies, but also in ongoing monitoring of critical timekeeping infrastructure, or even from within a future timing system, to police both performance and security goals. Transparent, reliable and readily available error reporting is a basic requirement to achieve a trusted system in the future, as sources of error come and go.

Network measurement re-evaluation The networking community, in particular the measurement community, routinely relies on client timestamps derived via the hierarchy from some S1 server assumed to be perfect. As (most of) the **ListLong** servers have been widely used over an extended period, the tag files may enable retrospective vetting of historical timestamp data in some cases, potentially triggering re-examinations of findings. More generally, the server

identification provided here should be a useful resource for future measurement campaigns seeking reliable servers in the medium term.

Detector development The tag files constitute a labelled dataset detailed enough for use in the development and evaluation of server error measurement techniques. They enable a step by step approach, as they support the testing for sub-problems such as Nice Zone, Context Zone and Server Anomaly Zone selection, **H** server detection, and reliable asymmetry measurement, separately or jointly. In our future work we expect this to be a critical resource toward achieving the goal of a fully automated classifier, capable of rapidly processing long, complex, traces with high precision and recall.

ACKNOWLEDGMENT

Partially supported by Australian Research Council's Linkage Projects funding scheme #DP170100451.

REFERENCES

[1] D. L. Mills, "Internet time synchronization: the Network Time Protocol," *IEEE Trans. Communications*, vol. 39, no. 10, pp. 1482–1493, October 1991.

[2] ——, *Computer Network Time Synchronization: The Network Time Protocol*. Boca Raton, FL, USA: CRC Press, Inc., 2006.

[3] K. Vijayalayan and D. Veitch, "Rot at the Roots? Examining Public Timing Infrastructure," in *Proc. of IEEE INFOCOM 2016*, San Francisco, CA, USA, April 10-15 2016. [Online]. Available: http://www.crin.eng.uts.edu.au/~darryl/Publications/Stratum1Check_camera.pdf

[4] Y. Cao and D. Veitch, "Network Timing, Weathering the 2016 Leap Second," in *Proc. of IEEE INFOCOM 2018*, Honolulu, USA, April 15-19 2018. [Online]. Available: http://www.crin.eng.uts.edu.au/~darryl/Publications/LeapSecond2016_camera.pdf

[5] ——, "TimeServer Dataset 2016-2017," 2019, https://data.research.uts.edu.au/public/DVTSD/. [Online]. Available: https://data.research.uts.edu.au/public/DVTSD/

[6] ntp.org, "NTP Pool Project," 2018, [Online; accessed 2-May-2018]. [Online]. Available: http://www.pool.ntp.org/en/

[7] T. Rytilahti, D. Tatang, J. Köpper, and T. Holz, "Masters of Time: An Overview of the NTP Ecosystem," in *2018 IEEE European Symposium on Security and Privacy*. IEEE, 2018, pp. 122–136.

[8] D. Veitch, J. Ridoux, and S. B. Korada, "Robust Synchronization of Absolute and Difference Clocks over Networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 417–430, April 2009. [Online]. Available: http://www.crin.eng.uts.edu.au/~darryl/Publications/synch_ToN.pdf

[9] NIST, 2019, ftp://time.nist.gov/pub/leap-seconds.3629404800. [Online]. Available: http://www.ietf.org/timezones/data/leap-seconds.list

[10] M. Burnicki, "Ntp li setting over leap second," http://bugs.ntp.org/show_bug.cgi?id=3537, 2018.

[11] D. L. Mills and P.-H. Kamp, "The Nanokernel," in *32nd Annual Precision Time and Time Interval (PTTI) Meeting*, Reston VA, November 2000, pp. 423–430.

[12] (2019) The Precision Time protocol (PTP), ptpd. http://ptpd.sourceforge.net/.

[13] K. Correll, N. Barendt, and M. Branicky, "Design Considerations for Software Only Implementations of the IEEE 1588 Precision Time Protocol," in *ISPCS*. Zurich, Switzerland: IEEE Computer Society, Oct. 10-12 2005, 6 pages.

[14] S. K. Mani, R. Durairajan, P. Barford, and J. Sommers, "Mntp: Enhancing time synchronization for mobile devices," in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16. New York, NY, USA: ACM, 2016, pp. 335–348. [Online]. Available: http://doi.acm.org/10.1145/2987443.2987484

[15] P. MOREIRA, J. SERRANO, T. WLOSTOWSKI, P. LOSCHMIDT, and G. GADERER, "White Rabbit: Sub-Nanosecond Timing Distribution over Ethernet," in *Proc. ISPCS 2009*, 200p.

[16] K. Lee, H. Wang, V. Shrivastav, and H. Weatherspoon, "Globally Synchronized Time via Datacenter Networks," in *Proc. ACM SIGCOMM*, 22-26 Aug. 2016.

[17] D. L. Mills, "On the accuracy and stablility of clocks synchronized by the network time protocol in the internet system," in *Computer Communication Review*, no. 1, 1989, pp. 65–75.

[18] J. D. Guyton and M. F. Schwartz, "Experiences with a survey tool for discovering network time protocol servers," in *Proc. USENIX Summer Conference*, 1994, pp. 257–265.

[19] N. Minar. (1999) A Survey of the NTP Network. Http://xenia.media.mit.edu/ nelson/research/ntp-survey99/ntp-survey99-minar.ps. [Online]. Available: http://alumni.media.mit.edu/~nelson/research/ntp-survey99/html/

[20] J. D. Guyton and M. F. Schwartz, "Experiences with a Survey Tool for Discovering Network Time Protocol Servers," 1994, [Online; accessed 31-July-2015]. [Online]. Available: http://static.usenix.org/publications/library/proceedings/bos94/full_papers/guyton.a

[21] D. Malone, "The Leap Second Behaviour of NTP Servers," in *Proceedings of the Traffic Monitoring and Analysis workshop*. IFIP Digital Library, April 7-8 2016, http://tma.ifip.org/2016/#program.

[22] C. Murta, P. Torres, and P. Mohapatra, "Characterizing quality of time and topology in a time synchronization network," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, Nov 2006, pp. 1–5.

[23] M. Burnicki, "Technical Aspects of Leap Second Propagation and Evaluation," in *Requirements for UTC and Civil Timekeeping on Earth Colloquium*, ser. Science and Technology Series, vol. 115. Univelt Inc., San Diego, 2015, aA 13-504.

[24] M. Lichvar, "Five different ways to handle leap seconds with NTP," July 1st 2015, http://developerblog.redhat.com/2015/06/01/five-different-ways-handle-leap-seconds-ntp/.

[25] C.-Y. Hong, C.-C. Lin, and M. Caesar, "Clockscalpel: Understanding root causes of Internet clock synchronization inaccuracy," in *Proceedings of the 12th international conference on Passive and active measurement*, ser. PAM'11, N. Spring and G. Riley, Eds., vol. 6579. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 204–213. [Online]. Available: http://dl.acm.org/citation.cfm?id=1987510.1987531

[26] Y. Cao and D. Veitch, "Where on Earth are the Best-50 Time Servers?" in *Proc. of PAM 2019*, Puerto Varas, Chile, March 27-29 2019. [Online]. Available: http://www.crin.eng.uts.edu.au/~darryl/Publications/Best50_PAM2019_camera.pdf

Yi Cao received the B.Sc. degree from Nanjing University of Science and Technology, Nanjing, China, in 2005, and the M.Sc. and Ph.D. degrees from the University of Wollongong, NSW, Australia, in 2011 and 2016, respectively, all in electrical, electronic, and information science. He was an electrical and systems engineer at Shanghai Eighth Division of China Aerospace Science and Technology Corporation, and a postdoctoral Research Associate at the University of Technology Sydney. He is now an IT Technical officer working at the National Institute for Applied Statistics Research Australia. His interests span networking and data analysis.

Darryl Veitch completed a BSc. Hons. at Monash University, Australia (1985) and a mathematics Ph.D. from DAMPT, Cambridge (1990). He worked at TRL (Telstra, Melbourne), CNET (France Telecom, Paris), KTH (Stockholm), INRIA (France), Bellcore (New Jersey), RMIT (Melbourne), Technicolor (Paris) and EMUlab and CUBIN at The University of Melbourne, where he was a Professorial Research Fellow until end 2014. He is now a Professor in the School of Electrical and Data Engineering at the University of Technology Sydney. His research interests are centered around computer networking and inference and include traffic modelling, parameter estimation, the theory and practice of active measurement, traffic sampling and sketching, information theoretic security, and clock synchronisation over networks, and drones. He is a Fellow of the IEEE.

APPENDIX / SUPPLEMENTARY MATERIAL

Server Anomaly Tag format Each potential server anomaly has an associated tag record, with the following fields.

1) *location_UTC*: start and end time of **E**-zone in UTC
2) *SABaselineWindows*: a vector containing the width (in # samples) of left and right components of the Context Zone (CZ) surrounding the Server Anomaly Zone (SAZ)
3) *ANice*: flag indicating whether underlying path asymmetry was constant or not over the Nice Zone
4) *uAandBL*: a vector $(\hat{\underline{a}}, \hat{\underline{r}}_{NZ})$ containing underlying asymmetry and baseline estimates in seconds
5) *expertise*: assessment of **E**-zone candidate complexity, values in $\{$Nice, Clean, Complex, Hard$\}$
6) *type*: values in $\{$Drift, Level Shift, Skew&Return, Other$\}$
7) *size*: measured error size $\hat{E}$ in seconds
8) *statistic*: measured 'significance ratio' μ
9) *readiness*: tag status, values in $\{$Ready, Unfinished, Checkit, Deleted$\}$

Table V holds the detailed per-server results for the metrics, introduced in Section V, that describe the protocol and error findings for servers in **ListNew** from experiment **Exp3**. For convenience the definition of these metrics are collected together in Table VI below.

G ID	3	4	5	9	17	22	26	29	37	49	50	52	64	70	71	74	84	89	90	92	93	94	97	101	Ave.
#P	0	0	0	5	3	0	**572**	1	2	2	0	0	1	0	0	0	0	0	1	0	1	1	2	0	24.6
Ptime(%)	0	0	0	$2.3e^{-3}$	$1.1e^{-4}$	0	**0.48**	$7.3e^{-5}$	$4.8e^{-4}$	$1.1e^{-4}$	0	0	$3.0e^{-4}$	0	0	0	0	0	$1.9e^{-3}$	0	$4.1e^{-4}$	$4.1e^{-4}$	$3.3e^{-3}$	0	0.02
Ztime(%)	0	0	0	$2.3e^{-3}$	$1.1e^{-4}$	0	**0.48**	$7.3e^{-5}$	$1.9e^{-5}$	$1.1e^{-4}$	0	0	$1.9e^{-5}$	0	0	0	0	0	$1.8e^{-5}$	0	$3.7e^{-5}$	$7.5e^{-5}$	$3.3e^{-3}$	0	0.02
ρ_L(%)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Plist	·	·	·	⊘	⊘	·	⊘	⊘	⊘,↑	⊘	·	·	⊘	·	·	·	·	·	⊘	·	⊘	⊘	⊘	·	·
Pvec	·	·	·	5	3	0	**572**	1	1,1	2	·	·	1	·	·	·	·	·	1	·	1	1	2	·	·
lossrate(%)	0.1	0.1	1.0	0.2	0.3	0.6	**19.0**	0.7	2.3	0.4	0.2	0.1	0.1	3.2	0.6	0.4	0.2	0.1	0.3	3.0	0.8	0.9	3.3	1.1	1.6

R ID	2	6	7	11	12	16	18	19	23	28	30	31	33	34	35	36	38	42	43	45
#E	3	4	4	1	1	1	4	2	6	1	3	2	2	1	1	2	1	7	2	4
Etime(%)	0.12	0.13	$1.1e^{-4}$	0.08	0.012	0.089	0.026	0.018	0.63	$2.9e^{-4}$	3.5	$5.8e^{-3}$	0.014	0.025	**8.5**	4.0	0.366	1.9	$3.5e^{-3}$	0.61
#P	12	6	0	4	1	1	3	3	13	1	10	1	1	0	0	7	1	**14**	3	6
Ptime(%)	0.096	0.04	0	0.011	$5.6e^{-3}$	$5.5e^{-3}$	$1.3e^{-4}$	1.6	0.38	$2.9e^{-4}$	0.38	$4.6e^{-3}$	$4.4e^{-3}$	0	0	**2.2**	0.023	0.023	$1.1e^{-3}$	$6.3e^{-3}$
Ztime(%)	0	0.037	0	$2.4e^{-4}$	$9.1e^{-5}$	$7.4e^{-5}$	$1.3e^{-4}$	$8.9e^{-5}$	0	$2.9e^{-4}$	0	0	0	0	0	$7.5e^{-5}$	$5.2e^{-3}$	$1.1e^{-3}$	$6.6e^{-4}$	$7.2e^{-4}$
ρ_L(%)	0	0	0	97.9	98.4	98.6	0	0	0	0	0	0	0	0	0	0	0	95.4	31.4	88.6
Plist	↑	⊘	·	⊘	⊘	⊘	⊘	⊘,↑	↑	⊘	↑	↑	↑	·	·	⊘,↑	⊘	⊘,L	⊘	⊘
Pvec	12	6	·	4	1	1	3	2,1	**13**	1	10	1	1	·	·	2,5	1	11, 3	3	6
#E&P	0	4	·	1	1	1	3	2	4	1	3	1	1	·	·	2	1	1	2	3
P∈E(%)	0	76.2	0	100	100	100	100	$5.6e^{-3}$	5.9	**100**	99.4	**100**	**100**	0	0	99.6	**100**	23.3	56.4	98.2
$\overline{\rho_P}$(%)	·	63.2	·	8.1	66.6	7.7	0.5	0.6	32.6	50	47	80	30.9	·	·	40.3	6.2	55.1	54.9	4.8
lossrate(%)	0.1	0.6	0.1	0.2	0.1	0.2	2.0	9.3	0.3	0.2	5.0	0.2	0.2	2.5	0.2	1.2	0.3	2.0	1.3	1.2

R ID	46	47	48	51	53	54	55	56	58	67	75	76	77	82	85	86	96	98	99	100	Ave.
#E	3	1	1	1	1	1	2	2	1	**8**	1	3	3	2	7	5	4	3	1	2	2.6
Etime(%)	0.085	$1.8e^{-5}$	$3.6e^{-5}$	$1.8e^{-5}$	0.041	$8.8e^{-4}$	1.5	0.065	$1.8e^{-5}$	0.042	0.043	0.31	0.46	0.016	0.40	0.032	4.5	$2.6e^{-3}$	$1.8e^{-5}$	0.016	0.69
#P	8	0	0	0	0	0	3	1	2	9	2	7	1	2	1	3	2	8	1	4	3.5
Ptime(%)	0.082	0	0	0	0	0	0.032	$7.1e^{-3}$	$9.2e^{-3}$	0.028	0.011	0.31	0.18	0.050	$2.8e^{-4}$	0.016	$4.0e^{-3}$	0.010	$1.2e^{-3}$	$2.9e^{-4}$	0.14
Ztime(%)	0.078	0	0	0	0	0	0.025	$7.1e^{-3}$	$3.7e^{-5}$	0	$4.6e^{-3}$	0	**0.18**	0.050	$1.9e^{-5}$	$2.9e^{-4}$	$4.0e^{-3}$	0.010	$1.2e^{-3}$	$2.2e^{-4}$	0.01
ρ_L(%)	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	98.2	0	0	0	0	17.7
Plist	⊘,↑	·	·	·	·	·	⊘	⊘	⊘	L	⊘,↑	↑	⊘	⊘	⊘	⊘,L	⊘	⊘	⊘	⊘,↑	·
Pvec	7,1	·	·	·	·	·	3	1	2	9	1,1	7	1	2	1	1,2	2	8	1	3,1	·
#E&P	3	·	·	·	·	·	2	1	1	**6**	1	3	1	1	0	3	1	3	0	1	1.8
P∈E(%)	94.3	0	0	0	0	0	**100**	**100**	0.2	99.9	97.3	95.8	**100**	93	0	**100**	**100**	36.4	0	25	72.6
$\overline{\rho_P}$(%)	58.9	·	·	·	·	·	10.9	11.7	**100**	46.7	24.1	79.4	37.8	**100**	·	64.3	0.09	**100**	·	1.6	40.8
lossrate(%)	7.0	0.1	0.1	0.1	1.5	0.2	0.1	0.2	0.4	0.5	0.4	0.2	33.7	**49.5**	1.1	1.0	0.5	8.3	2.2	0.6	3.3

C ID	8	24	44	65	79	87	95	Ave.
#E	50	11	10	27	**90**	12	12	30.3
Etime(%)	$1.0e^{-3}$	1.3	1.6	**8.2**	0.25	1.2	1.3	2.0
#P	0	2	6	27	0	**6806**	368	1.0e3
Ptime(%)	0	0.013	$3.7e^{-4}$	**21**	0	2.7	11	4.9
Ztime(%)	0	$3.3e^{-4}$	$3.7e^{-4}$	$6.1e^{-3}$	0	**2.7**	$4.0e^{-5}$	0.39
ρ_L(%)	0	**97.4**	0	79.2	0	0	0	25.2
Plist	·	⊘,L	⊘	⊘,↑	·	⊘	⊘,↑,↓	·
Pvec	·	1,1	6	6,21	·	**6806**	2,305,61	·
#E&P	·	2	2	13	·	4	3	4.8
P∈E(%)	0	**100**	85	23.1	0	0.7	0.049	41.8
$\overline{\rho_P}$	·	69.6	0.6	**73.3**	·	4.0	41.2	37.7
lossrate(%)	0.1	11.2	1.0	9.4	**36.3**	7.3	4.2	9.9

H ID	1	10	13	15	21	25	27	39	41	61	63	66	68	69	73	78	80	81	88	91	Ave.
Ttime	87.0	97.8	60.7	75.5	80.6	87	45.3	92.4	48.1	85.4	87.0	87.7	46.2	63.3	39.0	73.1	82.7	74.5	26.4	48.5	67.1
Etime(%)	**100**	**100**	60.7	75.5	**100**	87	45.3	**100**	+48.1	**100**	**100**	**100**	**100**	**100**	66	**100**	**100**	**100**	**100**	**100**	89.1
#**P**	3	0	170	50	0	17	6	0	52	75	2	0	**211**	11	1	3	46	5	4	3	32.9
Ptime(%)	0.022	0	0.26	0.66	0	0.40	5.1	0	0.88	0.060	0.42	0	28	$3.7e^{-4}$	**64.7**	0.23	2.0	0.15	0.013	0.47	5.2
Ztime(%)	0.015	0	0.021	$9.2e^{-5}$	0	0	5.1	0	$7.4e^{-4}$	$7.5e^{-4}$	0	0	**5.7**	$3.7e^{-4}$	0	0	0	0	$4.4e^{-3}$	0.01	0.54
ρ_L(%)	0	0	61.1	98.3	0	0	0	0	93.8	89.0	**100**	0	80.1	0	0	**100**	**100**	**100**	65.9	0	49.4
Plist	⊘	·	⊘,↑	⊘,↑	·	↑	⊘	·	⊘,↑,L	⊘,↑	↑,L	·	⊘,L	⊘	↑	L	L	L	⊘,L	⊘,↑	·
Pvec	3	·	120, 50	1,49	·	17	6	·	4,47,1	2,73	1,1	·	6,**205**	11	1	3	46	5	3,1	2,1	·
lossrate(%)	0.1	0	2.8	0.4	0.2	0.3	**17.3**	0.3	1.6	2.6	10.6	1.3	4.2	1.3	0.2	0.5	0.4	0.3	0.4	0.2	2.3

TABLE V

PROTOCOL AND ERROR ZONE METRICS FOR SERVERS ACCORDING TO PREVALENCE CLASS **G**, **R**, **C**, **H**. THE METRICS #**E**&**P**, $\overline{\rho_P}$, AND P∈E ARE DEFINED ONLY FOR **R** AND **C** SERVERS. METRIC AVERAGES OVER OVER ALL SERVERS ARE GIVEN IN THE RIGHTMOST COLUMN FOR EACH CLASS. AVERAGES FOR P∈E (FRACTION OF PTIME WITHIN **E**-ZONES) AND $\overline{\rho_P}$ (PROPORTION OF E-ZONES COVERED BY PATTERNS WITHIN THEM) ARE CONDITIONAL ON HAVING AT LEAST ONE PATTERN (RESP. WARNING). THE LARGEST VALUE(S) IN EACH ROW ARE IN BOLD FOR EMPHASIS.

#**E**	number of error zones (**E**-zones)
Etime	proportion of the trace taken up by **E**-zones (per-response probability of encountering an error)
#**P**	number of protocol zones (**P**-zones)
Ptime	proportion of the trace taken up by **P**-zones
Ztime	proportion of the trace taken up by ⊘ responses ($S = 0$)
ρ_L	ratio of $Ltime/(Ztime + Ltime)$, where Ltime is the proportion of the trace taken up by L responses
Plist	list of protocol pattern types seen in the trace
Pvec	population of each protocol pattern in Plist
#**E**&**P**	number of error zones which intersected at least one **P**=zone
P∈E	proportion of Ptime falling within **E**-zones
$\overline{\rho_P}$	for **E**-zones intersected by **P**-zones, the average proportion of the **E**-zones that is intersected
lossrate	1−empirical availability (ratio of response packets received to request packets sent)

TABLE VI

DEFINITION OF THE METRICS APPEARING IN TABLE V.