The Leap Second Behaviour of NTP Servers

David Malone
Hamilton Institute and Department of Mathematics & Statistics, Maynooth University, Ireland.

***Abstract*—The NTP network is an important part of the Internet's infrastructure, and one of the most challenging times for the NTP network is around leap seconds. In this paper we look at the behaviour of public servers in the NTP network in 2005 and over the period from 2008 to present, focusing on leap seconds. We review the evolution of the NTP reference implementation with respect to leap seconds and show how the behaviour of the network has changed since 2005. Our results show that although the network's performance has certain problems, these seem to be reducing over time.**

***Index Terms*—NTP, Leap Seconds, Internet Measurement**

I. Introduction

The NTP network provides time synchronisation for much of the Internet, and also for mobile devices, small cells, etc. As relatively good clock synchronisation is important for the correct functioning of many protocols and applications (including DNS, certificate verification, filesystems and caching) the operation of the NTP network is of indirect concern to essentially all Internet users.

The NTP network consists of NTP servers, which exchange UDP packets to establish the differences between their local clocks and then steer their local clocks towards the correct time. Servers that are directly connected to *reference clocks*, which know the correct time from external sources, are said to be at stratum 1. NTP servers that are synchronised to a server at stratum 1 are at stratum 2, and so on. In addition to the basic packets used for clock synchronisation, NTP supports various packets for management.

One subtle aspect of NTP, as it is usually deployed, is that it uses UTC as a reference timescale [1]. UTC is a compromise timescale between atomic time (TAI) and mean solar time, more precisely UT1. UTC consists of fixed-length SI seconds. To allow for variation in the rotation of the Earth, UTC allows minutes with 59 or 61 seconds resulting in *leap seconds* which are used keep UTC within 0.9s of UT1. As the rotation of the Earth is slowing, and the length of the SI second is based on the rotation of the Earth in approximately 1820 [2], in practice a minute with an extra second is required roughly every 18 months to keep UTC within 0.9s of UTC. The need for a leap second is determined by the International Earth Rotation and Reference Systems Service (IERS) based on measurements of the rotation of the Earth. Announcements are typically made six months in advance of a leap second, and leap seconds are currently scheduled at the end of June or December through IERS Bulletin C[1].

While a jump of one second is unlikely to cause direct problems for end users, UTC is the legal basis for timekeeping in many countries. On a practical basis, many systems depend on sub-second synchronisation, for example distributed systems running in data centres, financial trading systems, mobile systems handling call handover or playout in protocols such as RTP. The mis-introduction of leap seconds into such systems could cause substantial issues. Indeed, during the 2015 leap second, a number of stock exchanges were reported to suspended operations as a precaution. We outline some of the issues seen in operational systems in Section VI. The continued use of leap seconds is under debate at the ITU-R, and other fora[2], with resolution COM5/1 of the 2015 World Radio Conference calling for further and wider study of the matter. The original aim of this survey was to characterise the leap second behaviour of the NTP network, particularly misbehaviour, in order to provide input to this debate on the NTP network's performance.

The NTP network attempts to handle leap seconds gracefully by propagating information about pending leap seconds by using flags in exchanged synchronisation packets. Thus, for most users the leap second is supposed to be transparently managed by NTP without intervention. Historically, a limited number of reference clocks could automatically provide information about pending leap seconds into the network, and particularly conscientious server administrators could manually introduce the information. However, anecdotal concerns about missing, misapplied or maliciously manufactured leap seconds have led to changes in the NTP protocol and configuration relating to leap seconds. This paper chronicles these changes and relates them to the observed changes in behaviour of the NTP network.

In Section II we review work relating to the behaviour of NTP. In Section III we outline how data was collected for this survey. In Section IV we discuss what we can learn about the leap second-related behaviour of the NTP network from these results. In Section V we discuss how, when a part of the NTP network incorrectly implements a leap second, it may be possible to analyse subsequently available data to identify errant external servers. Finally, we discuss our results in Section VI and conclude in Section VII.

II. Related Work

There have been a number of previous surveys of the NTP network. In 1989 Mills conducted a survey of hosts responding to the NTP, ICMP and Time protocols [3]. A tool for finding NTP servers by querying known servers was described [4]. The authors collected clock statistics and the level of branching of the NTP-tree in order to assess the pressure on high-stratum servers. In 1995, another survey identified as many NTP

[1] http://www.iers.org/SharedDocs/News/EN/BulletinC.html

[2] Steve Allan has compiled a bibliography on the future of leap seconds at http://www.ucolick.org/~sla/leapsecs/onlinebib.html.

servers as possible by walking the NTP graph [5]. The survey looked at factors such as the stratum, time/frequency offsets and the roundtrip delays observed. Followup surveys in 1999 and 2005 [6], [7], again considered factors such as stratum, offset and number of peers. All these studies considered the NTP networks at particular points in time while comparing their results to previous studies.

Other studies have looked at the synchronisation of particular classes of hosts, such as web servers [8] or desktop clients [9], tried to identify the sources of poor synchronisation [10] and shown that NTP is not the only time synchronisation protocol with leap second issues [11]. More recently, some of the monitoring features of NTP software have been exploited to carry out amplification DoS attacks, which has prompted measurement studies and configuration advice [12], [13]. A new series of attacks exploiting features of the NTP and IP protocols were discovered [14]. Clock synchronisation has been of interest for other reasons, for example clock characteristics can be used to de-anonimise hosts [15]. In general, accuracy of timestamps is considered an important factor in digital forensics [16], [17] and the impact of leap seconds on timestamps in RTP is considered in RFC 7164.

III. Data Collection

Most NTP operators do not have servers with reference clocks directly attached, and so depend on other servers to provide timing and leap second related information. There are two main mechanisms used by users and vendors to find NTP servers. First, the ntp.org website lists Stratum 1 and Stratum 2 NTP servers[3] that provide a public NTP service, which can be used by those configuring their own NTP servers. Second, challenges around the use of the NTP server lists by vendors (e.g. [18], [19]) led to the establishment of the NTP pool[4] in 2003. The NTP pool is a DNS-based mechanism to locate NTP servers providing a public service.

In 2005, using the first mechanisms, we identified a list of stratum 1 and stratum 2 servers by using the lists provided at ntp.org. Preliminary data was then collected at the 2005 leap second. Subsequently, the continental NTP pool DNS entries were queried to identify a subset of the NTP pool servers available. Long-term data collection began in November 2008 and continues to the present (February 2016 as of writing, see Fig. 1). The usual query used by NTP admins to find the leap second status of a server is `ntpq -c rv leap`. This query was issued once an hour to each server, the output was recorded and the value of the leap flags was noted. The queries were issued from a host on a production DSL network and are subject to normal connectivity problems, upgrades and other similar operational issues.

In 2012, the list of NTP servers was updated, adding new servers that had appeared on the ntp.org server lists and adding a sample of new hosts from the NTP pool. The amount of data that had been collected also warranted rewriting the collection system in perl. For a period from September 2012 until June 2013, only the leap second flags are recorded due to an issue with the directory layout used to store raw results. When this issue was resolved, the query used was replaced with `ntpq -c rv`, which asks the server for the default list of available variables. In practice, this always includes the leap flags, but also includes a number of other useful indicators. A number of other cleanups were performed at this time, including the removal of long-term unresponsive servers.

Date	Change
2005-12	Preliminary measurement.
2008-11	Measurements start.
2009-05	OS upgrade.
2011-11	OS upgrades.
2012-04	Update stratum 1, 2 & pool list.
2012-06	OS upgrade.
2012-06	Rewrite in perl. Update pool list.
2012-09	Inode subdirectory limit reached.
2012-09	OS upgrade.
2013-02	OS upgrade.
2013-02	OS upgrade. Add IPv6 queries. Prune unresponsive. Update stratum 1, 2 & pool list.
2013-06	Inode subdirectory limit resolved. Default variable list requested..
2014-01	Add `ntpdate` queries.
2014-02	Update host list using `ntpdate`.
2014-03	Update host list using `ntpdate`.
2014-06	Update host list using `ntpdate`.

TABLE I
Changes to measurement system.

In December 2014, the use of NTP as part of an amplification DoS attack became widely known. This attack made use of the `monlist` administrative command, and so many NTP administrators began blocking administrative queries, including the `rv` command used in this survey. To counter the declining response rate to requests, in January 2014 servers that no longer responded to the `ntpq -c rv` command were identified, and for these servers a simple time synchronisation query was issued using the `ntpdate` command. The response to this command includes the value of the leap second flags and a number of other pieces of information, including stratum, precision, refid, etc. The list of servers responding to `ntpdate` was updated several times in 2014. The updates to the system are summarised in Table I.

IV. Results

The preliminary data covered the leap second on Saturday December 31st 2005, and three leap seconds occurred during the main measurement period: Wednesday December 31st 2008, Saturday June 30th 2012 and Tuesday June 30th 2015. There was also the potential for a leap second on 12 other days (the remaining final days of June and December). In order to understand the behaviour of the NTP network around these times, it is useful to understand the network's propagation of leap second information, both automatically and through configuration files.

Since reference clocks providing notification of upcoming leap seconds are relatively rare, the traditional approach of ntpd was to believe that a leap second was pending at the end

[3] Servers, locations and access conditions are listed at http://support.ntp.org/bin/view/Servers/StratumOneTimeServers and http://support.ntp.org/bin/view/Servers/StratumTwoTimeServers.

[4] http://www.pool.ntp.org/

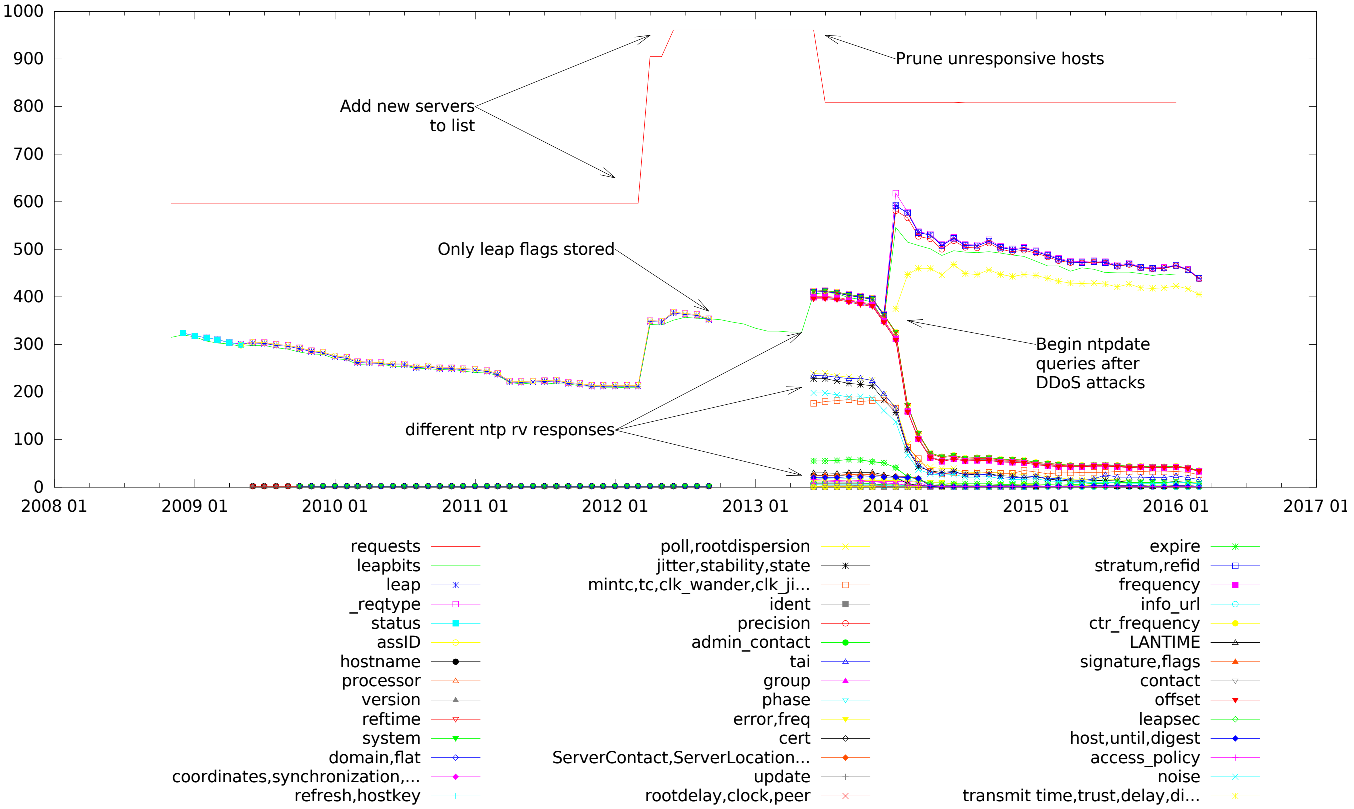

Fig. 1. Number of distinct hosts queried and responding with particular attributes. Attributes have been grouped if they always appear together.

of the month if any of the servers it was using for synchronisation advertised an upcoming leap second. Particularly attentive NTP server administrators could manually flag an upcoming leap second using the ntp administrative commands. In 2007, around fears of bogus (i.e. incorrect) leap seconds propagating through the NTP network, the rule was modified: now ntpd considers the advertisement of a leap second valid only if more than half its peers advertise it. If a more authoritative source of information is available, such as a manually configured leap second file or a locally attached reference clock, then ntpd will use this information.

Support for loading a list of historical and upcoming leap seconds was originally added to ntpd in 2000 as a side effect of new cryptographic features being added to ntpd. These extensions also allowed transfer of the table of leap seconds through the protocol associated with the cryptographic features. Anecdotally, deployment of these features beyond users with high-level security needs was minimal. In 2007, a feature that permitted the loading of the leap second file by adding a simple configuration directive to `ntp.conf` was added. In practice, this became available to ordinary ntpd users with the 4.2.6 release of ntpd at the end of 2009. Servers with a manually configured leap second file will ignore other leap second advertisements until the leap second file expires.

A. *Response Data*

Over the period, over 19 million responses to queries were recorded. A summary of the types of information recorded in the responses is shown in Fig. 1. For each month over the collection period, we show the number of requests made and the different information types returned. The graph shows that from December 2008 until September 2012 the information available consists of the leap flags and NTP status bits. `_reqtype` is a meta-response, indicating whether the request was made using `ntpq` or `ntpdate`. A small number of other variables are reported in this period, apparently by unusual NTP implementations that report hostname, processor and version strings even though they are not requested. The update to the list of queried hosts in April 2012 is also apparent in both the number of queries and responses. There is also a slight change to the way some of the data is reported from March 2009 due to an upgrade of the ntp tools used.

Starting in June 2013, we see a wider range of types of data being reported, reflecting queries requesting the default variable list. There are several different groups of variables corresponding to the default variable lists in various NTP implementations, and also manual configuration changes made by individual administrators. There is also a reduction in the overall number of requests, corresponding to unresponsive servers being pruned.

Between reconfigurations, we see a gradual decline in the number of responses, for example, from November 2008 until March 2012, there is a decline of, on average, 2.9 servers that respond per month, or a reduction of 1% per month in the responding population. The other periods of stable configuration up to December 2014 also show a reduction of about 1% per month in the responding population.

Beginning in December 2014 we see a sharp decline in the number of responses to rv queries: 10–40% per month. This reduction corresponds closely to the disclosure of the ntpd monlist DoS attack and levels out by about June 2014. While this reduction is sharp, it is not quite as dramatic as the 92% reduction over three months in the number of amplifiers reported in [13]. The sharper reduction in the number of amplifiers may be due to some administrators disabling the monlist feature to prevent the attack, rather than blocking all ntpq queries. Our consequent introduction of our ntpdate queries begins in January 2014. A significant number of new hosts provide responses to the ntpdate request, and also provide values for attributes relating to time synchronisation queries, such as delay, dispersion and various timestamps.

B. Leap Second Indicator Flags

Two bits are used in NTP packets to indicate information about upcoming leap seconds. A value of 00 indicates that no leap second is pending, a value of 01 means a leap second is to be inserted, a value of 10 means a leap second is to be removed and 11 indicates that the leap status is unknown because the clock is unsynchronised. Though RFC 1059, 1305 and 4330 state that these flags indicate a leap at the end of the current day, in practice NTP has known that the leap second can only occur at the end of the month and has processed the flag accordingly. In recognition of this, RFC 5905 now states that these flags indicate a leap second in the last minute of current month.

1) Behaviour at Scheduled Leap Seconds: The top left of Fig. 2 shows the leap indicator behaviour of the monitored servers for the preliminary measurement in 2005 for several days around the leap second. The point of the leap second is marked with a vertical line. We plot the fraction of servers in our stratum 1 and stratum 2 servers showing leap values of 01, 10 or 11.

We can see that more than 24 hours before the leap second, $< 10\%$ of stratum 1 and $< 20\%$ of stratum 2 servers know about the pending leap. The situation improves considerably in the day before, with a peak in the hour directly before the leap. These increases are likely to be due to reference clocks that provide notification of the leap either one day or one hour before the leap second. Curiously, one stratum 2 server advertises the removal of a leap second in the hour before the leap. After the leap, 10–20% of hosts fail to clear their leap indicators, but this number gradually dwindles over the following days.

Interestingly, there is also an increase in unsynchronised stratum 1 servers after the leap second. One might expect that these servers did not know about the leap second, and became unsynchronised on finding that they were one second ahead of other servers. However, looking at the individual servers involved, it appears that these servers knew about the upcoming leap second. We speculate that the problem may have been an interaction between the leap second implementation in ntpd and that of the attached reference clock.

Fig. 2 and Fig. 3 show show similar graphs around the time of potential leap seconds in the main data set (i.e. the end of each June and December), except for 2010 and 2011, which show similar behaviour to 2009. Results are shown for three days on either side of the end of the month. First, note that other than a smattering of servers in December 2008 and January 2009, we see no indicators for removal of a leap second.

As expected, only three graphs show substantial activity in the leap indicator flags, corresponding to the actual leap seconds. However, there is some evidence of unusual behaviour in the 24 hours before the end of 2012, 2013 and 2014. First, let us consider the behaviour around each of the three leap seconds, which we note is quite different.

In 2008, we see that even three days before the leap second is scheduled, already 20–30% of servers are advertising the forthcoming leap, with the percentage rising 2 days and 1 day before the leap second (up to 50–80%), and peaking the hour before the leap. We note that this is a marginal improvement compared to 2005, particularly more than 24 hours before. The proportion of stratum 2 and pool servers advertising the leap is higher than the proportion of stratum 1 servers, which is consistent with 2005. We also see a small number of servers becoming unsynchronised around the leap, about half of which share operators with those that became unsynchronised in 2005, indicating possible shared reference clock issues.

Again, a significant fraction of servers continue to advertise the leap second, in fact a larger proportion than in 2005. Most stratum 1 servers have stopped advertising the leap by mid-January, stratum 2 by mid-April and the pool before the end of June! A bug was identified that, in particular circumstances, resulted in the leap second information not being fully cleared[5]. Examining the source repository, it appears that this bug may have been introduced in overhauls of the leap second handling code in 2007, allowing leap flags to persist through a reference counting problem.

In 2012, a slightly different pattern is observed. Again, three days before the leap second there are already a number of servers advertising an upcoming leap, though a lower proportion ($< 20\%$) than seen in 2008. This time, there is no increase 48 hours before the leap, but there is a sharp increase 24 hours before the leap (to around 60–80%), again with a peak 1 hour before the leap, improving upon 2005 and 2008. This is consistent with what we expect based on the gradual adoption of majority-vote (beginning 2007), easy manual configuration of the leap second file (beginning 2007–2009) and a change in 2008 that limits the advertisement of a leap second to one day before the leap.

As in 2008, some servers continue to advertise the leap after it has occurred, though a lower proportion than in 2008. Many servers return to normal at the end of July, resulting in some discussion on mailing lists[6]. This is, presumably, the continued presence of the bug from 2007, which is resolved later in 2012.

A number of servers are unsynchronised in the period around the leap. Manually examining the logs for each of these, it seems some are simply servers that were unsynchronised for reasons apparently unrelated to the leap second

[5] https://bugs.ntp.org/show_bug.cgi?id=2246

[6] E.g. http://lists.ntp.org/pipermail/questions/2012-August/033611.html

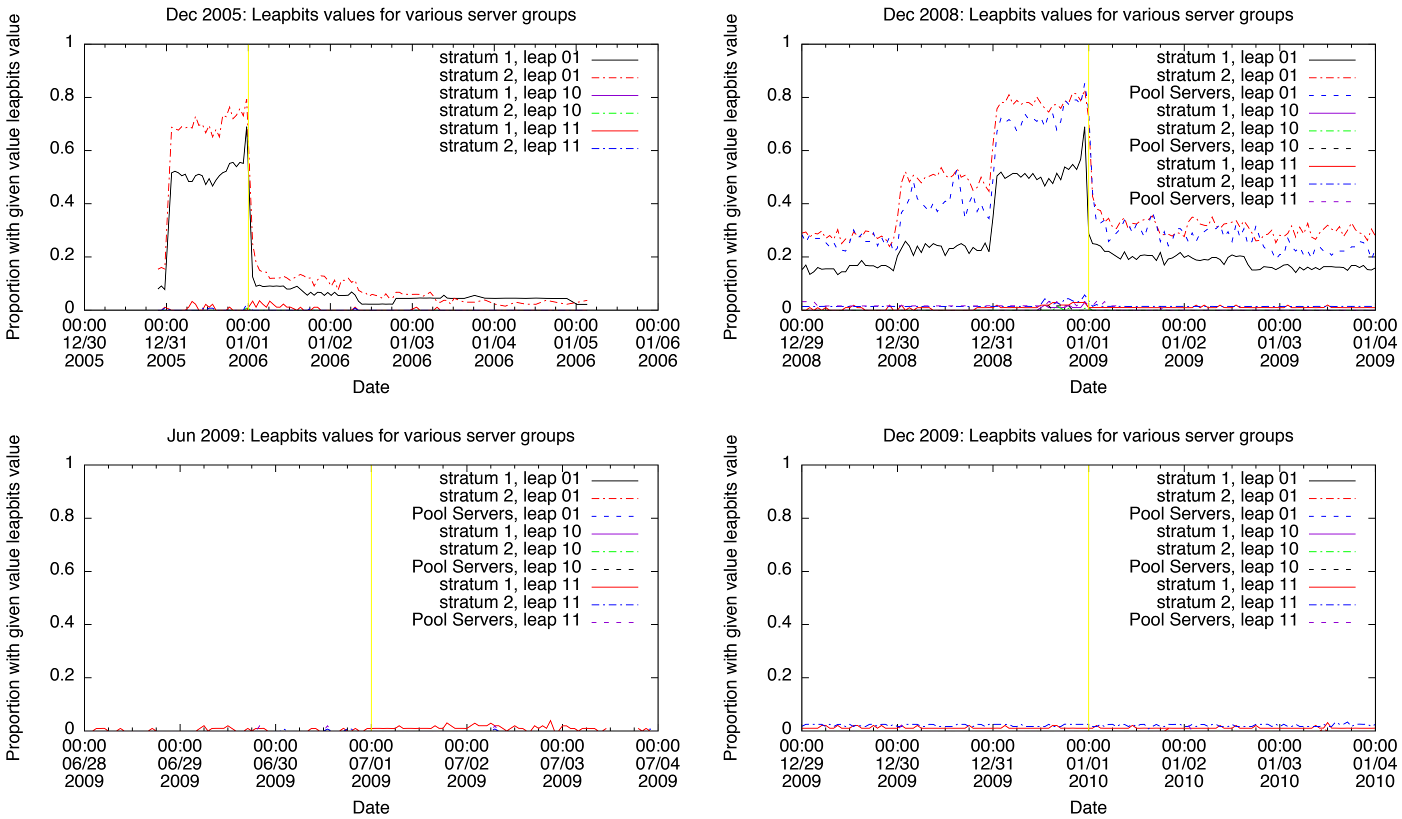

Fig. 2. Hourly behaviour of leap second flags around end of December 2005, December 2008, June 2009 and December 2009.

(e.g. becoming unsynchronised hours before the leap was scheduled, with no evidence of having stepped their clock), while a few appear to become unsynchronised at the leap because they were not aware that it was pending.

By June 2015, many NTP operators will have upgraded their ntpd to versions from early 2015 because of the DoS amplification attack. Now, almost no servers advertise the pending leap second until 24 hours before it is due. Then 60–80% of servers begin to advertise the leap second, again with a peak one hour before the leap. Behaviour after the leap second is considerably better in 2015, with the fraction of servers advertising the leap falling below 10% within an hour of the leap. This may either be due to deployed copies of the leap second file, which would reject leap seconds, or a change made in late 2013, which suppresses leap second advertisement during the first hour of each month.

In addition, a small number of servers in all groups become unsynchronised just after the leap second. The more detailed information available from both `ntpdate` and the default variable list let us say more about what happened to these servers. We see the following behaviours: (1) Many of these didn't know the leap second was scheduled, and we observe their clock is 1s ahead, and is subsequently stepped. (2) A number of these use GPS or PPS reference clocks, which usually have no way to indicate a forthcoming leap second. (3) A small number of servers seem to leap correctly, but then later step the clock, presumably due to an upstream peer that missed the leap second. (4) Some servers using the ACTS reference clock seem to have known about the upcoming leap second, but not implemented it, as timestamps are out by 1s after the leap second. This final behaviour matches that observed in 2005 and 2008.

2) Behaviour at Potential Leap Seconds: As noted above, there are also 12 days when a leap second could have potentially been scheduled, but was not. Fig. 2 and Fig. 3 also include graphs showing the behaviour around a subset. In June 2009, June 2010, December 2010 and June 2012, we see no leap second being advertised. In December 2009 and December 2011 we see small anomalies where, after the potential leap, a server briefly advertises the insertion of a leap second, but stops shortly afterwards.

In December 2015, there is also an anomaly: one Stratum 2 server has been advertising an upcoming leap since late June. This server is running a version of ntpd with the bug that results in flags not always being cleared, which seems a likely cause for this anomaly.

In each of December 2012, June 2013, December 2013, June 2014 and December 2014, close inspection of the graphs shows that there is a small increase in servers advertising an upcoming leap second in the 24 hours before the potential leap. These servers seem to appear in clusters. For example, in December 2012 a group of `.edu` servers and a group of `.ru` servers account for the majority of the aberrant advertisements. Slight variations of these groups reappear: the `.edu` group in June 2013 and June 2014; the `.ru` group in December 2013 and December 2014. Combined with a `.net` cluster in June 2014, these groups seem to explain almost all of the unexpected advertisements of leap seconds.

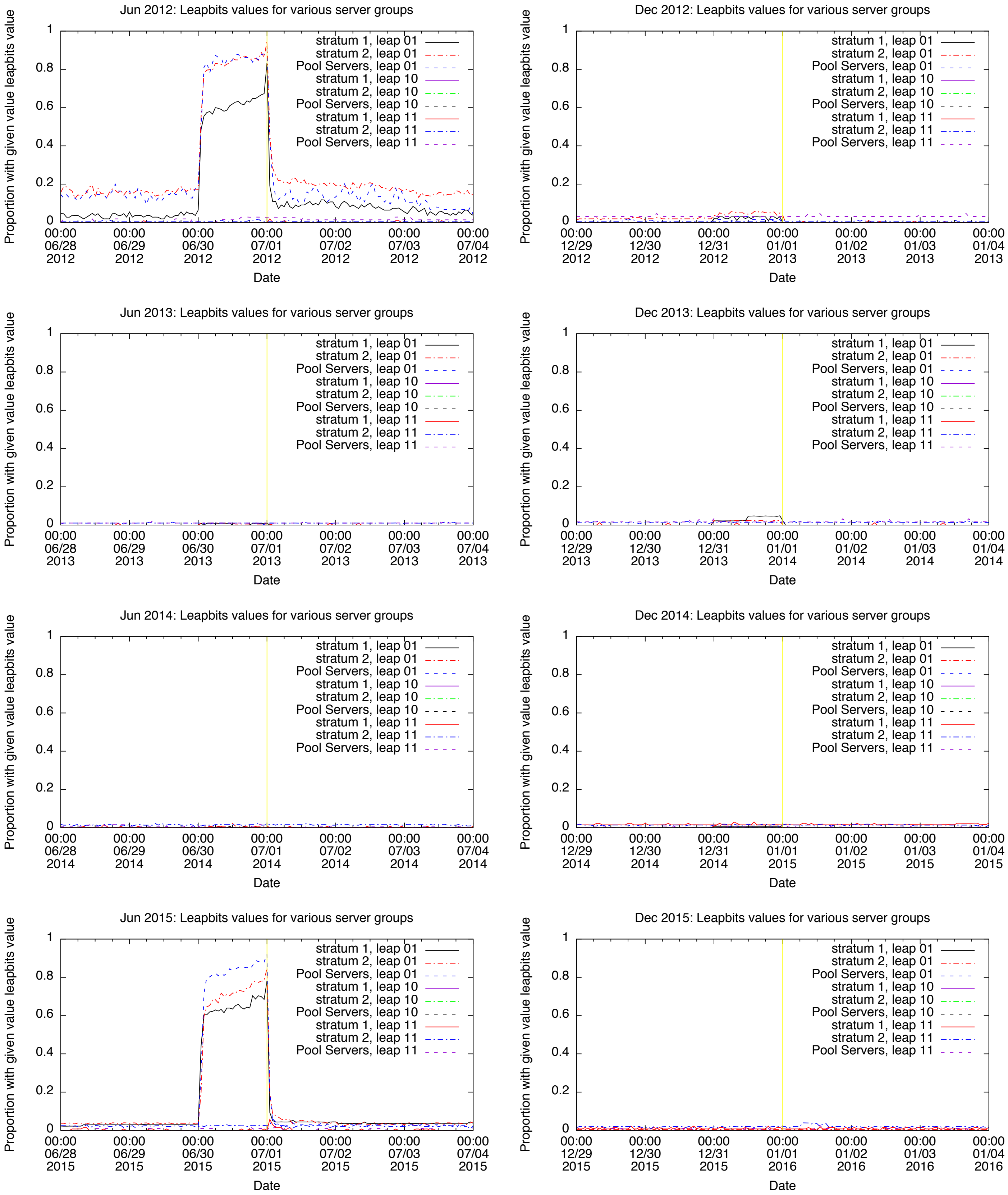

Fig. 3. Hourly behaviour of leap second flags around end of June/December 2012–2015. Graphs are chronologically ordered from top left to bottom right.

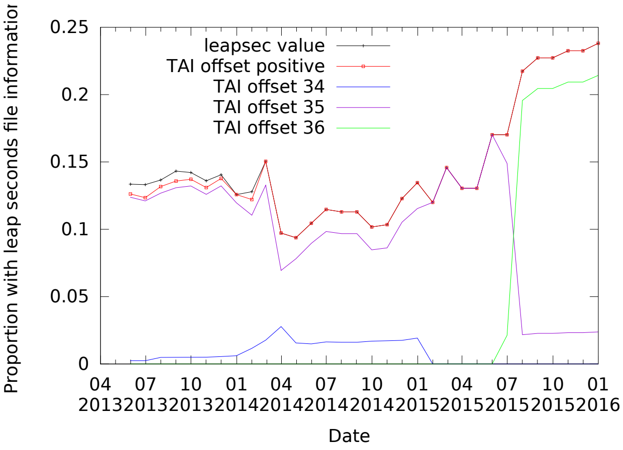

Fig. 4. Fraction of servers responding to ntpq queries reporting variables indicating the presence of a leap second file.

The clustering of these servers suggests that localised propagation of a bogus leap second is possible, but it does not become widespread. In the case of the `.edu` group, it appears that an old version of ntpd running on a stratum 1 server with an unusual reference clock may be a common factor. Overall, we see only localised advertisments of leap seconds at times when none are scheduled.

C. Presence of the Leap Seconds File

It is possible to gain some insight into the increased availability of the leap second file using the servers responding to ntpq queries. The presence of the leap second file results in two ntpd management variables being set. One, `tai`, gives the current difference between the atomic timescale and UTC. The other, `leapsec`, gives a timestamp associated with the generation of the leap second file (if this timestamp is available). For some ntpd versions, `tai` could be reported as 0 if no leap second file was present.

Fig. 4 shows the fraction of servers responding to ntpq queries that provide `leapsec` or `tai` values. We see that the total number of servers reporting a positive `tai` value or a leap second file timestamp are quite close, ranging from around 10% in mid-2013 to almost 25% by 2016. While this suggests that there has been an increase in the number of servers using leap second files, it is possible that the server administrators who configure the leap second file are correlated with those who continue to permit ntpq queries.

We also show a breakdown of the `tai` values reported. This shows that there are a number of NTP servers with an off-by-one error in their calculation of `tai`. We can also see the majority of servers going from advertising a `tai` value of 35 to 36 around the June 2015 leap second, as expected.

If we consider the groups of servers identified above that advertised a bogus leap second, we find that none of them had a leap second file configured when the problem happened. It appears that several operators subsequently configured the file and did not appear in errant groups thereafter.

V. Post-hoc Analysis of Leap Events

Suppose we measure some subset of the leap second graph at the time of a leap second, noting which nodes insert a leap second, and which do not. How would we diagnose which servers advertised the incorrect leap second flags after the fact, if they are not in our observed set of nodes? Such an analysis might be useful to an administrator to determine which external NTP servers provided incorrect information after a missed leap.

Let X_i be 1 if server i is advertising a leap second and be 0 otherwise. We observe $X_i = x_i$ for some observed subset W of the servers. Since the client-server relationships are persistent[7], we can also determine the set S_i of servers used by server i. We then wish to determine the X_i for the unobserved nodes. Remember that modern versions of ntpd use majority vote to determine if a leap second flag will be heeded[8], which amounts to finding X_i so that:

$$X_i = x_i \qquad \forall i \in W,$$

$$X_i \in \{0,1\} \qquad \forall i \notin W,$$

$$\frac{1}{|S_i|} \sum_{j \in S_i} X_j > 0.5 \text{ if } i \in W \text{ and } X_i = 1,$$

$$\frac{1}{|S_i|} \sum_{j \in S_i} X_j \leq 0.5 \text{ if } i \in W \text{ and } X_i = 0,$$

We could regard this as a constraint satisfaction problem, which will typically have multiple solutions, each solution indicating a configuration of leap second flags that could have lead to the observed conditions. By assigning probabilities to the correct/incorrect advertisement of the unobserved servers, we can sum over the solutions to calculate a probability of incorrect advertisement for the unobserved servers. Alternatively, based on our observations that most servers correctly advertise the leap second most of the time, then we can find the solution which maximises/minimises:

$$\sum_{i \notin W} X_i,$$

according to whether we expect there to be a leap second advertised at the time.

To test this method, we considered a real-life example of a cluster of 69 nodes that had each been configured to use 4 pool servers. Thus, each node had effectively configured 4 servers from a larger group of external NTP servers. In total, 53 distinct external servers were in use. In the cluster, 60 nodes correctly inserted the 2015 leap second and 9 did not. After the event, the external servers being used by each node in the cluster were identified using the `ntpq -p` command.

Clearly, evaluating all 2^{53} possibilities of external servers inserting a leap second or not to find which are consistent with the behaviour of the observed cluster nodes is an impractical debugging technique. We know from Section IV that the fraction of pool servers that correctly advertise the leap is bigger

[7] In many circumstances, they can be recovered from `ntpq -p` and/or from `ntp.conf` after the fact.

[8] We could replace some constraints if we knew older versions were running on some servers.

than 0.5, so we construct a binary optimisation programme to minimise the sum of the X_i. Using the SCIP Optimization Suite, a solution was found in 0.1s on an ordinary desktop PC, identifying a minimal set of 5 possible servers.

VI. Discussion

In Section IV we saw that while the leap second behaviour of the NTP network is not ideal, the majority of NTP servers discover that there is a leap second pending before the leap second is due. We observed servers that missed the leap becoming unsynchronised when the extra second was inserted, and subsequently stepping their clocks. We expect similar behaviour from any NTP client that missed the leap second. The alternative misbehaviour of inserting a bogus leap second appears rare and localised to particular subsets of servers. This suggests that NTP's majority-vote mechanism for propagating leap second information should be effective in providing the correct information when a diverse subset of upstream servers is used.

The level of good behaviour seems to be increasing, possibly driven by features in ntpd improving the propagation of leap seconds in the NTP network and allowing administrators to manually configure a file describing past and future leap seconds. Indeed, it may be worth including a leap second file as a part the standard ntpd configuration[9]. Keeping this file up to date represents a minor logistic challenge, however these files are issued with an expiry date, and so are not harmful when they become stale (in contrast to, say, routing bogon files).

While we have focused on the correct implementation of leap seconds in the NTP network, even when NTP has correct knowledge of leap seconds there is scope for issues. On POSIX systems, the leap second is commonly implemented by replaying the last second of the relevant minute. Thus the clock appears to step backwards, which may cause problems for some systems. For example, after the 2005 and 2008 leap seconds, Google reported that the leap second had caused some internal issues and that they would internally *smear* in the leap second using a windowed cosine[10]. In practice, it appears that Google used a linear smear to interpolate the leap second[11] in 2015.

Another issue that arose from the the insertion of the leap second was the Linux kernel printing a message to say that the leap second had been handled, however in some kernel versions this was at a point in the kernel where printing might lead to a deadlock. This led to a number of Linux machines hanging during the 2008, 2012 and 2015 leap seconds[12]. An issue with a later version of Linux futexes resulted in some applications busy-waiting rather than sleeping[13].

VII. Conclusion

In this paper we summarise the leap second behaviour of a group of servers providing a public NTP service from 2005 to 2015. The behaviour in each of the four leap seconds during this period is different and seems to show progressive improvement. We also consider the advertisement of bogus leap seconds and see that it is possible, but rare, and originates from small groups of servers. We also look at the deployment of the leap second file, and see it has grown over the period. Finally, we propose a technique for post-hoc diagnosis of incorrect leap seconds.

Acknowledgements

Thanks to Simon Leinen for providing the sample data for the post-hoc analysis. This publication emanated from research supported in part by a grant from Science Foundation Ireland (SFI) and co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

References

[1] D. Mills, *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space*. CRC Press, 2011.

[2] D. D. McCarthy, C. Hackman, and R. A. Nelson, "The physical basis of the leap second," *The Astronomical Journal*, vol. 136, no. 5, pp. 1906–1908, 2008.

[3] D. L. Mills, "On the accuracy and stablility of clocks synchronized by the network time protocol in the internet system," *Computer Communication Review*, vol. 20, no. 1, pp. 65–75, 1989.

[4] 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.

[5] D. L. Mills, A. Thyagarjan, and B. C. Huffman, "Internet timekeeping around the globe," DTIC Document, Tech. Rep., 1997.

[6] N. Minar, "A survey of the NTP network," 1999.

[7] C. D. Murta, P. R. Torres Jr, and P. Mohapatra, "Characterizing quality of time and topology in a time synchronization network." in *Proc. GLOBECOM*. IEEE, 2006.

[8] F. Buchholz and B. Tjaden, "A brief study of time," *Digital Investigation*, vol. 4, pp. 31–42, 2007.

[9] M. Laner, S. Caban, P. Svoboda, and M. Rupp, "Time synchronization performance of desktop computers," in *Proc. International Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. IEEE, 2011, pp. 75–80.

[10] C.-Y. Hong, C.-C. Lin, and M. Caesar, "Clockscalpel: Understanding root causes of Internet clock synchronization inaccuracy," in *Proc. Passive and Active Measurement Conference (PAM)*. Springer, 2011, pp. 204–213.

[11] P. V. Estrela and L. Bonebakker, "Challenges deploying PTPv2 in a global financial company," in *Proc. International Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. IEEE, 2012, pp. 1–6.

[12] H. Stenn, "Securing the network time protocol," *ACM Queue*, vol. 13, no. 1, 2015.

[13] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks," in *Proc. Internet Measurement Conference (IMC)*. ACM, 2014, pp. 435–448.

[14] A. Malhotra, I. E. Cohen, E. Brakke, and S. Goldberg, "Attacking the network time protocol," in *Proc. Network and Distributed System Security Symposium*. Internet Society, 2016, p. To Appear.

[15] S. J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *Proc. ACM conference on Computer and Communications Security*. ACM, 2006, pp. 27–36.

[16] J. Olsson and M. Boldt, "Computer forensic timeline visualization tool," *Digital Investigation*, vol. 6, pp. S78–S87, 2009.

[17] S. Y. Willassen, "Timestamp evidence correlation by model based clock hypothesis testing," in *Proc. International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia*. ICST, 2008, pp. A15, 1–6.

[18] D. Malone, "Unwanted HTTP: Who has the time?" *;login: The Magazine of USENIX & SAGE*, vol. 31, no. 2, pp. 49–54, 2006.

[19] D. Plonka, "Flawed routers flood University of Wisconsin Internet time server," 2003.

[9] This is being trialed in recent versions of FreeBSD.

[10] https://googleblog.blogspot.ie/2011/09/time-technology-and-leaping-seconds.html

[11] https://mlichvar.fedorapeople.org/leap2015/google_smear.png

[12] https://lkml.org/lkml/2009/1/2/373

[13] https://access.redhat.com/articles/15145