Technical Aspects of Leap Second Propagation and Evaluation

by **Martin Burnicki**
Email: martin.burnicki@meinberg.de

Meinberg Funkuhren GmbH & Co. KG
Bad Pyrmont, Germany
http://www.meinberg.de

Meinberg Sales Meeting
June 2015

Introduction

Before 1972 the length of a second was determined by Earth rotation

- When Earth rotation changed, seconds became longer or shorter
- Not good for technical solutions

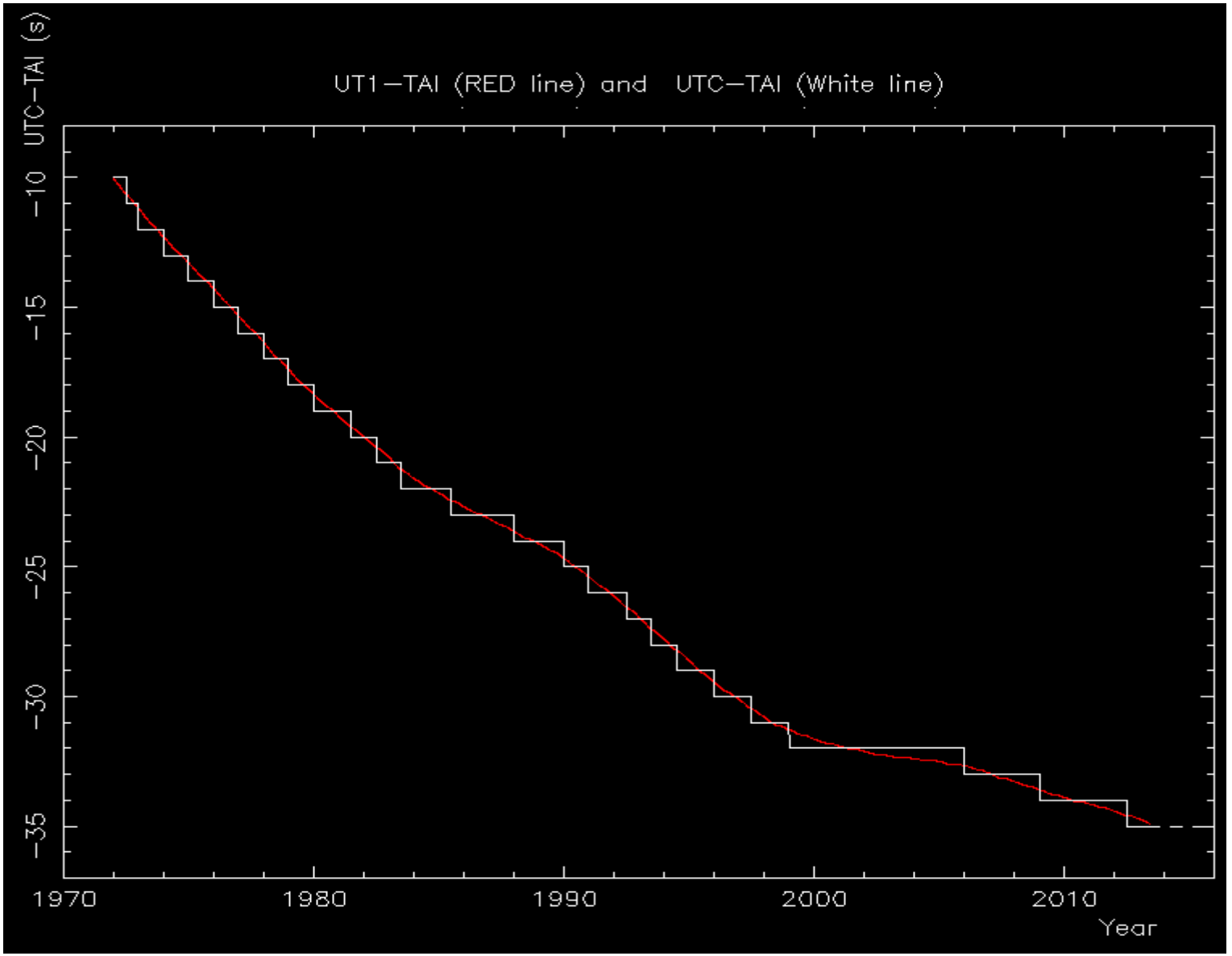
Since 1972 the length of a second is determined by atomic time, derived from Cesium atoms, base of UTC time

- Seconds have always the same length
- But Earth rotation still varies, and is a little bit slow

Leap seconds are scheduled by the International Earth Rotation Service (IERS) to keep UTC time synchronized with Earth rotation

- IERS Bulletin C sent by email twice per year
- Basically leap seconds can be inserted or deleted
- In the past there haven't been leap second deletions

Leap Seconds Inserted Since 1972



Source: http://hpiers.obspm.fr/eop-pc/earthor/utc/leapsecond.html

Leap Second Intervals

- From 1972 up to the 1990es leap seconds have been inserted once per year or at latest after 1.5 years
- From Dec 1998 to Dec 2005 there were 7 years without a leap second (!)
- Since then leap seconds about every 3 to 4 years

What To Do With Leap Seconds

Important: Distinguish between ***propagation*** of the leap second announcement ***to all devices***, and leap second ***handling by devices***

Different ways to *propagate* leap second announcements

- Radio signals, time codes, serial time strings
- Network protocols, data files

Different ways to *handle* leap second

- Step time at beginning or end of leap second
- Slew time over a certain interval around a leap second

All devices need to receive a leap second announcement *early enough*

- Propagated through a chain of devices and protocols

Leap Second Handling (1)

UTC specifies how time is counted when enumerating seconds:

- Skip last second “59” to delete a leap second
- Extend second count to “60” to insert leap second

Enumeration of UTC seconds when inserting a leap second:

2015-06-30 23:59:57
2015-06-30 23:59:58
2015-06-30 23:59:59
2015-06-30 23:59:60 <-- inserted leap second
2015-07-01 00:00:00
2015-07-01 00:00:01
2015-07-01 00:00:02

→ Normalized times before and after leap second are the same

→ Same system time stamps at beginning and end of leap second (POSIX)

→ How to handle fractions during the second?

Leap Second Handling (2)

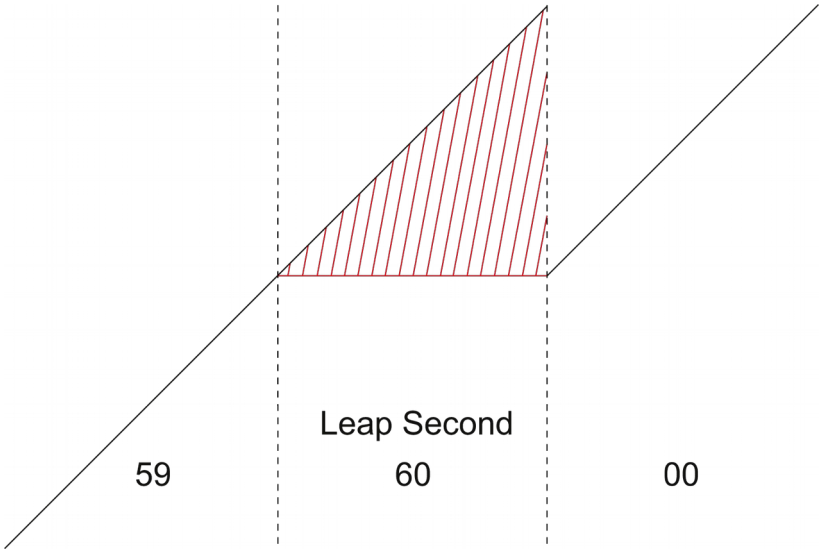
System time doesn't observe a second “60”, but only seconds since an epoch

Timestamp	Date/Time (UTC)	Offset from linear time [ms]	
D93DABFD.2A54001E	2015-06-30 23:59:57.165	-0.602	
D93DABFE.2AB85E4F	2015-06-30 23:59:58.166	-0.621	
D93DABFF.2B1FD406	2015-06-30 23:59:59.168	-0.646	<-- „59“, before leap second
D93DABFF.2B8DEDA5	2015-06-30 23:59:59.170	-1000.654	<-- „60“, leap second
D93DAC00.2BF7A2CE	2015-07-01 00:00:00.171	-1000.665	
D93DAC01.2C610C6B	2015-07-01 00:00:01.173	-1000.692	
D93DAC02.2CC3E439	2015-07-01 00:00:02.174	-1000.718	
D93DAC03.2D281C20	2015-07-01 00:00:03.176	-1000.762	

- Without additional status unable to tell if second is leap second, or not
- Conversion to date/time yields same result with second “59” twice

Leap Second Handling (3)

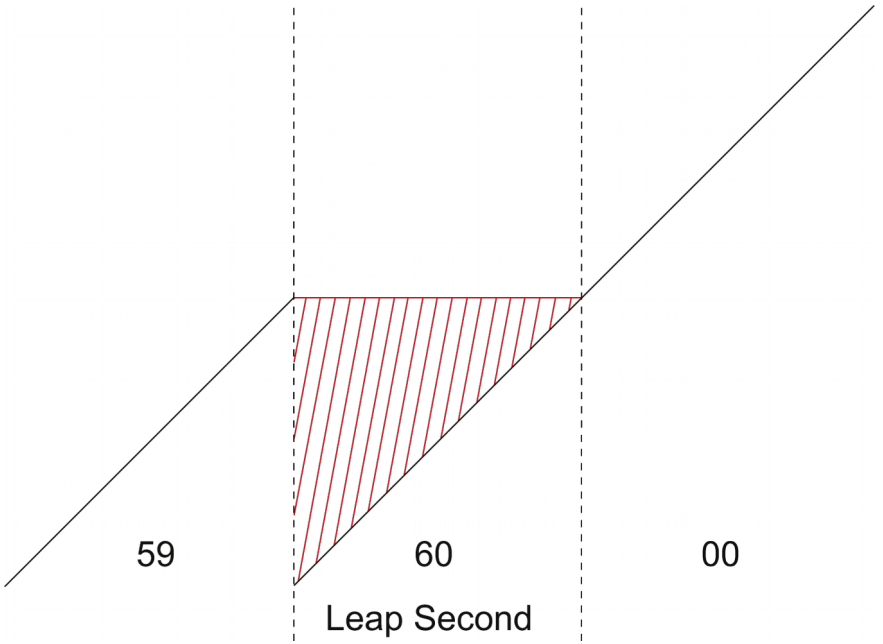
Stepping time back at the *end* of the leap second



- Time doesn't increase monotonically
- Duplicate time stamps *after* the leap second
- *Earlier* time stamps for *later* events
- Ambiguous beginning of new minute / hour / UTC day

Leap Second Handling (4)

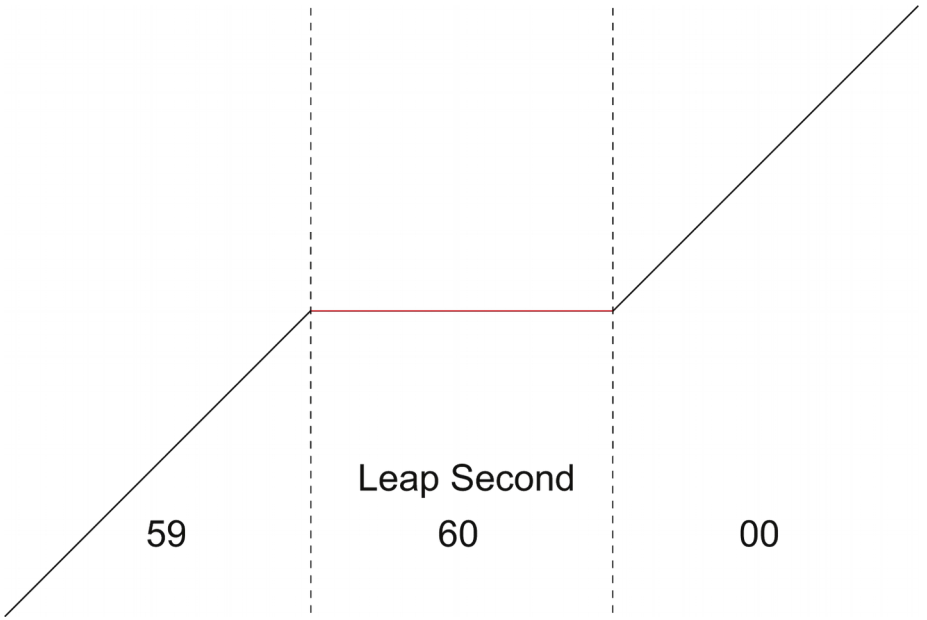
Stepping time back at the *beginning* of the leap second



- Time doesn't increase monotonically
- Duplicate time stamps *during* the leap second
- *Earlier* time stamps for *later* events
- Distinct beginning of new minute / hour / UTC day

Leap Second Handling (5)

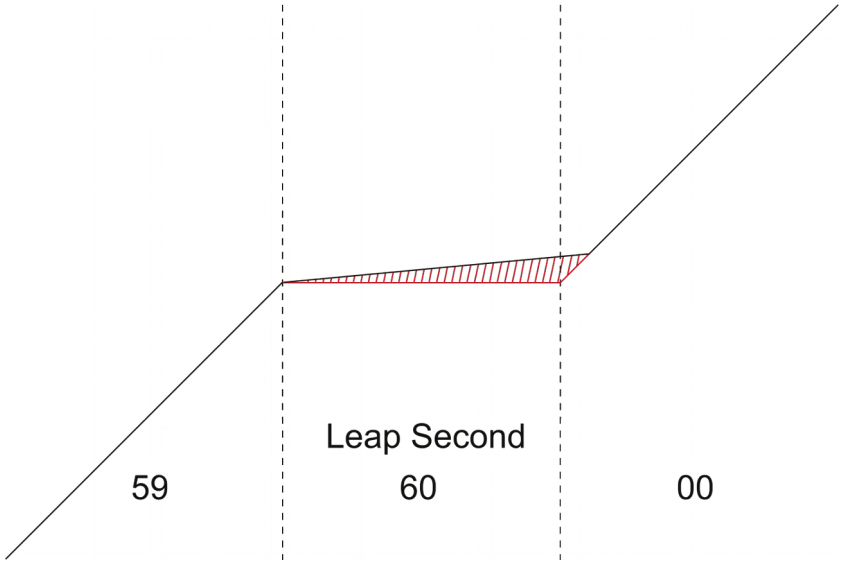
Stopping the clock for 1 second



- Time doesn't increase *strictly* monotonically
- *Same time stamps* for *subsequent events* during the leap second
- Distinct beginning of new minute / hour / UTC day

Leap Second Handling (6)

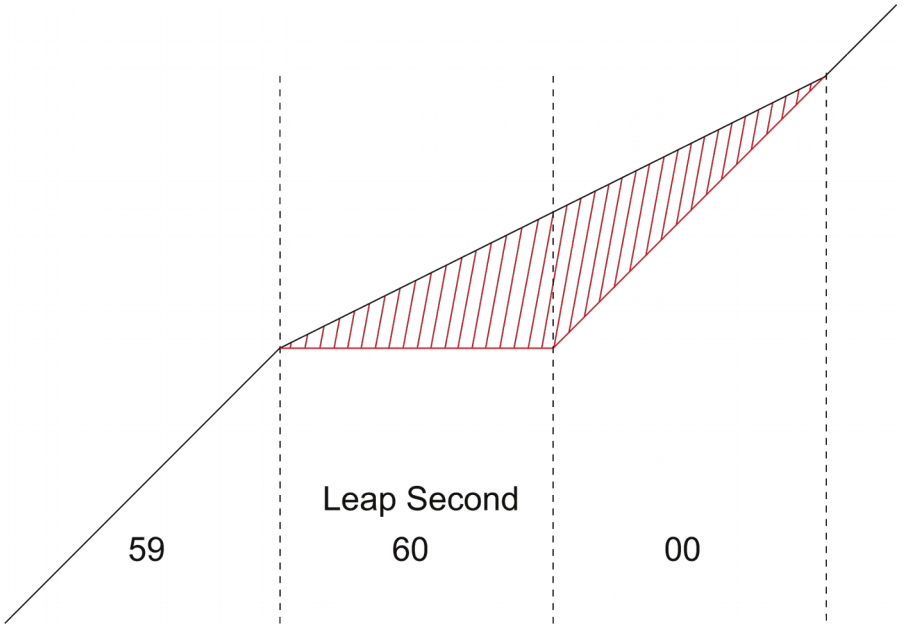
Increment time LSB on request



- LSB of time stamp is incremented whenever time is read
- Proposed by David L. Mills who invented NTP
- Time increases strictly monotonically
- No duplicate time stamps caused by leap second
- *Nearly* distinct beginning of new minute / hour / UTC day

Leap Second Handling (7)

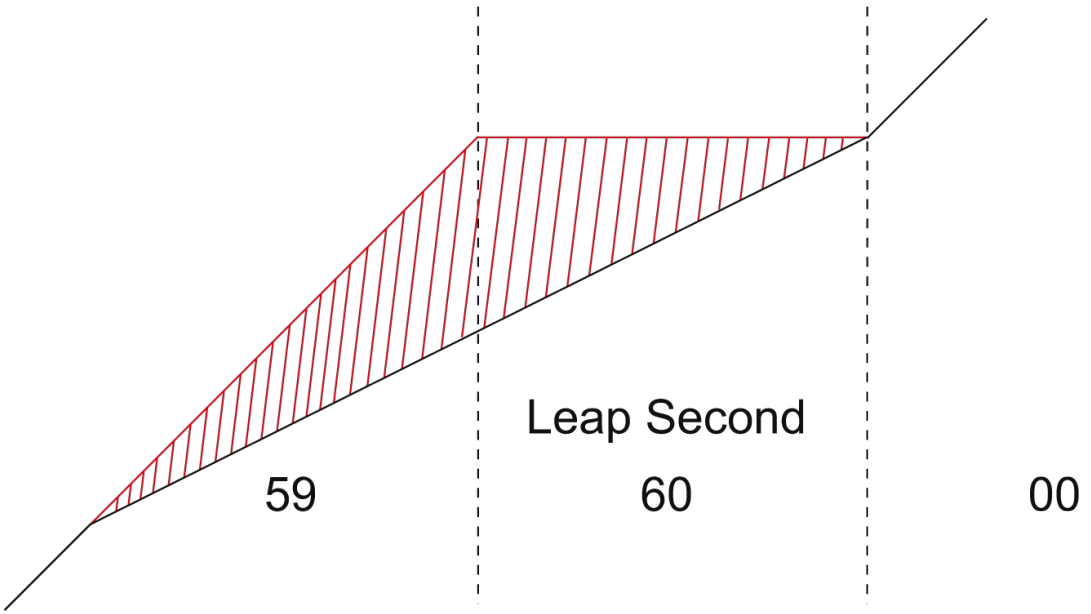
Slewing time half speed over 2 seconds *after beginning* of leap second



- Can be used if system clock does not support leap seconds
- Has been implemented in NTP port for Windows, before 4.2.8
- Initial workaround to support leap seconds under Windows
- No duplicate time stamps around leap second
- Incorrect beginning of new minute / hour / UTC day

Leap Second Handling (8)

Slewing time half speed over 2 seconds *before end* of leap second



- Can be used if system clock does not support leap seconds
- Has been implemented in NTP 4.2.8 for Windows
- Better workaround for Windows
- No duplicate time stamps around leap second
- Start of new minute / hour / UTC day accurate

Leap Second Handling (9)

UTC-SLS Proposal

- UTC with Smeared Leap Seconds
- Proposed by Markus G. Kuhn, Cambridge University
- System time gradually incremented during last 1000 s of day
- To be implemented in OS kernel
- Needs to receive leap second warning early enough so nodes can start smearing
- System time differs from UTC during smear interval
- Start of new minute / hour / UTC day accurate
- Used by NASA “Common Data Format” (CDF) software
- Andrew Main (Zefram)'s Time::UTC_SLS package for perl
- Oracle's Java: java.time.Clock class

Leap Second Handling (10)

Google Leap Smear

- Implemented by Google due to leap second problems observed in 2005
- Modified NTP servers send “modulated” time to clients
- Corrections applied smoothly, very slowly (cosine function)
- Clients follow modulated time from NTP servers
- Modification only required on NTP servers, not on clients
- Thus easy to implement for a huge number of clients
- Can only be used with “private” NTP servers
- System time differs from UTC during smear interval
- Start of new minute / hour / UTC day accurate
- In 2015 modified approach to smear time 10 hours before and 10 hours after leap second, smeared time always within ±0.5 s to UTC

Handling of leap second warnings (1)

GPS Satellite System

- GPS time is linear with constant offset to TAI, differs from UTC
- Counts weeks modulo 1024, and second-of-week
- Leap second warning broadcast nearly 6 months in advance
- Data structure contains week and day number for leap second as well as UTC offset before and after the leap time. Flexible since it
  - specifies an exact point in time for the leap second event
  - can handle both positive and negative leap seconds
  - can even specify the handling of several leap seconds at the same time
- Potential problem due to truncated week number in UTC parameters which allows only +/- 127 weeks
- How does a GPS receiver propagate a leap second warning to a devices to which it is connected?

Handling of leap second warnings (2)

German Long Wave Transmitter DCF-77

- Disseminates legal local time for Germany
- Only an announcement flag for a **positive** leap second
- No announcement of negative leap second possible
- Announcement flag only set 59 minutes in advance
- Too late for applications acting as NTP server
- Announcement missed if no reception during that hour

NIST Time Services

- Telephone service (ACTS) and LF transmitters (WWVH/WWVB)
- Leap second warning is transmitted
- Sign of DUT1 parameter can be used to distinguish insertion from deletion
- Announcement set manually by operators nearly 1 month in advance

Handling of leap second warnings (3)

IRIG Time Codes

- First codes introduced around 1960, long before leap seconds were known
- Commonly used code formats don't include leap second warning
- Even if transported seconds count to 60 this is usually too late

IEEE Time Codes

- IEEE 1344 from 1995 specifies how to use IRIG B122 reserved bits to transport additional information, including year number and local time offset, as well as DST and leap second status
- IEEE C37.118 from 2005 is same as IEEE1344, just uses reversed sign for local time offset
- Leap second flags not set earlier than 59 seconds in advance, which is much too short e.g. for usage with NTP servers, but may be early enough for other devices

Example of an IRIG Time Code Signal

Unmodulated IRIG signal (DC Level shift, DCLS)

Modulated IRIG signal, 1 kHz carrier frequency

Handling of leap second warnings (4)

Serial Time Strings

- Often used to transfer time and status information from a radio clock or GPS receiver to time synchronization software like the NTP daemon
- Very few string formats support leap second announcement
- Many popular formats including NMEA don't support this
- With some GPS receivers it is available via a proprietary binary format
- String formats commonly used by Meinberg devices include a leap second announcement
- No way for a GPS receiver to tell that a leap second is pending if the selected string format doesn't support this

Handling of leap second warnings (5)

NIST Leap Second File

- A text file containing a list of historic and current leap seconds
- Can be downloaded from FTP or HTTP servers
- Provides an expiration date
- Can be used with NTP daemon
- Beside USNO and NIST also the IERS now provides such file

Olson TZ Data Base

- Widely used on systems to convert UTC to local time
- Also includes a leap second file
- Provides historic time zone and leap second data
- Leap second data only used if a “right” time zone is used
- “Right” time zones currently not used on standard systems
- NTP software *chrony* uses TZ leap second file for leap second handling

Handling of leap second warnings (6)

PTP/IEEE1588 Network Protocol

- Uses TAI time stamps in the protocol by default
- UTC offset is transported by the protocol
- Two independent flags for positive and negative leap second
- Using 2 flags is prone to errors in implementation
- Leap second announcement 12 hours before UTC midnight
- Open source implementation of ptpd originally didn't evaluate leap second announcements
- Detected and implemented just a few weeks before the leap second event at the end of June, 2012
- Runs on Linux and BSD, so announcement is just passed to the kernel

Handling of leap second warnings (7)

Network Time Protocol (NTP)

- Uses TAI time stamps in the protocol by default
- The protocol supports positive or negative announcement
- The Reference implementation can receive announcement from
  - Upstream NTP servers
  - Hardware refclocks (e.g. GPS receivers)
  - NIST leap second file
- The exact way of evaluation varies with software version
- Priority of the sources
- *Usually* the leap second warning is just passed down to the kernel, if the kernel supports this (Linux, *BSD, Solaris)
- Leap second warning ***not passed down*** to the kernel if ntpd run with -x parameter --> behaviour depends on ntpd version

Handling of leap second warnings (8)

MEINBERG

NTP version dependencies

- Until v4.2.4 leap second file only evaluated if autokey enabled and configured properly
- Since v4.2.4 leap second handling code for Windows included in ntpd
- Until v4.2.4 leap second announcement from single upstream server accepted
- Since v4.2.6 announcement only accepted from majority of upstream servers
- Only since v4.2.6 leap second handling by ntpd on systems without kernel discipline (non-Windows)
- Since v4.2.6 leap second file is authoritative, if used; other sources ignored
- In v4.2.6 **leap second file even authoritative if expired** --> leap seconds not handled even if other sources provide a valid announcement
- Since v4.2.8 leap second file is ignored if expired
- Until v4.2.8 problem with leap second workaround on Windows 8 / Server 2012 due to changes in the Windows kernel
- Since v4.2.8p1 modifed leap second code also working on Windows 8 / Server 2012
- Since v4.2.6 **system time stepped even if ntpd runs with -x** (NTP Bug 2745), fixed in 4.2.8p3

Handling of leap second warnings (9)

Unix Kernels

- Usually can handle a leap second by itself if the kernel receives a leap second warning in time
- A programming interface is available which is used e.g. by ntpd and ptpd to pass a leap second announcement to the kernel
- Same programming interface can be used to retrieve the leap second status from the kernel
- Current Linux kernels just step time back at beginning of leap second

Windows

- Does not support leap seconds natively
- A workaround is available in the NTP port for Windows where the Windows system time is slewed over 2 seconds

Known Bugs (1)

Linux Kernel Deadlock

- During leap second processing on a busy machine the kernel could try to acquire a lock which it already had, causing a deadlock
- Machine stopped working, services immediatedly and unexpectedly became unavailable
- Affected kernels about 2.6.22 through 2.6.26.6

High CPU Load caused on Linux Systems

- Occurred after June 30, 2012
- CPU load continuously at 100%
- Significantly increased power consumption and heat in data centers
- Increased heat required increased air conditioning, and thus even more power consumption
- Affected kernels up to about 2.6.32

Known Bugs (2)

Absolute timers that expire at midnight UTC may fire early when the leap second is inserted

- The timer expiry should occur at the end of the leap second event but it does not always occur at the end. In fact it may occur during the leap second event and effectively one second too early
- Reported just on May 27, 2015
- Fix in progress

Known Bugs (3)

NTP Reference Implementation

- Up to v4.2.4 leap second was only evaluated if kernel discipline was available
- In 4.2.4 a workaround for Windows was introduced
- In 4.2.6 this was fixed for other systems without kernel discipline, time stepped back 1 s by ntpd
- Up to 4.2.6 there has been a bug potentially causing a leap second loop between groups of servers, causing ntpd to insert a leap second at the end of every month after a real leap second until the affected daemons are restarted
- Support for negative leap seconds has been removed in v4.2.6
- Leap second code has been reworked in v4.2.8
- Fix for -x (slew always vs. step leap second) eventuelly in 4.2.8p3

Open Source PTP Daemon

- Leap second support added in v2.2.2
- No automatic update in grandmaster mode; ptpd restart required
- Will eventuelly be fixed in v2.3.1

Known Bugs (4)

Invalid Leap Second Warning Issued By 3rd Party GPS Receivers

- Occurred after July 2005 when GPS started to announce a leap second for **December 2005**
- Faulty announcement was generated for the end of **September 2005**
- As a consequence NTP servers started to announce a leap second
- As another consequence ntpd was modified to accept announcements only from a majority of configured servers
- Can still affect NTP servers since refclocks are considered authoritative, if no current leap second file is being used
- As reported on the NTP pool mailing list this happened again in January 2015 when GPS satellites started to broadcast the leap second information for June 2015
- Errors are due to improper coding of the GPS receiver firmware, not due to NTP

Distinguish Types of Problems

Leap second not handled at all

- Time is off by 1 s after leap second event
- May not match legal requirements

Leap second handling causing system failure

- Fatal problem, affecting all applications on a system, e.g. Linux kernel deadlock

Leap second handling causing application problems

- e.g . duplicate timestamps due to stepped-back system time

Current Risks

Times of leap second insertion

- Dec 31, 1998 Thu/Fri New Years Day
- Dec 31, 2005 Sat/Sun New Years Day, weekend
- Dec 31, 2008 Wed/Thu New Years Day
- June 30, 2012 Sat/Sun weekend
- June 30, 2015 Tue/Wed **normal working day**

Leap Second 2015

- UTC: Wed 00:00
- Germany: Wed 02:00
- Asia / Japan: Wed 08:00
- USA / California: Tue 16:00 / 04:00 PM

→ **Lower risk in Europe / Africa, at night**

→ **Higher risk in Asia / America, during working hours**

→ **Problems potentially caused by local approaches to handle leap second in a non-standard way**

Avoiding Errors in System Designs

- Importance that leap second announcement is propagated early enough
- Several links with eventually different data connections
- Every link must fulfill worst case requirements, e.g. for cascaded NTP clients with 1024 s polling intervals
- Select appropriate data transfer methods when transferring data from a GPS receiver
- Time code signals and some serial time string formats may not be sufficient
- Usage of a leap second file / table may help
- Leap second file / table needs to be updated regularly, even if no leap second has been scheduled
- Without expiration date unable to determine if no leap second has been scheduled, or just no information is available.

Avoiding Errors in Applications

- **Careful software design and implementation is required**

- **Comparing time stamps from distributed services:**
  - Different nodes might increment the time over a leap second in different ways
  - Can result in abnormal large time offsets

- **Time stamps:**
  - Time may be stepped back
  - Duplicate time stamps may occur

- **Time intervals:**
  - A day may have 86400 ± 1 seconds
  - Even when working with historical data

Possible Future Improvements (1)

Try to fix ambiguity of time stamps

- When time is stepped back because a leap second is inserted
- Problem is similar to end of DST
- Leap second status flag would be helpful to distinguish

- **GSoC 2013 project by the Network Time Foundation (NTF)**
  - Investigated on a new general time stamp format and associated API calls
  - May not be appropriate for applications reading time at a high rate, with high resolution
  - Additional computation in kernel space increasing execution time
  - Locking mechanisms required to guarantee consistent time stamp and status information

Possible Future Improvements (2)

Ongoing and discussed improvements:

- New ***tzdist*** protocol allows update of the TZ rules and leap second information via network (HTTP/JSON)
- Proposal by Poul-Henning Kamp to use DNS to distribute leap second information (DNSSEC, pseudo DNS records)
- Long-lasting discussion to abolish leap seconds

  Decision taken eventually later in 2015

References (1)

- Mills, D.L., "A kernel model for precision timekeeping", Network Working Group Report RFC1589, University of Delaware, March 1994

- Kuhn, Markus G., "UTC with Smoothed Leap Seconds (UTC-SLS)", 2005-12-14, http://www.cl.cam.ac.uk/~mgk25/time/utc-sls/

- Google Blog, "Time, technology and leaping seconds", 2011-09-15, http://googleblog.blogspot.de/2011/09/time-technology-and-leaping-seconds.html

- Google Cloud Platform Blog, “Got a second? A leap second that is. Be ready for June 30th!“ http://googlecloudplatform.blogspot.de/2015/05/Got-a-second-A-leap-second-that-is-Be-ready-for-June-30th.html

- Linux Kernel Mailing List, "LKML Bug: Status/Summary of slashdot leap-second crash on new years 2008-2009", 2009-01-02, https://lkml.org/lkml/2009/1/2/373

- RedHat Bug 479765, "Leap second message can hang the kernel", 2009-01-12, https://bugzilla.redhat.com/show_bug.cgi?id=479765

- Linux Kernel Mailing List, "Potential fix for leapsecond caused futex related load spikes", 2012-07-01, https://lkml.org/lkml/2012/7/1/27

References (2)

- RedHat Bug 836803, "RHEL6: Potential fix for leapsecond caused futex related load spikes", 2012-07-01, https://bugzilla.redhat.com/show_bug.cgi?id=836803
- "Leap second bug in Linux wastes electricity", 2012-07-03, http://www.h-online.com/open/news/item/Leap-second-bug-in-Linux-wastes-electricity-1631462.html
- Linux Kernel Mailing List, “Leap Second, nohz, and ABSOLUTE timers problem”, https://lkml.org/lkml/2015/5/27/458
- Linux Kernel Mailing List, “timer changes for 4.2”, https://lkml.org/lkml/2015/6/22/110
- NTP bug 508: "ntpd doesn't handle leap seconds gracefully on systems without kernel support for leap seconds", https://bugs.ntp.org/show_bug.cgi?id=508
- NTP Bug 2246: "sys_leap is sticky", https://bugs.ntp.org/show_bug.cgi?id=2246
- NTP Bug 2745: "ntpd -x steps clock on leap second", https://bugs.ntp.org/show_bug.cgi?id=2745

References (3)

- Steven L. Allen, "Timekeeping System Implementations: Options For The Pontifex Maximus", October 2011 meeting "Decoupling Civil Timekeeping from Earth Rotation"

- Google Summer of Code 2013, "New Timestamp Format and API", http://www.google-melange.com/gsoc/project/details/google/gsoc2013/limifly/5727390428823552

- Network Time Foundation, "New Timestamp and Library API", Google Summer of Code 2013, https://wiki.nwtime.org/GSoC/GSoC2013NewTimestampAndAPI

- NASA, What is Common Data Format (CDF)? http://cdf.gsfc.nasa.gov/

- NASA, Requirements for handling leap seconds in CDF http://cdf.gsfc.nasa.gov/html/leapseconds_requirements.html

- Andrew Main (Zefram)'s Time::UTC_SLS package for perl http://search.cpan.org/~zefram/Time-UTC_SLS-0.003/lib/Time/UTC_SLS.pm

- Oracle's Java: java.time.Clock class https://docs.oracle.com/javase/8/docs/api/java/time/Clock.html

Thanks

Thanks for your attention!