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

Requirements for UTC and Civil Timekeeping on Earth
A Colloquium Addressing a Continuous Time Standard
May 2013

Introduction

Leap seconds are scheduled by the International Earth Rotation Service (IERS)

- To keep UTC time synchronized with earth rotation
- Announcement must be propagated to time keeping 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

Leap second announcement must be early enough

- Propagated through a chain of devices and protocols

Leap Second Handling (1)

Counting time in human readable format is easy

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

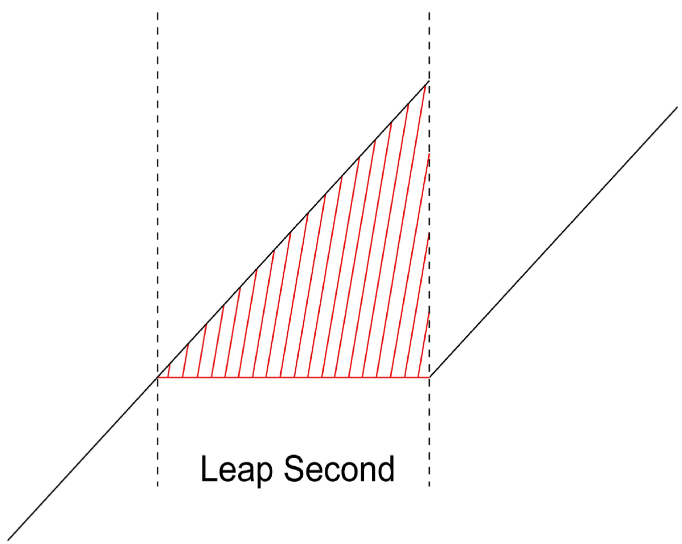
Enumeration of seconds when inserting a leap second:

2012-06-30 23:59:57
2012-06-30 23:59:58
2012-06-30 23:59:59
2012-06-30 23:59:60 <-- leap second
2012-07-01 00:00:00
2012-07-01 00:00:01
2012-07-01 00:00:02

- Normalized times before and after leap seconds are the same
- How to handle fractions during the second?

Leap Second Handling (2)

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

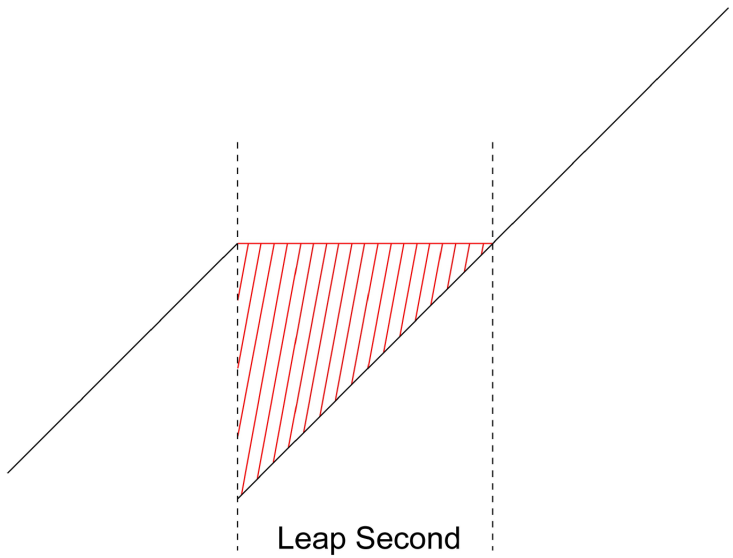


- Time increases not monotonically
- Duplicate time stamps *after* the leap second
- *Earlier* time stamps for *later* events

Leap Second Handling (3)

MEINBERG

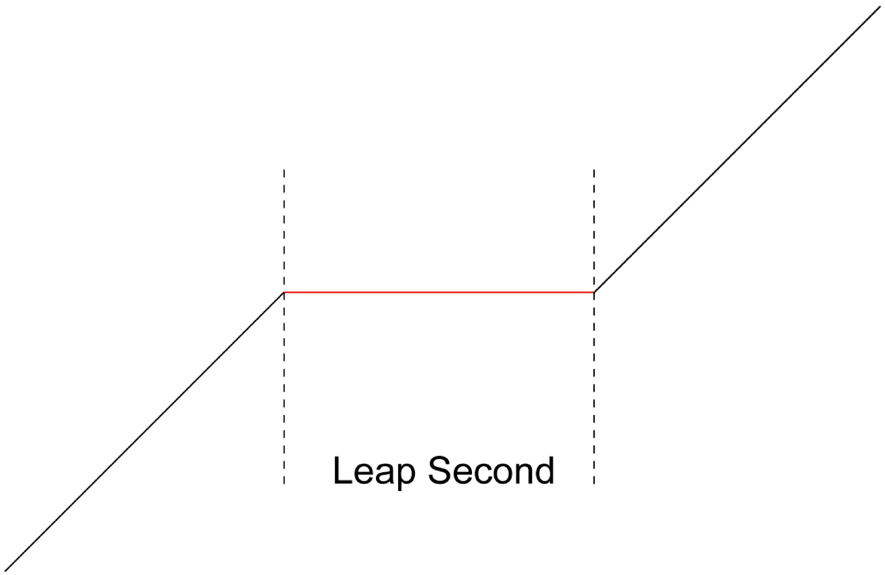
Stepping time back at the *beginning* of the leap second



- Time increases not monotonically
- Duplicate time stamps *during* the leap second
- *Earlier* time stamps for *later* events

Leap Second Handling (4)

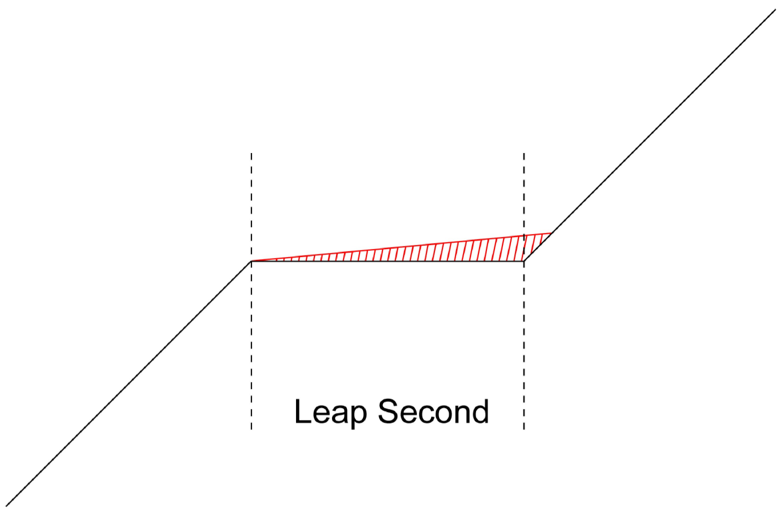
Stopping the clock for 1 second



- Time increases not strictly monotonic
- Same time stamps during the leap second

Leap Second Handling (5)

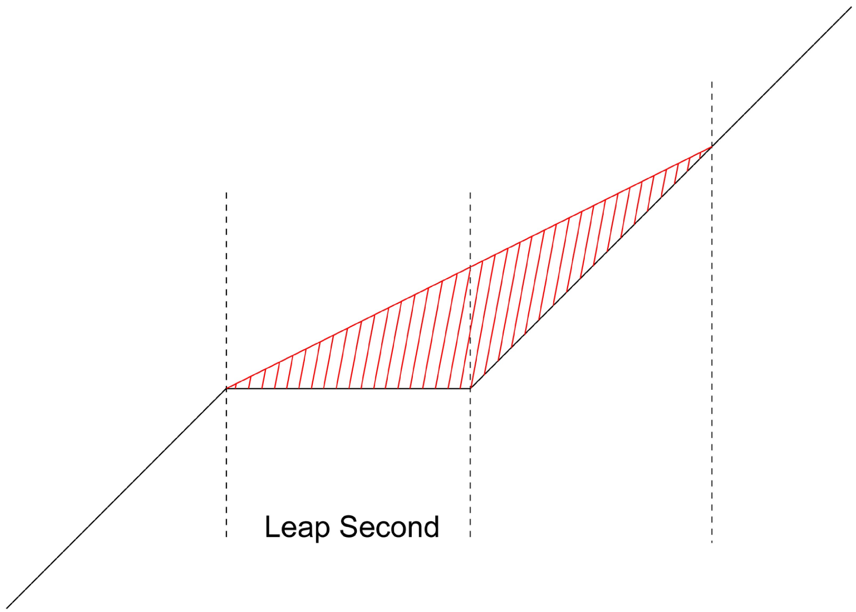
Increment time LSB on request



- LSB of timestamp is incremented whenever time is read
- Proposed by David L. Mills who invented NTP
- Time increases strictly monotonic
- No duplicate time stamps caused by leap second

Leap Second Handling (6)

Slewing time half speed over 2 seconds



- Can be used if system clock does not support leap seconds
- Has been implemented in NTP port for Windows
- Good workaround to reduce error

Leap Second Handling (7)

UTC-SLS Proposal

- UTC with Smeared Leap Seconds
- Proposed by Markus G. Kuhn
- 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
- Actually not implemented, as far as I know

Leap Second Handling (8)

Google Leap Smear

- Implemented by Google due to leap second problems observed in 2008
- Modified NTP servers send “modulated” time to clients
- 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
- System time differs from UTC during smear interval
- Can only be used with “private” NTP servers

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 broadcasted 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
- Very 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 with NTP
- Announcement missed if no reception during that hour

NIST Time Services

- Telephone service (ACTS) and LF transmitters (WWVH/WWVB)
- Leap second warning is transmitted
- How long 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 be set earlier than 59 seconds in advance, which is much too short e.g. for usage with NTP

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 don't support this
- 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 servers
- Provides an expiration date
- Can be used with NTP daemon

Olson TZ Data Base

- Widely use 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” timezone is used
- “Right” timezones currently not used on standard systems, as far as I know

Handling of leap second warnings (6)

PTP/IEEE1588 Network Protocol

- Works by default with TAI
- 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)

- 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
- Announcement from a single upstream server until v4.2.4
- With v4.2.6 and newer only from majority of upstream servers
- On systems with kernel discipline announcement is just passed to the kernel
- Workaround for Windows was introduced in v4.2.4
- Support for other systems without kernel discipline was introduced in v4.2.6

Handling of leap second warnings (8)

Unix Kernels

- Often implement the kernel clock implementation proposed by Dave Mills, or variations of it
- 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
- Status has to be queried by the time sync application

Windows

- Does not support leap seconds
- A workaround is available in the NTP port for Windows

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, which caused a deadlock
- Machine stopped working
- Services became unexpectedly 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
- Affected kernels up to about 2.6.32

Known Bugs (2)

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 added
- In 4.2.6 this was fixed for other systems without kernel discipline
- Actually there's 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

Open Source PTP Daemon

- Leap second support added in v2.2.2

Known Bugs (3)

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
- 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

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 may help

Careful software design and implementation is required anyway

Example: time synchronization between remote nodes:

- Different nodes might increment the time over a leap second in different ways
- Can result in abnormal large time offsets

Possibilities for Future Improvements (1)

Try to fix ambiguity of timestamps

- 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 NTF investigates 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 requires time
  - ✓ Locking mechanisms may be required to guarantee consistent time stamp and status information

Possibilities for Future Improvements (2)

A different approach proposed by Steve Allen: run computer system clock on GPS time and use the “right” zone tables

- Can be used to convert linear GPS time unambiguously to a time scale observing leap seconds
- Requires modification of the leap second file currently shipped with the tz database

Possible improvements:

- Avoid “proprietary” GPS time scale and use TAI instead
  Advantage:
  - ✓ Existing tz leap second file can be used unchanged
  - ✓ Easier interaction with PTP which uses TAI anyway

Wishlist:

- An API call to ask the kernel if it runs UTC or TAI, so that programs like ntpd or ptpd can find this out at runtime

Possibilities for Future Improvements (3)

Limitations using leap file and “right” time zone:

- Embedded systems often don't get firmware updates, and thus the tz data base might not be updated
- Still a requirement for safe online update via a protocol
- Ntpd does this with NIST leap second file and autokey
- Ptpd does this natively by transporting UTC offset

Wishlist:

- Add some expiration info to the tz leap second file
- Eventually use same formats for NIST and tz files

Conclusion

Every piece of the puzzle has its own properties, advantages and limitations.

It is important to account for these properties when putting things together.

Future improvements should try to reduce the number of potential errors.

Compatibility of existing application should be kept, or even improved, as good as possible.

MEINBERG – The Time and Frequency Company

- Founded in 1979 by Werner and Günter Meinberg
- Initial product range: DCF77 longwave radio receivers for industrial applications (1980)
- First self-developed GPS time receiver in 1993
- Full-depth manufacturer: research, development, design, production, sales and support in one hand
- In-house production of 90% of the mechanical (chassis/housing) and electronic (modules, integration) components
- 80 employees (20 R&D), one central campus in Bad Pyrmont
- approx. 70 km southwest of Hannover, Northern Germany

Thanks

Thanks for your attention!