

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>

**FOSDEM 2015
Brussels, Belgium
February 1, 2015**

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

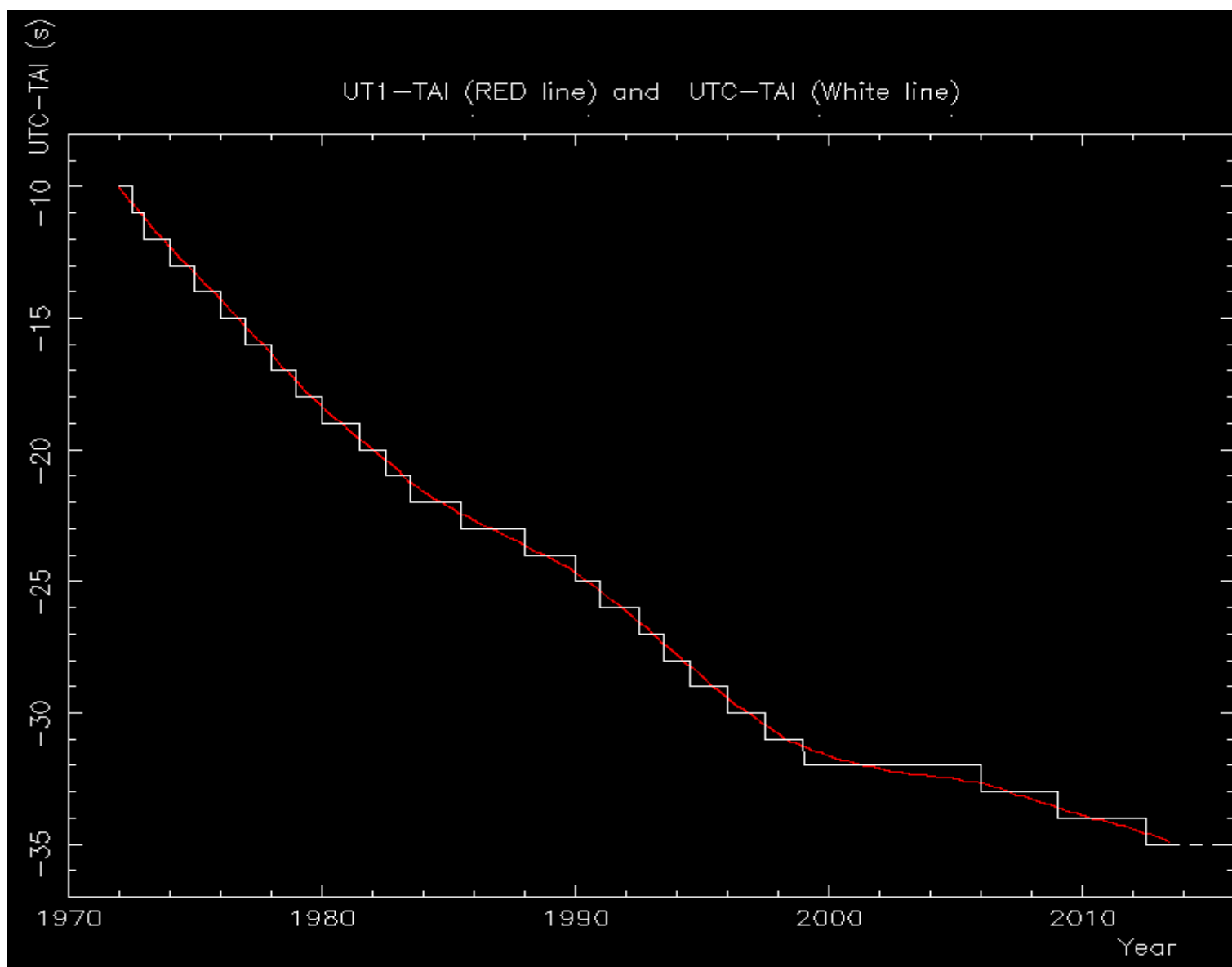
Since 1972 the length of a second is determined by atomic time, derived from Cesium atoms, base of UTC time

- 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>

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 years

INTERNATIONAL EARTH ROTATION AND REFERENCE SYSTEMS SERVICE (IERS)

SERVICE INTERNATIONAL DE LA ROTATION TERRESTRE ET DES SYSTEMES DE REFERENCE

SERVICE DE LA ROTATION TERRESTRE DE L'IERS

OBSERVATOIRE DE PARIS

61, Av. de l'Observatoire 75014 PARIS (France)

Tel. : 33 (0) 1 40 51 22 26

FAX : 33 (0) 1 40 51 22 91

e-mail : services.iers@obspm.fr

<http://hpiers.obspm.fr/eop-pc>

Paris, 5 January 2015

Bulletin C 49

To authorities responsible for the measurement and distribution of time

UTC TIME STEP
on the 1st of July 2015

A positive leap second will be introduced at the end of June 2015.

The sequence of dates of the UTC second markers will be:

2015 June 30, 23h 59m 59s

2015 June 30, 23h 59m 60s

2015 July 1, 0h 0m 0s

The difference between UTC and the International Atomic Time TAI is:

from 2012 July 1, 0h UTC, to 2015 July 1 0h UTC : UTC-TAI = - 35s

from 2015 July 1, 0h UTC, until further notice : UTC-TAI = - 36s

Leap seconds can be introduced in UTC at the end of the months of December or June, depending on the evolution of UT1-TAI. Bulletin C is mailed every six months, either to announce a time step in UTC or to confirm that there will be no time step at the next possible date.

Daniel Gambis

Head

Earth Orientation Center of IERS

Observatoire de Paris, France

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

UTC specifies how time is counted by enumerating seconds:

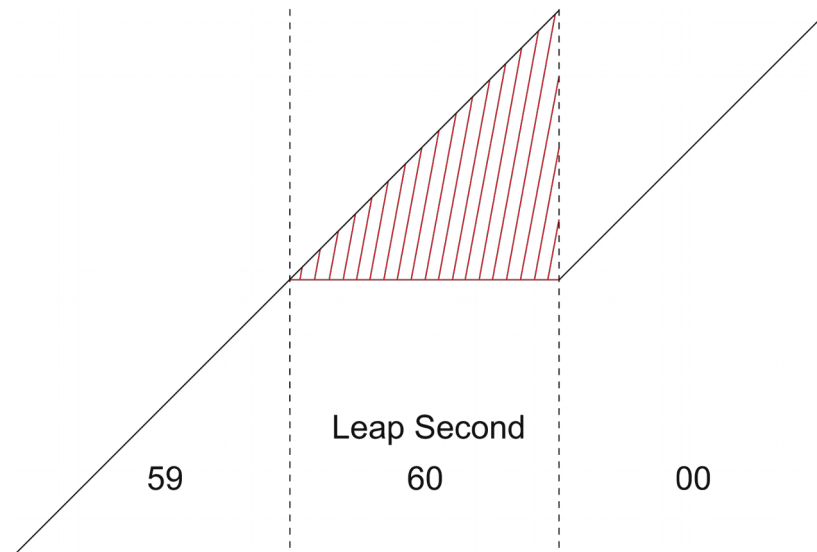
- Skip last second “59” to delete a leap second
- Extend second count to “60” to insert leap second

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
```

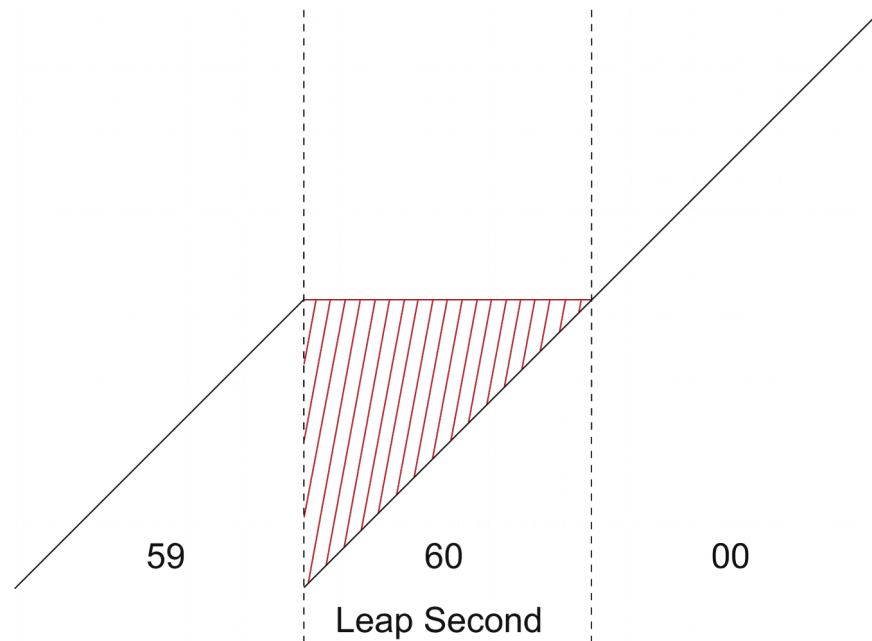
- Normalized times before and after leap second are the same
- Same system time stamps at beginning and end of leap second (POSIX)
- How to handle fractions during the second?

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



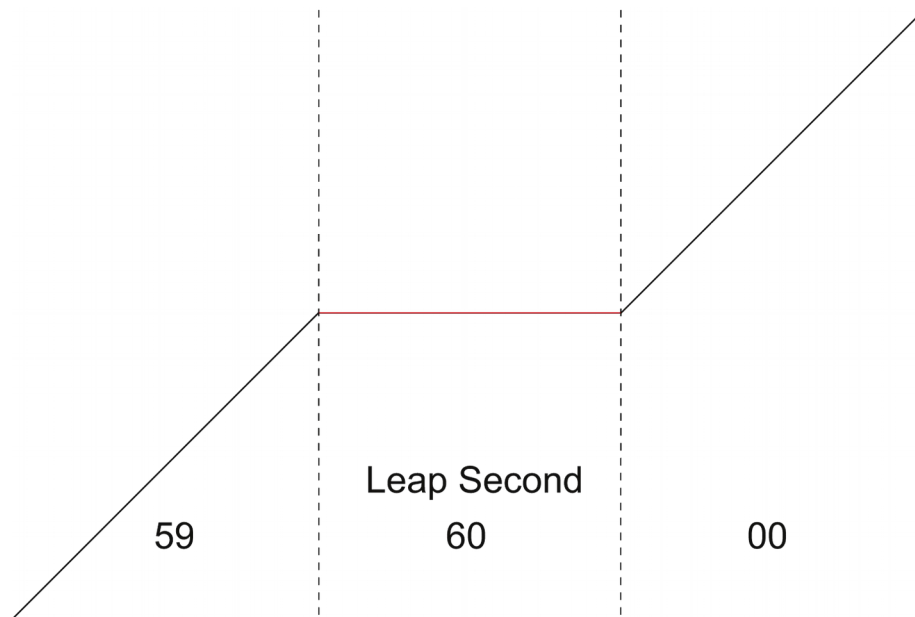
- Time doesn't increase monotonically
- Duplicate time stamps *after* the leap second
- *Earlier* time stamps for *later* events
- Ambiguous beginning of new minute / hour / UTC day

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

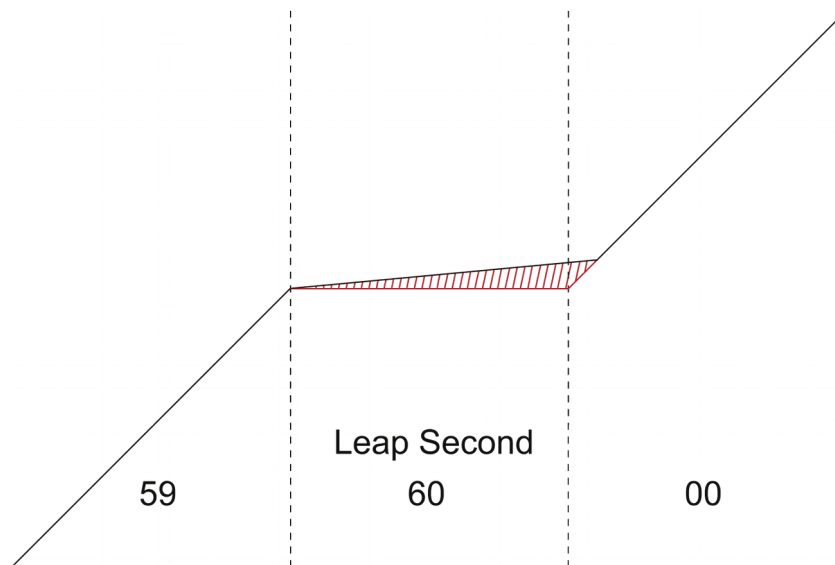
Stopping the clock for 1 second



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

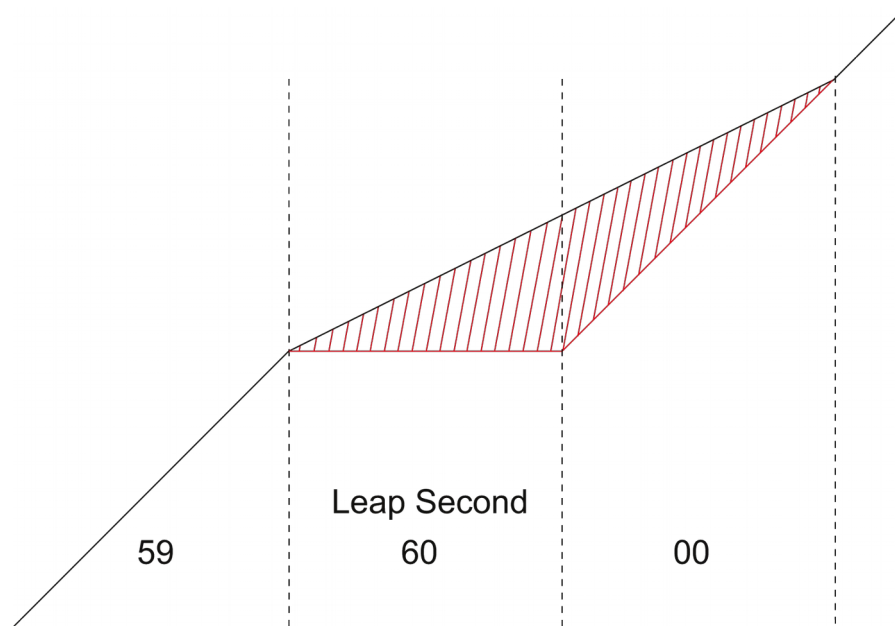
Leap Second Handling (5)

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

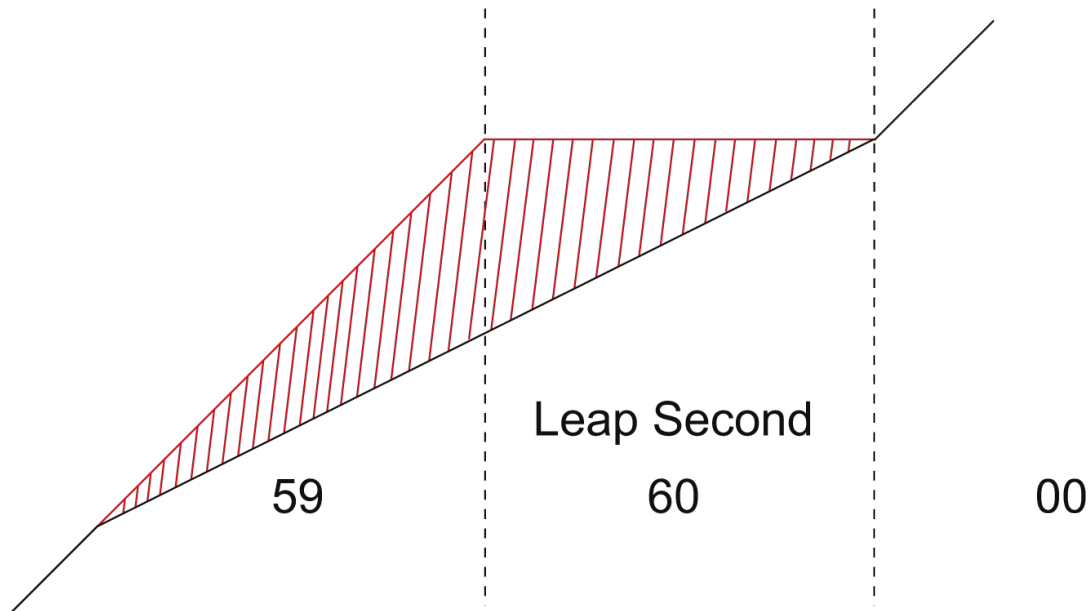
Slewing time half speed over 2 seconds



- Can be used if system clock does not support leap seconds
- Has been implemented in NTP port for Windows, before 4.2.8
- Initial workaround to support leap seconds under Windows
- Incorrect beginning of new minute / hour / UTC day

Leap Second Handling (7)

Slewing time half speed over 2 seconds



- 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
- Start of new minute / hour / UTC day accurate

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
- Start of new minute / hour / UTC day accurate
- Actually not implemented, as far as I know

Google Leap Smear

- Implemented by Google due to leap second problems observed in 2008
- Modified NTP servers send “modulated” time to clients
- Corrections applied smoothly, very slowly
- 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

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
- 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?

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
- How long in advance?

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

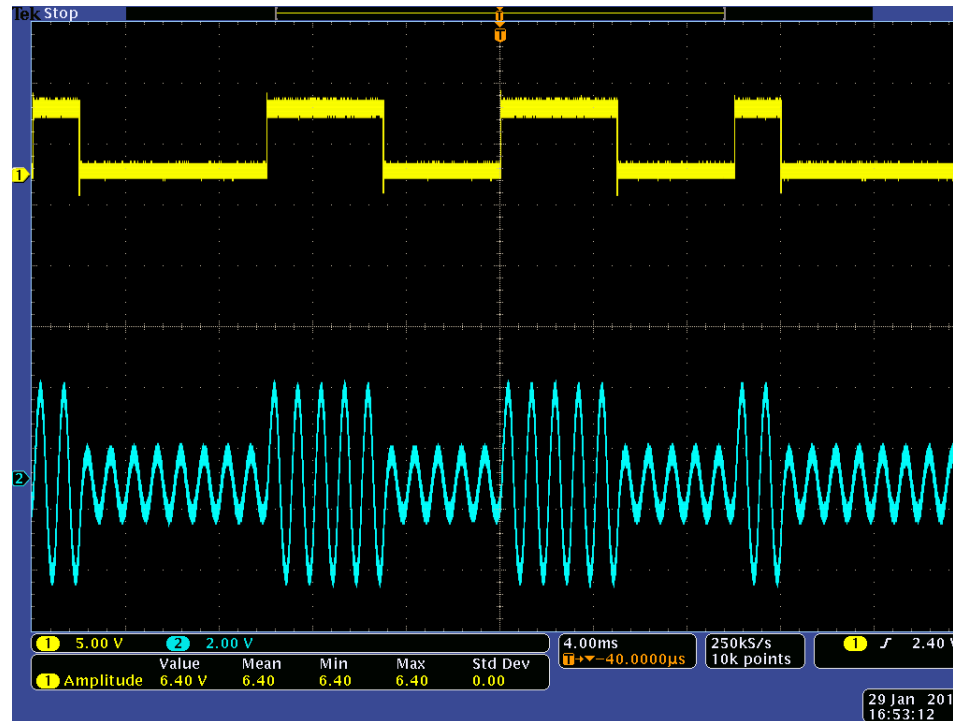
IEEE Time Codes

- 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 servers

Example of an IRIX Time Code Signal



Unmodulated IRIX signal (DC Level shift, DCLS)



Modulated IRIX signal, 1 kHz carrier frequency

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
- No way for a GPS receiver to tell that a leap second is pending if the selected string format doesn't support this

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
- Beside USNO and NIST the IERS is planning to provide 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
- chrony uses TZ leap second file for leap second warnings, AFAIK

PTP/IEEE1588 Network Protocol

- Works by default with TAI time stamps
- 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

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 the announcement is just passed down 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
- v4.2.6 uses leap second file if configured, **even if expired**

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
- Current Linux kernels step time back at beginning of leap second

Windows

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

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

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
- 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

Open Source PTP Daemon

- Leap second support added in v2.2.2

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
- Can still affect NTP servers since refclocks are considered authoritative
- As reported on the NTP pool mailing list this happened again in January 2015 when GPS satellites started to broadcast the leap second information
- Errors are due to improper coding

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

Without expiration date unable to determine if no leap second has been scheduled, or just no information is available.

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

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 NTF investigated on a new general time stamp format and associated API calls (Harlan Stenn's talk)

May not be appropriate for applications reading time at a high rate, with high resolution:

- Additional computation requires execution time
- Locking mechanisms may be required to guarantee consistent time stamp and status information

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

- PTP/IEEE1588 protocol uses TAI anyway
- Can be used to convert linear TAI or GPS time unambiguously to a time scale observing leap seconds
- Expiration date has recently been added to the TZ leap second file

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
- Ways to update leap second information automatically

Limitations when using leap second file or “right” time zone:

- Information needs to be updated regularly
- NIST format leap second file needs to be downloaded and replaced
- TZ data base updates usually deployed as software updates
- Embedded systems often don't get firmware updates, and thus might not be updated

Ongoing changes:

- New tzdist protocol allows update of the TZ rules and leap second information via network ([http/JSON](http://JSON))
- Proposal by Poul-Henning Kamp to use DNSSEC to distribute leap second information
- Long-lasting discussion to abolish leap seconds

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.

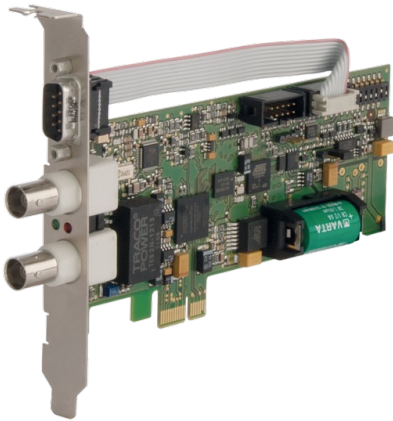
Compatibility of existing application should be kept, or even improved, as good as possible.



- 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
- 90 employees (20 R&D), one central campus in Bad Pyrmont
- approx. 70 km southwest of Hannover, Northern Germany

Products include:

- GPS receivers and radio clocks for DCF-77, MSF, and WWVB
- Full featured NTP / PTP time servers, PTP slave devices
- IRIG / AFNOR / IEEE time code generators / readers
- Frequency reference systems for telecommunications



Thanks



Thanks for your attention!