The UTC Time Scale: Internet timing issues

Judah Levine
Time and Frequency Division
NIST/Boulder
Jlevine@boulder.nist.gov

Focus is on time

- Time interval and frequency not considered
  - Network time services usually not used for this except at low accuracy
    - Calibrating stopwatches, timers, etc. where traceability to NIST or other NMI required

Outline of the presentation

- Realization of computer and network time
- Incorporating leap seconds
- Difficulties with current methods
- Possible solutions
- Conclusions

System time formats

- Seconds (and fractions) since epoch
  - Network Time Protocol uses 1900.0
  - Other choices: 1970.0, 1980.0, 17 Nov. 1858
- Time scale is almost always UTC
- Conversions done by applications
  - Local time zone, daylight saving time, …
  - Display formats, …

Computer clocks

- Oscillator generates periodic "ticks"
  - Hardware tick period not adjustable
- Register incremented on each tick
  - Increment value is adjustable in software
    - Normally always > 0
- Register can be over-written
  - Discontinuous setting of the clock
    - Strongly discouraged *except* during a cold start
- Register is basis for all system time functions

Realization of a leap second

- Time tags during a negative leap second:

	UTC
Day N	23:59:58
Day N+1	00:00:00

- Skipping a second does not present a very serious time problem
- Probably will never happen anyway

Realization of a leap second

- Time tags during a positive leap second:

	UTC
Day N	23:59:58
Day N	23:59:59
Day N	23:59:60
Day N+1	00:00:00

Realization of a leap second

- Time tags during a positive leap second:

	UTC	TAI	TAI-UTC
Day N	23:59:58	T	d
Day N	23:59:59	T+1s	d
Day N	23:59:60	T+2s	d+1
Day N+1	00:00:00	T+3s	d+1

Realization of a leap second

- Time tags during a positive leap second:

	UTC	Computers	
Day N	23:59:58	C	(23:59:58)
Day N	23:59:59	C+1s	(23:59:59)
Day N	23:59:60	C+1s	(23:59:59)
Day N+1	00:00:00	C+2s	(00:00:00)

Difficulties with the definition-1

- Computer clocks cannot represent a leap second and are effectively stopped when it occurs
  - *Most physical clocks have the same problem*
- Time sequence is:

23:59:59 .0, .1, …, .8, .9, .0, .1, …, .8, .9, …

Difficulties with the definition-2

- Time stamps can reverse time ordering of events and can violate causality:

  An event at 23:59:59.5 (#1) came

  before one at 23:59:59.4 (#2)

- Systems do not support adding flag to second time stamp to show leap second in progress

Difficulties with the definition-3

- Leap seconds can occur in the middle of a working day in Asia and Australia
  - Electronic commerce and digital transactions will be affected as soon as transactions depend on sub-second time resolution
    - This will be a problem sooner rather than later
    - Already a problem for NIST time services in supporting customers of online auctions (eBay)

Difficulties with the definition-4

- Implementation becomes more difficult as number of unsophisticated computer users who are engaged in e-commerce increases
  - Many PC operating systems do not support automatic insertion of leap seconds
  - Synchronization of wide-area networks lost or degraded by a leap second
  - Restoring synchronization places heavy load on time services
    - NIST time services currently handle 10^9 requests / day
    - Load immediately after leap second about 50X avg.

How many users are affected?

- NIST network time service receives about 10^9 requests per day
  - About 10^4 requests during leap second
  - Rate increasing about 8% per month
- Potential future impact: very serious
- Actual current impact: ?

All of these problems are going to get worse as the interval between leap seconds gets shorter.

What should we do?

1. Abandon leap seconds

- All previous problems disappear
- But –
  - ut1 correction becomes unbounded
    - Message format problems
    - Astronomy problems
    - Public relations problems
    - Legal time in US is MST (minor legal change)
- *Recommended only as a last resort*

2. Use TAI instead of UTC

- TAI time scale not readily available
  - NMIs and timing laboratories transmit only UTC
- Legal and commercial purposes require UTC
  - Conversion back from TAI possible but complicated and likely to produce lots of confusion
- NIST NTP Time servers transmit UTC and TAI
  - Does not help much *during* a leap second

3. Change leap second name

- Replace “23:59:60” with 23:59:59+flag to show leap second in progress
  - Flag could be used by applications to restore causality, etc.
  - Standard hardware clocks couldn’t do this, but they are broken in the current system too
  - Unknown, potentially large effects on lots of application software
  - Interesting, but probably not practical

4. Move leap second epoch

- Leap second epoch would be <u>only</u> on 1 January at 1200 UTC
  - Multiple leap seconds if needed
  - Business holiday in all time zones
  - Compromise:
    - Problems still remain but effects reduced
    - Advantages of current system preserved
      - ut1 correction remains bounded, might exceed 1s

Conclusions-1

- Any solution to leap second question will involve a compromise
  - Some undesirable effects will always remain
- Moving leap second epoch to 1200 UTC on 1 January is possible compromise
  - Minimal impact on all users
  - Preserves most current advantages

Conclusions-2

- Changing to TAI has lots of problems and will raise lots of objections
  - By using it directly
  - Implicitly by abandoning leap seconds
- Some form of leap second system is here to stay

Conclusions-3

- Changing the name of the leap second to be more compatible with digital time representations would be very helpful and should receive further study
  - 23:59:59 + "leap second in progress"
  - Use of 23:59:60 could remain for those systems that can support it
- Any change in leap second epoch should not depend on the outcome of this study