Leap Seconds in Digital Networks

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

Outline of the presentation

- Why are leap seconds necessary?
- How are leap seconds defined?
- How do digital systems keep time?
- How are leap seconds included?
- Difficulties with current methods
- Conclusions

Definition of Atomic Time

- Count cycles of frequency associated with a transition in cesium
  - 9,192,631,770 cycles = 1 s
- Cycle count chosen (13th CGPM, 1967) so that second was approximately continuous with previous astronomical definition
- Value was initially too small by about 3×10^{-8} relative to length of astronomical day at that time (UT1, 1967)

Atomic Time (UTC) vs. UT1

- Difference ~ 2.5 ms/day ≅ 0.9 s/year
  - Significant variability – difficult to predict
- Since 1972, discrepancy addressed by adding integer seconds to atomic time
  - $|UT1 - UTC| \leq 0.9$ s
- Additional "leap second" is named 23:59:60, usually added at end of June or December
  - Physical clocks cannot display this time

Digital 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
  - Direct comparison with other clocks
- Conversions done by applications
  - Local time zone, daylight saving time, …
  - Display formats, …

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

Step Realization of a leap second

- Time tags during a positive leap second:

	UTC	Digital System	
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)

Time difference= (C+2) – C = 2 s

Physical elapsed time= 3 s

Step Method Difficulty - 1

- Step Method: Clocks are effectively stopped
- Time sequence is:

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

Step Method Difficulty - 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)

- Step in time interval (frequency) across leap second
- Physical processes do not stop
  - Navigation systems have problems

Slew Method (Google)

- Time tags during a positive leap second

	UTC	Digital System Time
	UTC	Digital System Time
Day N:	23:58:00	23:58:00
Day N:	23:58:01	23:58:00.99
…		…
Day N:	23:59:00	23:58:59.50
…		…
Day N:	23:59:60	23:59:59.01
Day N+1:	00:00:00	00:00:00

Slew Method Difficulty - 1

- Time is monotonic, causality preserved
  - Significant time error for several minutes
- Smaller frequency error over longer time interval
  - Integrated time interval (frequency) the same as with step method

General Difficulties - 1

- Different implementations result in transient time and frequency offsets
- NTP phase-lock loop transient response
- Common systems do not recognize advance-notice flag or do not handle it correctly
  - No leap second, wrong sign, applied twice, applied late, …
    - Errors fixed in one version re-appear in next upgrade

General Difficulties - 2

- Physical processes do not stop during leap second
  - Navigation systems cannot include leap seconds
    - Use "system time" for navigation
- Additional time scales introduction confusion
  - Time stamps and time intervals
- Leap seconds occur in the middle of the next day in Asia and Australia, near end of day in California, Hawaii, …

If leap seconds were discontinued

- UTC – UT1 would increase
  - 1 minute after 100 years
- Uncertainty of UTC – UT1 << 1 s
- UTC no longer simple proxy for UT1
- Time stamps and time intervals would be less ambiguous
- Additional time scales no longer needed
  - GPS time, GLONASS time, Galileo time, …

Conclusions

- Leap seconds intended to maintain small value of |UT1 – UTC|
- Implementation encourages other time scales and adds confusion
- Implementation produces ambiguity in time stamps and time intervals
- Will becomes more difficult as number of unsophisticated computer users who need accurate time increases
- Leap second frequency predicted to increase
  - Problems will become more serious as this happens
- Advantage not worth the problems