NETWORK TIME FOUNDATION

Space-Based Positioning
Navigation & Time,
7-8 December, 2016

Communicating Time Looks Simple. So Why Haven't We Solved it Yet?

NETWORK TIME FOUNDATION

What I'm gonna say

- NTF: What, Why, Who.
- Time: From, To, Presumptions, Problems, and a Solution.
- Moving Forward: Timestamps, GTSAPI.
- Leapseconds, error bounds, Arithmetic.
- Timescales. POSIX.
- New issues, how does this help?
- GTSAPI in context with the wider scope.

NETWORK TIME FOUNDATION

Why NTF?

- The NTP Project needed the support and backing of a legal entity.
- I felt an “NTP Foundation” would have insufficient scope.
- Network Time Foundation seemed “right”.
- NTF is a registered 501(c)(3) US Charity.

- Combined Federal Campaign #18136

NETWORK TIME FOUNDATION

Who/What is NTF?

- NTF currently supports NTP, Ntimed, PTPd, LinuxPTP, RADclock and the General Timestamp API projects.
- Expecting to implement NTS this year.
- Looking to start a “Stratum 0 Consortium.”
- Developing Certification and Compliance programs.

NTF is all about the reliable communication and transfer of “time”.

Time Comes From…

USNO and NIST each know what they think the time is. They exchange time with each other and various other National Time Labs.

This collaboration produces TAI and UTC.

GPS and other methods “publish” this time.

NTF’s work on time protocols and open-source code “distribute” time.

NETWORK TIME FOUNDATION

Time Goes To…

NIST's time servers get 16 billion+ requests per day. 80% of these requests are via NTP. Their traffic is growing at 6%/month.

NTF's mission is the delivery of reliable time.

It is the responsibility of the "consumers" of time to have enough time sources, and to properly monitor them.

Presumptions

If we're on a computer and we need to know what time it is, we ask for a timestamp.

We simply assume the answer we get is correct. What other choice do we have?

Do we even consider any other conditions?

And later, when we use the timestamp?

NETWORK TIME FOUNDATION

The problems…

- Clocks are better now. We can easily measure:
  - earth rotation wobbles.
  - the effects of relativity.
- Timestamps are often only “locally” useful, and for somewhat limited span.
  - Daylight Savings Time.
  - Changes to Daylight Savings Time.
  - Leap Seconds.
  - Northeast blackout of 2003.
- Too many clocks are not synchronized.
  - Northeast blackout of 2003.
  - Medical Records.
- Too many sites use an insufficient number of time sources.
  - GPS signals can be jammed or spoofed.
  - Radio signals can be jammed or spoofed.
  - Network transmissions can be jammed/spoofed/altered.
- Current technology timestamps contain insufficient information. POSIX.

NETWORK TIME FOUNDATION

Current Timestamps

Current timestamps are mostly OK for “local use”. Mostly.

- seconds since some epoch
- <days since epoch>,<seconds since midnight>
- YYYYMMDD-HHMMSS - Long-standing hospital database does not bill millions of dollars each Fall’s daylight-savings correction

How To Do Better

Timestamps need to contain enough information so they can be reliably used in a wider scope, with better longevity. To do this, timestamps need to contain more information.

We propose NTF's General Timestamp API.

NETWORK TIME FOUNDATION

Timestamp Needs

- Monotonic time and databases
- System time may be known to be undergoing a correction.
- Error bounds?
- **What timescale is being used?**
- When comparing TS_0 and TS_1 did anything happen between those events that would affect the comparison? Did the clock change? Different timescales?

NETWORK TIME FOUNDATION

Timestamp Metadata

- A “clock discontinuity counter” is needed to show where “time steps” have occurred.
- A “host ID” is useful when comparing timestamps between multiple systems.
- A “clock ID” is useful if we need to know what the host is using to track the time.

NETWORK TIME FOUNDATION

About that Clock ID…

- With what degree of specificity should we know the source of time?

Multiple choice question:

- Is 13 microseconds very much time?

NETWORK TIME FOUNDATION

26 Jan 2016 and GPS

SVN 23 was the oldest GPS satellite still in operation on 26 Jan 2016, at the time it was being decommissioned. During that process, the legacy L-band signal was off by 13 µsecs from 00:49 MST until 06:10 MST.

13 microseconds *could* mean a position error of just under 4km / 2.5 miles.

More about this, later.

NETWORK TIME FOUNDATION

Timestamp Structure

- System time (or Elapsed time)
- Amount of pending correction
- Leapsecond correction (optional)
- Expected/Maximum error
- Timescale (and its revision #)
- Clock discontinuity counter
- Host and Clock ID
- Provable Signature
- Structure/API Version number, Flags

Putting it to Use

A new timestamp structure is only useful if it can be widely and generally portable:

- Kernel support
- Library support
- Application support (NTP, SQL, etc.)

NTPv5

The NTP model expects the other participants to play by the same rules.

Increasingly, this is not the case. With GTSAPI, we'd at least know the timescale the other system is using.

We'd also want to know some other behavioral choices.

NETWORK TIME FOUNDATION

Adjusting System Time

- Forward:
  - Adjust “system time”, or
  - Increase “amount of pending correction”
- Backward:
  - Decrease “amount of pending correction”

The OS applies pending corrections according to its policy rules.

Adjusting System Time

- Forward time adjustments seem to be pretty straightforward.
- Backward time adjustments are more challenging, as monotonic time is generally “good”.

If we want to step the time backwards, make tiny advances to the system clock when needed and decrease the pending correction by 1 second per second.

NETWORK TIME FOUNDATION

Leapsecond Timestamp Example

{System Time, Offset to Correct Time}

UTC	NTP-DLM (POSIX)	NTP-Windows59	NTP-Windows58	SMEAR24H
00:00:00.00	{00:00:00.00, 0}	{00:00:00.00, 0}	{00:00:00.00, 0}	{00:00:00.00, 0}
06:00:00.00	{06:00:00.00, 0}	{06:00:00.00, 0}	{06:00:00.00, 0}	{05:59:59.75, .25}
12:00:00.00	{12:00:00.00, 0}	{12:00:00.00, 0}	{12:00:00.00, 0}	{11:59:59.50, .50}
18:00:00.00	{18:00:00.00, 0}	{18:00:00.00, 0}	{18:00:00.00, 0}	{17:59:59.25, .75}
23:59:59.00	{23:59:59.00, 0}	{23:59:59.00, 0}	{23:59:59.00, .00}	{23:59:58.00, 1}
23:59:59.50	{23:59:59.50, 0}	{23:59:59.50, 0}	{23:59:59.50, .25}	{23:59:58.50, .5}
23.59.60.00	{23:59:59.99, .0}	{23:59:59.99, .0}	{23:59:59.99, .50}	{23:59:59.00, .0}
23.59.60.50	{23:59:59.99, .5}	{23:59:59.99, .25}	{23:59:59.99, .75}	{23:59:59.50, .0}
23.59.60.99	{23:59:59.99, 1}	{23:59:59.99, .50}	{23:59:59.99, 1}	{23:59:59.99, .0}
00:00:00.00	{00:00:00.00, 0}	{00:00:00.00, -.50}	{00:00:00.00, 0}	{00:00:00.00, 0}
00:00:00.50	{00:00:00.50, 0}	{00:00:00.50, -.25}	{00:00:00.50, 0}	{00:00:00.50, 0}
00:00:01.00	{00:00:01.00, 0}	{00:00:01.00, 0}	{00:00:01.00, 0}	{00:00:01.00, 0}

Using timestamps

The timestamp library API needs to handle:

- Adding/subtracting timestamps
  - Must accumulate error budgets
- Canonicalization of timestamps
- Comparing timestamps
- Converting timestamps (timescales)

NETWORK TIME FOUNDATION

Timestamp Arithmetic

T_A – Absolute Timestamp
T_D – Difference Timestamp

$T_A - T_A = T_D$

$T_A +/- T_D = T_A$

$T_D +/- T_D = T_D$

Of course, proper “accounting” of error budgets must be maintained.

Timestamp Error Budgets

NTP assumes that clocks accumulate error at the rate of 15ppm.

The initial error budget for a Difference timestamp is 0.

Otherwise, we generally care more about the magnitude of error as opposed to the error value.

Timescale Database

I’m operating on the belief that a timescale database won’t be that much harder to implement and maintain than Arthur David Olson’s Timezone Database.

There are groups actively working on tzdata dissemination.

NETWORK TIME FOUNDATION

Initial Timescales

Rare changes

- TAI/Satellite time (GPS, ...)
- Martian Standard Time
- UTC (leapseconds)
- Local Timezones (tzdata)
- IERS-A

Frequent changes

Should we bump the GPS version and track the error during 26 Jan 2016?

NETWORK TIME FOUNDATION

Timescale Identification

While it's overkill and likely way more than we need, using the 32-bit internet network class stuff is a useful way to start thinking about the problem.

The “network” portion can specify the timescale, and the “host” portion can specify the version of the timescale.

Ongoing Questions

POSIX mostly decided upon absolute event timers. It would be nice if we could come up with a mechanism to notify a process that the system time had changed so that the application had a way to decide if it wanted to adjust its events or not.

What about SIGWOKEUP, SIGTIMECHG?

NETWORK TIME FOUNDATION

How does this help?

Poor timekeeping and timestamps can be incredibly costly and terribly inefficient.

- Power Grid Failure
- Hospital E/R and healthcare data
- Vehicle Fleet Tracking

NETWORK TIME FOUNDATION

Certification and Compliance

- Being able to use a timestamp in a “provable” setting is very helpful.
- For a timestamp to be “provable” it needs to contain enough information to sufficiently understand its provenance, and know its boundaries and limits. GTSAPI.
- The entire “time chain” for the timestamp must be traceable and provable.

NETWORK TIME FOUNDATION

Free vs. Paid Time

Free timestamps must always be available.

Timestamps that cost money (even US$0.01 each) would be provable, traceable, and include liability insurance. The revenue from these would also help support Network Time and the time infrastructure.

NETWORK TIME FOUNDATION

Summary

- NTF: What, Why, Who.
- Time: From, To, Presumptions, Problems, and a Solution.
- Moving Forward: Timestamps, GTSAPI.
- Leapseconds, error bounds, Arithmetic.
- Timescales. POSIX.
- New issues, how does this help?
- GTSAPI in context with the wider scope.

NETWORK TIME FOUNDATION

Help NTF Help You

Visit http://nwtime.org and learn more about these issues and Network Time Foundation.

Join NTF and invite others to join, too!

Help NTF help you!

https://youtu.be/I-BYzaDwNoE