ARTWG J. Touch
Internet Draft Independent consultant
Intended status: Best Current Practice November 19, 2019
Expires: May 2020

Resolving Multiple Time Scales in the Internet
draft-touch-time-06.txt

Status of this Memo

Copyright Notice

Abstract

Internet systems use a variety of time scales, which can complicate time comparisons and calculations. This document explains these various ways of indicating time and explains how they can be used together safely. This document is intended as a companion to Internet time as discussed in RFC 3339.

Table of Contents

1. Introduction

A popular proverb reads, "a person with one clock always knows what time it is; a person with two clocks is never sure." Unfortunately, Internet systems rely on a variety of time references that often need to be reconciled. This document attempts to explain this issue

and provide advice on how to avoid temporal ambiguity.

There are various standards for expressing time, including Universal Coordinated Time (UTC) [ITU02], local time (UTC adjusted for time

zone location and daylight saving time shifts), and Unix time [OG08], as well as many others. Although the Internet has a standard for expressing time [RFC3339], this document explains the complexities of using any single such time scale and describes how to safely apply any one time scale and to accommodate concurrent use of different time scales. In particular, it focuses on the difficulties using a single time scale to indicate dates to users, to order events, and to measure intervals. This document ignores general and special relativistic effects.

Many time frames contain discontinuities, some of which are regular (e.g., time zones, leap days, and daylight saving time shifts), whereas others are irregularly introduced as needed (e.g., leap seconds or revisions to daylight savings time shifts). These discontinuities complicate interval computation, the latter requiring externally provided context (a table of mandated leap seconds and their scheduled occurrences). Other fine frames are non-uniform, in which the duration of an interval (e.g., a day, a year, or even a second) varies depending on its offset.

Despite many attempts, there is no single time scale that supports all common uses easily and without the need for updated external information. As a preview, this document makes the following recommendations for system designers:

1. System designers SHOULD NOT invent their own time scale. There are no simpler solutions and more than enough existing variants, although there is no known reason to exclude new variants.

2. System designers SHOULD use one time scale as their primary reference and derive all other time scales by conversion, to avoid confusion. Exceptions might optimize for more than one use.

3. System designers SHOULD use UTC as their primary time scale because it is most commonly accepted by governments and the basis for the Internet time [RFC3339] (based on ISO 8601 [ISO98]) and the Network Time Protocol (NTP) [RFC5905]. Exceptions optimize computation, e.g., to use TAI [BI06] for interval calculation or

local time [ISO98] for user interaction, e.g., calendars.

4. System designers SHOULD include location context (e.g., location or zone) as a part of all dates and MUST include that information if conversion to and between civil and local time is required.

5. System designers SHOULD maintain updated information regarding leap seconds and time zones and MUST maintain that information if accurate intervals or civil conversions are required.

6. System designers SHOULD be explicit about indicating whether intervals are inclusive or exclusive of start and end dates. Doing otherwise is an opportunity for ambiguity.

7. System designers SHOULD be very clear about whether timers expire on a date or when an interval has passed, to help understand the impact of continuous and monotonic aspects of time scales.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

The following terminology is used in this document. Note that some units are presented as "generally defined" (provided as approximations), whereas others are "precisely defined" (provided as specific).

- Instant: a specific moment in time.

- Time scale: a system for assigning names to intervals and dates.

- Interval: the elapsed time between instants.

- Date: the name of an instant in a time scale, typically indicated as an interval from an epoch.

- Epoch: an instant used as the origin (zero) of a time scale.

o Onset: an instant after which a time scale is valid. This term is introduced in this document.

o Expiry: an instant after which a time scale is invalid, in contrast to the onset. This term is introduced in this document. It applies most notably to the Julian calendar.

o Clock: a mechanism indicating the current date in a time scale.

o Civil time (or date): a time scale (or date) selected by a government authority.

o Solar day: a unit of time generally defined as the interval of one rotation of the earth as measured between the repeated position of the sun in the sky as viewed from a fixed location. Solar time relative to a single daily position is "apparent solar time". Solar time indicated as a mean over a year (one orbit of the earth around the sun) is "mean solar time". A given solar day can vary by as much as 30 seconds vs. the mean.

o Tropical year: a unit of time generally defined by the interval of one rotation of the earth around the sun as measured using the position of the sun in the sky in the same way as a solar day.

o Second: the unit of time, which has multiple definitions whose values differ, only the last of which is precise; others are derived from "generally defined" units:

  o 1/(24 * 60 * 60) of a solar day.

  o 1/(31,556,925.9747) of a tropical year as of the instant of 1900 "January 0" (i.e., December 31) at 12:00:00 Ephemeris time (Ephemeris time is defined later herein) [C61].

  o Exactly 9192631770 periods of the radiation corresponding to the hyperfine transition of the ground state of cesium 133 at 0K (precisely defined as an SI unit of time) [BI06].

o Leap seconds: an extra second irregularly inserted into or removed from the UTC time scale (based on SI seconds, see Sec.

4.2) to maintain it within 0.9 SI seconds of UT1 (based on solar days, see Sec. 4.2).

- Offset: an interval added or subtracted from a date or clock to convert between time scales with different epochs and leap seconds.

- Local time: a variation of a time scale intended to approximate that time scale at a given geographic location relative to that time scale at a reference geographic location, indicated as an offset.

- Time zone: an offset defined within a geographic region, used to compute local time.

4. Background

There are a variety of types of time scales in widespread use for scientific, civil, and computational purposes. Scientific time is

based on the International System (SI) standard definition of a second based on atomic clocks, and its goal is to provide a uniform standard for the passage of time. Civil time is based on the rotation of the earth and it attempts to coordinate a single geographic location with the same reference to the sun at the same time each day, including variations that support localized time to approximate that effect for other locations around the earth. Computational time is an approximation of civil time that is intended to be inexpensive for computers to maintain without external information.

Each of these time scales has different properties. Scientific time is intended to be continuous and uniform, so that one second of elapsed time always has the effect of moving a scientific clock forward one second. Civil time can be non-continuous, such as when leap seconds or leap days are added to compensate for the difference between elapsed time and the rotation of the earth relative to the sun. Computational time can be non-uniform, such as when the rate of Unix system clocks are varied to synchronize with civil time in a way intended to avoid discontinuity.

Each of these types of time scales also has different primary uses. Scientific time ensures uniform comparison of elapsed time and event

ordering. Civil time is used by people and their governments. Computational time is used by computers to approximate other time systems. Time represented in each of these systems can be converted to other representations, given sufficient additional information.

Time is used throughout the Internet, to govern protocols (e.g., timers in TCP [RFC793]), to improve efficiency (e.g., TCP RTT estimation using timestamps [RFC7323]), as well as to indicate a correlation with civilian time (e.g., NTP [RFC5905] and calendars [RFC5545]). Each of these types of uses has distinct requirements on the kind of time used.

4.1. Time uses and properties

Protocols use time for three primary purposes:

- Ordering: to determine the relative sequence of events across systems, such as with Lamport clocks [La78] or Vector clocks [Fi88][Ma88].

- Determining intervals: to determine actions to occur in a protocol in the absence of user requests and received messages (e.g., timers in TCP [RFC793]), to interact with physical systems (e.g., generating symbols at a given rate on a link), or to determine performance.

- Interacting with people: to exchange dates with the real world, as when indicating the civil date of an email transmission [RFC5321], web page [RFC7231], or managing calendars [RFC5545].

Each of these uses mandates a key property. Ordering requires that a time reference is monotonic and increasing, such that the time reference values change between any two events whose order needs to be established. Accurate interval calculation requires that a time reference also be continuous and uniform, such that the calculated differences between any two dates separated by the same interval yield the same value. Interacting with people requires the use of a time reference they already use, so that expressed dates have known

meaning.

These properties are not all supported by the variety of time references in widespread use. Some insert leap seconds and leap days, introducing discontinuities. Some vary their basic interval unit (e.g., to accommodate astronomical variances), which undermines their uniformity. These issues affect the choice of time reference and conversion between time references.

4.2. Time scales

The following is a description of the time scales in widespread use:

- TAI (International Atomic Time) [BI06]: a time scale based on the SI second at mean sea level ("on the geoid"), determined post-facto as a weighted average of a set of particular atomic clocks, adjusted to account for relativistic effects.

- UT (Universal Time) [Mc09][Sa78]: a time scale based on the solar day using zero degrees longitude as the earth location and a specific astronomical location (originally the sun, but now more distant objects). UT has several variants, of which the most common is UT1 (where UT is often synonymous with UT1), which includes corrections for earth axis variations.

- UTC (Coordinated Universal Time) [ITU02]: an approximation of UT based on TAI adjusted with leap seconds.

- Ephemeris time [C61]: an astronomical time reference, originating in Newcomb's tables [Ne1898] and standardized in 1952.

- Unix [OG08]: the POSIX/IEEE standard for Unix-based operating system software, in which dates are defined as the number of seconds that have elapsed since UTC 1970-1-1 00:00:00, increased by exactly 86,400 seconds per day (note that neither 'day' nor 'second' is defined in Unix time). Note that this is not the same as the POSIX time API (application programmer interface), which provides access to a variety of time scales.

The following are somewhat secondary to the time scales above:

o DUT1 [IERS]: the number of leap seconds between the current TAI date and the UTC epoch.

o GPS [Ha01]: the US Global Positioning System, defined as tracking time as TAI + 19 SI seconds.

o GLONASS [RI98]: Russia's satellite clock system, defined as tracking UTC.

o IRNSS/NAVIC [IRNSS]: The Indian Regional Navigation System.

o BeiDou-2 (prev. COMPASS) [NAE12]: China's satellite clock system.

o Galileo [Ga17]: the European Union's satellite clock system.

o NTP [RFC5905]: the Network Time Protocol, used in the Internet to synchronize local clocks, in which dates are indicated by UTC values. NTP times track the time of the clock they connect to.

Some of these time scales have a single reported value, such as GPS and NTP time. In other cases, the time scale is a weighted aggregate of contributions that are individually reported as well, such as UTC vs the component contributed by the U.S. Naval Observatory (UTC(USNO)) or the US National Institute of Standards and Time (UTC(NIST)). These components vary from their weighted averages, typically varying by only a few nanoseconds.

4.3. Comparison of properties

Time scales can be compared using the following properties, in addition to their epoch and the interval used as their unit of time:

o TAI error (Terr): a measure of the typically bounded precision on dates in the given time scale vs. TAI.

o Continuous (Cont): are dates in this time scale continuous, i.e., so that intervals can be calculated directly from the difference in dates.

o Uniform (Unif): are dates in this time scale uniform, i.e., so that all intervals of the same size represent the same amount of time.

o Onset (and expiry): the date after which a time scale is valid (or no longer valid), as introduced by this document.

Onset and expiry are not commonly indicated in many time scales. They are introduced here to help explain the difference between the zero time (epoch) of a time scale and the validity period of a time scale. Some time scales have no invalidity period, i.e., their onset is infinitely negative in the past, notably when values can be negative relative to their epoch. The Julian calendar has an expiry of 1852-10-05 and the Gregorian calendar has an onset of 1852-10-15, even though both calendars have an epoch of 0 AD and both calendars have been projected to dates in the past (at which point the difference is often not relevant, e.g., 100 BC).

The table below describes the time scales considered herein. All time scales use fixed epoch values except GLONASS, which reports dates relative to the current UTC. UT can drift in comparison to TAI by up to 0.9s, at which point a leap second is added. The satellite systems (BeiDou-2, Galileo, GPS, GLONASS, and IRNSS/NAVIC) attempt to track TAI, each with particular variances as design goals. NTP varies from TAI because of network latency variations, except where smearing is used [Go17]. Unix clocks typically use local quartz oscillators as clocks, which can drift from TAI by 1-2s/week unless continuously corrected, e.g., by NTP over the network.

Time scale	Epoch	Onset	Unit	Terr	Cont	Unif
TAI	1977-01-01	1960-01-01	SI	-	Yes	Yes
UT	0 AD (1)	1848-10-22	solar	0.9s	Yes	No
UTC	0 AD (1)	1972-01-01	SI	-	No	Yes
Unix	1970-01-01	epoch date	undef	~100s	Yes	undef

(1) The epoch of UT and UCT is when all fields are zero. Although time is expressed relative to that date, it precedes the onset. The onset dates indicate the onset of the most recent definition of the time scale indicated.

TAI was designed to be both continuous and uniform. UT was designed to be both uniform and track the solar day. The difference is

addressed in different ways in other time scales, which are largely derived from these two.

Unix time does not specify the definition of a 'second' or 'day', and so it is not clear whether it intends to track SI seconds (where time would be uniform) or solar time (where it would not).

5. Systems that report time

The following is a partial listing of widely used systems that report time.

Time scale	Epoch	Unit	Terr	Cont	Unif
BeiDou-2	2006-01-01	SI	100ns*	Yes	Yes
Galilelo	1999-08-22	SI	50ns*	Yes	Yes
GLONASS	UTC	SI	1ms*	No	Yes
GPS	1980-01-06	SI	25ns*	Yes	Yes
IRNSS/NAVIC	?	SI	?	Yes	Yes
NTP(1)	1900-01-01	SI	~100ms	No	Yes
NTP-smear(2)	1900-01-01	SI	1.1s	Yes	No

(1) As specified [RFC5905], error as per the FAQ [NTPfaq]
(2) Some servers, notably Google's, 'smear' leap seconds [Go17]
 * TAI comparisons from [Sa11]

6. Computing time

The concurrent use of multiple time scales results in the need to coordinate clocks and convert dates, and can complicate ordering. Conversions require more context than just the time units and epochs. It is also useful to be able to calculate the interval between two dates within a single time scale. Each of these calculations can require context, some of which cannot be statically encoded. Ordering depends on monotonically increasing clocks, which some time scales do not support.

6.1. Conversion

Dates in different time scales can be converted precisely as long as both time scales are uniform. When both time scales are also continuous, conversion is simple and relies only on the specification of the time scales. If either time scale is discontinuous, an additional table of discontinuities is required.

When either time scale is non-uniform, precise conversion is not defined unless the non-uniformity is also precisely indicated. The following subsections address each of these cases.

6.1.1. Continuous and uniform

For continuous and uniform time scales sharing the same unit of time, the difference in epoch is sufficient to convert one scale to the other, e.g.:

```
TS2_date = TS1_date - TS1_epoch + TS2_epoch
```

This conversion assumes both epochs are indicated in the same time scale (or can be converted to such - if not, no conversion is possible). For example, GPS reports the TAI date as an interval from January 6, 1980, whereas TAI reports the date as an interval from January 1, 1977, and both epochs are indicated in solar time. The difference between those two epochs is exactly 95,040,019 SI seconds, which is the total of 1,100 days of 86,400 SI seconds each and an additional 19 SI seconds, needed to align the epochs indicated as solar dates. As a result, dates indicated in year/month/day/second format need only have their seconds values adjusted as follows:

```
GPS_date = TAI_date + 19s
```

If time scales do not share the same unit of time, the conversion needs to account for the difference in the intervals from epoch. For example, a solar day is composed of 86,400 'solar seconds', but approximately 86,400.002 SI seconds. Conversion now requires that the epochs and units are expressed in a common time scale, and can be computed as follows:

```
TS2_date = (TS1_date - TS1_epoch)/TS1_unit * TS2_unit + TS2_epoch
```

Converting a common time frame to local time further requires knowing the location of each date and consulting a time zone database, which is also available online [tzdb] [RFC7808]. In this way, UTC can be converted to its local equivalent using a similar lookup operation (where TZDB is the time zone database):

```
UTC_localdate = UTC_date + TZDB[UTC_date, local_location]
```

6.1.2. Uniform but not continuous

Changes in the rotation of the earth cause variations in the difference between the unit of a second as defined by solar day,

tropical year, and SI methods. These differences are corrected by introducing "leap seconds", which are added (or removed) on specific dates [IERS]. E.g., UTC adds or removes leap seconds (known as DUT1) to TAI on specific dates to help it approximate UT.

Leap second dates can be approximated using a known calculation, but the exact date is determined administratively (rather than by calculation). Those dates are announced several months in advance and can be obtained online [tzdb][RFC6557][RFC7808]. As a consequence, conversion accounting for leap seconds requires a lookup operation (where "leapDB" is a database that indicates the number of leap seconds added since the epoch):

```
UTC_date = TAI_date + leapDB[TAI_date]
```

Between dates when leap second dates, precise differences in solar vs. SI time scales can be computed below 1s by accounting for the ratio between a solar second and an SI second, but this is rarely considered.

6.1.3. Not uniform

Some time scales are not uniform, i.e., the duration of an interval is indicated in units that vary and so are not easily directly comparable. Solar days vary by as much as 50 SI seconds because of the earth's variation in its axis of rotation. Because a solar day is defined as a fixed number of (solar) seconds, one solar second varies by as much as 0.06%. This variability is not simple to compute, but can be averaged out over long periods, but only in hindsight. Similarly, the earth's orbit around the sun varies and is slowing over time, resulting in an increasing stretching of a solar second.

Some time scales replace discrete leap seconds with a leap smear, stretching the interval of one second over 10-20 hours before (and sometimes after) the corresponding leap second date [Go17]. This allows a time scale to avoid discontinuities and non-conventional interval values (e.g., minutes with 59 or 61 seconds). Smearing causes non-uniformity of intervals that span the smear, especially because there is no current standard for the smear interval or algorithm.

Additionally, some time scales have no precise conversion, e.g., GPS

is coordinated to within 25ns of TAI, but the exact difference is known only as a post-facto measurement relative to NIST time (a subset of the clocks used to compute TAI) [NG]. This occurs because GPS uses its own set of atomic clocks rather than using the TAI

directly, and the same is true for other satellite systems. Other time scales are imprecise by definition, as with Unix time, which is based on clocks that vary widely by instance and with changes in temperature.

6.1.4. Ordering

Events in a distributed system often require ordering to ensure consistent views of their aggregate state [La78]. It can be important to know whether a bank deposit occurs before a withdrawal or if a license application has been submitted before a deadline. This can be accomplished for individual events using simple counters (Lamport clocks) but can become unwieldy for coordinating pairs or groups of events (Vector clocks [Fi88][Ma88]). Time scales can provide an alternative to these ordering mechanisms.

Time scales that are continuous enable easy ordering of dates, e.g., all dates comparisons correctly either indicate concurrence (when dates match) or a specific order. Time scales that are discontinuous can give false results, such as during a leap second. Consider the UTC date encodings indicated in Figure 1.

Instant	TAI date	UTC encoding (61s minute)
A	2016-01-01T00:00:34.0	2016-12-31T23:59:59.0
B	2016-01-01T00:00:34.5	2016-12-31T23:59:59.5
C	2016-01-01T00:00:35.0	2016-12-31T23:59:60.0
D	2016-01-01T00:00:35.5	2016-12-31T23:59:60.5
E	2016-01-01T00:00:36.0	2016-01-01T00:00:00.0

Figure 1 Leap seconds with long minutes

In both cases, two SI seconds progress between instants A and E. However, the last minute before midnight on December 31, 2016 has a minute that lasts 61 seconds (0..61), rather than 60. Ordering of these instants is unambiguous in this example.

Consider instead a system that cannot represent minutes with more

than 60 seconds. In such systems, the clock is either stalled or delayed during a leap second insertion, resulting in repeated values (Figure 2). Here, the order of instants B, C, and D cannot be established accurately from the dates. Additionally, intervals that span this "reset" are inaccurately calculated from date differences unless explicitly corrected.

Instant	TAI date	UTC encoding (60s minute)
A	2016-01-01T00:00:34.0	2016-12-31T23:59:59.0
B	2016-01-01T00:00:34.5	2016-12-31T23:59:59.5
C	2016-01-01T00:00:35.0	2016-12-31T23:59:59.0
D	2016-01-01T00:00:35.5	2016-12-31T23:59:59.5
E	2016-01-01T00:00:36.0	2016-01-01T00:00:00.0

Figure 2 Leap seconds with repeating dates

Ordering can be restored using leap smear, as shown in Figure 3, but at the expense of complicating the computation of intervals that span the duration of the smear, which can be several hours.

Instant	TAI date	UTC encoding (60s minute)
A	2016-01-01T00:00:34.0	2016-12-31T23:59:59.0
B	2016-01-01T00:00:34.5	2016-12-31T23:59:59.25
C	2016-01-01T00:00:35.0	2016-12-31T23:59:59.5
D	2016-01-01T00:00:35.5	2016-12-31T23:59:59.75
E	2016-01-01T00:00:36.0	2016-01-01T00:00:00.0

Figure 3 Leap seconds with smear

6.2. Calculating intervals

Intervals can be calculated directly between two dates of a uniform time scale directly as the difference between two dates A and B as follows (where "abs" is the absolute value function):

```
interval = abs(dateA - dateB)
```

It is important that the specification of an interval indicate

whether its endpoints are included or not, e.g., whether the interval is open, half-open, or closed. In common mathematical notation, the interval [1:24.00, 1:25.00] includes both 1:24.00 and 1:25.00. The interval [1.24.00, 1:25.00) starts at the instant of 1:24.00 and ends just before the instant of 1:25.00, i.e., 1:25.00 is excluded from the interval. System designers SHOULD clearly indicate whether intervals include or exclude their start and end instants.

A non-continuous time scale requires external information, e.g., the leap second dates that occur during the interval. Computing intervals in these time scales requires that the representation does not repeat or smear time. The interval is computed by converting

non-continuous time to continuous time by removing the effect of leap seconds and proceeding as with the continuous case, as follows:

```
interval = abs((dateA - leapDB(dateA)) - (dateB - leapDB(dateB)))
```

Non-uniform time scale intervals can sometimes be calculated, but this is rarely supported. Computing intervals into the future can be hazardous due to unpredicted changes, e.g., the addition of future leap seconds or changes in time zones and daylight savings time.

7. Advice

No single time scale serves all purposes. Use of multiple time scales requires conversion, which often requires external information. Maintaining accurate clocks can also require external information (to insert leap seconds), as can the computation of intervals.

7.1. Selecting a time scale

A primary time scale SHOULD be chosen from among existing time scales, if possible. Creating a new time scale increases complexity and is unlikely to avoid the issues already present with existing time scales, e.g., being continuous, uniform, requiring conversion, or needing external information for conversion or interval computation.

A primary time scale SHOULD be chosen to minimize the need for

repeated conversion and/or to minimize the complexity of computing intervals, depending on the expected frequency of these operations.

For example, if synchronizing clocks with other systems using NTP is the primary goal, implementers would probably select UTC [RFC5905]. If user presentation is primary, as for email or calendaring, implementers would probably select local time (which requires a comprehensive table) [RFC5545]. If interval computation is primary, implementers would probably select TAI.

As a consequence, in most cases, implementers seeking a primary time scale SHOULD select either TAI or UTC, or a system that closely approximates these (e.g., GPS-like systems or NTP), and expect to maintain updated leap second information [RFC7808].

7.2. Hazards of some time scales

Incorrect time scale selection can result in increased computational overhead and the need for increased storage. External information

might be needed for conversion, or conversion or calculation may not be possible (as with smearing).

Implementers SHOULD NOT use time scales that smear, for two reasons. First, there is no current standard for leap smearing, so the same time scale implemented on different systems are likely to indicate incorrect relative dates (i.e., incorrectly indicating instance ordering). Second, leap smearing complicates interval measurements computed over the smear which can be difficult to compensate.

7.3. Alternate solutions for ordering

Time scales can be used for ordering but other solutions can be simpler and less dependent on external sources. Notably, Lamport clocks [La78] provide individual event ordering and Vector clocks [Fi88][Ma88] and their derivatives provide event pair ordering, both without the need for precise time keeping and epoch coordination.

These mechanisms rely on the use of integer counters that increase with each event and tracking those counters where their uses provide a continuous trace that indicates an ordering. They rely on direct interactions and corresponding message exchanges to provide that trace, which can be complex and incur high overheads in some cases.

These systems become increasing complex as groups of events require ordering and may not be feasible when post-facto ordering is desired in the absence of direct communication.

8. Security Considerations

Time is used within security systems for a variety of reasons, including indicating the validity of certificates used for encryption and authentication [RFC5280]. Inaccurate dates, intervals, or ordering can affect the ability to use these protocols.

As a result, it can be important to secure the protocols used to coordinate time [RFC7384]. NTP, the most common such protocol, supports secure operation [RFC5905].

9. IANA Considerations

This document has no IANA considerations. This section should be removed prior to publication as an RFC.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words," RFC 2119, May 2017.

10.2. Informative References

[BI06] Bureau International des Poids et Mesures (International Bureau of Weights and Measures), "The International System of Units (SI)," 8th Edition, 2006.

[C61] http://www.bipm.org/en/CGPM/db/11/9/ Comptes Rendus de la 11e CGPM (1960), 1961, pp. 86.

[Fi88] Fidge, C., "Timestamps in Message-Passing Systems That Preserve the Partial Ordering," Proc. 11th Australian Computer Science Conference (ACSC'88), pp. 55-66.

[Ga17] Galileo system web pages, http://galileognss.eu/gst-galileo-system-time/

[Go17] Google's approach to NTP leap smearing, proposed in 2017. https://developers.google.com/time/smear

[Ha01] Hoffman-Wellenhof, B., H. Lichtenegger, J. Collins. Global positioning system: theory and practice. New York, Springer-Verlag, 2001.

[IERS] International Earth Rotation and Reference Systems Service (IERS). https://www.iers.org/

[IRNSS] Global Indian Navigation Satellites: Constellation studies, August ISRO-ISAC-IRNSS-TR-0887, 2009.

[ISO98] International Standards Organization, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO 8601, 1988.

[ITU02] International Telecommunication Union, "Standard-frequency and time-signal emissions," ITU-R Recommendations and Reports, TF.460-6, 2002.

[La78] Lamport, L., "Time, Clocks, and the Ordering of Events in a Distributed System," Communications of the ACM, V21 N7, Jul. 1978, pp. 558-565.

[Ma88] Mattern, F., "Virtual Time and Global States of Distributed Systems," Proc. Workshop on Parallel and Distributed Algorithms, pp. 215-226.

[Mc09] McCarthy, D., P. Seidelmann, "Time Applications," in Time - From Earth Rotation to Atomic Physics, Wiley-VCH, 2009. doi: 10.1002/9783527627943.ch19

[NAE12] National Academy of Engineering, "Global Navigation

Satellite Systems," National Academies Press, ISBN 978-0-309-22275-4, 2012.

[Ne1898] Newcomb, S., Tables of the Four Inner Planets," Second Edition, 1898. https://ia801005.us.archive.org/11/items/06AstronomicalPapersPreparedForTheUse/06-Astronomical_Papers_Prepared_for_the_Use_text.pdf

[NG] NIST vs. GPS time, https://www.nist.gov/pml/time-and-frequency-division/services/gps-data-archive

[NTPfaq] NTP FAQ pages, http://www.ntp.org/ntpfaq/NTP-s-algo.htm

[OG08] The Open Group, "Base Specifications, Issue 7, 2016 Edition," IEEE Std 1003.1 / POSIX.1-2008, 2008.

[RFC793] Postel, J., "Transmission Control Protocol" RFC 793, September 1981.

[RFC3339] Klyne, G., C. Newman, "Date and Time on the Internet: Timestamps," RFC 3339, July 2002.

[RFC5280] Cooper, D., S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, May 2008.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol," RFC 5321, Oct. 2008.

[RFC5545] Desruisseaux, B. (Ed.), "Internet Calendaring and Scheduling Core Object Specification (iCalendar)," RFC 5545, Sep. 2009.

[RFC5905] Mills, D., J. Martin (Ed.), J. Burbank, W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, June 2010.

[RFC6557] Lear, E., P. Eggert, "Procedures for Maintaining the Time Zone Database," RFC 6557, BCP 175, Feb. 2012.

[RFC7231] Fielding, R., J. Reshke (Eds), "Hypertext Transfer

Protocol (HTTP/1.1): Semantics and Content," RFC 7231, June 2014.

[RFC7323] Borman, D., R. Braden, V. Jacobson, R. Scheffenegger (Ed.), "TCP Extensions for High Performance," RFC 7323, Sep. 2014.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks," RFC 7384, Oct. 2014.

[RFC7808] Douglass, M., C. Daboo, "Time Zone Data Distribution Service," RFC 7808, Mar. 2016.

[RI98] Russian Institute of Space Device Engineering, "GLONASS Interface Control Document", 1998.

[Sa78] Sadler, D., "Mean Solar Time on the Meridian of Greenwich," Quarterly Journal of the Roayal Astronomical Society, V19, Sept. 1978, p.290. http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1978QJRAS..19..290S

[Sa11] Sanz Subirana, J., J. Juan Zornoza, M. Hernandez-Pajares, "Time References in GNSS," Navipedia web pages, 2011. http://www.navipedia.net/index.php/Time_References_in_GNSS

[tzdb] Time zone database, https://www.iana.org/time-zones

11. Acknowledgments

This work originated in response to a proposal for a new continuous time scale by Phillip Hallam-Baker, and benefitted from discussion on the IETF list, notably Steve Allen, Gerard Ashton, Patrik Falstrom, Tony Finch, Nicholas Mailhot, Juergen Schoenwaelder, Michael Thornburgh, and Nico Williams.

This work is partly supported by USC/ISI's Postel Center.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Joe Touch

Manhattan Beach, CA 90266 USA

Phone: +1 (310) 560-0334
Email: touch@strayalpha.com

Change Log:

draft-touch-time-06:

Update discussion of epoch to add onset date.

draft-touch-time-05:

Numerous clarifications to address imprecision of definitions.

Added discussion on alternate solutions for ordering.

draft-touch-time-04:

Revised terminology to indicate that some definitions are not precise

Clarified the use and benefits of integer (Lamport, Vector) clocks

Clarified that some time scales have individual (UTC(NIST)) and aggregate (UTC) values.

draft-touch-time-03:

Revise doc to more clearly target summarized recommendations.

Sec 4.2 definitions revised based on feedback:

- solar day now defined as two different things

- another scrub of existing definitions

draft-touch-time-02:

Sec 4.2 definitions revised based on feedback

Explain difference between Unix time and POSIX time API

draft-touch-time-01:

Sec 1 expanded to include list of recommendations.

Sec 5.2 more detailed description of intervals.

draft-touch-time-00:

(original version)