Will the INTERNET's Network Time Protocol Fail in 2000 or 2036?

by

Richard A. Painter, P.E.

Painter Engineering, Inc.

8470 Swan Rd.
Black Forest, CO 80908
719 495 7054

painter@ieee.org

Abstract

The INTERNET's Network Time Protocol (NTP) and Simple Network Time Protocol (SNTP) will likely fail on host computers in 2036 without INTERNET-wide changes. INTERNET primary time servers, local time servers and client computers world wide, numbering in the millions, and possibly these protocols, need to be changed to prevent serious time problems on this future date. Some people consider it highly unlikely that the NTP timescale, protocol design and reference implementation will produce incorrect time values in 2000. Even so, now is the time to correct the future traps before usage expands beyond a point to reasonably correct it. This paper reviews the NTP's essential time issues and describes the potential problems for 2000 and 2036.

Will the INTERNET's Network Time Protocol Fail in 2000 or 2036?

I. Introduction

The INTERNET's Network Time Protocol (NTP) is used throughout the INTERNET and intranetworks to synchronize many computer systems' notion of the current time. The protocol is only one part of the required elements to accomplish this. Reliable and accurate clock sources must be accessed by top-level, called stratum 1, primary time server computers. These computers must broadcast or reply to queries using the NTP to other computers to disseminate over networks the time information. Finally, the consumers' computers use the NTP information to maintain the local clocks.

NTP is more than a method to distribute time. Its protocol provides methods to account for the inherent network propagation delays and manages time information from multiple sources to increase accuracy and reliability.

Computers of all makes, models and operating systems use this time keeping system. Most UNIX systems are bundled with the client programs needed to receive network broadcasts or make requests over IP networks, both intranet and INTERNET. Personal computers, especially those running Linux and BSD386, are fast becoming consumers of the NTP. INTERNET Service Providers (ISP) use NTP and personal computer client programs, such as Frontier Technologies' SuperTCP, that run on Microsoft Windows support NTP. As more personal computers become connected to the INTERNET use of the NTP is likely to grow significantly.

The INTERNET has other time-bearing protocols. Some of these protocols have evolved since the 1980's with numerous versions obsolescing older ones. Below is a table of the latest INTERNET's Request For Comments (RFC) documents describing time protocols.

Table 1.- INTERNET Time Protocols

RFC	Date	Title	Time Data	Port
792	Sep. 1981	INTERNET Control Message Protocol	32-bit unsigned milliseconds since midnight UTC	ICMP Msg type 13, 14
867	May 1983	Daytime Protocol	ASCII string returned [Mon Apr 20 18:11:06 1998\n] with 25 characters	TCP|UDP Port 13
868	May 1983	Time Protocol	32-bit unsigned seconds since 00:00 1 January 1900 GMT	TCP|UDP Port 37
1305	Mar. 1992	Network Time Protocol Version 3	32-bit unsigned seconds, 32-bit unsigned fraction of seconds since 00:00 0 January 1900 UTC	TCP|UDP Port 123
1769	Mar. 1995	Simple Network Time Protocol	32-bit unsigned seconds, 32-bit unsigned fraction of seconds since 00:00 0 January 1900 UTC	TCP|UDP Port 123

II. NTP Analysis

As can be seen in Table 1, the NTP is very similar to the newer Simple Network Time Protocol (SNTP). In fact the protocol format is identical but SNTP does not employ all of the features of NTP, hence it's name: *Simple*.

All elements of the NTP packet are in Big-Endian format. That is, the bits are numbered starting at zero (0) from the high-order bits to the low-order bits.

The NTP packet format as represented in RFC-1769 is as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|LI | VN  |Mode |    Stratum    |     Poll      |   Precision   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Root Delay                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Root Dispersion                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Reference Identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Reference Timestamp (64)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Originate Timestamp (64)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Receive Timestamp (64)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Transmit Timestamp (64)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Authenticator (optional) (96)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This NTP packet contains four (4) timestamps: *Reference*, *Originate*, *Receive*, and *Transmit*. These timestamp formats are as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Seconds                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Seconds Fraction (0-padded)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 32-bit, unsigned integer representing *seconds* has a range from 0 to 4294967295 inclusive. This is equivalent to 138 years plus part of a year. This time span is known as the NTP ERA. The starting point is called an EPOCH and is defined for NTP as 00:00 0 January 1900 UTC. Therefore, this 32-bit integer will *roll over* in 2036. Clearly, this is reasonably past 2000 and is unlikely to be an issue for the next few years at least.

However, computers are creators of this packet's content and computers are its consumer. If the computers and their software have 2000 failures then they could disseminate bad time data within the NTP packet. David L. Mills states on his Web page that "...it is possible that external reference time sources used by NTP could misbehave and cause NTP server to distribute incorrect time values to significant portions of the INTERNET." But what are the potential problems in 2036?

> Please note that Mr. Mills' Web page states **incorrectly** that the UNIX internal time representation will wrap around in 2106! The correct date is Tue Jan 19 03:14:07 2038. See the reference paper *The UNIX Time will run out in 2038!* for details and a demonstration program. The author's papers are available at http://painterengineering.com

RFC-1305 expresses that it is the consumer computers' responsibility to interpret the EPOCH. Unfortunately, programmers forgot about this part just as they did for two of the four digits in year representations that have lead us to the 2000 crisis.

The starting EPOCH for both NTP and SNTP is 00:00 0 January 1900 where the EPOCH for the UNIX timestamp structure, *time_t*, is 00:00 1 January 1970. Since NTP sources base the time on UTC periodic adjustments are likely to happen, sometimes each year. This make it hard to precisely pinpoint the start date of the next NTP EPOCH. The following is an approximation of NTP's next EPOCH.

NTP ERA	= 4294967295 seconds
	= 71582788 minutes, 15 seconds
	= 1193046 hours, 28 minutes, 15 seconds
	= 49710 days, 6 hours, 28 minutes, 15 seconds
	= 136 years, 36 days, 6 hours, 28 minutes, 15 seconds
Next EPOCH	= 5 February 2036 (approximate)

It is not possible to set the next NTP EPOCH time until near the end of the current EPOCH due to unknown UTC adjustments! However, the software and possibly the NTP protocol could be modified now in a manner to be ready for the change. Also note while the first NTP EPOCH was a nice date the next will likely be in the early hours (UTC) of February 5th with nonzero minutes and seconds! Even with additional software the change over may be difficult for individual computers and may require human intervention. Here is an excerpt (emphasis added) from RFC-1769 that clearly indicates that the EPOCH interpretation was left for the future to bail out:

> `Note that, since some time in 1968 the most significant bit (bit 0 of the integer part) has been set and that the 64-bit field will overflow some time in 2036.` **`Should NTP or SNTP be in use in 2036, some external means will be necessary to qualify time relative to 1900 and time relative to 2036`** `(and other multiples of 136 years). Timestamped data requiring such qualification will be so precious that appropriate means should be readily available. There will exist a 200-picosecond interval, henceforth ignored, every 136 years when the 64-bit field will be 0, which by convention is interpreted as an invalid or unavailable timestamp.`

III. Conclusions

The Network Time and related protocols are likely to weather 2000 but the computers that use this information, including our personal ones, could fail to properly set the time due to the popular two-digit year 2000 problems. Stratum 1 time servers could produce discontinuities based upon 2000 date problems depending on their implementations.

Unfortunately, the present NTP client software appears to have no mechanisms for end users to set or manage EPOCH changeover short of reprogramming software. It is unlikely that vendor support or source code for many systems will be available as this date approaches, a similar situation for many for year 2000. If these same computers provide incorrect time to other networks and computers the propagation results are obvious.

Even when the NTP EPOCH issue is resolved and client computers are outfitted with reasonable mechanisms to affect the changeover the UNIX date problem in 2038 will likely cause both NTP and UNIX to behave improperly. Hence, both must be fixed.

Now is the time to join with the year 2000 efforts and make the needed modifications to handle the NTP EPOCH changeover. The cost and impact now will be a fraction of that 30 years hence.

IV. References

1. RFC 792, *INTERNET Control Message Protocol*, September 1981, J. Postel, ISI.
2. RFC 867, *Daytime Protocol*, May 1983, J. Postel, ISI.
3. RFC 868, *Time Protocol*, May 1983, J. Postel, ISI, K. Harrenstien, SRI.
4. *The UNIX Time will run out in 2038!*, December 1996, Richard A. Painter, P.E., Painter Engineering, Inc.
5. RFC 1305, *Network Time Protocol (Version 3) Specification, Implementation and Analysis*, March 1992, David Mills, University of Delaware.
6. RFC 1769, *Simple Network Time Protocol (SNTP)*, March 1995, David Mills, University of Delaware.
7. *INTERNET Time Synchronization: The Network Time Protocol*, IEEE Trans. On Comm. Vol. 39 No. 10 October 1991, David Mills.
8. *Improved Algorithms for Synchronizing Computer Network Clocks*, IEEE/ACM Trans. On Networking, Vol. 3 No. 3 June 1995, David Mills.
9. http://www.eecis.udel.edu/~mills/y2k.htm, *Year 2000 Conformance Statement*, 22 April 1998, David Mills, University of Delaware.
10. *UNIX Network Programming*, 1990, W. Richard Stevens, Prentice Hall ISBN 0-13-949876-1.
11. *TCP/IP Illustrated, Volume 1 The Protocols*, 1994, W. Richard Stevens, Addison-Wesley ISBN 0-201-63346-9.

NTP Packet Format

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|LI | VN  |Mode |    Stratum    |     Poll      |   Precision   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Root Delay                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Root Dispersion                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Reference Identifier                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Reference Timestamp (64)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Originate Timestamp (64)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Receive Timestamp (64)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Transmit Timestamp (64)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Authenticator (optional) (96)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NTP Timestamp Format

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Seconds                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Seconds Fraction (0-padded)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```