This document defines requirements for the Network Time Protocol (NTP) Version 4. NTP provides the mechanisms to synchronize time and coordinate time distribution amongst internet hosts.
Table of Contents

1. NTP Requirements Open Issues ........................................ 3
2. Introduction ......................................................... 3
3. Terminology ......................................................... 4
4. Algorithm Requirements ............................................... 6
   4.1 Clock Discipline .................................................. 6
   4.2 Accuracy ......................................................... 6
   4.3 Jitter ........................................................... 7
   4.4 Wander .......................................................... 7
5. Protocol Requirements ................................................. 7
   5.1 Configuration Requirements ....................................... 7
      5.1.1 Manual Configuration ....................................... 7
      5.1.2 Automatic, Autonomous Configuration ....................... 7
      5.1.3 Vendor Pre-configuration ................................... 7
      5.1.4 Administrative Domains ..................................... 8
      5.1.5 Key Distribution ............................................ 8
   5.2 System Performance ............................................... 8
      5.2.1 Scalability .................................................. 8
      5.2.2 Client Performance Requirements ............................ 8
      5.2.3 Server Performance Requirements ............................ 8
   5.3 Security Requirements ............................................ 8
   5.4 Internet Protocol Version 6 Requirements ........................ 8
   5.5 Robustness ...................................................... 8
      5.5.1 Authentication & Access Control ............................ 9
      5.5.2 Client/Peer Rejection ..................................... 9
   5.6 Longevity, Persistence .......................................... 9
      5.6.1 Reconfiguration ............................................. 9
7. Operational Requirements ............................................. 10
   7.1 Client Poll Interval ............................................... 10
8. Security Considerations ............................................. 10
9. IANA Considerations ................................................ 10
10. Acknowledgements .................................................. 11
11. References ......................................................... 11
   11.1 Normative References .......................................... 11
   11.2 Informative References ........................................ 11
   Author’s Address .................................................... 11
   Intellectual Property and Copyright Statements .................... 12
This Internet Draft’s editor maintains the most current revision at http://net.doit.wisc.edu/~plonka/ntp-reqs/ [5]. You may find an updated document there if draft submission cut-offs have delayed its availability elsewhere.

In this revision of this Internet Draft the keyword "FIXME" is used to mark locations where text will likely be added or modified. In subsequent revisions these might be changed to XML comments in the original source file, but for now they indicate the early stage of this draft.

1. NTP Requirements Open Issues

   1. How can we best address SNTP? Currently SNTP Version 4 is defined by its own Informational RFC (RFC 2030). This editor’s suggestion is that we either have our new NTP Version 4 documents each contain a SNTP section for the SNTP-pertinent content or to have a new standards-track SNTPv4 protocol document as a companion to the NTPv4 protocol document. The intent is to make our documents as clear as possible to implementors only interested in SNTP, since it is likely to enjoy (or suffer from...) the largest number of distinct, home-grown implementations. In either case, our new NTPv4 RFC(s) would then make RFC 2030 obsolete.

   2. Should Operation Requirements be included in our NTP Requirements document? One could argue this is BCP, but it also could have a major impact on the robustness of NTP as implemented, especially when utilizing public servers on the Internet.

   3. The requirements draft editor needs some contributed text and review especially for the Algorithm Requirements section.

2. Introduction

This document defines requirements for the Network Time Protocol (NTP) Version 4. NTP provides the mechanisms to synchronize time and coordinate time distribution amongst internet hosts. NTP Version 4 represents the latest improvements to NTP currently available and in use today. Earlier versions and portions of NTP have been specified by RFCs 1305 [1], 1769 [2], and 2030 [3].

Accurate and synchronized time is a requirement, or distinct advantage, for numerous applications. These applications include distributed databases, stock market operations, document timestamping, aviation traffic control, multimedia program synchronization and teleconferencing, network measurement and control,
and many forms of event logging.

NTP’s stated goals include:

Provide the best accuracy possible given network and server conditions.

Resist failures including malicious attacks and implementation bugs.

Be robust by utilizing Internet diversity and redundancy.

Automatically organize the subnet topology for best accuracy and reliability.

Perform host authentication, independent of non-NTP services.

Furthermore, ancillary issues such as access control and non-repudiation are considered goals as well.

The following requirements are prescribed or suggested by NTP applications, are direct consequences of NTP’s goals, or are expected for interoperability and end-user experience with the versions of NTP that are in widespread use today.

In this document, the words "must", "may", and "should" appear in lowercase since this is not a formal specification of the protocol. However, the use of these words here suggests that corresponding portions of the NTPv4 protocol specifications use these keywords in uppercase with the meanings defined by RFC 2119 [6].

3. Terminology

The following terms are used in this document:

host - an internet host that runs an implementation of NTP.

client - an NTP host that is the recipient of a disseminated time value.

server - an NTP host that is the source of a disseminated time value.

time - the value by which events are ordered in a given frame of reference. For NTP, the frame of reference is an epoch, and the time value is expressed in whole and fractional seconds since that epoch.
oscillator - a generator capable of a precise frequency (relative to the given frame of reference) to a specified tolerance.

clock - an oscillator together with a counter which records the (fractional) number of cycles since being initialized with a given value at a given time.

timescale - The NTP timescale is based on the UTC timescale, such that at the hour zero on 1 January 1972 (the beginning of the UTC era) the NTP clock was set to 2,272,060,800 (the number of seconds since hour zero on 1 January 1900).

epoch - the value of the counter at any given time. These are not continuous and depend on the precision of the counter.

calendar - a mapping from epoch in some frame of reference to the times and dates used in everyday life.

stability - a term used to classify the performance for clock oscillators, the systematic variation of frequency with time, synonymous with aging, drift, trends, etc.

jitter - a term used to classify the performance for clock oscillators, the short-term variations in frequency with components greater than 10 Hz.

wander - a term used to classify the performance for clock oscillators, the long-term variations in frequency with components less than 10 Hz.

stratum - the hierarchical layer at which an NTP host exists. The host(s) at the lowest layer (stratum 1) get their time value from a primary (non-NTP) time source and disseminate the time to hosts of the same or the next higher stratum.

subnet - the subset of network hosts that participate in a given NTP arrangement of servers and clients. Typically this arrangement is a hierarchical tree structure in which servers of the lowest strata are at the root and NTP servers of increasing strata branch toward the leaves of the tree, that are a set of NTP clients.

primary server - an NTP server host at stratum 1 that synchronizes to a non-NTP, typically national, time standard, such as by radio, satellite, or modem.

secondary server/client - an NTP host at stratum 2 or more that synchronizes to primary server(s) via a hierarchical subnet.
NTP modes - one of the modes in which an NTP host operates:

client/server mode - a unicast mode of operation in which an NTP server host disseminates a time value to an NTP client host. This mode has also been referred to as "master/slave".

symmetric mode - a mode of operation in which NTP hosts are equal peers, or servers of the same stratum.

multicast mode - a mode of operation in which NTP clients discover their NTP server(s) by receiving multicast advertisements from the available servers.

broadcast mode - a mode of intra-LAN operation in which NTP clients receive unsolicited broadcasts of the time value, typically from a single NTP server.

4. Algorithm Requirements

FIXME: consider common variable definitions which should be compile time or runtime configurable?: such as MAXSTRAT, MAXSKEW, MAXDISP, MINCLOCK, MAXCLOCK

FIXME: We need some help here from someone that knows the NTP reference implementation’s (ntpd) code. Which of the compile-time definitions (macros) are required to have the values defined in the implementation, as opposed to being configurable within a required range? We should also define the range required to be supported.

4.1 Clock Discipline

NTP implementations should include, at least, a clock discipline algorithm that utilizes a traditional linear phase-lock loop (PLL). Furthermore, NTP implementations may include a loop filter and variable frequency oscillator (VFO) that functions as a nonlinear, hybrid phase/frequency-lock (P/F) feedback loop to minimize jitter and wander and decrease time to converge as compared with a linear system only.

When available, NTP implementations should use host system-provided time adjustment routines so that clock readings are monotonically increasing such that no two successive clock readings could be the same.

4.2 Accuracy

Current NTP implementations and deployments generally have accuracies
of a few milliseconds in Local-Area Networks, and up to a few tens of milliseconds in global Wide-Area Networks. Given the best of implementation environments, worst-case error cannot exceed one-half the roundtrip delay measured by the client.

4.3 Jitter

FIXME

4.4 Wander

FIXME

5. Protocol Requirements

NTP server implementations must include support for unicast mode of client/server operation and symmetric mode so that a robust hierarchical subnet of NTP hosts can be constructed since this is NTP’s basis for reliability.

NTP server implementations may provide a multicast mode to serve multiple IP subnets in a network. It may also provide a broadcast mode in which unsolicited time values are disseminated to hosts on its LAN.

5.1 Configuration Requirements

Implementations must support the configuration of NTP servers using the Domain Name System. Multiple servers, e.g. up to six, should be able to be configured, since diverse network paths to multiple servers is the basis of NTP’s reliability.

5.1.1 Manual Configuration

FIXME

5.1.2 Automatic, Autonomous Configuration

FIXME: discuss autonomous configuration using multicast (for diversity and redundancy) with cryptographically secure source discovery.

Autonomously configured clients must periodically refresh their list of suitable servers.

5.1.3 Vendor Pre-configuration

5.1.4 Administrative Domains

FIXME

5.1.5 Key Distribution

FIXME

5.2 System Performance

FIXME

5.2.1 Scalability

FIXME: how many servers/peers can be configured? How many strata?

5.2.2 Client Performance Requirements

FIXME

5.2.3 Server Performance Requirements

FIXME

5.3 Security Requirements

Implementations must support the MD5-based symmetric key cryptography with preshared keys. This scheme is defined in RFC 1305 [1].

Implementations must support public key cryptography as defined by the so-called "Autokey" protocol, which is used to verify server identity. If employed, the implemetation must regenerate keys in a timely manner to resist compromise. FIXME: add details

5.4 Internet Protocol Version 6 Requirements

NTPv4 Requirements defined in this document apply without regard to whether the implementation runs atop IPv4 or IPv6, or both. So, an implementation that supports IPv4 must support all of its NTP modes and cryptographic features available using IPv6 whenever IPv6 is available on the underlying platform.

5.5 Robustness

FIXME
5.5.1 Authentication & Access Control

NTP has authentication requirements to protect the resulting system from faulty implementations, improper operation, and malicious attacks. These are important in a distributed protocol so that damage does not propagate throughout the NTP subnet.

NTP implementations must attempt to limit access to trusted peers. Trivially, this is first done by sanity checking NTP packet content to ignore duplicates and to timestamp packets as a one-time pad.

However, NTP implementations should also take measures to prevent unauthorized message-stream modification by using a crypto-checksum computed by the sender and checked by the receiver.

5.5.2 Client/Peer Rejection

NTP server implementations should include the ability to return a so-called "Kiss-o’-Death" (KoD) packet to a configured or discovered set of unwanted NTP clients. NTP clients, upon receiving the KoD packet, should cease communications with the given NTP server host that sent the packet, and instead rely on their other configured servers.

5.6 Longevity, Persistence

FIXME

5.6.1 Reconfiguration

FIXME: mention re-resolving DNS names here?


The Simple network Time Protocol (SNTP) is a slight variation of NTP in which the clients simply receive periodic time values to update their local clocks. Today, SNTP is the most common use of the NTP infrastructure. Also, SNTP is a small subset of the overall NTP functionality, so it has many unique client implementations. This plurality and ubiquity of SNTP clients warrants special attention as we define requirements for implementations.


An SNTP client should respect the KoD access-control mechanism.
7. Operational Requirements

FIXME: Do operational requirements belong here or in a separate
document? E.g. stratum 1 servers should be synchronized to a non-NTP
time standard, stratum 2 servers must synchronize to primary servers
in the NTP hierarchy.

7.1 Client Poll Interval

An SNTP client must not use a poll interval less than one minute.

An SNTP client should increase the poll interval using exponential
backoff if ever the server does not respond and also as its required
clock performance permits.

An SNTP client should randomize its initial inter-query timeout.

8. Security Considerations

A reliable network time service, such as NTP means to be, requires
provisions to prevent accidental or malicious attacks on its servers
and clients. Furthermore, reliability requires that NTP clients can
verify the authenticity of NTP messages it receives.

NTP implementations, whose requirements are described above, address
security threats in a number of ways:

The hosts in an NTP subnet should be able to be configured to
cryptographically authenticate servers using shared secret keys.
This is appropriate for hand-configured, engineered subnet
hierarchies amongst a relatively small set of trusted NTP hosts.

A specially crafted, NTP-specific public-key cryptography method
should be employed to simplify the authentication of servers by
hosts which are part of a potential large, possibly automatically
configured, NTP subnet.

The potentially large number and redundancy of NTP hosts and paths
amongst them, within an NTP subnet, mitigates some security
threats to the overall system. NTP takes advantage of this scale
by employing its algorithms to reject poorly performing, possibly
compromised, NTP servers to create an overall robust, reliable time
synchronization and dissemination system.

9. IANA Considerations

This document creates no new requirements on IANA namespaces.
10. Acknowledgements

Most of the NTP information used as background for this document was
drawn from David L. Mills' NTP documents, linked from [7] and [8].

11. References

11.1 Normative References


March 1995.


11.2 Informative References

<http://net.doit.wisc.edu/~plonka/ntp-reqs/>.

[6] Bradner, S., "Key words for use in RFCs to Indicate Requirement


<http://www.eecis.udel.edu/~mills/>.

Author’s Address

David Plonka
University of Wisconsin - Madison

Email: plonka@doit.wisc.edu
URI: http://net.doit.wisc.edu/~plonka/
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.