Requirements for Network Time Protocol Version 4
draft-ietf-ntp-reqs-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 27, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

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.

Editorial Notes

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 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.

NTP Requirements Open Issues

1. Is the leap seconds issue sufficiently addressed in the new Leap Seconds sub-section? Is it an issue that the (non-standard) AutoKey protocol was used to distribute the leapseconds table?

2. Amongst the NTP WG documents, where will the official terminology (definitions) be? Presumably there’s more content to take from the NTP community’s documents.

3. What’s "iburst" and how does it affect the polling interval requirements?

Revision History

The following is the recent revision history of this document.

$Log: draft-ietf-ntp-reqs.xml,v $  
Revision 1.6  2005/10/24 04:56:11  plonka  
Moved the open issues and revision history to an editorial notes preface to fix section numbers.

Mentioned SNTP clients when defining client in the terminology section.

Did the following edits as suggested by David Mills on the mailing list:

* in the definition of broadcast in the terminology section, mentioned that NTP’s broadcast mode utilizes multicast in IPv6.

* In the algorithm requirements section, removed mention of various macros such as MAXSTRAT, MAXSKew, etc. Introduced MAXPOLL, since that has clear requirements.

* reorganized and added text to the algorithm requirements section, including introducing the various algorithm sub-sections.
* clarified that SNTP clients are divorced from a number of algorithm requirements that "full" NTP hosts must use.

* Changed the accuracy section to note that NTP is best effort, but added some text about service expectations.

* added to the definitions of jitter and wander in the terminology section.

* with regard to configuring multiple servers, changed the text to say that 4 should be configurable.

* mentioned that implementations should signal pending leap-seconds to the OS.

* added some text regarding reconfiguration when NTP server(s) are unreachable.

As suggested by Danny Mayer on the mailing list, removed the reference to "master/slave" mode.

Added an item about leap seconds based on David Mill’s input on the mailing list.

Revision 1.5  2005/10/21 20:46:22  plonka
Resolved the SNTP open issue by devoting a section of this draft to SNTP.

Resolved the Operational Requirements open issue by temporarily devoting a section of this draft to Operational Requirements to subsequently be moved to a BCP draft.

Added the Management Information Requirements section. This was based on discussion noted in the minutes from the WG meeting at IETF63.

Revision 1.4  2005/10/21 19:37:54  plonka
fixed a typo: s/recipent/recipient/

removed reference to the AutoKey protocol and mentioned that implementations must use an IETF standard method to verify server identity, and should use a corresponding standard key distribution protocol. This is based on discussion noted in the minutes from the WG meeting at IETF63.

added the Revision History

upped the revision number in prep for next submission of this draft
Table of Contents

1. Introduction ........................................ 5
2. Terminology .......................................... 5
3. Algorithm Requirements .............................. 7
   3.1 Poll Algorithm .................................. 8
   3.2 Clock Discipline Algorithm ....................... 8
   3.3 Clock Filter Algorithm ........................ 9
   3.4 Clock Selection Algorithm ....................... 9
   3.5 Clustering Algorithm ............................ 9
   3.6 Combining Algorithm ............................. 9
   3.7 Accuracy ......................................... 9
   3.7.1 Leap Seconds ................................ 9
4. Protocol Requirements ............................... 9
   4.1 Configuration Requirements ...................... 10
      4.1.1 Manual Configuration ....................... 10
      4.1.2 Automatic, Autonomous Configuration ....... 10
      4.1.3 Vendor Pre-configuration ................... 10
      4.1.4 Administrative Domains ..................... 10
      4.1.5 Key Distribution ............................ 10
   4.2 System Performance .............................. 10
      4.2.1 Scalability ................................ 10
      4.2.2 Client Performance Requirements .......... 11
      4.2.3 Server Performance Requirements .......... 11
   4.3 Security Requirements .......................... 11
   4.4 Internet Protocol Version 6 Requirements ....... 11
5. Simple Network Time Protocol Requirements ....... 12
   5.1 SNTP Client Poll Interval ....................... 12
6. Management Information Requirements .............. 13
7. Operational Requirements .......................... 13
8. Security Considerations ............................ 13
9. IANA Considerations ................................ 14
10. Acknowledgements .................................. 14
11. References .......................................... 14
   11.1 Normative References ........................... 14
   11.2 Informative References ........................ 14
    Author’s Address .................................. 15
    Intellectual Property and Copyright Statements .. 16
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].

2. 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. A subset of clients are Simple Network Time Protocol (SNTP) clients.

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. As defined in NTP, jitter is the exponential average of the RMS differences between samples in a sliding window of time measurements.

wander - a term used to classify the performance for clock oscillators, the long-term variations in frequency with components less than 10 Hz. As defined in NTP, wander is the exponential average of the RMS differences between samples in a sliding window of frequency measurements.
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.

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. The details of the broadcast paradigm differ based upon the Internet address family. For IPv4, the broadcast mechanism is used. For IPv6, its multicast mechanism is used instead, even though this is still referred to as NTP’s broadcast mode.

3. Algorithm Requirements

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. MINPOLL is one example.
NTP hosts that are not merely SNTP clients are required to run the NTP algorithms. When using only one source, these include clock filter and clock discipline algorithms. When multiple sources are used, additional selection and clustering algorithms are required and a combining algorithm should be used. Since SNTP clients do not operate in symmetric modes, their requirements are somewhat relaxed. Any SNTP client that does not meet these NTP algorithm requirements cannot function as an NTP server.

3.1 Poll Algorithm

The NTP poll adjust algorithm is designed to protect the network. As such it is required of all NTP hosts, including SNTP clients. It backs off the poll rate both when the system clock converges within tolerance and when the source becomes unreachable.

An NTP client should randomize its initial inter-query timeout, and other intervals over a narrow range.

FIXME: add details

3.2 Clock Discipline Algorithm

Note that SNTP clients are not required to discipline their system clock. The following clock discipline requirements apply to all other NTP hosts.

If timestamps are determined directly by an attached reference clock, say by a shared register array with oscillator phase locked to a GPS receiver, then a clock discipline algorithm is not required.

Otherwise, such as when an implementation determines timestamps from an ordinary (uncompensated) quartz oscillator performing a time of day function, the implementation must discipline the clock in both time and frequency. This generally requires a second-order 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.
3.3 Clock Filter Algorithm

The clock filter algorithm is required of all NTP servers. FIXME

3.4 Clock Selection Algorithm

The clock selection algorithm applies to NTP hosts utilizing more than one source. FIXME

3.5 Clustering Algorithm

The clustering algorithm applies to NTP hosts utilizing more than one source. FIXME

3.6 Combining Algorithm

The combining algorithm is recommended of NTP hosts utilizing more than one source. FIXME

3.7 Accuracy

While NTP service is best effort with no guarantees, 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. As such, this sets the service expectation.

The absolute error relative to the time on the server (which could be in error itself) is not greater than half the roundtrip delay plus dispersion plus. It is correct to say the worst case error is bounded by the synchronization distance to the primary source. While these observations do not lead to strict requirements, it does, again, indicate the service expectation.

3.7.1 Leap Seconds

Implementations should signal the operating system of a pending leap second event on, but not before, the day of the leap second event. Ordinarily, this takes the form of an operating system function call.

4. 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.

Note that an SNTP client cannot operate in symmetric mode.
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.

4.1 Configuration Requirements

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

4.1.1 Manual Configuration

Implementations should be able to configure the minimum poll interval.

FIXME: what else?

4.1.2 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.

4.1.3 Vendor Pre-configuration


4.1.4 Administrative Domains

FIXME

4.1.5 Key Distribution

FIXME

4.2 System Performance

FIXME

4.2.1 Scalability

FIXME: how many servers/peers can be configured? How many strata?
4.2.2 Client Performance Requirements

FIXME

4.2.3 Server Performance Requirements

FIXME

4.3 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 the use of an IETF standard public key cryptography scheme to verify server identity. An accompanying IETF standard key distribution protocol should be supported.

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

4.4 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.

4.5 Robustness

FIXME

4.5.1 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.
4.5.2 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.

4.6 Longevity, Persistence

FIXME

4.6.1 Reconfiguration

NTP clients must attempt to reconfigure when they discover that their server is unreachable. This potentially involves, but is not necessarily limited to: performing a DHCP query to discover an NTP server, resolve the server DNS names, and restart the security protocol.

5. Simple Network Time Protocol Requirements

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 may use any means available to set its clock based on the received NTP packet. That is, an SNTP client is not required to conform to the same rules that an NTP client must regarding the NTP algorithms.

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

5.1 SNTP 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.

6. Management Information Requirements

Implementations may make management information available to remote managers. If provided, such implementations must use a IETF standard protocol. FIXME: If we elaborate on the management information base itself, consider what is made available using the ntpdc utility. This is the "special NTP query program" used to query the ntpd daemon.

7. Operational Requirements

Operational requirements identified during the preparation of this document may be collected here in anticipation of subsequently moving them to a BCP draft.

E.g. stratum 1 servers should be synchronized to a non-NTP time standard, stratum 2 servers must synchronized to primary servers in the NTP hierarchy.

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 authentication 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]. Danny Mayer, Brian Haberman, and David Mills provided useful comments or contributed text for this document.

11. References

11.1 Normative References


11.2 Informative References


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.