Network Working Group
Request for Comments: XXXX
Obsoletes: 2030, 1769
Category: Informational

D. Mills, University of Delaware
J. Montgomery, Netgear

August 2003

Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This memorandum describes the Simple Network Time Protocol (SNTP) Version 4, which is a subset of the Network Time Protocol (NTP) used to synchronize computer clocks in the Internet. SNTP can be used when the ultimate performance of the full NTP implementation described in RFC-1305 is not needed or justified. When operating with current and previous NTP and SNTP versions, SNTP Version 4 involves no changes to the NTP specification or known implementations, but rather a clarification of certain design features of NTP which allow operation in a simple, stateless remote-procedure call (RPC) mode with accuracy and reliability expectations similar to the UDP/TIME protocol described in RFC-868.

The only significant protocol change in SNTP Version 4 over previous versions of NTP and SNTP is a modified header interpretation to accommodate Internet Protocol Version 6 (IPv6) [DEE96] and OSI [COL94] addressing. However, SNTP Version 4 includes certain optional extensions to the basic NTP Version 3 model, including an anycast mode and a public-key based authentication scheme designed specifically for broadcast and anycast applications. While the anycast mode extension is described in this memo, the authentication scheme extension is described in another RFC submitted to the IETF. Until such time that a definitive specification is published, these extensions should be considered provisional. In addition, this memo introduces the kiss-o'-death message, which can be used by servers to suppress client requests as circumstances require.

This memorandum obsoletes RFC-1769, which describes SNTP Version 3, and RFC-2030, which describes SNTP Version 4. Its purpose is to correct certain inconsistencies in the previous documents and to clarify header formats and protocol operations for NTP Version 3 (IPv4) and NTP Version 4 (IPv4, IPv6 and OSI), which are also used for SNTP. A further purpose is to provide guidance for home and business client implementations in routers and other consumer devices to protect the server population from abuse. A working knowledge of the NTP Version 3 specification RFC-1305 is not required for an implementation of SNTP.

1. Introduction

The Network Time Protocol (NTP) Version 3 specified in RFC-1305 [MIL92] is widely used to synchronize computer clocks in the global Internet. It provides comprehensive mechanisms to access national time and frequency dissemination services, organize the time-synchronization subnet and adjust the local clock in each participating subnet peer. In most places of the Internet of today, NTP provides accuracies of 1-50 ms, depending on the characteristics of the synchronization source and network paths.

RFC-1305 specifies the NTP Version 3 protocol machine in terms of events, states, transition functions and actions and, in addition, engineered algorithms to improve the timekeeping quality and mitigate among several synchronization sources, some of which may be faulty. To achieve accuracies in the low milliseconds over paths spanning major portions of the Internet of today, these intricate algorithms, or their functional equivalents, are necessary. However, in many cases accuracies in the order of significant fractions of a second are acceptable. In simple home router applications accuracies of up to a minute may suffice. In such cases, simpler protocols such as the Time Protocol [POS83], have been used for this purpose. These protocols usually involve an RPC exchange where the client requests the time of day and the server returns it in seconds past some known reference epoch.

NTP is designed for use by clients and servers with a wide range of capabilities and over a wide range of network jitter and oscillator wander characteristics. Most users of NTP in the Internet of today use a software distribution available from www.ntp.org. The distribution, which includes the full suite of NTP options, mitigation algorithms and security schemes, is a relatively complex, real-time application. While the software has been ported to a wide variety of hardware platforms ranging from personal computers to supercomputers, its sheer size and complexity is not appropriate for many applications. Accordingly, it is useful to explore alternative strategies using simpler software appropriate for less stringent accuracy expectations.

This memo describes the Simple Network Time Protocol (SNTP) Version 4, which is a simplified access paradigm for servers and clients using NTP Version 4, as well as previous versions. The access paradigm is identical to the UDP/TIME Protocol and, in fact, it should be easily possible to adapt a UDP/TIME client implementation, say for a personal computer, to operate using SNTP. Moreover, SNTP is also designed to operate in a dedicated server configuration including an integrated radio clock. With careful design and control of the various latencies in the system, which is practical in a dedicated design, it is possible to deliver time accurate to the order of microseconds.

When operating with current and previous versions of NTP and SNTP, SNTP Version 4 requires no changes to the protocol or implementations now running or likely to be implemented specifically for NTP or SNTP Version 4. The NTP and SNTP packet formats are the same and the arithmetic operations to calculate the client time. clock offset and roundtrip delay are the same. To a NTP or SNTP server, NTP and SNTP clients are indistinguishable; to a NTP or SNTP client, NTP and SNTP servers are indistinguishable. Like

NTP servers operating in non-symmetric modes, SNTP servers are stateless and can support large numbers of clients; however, unlike most NTP clients, SNTP clients normally operate with only a single server at a time.

It is strongly recommended that SNTP clients be used only at the extremities of the synchronization subnet. SNTP clients should operate only at the leaves (highest stratum) of the subnet and in configurations where no NTP or SNTP client is dependent on another SNTP client for synchronization. SNTP servers should operate only at the root (stratum 1) of the subnet and then only in configurations where no other source of synchronization other than a reliable radio clock or telephone modem is available. The full degree of reliability ordinarily expected of primary servers is possible only using the redundant sources, diverse subnet paths and crafted algorithms of a full NTP implementation. This extends to the primary source of synchronization itself in the form of multiple radio or modem sources and backup paths to other primary servers should all sources fail or the majority deliver incorrect time. Therefore, the use of SNTP rather than NTP in primary servers should be carefully considered.

An important provision in this memo is the interpretation of certain NTP header fields which provide for IPv6 and OSI addressing. The only significant difference between the NTP Version 3 and NTP Version 4 header format is the four-octet Reference Identifier field, which is used primarily to detect and avoid synchronization loops. In Version 3 and Version 4 primary (stratum-1) servers, this field contains the four-character ASCII reference clock identifier defined later in this memo. In IPv4 secondary servers and clients, it contains the 32-bit address of the synchronization source. In IPv6 secondary servers and clients, it contains the first 32 bits of the MD5 hash of the 128-bit IPv6 address of the synchronization source.

In the case of OSI, the Connectionless Transport Service (CLTS) is used [ISO86]. Each SNTP packet is transmitted as the TS-Userdata parameter of a T-UNITDATA Request primitive. Alternately, the header can be encapsulated in a TPDU which itself is transported using UDP [DOB91]. It is not advised that NTP be operated at the upper layers of the OSI stack, such as might be inferred from [FUR94], as this could seriously degrade accuracy. With the header formats defined in this memo, it is in principle possible to interwork between servers and clients of one protocol family and another, although the practical difficulties may make this inadvisable. For the OSI protocol variant, the Reference Identifier field contains the first 32 bits of the MD5 hash of the NSAP address of the synchronization source.

In the following, indented paragraphs such as this one contain information not required by the formal protocol specification, but considered good practice in protocol implementations.

2. Operating Modes and Addressing

Unless excepted in context, reference to broadcast address means IPv4 broadcast address, IPv4 multicast group address or IPv6 site-local scope address. Further information on the broadcast/multicast model is in [DEE89]. Details of address format, scoping rules, etc., are beyond the scope of this memo. SNTP Version 4 can operate with either unicast (point

to point), broadcast (point to multipoint) or anycast (multipoint to point) addressing modes. A unicast client sends a request to a designated server at its unicast address and expects a reply from which it can determine the time and, optionally, the roundtrip delay and clock offset relative to the server. A broadcast server periodically sends an unsolicited message to a designated broadcast address. A broadcast client listens on this address and ordinarily sends no requests.

Anycast is designed for use with a set of cooperating servers whose addresses are not known beforehand. The anycast client sends an ordinary NTP client request to a designated broadcast address. One or more anycast servers listen on that address. Upon receiving a request, an anycast server sends an ordinary NTP server reply to the client. The client then binds to the server from which the first such message was received and continues operation with unicast addresses. Subsequent replies from other anycast servers are ignored.

Broadcast servers should respond to client unicast requests, as well as send unsolicited broadcast messages. Broadcast clients may send unicast requests in order to determine the network propagation delay between the server and client and then continue operation in listen-only mode.

The client and server addresses are assigned following the usual IPv4, IPv6 or OSI conventions. For NTP multicast, the IANA has reserved the IPv4 group address 224.0.1.1 and the IPv6 group address ending :101, with prefix determined by scoping rules. The NTP broadcast address for OSI has yet to be determined. Notwithstanding the IANA reserved addresses, other multicast addresses can be used which do not conflict with others assigned in scope. In the case of IPv4 multicast or IPv6 broadcast addresses, the client must implement the Internet Group Management Protocol (IGMP) [CAIN02], in order that the local router joins the multicast group and relays messages to the IPv4 or IPv6 multicast group. The scoping, routing and group membership procedures are determined by considerations beyond the scope of this memo.

It is important to adjust the time-to-live (TTL) field in the IP header of multicast messages to a reasonable value in order to limit the network resources used by this (and any other) multicast service. Only multicast clients in scope will receive multicast server messages. Only cooperating anycast servers in scope will reply to a client request. The engineering principles which determine the proper values to be used are beyond the scope of this memo.

In the case of SNTP as specified herein, there is a very real vulnerability that SNTP broadcast clients can be disrupted by misbehaving or hostile SNTP or NTP broadcast servers elsewhere in the Internet. It is strongly recommended that access controls and/or cryptographic authentication means be provided for additional security in such cases.

While not integral to the SNTP specification, it is intended that IP broadcast addresses will be used primarily in IP subnets and LAN segments including a fully functional NTP server with a number of dependent SNTP broadcast clients on the same subnet, while IP multicast group addresses will be used only in cases where the TTL is engineered specifically for each service domain.

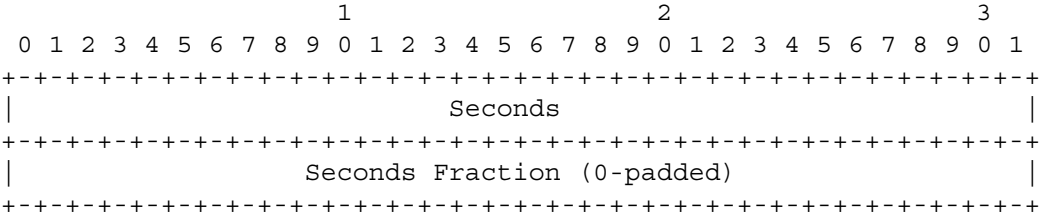
3. NTP Timestamp Format

SNTP uses the standard NTP timestamp format described in RFC-1305 and previous versions of that document. In conformance with standard Internet practice, NTP data are specified as integer or fixed-point quantities, with bits numbered in big-endian fashion from 0 starting at the left or most significant end. Unless specified otherwise, all quantities are unsigned and may occupy the full field width with an implied 0 preceding bit 0.

Since NTP timestamps are cherished data and, in fact, represent the main product of the protocol, a special timestamp format has been established. NTP timestamps are represented as a 64-bit unsigned fixed-point number, in seconds relative to 0h on 1 January 1900. The integer part is in the first 32 bits and the fraction part in the last 32 bits. In the fraction part, the non-significant low order can be set to 0.

It is advisable to fill the non-significant low order bits of the timestamp with a random, unbiased bitstring, both to avoid systematic roundoff errors and as a means of loop detection and replay detection (see below). It is important that the bitstring be unpredictable by a intruder. One way of doing this is to generate a random 128-bit bitstring at startup. After that, Each time the system clock is read the string consisting of the timestamp and bitstring is hashed with the MD5 algorithm, then the non-significant bits of the timestamp are copied from the result.

This format allows convenient multiple-precision arithmetic and conversion to UDP/TIME representation (seconds), but does complicate the conversion to ICMP Timestamp message representation, which is in milliseconds. The maximum number that can be represented is 4,294,967,295 seconds with a precision of about 200 picoseconds, which should be adequate for even the most exotic requirements.



Note that, since some time in 1968 (second 2,147,483,648) 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 (second 4,294,967,296). 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.

As the NTP timestamp format has been in use for the last 35 years, it remains a possibility that it will be in use 33 years from now when the seconds field overflows. As it is probably inappropriate to archive NTP timestamps before bit 0 was set in 1968, a convenient way to extend the useful life of NTP timestamps is the following convention: If bit 0 is set, the UTC time is in the range 1968-2036 and UTC time is reckoned from 0h 0m 0s UTC on 1 January 1900. If bit 0 is not set, the time is in the range

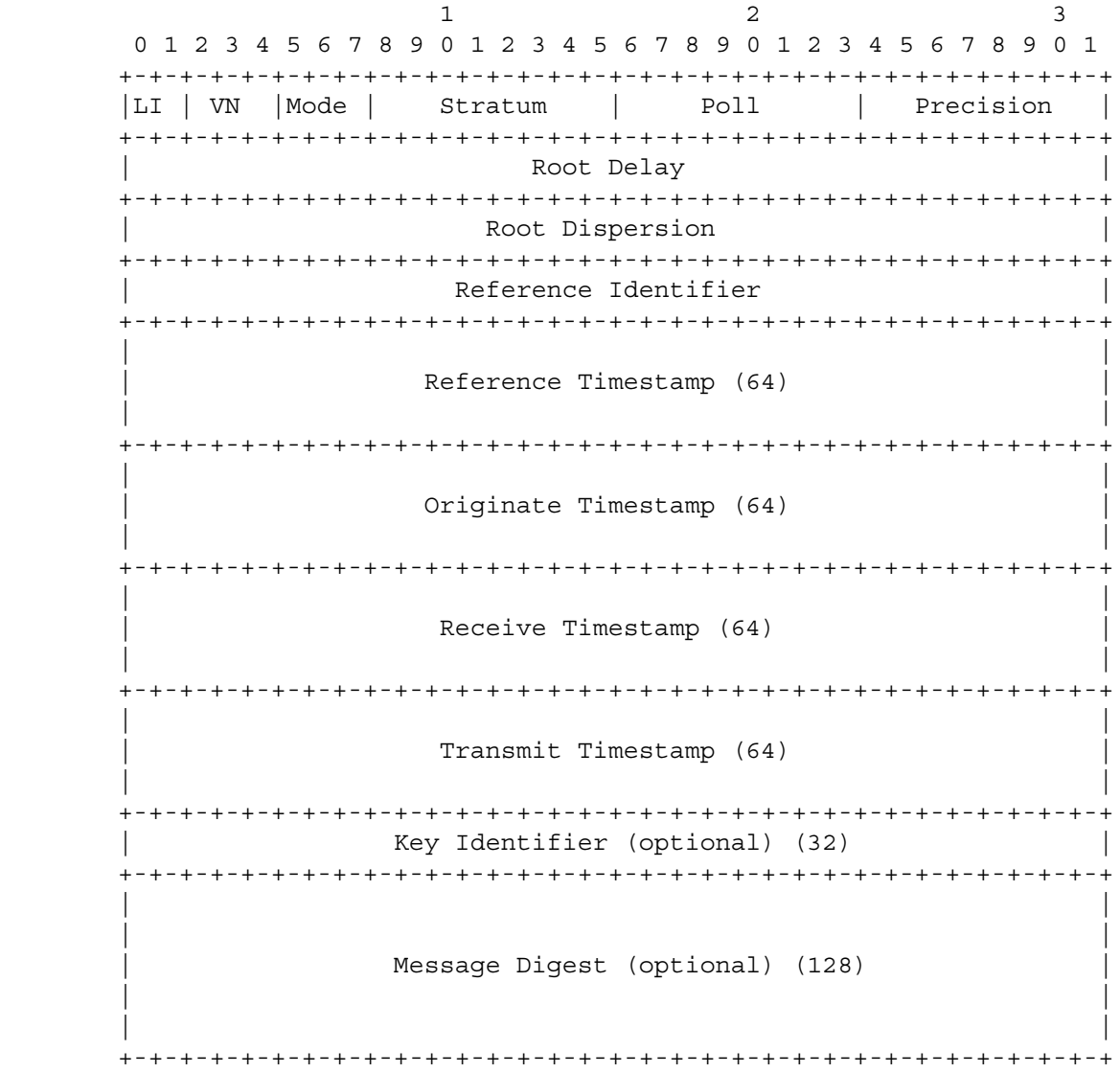


Figure 1. NTP Packet Header

2036-2104 and UTC time is reckoned from 6h 28m 16s UTC on 7 February 2036. Note that when calculating the correspondence, 2000 is a leap year and leap seconds are not included in the reckoning.

4. NTP Message Format

Both NTP and SNTP are clients of the User Datagram Protocol (UDP) [POS80], which itself is a client of the Internet Protocol (IP) [DAR81]. The structure of the IP and UDP headers is described in the cited specification documents and will not be detailed further here. The UDP port number assigned to NTP is 123, which should be used in the Destination Port field in the UDP header. The Source Port field can be any nonzero value chosen for identification or multiplexing purposes.

Figure 1 is a description of the NTP/SNTP Version 4 message format, which follows the IP and UDP headers. This format is identical to that described

in RFC-1305, with the exception of the reference identifier field. In SNTP most of these fields are initialized with pre-specified data. For completeness, the function of each field is briefly summarized below.

Leap Indicator (LI): This is a two-bit code warning of an impending leap second to be inserted/deleted in the last minute of the current day, with bit 0 and bit 1, respectively, coded as follows:

LI	Value	Meaning
00	0	no warning
01	1	last minute has 61 seconds
10	2	last minute has 59 seconds)
11	3	alarm condition (clock not synchronized)

SNTP servers ignore this field in client requests. SNTP clients interpret this field in server replies as above. At startup, SNTP servers set this field to 11 (clock not synchronized) and set this field to some other value when synchronized to the primary reference clock. Once set to other than 11, the field is never set to that value again, even if the primary reference clock becomes unreachable or defective.

Version Number (VN): This is a three-bit integer indicating the NTP/SNTP version number, currently 4. If necessary to distinguish between IPv4, IPv6 and OSI, the encapsulating context must be inspected. The values are defined as follows:

Mode	Meaning
0	reserved
1	symmetric active
2	symmetric passive
3	client
4	server
5	broadcast
6	reserved for NTP control message
7	reserved for private use

In unicast and anycast modes, the SNTP client sets this field to 3 (client) in the request and the server sets it to 4 (server) in the reply. In broadcast mode, the server sets this field to 5 (broadcast). The other modes are not used by SNTP servers and clients.

Stratum: This is a eight-bit unsigned integer indicating the stratum of the server or client, with values defined as follows:

Stratum	Meaning
0	kiss-o'-death message (see below)
1	primary reference (e.g., synchronized by radio clock)
2-15	secondary reference (synchronized by NTP or SNTP)
16-255	reserved

Poll Interval: This is an eight-bit unsigned integer used as an exponent of two, where the resulting value is the maximum interval between successive messages in seconds. For NTP servers and clients, the values that can appear in this field range from 4 (16 s) to 17 (131,072 s - about 36 h); however, most NTP servers and clients use only the default range from 6 (64 s) to 10 (1024 s). The poll interval is not used for SNTP servers and clients and ordinarily the value is set to zero. Optionally for consistency and possible error checking, if the value is nonzero, it should be set to the nearest exponent of two for the retry interval.

Precision: This is an eight-bit signed integer used as an exponent of two, where the resulting value is the precision of the local clock in seconds. The values that normally appear in this field range from -6 for mains-frequency clocks to -20 for microsecond clocks found in some workstations.

Root Delay: This is a 32-bit signed fixed-point number indicating the total roundtrip delay to the primary reference source, in seconds with fraction point between bits 15 and 16. Note that this variable can take on both positive and negative values, depending on the relative time and frequency offsets. The values that normally appear in this field range from negative values of a few milliseconds to positive values of several hundred milliseconds.

Root Dispersion: This is a 32-bit unsigned fixed-point number indicating the nominal error relative to the primary reference source, in seconds with fraction point between bits 15 and 16. The values that normally appear in this field range from zero to several hundred milliseconds.

Reference Identifier: This is a 32-bit bitstring identifying the particular reference source. In the case of NTP Version 4, stratum 0 (kiss-o'-death message) and 1 (primary server), this is a four-character ASCII string, left justified and zero padded to 32 bits. In IPv4 secondary servers, this is the 32-bit IPv4 address of the synchronization source. In IPv6 secondary servers, this is the first 32 bits of the MD5 hash of the 128-bit IPv6 address of the synchronization source. In OSI secondary servers, this is the MD5 hash of the NSAP address of the synchronization source.

NTP and SNTP primary (stratum 1) servers should set this field to a code identifying the external reference source according to Figure 2. If the external reference is one of those listed, the associated code should be used. Codes for sources not listed can be contrived as appropriate.

In NTP Version 3 this field was often used to walk-back the synchronization subnet to the root (primary server) for management purposes. In NTP Version 4 with IPv6 or OSI, this feature is not available, since the addresses are longer than 32 bits and only a hash is available. However, a walk-back can be accomplished using the NTP control message and the reference identifier field described in the NTP Version 3 specification [MIL92].

Reference Timestamp: This is the time at which the local clock was last set or corrected, in 64-bit timestamp format.

Code	External Reference Source
LOCL	uncalibrated local clock
CESM	calibrated Cesium clock
RBDM	calibrated Rubidium clock
PPS	calibrated quartz clock or other pulse-per-second source
IRIG	Inter-Range Instrumentation Group
ACTS	NIST telephone modem service
USNO	USNO telephone modem service
PTB	PTB (Germany) telephone modem service
TDF	Allouis (France) Radio 164 kHz
DCF	Mainflingen (Germany) Radio 77.5 kHz
MSF	Rugby (UK) Radio 60 kHz
WWV	Ft. Collins (US) Radio 2.5, 5, 10, 15, 20 MHz
WWVB	Boulder (US) Radio 60 kHz
WWVH	Kaui Hawaii (US) Radio 2.5, 5, 10, 15 MHz
CHU	Ottawa (Canada) Radio 3330, 7335, 14670 kHz
LORC	LORAN-C radionavigation system
OMEG	OMEGA radionavigation system
GPS	Global Positioning Service
GOES	Geostationary Orbit Environment Satellite

Figure 2. Reference Identifier Codes

Originate Timestamp: This is the time at which the request departed the client for the server, in 64-bit timestamp format.

Receive Timestamp: This is the time at which the request arrived at the server, in 64-bit timestamp format.

Transmit Timestamp: This is the time at which the reply departed the server for the client, in 64-bit timestamp format.

Authenticator (optional): When the NTP authentication scheme is implemented, the Key Identifier and Message Digest fields contain the message authentication code (MAC) information defined in Appendix C of RFC-1305.

5. SNTP Client Operations

A SNTP client can operate in unicast, broadcast or anycast modes. In unicast mode, the client sends a request (NTP mode 3) to a designated unicast server and expects a reply (NTP mode 4) from that server. In broadcast client mode, it sends no request and waits for a broadcast (NTP mode 5) from one or more broadcast servers. In anycast mode, the client sends a request (NTP mode 3) to a designated broadcast address and expects a reply (NTP mode 4) from one or more anycast servers. The client uses the first reply received to establish the particular server for subsequent unicast operations. Later replies from this server (duplicates) or any other server are ignored. Other than the selection of address in the request, the operations of anycast and unicast clients are identical. Requests are normally sent at intervals depending on the frequency tolerance of the client clock and the required accuracy.

A unicast or anycast client initializes the NTP message header, sends the request to the server and strips the time of day from the Transmit Timestamp field of the reply. For this purpose, all of the NTP header fields shown above can be set to 0, except the Mode, which is set to 3 (client), VN and optional Transmit Timestamp fields. The VN field can be set to any version number supported by the NTP/SNTP server. By rule, NTP servers for a particular version support previous versions as well, so a prudent SNTP client can specify the earliest acceptable version on the expectation that any server of that or later versions will respond. Version 3 (RFC-1305) and Version 2 (RFC-1119) servers already accept all previous versions, including Version 1 (RFC-1059). Note that Version 0 (RFC-959) is no longer supported by any other version.

Since there will probably continue to be NTP and SNTP servers of all four versions interoperating in the Internet, careful consideration should be given to the VN field. It is recommended that clients use the latest version known to be supported by the selected server in the interest of the highest accuracy and reliability. SNTP Version 4 clients can interoperate with all previous version NTP and SNTP servers. Each version NTP server replies with the same version as the request, so the VN field of the request also specifies the VN field of the reply.

While not necessary in a conforming client implementation, in unicast and anycast modes it highly recommended that the Transmit Timestamp field in the request is set to the time of day according to the client clock in NTP timestamp format. This allows a simple calculation to determine the propagation delay between the server and client and to align the local clock generally within a few tens of milliseconds relative to the server. In addition, this provides a simple method to verify that the server reply is in fact a legitimate response to the specific client request and avoid replays. In broadcast mode, the client has no information to calculate the propagation delay or determine the validity of the server, unless one of the NTP authentication schemes is used.

To calculate the roundtrip delay d and local clock offset t relative to the server, the client sets the Transmit Timestamp field in the request to the time of day according to the client clock in NTP timestamp format. For this purpose the clock need not be synchronized. The server copies this field to the originate timestamp in the reply and sets the Receive Timestamp and Transmit Timestamp fields to the time of day according to the server clock in NTP timestamp format.

When the server reply is received, the client determines a Destination Timestamp variable as the time of arrival according to its clock in NTP timestamp format. The following table summarizes the four timestamps.

Timestamp Name	ID	When Generated
Originate Timestamp	T1	time request sent by client
Receive Timestamp	T2	time request received by server
Transmit Timestamp	T3	time reply sent by server
Destination Timestamp	T4	time reply received by client

The roundtrip delay d and local clock offset t are defined as

d = (T4 - T1) - (T3 - T2) t = ((T2 - T1) + (T3 - T4)) / 2.

Note that both delay and offset are signed quantities and can in general be less than zero; however, a delay less than zero is possible only in symmetric modes, which SNTP clients are forbidden to use. The following table summarizes the required SNTP client operations in unicast, anycast and broadcast modes. The recommended error checks are shown in the Reply and Broadcast columns in the table. The message should be considered valid only if all the fields shown contain values in the respective ranges. Whether to believe the message if one or more of the fields marked "ignore" contain invalid values is at the discretion of the implementation.

Field Name	Unicast/Anycast		Broadcast
	Request	Reply	
LI	0	0-2	0-2
VN	1-4	copied from request	1-4
Mode	3	4	5
Stratum	0	1-15	1-15
Poll	0	ignore	ignore
Precision	0	ignore	ignore
Root Delay	0	ignore	ignore
Root Dispersion	0	ignore	ignore
Reference Identifier	0	ignore	ignore
Reference Timestamp	0	ignore	ignore
Originate Timestamp	0	(see text)	ignore
Receive Timestamp	0	(see text)	ignore
Transmit Timestamp	(see text)	nonzero	nonzero
Authenticator	optional	optional	optional

While not required in a conforming SNTP client implementation, it is wise to consider a suite of sanity checks designed to avoid various kinds of abuse that might happen as the result of server implementation errors (Windows comes to mind) or malicious attack. Following is a list of suggested checks.

1. The UDP Destination Port number in the server reply should match the Source Port number used in the request.

2. The Originate Timestamp in the server reply should match the Transmit Timestamp used in the request.

3. The server reply should be discarded if either Transmit Timestamp field is zero or the Stratum is zero or the Mode is not 4 (unicast) or 5 (broadcast).

4. A truly paranoid client can check the Root Delay and Root Dispersion fields are each greater than or equal to zero and less than infinity, where infinity is currently a cozy number like 16 seconds.

6. SNTP Server Operations

A SNTP Version 4 server operating with either a NTP or SNTP client of the same or previous versions retains no persistent state. Since a SNTP server ordinarily does not implement the full set of NTP algorithms intended to support redundant peers and diverse network paths, a SNTP server should be operated only in conjunction with a source of external synchronization, such as a reliable radio clock or telephone modem. In this case it operates as a primary (stratum 1) server.

A SNTP server can operate with any unicast, anycast or broadcast address or any combination of these addresses. A unicast or anycast server receives a request (NTP mode 3), modifies certain fields in the NTP header, and sends a reply (NTP mode 4), possibly using the same message buffer as the request. A anycast server listens on the designated broadcast address, but uses its own unicast address in the source address field of the reply. Other than the selection of address in the reply, the operations of anycast and unicast servers are identical. Broadcast messages are normally sent at poll intervals from 64 s to 1024 s, depending on the expected frequency tolerance of the client clocks and the required accuracy.

Unicast and anycast servers copy the VN and Poll fields of the request intact to the reply and set the Stratum field to 1.

Note that SNTP servers normally operate as primary (stratum 1) servers. While operating at higher strata (up to 15) and at the same time synchronizing to an external source such as a GPS receiver is not forbidden, this is not recommended.

If the Mode field of the request is 3 (client), the reply is set to 4 (server). If this field is set to 1 (symmetric active), the reply is set to 2 (symmetric passive). This allows clients configured in either client (NTP mode 3) or symmetric active (NTP mode 1) to interoperate successfully, even if configured in possibly suboptimal ways. For any other value in the Mode field, the request is discarded. In broadcast (unsolicited) mode, the VN field is set to 4, the Mode field is set to 5 (broadcast), and the Poll field set to the nearest integer base-2 logarithm of the poll interval.

Note that it is highly desirable that a broadcast server also supports unicast clients. This is so a potential broadcast client can calculate the propagation delay using a client/server exchange prior to regular operation using only broadcast client mode. A anycast server by design also is a unicast server. There does not seem to be a great advantage for a server to operate as both broadcast and anycast at the same time, although the protocol specification does not forbid it.

A broadcast or anycast server may or may not respond if not synchronized to a correctly operating reference source, but the preferred option is to respond, since this allows reachability to be determined regardless of synchronization state. If the server has never synchronized to a reference source. the LI field is set to 11 (unsynchronized). Once synchronized to a reference source, the LI field is set to one of the other three values and remains at the last value set even if the reference source becomes unreachable or turns faulty.

If synchronized to a reference source the Stratum field is set to 1 and the Reference Identifier field is set to the ASCII source identifier shown in Figure 2. If not synchronized, the Stratum field is set to zero and the Reference Identifier field set to an ASCII error identifier described below. In broadcast mode, the server sends broadcasts only if synchronized to a correctly operating reference source.

The Precision field is set to reflect the maximum reading error of the local clock. For all practical cases it is computed as the negative base-2 logarithm of the number of significant bits to the right of the decimal point in the NTP timestamp format. The Root Delay and Root Dispersion fields are set to 0 for a primary server; optionally, the Root Dispersion field can be set to a value corresponding to the maximum expected error of the radio clock itself.

The timestamp fields are set as follows. If the server is unsynchronized or first coming up, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the reference source. In unicast and anycast modes, the Receive Timestamp and Transmit Timestamp fields are set to the time of day when the message is sent and the Originate Timestamp field is copied unchanged from the Transmit Timestamp field of the request. It is important that this field be copied intact, as a NTP client uses it to avoid bogus messages. In broadcast mode, the Originate Timestamp and Receive Timestamp fields are set to 0 and the Transmit Timestamp field is set to the time of day when the message is sent. The following table summarizes these actions.

Field Name	Unicast/Anycast		Broadcast
	Request	Reply	
LI	ignore	as needed	as needed
VN	1-4	copied from request	4
Mode	1 or 3	2 or 4	5
Stratum	ignore	1	1
Poll	ignore	copied from request	log2 poll interval
Precision	ignore	-log2 server significant bits	-log2 server significant bits
Root Delay	ignore	0	0
Root Dispersion	ignore	0	0
Reference Identifier	ignore	source ident	source ident
Reference Timestamp	ignore	time of last source update	time of last source update
Originate Timestamp	ignore	copied from transmit timestamp	0
Receive Timestamp	ignore	time of day	0
Transmit Timestamp	(see text)	time of day	time of day
Authenticator	optional	optional	optional

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when

the reference source is inoperative. The most important indicator of an unhealthy server is the Stratum field, in which a value of 0 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

7. Configuration and Management

Initial setup for SNTP servers and clients can be done using a configuration file if a file system is available, or a serial port if not. Some folks hoped that in-service management of NTP and SNTP Version 4 servers and clients be performed using SNMP and a suitable MIB to be published, and this has happened in some commercial SNTP servers. But, the means used in the last decade and probably in the next is the NTP control and monitoring protocol defined in RFC-1305. Ordinarily, SNTP servers and clients are expected to operate with little or no site-specific configuration, other than specifying the IP address, subnet mask, gateway and DNS server.

Unicast clients must be provided with one or two designated server names or addresses. If two servers are provided, either can be used for active operation and the other for backup should the active one fail or show an error condition. Broadcast servers and anycast clients must be provided with the TTL and local broadcast or multicast group address. Unicast and anycast servers and broadcast clients may be configured with a list of address-mask pairs for access control, so that only those clients or servers known to be trusted will be accepted. Multicast servers and clients must implement the IGMP protocol and be provided with the local broadcast or multicast group address as well. The configuration data for cryptographic authentication is beyond the scope of this memo.

There are several scenarios which provide automatic server discovery and selection for SNTP clients with no pre-specified server configuration. For instance a role server with CNAME such as pool.ntp.org returns a randomized list of volunteer secondary server addresses and the client can select one or more as candidates. For an IP subnet or LAN segment including a NTP or SNTP server, SNTP clients can be configured as broadcast clients. The same approach can be used with multicast servers and clients. In both cases, provision of an access control list is a good way to insure only trusted sources can be used to set the local clock.

In another scenario suitable for an extended network with significant network propagation delays, clients can be configured for anycast addresses, both upon initial startup and after some period when the currently selected unicast source has not been heard. Following the defined protocol, the client binds to the server from which the first reply is received and continues operation in unicast mode.

8. The Kiss-o'-Death Packet

In the rambunctious Internet of today, it is imperative that some means be available to tell a client to stop making requests and go somewhere else. A recent experience involved a large number of home/office routers all configured to use a particular university time server. Under some error conditions a substantial fraction of these routers would send packets at

Code	Meaning
ACST	The association belongs to a anycast server
AUTH	Server authentication failed
AUTO	Autokey sequence failed
BCST	The association belongs to a broadcast server
CRYP	Cryptographic authentication or identification failed
DENY	Access denied by remote server
DROP	Lost peer in symmetric mode
RSTR	Access denied due to local policy
INIT	The association has not yet synchronized for the first time
MCST	The association belongs to a manycast server
NKEY	No key found. Either the key was never installed or is not trusted
RATE	Rate exceeded. The server has temporarily denied access because the client exceeded the rate threshold
RMOT	Somebody is tinkering with the association from a remote host running ntpdc. Not to worry unless some rascal has stolen you keys
STEP	A step change in system time has occurred, but the association has not yet resynchronized

Figure 3. Kiss Codes

intervals of one second. The resulting traffic spike was dramatic, and extreme measures were required to diagnose the problem and bring it under control. The conclusion is that clients must respect the means available to targeted servers to stop them from sending packets.

According to the NTP Version 3 specification, if the Stratum field in the NTP header is 1, indicating a primary server, the Reference Identifier field contains an ASCII string identifying the particular reference clock type. However, in NTP Version 3 nothing is said about the Reference Identifier field if the Stratum field is 0, which is called out as "unspecified". In NTP Version 4, if the Stratum field is 0, the Reference Identifier field can be used to convey messages useful for status reporting and access control. Packets of this kind are called Kiss-o'-Death (KoD) packets and the ASCII messages they convey are called kiss codes. The KoD packets got their name because an early use was to tell clients to stop sending packets that violate server access controls.

In general, a SNTP client should stop sending to a particular server if that server returns a reply with a Stratum field of 0, regardless of kiss code, and an alternate server is available. If no alternate server is available, the client should retransmit using an exponential-backoff algorithm described in the next section.

The kiss codes can provide useful information for an intelligent client. These codes are encoded in four-character ASCII strings left justified and zero filled. The strings are designed for character displays and log files. Usually, only a few of these codes can occur with SNTP clients, including DENY, RSTR and RATE. Others can occur more rarely, including INIT and STEP, when the server is in some special temporary condition. Figure 3 shows a list of the kiss codes currently defined.

9. On Being a Good Network Citizen

SNTP and its big brother NTP have been in explosive growth over the last few years, mirroring the growth of the Internet. Just about every Internet appliance has some kind of NTP support, including Windows XP, Cisco routers, embedded controllers and software systems of all kinds. This is the first edition of the SNTP RFC where it has become necessary to lay down rules of engagement in the form of design criteria for SNTP client implementations. This is necessary to educate software developers regarding the proper use of Internet time server resources as the Internet expands and demands on time servers increase, and to prevent the recurrence of the sort of problem mentioned above.

The following algorithm can be used as a pattern for specific implementations. It uses the following variables:

Timer: This is a counter that decrements at a fixed rate. When it reaches zero, a packet is sent and the timer initialized with the timeout for the next packet.

Maximum timeout: This is the maximum timeout determined from the given oscillator frequency tolerance and the required accuracy.

Server Name: This is the DNS name of the server. There may be more than one of them to be selected by some algorithm not considered here.

Server IP Address: This is the IPv4, IPv6 or OSI address of the server.

We assume the manufacturer operates one or more NTP time servers as a customer convenience. We further assume that a DNS request for a generic server name such as ntp.mytimeserver.com results in a random selection of server IP addresses available for that purpose. Each time a DNS request is received, a new randomized list is returned. The client ordinarily uses the first address on the list. The client operates as follows (note that steps 2 – 4 comprise a synchronization loop):

1. Consider the specified frequency tolerance of the system clock oscillator. Define the required accuracy of the system clock, then calculate the maximum timeout. For instance, if the frequency tolerance is 200 parts-per-million (PPM) and the required accuracy is one minute, the maximum timeout is about 3.5 days. Use the longest maximum timeout possible given the system constraints to minimize time server aggregate load, but never less than 15 minutes.

2. When first coming up or after reset, randomize the timeout from one to five minutes. This is to minimize shock when 3000 PCs are rebooted at the same time power is restored after a blackout. Assume at this time the IP address is unknown and the system clock is unsynchronized. Otherwise use the timeout value as calculated in previous loop steps. Note that it may be necessary to refrain from implementing the aforementioned random delay for some classes of ICSA certification.

3. When the timer reaches zero, if the IP address is not known, send a DNS query packet; otherwise send a NTP request packet to that address. If no reply packet has been heard since the last timeout, double the timeout, but not greater than the maximum timeout. If primary and secondary time servers have been configured, alternate queries between the primary and secondary servers when no successful response has been received.

4. If a DNS reply packet is received, save the IP address and continue in step 2. If a KoD packet is received remove that time server from the list, activate the secondary time server and continue in step 2. If a received packet fails the sanity checks, drop that packet and also continue in step 2. If a valid NTP packet is received, update the system clock, set the timeout to the maximum, and continue to step 2.

10. Acknowledgements

Jeff Learman was helpful in developing the OSI model for this protocol. Ajit Thyagarajan provided valuable suggestions and corrections.

11. References

[CAIN02] Cain, B., S. Deering, I. Kouvalas, B. Fenner, and A. Thyagarajan. "Internet group management protocol, version 3", RFC 3376, Cereva Networks, October 2002.

[COL94] Colella, R., R. Callon, E. Gardner, Y. Rekhter, "Guidelines for OSI NSAP allocation in the Internet", RFC 1629, NIST, May 1994.

[DAR81] Postel, J., "Internet Protocol", STD 5, RFC 791, USC Information Sciences Institute, September 1981.

[DEE89] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, Stanford University, August 1989.

[DEE96] Deering, S., R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, Xerox and Ipsilon, January 1996.

[DOB91] Dobbins, K, W. Haggerty, C. Shue, "OSI connectionless transport services on top of UDP - Version: 1", RFC 1240, Open Software Foundation, June 1991.

[EAS95] Eastlake, D., 3rd., and C. Kaufman, "Domain Name System Security Extensions", Work in Progress.

[FUR94] Furniss, P., "Octet sequences for upper-layer OSI to support basic communications applications", RFC 1698, Consultant, October 1994.

[HIN96] Hinden, R., and S. Deering, "IP Version 6 addressing Architecture", RFC 1884, Ipsilon and Xerox, January 1996.

[ISO86] International Standards 8602 - Information Processing Systems - OSI: Connectionless Transport Protocol Specification. International Standards Organization, December 1986.

[MIL92] Mills, D., "Network Time Protocol (Version 3) specification, implementation and analysis", RFC 1305, University of Delaware, March 1992.

[PAR93] Partridge, C., T. Mendez and W. Milliken, "Host anycasting service", RFC 1546, Bolt Beranek Newman, November 1993.

[POS80] Postel, J., "User Datagram Protocol", STD 6, RFC 768, USC Information Sciences Institute, August 1980.

[POS83] Postel, J., "Time Protocol", STD 26, RFC 868, USC Information Sciences Institute, May 1983.

Security Considerations

Security issues are not discussed in this memo.

Author's Address

David L. Mills
Electrical and Computer Engineering Department
University of Delaware
Newark, DE 19716
Phone: (302) 831-8247