```
Network Working Group                                          D. Mills
Request for Comments: 2030                      University of Delaware
Obsoletes: 1769                                          October 1996
Category: Informational
```

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

Status of this Memo

Abstract

   This memorandum describes the Simple Network Time Protocol (SNTP)
   Version 4, which is an adaptation 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 Version 3 model, including an anycast mode
   and an authentication scheme designed specifically for multicast and
   anycast modes. While the anycast mode extension is described in this
   document, the authentication scheme extension will be described in
   another document to be published later. Until such time that a
   definitive specification is published, these extensions should be
   considered provisional.

   This memorandum obsoletes RFC-1769, which describes SNTP Version 3.
   Its purpose is to correct certain inconsistencies in the previous
   document and to clarify header formats and protocol operations for
   current NTP Version 3 (IPv4) and proposed NTP Version 4 (IPv6 and
   OSI), which are also used for SNTP. 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 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 delays and jitter
   characteristics. Most users of the Internet NTP synchronization
   subnet of today use a software package including the full suite of
   NTP options and algorithms, which are relatively complex, real-time
   applications (see http://www.eecis.udel.edu/~ntp). 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 access strategies using simpler software
   appropriate for less stringent accuracy expectations.

   This document describes the Simple Network Time Protocol (SNTP)
   Version 4, which is a simplified access strategy for servers and
   clients using NTP Version 3 as now specified and deployed in the
   Internet, as well as NTP Version 4 now under development. 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.

SNTP Version 4 is designed to coexist with existing NTP and SNTP
Version 3 clients and servers, as well as proposed Version 4 clients
and servers. 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 ir SNTP Version 4. To a NTP or SNTP server, NTP and SNTP
clients are undistinguishable; to a NTP or SNTP client, NTP and SNTP
servers are undistinguishable. 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. NTP and SNTP Version 3
servers can operate in unicast and multicast modes. In addition, SNTP
Version 4 clients and servers can implement extensions to operate in
anycast mode.

It is strongly recommended that SNTP 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 or modem
time service 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 document is the reinterpretation of
certain NTP Versino 4 header fields which provide for IPv6 and OSI
addressing and optional anycast extensions designed specifically for
multicast service. These additions are in conjunction with the
proposed NTP Version 4 specification, which will appear as a separate
document. The only difference between the current NTP Version 3 and
proposed NTP Version 4 header formats is the interpretation of 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 identifier defined later in this document. In Version
3 secondary servers and clients, it contains the 32-bit IPv4 address
of the synchronization source. In Version 4 secondary servers and
clients, it contains the low order 32 bits of the last transmit
timestamp received from the synchronization source.

In the case of OSI, the Connectionless Transport Service (CLTS) is
used [ISO86]. Each SNTP packet is transmitted as tht 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 document, 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.

    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

SNTP Version 4 can operate in either unicast (point to point),
multicast (point to multipoint) or anycast (multipoint to point)
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 local clock offset
relative to the server. A multicast server periodically sends a
unsolicited message to a designated IPv4 or IPv6 local broadcast
address or multicast group address and ordinarily expects no requests
from clients. A multicast client listens on this address and
ordinarily sends no requests. An anycast client sends a request to a
designated IPv4 or IPv6 local broadcast address or multicast group
address. One or more anycast servers reply with their individual
unicast addresses. The client binds to the first one received, then
continues operation in unicast mode.

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

In unicast mode, the client and server end-system addresses are
assigned following the usual IPv4, IPv6 or OSI conventions. In
multicast mode, the server uses a designated local broadcast address
or multicast group address. An IP local broadcast address has scope
limited to a single IP subnet, since routers do not propagate IP
broadcast datagrams. On the other hand, an IP multicast group address
has scope extending to potentially the entire Internet. The scoping,
routing and group membership procedures are determined by
considerations beyond the scope of this document. For IPv4, the IANA
has assigned the multicast group address 224.0.1.1 for NTP, which is

used both by multicast servers and anycast clients. NTP multicast
addresses for IPv6 and OSI have yet to be determined.

Multicast clients listen on the designated local broadcast address or
multicast group address. In the case of local broadcast addresses, no
further provisions are necessary. In the case of IP multicast
addresses, the multicast client and anycast server must implement the
Internet Group Management Protocol (IGMP) [DEE89], in order that the
local router joins the multicast group and relays messages to the
IPv4 or IPv6 multicast group addresses assigned by the IANA. Other
than the IP addressing conventions and IGMP, there is no difference
in server or client operations with either the local broadcast
address or multicast group address.

    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 value to be used are beyond the scope of this
    document.

Anycast mode is designed for use with a set of cooperating servers
whose addresses are not known beforehand by the client. An anycast
client sends a request to the designated local broadcast or multicast
group address as described below. For this purpose, the NTP multicast
group address assigned by the IANA is used. One or more anycast
servers listen on the designated local broadcast address or multicast
group address. Each anycast server, upon receiving a request, sends a
unicast reply message to the originating client. The client then
binds to the first such message received and continues operation in
unicast mode. Subsequent replies from other anycast servers are
ignored.

    In the case of SNTP as specified herein, there is a very real
    vulnerability that SNTP multicast clients can be disrupted by
    misbehaving or hostile SNTP or NTP multicast servers elsewhere in
    the Internet, since at present all such servers use the same IPv4
    multicast group address assigned by the IANA. Where necessary,
    access control based on the server source address can be used to
    select only the designated server known to and trusted by the
    client. The use of cryptographic authentication scheme defined in
    RFC-1305 is optional; however, implementors should be advised that
    extensions to this scheme are planned specifically for NTP
    multicast and anycast modes.

   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 multicast 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.

   In NTP Version 3, the reference identifier was often used to
   walk-back the synchronization subnet to the root (primary server)
   for management purposes. In NTP Version 4, this feature is not
   available, since the addresses are longer than 32 bits. However,
   the intent in the protocol design was to provide a way to detect
   and avoid loops. A peer could determine that a loop was possible
   by comparing the contents of this field with the IPv4 destination
   address in the same packet. A NTP Version 4 server can accomplish
   the same thing by comparing the contents of this field with the
   low order 32 bits of the originate timestamp in the same packet.
   There is a small possibility of false alarm in this scheme, but
   the false alarm rate can be minimized by randomizing the low order
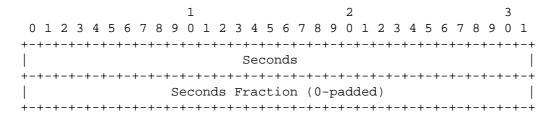   unused bits of the transmit timestamp.

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 high-order, position. 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). One way of doing this is to generate
   a random bitstring in a 64-bit word, then perform an arithmetic
   right shift a number of bits equal to the number of significant
   bits of the timestamp, then add the result to the original
   timestamp.

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.

```
                      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)                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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).
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). 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 17 years,
   it remains a possibility that it will be in use 40 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 2036-
   2104 and UTC time is reckoned from 6h 28m 16s UTC on 7 February
   2036. Note that when calculating the correspondence, 2000 is not a
   leap year. Note also that leap seconds are not counted 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
   both the Source Port and Destination Port fields in the UDP header.
   The remaining UDP header fields should be set as described in the
   specification.

Below 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 contents of the
reference identifier field. The header fields are defined 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)                    |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Key Identifier (optional) (32)                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                                                               |
   |                 Message Digest (optional) (128)               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

As described in the next section, 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)
```

Version Number (VN): This is a three-bit integer indicating the
NTP/SNTP version number. The version number is 3 for Version 3 (IPv4
only) and 4 for Version 4 (IPv4, IPv6 and OSI). If necessary to
distinguish between IPv4, IPv6 and OSI, the encapsulating context
must be inspected.

Mode: This is a three-bit integer indicating the mode, with values
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 client sets this field to 3
(client) in the request and the server sets it to 4 (server) in the
reply. In multicast mode, the server sets this field to 5
(broadcast).

Stratum: This is a eight-bit unsigned integer indicating the stratum
level of the local clock, with values defined as follows:

```
    Stratum   Meaning
    -----------------------------------------------
    0         unspecified or unavailable
    1         primary reference (e.g., radio clock)
    2-15      secondary reference (via NTP or SNTP)
    16-255    reserved
```

Poll Interval: This is an eight-bit signed integer indicating the
maximum interval between successive messages, in seconds to the
nearest power of two. The values that can appear in this field
presently range from 4 (16 s) to 14 (16284 s); however, most
applications use only the sub-range 6 (64 s) to 10 (1024 s).

Precision: This is an eight-bit signed integer indicating the
precision of the local clock, in seconds to the nearest power of two.
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 0 to several
hundred milliseconds.

Reference Identifier: This is a 32-bit bitstring identifying the
particular reference source. In the case of NTP Version 3 or Version
4 stratum-0 (unspecified) or stratum-1 (primary) servers, this is a
four-character ASCII string, left justified and zero padded to 32
bits. In NTP Version 3 secondary servers, this is the 32-bit IPv4
address of the reference source. In NTP Version 4 secondary servers,
this is the low order 32 bits of the latest transmit timestamp of the
reference source. NTP primary (stratum 1) servers should set this
field to a code identifying the external reference source according
to the following list. 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.

```
      Code      External Reference Source
      ---------------------------------------------------------------
      LOCL      uncalibrated local clock used as a primary reference for
                a subnet without external means of synchronization
      PPS       atomic clock or other pulse-per-second source
                individually calibrated to national standards
      ACTS      NIST dialup modem service
      USNO      USNO modem service
      PTB       PTB (Germany) 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
```

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

   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 multicast mode, unicast mode or anycast
   mode. In multicast mode, the client sends no request and waits for a
   broadcast (mode 5) from a designated multicast server. In unicast
   mode, the client sends a request (mode 3) to a designated unicast
   server and expects a reply (mode 4) from that server. In anycast
   mode, the client sends a request (mode 3) to a designated local
   broadcast or multicast group address and expects a reply (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 from 64 s to 1024 s,
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 first octet
and (optional) Transmit Timestamp fields. In the first octet, the LI
field is set to 0 (no warning) and the Mode field is set to 3
(client). The VN field must agree with the version number of the
NTP/SNTP server; however, Version 4 servers will also accept previous
versions. 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 version used by SNTP Version 4 clients. 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, since the header fields used
by SNTP clients are unchanged. Version 4 servers are required to
reply in the same version as the request, so the VN field of the
request also specifies the version of the reply.

While not necessary in a conforming client implementation, in unicast
and anycast modes it highly recommended that the transmit timestamp
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 multicast mode, the
client has no information to calculate the propagation delay or
determine the validity of the server, unless the NTP authentication
scheme is used.

To calculate the roundtrip delay d and local clock offset t relative
to the server, the client sets the transmit timestamp in the request
to the time of day according to the client clock in NTP timestamp

format. The server copies this field to the originate timestamp in
the reply and sets the receive timestamp and transmit timestamp 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

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

The following table summarizes the SNTP client operations in unicast,
anycast and multicast modes. The recommended error checks are shown
in the Reply and Multicast 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 | | Multicast |
	Request	Reply	
LI	0	0-2	0-2
VN	1-4	copied from request	1-4
Mode	3	4	5
Stratum	0	1-14	1-14
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

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 always operates as a primary
   (stratum 1) server.

   A SNTP server can operate in unicast mode, anycast mode, multicast
   mode or any combination of these modes. In unicast and anycast modes,
   the server receives a request (mode 3), modifies certain fields in
   the NTP header, and sends a reply (mode 4), possibly using the same
   message buffer as the request. In anycast mode, the server listens on
   the designated local broadcast or multicast group address assigned by
   the IANA, 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.
   Multicast 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.

   In unicast and anycast modes, the VN and Poll fields of the request
   are copied intact to the reply. If the Mode field of the request is 3
   (client), it is set to 4 (server) in the reply; otherwise, this field
   is set to 2 (symmetric passive) in order to conform to the NTP
   specification. This allows clients configured in symmetric active
   (mode 1) to interoperate successfully, even if configured in possibly
   suboptimal ways. In multicast (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, if a server supports
      multicast mode, it also supports unicast mode. This is so a
      potential multicast client can calculate the propagation delay
      using a client/server exchange prior to regular operation using
      only multicast mode. If the server supports anycast mode, then it
      must support unicast mode. There does not seem to be a great
      advantage to operate both multicast and anycast modes at the same
      time, although the protocol specification does not forbid it.

   In unicast and anycast modes, the server may or may not respond if
   not synchronized to a correctly operating radio clock, but the
   preferred option is to respond, since this allows reachability to be
   determined regardless of synchronization state. In multicast mode,
   the server sends broadcasts only if synchronized to a correctly

   operating reference clock.

   The remaining fields of the NTP header are set in the following way.
   Assuming the server is synchronized to a radio clock or other primary
   reference source and operating correctly, the LI field is set to 0
   and the Stratum field is set to 1 (primary server); if not, the
   Stratum field is set to 0 and the LI field is set to 3. 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 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 Reference Identifier is set to designate the
   primary reference source, as indicated in the table of [Section 5](#) of
   this document.

   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 radio clock or modem. 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 replays. In multicast
   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 | | Multicast |
	Request	Reply	
LI	ignore	0 or 3	0 or 3
VN	1-4	copied from request	4
Mode	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 radio update	time of last radio update

```
Originate Timestamp      ignore      copied from  0
                                     transmit
                                     timestamp
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 primary reference source is inoperative. The most
important indicator of an unhealthy server is the LI field, in which
a value of 3 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. It is intended that in-service management of NTP and SNTP
Version 4 servers and clients be performed using SNMP and a suitable
MIB to be published later. Ordinarily, SNTP servers and clients are
expected to operate with little or no site-specific configuration,
other than specifying the IP address and subnet mask or OSI NSAP
address.

Unicast clients must be provided with the designated server name or
address. If a server name is used, the address of one of more DNS
servers must be provided. Multicast servers and anycast clients must
be provided with the TTL and local broadcast or multicast group
address. Anycast servers and multicast 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 used. These 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
document.

There are several scenarios which provide automatic server discovery
and selection for SNTP clients with no pre-specified configuration,
other than the IP address and subnet mask or OSI NSAP address. For a
IP subnet or LAN segment including a fully functional NTP server, the
clients can be configured for multicast mode using the local
broadcast address. The same approach can be used with other servers
using the multicast group address. 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
   mode, 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 first reply heard and
   continues operation in unicast mode. In this mode the local clock can
   be automatically adjusted to compensate for the propagation delay.

   In still another scenario suitable for any network and where
   multicast service is not available, the DNS can be set up with a
   common CNAME, like time.domain.net, and a list of address records for
   NTP servers in the same domain. Upon resolving time.domain.net and
   obtaining the list, the client selects a server at random and begins
   operation in unicast mode with that server. Many variations on this
   theme are possible.

8. Acknowledgements

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

9. References

   [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 Engineering Department
   University of Delaware
   Newark, DE 19716

   Phone: (302) 831-8247