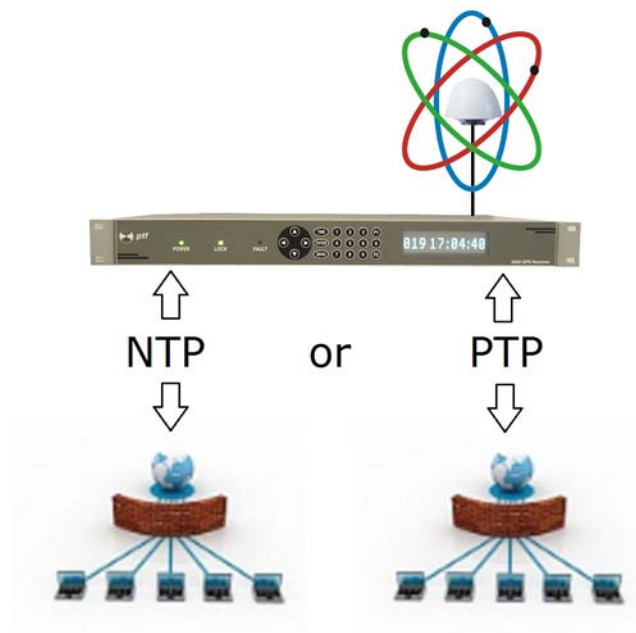# NTP (Network Time Protocol)

and

PTP (Precision Time Protocol)

Introduction

This paper addresses two protocols available for time transfer over Ethernet, NTP (Network Time Protocol) and PTP (Precision Time Protocol). The paper is in two parts, Part 1 for NTP, and Part 2 for PTP. Both protocols rely upon the exchange of information packets over a network. The packets contain time stamps defining when a packet was transmitted and when it was received, and by analyzing these timestamps it is possible to determine the relative time between the exchanging elements. Both protocols also contain mechanisms for the host to act as either a provider (server) of time or a recipient (client) receiving time. This paper is written more from the perspective of a time provider and does not cover in as much detail the receiving end of the transfer.

NTP has been around for many years (since the early 1980's) and is widely used in many time and frequency applications today either just for a coarse time measure, supplemented by some other precision timing mechanism (e.g. 1PPS), or can be used directly for time stamping events in applications where precision of a few milliseconds is sufficient .   There are many NTP "clients" freely available for download on the internet which makes this a popular choice for many applications.

The newer PTP (often referred to by the name of the standard that defines it, IEEE 1588) was first introduced in the early 2000's and adds additional capability allowing much higher precision of time transfer (sub microseconds) albeit at greater complexity. PTP is now in its second iteration (IEEE 1588 - 2008, or PTP v2). This second iteration includes additional features to achieve higher accuracies, e.g. inclusion of a capability that allows application specific "profiles" that provide better estimation of timing delays etc. to be expected in particular application situations.

Both protocols also provide different "modes" of operation whereby a provider (server) may interact directly with a request from a receiver (client) in "Unicast" mode, or when operating in "Multicast" mode, may send the time message at regular intervals to a pre-defined IP address, from which multiple clients receive the data.

PART 1 - Network Time Protocol (NTP)

Since its introduction in the early 1980s NTP (Network Time Protocol) has become widely accepted as a relatively easy and convenient way of transferring time over the Ethernet to the point where virtually every Personal Computer shipped today includes an NTP client as standard to set the computer time from the internet.

The NTP concept basically consists of two elements, an NTP server and an NTP client.

An NTP server is a device that acquires time (typically from the Global Positioning Satellite system) and outputs NTP using TCP/IP (either on request or at regular preset intervals) on Ethernet port 123. The output packet is described below and includes information regarding the accuracy/origin of the time being transmitted as well as the time. The time is always based on UTC (Universal Time Coordinated - the worldwide standard for time) and is converted to NTP seconds/milliseconds. In many cases where time is directly acquired by the NTP server (e.g. from the GPS system) the server is actually an SNTP server, where the "S" stands for Simple. The NTP model is less sophisticated however the protocol is identical.

An NTP client receives the transmitted NTP packet, on Ethernet port 123, and converts back from NTP seconds to UTC and often in turn to "local" time. There is no facility within NTP to transmit local time offsets or daylight savings time.

The conversion to and from NTP seconds may seem strange at first, however it is clearly important that all NTP servers and NTP clients refer to the same timescale, and the background of the NTP second is connected to the commencement of UTC at 00:00 hours on January 1, 1972. At that time it was decided to set the NTP second to a value of 2,272,060,800.0 which meant that extrapolating back, the NTP second was 0 (i.e. started) at 00:00 hours on January 1, 1900.

NTP time is represented by a 64 bit unsigned integer field. The integer part of NTP time (i.e. whole seconds) is contained within the first 32 bits and therefore will "roll over" approximately every 136 years (2^{32} = 4294967296 seconds which equates to around 136.19 years). This means that if still in use, NTP will "roll over" some time in 2036. The fractional part of NTP is contained within the second 32 bits which actually gives a resolution of $1/2^{32}$ or approximately 0.23 nano seconds - more than adequate for most applications, especially when considering that very few primary clocks provide time to an accuracy of better than a few nano seconds.

In many applications requiring precision timing as well as absolute time, NTP is used in combination with a precise one pulse per second signal (1PPS). This allows the application equipment to be synchronized to a precision of around 20 to 50 nano seconds in absolute time, with the coarse time being provided by NTP, and the precision one second MARK being provided by a 1PPS signal (usually from a GPS receiver). An internal flywheel oscillator within the equipment is more than capable of maintaining the precision from second to second between updates.

First the structure/content of the NTP messages is described, and then the message interchange to calculate time offset between servers and clients.

The structure of the NTP packet is shown below:

The NTP indicators are contained in a 32 bit word 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   |

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

LI = an indicator for upcoming leap seconds

```
        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)
```

VN   = the NTP version number (latest/current version is NTPv4)

MODE = a 3 bit number indicating the current mode of NTP operation. Typically when used in a precision time standard as an NTP server responding to a request from a client (i.e. Unicast mode) , the received mode will be 3, and the returned mode will be set to 4.

```
0       = reserved
1       = symmetric active
2       = symmetric passive
3       = client
4       = server
5       = broadcast/multicast
6       = NTP control message
7       = reserved
```

```
STRATUM = an 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
```

Normally in a precision time server locked to GPS or another accurate source the STRATUM value will be set to 1 (Primary Clock). If the time server is not locked to GPS (i.e. in holdover) or some other source then the STRATUM would be set to 0 (undefined). It is important to note that many available NTP clients (e.g. on personal computers)will not synchronize to an NTP server unless the STRATUM flag is set to 1.

`PRECISION` = an eight-bit signed integer indicating the precision of the local clock, in seconds to the nearest power of two.

Typically a GPS locked time server will set the precision to a value around 10 microseconds i.e. -17 (2^{-17} = 0.0000076 or 7.6 microseconds)

The next three 32 bit words are:

```
ROOT DELAY
ROOT DISPERSION
REFERENCE IDENTIFIER
```

`ROOT DELAY` = the round trip delay to the primary reference source. Where the NTP packet is being sent from a GPS locked time server this round trip delay is effectively 0

`ROOT DISPERSION` = represents the maximum error of the local clock relative to the primary reference. Again, for an NTP server locked to GPS this would be effectively 0.

`REFERENCE IDENTIFIER` = identifies the reference source. In the case of stratum 2 source or worse, this is usually the IP address of the primary source.

Following the ROOT DISPERSION are four 64 bit timestamps as follows;

```
REFERENCE TIMESTAMP
ORIGINATE TIMESTAMP
RECEIVE TIMESTAMP
TRANSMIT TIMESTAMP
```

Each of the above 64 bit words take the form:

first 32 bits   = integer seconds

second 32 bits = fractional seconds (0.23 nano second resolution)

`REFERENCE TIMESTAMP` is set to time when the NTP server was last updated, effectively the current time.

`ORIGINATE TIMESTAMP` is the time that the originating message was sent. This is copied from the received message i.e. from the client's "`TRANSMIT`" timestamp.

`RECEIVE TIMESTAMP` is the time at which the message from the client is received

`TRANSMIT TIMESTAMP` is the time at which the NTP message is re-transmitted from the NTP server.

There is one final optional part of the message:

a 96 bit "Authenticator". If used this contains the "key" identifier and encrypted checksum of the message contents.

The complete NTP message then looks like :

```
|LI|VN |MODE|STRATUM|POLL|PRECISION|      32 bits
ROOT DELAY                                32 bits
ROOT DISPERSION                           32 bits
REFERENCE IDENTIFIER                      32 bits
REFERENCE TIMESTAMP                       64 bits
ORIGINATE TIMESTAMP                       64 bits
RECEIVE TIMESTAMP                         64 bits
TRANSMIT TIMESTAMP                        64 bits
AUTHENTICATOR (optional)                  96 bits
```

The message interchange between an NTP client and NTP server can be graphically represented as shown below;
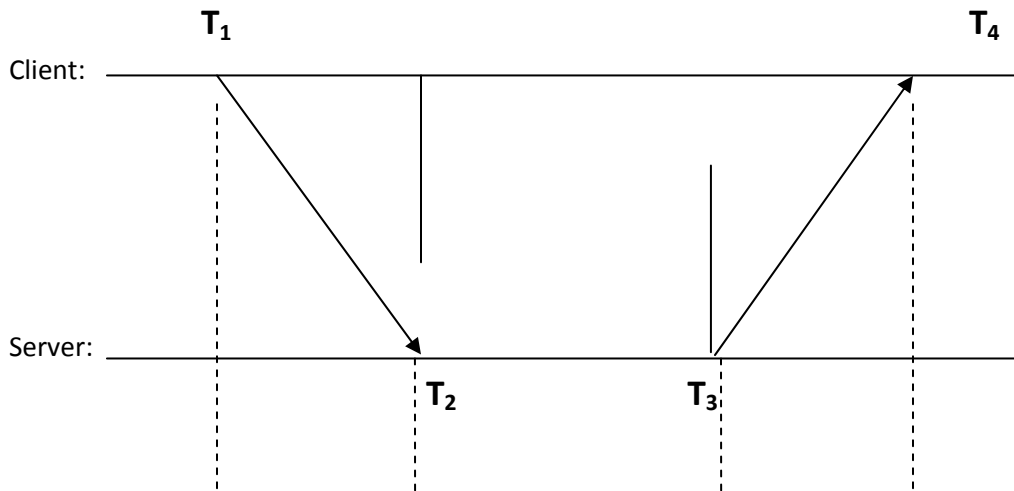
The synchronization process goes as follows;

Client sends Server an NTP request, and timestamps the request: T_1

When the server receives the request it timestamps it: T_2

When the server responds to the request it timestamps the response: T_3

When the client receives the response, it timestamps it: T_4

Now the client can calculate two important parameters:

Roundtrip delay (to the server)	=	difference between the time client sent the message and the time it received it back **less** the delay time within the server.
	=	$(T_4-T_1) - (T_3-T_2)$

Time offset (Client & Server) θ	=	[(timestamp when sent by client **less** timestamp when received by server) **plus** timestamp when sent back by server less timestamp when received)]/2
	=	$[(T_2-T_1) + (T_3-T_4)]/2$

Knowing these two values, the time in the client can now be set to match the time in the server. In practice, due to variability in delays through the network, typical synchronization accuracies that can be achieved with NTP are in the order of a few milliseconds. This is quite impressive when considering that the round trip delay between client and server can be several seconds.

In practice for a server consisting of an embedded processor locked to GPS, T_2 and T_3 will be effectively the same value.

Leap Seconds

NTP seconds are always synchronized with UTC seconds. Although NTP has a leap indicator bit, it has no mechanism for dealing with leap seconds, and therefore when a leap second is added, NTP seconds are suspended for 1 second.