A Basic Principle of NTP Time Synchronization

Duy-Ky Nguyen, PhD
2002-Sep-26

1. Introduction

The **Network Time Protocol** (**NTP**) is a protocol currently defined in the RFC-1305 standard that allows time synchronization over networks. At any time on a network, there is only one server NTP clock to do time synchronizing many client ones attached to the network. In this note, we present only the basic principle how NTP provides time synchronization.

There are 4 timestamps associated with 4 timing operations as described below

1. A client NTP stamps the time upon sending a request message (*Originate Timestamp*[1]);
2. The server stamps the time upon receiving the request message (*Receive Timestamp*[2]);
3. The server stamps the time upon sending the response message (*Transmit Timestamp*[3]);
4. The client stamps the time upon receiving the response message (*Reference Timestamp*[4]);

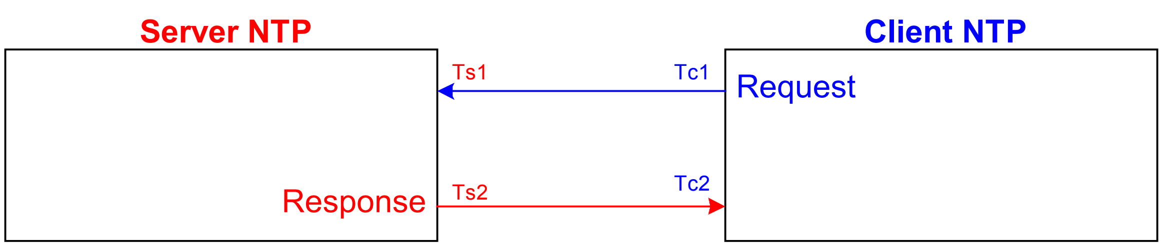
2. Computation

Practically we have a running server NTP clock and a client one just powered up, so

$$T_s = T_c + T_o \tag{1}$$

where T_s, T_c are server and client clock time, respectively; and T_o is offset time of client NTP from the server. The goal is to sync a client to a server NTP, in other word to get zero offset time, *i.e.* $T_o = 0$.

The client NTP clock will do all computations for its clock synchronized to the server, *i.e.* to achieve $T_o = 0$.



For the direction from server to client NTP, we have

$$T_{s1} = T_{c1} + T_o + T_{c2s} \tag{2}$$

where

- T_{s1} is server timestamp upon receiving the request message (receive timestamp);
- T_{c1} is client timestamp upon sending a request message (originate timestamp);
- T_{c2s} is traveling time from client to server.

For the reverse direction in response, we have

$$T_{c2} = T_{s2} - T_o + T_{s2c} \tag{3}$$

where

- T_{c2} is client timestamp upon receiving response message (receive timestamp);
- T_{s2} is server timestamp upon sending response message (transmit timestamp);
- T_{s2c} is traveling time from server to client, i.e. in reverse direction.

Note opposite sign associated with T_o in Eqs. (2) & (3).

So, we have a system equation of only 2 equations (2) & (3) for 3 unknowns T_o, T_{c2s} and T_{s2c}. In algebra theory, a system equation can be solved only if number of *independent* equations and unknowns are equal. An independent equation cannot be derived from another one. Therefore, we have to assume symmetric transmission paths, so the **round trip delay** is given by

$$T_d = T_{c2s} + T_{s2c} \quad \Rightarrow \quad T_{c2s} = T_{s2c} = T_d/2 \tag{4}$$

and reduce to 2 unknowns T_o and T_d

$$\left.\begin{aligned} T_{s1} &= T_{c1} + T_o + T_d/2 \\ T_{c2} &= T_{s2} - T_o + T_d/2 \end{aligned}\right\} \Leftrightarrow \left.\begin{aligned} T_o + T_d/2 &= T_{s1} - T_{c1} \\ T_o - T_d/2 &= T_{s2} - T_{c2} \end{aligned}\right\} \tag{5}$$

Adding and subtracting Eq.(5), we have the solutions

$$\begin{cases} T_o = \dfrac{\Delta_{sc1} + \Delta_{sc2}}{2} \\ T_d = \Delta_{sc1} - \Delta_{sc2} \end{cases} \tag{6}$$

where

$$\left.\begin{aligned} \Delta_{sc1} &= T_{s1} - T_{c1} \\ \Delta_{sc2} &= T_{s2} - T_{c2} \end{aligned}\right\} \tag{7}$$

The client NTP is synchronized to the server when $T_o = 0$, so, based on Eq.(1), the client update clock given by

$$T_{c,update} = T_{c,current} + T_o \tag{8}$$

3. Conclusion

Client is responsible for getting timestamp from received messages and calculating offset and round-trip delay time.