NTP sync of Wharton display clocks relies on the client clock polling the NTP server where messages at sent back and forth to establish 'what time it is' taking into account network delays (see a more detailed explanation at the end of this document -see page 3 '**The Clock Synchronization Algorithm'** [extract from How Does NTP Work? - Kevin Sookocheff]).

Customer wanted to know the standard polling frequency of Wharton display clocks:

From: Colin Hird
Sent: 14 June 2022 12:52
To: sales@wharton.uk
Subject: CIRS: 99034 - Wharton NTP updates from display clocks

Hello there,

A client has posed an interesting question!

With respect to the Wharton display clocks that slave to NTP
e.g. WHARTON 4900N, WHARTON 4900NE

Customer is asking how often the clock itself checks the incoming NTP timecode.
i.e. what is the reaction time should the NTP timecode change for whatever reason.

They have a simulation scenario whereby their NTP signal can change in an instant
and they need a display that would update immediately.

I would imagine that the reaction time would be immediate given that I'm of the belief that the clocks only run from their internal clock if the NTP is interrupted.

Please advise how often the NTP timecode is checked and what the reaction time to display such maybe if a change is detected.

Cheers!

Regards.

Colin Hird
Canford Technical Support

From: ga@wharton.uk <ga@wharton.uk> On Behalf Of Wharton Sales
Sent: 14 June 2022 14:04
To: Colin Hird <ch@canford.co.uk>
Subject: Re: CIRS: 99034 - Wharton NTP updates from display clocks

Hi Colin,

The way NTP works is the client device polls the NTP server which then returns an NTP packet. With our clocks this happens every few minutes once synchronisation has been established. Under normal circumstances there is just no need for this to happen more regularly and it would cause unnecessary network traffic.

Do you know how many units your client is looking for at all? I'm not sure if it would be possible but I can check with our technical team to see if we can do anything special.

Kind regards,

Gary Amor

Continued…………

Wharton Clocks NTP Client Polling interval

Whartons response of a ‘few minutes’ polling interval wasn’t adequate for the client and eventually:
(Edited for clarity)

From: jb@wharton.uk <jb@wharton.uk> On Behalf Of Wharton Sales
Sent: 15 June 2022 16:57
To: Colin Hird <ch@canford.co.uk>
Cc: ga@wharton.uk
Subject: Re: Response asap please - CIRS: 99034 - Wharton NTP updates from display clocks

Hi Colin,

>However as stated before in my last email the customer would like to know by close of business today
>the precise time scale between polling that the standard product may offer.

Typically 64 seconds but it may vary under certain conditions.

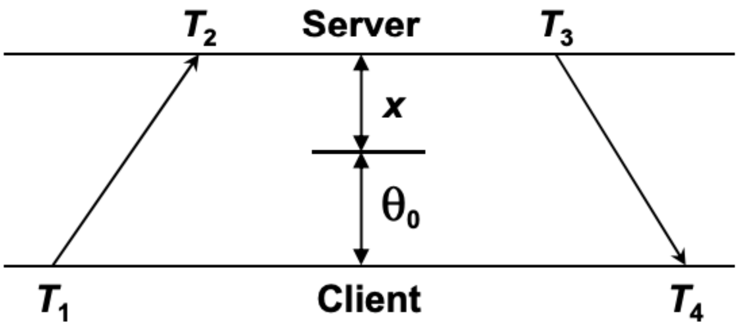
If your client is interested in a special unit just let us know and we will speak to our technical team, as the standard unit is not designed to work well in this system.

Kind regards,

Jimena Barillas

The Clock Synchronization Algorithm

During typical operation, an NTP client regularly polls one or more NTP servers to receive updated time data. Upon receipt of new data, the client computes the time offset and round-trip delay from the server as in the following figure.



$$\theta = \tfrac{1}{2}[(T_2 - T_1) + (T_3 - T_4)]$$
$$\delta = (T_4 - T_1) - (T_3 - T_2)$$

Offset and Delay Calculations

The time offset θ, the difference in absolute time between the two clocks, is defined mathematically by

$$\theta = \left| \frac{(t_2 - t_1) + (t_3 - t_4)}{2} \right| .$$

More intuitively, the offset calculation is the absolute time difference between the two clocks time it takes for packets to transmit between the client and the server.

Using the time data, we can also calculate the network delay δ:

$$\delta = (t_4 - t_1) - (t_3 - t_2).$$

where,

- *t*1 is the client's timestamp of the request packet transmission,
- *t*2 is the server's timestamp of the request packet reception,
- *t*3 is the server's timestamp of the response packet transmission and
- *t*4 is the client's timestamp of the response packet reception.

At the very heart of NTP are the algorithms used to improve the accuracy of the values for θ and δ using filtering and selection algorithms. The complexity of these algorithms varies depending on the statistical properties of the path between peers and the accuracies required. For example, if two nodes are on the same gigabit LAN, the path delays between messages sent between peers are usually within or below any required clock accuracies. In a case like this, the raw offsets delivered by the receive procedure can be used to directly adjust the local clock. In other cases, two nodes may be distributed widely over the Internet and the delay might be much larger than acceptable.