

What is the Time? Part 2

Call: +44 (0)23 8098 8890

E-mail: sales@itdev.co.uk (<mailto:sales@itdev.co.uk>)

[Home \(/\)](#) / [Blog \(/blog\)](#) / What is the Time? Part 2

Posted 14th December 2018, By James H ([/company/about/team/james-h](#))

Making a Stratum 1 Linux Time Server: Part 2

In the previous article we saw that getting distributed devices to agree on the current time is hard. This article, the second in a series of three, will introduce the concept of a time server – a critical piece of infrastructure required to make everyone agree on the time.

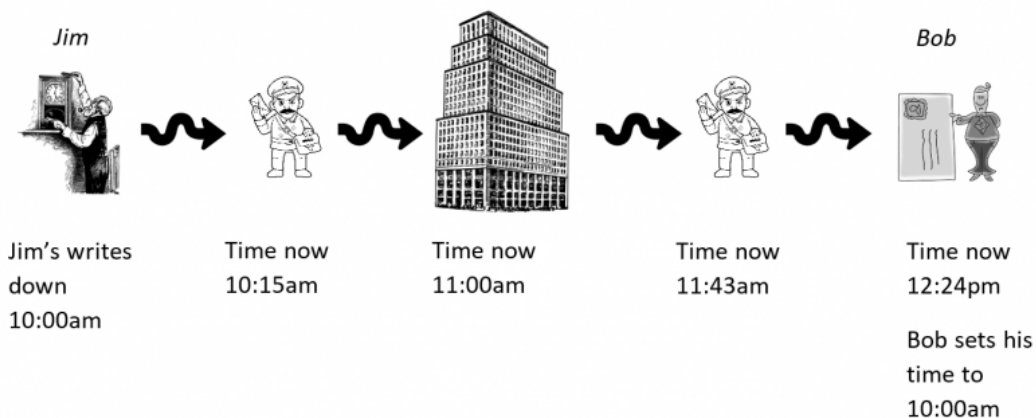
What is a Time Server

We've seen how our wrist watches need synchronisation. This was an allegory for what we need in our various bits of technology. Normally, whether we are dealing with a network of IoT sensors or providing a frequency source to a base station, etc. we need a high degree of accuracy, way more than to the nearest second! Normally, we wouldn't specify the instantaneous accuracy but provide a "mask" that makes it explicit as to how much a device can drift from a reference before it can no longer function. We will also limit the allowed jitter on a reference.

The question that a time server answers is “how do we deliver time to our devices?” It is a point of contact where devices can ask for the time, normally using protocols such as NTP or PTP. These protocols are used to overcome the difficulties in delivering time in networks that do not transmit a clock signal, most notably over Ethernet. By delivering time to our devices we allow them all to synchronise and agree on the same time (within tolerances).

However, delivering time across any network is not a trivial task. When “sending time” it is not enough for the time server to “write down” its time and then send it.

Consider an example. Let's say Jim owns the master clock. To send time to Bob he looks at the clock and writes down the current time. He then puts it in an envelope and mails it to Bob. The letter gets collected by the post man, delivered to the sorting centre and from there to Bob. Bob then opens the envelope, reads the time and sets his watch to this time. Will Bob's watch now be in sync with Jim's master clock? Clearly not as we see below:



Bob needs to know that it took 2hrs 24mins to deliver the message. If he knew that, he could set his watch to 10:00am plus 2hrs 24min to get the correct time of 12:24pm. Protocols like NTP and PTP allow Bob to figure out this delay.

However, it gets more complicated. The mail service is not perfect. The time it takes for the letter to get from Jim to Bob isn't constant. Sometimes the sorting office will be busier, so the letter will spend longer there. Sometimes the postman is stuck in a traffic jam or must take detours to avoid road works. This means that each time Bob gets a time from Jim, he can't be sure how long the message took to get to him. Thus, he can't apply a constant 2hrs 24 mins to get the correct time!

This is a problem because Bob knows his time piece is inherently inaccurate. When he gets a time and syncs his watch to it, over time his watch will drift and slowly begin to tell the wrong time. To solve this Bob needs to get constant updates. NTP and PTP step in to solve this problem by making a best-estimate of the transmission delay.



On a real-life Ethernet network, the analogy holds up quite well. The situation is more complex still, but one can get an appreciation that the problem of “transmitting time” is not at all easy.

Not all reference clocks are created equal. Some are better than others and a server’s “stratum” is used to indicate how accurate it is. The stratum is the distance from the reference clock/grand master. So, the first layer of time servers that lock to the grand master are stratum 1. The time servers that lock to stratum 1 servers are stratum 2 and so on...

At each stratum accuracy is lost, so the further down the chain you are the less accurate you will be.

A GPS receiver is considered a stratum 0 source, so if you sync your time server to it you have a stratum-1 time server.

Having found out why we need a time server and, in general, what one is, we can continue to the next article which will tell us how to set a time server up on a Debian based Linux server.

How ITDev Can Help

As a provider of software and electronics design services, we are often working with the Linux and Android operating systems. We have extensive experience in developing, modifying, building, debugging and bring up of these operating systems be it for new developments or working with existing kernels to demonstrate new device technologies and features.

We offer advice and assistance to companies considering to use or already using Linux or Android on their embedded system. Initial discussions are always free of charge, so if you have any questions, or would like to find out more, we would be delighted to speak to you. If you are interested in attending a workshop on embedded Linux/Android, or receiving more information on this topic, please sign up to our Embedded Linux Interest Group (<http://eepurl.com/dCms-P>).



Click to find out about our Embedded Linux Interest Group, including upcoming events

(<http://eepurl.com/dCms-P>)

- Follow us on LinkedIn (<https://www.linkedin.com/company/itdev/>), Twitter (<https://twitter.com/ITDevLtd>) or Facebook (<https://www.facebook.com/itdevltd/>) to find out when Part 3, the final part of the 'What is the Time' blog series will be published as well as our forthcoming blog posts.

< prev (/blog/what-time-part-1) next > (/blog/what-time-part-3)



(<http://www.nmi.org.uk/>)



(<https://www.theiet.org/>)



(<http://accu.org/>)



(<http://www.ktn-uk.co.uk/>)

Latest Blog Posts