TSCclock Goes Live.

A demonstration of a robust, accurate replacement to *ntpd*.

Julien Ridoux and Darryl Veitch
ARC Special Research Center for Ultra-Broadband Information Networks (CUBIN), an affiliated program of National ICT Australia, The University of Melbourne, Australia
{jridoux,dveitch}@unimelb.edu.au

ABSTRACT

Clock synchronization is increasingly critical as networks become faster, for example in distributed gaming and information systems, and network measurement. The *TSCclock* is the result of an intensive research effort [1] to provide a replacement to the Network Time Protocol daemon *ntpd* for network based synchronization. The TSCclock is a complete redesign of synchronization over noisy networks which differs fundamentally from *ntpd* in several respects. It has been extensively tested running live in many benchmarked experiments months long and has shown to deliver an order of magnitude improvement in accuracy and robustness. This demo illustrates each of these both on and off line.

Categories and Subject Descriptors: C.2.m [Computer-Communication Networks]: Miscellaneous; D.4.m [Operating Systems]: Miscellaneous

General Terms: Algorithms, Reliability.

Keywords: Timing, synchronisation, software clock

1. GOALS OF THE DEMO

The demonstration of the TSCclock will be organised along three main themes. We will:

1. Show how it is easy to install and use.
2. Demonstrate it running live, test it interactively against disruptive events and compare its performance with other algorithms running in parallel.
3. Use replays to show long term behaviour, and to zoom into and explore anomalous events.

These themes, elaborated below, together provide a complete 'guide' to how the networking community could incorporate the TSCclock into any application or tool, as well as insight into what it can provide, and how it will behave, under different circumstances.

1.1 The TSCclock runs out of the box

This theme shows that the TSCclock has evolved beyond the initial research and proof-of-concept stages and can now be easily used. We will describe its installation, using packages, for supported Linux and FreeBSD systems. To demonstrate how easy it is, we will offer attendees the possibility of installing, configuring and start the TSCclock on test systems in less than 5 minutes (with a prize for the fastest install).

A clock is useless if it cannot be accessed by applications. The TSCclock package provides a compact API to support the integration into applications, for example precision network monitors (available in C and Python). The API will be described with a poster, HOWTOs, and GUI tools.

1.2 The TSCclock runs live

The second theme of the demonstration is to show the TSCclock running live and coping with real-world conditions. Using the Stratum-1 servers available around Seattle we will demonstrate how the TSCclock effectively turns a commodity PC into a reliable source of timing, and in particular for timestamping network packets.

The design of the TSCclock is non-intrusive, allowing other synchronization algorithms to run in parallel on the host system. We will exploit this to display comparisons against competitors such as the *ntpd* daemon and the *ptpd* daemon, a software implementation of the new IEEE-1588 standard. We will use a DAG card donated by Endace as a reliable external reference.

We will also demonstrate the high quality and stability of the difference clock mode of the TSCclock, which enables RTT's to be measured to less than 1μs even when disconnected from the server for periods of weeks! The competing clocks do not have a difference clock mode.

1.3 TSCclock performance: years of replay

The final theme is to give a survey of TSCclock performance under a variety of conditions, based on virtual replay of many traces collected in our testbed during live operation of the TSCclock at the University of Melbourne. The testbed benefits from Linux and BSD kernels modified to enable accurate clock comparisons, DAG cards, high quality GPS receivers, a Rubidium atomic clock, and extensive analysis software.

We will provide replays accompanied by detailed data analysis allowing the performance of the clock, as well as network and host characteristics and events, to be put under the microscope. Long term average clock performance as well as peculiar and isolated events, and limitations due to the OS and hardware, will be illustrated using servers which are on a LAN, close by, or far away.

2. REFERENCES

[1] J. Ridoux and D. Veitch, "TSCclock Webpage." (http://www.cubinlab.ee.unimelb.edu.au/tscclock/)

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.