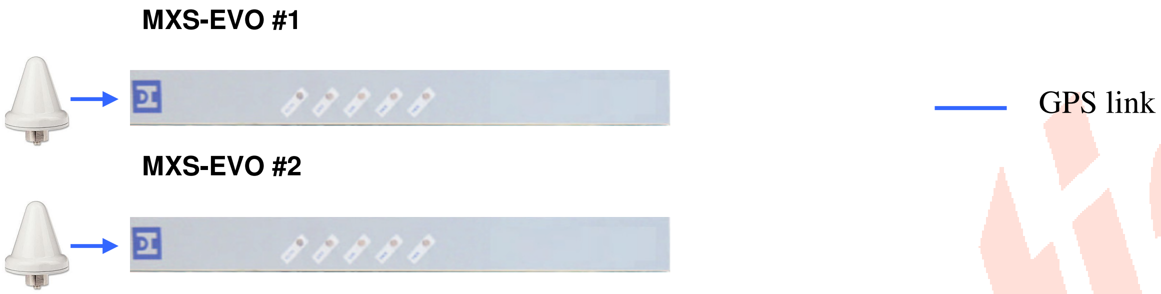
NTP HARDWARE TIMESTAMPING

Goal

- To achieve better performance with existing NTP server hardware

Requirements

- 2x MXS-EVO



Equipment

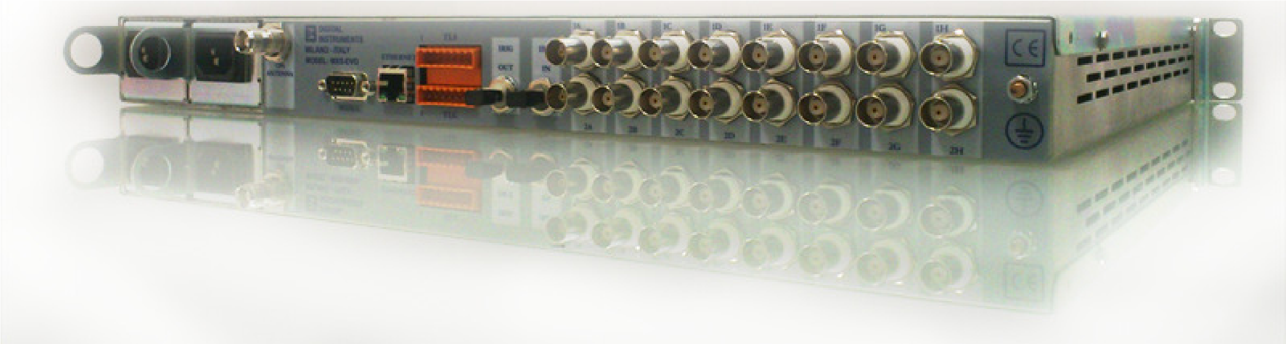
MXS-EVO

MXS-EVO is a multi-output generator for high stability Time & Frequency signals, aimed to synchronization of systems and devices in many areas like Broadcast, Defence, Space, Telecommunication etc.

The unit has 12 programmable outputs designed to make the equipment adaptable to di_erent situation and meet user's needs. Furthermore it has an Ethernet interface for Time Protocol Synchronization (NTP or PTPv2 Grandmaster Clock).

The unit is also capable of two (1 input and 1 output) optical ST connectors designed for IRIG B.

MXS-EVO can get external reference input from GPS receiver, E1/T1, PPS, 1 to 10 MHz analogue, IRIG-B Time code, PTPv2 IEEE 1588-2008, in order to have maximum reliability that is completed by dual independent Power Supply Unit.



Features

- Internal high stability OCXO aging rate of ± $1*10^{-10}$/day
- 12 channels GPS receiver with automatic tracking and timing error management
- New generation DPLL fast lock with holdover
- Multi reference inputs:
  - GPS
  - E1 (G.703/9) or T1
  - PPS
  - 1, 2, 2.048, 5, 10 MHz
  - IRIG-B Time Code
  - PTPv2 (IEEE 1588-2008)
- 1x Fast-Ethernet interface for NTP and/or PTP synchronization
- 1x Optical multimode I/O via ST connectors
- 12x programmable outputs configurable between:
  - PPS
  - IRIG B DCLS
  - IRIG B AM
  - E1 (G.703/9) / T1 with SSM
  - 2.048 MHz (G.703/13)
  - 10 MHz (Low Phase Noise)
- 2x PSU (AC or DC)

Implementation

Software timestamps are usually collected in userland code or, at best, in driver interrupt routine.
In order to reduce the jitter introduced by the operating system timestamps can be collected by dedicated hardware (usually FPGA or ASIC).

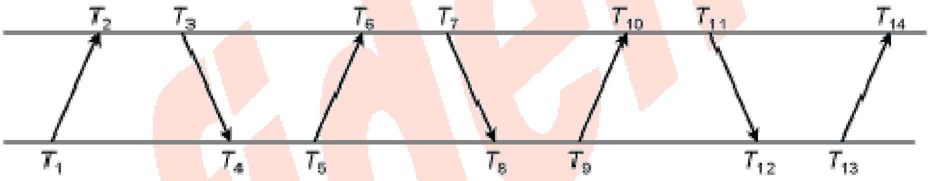
The improvement gain is huge: a precision of tens of nanoseconds instead of tens or hundred of microseconds.
The drawback is that in many cases the sending timestamp is only available after the packet has been delivered and cannot be inserted live in the same packet.

The solution consist to insert the timestamp in the following packet by slightly modifying the NTP on-wire protocol to support an interleaved mode of operation.

This is described in depth in the reference page of *NTP Interleaved Modes*[1] and in the *Analysis and Simulation of the NTP On-Wire Protocols*[2].

In order to achieve this the server is stateful and maintains a table of timestamps delivered to client.

	Sync		Valid		Valid		Valid	
rec	0	0	T_4	T_4	T_8	T_8	T_{12}	
dst	T_2	T_2	T_6	T_6	T_{10}	T_{10}	T_{14}	
aorg	0	T_3	T_3	T_3	T_3	T_{11}	T_{11}	B
borg	0	0	0	T_7	T_7	T_7	T_7	



t_{org}	0	0	T_2	T_4	T_6	T_8	T_{10}	
t_{rec}	0	T_2	T_4	T_6	T_8	T_{10}	T_{12}	
t_{xmt}	0	0	T_1	T_3	T_5	T_7	T_9	

rec	0	T_2	T_2	T_6	T_6	T_{10}	T_{10}	
dst	0	T_4	T_4	T_8	T_8	T_{12}	T_{12}	A
aorg	T_1	T_1	T_1	T_1	T_9	T_9	T_9	
borg	0	0	T_5	T_5	T_5	T_5	T_{13}	
		Sync		Valid		Valid		

Round 1
Round 2
Round 3

The clients should be able to support this interleaved operating mode, that has been introduced in ntpd since version 4.2.6.

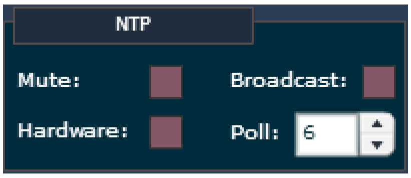
It only works for peer and broadcast mode.

[1] http://www.eecis.udel.edu/~mills/ntp/html/xleave.html
[2] http://www.eecis.udel.edu/~mills/onwire.html

Test case

We would like to measure the performance gain in using NTP hw timestamping.

The first device is configured to use the legacy software timestamping.

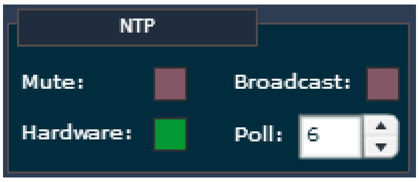


NTP

Mute: 🟥 Broadcast: 🟥

Hardware: 🟥 Poll: 6

The second device is configured to use the new hardware timestamping.



NTP

Mute: 🟥 Broadcast: 🟥

Hardware: 🟩 Poll: 6

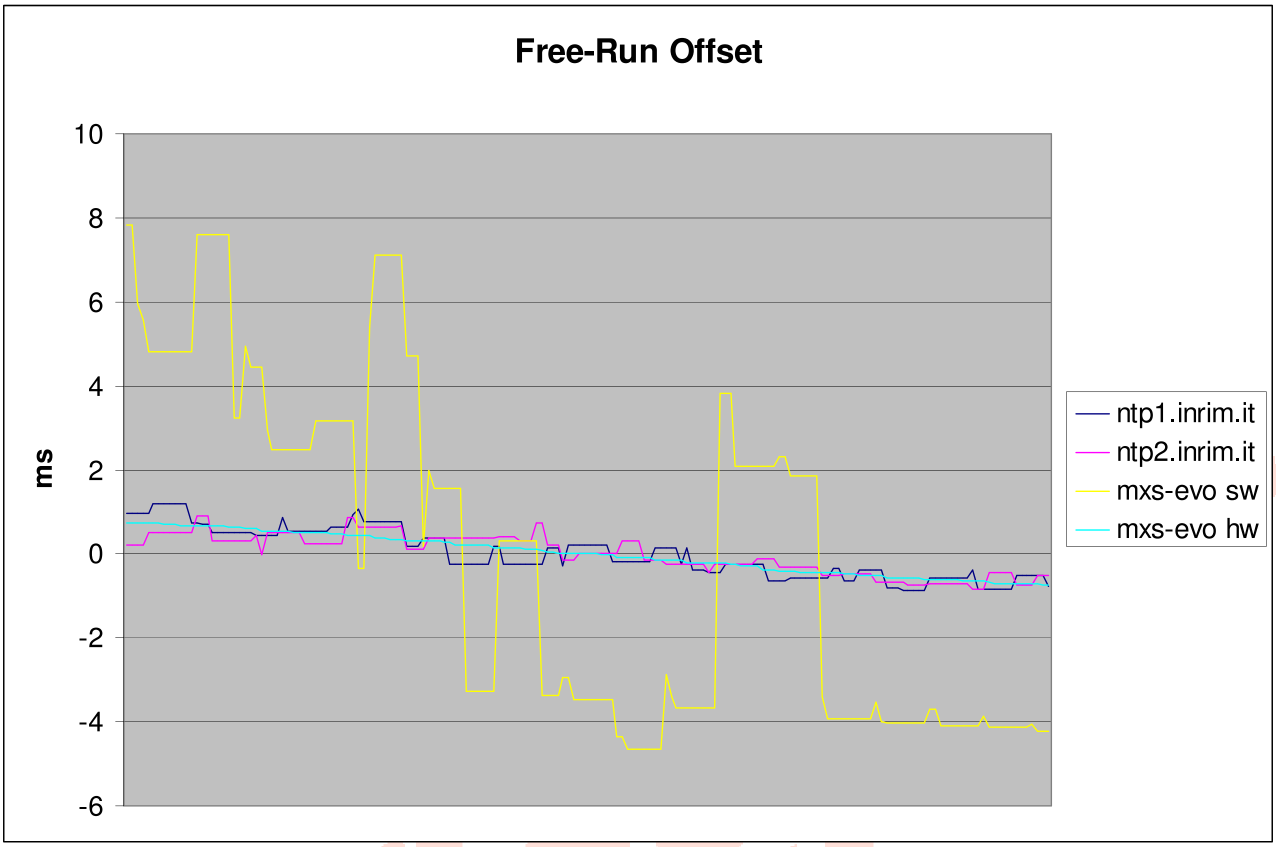
In this test we want to check the NTP accuracy and coherence on the two devices (without disciplining the system clock)

1. We first need to synchronize the system clock with one of the available NTP server
   ```
   ntpdate <server ip>
   ```
2. Then we need to setup a NTP daemon[3]
3. And add a line for every device. It is also a good idea to specify an external public NTP server in order to compare the time stamp with an external source and a local clock in order to let the system clock freely run.
   ```
   ## ntpd.conf
   server 127.127.1.1
   server ntp1.ien.it noselect
   server ntp2.ien.it noselect
   server 192.168.200.184 noselect
   peer 192.168.200.183 xleave noselect
   ```
4. Ask the timing information with `ntpq -c peer`
   ```
        remote           refid      st t when poll reach   delay   offset  jitter
   ==============================================================================
   *LOCAL(1)        .LOCL.          12 l   55   64  377    0.000    0.000   0.000
    ntp1.inrim.it   .CTD.            1 u    4   16  377   52.978   18.862   4.491
    ntp2.inrim.it   .CTD.            1 u    3   16  377   57.822   18.053   0.373
    192.168.200.181 .GPS.            1 u    2   16  377   11.564   14.875  51.929
    192.168.200.183 .GPS.            1 u    3   16  377    0.065   13.558   0.478
   ```

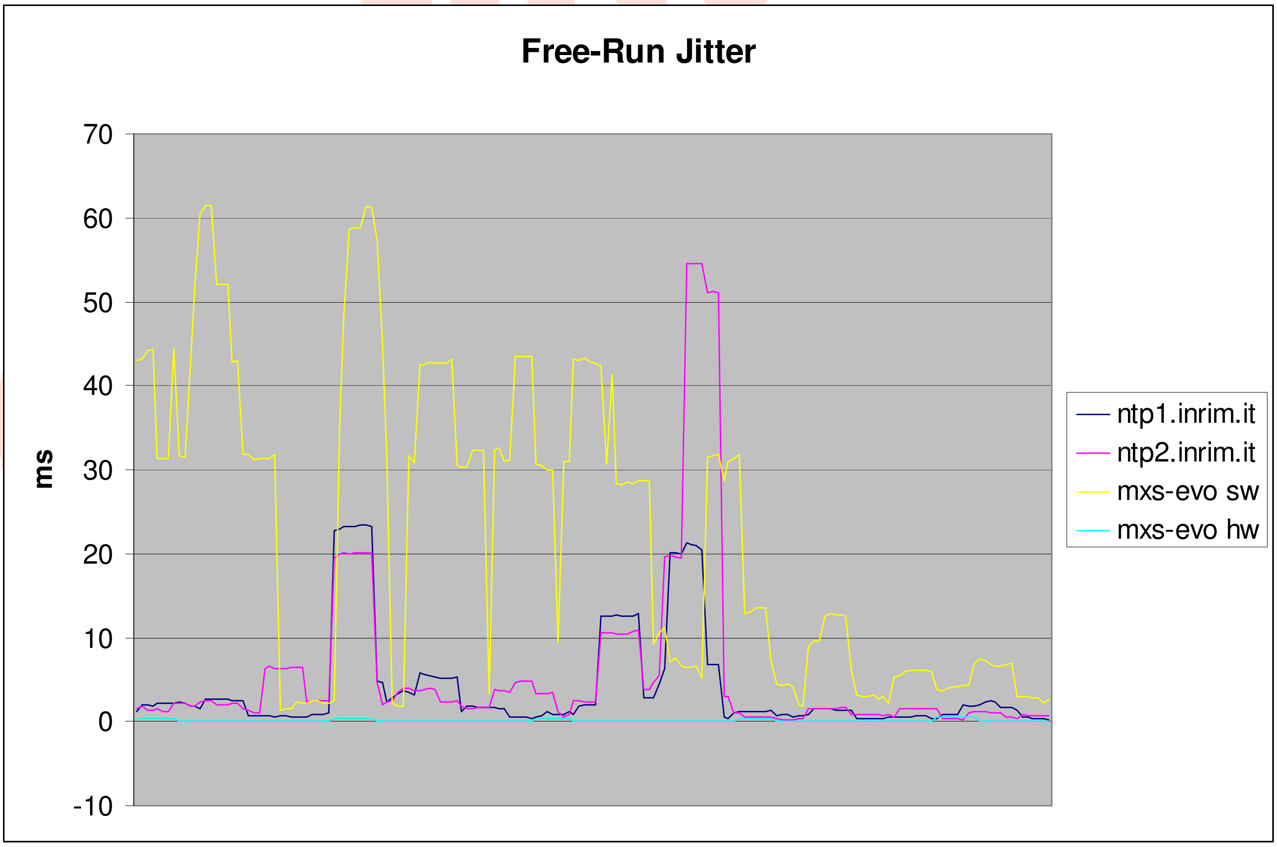
The time stamps should be coherent to each other and to the external servers.

[3] GNU/Linux version can be downloaded from http://www.ntp.org/downloads.html
Microsoft Windows version can be downloaded from http://norloff.org/ntp/ntp-4.2.6p3-RC8.zip

After a few hours the following measures have been carried out.
From the collected offsets the mean value has been removed in order to better compare the jitter.



ntpd also records the measured jitter and this is the result.



After having let the daemon synchronize a bit the offset from the peer reference is reduced below the millisecond.

```
$ ntpq -c peer
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 ntp1.inrim.it   .CTD.            1 u   10   16  377   54.921    4.582   5.738
 ntp2.inrim.it   .CTD.            1 u    8   16  377   56.543    3.343  18.484
 192.168.200.181 .GPS.            1 u   10   16  377   50.970   15.814   3.758
*---.homenet.tel .GPS.            1 u    1   16  377    0.150   -0.549   0.059
```

Conclusions

The NTP hardware time stamping support greatly improves the performance on the server side. With common PC hardware it should be possible to achieve better than 1 ms precision on client side. If this is not achieved you may also consider switching to a more performing hardware, to a newer protocol (like PTP) or modifying the synchronization design (e.g. by distributing stratum-2 miniature PTP/NTP bridges closer to the end devices).

If you would like to test our solutions or ask some questions about these or other subjects please do not hesitate to contact us!

Digital Instruments S.r.l.
Via Parco degli Scout,13
20091 BRESSO (MI) - ITALY
Tel: +39-0266506250
Fax: +39-0266506103
Info e-mail: info@digital-instruments.it
Commercial e-mail: marco.muritti@digital-instruments.it
http://www.digital-instruments.it