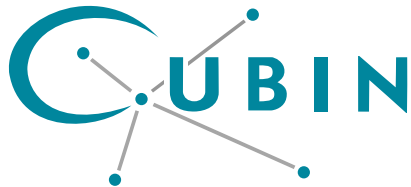

A Methodology for Clock Benchmarking

Julien Ridoux j.ridoux@ee.unimelb.edu.au

Darryl Veitch d.veitch@ee.unimelb.edu.au



**ARC Special Research Centre for
Ultra-Broadband Information Networks
THE UNIVERSITY OF MELBOURNE**

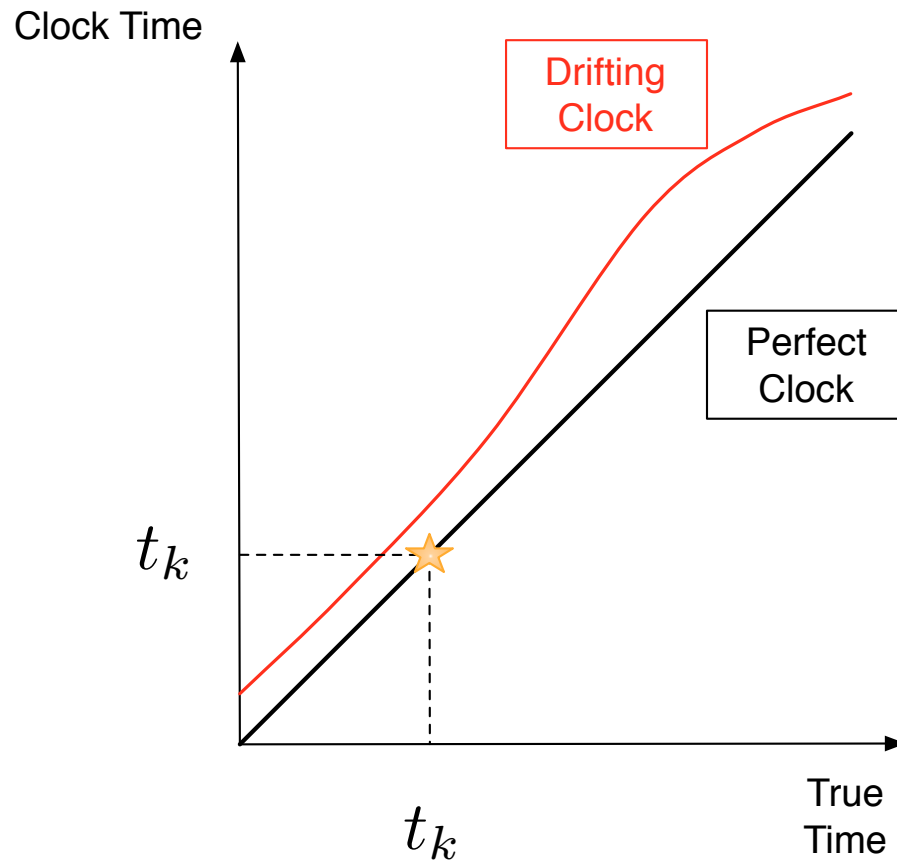
**NICTA Victoria Research Laboratory
Dept. of Electrical & Electronic Engineering
THE UNIVERSITY OF MELBOURNE**

Introduction

- **Higher demand on the network, better clocks**
 - Network applications are more and more distributed
 - Users/Providers need higher reactivity and more precision
 - Essential for testbeds and performance evaluations
- **Limitation: the quality of clock synchronisation**
 - Need for higher accuracy
 - Need for higher reliability
- **Difficult to benchmark timekeeping systems**
 - Against which reference?
 - How to access clock instantaneously?

Clock and Timestamping errors

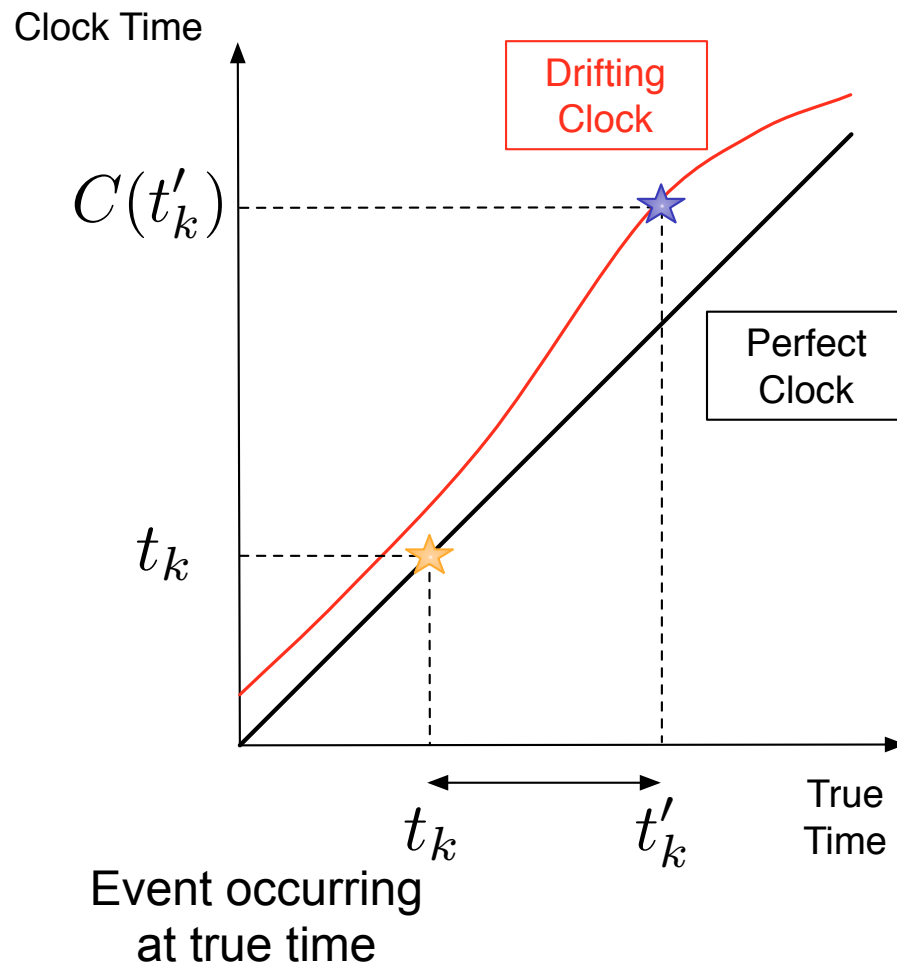
Timestamps



Event occurring
at true time t_k

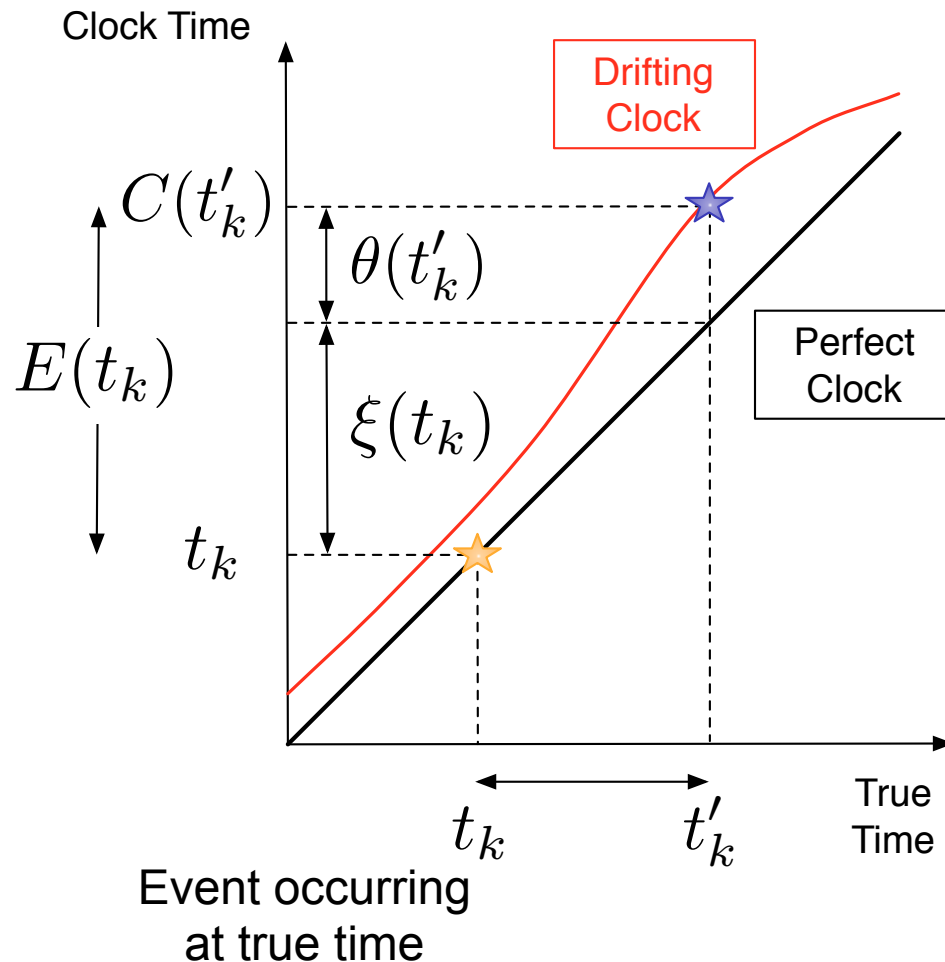
Clock and Timestamping errors

Timestamps



Clock and Timestamping errors

Timestamps



- **Clock error or offset**

$$\theta(t_k) = C(t_k) - t_k$$

- **Timestamping error**

$$\xi(t_k) = t'_k - t_k$$

- **Total error:**

$$\begin{aligned} E(t'_k) &= C(t'_k) - t_k \\ &= \theta(t'_k) + \xi(t_k) \end{aligned}$$

Clock and Timestamping errors

- **In practice, no perfect clock for benchmarking**
- **Total relative error:**

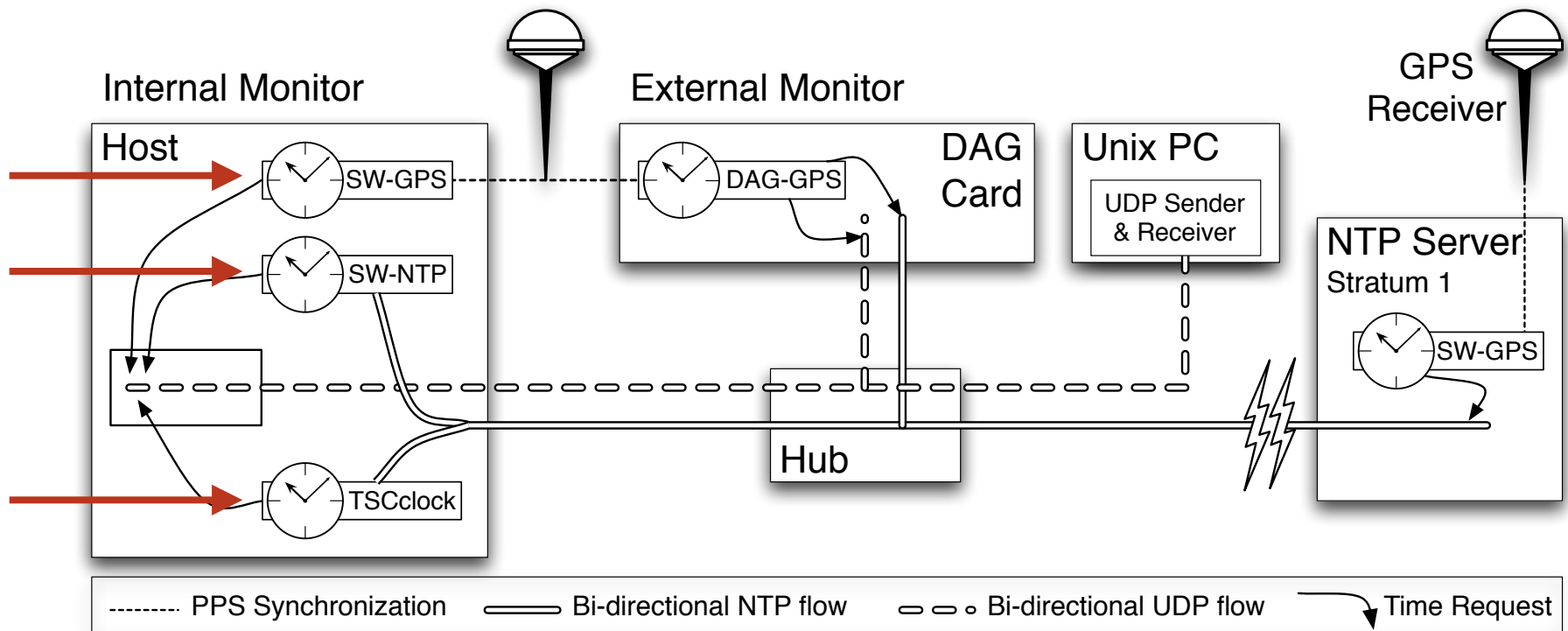
$$\begin{aligned} E_{C_1, C_2}(t_k) &= C_1(t'_k) - C_2(t''_k) \\ &= \theta_{C_1}(t'_k) - \theta_{C_2}(t''_k) + \xi_{C_1}(t_k) - \xi_{C_2}(t_k) \end{aligned}$$

- The clock and timestamping errors combine
- **Without a perfect clock; benchmarking a challenge**
 - Timestamping error: eliminate / estimate
 - Clock error: relative / absolute
- **Need a strong methodology**

CubinLab Testbed

■ 3 Clocks under study (Linux & FreeBSD)

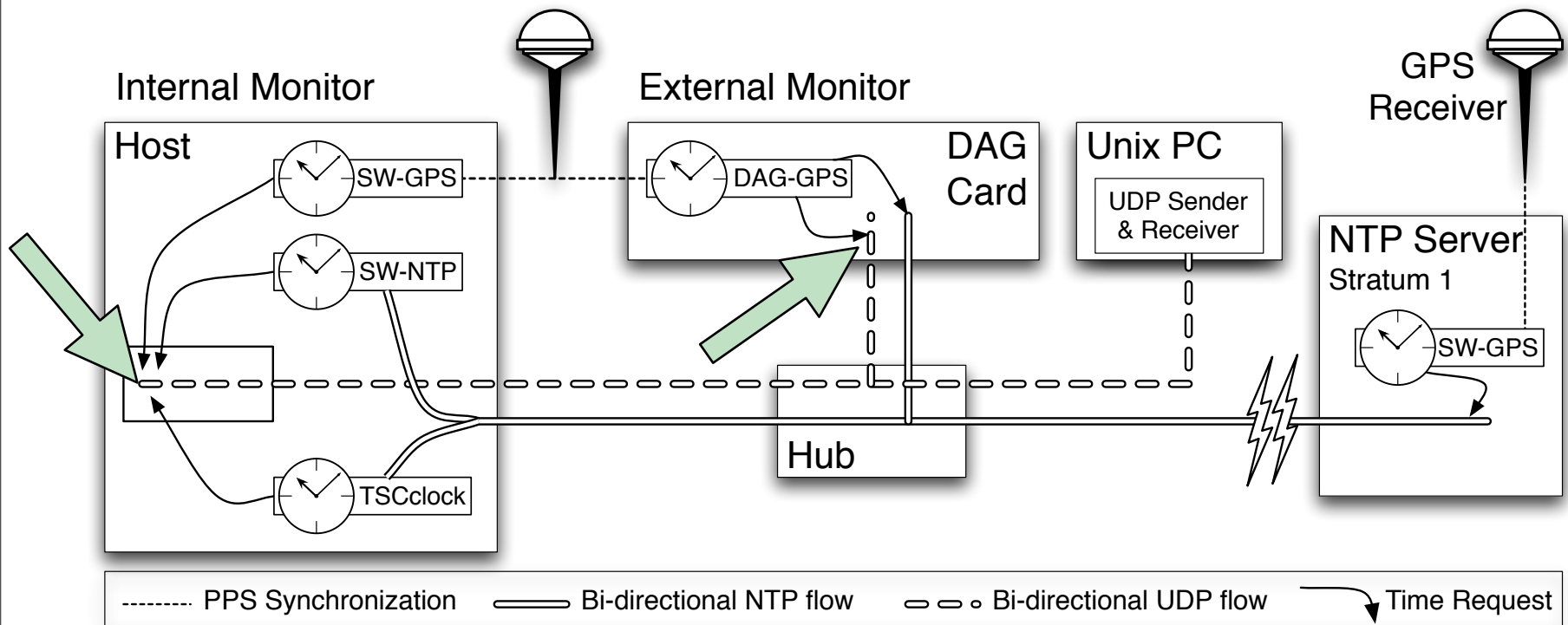
- SW-GPS: ntpd + GPS sync. Absolute Clock
- SW-NTP: ntpd + Net. sync. Absolute Clock
- TSCclock: Net. sync, Absolute & Difference Clock



CubinLab Testbed

Kernel timestamping of UDP packets

- Outgoing / Incoming directions
- External: DAG Card
- Internal: Multiple clocks simultaneously



Internal comparison

- **Modified kernels: timestamps taken “back to back”**

- Identical delay accessing the clocks $\rightarrow t'_k = t''_k$
- Timestamping errors cancel $\rightarrow \xi_{C_1}(t_k) = \xi_{C_2}(t_k)$

- **Obtain a comparison of the two clocks offsets**

$$E_{C_1, C_2}(t_k) \approx \theta_{C_1}(t_k) - \theta_{C_2}(t_k)$$

- Free of timestamping error
- No absolute performance with respect to true time

External comparison

- **Use of the DAG card**

- Considered best absolute time reference available →

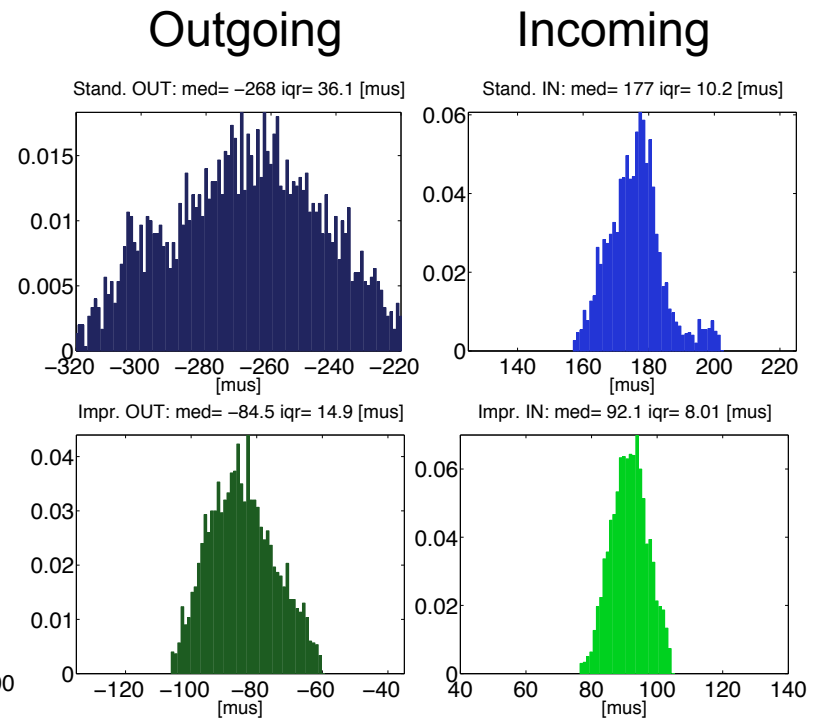
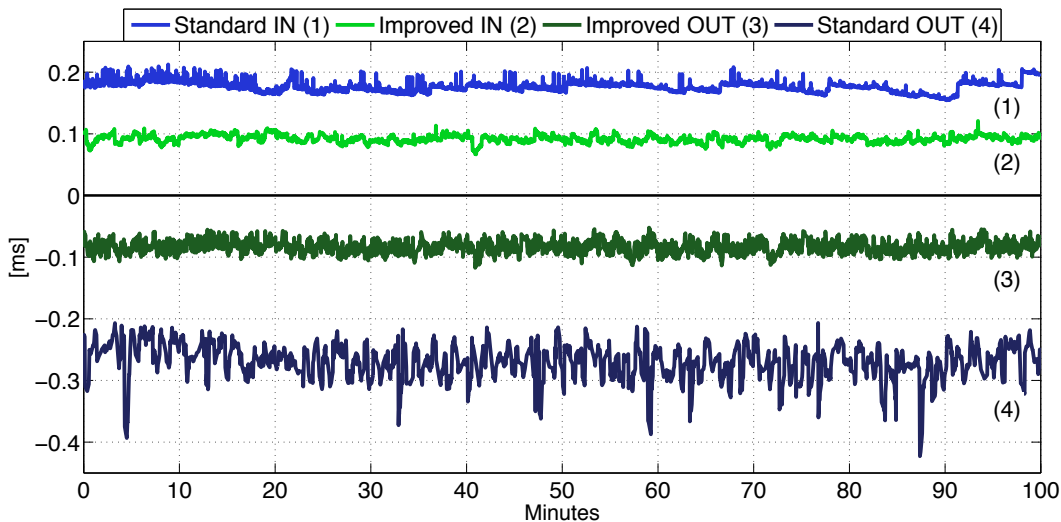
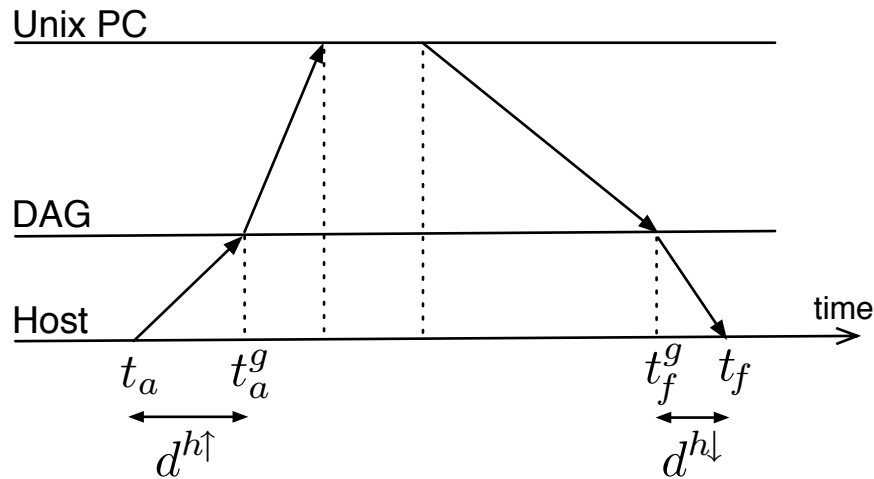
$$E_{C,Dag}(t_k) = \theta_C(t_k) + \xi_C(t_k) - \xi_{Dag}(t_k)$$

- Provides absolute reference
- But suffers from timestamping error

- **Additional kernel modifications to reduce noise**

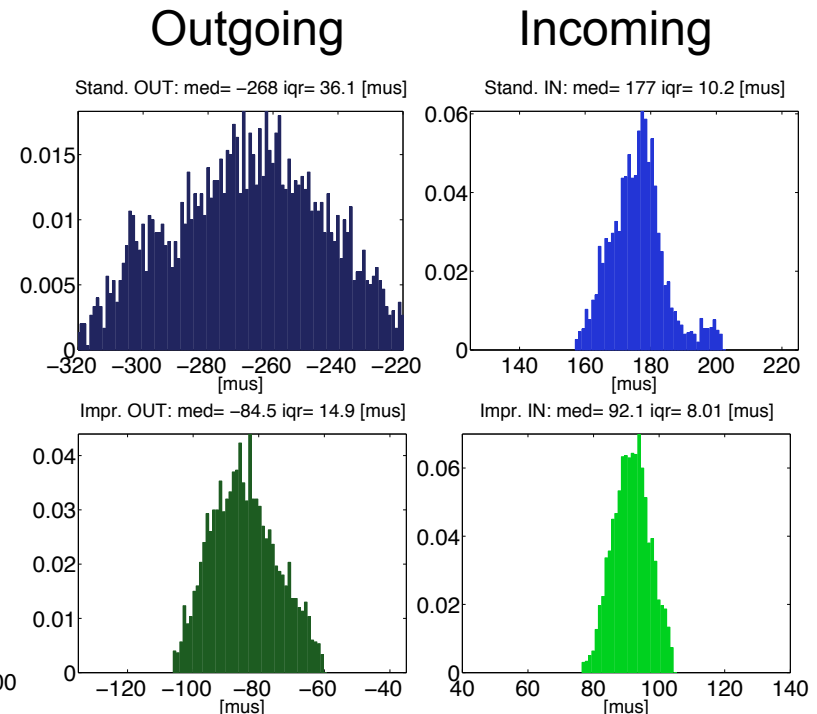
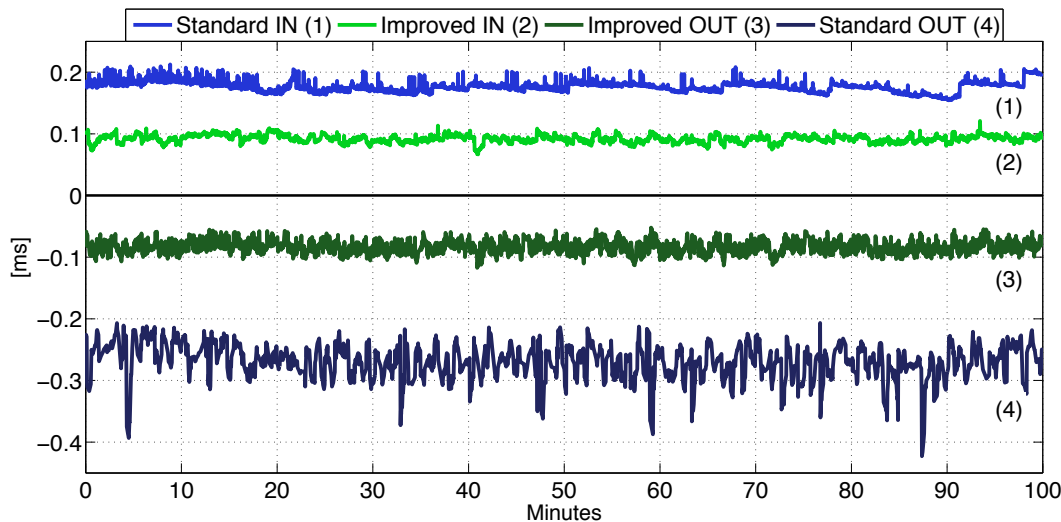
- Standard location in kernel for all clocks
- Improved locations for the TSCclock
 - As close as possible to the last bit transmitted/received
 - Interrupt bottom-half / driver implementation

Reducing kernel timestamping error



Reducing kernel timestamping error

- **Standard timestamping location with the TSCclock**
 - Outgoing direction **noisier**: IQR 100 μs larger
 - **Asymmetry** of 100 μs between Outgoing / Incoming
 - Improved location much better



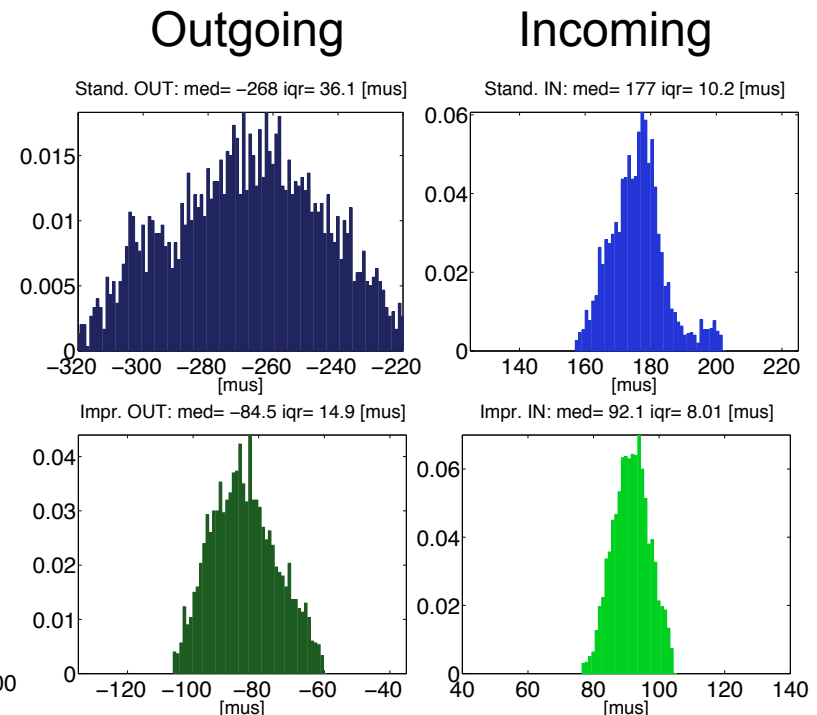
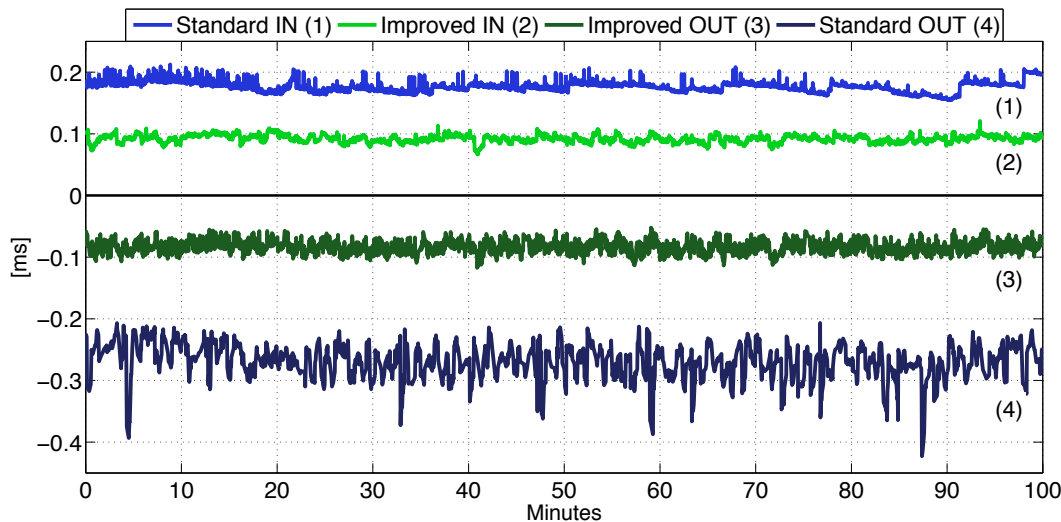
Reducing kernel timestamping error

■ Standard timestamping location with the TSCclock

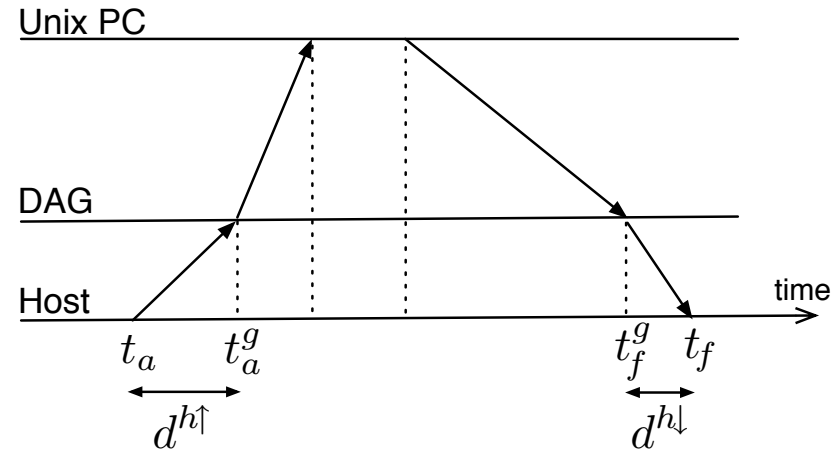
- Outgoing direction **noisier**: IQR 100 μs larger
- **Asymmetry** of 100 μs between Outgoing / Incoming
- Improved location much better

■ The same clock in both directions !!!

- **Which direction to trust?**



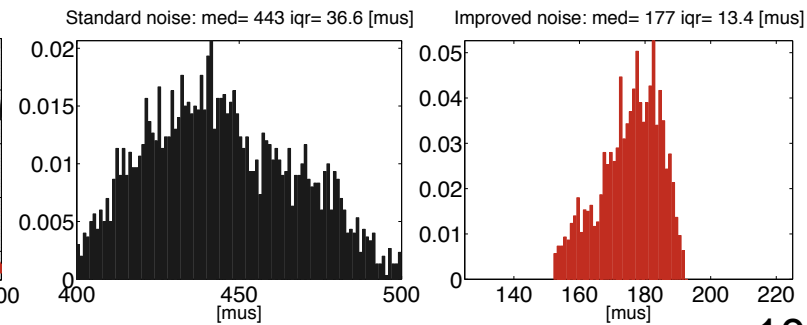
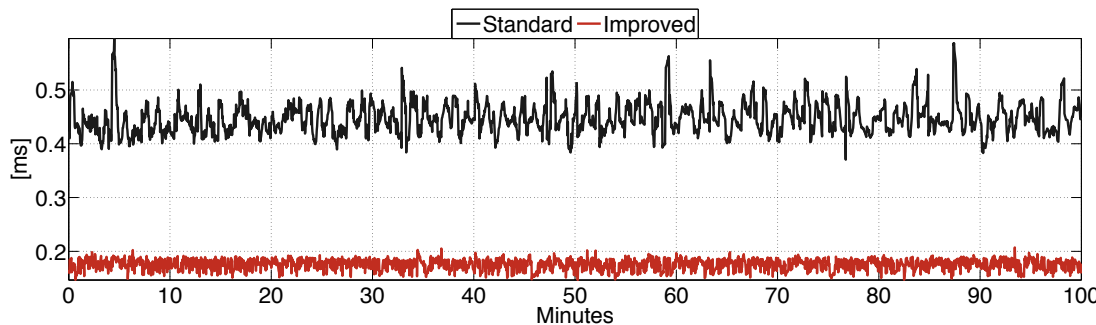
Host RTT measurement



■ If we use both directions

- Minimum Host RTT: $r^h = d^{h\uparrow} + d^{h\downarrow}$
- Available timestamps: $R^h = r^h + \xi(t_f) - \xi(t_a)$
- Host RTT available since measured with the same clock

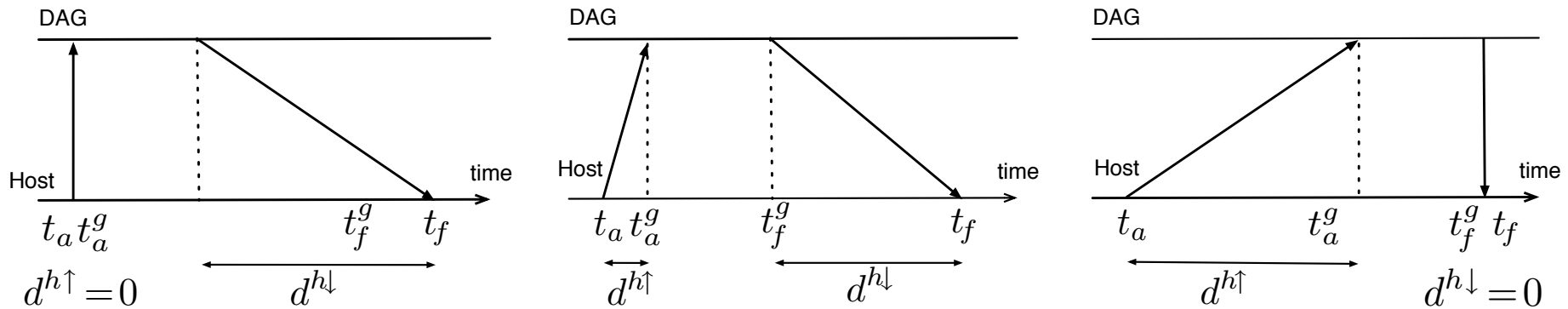
- Minimum can be filtered; noise is the width of histogram of R^h



Recovering one-way measurements

- **Ambiguity due to the asymmetry that we can't evaluate**

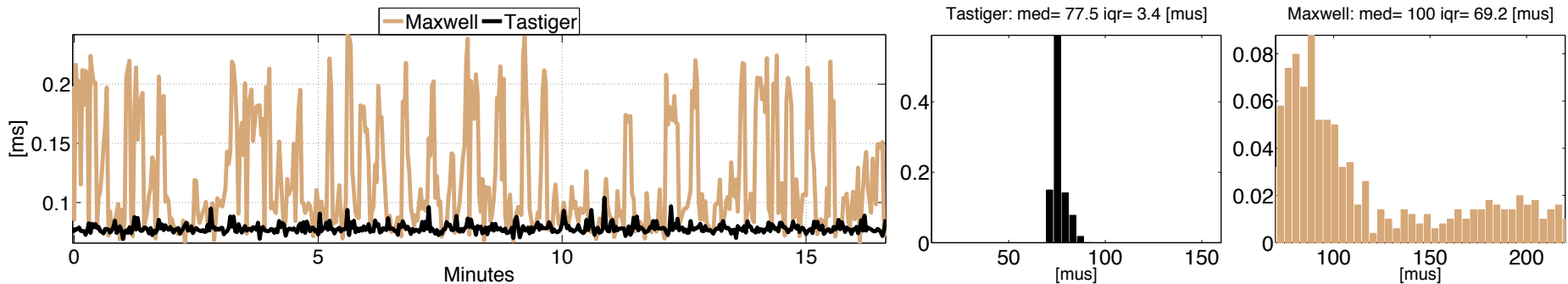
$$asym = d^{h\downarrow} - d^{h\uparrow} \quad asym \in [-r^h, r^h]$$



- **One way delays can't be recovered individually**
- **Host RTT \rightarrow impact of noise on one-way measurement**
 - Median is **ambiguous** but bounded by $2 \times r^h$
 - Histograms are **broadened** because of $IQR(R^h)$

Beware of problematic drivers/NIC

- **Two hosts with same OS / hardware**
 - FreeBSD 6.1
 - Pentium-D architecture
 - But different NIC / Driver
 - Maxwell : Broadcom 5157 Gig-E (Brown)
 - Tastiger : 3Com 10/100 Mbps (Black)



- **Choose carefully!**
 - The quality of R^h measurement drives the accuracy of the methodology!

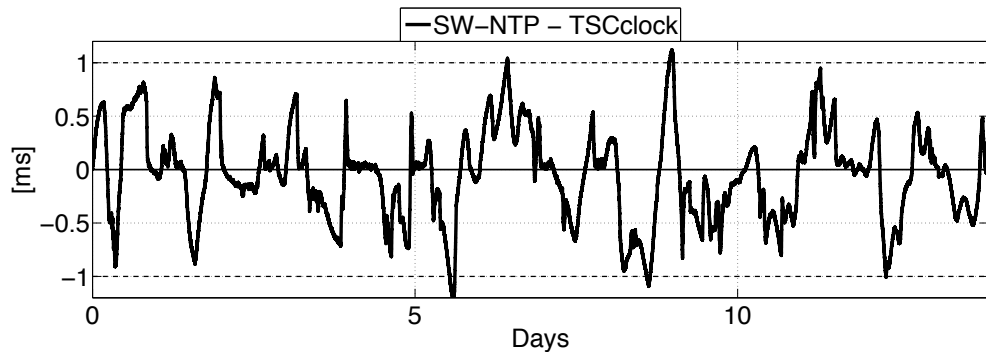
Let's get started

- **Now that we have**
 - an accurate testbed
 - internal / external timestamping and validation
 - improved kernel timestamping
 - removed problematic hardware

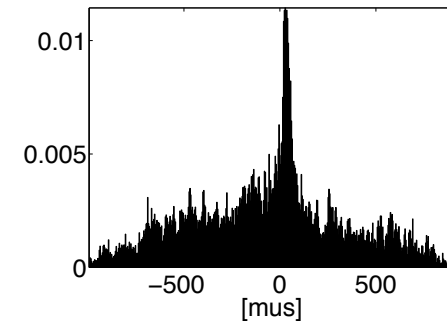
- **... we can start the detective work**

SW-NTP vs. TSCclock

- Internal comparison → large oscillations $\pm 1\text{ms}$

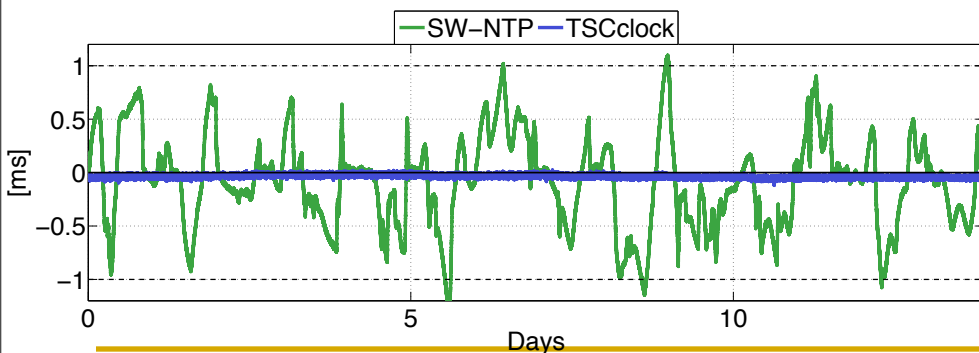


ockdiff: SW - TSC: med= -14.6 iqr= 546 [n

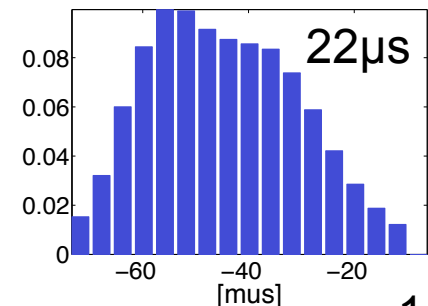
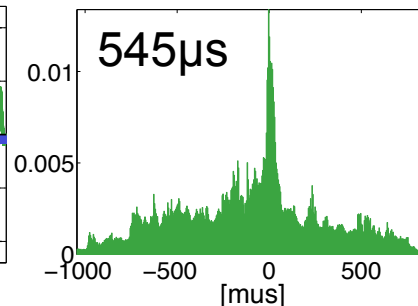


- External comparison → SW-NTP responsible

- Noise: $\text{IQR}(R^h) = 37\mu\text{s}$ $r^h = 10\mu\text{s}$ (ambiguity = $20\mu\text{s}$)
- Accurate view for SW-NTP, difficult diagnosis for TSCclock

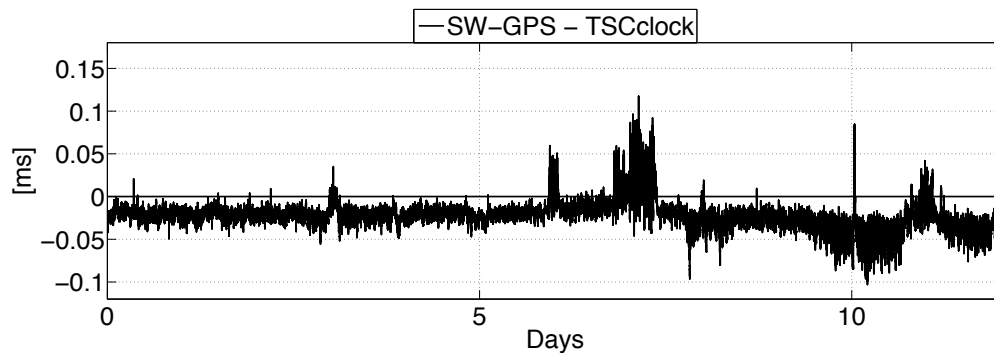


SW-NTP: med= -58.6 iqr= 545 [mus] TSCClock: med= -43.4 iqr= 22.2 [mus]

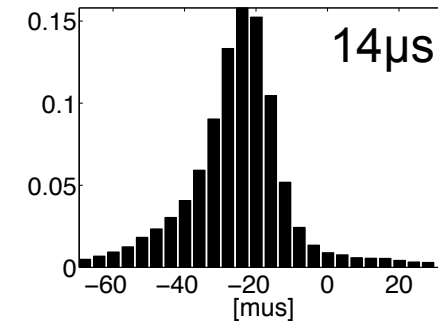


SW-GPS vs. TSCclock

Internal comparison → similar behavior (IQR = 14 μ s)

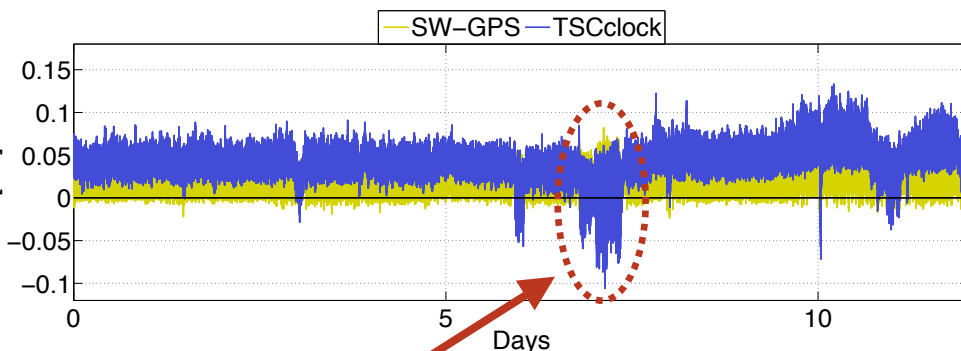


clockdiff: SW - TSC: med= -22.3 iqr= 14.2 [n

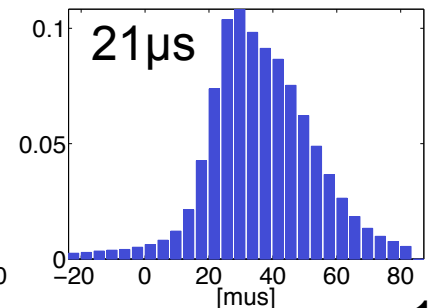
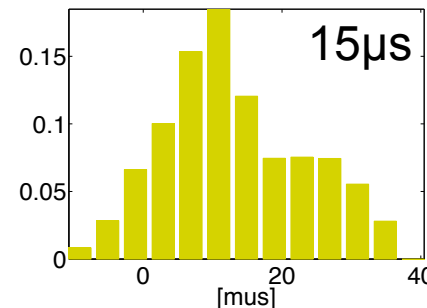


External Comparison

- ❑ Noise: $IQR(R^h) = 23\mu$ s $r^h = 10\mu$ s (ambiguity = 20 μ s)
- ❑ One clock may have (IQR = 0 + noise). But can't be verified!
- ❑ TSCclock slightly ahead but which clock is worse?

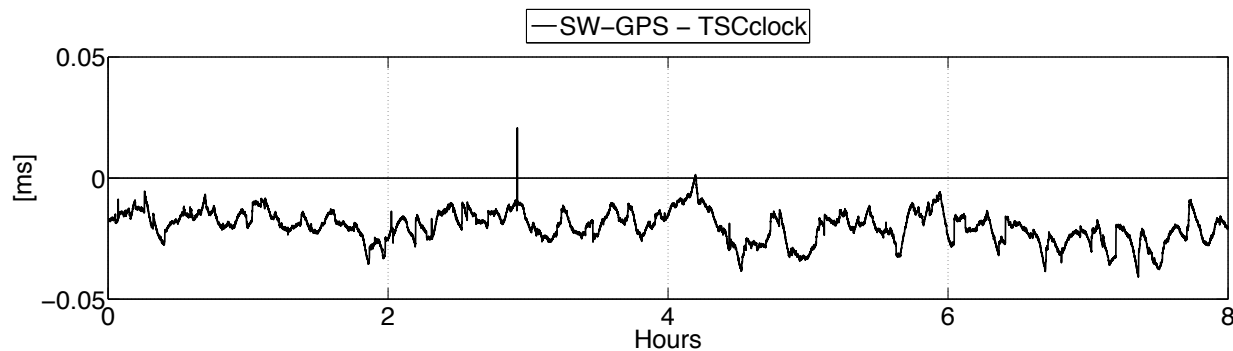


SW-GPS: med= 13.7 iqr= 15.1 [mus] TSCClock: med= 37.4 iqr= 21.2 [mus]

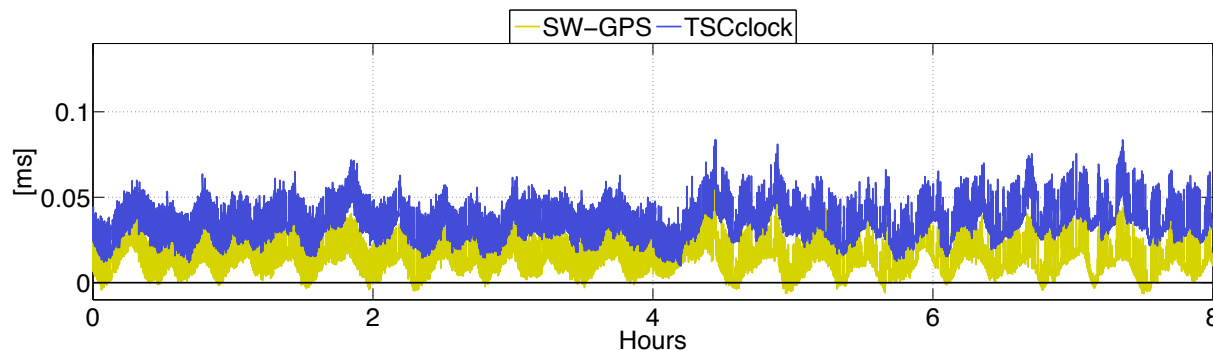


SW-GPS vs. TSCclock (Zoom)

- **Observe SW-GPS and TSCclock more closely**
 - Observe oscillations with a 20mn period

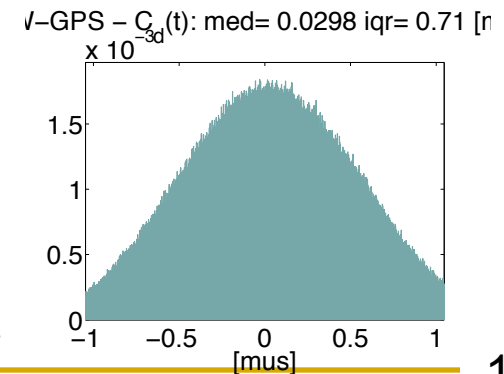
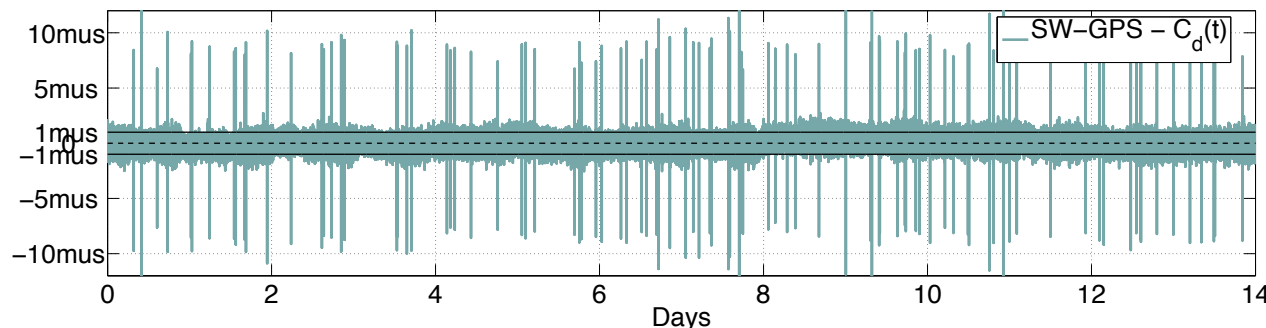


- **Both clocks show oscillations**
- **Temperature effect (air-conditioning variations)**



SW-GPS and (TSCclock) Difference Clock

- **Measure UDP packets inter-arrivals**
 - Compare SW-GPS IAT and TSCclock Difference Clock IAT
- **Final error in a $\pm 1\mu\text{s}$ band**
 - Resolution of SW-GPS is $1\mu\text{s}$ (struct timeval)
 - Can't interpret errors within this band
- **Spikes of up to $10\mu\text{s}$ magnitude**
 - By construction, can't be due to the difference clock
 - Small time scale stability of the oscillator
 - Due to the SW-GPS clock!!



Conclusion

■ **Clock benchmarking**

- ❑ is a challenge
- ❑ requires good quality hardware
- ❑ a strong methodology
- ❑ requires rigour and attention to details

■ **Our methodology and testbed**

- ❑ highlights the need for kernel modifications
- ❑ presents Internal / External complementary comparisons
- ❑ provides comparison down to the system clock resolution
- ❑ allows to track causes of observed strange behaviors

j.ridoux@ee.unimelb.edu.au

<http://www.cubinlab.ee.unimelb.edu.au>