Using a multi-source NTP watchdog to increase the robustness of PTPv2 in Financial Industry networks

Pedro V. Estrela
IMC Financial Markets
Amsterdam, Netherlands
pedro.estrela@imc.nl

Sebastian Neusüß
Deutsche Börse AG
Frankfurt, Germany
Sebastian.Neusuess@deutsche-boerse.com

Wojciech Owczarek
NYSE Euronext
Belfast, UK
wowczarek@nyx.com

Abstract — This paper describes a fundamental single point of failure in the PTPv2 protocol that affects its robustness to failure in specific error scenarios. The architecture design of electing a single unique time source to a PTP domain – the PTP GrandMaster – makes this protocol vulnerable to byzantine failures.

Previous work has described this vulnerability from both a theoretical and practical point of view - and in particular how this affects the financial industry. This paper advances the discussion by contributing a description of the latest high-accuracy regulatory requirements on the financial industry, and by documenting new examples of failures in real-world customer-facing operations. It then describes an example of one of possible ways to increase PTP robustness while preserving its accuracy (using a multi-source NTP watchdog), and a laboratory test that shows how different protocol implementations are affected by this problem.

In all, the current paper attempts to raise awareness of the robustness requirements within the financial industry today. As only PTP is accurate enough for both current and upcoming regulatory requirements, we hope that these issues are addressed in the forthcoming PTPv3 protocol, by adding multi-time source querying capabilities at the PTP end-slaves themselves.

Keywords: PTP, Financial Industry, Byzantine robustness, regulation, IEEE 1588

I. INTRODUCTION

Under pressure to comply with the new regulatory and performance requirements, the financial industry has performed major upgrades to their time synchronization. In this process it replaced the tried-and-tested NTP protocol with newer PTP solutions, because PTP offers much higher *actual* accuracy than NTP.

We believe that PTP is the future of network time synchronization because is the only solution *today* that supports time-aware functionality on the *switches* themselves (i.e., boundary clocks and transparent clocks). This enables symmetric paths in packet-switched networks by removing the buffering queuing effects on the clock disciplining feedback loops. In our view, this explains why the majority of the time industry's innovation is currently focused on PTP.

However, there have been recurrent reports that PTP is not as robust as NTP. In particular, PTP goes to great efforts to keep a *single* active time source for the whole PTP domain known as the Grandmaster (GM). This introduces a fundamental single point of failure that renders this protocol vulnerable to "byzantine failures" – the worst possible class of failures where failing GMs do not shutdown, but instead start to send misleading time information to their slaves.

Previous work has described this exact vulnerability from both a theoretical [2] and practical point of view [3] - and in particular how this affects the financial industry [4].

To advance the discussion, this paper makes the following contributions:

- a description of the latest regulatory requirements that are pushing higher accuracy obligations to the financial industry ([1] / [13] / [15])
- a description of new examples of failures in real-world customer-facing operations [10]
- an example of one of the possible ways to increase PTP robustness while preserving its accuracy (using a multi-source NTP watchdog to prevent failure scenarios)
- a laboratory test that shows how different protocol implementations are affected by this problem

In all, the current paper attempts to raise awareness on the robustness requirements that the financial industry are facing today. As only PTP is accurate enough for both current and upcoming regulatory requirements, we hope that these issues are addressed on the forthcoming PTPv3 protocol, by adding multi-time source querying capabilities at the PTP end-slaves themselves.

TABLE I
A SUMMARY OF THE ACRONYMS USED IN THIS PAPER

BC	Boundary Clock
BMC	Best Master Clock
CAT	Consolidated Audit Trail
FINRA	Financial Industry Regulatory Authority
GM	GrandMaster
NBBO	National Best Bid / Offer
NTP	Network Time Protocol
PTP	Precision Time Protocol
SPOF	Single point of failure
UTC	Universal Coordinated Time
TC	Transparent Clock

II. FINANCIAL INDUSTRY TIMEKEEPING SUMMARY

A. Regulatory / Compliance requirements

Initially, the regulatory norms for the financial industry required clocks to be synchronized up to one second of UTC (OATS FINRA rule 7430 [1]). This rule is obsolete, as it references "wall clocks" and manual synchronization to be performed "every business day before the market open".

Recently these rules have been tightened substantially. In 2012, the Foresight Committee, a group analyzing the requirements of computer-based trading for the UK government, concluded that "*Legislators and regulators should consider implementing accurate, high resolution, synchronized timestamps because this could act as a key enabling tool for analysis of financial markets*" (page 16 of [13]).

More recently, the USA Securities Exchange Commission (SEC) has conducted a multi-year study on how to build a Consolidated Audit Trail between the fragmented USA exchanges that contribute to the NBBO (National Best Bid / Offer). The outcome is Rule 613 (d) (3) [15], which require all market participants (both the exchanges and the participating members) to synchronize their clocks "consistent with industry standards", and to have timestamps "with at least millisecond precision".[1]

B. Specific PTP solutions for the Financial markets

As described in [3] [19], the initial time distribution systems for the financial markets were based on NTP, or older solutions like IRIG. Nevertheless, today this industry has now mostly upgraded their time synchronization to microsecond-accurate PTP. This became possible because all the major network vendors now support PTP in their equipment, so the market participants can now efficiently distribute network time on the same network that they already have for general-purpose data.

In addition, there are now specific offerings to further simplify receiving UTC time, all based on PTPv2:

In the UK, a new service called NPLTime [11], proposed by the National Physics Laboratory, is able to bypass the well-known GPS spoofing vulnerabilities [16] by sending UTC (NPL) time directly from the Atomic clock at Teddingdon into the Financial City data-centers, via a fiber optical link. This whole setup is based on PTPv2 boundary clocks. In the US, a similar setup to UTC (NIST) was proposed by Perseus Telecom [12], named "High Precision Time".

On the other hand, leading exchanges enable PTP time synchronization from their systems towards the market participants, for these to synchronize their clocks against the exchange. Some examples will be given in the below sections.

C. IMC PTP usage

IMC Financial Markets is a leading proprietary trading firm and a key market maker in various products listed on all the world's major exchanges, with offices in Amsterdam, Zug, Chicago and Sydney.

Our company provides liquidity to the users of the equities (e.g., stocks and indices) and derivatives markets (e.g., futures and options). In this business, IMC vigorously competes with other similar companies to provide the best prices to the other market participants (e.g. pension funds, investment houses, banks and individual end-users). The competitive nature of our business drives a strong focus on technology, compounded by the regulatory and compliance requirements demanded of financial market participants.

IMC has built and continuously maintains a state-of-the-art technological infrastructure that keeps us competitive. In particular, IMC features a global network that directly connects to over 40 exchanges worldwide, in all major financial locations and spanning all time zones. This network has dozens of data-centers (DCs), either co-located or in proximity to the financial exchanges, all with state-of-the-art switching backbones and inter-connected with a variety of both leased and partially shared high-speed interconnection lines.

Together, all these DCs host thousands of servers, most of them performing critical real-time activities. All these servers require strict traceability of their clocks to UTC for both current and anticipated compliance reasons, for risk mitigation and for internal performance testing. All trading servers and underlying network equipment were upgraded from NTP to PTP over a two-year period, between 2010 and 2012. The challenges found during this work were detailed in a previous 2012 paper [3].

D. Eurex PTP usage

Eurex is one of the world's leading derivatives exchanges offering a broad range of international benchmark products, operating the most liquid fixed income markets in the world.

Eurex has a long history of offering its clients the utmost transparency which also includes its latency figures [5]. On the latest "T7" system, each order entered into the system carries 7 timestamps taken at various points in the message flow. The accuracy and synchronicity of these times is of greatest importance to the exchange and its customers (see also [18]).

Therefore a dedicated PTP network has been built serving numerous clients with high precision time. Where necessary, PTP clients incorporate hardware timestamping and stable oscillators to guarantee highest precision. Great effort has been made to closely monitor the quality of time synchronization, especially after the exchange had to deal with a leap second bug in its PTP Grandmaster [10].

---

[1] From page 339 of [15]:
"*(d) Clock Synchronization and Time Stamps.*
(1) (...) to synchronize its business clocks that are used for the purposes of recording the date and time of any reportable event that must be reported pursuant to this section to the time maintained by the National Institute of Standards and Technology, ***consistent with industry standards;***
(3) (...) to utilize the time stamps required by paragraph (c)(7) of this section, with at minimum the granularity set forth in the national market system plan submitted pursuant to this section, which shall ***reflect current industry standards and be at least to the millisecond.****"*

Eurex exchange makes use of redundant PTP grandmasters installed in multiple locations. These are driven by independent time signals (GPS, DCF77 [17] and NTP). All three sources are continuously monitored. Additionally alerts have been put into place that report time differences of 10 microseconds or more on application (client) level. Even though the vulnerability against byzantine failures is not completely eradicated, having this monitoring guarantees timely reaction to such events.

The incorporation of robustness against these failures within the PTP layer will be of great benefit. Additionally an extension of PTP to enable central monitoring of all clients in a time domain would make proprietary efforts to continuously monitor each and every PTP client partly unnecessary.

E. NYSE Euronext PTP usage

NYSE Euronext is one of the world's leading exchange operators and market access providers. NYSE Euronext has been offering PTP as a service to their co-location customers since 2012 and has deployed PTP internally across the matching engine clusters and other environments. NYSE is a heavily regulated exchange and has set an initial timing precision target to sub-100 microseconds, with a long term goal to improve it further to 10 microsecond range or better.

Some of their PTP deployment challenges have been described in [4], being similar to those presented in [3] and in the current paper. NYSE uses hardware PTP GMs and software PTP slaves based on PTPd. The slaves are protected from clock jumps (including the leap second failure events) by means of a modified PTPd version supporting both NTP failover, and ignoring the UTC validity flags (instead, it always respect the previous announced UTC offset). This is one of the several features that have been contributed back to PTPd and included in version 2.3.0.

NYSE is also in the process of developing an internal time arbitration solution to protect against other byzantine failure scenarios. More generally, NYSE is a participant in the IEEE 1588 standard work and an advocate of improving timing resiliency by influencing the standard. Currently, simultaneous use of multiple domains is one of the options being looked into for the next IEEE 1588 revision.

III. PTP SINGLE-MASTER VULNERABILITIES TO BYZANTINE FAULTS

A. Failure descriptions

Section V.B of the IMC paper [3] described one specific example where a leap second error in the PTP GM resulted in failure of the time accuracy, causing the cautionary shutdown of hundreds of trading applications.

In short, on several occasions a GM bug caused the (single) PTP time source to send time without leap seconds information, for several hours. As the active GM continued to send "Announce" packets as normal, with the same BMC parameters (in particular priority, clock class and variance), the inactive GMs had no reason to take over. All clients, however, saw a 34 second offset without any indication that this time might be invalid or suspicious in any way. Consequently, they either "corrected" this situation by stepping (jumping) the clock backwards, or by slewing (slowing down) the clock at maximum speed to the "new" value.

In the period following the IMC paper, other market participants have informally confirmed that they randomly experienced the same problem as well. However, up until today the worst reported problem happened at the Eurex Exchange on 26th August 2013, as it affected *all* their hundreds of market participants [10]. In this case, Eurex had to postpone their market opening, which occurs every day at 9:00 CET, because their critical systems were in the wrong time due to the same leap second problem.[2]

B. Multi-source slaves

Even though the specific issues were corrected in cooperation with the vendors, unfortunately we believe that new instances of the problem will continue to appear. We consider that the root cause lies in the PTPv2 standard *itself:* the standard is vulnerable to byzantine failures, so it affects any PTPv2 implementation in which clients trust a single time source.

This very fact – single-source time synchronization protocols suffer from having a single point of failure – was proven mathematically in 1997 by a paper by Fetzer and Cristian [2], which states that "*we prove that at least 2F+1 reference time servers are necessary for achieving external clock synchronization when up to F reference time servers can suffer arbitrary failures*".

Thus, to address Byzantine failures, where "traitors" actively send the worst possible misleading time - for whatever reason - it is imperative that the clients *themselves* are able to listen and reply to multiple sources simultaneously, using a minimum of 2*T+1 reference time servers to successfully absorb T "traitors".[3]

We also believe that having an accurate, but robust multi-source solution will also substantially improve safety to malicious attacks. The most common are the spoofing of GPS time sources, which has been described in general in [16], and in particular for financial services in [14].

IV. EXAMPLE SOLUTION USING AN NTP WATCHDOG

This section describes an example of one of the possible ways to increase PTP robustness to the byzantine failure problems described in the previous section – but keeping the accuracy already provided by PTP in non-failure scenarios. It was developed during the IMC research work performed in 2010/11 [9], and it focus on minimizing the changes to the

[2] From [10], *"'An incorrect time synchronization within the system' triggered the market halt. The problem was solved, pre-trading started at 07:20 GMT and, as of 07:30 GMT all products were again tradable, the exchange said"*

[3] In a simple metaphor, *"A person with one clock believes to know what time it is; A person with two clocks is never sure; A person with three clocks is way surer than the previous two."*

already existing PTP implementations. It should be noted that this solution is only intended to be illustrative, to show that any robust PTPv3 solution will require adding multi-time source querying capabilities at the PTP end-slaves themselves.

A. *Multi-PTP multicast watchdog*

The initial idea was to force all internal PTP GMs to be active simultaneously. As there was no client unicast support at the time, this would require all the globally distributed devices to be isolated in different PTP domains, because they all share the same multicast group [9]. Then, we could have modified the PTPd open-source client [6] to support multiple PTP state machines simultaneously, and somehow combine the results into a single robust offset to be applied to the slave clocks.

However, the very poor maturity of the PTP implementations implied that only a combination of different GMs from *different* vendors would result in a truly robust solution. If we combine this with the quite high requirements and complexity of PTP itself (especially the problem of having thousands of machines continuously multicasting their PTP messages to a shared multicast group spanning the entire globe, effectively implementing a global broadcast system [3] [4]), then it became clear that such solution would have a very high associated cost.

B. *Multi-NTP unicast watchdog*

Instead, by noting that we would only want a solution to cover the failure scenarios with *fair* robustness, it was recognized that a *degraded* robust time could be achieved using the far simpler – and far more mature – NTP unicast protocol.

For this, open source PTPd source code was changed[4] to include a new NTP state machine that independently calculates the current robust offset from multiple dispersed NTP servers. While this still requires the usage of different NTP vendors, the availability of thousands of public NTP servers on the Internet substantially lowers the cost of this solution.

The integration of the two state machines is outlined in Figure 1. The NTP state machine periodically queries the multiple servers specified from /etc/ntp.conf, up to the limits specified using minpoll / maxpoll configs, and calculates the estimated offsets from every server independently.

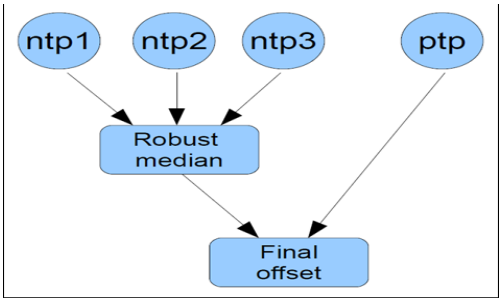

Figure 1. NTP and PTP state machines running in parallel

[4] To promote discussion, the code will be made public as a patch to the latest PTPd implementation, available on the main author's academic site

Then, the found offsets are combined using the median() function. This function is robust to outliers as it separates the higher half of a data sample from its lower half. For this example implementation we choose not to implement the full NTP algorithms (ie, falsetickers / truechimers support [8]), because the median function is enough to add a robustness watchdog to the already highly accurate PTP operations. Further work could be considered to extend the median with estimated confidence intervals, by using the Marzullo / Intersection algorithm of NTP [8].

On the other hand, the PTP state machine operates in parallel using the normal way, and determines its very accurate (but potentially unrobust) offset value continuously.

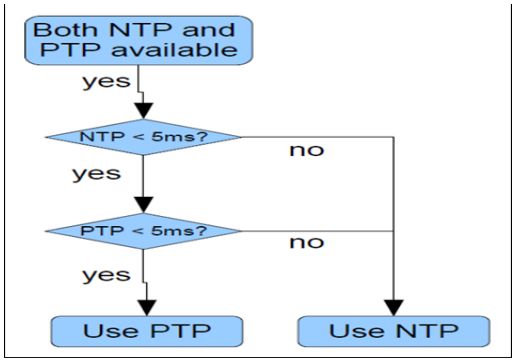

Figure 2. Decision tree to choose PTP or NTP offsets

Figure 2 describes the only point where the two state machines communicate. Here, the clock disciplining servo [6] chooses which offset to apply: either the accurate PTP one, or the robust NTP one. The algorithm works as follows:

- If the NTP side considers the clock to be far away from UTC, (i.e. "far" being higher than a config value, like e.g. 5 milliseconds), then NTP fully controls the clock by applying its own offset. In this case the offset from PTP is ignored, regardless if it is accurate and/or robust.
  The config value is selectable according to the distance to the time servers, and enables a trade-off between the expected NTP accuracy and its provided robustness. A hysteresis effect is added by adding a +-10% uniform offset on every usage, which avoids possible edge behavior around this config value.
- If NTP considers the clock to be reasonably close to UTC (i.e., the robust median offset is lower than the config value), then the PTP value is checked as well against the config value. If both PTP and NTP agree on this offset to be small, then PTP is allowed to control the clock, and accurately drive the offset to sub-microsecond values.
- The interesting case is when NTP and PTP have differing opinions: eg NTP robustly found a small offset, but PTP contradicts this (perhaps because of the leap second problem). In this case NTP continues to control the clock and ignores the PTP offset completely.
  This results in a degraded lower-accuracy operation (but still expected to be sub-millisecond), but which is fully

robust to byzantine errors that would easily create inaccuracies of thousands of milliseconds, or more.

V. EXPERIMENTAL TESTS

An experimental test was initially carried out in late 2011 to show the effects of having no multi-source features in PTP. This was repeated in 2014 to confirm the behavior with the latest version of the open source PTPd client.

The testbed, described in Figure 3, comprises four machines running standard Linux versions, which will have their kernel clocks disciplined in various ways. Independent validation is performed using a backdoor GPS card per machine, which measures the *actual* UTC offset of the Linux kernel clock against GPS.

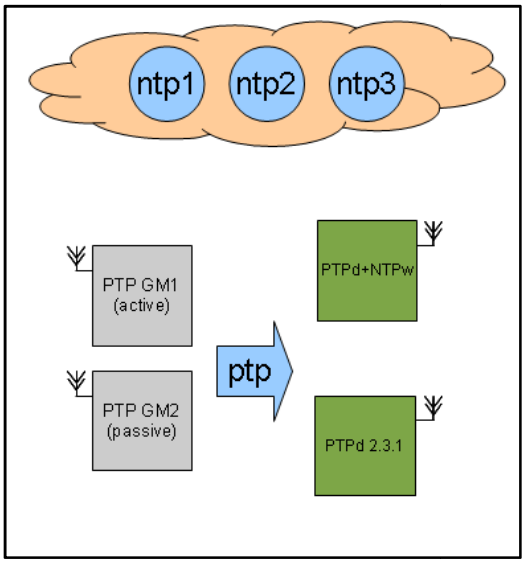

Figure 3. Example testbed, connecting two PTP GMs to multiple PTP clients. All involved machines have a backdoor GPS card to measure their actual UTC offset

Two machines will be the PTP time sources, while the other will be slaves to them. The scenario is as follows:

- GM1 runs PTPd in master mode, being the active GM for this network. This clock will be manipulated in various ways to simulate byzantine failures on demand, by accelerating or slowing down its kernel clock frequency away from its nominal value.
- GM2 also runs PTPd in master mode, but it only provides backup PTP time, by being the passive GM. It should be stressed that no time manipulations are performed in this machine.
- Slave 1 (PTPd+NTPw) runs a standard PTPd daemon [6], modified with the multiple NTP servers watchdog feature described in the previous section. For this, PTPd+NTPw is configured to periodically query 3 far-away NTP servers located away from the testlab (NTP1-3).
- Slave 2 (PTPd-v2.3.1) runs latest version of the standard open source client, which at time of writing is v2.3.1-rc from June 2014. This slave is representative of many PTP-only slaves used to build accurate PTP solutions today, which are available either as commercial solutions or as open-source projects.

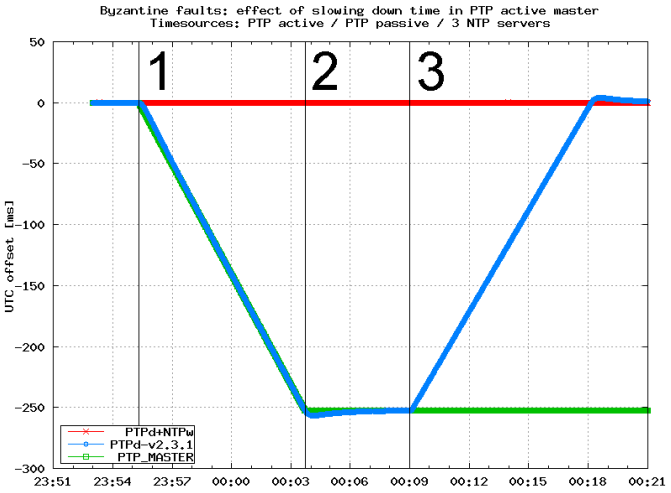

Figure 4. Experimental test, showing the measured UTC offsets during a byzantine failure of the active GM, where the clock is slowed-down to ~0.25 UTC seconds for 8 minutes

The test results are depicted in Figure 4. This graph shows the error of the kernel clocks of the various machines against UTC, using the previously mentioned backdoor GPS cards. The results of the passive GM and of PTPd+NTPw are very similar, thus only the latter is shown.

This test has a total duration of ~25 minutes, and had the following steps:

- At first, all involved clocks were synchronized with PTP to well below sub-millisecond accuracy. This was confirmed by both the backdoor GPS cards and the PTP log files.
- At **step 1**, the GM clock frequency stops being automatically disciplined, and instead was slowed-down at the maximum rate permitted by the "adjtimex" linux tool. This result in the GM having bigger and bigger offsets every second; such is visible on the graph by the negative green line of Figure 4. At this stage the PTPd GM daemon is still the active GM of the network, even though it starts sending its own (wrong) time to slaves.
- Slave 2 (blue line) successfully tracks the single GM time as closely as possible - and thus accumulate its offsets at the same rate. This results in the blue line tracking very closely the green line in Figure 4. The log file of slave 2 consistently shows very small offsets against the active GM.
- Initially, PTPd+NTPw (red line) did accumulate the very same UTC offsets as the other slave. However, this was only the case until the NTP state machine queried the NTP servers again, and this offset went higher than the configured value of 5ms. At this stage, the NTP watchdog

seamlessly took control of the clock, and disciplined it to NTP accuracy (in this case, to hundreds of microseconds).

- **Step 2:** after the GM accumulated a ~0.25 second UTC error, its frequency was manually re-adjusted to the previous nominal value. While this prevented further drifting, it also resulted in the GM keeping a constant phase offset for the remaining of the test. As before, its PTP daemon continues to be the active GM for the network, and is thus allowed to send PTP packets as normal with the wrong time information.
- **Step 3:** at this stage the active GM daemon is killed, stopping sending any more PTP packets. No further manual changes are performed until the end of the test. When this happens the regular PTP robustness facilities elect GM2 as the new active GM.
- It is only at this stage that the PTP-only client becomes aware of the ~0.25 second UTC offset. When this happens it proceeds to discipline the clock as fast as possible to remove it, a process that takes more than ~10 minutes.

VI. CONCLUSIONS

This paper has described how PTPv2 - the only high-accuracy solution available today to meet the latest financial services regulations - is vulnerable to a variety of robustness problems due to having its slaves listen to a single time source only. This results in PTP having a fundamental single point of failure – the single GM – that renders it vulnerable to byzantine failures, and other types of malicious issues like spoofing.

This problem has been observed multiple times in the financial industry, making it impossible to fully comply with existing regulations using the PTP protocol.

To raise awareness of this problem, the paper has described one of the several possible solutions on how to add multi-source robustness to existing PTP slaves, coupled with a laboratory test that shows how different protocol implementations are affected by this problem.

The test results show that having multiple PTP time sources available is not enough – instead, it is really required that end-slaves *themselves* are able to continuously query and apply offsets from multiple geographically disperse time sources, repeatedly.

Thus it is hoped that this paper will influence the forthcoming PTPv3 protocol to add multi-time sources at the PTP end-slaves themselves, enabling a self-sufficient accurate and robust time solution.

ACKNOWLEDGMENTS

The authors would like to thank their colleagues and the other financial industry entities for their collaboration, discussions and review.

In particular, the first author would like to thank L. Bonebakker for the 2010 / 2011 research collaboration that trigged the solution presented in section IV.

REFERENCES

[1] Financial Industry regulatory Authority, “Order Audit Trail System (OATS) Reporting Technical Specifications”, December 2011

[2] C. Fetzer and F. Cristian, “Integrating external and internal clock synchronization,” J. Real-Time Systems, vol. 12, no. 2, pp. 123–172, March 1997

[3] P. Estrela, L. Bonebakker, "Challenges deploying PTPv2 in a global financial company", ISPCS 2012, September 2012

[4] W. Owczarek, "Deploying PTP as an Enterprise Service: Issues, challenges and design considerations", ISPCS 2013, September 2013

[5] Deutsche Boerse, "ExServes Time Service", April 2012

[6] K. Correl, et al, “Design considerations for software only implementations of the IEEE 1588 Precision Time Protocol”, IEEE 1588 Conference for Networked Measurement and Control Systems, 2006

[7] http://ptpd.svn.sourceforge.net/viewvc/ptpd/trunk/ChangeLog?revision=179, accessed in April 2012

[8] D. Mills, “A Brief History of NTP Time: memoirs an Internet Timekeeper”, available in http://www.eecis.udel.edu/~mills/database/papers/history.pdf, ACM SIGCOMM Computer Comm. Rev., vol. 33, no. 2, 2003

[9] P. Estrela, “Clock Sync vulnerabilities in the Financial sector”, NAV Series: GNSS Vulnerabilities and Resilient PNT - RIN, Teddington, February 2014

[10] “Eurex restarts trading after technical glitch”, available in http://online.wsj.com/news/articles/SB10001424127887324591204579036503337130322, August 2013

[11] L. Lobo, “NPLTime – Trusted Time”, available in http://www.npl.co.uk/upload/pdf/npltime-brochure.pdf, May 2013

[12] Perseus Telecom, “High Precision Time”, available at http://perseustelecom.com/products-2/certified-time-by-perseus-telecom/, 2014

[13] “Foresight: The Future of Computer Trading in Financial Markets (2012)”, Final Project Report, The Government Ofice for Science, London, available in http://www.cftc.gov/ucm/groups/public/@aboutcftc/documents/file/tacfuturecomputertrading1012.pdf, page 16, 2012.

[14] T. Humphreys, "GPS Spoofing and the Financial Sector", available at http://rntfnd.org/wp-content/uploads/summary_financial_sector_implications.pdf, June 2011

[15] Securities Exchange Commission, "Consolidated Audit Trail - Rule 613(d)(3)", available in https://www.sec.gov/rules/final/2012/34-67457.pdf, page 339, October 2012

[16] J. Volpe, "Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System", National Transportation Center, 2001

[17] DCF77: http://en.wikipedia.org/wiki/DCF77, accessed May 2014

[18] “Wolfgang Eholzer on the advantages of PTP for Eurex Exchange's new trading architecture”, available in *http://www.eurexchange.com/exchange-en/about-us/news/187084/*, August 2012

[19] M. Weiss, et al, “The Case for Cross Disciplinary Research on Time Aware Applications, Computers and Communication Systems (TAACCS), White Paper, available in http://tf.nist.gov/seminars/WSTS/TAACCS/TheCaseforTAACCS.pdf, September 2013