

Network Working Group
Internet Draft
Intended status: Experimental
Expires: April 2017

A. Shpiner
Mellanox
R. Tse
PMC-Sierra
C. Schelp
Oracle
T. Mizrahi
Marvell
October 27, 2016

Multi-Path Time Synchronization
draft-ietf-tictoc-multi-path-synchronization-07.txt

Abstract

Clock synchronization protocols are very widely used in IP-based networks. The Network Time Protocol (NTP) has been commonly deployed for many years, and the last few years have seen an increasingly rapid deployment of the Precision Time Protocol (PTP). As time-sensitive applications evolve, clock accuracy requirements are becoming increasingly stringent, requiring the time synchronization protocols to provide high accuracy. This memo describes a multi-path approach to PTP and NTP over IP networks, allowing the protocols to run concurrently over multiple communication paths between the master and slave clocks, without modifying these protocols. The multi-path approach can significantly contribute to clock accuracy, security and fault tolerance. The multi-path approach that is presented in this document enables backward compatibility with nodes that do not support the multi-path functionality.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 27, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	3
2. Conventions Used in this Document.....	4
2.1. Abbreviations.....	4
2.2. Terminology.....	5
3. Multiple Paths in IP Networks.....	5
3.1. Load Balancing.....	5
3.2. Using Multiple Paths Concurrently.....	5
3.3. Two-Way Paths.....	6
4. Solution Overview.....	6
4.1. Path Configuration and Identification.....	6
4.2. Combining.....	7
5. Multi-Path Time Synchronization over IP Networks.....	7
5.1. Overview.....	7
5.2. Single-Ended Multi-Path Synchronization.....	8
5.2.1. Single-Ended MPPTP Synchronization Message Exchange..	8
5.2.2. Single-Ended MPNTP Synchronization Message Exchange..	9
5.3. Dual-Ended Multi-Path Synchronization.....	10
5.3.1. Dual-Ended MPPTP Synchronization Message Exchange...	10
5.3.2. Dual-Ended MPNTP Synchronization Message Exchange...	11
5.4. Using Traceroute for Path Discovery.....	12
5.5. Using Unicast Discovery for MPPTP.....	13
6. Combining Algorithm.....	13
7. Security Considerations.....	14
8. Scope of the Experiment.....	14
9. IANA Considerations.....	14

10. Acknowledgments.....	15
11. References.....	15
11.1. Normative References.....	15
11.2. Informative References.....	15

1. Introduction

The two most common time synchronization protocols in IP networks are the Network Time Protocol [NTP], and the Precision Time Protocol (PTP), defined in the IEEE 1588 standard [IEEE1588].

The accuracy of the time synchronization protocols directly depends on the stability and the symmetry of propagation delays on both directions between the master and slave clocks. Depending on the nature of the underlying network, time synchronization protocol packets can be subject to variable network latency or path asymmetry (e.g. [ASSYMETRY], [ASSYMETRY2]). As time sensitive applications evolve, accuracy requirements are becoming increasingly stringent.

Using a single network path in a clock synchronization protocol closely ties the slave clock accuracy to the behavior of the specific path, which may suffer from temporal congestion, faults or malicious attacks. Relying on multiple clock servers as in NTP solves these problems, but requires active maintenance of multiple accurate sources in the network, which is not always possible. The usage of Transparent Clocks (TC) in PTP solves the congestion problem by eliminating the queueing time from the delay calculations, but does not address security or fault-tolerance aspects.

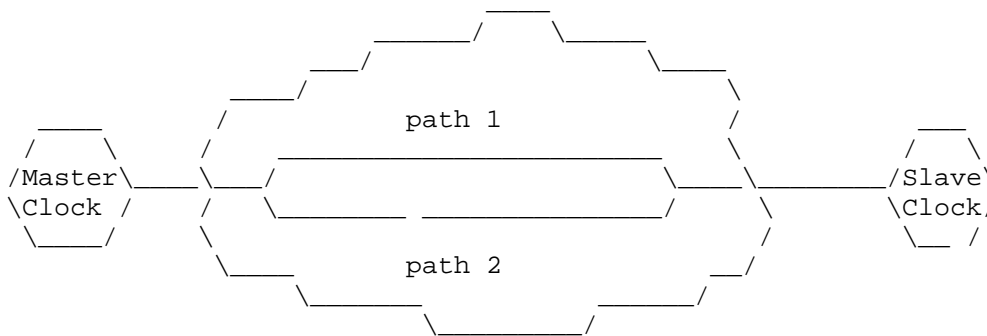


Figure 1 Multi-Path Connection

Since master and slave clocks are often connected through more than one path in the network, as shown in Figure 1, [SLAVEDIV] suggested that a time synchronization protocol can be run over multiple paths, providing several advantages. First, it can significantly increase the clock accuracy as shown in [SLAVEDIV]. Second, this approach provides additional security, allowing to mitigate man-in-the-middle attacks against the time synchronization protocol [DELAY-ATT]. Third, using multiple paths concurrently provides an inherent failure protection mechanism.

This document introduces Multi-Path PTP (MPPTP) and Multi-Path NTP (MPNTP). The functionality of the multi-path approach is defined at the network layer and does not require any changes in the PTP or in the NTP protocols.

MPPTP and MPNTP are defined over IP networks. As IP networks typically combine ECMP routing, this property is leveraged for the multiple paths used in MPPTP and MPNTP. The key property of the multi-path approach is that clocks in the network can use more than one IP address. Each {master IP, slave IP} address pair defines a path. Depending on the network topology and configuration, the IP combination pairs can form multiple diverse paths used by the multi-path synchronization protocols. It has been shown [MULTI] that using multiple IP addresses over the wide Internet indeed allows two end-points to attain multiple diverse paths.

This document introduces two variants of the multi-path approach; a variant that requires both master and slave nodes to support the multi-path functionality, referred to as the dual-ended variant, and a backward compatible variant that allows a multi-path clock to connect to a conventional single-path clock, referred to as the single-ended variant.

2. Conventions Used in this Document

2.1. Abbreviations

BMC	Best Master Clock [IEEE1588]
ECMP	Equal Cost Multiple Path
LAN	Local Area Network
MPNTP	Multi-Path Network Time Protocol
MPPTP	Multi-Path Precision Time Protocol

NTP Network Time Protocol [NTP]

PTP Precision Time Protocol [IEEE1588]

2.2. Terminology

In the NTP terminology, a time synchronization protocol is run between a client and a server, while PTP uses the terms master and slave. Throughout this document, the sections that refer to both PTP and NTP generically use the terms master and slave.

3. Multiple Paths in IP Networks

3.1. Load Balancing

Traffic sent across IP networks is often load balanced across multiple paths. The load balancing decisions are typically based on packet header fields: source and destination addresses, Layer 4 ports, the Flow Label field in IPv6, etc.

Three common load balancing criteria are per-destination, per-flow and per-packet. The per-destination load balancers take a load balancing decision based on the destination IP address. Per-flow load balancers use various fields in the packet header, e.g., IP addresses and Layer 4 ports, for the load balancing decision. Per-packet load balancers use flow-blind techniques such as round-robin without basing the choice on the packet content.

3.2. Using Multiple Paths Concurrently

To utilize the diverse paths that traverse per-destination load-balancers or per-flow load-balancers, the packet transmitter can vary the IP addresses in the packet header. The analysis in [PARIS2] shows that a significant majority of the flows on the internet traverse per-destination or per-flow load-balancing. It presents statistics that 72% of the flows traverse per-destination load balancing and 39% of the flows traverse per-flow load-balancing, while only a negligible part of the flows traverse per-packet load balancing. These statistics show that the vast majority of the traffic on the internet is load balanced based on packet header fields.

The approaches in this draft are based on varying the source and destination IP addresses in the packet header. Possible extensions have been considered that also vary the UDP ports. However some of the existing implementations of PTP and NTP use fixed UDP port values in both the source and destination UDP port fields, and thus do not allow this approach.

3.3. Two-Way Paths

A key property of IP networks is that packets forwarded from A to B do not necessarily traverse the same path as packets from B to A. Thus, we define a two-way path for a master-slave connection as a pair of one-way paths: the first from master to slave and the second from slave to master.

If possible, a traffic engineering approach can be used to verify that time synchronization traffic is always forwarded through bidirectional two-way paths, i.e., that each two-way path uses the same route on the forward and reverse directions, thus allowing propagation time symmetry. However, in the general case two-way paths do not necessarily use the same path for the forward and reverse directions.

4. Solution Overview

The multi-path time synchronization protocols we present are comprised of two building blocks; one is the path configuration and identification, and the other is the algorithm used by the slave to combine the information received from the various paths.

4.1. Path Configuration and Identification

The master and slave clocks must be able to determine the path of transmitted protocol packets, and to identify the path of incoming protocol packets. A path is determined by a {master IP, slave IP} address pair. The synchronization protocol message exchange is run independently through each path.

Each IP address pair defines a two-way path, and thus allows the clocks to bind a transmitted packet to a specific path, or to identify the path of an incoming packet.

If possible, the routing tables across the network should be configured with multiple traffic engineered paths between the pair of clocks. By carefully configuring the routers in such networks it is possible to create diverse paths for each of the IP address pairs between two clocks in the network. However, in public and provider networks the load balancing behavior is hidden from the end users. In this case the actual number of paths may be less than the number of IP address pairs, since some of the address pairs may share common paths.

4.2. Combining

Various methods can be used for combining the time information received from the different paths. The output of the combining algorithm is the accurate time offset. Combining methods are further discussed in Section 6.

5. Multi-Path Time Synchronization over IP Networks

5.1. Overview

This section presents two variants of MPPTP and MPNTP; single-ended multi-path time synchronization and dual-ended multi-path time synchronization. In the first variant, the multi-path approach is only implemented by the slave and the master is not aware of its usage. In the second variant, all clocks use multiple paths.

The dual-ended variant provides higher path diversity by using multiple IP addresses at both ends, the master and slave, while the single-ended variant only uses multiple addresses at the slave. Consequently, the single-ended approach is can interoperate with existing implementations, which do not use multiple paths. The dual-ended and single-ended approaches can co-exist in the same network; each slave selects the connection(s) it wants to make with the available masters. A dual-ended slave could switch to single-ended mode if it does not see any dual-ended masters available. A single-ended slave could connect to a single IP address of a dual-ended master.

Multi-path time synchronization, in both variants, requires clocks to use multiple IP addresses. Using multiple IP addresses introduces a tradeoff. A large number of IP addresses allows a large number of diverse paths, providing the advantages of slave diversity discussed in Section 1. On the other hand, a large number of IP addresses is more costly, requires the network topology to be more redundant, and exacts extra management overhead.

If possible, the set of IP addresses for each clock should be chosen in a way that enables the establishment of paths that are the most different. If the load balancing rules in the network are known, it is possible to choose the IP addresses in a way that enforces path diversity. However, even if the load balancing scheme is not known, a careful choice of the IP addresses can increase the probability of path diversity. For example, choosing multiple addresses with different prefixes is likely to produce higher path diversity, as BGP routers are more likely to route these different prefixes through different routes.

The use of Network Address Translation (NAT) may significantly reduce the effectiveness of multi-path synchronization in some cases. For example, if a master uses multiple IP addresses that are translated to a single IP address, the path diversity can be dramatically reduced compared to a network that does not use NAT. Thus, path discovery should be used to identify the possible paths between the master and slave. Path discovery is further discussed in Section 5.4.

The concept of using multiple IP addresses or multiple interfaces is a well-established concept that is being used today by various applications and protocols, e.g., [MPTCP]. Using multiple interfaces introduces some challenges and issues, which were presented and discussed in [MIF].

The descriptions in this section refer to the end-to-end scheme of PTP, but are similarly applicable to the peer-to-peer scheme. MPNTP, as described in this document, refers to the NTP client-server mode, although the concepts described here can be extended to include the symmetric variant as well.

Multi-path synchronization by nature requires protocol messages to be sent as unicast. Specifically in PTP, the following messages must be sent as unicast in MPPTP: Sync, Delay_Req, Delay_Resp, PDelay_Req, PDelay_Resp, Follow_Up, and PDelay_Resp_Follow_Up. Note that [IEEE1588] allows these messages to be sent either as multicast or as unicast.

5.2. Single-Ended Multi-Path Synchronization

In the single-ended approach, only the slave is aware of the fact that multiple paths are used, while the master is agnostic to the usage of multiple paths. This approach allows a hybrid network, where some of the clocks are multi-path clocks, and others are conventional one-path clocks. A single-ended multi-path clock presents itself to the network as N independent clocks, using N IP addresses, as well as N clock identity values (in PTP). Thus, the usage of multiple slave identities by a slave clock is transparent from the master's point of view, such that it treats each of the identities as a separate slave clock.

5.2.1. Single-Ended MPPTP Synchronization Message Exchange

The single-ended MPPTP message exchange procedure is as follows.

- o Each single-ended MPPTP clock has a fixed set of N IP addresses and N corresponding clockIdentities. Each clock arbitrarily defines one of its IP addresses and clockIdentity values as the clock primary identity.
- o A single-ended MPPTP port sends Announce messages only from its primary identity, according to the BMC algorithm.
- o The BMC algorithm at each clock determines the master, based on the received Announce messages.
- o A single-ended MPPTP port that is in the 'slave' state uses unicast negotiation to request the master to transmit unicast messages to each of the N slave clock identities. The slave port periodically sends N Signaling messages to the master, using each of its N identities. The Signaling message includes the REQUEST_UNICAST_TRANSMISSION_TLV.
- o The master periodically sends unicast Sync messages from its primary identity, identified by the sourcePortIdentity and IP address, to each of the slave identities.
- o The slave, upon receiving a Sync message, identifies its path according to the destination IP address. The slave sends a Delay_Req unicast message to the primary identity of the master. The Delay_Req is sent using the slave identity corresponding to the path the Sync was received through. Note that the rate of Delay_Req messages may be lower than the Sync message rate, and thus a Sync message is not necessarily followed by a Delay_Req.
- o The master, in response to a Delay_Req message from the slave, responds with a Delay_Resp message using the IP address and sourcePortIdentity from the Delay_Req message.
- o Upon receiving the Delay_Resp message, the slave identifies the path using the destination IP address and the requestingPortIdentity. The slave can then compute the corresponding path delay and the offset from the master.
- o The slave combines the information from all negotiated paths.

5.2.2. Single-Ended MPNTP Synchronization Message Exchange

The single-ended MPNTP message exchange procedure is as follows.

- o A single-ended MPNTP client has N separate identities, i.e., N IP addresses. The assumption is that the server information, including its IP address is known to the NTP clients. This is a fair assumption, as typically the address(es) of the NTP server(s) are provided to the NTP client by configuration.
- o A single-ended MPNTP client initiates the NTP protocol with an NTP server N times, using each of its N identities.
- o The NTP protocol is maintained between the server and each of the N client identities.
- o The client sends NTP messages to the master using each of its N identities.
- o The server responds to the client's NTP messages using the IP address from the received NTP packet.
- o The client, upon receiving an NTP packet, uses the IP destination address to identify the path it came through, and uses the time information accordingly.
- o The client combines the information from all paths.

5.3. Dual-Ended Multi-Path Synchronization

In dual-ended multi-path synchronization each clock has N IP addresses. Time synchronization messages are exchanged between some of the combinations of {master IP, slave IP} addresses, allowing multiple paths between the master and slave. Note that the actual number of paths between the master and slave may be less than the number of chosen {master, slave} IP address pairs.

Once the multiple two-way connections are established, a separate synchronization protocol exchange instance is run through each of them.

5.3.1. Dual-Ended MPPTP Synchronization Message Exchange

The dual-ended MPPTP message exchange procedure is as follows.

- o Every clock has N IP addresses, but uses a single clockIdentity.
- o The BMC algorithm at each clock determines the master. The master is identified by its clockIdentity, allowing other clocks to know the multiple IP addresses it uses.

- o When a clock sends an Announce message, it sends it from each of its IP addresses with its clockIdentity.
- o A dual-ended MPPTP port that is in the 'slave' state uses unicast negotiation to request the master to transmit unicast messages to some or all of its N_s IP addresses. This negotiation is done individually between a slave IP address and the corresponding master IP address that the slave desires a connection with. The slave port periodically sends Signaling messages to the master, using some or all of its N_s IP addresses as source, to the corresponding master's N_m IP addresses. The Signaling message includes the REQUEST_UNICAST_TRANSMISSION_TLV.
- o The master periodically sends unicast Sync messages from each of its IP addresses to the corresponding slave IP addresses for which a unicast connection was negotiated.
- o The slave, upon receiving a Sync message, identifies its path according to the {source, destination} IP addresses. The slave sends a Delay_Req unicast message, swapping the source and destination IP addresses from the Sync message. Note that the rate of Delay_Req messages may be lower than the Sync message rate, and thus a Sync message is not necessarily followed by a Delay_Req.
- o The master, in response to a Delay_Req message from the slave, responds with a Delay_Resp message using the sourcePortIdentity from the Delay_Req message, and swapping the IP addresses from the Delay_Req.
- o Upon receiving the Delay_Resp message, the slave identifies the path using the {source, destination} IP address pair. The slave can then compute the corresponding path delay and the offset from the master.
- o The slave combines the information from all negotiated paths.

5.3.2. Dual-Ended MPNTP Synchronization Message Exchange

The MPNTP message exchange procedure is as follows.

- o Each NTP clock has a set of N IP addresses. The assumption is that the server information, including its multiple IP addresses is known to the NTP clients.

- o The MPNTP client chooses N_{svr} of the N server IP addresses and N_{c} of the N client IP addresses and initiates the $N_{\text{svr}}*N_{\text{c}}$ instances of the protocol, one for each {server IP, client IP} pair, allowing the client to combine the information from the $N_{\text{s}}*N_{\text{c}}$ paths.
(N_{svr} and N_{c} indicate the number of IP addresses of the server and client, respectively, which a client chooses to connect with)
- o The client sends NTP messages to the master using each of the source-destination address combinations.
- o The server responds to the client's NTP messages using the IP address combination from the received NTP packet.
- o Using the {source, destination} IP address pair in the received packets, the client identifies the path, and performs its computations for each of the paths accordingly.
- o The client combines the information from all paths.

5.4. Using Traceroute for Path Discovery

The approach described thus far uses multiple IP addresses in a single clock to create multiple paths. However, although each two-way path is defined by a different {master, slave} address pair, some of the IP address pairs may traverse exactly the same network path, making them redundant.

Traceroute-based path discovery can be used for filtering only the IP addresses that obtain diverse paths. 'Paris Traceroute' [PARIS] and 'TraceFlow' [TRACEFLOW] are examples of tools that discover the paths between two points in the network. It should be noted that this filtering approach is effective only if the Traceroute implementation uses the same IP addresses and UDP ports as the synchronization protocol packets. Since some Traceroute implementations vary the UDP ports, they may not be effective in identifying and filtering redundant paths in synchronization protocols.

The Traceroute-based filtering can be implemented by both master and slave nodes, or it can be restricted to run only on slave nodes to reduce the overhead on the master. For networks that guarantee that the path of the timing packets in the forward and reverse direction are the same, path discovery should only be performed at the slave.

Since network routes change over time, path discovery and redundant path filtering should be performed periodically. Two {master,slave} pairs that produce two diverse paths may be rerouted to use the same

paths. Thus, the set of addresses that are used by each clock should be reassessed regularly.

5.5. Using Unicast Discovery for MPPTP

As presented above, MPPTP uses Announce messages and the BMC algorithm to discover the master. The unicast discovery option of PTP can be used as an alternative.

When using unicast discovery the MPPTP slave ports maintain a list of the IP addresses of the master. The slave port uses unicast negotiation to request unicast service from the master, as follows:

- o In single-ended MPPTP, the slave uses unicast negotiation from each of its identities to the master's (only) identity.
- o In dual-ended MPPTP, the slave uses unicast negotiation from its IP addresses, each to a corresponding master IP address to request unicast synchronization messages.

Afterwards, the message exchange continues as described in sections 5.2.1. and 5.3.1.

The unicast discovery option can be used in networks that do not support multicast or in networks in which the master clocks are known in advance. In particular, unicast discovery avoids multicasting Announce messages.

6. Combining Algorithm

Previous sections discussed the methods of creating the multiple paths and obtaining the time information required by the slave algorithm. Once the time information is received through each of the paths, the slave should use a combining algorithm, which consolidates the information from the different paths into a single clock. Various methods have been suggested for combining information from different paths or from different clocks, e.g., [NTP], [SLAVEDIV], [HIGH-AVAI], [KALMAN]. The choice of the combining algorithm is local to the slave, and does not affect interoperability. Hence, this document does not define a specific method to be used by the slave. The combining algorithm should be chosen carefully based on the system properties, as different combining algorithms provide different advantages. For example, some combining algorithms (e.g., [NTP], [DELAY-ATT]) are intended to be robust in the face of security attacks, while other combining algorithms (e.g., [KALMAN]) are more resilient to random delay variation.

7. Security Considerations

The security aspects of time synchronization protocols are discussed in detail in [RFC7384]. The methods described in this document propose to run a time synchronization protocol through redundant paths, and thus allow to detect and mitigate man-in-the-middle attacks, as described in [DELAY-ATT]. Specifically, multi-path synchronization can mitigate the following threats (as per [RFC7384]):

- o Packet manipulation (Section 3.2.1 of [RFC7384]).
- o Packet Interception and Removal (Section 3.2.5 of [RFC7384]).
- o Packet delay manipulation (Section 3.2.6 of [RFC7384]).

It should be noted that when using multiple paths, these paths may partially overlap, and thus an attack that takes place in a common segment of these paths is not mitigated by the redundancy. Moreover, an on-path attacker may in some cases have access to more than one router, or may be able to migrate from one router to another. Therefore, when using multiple paths it is important for the paths to be as diverse and as independent as possible, making the redundancy scheme more tolerant to on-path attacks.

It should be noted that the multi-path approach requires the master (or NTP server) to dedicate more resources to each slave (client) than the conventional single-path approach. Hence, well-known Distributed Denial-of-Service (DDoS) attacks may potentially be amplified when the multi-path approach is enabled.

8. Scope of the Experiment

This memo is published as an experimental RFC. The purpose of the experimental period is to allow the community to analyze and to verify the methods defined in this document. An experimental evaluation of some of these methods has been published in [MULTI]. It is expected that the experimental period will allow the methods to be further investigated and verified by the community. The duration of the experiment is expected to be no less than two years from the publication of this document.

9. IANA Considerations

There are no IANA actions required by this document.

RFC Editor: please delete this section before publication.

10. Acknowledgments

The authors gratefully acknowledge the useful comments provided by Peter Meyer, Doug Arnold, Joe Abley, Zhen Cao, Watson Ladd, and Mirja Kuehlewind, as well as other comments received from the TICTOC working group participants.

This document was prepared using 2-Word-v2.0.template.dot.

11. References

11.1. Normative References

- [IEEE1588] IEEE Instrumentation and Measurement Society, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588, 2008.
- [NTP] D. Mills, J. Martin, J. Burbank, W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", IETF, RFC 5905, 2010.

11.2. Informative References

- [ASSYMETRY] Yihua He and Michalis Faloutsos and Srikanth Krishnamurthy and Bradley Huffaker, "On routing asymmetry in the internet", IEEE Globecom, 2005.
- [ASSYMETRY2] Abhinav Pathak, Himabindu Pucha, Ying Zhang, Y. Charlie Hu, and Z. Morley Mao, "A measurement study of internet delay asymmetry", PAM'08, 2008.
- [DELAY-ATT] T. Mizrahi, "A Game Theoretic Analysis of Delay Attacks against Time Synchronization Protocols", ISPCS, 2012.
- [HIGH-AVAI] P. Ferrari, A. Flammini, S. Rinaldi, G. Prytz, "High availability IEEE 1588 nodes over IEEE 802.1 aq Shortest Path Bridging networks" ISPCS, 2013.
- [KALMAN] G. Giorgi, C. Narduzzi, "Kalman filtering for multi-path network synchronization" ISPCS, 2014.
- [MIF] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, DOI 10.17487/RFC6418, November 2011, <<http://www.rfc-editor.org/info/rfc6418>>.

- [MPTCP] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <<http://www.rfc-editor.org/info/rfc6824>>.
- [MULTI] A. Shpiner, Y. Revah, T. Mizrahi, "Multi-path Time Protocols" ISPCS, 2013.
- [PARIS] Brice Augustin, Timur Friedman and Renata Teixeira, "Measuring Load-balanced Paths in the Internet", IMC, 2007.
- [PARIS2] B. Augustin, T. Friedman, and R. Teixeira, "Measuring Multipath Routing in the Internet", IEEE/ACM Transactions on Networking, 19(3), p. 830 - 840, June 2011.
- [RFC7384] T. Mizrahi, "Security Requirements of Time Protocols in Packet Switched Networks", IETF, RFC 7384, 2014.
- [SLAVEDIV] T. Mizrahi, "Slave Diversity: Using Multiple Paths to Improve the Accuracy of Clock Synchronization Protocols", ISPCS, 2012.
- [TRACEFLOW] J. Narasimhan, B. V. Venkataswami, R. Groves and P. Hoose, "Traceflow", IETF, draft-janapath-intarea-traceflow, work in progress, 2012.

Authors' Addresses

Alex Shpiner
Mellanox Technologies, Ltd.
Hakidma 26
Ofer Industrial Park
Yokneam, 2069200, Israel

Email: alexshp@mellanox.com

Richard Tse
PMC-Sierra
8555 Baxter Place
Burnaby, BC
Canada
V5A 4V7

Email: Richard.Tse@pmcs.com

Craig Schelp
Oracle

Email: craig.schelp@gmail.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692, Israel

Email: talmi@marvell.com

