Multi-Path Time Protocols

Alexander Shpiner[†*], Yoram Revah[†], Tal Mizrahi[†*]
[†]Marvell Israel, [*]Technion – Israel Institute of Technology
shalex@tx.technion.ac.il, yoramr@marvell.com, talmi@marvell.com

***Abstract*—Over the last few years, packet based networks have become the common transport for applications requiring clock synchronization. Classical time distribution protocols are run between a master clock and a slave clock using a single network path between the two clocks. A recently introduced approach called Slave Diversity uses multiple paths between a master-slave pair to reduce the effect of temporal congestion or errors in a specific path. The current paper applies the multi-path approach to the most widely-used packet based time protocols, PTP and NTP. We introduce extensions to the PTP and NTP standards called Multi-Path PTP (MPPTP) and Multi-Path NTP (MPNTP), respectively, and describe their application over various transport protocols. Our experimental evaluation shows that a large number of paths can be utilized when running the multi-path protocols over the internet, and thus that our multi-path approach can be effectively deployed over existing IP networks.**

Keywords: multiple paths, slave diversity, IEEE 1588, PTP, NTP, time protocol, clock synchronization.

I. Introduction

The two most common time and frequency distribution protocols in packet networks are the Network Time Protocol (NTP) [1], and the Precision Time Protocol (PTP) [2]. In both protocols the master (also known as a time server) uses dedicated protocol packets to distribute time information to slave clocks (also known as clients).

The challenge in clock synchronization over packet-based networks is the variability of the network behavior. The accuracy of the clock synchronization service directly depends on the stability and the symmetry of the propagation delay in both directions between the master clock and the synchronized slave clock. Depending on the nature of the underlying network, time protocol packets can be subject to variable network latency or path asymmetry [3].

A well-known approach for mitigating these challenges is to receive time information from multiple clock sources ([1], [4]), and to combine this information to a more accurate time.

Another approach is to use multiple network paths; since master and slave clocks are often connected through more than one path in the network, an approach called Slave Diversity [5] suggests running the time protocol over multiple paths concurrently. This approach has several advantages. First, it significantly increases the clock accuracy in the presence of a congested network, or asymmetric communication paths by combining the time information received through multiple paths. Second, this approach improves security and fault-tolerance by creating multiple paths between the pairs to help overcome man-in-the-middle attacks [6]. The concurrent use of redundant paths inherently provides path protection.

The analysis in [5] includes various approaches for combining the time information received from different paths, but it does not discuss *how* a time protocol can be run over multiple paths. Recent works have discussed the usage of multiple paths in PTP ([7], [8], [9]) over Layer 2 networks, but we are not aware of any work that generalized this approach to other transport technologies, or to NTP.

In this paper we analyze how multi-path synchronization can be achieved in various types of networks and various transport types. Our Multi-path Time Protocol approach is an extension to the PTP and NTP protocols, allowing, the time protocol to run between the master and slave concurrently through multiple paths.

The main contributions of this paper are as follows:

- We describe the main building blocks of a multi-path clock synchronization solution.
- We define multi-path extensions to NTP and PTP. We analyze how these extensions can be applied in various Layer 2 transport technologies, and then focus on how the extensions are applied in IP networks. Based on the analysis in this paper, the multi-path approach in IP networks has been proposed as an internet draft [10] in the IETF.
- We show that the proposed extensions are relatively light-weight, and allow interoperability with existing implementations of the protocol.
- We present experimental results that analyze the number of paths that can be used by a multi-path time protocol in real-life IP networks. The analysis also includes an evaluation of the path diversity, i.e., a measure of how diverse the different paths are.

II. Overview of the Multi-Path Approach

Time protocols are often deployed in network topologies that offer more than one available path between the master and slaves. In such cases a multi-path time protocol approach can be applied, utilizing multiple paths concurrently to improve the time protocol.

A. Basic Building Blocks

The multi-path approach we present consists of three building blocks:

- *Path configuration or discovery*: in a locally administered network, i.e., a network in which the operator has the ability to control the network devices, traffic engineering can be used to configure dedicated paths for the multi-path protocol rather than to discover the available paths. In other cases the underlying network may be a public or a provider network, where the master and slave have only partial information about the network topology. In these cases the master and slave clocks need to discover available paths that can be used for the multi-path protocol. Note that the multi-path protocol does not require the different paths to have the same path delay. As in the conventional single-path approach it is preferable for each path to use the same physical path for the forward and reverse directions, although it is not a requirement.

It should be noted that after the paths are discovered and used by the multi-path protocol they may be subject to failures or to topology changes. Thus, when path discovery is used, it should typically be invoked periodically to allow the protocol to choose the best set of paths based on an updated picture of the network topology. Although a topology change can potentially cause a transient effect to the time protocol, the usage of multiple paths significantly reduces the impact of a topology change in a single path.

- *Path identification*: when a clock transmits a protocol packet, it must be able to determine which path the packet is transmitted through. Similarly, when a protocol packet is received clocks must be able to determine the path from which an incoming protocol packet was received.
- *Combining*: a slave clock that receives time information through multiple paths uses a combining algorithm, resulting in a single accurate clock. The *Cluster* and *Combine* algorithms in [1] are defined for combining information from multiple clock sources, and can similarly be applied to a multi-path setting. Several multi-path combining algorithms were also presented in [5]. We note that the combining algorithm does not impose any interoperability requirements, i.e., each slave clock can independently use a different combining algorithm.

In this paper we focus on the first two building blocks, focusing on how the time protocol interacts with the transport protocol. The combining algorithm is a non-goal of this paper.

B. *Single-Ended vs. Dual-Ended Approach*

The multi-path approach can be applied in one of two possible approaches.

The dual-ended approach: both the master and the slave are aware of the multi-path protocol. Each clock maintains a list of paths to its peer clock, and sends and receives protocol packets over each of the paths.

The single-ended approach: only the slave is aware of the multi-path protocol. The slave maintains a list of N paths between itself and the master, and leads the master to believe that the master is connected to N different slaves.

The dual-ended approach allows a more flexible selection of paths, since both ends take part in the path selection. This advantage is further discussed in Section IV. The single-ended approach, on the other hand, does not require the master to be multi-path aware, and thus allows interoperability with existing implementations; it enables the deployment of hybrid networks, where some of the nodes are multi-path aware and some are not.

III. Multi-Path Time Protocols in Layer 2 Networks

This section surveys multi-path approaches for Layer 2 networks. The underlying assumption is that the network is locally administered.

A. *The Multi-Path Approach over Multiple Virtual LANs*

Virtual LAN (VLAN) [11] is a concept of partitioning a physical network. For each VLAN a dedicated spanning tree is created for the set of nodes in the VLAN. Two packets with a different VLAN ID may be forwarded through different trees.

Hence, multiple VLANs can be used to define multiple network paths. This can be achieved by defining a different spanning tree for each VLAN, defining the different paths. Another approach is to use a routing protocol such as IS-IS to define the different network paths, as suggested in [9].

Fig. 1 illustrates allocation of three VLANs between the master and the slave, creating three diverse paths. To implement the multi-path protocol over VLANs, a dedicated VLAN has to be assigned for each path between a master and a slave. Clocks can identify the path by the packet's VLAN ID. In some networks the 12-bit VLAN ID is included in the packet header using a VLAN tag. In other cases, the VLAN ID is implied by the network topology, e.g., based on the physical port a packet is received from. When receiving a time protocol packet, the switch binds the packet to a specific VLAN, and hence to a specific path.

The main advantage of using VLANs for the multi-path protocol is simplicity. However, several drawbacks exist. First, the network has to be reconfigured when adding a new slave or master, or when network topology changes occur. Second, the number of VLANs used for this method is equal to the number of slaves multiplied by the number of paths between each slave and the master. Thus, the usage of VLANs for the multi-path protocol is not scalable for large networks.

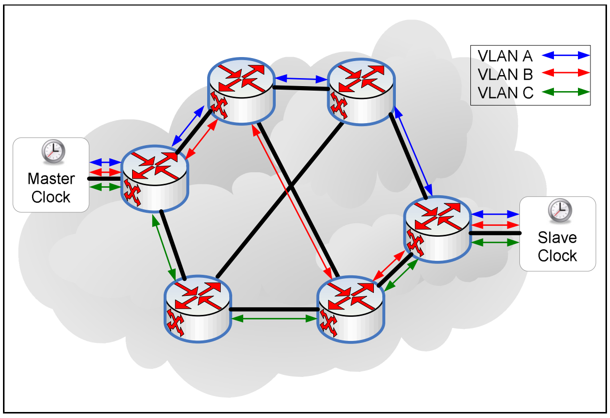

Fig. 1. Multi-Path Clock Synchronization over Virtual LANs

B. *The Multi-Path Approach over HSR and PRP*

High-availability Seamless Redundancy (HSR) [12] and Parallel Redundancy Protocol (PRP) [12] are two Layer 2 protocols that establish two diverse paths between network nodes to provide path protection. The transmitter sends each packet through both paths, and the receiver accepts one of the two duplicate packets, and drops the other.

The work presented in [7] was the first to show how the multi-path time protocol approach can be applied in HSR and PRP, requiring the receiver to accept and use both copies of each time protocol packet, rather than to drop one of them. However, the implementation in [7] does not allow the clocks to transmit PTP message on a specific path and to distinguish the path on which the PTP message was received. Thus, we propose an extension to the approach in [7] by using the HSR_path field in the HSR tag to identify the path through which each time protocol packet is received and by transmitting each PTP packet on a specific path. This allows more accurate combining algorithms, especially when the multiple paths have different characteristics.

C. *The Multi-Path Approach over 1+1 Ethernet Protection*

In 1+1 Ethernet protection switching [13] the traffic is concurrently sent over two redundant paths. In order to support 1+1 protection switching the network infrastructure has to (1) disable the spanning tree protocol (STP), otherwise two diverse paths between the master-slave pair are not available, or (2) allocate a different VLAN for each of the paths. The configuration of the VLANs has to guarantee that two diverse paths are established by the switching algorithm between the master and the slave.

Two clocks connected through a 1+1 protection scheme can use the redundant paths for running a multi-path time protocol scheme. Upon receiving a time protocol packet, the receiving switch must keep and process both copies. Similar to the approach in Section A, the VLAN ID can be used to identify which path a protocol packet is sent or received through.

IV. Multi-Path Time Protocols in IP Networks

This section surveys multi-path approaches for IP networks. Time protocols can be used to synchronize clocks that are connected by an IP network. NTP is a typical example, as well as PTP when run over IP networks[1], either with or without on-path support (e.g., [14]).

A. *Path Identification using Multiple IP Addresses*

1) Load Balancing and Multiple IP Addresses

Traffic sent across IP networks is often load balanced across multiple paths. The load balancing decisions are typically based on packet header fields, e.g., source and destination addresses, Layer 4 ports, or the Flow Label field in IPv6. Thus, having multiple paths can be leveraged for the multi-path time approach by varying the header fields of the packet.

Load balancers use *per-destination*, *per-flow* or *per-packet* balancing schemes. Per-destination load balancers make their routing decision based on the destination IP address field in the packet header. Per-flow load balancers use both source and destination IP addresses and 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. To utilize the diverse paths that traverse per-destination load-balancers or per-flow load-balancers, the packet transmitter can vary the destination IP address or the L4 ports, respectively, in the packet header. However, when traversing per-packet load balancing the packet header does not affect the load balancing decision, and hence the transmitter has no control over path selection. Fortunately, [15] shows that the vast majority of the flows traverse per-destination or per-flow load-balancing.

MPPTP and MPNTP use multiple IP addresses for each of the clocks participating in the protocol. This approach is well-known in other applications, such as Multi-Path TCP (MPTCP) [16]. Possible extensions have been considered that also vary the UDP ports. However, PTP and NTP typically use fixed values in both the source and destination UDP port, thus preventing this approach.

The multiple IP address approach is a good match for per-destination and per-flow load balancing schemes, as varying the IP address affects the path selection. Nevertheless, even in the presence of per-packet load-balancers the multi-path approach can be used, but does not benefit from multiple paths formed by per-packet load balancers. For example, in the topology in Fig. 2 path 0 traverses a per-packet load-balancer, but is treated as a single path by the master and the slave.

Each clock that participates in the multi-path time protocol should use a range of IP addresses that belong to the same subnet to avoid additional configuration of the routing tables of the intermediate routers.

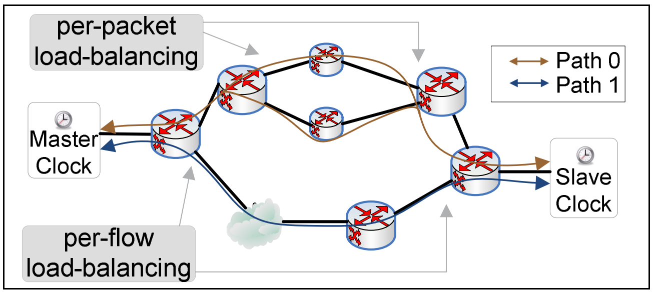

Fig. 2. Example: Multiple Paths with Various Load Balancing Schemes

2) 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 between a master and a slave as a pair of one-way paths: from the master to the slave and from the slave to the master. In the multi-path time protocol approach a slave can run the time protocol over each of the two-way paths independently.

Fig. 3 illustrates an example of the multi-path connection between a pair of nodes. Two paths are established in each direction, thus a total of four two-way paths are used by the protocol. Each clock maintains a list of two-way paths, identified by {master IP, slave IP} pairs.

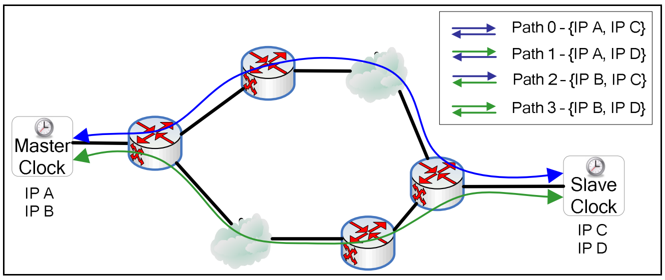

Fig. 3. Example: Multi-Path Clock Synchronization over an IP Network

B. *Path Configuration / Path Discovery*

In locally administered Layer 3 networks, the routing tables of network devices can be configured with multiple traffic engineered paths between the master and slave clock. However, in some cases time protocol packets are sent over a public or a provider network, and thus traffic engineering is not an option. Moreover, the topology and the load balancing behavior are hidden from the end users. Therefore, a multi-path time protocol deployed in such networks must use a path discovery mechanism, allowing the participating end-points to first discover the available paths, and then to use them in the protocol.

Although each two-way path is defined by a different {master, slave} address pair, some of the IP address pairs may share the

[1] On-path support in PTP refers to a network in which some or all of the intermediate switches and routers function as PTP Transparent Clocks or Boundary Clocks.

same network path, making them redundant. Traceroute-based path discovery can be used for filtering only the IP address pairs that obtain diverse paths.

For the multi-path time protocols we propose to use Paris Traceroute [15], a tool that discovers all available paths between two points in the network by scanning the values of some of the packet header fields, and probing the corresponding paths.

Path discovery 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. In networks that guarantee that the forward and reverse directions use the same physical path, path discovery should only be performed at the slave.

Following the path discovery or path configuration, a set of IP addresses is assigned to each clock, and used in the protocol. If possible, the set of IP addresses for each clock should be chosen in a way that enables the establishment of paths that are as diverse as possible. 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 [5]. On the other hand, a large number of IP addresses is more costly, and imposes extra management overhead.

C. *Theory of Operation*

In this subsection we present the theory of operation of MPPTP and MPNTP in a nutshell. The descriptions in this section refer to the end-to-end scheme of PTP, but are similarly applicable to the peer-to-peer scheme. The MPNTP protocol 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.

1) Network Layer Protocol Requirements

Multi-path synchronization protocols by nature require protocol messages to be sent as unicast, allowing different messages to be sent over different paths. 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. Thus, we assume that clocks taking part in the protocol use the unicast negotiation procedure defined in [2], whereby a master and a slave agree to use unicast messages. Announce messages, on the other hand are sent as multicast, as they are used by the master to announce its properties to the network, and can be sent over a single path.

2) MPPTP: Message Exchange in Single-Ended Mode

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.

The following procedure describes the single-ended MPPTP message exchange, which is also illustrated in Fig. 4.

1. 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 its *primary clock identity*.

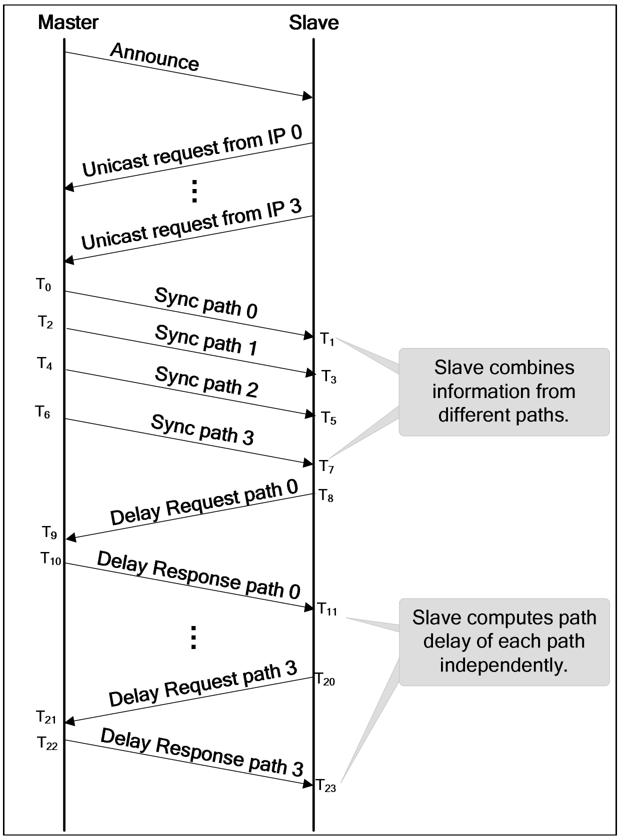

Fig. 4. Single-Ended MPPTP Message Exchange: Example with 4 Paths

2. When a single-ended MPPTP port sends an Announce message, it is only sent from its primary identity.
3. The BMC algorithm at each clock determines the master, based on the received Announce messages.
4. A single-ended MPPTP port that is in the 'slave' state uses unicast negotiation to request unicast transmission from the master. N separate Signaling messages are sent, corresponding to the N slave clock identities. The Signaling messages incorporate the REQUEST_UNICAST_TRANSMISSION_TLV.
5. The master periodically sends unicast Sync messages from its primary identity, identified by the sourcePortIdentity and IP address, to each of the slave identities that have requested unicast transmission.
6. The slave, upon receiving a Sync message, identifies which path the packet was received from according to the {source IP, destination IP} pair. 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.
7. 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.
8. Upon receiving the Delay_Resp message, the slave identifies the path using the {source IP, destination IP} pair and the requestingPortIdentity. The slave can then compute the

corresponding path delay and the offset from the master.

9. The slave combines the information from all negotiated paths.

3) MPPTP: Message Exchange in Dual-Ended Mode

In dual-ended multi-path synchronization each clock has N IP addresses. Clock synchronization messages are exchanged between some of the combinations of {master IP, slave IP} addresses, allowing multiple paths between the master and slave. A separate instance of the time protocol exchange is run through each of the paths.

The message exchange in dual-ended mode is mostly similar to the description in subsection 1), except for the following differences:

- Every clock has N IP addresses, but uses a single clockIdentity.
- The master transmits Announce messages from each of its N IP addresses. Each slave can consequently learn the IP addresses of the master, and send a unicast negotiation request to each of these addresses. As a result of this negotiation process, the two ends use multiple {master IP, slave IP} pairs in the time protocol.

4) MPNTP Message Exchange

The message exchange procedure in MPNTP is very similar to MPPTP. The server information, including its IP address range, is assumed to be generally known to the clients. The assumption is reasonable due the fact that the NTP standard does not provide a method of spreading the server information. Although NTP does not have a 'clock identity', or use Announce messages, other details in the MPPTP message exchange procedure apply to MPNTP; the {server IP, client IP} pair is used for identifying paths, and clients use Paris Traceroute for path discovery. The detailed message exchange for MPNTP is shown in [10].

V. Evaluation

In this section we evaluate the number of paths between NTP client-server pairs in wide-area IP networks. We also evaluate the path diversity, i.e., how diverse the different paths are. This analysis is relevant to MPPTP and MPNTP in IP networks.

Our experiments simulated a single-ended MPNTP topology. Our server at the Technion lab played the role of an MPNTP client with 160 IP addresses. Each experiment was conducted with a different NTP server; we ran our experiments with each of the 234 stratum 1 NTP servers [17]. [2] We used Paris Traceroute [15] for discovering the available paths between our client and the NTP server. Since our experiments focused on analyzing the paths, we did not actually run the NTP protocol, but only the Paris Traceroute probes for discovering all available paths between the end-points.

A. Number of Paths

In each experiment we used Paris Traceroute to find all available paths between our client and the NTP server. The Traceroute probes were sent using NTP-reserved source and destination port of 123 and scanning 163 values of the source IP address. For each NTP server we counted the number of different paths found. To eliminate the effect of per-packet load balancers, we removed the paths achieved by per-packet load balancers from the total number of paths, as in the example in Fig. 2.

Fig. 5 presents the distribution of the number of distinct paths achieved by varying source IP address. The experimental results show that 86% of NTP servers were reachable through multiple paths. We conclude that single-ended MPNTP can be used with the vast majority of the existing stratum 1 NTP servers. We believe that similar results can be observed from other probing locations and for other NTP servers.

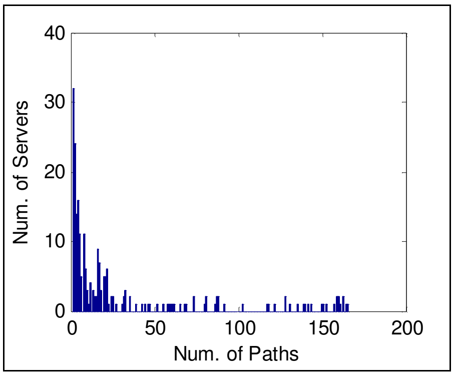

Fig. 5. Distribution of the number of paths to the Stratum 1 NTP servers

B. Path Diversity

For each link[3] along the path between the client and the server we counted the number of paths that traversed this link. Fig. 6, Fig. 7 and Fig. 8 show the results for three selected NTP servers. Each mark on the graph represents a link. The x-axis of the graph indicates the link distance (the number of hops) from our lab computer and the y-axis indicates the number of paths that traverse through the corresponding link.

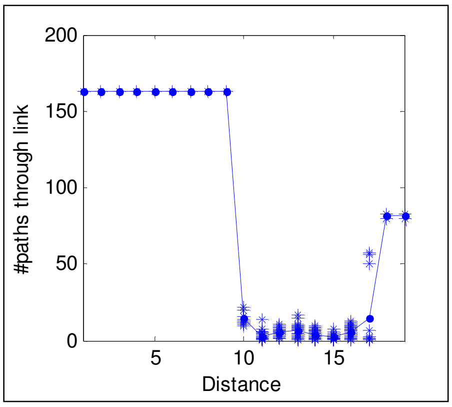

Fig. 6. Path Diversity to nist.expertsmi.com NTP server

For example, the results in Fig. 6 show that all the paths from our lab to the nist.expertsmi.com server share the same links on the first nine hops and starting from the 10th hop the multiple paths exist. The graph shows that every link at a distance of 10 to 16 hops from the source is used by at most 30 paths. The continuous line presents the mean number of paths-per-link at a specific distance, in other words it is an average of the points for the same value on the x-axis.

[2] There are ~250 servers in the list in [17], but some of them do not respond to Traceroute.

[3] In this context the term 'link' refers to a segment between two adjacent routers.

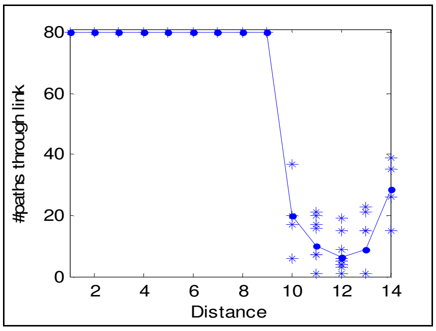

Fig. 7. Path Diversity to ntp.melbourne.nmi.gov.au NTP server

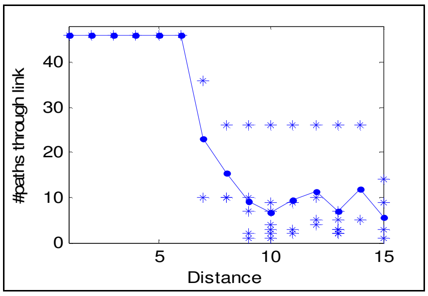

Fig. 8. Path Diversity to ntp1.niiftri.irkutsk.ru NTP server

Next we evaluate the path diversity of a single master-slave pair. Refs. [18][19][20] define different metrics to describe the parallelism and diversity of the paths. However, they all either model the communication diversity between all the nodes in the network and/or consider the topology of the network rather than considering the achievable paths only. Therefore, we define a slightly different metric for the *path diversity*. First, we define the *path similarity* of the paths between a client and a server as the mean percentage of the paths traversing each link, i.e. the mean y-axis value of the marks on Fig. 6,Fig. 7 or Fig. 8 divided by total number of paths. Another interpretation of path similarity is the percentage of affected paths given that one uniformly chosen link fails. Next, we define the *path diversity* as 1-(path similarity). Note that the path diversity is in the range [0,1). Intuitively, the multi-path advantages are correlative to the path diversity value.

In Fig. 9 we present the path diversity distribution of all the examined NTP servers. Note that the number of servers indicated by the path diversity of 1 on Fig. 9 corresponds to the number of servers with only single path on Fig. 5.

VI. Conclusion

PTP and NTP are the most common time protocols over packet networks. We presented an extension to these protocols that utilizes multiple diverse network paths between the synchronizing clocks. Using diverse paths improves the clock accuracy, security and fault tolerance. We suggested methods for establishing multiple paths between a pair of network nodes in Layer 2 and in Layer 3 networks. The methods we presented can be applied to other transport protocols such as MPLS, PBB, or TRILL. The approach and methods presented in this paper can be used in future versions of PTP and NTP, in standards that define PTP profiles, and in other standards that rely on PTP or NTP to obtain clock synchronization.

Our experimental evaluation shows that a large number of paths can be utilized when running the multi-path protocols over the internet, and thus that our multi-path approach can be effectively deployed over existing IP networks.

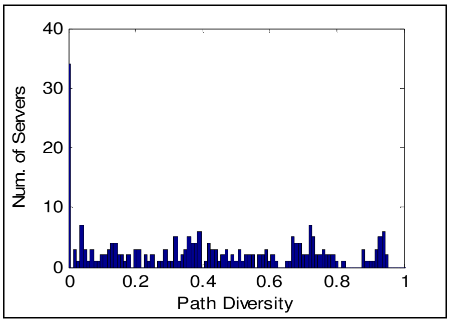

Fig. 9. Path Diversity Distribution

Acknowledgement

This work was partly supported by the Hasso Plattner Institute Research School and the Israel Ministry of Science and Technology.

References

[1] D. Mills, J. Martin, J. Burbank and W. Kasch, “Network Time Protocol Version 4: Protocol and Algorithms Specification”, *IETF, RFC 5905*, 2010.

[2] IEEE Instrumentation and Measurement Society, “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”, *IEEE Std 1588™*, 2008.

[3] Y. He, M. Faloutsos, S. Krishnamurthy and B. Huffaker, “On routing asymmetry in the internet”, *Globecom*, 2005.

[4] O. Gurewitz, I. Cidon, and M. Sidi, “Network Time Synchronization Using Clock Offset Optimization”, *ICNP*, 2003.

[5] T. Mizrahi, “Slave Diversity: Using Multiple Paths to Improve the Accuracy of Clock Synchronization Protocols”, *ISPCS*, 2012.

[6] T. Mizrahi, “A Game Theoretic Analysis of Delay Attacks against Time Synchronization Protocols”, *ISPCS*, 2012.

[7] H. Kirrmann, C. Honegger, D. Ilie, and I. Sotiropoulos, "Performance of a full-hardware PTP implementation for an IEC 62439-3 redundant IEC 61850 substation automation network", *ISPCS* 2012.

[8] A. Komes and C. Marinescu, “IEEE 1588 for Redundant Ethernet Networks”, *ISPCS*, 2012.

[9] F.J. Goetz, "High Available Synchronization with IEEE 802.1AS bt", http://www.ieee802.org/1/files/public/docs2013/asbt-goetz-HighAvailableSync-0319-v02.pdf, 2013.

[10] A. Shpiner, R. Tse, C. Schelp, T. Mizrahi, “Multi-Path Time Synchronization”, draft-shpiner-multi-path-synchronization (work in progress) *IETF*, 2013.

[11] “IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks”, *IEEE Std 802.1Q-2011*, 2011.

[12] Industrial Communication Networks – High Availability Automation Networks – Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR), *IEC*, 2010.

[13] ITU-T, “G.8031/Y.1342 , Ethernet linear protection switching”, 2011.

[14] ITU-T, “G.8265.1/Y.1365.1 Precision time protocol telecom profile for frequency synchronization”, 2010.

[15] B. Augustin, T. Friedman, and R. Teixeira, “Measuring Load-balanced Paths in the Internet”, *IMC*, 2007.

[16] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, IETF, 2013.

[17] http://support.ntp.org/bin/view/Servers/StratumOneTimeServers

[18] S. Huang, Y. Xu and L. Zhang, "A Path Diversity Metric for End-to-End Network," *PRDC*, 2007.

[19] C. de Launois, B. Quoitin and O. Bonaventure, “Leveraging network performance with IPv6 multihoming and multiple provider-dependent aggregatable prefixes”, *Computer Networks*, Vol. 50, Issue 8, 2006.

[20] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, “In search of path diversity in ISP networks”, *IMC*, 2003.