A Study of the Merits of Precision Time Protocol (IEEE-1588) Across High-Speed Data Networks

Public Note

Issue:	1
Revision:	0
Reference:	LHCb-PUB-2015-022
Created:	August 17, 2015
Last modified:	September 18, 2015

Prepared by: David Oliver[a b], Niko Neufeld[a], Adam Otto[a]
[a]CERN, Switzerland
[b]University of Cambridge, United Kingdom

Abstract

By using Precision Time Protocol across high-speed data networks, it is possible to achieve good time synchronisation using only commercial, off-the-shelve equipment. Even under heavy network loads, the attainable precision far exceeds that which is possible with Network Time Protocol, and is sufficient for many applications. This note explores the time precision possible with PTP under various conditions and attempts to provide a measurement of its performance.

Document Status Sheet

1. Document Title: A Study of the Merits of Precision Time Protocol (IEEE-1588) Across High-Speed Data Networks			
2. Document Reference Number: LHCb-PUB-2015-022			
3. Issue	**4. Revision**	**5. Date**	**6. Reason for change**
Draft	1	August 17, 2015	First version.
Draft	2	September 17, 2015	Second draft.
Final	1	September 18, 2015	Final Release.

Contents

1 Introduction

The current use of Network Time Protocol (NTP) for time synchronisation, while sufficient for many applications, provides inadequate precision and convergence time for high-speed, time-sensitive processes such as data acquisition at the LHC. The specification of the Precision Time Protocol (PTP, IEEE 1588-2002), and its subsequent revision in 2008 (IEEE 1588-2008), have allowed for much better clock synchronisation across networks, and is capable of achieving precision in the micro or nanosecond range rather than the millisecond range of NTP [1]. While the CERN White Rabbit project can achieve such precision across large, gigabit networks using PTP, it requires custom switches to act as boundary clocks which are infeasible for many applications [2]. However, the use of higher network speeds, such as 40G or 100G, should enable equivalent precision to be achieved using standard switches, by reducing the variation of the latency in the network.

Machine	Network Card	Firmware	PTP Implementation
lab09	Solarflare Communications SFN7141Q-R1 Flareon Ultra 7000 Series 10/40G Adapter	sfc 4.5.1.1010	sfptpd 2.2.4.70
lab10	Solarflare Communications SFN7141Q-R1 Flareon Ultra 7000 Series 10/40G Adapter	sfc 4.5.1.1010	sfptpd 2.2.4.70
lab11	Intel Corporation Ethernet Converged Network Adapter XL710-Q2	i40e 1.2.48	PTPd 2.3.1
lab12	Intel Corporation Ethernet Converged Network Adapter XL710-Q2	i40e 1.2.48	PTPd 2.3.1
lab13	Solarflare Communications SFN7141Q-R1 Flareon Ultra 7000 Series 10/40G Adapter	sfc 4.5.1.1010	sfptpd 2.2.4.70
lab15	Solarflare Communications SFN7141Q-R1 Flareon Ultra 7000 Series 10/40G Adapter	sfc 4.5.1.1010	sfptpd 2.2.4.70
Switch	Mellanox SX6036	SX_PPC_M460EX SX_3.4.0012	—

Table 1: Details of machines on test network, including network card model, driver firmware, and the implementation of PTP that is running on the machine.

Machine	Network Connection Speed to lab13 (Gbps)
lab09	39.5
lab10	39.6
lab11	39.7
lab12	39.7
lab15	39.6

Table 2: Achieved bandwidth of each connection to lab13 measured using iperf

2 Implementation of the Test Network

To initially benchmark the performance of the protocol, six machines were connected across a single switch using 40 Gigabit Ethernet. Table 1 shows details of the machines and the switch on the network. Two different implementations of the protocol were used, one produced by Solarflare, and the other an open source distribution [3]. This was done to ensure each fully complied with the IEEE 1588-2008 standard by being fully compatible. Lab13 was assigned to be the grandmaster clock, and the other five as slaves. For each of the machines, including lab13, NTP was disabled and lab13 was allowed to free run while the other machines maintained synchronisation. Table 2 shows the measured speed of the connection from each machine to lab13. These were measured using iPerf (version 2.0.5) to send data using TCP from each of the computers to lab13.

3 Benchmarking the Achievable Synchronisation Precision

In order to test the precision of the protocol, each of the machines (excluding lab13) were assigned a time slot in which to broadcast data to lab13 at the maximum bandwidth of the network. By comparing the data rate of this barrel-shifting broadcast to the maximum bandwidth of the network, it is possible to create a metric of the precision of the synchronisation. This is because a lack of synchronisation leaves a time portion unused and hence reduces the network throughput. This is illustrated in figures 1 and 2. Furthermore, by comparing the start time of each time slot, the relative synchronisation of the machines can be measured with reasonable precision. This was considered to be a suitable test of the system as, for many data acquisition applications, similar methods of data handling are used to condense small amounts of data from several sources into a single output stream through a switch. Furthermore, as the PTP packets are operating over the same switched network as the data, this test would be expected to cause a degradation in synchronisation as the network is swamped by the data flow, increasing queue lengths in the switch and therefore variations in network latency. This is therefore a test of the performance of the protocol in almost worst case conditions. For further details on the programs used, see Section 6.

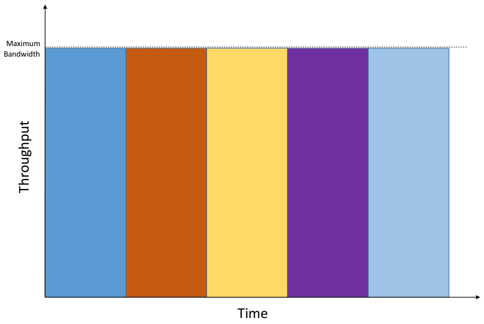

Figure 1: Maximum bandwidth can be achieved with high time precision

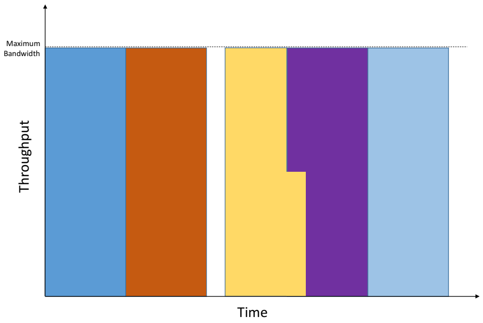

Figure 2: When there is a lack of synchronisation, network throughput is decreased

Machine	Network Connection Speed to lab13 (Gbps)
lab09	37.2
lab10	37.2
lab11	37.1
lab12	37.1
lab15	37.2

Table 3: Achieved bandwidth of each connection to lab13 using TCP

These test were originally performed using UDP. UDP is attractive because it is connectionless, and hence lacks the delays and retransmissions introduced by the management processes of similar protocols that guarantee quality of service. However, it was found that to achieve full bandwidth over the network, the packet loss became unacceptably high, and therefore was considered to be a poor test of the system. Furthermore, it was found that each process would load the network card queue with packets. This would mean that a lack of synchrony would have little effect on the overall bandwidth of the network as the queued packets would continue to send after the kernel had seized handing packets to the network card. Although this effect could be reduced by shorting the network card queue length, this was found to have a detrimental effect on the achievable throughput when the queue lengths became too short.

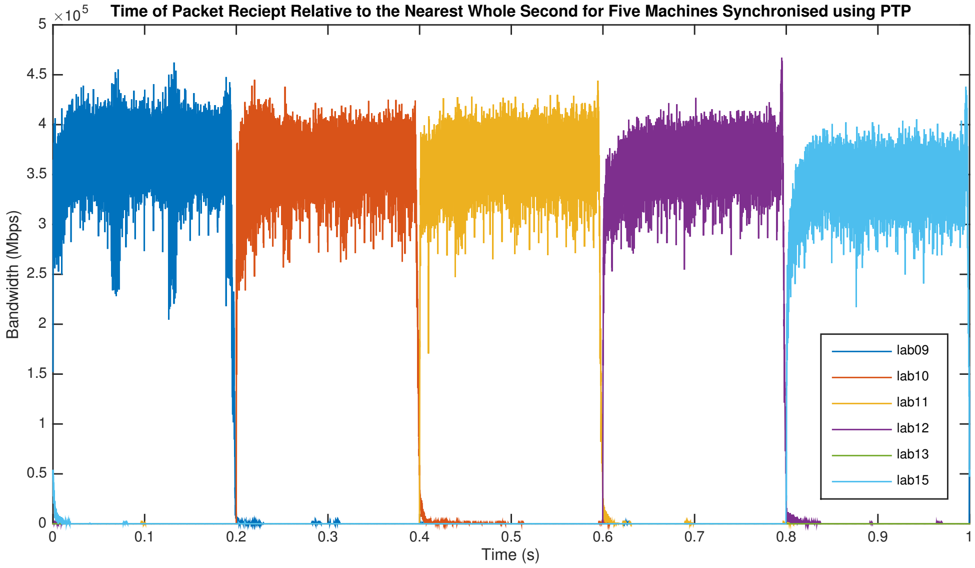

Figure 3: Number of packets received at each time relative to the previous whole second for machines synchronised using PTP

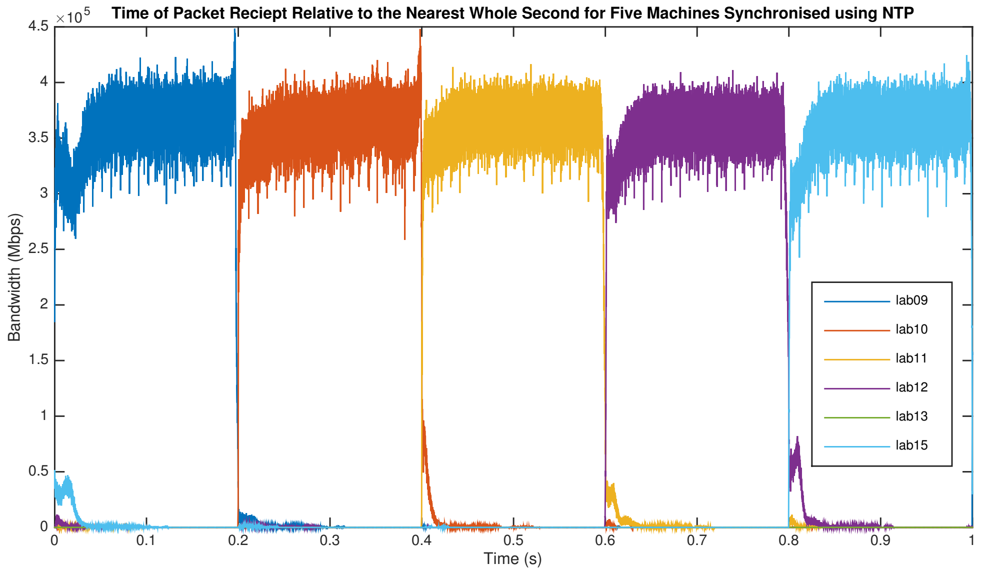

Figure 4: Number of packets received at each time relative to the previous whole second for machines synchronised using NTP

For these reasons, the decision was made to use TCP to test the system. TCP has the problem that, because it ensures quality of service, it is significantly harder to achieve the full bandwidth across the network and requires tuning. Using tuning [a], it was possible to achieve almost full bandwidth utilisation, which was sufficient for this testing (see Table 3). TCP also has the issue of loading the network card queue, however it is less pronounced than with UDP.

Using this barrel-shifting program with a timeslot width of 200,000 μs, such that the sum of the five machines fill a whole second, a bandwidth of 36.2 Gbps was achieved. The distribution of data received against time is shown in Figure 3. This is over 97% of the expected bandwidth, and it is prob-

[a] This tuning included increasing the MTU to jumbo frames (9000 Bytes) and allocating a larger amount of memory for the read and write queues (particularly for lab13). Disabling slow start was explored, but found to have a detrimental effect on the bandwidth while giving little improvement.

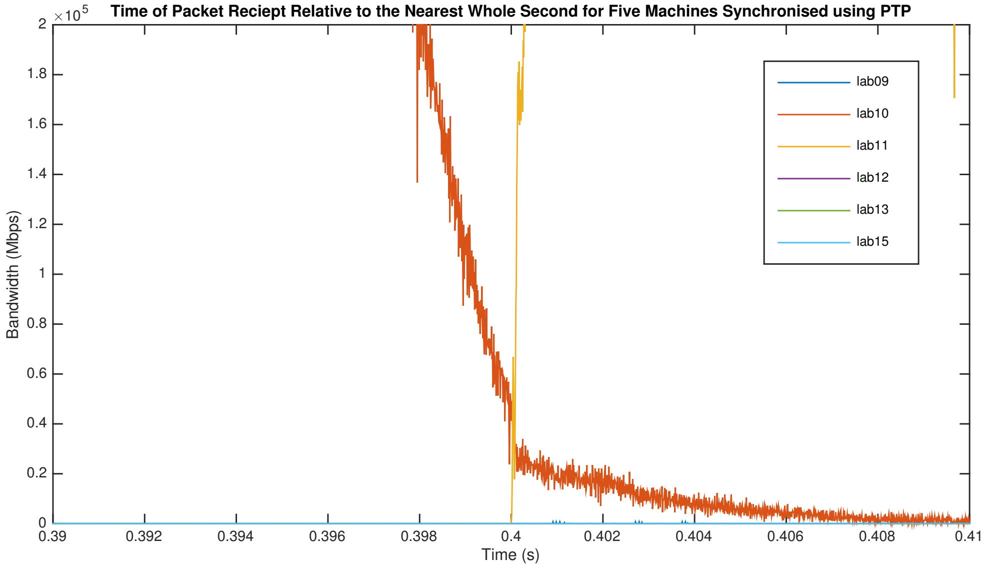

Figure 5: Portion of Figure 3 showing the offset of the start of the broadcast

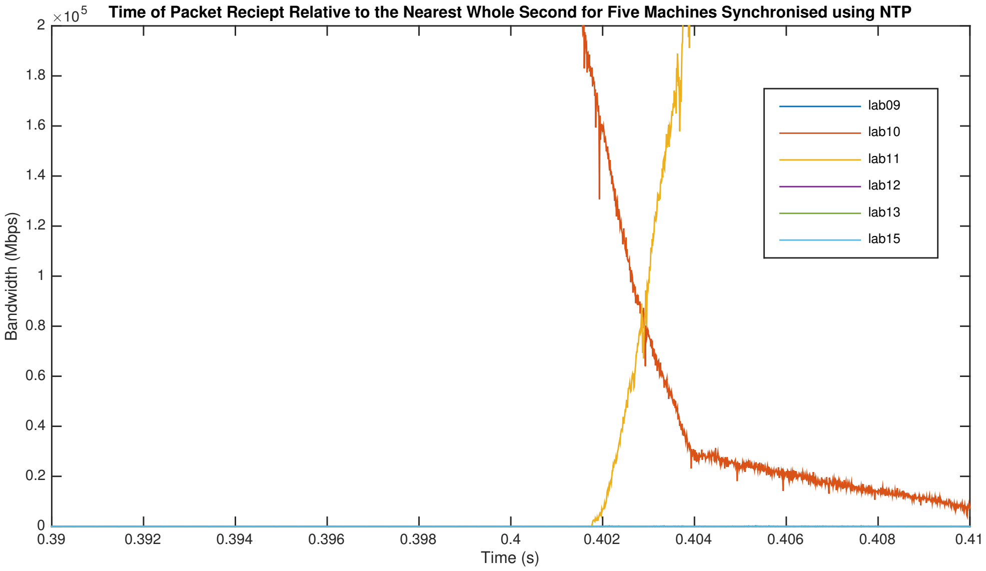

Figure 6: Portion of Figure 4 showing the offset of the start of the broadcast

able that much of the lost bandwidth is due to the ramp up and the tail from clearing the network card queue and resending dropped packets. The maximum discrepancy between the expected and measured start time of a broadcast was 30 μs, however, as this data is averaged over a significant time period (approximately an hour), this is the earliest that a packet was received during the run, and is therefore an estimate of the maximum negative asynchrony that occurred during this time.

A similar plot for the machines synchronised using NTP is shown in Figure 4. The throughput is slightly, but noticeably, less, and also the discrepancies between the expected and measured start times are significantly greater. Figures 5 and 6 show zoomed in versions of the above figures to more clearly highlight this discrepancy. The average measured offset between expected and measured start time for NTP was greater than one millisecond. It should be noted that in this case, NTP was running on a separate network to the data and thus would not have been impeded by congestion in the switch.

Table 4 shows typical precision statistics generated by the PTP daemon on the lab09 slave during a period of standard minimal use. It need be noted that these statistics are generated with the assumption that the network is symmetric, and any asymmetry will cause errors, however the used test network is symmetric to reasonable degrees of approximation and therefore these statistics should be

time	mean	min	max	samples	start-time	end-time
minute[-1]	-112.510	-2932	3344	1426	2015-09-07 09:49:05	2015-09-07 09:50:05
minute[-2]	-87.184	-2574	2965	1388	2015-09-07 09:48:05	2015-09-07 09:49:05
minute[-3]	-4.018	-2111	2101	1410	2015-09-07 09:47:05	2015-09-07 09:48:05
ten-minutes[0]	-37.813	-2932	3344	8480	2015-09-07 09:44:05	—
ten-minutes[-1]	3.031	-3765	4774	14212	2015-09-07 09:34:05	2015-09-07 09:44:05
ten-minutes[-2]	-10.325	-10798	4288	14247	2015-09-07 09:24:05	2015-09-07 09:34:05
hour[0]	-2.648	-12192	5393	79276	2015-09-07 08:54:05	—
hour[-1]	1.506	-13392	5290	84517	2015-09-07 07:54:05	2015-09-07 08:54:05
hour[-2]	2.523	-15315	5984	84828	2015-09-07 06:54:05	2015-09-07 07:54:05
day[0]	-0.058	-15315	13624	2029056	2015-09-06 09:54:05	—
day[-1]	0.307	-23005	48754	2035584	2015-09-05 09:54:05	2015-09-06 09:54:05
week[0]	0.125	-23005	48754	4064640	2015-09-05 09:54:05	—

Table 4: Automatically generated statistics of time offset from master (in nanoseconds) for a period of minimal usage.

time	mean	min	max	samples	start-time	end-time
minute[-1]	-64.770	-22058	23510	1414	2015-09-08 09:45:05	2015-09-08 09:46:05
minute[-2]	-54.239	-12059	20901	1470	2015-09-08 09:44:05	2015-09-08 09:45:05
minute[-3]	-1.527	-9334	22301	1470	2015-09-08 09:43:05	2015-09-08 09:44:05
ten-minutes[0]	-59.403	-22058	23510	2884	2015-09-08 09:44:05	—
ten-minutes[-1]	-135.321	-22425	28597	14612	2015-09-08 09:34:05	2015-09-08 09:44:05
ten-minutes[-2]	200.496	-17378	36214	14526	2015-09-08 09:24:05	2015-09-08 09:34:05
hour[0]	-32.646	-27315	98409	87370	2015-09-08 08:54:05	—
hour[-1]	-41.756	-24874	66377	87244	2015-09-08 07:54:05	2015-09-08 08:54:05
hour[-2]	18.422	-24942	29746	87083	2015-09-08 06:54:05	2015-09-08 07:54:05
day[0]	-41.117	-711044	3644672	2044559	2015-09-07 09:54:05	—

Table 5: Automatically generated statistics of time offset from master (in nanoseconds) for a period of heavy usage.

adequately reliable [4] [b]. It can be seen that the precision is generally within 50 μs, and has accuracy in the nanosecond, or even subnanosecond, range over longer time periods. This is achievable because the switch is under low loading and is able to pass PTP packets through promptly, causing little jitter in the network latency. This allows the protocol to accurately estimate the one way delay to the master and adjust the system clock frequency to maintain excellent synchronisation with the master clock.

In comparison, Table 5 show the same statistics after the network has experienced 24 hours of heavy use [c]. It is clear that the accuracy and precision of the synchronisation is degraded, reporting an average accuracy of tens of nanoseconds, and a maximum asynchrony of the order of a few milliseconds. However, it is likely that these statistics under high loads overestimate the true asynchrony of the system. Because there is no way for a slave machine to objectively compare its clock to that of the master, these statistics are generated by comparing the internal clock of the slave to the time stamp of a received packet adjusted with the current estimate of the network latency. PTP relies upon consistent network latency and, when there is a high network load, the buffers in the switch and the network cards fill up causing erratic delays that make it practically impossible to accurate determine the network latency. This means that using the comparison of the slave's clock time to that of the incoming packet as an accurate metric of the synchronisation between the two becomes, if not fruitless, then vacuous. Despite neither this measurement, nor the discrepancy between the expected and measured start time discussed above, provide accurate estimations of the time difference between the master and slave, they show worst case values of the discrepancy which is sufficient to determine that PTP significantly outperforms NTP, and provides sufficient synchronisation for many applications.

4 Conclusion

Precision Time Protocol, while not able to obtain the same time synchronisation as a dedicated clock line, is an impressive step up from previous technologies. Using high speed networks, PTP is capable of achieving this precision without the need for custom switches or modified network cards, which makes it viable for use in data acquisition systems. It is possible that greater precision with improved resilience to heavy network loads could be achieved by using higher speed networks such as Inifiniband or 100 Gigabit Ethernet. Further tests on a larger test network would be required before implementation on a significantly larger scale system to ensure that the performance is independent of the network size, however one would not expect this to be a problem.

[b] Symmetry is a reasonable assumption for local area networks, however wide area networks often use more sophisticated routing which can cause asymmetry.

[c] This use was the barrel-shifting traffic shaping described above running at full bandwidth.

5 References

[1] A. Dreher, D. Mohl. (2010). *Precision Clock Synchronization IEEE 1588*. Hirschmann Automation and Control GmbH.
Available at: `http://www.industrialnetworking.com/pdf/Hirschmann_IEEE_1588.pdf` [2015, August 28].

[2] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, G. Gaderer. *White Rabbit: Sub-Nanosecond Timing Distribution over Ethernet*, ISPCS 2009, Brescia, Italy.

[3] PTPd website,
see `http://ptpd.sourceforge.net/`

[4] Solarflare Communications. (2014). *Solarflare Enhanced PTP User Guide*. SOLARFLARE Communications, Inc.
Available at: `https://support.solarflare.com/` [2015, July 14]

[5] White Rabbit Project,
see `http://www.ohwr.org`

6 Appendix

`https://gitlab.cern.ch/doliver/LHCb_PTP_Test` contains a Git repository of the programs used for these results. This includes a server application to be run on the master server, and a client to be run on each of the slaves. An included README gives further details for those who are interested.