FPGA-Based Testbed for Synchronization on Ethernet Fronthaul with Phase Noise Measurements

Joary Paulo*, Igor Freire*, Ilan Sousa*, Chenguang Lu†, Miguel Berg†, Igor Almeida† and Aldebaro Klautau*
*Signal Processing Laboratory, Federal University of Pará, Brazil
{joary,igorfreire,ilan,aldebaro}@ufpa.br
†Ericsson Research, Kista, Sweden
{chenguang.lu, miguel.berg, igoralmeida}@ericsson.com

***Abstract*—Cloud radio access network (C-RAN) is a recent trend of RAN architecture positioned to help the operators to address challenges of new wireless services, such as emerging 4G and 5G mobile networks. C-RAN uses baseband processing units in a central server which connects to the radio front-ends at cell sites via the so-called fronthaul network. The fronthaul infrastructure is currently provided by CPRI (Common Public Radio Interface) and OBSAI (Open Basestation Architecture Initiative) industry standards which use dedicated optical links with high deployment costs. An alternative is to use Ethernet technology aiming to reuse of network infrastructure available in many commercial buildings. However, in contrast to the traditional synchronous fronthaul, Ethernet suffers with packet delay variation (PDV) and challenging synchronization recovery. This work presents a complete and flexible testbed to evaluate Ethernet-based fronthaul. The system is validated via extensive measurements that show the effects of synchronization procedures and network impairments on regenerated clock phase noise.**

I. Introduction

Cloud radio access networks (C-RAN) provide key solutions for efficient allocation and management of baseband processing resources [1], which are essential to forthcoming *ultra-dense* [2] deployments. However, centralization demands for increased flexibility in the *fronthaul*, either in terms of infrastructure for new installations or in terms of IQ (In-phase and Quadrature) data traffic routing. Thus, Ethernet has been investigated by standardization task forces such as IEEE 1904.3, IEEE 802.1CM and IEEE1914.1, aiming at further evolving current fronthaul protocols such as CPRI and OBSAI to support Ethernet [1], [3].

Synchronous fronthaul implementations deliver synchronization signals through line timing paths formed at the physical layer of cascaded nodes, while conventional Ethernet deliberately uses free-running clocks on network nodes. In order to provide synchronization on Ethernet systems, other solutions can be used such as the Synchronous Ethernet (SyncE) [4] and the Precision Time Protocol (PTP) [5]. Since fronthaul networks are relatively recent, a current problem is to ensure the accuracy required by 3GPP [3] is achievable through such solutions.

Particularly for PTP, Packet Delay Variations (PDV) represents the main limitation to accuracy. For example, the work shown in [6] concludes that the fronthaul requirements for jitter cannot be satisfied unless schemes such as frame preemption [7], traffic scheduling [8] or de-jitter buffering are used to alleviate PDV. Such strategies effectively reduce PDV in the network, but generally require equipment upgrades.

A related effort was published in [9], namely an Ethernet-based fronthaul testbed. The work, however, considers a scenario of Ethernet over optical links. In contrast, we assume our usage scenario is over legacy Ethernet infrastructure that exploits copper cables, aiming to allow reuse of existent infrastructure in many buildings.

This work presents a complete and flexible testbed to evaluate Ethernet-based fronthaul over copper cables. The prototype described in this work transmits IQ data traffic and PTP synchronization packets over the same shared fronthaul link. The testbed allows investigation of PDV and network impairments on PTP packets and is validated via measurements showing the effects of synchronization procedures and network impairments on the regenerated clock phase noise.

The work is organized as follows: Section II presents the background concepts on CPRI over ethernet and PTP synchronization, Section III explains the testbed implementation details, Section IV evaluates Phase-Noise results of network synchronized clock and, finally, Section V shows the conclusions of testbed evaluation.

II. Background Concepts

This section describes basic concepts used throughout this work.

A. CPRI and Ethernet

The CPRI specification [10] defines a point-to-point link between a Base Band Unit (BBU) and a Remote Radio Unit (RRU), or between two cascaded RRUs. CPRI defines layer 1 (physical layer) and layer 2 (link layer) for single-hop and multi-hop topologies. Its PHY layer supports both electrical and optical interface, while the link layer includes features such as media access control, flow control and protection of the data in the control and management flows. Since CPRI PHY is synchronous, the RRU can recover BBU's clock directly from received signal.

The CPRI framing is formed by a Basic Frame (BF) with 16 words: one for control, the so-called control word (CW), and 15 for IQ data, as shown in Fig. 1. The BF rate is the same Chip Rate (3.84 MHz) inherited from CDMA (Code

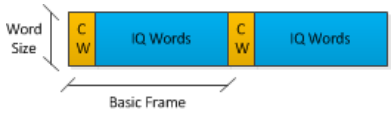

Fig. 1. CPRI framing format.

Division Multiple Access) mobile technologies and adopted in LTE standards. The link data rate is defined by the word length, for example in CPRI profile 1 the word size is 8 bits so that the BF is of 128 bits and the line bit-rate becomes 128 [bits] $\times$ 3.84 [MHz] $\times$ $\frac{8}{10}$ $=$ 614.4 [Mbps], the last factor is due to 8b/10b line coding, deployed in CPRI.

B. PTP Engine

The clock synchronization is performed in this testbed using PTP specified in IEEE1588 [5], that defines a generic two-way exchange of timestamped packets to provide time synchronization between two clock units. PTP also defines a master-slave architecture, where a node in the network can operate in either master or slave mode. The master must provide a very accurate time-of-day information, and the slave disciplines its local clock based on estimations made with network exchanged timestamps.

PTP also defines the so-called *ordinary clock* (OC) a node that contains a single physical port (e.g. a single Ethernet interface) and, therefore, can only operate as master or slave. In contrast, *boundary clock* (BC) and *transparent clocks* (TC) are PTP nodes with multiple ports. Example of BC and TC are Ethernet switches with PTP support, some of it ports are master and a single port necessarily is slave, since there is just one clock source in the network. Ideally, like in the ITU-T 8275.1 profile [11], BCs or TCs would be used in the network because of their features to avoid effects of PDV, by updating the so-called correction field of PTP messages. However, nothing prevents an end-to-end architecture with regular switches (without PTP support) to be used, which is the case of interest in this work.

When operating in master mode, a PTP node initiates a PTP processing cycle by emitting the so-called SYNC message to slave. This message can be sent with a minimal rate of one packet every 16 seconds, up to a maximum rate of 128 packets-per-second, which ideally is sufficient with respect to the frequency of physical variations in telecom-grade oscillators. Upon reception of a SYNC packet, the slave device immediately sends DELAY_REQ message back to master. Finally, the master replies with DELAY_RESP message. This exchange is known as *delay request-response mechanism*, which is illustrated in Fig. 2.

The operation of Fig 2 involves exchange of four timestamps: t_1, t_2, t_3 and t_4. Timestamps t_1 and t_4 are taken in master, while t_2 and t_3 are taken in slave. The first timestamp (t_1) represents the time that the master emits a SYNC message. It can be conveyed either within a SYNC message itself (defined as *one-step mode*) or in the so-called FOLLOW_UP message (*two-step mode*), depending on the accuracy of the

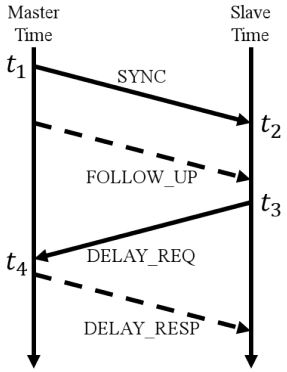

Fig. 2. Basic message exchange in the IEEE 1588 delay request-response mechanism [5].

embedded timestamping unit. Upon reception of a SYNC message, the slave measures the SYNC arrival time t_2. Next, the slave sends a DELAY_REQ message to master and stores its transmit time t_3. Finally, when the DELAY_REQ message is recognized at the master, its arrival time t_4 is taken and replied back to the slave in the DELAY_RESP message. The timestamps are also illustrated in Fig. 2.

The aforementioned timestamps are specified [5] to be composed of seconds and nanoseconds fields, represented with 48 and 32 bits, respectively. The timestamp nanoseconds field itself should be always less than 10^9, namely within 30 bits. The nodes also generally store an additional information, the fractional nanoseconds count, which is not sent over the network, except by transparent switches in the so-called correction fields. Furthermore, messages exchanged in the protocol have an associated source port identity, which informs the egress PTP node and interface within the node. A responder replies back this information so that the requestor can check whether a valid exchange of timestamps was carried.

The master is supposed to be locked to a Primary Reference Clock (PRC), which is a highly precise clock with typical accuracy of 10 parts-per-trillion, also known as Stratum 1 accuracy. As a result the slave is the one that performs clock corrections, computing through PTP the time and frequency-offset of its local RTC module with respect to the master clock.

III. Developed Testbed

The proposed testbed uses legacy Ethernet network and FPGA boards to evaluate packet switched CPRI traffic, addressing solutions for baseband IQ data exchange and synchronization. The prototype was designed with VHDL and C languages, therefore highly programmable. IP Cores have also been used.

The testbed is formed by a BBU which is also a master PTP OC, connected through a switch to a RRU which is also a regular PTP OC. The scheme is illustrated by the building blocks in Fig. 3. Fig. 4 shows how BBU and RRU are connected to the Ethernet switch during tests. The next subsections describe more about the building blocks of the platform.

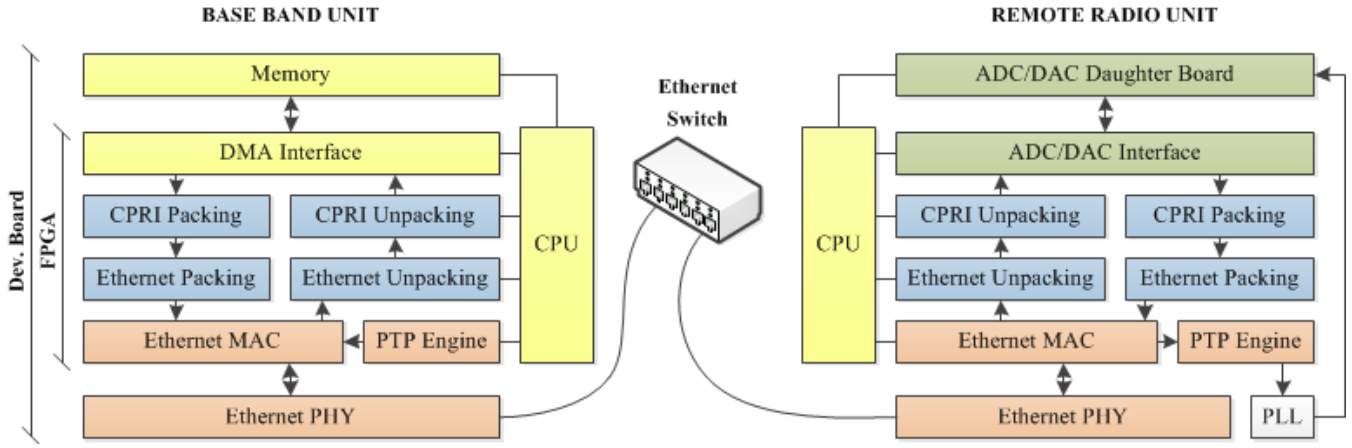

Fig. 3. Testbed flow-graph overview.

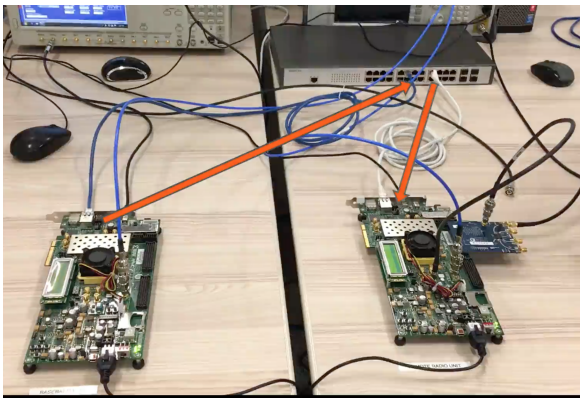

Fig. 4. Hardware used in the testbed, emphasizing the two FPGA boards emulating the BBU and RRU, and the switch that connects them.

A. CPRI to Ethernet mapping

The encapsulation of CPRI BFs into Ethernet frame is an ongoing discussion within the IEEE1904.3 and IEEE1914.1 working groups. These specifications try to optimize the usage of Ethernet by splitting data and control into different streams. The presented testbed uses a simpler approach encapsulating a fixed number of BFs into an Ethernet frame: to reduce the overhead there is no IP header on the generated frame, which forces the network to be an L2-switch only.

In order the reduce the data rate requirements, BFs are encapsulated into Ethernet frames without the 8b/10b coding, which results in a data rate of 491 Mbps for profile 1 of CPRI. Hence, it is theoretically possible to transport a CPRI profile 2 with un-encoded line-rate of 983.04 Mbps, but PTP streams with concurrent radio traffic close to the network maximum will be more affected by network fluctuations. Moreover, it is necessary to take into account concurrent streams such as the PTP packets used on synchronization procedures.

B. PTP Synchronization

As discussed before, the PTP protocol exchanges packets with timestamps over the network as shown in Fig. 2. The first estimation to be made with this data is the one-way link delay and can be calculated according to (1), where master-to-slave (t_{ms}) and slave-to-master (t_{sm}) delays are assumed to be the same.

$$\hat{d} = \frac{t_{ms} - t_{sm}}{2} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}, \tag{1}$$

Based on the delay estimation it is possible to calculate the time error between clocks according to (2), where both the estimated delay ($\hat{d}$) and also a random variable representing the impairments from network uncertainty (γ) are used.

$$x = t_2 - (t_1 + \hat{d} + \gamma) \tag{2}$$

Finally it is possible to estimate the instantaneous frequency-offset based on two sequential realizations of SYNC messages, according to (3). In this expression the value of θ_d is the packet delay difference between master-to-slave and slave-to-master realizations, which represents noise to the frequency-offset estimation. Since the Ethernet network have naturally high PDV the y value is also processed by a moving-average filter to ensure more stable estimation.

$$y = \frac{(t'_{2,k} - t_{1,k}) - (t'_{2,k+1} - t_{1,k+1} + \theta_d)}{t_{1,k} - t_{1,k+1} + \theta_d} \tag{3}$$

Due to the relative stable clock nature the relation between master and slave clocks can be assumed to be a first-order function $t_s = f(t_m) = y \times t_m + x$, where y is the frequency-offset and x is the time-offset. This model is a simplistic approximation and ignores the frequency drift and phase noise terms from the model of [12]. It is a reasonable assumption considering the frequency drift is a slow process with respect to the PTP packet exchange periods. Even with noisy offset estimations it is possible by processing then to choose a correction value to assure a better synchronization quality. These estimations were implemented as part of the testbed firmware.

Finally, the external jitter attenuator PLL filters the PTP-synchronized clock. The PTP recovered clock, which has a

frequency of 8 kHz, is delivered to a chip outside the FPGA fabric in order to convert its frequency to 40 MHz and reduce the signal jitter. The higher frequency is required by analog-to-digital converter board as explained in the next section. The phase noise measurements presented in this work have been taken from the 40 MHz signal.

C. FPGA Resources

The presented testbed is implemented on FPGA evaluation boards. The hardware components used are two Virtex-7 VC707 boards and one Ethernet switch along with measurement equipments. Each FPGA represents one endpoint of the fronthaul link (BBU and RRU). The VHDL code can be divided into subsystems, namely: Ethernet, CPU, ADC/DAC and EthernetFronthaulController.

The Ethernet infrastructure is provided by an on-board PHY (Marvel 88E1111) and a Xilinx's MAC soft-core [13] in the FPGA chip. The MAC has a hardware-assisted PTP support, which means the hardware sends PTP frames but packet data (timestamps) are processed in software.

The CPU is implemented by a microprocessor soft-core [14] with support to on-board peripherals, such as DDR3, UART and JTAG. This subsystem plays an orchestrator role. Its function is to configure and run systems, since the majority of processing is made by coprocessors in hardware by other subsystems.

The ADC/DAC subsystem is a soft-core block that interfaces to the FMCOMMS2 board, a complete software-configurable RF frontend. This board uses the AD9361 chip, which is a transceiver designed for 3G and 4G base stations applications.

Finally, EthernetFronthaulController is an in-house developed subsystem that takes care of CPRI encapsulation and Ethernet packing, being responsible for transmission of IQ data over the network, flow-control and transmit/receive queue management on both ends of the link. This subsystem also includes the software necessary to process and filter PTP timestamps gathered on the Ethernet subsystem.

The described subsystems integrates the testbed as shown in Fig. 3. An important observation is how much FPGA resource is used on the setup and these values are shown in Tab. I. The next section shows validation results of the implemented infrastructure.

TABLE I
HARDWARE UTILIZATION FOR BBU AND RRU.

Resource	Utilization				Available
	BBU		RRU		
	Qty	%	Qty	%	
FlipFlops	62593	10.14	91871	15.12	607200
LUT	61076	20.12	87431	28.80	303600
Memory	4891	3.74	6142	4.70	130800
BRAM	33.5	3.25	98.5	9.56	1030
DSP48	4	0.14	52	1.86	2800
MMCM	2	14.29	2	14.29	14

IV. PHASE-NOISE EVALUATION OF PTP SYNCHRONIZED CLOCK

The testbed was validated with measurements of *phase noise* made using the vector signal analyzer Keysight 9010A with typical specification of -116 dBc/Hz at 100 kHz. As explained before, the synchronized clock has a frequency of 8 kHz and is used as a reference to a jitter attenuator (Si5324) in order to generate a 40 MHz clock. The measurements in this work have been made on the filtered and jitter-attenuated 40 MHz signal. Furthermore, throughout the experiment PTP messages rates of 128 packets-per-second (pps) and 8 pps were used for the one-way sync transmissions (from master to slave) and peer-delay mechanism (c.f. [5]), respectively.

The synchronized 8 kHz clock is derived at the PTP-synchronized real-time clock that is used in the PTP timestamping unit. Since messages are traversed by legacy Ethernet, timestamps have a great amount of noise introduced by packet delay variations, specially due to queuing delay and other network impairments. Hence, it is common practice [15]–[17] to filter them to assure attenuation of noise on measurements.

The measurements of one-way delay ($\hat{d}$) and frequency-offset (y) are filtered to attenuate estimation noise due to Ethernet PDV. The frequency-offset value is processed by a moving-average filter with window length configured to 128 samples. The time offset is processed by a selection filter implemented as a estimation buffer, when the buffer is full the selection algorithm chooses the best time-offset value. The testbed offers configurable possibilities of selection algorithm and buffer sizes, but this work uses sample mean [16] with a buffer size of 256.

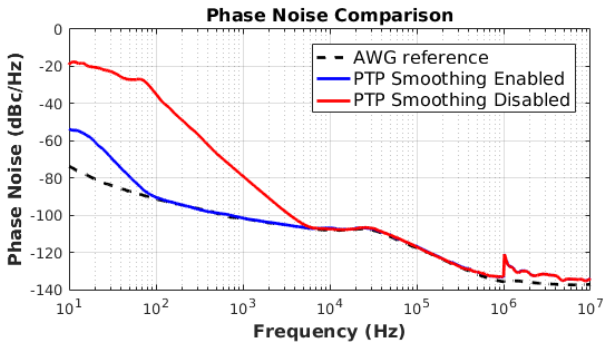

Fig. 5. Phase noise comparison of enabled versus disabled timestamp smoothing.

Fig. 5 compares the phase noise measurements with and without the smoothing of time-offset estimations. It is possible to observe a difference of approximately 20 dBc/Hz between both phase noises. Furthermore, a reference measurement made with an Arbitrary Waveform Generator (AWG model 7082C) is presented for comparison, serving as a reference low-noise clock source.

Another relevant measurement is the phase noise difference between a PTP synchronized and a free-running (not disciplined) clock. Fig. 6 shows that a free-running clock has a

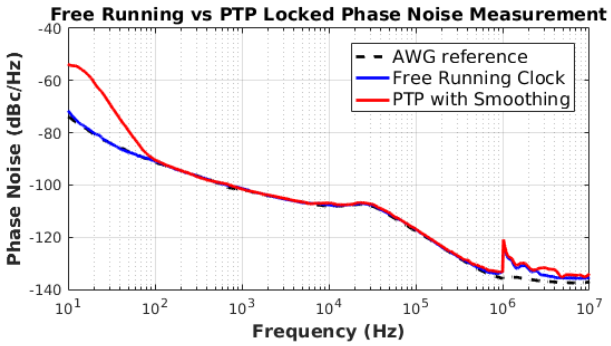

Fig. 6. Phase noise comparison of PTP locked and free-running clocks.

lower phase noise, as expected due to the fact that the RTC time used to generate the clock is not being frequently updated using time and frequency-offset estimations. However, in this case the fronthaul would not achieve time synchronization, Hence there is a tradeoff between phase-noise and timing alignment. The PTP corrections (primarily of time error) insert some noise aiming a better time alignment, so the processing made on PTP timestamps should be both precise for better timing performance and smooth for low phase-noise.

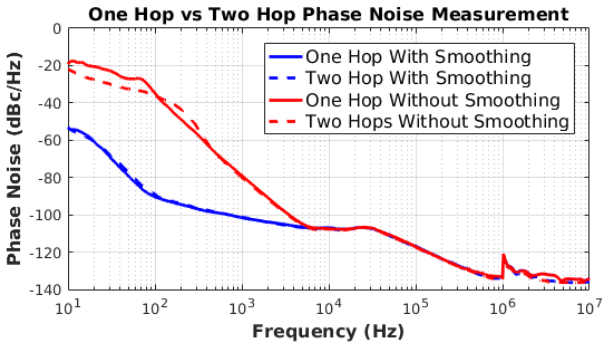

Fig. 7. Phase noise comparison of two vs one hop network link.

Finally, it is essential to evaluate phase-noise impairments due to the increase on the number of hops over the network link. Fig. 7 shows the results for one and two network hops when smoothing is enabled and disabled. It is possible to see that in this particular scenario an increment in the number of hops on the fronthaul link did not impact the phase-noise. This is the kind of issue that a flexible fronthaul, as the proposed one, allows to investigate.

V. Conclusions

This work presented a flexible FPGA-based testbed to evaluate Ethernet-based fronthauls. This testbed is very useful for research in this area given the increasing interest in packet-based fronthauls, motivated by the decrease it provokes in the total cost of C-RAN deployments. The testbed was validated by experiments that observed the effects of synchronization procedures and network impairments on the regenerated clock phase noise. The obtained results indicated that the testbed allows the development and tuning of synchronization algorithms for packet-based.

Acknowledgment

This work was supported in part by the Innovation Center, Ericsson Telecomunicações S.A., Brazil, the Capes Foundation, Brazil, and by the European Union through the 5G-Crosshaul project (H2020-ICT-2014/671598).

References

[1] C.-L. I, J. Huang, R. Duan, C. Cui, J. Jiang, and L. Li, "Recent progress on C-RAN centralization and cloudification," *Access, IEEE*, vol. 2, pp. 1030–1039, 2014.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5g be?" *Arxiv preprint*, pp. 1–17, 2014.

[3] D. Bladsjö, M. Hogan, and S. Ruffini, "Synchronization aspects in LTE small cells," *Communications Magazine, IEEE*, vol. 51, no. 9, pp. 70–77, sep, 2013.

[4] ITU-T, "Packet delay variation network limits applicable to packet-based methods (frequency synchronization)," feb, 2012.

[5] IEEE Instrumentation and Measurement Society, "IEEE 1588-2008: Standard for a precision clock synchronization protocol for networked measurement and control systems," jul, 2008.

[6] T. Wan and P. Ashwood, "A performance study of CPRI over ethernet," *IEEE 1904.3 Task Force*, 2015.

[7] IEEE Computer Society, "802.1Qbu - Frame Preemption for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks," mar 2012, IEEE Standard for Local and metropolitan area networks.

[8] ——, "802.1Qca - Path Control and Reservation for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks," mar 2012, IEEE Standard for Local and metropolitan area networks.

[9] D. Riscado, J. Santos, D. Dinis, G. Anjos, D. Belo, N. B. Carvalho, and A. S. R. Oliveira, "A flexible research testbed for c-ran," in *Digital System Design (DSD), 2015 Euromicro Conference on*, Aug 2015, pp. 131–138.

[10] "Common Public Radio Interface (CPRI) specification v6.1," http://www.cpri.info, Jul. 2014.

[11] ITU-T, "Precision time protocol telecom profile for phase/time synchronization with full timing support from the network," jul, 2014.

[12] ——, "Definitions and terminology for synchronization networks," International Telecommunication Union, Rec. G. 810, aug 1997.

[13] Xilinx, Inc., "AXI 1G/2.5G Ethernet Subsystem v7.0," PG 138, sep 2015.

[14] ——, "MicroBlaze Processor Reference Guide," mar 2009, UG081.

[15] M. Anyaegbu, C. X. Wang, and W. Berrie, "A sample-mode packet delay variation filter for ieee 1588 synchronization," in *ITS Telecommunications (ITST), 2012 12th International Conference on*, Nov 2012, pp. 1–6.

[16] I. Hadžić and D. R. Morgan, "Adaptive packet selection for clock recovery," in *2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Sept 2010, pp. 42–47.

[17] ——, "On packet selection criteria for clock recovery," in *2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Oct 2009, pp. 1–6.