Freire, I., Sousa, I., Bemerguy, P., Klautau, A., Almeida, I., Lu, C. y Berg, M. (2018). Analysis of Controlled Packet Departure to Support Ethernet Fronthaul Synchronization via PTP. In: *2018 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS),* Geneva, Switzerland, pp. 1-6.

DOI: 10.1109/ISPCS.2018.8543068

Analysis of Controlled Packet Departure to Support Ethernet Fronthaul Synchronization via PTP

Igor Freire, Ilan Sousa, Pedro Bemerguy and Aldebaro Klautau
LASSE - 5G & IoT Research Group,
Federal University of Pará, Brazil
{igorfreire, ilan, aldebaro}@ufpa.br
pedro.bemerguy@itec.ufpa.br

Igor Almeida*, Chenguang Lu† and Miguel Berg†
*Ericsson Research, Indaiatuba, Brazil
†Ericsson Research, Kista, Sweden
{igor.almeida, chenguang.lu, miguel.berg}@ericsson.com

***Abstract*—The synchronization accuracy achieved via the IEEE 1588 Precision Time Protocol (PTP) in packet-based fronthaul networks is substantially impaired by packet delay variation (PDV). Nevertheless, in the particular case of deployment over tree topologies, it is known that PDV can be avoided by controlling the departure of PTP packets such that they experience close to constant delays over the fronthaul. This paper analyzes controlled PTP departure under constraints that are peculiar to a fronthaul scenario of interest and considering that radio traffic itself behaves as background traffic relative to PTP. Since the method involves buffering of radio traffic prior to controlled PTP transmissions, its impact on buffer sizes at the baseband and radio units, and the corresponding increase in fronthaul latency are also analyzed. In the end, results collected through a self-developed FPGA-based testbed are presented.**

***Index Terms*—PTP, IEEE 1588, Fronthaul, CPRI, Ethernet.**

I. Introduction

There is widespread interest in exploiting packet-based networking and especially Ethernet for the fronthaul (FH) of cloud radio access networks (C-RAN). The goal is to improve the flexibility and cost efficiency of the FH with respect to current interfaces based on synchronous and dedicated transport channels. This is observed for example in efforts such as IEEE 802.1CM and the eCPRI specification of the Common Public Radio Interface [1], which support packet and Ethernet-based FH architectures.

A distinguishing property of the packet-based FH is that it does not inherently achieve time synchronization between baseband units (BBUs) and remote radio units (RRUs). Hence, it requires an explicit packet exchange dedicated for time synchronization, such as the one provided by the IEEE 1588 Precision Time Protocol (PTP) [2]. This is required in order to support time-aligned transmissions among geographically apart RRUs, where the RRUs rely on time recovered from a common primary reference [3], [4]. Furthermore, time recovery supports the disciplining of the carrier and sampling frequencies at the RRUs, and can also benefit the FH itself, e.g. for flow control between BBUs and RRUs [5].

PTP is a cost-efficient solution for timing distribution in the FH and especially interesting for indoor RRUs, where Global Navigation Satellite System (GNSS) becomes inviable. An open challenge, however, is to transport PTP time-multiplexed with FH streams over legacy (PTP-unaware) Ethernet networks and yet comply with the accuracy required for radio transmissions. For instance, the ± 1.5 µs maximum absolute time error required between adjacent LTE timing division duplexing (TDD) small cell base stations or e.g. the 260 ns relative time error between inter-band carrier aggregation transmitters [6]. Currently, the common practice is to rely on PTP-aware transport nodes that are fed with physical layer frequency references and that introduce constant time error below ± 50 ns per hop [7]. Such accuracy is challenging to achieve in PTP-unaware networks due to PDV, which adds noise to the estimations that the PTP slave executes to discipline its clock.

Controlled PTP departure is one way to alleviate PDV effects. As discussed in [8], it is a mechanism for transmitting PTP frames solely when it is guaranteed that their network transit times will not vary considerably. Such a strategy was also alluded for the FH in [9], as bursting of radio traffic prior to PTP transmission on silent periods. Importantly, the technique requires processing solely at BBUs and RRUs and, thus, can be deployed over legacy Ethernet. It is specially interesting for leased FH network segments lacking synchronization, as considered for the ITU-T G.8275.2 Partial Timing Support profile. Nevertheless, the strategy works solely if the path from PTP master to slaves follows a tree topology [8]. Otherwise, it depends heavily on the other (non-PTP) streams in the network, i.e. on background (BG) traffic.

Particularly in the FH, the main BG traffic for PTP is the FH stream itself, namely the stream carrying radio control and user data [10]. Also, backhaul traffic can compose the BG stream in converged fronthaul-backhaul networks [11]. In any case, the vision is that the BBUs and RRUs can coordinate these relative to PTP in order to reduce the PDV.

This work analyzes PTP departure control with formulations in terms of radio and FH parameters. It evaluates the PDV improvement provided by the technique, the added latency and the buffer implications due to the requirement of temporarily pausing the FH traffic prior to controlled PTP transmissions. Results are collected in an FPGA-based testbed that was thoroughly described in our earlier exposition [5].

This work is organized as follows. Section II models the system. Section III investigates the feasibility of departure control under FH traffic. Section IV presents results obtained using the testbed and Section V concludes.

II. System Model and Requirements

When a background (BG) frame and a subsequent PTP frame traverse an Ethernet switch, the interval between them changes due to the *store-and-forward* procedure. More specifically, for a BG frame whose transmission delay is t_{bg} and a PTP message whose transmission delay is t_{ptp}, after a switch, the interval reduces by $t_{\text{bg}} - t_{\text{ptp}}$. This is because a switch stores the incoming frames completely before forwarding them, which introduces a delay of t_{bg} for the BG frame and t_{ptp} for the PTP frame. Hence, the two frames are delayed by different amounts within the switch, so that their relative distance in the outbound link is altered. Considering $t_{\text{bg}} > t_{\text{ptp}}$, the PTP frame is approximated to the preceding BG frame.

Based on this behavior, when a PTP event[1] transmission [2] is requested, a departure control scheme assigns a clearance interval with respect to the preceding BG frame. This interval, henceforward denominated *gap*, should guarantee that after all "approximations" between the two frames over the network, the theoretical starting instant of the PTP frame at the destination's input interface remains separated from the BG start by at least more than t_{bg} plus an inter-packet gap t_{ipg}. Equivalently, the gap must ensure that no store-and-forward procedure will approximate the PTP frame such that it approaches the BG frame and consequently suffers queuing delay.

To derive the required gap, similar to [8], consider that the end-of-frame instant r_{pkt} of a generic frame arriving at the destination after N hops without queuing delay is given by:

$$r_{\text{pkt}} = s_{\text{pkt}} + \sum_{i=0}^{N} \left(t_{\text{pkt}}^{(i)} + \tau^{(i)} \right) + \sum_{i=1}^{N} \xi^{(i)}, \tag{1}$$

where s_{pkt} denotes the start time at the transmitter, $t_{\text{pkt}}^{(i)}$ is the transmission delay to serialize the frame at the i-th device ($i = 0$ is the transmitter), $\tau^{(i)}$ is the propagation delay between the i-th device and its link peer, and $\xi^{(i)}$ is the processing delay at the i-th transport node.

For the sake of simplicity, assume that the processing delay is constant and that the transmission and propagation delays are the same over all hops, such that (1) simplifies to:

$$r_{\text{pkt}} = s_{\text{pkt}} + (N+1)\left(t_{\text{pkt}} + \tau\right) + (N)(\xi). \tag{2}$$

Then, note that queuing delays are avoided for a particular PTP frame when the following condition is satisfied:

$$r_{\text{ptp}} \geq r_{\text{bg}} + t_{\text{ptp}} + t_{\text{ipg}}.$$

Thus, by substituting (2) in r_{ptp} and r_{bg}, it is inferred that the PTP departure instant should be controlled to satisfy:

$$s_{\text{ptp}} \geq s_{\text{bg}} + N(t_{\text{bg}} - t_{\text{ptp}}) + t_{\text{ipg}} + t_{\text{bg}}. \tag{3}$$

[1]Only PTP event messages benefit from departure control, since only these are timestamped upon arrival at the destination.

In this expression, the $t_{\text{ipg}} + t_{\text{bg}}$ parcel is inherent, since the start of a PTP frame succeeding a BG frame of reference must naturally be after the end of this BG frame and with the inter-packet gap. Particularly due to departure control, however, an additional interval equivalent to the sum of "approximations" between the two frames along the network must be provided, i.e. the $N(t_{\text{bg}} - t_{\text{ptp}})$ parcel.

In conclusion, the gap D relative to the start of the preceding BG frame should satisfy:

$$D \geq N(t_{\text{bg}} - t_{\text{ptp}}) + t_{\text{ipg}} + t_{\text{bg}}. \tag{4}$$

This model is valid only when the BG frame of reference does not undergo queuing delays. For example, it is invalidated if a frame of a *cross-traffic* [12] flow is inserted in between the BG and PTP frames by a particular switch over the network. Consequently, it only becomes a valid model under restricted network topologies. For instance, when considering master-to-slave PTP transmissions in a tree FH topology where the BBU (master) is the root node, such as the one evaluated in [13]. The reason is that in a tree topology the communication from BBU (tree root) to RRUs (tree leaves) does not suffer contention in the absence of cross-communication between the RRUs themselves, which is likely a practical case.

In the remainder of the paper, this specific tree topology is assumed. In addition, it is assumed that the BBU and RRUs transmit their radio streams through the same MAC and PHY interfaces that exchange PTP frames, and that the radio data frames alone play the role of in-line BG traffic (see definition in [12]) with respect to the PTP. Furthermore, without loss of generality, we assume the adoption of the FH functional "split E" from [1], namely raw in-phase and quadrature (IQ) samples conveyed within the FH. Finally, it is assumed that the number of samples encapsulated per frame and the IQ size in bits are set to fixed values for relatively long periods such that the BG (radio) frames are transmitted periodically in time with a constant period i_{bg}. Ultimately, the BG traffic becomes composed of constant bit rate fixed-length (CBRFL) frames.

Although the collection of assumptions forms a very specific scenario, it should be noted that the departure control scheme is generic for operation under other FH functional splits and correspondingly other types of BG traffics. The motivation for assuming split E and CBRFL BG traffic as a baseline is to exploit the simplicity in the analytical formulation and, with that, gain insights into using the method in the FH. Besides, CBRFL is also analyzed to match the hardware implementation developed for the FPGA-based testbed.

In what follows, an important point of concern is buffering. The model is such that when the target gap D is larger than the BG interval i_{bg}, the succeeding BG frame must be delayed, as illustrated in Fig. 1. Moreover, under practical deployments, $D < i_{\text{bg}}$ is seldom the case, given D scales with the number of hops N. Therefore, the implementation must properly provide buffering and logic for temporarily pausing the BG stream.

To allow the ensuing buffering analysis, it is instructive to model the gap based on radio parameters. First, note that the

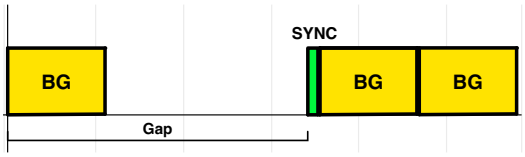

Fig. 1. Serialized frames illustrating the postponing of BG frames past a departure-controlled PTP transmission.

BG frame interval i_{bg} is determined by:

$$i_{\text{bg}} = \frac{(n_{\text{spf}})(T_s)}{n_a}, \tag{5}$$

where T_s is the sampling period, n_{spf} is the number of IQ samples encapsulated per frame, and n_a is the number of antenna streams in the destination RRU that concurrently consume (in the downlink) or produce (in the uplink) IQ samples from (to) the FH. Also, note that n_{spf} includes IQ samples to (from) all antennas of a RRU. These samples are acquired in parallel during n_{spf}/n_a sampling periods.

Meanwhile, the transmission delay t_{bg} is given by:

$$t_{\text{bg}} = \gamma(n_{\text{spf}}) + \mu, \tag{6}$$

where $\gamma = \frac{l_{\text{iq}}}{R_{\text{line}}}$, $\mu = \frac{n_{\text{oh}}}{R_{\text{line}}}$, n_{oh} is the Ethernet overhead length in bits, R_{line} is the Ethernet bit rate and l_{iq} is the IQ sample size in bits.

For multiple RRUs served by a single BBU, depending on how the time-multiplexing of frames to different RRUs is implemented, (5) and (6) may be adapted accordingly. Here, it is considered that i_{bg} does not change with the number of RRUs, since the radio samples are acquired in parallel. Additionally, it is considered that the BBU consecutively sends one BG frame to each RRU during the interval i_{bg}.

III. PTP Departure Control in Fronthaul

The departure control mechanism relies on a module that receives the PTP transmission requests and waits to effectively release the PTP frame only after the gap of D seconds is measured with respect to the start-of-frame (SOF) of the last transmitted radio frame. Once the module receives a PTP transmission request, it temporarily pauses the BG traffic to ensure that no other BG frame reaches the Ethernet MAC (EMAC) before the scheduled PTP departure. After the PTP departure, it resumes the BG traffic, releasing the buffered BG frames in a burst. Fig. 1 illustrates this behavior.

The pause mechanism introduces delay to BG frames and also causes accumulation of radio data on the transmit buffer of the sender, with corresponding depletion of radio data at the destination's receive buffer. In the sequel, this is analyzed with two goals: 1) to assess the capability of maintaining the required FH bit rate even when sporadically pausing the FH stream and 2) to formulate the buffer dimensions that are needed to avoid overflow and underflow problems. For this analysis, it should be assumed that, after crossing the FH, the IQ samples contained in the radio frames are first written into dual-port elastic buffers at the destination (BBU or RRU). On the read side of these buffers, in turn, IQ samples are continuously and synchronously read based on the sample rate.

A. Delay added to buffered BG frames

A controlled PTP frame is placed with interval D from the previous BG frame (for instance BG frame 1). Thus, the PTP end-of-frame is at $D + t_{\text{ptp}}$. Also, considering the inter-packet gap t_{ipg} and assuming no extra delay in the process of restarting the BG traffic past the PTP departure, the first succeeding BG frame (i.e. BG frame 2) to be released after the PTP transmission is spaced by $D + t_{\text{ptp}} + t_{\text{ipg}}$ relative to BG frame 1. Specifically in the case of CBRFL, BG frame 2 otherwise would be only i_{bg} seconds away from BG frame 1, therefore the delay due to departure control becomes:

$$\Delta s_{\text{bg}} = D + t_{\text{ptp}} + t_{\text{ipg}} - i_{\text{bg}}. \tag{7}$$

This delay propagates over subsequent BG frames until it is entirely "self-healed" by transmitting buffered BG frames with intervals shorter than i_{bg} after this event. That is, after the controlled departure, the idle time between delayed BG frames is progressively consumed until the interval between them returns to normal. Since the PTP frames are spaced by much longer periods than BG frames, it is expected that there will be enough time for recovering from the buffering delay.

Considering CBRFL BG traffic with idle interval of $i_{\text{bg}} - t_{\text{bg}}$ between BG frames, the delay can be recovered provided that:

$$\left(\frac{i_{\text{ptp}}}{i_{\text{bg}}}\right)(i_{\text{bg}} - t_{\text{bg}}) > \Delta s_{\text{bg}}, \tag{8}$$

where the ratio between i_{ptp} and i_{bg} approximates the number of BG frames and corresponding idle intervals that exist between two consecutive PTP message transmissions.

Using (4) and (7), it can be shown that the lower bound on the PTP period that ensures sufficient recovery is given by:

$$i_{\text{ptp}} > \frac{N(t_{\text{bg}} - t_{\text{ptp}}) + t_{\text{bg}} + t_{\text{ptp}} + 2t_{\text{ipg}} - i_{\text{bg}}}{\left(1 - \frac{t_{\text{bg}}}{i_{\text{bg}}}\right)}, \tag{9}$$

which is a function of the number of hops N, radio and Ethernet parameters. As N increases, so does the gap D and the delay added to buffered BG frames. Hence, the minimum required PTP period also increases to allow recovery.

Fig. 2 evaluates the minimum PTP period of (9) assuming 1 Gbps Ethernet, 32-bit IQ samples, sample rate of 7.68 MHz, 2 antennas ($n_a = 2$), $n_{\text{oh}} = 224$ bits and $t_{\text{ptp}} = 0.512$ µs, for varying number N of hops and varying n_{spf}. Note the minimum period grows with n_{spf}, since the latter increases the gap D by more than the total idle interval. More importantly, note in all cases $\min\{i_{\text{ptp}}\}$ is well below all standard PTP periods from [2] (minimum is 7.8125 ms). Similar conclusions hold also for other practical Ethernet and FH rates.

Finally, it should be noted that for the sake of simplicity this analysis does not consider the fact that multiple departure-controlled PTP frames can be requested within a given period i_{ptp} (e.g. a "Sync" and a "Delay_Req"). However, since $\min\{i_{\text{ptp}}\}$ is orders of magnitude lower than the actual used PTP periods, this is not expected to be problematic.

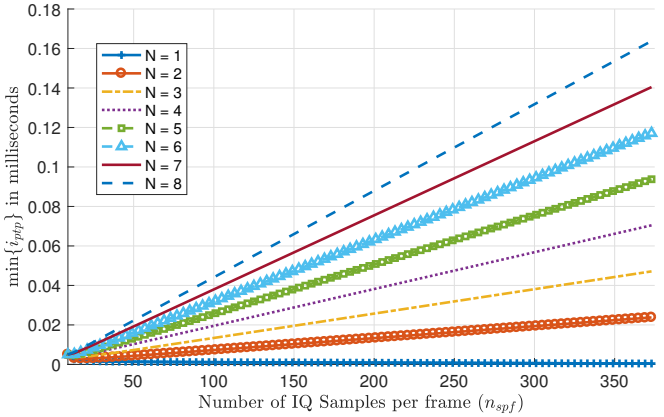

Fig. 2. Minimum PTP periods that provide sufficient time for recovery from an instantaneous delay added by the departure control scheme to BG frames.

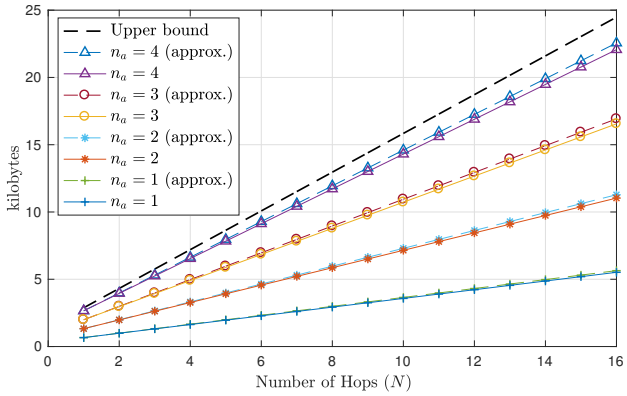

Fig. 3. RRU buffer depletion for varying number of concurrent antenna streams and network hops.

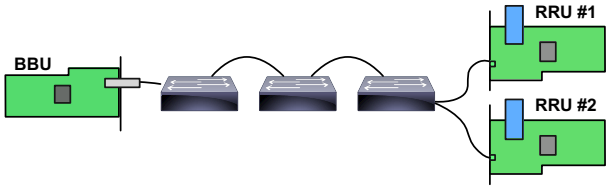

Fig. 4. Testbed composed by three Xilinx FPGA boards, connected via 1 Gbps Ethernet to a switch configured with multiple VLANs.

B. Buffer Design

We now focus on the buffer dimensions required to withstand the BG pause mechanism. In the particular case of master-to-slave PTP transmissions and CBRFL BG traffic, the number of BG frames that would arrive at a given RRU within the gap interval is given by (D/i_{bg}). Hence, the number of bits depleted from the RRU buffer in this interval is approximately:

$$b = \left(\frac{D}{i_{\text{bg}}}\right)(n_{\text{spf}})(l_{\text{iq}}). \tag{10}$$

Meanwhile, the amount of data accumulated at the BBU transmit buffer is scaled by the number of RRUs.

Also, it can be shown that, if the frame's IQ content in bits is much longer than the Ethernet frame overhead, that is, if $(\gamma \cdot n_{\text{spf}}) \gg \mu$, (10) becomes tightly approximated by:

$$b \approx \left(\frac{(N+1)(n_a)(\gamma)}{T_s}\right)(n_{\text{spf}})(l_{\text{iq}}). \tag{11}$$

For a given N, maximum buffering occurs as γ approaches T_s/n_a, i.e. as the IQ sample transmission delay γ over the given Ethernet link approaches the time T_s/n_a to acquire an IQ sample in the radio interface.[2] In other words, it occurs when the FH rate approaches the link capacity. Thus, the RRU buffer depth can be designed considering an upper bound of:

$$b_{\text{ub}} = (N+1)(n_{\text{spf}})(l_{\text{iq}}), \tag{12}$$

which is independent of the Ethernet bit rate, the sampling frequency and the number of concurrent antenna streams.

Fig. 3 considers a 1 Gbps Ethernet FH, with 64-byte PTP frames, $t_{\text{ipg}} = 96$ ns, 30-bit IQ samples ($l_{\text{iq}} = 30$), 384 IQ samples per frame ($n_{\text{spf}} = 384$), $n_{\text{oh}} = 224$ bits, sampling frequency of 7.68 MHz and, more importantly, a varying number n_a of concurrent antenna streams serving a single RRU. Since all antenna streams are acquired in parallel at the RF interface, a higher n_a makes the BG frame content available in less time, for a fixed number of IQ samples. Consequently, as n_a increases, so does the FH rate requirement, such that the curve is expected to approach the upper bound.

The n_a values 1 to 4 in Fig. 3 require FH rates of 234.88 Mbps, 469.76 Mbps, 704.64 Mbps and 939.52 Mbps, respectively. For $n_a = 4$, note that γ (equal to 30 ns) is indeed close to T_s/n_a (32.55 ns) and the curve approaches the upper bound. In particular, note that the highest value in the upper bound b_{ub} curve (for $N = 16$ hops) is equal to 24.48 kilobytes. In terms of latency, this corresponds to approximately 195 µs for $R_{\text{line}} = 1$ Gbps. Nevertheless, note that this worst-case latency drops to 19.5 µs with 10 Gbps Ethernet, since the upper bound in (12) is independent of the line rate. This suggests the latency introduced by the method can be within end-to-end limits of practical FH networks.

IV. Hardware Evaluation

The adopted testbed was comprehensively described in [5]. It was developed using Xilinx Virtex 7 FPGAs to implement BBU and RRU devices. A PTP-capable EMAC with hardware timestamping is instantiated in the FPGA together with a time counter that is fed by a free-running clock and the module that implements the departure control mechanism. At the RRU side (slave) the time counter is disciplined through PTP and used to synthesize a frequency reference that ultimately determines the sampling clock and, correspondingly, the rate that IQ samples are read and written from (into) the FH buffers.

The adopted FH topology is as shown in Fig. 4. The network hops are implemented using independent dedicated port-based

[2]Observe that the radio interface is assumed to acquire n_a IQ samples concurrently during a single sampling period T_s.

VLANs[3] configured in a commercial PTP-unaware switch (Intelbras SG 2404 MR). Also, only FH and PTP traffics are exchanged between the BBU and RRU. The FH traffic, in turn, is configured to convey independent 2x2 LTE 5 MHz baseband streams to each of the two RRUs, particularly encapsulated into frames with $n_{\text{oh}} = 224$ bits. The IQ sample size and the number of IQ samples per frame are varied in the experiments.

To evaluate the PDV performance, first the two-way Pdelay_Req/Pdelay_Resp exchange[4] is departure controlled. The rationale is that a reasonable way to infer PDV improvements is by observing the PTP delay estimations themselves, and these estimations, in turn, would only experience reduction in fluctuation if the frames of the peer-delay mechanism are controlled. However, note that only master-to-slave PTP frames are free from contention in the adopted tree topology, so that departure control of Pdelay_Req/Pdelay_Resp messages would only hold for a single RRU served in the FH.

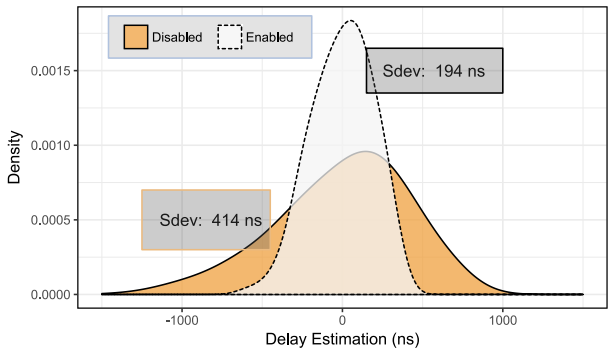

Fig. 5. Delay estimation distribution for 3 hops and 1 RRU.

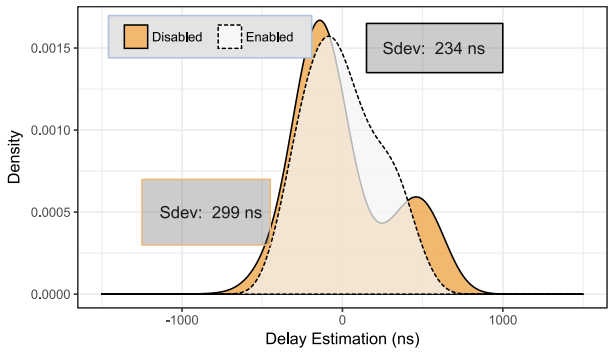

Fig. 6. Delay estimation distribution for 3 hops and 2 RRUs

Fig. 5 shows the distribution of the delay estimations taken at the RRU side[5] during 10^4 Pdelay_Req/Pdelay_Resp exchanges with 3 hops ($N = 3$), $l_{\text{iq}} = 24$ and $n_{\text{spf}} = 64$.

[3]It should be pointed that the method does not work with IEEE 802.1Q VLAN trunks, since these can lead to traffic contention between VLANs.

[4]The peer-delay mechanism [2] is employed in the experiment. Similar results can be obtained with two-way delay request-response exchanges.

[5]Both peers of a link run their own delay estimations.

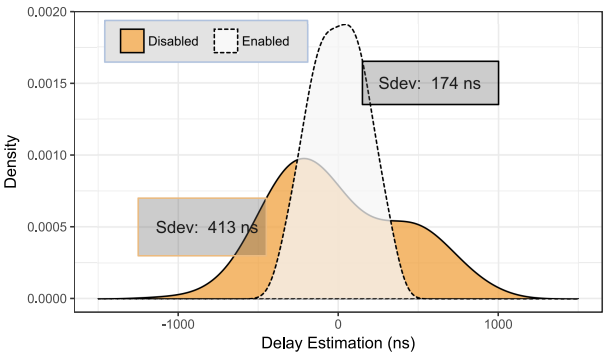

Fig. 7. Delay estimation distribution with gap calculated as if N was 8.

To omit any estimation bias,[6] the distributions were centered around the origin. Note that, with controlled departure, the delay estimation distribution approaches a Gaussian shape, as queuing delays are avoided and the FH delay becomes mostly given by processing delays [8]. Note also that the scheme reduces the standard deviation (Sdev) substantially, in this case by more than a factor of 2.

Next, to gain insight regarding how contention disturbs the departure control in the slave-to-master direction, Fig. 6 plots the delay estimations obtained when serving two RRUs. The controlled departure distribution indeed loses the Gaussian shape, as the departure gap is not guaranteed to be achieved. Nevertheless, since the amount by which a contending BG frame from the other RRU reduces the desired gap is random, it is still possible to benefit from departure control at a lower extent. In Fig. 6, it can be seen that the scheme was still beneficial, as the Sdev was reduced by approximately 22%.

A further analysis concerns the need for margin in the gap computation. Due to simplifications made along the derivation, a perfectly dimensioned gap likely does not fully avoid queuing delays. This is especially due to the assumption of constant processing delays made prior to (2). To assess this, the gap is computed as if N was 8, since this leads to a much higher D. The result is shown in Fig. 7. Note the Sdev is slightly improved in relation to the previous evaluations.

Lastly, the buffer impact is analyzed. For that, it should be assumed that, in normal operation, the rate of samples arriving at the RRUs via the FH is matched to the rate that these samples are consumed towards the analog frontend for transmission in the air interface. As a result, the buffer occupancy oscillates around its midpoint (see [5]). During a departure gap, however, this occupancy is disturbed.

Fig. 8 shows two independent RRU buffer occupancy time series, captured when using one and two RRUs in the FH. To facilitate the interpretation, the curves are shifted vertically by their average values, so that they oscillate around zero, and horizontally, such that the minimum occupancy is centered at the origin. The sawtooth pattern is due to packets arriving

[6]Delay asymmetry and PDV over a PTP-unaware path introduces random biases in both delay and time offset estimations [14].

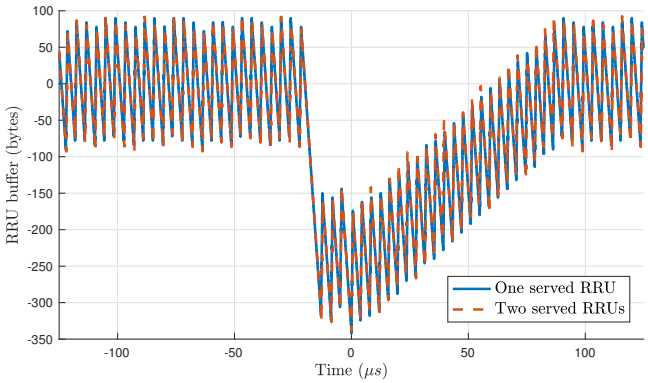

Fig. 8. RRU buffer occupancies during the PTP departure gap.

TABLE I
DEPLETION IN BYTES AT THE RRU BUFFER FOR DISTINCT n_{SPF} AND l_{IQ}

l_{iq} \ n_{spf}	64	96
16	166	188
24	342	486

from the FH (when the occupancy rises) and the continuous consumption of IQ samples for transmission via the air interface. The capture highlights the exact moment when the RRU buffer occupancy drops due departure control at the BBU side and the subsequent recovery once the BG stream is resumed.

The experiment in Fig. 8 adopts $n_{\text{spf}} = 64$, $l_{\text{iq}} = 24$, departure control of 44-byte PTP Sync frames and the 3-hop topology of Fig. 4 with 1 Gbps Ethernet links ($R_{\text{line}} =$ 1 Gbps). The IQ sample rate is 7.68 MHz and each RRU has two independent antenna streams ($n_a = 2$). Consequently, the minimum gap becomes approximately 6 µs and a buffer depletion of $b = 280$ bytes is expected. Note that the depletion in Fig. 8 is close to this value, except for the aforementioned occupancy oscillations. Furthermore, note that the interval from the instant that the occupancy stops dropping to when it restarts rising back to midpoint is within a few microseconds. This is compliant to the interval of roughly 4 µs from (9).

Table I presents the measured buffer depletion for other configurations of l_{iq} and n_{spf}. The number of bytes was captured exactly after a period corresponding to the minimum gap from (4) starting from when the buffer occupancy crosses its mean value for the last time before the gap.

V. CONCLUSIONS

This paper analyzed several aspects of PTP departure control operating under FH background traffic. The work investigated the PDV improvements provided by the method and two feasibility aspects: the impact of the method on the FH stream and the amount of buffering involved in this process. The work presents measurements taken in an FPGA-based testbed and demonstrates that the departure control mechanism is able to provide substantial reduction in the end-to-end delay distribution, which is highly beneficial for PTP performance. It also shows that, even though FH data needs to be buffered prior to every controlled PTP transmission, the PTP message rate is such that buffer occupancy can seamlessly recover its stable level between PTP transmissions. Experimental results validate the formulations used to design the buffer depth.

Future extensions of this work shall adapt the theory to other practical FH topologies and traffic models, for example including aggregation links and variable rate traffic from distinct functional splits. Also, forthcoming work can focus on the time accuracy performance achieved with the method, as well as mechanisms to calibrate the gap in runtime without prior knowledge of the network parameters.

ACKNOWLEDGMENT

This work was supported in part by the Innovation Center, Ericsson Telecomunicações S.A., Brazil, CNPq/Capes, Brazil, and by the European Union through the H2020 collaborative Europe/Taiwan research project 5G-CORAL (grant agreement no. 761586).

REFERENCES

[1] "Common Public Radio Interface: (eCPRI) Specification v1.0," ago 2017.
[2] IEEE Instrumentation and Measurement Society, "IEEE 1588-2008: Standard for a precision clock synchronization protocol for networked measurement and control systems," July 2008.
[3] V. Jungnickel, T. Wirth, M. Schellmann, T. Haustein, and W. Zirwas, "Synchronization of cooperative base stations," in *2008 IEEE International Symposium on Wireless Communication Systems*, Oct 2008, pp. 329–334.
[4] A. Checko et al., "Synchronization challenges in packet-based cloud-RAN fronthaul for mobile networks," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, June 2015, pp. 2721–2726.
[5] Igor Freire et al., "An FPGA-based design of a packetized fronthaul testbed with IEEE 1588 clock synchronization," in *European Wireless 2017; 23th European Wireless Conference*, May 2017, pp. 1–6.
[6] H. Li, L. Han, R. Duan, and G. M. Garner, "Analysis of the synchronization requirements of 5g and corresponding solutions," *IEEE Communications Standards Magazine*, vol. 1, no. 1, pp. 52–58, March 2017.
[7] ITU-T, "Timing characteristics of telecom boundary clocks and telecom time slave clocks," Jan. 2017.
[8] B. Mochizuki and I. Hadzic, "Improving IEEE 1588v2 clock performance through controlled packet departures," *IEEE Communications Letters*, vol. 14, no. 5, pp. 459–461, May 2010.
[9] P. Assimakopoulos et al., "Switched ethernet fronthaul architecture for cloud-radio access networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 12, pp. B135–B146, December 2016.
[10] P. Assimakopoulos et al., "Statistical distribution of packet inter-arrival rates in an Ethernet fronthaul," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 140–144.
[11] T. Deißet al., "Packet forwarding for heterogeneous technologies for integrated fronthaul/backhaul," in *2016 European Conference on Networks and Communications (EuCNC)*, June 2016, pp. 133–137.
[12] I. Hadzic and D.R. Morgan, "On packet selection criteria for clock recovery," in *Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on*, Oct. 2009, pp. 1–6.
[13] T. Wan and P. Ashwood-Smith, "A performance study of CPRI over Ethernet with IEEE 802.1qbu and 802.1qbv enhancements," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.
[14] M. Hajikhani, T. Kunz, and H. Schwartz, "A recursive method for bias estimation in asymmetric packet-based networks," in *2014 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, Sept 2014, pp. 47–52.