

Contention management for Cloud Ran over an optical ring

Dominique Barth¹ et Dominique Chiaroni² et Maël Guiraud¹ et Yann Strozecki¹

¹David Laboratory, UVSQ

²BNokia Bell Labs France

Nowadays challenge for network providers is to design some network which satisfies in the same time a good Quality of Services and a cheap hardware development. This study is a contribution to the N-GREEN project, which has for goal to develop a low cost optical ring technology that ensures some good performances (dataflow, latency,...) without increasing the hardware with some expensive E2E connections. In this paper, we study the compatibility of such a technology with the development of the Cloud Ran, a latency critical application which is a major aspect of the 5G deployment.

Mots-clefs : N-GREEN, Optical ring, Latency, 5G

1 Introduction

Network providers has to design cheapest networks while the amount of datas and online applications significantly increases. Much of these applications have some QoS criteria, like a minimal bandwidth or a maximal latency. The N-GREEN project aims to propose an high performing optical ring while ensuring a minimal cost for providers. Indeed adversely to most of the nowadays solutions to ensure some good QoS [huawei, demander a dominique C], that establish some End to End connection between the nodes, which is extremely expensive, the N-GREEN optical ring is designed to ensure some good performances with few hardware solutions.

This study is based on the Cloud RAN application in the N-GREEN optical ring. Indeed, Cloud RAN is one of the major area of development for 5G, that consists in centralizing the computation units (or **BaseBand Units, BBU**) of the Remote Radio Headers (**RRHs**) in one **datacenter**. The latency of the messages between the BBU and the RRH is critical []. Here, we study the latency of some C-RAN equipment connected through an N-GREEN optical ring.

In the next part of this paper, we expose the specificity of the optical ring, and the model we studied. In sec 3, we study two method to manage the messages arrivals, and their impact on the latency. Finally, in sec 4, we propose and study some deterministic algorithm which allows the C-RAN to not be buffered at all, while the QoS of others messages is also increased. **TODO: Je sais pas si il faut parler de SDN, qui est un keyword a la mode mais peut etre que l'architecture ngreen nous propose, demander a chiaroni**

Related work

TODO: ?

2 Description of the network, model, problematic

2.1 Network modeling

The model exposed here has already been studied for another particular topology of network and with different problematics from the ones of this paper. One can find in [?] the complexity results on the problem proposed, and some algorithmic solutions to solve it in star networks.

2.2 N-GREEN Optical ring

We consider a circular digraph $G = (V, A)$ representing an unidirectional optical ring. The vertices of the graph represents the nodes of the ring. Each arc (u, v) in A is labeled by an integer weight $\omega(u, v)$ which represents the time taken by a message to go from u to v using this arc. A **route** r in G is a directed path, that is, a sequence of adjacent vertices. We assume that each RRH has an associated BBU. In the process we study, a message is sent from an RRH to a BBU on each route at each period, denoted by P . The time is discretized, i.e. a period is sliced into P **slots**. We call forward routes the routes going from the RRH to the BBU and Backward routes the routes going from the BBU to the RRH. The addition of the weight of the two routes is then equal to RS , the size of the ring. We only consider one datacenter into the ring. This mean that all the forward routes have the same destination nodes, and all the backward routes have the same source node.

In this entire subsection, the ring is in broadcast and select mode with an emission release policy. We first propose an algorithmic model for the ring. In 3, we study the behavior of the ring with an opportunistic insertion policies, in which the insertion buffer is managed by two different methods. Finally, we propose and study in 4 some algorithm allowing the C-RAN PDUs to have 0 sojourn time in the stations, while the BE PDUs are as little buffered as possible.

2.3 Into the node : Two kind of traffics

In this study, we focus on the behavior of the N-GREEN optical ring. Therefore, we do not want to consider the issues caused by the upstream network. To refine our model around the N-GREEN study, one must consider the opto-electronic stations behavior. Each N-GREEN station is connected to at least one electronic interface of rate D . as mentioned in sec.?? a station is able to use all the K channels during a slot of duration Z/KD to send a container of capacity Z which contain all the data received by an electronic interface during a macro-slot, that is, the time needed to fill a container at rate D , i.e. Z/D In other words, an N-GREEN station is able to fill a PDU with the datas received during a macro-slot in the electronic domain.

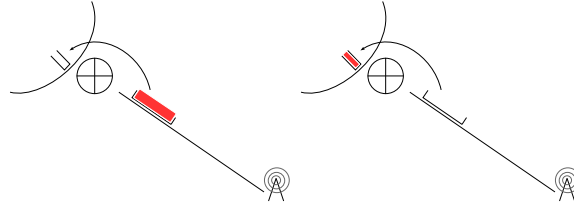


Figure 1: Opto-electronic interface.

An N-GREEN station deals with two kinds of traffics. Some C-RAN and some Best Effort. We assume that each traffic has its corresponding electronic interface. Also, a C-RAN message is of size Z and completely fills a container let us call **C-RAN PDU** such a message. In the same time, the Best Effort SDUs are buffered in a first *gathering buffer* before being gathered together into a **BE PDU**. Once the PDUs are created, there join the *contention buffer* before being sent in the ring.

In this study, we measure the sojourn time W of a PDU in a station. The model we consider in our experiments to generate the BE PDUs is described in sec ??.

C-RAN PDU generation model An RRH usually emit an huge quantity of information during a long time. Also, as we just see, an RRH can send no more than a PDU every $10\mu s$ on the ring. This gap between two PDU from the same RRH is defined by K , the number of channels considered in the ring. We denote by ET , for **Emission time**, the time during which an RRH emits a PDU every K slots on the ring. An RRH emits $\frac{ET}{K}$ PDUs on the ring, every period P .

Back and forth communications As we explained in the introduction of this section, once the datacenter receives some PDUs from an RRH, it computes an answer and send it to the RRH through the ring. To simplify our model, we assume that the computation times is the same for all BBUs and that it is a multiple

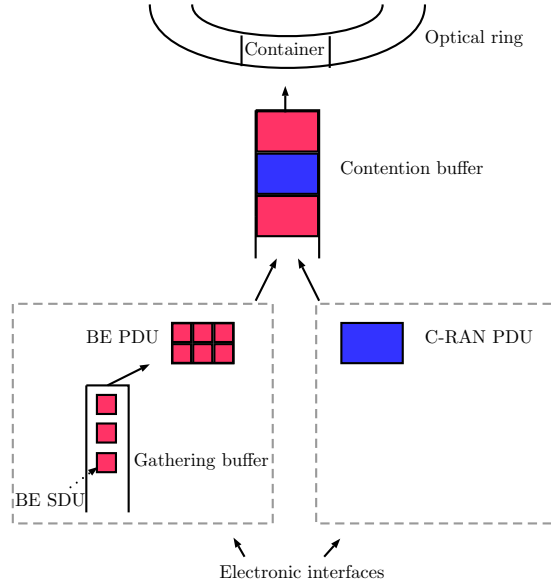


Figure 2: An N-GREEN station.

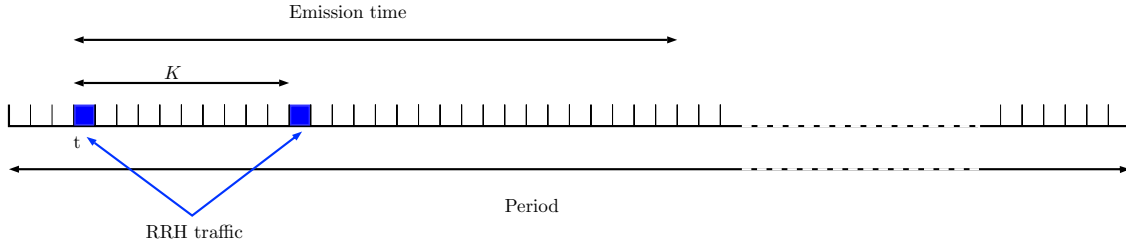


Figure 3: C-RAN PDU generation.

of the period. In this situation, we can assume that, each time the datacenter receives a container with a C-RAN PDU from an RRH, it immediately sends an answer (which is in fact the answer to the two periods ago C-RAN PDU's).

The behavior of the C-RAN PDU generation leads us a new problem, similar to PALL. The set of time slots used by route r at vertex v in a period P , is now defined by $[t(v, r)]_{P, K} = \{t(v, r) + i \bmod P \mid i \in \{0, EP, 2 \times K, \dots, \frac{ET}{K} \times K\}\}$. Consequently, we now want to find a (P, ET) -periodic assignment that set the offsets of the routes m_r , such that there is no collision between two routes r_1 and r_2 , that is, $[t(v, r_1)]_{P, K} \cap [t(v, r_2)]_{P, K} \neq \emptyset$.

The rest of this study is divided in two parts. In a first time, we look at the behavior of the ring and the impact of the PDUs sojourn time with two different method of managing the contention buffer. Then, we propose some algorithms that solve the previous problem of finding a (P, ET) -periodic assignment for the C-RAN PDUs and we compare the impact of those different algorithms on the BE PDUs sojourn time.

2.4 Parameters of the study

The experiment of this subsections are made following the same parameters, until sec 4.2. We set $K = 10$ chanel, some electronic interface of rate $D = 10\text{Gbps}$ and $Z = 12500\text{B}$ so, $Z/KD = 1\mu\text{s}$. The optical ring has a rate $KD = 100\text{Gbps}$, while a slots has a duration of $1\mu\text{s}$. The length of the ring is to 20km in our simulations, this means that the ring has a length of $RS = 100$ slots. We consider some BE SDUs of 125B , and we parametrized BE PDU generator of sec ?? such that each station load the network with an average

Number of channels K	10
Rate of an electronic interface D	10 Gbps
Container size Z	12,500 B
Slot duration Z/KD	1 μ s
Optical ring rate KD	100 Gbps
Length of the ring RS	100 slots
BE SDU size	125B
C-RAN PDU size	12,500B = Z
Emission time ET	500 slots
Period P	1,000 slots
Number of RRH	5
Number of stations n	5

Figure 4: Parameters of the following experiments based on N-GREEN

of 3,5% of BE traffic. We set $ET = 500$ slots, that correspond to a C-RAN traffic of 5Gbps. The period P is to 1ms, that is 1000 slots. We set the number of RRH to $n = 5$. We take 5 stations and one of them is the datacenter. The antennas are apportioned between the others station. With those parameters, the network is theoretically loaded to 67.5%.

3 Performance evaluation of the N-GREEN optical ring

In a first time, we will look at the performance of the N-GREEN optical ring with an opportunistic insertion policy ???. When a station wants to send a PDU on the ring, it uses the first available container . We study the performance of the ring through two methods; a full opportunistic method, and a method that prioritize the C-RAN PDUs. Since we only consider some PDUs arrivals, those two methods are some variants of the management of the PDUs in the station. For those two methods, the offsets m_r of the routes are chosen randomly in the period, i.e. we do not choose the first time at which the RRH start to emit. Since we do not manage those offsets, the two following methods do not gives us some $(P - ET)$ -periodic assignment. No choices are made on the offsets and thus, two routes can collides.

3.1 Full opportunistic method

We first want to look at a simple configuration in which we do not manage the PUDs (BE or C-RAN) arriving into a station. At their arrival, the BE and C-RAN PDUS are buffered in the contention buffer, and this buffer is emptied following the FIFO rule.

The following experiment shows us the performance this **full opportunistic method** with some parameters issued from N-GREEN. On fig 3.1, one can see the cumulative distribution of the sojourn times of the C-RAN and BE PDUs. Those results are computed on 100 instances of optical rings that have run during 10,000,000 slots. The measures are taken after 3,000 slots of simulation, to let the ring initialize it's behavior. The parameters of the simulations are stated in sec 2.5.

As we can see on fig 3.1, the sojourn times for the different kind of PDU is notably similar because we do not manage anything. Furthermore, we can remark that about 10% of the C-RAN messages wait over 100 μ s, that is already critical for C-RAN **TODO: ref.**

3.2 C-RAN priority method

In order to improve the C-RAN PDUs sojourn time, we imagined the following solution, called **C-RAN priority method**. Each stations have two contention buffers. One for the BE PDUs, and one fore the C-RAN PDUs. When a station is able to send a PDU on the ring, i.e. its available container free, the container is filled with the C-RAN PDUs first.

With this method, one can imagine improve the sojour time of the C-RAN PDUs. Fig 3.2 shows us the performance of the C-RAN priority algorithm for different kinds of traffics. The parameters of the simulations still the same, stated in sec 2.5.

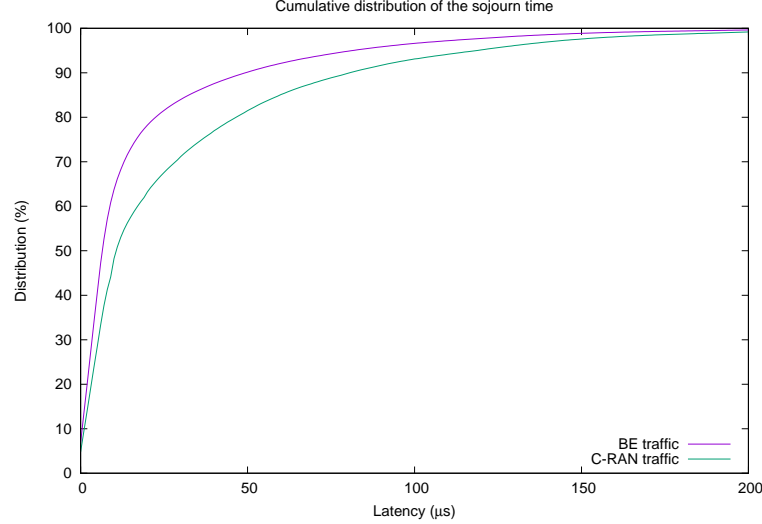


Figure 5: Full opportunistic method performances.

As expected, the sojourn time of the C-RAN PDUs is decreased compared to the full opportunistic method. Nevertheless, there is still 10% of the C-RAN PDUs that have more than $50\mu s$ of latency, and the best effort PDUs are penalized, indeed, close to 10% of the BE PDUs awaits more than $150\mu s$ before being sent in the ring, versus close to 0% with the previous method.. Thus, we propose a deterministic approach in which we set the offsets of the RRH and we reserve the containers needed for the C-RAN PDUs. **TODO: il manque une partie sur le travail de karim,**

4 Deterministic approach for 0 latency on C-RAN

For the rest of the subsection, the policy of slot insertion is the slot reservation. We are still in emission release policy. We present three algorithms that allows the C-RAN PDUs to have 0 sojourn time in the insertion buffers by establishing a static reservation of the slots in the period. Those algorithm are **Centralized** and requires a global supervisor of the network to be implemented. **TODO: phrase + ref sdn?**

Slot reservation To decrease the C-RAN messages sojourn time, one plan to reserve the container for the stations. Indeed, if a station i needs to send a PDU at time t , a container needs to be reserved since the previous lap of the container, i.e. at time $t - RS$ by i . In this situation, if the container is already taken by a PDU which has been sent by a station j , j will remove the PDU of the container during the next lap of the ring, and no other stations are able to write in this container before it comes back to the station i a date t . By finding a (P, ET) -periodic assignment and reserving the corresponding container, we allow the C-RAN PDUs to have 0 sojourn time in the stations. Though, it impacts the best effort PDUs, by avoiding them to use free reserved containers. It is equivalent to increase the load of the network, without increasing the number of PDUs carried by the network.

RRHs synchronisations This slot reservation implies that the RRHs are synchronized with the ring. The following proposed algorithms set a sequence of reserved slots for each RRH in each period. A technical application implies to simultaneously ensure the reservation of the slot to the given dates, and the synchronization of the RRH with those reserved slot. One must ensure that the C-RAN PDU arrives in the contention buffer of the station just before the reserved container.

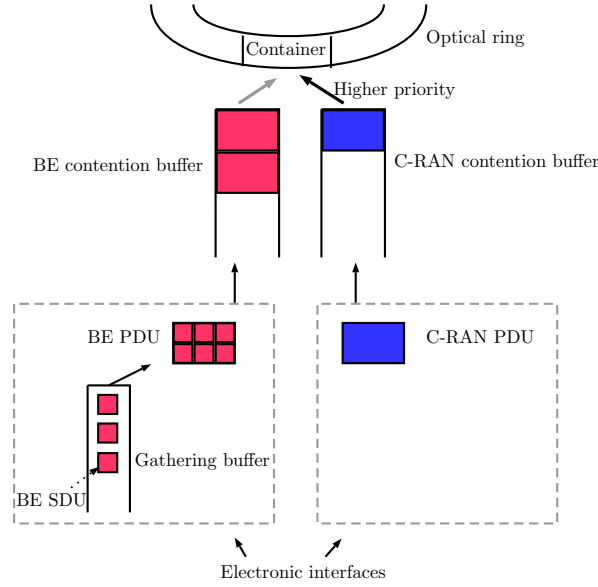


Figure 6: C-RAN priority method.

In this section, we explain the different algorithm we use to ensure 0 sojourn time on C-RAN PDUs and we study their impact on the BE PDUs sojourn times.

Unlike the previous methods with opportunistic insertion policy, we now want to choose a (P, ET) -periodic assignment for the routed network (G, \mathcal{R}) . Since all the forward routes have the same last vertex v (the datacenter), which is also the first vertex of all the backward routes, we choose to manage the collision in this vertex only. The purpose is to determine the time $t(v, r_i)$ at which the first C-RAN PDU of the route i arrives at the datacenter v for each routes i , such that there is no collisions between the routes. Let us call BBU-offset this time. Once the BBU-offsets are set, for all routes i , it is easy to compute $m_{r_i} = t(v, r_i) - \lambda(v, r_i)$.

4.1 *N-GREEN parameters adapted algorithms*

An RRH can not emit a message more often than once slot every K slots. Remember that **macro slot** is a duration of K slots, during which each RRH can emit at most once. Thus, we consider the simple case in which there is less C-RAN traffic than slots in a macro slot. Indeed, in that case, if we manage the collision in one macro slot, i.e. we assign a C-RAN traffic to each slot of the macro-slot, we find a (P, ET) -periodic assignment. Since a BBU always instantly emits an answer to it's RRH, for each RRH, a group of two following containers (called **bunch**) can be reserved. Thus, we first study the case $K \geq 2 \times n$. As a reminder, n is the number of couple BBU-RRH.

Because there is at most as much couples BBU-RRH as bunches, we want to assign one RRH to one bunch. We propose the following **naive reservation algorithm**. For each RRH $i, i \in \{0, \dots, n-1\}$, set $t(v, r_i) = 2 \times i$ with v the vertex of the BBUs. The model automatically fixes the offset of the backward route $p(r_i)$ to $m_{p(r_i)} = 2 \times i + 1$. This is the reason why we do not care about the backward routes anymore, and we only deal with the BBU-offsets of the forward routes..

We kept the same parameters than in the two previous experiments (see 2.5) and we made an experiment to show the impact of this algorithm on the BE traffic. Fig 10 shows us the distribution of the sojourn time for BE PDUs with this algorithm.

As we can see, the BE traffic is highly penalized by this algorithm, indeed, while almost 100% of the BE PDUs have a sojourn time to less than $200\mu s$ with the full opportunistic method (see fig 3.1), here, more than half of the BE PDUs are buffered more than $200\mu s$. Those performances are explained by the structure of the algorithm. The messages are grouped together, and with our parameters, $K = 2 \times n$. It

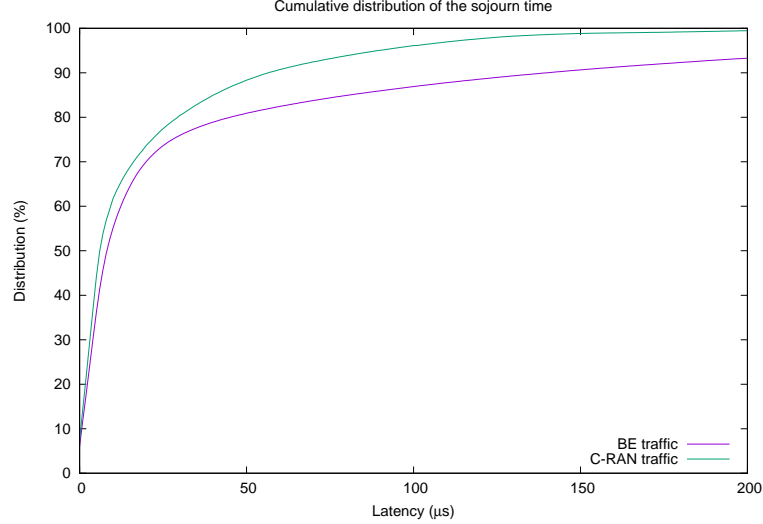


Figure 7: C-RAN priority method performances.

creates a long sequence of slots during which there is no free containers in the ring.

To improve the BE PDUs sojourn time, we propose a **split reservation algorithm** that balances the load of the CRAN traffic over the period. Instead of giving some close BBU-offsets for all routes i , with v the datacenter, we now uniformly distribute the different BBU-offsets in the period. Each couple BBU-RRH is still assigned to its bunch, but we split the BBU-offsets of $\frac{P}{n}$. Also, if $2 \times n < K$, the RRH are balanced over the bunches by the same way, we try to give a bunch to a RRH each $\frac{K}{2 \times n}$ bunches.

On fig 12 one can observe the performance of this algorithm on the BE PDUs sojourn time. Once again, to have a reference point, we take the same parameters for the simulation.

As we can see, the sojourn time of the BE PDUs is highly better with this this algorithm. Indeed almost 100% of CRAN PDUs have a sojourn time of less than $50\mu s$ which is better than the full opportunistic method. Thus, we decreased the latency of every kind of traffic.

4.2 With more antennas

In this section, we study the case $n > \frac{K}{2}$, we want to assign more than one RRH to a bunch. Let us consider a particular behavior of the ring. Instinctively, we can think that if we have enough space in the period for many traffics on the same frequency that is, $ET \leq \frac{P}{2}$, we can assign several couples BBU-RRH to the same bunch.

Let us take $ET = \frac{P}{2}$ and $n = \frac{K}{2} + 1$. We assign a bunch to each $\frac{K}{2}$ first RRH and we have only one RRH left. Then, we only look at one bunch, on which we want to add the new couple BBU-RRH.

We take a ring with two stations A and B . We simplify the model to only look at the behavior of one bunch, thus, both of the nodes alternatively emits some containers every slots, during ET slots in the period P . By construction, we allow A to emit a time 1 during ET slots. This container arrives in B at $\omega(A, B)$, that is $[t(B, r_A)]_{P, K} = [\omega(A, B); \omega(A, B) + ET]$. Then, B can immediately emit its containers at time $\omega(A, B) + ET$. This containers from B arrives at A at time $\omega(A, B) + ET + \omega(B, A) = ET + RS$, and thus, $[t(A, r_B)]_{P, K} = [ET + RS; ET + RS + ET] = [ET + RS; RS + P] = [ET + RS; RS] \mod P$. Since $[t(A, r_A)]_{P, K} = [1; ET]$, there is a collision at the interval $[1; RS]$ in the period between the two routes A and B .

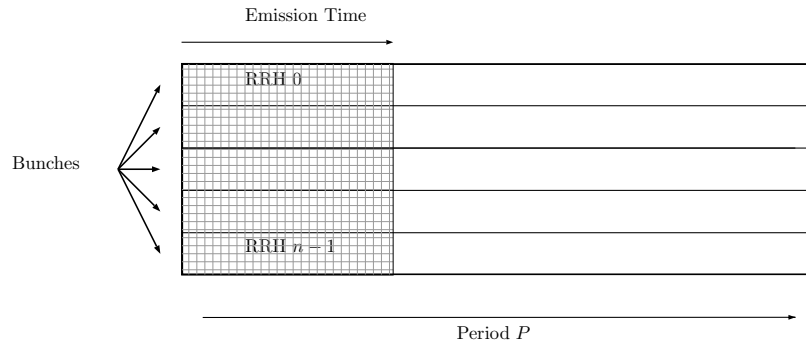


Figure 8: Grouped assignment of the bunches to the RRH.

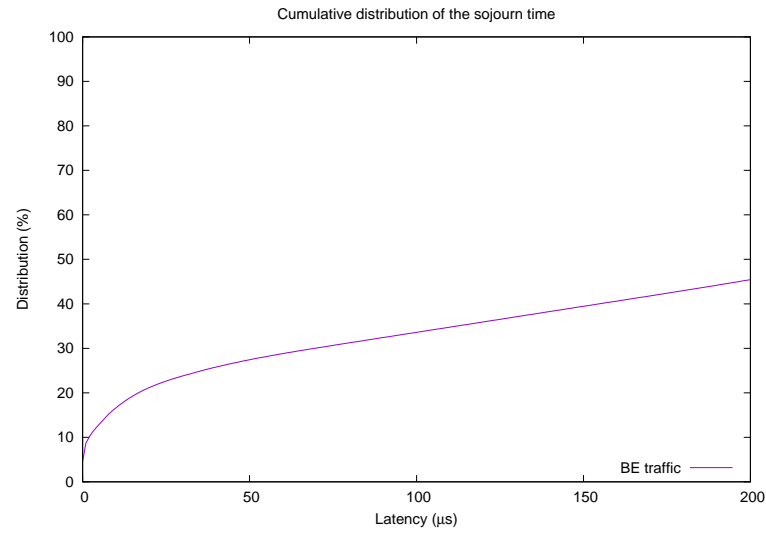


Figure 9: Impact of the naive reservation algorithm on the Best effort PDUs.

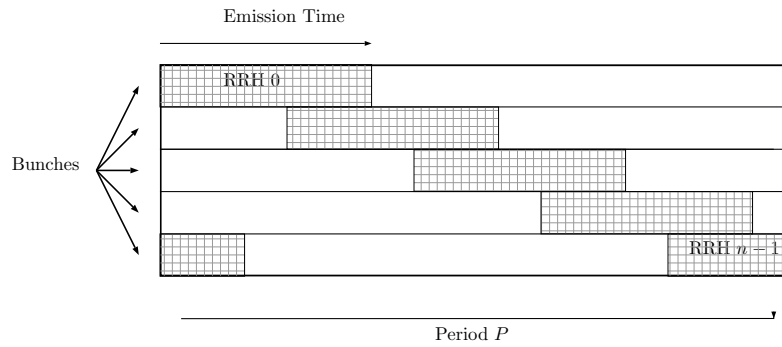


Figure 10: Splited assignment of the bunches to the RRH.

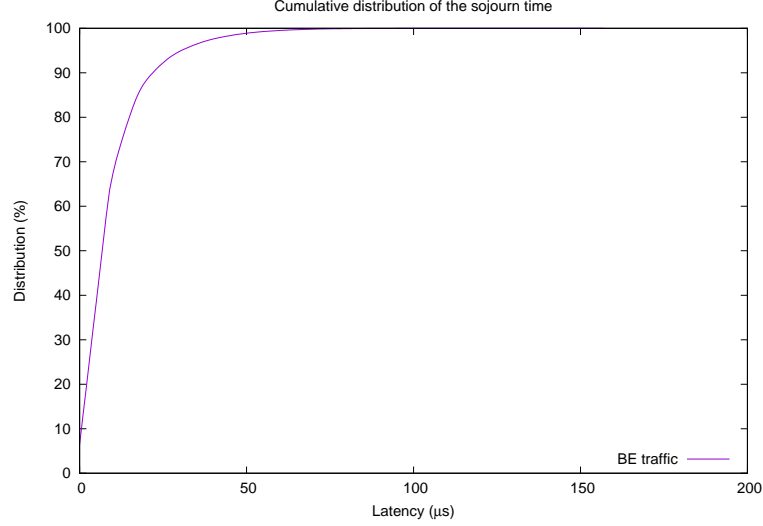


Figure 11: Impact of the splitted reservation algorithm on the Best effort.

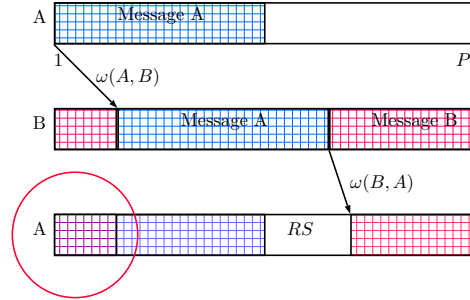


Figure 12: Collision between two routes on the same bunch when the period is too small.

As we have just noted, if several RRH share the same bunch, the period must have an additional budget in addition to the time needed to send the messages. Let us study the size of the additional budget.

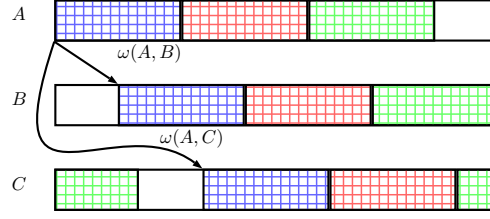
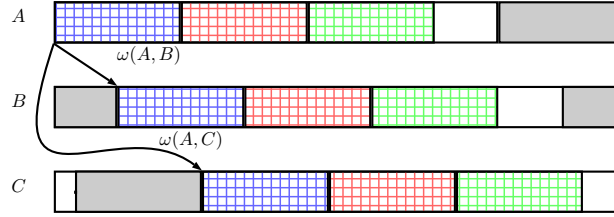
Proposition 1. *If several RRH shares a bunch, it is possible to find a (P, ET) -periodic assignment if $P \geq n \times ET + RS$, with n the number of RRHs.*

Proof. We seek to show that proposition by induction.

Base case: For $n = 2$, the proof of the previous example shows us that it is possible to assign two RRH on the same bunch with $P = 2 \times ET + RS$.

Induction step: We consider that we have a frequency shared by n RRH, with a period of size $P = n \times ET + RS$. We want to show that if we want to add one antenna to the bunch, we need a period of size at least $P = (n + 1) \times ET + RS$. If we look at the sequence period of the bunch with n RRH. We can observe that this sequence is the same in every station, shifted of the length of the routes between two stations. For instance, if we take two stations A and B , the sequence in A is the same that the sequence in B shifted of $\omega(A, B)$.

Thus, if we add ET slots in this frequency, the periodic emissions of all others nodes are not impacted, there is just a new interval in the sequence.


 Figure 13: An example of sequence with $n = 3$.

 Figure 14: The same sequence with $n = 4$.

Note that a station can send some traffic from more than one antenna, the figures 14 and [reffig:proofperiod2](#) are just an example, but this proof stays the same. \square

Considering those results, we propose an algorithm to allow several RRH to share the same bunch, as long as $P \geq n \times ET + RS$. Since we need RS slots on a each bunch with at least an RRH to assign another RRH to this latter, it is interesting to fill as far as possible the bunches. We then compute the number of bunches needed to carry the number of RRH n . Then, the algorithms tries to balance those bunches into the macro-slot, i.e. it avoids to group the used bunches together. The used bunches have RS or more free slots in the period. Thus, the algorithm subsequently balance BBU-offsets of the first RRH of each bunch, with the same idea than the splitted algorithm. We call this algorithm the **full repartition algorithm**.

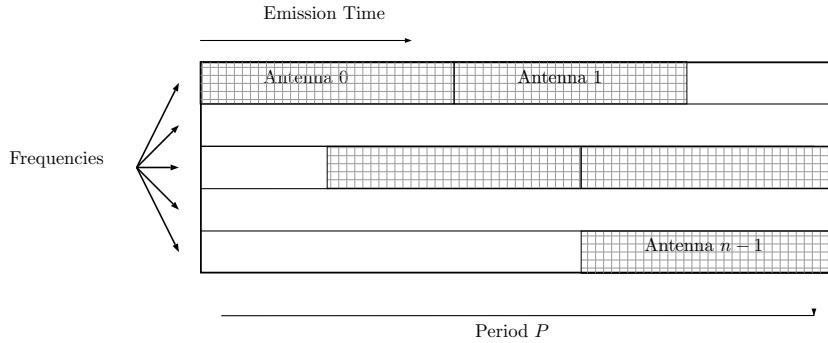


Figure 15: Organization of the bunches with the balancing algorithm.

We want to observe the impact of this algorithm on BE traffic. Since our previous parameters did not allow several RRH to share a frequency, we now set $ET = 200$, that correspond to a C-RAN flow of 2Gbps. This is not out of context since the exact split (the C-RAN split is the degree of centralization of the computation units in the cloud) of the C-RAN is not fully determined yet [?].**TODO: la ref** To keep a load similar to the load in the previous experiment, we set the number of antennas to $n = 12$. The others parameters are all the one described in the sec 2.5. The following experiment shows us the impact of the balancing algorithm on the BE traffics. We chose to compare those results to a simulation with the same parameters but with the full opportunistic method of sec 3.2.

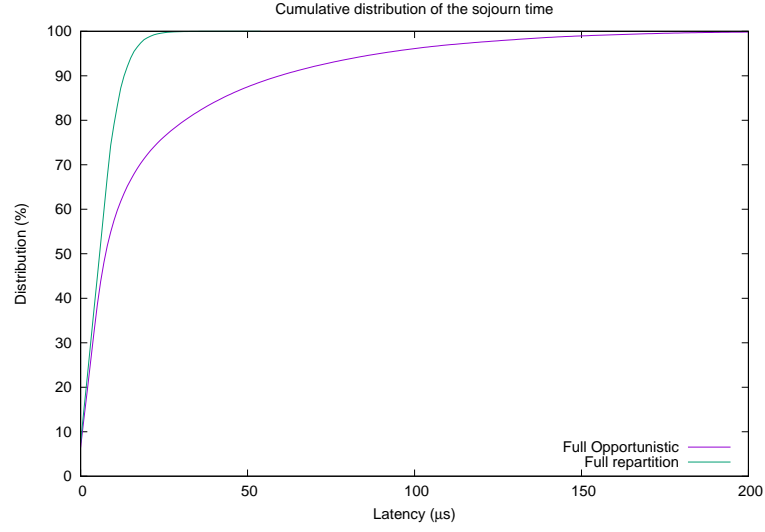


Figure 16: Balancing algorithm performance on the N-GREEN optical ring.

As fig 17 shows, the full repartition algorithm ensures a lower sojourn time to BE PDUs than the full opportunistic method, which gave the best result for the BE traffic. Indeed, since we remove the random aspect of the C-RAN PDUs generation, we balanced the load over the period and then increased the BE performances while we totally removed the C-RAN PDUs sojourn time.