Contention management for Cloud RAN over an optical ring

Dominique Barth[1] et Maël Guiraud [1,2] et Yann Strozecki [1]

[1]*David Laboratory, UVSQ*
[2]*BNokia Bell Labs France*

N-GREEN project has for goal to design a low cost optical ring technology with good performances (throughput, latency ...) without using expensive end to end connections. 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 5G deployment. We show that deterministically managing Cloud RAN traffic minimizes its latency without impacting the latency of others traffics.

Mots-clefs : Optical ring, Latency, C-RAN, SDN

TODO: ça marque et dans le titre, ça ne devrait pas
Il faut citer l'ANR NGREEN + n'importe quel truc de charoni qui explique leur architecture NGREEN

1 Introduction

Network providers have to design inexpensive networks while the amount of data and online applications significantly increases. Much of these applications have QoS criteria, like a minimal throughput or a maximal latency. The N-GREEN project aims to design a high performing optical ring while ensuring a minimal cost for providers. The current solutions with good QoS [huawei, demander a dominique C], establishes end to end connection (E2E) between the nodes, which is extremely expensive. The N-GREEN optical ring is designed to ensure good performances and the hardware it requires scales linearly with the number of nodes while E2E scales quadratically making it impractical for more than a few nodes.

In this article, we study a Cloud RAN (C-RAN) application in the N-GREEN optical ring. C-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 Heads** (RRH) in one datacenter. The latency of the messages between the BBU and the RRH is critical. TODO: mettre les citations sur la latence dans la 5G, archi ngreen et concurents E2E à reprendre de nos précédents articles The aim of the article is to propose simple methods to deterministically manage the periodic C-RAN traffic. In a previous work [1], the authors have studied this problem for a star shaped network. Here the load due to the CRAN traffic is small which makes the problem easier. However, we add several new difficulties: the messages from RRHs are fragmented, there are other traffics which must not be to impacted and the broadcast and select policy of the ring makes reservation of a specific time for emission somewhat wasteful.

In Sec. 2, we model the optical ring and the traffic flow. In Sec. 3, we study the latency when using stochastic multiplexing to manage packets insertions on the ring, with or without priority for C-RAN packets. In Sec. 4, we propose deterministic ways to manage C-RAN packets without buffers and thus with a minimal latency compared to stochastic multiplexing. The best method to do so also improves the latency of best effort packets. 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, Yann : je pense qu'il faut parler de SDN, reprendre peut être des choses de notre article précédent sur Cloud RAN

2 Model of C-RAN traffic over an optical ring

N-GREEN Optical ring The unidirectional optical ring is represented by an oriented cycle. The vertices of the cycle represents the nodes of the ring, where traffic arrives. The edges (u,v) of the cycle have an integer weight $\omega(u,v)$ which represents the time to transmit a unit of information from u to v. We denote by RS (ringsize) the length of the cycle, that is the sum of $\omega(u,v)$ for all arcs (u,v). Each arc (u,v) can be seen as a sequence of $\omega(u,v)$ **containers**, of capacity C, expressed in Bytes. The containers are filled by the nodes. The node u can put some data in the first container of the arc (u,v) *only if it is free*. The dynamic of the network is simple: at each unit of time data contained in a container go to the next. The ring follows a **broadcast and select with emission release policy**. When a container is filled by some node u, it is freed when it comes back at u after a whole round trip.

C-RAN traffic There are n RRHs attached to the ring, the RRH i being attached to the vertex u_i (several antennas can be attached to the same vertex). The antenna are linked to the nodes of the ring through an electronic interface of bit rate R Bps. The ring has a larger rate of $F \times R$ Bps. The integer F is called the **acceleration factor** between the electronic and the optical domains. A node aggregates the data received on the electronic interface during F unit of times to create a packet of size C which completely fills one container of the ring. An RRH emits a C-RAN traffic during a time ET (emission time) in each period P, which consists in a packet of size C each F unit of times.

The data of the RRH i arrives at node u_i at a time m_i called the **offset**. The offset can be determined by the designer of the system and can be different for each RRH but must remain the same over all periods. We assume that all BBUs are contained in the same data-center attached to the node v. The data from u_i is routed to its BBU at node v through the ring and arrives at time $m_i + \omega(u,v)$ if it has been inserted in the ring upon arrival. Then after some computation time (which is supposed to be zero to simplify the model), an answer is sent back from the BBU to the RRH. The same quantity of data is emitted by each BBU or RRH during any period. The **latency** of a message is the time it waits in a node before being put in a free container on the ring. The aim of our study is to minimize the latency of the C-RAN traffic, both from the RRHs and the BBUs. In fact in Sec.4 we propose a deterministic mechanism with zero latency. The only downside is that other data used the optical ring and that their latency may be increased. We shortly describe the nature of this additional traffic in the next subsection.

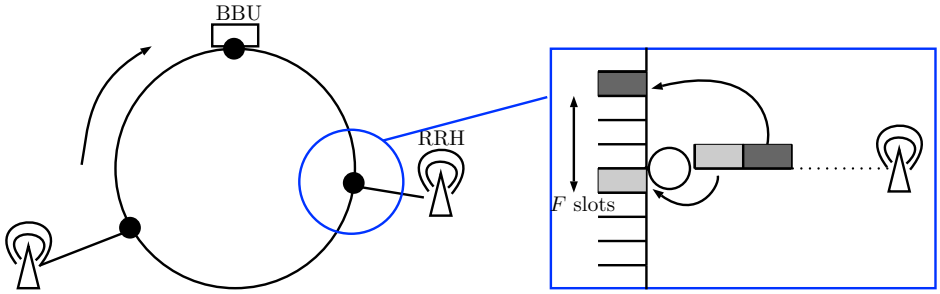

Figure 1: Opto-electronic interface.

TODO: Améliorer le dessin: un seul schéma avec les notations et horizontal pour pas perdre de place. Éventuellement faire une représentation du ring et en zoom faire l'interface

TODO: Décrire ici les degrés de liberté -¿ quand ça part + la latence

Best effort traffic The optical ring supports other traffics, corresponding to the internet flow. We call this traffic **Best Effort** (BE), since it is less critical and one can increase the latency of BE messages to improve the latency of C-RAN messages. A contention buffer is filled by a batch arrival process of BE data. Then according to some parameters (fill rate and maximum wait time) a container is sent on the ring from the contention buffer. The arrival of BE messages is modeled by a temporal law that gives the distribution

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

Figure 2: Parameters of the following experiments based on N-GREEN

of times between two arrivals of a container of BE messages in a node. This distribution is obtained by choosing optimal parameters for the contention buffer [].

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 node 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 node. 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.

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 node. 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 latency 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 **??**.

As we can see on fig 3.1, the latency 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 100μs, that is already critical for C-RAN TODO: ref.

3.2 C-RAN priority method

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

With this method, one can imagine improve the latency 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 **??**.

As expected, the latency 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 50μs of latency, and the best effort PDUs are penalized, indeed, close to 10% of the BE PDUs awaits more than 150μs before being sent in the ring, versus close to 0% with the previous method.. Thus, we propose a deterministic approach in which we

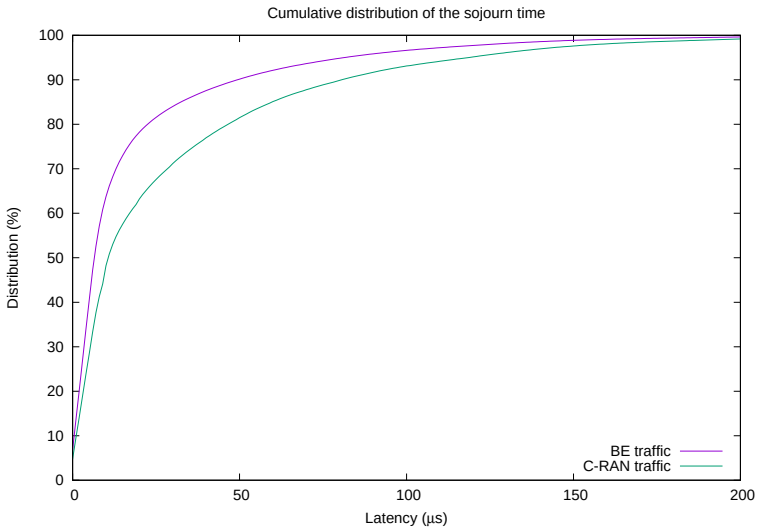

Figure 3: Full opportunistic method performances.

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

In this section We present three algorithms that allows the C-RAN PDUs to have 0 latency 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 latency, one plan to reserve the container for the nodes. Indeed, if a node 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 node j, j will remove the PDU of the container during the next lap of the ring, and no other nodes are able to write in this container before it comes back to the node i a date t. By choosing the offsets of the route and reserving the corresponding containers, we allow the C-RAN PDUs to have 0 latency in the nodes. 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 node just before the reserved container.

Unlike the previous methods with opportunistic insertion policy, we now want to choose a the offsets for the routes. 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 then to determine the time BBU offset of the routes. Once the BBU-offsets are set, for all routes i, it is easy to compute $m_{r_i} = t(v, r_i) - \omega(u, v)$.

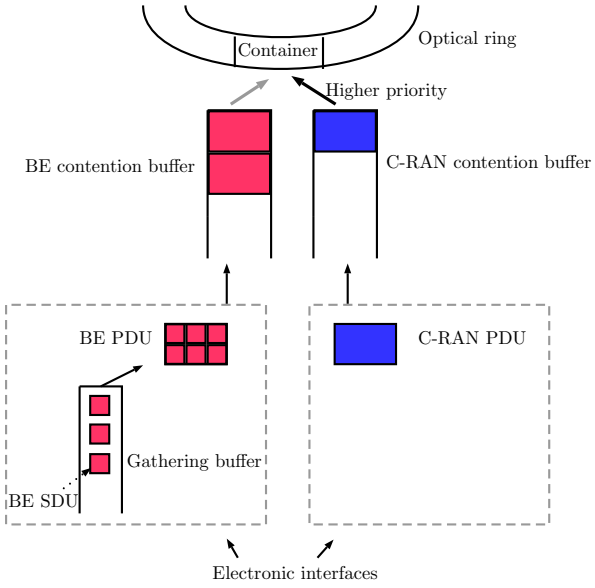

Figure 4: C-RAN priority method.

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, we only manage the collision in one macro slot, i.e. we assign a C-RAN traffic to each slot of the macro-slot. 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 $i, i \in \{0, ..., n-1\}$, 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 $\rho(r_i)$ to $m_{\rho(r_i)} = 2 \times i + 1$. 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 **??**) and we made an experiment to show the impact of this algorithm on the BE traffic. Fig 7 shows us the distribution of latency 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 latency to less than 200μs with the full opportunistic method (see fig 3.1), here, more than half of the BE PDUs are be buffered more than 200μs. Those performances are explained by the structure of the algorithm. The messages are grouped together, and with our parameters, $K = 2 \times n$. It creates a long sequence of slots during which there is no free containers in the ring.

To improve the BE PDUs latnecy, we propose a **splited 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 $\frac{K}{2 \times n}$ buches.

On fig 9 one can observe the performance of this algorithm on the BE PDUs latency. Once again, to have a reference point, we take the same parameters for the simulation.

As we can see, thelatency of the BE PDUs is highly better with this this algorithm. Indeed almost 100% of CRAN PDUs have a latency of less than 50μs which is better than the full opportunistic method. Thus,

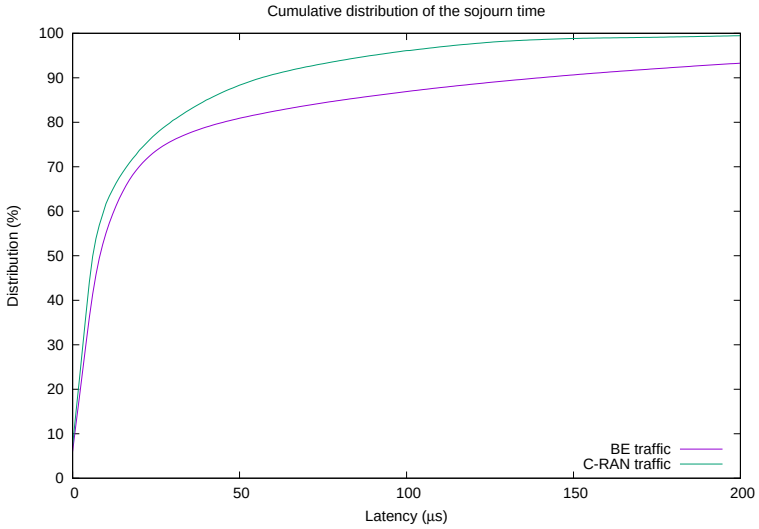

Figure 5: C-RAN priority method performances.

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 nodes A and B. We simplify the model to only look at the behaviour of one bunch, thus, both of the nodes alternatively emits some containers every slots, during ET slots in the period P. We assume that A emits 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 traffics from A and B.

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 set the offsets of the routes such that there is no collisions between the messages 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 bunch 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 RRH 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

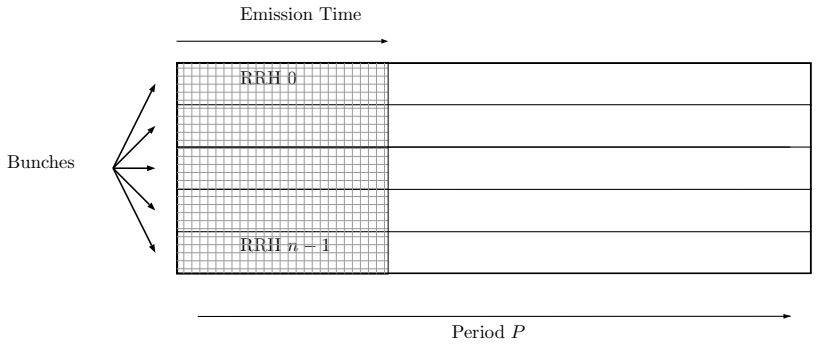

Figure 6: Grouped assignment of the bunches to the RRH.

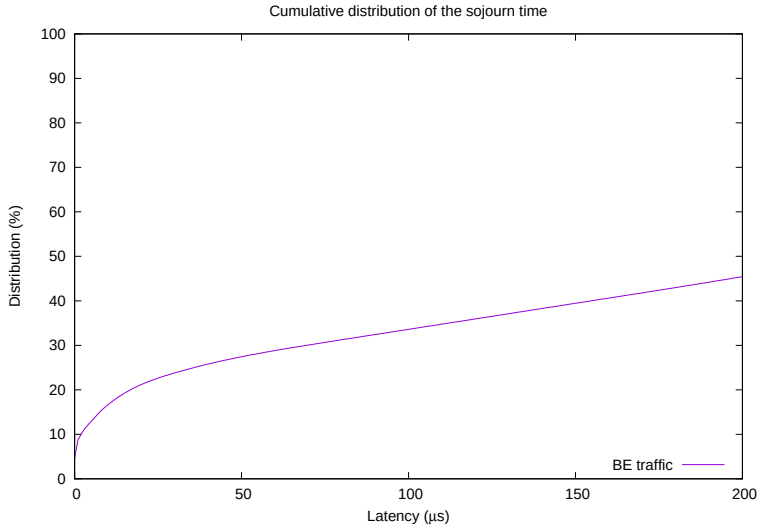

Figure 7: Impact of the naive reservation algorithm on the Best effort PDUs.

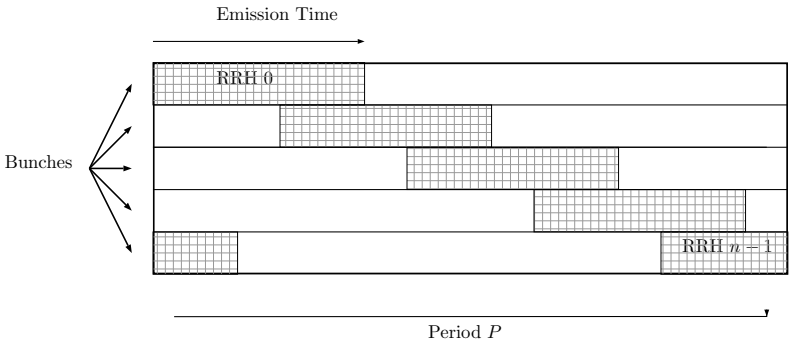

Figure 8: Splited assignment of the bunches to the RRH.

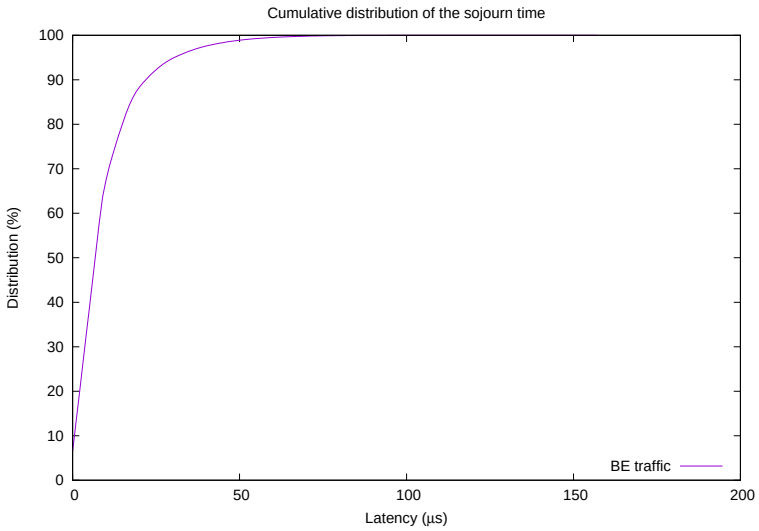

Figure 9: Impact of the splited reservation algorithm on the Best effort.

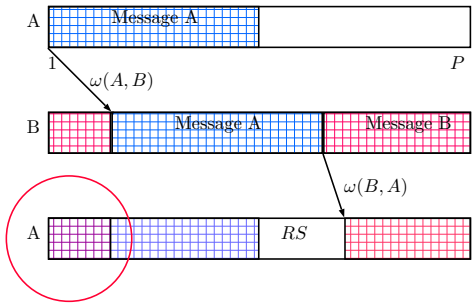

Figure 10: Collision between two routes on the same bunch when the period is too small.

sequence is the same in every nodes, shifted of the length of the routes between two nodes. For instance, if we take two nodes 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 bunch, the periodic emissions of all others nodes are not impacted, there is just a new interval in the sequence.

Note that a station can send some traffic from more than one RRH, the figures 11 and ẽffig:proofperiod2 are just an example, but this proof stays the same. □

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 splited algorithm. We call this algorithm the **full repartition 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 bunch, we now set $ET = 200$, that correspond to a C-RAN flow of 2Gbps. This

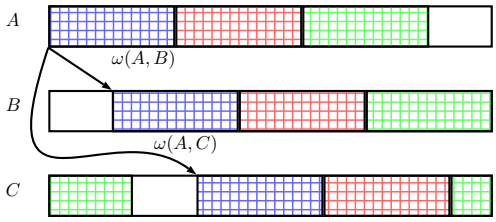

Figure 11: An example of sequence with n = 3.

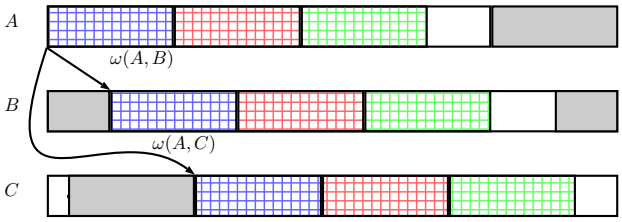

Figure 12: The same sequence with n = 4.

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 RRH to $n = 12$. The others parameters are all the one described in the sec **??**. 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 3.1.

As fig 14 shows, the full repartition algorithm ensures a lower latencye 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 an then increased the BE performances while we totally removed the C-RAN PDUs latency.

References

[1] B. Dominique, G. Maël, and S. Yann, "Deterministic scheduling of periodic messages for cloud ran," *arXiv preprint arXiv:1801.07029*, 2018.

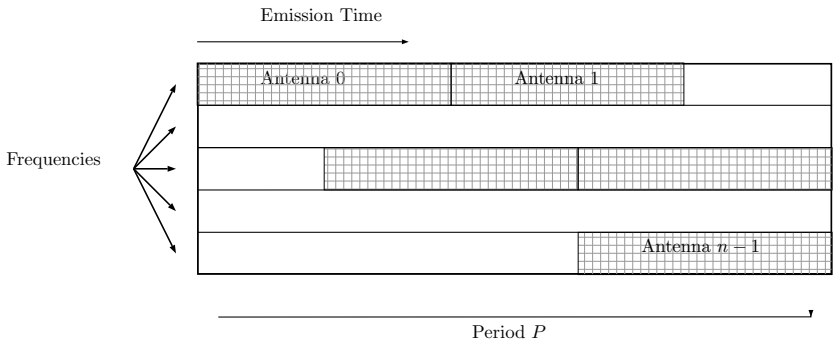

Figure 13: Organization of the bunches with the balancing algorithm.

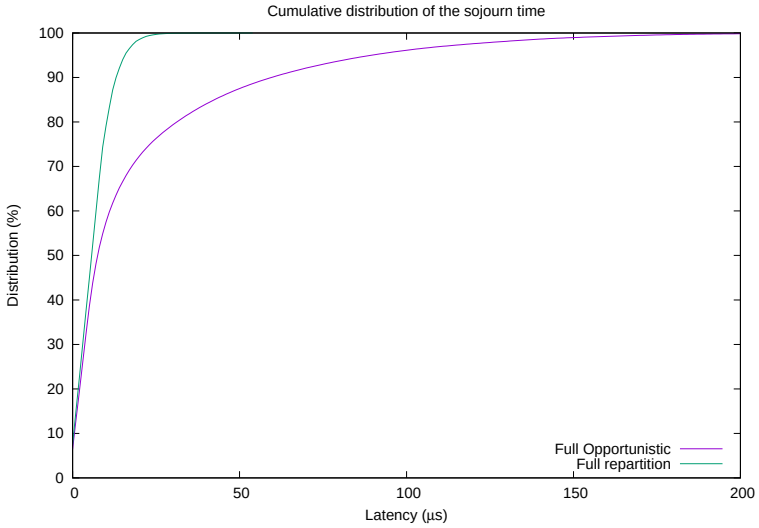

Figure 14: Balancing algorithm performance on the N-GREEN optical ring.