Deterministic scheduling of periodic datagrams for low latency in 5G and beyond

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et technologies de l'information et de la communication (STIC)
Unités de recherche: (1) Université Paris-Saclay, UVSQ, Données et Algorithmes pour une ville intelligente et durable, 78035, Versailles, France.
(2) Nokia Bell Labs France, 91620 Nozay, France.
Référent: : Université de Versailles-Saint-Quentin-en-Yvelines.

Thèse présentée et soutenue à ....., le .... 2021, par

Maël Guiraud

Composition du jury:

(**Johanne Cohen** Directeur de Recherche , LRI	Président/e)
Prénom Nom Titre, Affiliation	Rapporteur
Prénom Nom Titre, Affiliation	Rapporteur
(**Safia Kedad-Sidhoum** Professeur, CNAM	Examinatrice)
Dominique Barth Professeur, UVSQ	Directeur de thèse
Yann Strozecki Maître de conférence, UVSQ	Coencadrant
Olivier Marcé Ingénieur de recherche, Nokia Bell Labs France	Coencadrant
Brice Leclerc Ingénieur de recherche, Nokia Bell Labs France	Examinateur

Thèse de doctorat

NNT: XXXXXXXX

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Ordonnancement periodiques de messages pour minimiser la latence dans les réseaux dans un contexte 5G et au delà

Mots clés: Cloud Radio Access Network, Ordonnancement periodique, Heuristiques de recherche locale, Analyse de complexité, Réduction de la latence, Theorie des graphes

Résumé: Cette thèse est le fruit d'une collaboration entre les laboratoires DAVID et Nokia Bell Labs France. L'idée originale est de trouver des solutions algorithmiques pour gérer des flux periodiques de manière déterministe dans les réseaux afin de contrôler et de minimiser le temps de transmission, appelé latence. L'un des objectifs de la 5G (le C-RAN, pour Cloud Radio Access Network) est de centraliser les unités de calculs des antennes radio des réseaux de télécommunications (appelé Radio Access Network) dans un même centre de calcul (le Cloud). Le réseau entre le centre de calcul et les antennes doit être capable de satisfaire les contraintes de latence imposées par les protocoles.

Nous définissions le problème de trouver un ordonnancement periodique pour les messages de façon à ce qu'ils ne se disputent jamais la même ressource, et prouvons que les différentes variantes du problème étudiés sont NP-complets. Nous étudions dans un premier temps le problème pour une topologie particulière dans laquelle tous les flux partagent un même lien. Nous proposons dans un premier temps des algorithmes polynomiaux, de plus en plus evolués, ainsi que des algorithmes FPT peremettant de trouver une solution quand le nombre de route est raisonnable, ce qui est le cas des réseaux C-RAN.

Les algorithmes développés dans cette première partie n'étant pas applicables directement aux topologies plus générales, nous proposons ensuite une forme cannonique au problème qui nous permet de définir une notion de voisinage efficace pour des heuristiques de recherches locales (descente, recherche tabou, recuit simulé). Nous utilisons cette forme cannonique pour définir un algorithme Branch and Bound efficace quand le nombre de route est modéré. Nous proposons aussi une évaluation de performance des solutions proposés par rapport aux solutions courantes de gestion des flux, et montrons que notre modèle est réalisable en pratique grace aux nouveaux equipements en cours de développement.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Deterministic scheduling of periodic datagrams for low latency in 5G and beyond

Keywords: Cloud Radio Access Network, Periodic scheduling, Local search heuristics, complexity analysis, Latency reduction, Graph theory

Abstract: This thesis is the result of a collaboration between DAVID Laboratory and Nokia Bell Labs France. The original idea is to find algorithmic solutions to deterministically manage periodic flows in networks in order to control and minimize the transmission time, called latency. One of the objectives of 5G (C-RAN, for Cloud Radio Access Network) is to centralize the calculation units of the radio antennas of telecommunications networks (called Radio Access Network) in the same computer center (the Cloud). The network between the computing center and the antennas must be able to satisfy the latency constraints imposed by the protocols.

We define the problem of finding a periodic scheduling for messages so that they never compete for the same resource, and prove that the different variants of the problem studied are NP-complete. We first study the problem for a particular topology in which all the streams share the same link. We first propose polynomial algorithms of increased sophistication, and FPT algorithms that allow us to find a solution when the number of routes is reasonable, which is the case for C-RAN networks.

Since the algorithms developed in this first part are not directly adaptable to more general topologies, we then propose a canonical form to the problem which allows us to define an efficient neighborhood notion for local search heuristics (hill climbing, taboo search, simulated annealing). We use this canonical form to define an efficient Branch and Bound algorithm when the number of routes is moderate. We also propose a performance evaluation of the proposed solutions compared to current flow management solutions, and show that our model is feasible in practice thanks to new equipment under development.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Contents

List of Figures

Chapter 1

Mixing Periodic Datagrams and Stochastic Datagrams

In previous chapters, we presented algorithms that solve the problems of scheduling deterministic traffic in the network. In practice, networks are shared between deterministic and stochastic traffics. This is possible in practice using Time Sensitive Networking technology, that allows to manage traffics independently. The objective of this chapter is to study the impact on stochastic traffic of the algorithms we have designed to minimize the latency of deterministic traffic. We propose a method using the algorithms we have designed, to improve the latency of *all* traffics of the network.

1.1 Periodic Assignment and Random Traffic on Star Routed Networks

This section is taken from [1]. The algorithms proposed in this thesis are designed to manage deterministic periodic flows in dedicated networks. In this section, the objective is to determine the effect of adding in the network non-deterministic flows (internet traffic, best-effort) managed by statistical multiplexing.

The algorithms solving PALL are not designed to take into account best-effort traffic. In particular, they often build very compact assignments, with all messages following one another in a contention point, which is bad for the latency of best-efforts packets trying to go through the same contention point. Thus, we propose an adaptation of any algorithm solving PALL, to find assignments where the unused tics are as evenly spaced as possible to minimize the maximal latency of any random packet trying to go through the contention point.

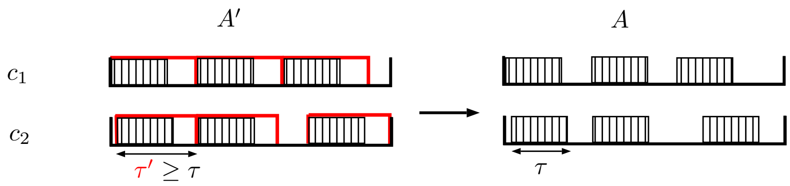

Figure 1.1 – A (P,τ')-assignment interpreted as a (P,τ)-assignment

1.1.1 Spaced Assignments

Most algorithms for PALL, when determining the waiting times, send datagrams as early as possible and thus create long sequences of datagrams in c_2, without free tics between them. We propose to modify any algorithm solving PALL on an instance with datagram size τ as follows: compute a (P,τ') assignment using the algorithm, for the largest possible $\tau' \geq \tau$.

Lemma 1. *Let $I' = (N,P,\tau',d)$ be an instance of* PALL*, for which there is an assignment, and let $\tau \leq \tau'$, then there is also an assignment for $I = (N,P,\tau,d)$.*

Proof. Let A be the assignment of I', the absence of collision is the absence of intersection between intervals $[r_i,c_1]_{P,\tau'}$ (and $[r_i,c_2]_{P,\tau'}$). If we consider A as an assignment of I, then the intervals are $[r_i,c_1]_{P,\tau}$ and are strictly included in $[r_i,c_1]_{P,\tau'}$, hence they do not have an intersection either. □

Lemma 1 gives a way to obtain a solution to the original instance from the instance with a larger message size as illustrated in Figure 1.1, with the additional property that all datagrams are separated by at least $\tau' - \tau$ free tics in each contention point. We are interested in finding the maximal τ' for which there is an assignment. Since the property of having an assignment is monotonous with regards to τ, we can do so by a dichotomous search.

We call `SPMLS`, for **S**paced `PMLS`, the adaptation of `PMLS` which finds an assignment for the largest possible τ by dichotomous search on τ. We now investigate how large are the τs for which `SPMLS` finds an assignment. In Figure 1.2, we represent the probability to find a (P,τ')-assignment function of τ'. The star routed networks are generated as in Chapter **??**, with 8 routes and length of the arcs drawn in $[P]$. The network has a load of 60% of C-RAN traffic, hence the period is set to 33,333 for $\tau = 2500$. The network is less loaded with C-RAN traffic than in the previous sections because it will also support non deterministic traffic, incurring an additional load.

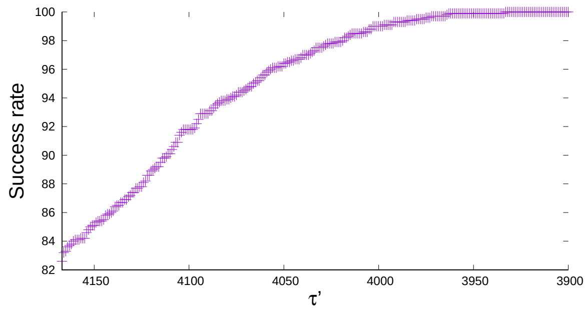

Figure 1.2 – Probability of finding a (P,τ')-assignment over 10,000 instances

For more than 80% of the instances, there is an assignment for the maximal size of a message $\tau' = \frac{P}{n} = 4166$. This means that SPMLS perfectly balances the free tics in the period. In the worst case, a solution with $\tau' = 3925$ is found, which still yields $3925 - 2500 = 1425$ unused tics between datagrams. Hence, we expect SPMLS to work well in conjunction with random traffic. The excellent performance of PMLS when the load is high explains this result and justifies the work we have done to solve PALL efficiently under high load rather than just requiring a mild load in applications.

1.1.2 Performance Evaluation

We evaluate in this section different ways to manage both statistical and deterministic traffics together in the same network.

1.1.2.1 Best-effort datagrams generation

Let us denote best-effort by BE. The BE traffic is generated as follows. The size of a BE datagram is small in practice, and set to 50 tics in our experiments. We generate 20% of average load of BE traffic in our experiment, to obtain a total load of 80%. The BE datagrams do not make a round trip in the network as the C-RAN datagrams, they go through a single contention point. We simulate that, by generating 20% of average load of BE datagrams for each of the two contention points c_1 and c_2. The **latency** of a BE datagram is defined as the time it must wait before going through its contention point.

On each contention point, the generation is split in two exponential distributions which give the time before the next arrival of datagrams. The first one models background traffic, it has an average load of 15% by generating one BE datagram every 333 tics on average. The second models a burst of BE datagrams with an average load of 5%, that is, a generation of 10 datagrams, but every 10,000 tics on average.

1.1.2.2 Statistical multiplexing policy

We test several policies to deal with all traffics using statistical multiplexing. The BE traffic is managed using `FIFO`, and we propose two policies to deal with C-RAN. First, all datagrams, BE or C-RAN, are stored in the same buffer and dealt with the `FIFO` policy regardless of their type. We call this policy `FIFO`.

In order to minimize the latency of C-RAN traffic, we can store the two types of datagrams in two different buffers, managed each with `FIFO`, but we prioritize the C-RAN datagrams which are always sent first. It can be technically implemented by using TSN 802.1Qbu [2], that allows to define priority class in the traffic to schedule first the traffic with the highest priority, here the C-RAN traffic. We call this policy `FramePreemption`.

We also consider the case of C-RAN traffic scheduled by `PMLS` or `SPMLS`. Then, we need to forbid the transit of a BE datagram which collides with a C-RAN datagram. Thus, in each contention point, we reserve 50 tics (the size of a BE datagram) before the arrival of a C-RAN message. Observe that it wastes some ressources and thus slightly decreases the maximal throughput and may worsen the latency of BE datagrams.

Figure 1.3 shows the cumulative distribution of the logical latency of BE datagrams, that is the probability that a BE datagram has a latency less than some value. The distribution is computed over 1000 random instances, and for each the traffic is simulated for ten periods.

If we compare `FIFO` and `FramePreemption`, we see that the latency of BE datagrams is better (1977 tics on average) with `FIFO`. It is expected, since in `FramePreemption` the C-RAN datagrams are prioritized and thus the latency of the BE datagrams is strictly worse, 3256 tics on average. However, this is a trade-off with the margin of the C-RAN datagrams, which is strictly better for `FramePreemption`: 1919 tics on average versus 5265 tics for `FIFO`.

Using a deterministic approach for C-RAN with `PMLS`, the trade-off is even stronger: the CRAN margin is down to 0, but the BE traffic is more impacted, at a latency of 4909 tics on average. This can be explained by both reservation of tics to deal with the

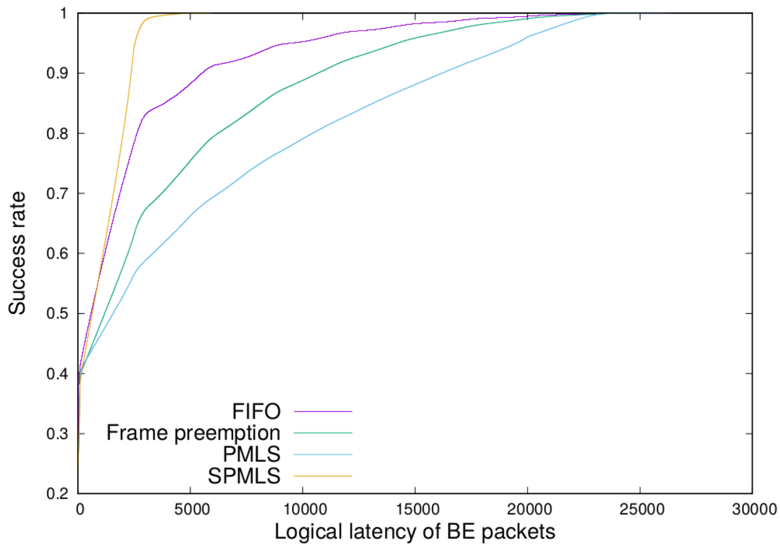

Figure 1.3 – Cumulative distribution of the latency of BE datagrams for several network management schemes

periodic sending scheme and the long sequences of CRAN datagrams without free time in contention points.

In `SPMLS`, the C-RAN traffic is smoothed over the period, in order to regularly leave some free tics for BE traffic. By construction, we still have CRAN margin of 0 but it improves the latency of BE datagrams to 949 tics on average, which is even better than with `FIFO`. This result shows that managing deterministic traffic deterministically is also good for the other sources of traffic on the network. We have already observed such a phenomenon in [3], a similar problem on an optical ring, that we describe in the next section.

1.2 Both Traffics On Optical Ring : An Industrial product

In this section, taken from [3], we study a C-RAN application based on an optical ring. We work on an industrial product which was developed in the ANR project N-GREEN described in [4, 5]. In contrast with the previous chapters, finding emission timings so that different periodic sources do not use the same resource is easy in the context of the N-GREEN optical ring with a single data-center. However, we deal with two additional difficulties arising from practice: the messages from RRHs are scattered because of the electronic to optic interface and there are other traffics whose latency must be preserved. It turns out that the deterministic management of CRAN traffic we propose reduces the latency of CRAN traffic to the physical delay of the routes, while reducing the latency of the other traffics by smoothing the load of the ring over the period. To achieve such a good latency, our solution needs to reserve resources in advance, which slightly decreases the maximal load the N-GREEN optical ring can handle. Such an approach of reservation of the network for an application (CRAN in our context) relates to network slicing [6] or virtual-circuit-switched connections in optical networks [7, 8].

In Section 1.2.1, we model the optical ring and the traffic flow. In Section 1.2.2, we experimentally evaluate the latency when using stochastic multiplexing to manage packets insertion on the ring, with or without priority for C-RAN packets. In Section 1.2.3, we propose a deterministic way to manage C-RAN packets without buffers, which guarantees to have zero latency from buffering. We propose several refinements of this deterministic sending scheme to spread the load over time, which improves the latency of best-effort packet, or in Section 1.2.3.3, to allow the ring to support a maximal number of antennas at the cost of a very small latency for the C-RAN traffic.

1.2.1 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 represent the nodes of the ring, where the traffic arrives. The arcs (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. By extension, if u and v are not adjacent, we denote by $\omega(u,v)$ the size of the directed path from u to v. The **ring size** is the length of the cycle, that is $\omega(u,u)$ and we denote it by RS. A **container**, of capacity C expressed in bytes, is a basic unit of data in the optical ring.

The time is discretized: a unit of time corresponds to the time needed to fill a container with data. As shown in Figure 1.4, the node u can fill a container with a data packet of size less than C bytes at time t if the container at position u at time t is *free*. If there are several packets in a node or if a node cannot fill a container, because it is not free, the remaining packets are stored in the **insertion buffer** of the node. A container goes from u to v in $\omega(u,v)$ units of time. The ring follows a **broadcast and select scheme with emission release policy**: When a container is filled by some node u, it is freed when it comes back at u after going through the whole cycle.

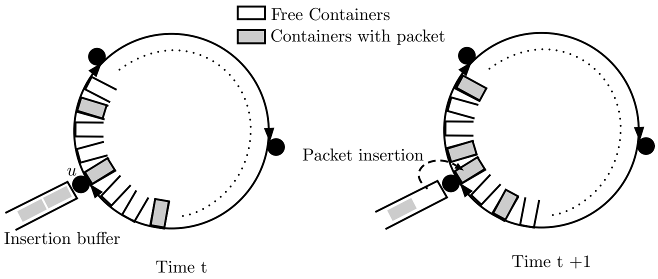

Figure 1.4 – Dynamic behavior of the ring.

C-RAN traffic The RRHs are the source of the **deterministic and periodic** C-RAN traffic. There are k RRHs attached to the ring and several RRHs can be attached to the same vertex. An RRH is linked to a node of the ring through an electronic interface of bit rate R Bps. The ring has a larger bit 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 units of time to create a packet of size C and then puts it in the insertion buffer. In each period P, an RRH emits data during a time called **emission time** or ET. Hence the RRH emits ET/F packets, i.e. requires a container of size C each F units of time during the emission time, as shown in Figure 1.5.

At each period, the data of the RRH i begins to arrive in the insertion buffer at a time o_i called **offset**. The offsets 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 is routed to its BBU at node v through the ring and arrives at time $o_i + \omega(u,v)$ if it has been inserted in the ring upon arrival. Then, after some computation time, which w.l.o.g. is supposed to be zero, 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 data packet is defined as the time it waits in an insertion buffer. Indeed, because of the ring topology, the routes between RRHs and BBUs are fixed, thus we cannot reduce the physical transmission delay of a data which depends only on the size of the arcs used. Moreover, there is only one buffering point in the N-GREEN optical ring, the insertion buffer of the node at which the data arrives. Hence, in this context, to minimize the end-to-end delay, we need to minimize the (logical) latency. More precisely, we want to reduce the latency of the C-RAN traffic to **zero**, both for the RRHs (uplink) and the BBUs (downlink). In Section 1.2.3 we propose a deterministic mechanism with zero latency for C-RAN which also improves the latency of other data going through the optical ring. We shortly describe the nature of this additional traffic in the next paragraph.

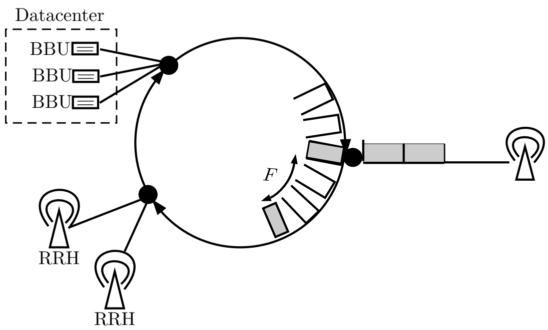

Figure 1.5 – Insertion of C-RAN traffic in the N-GREEN optical ring.

Best-Effort traffic The optical ring supports other traffics, corresponding to the internet flow. We call this traffic **Best-Effort** (BE). We want it to have the best possible distribution of latency, but since BE traffic is less critical than C-RAN traffic, we impose no hard constraint on its latency. At each node of the ring, a **contention buffer** is filled by a batch arrival process of BE data. This batch arrival process consists in generating, at each unit of time, a quantity of data drawn from a bimodal distribution to model the fact that internet traffic is bursty. Then, according to the fill rate of the contention buffer and the maximum waiting time of the data, a packet of size at most C may be created by aggregating data in the contention buffer. This packet is then put in the insertion buffer of the node. Hence, the arrival of BE messages can be modeled by a temporal law that gives the distribution of times between two arrivals of a BE packet in the insertion buffer. The computation of this distribution for the parameters of the contention buffer used in the N-GREEN optical ring is described in [9]. We use this distribution in our experiments to model arrivals of BE packets in the insertion buffer.

1.2.2 Evaluation of the latency on the N-GREEN optical ring

We first study the latency of the C-RAN and BE traffics when the ring follows an opportunistic insertion policy: When a free container goes through a node, it is filled with a packet of its insertion buffer, if there is one. Two different methods to manage the insertion buffer are experimentally compared. First, the **FIFO** rule, which consists in managing the C-RAN and BE packets in the same insertion buffer. Then, when a free container is available, the node fills it with the oldest packet of the insertion buffer, without distinction between C-RAN and BE. This method is compared to a method called **C-RAN priority** that uses two insertion buffers: one for the BE packets, and another for the C-RAN packets. The C-RAN insertion buffer has the priority and is used to fill containers on the ring while it is non empty before considering the BE insertion buffer.

We compare experimentally these two methods in the simplest topology: The lengths of the arcs between nodes are equal and there is one RRH by node. The experimental parameters are given in Table 1.1 and chosen following [4]. In each experiment, the offsets of the RRHs are drawn uniformly at random in the period. The results are computed over 1,000 experiments in which the optical ring is simulated during 1,000,000 units of time. Fig. 1.6 gives the cumulative distribution of both C-RAN and BE traffics latencies for the FIFO and the C-RAN priority methods. The source code in C of the experiments can be found on the webpage [10].

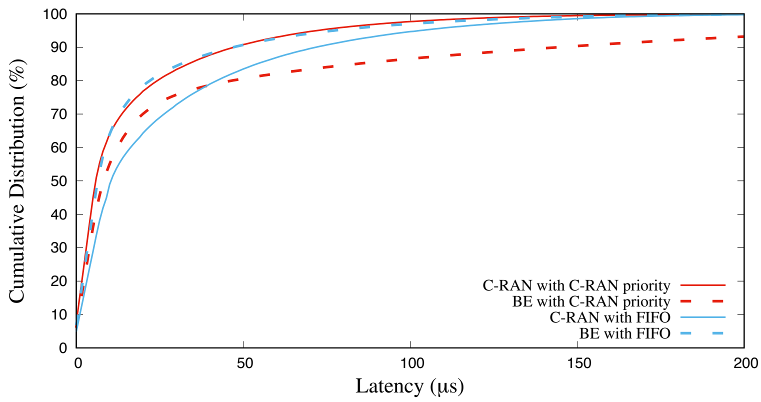

Figure 1.6 – Distribution of latencies for FIFO and C-RAN first

Bit rate of an electronic interface R	10 Gbps
Optical ring bit rate $F \times R$	100 Gbps
Acceleration factor F	10
Container size C	100 kb
Unit of time (UoT) $C/(F \times R)$	1 μs
Length traveled during one UoT	200 m

Time to go through the cycle RS	100 UoT
Emission time ET	500 UoT
Period P	1,000 UoT
Number of RRH	5
Number of nodes k	5
Load induced by C-RAN traffic	50%
Load induced by BE traffic	40%

Table 1.1 – Parameters of the N-GREEN architecture.

Unsurprisingly, the latency of the C-RAN traffic is better when we prioritize the C-RAN messages, while the BE traffic is heavily penalized. Furthermore, there is still 10% of the C-RAN traffic with a latency higher than $50\mu s$, a problem we address in the next section.

Remark that, due to the broadcast and select mode, a message coming from any node induces the same load for all the nodes of the ring. Hence the latency of the traffics coming from any RRHs or from the BBUs are the same, which may seem couterintuitive knowing that all BBUs share the same node on the ring. This is why in Fig. 1.1 we do not ditinguish between uplink C-RAN traffic (RRH to BBU) and downlink C-RAN traffic (BBU to RRH).

1.2.3 Deterministic approach for zero latency

1.2.3.1 Reservation

Finding good offsets for the C-RAN traffic is a hard problem even for simple topologies and without BE traffic, as we have shown in previous chapters. In this section, we give a simple solution to this problem in the N-GREEN optical ring, and we adapt it to minimize the latency of the BE traffic.

Let u be the node to which is attached the RRH i. To ensure zero latency for the C-RAN traffic, the container which arrives at u at time o_i must be free so that the data from the RRH can be sent immediately on the optical ring.

To avoid latency between the arrival of the data from the RRH and its insertion on the optical ring, we allow nodes to **reserve** a container one round before using it. A container which is reserved cannot be filled by any node except the one which has reserved it (but it may not be free when it is reserved). If u reserves a container at time $o_i - RS$, then it is guaranteed that u can fill a free container at time o_i with the data of the RRH i. In the method we now describe, the C-RAN packets never wait in the node: The message sent by the RRH i arrives at its BBU at node v at time $o_i + \omega(u,v)$ and the answer is sent from the BBU at time $o_i + \omega(u,v) + 1$.

Recall that an RRH fills a container every F units of time, during a time ET. Thus if we divide the period P into **slots** of F consecutive units of time, an RRH needs to fill at most one container each slot. If an RRH emits at time o_i, then we say it is at **position** $o_i + \omega(u,v) \pmod F$. The position of an RRH corresponds to the position in a slot of the container it has emitted, when it arrives at v, the node of the BBU. If an RRH is at position p, then by construction, the corresponding BBU is at position $p+1 \pmod F$. For now, we do not allow waiting times for C-RAN traffic, hence each RRH uses a container at *the same position during all the emission time.*

Given a ring, a set of RRH's, a period and an acceleration factor F, the problem we solve here is to find an **assignment** of values of the offsets o_i's which is **valid**: two RRHs must never use the same container in a period. Moreover we want to preserve the latency of the BE traffic. It means that the time a BE packet waits in the insertion buffer must be minimized. To do so, we must minimize the time a node waits for a free container at any point in the period, by spreading the C-RAN traffic as uniformly as possible over the period.

Figure 1.7 represents an assignment of two couples of RRH and BBU by showing the containers going through the node of the BBU during a period. Each slot has a duration of F unit of times, and, since an RRH/BBU emits a packet each F UoT during ET UoT, if we take the granularity of a slot to represent the time, the emission of a BBU/RRH is continuous in our representation, during ET/F slots. A date t in the period corresponds in Figure 1.7 to the slot t/F and is at position $t \mod F$.

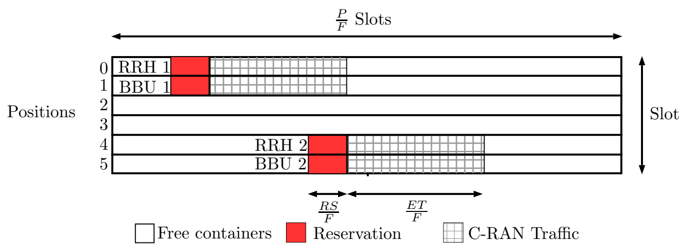

Figure 1.7 – A valid assignment with $F = 6$.

1.2.3.2 Building valid assignment with zero C-RAN latency

Remark that two RRHs which are not at the same position never use the same containers. Moreover, if we fix the offsets of the RRHs to even positions so that they do not reserve the same containers, then, because the answers of the BBU are sent without delay in our model, it will fix the offsets of the BBUs to odd positions which do not reserve the same containers. Hence, we need to deal with the RRHs only. The next proposition gives a simple method to find an assignment.

Proposition 1. *There is a valid assignment of the offsets $o_1, \ldots, o_k$ on the same position if $kET + RS \leq P$.*

Proof. W.l.o.g we fix o_1 to 0 and all the other offsets will then be chosen at position 0. Let $u_1, \ldots, u_k$ be the nodes attached to the RRHs $1, \ldots, k$. We assume that $u_1, \ldots, u_k$ are in the order of the oriented cycle. The last message emitted by the RRH 1 arrives at u_2 at time $ET - 1 + \omega(u_1, u_2)$. Therefore we can fix $o_2 = ET + \omega(u_1, u_2)$. In general we can set $o_i = (i-1) \times ET + \omega(u_1,u_i)$ and all RRHs will use different containers at position 0 during a period. By hypothesis $k \times ET + \omega(u_1,u_1) \leq P$, thus the containers filled by the k-th RRH are freed before P. Hence, when the RRH 1 must emit something at the first unit of time of the second period, there is a free container. □

Remark that reserving free containers make them unusable for BE traffic which is akin to a loss of bandwidth. However, with our choice of emission times of the RRHs in the order of the cycle, most of the container we reserve are used by the data from some RRH. If all containers at some position are used, that is $kET + RS = P$, then there are only RS free containers wasted. In the worst case, less than $2RS$ containers are wasted by the assignment of Proposition 1.

It is now easy to derive the maximal number of antennas which can be supported by an optical ring, when using reservation and the same position for an RRH for the whole period.

Corollary 1. *There is a valid assignment with* $\lfloor \frac{P-RS}{ET} \rfloor \times \frac{F}{2}$ *antennas and zero latency.*

Proof. Following Proposition 1, the maximal number of antennas for which there is an assignment on the same position is $k = \lfloor \frac{P-RS}{ET} \rfloor$. In such an assignment, we need a second position to deal with the traffic coming from the BBUs coming back to those k antennas. Since we got F positions in the slot, the number of antennas supported by the ring is thus equal to $k \times \frac{F}{2}$. □

With the parameters of the N-GREEN ring given in Figure 1.1, we can support 5 antennas, while stochastic multiplexing can support 10 antennas albeit with extreme latency. There are two sources of inefficiency in our method. The first comes from the reservation and cannot be avoided to guarantee the latency of the C-RAN traffic. The second comes from the fact that an RRH must emit at the same position during all the emission time (to guarantee zero latency). We relax this constraint in Section 1.2.3.3 to maximize the number of antennas supported by the ring, while minimizing the loss of bandwidth due to reservation.

We now present an algorithm using reservation as in Proposition 1 to set the offsets of several RRHs at the same position. In a naive assignment, we put each RRH in an arbitrary position, for instance one RRH by position. We then propose three ideas to optimize the latency of the BE traffic, by spacing as well as possible the free containers in a period.

Balancing inside the period With the parameters of the N-GREEN ring given in Figure 1.1 ($ET = \frac{P}{2}$, $F = 10$ and $n = 5$), there are no unused position. Any assignment has exactly one BBU or RRH at each position. If all the RRHs start to emit at the first slot, then during ET there will be no free container anywhere on the ring, inducing a huge latency for the BE traffic. To mitigate this problem, in a period, the time with free containers in each position must be uniformly distributed over the period as shown in Fig. 1.8.

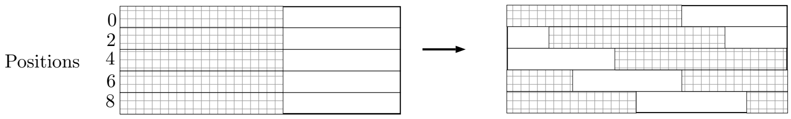

Figure 1.8 – Balancing inside the period.

Compacting positions For each position which is used by some RRH, and for each period, at least RS free containers are reserved which decreases the maximal load the system can handle. Therefore to not waste bandwidth, it is important to put as many RRHs as possible on the same position as shown in Fig. 1.9. Indeed, for any position which is not used at all, no container needs to be reserved. This strategy is also good to spread the load during the period since it maximizes the number of unused positions and for each unused position there is a container free of C-RAN traffic each F unit of times.

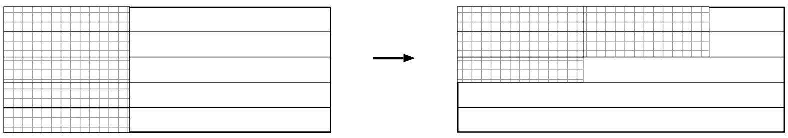

Figure 1.9 – Compacting positions.

Balancing used positions The free positions can be distributed uniformly over a slot, to minimize the time to wait before a node has access to a container from a free position, as shown in Fig. 1.10. To do so, compute the number of needed positions $x = \lceil k \times \frac{ET}{P-RS} \rceil$, with k the number of antennas using the previous strategy. Then, set the x used positions in the following way: $\lfloor \frac{F}{x} \rfloor - 1$ free positions are set between each used positions. If $\frac{F}{x}$ has a reminder r, then we set the r free remaining positions uniformly over the interval in the same way and so on until there are no more free position. It is a small optimization, since it decreases the latency by at most $F/2$.

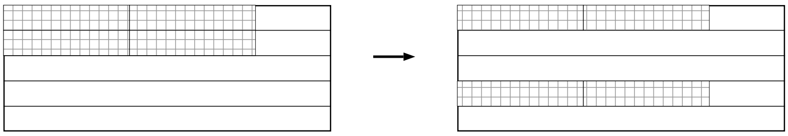

Figure 1.10 – Balancing used positions.

Experimental evaluation Our algorithm *combines the three methods* we have described to spread the load over the period. In order to understand the interest of each improvement, we present the cumulative distribution of the latency of the BE traffic using them either alone or in conjunction and we compare our algorithm to stochastic multiplexing with C-RAN priority.

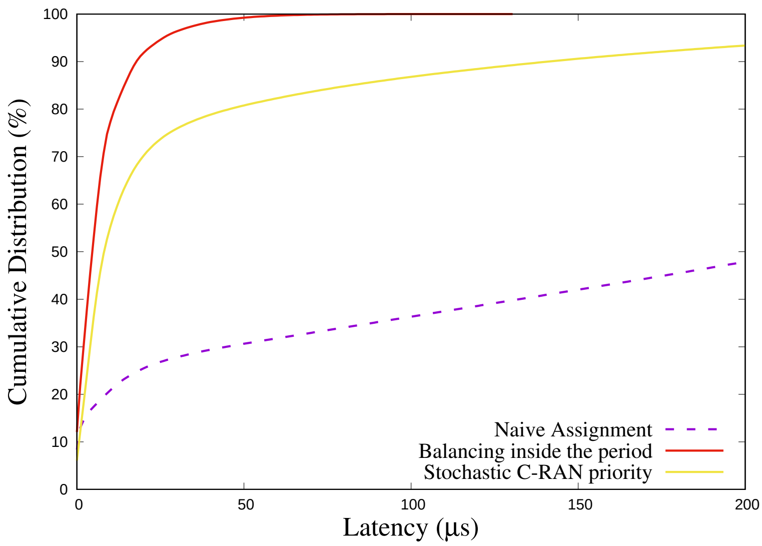

Figure 1.11 – BE latencies of a naive assignment and balancing inside the period for 5 antennas.

Figure 1.11 shows the performance of balancing the C-RAN traffic inside the period against a naive assignment in which all the RRH begin to emit at the same slot. We keep the same parameters as in Section 1.2.2 (see Table 1.1). As expected, the BE traffic latency is much better when we balance the C-RAN traffic inside the period and already much better than stochastic multiplexing.

To show the interest of compacting the positions, we must be able to put several RRHs at the same position. Hence, we change the emission time to $ET = 200$ and the number of antennas to $k = 12$ to keep the load around 90% as in the experiment of Figure 1.6. This is not out of context since the exact split of the C-RAN (the degree of centralization of the computation units in the cloud) is not fully determined yet [11].

As shown in Figure 1.12, the performance of the naive assignment is really bad. Compacting the RRHs on a minimal number of positions decreases dramatically the latency. If in addition, we balance over a period, we get another gain of latency of smaller magnitude: the average (respectively maximum) latency for BE traffic goes from 4.76μs (respectively 48μs) to 3.28μs (resp. 37μs). We did not represent the benefit of balancing used positions because the reduction in latency it yields is small as expected: the average (respectively maximum) latency for BE traffic goes from 4.76μs (resp. 48μs) to 4.43μs (resp. 44μs).

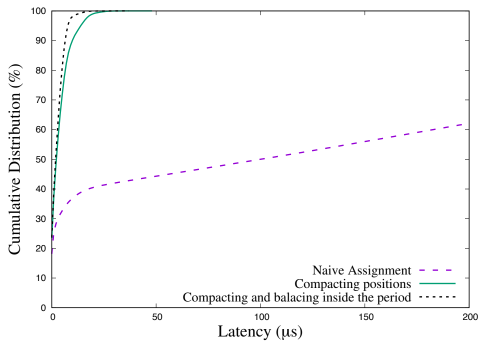

Figure 1.12 – BE latencies of compacting positions and balancing inside the period for 12 antennas.

In Figure 1.13, we compare the cumulative distribution of the latency of the BE traffic using the FIFO rule to our reservation algorithm with the three proposed improvements. The parameter are the same as in the previous experiment. The performance of our reservation algorithm is excellent, since the C-RAN traffic has *zero latency* and the BE traffic has a *better latency* than with the FIFO rule despite the cost of reservation. It is due to the balancing of the load of the C-RAN traffic over the period, that guarantee a more regular bandwidth for the BE traffic.

1.2.3.3 Building Valid Assignments with Additional C-RAN Latency

The previous approach limits the number of antennas supported by the ring when $P - RS \mod ET \neq 0$, which is the case with N-GREEN parameters. The method we present in

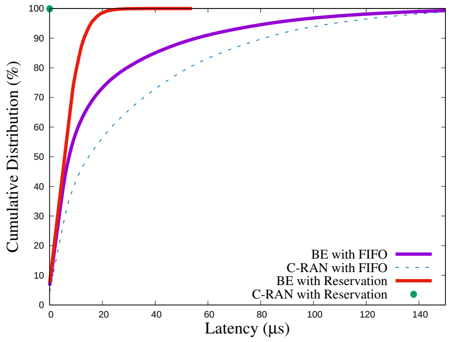

Figure 1.13 – FIFO buffer compared to the best method with reservation for 12 antennas.

this section enables us to support more antennas and improves the latency of BE traffic (it reserves less free containers) by *allowing the data from an RRH to use two positions.* It is at the cost of a slightly worse latency for C-RAN traffic and it also requires in practice to implement some buffering for the C-RAN packets.

In order to support as much antennas as possible on the ring, we use *all* containers in a given position, improving on the compacting position heuristic.

Proposition 2. *There is a valid assignment for k antennas when $k \leq \lfloor \frac{P-RS}{ET} \times \frac{F}{2} \rfloor$.*

Proof. We consider the RRHs in the order of the ring. Let $l = \lfloor \frac{P-RS}{ET} \rfloor$, then we set the offsets of the first l RRHs as in Proposition 1. These RRHs are at position zero and the $(l+1)$th RRH first emits at position zero, with offset $o_{l+1} = l * ET + \omega(u_0, u_{l+1})$.

The $(l+1)$th RRH emits up to time $P - \omega(u_{l+1}, u_0)$ at position zero, so that there is no conflict with RRH 0 during the next period. Hence, it has used the position zero during $x = P - \omega(u_{l+1}, u_0) - l * ET - \omega(u_0, u_{l+1}) = P - l * ET - RS$. From time $P - \omega(u_{l+1}, u_0) + 2$, the $(l+1)$th RRH emits at position 2 and during a time $ET - x$. Then the next RRH in the order is assigned to position 2, and begins to emit at time $P - \omega(u_{l+1}, u_0) + ET - x$ instead of zero. The rest of the assignment is built in the same way filling completely all first positions, until there are no more RRH. □

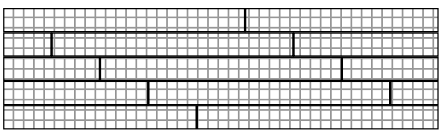

Figure 1.14 – Valid assignment for 9 antennas and the N-GREEN parameters.

Figure 1.14 illustrates the construction of Proposition 2 for the N-GREEN parameters. The loss due to reservation is exactly RS containers by used positions. Hence, it is possible to support 9 antennas (but no BE traffic in this extreme case), rather than 5 with the method of Section 1.2.3.2.

We call this new reservation algorithm **saturating positions** since it improves on compacting positions of the previous subsection. Moreover, there are no free slots in used positions, hence the idea of balancing into the period is not relevant. The only possible optimisation would be to balance the used positions, but it is not worth it since it adds additional latency for the RRHs using two different positions.

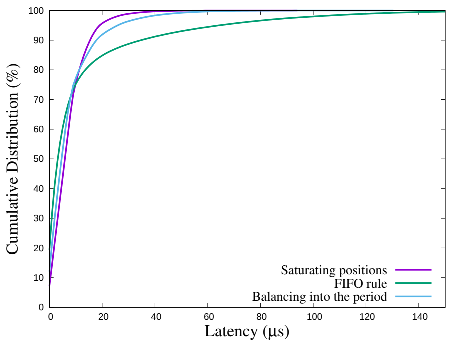

Figure 1.15 – Latencies of saturating positions, balancing into the period and FIFO rule for 5 antennas.

Figure 1.15 represents the cumulative distribution of the latency of BE traffic for the FIFO rule, saturating position, and balancing into the period using the N-GREEN parameters. Saturating positions reduces the BE traffic latency more than balancing into the period. This is easily explained by its lesser use of reservation. It is at the cost of a maximal latency of 2 μs for C-RAN traffic, so the designer can choose any of the two algorithms, according to the desired latency for C-RAN and BE traffic.

Conclusion

The concept of Cloud-RAN is to use non-dedicated networks, that is, networks shared with other applications. In this chapter, we study the impact of the scheduling of C-RAN flows on the latency of Best-Effort flows.

Our algorithm `ASPMLS` used to solve PALL on star routed networks tend to create long sequence of contiguous messages that monopolize the ressources during a long time. Hence, the latency of the Best-Effort flows is consequently worsen. To solve this issue, we virtually change the size τ of the datagram to the largest possible value τ' for which `ASPMLS` finds a solution. Then, we use the computed scheduling with the size of datagram τ, that leaves $\tau' - \tau$ free tics of time between every datagrams. Such an approach is possible because we are free to chose any offset in the period without impacting the latency. When solving SPALL, this approach would not be reasonable in term of latency, and we did not investigate yet the impact of our algorithms on Best-Effort latency.

We also present similar results on optical ring, developed for ANR project N-Green. In N-Green optial ring, the technical conception of the equipments makes the problem of scheduling the C-RAN flows trivial. We developped several techniques to smooth the load of the C-RAN flow over the period in order to let regular free tics for the Best-Effort traffic.

In both case, in order to ensure that a ressource scheduled for a route is free at the exact moment it is needed, we propose some reservation mechanisms. Reservation creates artificial use of bandwidth, which should result in lower latency for Best-Effort flows. In fact, it appears that Best-Effort latency is better when the C-RAN traffic is managed while smoothing the load on the period, even with the reservation than when all flows follows statistical multiplexing laws.

Mixing several kinds of flow and following a scheduling for a part of them is one of the major technical issue currently studied for deterministic networking. We detail in next chapter how released standards leads us to deterministic management of the flows.

Bibliography

[1] Dominique Barth, Maël Guiraud, and Yann Strozecki. "Deterministic Scheduling of Periodic Messages for Cloud RAN". In: *CoRR* abs/1801.07029 (2018). arXiv: 1801.07029. URL: http://arxiv.org/abs/1801.07029.

[2] *Time-Sensitive Networking Task Group*. http://www.ieee802.org/1/pages/tsn.html. Accessed: 2021-02-10.

[3] Dominique Barth, Maël Guiraud, and Yann Strozecki. "Deterministic Contention Management for Low Latency Cloud RAN over an Optical Ring". In: *Optical Network Design and Modeling - 23rd IFIP WG 6.10 International Conference, ONDM 2019*. Vol. 11616. Lecture Notes in Computer Science. Springer, 2019, pp. 479–491.

[4] Dominique Chiaroni. "Network Energy: Problematic and solutions towards sustainable ICT". In: *invited paper, International Commission of Optics (ICO-24)* (2017).

[5] Bogdan Uscumlic et al. "Scalable deterministic scheduling for WDM slot switching Xhaul with zero-jitter". In: *2018 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE. 2018, pp. 100–105.

[6] Menglan Jiang, Massimo Condoluci, and Toktam Mahmoodi. "Network slicing management & prioritization in 5G mobile systems". In: *European wireless*. 2016, pp. 1–6.

[7] Christian Cadéré et al. "Virtual circuit allocation with QoS guarantees in the ECOFRAME optical ring". In: *Optical Network Design and Modeling (ONDM), 2010 14th Conference on*. IEEE. 2010, pp. 1–6.

[8] Ted H Szymanski. "An ultra-low-latency guaranteed-rate Internet for cloud services". In: *IEEE/ACM Transactions on Networking* 24.1 (2016), pp. 123–136.

[9] Youssef Ait el mahjoub, Hind Castel-Taleb, and Jean-Michel Fourneau. "Performance and energy efficiency analysis in NGREEN optical network". In: *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (WiMob 2018)*. Limassol, Cyprus, Oct. 2018.

[10] *Maël Guiraud's website*. https://mael-guiraud.github.io/.

[11] China Mobile. "C-RAN: the road towards green RAN". In: *White Paper, ver* 2 (2011).