Deterministic Scheduling of Periodic Messages for Cloud RAN

Dominique Barth[1], Maël Guiraud[1,2], Brice Leclerc[2], Olivier Marcé[2], and Yann Strozecki[1]

[1]David Laboratory, UVSQ
[2]Nokia Bell Labs France

May 25, 2018

1 Model and Problems

We use the notation $[n]$ to denote the interval of n integers $\{0,\ldots,n-1\}$.

1.1 Network modeling

The network is modeled as a directed graph $G = (V,A)$. 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 $u_0,\ldots,u_l$, with $(u_i,u_{i+1}) \in A$. The **delay** of a vertex u_i in a path $r = (u_0,\ldots,u_l)$ is defined by $\lambda(u_i,r) = \sum_{0\leq j<i} \Omega(u_j,u_{j+1})$. We also define $\lambda(u_0,r) = 0$. The length of the route r is defined by $\lambda(r) = \lambda(u_l,r)$. We denote by $\mathcal{R}$ a set of routes, the pair $(G,\mathcal{R})$ is called a **routed network** and represents our telecommunication network. The first vertex of a route models an antenna (RRH) and the last one a data-center (BBU) which computes the messages sent by the antenna.

1.2 Messages dynamic

Time is discretized, hence the unit of all time values is a **tic**, the time needed to transmit a minimal unit of data over the network. The weight of an arc is also expressed in tics, it is the time needed by a message to go through this arc. In the process we study, a **frame** is sent on each route at each period, denoted by P. A frame F is composed of several **datagrams**. The size of a frame, denoted $|F|$ is the number of datagrams in the frame. The datagrams $\{d_0,\ldots,d_{|F|}\}$ of a

frame represent one or several *consecutive tics*, called length of the datagram. Once a datagram have been emitted, it can not be severed during its travel in the network. Thus, the length of a frame is $\sum_{i=0}^{|F|} d_i = \tau$. In this paper, we assume that τ is the same for all routes. Indeed, the data flow sent by an RRH to its BBU is the same, regardless of the route.

Let $r = (u_0, \ldots, u_l)$ be a route, a datagram d_j of lenght $|d_j|$ can be buffered in a node u_i, during at least $|d_j|$ tics. A datagram d_j is thus characterized by a sequence $\{b_j^0, \ldots, b_j^l\}$ of buffers corresponding to the buffering time of this datagram in a node u_i. We denote by $m_r(d_j)$ the time at which a datagram d_j is sent at time from u_0 the first vertex of r then it will arrive at vertex u_i in r at time $m_r(d_j) + \lambda(u_i,r) + \sum_{k=0}^{i-1} b_j^k$. Since the process is periodic, if the datagram from r goes through an arc at time $t \in [0,P-1]$, then it goes through the same arc at time $t + kP$ for all positive integers k. Therefore, every time value can be computed modulo P and we say that the first time slot at which a datagram d_j sent at time $m_r(d_j)$ on r reaches a vertex u_i in r is $t(d_j,u_i,r) = m_r(d_j) + \lambda(u_i,r) + \sum_{k=0}^{i-1} b_j^k \mod P$.

A frame is thus defined as follow on the route i :

$$F_i = \{(m_i(d_0),\{b_0^{0,i}, \ldots, b_0^{l,i}\}), \ldots, (m_i(d_{|F_i|}),\{b_{|F_i|}^{0,i}, \ldots, b_{|F_i|}^{l,i}\})\}$$

, where $|F|$ is the number of datagrams in the frame, and l the number of nodes in the routes.

Let us call $[t(F,u,r)]_{P,\tau}$ the set of time slots used by a frame F on a route r at vertex u in a period P, that is $[t(F,u,r)]_{P,\tau} = \{t(d_0,u,r) + i \mod P \mid 0 \leq i < |d_0|\} \cup \ldots \cup \{t(d_{|F|},u,r) + i \mod P \mid 0 \leq i < |d_{|F|}|\}$. Let r_1 and r_2 be two routes, on which frames F_1 and F_2 are sent in their first vertex. We say that the two routes have a **collision** if they share an arc (u,v) and $[t(F_1,u,r_1)]_{P,\tau} \cap [t(F_2,u,r_2)]_{P,\tau} \neq \emptyset$.

A (P,τ)**-periodic assignment** of a routed network $(G,\mathcal{R})$ is a function that associates to each datagram of each frame of each route $r \in \mathcal{R}$ its **offset** $m_r(d_i)$ that is the time at which the datagram d_i is emitted at the first vertex of the route r and its buffers $\{b_i^{0,r}, \ldots, b_i^{l,r}\}$ that is, the buffering time of the datagram in each nodes of the routes. In a (P,τ)-periodic assignment, *no pair of routes has a collision.*

1.3 Periodic route assignment

We want to find an assignment which allows to send periodic messages from sources to targets without collisions. We introduce the following associated decision problem, useful for hardness proofs.

Periodic Routes Assignment (pra)

Input: a routed network $(G,\mathcal{R})$, an integer τ and an integer P, a set of frames $\mathcal{F}$

Question: does there exist a (P,τ)-periodic assignment of $(G,\mathcal{R})$?

1.4 Periodic assignment for low latency

In the context of cloud-RAN applications, we need to send a frame from an RRH u to a BBU v and then we must send the answer from v back to u. We say that a routed network $(G,\mathcal{R})$ is **symmetric** if the set of routes is partitioned into the sets Fr of **forward routes** and Br of **backward routes**. There is a bijection ρ between A and B such that for any forward route $r \in A$ with first vertex u and last vertex v, the backward route $\rho(r) \in B$ has first vertex v and last vertex u. In all practical cases the routes r and $\rho(r)$ will be the same with the orientation of the arcs reversed, which corresponds to bidirectional links in *full-duplex* networks, but we need not to enforce this property.

We now give a new interpretation of a (P,τ)-periodic assignment of a $(G,\mathcal{R})$ symmetric routed network, so that it represents the sending of a frame and of its answer. This assignment represents the following process: First, the datagrams of a frame F_r is sent at u, through the route $r \in A$, at time $\{m_r(d_0),\ldots,m_r(d_{|F_r|})\}$. Those datagrams are received by v, i.e., the last vertex of r at times $\{t(d_0,v,r),\ldots,t(d_{|F_r|},v,r)\}$. Once v has received all the datagrams of a frame, the answer is computed and sent back in a frame F'_r. Note that $|F'_r|$ may not be equal to $|F_r|$, i.e. the answer is not necessarily under the same form as the initial frame. Thus the node v send F'_r to u on the route $\rho(r)$. The time between the arrival of the last datagram of a frame and the time the answer is sent back is called the **waiting time** and is defined by w_r

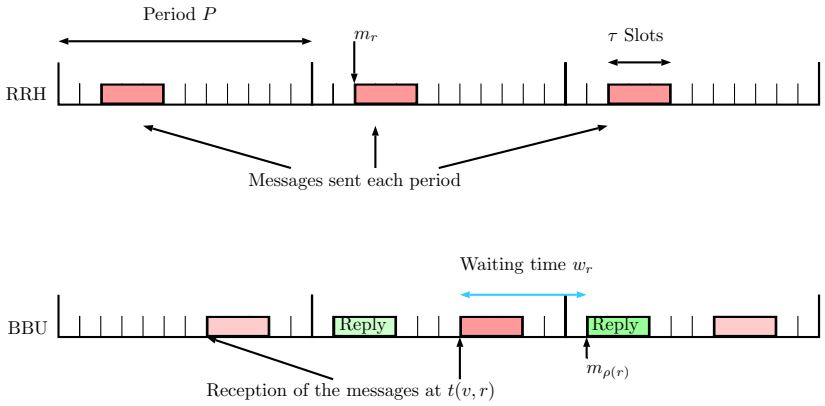

Figure 1: Periodic process

Note that, in the process we describe, we do not take into account the computation time a BBU needs to deal with one message. It can be encoded in the weight of the last arc leading to the BBU and thus we do not need to

consider it explicitly in our model.

We define by **jitter** of a frame the time at which the BBU receive the last datagram of the frame, minus the time at which the RRH has sent the first datagram of the frame, minus the length of the frame, minus the length of the route. In other words, if the RRH sends the datagram d_0 first on the route r at time $m_r(d_0)$, and the BBU v receive the last datagram of the frame d_i at time $t(d_i,v,r)$, the jitter J_r of a frame is then $J_r = t(d_i,v,r) - m_r(d_0) - \tau - \lambda(r)$. The whole process time for a route r is equal to $PT(r) = \lambda(r) + w_r + \lambda(r) + J_r$. In the process time, we count the time between the time the first slot of the frame is emitted and the first time at which the first slot of the message comes back. Alternatively we could consider the time between the emission of the first slot and the reception of the last slot of the message, which adds τ to the process time. Both definitions are equivalent in our context where all messages are of size τ, hence we chose the first definition which is slightly simpler. Each route must respect a time limit that we call *deadline*. To represent these deadlines, we use a deadline function d, which maps to each route r an integer such that $PT(r)$ must be less than $d(r)$.

We consider the following decision problem.

Periodic Assignment for Low Latency (pall)

Input: A symmetric routed network $(G,\mathcal{R})$, the integers P, τ and a deadline function d.

Question: does there exist a (P,τ)-periodic assignment m of $(G,\mathcal{R})$ such that for all $r \in \mathcal{R}$, $PT(r) \leq d(r)$?

As a consequence of the NP-hardness of PRA, we show in the next subsection that this problem is NP-hard. In Section **??** we will study heuristics used to solve the search version of PALL (computing an assignment), also denoted by PALL for simplicity. In PALL, we have chosen to bound the process time of each route, in particular we can control the worst case latency. It is justified by our C-RAN application with hard constraint on the latency. It would be interesting to study the case of a constraint on the *average* of process times (or equivalently waiting times) of routes, which may be more relevant in other contexts.

The following table summarize the main notations used in the paper.

$(G,\mathcal{R})$	Routed network
$\Omega(u,v)$	Weight of the arc $(u,v) \in A$
$\lambda(u_i,r)$	Latency of the vertex u_i in r
$\lambda(r)$	Length of the route r
P	Period
τ	Size of a message
$[t(v,r)]$	Set of time slots used by route r at vertex v in a period P
$m = (m_0, \ldots, m_{n-1})$	Assignment: an offset for each route
w_r	Waiting time of the route r
$PT(r)$	Process time of the route r
$d(r)$	Deadline of the route r