Contention Management for 5G

DB,CC,MG,OM,YS

January 20, 2017

Abstract

This article treats about Contention Management for 5G.

1 Introduction

- Context and problematic
- Related works
- Article contribution

2 Model, Problems

2.1 Definitions

A network can be considered as a directed graph $G = (V, A)$ with two non intersecting subsets of vertices: a subset L of nodes which are called **leaves** and a subset S of nodes which are called **sources**. The indegree of nodes in S, and the outdegree of nodes in L are equal to 0. We denote by $\mathcal{L}$ the cardinal of L and by $\mathcal{S}$ the cardinal of S. Each arc (u, v) in A has an integer weight $Dl(u, v)$, called **delay**, representing the time taken by a data to go from u to v using this arc.

We consider $G^r = (V, A^r)$ wherein the set of vertices is the same as in G, and A^r represents the edges of A directed in the other way.

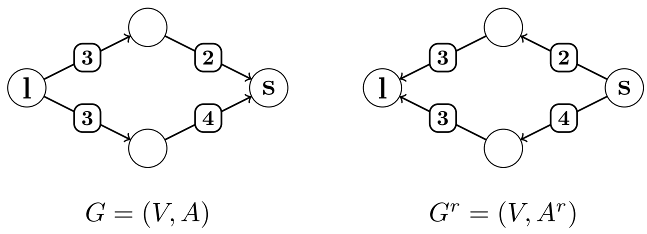

Figure 1: G and G^r with $\mathcal{L} = \mathcal{S} = 1$.

A **route** r is a sequence of arcs $a_0, \ldots, a_{n-1}$, with $a_i = (u_i, u_{i+1}) \in A$ such that $u_0 \in L$ and $u_n \in S$. The **latency** of a vertex u_i in r, with $i \geq 1$, is defined by

$$\lambda(u_i, r) = \sum_{0 \leq k < i} Dl(a_k)$$

We also define $\lambda(u_0, r) = 0$. The latency of the route r is defined by $\lambda(r) = \lambda(u_n, r)$. In graph theory, a route is a simple path in the graph, and its latency is its weight.

A **routing function** $\mathcal{R}$ is an application associating a route $\mathcal{R}(s, l)$ to each couple $(s, l) \in S \times L$ in G. Moreover $\mathcal{R}$ satisfies the **coherent routing** property: the intersection of two routes must be a path.

For simplicity, we assume that we have as many source nodes as we have leaves ($\mathcal{S} = \mathcal{L}$). A $\mathcal{R}$**-matching** is a bijection $\rho : S \to L$ which associates to each $s_i \in \{s_0, ..., s_{\mathcal{S}-1}\}$ a $l_i \in \{l_0, ..., l_{\mathcal{L}-1}\}$. A $\mathcal{R}$-matching defines a set $\{r_0, \ldots, r_{\mathcal{L}-1}\}$ of $\mathcal{L}$ routes in $\mathcal{R}$ such that $\forall i, r_i = \mathcal{R}(s_i, l_i)$.

The quintuplet $N = (G, S, L, \mathcal{R}, \rho)$ defines a **matched graph**. We call N^r the quintuplet $(G^r, S, L, \mathcal{R}, \rho^r)$, where ρ^r is the $\mathcal{R}$-matching obtained using the same routes, with inverted arcs.

2.2 Slotted time Model

In our model, the time is discrete. The unit of time is one slot. Two values are expressed in time slots:

1. The emission time of a message by a node, the **message length**, We denote by **T** this time.
2. The time taken by a message to cross a link, the delay of an arc.

   Let $P > 0$ be an integer called **period**. A P**-periodic affectation** of N(a matched graph) consists in a set $\mathcal{M} = (m_0, \ldots, m_{\mathcal{L}-1})$ of $\mathcal{L}$ integers that we call **offset**. Each period is divided in P slots and the number m_i represents the first slot number used by the route r_i at its source. We define the first time slot at which a message arrive at any vertex v of the route by

$$t(v, r_i) = m_i + \lambda(u, r_i) \mod P.$$

Let us call $[t(v, r_i)]_{T,P}$ the values of the time slots used by a route r_i in a vertex v. Those values are forming a continuous set of values starting at $t(v, r_i) \mod P$ and ending at $t(v, r_i) + T \mod P$. For a given instance, P and T does not change at any moment, indeed, the size of the messages is the same for any route, and the period is also the same for any vertex. So, since P and T are fixed, we simplify the notation by $[t(v, r_i)]$.

A P-periodic affectation must have no **collision** between two routes in ρ, that is $\forall r_i, r_j \in \rho, i \neq j$, we have

$$[t(u, r_i)] \cap [t(u, r_j)] = \emptyset.$$

2.3 Problems

The application we address here by studying the problems defined above is the following. Consider a fronthaul network in which source nodes in S represent BBU. Each of the source nodes do a computation for its associated RRH represented by a leaf node in L. Consider a $\mathcal{R}$-matching ρ of S in L. Consider a leaf l, its dedicated source node s and $R(l, s)$ the route from l to s in R. We consider a P-periodic affectation of N, and also another P-periodic affectation of N^r. The periodic process is, for each route, the following one:

(a) During each period of duration P, l sends a message to its source $s = \rho(l)$, using its routes according to the P-periodic affectation of N.

(b) When a source $s = \rho(l)$ receive a message from l, it makes a computation. This computation time is the same for all sources.

(c) After this computation time, s sends a message to $l = \rho(l)$ according to the P-periodic affectation of N^r.

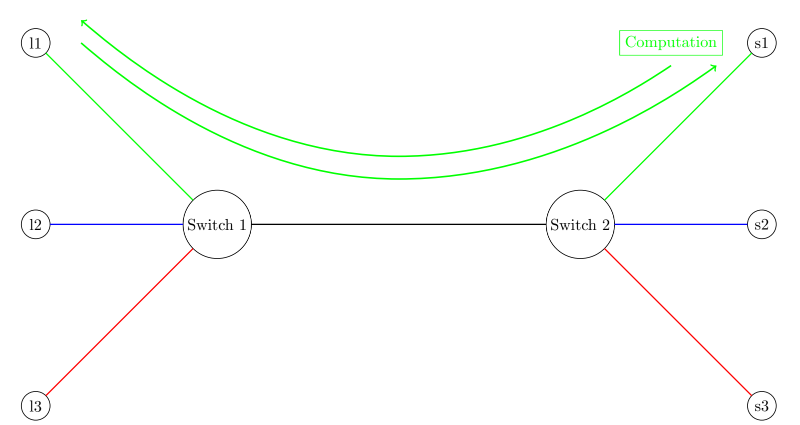

Figure 2: The periodic process of a message on a route in the fronthaul.

On source nodes, between the end of the computation time and the emission time of the message (given by the P-periodic affectation), there is a **waiting time**. We define by $\omega : r \to \mathbb{N}$ the waiting time of a route r in the $\mathcal{R}$-matching considered, i.e. the time during which the message is "sleeping", waiting to be sent through the network.

We have to find two P-periodic affectations, one for the messages going from the RRH to the BBU, and another one for the answers going from BBU to the RRH. Those two P-periodic affectations have the same period P. If the messages have to be buffered, we can only do it in sources nodes. We define by θ the computation time required at the source node before sending an answer to its leaf node.

A **TwoWayTrip** is, for a route, the full travel leaf-leaf, including the waiting time and the computation time.

Let us call $T(r)$ the time of a TwoWayTrip on a route r:

$$T(r) = 2\lambda(r) + \omega(r) + \theta$$

Since θ is the same on every route, we can simplify the model by removing θ. Whether we want to consider it, we only have to lengthen all links before source nodes by $\frac{\theta}{2}$.

In our network application, since P and the $\mathcal{R}-matching$ are given, we do not need to minimize P. Therefore we want to optimize the time taken by the messages to do the TwoWayTrip in order to ensure a good level of quality of service.

A **TwoWayTrip affectation** of N is a set of pairs $((m_0, x_0), ..., (m_{\mathcal{L}-1}, x_{\mathcal{L}-1}))$ in which $\mathcal{M} = (m_0, ..., m_{\mathcal{L}-1})$ is a P-periodic affectation of N,$\mathcal{X} = (x_0, ..., x_{\mathcal{L}-1})$ is a P-periodic affectation of N^r, and we define the waiting time of the route r_i by:

$$\omega_i = x_i - (m_i + \lambda(r_i)) \mod P.$$

Since $\mathcal{M}$ and $\mathcal{X}$ are some P-periodic affectation, they must have no collisions as we have already defined.

In the network problem, it is not allowed to have a route such that $T(r) > D$. This maximal value D is called the **deadline**. Thus, the problem we want to solve is the following:

Periodic Assignment for Low Latency (PALL)

Input: Matched graph N, integer P, T_{max}.

Question: does there exist a TwoWayTrip affectation of N, such that $\forall r \in \rho$, $T(r) \leq T_{max}$.

Once this decision problem established, we can consider two optimization problems, derived from the previous problem in which we try to minimize different functions of $T(r)$.

Optimization goal 1: minimizing $\max(T(r))$.

Minimizing the longest TwoWayTrip time of all routes allows us to win some time, and consequently some distance between the BBU and the RRH.

Optimization goal 2: minimizing $\sum_{r\in\rho} T(r)$ (equivalent to minimizing $\sum_{r\in\rho} \omega(r)$.
By minimizing the sum of all the routes, we allow a better global Quality of Service through the network.

3 PRA Solving

3.1 NP-Hardness

Theorem 1. *Problem PRA cannot be approximate within a factor* $n^{1-o(1)}$ *unless* $\mathsf{P} = \mathsf{NP}$ *even when the load is two and* n *is the number of vertices.*

Proof. We reduce PRA to graph coloring. Let G be a graph instance of the k-coloring problem. We define H in the following way: for each vertex v in G, there is a route r_v in H. Two routes r_v and r_u share an edge if and only if (u, v) is an edge in G and this edge is only in this two routes. We put weight inbetween shared edges in a route so that there is a delay k between two such edges.
As in the previous proof, a k-coloring of G gives a k-periodic schedule of H and conversly. Therefore if we can approximate the value of PRA within a factor f, we could approximate the minimal number of colors needed to color a graph within a fator f, by doing the previous reduction for all possible k. The proof follows from the hardness of approximability of finding a minimal coloring [**?**]. □

3.2 MIN-PRA

Exemple de cas simple

4 Proposed Solutions, solving PALL

4.1 Intro

PALL NP-Hard car PRA NP-Hard
Résultats valables sur Topologie 1 avec nos paramètres

4.2 No waiting times

4.2.1 Star affectation

Définir star affectation en partant de PALL

4.2.2 Shortest-longest

Algo

Period

4.2.3 Exhaustive generation

Décrire l'algo, expliquer les coupes

4.2.4 Results

Resultats des simulations : Shortest-longest optimal pour ces parametres.

4.3 Allowing waiting times

4.3.1 Intro

Importance des waiting times quand la période est donnée (Résultats D'éxepriences et preuve avec l'exemple)

4.3.2 LSG

Algorithm

Analysis Parler de LSO et expliquer pourquoi LSG mieux avec nos params

4.3.3 Results

Random

Distributions

5 Conclusion