Contention Management for 5G

DB,CC,MG,OM,YS

January 24, 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

We consider a symmetric directed graph $G = (V, A)$ modeling a slotted network. Each arc (u, v) in A is characterized by an integer delay $Dl(u, v) \geq 1$ representing the number of slots from u to v on this arc. Note that for any arc (u, v), $Dl(u, v) = Dl(v, u)$.

A route r in G is a sequence of consecutive arcs $a_0, \ldots, a_{k-1}$, with $a_i = (u_i, u_{i+1}) \in A$. The *latency* of a vertex u_i in r, with $i \geq 1$, is defined by

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

We also define $\lambda(u_0, r) = 0$. The latency of the route r is defined by $\lambda(r) = \lambda(u_k, r)$. A routing function $\mathcal{R}$ in G associates a route from u to v for any couple of different vertices $< u, v >$ in G.

Let $\mathcal{C}$ be an assignment in G, i.e., set of couples of different vertices of G. We denote by $\mathcal{R}_{\mathcal{C}}$ the set of routes $\mathcal{R}(u, v)$ for any $< u, v >$ in $\mathcal{C}$.

2.2 Slotted time Model

Consider now a positive integer P called **period**. A P**-periodic affectation** of $\mathcal{C}$ in $(G, \mathcal{R})$ consists in a set $\mathcal{M} = (m_0, \ldots, m_{c-1})$ of c integers that we call **offset**, with c the cardinal of $\mathcal{C}$. Indeed, time is consider as consecutive periods of P slots each and the number m_i represents the first slot number used by the route $r_i \in \mathcal{R}_\mathcal{C}$ at its source. We define the first time slot at which a message reaches any vertex v in this route by

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

Let us call $[t(v, r_i)]$ the values of the time slots used by a route r_i in such a vertex v. Those values are forming a continuous set of values starting at $t(v, r_i) \mod P$ and ending at $t(v, r_i) + \tau \mod P$. A P-periodic affectation must have no **collision** between two routes in $\mathcal{R}_\mathcal{C}$, that is $\forall r_i, r_j \in \mathcal{R}_\mathcal{C}, i \neq j$, with τ the size (in number of consecutive slots) of each message that must be periodically sent on each route of $\mathcal{R}_\mathcal{C}$, we have

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

2.3 Problems

The main theoretical problem we have to deal with in this context is the following.

Problem Periodic Routes Assignment (PRA)

Input: graph $G = (V, A)$, set $\mathcal{C}$ of couples of vertices, routing function $\mathcal{R}$, integer P.

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

We deal in next section with complexity of Problem PRA.

Total-latence.pdf

Figure 1: Complete process for a leaf in L.

In the context of cloud-RAN applications, we consider here in digraph $G = (V, A)$ modeling the target network two disjoint subsets of vertices S and L, where S is the set of BBU and L is the set of RRH. We consider a **matching** defined as an application $\rho : S \to L$. We consider a P-periodic affectation $\mathcal{M}$, associating respectively integers m_l and $m_{\bar{l}}$ to couples $(l, \rho(l))$ and $(\rho(l), l)$ for all $l \in L$. The process realized periodically (i.e. initiated in each window of size P) for each leaf $l \in L$ is the following one (see Figure **??**). First, a message of τ slots is sent from l to $\rho(l)$ on $\mathcal{R}(l, \rho(l))$ at slot m_l in the current window. After receiving this message,

$\rho(l)$ computes it during a time equal to θ slots. Then, a message of τ slots containing the computed result is sent back from $\rho(l)$ to l on $\mathcal{R}(\rho(l), l)$, at the first occurrence of step $m_{\bar{l}}$ in a window (i.e., in the current window at the end of computation or the next one). We denote by $\omega(l)$ the **waiting time**of l, i.e., the number of slots between the end of the computation time in $\rho(l)$ and the first occurrence of step $m_{\bar{l}}$ in a window. Thus, the whole proccess time for l is equal to

$$PT(l) = \lambda(\mathcal{R}(l, \rho(l))) + \theta + \omega(l) + \lambda(\mathcal{R}(\rho(l), l)).$$

Let us denote by $\mathcal{C}_\rho$ the set of couples $< l, \rho(l) >$ and $< \rho(l), l >$, for any $l \in L$. Consider a P-periodic affectation $\mathcal{M}$ of $\mathcal{C}_\rho$ in $(G, \mathcal{R})$. The maximum process time of $\mathcal{M}$ is equal to equal to $MT(\mathcal{M}) = \max_{l \in L} PT(l)$. Thus, we have to deal with the following problem.

Problem Periodic Assignment for Low Latency(PALL)

Input: a digraph G, a matching ρ of a set S into a set L in G, a routing function $\mathcal{R}$, a period P, an integer T_{max}.

Question: does there exist a P-periodic affectation $\mathcal{M}$ of $\mathcal{C}_\rho$ in $(G, \mathcal{R})$ such that $MT(\mathcal{M}) \leq T_{max}$?

The related optimisation problem we will focus on consists in minimizing $MT(\mathcal{M})$. Note that in the context of cloud-RAN networks, we consider $P = 1ms$, $\theta = 2.6ms$ and T_{max} must be less or equal to $3ms$.

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