Service Placement under Affine Delay Constraint

Yannick Carlinet, Orso Forghieri, Nancy Perrot

Service Placement under Affine Delay Constraint

Yannick Carlinet[1], Orso Forghieri[2], Nancy Perrot[1]

[1] Orange Innovation, Chatillon, France
yannick.carlinet@orange.com, nancy.perrot@orange.com
[2] ENS Paris Saclay, France
orso.forghieri@ens.fr

Keywords : *Combinatorial Optimization, Edge Computing, Maximum Flow, 6G*

1 Introduction

Many future network services, like Virtual Reality or computer vision, rely on Multi-Access Edge Computing (MEC) architectures to have their strong requirements met, in terms of throughput and latency. In MEC, Computing and storage servers are deployed close to the users, so that network services and applications can be processed and delivered locally. These edge computing nodes will allow to reach the very low level latency required by these new services, while reducing the network congestion.

In this work, we present a network optimization problem related to the placement of storage and computing units in these architectures, the so-called SP-AD (Service Placement with Affine Delay) problem. This problem is studied in the context of an European project [2] which deals more broadly with the intelligence distribution and placement problems at the edge for future 6G Networks.

2 Problem Description

We consider the problem of service placement in an Edge Computing platform. The aim is to allocate physical resources to services in the edge network, the edge cloud or in the cloud.

An example of such services is Augmented Reality (AR) : the end-user terminal (e.g. connected glasses) is streaming to an application in the cloud, which in turn processes the images and encodes information in it. The resulting enriched video stream is sent back to the terminal which displays it. This service is characterized by a very strict latency requirement, because the enriched video stream must be almost on sync with the original one. It is also characterized by requiring a lot of computing resources for processing the original video stream, which prevents the processing to be performed by the terminal itself. The enriched stream can also be sent to the cloud for storage. In that case, the latency requirement is less strict, for this flow only.

Services are modeled as one task, and several flows connecting the task with different destinations.

In this problem, we have to consider the capacity of the network links, for two reasons : firstly, most of the services have bandwidth requirements, so they have to share the link resources, secondly, experience has shown that the latency induced by a link is dependent on the load on this link. As a consequence, we need to model the latency of a link as a function of its load. As a first approximation, we assume in the following that the latency is an affine function of the load, i.e. a function of the form $f(x) = \alpha x + \beta$. The β term is necessary because there is always a threshold delay when using a link, even when the link is not congested.

Each service has a requirement in term of a maximum latency that must be met. The latency of the service is computed by adding the latency of each link crossed by the service traffic. Each flow of a given service may have different latency requirements. With the example given

above, the flows of AR streams to the AR glasses have a stricter latency requirement than the flow going to the cloud for storage. In addition, services have requirements in terms of minimum resources that must be available on the host on which they run. These resources include computation (CPU or GPU), memory, storage, electrical power. These resources are additive, which means that the total resource of each type used on a particular node is the sum of the resources of all the services running on this node.

The flows from the service task may have variable throughput. Indeed the service may still be functional even with a degraded quality of service. In certain conditions, it might be necessary to lower the throughput of some flows in order to meet the latency requirement of the flow. Each service should define whether or not they accept a lower throughput, and in which range. This range is defined as a set of possible throughputs (e.g. High Definition, Low Definition). Of course, when possible, the flows should be at the highest level of throughput, so as to offer the best quality of service as possible.

The objective of SP-AD is to maximize the sum of the (weighted) throughputs of all flows. Alternative objectives include maximizing the number of flows admitted, minimizing the overall energy consumption, or minimizing a cost function.

3 Problem Formulation

The SP-AD problem is a generalization of the multi-commodity flow problem under proportional delay constraint, described by Bonami et al. [1]. In consequence, it is NP-complete.

Let G = (V, A) a connected, directed graph. We consider a set S of services and, for each service s, a set of flows F_s. The decision variables of the problem are x_f, the throughput of flow f ($x_f \in \{t_1, \ldots, t_N\}$), binary variable y_{su}, the indicator whether service s is placed on node u, and binary variable z_{fa}, the indicator whether flow f is using arc a.

Due to space limitation, we detail only the latency constraints :

$$\sum_{a \in A} z_{fa} \left[\alpha_a \left(\sum_{s' \in S} \sum_{f' \in F_{s'}} x_{f'} z_{f'a} \right) + \beta_a \right] \leq L_f \qquad \forall s \in S, \forall f \in F_s$$

with L_f the maximum admitted latency for flow f, and with the latency of arc a equal to $\alpha_a L_a + \beta_a$, with L_a the total traffic on arc a.

These constraints are clearly not linear, therefore in order to ease the resolution of the problem, the first step was to provide a linearized version of the compact formulation. Afterwards, we studied relaxation-based heuristics, as well as greedy and genetics heuristics. This Work is still in progress.

Acknowledgment

This work was partly supported by the European Union's Horizon 2020 research and innovation programme in project DEDICAT6G [2] under Grant Agreement No. 101016499.

Références

[1] Pierre Bonami, Dorian Mazauric, and Yann Vaxès. Maximum flow under proportional delay constraint. *Theoretical Computer Science*, 689 :58–66, 2017.

[2] https ://dedicat6g.eu. Dedicat6g dynamic coverage extension and distributed intelligence for human centric applications with assured security, privacy and trust : from 5g to 6g. 2020.