



**HAL**  
open science

# Mathematical Formulation for the Network Slice Design Problem

Wesley da Silva Coelho, Amal Benhamiche, Nancy Perrot, Stefano Secci

► **To cite this version:**

Wesley da Silva Coelho, Amal Benhamiche, Nancy Perrot, Stefano Secci. Mathematical Formulation for the Network Slice Design Problem. 2020. hal-02448028

**HAL Id: hal-02448028**

**<https://hal.science/hal-02448028>**

Preprint submitted on 15 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Mathematical Formulation for the Network Slice Design Problem

Wesley da Silva Coelho  
CNAM, Orange Labs, France  
wesley.dasilvacoeelho@orange.com

Amal Benhamiche  
Orange Labs, France  
amal.benhamiche@orange.com

Nancy Perrot  
Orange Labs, France  
nancy.perrot@orange.com

Stefano Secci  
CNAM, France  
stefano.secci@cnam.fr

**Abstract**—In this document, we provide a Mixed Integer Linear Program to the Network Slice Design problem, which includes novel mapping decision points rising with new 5G radio and core function placement policies. The model in particular encompasses flexible functional splitting, with possibly different splitting for different slices, and sub-slice and network function decomposition.

## I. PROBLEM DEFINITION

We provide a 5G network model and design problem statement, taking into account the presented requirements. Table I summarize the used notations.

### A. Physical layer model

We associate with the physical layer a directed graph  $G_p = (V_p, A_p)$  where  $V_p$  is the set of nodes and  $A_p$  the set of arcs.  $V_p$  is composed of disjoint sub-sets,  $V_p^{du}$ ,  $V_p^{ac}$ , and  $V_p^{ap}$ , containing the distributed unities, aggregation and core servers, and application nodes, respectively, in such a way that  $V_p^{du} \cup V_p^{ac} \cup V_p^{ap} = V_p$  and  $V_p^{du} \cap V_p^{ac} = V_p^{du} \cap V_p^{ap} = V_p^{ac} \cap V_p^{ap} = \emptyset$  hold. Every node  $u \in V_p$  is associated a number of available CPU  $c_u$ . Moreover, an arc  $a = (u, v) \in A_p$  corresponds to a physical link connecting nodes  $u$  and  $v \in V_p$ . We denote by  $\delta^+(u)$  (resp.  $\delta^-(u)$ ) the sub-set of arcs going from (resp. to) node  $u \in V_p$ . Finally, each arc  $a \in A_p$  has a bandwidth capacity denoted  $b_a$ , and a latency value  $d_a$  expressing the time needed by a flow to traverse  $a$ .

### B. Virtual layer model

The virtual layer is modeled as a set of directed graphs corresponding to network slices. Every NS is composed of one or more network slice subnets with different network functions, which, in turn, are composed of a specific set of NFSs. In this work, we define an NSS as any sub-set of network functions shared among the same group of network slices.

1) *Network Function Services*: We denote by  $F$  the set of different NFS types.  $F$  is composed of the sub-set  $F^d$  of data-plane NFSs, the sub-set  $F^c$  of control-plane NFSs, and an auxiliary dummy function  $f_0$ , in such a way that  $F^d \cup F^c \cup \{f_0\} = F$  and  $F^d \cap F^c \cap \{f_0\} = \emptyset$  hold<sup>1</sup>. Regarding the uplink direction,  $F^d$  is an ordered set composed of data-plane NFSs from both access and core networks. Every

<sup>1</sup>Note we do not consider any service function (e.g. Firewall and Proxy), which can be easily added in model extensions.

TABLE I: Main notation: sets

Set	
$V_p$	Set of all nodes.
$V_p^{du}$	Set of all access nodes.
$V_p^{ac}$	Set of all non-access nodes.
$V_p^{ap}$	Set of all applications server nodes.
$A_p$	Set of all arcs.
$\delta^+(u)$	Set of all arcs going from node $u$ .
$\delta^-(u)$	Set of all arcs going to node $u$ .
$F$	Set of all NFS types.
$F^d$	Set of all data-plane NFS types.
$F^c$	Set of all control-plane NFS types.
$S$	Set of all network slice requests.
$F(s)$	Set of all CP NFS pairs that must be connected in slice $s$ .
$G(s)$	Set of all pairs of NFSs from different type sets that must be connected to each other in slice $s$ .
$K(s)$	Set of all demands of slice request $s$ .
$O(s)$	Set of origin nodes of all traffic demand from slice $s$ .
$N$	Set of all NFs.
Parameter	
$c_u$	number of available CPUs on node $u$ .
$b_a$	bandwidth value on arc $a$ .
$d_a$	delay value on arc $a$ .
$c_f$	number of CPU required by NFS $f$ .
$cap(f)$	traffic processing capacity of NFS $f$ .
$b_{fg}$	total amount of traffic generated between NFSs $f$ and $g$ by an UE.
$b_f$	expected data rate of NFS $f$ given one UE.
$d_{fg}$	the maximum accepted delay between NFSs $f$ and $g$ .
$\lambda_f$	compression coefficient of NFS $f$ .
$\alpha_f^s$	equals to 1 if a NFS type $f$ must be present in slice $s$ ; 0 otherwise.
$q_{fg}^{st}$	equals to 1 if slice request $s$ admits sharing a NFS of type $f$ with a NFS of type $g$ of slice $t$ ; 0 otherwise.
$\eta_s$	expected number of UEs connected to slice $s$
$d_s$	maximum accepted delay on data plane of slice $s$ .
$o_k$	origin node of demand $k$
$t_k$	target node of demand $k$
$b_k$	expected volume of data between sent by origin node of demand $k$ .

network function service  $f \in F$  requires the minimum number of CPUs  $c_f$  needed to be packed into a NF. Also, every NFS  $f \in F$  is associated with a traffic processing capacity  $cap(f)$ , expressed in Mbps, and an expected data rate  $b_f$  within a physical node given one UE connected to the related slice. We denote by  $b_{fg} \geq 0$  the total amount of traffic generated between NFSs  $f$  and  $g$  given one UE connected to the related NS. Additionally, we denote by  $d_{fg} \geq 0$  the maximum accepted delay<sup>2</sup> between NFSs  $f$  and  $g$ . Finally, for

<sup>2</sup>This is important when flexible functional splitting is applied on the radio access; the selected split must respect the maximum fronthaul latency proposed by standard organizations.

every  $f \in F^d$ , we denote by  $\lambda_f$  the compression coefficient on the data-plane traffic flow related to the initial volume sent by any traffic request's origin node. Lastly, all aforementioned parameters related to the auxiliary dummy function  $f_0$  are set to 0, except the compression coefficient  $\lambda_{f_0}$ , which is equal to 1.

2) *Network Functions*: We denote by  $N$  the set of network functions available to pack NFS copies. An NF  $n \in N$  might gather several NFS copies<sup>3</sup>, potentially of different types. In our model, network functions are uncapacitated entities with no resource requirements other than those demanded by the hosted NFSs.

3) *Network Slice Requests*: The set of network slice requests is denoted by  $S$ . Each request  $s \in S$  is associated with a binary parameter  $\alpha_f^s$  that takes value 1 (resp. 0) if an NFS type  $f \in F$  is (resp. is not) required to be present in the final associated virtual network. We denote by  $G_s = (V_s, A_s)$  the final directed graph associated with  $s \in S$ , with  $V_s$  being the set of virtual nodes representing the sub-set of NFs (and the hosted NFSs) serving the given slice, and  $A_s$  being a set of arcs connecting two nodes from  $V_s$ . For the control plane, we denote by  $F(s) \subseteq A_s$  the set of arcs between CP NFSs such that for any pair  $(f, g) \in F(s)$ ,  $(f \in F^c) \wedge (g \in F^c)$  holds. Additionally, we denote by  $G(s) \subseteq A_s$  the set of arcs between NFSs from different sub-sets of NFS types such that for any pair  $(f, g) \in G(s)$ ,  $(f \in F^c) \oplus (g \in F^c)$  holds. To represent the isolation requirements on the virtual layer, we denote by  $q_{fg}^{st}$  the binary parameter that takes value 1 (resp. 0) if slice request  $s \in S$  admits (resp. does not admit) packing an NFS of type  $f \in F$  with an NFS  $g$  from slice request  $t \in S$  in the same NF. Finally,  $\beta_{st}$  is binary parameter that is equal to 1 (rep. 0) if slice requests  $s$  and  $t$  are (rep. are not) NS subnets of a higher-level network slice. In addition, every request  $s \in S$  is also associated with a set  $K(s)$  of traffic demands to be routed in the physical layer. Each demand  $k \in K(s)$  is defined by a pair  $(o_k, t_k)$ , being the origin and the destination nodes of  $k$ . For any  $k$ ,  $o_k \in V_p^{du}$  and  $t_k \in V_p^{ap}$ . We denote by  $O(s)$  the set of origin nodes of all traffic demands from  $K(s)$ . Also, we denote by  $b_k$  the initial data rate sent by node  $o_k$ , in Mbps, and  $d_s$  the maximum end-to-end latency for all traffic demands in  $K(s)$ . We assume that uplink and downlink flows follow the same physical path and are treated by the same DP NFSs, in a reverse order related to each other. Due to this assumption and for the sake of simplicity, in our model we take into consideration only the uplink direction on the data-plane flow. Finally, we denote by  $n_s$  the expected number of UEs that are to be connected to slice  $s$ .

### C. Problem Statement

We define our Network Slice Design Problem (NSDP) as follows. Given a directed graph  $G_p$  representing the physical network, a set of slice requests  $S$ , a directed graph  $G_s$ , a set of traffic demands  $K(s)$  associated with each request  $s \in S$ , and

<sup>3</sup>We assume that every NF already contains an intelligent entity responsible for directing the incoming flow to the right hosted NFS.

TABLE II: Decision variables

Variable		Type
$z_f^s$	1, if functions $f$ is centralized; 0 otherwise.	Binary
$x_{nu}^{sf}$	1, if NFS $f$ installed on node $u$ is packed into NF $n$ serving slice $s$ ; 0 otherwise.	Binary
$w_{nu}^{sf}$	amount of NFS $f$ serving slice $s$ packed in NF $n$ and installed on node $u$ .	Real
$y_{nu}^f$	total number NFSs of type $f$ packed into NF $n$ and installed on node $u$ .	Integer
$\gamma_{fg}^{ka}$	1, if arc $a$ is used to route the flow between data-plane NFSs $f$ and $g$ from traffic demand $k$ ; 0 otherwise.	Binary

a set of available NFS types denoted  $F$ , the NSDP consists in determining the number of NFSs to install on the nodes of  $G_s$  for each  $s \in S$  and the size of NF hosting them, so that:

- $K(s)$  demands can be routed in  $G_s$  using these NFs,
- the NFs installed on  $G_s$  can be packed into the NFs, while satisfying the isolation constraints,
- a path in  $G_p$  is associated with each pair of NFs installed,
- the total cost is minimum,
- all technical constraints imposed by both physical and virtual layers are respected.

The objective is to minimize the total cost of deploying the network slice request while ensuring all technical constraints imposed both physical and virtual layers.

## II. MATHEMATICAL PROGRAMMING FORMULATION

This section is dedicated to introduce the mathematical model to address the Network Design Problem.

### A. Decision variables

The binary variable  $z_f^s$  takes value 1 if NFS  $f$  is centralized, and 0 otherwise.  $x_{nu}^{sf}$  is a binary variable that takes value 1 if NFS  $f$ , installed on node  $u$ , is packed into NF  $n$  serving slice  $s$ , and 0 otherwise. The variable  $w_{nu}^{sf}$  is the amount of NFS  $f$  serving slice  $s$  packed in NF  $n$  and installed on node  $u$ . The variable  $y_{nu}^f$  is the total number of NFSs of type  $f$  packed into NF  $n$  and installed on node  $u$ .  $\gamma_{fg}^{ka}$  is a binary variable that takes value 1 if arc  $a$  is used to route the flow between NFSs  $f$  and  $g$  for demand  $k$ , and 0 otherwise. Table II summarizes all decision variables used in this model.

### B. Constraints

1) *Split Selection*: Inequalities (1) decide whether a NFS  $f$  serving a slice  $s$  is installed locally or centrally. Since the RAN NFSs are chained in a specific order, all NFSs on the same side of the selected split must be installed in the same way, that is, either locally or centrally. This ordering constraint is also represented by inequalities (1). Note that we consider the uplink direction of the flow, (i.e., from DUs to application servers).

$$z_f^s \leq z_{f+1}^s, \quad \forall s \in S, \forall f \in F^d \setminus \{f_{|F^d|}\} \quad (1)$$

2) *NFS Placement*: Given a set  $K(s)$ , constraints (2) ensure that all distributed NFSs will be installed on all related origin nodes; we assume that NFSs from CP cannot be installed in a distributed manner. Constraints (3), in turn, ensure that all copies of centralized NFSs will be installed in the same physical node.

$$\sum_{n \in N} x_{nu}^{sf} = \begin{cases} 1 - z_f^s & , \text{ if } f \in F^d, u \in O(s); \\ 0 & , \text{ otherwise.} \end{cases} \quad , s \in S, \forall f \in F, u \in V_p^{du} \quad (2)$$

$$\sum_{n \in N} \sum_{u \in V_p \setminus V_p^{du}} x_{nu}^{sf} = \begin{cases} z_f^s & , \text{ if } f \in F^d; \\ \alpha_f^s & , \text{ otherwise.} \end{cases} \quad s \in S, \forall f \in F \quad (3)$$

3) *NF dimensioning*: (4) calculate the exact amount of distributed centralized NFSs for each NS request. It is important to mention that, to minimize the residual virtual resources from each NFS, this amount might be a fractional value; regarding the sharing possibilities, these values are rounding up with inequalities related to packing and capacity constraints.

$$cap(f)w_{nu}^{sf} = \begin{cases} \lambda_{f-1} b^k x_{nu}^{sf} & , \text{ if } f \in F^d, u \in V_p^{du} \\ n_s b_f x_{nu}^{sf} & , \text{ if } f \in F^c; \\ \sum_{k \in K(s)} \lambda_{f-1} b_k x_{nu}^{sf} & , \text{ otherwise.} \end{cases} \quad , \forall s \in S, \forall f \in F, \forall n \in N, \forall u \in V_p \quad (4)$$

4) *NFS Packing*: (5) represent the isolation constraints on the virtual layer. These constraints are responsible for applying different sharing policies imposed by each NS demand type. Constraints (6), in turn, ensure that a NF will not be present in more than one physical node.

$$x_{nu}^{sf} + x_{nu}^{tg} \leq 1 + q_{fg}^{st} q_{gf}^{ts} \quad , \forall s, t \in S, u \in V_p, n \in N, f, g \in F \quad (5)$$

$$x_{nu}^{sf} + x_{nv}^{tg} \leq 1, \forall s, t \in S, f, g \in F, n \in N, u, v \in V_p : v \neq u \quad (6)$$

$$\sum_{s \in S} w_{nu}^{sf} \leq y_{nu}^f \quad , \forall n \in N, \forall v \in V_p, \forall f \in F \quad (7)$$

Let us explain in detail the inequalities (7) with some examples. Suppose that NFSs of type  $f$  from  $s$  and  $t$  cannot be packed together ( $\forall n \in N, x_{nu}^{sf} \oplus x_{nu}^{tf}$ ). Hence, all copies of  $f$  installed on node  $u$  and serving  $s$  are not shared with  $t$ . In this way, if (4) set  $w_{mu}^{sf}$  to 4.60 and  $w_{nu}^{tf}$  to 1.25, for example, we must install at least seven ( $\lceil 4.60 \rceil + \lceil 1.25 \rceil$ ) NFSs of type  $f$  on the node  $u$  using two different NFs. Now, let  $s$  and  $t$  be two slices with no isolation constraints and using the same NF for a given NFS  $f$  ( $x_{nu}^{sf} \wedge x_{nu}^{tf}$ ). Suppose that (4) have set  $w_{bu}^{sf}$  and  $w_{nu}^{tf}$  equal to 4.60 and 1.25, respectively. Since both  $s$  and  $t$  accept NFS sharing with each other ( $q_{ff}^{st} \wedge q_{ff}^{ts}$ ), we need to install only six ( $\lceil 4.60 + 1.25 \rceil$ ) NFSs of type  $f$  on node  $u$  instead of seven of them. Using this approach on residual capacities, this saving can be even greater if we have a bigger sub-set of slices having  $q_{fg}^{st} = 1$  for a given tuple  $(s, t, f, g)$ .

5) *Physical node capacity*: (8) ensure that there will not be more installed NFs than a node can support.

$$\sum_{n \in N} \sum_{f \in F} c_f y_{nu}^f \leq c_u \quad , \forall u \in V_p \quad (8)$$

6) *Routing*: Constraints (9) represent the conservation flow constraints on control-plane traffic. Note that, since there can be only one virtual control-plane for each slice request,  $\gamma$  variables related to the set  $F(s)$  can be indexed to only one  $k$ ; we chose the first traffic demand to represent the whole control-plane on each slice. These constraints also represent the conservation flow constraints on the data-plane for each traffic demand  $k$ ; they provide a path between each pair NFSs from DP, from the origin node of each traffic demand  $k$  to the first related data-plane NFS, and between the last data-plane NFS and the target node for each traffic demand  $k$ . Note that we use the dummy function  $f_0$  in order to find a physical path between it and the data-plane chain if and only if its first NFS is installed centrally.

$$\begin{aligned} \sum_{a \in \delta^+(u)} \gamma_{fg}^{ka} - \sum_{a \in \delta^-(u)} \gamma_{fg}^{ka} = & \begin{cases} \sum_{n \in N} x_{nu}^{sf} - x_{nu}^{sg} & , \text{ if } (f, g) \in F(s), k = k_1 \\ \sum_{n \in N} x_{nu}^{sf} - x_{nu}^{sg} & , \text{ if } u \in V_p^{ac}, (f, g) \in G(s), \\ z_f^s - 1 & , \text{ if } (f, g) \in G(s), f \in F^c, u = o_k, \\ 1 - z_f^s & , \text{ if } (f, g) \in G(s), f \in F^d, u = o_k, \\ - \sum_{n \in N} x_{nu}^{sg} & , \text{ if } u \in V_p \setminus V_p^{du}, f = f_0, g = f_1 | g \in F^d \\ z_g^s & , \text{ if } u = o_k, f = f_0, g = f_1 | g \in F^d \\ 1 - z_f^s & , \text{ if } u = o_k, f = f_1 | f \in F^d, g = f_0 \\ -1 & , \text{ if } u = t_k, f = f_1 | f \in F^d, g = f_0 \\ \sum_{n \in N} x_{nu}^{sf} & , \text{ if } u \in V_p \setminus V_p^{du}, f = f_1 | f \in F^d, g = f_0 \\ \sum_{n \in N} x_{nu}^{sf} - x_{nu}^{sg} & , \text{ if } u \in V_p \setminus V_p^{du}, \forall f, g \in F^d | g = f + 1 \\ z_g^s - z_f^s & , \text{ if } u = o_k, \forall f, g \in F^d | g = f + 1 \\ 0 & , \text{ otherwise.} \end{cases} \\ \forall k \in K(s) : s \in S, \forall f, g \in F, \forall u \in V_p & \quad (9) \end{aligned}$$

*Latency*: Inequalities (10) ensure that the maximum end-to-end latency imposed by each slice request  $s$  will be respected on the path between  $o_k$  and  $t_k$  for every commodity  $k$ . Note that these technical constraints are applied only on the data-plane and only on the uplink direction as discussed before; we assume an arc has the same latency value  $d_a$  on both directions. Inequalities (11) ensure that the maximum latency between NFSs  $f$  and  $g$  will be respected on both data and control planes.

$$\sum_{a \in A_p} d_a (\gamma_{f_1 | F^d | f_0}^{ka} + \sum_{f \in \{f_0\} \cup F^d \setminus \{f_1 | F^d | \}} \gamma_{ff+1}^{ka}) \leq d_s \quad , \forall k \in K(s) : s \in S \quad (10)$$

$$\sum_{a \in A_p} d_a \gamma_{fg}^{ka} \leq d_{fg} \quad , \forall k \in K(s) : s \in S, \forall f, g \in F \quad (11)$$

*Physical link capacity* Inequalities (12) ensure that a arc will not carry more data than it can support. Note that comprehension coefficients are considered in these constraints.

$$\begin{aligned} & \sum_{s \in S} \sum_{k \in K(s)} b^k (\lambda_{f_{|F^d|}} \gamma_{f_{|F^d|} f_0}^{ka} + \sum_{f \in \{f_0\} \cup F^d \setminus \{f_{|F^d|}\}} \lambda_f \gamma_{ff+1}^{ka}) \\ & + \sum_{s \in S} n_s \left( \sum_{(f,g) \in F(s)} b_{fg} \gamma_{fg}^{ksa} + \sum_{(f,g) \in G(s)} \sum_{k \in K(s)} \frac{b_{fg} \gamma_{fg}^{ka}}{|K(s)|} \right) \leq b_a \\ & , \forall a \in A_p \quad (12) \end{aligned}$$

### C. Formulation

We minimize the total cost of deploying all network slice requests. To this end, the objective is to share as many NFSs as possible while respecting physical capacity constraints and assuring QoS imposed by each slice request. Being  $\Omega$  the scaling coefficient related to link utilization, the NSDP is then equivalent to the following formulation:

$$\min \sum_{f \in F} \sum_{n \in N} \sum_{u \in V_p} y_{nu}^f + \Omega \sum_{a \in A_p} \sum_{s \in S} \sum_{k \in K(s)} \sum_{f,g \in F} \gamma_{fg}^{ka} \quad (13)$$

s.t. (1)-(12) and

$$y_{nu}^f \geq 0 \quad \in \mathbb{Z} \quad , \forall f \in F, \forall n \in N, \forall u \in V_p \quad (14)$$

$$x_{nu}^{sf} \in \{0, 1\} \quad , \forall s \in S, \forall f \in F, \forall n \in N, \forall u \in V_p \quad (15)$$

$$z_f^s \in \{0, 1\} \quad , \forall s \in S, \forall f \in F \quad (16)$$

$$\gamma_{fg}^{ka} \in \{0, 1\} \quad , \forall k \in K(s) : s \in S, \forall f, g \in F \quad (17)$$

$$w_{uf}^s \geq 0 \quad \in \mathbb{R} \quad , \forall s \in S, \forall f \in F, \forall n \in N, \forall u \in V_p \quad (18)$$

While the first term in 13 is related to the number of installed functions, the second one refers to the number of active links. By simply changing the coefficient Omega (which multiplies one of the terms), slice providers can modify the objective function to a more suitable one (e.g., to emphasize the number of NFSs over the number of links in the optimization process)