Flow-Aware Service Function Embedding Algorithm in Programmable Data Plane

Jaewook Lee, Haneul Ko, *Member, IEEE*, Hochan Lee, and Sangheon Pack, *Senior Member, IEEE*

I. Flow-Aware Service Function Embedding (FASE) Algorithm

In this section, we propose the FASE algorithm to minimize the SFC completion time while providing efficient resource usage. We formulate an optimization problem to find the optimal embedding strategy. Subsequently, we provide the details of FASE algorithm. The important notations for the formulated problem are summarized in Table I.

A. *Optimization Problem*

As the SFC completion time is also highly affected by the incoming rates of SFC flows, we define the objective function that calculates the required number of re-circulations to process the incoming packets of all requested SFCs. The objective function can be expressed as $\sum_{c} I_c \sum_{k} R(Y, c, k)$, where I_c represents the incoming rate of SFC c, and the function $R(Y, c, k) = \sum_{n=1}^{N} \sum_{n'=n}^{N} y_{k-1,c,n'} y_{k,c,n}$ indicates whether a re-

TABLE I

SUMMARY OF NOTATIONS.

Notation	Description
N	The number of stages
S	The number of SF types
C	The number of the requested SFCs
r_s	The required number of stages to embed SF s
f_c	The number of SFs in SFC c
$t_{k,c}$	SF type of kth SF in SFC c
I_c	The incoming rates of SFC c
$x_{n,s}$	The decision variable to denote whether SF s is embedded from nth stage
$a_{n,s}$	The auxiliary decision variable to denote whether SF s is embedded at nth stage
$y_{k,c,n}$	The decision variable to denote whether kth SF in SFC c is allocated to the embedded SF in nth stage

circulation is required to process the kth SF in SFC c for the mapping result Y. The optimization problem can be formulated as follows.

$$\min \sum_{c=1}^{C} I_c \sum_{k=2}^{f_c} \sum_{n=1}^{N} \sum_{n'=n}^{N} y_{k-1,c,n'} y_{k,c,n} \tag{1}$$

We also define five constraints. First of all, since all SF types should be embedded at least once to support all SFC requests, the corresponding constraint is defined as equation (2), where $x_{s,n}$ is another decision variable to indicate the SF type s is embedded from nth stage (constraint 1).

$$\sum_{n=1}^{N} x_{s,n} \geq 1 \quad \forall s \tag{2}$$

Second, each stage, which is the resource of the PDP switch for embedding the match-action table, cannot be simultaneously used by more than one SF (constraint 2). Thus, the corresponding constraint is defined as (3) where $a_{s,n}$ is the auxiliary decision variable to denote whether SF s is embedded at nth stage.

$$\sum_{s=1}^{S} a_{s,n} \leq 1 \quad \forall n \tag{3}$$

Third, each SF should be consecutively embedded over the required number of stages (constraint 3). The corresponding constraint is defined as (4,5)

$$\sum_{n=1}^{N} a_{s,n} = r_s \sum_{n=1}^{N} x_{s,n} \quad \forall s \tag{4}$$

$$x_{s,n} \leq a_{s,n} \leq ... \leq a_{s,n+r_s-1} \quad \forall n \leq N - r_s + 1, s \tag{5}$$

Due to the limited stages of a PDP switch, the required number of stages used to embed the SFs cannot be more than the total number of available stages (constraint 4). The corresponding constraint is defined as (6).

$$\sum_{s=1}^{S} \sum_{n=1}^{N} a_{s,n} \leq N \tag{6}$$

Finally, each SF in the requested SFCs should be mapped to only one embedded SF (constraint 5). The corresponding constraint is defined as (7,8).

$$y_{k,c,n} \leq x_{t_{k,c},n} \quad \forall k, c, n \tag{7}$$

$$\sum_{n=1}^{N} y_{k,c,n} = 1 \quad \forall k, c \tag{8}$$

B. Description of FASE

The formulated optimization problem is an integer non-linear programming (INLP) with high complexity. We devised the FASE algorithm to find out a practical solution to the formulated problem. The FASE algorithm first determines the embedding order and mapping result without any redundant SF and re-circulation to minimize the SFC completion time under the available resources. At this step, all the SFC requests are sorted according to their incoming rates. Subsequently, the SFs in the sorted SFCs are included in the embedding order, and the SFCs are first mapped to the included SFs. Note that an SF can be included only when it does not exist in the embedding order. As all the SFs are embedded at once, all the requested SFCs can be supported. However, if several SFs are redundantly embedded, the SFC completion time can be further reduced. Thus, if some resources remain, several SFs are redundantly embedded in the FASE. To reduce the SFC completion time, it is intuitive to allow redundant SFs for the SFC requests with high incoming rates. Therefore, after sorting all the SFC requests according to their incoming rates, redundant SFs are embedded one-by-one until no resources remain in the PDP switch. When no more SFs can be embedded owing to limited resources, the embedding order is determined. In contrast, if not all SFs in the requested SFCs are not mapped to the embedded SFs, the remaining SFCs are supported by re-circulation.