Routing and Resource Allocation for Service Function Chain in Service-Oriented Network

Ziyu Liu(✉), Zeming Li, Chengchao Liang, and Zhanjun Liu

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
s190101097@stu.cqupt.edu.cn

Abstract. Service function chain (SFC) and in-network computing have become popular service provision approaches in 5G and next-generation communication networks. Since different service functions may change the volume of processed traffic in different ways, inappropriate resource allocation will lead to resource wastage and congestion. In this paper, we study the traffic change effects of service nodes with in-network computing and propose a software defined network based SFC routing and resource allocation scheme. With the objective of minimizing the difference between the service delay of adjacent function pairs in SFC and the corresponding expected delay, an optimization problem is established. Due to the coupling of computing resource provision and traffic engineering, and the non-convexity of the objective function and constraints, the problem becomes difficult to solve in practice. Therefore, we first transform the problem into a convex optimization problem using linear relaxation and variable substitution. Using the dual decomposition method, we decouple the different sets of variables. With this decoupling, the network controller can efficiently design the users' service nodes and traffic engineering. Finally, we use a rounding method to obtain a feasible solution set of the problem performed by the service nodes. Extensive simulations are performed under different system settings to verify the effectiveness of the scheme.

Keywords: Service function chain · In-network computing · Resource allocation · Traffic engineering

1 Introduction

With the large-scale application of novel service models (e.g., cloud computing, virtual reality, or the Internet of Things) and the stricter service quality requirements (e.g., ultra-high-definition video), the service-oriented network needs to have massive data processing capability and data transmission capability for

H. Gao et al. (Eds.): ChinaCom 2021, LNICST 433, pp. 465–480, 2022.
https://doi.org/10.1007/978-3-030-99200-2_35

large-scale connectivity scenarios [1]. To cope with the challenge of high coordination between computing and network requirements, the current key technologies such as service function chain (SFC), in-network computing, and software defined network (SDN) can be explored [2,3].

For network operators, proper interconnection of service functions is necessary to achieve complete end-to-end services [4]. For example, network functions required by the network protection service may include firewall, deep packet inspection, and virus scanning [5]. In practice, a service consisting of a set of network functions arranged in a predefined order is defined as SFC. Each network function is provided by some specific network nodes. To support different on-demand services, operators can use SFCs to direct traffic from different users through the required network nodes in a predefined order. In the service-oriented network, SFC can flexibly customize and rapidly allocate the network resources of operators [6]. Figure 1 shows an example of SFC with three network functions.

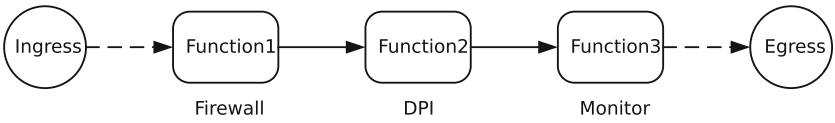

Fig. 1. The example of SFC.

In-network computing is a kind of communication acceleration technology based on the concept of collaborative design. Traditional networks typically deploy network functions at end hosts, while in-network computing enables the offload of network functions from end hosts to network nodes [7]. With in-network computing, the network nodes can provide online computing services during data transmission. Since the computing tasks are performed within the network, the efficiency of services can be significantly improved. Recently, with the advancement of programmable network devices, the in-network computing paradigm has received numerous attentions from researchers.

SDN enables the programmability of networks so that the complexity and the cost of networks can be reduced [8]. SDN decouples the control plane from the data plane, which enables more flexible management of network information. The network controller can efficiently select service nodes for users based on valid network information. The programming ability of SDN is also considered as a compelling candidate to enhance the performance of traffic engineering (TE), which is a critical component in the communication network [9]. In SDN-enabled network, TE is regarded as an effective tool to optimize the service path selection and resource allocation [10]. Recently, the authors in [11] integrated all functions required by a specific SFC on the same CPU with multiple cores to save transmission resource. In [12], the authors described the service node selection and routing problem under the condition of limited links and nodes capacities. Reference [13] considered the differentiated characteristics of SFC requests and proposed a heuristic algorithm. The network function deployment

and routing problem between adjacent network function pairs on the edge cloud was studied in [14]. The author proposed an approximate algorithm based on linear relaxation and random rounding.

The above work proposed some routing and resource allocation schemes for SFC scenarios. However, unlike classical network nodes that only forward data, network nodes with in-network computing may change the traffic volumes of the data being processed [15]. For example, the encoder used for satellite communication can increase the traffic by 31% due to the checksum, and the WAN optimizer may reduce the traffic by up to 80% before sending it to the next hop [16]. Figure 2 is an example of the traffic changing effects of network functions. Since different service functions may change the volume of processed traffic in different ways, the fixed resource allocation scheme will lead to inappropriate allocation of resource, e.g., wasted or insufficient resource. Further, the changed traffic will affect the service delay between adjacent function pairs in SFCs and create congestion at the service nodes. The authors in [16] proposed a service deployment scheme that considered the traffic changing effects to achieve optimal load balancing. In the SFC routing and resource allocation problem for meeting the end-to-end service requirements of different users, the traffic changing effects have not received sufficient attention.

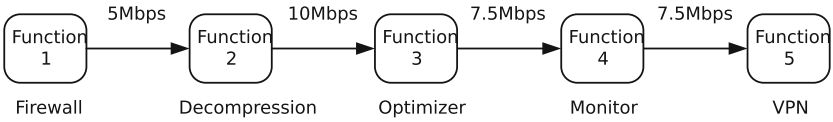

Fig. 2. The example of the traffic change effects.

Motivated by these challenges, we consider the end-to-end routing and resource allocation problem for different user requests with specific SFCs, where the required service functions have different capabilities to change the traffic volumes. The main contributions are shown as follows:

- Considering the capabilities of functions to change traffic volumes and the different user requests with specific SFCs, we establish an optimization problem. According to the ratio of traffic changes, we adjust the resources allocated to the traffic at the service node. The limited transmission resource of each link and the computing resource of each node are considered.
- Due to the coupling of TE and computing resource provision, and the non-convexity of the objective function and constraints, the problem becomes difficult to solve in practice. Therefore, we use linear relaxation and dual decomposition method to solve the problem. After obtaining the linear solution, we use a rounding method to obtain the feasible binary solution.
- Extensive simulations are conducted with different parameter settings to verify the performance of the proposed scheme.

The rest of this paper is organized as follows. In Sect. 2, we introduce the system model and formulate the problem. The transformed problem and the approximation algorithm are proposed in Sect. 3. Simulation results are discussed in Sect. 4. Finally, Sect. 5 concludes the paper.

2 System Model and Problem Formulation

2.1 Network Communication Model

As shown in Fig. 3, we consider a service-oriented wireless network that supports SFC. The considered communication network is modeled as a directed graph $\mathcal{G} = (\mathcal{I}, \mathcal{L})$. $\mathcal{I}$ is the nodes set, which includes the network nodes set $\mathcal{V}$ and the users set $\mathcal{K}$, namely, $\mathcal{I} = \mathcal{V} \cup \mathcal{K}$. Similarly, $\mathcal{L}$ is the set of directed links. It comprises the wired links set $\mathcal{L}^w$ and the wireless links set $\mathcal{L}^{wl}$ respectively.

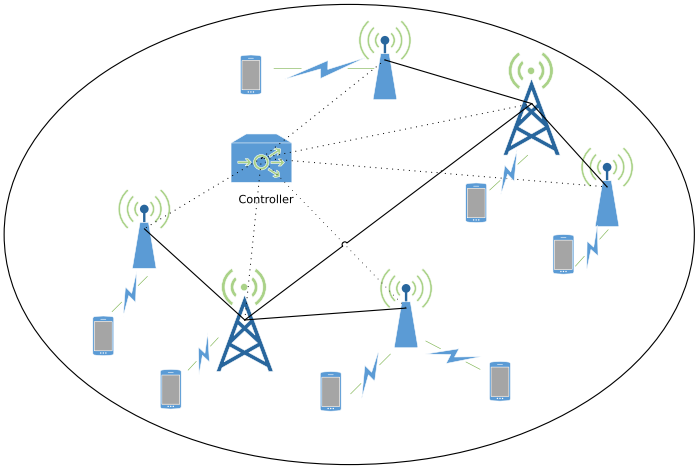

Fig. 3. The service-oriented wireless communication network.

If link (i, j) is a wired link, we assume that the link can provide a fixed available transmission capacity C_{ij}. If link (i, j) is a wireless link, the transmission capacity of the link depends on the available wireless resource of the link. The wireless resource is allocated by the network controller. By using Shannon's theorem, the spectral efficiency of wireless link (i, j) can be defined as follows:

$$\gamma_{ij} = log\left(1 + \frac{g_{ij}p_i}{\sigma_0 + \sum_{i' \in \mathcal{I}, i' \neq i} g_{i'j}p_{i'}}\right). \tag{1}$$

where g_{ij} is the channel gain between link (i, j) including large-scale path loss and shadowing, and p_i is the transmission power. To simplify the analysis, we

use the fixed equal power allocation mechanism to make the transmission power is identical for all wireless links. Each node treats the interference from any other transmission node $i^{'}$ as noise. The aggregated interference can be written as $\sum_{i' \in \mathcal{I}, i' \neq i} g_{i'j} p_{i'}$. Since the change of the small-scale fading is much faster than the transmission resource allocation, in this paper, we ignore the small-scale fading when evaluating the SINR [10]. Besides, σ_0 is the power spectrum density of additive white Gaussian noise. Therefore, the achievable data transmission rate of the wireless link (i, j) is $R_{ij} = W\gamma_{ij}$, where W is the total available wireless spectrum resource in the network.

2.2 SFC Service Model

The network supports multiple types of network functions. We use $\mathcal{V}_f \subset \mathcal{V}$ to express the subset of network nodes that can provide function f. We call the network nodes which can provide service functions are service nodes [12]. Each service node $i \in \mathcal{V}$ has a known computing capacity π_i. The service request of the user $k \in \mathcal{K}$ is described by a service function chain $\mathcal{F}(k)$, which consists of m functions that need to be processed in a given order, i.e., $\mathcal{F}(k) = (f_1^k \rightarrow f_2^k \rightarrow ... \rightarrow f_m^k)$.

For example, there are four network nodes and three users in Fig. 4. For simplicity, each network node only deploys one network function. The SFC of user 1 can be expressed as $\mathcal{F}(1) = (f_1^1 \rightarrow f_2^1)$, where f_1^1 and f_2^1 are $f2$ and $f4$ respectively. User 1 prefers to access the network from node B because function $f2$ is deployed on node B. However, there is no direct link between node B and node D. Then, node A or node C will act as a routing node, which needs to route

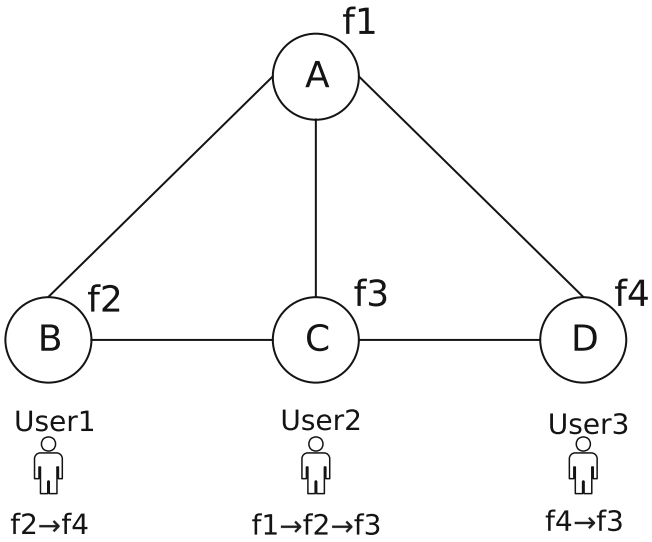

Fig. 4. An example of SFC requests in service-oriented network.

the traffic after processing $f2$ to node D. Similarly, before providing service to user 2, the network needs to route the traffic to node A first. The traffic of user 3 only needs to pass node D and node C in sequence.

After the data traffic traverses the service node and is processed by the corresponding function, the traffic rate may increase or decrease (e.g., video decompression and compression) [17]. The traffic rate of user k after receiving the function f_n^k can be expressed as $\delta^n(k)$. We denote a parameter α_n to express the traffic inflation factor of f_n^k. Therefore, we have $\delta^n(k) = \alpha_n \delta^{n-1}(k)$.

To avoid the coordination overhead caused by function splitting in practical application, we require each service function of service request k should be processed by one corresponding service node [12]. We use a binary parameter $h_i^n(k)$ to indicate a potential hitting event between service node i and the nth function of service request k. If node i can provide the nth function of service request k, $h_i^n(k) = 1$; otherwise, $h_i^n(k) = 0$. According to the user's request (SFC $\mathcal{F}(k)$) and the current network information (subset $\mathcal{V}_f$), the controller can quickly get the value of $h_i^n(k)$.

2.3 Problem Formulation

For the convenience of description, we add two dummy functions f_0^k and f_{m+1}^k to each service request k as the first and the last service function, respectively [14]. Both the two functions cannot cause any consumption of computing resource and computing delay. We define the service of the adjacent function pair (f_n^k, f_{n+1}^k) as the nth segment service of the service request k.

We use a binary variable $x_i^n(k)$ to define whether the network node i provides the nth service function for the user k. If node i provides the nth service function of $\mathcal{F}(k)$, $x_i^n(k) = 1$; otherwise, $x_i^n(k) = 0$.

We use another binary variable $y_{ij}^n(k)$ to describe the path selection of user k. If link (i, j) belongs to the nth segment service of the service request k, $y_{ij}^n(k) = 1$; otherwise, $y_{ij}^n(k) = 0$. We define $r_{ij}(k)$ as the data rate of user k over link (i, j), and we can conveniently get $r_{ij}(k) = \sum_n y_{ij}^n(k)\delta^n(k)$.

We define the service delay consumed by adjacent function pair (f_n^k, f_{n+1}^k) as $t^n(k)$, which equals to the sum of the transmission delay and the computing delay of the source node of the pair. The transmission delay $t_{tra}^n(k)$ of adjacent function pair (f_n^k, f_{n+1}^k) can be written as:

$$t_{tra}^n(k) = \sum_{(i,j)} \frac{d^n(k)}{\delta^n(k)} y_{ij}^n(k) = \sum_{(i,j)} t_h^n(k) y_{ij}^n(k). \tag{2}$$

where $t_h^n(k) = d^n(k)/\delta^n(k)$, and $d^n(k)$ is the task volume of the nth segment service of service request k.

Similarly, the computing delay $t_{com}^n(k)$ of adjacent function pair (f_n^k, f_{n+1}^k) is determined by the ratio of the current user demand and the corresponding computing rate $\eta(f_n^k)$ [18], which is defined as follows:

$$t_{com}^n(k) = \frac{d^{n-1}(k)}{\eta(f_n^k)}. \tag{3}$$

Therefore, the service delay of adjacent function pair can be expressed as:

$$t^n(k) = t^n_{com}(k) + \sum_{(i,j)} t^n_h(k) y^n_{ij}(k). \tag{4}$$

We define the expected service delay of adjacent function pair (f^k_n, f^k_{n+1}) as $t^n_{exp}(k)$. In order to minimize the absolute delay difference between the service delay $t^n(k)$ and the expected delay $t^n_{exp}(k)$ for all adjacent function pairs, the objective function can be written as:

$$\sum_{k\in\mathcal{K}} \sum_{n} |t^n(k) - t^n_{exp}(k)|. \tag{5}$$

In summary, the optimization problem can be expressed as:

$$\begin{aligned}
\mathbf{P1:} \min_{\mathbf{X},\mathbf{Y}} & \sum_{k\in\mathcal{K}} \sum_{n} |t^n(k) - t^n_{exp}(k)| \\
s.t. \quad & C1: x^n_i(k) \le h^n_i(k), \quad \forall i,k,n, \\
& C2: \sum_{i\in\mathcal{V}} x^n_i(k) = 1, \quad \forall k,n, \\
& C3: \sum_{(i,j)\in\mathcal{L}} y^n_{ij}(k) - \sum_{(j,i)\in\mathcal{L}} y^n_{ji}(k) = x^n_i(k) - x^{n+1}_i(k), \\
& C4: \sum_{n} \sum_{k\in\mathcal{K}} \eta(f^k_n) x^n_i(k) \le \pi_i, \quad \forall i, \\
& C5: \sum_{k\in\mathcal{K}} \sum_{n} y^n_{ij}(k)\delta^n(k) \le C_{ij}, \quad \forall (i,j) \in \mathcal{L}^w, \\
& C6: \sum_{(i,j)\in\mathcal{L}^{wl}} \frac{\sum_n y^n_{ij}(k)\delta^n(k)}{\gamma_{ij}} \le W, \\
& C7: x^n_i(k) \in \{0,1\}, y^n_{ij}(k) \in \{0,1\}.
\end{aligned} \tag{6}$$

The constraints C1 and C2 enforce that each service function must be served by exactly one service node in the network. The path selection constraint for service request k can be written as the constraint C3. This constraint is an essential condition for the successful construction of the routing path [19]. The constraint C4 enforces that the allocated computing resource of node i cannot exceed its available computing resource. The constraint C5 means the allocated data rate of wired link (i,j) for all users should be less than the link capacity. Similarly, the constraint C6 means the total allocated spectrum cannot exceed the available spectrum bandwidth. The constraint C7 means that $x^n_i(k)$ and $y^n_{ij}(k)$ are binary variables.

3 Solution to the Problem

Since $x^n_i(k)$ and $y^n_{ij}(k)$ are binary variables, and they are coupled in constraint, the problem becomes difficult to solve in practice. In this section, we first obtain

the fractional solution by linear relaxation and dual decomposition. After that, we use a rounding method to get a feasible solution of the problem.

3.1 Problem Decomposition

In order to overcome the obstacle of binary variables, we relax them into real value $[0, 1]$. Due to the non-convexity of the objective function, we transform it into a linear form. Firstly, we introduce a new variable $z^n(k)$ to replace the original objective function, namely, $z^n(k) = |t^n(k) - t^n_{exp}(k)|$. To limit the new variable, we introduce two new constraints, which are shown below:

$$\begin{aligned} z^n(k) &\geq t^n(k) - t^n_{exp}(k), \\ z^n(k) &\geq t^n_{exp}(k) - t^n(k). \end{aligned} \tag{7}$$

Finally, we get the convex optimization problem:

$$\begin{aligned} \mathbf{P2:} \min_{\mathbf{X},\mathbf{Y},\mathbf{Z}} & \sum_{k\in\mathcal{K}} \sum_{n} z^n(k) \\ s.t. \quad & C1: x^n_i(k) \leq h^n_i(k), \quad \forall i, k, n, \\ & C2: \sum_{i\in\mathcal{V}} x^n_i(k) = 1, \quad \forall k, n, \\ & C3: \sum_{(i,j)\in\mathcal{L}} y^n_{ij}(k) - \sum_{(j,i)\in\mathcal{L}} y^n_{ji}(k) = x^n_i(k) - x^{n+1}_i(k), \\ & C4: \sum_{n} \sum_{k\in\mathcal{K}} \eta(f^k_n) x^n_i(k) \leq \pi_i, \quad \forall i, \\ & C5: \sum_{k\in\mathcal{K}} \sum_{n} y^n_{ij}(k)\lambda^n(k) \leq C_{ij}, \quad \forall (i,j) \in \mathcal{L}^w, \\ & C6: \sum_{(i,j)\in\mathcal{L}^{wl}} \frac{\sum_n y^n_{ij}(k)\lambda^n(k)}{\gamma_{ij}} \leq W, \\ & C7: t^n_{com}(k) + \sum_{(i,j)\in\mathcal{L}} t^n_h(k) y^n_{ij}(k) - t^n_{exp}(k) \leq z^n(k), \\ & C8: t^n_{exp}(k) - t^n_{com}(k) - \sum_{(i,j)\in\mathcal{L}} t^n_h(k) y^n_{ij}(k) \leq z^n(k), \\ & C9: x^n_i(k) \in [0,1], y^n_{ij}(k) \in [0,1]. \end{aligned} \tag{8}$$

Although we transform the problem **P1** into the convex problem **P2**, the problem is still complex because the coupling of variables. To make the network more effective in routing and resource allocation, we compose the problem **P2** into three parts: service provisioning problem, traffic engineering problem, and delay optimization problem.

There are three coupling constraints C3, C7, and C8 in the relaxation problem. Therefore, by using dual variables $\{\lambda_{nk}\}$, $\{\mu_{ink}\}$ and $\{\nu_{nk}\}$, Lagrangian can be written as

$$\begin{aligned}
&\min_{\mathbf{X},\mathbf{Y},\mathbf{Z}} \sum_{k\in\mathcal{K}} \sum_{n} z^n(k) + \\
&\quad \sum_{i,n,k} \mu_{ink} \left[\sum_{(i,j)\in\mathcal{L}} y_{ij}^n(k) - \sum_{(j,i)\in\mathcal{L}} y_{ji}^n(k) - x_i^n(k) + x_i^{n+1}(k) \right] \\
&\quad + \sum_{n,k} \lambda_{nk} \left[t_{com}^n(k) + t_h^n(k) \sum_{(i,j)\in\mathcal{L}} y_{ij}^n(k) - t_{exp}^n(k) - z^n(k) \right] \\
&\quad + \sum_{n,k} \nu_{nk} \left[t_{exp}^n(k) - t_{com}^n(k) - t_h^n(k) \sum_{(i,j)\in\mathcal{L}} y_{ij}^n(k) - z^n(k) \right] \\
&s.t. \quad \mathbf{X} \in \Pi_x, \quad \mathbf{Y} \in \Pi_y, \quad \mathbf{Z} \in \Pi_z.
\end{aligned} \tag{9}$$

where Π_x, Π_y, and Π_z are independent local feasible sets of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, respectively. The relaxation problem is divided into two optimization levels: the dual variables updating and dual functions finding [20]. Naturally, the related dual problem (DP) is defined as:

$$\begin{aligned}
\mathbf{DP:} \max_{\lambda_{nk},\mu_{ink},\nu_{nk}\in\mathbb{R}} \quad & D(\lambda_{nk},\mu_{ink},\nu_{nk}) = g_x(\mu_{ink}) \\
& + g_y(\lambda_{nk},\mu_{ink},\nu_{nk}) + g_z(\lambda_{nk},\nu_{nk}) \\
s.t. \quad & \lambda_{nk} \geq 0 \qquad \forall k,n, \\
& \nu_{nk} \geq 0 \qquad \forall k,n.
\end{aligned} \tag{10}$$

where the three dual functions $g_x(\mu_{ink})$, $g_y(\lambda_{nk},\mu_{ink},\nu_{nk})$ and $g_z(\lambda_{nk},\nu_{nk})$ will be solved by the given dual variables $\{\lambda_{nk}\}$, $\{\mu_{ink}\}$ and $\{\nu_{nk}\}$ in the following three problems:

$$g_x(\mu) = \inf_{x\in\Pi_x} \left\{ \sum_{i,n,k} \mu \left[x_i^{n+1}(k) - x_i^n(k) \right] \right\}. \tag{11}$$

$$g_y(\lambda,\mu,\nu) = \inf_{y\in\Pi_y} \left\{ \begin{array}{c} \sum_{i,n,k} \mu \left[\sum_{(i,j)} y_{ij}^n(k) - \sum_{(j,i)} y_{ji}^n(k) \right] \\ + \sum_{n,k} t_h^n(k)(\lambda-\nu) \sum_{(i,j)} y_{ij}^n(k) \end{array} \right\}. \tag{12}$$

$$g_z(\lambda,\nu) = \inf_{z\in\Pi_z} \left\{ \begin{array}{c} \sum_{n,k} (1-\lambda-\nu) z^n(k) \\ + \sum_{n,k} (\lambda-\nu) \left[t_{com}^n(k) - t_{exp}^n(k) \right] \end{array} \right\}. \tag{13}$$

We can deploy sub-gradient method to solve (10). In each iteration, we first solve (11), (12) and (13) by the given dual variables. After obtaining the solutions of the three problems, we update the dual variables. Finally, we are able to obtain the optimal solution of the problem.

As for problem (11), it can be written as the following form:

$$\begin{aligned} \mathbf{P3:} \min_{x} \; & g_x(\mu) \\ s.t. \quad & C1: x_i^n(k) \le h_i^n(k), \quad \forall i,k,n, \\ & C2: \sum_{i\in\mathcal{V}} x_i^n(k) = 1, \quad \forall k,n, \\ & C3: \sum_{n}\sum_{k\in\mathcal{K}} \eta(f_n^k) x_i^n(k) \le \pi_i, \quad \forall i, \\ & C4: x_i^n(k) \in [0,1]\,. \end{aligned} \tag{14}$$

Similar to problem (11), the problem (12) as for the link selection variable $y_{ij}^n(k)$ is shown as:

$$\begin{aligned} \mathbf{P4:} \min_{y} \; & g_y(\lambda,\mu,\nu) \\ s.t. \quad & C1: \sum_{k\in\mathcal{K}}\sum_{n} y_{ij}^n(k)\lambda^n(k) \le C_{ij}, \quad \forall (i,j)\in\mathcal{L}^w, \\ & C2: \sum_{(i,j)\in\mathcal{L}^{wl}} \frac{\sum_n y_{ij}^n(k)\lambda^n(k)}{\gamma_{ij}} \le W, \\ & C3: y_{ij}^n(k) \in [0,1]\,. \end{aligned} \tag{15}$$

At last, the problem (13) is shown as:

$$\begin{aligned} \mathbf{P5:} \min_{z} \quad & g_z(\lambda,\nu) \\ s.t. \quad & z^n(k) \ge 0. \end{aligned} \tag{16}$$

The above three sub-problems are all linear programming (LP) problems with only one type of variable. The optimal global solutions of these sub-problems can be easily obtained in polynomial time. However, the solution of the problem **P2** may not be feasible for the original problem **P1** because the optimal solution of the LP problem may not be binary. The optimal solution of the LP problem is the lower bound of the solution of the original problem [12]. Next, we will use an effective rounding method to obtain the feasible solution.

3.2 Binary Recover

We combine some methods of existing work to construct our rounding strategy. Suppose that the fractional solution set obtained after iteratively solving the relaxation problem is $\{\tilde{x}_i^n(k)\}$ (and $\{\tilde{y}_{ij}^n(k)\}$). Our goal is to construct a binary solution set $\{x_i^n(k)\}$ based on the non-zero values in $\{\tilde{x}_i^n(k)\}$. However, due to the resource capacities couple the transmission of all service requests k, the system rounding is difficult [12]. We use a heuristic method to round the elements to 1 or 0.

In particular, if $\tilde{\mathbf{x}}^n(k)$ itself is binary, then we simply let $\mathbf{x}^n(k) = \tilde{\mathbf{x}}^n(k)$; if $\tilde{\mathbf{x}}^n(k)$ is not binary, we will round according to its value.

Firstly, we check the value of the largest element. If $\tilde{x}_j^n(k) = \max_{i \in V_f} \tilde{x}_i^n(k) \geq \theta$, where $\theta \in (0, 1)$ is the threshold we set, and node j has enough computing capacity, we adopt the following strategy:

$$x_j^n(k) = 1, \quad and \quad x_i^n(k) = 0, \quad \forall i \in \mathcal{V}_f \setminus \{j\}. \tag{17}$$

Otherwise, we give priority to the node $v \in V_f$ with the most computing capacity to provide the service functions for service request k. We set:

$$x_v^n(k) = 1, \quad and \quad x_i^n(k) = 0, \quad \forall i \in \mathcal{V}_f \setminus \{v\}. \tag{18}$$

After the rounding process, we get a binary solution set $\{x_i^n(k)\}$ while satisfying the constraint of the computing capacity. We use the rounded binary solution $\{x_i^n(k)\}$ to solve the problem **P2** again to find the solution under the condition of satisfying other constraints [14].

4 Simulation Results

We consider a service-oriented network with a range of $500\,m \times 500\,m$ in simulation. The network consists of one MBS and several SBSs. Users are scattered in the network, and we assume that the user can establish a connection with any BS within 200 m. The parameters of the network part are shown in Table 1.

Table 1. Network parameters settings

Notations	Definition
Frequency bandwidth	20 MHz
Transmission power	SISO with maximum power: 23 dBm
Pathloss	L(distance) = 34 + 40log(distance) (dB)
Lognormal shadowing	8 dB
Power density of the noise	−174 dBm/Hz
Number of SBSs	15
Wired link transmission capacity	[20, 30] Mbps
Computing capacity of SBS	[40, 50] Mbps
Computing capacity of MBS	75 Mbps

The network can provide five types of network functions, which are denoted as $\{f_1, f_2, ..., f_5\}$. Network functions can be placed on all nodes. In the simulation, each SBS randomly deploys one type of these functions, while MBS can deploy three functions. The network functions have traffic changing effects. Among them, The functions change traffic to $\{0.5, 0.75, 1, 1.25, 1.5\}$ times the

original, respectively. The computing capacity required by the five functions is set to $\{5, 7.5, 10, 7.5, 5\}$ Mbps, respectively.

We assume that each user demand comes with a service request for a specific SFC. The SFC $\mathcal{F}(k)$ with a length of m is generated by randomly selecting m unique functions from the function pool of the network and arranging them in order. Each data packet size sent by different users is randomly generated from [300, 500] KB, and the initial flow rate is randomly generated from [1, 4] Mbps. The expected delay of the user is equal to the service delay required by the user to complete the service by the shortest path, and the resource constraints are ignored in the calculation [21].

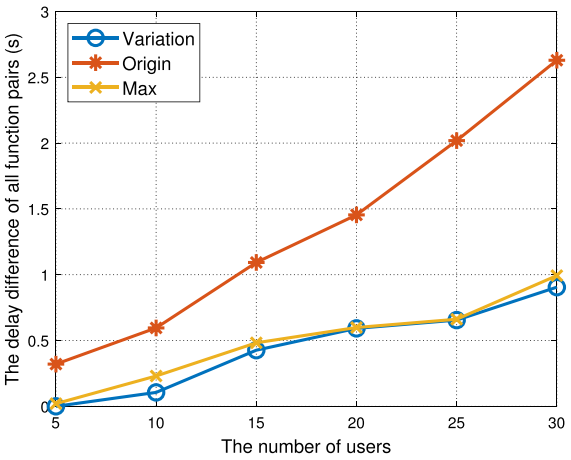

Fig. 5. The delay difference of all function pairs with different number of users.

In the simulation, we test the impact of the different number of users and the different SFC lengths on network performance. We compare the following three different schemes:

- Variation: Our proposed scheme. When traffic leaves the service node, we adjust the transmission resource allocated for traffic according to the corresponding traffic inflation factor.
- Origin: The network adopts the fixed transmission rate scheme, which equals the initial rate of traffic.
- Max: During the whole service process, the network controller reserves transmission resource according to the maximum transmission rate of traffic.

We first observe the change in the objective function of the optimization problem as the number of users increases. In Fig. 5, when the number of users increases, the absolute delay difference of all function pairs in the network also increases. The performance in delay difference of the Max scheme is approximately the same as that of our scheme. In most cases, our scheme is slightly

better than the Max scheme. Moreover, the delay difference of our scheme is much smaller than the Origin scheme. When the number of users is 30, the delay difference of the Origin scheme is even 2.6 times higher than the other two schemes. We consider that because the Origin scheme uses a fixed transmission rate, the delay will jitter dramatically in scenarios where the traffic volume changes. On the other hand, the Max scheme reserves enough transmission resource to maintain similar results as our scheme.

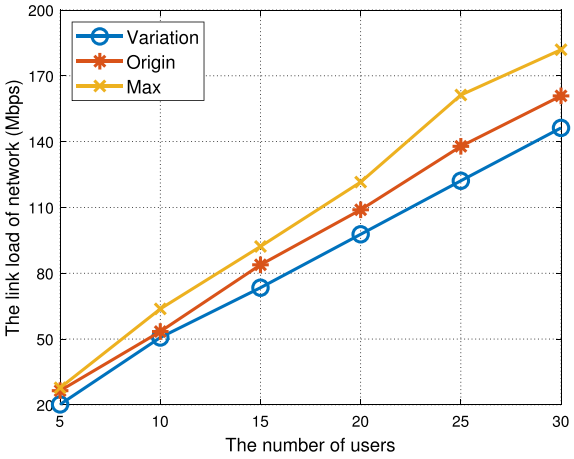

Fig. 6. The link load of network with different number of users.

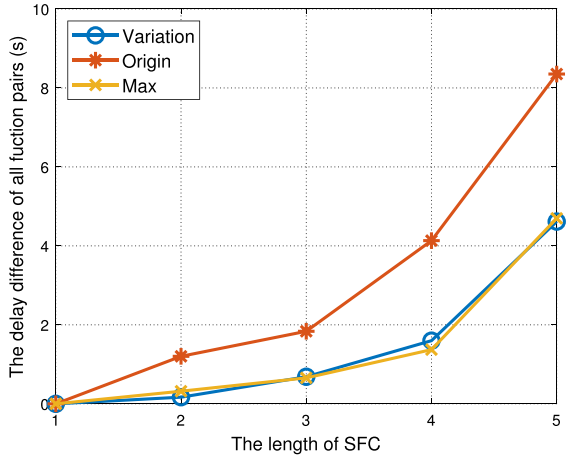

Fig. 7. The delay difference of all function pairs with different SFC lengths.

Figure 6 shows the transmission resource load of the network. It is worth noting that although network nodes with in-network computing can increase or

reduce user traffic, the total link load in our scheme is still smaller than the link load in the Origin scheme. The link load of the Max scheme is the largest of the three schemes. Compared with the Max scheme, in serving the same number of users with SFC requests, our scheme can save the link transmission resource consumption as much as possible.

With the fixed number of users, we change the length of the SFC requested by each user. Specifically, the increase of service chain length leads to the increasing complexity of traffic engineering, which affects the network delay difference performance. This result can be obtained from Fig. 7. The delay difference of the Origin scheme is still higher than the other two schemes. The delay difference of all function pairs of our scheme is about half of the Origin scheme.

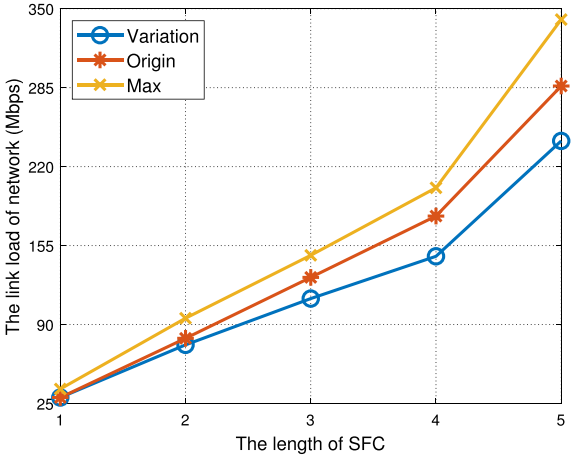

Fig. 8. The link load of network with different SFC lengths.

As shown in Fig. 8, our scheme consumes the least transmission resource in the case of the same number of processed service requests. The total link load of the Max scheme is higher than the other two schemes. Compared with the Max scheme, our scheme reduces transmission resource consumption by about 29%.

As a result, our scheme can achieve better network performance in both delay difference and network load. The Max scheme is similar to our scheme in delay difference performance, but it wastes more link resources. The Origin scheme has good link load performance in some cases, but its delay difference performance is poor. In brief, during the whole service process, properly adjusting the traffic rate is better than the mechanism for fixed allocation of transmission bandwidth.

5 Conclusions

This paper studied the routing and resource allocation of user data with SFC requests in the service-oriented network. Unlike previous work, this paper considered the traffic changing effects with in-network computing and established

an optimization problem to minimize the delay difference between the service delay and the expected delay for all adjacent function pairs. Due to the complexity of the problem, we used linear relaxation, dual decomposition, and rounding methods to solve the problem. The numerical results showed that our proposed scheme could achieve better results in both delay difference and link transmission load performance.

References

1. Pham, Q.V., et al.: A survey of multi-access edge computing in 5G and beyond: fundamentals, technology integration, and state-of-the-art. IEEE Access **8**, 116974–117017 (2020)
2. Tang, X., et al.: Computing power network: the architecture of convergence of computing and networking towards 6G requirement. China Commun. **18**(2), 175–185 (2021)
3. Qu, K., Zhuang, W., Ye, Q., Shen, X.S., Li, X., Rao, J.: Traffic engineering for service-oriented 5G networks with SDN-NFV integration. IEEE network **34**(4), 234–241 (2020)
4. Bhamare, D., Jain, R., Samaka, M., Erbad, A.: A survey on service function chaining. J. Netw. Comput. Appl. **75**, 138–155 (2016)
5. Mirjalily, G., Zhiquan, L.: Optimal network function virtualization and service function chaining: a survey. Chin. J. Electron. **27**(4), 704–717 (2018)
6. Halpern, J., Pignataro, C., et al.: Service Function Chaining (SFC) architecture. In: RFC 7665 (2015)
7. Mai, T., Yao, H., Guo, S., Liu, Y.: In-network computing powered mobile edge: toward high performance industrial IoT. IEEE network **35**(1), 289–295 (2020)
8. Liao, W.C., Hong, M., Farmanbar, H., Li, X., Luo, Z.Q., Zhang, H.: Min flow rate maximization for software defined radio access networks. IEEE J. Sel. Areas Commun. **32**(6), 1282–1294 (2014)
9. Mendiola, A., Astorga, J., Jacob, E., Higuero, M.: A survey on the contributions of software-defined networking to traffic engineering. IEEE Commun. Surv. Tutor. **19**(2), 918–953 (2016)
10. Liang, C., He, Y., Yu, F.R., Zhao, N.: Enhancing video rate adaptation with mobile edge computing and caching in software-defined mobile networks. IEEE Trans. Wireless Commun. **17**(10), 7013–7026 (2018)
11. Zheng, J., et al.: Optimizing NFV chain deployment in software-defined cellular core. IEEE J. Sel. Areas Commun. **38**(2), 248–262 (2019)
12. Zhang, N., Liu, Y.F., Farmanbar, H., Chang, T.H., Hong, M., Luo, Z.Q.: Network slicing for service-oriented networks under resource constraints. IEEE J. Sel. Areas Commun. **35**(11), 2512–2521 (2017)
13. Hong, P., Xue, K., Li, D., et al.: Resource aware routing for service function chains in SDN and NFV-enabled network. IEEE Trans. Serv. Comput. (2018)
14. Yang, S., Li, F., Trajanovski, S., Chen, X., Wang, Y., Fu, X.: Delay-aware virtual network function placement and routing in edge clouds. IEEE Trans. Mob. Comput. **20**, 445–459 (2019)
15. Jang, I., Suh, D., Pack, S., Dán, G.: Joint optimization of service function placement and flow distribution for service function chaining. IEEE J. Sel. Areas Commun. **35**(11), 2532–2541 (2017)

16. Ma, W., Sandoval, O., Beltran, J., Pan, D., Pissinou, N.: Traffic aware placement of interdependent NFV middleboxes. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)
17. Yang, S., Li, F., Trajanovski, S., Yahyapour, R., Fu, X.: Recent advances of resource allocation in network function virtualization. IEEE Trans. Parallel Distrib. Syst. **32**(2), 295–314 (2021)
18. Woldeyohannes, Y.T., Mohammadkhan, A., Ramakrishnan, K., Jiang, Y.: ClusPR: balancing multiple objectives at scale for NFV resource allocation. IEEE Trans. Netw. Serv. Manage. **15**(4), 1307–1321 (2018)
19. Wang, G., Zhou, S., Zhang, S., Niu, Z., Shen, X.: SFC-based service provisioning for reconfigurable space-air-ground integrated networks. IEEE J. Sel. Areas Commun. **38**(7), 1478–1489 (2020)
20. Palomar, D.P., Chiang, M.: A tutorial on decomposition methods for network utility maximization. IEEE J. Sel. Areas Commun. **24**(8), 1439–1451 (2006)
21. Chen, W.K., Liu, Y.F., De Domenico, A., Luo, Z.Q.: Network slicing for service-oriented networks with flexible routing and guaranteed E2E latency. In: 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–5. IEEE (2020)