EasyChair Preprint

№ 1723

# Service Function Chain Deployment Based on Improved Viterbi Algorithm in 5G-C-RAN

Kaiming Liu and Zhengbo Peng



October 20, 2019

# Service Function Chain Deployment Based on Improved Viterbi Algorithm in 5G-C-RAN

Kaiming Liu, Zhengbo Peng
Institution of Smart Wireless Information Technology, Beijing University of Posts and Telecommunications
{kmliu,pzb}@bupt.edu.cn

***Abstract*—This paper designs a service function chain deployment algorithm. The service function chain is under the 5G-C-RAN architecture. We build the combine profit model to maximize the utility for the virtual wireless access network. It optimizes the end-to-end delay limit and service rate requirements of mobile virtual network operator(MVNO) with resource Infrastructure Provider (InP) constraint. The Viterbi algorithm is modified to realize the deployment of the service function chain. Experiments show that the method has good performance in terms of service processing time, request acceptance rate, combine profit and algorithm execution time.**

***Keywords—C-RAN, NFV, SFC, deployment, Viterbi algorithm***

## I. INTRODUTION

Network Functions Virtualization (NFV) framework is an important technique in the fifth generation (5G) networks [1, 3-4]. And the Cloud-Radio access network (C-RAN) architecture with 5G (5G-C-RAN) is an important scenario application for NFV. It has flexible wireless resources. The NFV technology has transformed traditional operators into two roles: infrastructure provider (InP) and mobile virtual network operator (MVNO). InP provides VNFs and various physical resources for MVNO while MVNO operates a corresponding virtual network to provide network services. It promotes the efficient sharing of infrastructure resources, which greatly improve the cost efficiency of network construction. This can reduce the network's capital expenditure (CAPEX) and operating expense (OPEX), as well as increase infrastructure revenue [2]. The core goal of NFV is collecting information integration resources through management and orchestration functions, then deploying Virtualized Network Functions (VNFs) on general-purpose network devices by using an optimized deployment algorithm to form a service function chain (SFC).

Existing literature on SFC deployment problem modeled the problem as an integer linear programming (ILP) problem and proposed a series of precise and heuristic algorithms. [5-7] studied VNF embedding and deployment issues. In [8], joint topology design and SFC mapping in NFV was studied. In [9], considering the impact of different SFC deployment schemes on end-to-end delay, an SFC deployment strategy was designed with the goal of minimizing delay. Literature [10] considered NFV deployment issues into three aspects, namely SFC orchestration, VNF forwarding map mapping, and VNF scheduling. It modeled the optimization goals with minimal deployment costs.

Reference [17] proposed a heuristic algorithm aiming at minimizing the overhead of the whole network while the literature [18] proposed a network function virtualization orchestration model, and designed a corresponding greedy-based meta-heuristic algorithm. In [19], a heuristic algorithm based on simulated annealing was designed. It can find the approximate optimal solution in a short time with one type of VNF. In [20], a greedy-based minimum load (GLL) algorithm and a tabu search (TS)-based algorithm was designed. The former preferentially deployed VNF on the underlying node with the largest available remaining resources, and the latter constantly sought through taboo search. The optimal solution satisfies the condition, but the two algorithms do not fully consider the link state through which the traffic passes.

Reference [21] focused on deploying BBUs in a virtualized resource pool to the remote radio head (RRH) to minimize the delay between the RRH and the BBU and the settings of the server and the cost of pre/retransmission link (between the RRH and the BBU). Reference [22] studied the problem of virtual network function deployment in wireless access networks, and proposed a slice management mechanism for resource isolation in network slices. Reference [23] proposed a virtual node resource remapping algorithm that considers radio link interference. According to the remapping impact factor, when the service request ends, the changed physical resources are remapped to achieve load balancing.

One of the major defect of the algorithms proposed by the former references is that these algorithms occupy too much Inp computing resource. The impact of different SFC deployment scenarios on the wireless access network side is usually not considered.

This paper designs a service function chain deployment algorithm. It maximizes utility for the virtual wireless access network scenario under the 5G-C-RAN architecture. The main contribution of this paper is three-folds. First, the algorithm optimizes end-to-end delay limit and service rate requirements of MVNO. The constraint of optimizing goal contains InP Computing resources. Second, the utility model is built with the goal of maximizing the combine benefits of MVNO and InP generated by SFC deployment. Third, the hidden Markov model is used to describe available underlying network topology, and the improved Viterbi algorithm is used to deploy the service function chain. Simulation and testing results shows that the proposed algorithm has higher acceptance rate, higher combined profit, lower processing time and lower execute time than referred schemes.

The rest of this paper is organized as following. Section II gives a brief introduction of NFV architecture in 5G-C-RAN. In section III, the SFC deployment problem formulation is presented based on integral linear programming function. Section IV proposes SFC deployment algorithm. In Section V,

simulation and testing results have been given and and the paper concluded in Section VI.

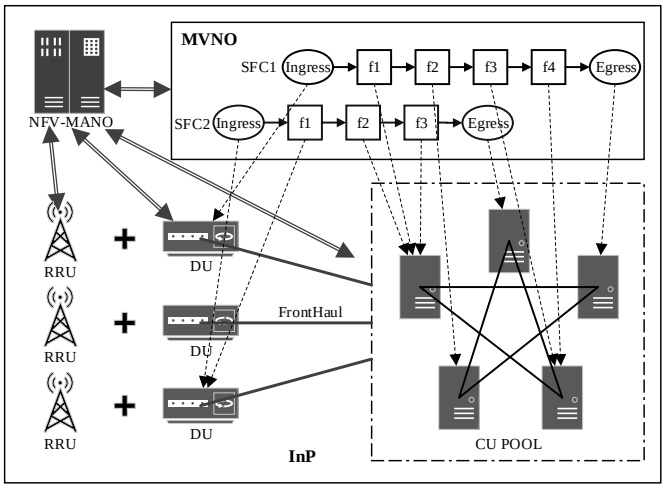


Fig. 1. 5G-C-RAN Virtualization Network Architecture

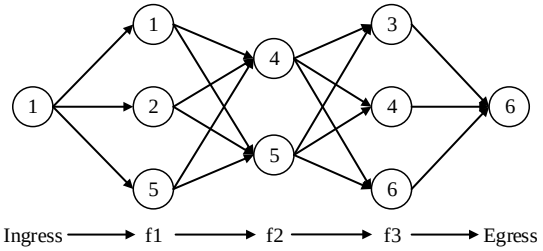


Fig. 2. Logical View of VNF Deployable Location

## II. NFV ARCHITECTURE IN 5G-C-RAN

In the NFV architecture, the virtualized wireless access network consists of an InP and a MVNO. InP has physical infrastructure and wireless resources, while MVNO leases resources from InP to create and operate virtual resources and distribute them to their users. InP, according to the SFC and network performance requirements requested by the MVNO, allocates spectrum resources to the MVNO through the network function virtualization management and orchestrator (NFV-MANO), provides and deploys the VNF, allocates CU and DU node computing resources, and terminates the FrontHaul link. The resources and the link resources between the CU nodes complete the deployment of the SFC.

### *A. Substrate Network*

The substrate network topology is represented by an undirected graph $G = (N, L)$, where $N$ denotes the set of nodes and $L$ denotes the set of links in the network. As mentioned earlier, the FrontHaul link is the link between the DU node and the CU node. Let $r_n \in \{0,1\}$ indicate whether node $n \in N$ is DU node.

$$r_n = \begin{cases} 1 & \text{if node } n \in N \text{ is DU node,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Let, $C_n$ and $C_n^f$ denote the capacity and available CPU resource of node $n \in N$, respectively, where $C_n^f \le C_n$. $B_{ij}$ and $B_{ij}^f$ denotes the capacity and available bandwidth resource of link between node $i$ and $j$, respectively, where $i, j \in N$ and $B_{ij}^f \le B_{ij}$. $D_{ij}$ denotes the delay of link between node $i$ and $j$, where $i, j \in N$.

### *B. Virtualized Network Functions (VNFs)*

The substrate network can provide VNFs of different protocol layer types, including physical layer (PHY), media access layer (MAC), radio link control layer (RLC), packet data convergence protocol layer (PDCP). Let $F = \{f_p \mid p = 1,2,3,\ldots P\}$ represent the set of VNFs, where $p = tp(f_p)$ denotes the type of VNF. It should be noted that each node does not necessarily provide all types of VNFs. So, we assume that each VNF type has a set of nodes on which to deploy. The binary variable $\gamma_{np}$ indicates whether node $n \in N$ can deploy VNF $f_p \in F$.

$$\gamma_{np} = \begin{cases} 1 & \text{if node } n \in N \text{ can deploy VNF } f_p \in F, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Running VNF requires resources on the general-purpose processor nodes, such as CPU, memory, disk. The resource requirements are usually related to the amount of data that the VNF needs to process.

The SFC requests consists of multiple VNFs, Ingress and Egress nodes with sequential constraints. Let $S = \{s_k \mid k = 1,2,3,\ldots K\}$ represent the set of SFC requests, $s_k = < i_k, e_k, \varphi_k, v_k, d_k, \tau_k >$ indicates a SFC request, where $i_k, e_k \in N$ denote the ingress and egress node, respectively. $v_k$ $d_k$ and $\tau_k$ denote the maximum rate demand, delay limit and life cycle of the SFC. When the SFC service time exceeds the life cycle, the service ends and the allocated resources are reclaimed. $\varphi_k = \{ f_m^k \mid m = 1,2,3,\ldots M^k\}$ represents ordered VNF sequence of SFC, where $f_m^k \in F$.

Finally, we define a binary variable $\eta_m^k \in \{0,1\}$ to indicate whether VNF $f_m^k \in \varphi_k$ deployed on node $n \in N$.

$$\eta_{mn}^k = \begin{cases} 1 & \text{if node } f_m^k \in \varphi_k \text{ deployed on node } n \in N, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

### *C. SFC Deployment Problem*

In the SFC deployment problem, we assume that MVNO deploys the SFC requests one by one according to the arrival time. The MVNO needs to find a suitable mapping location for each VNF and its link in the substrate network given a set of service function chain requests, and needs to meet the corresponding optimization objective.

## III. INTEGER LINEAR PROGRAMMING (ILP) FORMULATION

In this paper, our objective is to maximize the combined profits of MVNO and InP generated by SFC deployment. Among them, InP's revenue is mainly VNF deployment revenue, and expenditure mainly includes the cost of node computing resources and link bandwidth resources. And the revenue of MVNO mainly comes from its service rate, and the expenditure comes from the QoS loss of MVNO.

### *A. InP revenue and expenditure model*

In the business model of this paper, the revenue of InP comes

from the lease of VNF, and the cost comes from the processor node computing resources and inter-node link resources that are required to deploy the VNF. The deployment of each VNF module in the SFC will bring CAPEX, including VNF development, maintenance, and copyright fees, which are borne by the MVNO and are one of the revenue components of the InP. This paper refers to the discussion of the function splitting of CU and DU in the logical architecture of the next generation radio access network in 3GPP. Considering the different protocol layer splitting schemes, the cost and revunue of deploying VNF in the SFC will be different. The revunue of VNF deployment generated by deploying a SFC can be expressed as

$$P_{vnf} = \sum_{s_k \in S} \sum_{f_m \in s_k} \sum_{n \in N} (1-r_n)\eta_{mn}^k CUP_{f_m} + r_n \eta_{mn}^k DUP_{f_m}, \quad (4)$$

where, *CUP* and *DUP* represent the revenue generated by VNF $f_m$ deployed on the CU and DU, respectively.

The cost of deploying the computing resources occupied by the VNF on the processor node is expressed as

$$E_{cmp} = \sum_{s_k \in S} \sum_{f_m \in s_k} \sum_{n \in N} \eta_{mn}^k C_m^k \delta_c, \quad (5)$$

where, $\delta_c$ represents the unit price of the computing resource on processor node.

In the model constructed in this paper, the demand for link resources between nodes is equal to the rate of data streams transmitted on the link [16]. Since the FrontHaul link resource is relatively precious in the C-RAN architecture, in order to reflect the difference between the FrontHaul link and the CU cluster node in the C-RAN architecture, the two links are separately charged and billed separately. The link bandwidth cost in the CU cluster is expressed as

$$E_{cu} = \sum_{s_k \in S} \sum_{f_m \in s_k} \sum_{i,j \in N} (1-r_j)\eta_{m,i}^k \eta_{m+1,j}^k h_{i,j} V_{m+1}^k \delta_b, \quad (6)$$

where, $\delta_b$ denotes the unit price of the link bandwidth in the CU cluster, $h_{ij}$ denotes the hop count between the processor node $i$ and $j$, that is, the number of links passing through.

The cost of fronthaul link resources brought by deploying SFC is expressed as

$$E_{fh} = \sum_{s_k \in S} V_{x+1}^k \delta_{fh}, \quad (7)$$

where, $\delta_{fh}$ represents the unit price of the FrontHaul link. $x$ denotes the number of VNF modules deployed on the DU node. The $V_{x+1}^k$ denotes the bandwidth required for the SFC $s_k$ on the FrontHaul. The calculation of $x$ is

$$x = \sum_{f_m \in s_k} \sum_{n \in N} \eta_{mn}^k r_n. \quad (8)$$

In summary, the total profit of the infrastructure operator can be expressed as

$$P_{InP} = P_{vnf} - E_{cmp} - E_{fh} - E_{cu}. \quad (9)$$

### B. MVNO revenue and expenditure model

The revenue of the MVNO comes from its service rate, and the cost comes from the QoS loss. Service rate revenue is expressed as

$$P_v^k = v_k \delta_v, \quad (10)$$

where $\delta_v$ denotes the service rate unit price of MVNO.

Different deployment results of SFC will bring different end-to-end delays, which will affect the service quality of virtual operators. Therefore, the QoS loss generated by SFC end-to-end delay is used as the expenditure of virtual operation. The end-to-end delay of an SFC can be expressed by the propagation delay and processing delay as

$$D^k = \sum_{f_m \in s_k} \sum_{i,j \in N} \eta_{m,i}^k \eta_{m+1,j}^k h_{i,j} D_{i,j} + \sum_{f_m \in s_k} D_{f_m}. \quad (11)$$

The QoS loss caused by the end-to-end delay can be expressed as

$$E_{QoS}^k = D^k \delta_d, \quad (12)$$

where, $\delta_d$ represents the unit price of the end-to-end delay. In summary, the total profit of MVNO can be expressed as

$$P_{MVNO} = \sum_{s_k \in S} P_V^k - E_{QoS}^k. \quad (13)$$

### C. Optimize Objective

Our objective is to maximize the combined profits of MVNO and InP generated by SFC deployment that can be expressed as a weighted sum of the two profits mentioned.

$$\begin{aligned} &\arg\max_{\eta} \left(\omega_0 P_{InP} + \omega_1 P_{MVNO}\right) \\ &\textit{subject to} \\ &C_1 : D^k \le d_k \quad \forall s_k \in S, i \ne j \\ &C_2 : \sum_{f_m \in s_k} \sum_{n \in N} \eta_{mn}^k = 1 \quad \forall s_k \in S \\ &C_3 : \sum_{s_k \in S} \sum_{f_m \in s_k} \alpha_m \eta_{mi}^k V_m^k \le C_n^f \quad \forall i \in N \\ &C_4 : \sum_{s_k \in S} \sum_{f_m \in s_k} \sum_{i,j \in N} \eta_{m,i}^k \eta_{m+1,j}^k V_m^k \le B_{i,j}^f \quad \forall i \ne j \end{aligned} \quad (14)$$

Here, the two weighting coefficients $\omega_0$ and $\omega_1$ satisfy $\omega_0+\omega_1=1$. Constraint $C_1$ indicates that the end-to-end delay limit of the SFC needs to be met during deployment. Constraint $C_2$ denotes that the VNFs in each SFC are mapped to a unique node. Constraint $C_3$ indicates that the total amount of computing resources occupied by the VNF on any node is less than the total amount of computing resources of the node. Constraint $C_4$ denotes that the total amount of link bandwidth occupied by the data stream on the link is less than the total bandwidth of the link.

In this paper, in order to simplify the complexity of the model, the various resources on the node are unified into computing resources, and define the amount of computing resources required by VNF to be linear with the data rate that VNF needs to process. Let $\alpha_p$ denote the correlation coefficient between the VNF computing resource requirement and the data processing rate. That is, the computing resource requirement of VNF is expressed as $C_p = \alpha_p V_p$, where $V_p$ denotes the data processing rate of VNF. In addition, this model considers the processing of the uplink protocol function in wireless communication. Therefore, when data processing of the virtual network is performed, the rate of the data stream may change after the data stream is processed by each VNF [1], for example, in the uplink of the LTE protocol. In the protocol function processing, after the data stream is processed by the PHY layer and then to the MAC layer, the data stream rate is greatly reduced. Therefore,

in the C-RAN architecture, the pressure of the FrontHaul link can be reduced by deploying part of the protocol function on the RRU side [15]. This paper takes this feature into account and sets the data rate expansion rate $\beta_p$ for each VNF.

## IV. Algorithm Description

In the SFC deployment problem, the VNF component of the SFC and the location where the VNF can be deployed are observable, but the specific service path of the SFC is unobservable. Therefore, the SFC deployment problem has a hidden Markov property, and the service path is a hidden Markov chain.

At time $t$, we use the hidden Markov model to describe the logical view of the deployable locations of all VNFs in the substrate network based on the SFC request and the state of the network, as shown in Fig. 2. In a directed graph network, the direction of the arrow indicates the VNF order in the SFC request. Except for the Ingress and Egress, each column in the directed graph network represents a type of VNF, and the different columns in the same column represent different locations at which the type VNF can be deployed at time $t$. The Viterbi algorithm can be used to find hidden state sequences of observed events for Hidden Markov Models.

We assume that the m-layer node set is $N_m$, that is, the VNF $f_m$ deployable node set. Let $P_t[f_m(n_a)]$ denotes the profit of $f_m$ deployed at node $n_a$ at time $t$, $P_t[f_{m+1}(n_b)]$ denotes the profit of $f_{m+1}$ deployed at node $n_b$ at time $t$, and $P_t[f_{m,m+1}(n_a, n_b)]$ denotes the profit when $f_m$ and $f_{m+1}$ are deployed at nodes $n_a$ and $n_b$ respectively at time $t$, where $n_a \in N_m$, $n_b \in N_{m+1}$. The Viterbi algorithm recursively calculates the maximum cumulative profit $P_m(n_a)$ for all $n_a$ in $N_m$, from Ingress until the maximum cumulative profit of the Egress is obtained, which is the cumulative profit of the optimal path. Finally, the optimal path can be obtained by backtracking from the Egress node. Here, $P_{m+1}(n_b)$ can be expressed as

$$P_{m+1}(n_b) = \max_{n_a \in N_m} \left\{ P_m(n_a) + P_t\left[ f_{m \to m+1}(n_a, n_b) \right] + P_t\left[ f_{m+1}(n_b) \right] \right\}. \quad (15)$$

However, in practical applications, the amount of computation required by the algorithm is still large. We improve the algorithm and cut down those impossible paths.We use the cumulative profit as the score of the path. After each NVF deployment, we find the path with the highest and lowest score whitch can be expressed as

$$P_m(n_{\max}) = \max_{n_a \in N_m} \left[ P_m(n_a) \right] \quad (16)$$

$$P_m(n_{\min}) = \min_{n_a \in N_m} \left[ P_m(n_a) \right]. \quad (17)$$

If other paths score much less than $P_m(n_{max})$, then the probability of becoming the best candidate path in the next layer is very low. Therefore, it can be assumed that any path with ascore lower than $P_{m\text{-}threshold}$ can be cut off in advance. We define $P_{m\text{-}threshold}$ as

$$P_{m-shreshold} = P_m(n_{\min}) + q\left[ P_m(n_{\max}) - P_m(n_{\min}) \right]. \quad (18)$$

**Algorithm 1** SFC deployment

**Input:** Network topology $G = (N, L)$, SFC request $s_k$=<$i_k$, $e_k$, $\varphi_k$, $v_k$, $d_k$, $\tau_k$>.
**Output:** SFC deployment results $\eta$.

Initialize $P_0(i_k)$, $\pi_{1,n}$ , $n \in N_1$
**for** all $f_m \in \varphi_k$ **do**
  $p_m = tp(f_m)$, $N_m = \varnothing$
  **for** all $n \in N$ **do**
    **if** $\gamma_{np}$==1 **and** $C_{p_m} \le C_n^f$ **then**
      $N_m = N_m + n$
  **end for**
  **for** $n_a \in N_{m-1}$, $n_b \in N_m$ **do**
    **if** $B_m \le B_{n_a,n_b}^f$ **then**
      $P_m(n_b) = \max\left\{ P_m(n_b), P_{m-1}(n_a) + P_t\left[ f_{m-1 \to m}(n_a, n_b) \right] + P_t\left[ f_m(n_b) \right] \right\}$
      $\pi_{m,n_b} = \arg P_m(n_b)$
  **end for**
  $P_{m-shreshold} = P_m(n_{\min}) + q\left[ P_m(n_{\max}) - P_m(n_{\min}) \right]$
  **for** all $n_b \in N_m$ **do**
    **if** $P_m(n_b) < P_{m\text{-}shreshold}$ **then**
      $N_m = N_m - n_b$
  **end for**
**end for**
$P_{M^k+1}(o_k) = \max_{n_a \in N_M} \left\{ P_{M^k}(n_a) + P_t\left[ f_{M^k \to M^k+1}(n_a, o_k) \right] \right\}$
**if** $D^k \le d_k$ **then**
  Deployed successfully,return deployment results $\eta$.
  Update network topology $G = (N, L)$.
**while**(1)
  **if** $T > t_0 + \tau$ **then**
    End the service, reclaim resources allocated to the service.
    Update network topology $G = (N, L)$.
**end while**

Here $q$ is related to the utilization of resources, which can be expressed as

$$q = \frac{1}{2}\left[ \operatorname{avg}_{n \in N}\left( C_n^f / C_n \right) + \operatorname{avg}_{l \in L}\left( B_l^f / B_l \right) \right] q_0, \quad (19)$$

$q_0$ is a predefined constant, we set $q_0$=0.6. The VNF deployment algorithm based on the improved Viterbi algorithm is shown in Algorithm 1. We define the variable $\pi_{m,n}$ to record the pre-VNF deployment node in the optimal path when $f_m$ is deployed at node $n$.

## V. Evaluation

In order to evaluate our models and algorithms with respect to service request processing time, acceptance rate, total profit and average profit cost ratio, experimental test is presented for NFV-MVNO system. Three other algorithms, namely, DP, TS and GLL [20,24] are also included in this work for comparison.

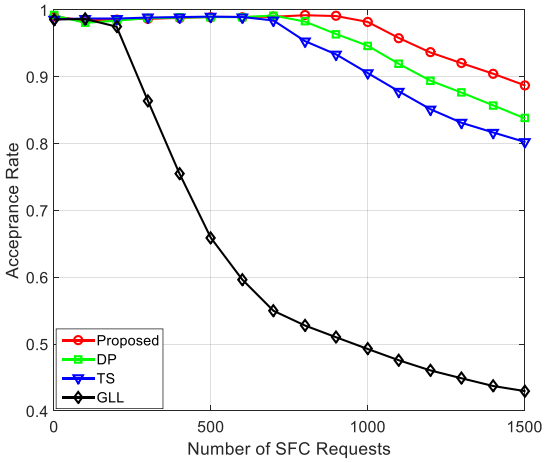


Fig. 3. Service Acceptance Ratio

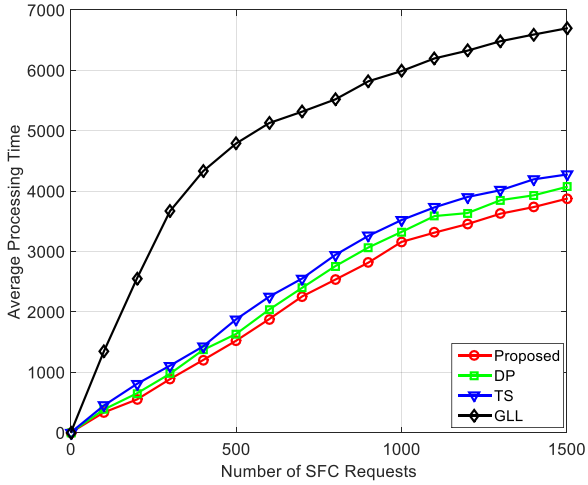


Fig. 4. Request Averaging Processing Time

We implemented a discrete event simulator in Python with parameters as shown in Tables I and II. Each pair of CU nodes generates a physical link with equal probability, and the DU nodes randomly connect the CU nodes to generate a FrontHaul link.

TABLE I
SETTING OF SUBSTRATE NETWORK PARAMETER

| Parameter | Minmum | Maximum |
|---|---|---|
| Number of DU nodes | 15 | 15 |
| Number of CU nodes | 35 | 35 |
| Node CPU capacity | 15 | 30 |
| FrontHaul link bandwidth capacity | 150 | 150 |
| CU link bandwidth capacity | 50 | 100 |
| FrontHaul link latency | 1 | 1 |
| CU link latency | 1 | 3 |
| Function processed by each node | 2 | 4 |
| Unit price of the CPU resource | 6 | 6 |
| Unit price of the FrontHaul link | 4 | 4 |
| Unit price of the CU link | 2 | 2 |

In these evaluations, we assume that the SFC request satisfies the Poisson distribution and arrives on average once every 5 seconds.

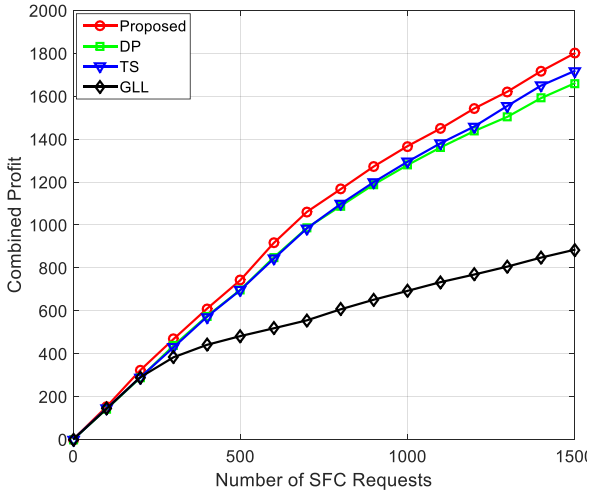


Fig. 5. Combined Profit

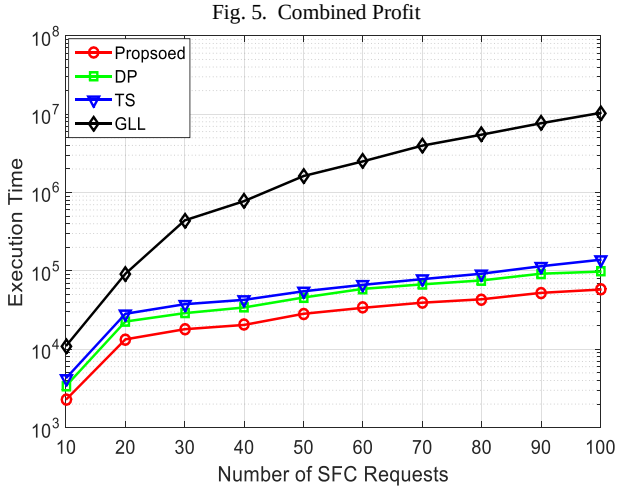


Fig. 6. Algorithm Execution Time

TABLE II
SETTING OF VNF AND SFC PARAMETER

| Parameter | Minmum | Maximum |
|---|---|---|
| Number of VNF types | 10 | 10 |
| Revenue of each VNF in the DU | 80 | 80 |
| Revenue of each VNF in the CU | 60 | 60 |
| Function processing times | 5 | 10 |
| CPU demand coefficient $\alpha_m$ | 0.1 | 0.2 |
| Data rate expansion rate $\beta_m$ | 0.5 | 1.0 |
| Number of VNFs in SFC | 3 | 6 |
| Maximum rate demand | 10 | 20 |
| End-to-end delay limit | 10 | 20 |
| Life cycle | 4000 | 8000 |
| Unit price of service rate | 50 | 50 |
| Unit price of end-to-end delay | 10 | 10 |

Fig. 3 shows the service request acceptance rate with respect to the number of SFC requests. As shown in Fig. 3, as the number of requests increases, the acceptance rate of the algorithm decreases. With the improved Viterbi algorithm being used, the proposed algorithm has the highest acceptance rate. For example, when the number of SFC requests is 1000, the Acceptance rate is 0.98, which is twice as GLL algorithm.

Fig. 4 shows the service request averaging processing time with respect to the number of SFC requests. As the number of

requests increases, the average processing time of the algorithm is increasing. The proposed algorithm has the lowest average processing time. For example, when the number of SFC requests is 1000, the average processing time is 3000, which is half of GLL algorithm.

Fig. 5 shows the combined profit with respect to the number of SFC requests. As the number of requests increases, we can find that the combined profit of proposed algorithm increases. For example, when the number of SFC requests is 1000, the combined profit is 1400, which is twice than the GLL algorithm.

Fig. 6 shows the execution time of the algorithm with respect to the number of SFC requests. As shown in Fig. 6, when the number of SFC requests increases, the execution time increases. And the proposed algorithm has the lowest execution time. For example, when the number of SFC requests is 1000, the execution time is half of GLL algorithm.

## VI. Conclusion

This paper has designed a service function chain deployment. The service function chain is under the 5G-C-RAN architecture. It maximizes utility for the virtual wireless access network scenario under the 5G-C-RAN architecture. the algorithm optimizes end-to-end delay limit and service rate requirements of MVNO. The constraint of optimizing goal contains InP. Computing resources. The constraint of optimizing goal contains InP. Computing resources. the utility model is built with the goal of maximizing the combine benefits of MVNO and InP generated by SFC deployment. The hidden Markov model is used to describe available underlying network topology, and the improved Viterbi algorithm is used to deploy the service function chain.

Experiment tests show that with the same number of SFC requests, the proposed algorithm can achieve the highest combined profit, the highest acceptance rate, the lowest average processing time and the lowest execution time.

## References


[1] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," *IEEE Access*, vol. 3, pp. 1206-1232, 2015.

[2] Hawilo H, Shami A, Mirahmadi M et al. NFV:State of the art, challenges, and implementation in next generation mobile networks (vEPC). IEEE Network, vol. 6, pp. 18-26, 2014.

[3] Chih-Lin I, Huang J, Yuan Y, et al, "5G RAN Architecture: C-RAN with NGFI. 5G Mobile Communications", Springer International Publishing, 2017.

[4] 3GPP V14.0.0. Study on new radio access technology: Radio access architecture and interfaces (Release 14). TR 38.301.Mar.2017

[5] J. Liu, Y. Li, Y. Zhang, L. Su and D. Jin, "Improve Service Chaining Performance with Optimized Middlebox Placement," *IEEE Transactions on Services Computing*, vol. 10,pp. 560-573, 2017.

[6] S. Herker, X. An, W. Kiess, S. Beker and A. Kirstaedter, "Data-Center Architecture Impacts on Virtualized Network Functions Service Chain Embedding with High Availability Requirements," *2015 IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, 2015, pp. 1-7.

[7] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions,"*2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, ON, 2015, pp. 98-106.

[8] Z. Ye, X. Cao, J. Wang, H. Yu and C. Qiao, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Network*, vol. 30, pp. 81-87, 2016.

[9] L. Qu, C. Assi and K. Shaban, "Delay-Aware Scheduling and Resource Optimization With Network Function Virtualization," *IEEE Transactions on Communications*, vol. 64, pp. 3746-3758, 2016.

[10] L. Wang, Z. Lu, X. Wen, R. Knopp and R. Gupta, "Joint Optimization of Service Function Chaining and Resource Allocation in Network Function Virtualization," *IEEE Access*, vol. 4, pp. 8084-8094, 2016.

[11] Y. Li, F. Zheng, M. Chen and D. Jin, "A unified control and optimization framework for dynamical service chaining in software-defined NFV system," in, *IEEE Wireless Communications*, vol. 22, pp. 15-23, 2015.

[12] BASTA A, HOFFMANN K, HOFFMANN K, et al. "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem", The Workshop on All Things Cellular: Operations, Aug 17-22, 2014, Chicago, IL, USA. New York: ACM Press, 2014: 33-38.

[13] F. Ben Jemaa, G. Pujolle and M. Pariente, "QoS-Aware VNF Placement Optimization in Edge-Central Carrier Cloud Architecture,"*2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, 2016, pp. 1-7.

[14] A. Baumgartner, V. S. Reddy and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization,"*Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, 2015, pp. 1-9.

[15] A. Baumgartner, V. S. Reddy and T. Bauschert, "Combined Virtual Mobile Core Network Function Placement and Topology Optimization with Latency Bounds,"*2015 Fourth European Workshop on Software Defined Networks*, Bilbao, 2015, pp. 97-102.

[16] S. Mariia and S. Svitlana, "Service deployment aspects in the systems with network function virtualization,"*2016 International Conference Radio Electronics & Info Communications (UkrMiCo)*, Kiev, 2016, pp. 1-7.

[17] R. Cohen, L. Lewin-Eytan, J. S. Naor and D. Raz, "Near optimal placement of virtual network functions," *2015 IEEE Conference on Computer Communications (INFOCOM)*, Kowloon, 2015, pp. 1346-1354.

[18] B. Addis, D. Belabed, M. Bouet and S. Secci, "Virtual network functions placement and routing optimization,"*2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, Niagara Falls, ON, 2015, pp. 171-177.

[19] X. Li and C. Qian, "The virtual network function placement problem,"*2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Hong Kong, 2015, pp. 69-70.

[20] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions,"*Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, 2015, pp. 1-9.

[21] R. Mijumbi, J. Serrat, J. Gorricho, J. Rubio-Loyola and S. Davy, "Server placement and assignment in virtualized radio access networks,"*2015 11th International Conference on Network and Service Management (CNSM)*, Barcelona, 2015, pp. 398-401.

[22] R. Riggio, A. Bradai, D. Harutyunyan, T. Rasheed and T. Ahmed, "Scheduling Wireless Virtual Networks Functions," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 240-252, 2016.

[23] R. Riggio, T. Rasheed and R. Narayanan, "Virtual network functions orchestration in enterprise WLANs," *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, ON, 2015, pp. 1220-1225. (IM), May 11-15, 2015, Ottawa, ON, Canada. Piscataway: IEEE Press, 2015.

[24] Y. Xu and V. P. Kafle, "A Delay-Aware Service Function Chain Placement Scheme Based on Dynamic Programming," *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, Washington, DC, 2018, pp. 110-111.

[25] K. Liu, Q. Yuan and Y. Liu, "QoS-aware Resource Reallocation Based on Flow Priority in Software Defined Network," *2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Beijing, China, 2018, pp. 329-334.