Service Function Chain Planning with Resource Balancing in Space-Air-Ground Integrated Networks

Guangchao Wang, Sheng Zhou, Zhisheng Niu

Beijing National Research Center
for Information Science and Technology,
Department of Electronic Engineering,
Tsinghua University, Beijing 100084, China
Email: wgc15@mails.tsinghua.edu.cn,
{sheng.zhou, niuzhs}@tsinghua.edu.cn

Shan Zhang

Beijing Key Laboratory
of Computer Networks,
School of Computer
Science and Engineering,
Beihang Univesity,
Beijing 100191, China
Email: zhangshan18@buaa.edu.cn

Xuemin (Sherman) Shen

Department of Electrical
and Computer Engineering,
University of Waterloo,
200 University Avenue West,
Waterloo N2L 3G1,
Ontario, Canada
Email: sshen@uwaterloo.ca

***Abstract*—Space-air-ground integrated network (SAGIN) brings great potentials to extend the terrestrial networks and satisfy the diverse service demands from many emerging applications. The major challenge is the coordination of large-scale networks with heterogeneous communication and computation resources. In this paper, flexible and reconfigurable service provisioning based on service function chaining (SFC) is exploited to address the challenge, where the traffic flow of the network services need to pass through specified virtual network functions (VNFs) in a given order. Our main target is to optimize the planning of the service function chains under limited heterogeneous resources and to map them on physical networks, considering the balance of resource utilization of both communication and computation. The SFC planning problem is formulated as an integer non-linear programming problem, which is NP-hard. Then, we propose a heuristic SFC planning algorithm (HSP) to reduce the computational complexity. Moreover, we propose a new metric, aggregation ratio (AR), to observe the tradeoff between communication and computation resource consumptions. The simulations results demonstrate that the HSP achieves near-optimal performance and the communication and computation resources can be well tradeoffed via tuning AR. The service blockage probability is significantly decreased and the efficiency of resource utilization is improved by integrating SAGIN based on SFC.**

I. Introduction

With the development of 5G mobile networks, diverse emerging applications arise, such as virtual reality, smart cities, autonomous driving, and etc [1]. However, these applications require not only huge amounts of computing and storage resources, but also seamless communication coverages, as well as highly reliable and low-latency data transmission. Currently, various networks, including terrestrial mobile networks, space information networks, and airborne communication networks, have their own advantages and limitations, and lack efficient collaborations. The stringent requirements of emerging applications can not be met solely by the stand-alone networks.

The architecture of space-air-ground integrated network (SAGIN) has been proposed [2], [3], to make the best use of complementary advantages of each network segment, and thus can accommodate various different types of services simultaneously. However, it brings challenges to coordinate heterogeneous physical resources in such a large-scale dynamic network. In literatures, software defined networking (SDN) and network function virtualization (NFV) technologies have been proposed to be used for resource allocation and network management in SAGIN [4], [5]. Even so, how to identify different types of services and meet the dynamic service requirements with heterogeneous network resources is still yet to be addressed. In this paper, we exploit the concept of service function chaining (SFC), to provide flexible and reconfigurable services in SAGIN. Thus, different services are identified by service function chains, in which the virtual network functions (VNFs) are orchestrated in a given order. The problem is how to plan and map the service function chains to the physical networks. Specifically, we need to place the VNFs into proper physical nodes and design the routes of the service data, under the resource and quality of service (QoS) constraints.

Existing investigations on SFC planning and resource allocation in NFV have been summarized in [6]–[8]. In [9], the VNF chain composition and VNF embedding are jointly considered in NFV-based wireline networks, and is shown to be a mixed integer programming problem. A coordinated heuristic-based algorithm JoraNFV is proposed to get near-optimal solution. In [10], the VNFs placement is investigated to optimize the network operational costs and resource utilization, while guaranteeing the QoS. A dynamic programming-based heuristic algorithm is provided to solve the problem in large-scale NFV-based networks. The authors in [11] further formulate the readjustment of VNFs with dynamic service requests. The VNF placement in datacenters are investigated in [12] and [13]. Specifically, the objective in [12] is to minimize the number of used physical machines considering the time-varying workloads and basic resource consumptions,

This work is sponsored in part by Nature Science Foundation of China (No. 91638204, No. 61871254, No. 61571265, No. 61861136003, No. 61621091), National Key R&D Program of China 2018YFB0105005, and Hitachi Ltd.

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

and a two-stage heuristic algorithm is proposed to solve the problem. In [13], the joint VNF placement and traffic routing are formulated as an mixed integer programming problem to jointly maximize the reliability of network service and minimize the end-to-end delay. Nevertheless, most existing works concentrating on NFV-based wireline networks, where the communication is not the main bottleneck and there are no differences between physical nodes, and thus they are not suitable for SAGIN.

In this paper, we investigate the SFC planning problem in SAGIN, where the distinguishing features of aerial nodes are identified, and the resource consumptions of both computations and communications are balanced. Our goal is to maximize the number of user requests that can be successfully served, while minimizing total resource costs. The SFC planning problem is first formulated as an integer non-linear programming (INLP) problem in SAGIN. Then, a heuristic SFC planning algorithm (HSP) with low complexity is proposed, and is shown to achieve near-optimal performance through extensive simulations. Additionally, a new metric, aggregation ratio (AR), is proposed to elaborate the tradeoff between communication and computation cost, which is validated by simulation results. We find that when the bandwidth requirement of the service is small, consuming a small amount of communication resources can save a large amount of computation resources via increasing AR. Reversely, when the bandwidth requirement of the service is large, a large amount of communication resources can be saved at the expense of slightly increasing computation resource consumptions by decreasing AR.

The rest of this paper is organized as follows. System model is described in Section II, followed by the detailed explanation of the aggregation ratio. Then, the problem formulation and proposed algorithm are provided in Section III, and the performance is evaluated in Section IV. Finally, Section V concludes this paper.

II. System Model

We consider a general SAGIN, the topology of which is shown in Fig. 1. In this paper, we identify the space-air network via an aerial node, such as an HAP or a satellite. The aerial node has full connections to all terrestrial nodes within its coverage through wireless communications. The terrestrial nodes connects to each other by wired or wireless links depending on the network architecture.

A. Heterogeneous Physical Network

We model the physical network as a directed graph $G = (V, E)$, where V is the set of physical nodes and E is the set of transmission links that interconnect these network nodes. In SAGIN, we have $V = V_{\mathrm{A}} \cup V_{\mathrm{G}}$, where V_{A} is the set of aerial nodes and V_{G} is the set of terrestrial nodes. We assume that each physical node has a fixed computation capacity, denoted by $\{C_v \mid C_v > 0, v \in V\}$. We also have $E = E_{\mathrm{A}} \cup E_{\mathrm{G}}$, where E_{A} is the set of links that connect aerial nodes and E_{G} is the set of links that interconnect terrestrial nodes. The direction of the physical link is denoted by $\{(v, u) \in E \mid v, u \in V\}$, where v is the starting node and u is the destination node. For each link, the one hop delay is $\{D_{v,u} \mid D_{v,u} > 0, (v, u) \in E\}$. The amount of available bandwidth resources of physical links is denoted by $\{B_{v,u} \mid B_{v,u} > 0, (v, u) \in E\}$.

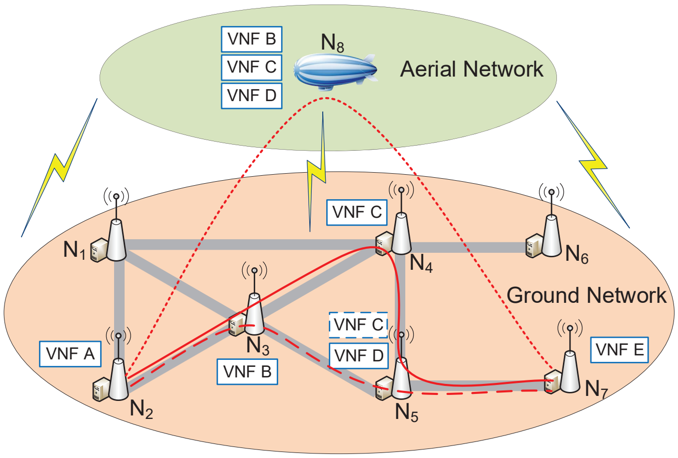

Fig. 1. The topology of a general SAGIN.

B. Service Requests

We assume that the set of service requests is known in advance, denoted by $R = \{r \mid r = 1, 2, ..., |R|\}$. We denote F as the set of VNFs that are used to compose the service function chains. The sets of source and destination nodes of service requests r is denoted by s_r and d_r, respectively. One service request typically contains the service data of multiple users, and thus has a constant bandwidth requirement, denoted by $\{B_r \mid r \in R\}$. The service request must be served within the delay requirement, denoted by $\{D_r \mid r \in R\}$. The set of VNFs composing request r is denoted by Π_r ($r \in R$). Then, we denote the service function chain as $\{\pi_r = (f_{r,1}, f_{r,2}, ..., f_{r,|\pi_r|}) \mid f_{r,1}, f_{r,2}, ..., f_{r,|\pi_r|} \in \Pi_r, r \in R\}$. The virtual links are defined to describe the connections between two VNFs in service function chains, which is denoted by $E_r = \{(i, j) \mid i, j \in \Pi_r, r \in R\}$.

C. Decision Variables

We define four types of decision variables to describe the SFC planning problem. The binary variable $a_{f,v}$ is used to indicate if VNF f is placed on node v, and the binary variable $x_{f,v,r}$ indicates whether the VNF f of request r is placed on network node v. Obviously, there is a non-linear relationship between $x_{f,v,r}$ and $a_{f,v}$, as

$$a_{f,v} = \mathbb{I}\left(\sum_{r \in R} x_{f,v,r} \geq 1\right), \forall f \in F, \forall v \in V, \tag{1}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. Similarly, the binary variable $y_{(v,u)}^{(i,j),r}$ is introduced to indicate whether the virtual link (i, j) of request r uses physical link (v, u). The binary variable z_r is defined to indicate whether service request r is successfully served.

D. Cost Modeling

Due to load limitations, the computation and bandwidth resources of aerial nodes are very scarce, leading to higher resource costs. Using a weighted linear model, we define the resource costs of both communication and computation into a uniform abstraction, as

$$\begin{aligned}\Xi_{\text{total}} =&\beta_{\text{A}}^{\text{cm}} \sum_{r\in R} \xi_{\text{A},r}^{\text{cm}} + \beta_{\text{G}}^{\text{cm}} \sum_{r\in R} \xi_{\text{G},r}^{\text{cm}} + \\ &\beta_{\text{A}}^{\text{cp}} \sum_{v\in V_{\text{A}}} \sum_{f\in F} \xi_{f,v}^{\text{cp}} + \beta_{\text{G}}^{\text{cp}} \sum_{v\in V_{\text{G}}} \sum_{f\in F} \xi_{f,v}^{\text{cp}},\end{aligned} \tag{2}$$

where Ξ_{total} is the total resource cost, $\beta_{\text{A}}^{\text{cm}}$, $\beta_{\text{G}}^{\text{cm}}$, $\beta_{\text{A}}^{\text{cp}}$ and $\beta_{\text{G}}^{\text{cp}}$ are the corresponding weights. $\xi_{\text{A},r}^{\text{cm}}$ is the communication resource consumption of aerial links, as

$$\xi_{\text{A},r}^{\text{cm}} = \sum_{(v,u)\in E_{\text{A}}} \mathbb{I}\left(\sum_{(i,j)\in E_r} y_{(v,u)}^{(i,j),r} \geq 1 \right) B_r, \tag{3}$$

and $\xi_{\text{G},r}^{\text{cm}}$ is the communication resource consumption of terrestrial links, as

$$\xi_{\text{G},r}^{\text{cm}} = \sum_{(v,u)\in E_{\text{G}}} \mathbb{I}\left(\sum_{(i,j)\in E_r} y_{(v,u)}^{(i,j),r} \geq 1 \right) B_r. \tag{4}$$

Note that the number of communication hops dominates the communication resource costs. Thus, the communication resources can be saved by routing the service data with less communication hops.

$\xi_{f,v}^{\text{cp}}$ is the computation resource consumption for a given VNF f on node v, as

$$\xi_{f,v}^{\text{cp}} = \xi_{\text{O},f} + \sum_{r\in R} \xi_{\text{P},f} x_{f,v,r}, \tag{5}$$

where $\xi_{\text{O},f}$ is the fixed computation resource consumption for operating and maintaining the VNF f, $\xi_{\text{P},f}$ is the actual computation resource consumption incurred by a service request when using VNF f. Thus, deploying fewer VNFs can save the fixed computation resources.

E. Aggregation Ratio

To save the computation resources, we can aggregate the VNFs which can be shared by multiple service requests, meaning that we only deploy one VNF of a specific type on one physical node. The new metric, aggregation ratio η, is defined to measure the *level of VNFs sharing* in SFC planning, as the ratio of the number of VNFs decreased by aggregation, to the total number of VNFs required by all service requests, which is

$$\eta = \frac{N_T - N_I}{N_T}, \tag{6}$$

where N_T is the total number of VNFs required by all services requests, N_I is the number of deployed VNFs.

Although computation resources can be saved by aggregation, extensive communication resources would be paid out. This is because that if the VNFs are sparsely deployed, the route of service data might be rather long to seek the required VNFs on the limited physical nodes, leading to higher transmission delay and the waste of communication resources. However, if the VNFs are densely deployed, the service function chains for each request can be easily built with less routing hops while bringing much more computation resource costs.

III. Problem Formulation and Solutions

We first elaborate the constraints of the SFC planning problem. In SAGIN, the flow conservation must be guaranteed to successfully establish the route of service data, meaning that for a given physical node, the data that flows in must be equal to the data that flows out, as

$$\begin{aligned}\sum_{u\in V} y_{(v,u)}^{(i,j),r} - \sum_{u\in V} y_{(u,v)}^{(i,j),r} = x_{i,v,r} - x_{j,v,r}, \\ \forall v \in V, \forall r \in R, \forall (i,j) \in E_r.\end{aligned} \tag{7}$$

To successfully serve a service request, all indispensable VNFs must be placed to one and only one physical nodes. That is

$$\sum_{v\in V} x_{f,v,r} = z_r, \forall f \in \Pi_r, \forall r \in R. \tag{8}$$

The first VNF of the service function chain is placed on the source node, and the last VNF is placed on the destination node, as

$$\begin{aligned}x_{f_r^{\text{f}},s_r,r} = z_r, \forall r \in R, \\ x_{f_r^{\text{l}},d_r,r} = z_r, \forall r \in R,\end{aligned} \tag{9}$$

where the subscript f_r^{f} and f_r^{l} denote the first VNF and last VNF of service request r respectively.

For a given physical node, the computation resource constraints must be satisfied, as

$$\sum_{r\in R} \sum_{f\in F} x_{f,v,r} \xi_{O,f} + \sum_{f\in F} a_{f,v} \xi_{P,f} \leq C_v, \forall v \in V. \tag{10}$$

For a given physical link, the communication resource constraints must be satisfied, as

$$\sum_{r\in R} \sum_{(i,j)\in E_r} y_{(v,u)}^{(i,j),r} B_r \leq B_{v,u}, \forall (v,u) \in E. \tag{11}$$

Besides, the total delay of the service request must be less than the delay requirement to guarantee the QoS, and thus

$$\sum_{(i,j)\in E_r} \sum_{(v,u)\in E} y_{(v,u)}^{(i,j),r} D_{v,u} \leq D_r, \forall r \in R. \tag{12}$$

Our goal is to maximize the number of service requests that can be successfully served. Meanwhile, the costs of using communication and computation resources need to be minimized. Therefore, the objective function is

$$\mathcal{U} = \sum_{r\in R} z_r K_r - \sum_{f\in F} \xi_{O,f}\left(\sum_{v\in V_A}\beta_{\mathrm{A}}^{\mathrm{cp}} a_{f,v} + \sum_{v\in V_G}\beta_{\mathrm{G}}^{\mathrm{cp}} a_{f,v}\right) - \sum_{r\in R}\sum_{(i,j)\in E_r}\sum_{v\in V} B_r\left(\sum_{u\in V_A}\beta_{\mathrm{A}}^{\mathrm{cm}} y_{(v,u)}^{(i,j),r} + \sum_{u\in V_G}\beta_{\mathrm{G}}^{\mathrm{cm}} y_{(v,u)}^{(i,j),r}\right), \tag{13}$$

where K_r is the revenue for serving a service request r. The objective function is the net revenue of SFC planning, which equals to the total revenue earned from successfully serving the requests minus the total costs of communication and computation resources.

Then, the joint VNF embedding and virtual link mapping problem can be formulated as

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y},\mathbf{a},\mathbf{z}} \quad & \mathcal{U} \\ \text{s.t.} \quad & a_{f,v} = \mathbb{I}\left(\sum_{r\in R} x_{f,v,r} \geq 1\right), \forall f \in F, \forall v \in V, \quad (14) \\ & \text{Eq.}(7) - \text{Eq.}(12). \end{aligned}$$

There is a non-linear constraint due to the indicator function, and all the variables are integers. Thus, it is an INLP problem, which has been proved as a NP-hard problem [10]. The exact solutions can be obtained only when the network scale is small. However, the computing time will be intolerable as the network scale increases. To this end, a heuristic SFC planning algorithm is proposed to reduce the computational complexity.

In Algorithm 1, we allocate different priorities to the ground and aerial networks for resource utilization. Intuitively, we prefer to meet the service demands solely by ground network to decrease the resource costs. However, if the service request is blocked due to the QoS violation, the aerial resources will be used to aid the services.

Algorithm 1 HSP for SAGIN

```
 1: Initialize the network status and SFC service requests
 2: for each service request do
 3:   Find optimal path in ground network with Algorithm 2
 4:   if there is at least one feasible path then
 5:     Embed the VNFs into the nodes along the optimal
        path with Algorithm 3
 6:     if Resource constraints are violated then
 7:       The service request is blocked
 8:     else
 9:       The service request is successfully served
10:       Break loop
11:     end if
12:   else
13:     Find optimal path in SAGIN with Algorithm 2
14:     if there is at least one feasible path then
15:       Do step 5 to 12
16:     else
17:       The service request is blocked
18:     end if
19:   end if
20: end for
```

Another key idea is to decouple the VNF placement and service data routing by Algorithm 2 and Algorithm 3. The Algorithm 2 is used to search optimal path from source node to destination node of each service request in a greedy manner based on Dijkstra algorithm. Then, we deploy all involved VNFs along the selected path using Algorithm 3. Such decoupling will not significantly degrade the performance as the resources are used efficiently. However, the weights of the links should be carefully designed. A feasible route must satisfy the bandwidth resource requirements of the services, and the end-to-end delay requirement can not be violated. Furthermore, the level of function sharing should also be considered to balance the resource utilization. We define the function sharing factor as

$$S_{v,r} = \sum_{f\in\Pi_r} A_{f,v}, \forall r \in R, \forall v \in V, \tag{15}$$

where $A_{f,v}$ is the VNF deployment status, meaning that if the VNF f has been placed on node v. Based on the function sharing factor, we design the edge weight for physical links in the network graph, as

$$W_r(v,u) = \frac{D_{v,u}\mathbb{I}(B_{v,u}^{\mathrm{Re}} - B_r)}{\exp(\rho(S_{v,r} + S_{u,r}))}, \forall r \in R, \tag{16}$$

where $B_{v,u}^{\mathrm{Re}}$ denotes the remaining bandwidth resources of physical link (v,u), and ρ is the aggregation factor which is used to tune the aggregation ratio. If the remaining bandwidth resources are not enough for the service request, the weight will be 0, meaning that the link is broken. The weight is negatively correlated to the function sharing factor. Note that the weight is positively correlated to the hopping delay. However, if the remaining bandwidth resources are not enough for serving the service request, the weight will be 0, indicating that the link is broken. Furthermore, the weight is negatively correlated to the function sharing factor and the aggregation factor ρ. The HSP will choose a feasible path with minimal total weights. Intuitively, when ρ is large, the weight is dominated by the function sharing factor. Then, the path that can potentially share more VNFs will be selected. However, the delay requirements can be violated. If ρ equals to 0, the weight only depends on the hopping delay. Then, the HSP will select the path with minimum hopping delay regardless the VNFs sharing. In HSP, to guarantee the delay performance, the proper value of ρ is searched by a step Δ.

For any physical node v, we design the node weight as

$$W_r(v) = S_{v,r}(C_v^{\mathrm{Re}} - C_{r,v}^{\mathrm{Ne}}), \forall r \in R, \tag{17}$$

where C_v^{Re} denotes the remaining computation resources of physical node v, and $C_{r,v}^{\mathrm{Ne}}$ is the necessary computation resources for placing all VNFs of service request r on node v.

Algorithm 2 Service Flow Routing

```
Initialize aggregation factor $\rho$ and searching step $\Delta$
while $\rho >= 0$ do
    Calculate the weight of edges in the network graph using Eq. (16)
    Find the shortest path using Dijkstra algorithm
    if the delay requirement is not violated then
        Return this path as the routing path
        Break the loop
    else
        $\rho = \rho - \Delta$
    end if
end while
```

The set of potential nodes P_v first contains all nodes in the path. Then the node with maximum weight will be selected and all VNFs are embedded on this node. If the computation capacity is violated, the candidate node will be removed from P_v. The service request is blocked until P_v is empty.

Algorithm 3 VNFs Placement

```
Initialize the set of potential nodes $P_v$ as all nodes in the optimal path obtained by Algorithm 2
while $P_v$ is not empty do
    Calculate the weight of the nodes in $P_v$ using Eq. (17)
    Select the node with maximal weight as candidate node
    if Computation capacity is not violated then
        Place all VNFs to the candidate node
        Break the loop
    else
        Remove the candidate node from $P_v$
    end if
end while
```

For Algorithm 2, the complexity mainly results from the *while* loop and Dijkstra algorithm. We use $|V|$ to denote the total number of network nodes. Then, the complexity of Algorithm 2 is $\mathcal{O}(\lceil \frac{\rho}{\Delta} \rceil |V|^2)$. Since the number of nodes in P_v cannot exceed $|V|$, the worst case of Algorithm 3 has a complexity of $|V|^2$. The complexity of *while* loop of Algorithm 1 is dominated by the total number of service requests, which is denoted by $|R|$. In summary, the complexity of HSP is $\mathcal{O}(|R| \lceil \frac{\rho}{\Delta} \rceil |V|^2)$.

IV. Performance Evaluation

In the simulations, the topology of the SAGIN is shown in Fig. 1. The service requests are randomly generated with 3-6 VNFs. The bandwidth capacity of links and computation capacity of nodes are uniformly distributed in $[500, 600]$ Mbps and $[800, 1000]$ TFLOPS. The single hop delay is set to $[10, 15]$ ms following uniform distribution.

We evaluate the performance of our proposed algorithm and compare it with the exact solution obtained by *Solving Constraint Integer Programs* (SCIP) solver [14], as shown in Fig. 2. The upper bound is obtained by linearizing and relaxing the primary problem into a linear optimization problem. We also provides an function sharing optimal scheme as the benchmark, in which the VNFs are shared as many as possible neglecting the communication resource consumptions. We can observe that the function sharing optimal scheme has the worst performance, because the bandwidth resources are consumed in a greedy manner and are soon exhausted, leading to inevitable service blockages. It can be seen that the HSP achieves near-optimal performance when the number of service requests is not too large. However, the gap between HSP and optimal solution increases as the number of service requests increases. This is because that some service requests can be blocked in HSP when the network resources are gradually approaching the bottleneck.

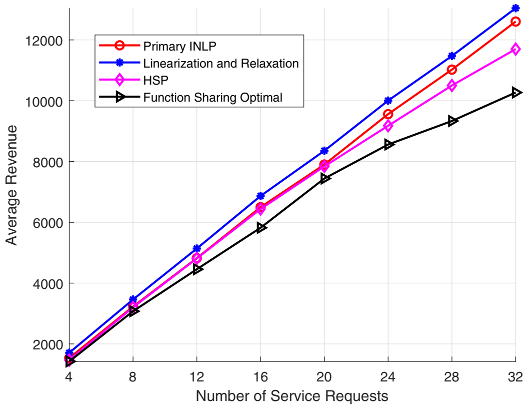

Fig. 2. The performance of HSP compared with benchmarks.

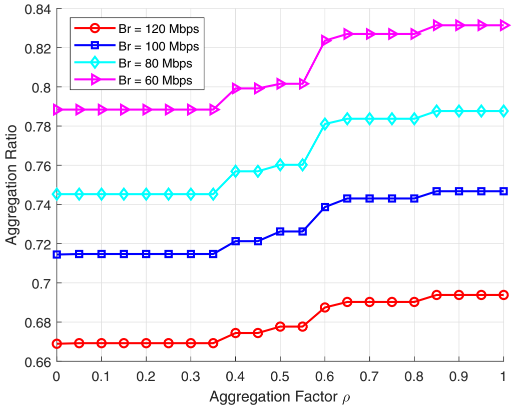

Fig. 3. Aggregation ratio versus aggregation factor ρ in HSP under different bandwidth requirement B_r.

Fig. 3 validates that the AR can be tuned by the aggregation factor ρ in HSP. Note that the AR increases as ρ increases, because the function sharing factor gradually dominates the selection of optimal route. Correspondingly, the tradeoff between communication and computation resource costs is presented in Fig. 4. It can be observed that the computation resource consumption increases and bandwidth resource consumption

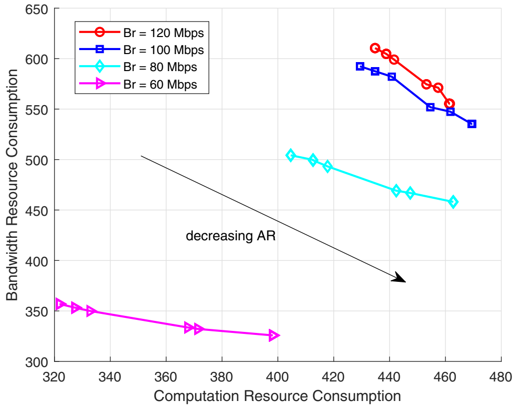

Fig. 4. Tradeoff between bandwidth resource consumption and computation resource consumption under different bandwidth requirement B_r.

decreases as AR decreases. Specifically, when B_r is small, as illustrated by the magenta right-pointing triangle line, sacrificing a small amount of communication resources can trade for a large amount of computation resource savings. On the other hand, when B_r is large, as illustrated by the red circle line, we can use a small amount of computing resources to trade for a large amount of communication resources.

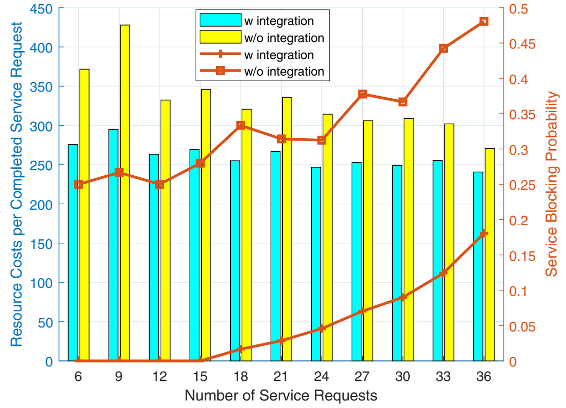

Fig. 5. Comparison between the network with integration and without integration. The bars correspond to the resource costs while the lines correspond to the service blocking probability.

We also compare the resource costs and the blocking probability between the SFC-based SAGIN and the stand-alone terrestrial or aerial networks in Fig. 5. The cost per completed service requests is significantly decreased by 12.5% to 45.1% via integration, indicating that higher resource utilization efficiency is achieved. Furthermore, the blocking probability is also reduced by integration.

V. Conclusions

In this work, we propose to exploit SFC in a large-scale heterogeneous network, i.e. SAGIN. The SFC planning problem is addressed to identify various services and flexibly meet service demands with limited heterogeneous resources, which is formulated as an INLP. Then HSP is proposed to reduce the computational complexity. Its heuristic is based on different features of terrestrial and aerial nodes, and the weights of network nodes and links are carefully designed jointly considering the resource balance and QoS guarantees. We also propose AR to elaborate the tradeoff between communication and computation resource costs. Simulation results show that the HSP can obtain near-optimal results. When the amount of service data is low, a large amount of computation resources can be saved by consuming a small amount of additional communication resources via increasing AR. On the other hand, when the amount of service data is high, slightly increasing computation resource consumptions can save a large amount of communication resources by decreasing AR. We also find that the SFC-based SAGIN reduces 12.5% to 45.1% total resource costs per completed service request compared with stand-alone terrestrial or aerial networks and significantly decreases the service blocking probability.

References

[1] M. Agiwal, A. Roy and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617-1655, thirdquarter 2016.

[2] J. Liu, Y. Shi, Z. M. Fadlullah and N. Kato, "Space-Air-Ground Integrated Network: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714-2741, Fourthquarter 2018.

[3] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou and X. S. Shen, "Air-Ground Integrated Mobile Edge Networks: Architecture, Challenges, and Opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26-32, August 2018.

[4] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang and X. S. Shen, "Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101-109, 2017.

[5] M. Sheng, Y. Wang, J. Li, R. Liu, D. Zhou and L. He, "Toward a Flexible and Reconfigurable Broadband Satellite Network: Resource Management Architecture and Strategies," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 127-133, Aug. 2017.

[6] G. Mirjalily and Z. Luo, "Optimal Network Function Virtualization and Service Function Chaining: A Survey," *Chinese J. Elect.*, vol. 27, no. 4, pp. 704-717, 2018.

[7] Y. Xie, Z. Liu, S. Wang and Y. Wang , "Service function chaining resource allocation: A survey," *arXiv preprint arXiv:1608.00095*, 2016.

[8] J. Gil Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518-532, Sept. 2016.

[9] 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.

[10] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba and O. C. M. B. Duarte, "Orchestrating Virtualized Network Functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 725-739, Dec. 2016.

[11] J. Liu, W. Lu, F. Zhou, P. Lu and Z. Zhu, "On Dynamic Service Function Chain Deployment and Readjustment," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 543-553, Sept. 2017.

[12] D. Li, P. Hong, K. Xue and j. Pei, "Virtual Network Function Placement Considering Resource Optimization and SFC Requests in Cloud Datacenter," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1664-1677, July. 2018.

[13] L. Qu, C. Assi, K. Shaban and M. J. Khabbaz, "A Reliability-Aware Network Service Chain Provisioning With Delay Guarantees in NFV-Enabled Enterprise Datacenter Networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 554-568, Sept. 2017.

[14] SCIP Optimization. *SCIP Doxygen Documentation*. [Online]. Available: https://scip.zib.de