SFC-Based Service Provisioning for Reconfigurable Space-Air-Ground Integrated Networks

Guangchao Wang, Sheng Zhou, *Member, IEEE*, Shan Zhang, *Member, IEEE*, Zhisheng Niu, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Space-air-ground integrated networks (SAGIN) extend the capability of wireless networks and will be the essential building block for many advanced applications, like autonomous driving, earth monitoring, and etc. However, coordinating heterogeneous physical resources is very challenging in such a large-scale dynamic network. In this paper, we propose a reconfigurable service provisioning framework based on service function chaining (SFC) for SAGIN. In SFC, the network functions are virtualized and the service data needs to flow through specific network functions in a predefined sequence. The inherent issue is how to plan the service function chains over large-scale heterogeneous networks, subject to the resource limitations of both communication and computation. Specifically, we must jointly consider the virtual network functions (VNFs) embedding and service data routing. We formulate the SFC planning problem as an integer non-linear programming problem, which is NP-hard. Then, a heuristic greedy algorithm is proposed, which concentrates on leveraging different features of aerial and ground nodes and balancing the resource consumptions. Furthermore, a new metric, aggregation ratio (AR) is proposed to elaborate the communication-computation tradeoff. Extensive simulations shows that our proposed algorithm achieves near-optimal performance. We also find that the SAGIN significantly reduces the service blockage probability and improves the efficiency of resource utilization. Finally, a case study on multiple intersection traffic scheduling is provided to demonstrate the effectiveness of our proposed SFC-based service provisioning framework.**

***Index Terms*—Service function chaining (SFC), space-air-ground integrated networks (SAGIN), heterogeneous networks, virtual network functions (VNFs).**

I. Introduction

IN THE era of 5G mobile networks, many new applications are emerging, such as autonomous driving, Internet of things, earth monitoring, smart cities, and etc., [1]. However, these applications not only require wide network coverages, seamless access and low latency transmissions, but also have a huge amount of real-time data to be stored and processed. This inevitably imposes much more stringent requirements on network infrastructures and service provisions. Unfortunately, current stand-alone networks, such as terrestrial mobile networks, space information networks [2]–[5], and airborne communication networks [6]–[8], are constructed for exclusive purposes and specific missions. For instance, space and air networks are suitable for broadcasting services [9] and remote data sensing, collection and dissemination with wide-range coverage, especially when the terrestrial networks fail [10]–[12]. However, the storage and computation resources are scarce on the aerial nodes, which makes it difficult to handle intensive computation tasks. On the other hand, ground networks have more resources to accomplish sophisticated tasks, but have limited coverage. Thus, existing stand-alone networks are operated independently and lack collaboration mechanisms. To make the best use of complementary advantages, space-air-ground integrated network (SAGIN) has been proposed [13], [14] as a promising architecture to satisfy diverse quality of service (QoS) requirements of emerging applications. Via integration, the networks are able to provide seamless global connectivity, as well as support computation intensive services with high data rate requirements.

However, coordinating heterogeneous physical resources in such a large-scale dynamic network is very challenging. In [28], the cooperative multicast transmission with spectrum reusing in integrated terrestrial-satellite networks is investigated, which is shown to improve the system performance compared with the stand-alone terrestrial network. A novel spectrum exploitation framework is proposed in [29] to solve the spectrum scarcity problem in satellite and terrestrial communication systems. Some existing researches have proposed to use software defined networking (SDN) and network function virtualization (NFV) technologies to manage virtual resources abstracted from physical infrastructures [15]–[17]. In [30], an SDN-based spectrum sharing and traffic offloading mechanism is proposed to achieve coordination between satellite and terrestrial networks. Nevertheless, under SDN and NFV, it is still essential to differentiate the services and flexibly match the network resources to service demands. In this paper, we propose to leverage the concept of service function chaining (SFC) in SAGIN, to enable flexible and reconfigurable service provisioning, as shown in Fig. 1. Diverse applications

Manuscript received April 15, 2019; accepted February 28, 2020. Date of publication April 9, 2020; date of current version June 29, 2020. This work was supported in part by the Nature Science Foundation of China under Grant 91638204, Grant 61871254, Grant 61861136003, and Grant 61801011, in part by the National Key Research and Development Program of China under Grant 2018YFB0105005 and Grant 2018YFB1800800, and in part by the Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles. *(Corresponding author: Sheng Zhou.)*

Guangchao Wang, Sheng Zhou, and Zhisheng Niu are with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: sheng.zhou@tsinghua.edu.cn).

Shan Zhang is with the Beijing Key Laboratory of Computer Networks, School of Computer Science and Engineering, Beihang University, Beijing 100191, China.

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2020.2986851

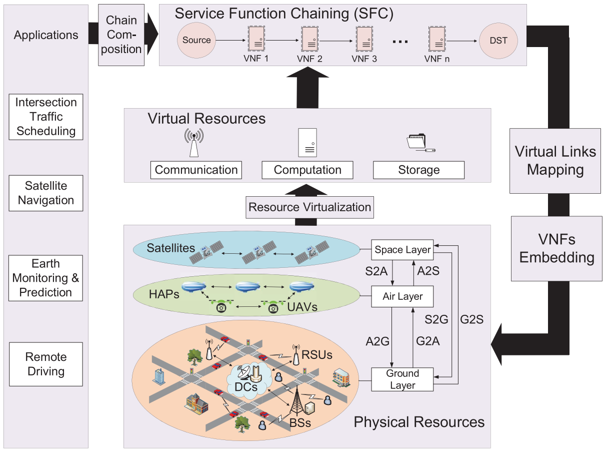


Fig. 1. SFC-based reconfigurable service provisioning framework for SAGIN.

such as intersection traffic scheduling, satellite navigation, earth monitoring, remote driving, and etc, can be identified by service function chains. Using resource virtualization, the heterogeneous physical resources from different network segments, e.g. satellites, high altitude platforms (HAPs), unmanned aerial vehicles (UAVs), base stations (BSs), and etc, are abstracted into unified virtual resource pools. Then, via NFV management, the virtual resources are flexibly provisioned to support various virtual network functions (VNFs), which are orchestrated in a specified sequence to compose a service function chain. The main challenge herein is SFC planning problem, which consists of two fundamental issues:

1) VNFs embedding problem: how to place VNFs into proper physical network nodes?
2) Virtual link mapping problem: how to route the data flow among VNFs?

To address above two issues, recent studies on SFC planning and resource allocation have been adopted, as summarized in [18]–[20]. In [22] and [24], the VNF placement in NFV-based wireline networks is formulated as an integer programming problem. Different heuristic algorithms are proposed to obtain near-optimal solutions. The VNF placement in datacenters are investigated in [21] and [23]. Specifically, the objective in [21] is to minimize the number of used physical machines considering the time-varying workloads and basic resource consumptions, and a two-stage heuristic algorithm is proposed to solve the problem. In [23], the joint VNF placement and traffic routing are formulated as a mixed integer programming problem to jointly maximize the reliability of network service and minimize the end-to-end delay. In their heuristic algorithm, the link weight of network graph is designed solely based on the end-to-end delay. Then the shortest path is searched in a greedy manner and expanded to enhance the reliability. However, the network resources are not used efficiently, resulting in potential drops of the requests. Thus, the weights of physical links need to be carefully designed jointly considering the resource utilization and QoS gruarantee in SAGIN. Most existing works focus on wireline networks where the communication is not the main bottleneck and physical nodes have no differences, and thus their solutions cannot be directly applied for SAGIN.

In this paper, we optimize the resource management of heterogeneous nodes in SAGIN, to balance the resource consumptions of computations and communications. First, the differences of various types of network nodes, in terms of coverage and processing capabilities, need to be identified. Second, function sharing is an effective approach to save computation resources in SFC planning. Specifically, one VNF can be shared by multiple SFC requests to reduce the number of VNF instances deployed on physical nodes. However, it consumes more communication resources because the data flow may experience more hops to reach the VNF. Thus, the tradeoff between communication and computation resource consumptions needs to be carefully investigated. In addition, a solution with low complexity is desired considering the large-scale nature of SAGIN.

To cope with these challenges, we first formulate the SFC planning problem in SAGIN as an integer non-linear programming problem (INLP). Then, a heuristic decoupled greedy algorithm with low complexity is proposed, and is shown to achieve near-optimal performance. The heuristic is based on different features of aerial and ground nodes. Specifically, we allocate a higher priority to the ground network to make full use of its computation resources. When the end-to-end delay can not be guaranteed solely by the ground network, the aerial nodes are leveraged to reduce the number of transmission hops. Then, the optimal path is searched in a greedy manner according to the QoS requirement, the level of function sharing and bandwidth capacity. At last, we choose optimal nodes along the optimal path according to available computation resources and the level of function sharing to embed corresponding VNFs. Furthermore, we propose a new metric, aggregation ratio (AR), to measure the *the level of function sharing* in SFC planning. Extensive simulations are performed to evaluate our proposed algorithm and the communication-computation tradeoff is shown via tuning AR. Compared with stand-alone networks, the SAGIN are shown to significantly decrease the service blockage probability and reduce 12.5% to 45.1% of total resource costs per completed service request.

Finally, a case study on multiple intersection traffic scheduling supported by SAGIN is provided in order to show how the service function chains are planned and mapped on heterogeneous physical resources for a practical application. When the traffic load is low, only ground nodes are used and the computation intensive functions (like *Vehicle Detection*) are placed in a centralized manner to save computation resources. When the traffic load is high, the aerial nodes are used to relieve the traffic burden. Specifically, the *Radio Access* is placed on aerial nodes when the communications become the bottleneck of the network, while the *Information Fusion* is placed on aerial nodes when the computations become the bottleneck of the network. We also find that the number of vehicles with vehicle-to-everything communication capability that the network can simultaneously accommodate is increased by 50% by leveraging the aerial node.

The rest of this paper is organized as follows. System model is described in Section II, and the AR is introduced in

TABLE I

NOTATIONS

Notations	Descriptions
N	The set of physical nodes
N_{A}	The set of aerial nodes
N_{G}	The set of ground nodes
E	The set of physical links
E_{A}	The set of physical links that connect to aerial nodes
E_{G}	The set of physical links that solely connect to ground nodes
C_n	The computation capacity of physical node n
$B_{n,m}$	The available bandwidth resource of physical link (n,m)
$D_{n,m}$	The single hop delay of physical link (n,m)
F	The set of VNFs
Q	The set of service requests
s_q	The source node of service request q
d_q	The destination node of service request q
B_q	The bandwidth requirement of service request q
D_q	The deadline of service request q
Π_q	The set of VNFs required by service request q
π_q	The service function chain of service request q
E_q	The set of virtual links of service request q
$a_{f,n}$	The binary variable that indicates whether VNF f is installed and maintained on node n
$x_{f,n,q}$	The binary variable that indicates whether VNF f of service request q is embedded on node n
$y_{(n,m)}^{(i,j),q}$	The binary variable that indicates whether the virtual link (i,j) of request q is mapped on physical link (n,m)
z_q	The binary variable that indicates whether the service request q is successfully served
R_q	The revenue coefficient for serving request q
Φ_{total}	The total resource cost
$\phi_{B,f}$	The constant computation resource consumption for the installation and maintaining of VNF f
$\phi_{A,f}$	The computation resource consumption for serving the VNF f of a service request

Section III. The problem formulation and proposed algorithm are provided in Section IV, and the performance is evaluated in Section V. The case study is shown in Section VI. Finally, Section VII concludes this paper.

II. System Model

The symbols used in this paper are listed in Table I.

We consider a general topology of SAGIN, as shown in Fig. 2. The space-air network is identified by an aerial node (such as a satellite or HAP), i.e. N_8 in illustration, which has a large coverage and connects all ground nodes via wireless links. The ground nodes are connected by wired or wireless links. The tasks from end users are represented by service function chains which are composed of several specific VNFs. Based on NFV technology, these VNFs can be flexibly placed on both the aerial and ground nodes.

Fig. 2 shows three strategies of SFC planning schematically. Considering a service function chain that requires 5 VNFs from VNF A to VNF E sequentially, the traffic starts from node N_2 and eventually flows into N_7. Three strategies are represented by solid red line, dashed red line, and dotted red line respectively. The solid line shows a basic strategy of SFC planning, where the VNFs are placed on five network nodes respectively and the traffic experiences four-hop transmissions. In comparison, the traffic only experiences three-hop transmissions using the strategy shown as the dashed line, indicating that the communication resource consumption is reduced. However, an additional VNF C should be installed and maintained on node N_5, which consumes more computation resources. Note that only terrestrial resources are used in these two strategies. As shown in dotted line, the traffic only experiences two-hop transmissions, which can significantly reduce the hopping delay at the expense of the resource consumptions of the aerial network.

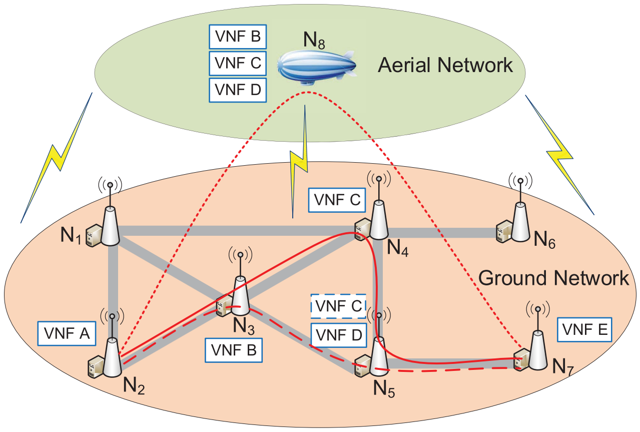


Fig. 2. An example of SFC planning and the topology of a general SAGIN.

A. Physical Network and VNFs

The physical network is represented by a directed graph $G = (N, E)$, where N is the set of network nodes and E is the set of physical links that interconnect these network nodes. In SAGIN, we have $N = N_{\mathrm{A}} \cup N_{\mathrm{G}}$, where N_{A} is the set of aerial nodes and N_{G} is the set of ground nodes. We also have $E = E_{\mathrm{A}} \cup E_{\mathrm{G}}$, where E_{A} is the set of physical links that connect aerial nodes and E_{G} is the set of physical links that solely connect ground nodes. We use $\{(n,m) \in E \mid n,m \in N\}$ to denote the directional link, where n is the initial node and m is the terminal node. As we consider a one-shot optimization problem given the network conditions, each physical node is assumed to have a fixed computation capacity [21]–[24], denoted by $\{C_n \mid C_n > 0, n \in N\}$. The physical links are identified by available bandwidth resources, denoted by $\{B_{n,m} \mid B_{n,m} > 0, (n,m) \in E\}$. The delay of single hop is denoted by $\{D_{n,m} \mid D_{n,m} > 0, (n,m) \in E\}$.

We consider a set of VNFs that are required to provide network services, denoted by F. These VNFs can be flexibly embedded on any physical nodes. A binary variable $a_{f,n}$ is introduced to indicate whether VNF f is installed and maintained on node n. The solution vector is denoted by $\mathbf{a} = \{a_{f,n} \mid f \in F, n \in N\}$. We have

$$a_{f,n} = \begin{cases} 1, & \text{VNF } f \text{ is installed and maintained on node } n, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

B. Service Requests

We consider a reconfiguration period of length T. At the starting point of each period, we assume that a batch of service requests arrive simultaneously with different service deadlines. Actually, these service requests probably arrive at any time during the last reconfiguration period. We accumulate these arrived service requests and treat them as a batch in the current period, and the deadline of each service is assigned according to the delay requirement and its arrival time. Besides, the topology of the SAGIN is assumed to be fixed in each period and can change in different reconfiguration periods due to the quasi stationary nature of the HAPS [26], [27]. The service procedure will not be interrupted by the reconfigurations, which means that if a service can not be accomplished in the current period, it will continue to be served in the next reconfiguration period. However, the remaining available resources, including computation resources of physical nodes and bandwidth resources of physical links, must be updated. Accordingly, a general one-shot optimization problem is considered, based on the current network status consisting of the network topology and residual available resources.

We consider a set of service requests, denoted by $Q = \{q \mid q = 1, 2, \ldots, |Q|\}$. The sets of source and destination nodes of corresponding service requests are denoted by $\{s_q \mid q \in Q\}$ and $\{d_q \mid q \in Q\}$. We assume that each service request has a fixed bandwidth requirement, denoted by $\{B_q \mid q \in Q\}$ [23], [24]. A service request is successfully served only if it is allocated with required bandwidth. The deadline of the service request is denoted by $\{D_q \mid q \in Q\}$. According to the bandwidth requirement and the deadline, the service requests are divided into four types: delay sensitive low bandwidth request (DSLBR), delay tolerant low bandwidth request (DTLBR), delay sensitive high bandwidth request (DSHBR), and delay tolerant high bandwidth request (DTHBR). In this paper, we assume that the service function sequence is given when the service request arrives [21]–[24]. The function chain composition is out of the scope of this paper. The set of virtual functions required by request q is denoted by Π_q $(q \in Q)$. The set of the service function chain is denoted by $\{\pi_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,|\pi_q|}) \mid f_{q,1}, f_{q,2}, \ldots, f_{q,|\pi_q|} \in \Pi_q, q \in Q\}$. The interconnections among VNFs in service function chains are regarded as virtual links. We use $E_q = \{(i,j) \mid i, j \in \Pi_q, q \in Q\}$ to represent the virtual links from VNF i to VNF j in service function chain π_q.

The binary variable $x_{f,n,q}$ is defined to indicate whether the virtual function f of request q is embedded on node n, as

$$x_{f,n,q} = \begin{cases} 1, & \text{VNF } f \text{ of request } q \text{ is embedded} \\ & \text{on node } n, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The solution vector is denoted by $\mathbf{x} = \{x_{f,n,q} \mid f \in F, n \in N, q \in Q\}$. According to the definitions, there exists a nonlinear relationship between $x_{f,n,q}$ and $a_{f,n}$, as

$$a_{f,n} = \mathbb{I}\left(\sum_{q \in Q} x_{f,n,q} \geq 1\right), \quad \forall f \in F, \forall n \in N, \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Similarly, we define the binary variable $y_{(n,m)}^{(i,j),q}$ to denote whether the virtual link (i,j) of request q is mapped on physical link (n,m), as

$$y_{(n,m)}^{(i,j),q} = \begin{cases} 1, & \text{virtual link } (i,j) \text{ of request } q \text{ is} \\ & \text{mapped on physical link } (n,m), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The solution vector is denoted by $\mathbf{y} = \{y_{(n,m)}^{(i,j),q} \mid (n,m) \in E, (i,j) \in E_q, q \in Q\}$. We also define another binary variable z_q to describe whether service request q is successfully served, as

$$z_q = \begin{cases} 1, & \text{request } q \text{ is successfully served,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The solution vector is denoted by $\mathbf{z} = \{z_q \mid q \in Q\}$.

C. Delay Modeling

As the SAGIN is an extremely large-scale network, we assume that the end-to-end delay of the service is dominated by transmission hops. The delay of single hop consists of transmission delay, propagation delay, processing delay and queuing delay, which is

$$D_{\text{hop}} = D_{\text{tran}} + D_{\text{prop}} + D_{\text{proc}} + D_{\text{que}}. \tag{6}$$

Then the end-to-end delay can be expressed as

$$D_{\text{e2e}} = D_{\text{hop}} N_{\text{hop}}, \tag{7}$$

where the N_{hop} is the number of experienced hops for serving the request. Note that the value of the end-to-end delay for serving the request highly depends on the number of hops, indicating that reducing the number of hops by using aerial nodes can significantly decrease the end-to-end delay.

D. Cost Modeling

For a given service request q, the communication resource consumption of aerial links is

$$\phi_{\text{A},q}^{\text{cm}} = \sum_{(n,m) \in E_{\text{A}}} \mathbb{I}\left(\sum_{(i,j) \in E_q} y_{(n,m)}^{(i,j),q} \geq 1\right) B_q, \tag{8}$$

and the communication resource consumption of ground links has similar expression as

$$\phi_{\text{G},q}^{\text{cm}} = \sum_{(n,m) \in E_{\text{G}}} \mathbb{I}\left(\sum_{(i,j) \in E_q} y_{(n,m)}^{(i,j),q} \geq 1\right) B_q. \tag{9}$$

Note that the communication resource consumption highly relies on the number of communication hops. Thus, we can save the communication resources by reducing the number of communication hops.

For a given VNF f on node n, the computation resource consumption is

$$\phi_{f,n}^{\text{cp}} = \phi_{\text{B},f} + \sum_{q \in Q} \phi_{\text{A},f} x_{f,n,q}, \tag{10}$$

where $\phi_{\text{B},f}$ is the constant computation resource consumption for VNF installation and maintaining, $\phi_{\text{A},f}$ is the additional computation resource consumption for servicing a request.

Note that the constant computation resources can be saved by reducing the number of deployed VNFs. An efficient approach is to aggregate the VNFs that can be shared by different service requests.

The resource costs vary among aerial nodes and ground nodes, because the resources of aerial nodes are more scarce. Therefore, we define a uniform abstraction of resource costs for both communication and computation using a linear model

$$\begin{aligned}\Phi_{\text{total}} = \alpha_{\text{A}}^{\text{cm}} \sum_{q\in Q} \phi_{\text{A},q}^{\text{cm}} + \alpha_{\text{G}}^{\text{cm}} \sum_{q\in Q} \phi_{\text{G},q}^{\text{cm}} \\ + \alpha_{\text{A}}^{\text{cp}} \sum_{n\in N_{\text{A}}} \sum_{f\in F} \phi_{f,n}^{\text{cp}} + \alpha_{\text{G}}^{\text{cp}} \sum_{n\in N_{\text{G}}} \sum_{f\in F} \phi_{f,n}^{\text{cp}},\end{aligned} \tag{11}$$

where Φ_{total} is the total resource cost, $\alpha_{\text{A}}^{\text{cm}}$, $\alpha_{\text{G}}^{\text{cm}}$, $\alpha_{\text{A}}^{\text{cp}}$ and $\alpha_{\text{G}}^{\text{cp}}$ are the corresponding weights.

III. Aggregation Ratio

In this section, we propose a new metric, aggregation ratio (AR), to elaborate the tradeoff between communication and computation resource costs in SFC planning. When we place the VNFs to physical nodes, we can aggregate these VNFs, which means that only one VNF instance of a specific type is required to be installed and maintained on one physical node. Multiple service requests can share this VNFs to save the basic computation resources. To measure the *level of VNFs sharing* in SFC planning, we define the aggregation ratio as the ratio of the number of VNF instances that are saved by aggregation to the total number of VNFs that are required by all service requests, as

$$AR = \frac{N_V - N_I}{N_V}, \tag{12}$$

where N_V is the total number of VNFs that are required for all service requests, N_I is the actual number of VNFs that are deployed on physical nodes.

Although computation resources can be saved through VNFs aggregation, additional communication resources will be consumed. This is because that the data flow of service requests will deviate from the initial routing path to reach the shared VNFs, which potentially increases the number of communication hops. Thus, the AR can reflect the tradeoff relationship between communication and computation resource consumptions. In SFC planning, a low AR indicates a high computation cost, while a high AR indicates a high communication cost. Therefore, if we can control the AR while optimizing the SFC planning, the minimal resource cost can be obtained. Specifically, when the computation resources are the main bottleneck of the network, we can trade the communication resources for computation resources by tuning up the value of AR, and vise versa.

IV. Problem Formulation and Solutions

In this section, we formulate the SFC planning problem as an INLP problem, and a decoupled greedy algorithm is proposed to reduce the computational complexity.

A. Capacity Constraints

For any physical nodes, the computation resource consumptions can not exceed the computation capacity, which is expressed as

$$\text{C}_1 : \sum_{q\in Q} \sum_{f\in F} x_{f,n,q}\phi_{A,f} + \sum_{f\in F} a_{f,n}\phi_{B,f} \leq C_n, \quad \forall n \in N. \tag{13}$$

Similarly, for any physical links, the bandwidth resource consumptions can not exceed the bandwidth capacity, which is expressed as

$$\text{C}_2 : \sum_{q\in Q} \sum_{(i,j)\in E_q} y_{(n,m)}^{(i,j),q} B_q \leq B_{n,m}, \quad \forall (n,m) \in E. \tag{14}$$

B. Service Provision Constraints

For any service requests that are successfully served, all the required VNFs must be embedded to one and only one physical nodes. That is

$$\text{C}_3 : \sum_{n\in N} x_{f,n,q} = z_q, \quad \forall f \in \Pi_q,\ \forall q \in Q. \tag{15}$$

In this paper, we consider that the first VNF of the service function chain is embedded on the source node, and the last VNF is embedded on the destination node, which is expressed as

$$\begin{aligned}\text{C}_4 : x_{f_q^{\text{f}},s_q,q} = z_q, \quad \forall q \in Q, \\ \text{C}_5 : x_{f_q^{\text{l}},d_q,q} = z_q, \quad \forall q \in Q,\end{aligned} \tag{16}$$

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

Besides, the service request must be served before the deadline, which is

$$\text{C}_6 : \sum_{(i,j)\in E_q} \sum_{(n,m)\in E} y_{(n,m)}^{(i,j),q} D_{n,m} \leq D_q, \quad \forall q \in Q. \tag{17}$$

C. Flow Conservation Constraints

In the graph model, the flow conservation is an essential condition to build the routing path successfully. For any physical nodes, the traffic that flows in must be equal to the traffic that flows out, which is express as

$$\begin{aligned}\text{C}_7 : \sum_{m\in N} y_{(n,m)}^{(i,j),q} - \sum_{m\in N} y_{(m,n)}^{(i,j),q} = x_{i,n,q} - x_{j,n,q}, \\ \forall n \in N, \forall q \in Q, \forall (i,j) \in E_q.\end{aligned} \tag{18}$$

D. INLP Problem

In fact, the service request will be blocked due to two factors: 1) the physical resources are not enough, 2) the end-to-end delay exceeds the deadline. From the perspective of the network operator, the objective is to maximize the number of service requests that can be successfully served to achieve the highest revenue. Meanwhile, the costs of using communication and computation resources in both aerial and ground nodes should be minimized. Therefore, the objective function is designed as follows

$$\mathcal{U} = \sum_{q\in Q} z_q R_q - \sum_{f\in F} \phi_{B,f}\left(\sum_{n\in N_A}\alpha_{\mathrm{A}}^{\mathrm{cp}} a_{f,n} + \sum_{n\in N_G}\alpha_{\mathrm{G}}^{\mathrm{cp}} a_{f,n}\right)$$
$$-\sum_{q\in Q}\sum_{(i,j)\in E_q}\sum_{n\in N} B_q\left(\sum_{m\in N_A}\alpha_{\mathrm{A}}^{\mathrm{cm}} y_{(n,m)}^{(i,j),q} + \sum_{m\in N_G}\alpha_{\mathrm{G}}^{\mathrm{cm}} y_{(n,m)}^{(i,j),q}\right), \tag{19}$$

where R_q is the revenue weight for successfully serving a request. The physical meaning of 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. Note that only the basic computation resource costs are accounted in objective function because only this part can be saved by function sharing. Besides, only the communication resource consumptions of the ingress traffic are calculated due to the flow conservation.

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

$$\begin{aligned}(\mathbf{P1}):\ & \max_{\mathbf{x},\mathbf{y},\mathbf{a},\mathbf{z}} \mathcal{U}\\ & \text{s.t. } \mathrm{C}_1-\mathrm{C}_7,\\ & a_{f,n}=\mathbb{I}\left(\sum_{q\in Q} x_{f,n,q}\geq 1\right),\quad \forall f\in F, \forall n\in N,\\ & \mathbf{x},\mathbf{y},\mathbf{a},\mathbf{z}\in\{0,1\}.\end{aligned} \tag{20}$$

Note that there is a non-linear constraint due to the indicator function, and all the variables are integers in (P1). Thus, it is an INLP problem. Existing works have proven that the general SFC planning optimization problem is NP-hard [24]. We can obtain the exact solutions only when the network scale is small and the number of service requests is not too large, using the existing optimization solvers. However, the problem can become computationally prohibited as the network scale increases. One approach is to transform the primary problem into a linear programming problem which can be solved in polynomial time. Notice that the only non-linear constraint in (P1) is the indicator constraint, which can be transformed into a linear constraint using Lemma 1.

Lemma 1: The indicator constraint of Eq. (3) can be transformed into a linear constraint as

$$\sum_{q\in Q} x_{f,n,q} < |Q|a_{f,n} + 1,\quad \forall f\in F,\ \forall n\in N, \tag{21}$$

where $|Q|$ is the total number of service requests.

Proof: For a given VNF f and a given node n, the $a_{f,n}$ equals to 1 only when there is at least one $x_{f,n,q}$ that equals to 1 in all service requests, as shown in equation (3). Otherwise, $a_{f,n}$ equals to 0. In equation (21), when $\sum_{q\in Q} x_{f,n,q}\geq 1$, $a_{f,n}$ must equal to 1. Note that $\sum_{q\in Q} x_{f,n,q}$ can not exceed $|Q|$, thus equation (21) always holds when $a_{f,n}$ equals to 1. On the other hand, if $\sum_{q\in Q} x_{f,n,q}=0$, the equation (21) always holds whatever $a_{f,n}$ is. However, one of the objective in (P1) is to minimize the constant computation costs which are positively correlated to the value of $a_{f,n}$. Thus $a_{f,n}$ must equal to 0, when $\sum_{q\in Q} x_{f,n,q}=0$. The proof is finished.

Then, using the relaxations for the integer variables and the strict inequality constraint, we can transform (P1) into a linear programming (LP) problem as

$$\begin{aligned}(\mathbf{P2}):\ & \max_{\mathbf{x},\mathbf{y},\mathbf{a},\mathbf{z}} \mathcal{U}\\ & \text{s.t. } \mathrm{C}_1-\mathrm{C}_7,\\ & \sum_{q\in Q} x_{f,n,q}\leq |Q|a_{f,n}+1,\quad \forall f\in F,\ \forall n\in N,\\ & \mathbf{x},\mathbf{y},\mathbf{a},\mathbf{z}\in[0,1].\end{aligned} \tag{22}$$

(P2) can be solved by existing optimization solvers in polynomial time. The optimal value V_{opt} of the objective function in (P1) is upper bounded by the optimal value V_{opt}^* of the objective function in (P2). This is because the feasible domain of (P1) is a subset of the feasible domain of (P2). Note that we can derive the upper bound of the revenue that the network operator can obtain by solving (P2). However, the solution of (P2) may not be feasible for (P1). Therefore, an approximation algorithm is needed to obtain the sub-optimal solution to (P1) in polynomial time.

E. Decoupled Greedy Algorithm

In this part, a decoupled greedy algorithm with low complexity is proposed, which is composed of three sub-algorithms.

In Algorithm 1, the ground and aerial networks are provided with different priorities for resource utilization. At the start point of SFC planning, the network status, including network topology, available physical resources, and channel conditions, are known in advance. Besides, the service requests are arrived with different deadlines. Then, each service request is served by the stand-alone ground network first, using Algorithm 2 to find the best routing path for the traffic and Algorithm 3 to embed corresponding VNFs. This is because the physical resources in aerial network are much more scarce with higher cost weights. Thus, the service requests are preferred to be served by ground network to reduce the resource cost. However, if the service request is blocked by the stand-alone ground network, the aerial resources are leveraged. The priority mode for resource utilization can make the best use of the advantages of both ground and aerial networks to reduce the resource costs.

Another key idea is to decouple the VNF embedding and virtual link mapping processes by Algorithm 2 and Algorithm 3. The Algorithm 2 is a greedy algorithm to find the optimal path from source node to destination node for each service request based on Dijkstra algorithm. A feasible routing path for the traffic should satisfy the bandwidth resource requirement, and the total hopping delay can not exceed the deadline. Furthermore, the level of function sharing must be considered. We define the function sharing factor as

$$W_{n,q} = \sum_{f\in\Pi_q} A_{f,n},\quad \forall q\in Q,\ \forall n\in N, \tag{23}$$

Algorithm 1 Decoupled Greedy Algorithm for SFC Planning in SAGIN

```
1:  Initialize the network status and SFC service requests
2:  for each service request do
3:    Find optimal path using Algorithm 2 in Ground Network
4:    if there is at least one feasible path then
5:      Map the virtual links to the physical links in the optimal path
6:      Embed the VNFs into the nodes in the optimal path using Algorithm 3
7:      if Required resource exceeds available resource then
8:        The service request is blocked
9:      else
10:       The service request is successfully served
11:       Break loop
12:     end if
13:   else
14:     Find optimal path using Algorithm 2 in SAGIN
15:     if there is at least one feasible path then
16:       Do step 5 to 12
17:     else
18:       The service request is blocked
19:     end if
20:   end if
21: end for
```

where $A_{f,n}$ is defined as the VNF deployment status, which indicates that whether the instance of VNF f has been deployed on node n. Thus $A_{f,n}$ is required to be updated after each service request is served. For service request q, it can be noticed that the function sharing factor of node n is the total number of VNFs that have already been deployed. Based on this concept, we design the edge weight for physical links in the network graph. For any physical links $(n, m) \in E$, the weight is

$$Weight_q(n, m) = \frac{D_{n,m}\mathbb{I}(B^R_{n,m} - B_q)}{\exp(\rho(W_{n,q} + W_{m,q}))}, \quad \forall q \in Q, \quad (24)$$

where $B^R_{n,m}$ denotes the remaining bandwidth resources of physical link (n, m) after the last service request is served, and ρ is the aggregation factor which is a tunable nonnegative parameter to control the aggregation ratio. The design principles of the link weights are as follows:

1) The physical link with higher hopping delay has higher weight;
2) If the remaining bandwidth resources are not enough for the service request, the weight will be 0, meaning that the link is broken;
3) The weight is negatively correlated to the function sharing factor.

Our algorithm will choose a feasible path with minimal total weights. Intuitively, when ρ is large, the weight highly depends on the function sharing factor. Then, the algorithm prefers to choose the path that shares more VNFs, which increases the aggregation ratio. However, the delay requirements may not be satisfied. If ρ equals to 0, the weight only relies on the hopping delay. Then, the algorithm will choose the path with minimal hopping delay neglecting the VNFs sharing. In Algorithm 2, the proper value of ρ is searched by searching step δ to guarantee the delay performance.

Algorithm 2 Optimal Path Searching

```
Initialize aggregation factor $\rho$ and searching step $\delta$
while $\rho >= 0$ do
  Calculate the weight of edges in the network graph using expression (24)
  Find the shortest path using Dijkstra algorithm
  if the delay of the choosing path is less than the delay requirement then
    Return this path as optimal path
    Break the loop
  else
    $\rho = \rho - \delta$
  end if
end while
```

Algorithm 3 is used to embed the VNFs to the physical nodes along the optimal routing path in a greedy manner. For any physical node n in the network graph, we design the weight as

$$Weight_q(n) = W_{n,q}(C^R_n - C^{Req}_{q,n}), \quad \forall q \in Q, \quad (25)$$

where C^R_n denotes the remaining computation resources of physical node n after the last service request is served, and $C^{Req}_{q,n}$ is the required computation resources for embedding service request q on node n. It can be seen that the weight is determined by the function sharing factor and remaining computation resources. Thus, the node that has large remaining computation resources and has potentials to share more functions is tend to be selected. Initially, the set of potential nodes P_n contains all nodes in the optimal path. Then the algorithm will choose the node with maximal weight and embed all VNFs to this node. If the required computation resources exceed the capacity, the candidate node will be removed from potential nodes. The service request is blocked until P_n is empty.

Algorithm 3 VNFs Embedding

```
Initialize the set of potential nodes $P_n$ as all nodes in the optimal path obtained by Algorithm 2
while $P_n$ is not empty do
  Calculate the weight of the nodes in $P_n$ using expression (25)
  Choose the candidate node with maximal weight
  if Resource is enough then
    Embed all VNFs to the candidate node
    Break the loop
  else
    Remove the candidate node from $P_n$
  end if
end while
```

TABLE II
SIMULATION SETTINGS

Parameter	Value	Distribution
N_A	7	-
N_G	1	-
C_n	$[800, 1000]$ Tflops	Uniform
$B_{n,m}$	$[500, 600]$ Mbps	Uniform
$D_{n,m}$	$[10, 15]$ ms	Uniform
$\|F\|$	6	-
α_{cm}^{A}	0.5	-
α_{cm}^{G}	3	-
α_{cp}^{A}	10	-
α_{cp}^{G}	30	-
R_q	500	-

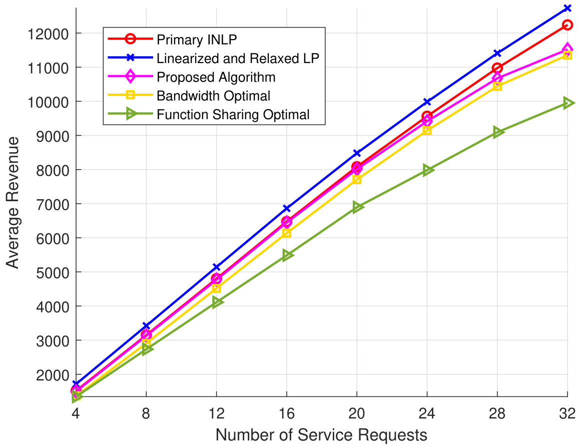


Fig. 3. Performance of different algorithms compared with exact optimal solutions and the upper bound.

For Algorithm 2, the complexity mainly results from the $while$ loop and Dijkstra algorithm. We use $|N|$ to denote the total number of network nodes. Then, the complexity of Algorithm 2 is $\mathcal{O}(\lceil \frac{\rho}{\delta} \rceil |N|^2)$. Since the number of nodes in P_n cannot exceed $|N|$, the worst case of Algorithm 3 has a complexity of $|N|^2$. The complexity of $while$ loop of Algorithm 1 depends on the total number of service requests, which is denoted by $|Q|$. Thus, the complexity of our proposed algorithm is $\mathcal{O}(|Q| \lceil \frac{\rho}{\delta} \rceil |N|^2)$.

V. PERFORMANCE EVALUATION

In this section, we present the simulation results to evaluate our proposed algorithm. The main simulation parameters [21], [22], [24] are listed in Table II.

We first evaluate the performance of the proposed algorithm and compare it with the exact solutions of primary problem (P1) obtained by *Solving Constraint Integer Programs* (SCIP) solver [31] and the solutions obtained by solving (P2), as shown in Fig. 3. We also modify the algorithms developed in [21] and [25] to fit in our scenario (i.e. *Function Sharing Optimal* [21] and *Bandwidth Optimal* [25]). Specifically, the *Function Sharing Optimal* algorithm shares VNFs as many as possible to reduce the number of occupied physical nodes. The *Bandwidth Optimal* algorithm greedily deploys the VNF one by one to minimize the bandwidth utilization. The performance is measured by the average revenue which is the total revenue earned by successfully serving the service requests minus the total communiaction and computation resource costs. It can be seen that the *Function Sharing Optimal* algorithm has the worst performance. This is because the physical links become saturated very soon, resulting in inevitable drops of service requests. Our proposed algorithm also performs better than the *Bandwidth Optimal* algorithm because the latter can not optimize the tradeoff between communication and computation resource costs and can not handle the service priorities of aerial and ground networks. We also find that the proposed algorithm achieves the near-optimal performance when the number of service requests is not too large. However, the gap increases as the number of service requests increases. This is because that some service requests can be blocked in our proposed algorithm when the network resources are gradually approaching the bottleneck.

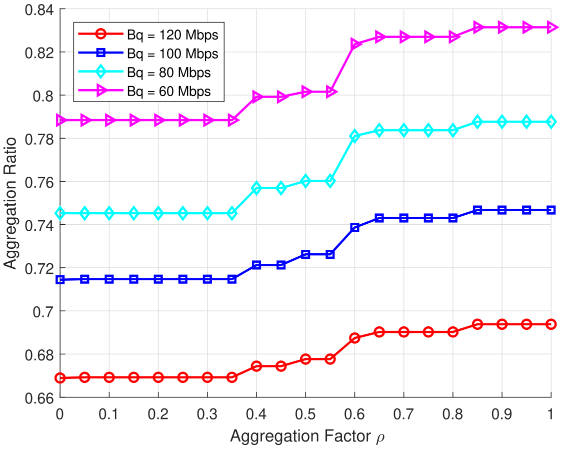


Fig. 4. Aggregation ratio versus aggregation factor ρ in proposed algorithm under different bandwidth requirement B_q.

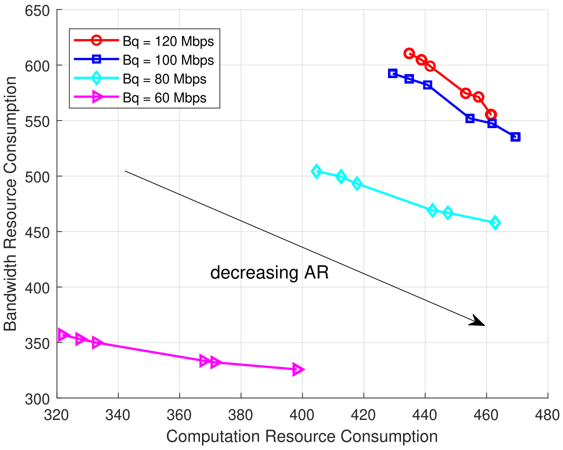


Fig. 5. Tradeoff between bandwidth resource consumption and computation resource consumption under different bandwidth requirement B_q.

Fig. 4 validates that the AR is controlled by the aggregation factor ρ in our proposed algorithm. We can observe that the AR increase as ρ increases, because the function sharing factor gradually dominates the choice of optimal routing path. Besides, the AR is influenced by the bandwidth requirement of service request q. Specifically, a higher bandwidth requirement B_q indicates a lower AR, because the physical links are more likely to be saturated and additional routing paths are required to be explored. Correspondingly, the tradeoff between communication and computation resource costs is

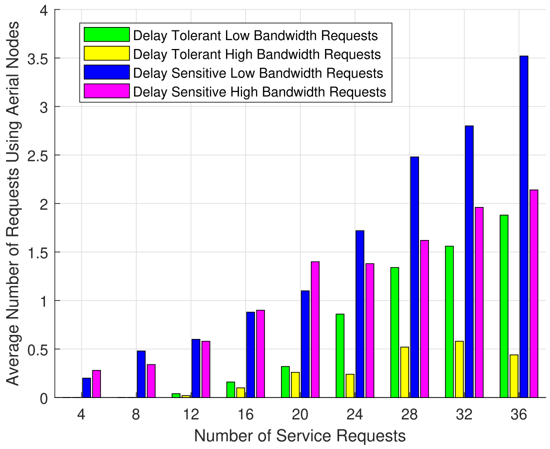


Fig. 6. Resource utilization for different types of service requests.

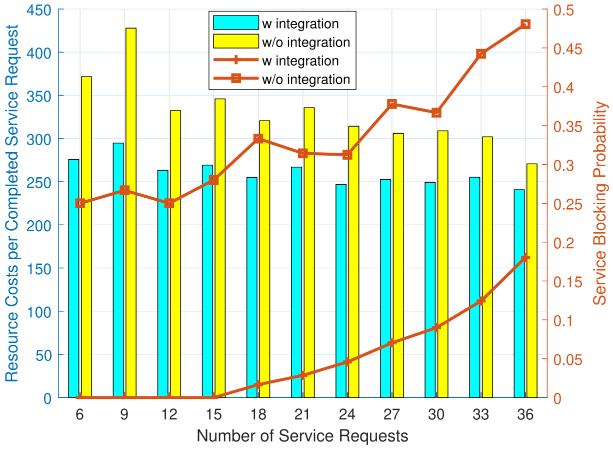


Fig. 7. Comparison between the network with integration and without integration. The bars correspond to the resource costs while the lines correspond to the service blocking probability.

shown in Fig. 5. By tuning AR, we can significantly reduce the communication resource consumptions at the expense of the increased computation resource consumptions, and vice versa. Specifically, in the case of a small B_q, as illustrated by the magenta right-pointing triangle line, consuming a small amount of communication resources can save a large amount of computation resources when we increase AR. Reversely, in the case of a large B_q, as illustrated by the red circle line, a large amount of communication resources can be saved at the expense of slightly increasing computation resource consumptions when we decrease AR.

Fig. 6 shows the resource utilization of aerial nodes for different types of service requests. Note that the aerial nodes mainly provide services for delay sensitive requests, because the traffic only experience two communication hops which significantly reduces the hopping delay. Besides, low bandwidth requests are more like to be served due to the limited available bandwidth resources.

We compare the resource costs and the blocking probability between SAGIN and the stand-alone ground or aerial networks in Fig. 7. It can be seen that the cost per completed service requests is significantly reduced by 12.5% to 45.1% via integrating the ground and air networks, indicating that higher resource utilization efficiency is achieved. Furthermore, the blocking probability is also reduced by integration, which potentially increase the revenue earned by the network operators.

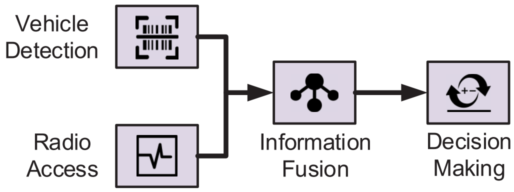


Fig. 8. Two function chains for multiple intersection traffic scheduling.

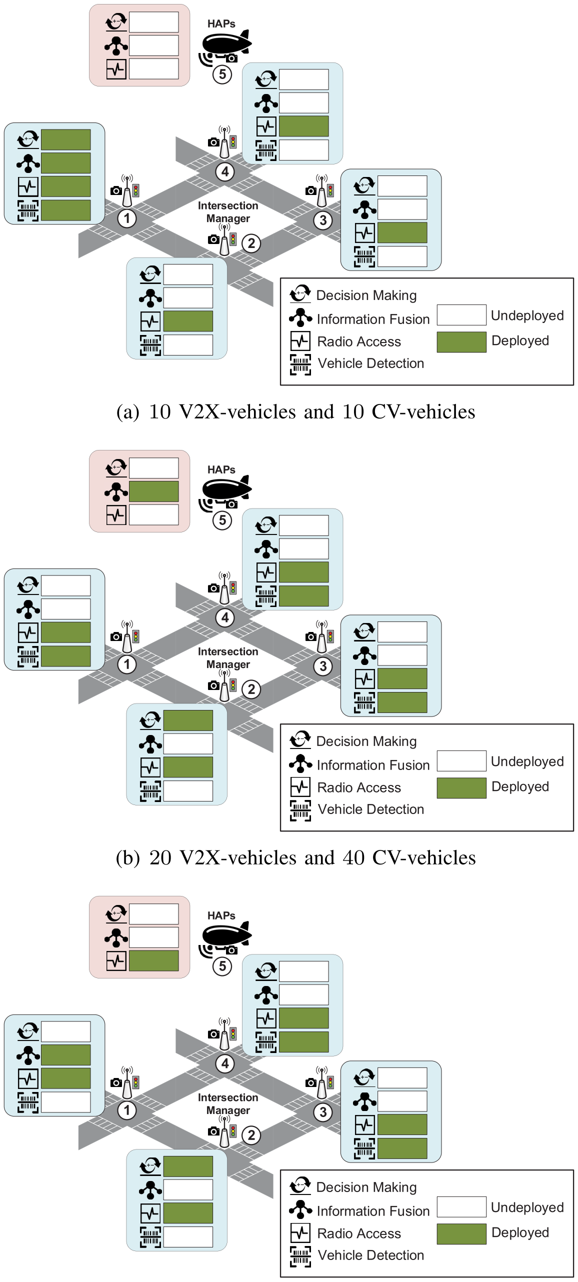
VI. Case Study: Multiple Intersection Traffic Scheduling

In this section, we provide a case study on multiple intersection traffic scheduling to elaborate how our proposed SFC-based network reconfiguration framework works under a practical application. Extensive investigations have been done to improve the road safety and traffic efficiency of the intersections [32]–[34]. Basically, a centralized scheduler, i.e. the intersection manager, is responsible for allocating time-space resources of the intersection to the vehicles based on the vehicle and traffic informations, such as location, velocity, and acceleration. There are two approaches for the vehicle detection and information collection, i.e. the computer vision (CV) approach [35] and the vehicle-to-everything (V2X) communication approach [36], [37]. In our scenario, we have two types of vehicles:

1) V2X-vehicle: the vehicles that can collect the vehicle information by powerful local sensor network and upload the information to the intersection manager using V2X communications;
2) CV-vehicle: the vehicles that are directly detected by the intersection manager using computer vision technology.

Therefore, we have two function chains in our scenario, as shown in Fig. 8. Note that *Vehicle Detection* function is responsible for computationally intensive target recognition and tracking, which is not suitable to be deployed on the aerial node. *Information Fusion* function can be deployed on only one physical node, because it is responsible for gathering and processing the vehicle information from all involving vehicles. *Radio Access* function mainly consumes communication resources for data transmission but still consumes a small amount of constant computation resources for the VNF installation and maintenance. Note that both the two chains (V2X packets update and compute-vision based vehicle detection) have stable data flow. Thus, the bandwidth requirements of the two service requests are still assumed to be constants.

We consider a 2×2 grid of intersections with 4 intersection managers and 1 HAP, as shown in Fig. 9. The vehicles are uniformly distributed at each intersection. In this scenario, the vehicles are regarded as users. The VNFs can be flexibly deployed on the intersection managers or the HAP, which has low mobilities. Thus, the topology of the network is assumed to be fixed. The V2X-vehicles can only access the nearest intersection manager or the HAP. We assume that the bandwidth of the backbone network for the data transmissions



(a) 10 V2X-vehicles and 10 CV-vehicles

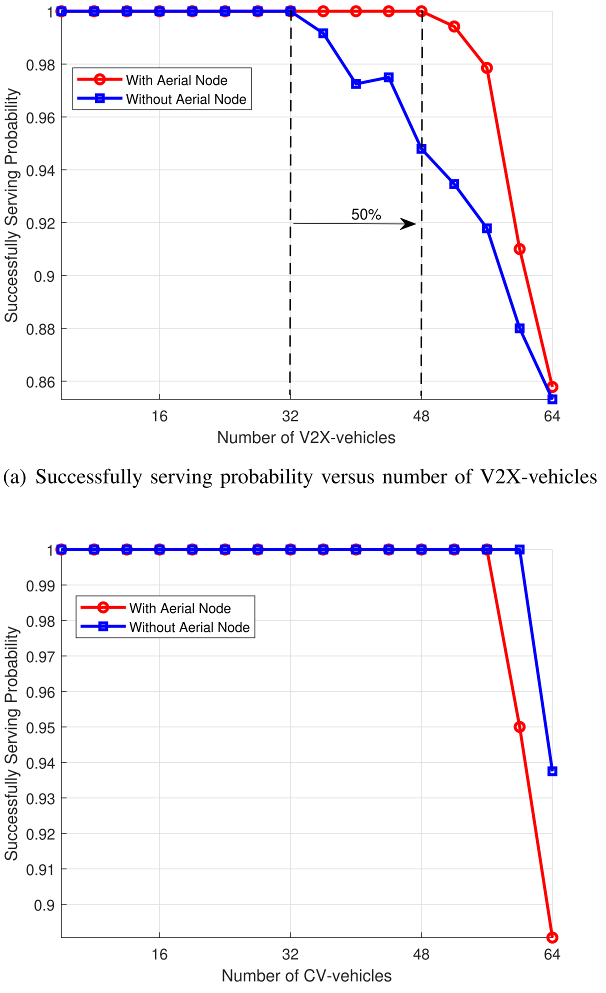
(b) 20 V2X-vehicles and 40 CV-vehicles

(c) 40 V2X-vehicles and 20 CV-vehicles

Fig. 9. Function placement on a 2×2 grid of intersections with one HAP under different numbers of vehicles.

among the intersection managers and the HAP is guaranteed [41]. Thus, we can focus on deploying the functions on physical nodes without considering the traffic routing problem. Besides, the communication resource limitation mainly results from the radio access of the V2X-vehicles. For each intersection manager or HAP, the communication resources, i.e. the bandwidth resources, are uniformly allocated the V2X-vehicles connecting to it.

The parameter setting [38]–[40] is provided as follows. The computation capacity is 25 GHz for each node, where we use CPU cycles as the unit of computation resources. The bandwidth capacity is 40 Mbps for each node. The packet size is 10 KB. The delay requirement is 20 ms. The basic computation resource consumptions for *Vehicle Detection*, *Radio Access*, *Information Fusion*, and *Decision Making* are randomly selected within $[2.5, 3.5]$ GHz, $[0.4, 0.6]$ GHz, $[0.8, 1.2]$ GHz and $[0.8, 1.2]$ GHz for different service requests, respectively. The corresponding additional computation resource consumptions per service request are 1 GHz, 0 GHz, 0.3 GHz and 0.1 GHz, respectively. The simulation is done in Matlab and the INLP problem is solved by the SCIP solver.



(a) Successfully serving probability versus number of V2X-vehicles

(b) Successfully serving probability versus number of CV-vehicles

Fig. 10. Comparison of successfully serving probability between the network with aerial node and the network without aerial node under different number of vehicles.

Fig. 9 shows how the functions deployed on physical nodes under different system conditions. It can be observed that, when the number of vehicles is small, as shown in Fig. 9(a), the functions are deployed on one intersection manager for function sharing and the HAP is not used. This is because the resources in ground network are sufficient and the computation resources can be saved by centralized function deployment. As the number of vehicles increases, the aerial nodes are used to relieve the traffic burden and the types of VNFs that deployed on aerial nodes depend on the bottleneck of network resources. Specifically, when the number of CV-vehicles is large, as shown in Fig. 9(b), the *Information Fusion* is deployed on the HAP and each intersection manager is deployed with at least one *Vehicle Detection* or *Decision Making*. This is because the computation resources become the bottleneck of the network and the computation resources of the HAP are utilized. When the number of V2X-vehicles is large, as shown in Fig. 9(c), the *Radio Acess* is deployed on the HAP. This is because the communication resources become the bottleneck of the network and the communication resources of the HAP are utilized.

We also compare the successfully serving probability between SAGIN and the stand-alone ground network, as shown in Fig 10. For fairness, the two networks have the same amount of computation and communication resources. It can be observed that when the number of V2X-vehicles is large, the successfully serving probability is significantly increased thanks to the aerial node, and the number of V2X-vehicles the network can accommodate is increased by 50% in SAGIN. While the performance of SAGIN is slightly worse than the stand-alone ground network as the number of CV-vehicles increases. This is because the computation resources can be allocated in a more centralized manner in stand-alone ground network. These results indicate that V2X enabled vehicles are really welcome in SAGIN.

VII. Conclusion

In this work, we propose to apply SFC in large-scale heterogeneous networks, i.e. the SAGIN, and investigate the SFC planning problem which is formulated as an INLP. Then a heuristic decoupled greedy algorithm is proposed to solve it, which is shown to achieve near-optimal performance. Different from most existing works, the heuristic is based on different features of physical nodes, in which the weights of network nodes and links are carefully designed jointly considering the resource utilizations and QoS guarantees. We also focus on balancing the communication and computation resource consumptions, which is vital for SAGIN due to the uneven distributions of the resources and dynamic service demands. Therefore, AR is proposed to tackle the tradeoff between communication and computation resource costs. 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 small, a large amount of communication resources can be saved at the expense of slightly increasing computation resource consumptions by decreasing AR. It also shows that the SAGIN outperform the stand-alone networks by saving 12.5% to 45.1% total resource costs per completed service request and significantly reduce the service blocking probability. At last, the case study on multiple intersection traffic scheduling validates our proposed SFC-based reconfigurable service provision framework under a practical application in SAGIN. As a future work, we plan to optimize the SFC migration when the network conditions change during the service procedures, which will further cope with the challenges of dynamic SAGIN.

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, 3rd Quart., 2016.

[2] W. Qi, W. Hou, L. Guo, Q. Song, and A. Jamalipour, “A unified routing framework for integrated space/air information networks,” *IEEE Access*, vol. 4, pp. 7084–7103, 2016.

[3] J. Du, C. Jiang, Q. Guo, M. Guizani, and Y. Ren, “Cooperative Earth observation through complex space information networks,” *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 136–144, Apr. 2016.

[4] Y. Wang *et al.*, “Multi-resource coordinate scheduling for Earth observation in space information networks,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 268–279, Feb. 2018.

[5] Y. Zhu *et al.*, “Utilization and analysis of resource mobility in space information networks,” *J. Commun. Inf. Netw.*, vol. 4, no. 1, pp. 67–77, Mar. 2019.

[6] X. Cao, P. Yang, M. Alzenad, X. Xi, D. Wu, and H. Yanikomeroglu, “Airborne communication networks: A survey,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1907–1926, Sep. 2018.

[7] K. Kwak *et al.*, “Airborne network evaluation: Challenges and high fidelity emulation solution,” *IEEE Commun. Mag.*, vol. 52, no. 10, pp. 30–36, Oct. 2014.

[8] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in UAV communication networks,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[9] G. Cocco, N. Alagha, and C. Ibars, “Cooperative coverage extension in vehicular land mobile satellite networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 5995–6009, Aug. 2016.

[10] M. Casoni, C. A. Grazia, M. Klapez, N. Patriciello, A. Amditis, and E. Sdongos, “Integration of satellite and LTE for disaster recovery,” *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 47–53, Mar. 2015.

[11] J. Jiao, S. Wu, Y. Sun, Y. Wang, and Q. Zhang, “Power allocation optimization of multibeam high-throughput satellite communication systems,” *J. Commun. Inf. Netw.*, vol. 4, no. 1, pp. 33–41, Mar. 2019.

[12] A. M. Hayajneh, S. A. R. Zaidi, D. C. McLernon, M. Di Renzo, and M. Ghogho, “Performance analysis of UAV enabled disaster recovery networks: A stochastic geometric framework based on cluster processes,” *IEEE Access*, vol. 6, pp. 26215–26230, 2018.

[13] 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, 2018.

[14] N. Cheng *et al.*, “Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities,” *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[15] 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.

[16] 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.

[17] S. Zhou, G. Wang, S. Zhang, Z. Niu, and X. S. Shen, “Bidirectional mission offloading for agile space-air-ground integrated networks,” *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 38–45, Apr. 2019.

[18] G. Mirjalily and Z. Luo, “Optimal network function virtualization and service function chaining: A survey,” *Chin. J. Electron.*, vol. 27, no. 4, pp. 704–717, Jul. 2018.

[19] Y. Xie, Z. Liu, S. Wang, and Y. Wang, “Service function chaining resource allocation: A survey,” 2016, *arXiv:1608.00095*. [Online]. Available: http://arxiv.org/abs/1608.00095

[20] J. Gil Herrera and J. F. Botero, “Resource allocation in NFV: A comprehensive survey,” *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[21] 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, Jul. 2018.

[22] 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.

[23] 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 Manage.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[24] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, “Orchestrating virtualized network functions,” *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[25] M. T. Beck and J. F. Botero, “Coordinated allocation of service function chains,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2014, pp. 1–6.

[26] A. Mohammed, A. Mehmood, F.-N. Pavlidou, and M. Mohorcic, “The role of high-altitude platforms (HAPs) in the global wireless connectivity,” *Proc. IEEE*, vol. 99, no. 11, pp. 1939–1953, Nov. 2011.

[27] F. Dong, H. Han, X. Gong, J. Wang, and H. Li, "A constellation design methodology based on QoS and user demand in high-altitude platform broadband networks," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2384–2397, Dec. 2016.

[28] X. Zhu, C. Jiang, L. Yin, L. Kuang, N. Ge, and J. Lu, "Cooperative multigroup multicast transmission in integrated terrestrial-satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 981–992, May 2018.

[29] E. Lagunas, S. K. Sharma, S. Maleki, S. Chatzinotas, and B. Ottersten, "Resource allocation for cognitive satellite communications with incumbent terrestrial networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 1, no. 3, pp. 305–317, Sep. 2015.

[30] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, "Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202–2217, Oct. 2018.

[31] SCIP Optimization. *SCIP Doxygen Documentation*. Accessed: Apr. 15, 2019. [Online]. Available: https://scip.zib.de

[32] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.

[33] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[34] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.

[35] M. S. Shirazi and B. T. Morris, "Looking at intersections: A survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 4–24, Jan. 2017.

[36] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[37] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 26–32, Sep. 2018.

[38] *Study on enhancement of 3GPP Support for 5G V2X Services (Release 16)*, document 3GPP TR 22.886 V16.1.0, Sep. 2018.

[39] M. A. S. Kamal, J.-I. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1136–1147, Jun. 2015.

[40] N. Sundaram, "Making computer vision computationally efficient," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., UC Berkeley, Berkeley, CA, USA, 2012.

[41] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, Sep. 2018.

Guangchao Wang received the B.S. degree in communications engineering from Beijing Jiaotong University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree in electronic engineering with Tsinghua University, Beijing. His research interests include space-air-ground integrated network reconfiguration, service function chaining, and UAV-aided traffic offloading.

Sheng Zhou (Member, IEEE) received the B.E. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2005 and 2011, respectively. In 2010, he was a Visiting Student with the Wireless System Lab, Department of Electrical Engineering, Stanford University, Stanford, CA, USA. From 2014 to 2015, he was a Visiting Researcher with the Central Research Lab, Hitachi Ltd., Japan. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. His research interests include cross-layer design for multiple antenna systems, mobile edge computing, vehicular networks, and green wireless communications. He received the IEEE ComSoc Asia Pacific Board Outstanding Young Researcher Award in 2017.

Shan Zhang (Member, IEEE) received the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2016. She was a Post-Doctoral Fellow with the Department of Electronical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from 2016 to 2017. She is currently an Assistant Professor with the School of Computer Science and Engineering, Beihang University, Beijing. Her research interests include mobile edge computing, wireless network virtualization, and intelligent management. She received the Best Paper Award at the Asia-Pacific Conference on Communication in 2013.

Zhisheng Niu (Fellow, IEEE) received the B.E. degree from Beijing Jiaotong University, China, in 1985, and the M.E. and D.E. degrees from the Toyohashi University of Technology, Japan, in 1989 and 1992, respectively.

From 1992 to 1994, he was with Fujitsu Laboratories Ltd., Japan. In 1994, he joined Tsinghua University, Beijing, China, where he is currently a Professor with the Department of Electronic Engineering. His major research interests include queuing theory, traffic engineering, mobile Internet, radio resource management of wireless networks, and green communication and networks. He is a fellow of IEICE. He received the Outstanding Young Researcher Award from the Natural Science Foundation of China in 2009 and the Best Paper Award from the IEEE Communication Society Asia Pacific Board in 2013. He has served as the Chair for the Emerging Technologies Committee from 2014 to 2015, the Director for the Conference Publications from 2010 to 2011, and the Director for the Asia Pacific Board in the IEEE Communication Society, from 2008 to 2009. He is currently serving as the Director of the Online Contents from 2018 to 2019 and as an Area Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He was also selected as a Distinguished Lecturer of the IEEE Communication Society from 2012 to 2015 and the IEEE Vehicular Technologies Society from 2014 to 2018.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, social networks, 5G and beyond, and vehicular ad hoc and sensor networks. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the R. A. Fessenden Award in 2019 from the IEEE, Canada, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015, and the Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award five times from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He has served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, the IEEE Infocom'14, the IEEE VTC'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL and the Vice President on Publications of the IEEE Communications Society.