Service-Oriented Dynamic Resource Slicing and Optimization for Space-Air-Ground Integrated Vehicular Networks

Feng Lyu, *Member, IEEE*, Peng Yang, *Member, IEEE*, Huaqing Wu, *Graduate Student Member, IEEE*, Conghao Zhou, *Graduate Student Member, IEEE*, Ju Ren, *Member, IEEE*, Yaoxue Zhang, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— In this paper, we study Space-Air-Ground integrated Vehicular Network (SAGVN), and propose an online control framework to dynamically slice the SAG spectrum resource for isolated vehicular services provisioning. In particular, at a given time slot, the system makes online decisions on the request admission and scheduling, UAV dispatching, and resource slicing for different services. To characterize the impact of those parameters, we construct a time-averaged queue stability criteria by taking queue backlogs of all services into consideration, and formulate a system revenue function which incorporates the time-averaged system throughput and UAV dispatching cost. The objective is to maximize the system revenue while stabilizing the time-averaged queue, which falls into the scope of Lyapunov optimization theory. By bounding the drift-plus-penalty, the original problem can be decoupled into four independent subproblems, each of which is readily solved. The merits of our control framework are three-fold: 1) the system is able to admit and process as many requests as possible (i.e., maximizing the time-averaged throughput); 2) the time-averaged UAV dispatching cost is minimized; and 3) service queues are stabilized in the long-term. Extensive simulations are carried out, and the results demonstrate that the control framework can effectively achieve the system revenue maximization and queueing stabilization. Moreover, it can balance the trade-off among system throughput, UAV dispatching cost, and queueing states via parameter tuning. Compared with the fixed slicing, our dynamic slicing can react to the vehicular environment rapidly and achieve an average 26% of throughput improvement.**

***Index Terms*— Space-air-ground integrated vehicular network, dynamic slicing, isolated service provisioning.**

Manuscript received 21 July 2020; revised 17 January 2021; accepted 11 March 2021. Date of publication 15 April 2021; date of current version 8 July 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62002389, Grant 62001180, Grant 62072472, and Grant U19A2067; in part by the National Key Research and Development Program of China under Grant 2019YFA0706403; in part by the Natural Science Foundation of Hunan Province of China under Grant 2020JJ2050; in part by the 111 Project under Grant B18059; in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2018QNRC001; in part by the Young Talents Plan of Hunan Province of China under Grant 2019RS2001; and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. This article was presented in part at IEEE VTC2020-Fall. The Associate Editor for this article was R. Malekian. *(Corresponding author: Peng Yang.)*

Feng Lyu and Ju Ren are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: fenglyu@csu.edu.cn; renju@csu.edu.cn).

Peng Yang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yangpeng@hust.edu.cn).

Huaqing Wu, Conghao Zhou, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: h272wu@uwaterloo.ca; c89zhou@uwaterloo.ca; sshen@uwaterloo.ca).

Yaoxue Zhang is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China, and also with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: zhangyx@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2021.3070542

I. Introduction

In the era of automated and connected driving, vehicular communications become prevailing in many cases, including vehicular platooning, extended sensing, automated driving, and remote driving [2]. To support those automotive services, an extremely versatile network is required to simultaneously guarantee ultra-reliability and low latency communications (URLLC), deliver bandwidth-consuming contents, and enable massive connections [3]–[7]. To this end, Space-Air-Ground integrated Vehicular Network (SAGVN) has been proposed as a promising networking architecture. In addition to the terrestrial network, the space satellites and aerial network (i.e., unmanned aerial vehicles (UAVs)) are leveraged to complement the terrestrial communications. Particularly, as the terrestrial network has coverage holes due to the considerable deployment and maintenance cost, the satellite network can assist in providing full network connectivity anywhere and anytime. On the other hand, with fully controllable mobility and altitude, UAVs can be flexibly deployed to keep pace with the dynamic vehicular environments [8]–[11].

However, as different vehicular services share the underlying spectrum resource, the Quality of Service (QoS) of vehicular applications are difficult to be guaranteed since they have distinct traffic features and can affect each other significantly, especially when the network resource is insufficient. For instance, bandwidth-extensive services are very likely to block the channel, resulting in packet drop or transmission delay for periodical lightweight URLLC services. Recently, network softwarization and slicing have been identified as an indispensable component for the next-generation network, which can shift the one-size-fits-all network to a set of dedicated networks, each customized to serve one type of service [12]–[15]. Generally, network slicing is the embodiment of running multiple dedicated networks on a shared physical infrastructure, whereby slices are completely isolated

from each other, and each slice allows the implementation of service-oriented functionalities, providing a mix of capabilities to meet diverse QoS demands. However, to achieve efficient slicing in SAGVN is quite challenging due to the following three significant reasons. First, how to slice the resource to accommodate different services is intricate as SAG spectrum resources are multi-dimensional (i.e., satellite, UAVs, cellular) and heterogeneous (with distinct access control technologies and channel conditions) while each slice is customized for a specific service, calling for differentiated spectrum provisioning. Second, in vehicular environments, a slicing policy may become inefficient shortly, since vehicular mobility results in dramatic topology variation, and the requests are also unpredictable and possibly bursty. Hence, a deterministic hard slicing strategy can hardly accommodate the time-varying service requests. Therefore, *dynamic slicing* is requisite to keep pace with fast changes of the vehicular environment. Third, for the standby UAV resources, when and how many it should be used is non-trivial. On the one hand, if excessive UAVs are dispatched at an off-peak time, UAV resources will be wasted. On the other hand, in the case of high-density scenario, if insufficient UAVs are dispatched, service QoS can be severely deteriorated.

In the literature, there have been researches on the SAG integrated network and network slicing independently. Compared to the terrestrial network, satellite and aerial networks have different features that can assist the terrestrial network. Therefore, most existing works focus on the SAG network architecture design and resource management [15]–[17], air-to-ground channel modeling [18], aerial network deployment and trajectory design [19], [20], as well as energy efficiency and throughput maximization [21], [22], etc. Due to the promising merits of SAG, in this paper, we concentrate on the network slicing under a given SAGVN scenario, which however is quite different from previous works. For network slicing, there are very limited studies, and they are unable to resolve above challenges. Particularly, early works [23], [24] discussed V2X slicing at high level without giving detailed implementation techniques. Although Zhang *et al.* [25] have considered the slicing under air-ground integrated vehicular network, they focused on content pushing and caching, i.e., slicing storage resource, and providing theoretical analysis for each slice. In contrast, we investigate the SAG spectrum resource slicing, and concentrate on dynamic slicing scheme design towards service-oriented performance enhancement.

In this paper, to adapt to the varying vehicular environments, we propose an online control framework to dynamically slice the SAG spectrum resource in order to achieve isolated service provisioning. With the spectrum resource softwarization and slicing, each type of services would be processed within an independent queue. Considering a time-slotted system, at each time slot, the framework has to make the following online decisions: 1) how many requests from different services should be admitted into the system; 2) for each type of admission requests, how to dispatch them into satellite, UAV and cellular resource planes for processing; 3) how many UAVs should be dispatched to supplement the spectrum resource if necessary; 4) for service queues at each resource plane, how to slice the spectrum for them in order to reduce the queue backlogs. The upshot of the system falls into three folds: 1) the system should admit and process as many service requests as possible (i.e., maximizing the system throughput); 2) after guaranteeing the performance, the UAV dispatching cost should be minimized; 3) for each type of services, the queue backlog should be stable and fairly bounded. Under the dynamic vehicular environments, the service requests are *stochastic* and may be *bursty*, making the objectives intractable directly. To this end, we adopt the Lyapunov optimization techniques to make the online decisions towards the long-term system revenue maximization. Specifically, we first take the time-averaged queue backlogs of all services into consideration and construct a quadratic Lyapunov function. We then formulate a system *revenue maximization* problem which incorporates the time-averaged system throughput and UAV dispatching cost. To maximize the system revenue, an inverse penalty function is constructed. In order to stabilize the system while minimizing the time-averaged penalty, the drift-plus-penalty should be bounded to the minimum. Based on the derived drift-plus-penalty bound, at each time slot, the controlling problem is then decomposed into several deterministic subproblems including auxiliary variable determination, request admission and scheduling, UAV dispatching, and resource slicing.

For performance evaluation, we generate the realistic vehicular trace by simulation tool, and emulate diverse and differentiated communication services, including lightweight periodical communication, medium constant downloading, and heavy yet stochastic entertainment. Extensive simulations are conducted and results show that: 1) the queueing stability can be maintained; 2) by tuning the control parameters, the trade-offs between the system stability and revenue, and between the system throughput and UAV dispatching cost, are flexible to be balanced; and 3) compared with the fixed slicing scheme, an average 26% of throughput improvement can be achieved by the proposed dynamic slicing scheme.

We highlight our major contributions in this paper as follows.

- We propose a dynamic slicing framework, by which, without prior knowledge of services arrival, the system can conduct the admission control, request scheduling, UAV dispatching and resource slicing in real time for isolated service provisioning.
- Based on the Lyapunov optimization, both the long-term system stability and revenue maximization can be achieved simultaneously. Meanwhile, by tuning the control parameters, the UAV usage bound and the trade-off between the system stability and revenue are controllable.
- For isolated services, by giving different utility functions, differentiated service provisioning is enabled. In addition, to guarantee ultra-low delay for URLLC services, an upgraded dynamic slicing scheme is proposed, which can process the URLLC service requests with the first priority. It can effectively bound the URLLC queue within a small size with negligible throughput degradation since the URLLC service traffic is usually lightweight.

We organize the remainder of this paper as follows. Section II gives the related work. We describe the system

model and SAG resource management architecture in Section III. The work flow of dynamic slicing and problem formulation are elaborated in Section IV. The dynamic slicing algorithm for the system stabilization and revenue maximization is given in Section V. Simulation setup and performance evaluation are carried out in Section VI. Finally, we conclude the paper and direct the future work in Section VII.

II. Related Work

A. Space-Air-Ground Integrated Network

To meet demands of ever increasing wireless services, SAG integrated network is proposed in recent years to assist the terrestrial network in providing seamless wireless coverage, massive connections, URLLC services, etc.

Liu *et al.* [9] organized a survey on SAG network, in which they first provided an overview of the available related research works, then discussed the applicable SAG network architectures, and finally pointed out some technical challenges and future research directions. Zhang *et al.* [15] and Zhou *et al.* [16] revisited the SAG architecture at high level; specifically, Zhang *et al.* proposed a software defined space-air-ground integrated network architecture, which aims at providing various vehicular services in a cost-effective, seamless, and efficient manner, while Zhou *et al.* proposed a programmable, scalable, and flexible framework, named as SAGECELL, in order to combine SAG resources in a complementary fashion to well match the underlying dynamic traffic demands. Considering the different features of three network segments in SAG integrated work, Kato *et al.* [17] advocated the usage of Artificial Intelligence (AI) technique to manage the network resources in order to enhance the network performance, where they provided an example of exploiting deep learning to conduct satellite traffic control. Likewise, Cheng *et al.* [11] presented an SAG integrated network aided edge/cloud computing architecture to assist in offloading the computation-intensive applications for remote Internet of things (IoT) users, where they adopted a deep reinforcement learning-based approach to learn the optimal offloading policy. For aerial networking, Al-Hourani *et al.* [18] investigated the air-to-ground channel modeling and provided a mathematical model in order to obtain optimal network altitude that maximizes the coverage on the ground. In the work [19], the authors proposed the UAV-enabled caching for vehicular networks and jointly optimized the UAV trajectory and caching scheme performance by adopting a learning-based approach. Wu *et al.* jointly optimized the user scheduling and association, UAV trajectories, and transmission power, to maximize the average rate among all users [20]. In addition, the UAV endurance limitation and UAV-aided relay system were also investigated in [21] and [22], respectively.

These studies have demonstrated the potential of SAG integrated network, which is quite promising to support future connected and automated vehicle services. Therefore, in this paper, we focus on a specific SAGVN scenario, and concentrate on the network slicing under the case, the scheme of which can adapt to the rapidly-evolving network architecture.

B. Network Slicing

Network slicing is an emerging softwarization technique to enable resource isolation for dedicated service provisioning. Particularly, Campolo *et al.* [23] justified the role of network slicing to enable the isolated V2X services delivery and provided some high-level suggestions of designing dedicated V2X slices. In the work [24], after revisiting the key characteristics and requirements of V2X services, they further presented a set of design guidelines. However, in both pieces of work, only high-level network slicing directions are discussed while no technical details are given. To shape next-generation smart factories, Taleb *et al.* [14] proposed a 5G-based network slicing framework in order to accommodate the requirements of Industry 4.0. In the works [26]–[28], network slicing is considered under the terrestrial heterogeneous radio access network. Specifically, Ye *et al.* [26] formulated a network utility maximization problem to determine the optimal bandwidth slicing ratios and BS-device association patterns, and transformed it to a biconcave maximization problem. Bahlke *et al.* [27] jointly optimized the resource distribution among network slices, the allocation of cells to operate on different slices, and the association of users to cells. They then proposed a convex inner approximation to solve the formulated problem since it is computationally intractable. Kim *et al.* [28] considered a multiple-infrastructure-provider scenario and formulated a slice allocation problem to guarantee the performance of users (with different service requests). A matching game theory framework is then leveraged to tackle the problem.

For above works, as the targeted services are quite different from the vehicular services, where the user mobility and fast topology variation are not considered in the scenarios, they are unable to work efficiently in SAGVN. In the work [25], Zhang *et al.* considered the slicing under air-ground integrated vehicular network, the scenario of which is similar to ours. However, they concentrate on content pushing and caching, i.e., slicing storage resource, and providing the theoretical analysis for each slice. Differently, we focus on SAG spectrum resource slicing and push the theoretical analysis further by designing the dynamic slicing scheme in accordance with the time-varying vehicular environment. In our previous work [1], we have demonstrated the efficiency of the dynamic slicing framework. In this paper, we further improve it by embodying the system models, completing the technical methodology, extending the experiments, and including a new upgraded dynamic slicing scheme design.

III. System Model

In this section, we present our system model, including the space-air-ground integrated communication models and software defined network (SDN)-based network resource management architecture. Table I summarizes the main notations that will be used in this paper.

A. Space-Air-Ground Communication Models

Figure 1 illustrates a typical SAGVN scenario, where ground base stations (BS) constitute the fixed terrestrial network to provide Cellular-V connections, UAVs with high

TABLE I
SUMMARY OF MAIN NOTATIONS

Symbol	Description	Symbol	Description
Ψ	The set of vehicles in the network.	r_{Si}	Downlink rate for the satellite sub-channel.
v_i	i-th vehicle in the set Ψ.	M	The number of sub-channels for Satellite-V.
N	The number of vehicles in the set Ψ.	$\mathcal{I}$	The set of services to be supported.
K	The number of sub-channels for LTE-V.	$\mathcal{J}$	The set of downlink resources.
W_B	Total bandwidth of cellular BS.	$\mathcal{T}$	The set of time slots.
r_{Bi}	Downlink rate for the cellular sub-channel.	$A_i(t)$	The amount of packet arrival from the i-th service.
p_{Bi}	Transmission power from the BS to v_i.	$a_i(t)$	The amount of admitted requests.
g_{Bi}	Channel power gain from the BS to v_i.	$a_{ij}(t)$	The number of requests scheduled to the j-th plane.
n_0	Background noise density of LTE-V.	$Z_j(t)$	The packet transmission capacity of the j-th plane.
$\mathcal{Z}$	The communication zone set of WiFi-V.	$C(t)$	The amount of capacity provided by UAVs.
L	The number of communication zones.	$b_{ij}(t)$	Transmission capacity sliced to the i-th service.
z_i	i-th communication zone in the set $\mathcal{Z}$.	$Q_{ij}(t)$	Queue backlog of the i-th service at the j-th plane.
l_j	Horizontal length of the zone z_i.	$\overline{Q}$	The time average queue stability criteria.
r_j	Associated date rate in the zone z_i.	λ_i	A constant to characterize the service weight.
r_{Ui}^j	Downlink rate for the vehicle v_i.	$\overline{C}$	Time averaged UAV dispatching cost.
ρ	The factor of WiFi throughput efficiency.	$\mathbf{\Theta}(t)$	The vector indicating queue sizes of all the queues.
n	The number of vehicles contending for WiFi.	$\Delta_V(t)$	One-slot conditional Lyapunov drift-plus-penalty.

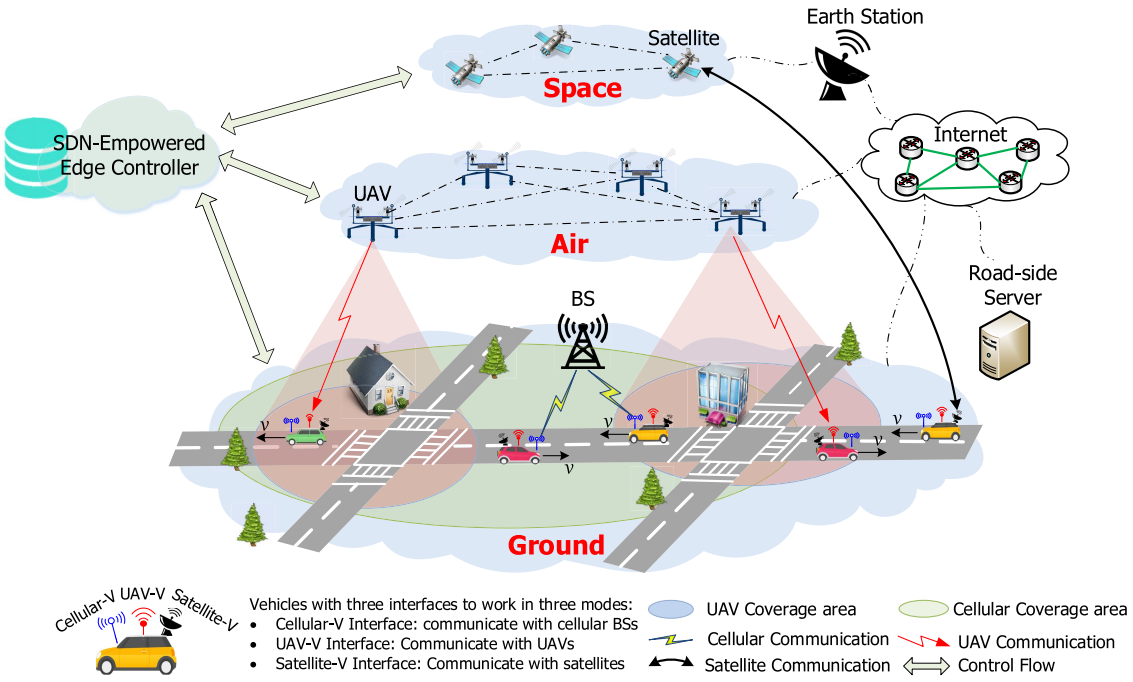


Fig. 1. Illustration of the considered SAGVN scenario.

agility are dynamically dispatched to provide the on-demand aerial network and support UAV-V connections, and the satellite constellation (containing hundreds of low Earth orbits (LEOs)) can provide the seamless wireless coverage within the area of interest to assist Satellite-V connections.[1] We assume each vehicle has three network interfaces to support Cellular-V, UAV-V, and Satellite-V communications, respectively. As vehicular network services mainly rely on the downlink communications, in this paper, we focus on the downlink services.

1) Cellular-V Communication: Suppose there is a BS which serves a set of vehicles, denoted by $\Psi = \{v_1, v_2, \ldots, v_N\}$. Following Cellular-V protocol like LTE-V [29], the BS allocates orthogonal bands to vehicles for simultaneous downlink transmissions without interference. At each time slot, we consider there are K sub-channels that can be allocated to users. Given the total BS bandwidth of W_B, each sub-channel can occupy a bandwidth of $\frac{W_B}{K}$. Therefore, for a vehicle v_i ($\forall v_i \in \Psi$), if one sub-channel is allocated to it, then its downlink rate

[1]In this paper, Cellular-V, UAV-V, and Satellite-V means vehicle communications are enabled by the underlying cellular, UAV, and satellite spectrum, respectively.

r_{Bi} yields

$$r_{Bi} = \frac{W_B}{K} \log_2 (1 + \frac{p_{Bi} g_{Bi}}{\frac{W_B}{K} n_0}), \quad (1)$$

where p_{Bi} and g_{Bi} mean the transmission power and channel power gain from the BS to vehicle v_i, respectively, and n_0 denotes the background noise density. Note that, the channel gain consists of both large scale fading (path loss, shadowing) and small scale fading (fast fading, multi-path fading, etc.). In this paper, we use g_{B_i} to incorporate the overall channel gain of all kinds of fading. However, characterizing the effect of detailed channel parameters, such as transmission distance, path loss component, number of multi-path, and the type of fading (Rayleigh, Gaussian, Nakagami, etc.), on the channel gain can be referred to dedicated research works [30], [31].

2) WiFi Enabled UAV-V Communication: For UAV-V communications, both cellular and WiFi bands are widely considered in the literature. In this paper, we adopt WiFi bands to support UAV-V communications due to its low cost. Commercial drone products using WiFi bands are also available [32], [33]. Besides, there are two additional significant reasons to employ WiFi band in our scenario. First, as the UAV is usually dispatched when the spectrum resource is insufficient, using the WiFi band can supplement the spectrum resource. Second, communication interference on other links can be avoided without complicated interference management. As stated in the IEEE 802.11 standard [34], the data rates of WiFi are discrete values which are determined by the SNR (signal-to-noise ratio) level, mainly impacted by the distance between the UAV and vehicle. Therefore, we adopt the drive-thru zone-based model to enable the UAV-V throughput acquisition, which has been widely verified and investigated in the previous studies [35]. Suppose there are L zones, denoted by $\mathcal{Z} = \{z_1, z_2, \ldots, z_L\}$. For each zone z_j ($\forall z_j \in \mathbf{Z}$), let l_j and r_j be its horizontal length and the associated data rate, respectively. The length and associated data rate of each zone can be calculated according to the WiFi Modulation and Coding Scheme (MCS) and SNR data sheet [36], where the WiFi channel is shared by vehicles under a contention-based mechanism. Therefore, for a vehicle v_i ($\forall v_i \in \Psi$), if it successfully acquires the WiFi channel within the zone z_j, then its downlink rate r_{Ui}^j can be calculated as

$$r_{Ui}^j = \frac{\rho r_j}{n}, \quad (2)$$

where ρ and n are the factor of WiFi throughput efficiency and the number of vehicles contending for the UAV channel, respectively. The parameter ρ characterizes the overhead of protocol negotiations and packet headers. For instance, the maximum throughput of IEEE standards is normally smaller than the claimed data rate, due to the protocol overhead. The detailed UAV-V communication set and WiFi zone parameters are given in the performance evaluation section.

3) Satellite-V Communication: For the future space network, there will be a satellite constellation constituted by a group of LEO satellites that work together to provide permanent global coverage. Thanks to the high mobility of LEO satellites, seamless connections in interested areas can be guaranteed by the cooperation of different LEOs.

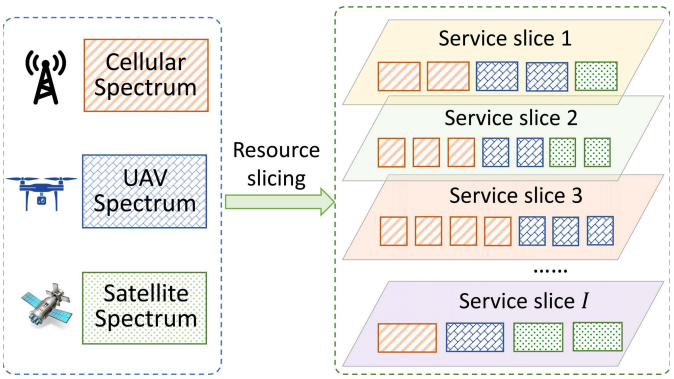


Fig. 2. SAG spectrum softwarization and slicing.

The satellites are considered to operate at Ku band or above, where rain attenuation becomes the major fading component as the LEO-to-vehicle distance remains constant, considering the relative moving speed of vehicles and satellites. The Weibull-based channel model can be utilized to characterize the rain attenuation for satellite links that operate at 10 GHz or above [37], [38]. However, within the limited targeted area, each vehicle usually undergoes the same rain attenuation. Therefore, for each sub-channel, if it is allocated to a vehicle v_i, the data rate r_{Si} is a constant, which is usually smaller than that of Cellular-V and UAV-V links. Likewise, we consider there are M orthogonal sub-channels that can be allocated to users for simultaneous transmissions.

B. SDN-Based Spectrum Resource Management

For resource management, we assume that at each area of interest, there are SDN-empowered edge controllers in charge of local SAG spectrum resource.[2] The edge controller is responsible for conducting spectrum resource softwarization and slicing. It should be noted that for SDN control, there will be significant signaling overhead to collect the network status and control the network. Therefore, various edge controllers should be deployed cooperatively to monitor the entire network while being responsive to react in real time. Moreover, edge controllers can negotiate the usage of large-domain resources, e.g., satellite spectrum. However, how to deploy the SDN controllers and optimize the control efficiency are out of the scope of this paper [39].

As shown in Fig. 2, the goal of this work is to slice SAG spectrum resources to accommodate the isolated service provisioning. In doing so, each type of services can be served by a customized resource slice, where service qualities would not be impacted by other types of services that have completely different request patterns. However, the hurdle is that service slices are with distinct QoS requirements, calling for customized and dedicated spectrum resources. Particularly, to enable high-frequency periodical communications, the ultra-reliability and low-latency communications are essential, while for high-precision map downloading, the stable and high

[2]Note that, we only consider the spectrum resource slicing which can be incrementally integrated with computing and storage slicing.

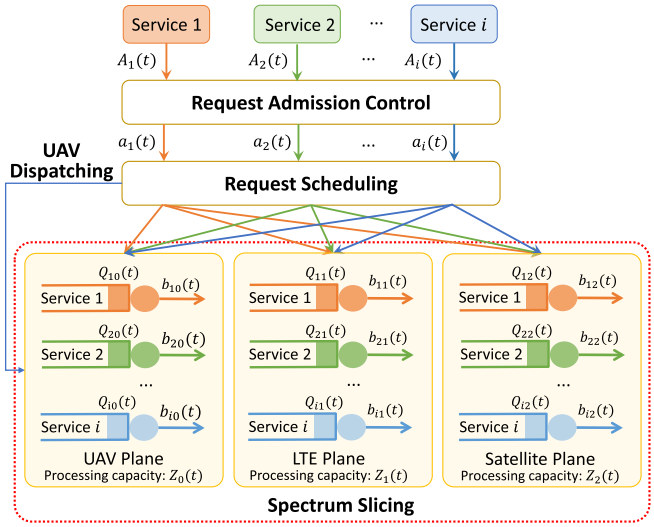


Fig. 3. The work flow of dynamic slicing.

throughput is more important. In addition, for vehicular Internet and entertainment services, the data request is unpredictable and normally data-intensive, posing challenges of resource feeding for the slice. On the other hand, caused by the mobility, the vehicular density variation can result in uncertainty and randomness of the service arrivals, making the resource slicing more intricate that might hamper service provisioning. In the following sections, we will elaborate on our dynamic slicing methodology to tackle those challenges.

IV. Work Flow of Dynamic Slicing and Problem Formulation

In current networks, requests of different services share the substrate network resources via multiplexing, where diversified QoS requirements of different services cannot be guaranteed. Within the SAGVN slicing framework, requests from the same service, which are of similar QoS requirements, are provisioned with a dedicated resource slice, contributing to service QoS satisfaction. In this section, we elaborate on the work flow of the proposed dynamic slicing, followed by the problem formulation and analysis.

A. Work Flow of Dynamic Slicing

A downlink scenario in the SAGVN is considered, where there is a set of downlink resources $\mathcal{J} = \{0, 1, \cdots, J\}$,[3] with supporting a set of services $\mathcal{I} = \{1, 2, \cdots, I\}$. Consider the network is operated in slotted time, denoted by $\mathcal{T} = \{0, 1, 2, \cdots, t, \cdots\}$, and the length of each time slot can be adjusted to the scheduling unit of the communication system. Note that, the control decisions are made every time slot. The value of slot duration can be the basic transmission slot in cellular networks, which means that the most fine-grained control actions are enabled in the system. Optionally, the duration of slot can be also the multiple of basic transmission slot in cellular network to reduce control overhead in accordance with the system demand. As shown in Fig. 3, we propose a dynamic slicing scheme, and it has the following four steps in each time slot t:

a) Request admission: For a certain type of service $i \in \mathcal{I}$, we use $A_i(t)$ to denote the amount of service arrival (in packets) during time slot t. $A_i(t)$ is assumed to be independent and identically distributed (i.i.d.) over different time slots, which is also independent from other service arrivals. Generally, there is an upper bound for all service arrivals, denoted by $A^{\max}$, and then $A_i(t) \in [0, A^{\max}]$, $\forall i \in \mathcal{I}$ and $\forall t \in \mathcal{T}$. Considering the limitation of network downlink capacity, the amount of admitted requests should be constrained, otherwise the system can be seriously overloaded, resulting in performance degradation. The amount of admitted requests from the i-th service is denoted by $a_i(t)$ during time slot t, which holds

$$0 \le a_i(t) \le A_i(t), \quad \forall i \in \mathcal{I}, \ \forall t \in \mathcal{T}. \tag{3}$$

b) Request scheduling: After admitting the requests into the system, we then need to schedule them to different resource pipelines for service processing. Particularly, we use $a_{ij}(t)$ to denote the number of requests out of $a_i(t)$ that are scheduled to the j-th downlink resource plane, which follows $a_i(t) = \sum_{j \in \mathcal{J}} a_{ij}(t)$. As shown in Fig. 3, with the slicing technology, each resource plane can maintain an independent service queue for each type of requests, isolating the service provisioning.

c) UAV dispatching: The packet transmission capacity of the j-th resource plane during the t-th time slot is denoted by $Z_j(t)$ ($j \in \mathcal{J}$). Unlike examining the advantages of agile deployment and efficient offloading provided by UAVs [19], [20], [40], the dynamic UAV dispatching case is considered under the high-density vehicular scenario, where communication resources are insufficient to accommodate the instantaneous service arrivals. In particular, we use $j = 0$ and $Z_0(t)$ to denote the UAV resource plane and the available UAV packet transmission capacity[4] during time slot t, respectively. Denote by $C(t)$ the amount of transmission capacity provided by dynamically dispatched UAVs during time slot t, which satisfies the boundedness assumption $C(t) \in [0, C^{\max}]$ since the number of dispatchable UAVs is limited in practice. To accommodate the dynamic packet requests from services, the edge server needs to determine the amount of dispatched UAVs during each time slot t, and the dispatched UAVs are available to the services from the $(t+1)$-th time slot.

d) Resource slicing: During time slot t, denote by $b_{ij}(t)$ the packet transmission capacity sliced to the i-th service requests by the j-th resource plane, which holds

$$\sum_{i \in \mathcal{I}} b_{ij}(t) \le Z_j(t), \ \forall j \in \mathcal{J}, \ t \in \mathcal{T}. \tag{4}$$

For requests from each type of service, the number of them to be processed is limited within one time slot, which has the boundedness $b_{ij}(t) \in [0, b^{\max}]$, $\forall i \in \mathcal{I}$, $\forall j \in \mathcal{J}$, and $\forall t \in \mathcal{T}$.

[3]In this work, $|\mathcal{J}|$ is equivalent to 3, which are in accordance with UAV, cellular, and satellite downlink data pipelines. The developed dynamic slicing framework can be readily applied to more general scenarios.

[4]Without loss of generality, we assume the available packet transmission capacity of both cellular plane and satellite plane are static, i.e., $Z_j(t) = Z_j$, $\forall j \in \{1, 2\}$ and $t \in \mathcal{T}$.

To finish the above four steps during each time slot, the following control parameters need to be determined, i.e., $a_i(t)$, $a_{ij}(t)$, $C(t)$, and $b_{ij}(t)$. To characterize their impacts on the system performance, we use $Q_{ij}(t)$ to denote the queue backlog of the i-th type of service request at the j-th resource plane during time slot t, which evolves as

$$Q_{ij}(t+1) = \max[Q_{ij}(t) - b_{ij}(t), 0] + a_{ij}(t), \tag{5}$$

where $Q_{ij}(0) \geq 0$, $\forall i \in \mathcal{I}$ and $\forall j \in \mathcal{J}$. To enable feasible iterations, the system should guarantee the stability of all the queues. Therefore, we adopt the time average queue stability criteria, i.e., defining

$$\overline{Q} \triangleq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} Q_{ij}(t), \tag{6}$$

where the system is stable if $\overline{Q} < \infty$ [41].

For the UAV resource plane, its packet transmission capacity depends on the dynamic UAV dispatching, which evolves as

$$Z_0(t+1) = Z_0(t) - \sum_{i\in\mathcal{I}} b_{i0}(t) + C(t). \tag{7}$$

In practice, to guarantee the service continuity, the available UAV transmission capacity should be maintained within a certain level considering the time lag between UAV dispatching and its operation. Formally, it translates to the following long-term time averaged equivalence

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} [Z_0(t) - \theta] \to 0, \tag{8}$$

where $Z_0(0)$ can be initialized to be a nonnegative value, and θ is the expected available UAV transmission capacity across the time span [41], [42].

B. Problem Formulation

For system management, different services usually call for differential QoS requirement, and they can bring different utility levels to the system provider in turn. For the service provider, the first objective is to maximize the network utilities of services, to achieve which we define the following system utility function for fair service provisioning [41]

$$U_i(\overline{a}_i) = \log(1 + \lambda_i \overline{a}_i), \ \forall i \in \mathcal{I} \tag{9}$$

where $\overline{a}_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[a_i(t)]$ is the long-term time-averaged amount of admitted requests from the i-th service, and λ_i is a positive constant to characterize the weight of the i-th service. Specifically, as utility functions $\{U_i(\cdot)\}_{i\in\mathcal{I}}$ are all concave, (9) guarantees diminishing return property, i.e., the network service provider cannot keep increasing its total utility by solely increasing the admitted requests from one service.

In the meantime, operational revenue is also critical to the system provider. Therefore, it is necessary to minimize the service provisioning costs from all the resource planes. It turns to minimize the UAV dispatching cost since the transmission capacity of the cellular and satellite are static. Denote by $\overline{C} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[C(t)]$ the long-term time averaged UAV dispatching cost. Considering the conflicting objectives of UAV dispatching cost and service provisioning utility, it is impossible to achieve the minimal cost and maximal utility simultaneously. To deal with it, a weighted sum of both objectives is adopted and the following optimization problem is formulated

$$\begin{aligned} \mathcal{P}_1: \ \max \ & \sum_{i\in\mathcal{I}} U_i(\overline{a}_i) - \beta\overline{C}, \\ \text{s.t. } & (3), (4) \text{ and } \overline{Q} < \infty, \end{aligned} \tag{10}$$

in which the per-slot optimization variables are the amount of admitted requests of each service $a_i(t)$, the amount of scheduled requests to each resource plane $a_{ij}(t)$, the transmission capacity of the dispatched UAVs $C(t)$, and the amount of sliced resources to each service queue $b_{ij}(t)$. In addition, β is a weight constant that specifies the unit cost of dispatched UAV transmission capacity.

It is challenging to directly attack the above optimization problem due to the following reasons. Firstly, the amount of request arrivals of different services are time-varying and unknown *a priori*, hence it is hard to make optimal offline decisions that satisfy constraint (3). Secondly, as the transmission capacity of the UAVs dispatched in the t-th time slot is only available from the $(t+1)$-th time slot, a set of decision parameters that satisfy constraint (4) may not be able to serve all the admitted request in the next time slot, which leads to potential request accumulation and queue instability in the long run. In the next section, we resort to the Lyapunov optimization framework to decouple the correlation between the optimization parameters, and provide effective per-slot control decisions across the time span, which collectively solve the above optimization problem.

V. The Dynamic Slicing Algorithm

A. Problem Transformation

Generally, the Lyapunov drift-plus-penalty minimization algorithm in its simplest form is powerful in optimizing long-term time averaged objectives. Yet, the presented optimization objective (10) is a function of time averages. Fortunately, by applying Jensen's inequality on the set of concave utility functions $\{U_i(\cdot)\}_{i\in\mathcal{I}}$, the problem $\mathcal{P}_1$ can be transformed via introducing a set of auxiliary variables [41]. Let $\gamma_i(t)$ be the nonnegative auxiliary variable corresponding to $a_i(t)$, $\forall i \in \mathcal{I}$ and $t \in \mathcal{T}$. Then, we can have the following lemma.

Lemma 1: The optimization problem $\mathcal{P}_1$ is equivalent to

$$\mathcal{P}_2: \ \max \ \sum_{i\in\mathcal{I}} U_i(\overline{\gamma}_i) - \beta\overline{C} \tag{11}$$

$$\begin{aligned} & s.t. \ \overline{\gamma}_i < \overline{a}_i, \ \forall i \in \mathcal{I} \\ & (3), (4) \textit{ and } \overline{Q} < \infty. \end{aligned} \tag{12}$$

The transformed problem $\mathcal{P}_2$ falls into the scope of Lyapunov optimization framework, which can be solved by constructing virtual queues corresponding to the auxiliary variable constraints in (12). In specific, we define a virtual queue for each type of service, which evolves as

$$H_i(t+1) = \max[H_i(t) + \gamma_i(t) - a_i(t), 0], \quad \forall i \in \mathcal{I}. \tag{13}$$

The virtual queue, $H_i(t)$, can transform inequality constraint (12) into a queue stability problem (Chapter 4.4 in [41]). Then, this problem can be addressed together with the stability of other queues, $Q_{ij}(t)$ and $Z_j(t)$. Otherwise, it would be difficult to directly satisfy constrain (12), since both sides of the inequality are in time-averaged form. Therefore, constraint (12) is satisfied as long as the set of queues $\{H_i(t)\}_{i\in\mathcal{I}}$ are stable [41]. Meanwhile, to further elaborate on the resource slicing constraint (4), the heavy-traffic scenario is considered, where the transmission capacity of both satellite and cellular are always fully utilized. Then, in each time slot $t \in \mathcal{T}$, we have $Z_j(t) - \sum_{i\in\mathcal{I}} b_{ij}(t) = 0$ for $j \in \{1, 2\}$, and $\sum_{i\in\mathcal{I}} b_{i0}(t) \leq Z_0(t)$.

Given the queue dynamics $\{Q_{ij}(t)\}_{i\in\mathcal{I},\, j\in\mathcal{J}}$, $\{H_i(t)\}_{i\in\mathcal{I}}$ and $\{Z_j(t)\}_{j\in\mathcal{J}}$ across the time span, we can define the following Lyapunov function corresponding to problem $\mathcal{P}_2$

$$L(t) \triangleq \frac{1}{2}\Big\{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} Q_{ij}(t)^2 + \sum_{i\in\mathcal{I}} H_i(t)^2 + \sum_{j=1}^{2} Z_j(t)^2 + [Z_0(t)-\theta]^2\Big\}. \quad (14)$$

The above definition represents a scalar measure of all the queue sizes, which is always nonnegative and $L(t) = 0$ if and only if all the queue sizes equal 0. Let $\Theta(t) \triangleq (\{Q_{ij}(t)\}_{i\in\mathcal{I},\, j\in\mathcal{J}}, \{H_i(t)\}_{i\in\mathcal{I}}, \{Z_j(t)\}_{j\in\mathcal{J}})$ be the concatenated vector that indicates the instantaneous queue sizes of all the queues by the end of the t-th time slot. Then, we can define the one-slot conditional Lyapunov drift-plus-penalty as

$$\Delta_V(t) \triangleq \Delta(t) - V\mathbb{E}\Big[\sum_{i\in\mathcal{I}} U_i(\gamma_i(t)) - \beta C(t)|\Theta(t)\Big], \quad (15)$$

where V is a weight constant that determines the tradeoff between system revenue and queueing stability, and $\Delta(t) \triangleq \mathbb{E}\{L(t+1) - L(t)|\Theta(t)\}$ is the one-slot conditional Lyapunov drift [41]. A smaller value of $\Delta_V(t)$ indicates more stable queue sizes and higher system revenue, and vice versa. The upper bound of $\Delta_V(t)$ can be given by the following lemma.

Lemma 2: For all possible queue states and control actions, $\Delta_V(t)$ is upper bounded by

$$\begin{aligned}\Delta_V(t) \leq\ & B - \sum_{i\in\mathcal{I}} \mathbb{E}\{VU_i(\gamma_i(t)) - H_i(t)\gamma_i(t)|\Theta(t)\} \\ & - \sum_{i\in\mathcal{I}} \mathbb{E}\{H_i(t)a_i(t) - \sum_{j\in\mathcal{J}} Q_{ij}(t)a_{ij}(t)|\Theta(t)\} \\ & + \mathbb{E}\Big\{\beta VC(t) + [Z_0(t)-\theta]C(t) + \sum_{j=1}^{2} Z_j^2|\Theta(t)\Big\} \\ & - \Big\{\mathbb{E}\Big[\sum_{i\in\mathcal{I}}(Q_{i0}(t) + Z_0(t) - \theta)b_{ij}(t)\Big] \\ & + \sum_{j=1}^{2}\mathbb{E}\Big[\sum_{i\in\mathcal{I}}(Q_{ij}(t) + Z_j)b_{ij}(t)\Big]|\Theta(t)\Big\}, \quad (16)\end{aligned}$$

where B is a constant determined by the general upper bounds A^{max}, b^{max}, and C^{max}.

The derivation of (16) in Lemma 2 is a direct application of Lemma 4.6 of [41]. We only provide a proof sketch here. First, following (4.46) of [41], three inequalities can be obtained by squaring the queue update equation (14), then taking conditional expectations of those inequalities and summing over $i \in \mathcal{I}$ and $j \in \mathcal{J}$ gives a bound on $\Delta(t)$. Adding the penalty to both sides concludes the proof.

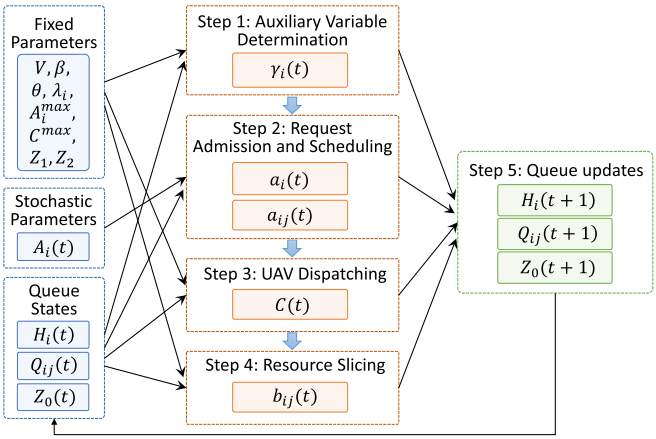


Fig. 4. Dynamic slicing algorithm.

The decoupling of control parameters in the right-hand side of the drift-plus-penalty upper bound (16) allows us to make independent and sequential decisions on $\gamma_i(t)$, $a_i(t)$, $a_{ij}(t)$, $C(t)$, and $b_{ij}(t)$. According to Lemma 2, by minimizing the value on the right-hand side of the inequality, which can be viewed as an upper bound of $\Delta_V(t)$, we are able to minimize the sum of system drift and system cost. Note that, however, this is not equivalent to simultaneously achieving the maximal system revenue and the best system stability. Setting the value of V to be 0 or $+\infty$ can achieve those two objectives respectively. Therefore, by tuning the value of V, we are able to balance the tradeoff between system revenue and system stability. In what follows, instead of directly attacking the optimization problem $\mathcal{P}_2$, we minimize the four separated conditional terms on the right-hand side of (16).

B. Dynamic Slicing Algorithm

For the system control, all the queue states $\Theta(t)$ are firstly observed in each time slot t, based on which the following five control actions are then performed.

1) Auxiliary Variable Determination: Given the observed virtual queue state $H_i(t)$, it turns to maximize $\sum_{i\in\mathcal{I}} \mathbb{E}\{VU_i(\gamma_i(t)) - H_i(t)\gamma_i(t)\}$, the optimization problem of which is to solve three single variable optimization problems over the feasible set of γ_i. It is readily solved since they all have closed-form solutions with the utility functions $\{U_i(\cdot)\}_{i\in\mathcal{I}}$ being well defined in (9).

2) Request Admission and Scheduling: Considering the coupling constraint $a_i(t) = \sum_{j\in\mathcal{J}} a_{ij}(t)$, it is difficult to determine a set of actions $a_i(t)$ and $a_{ij}(t)$ to maximize $\mathbb{E}\{H_i(t)a_i(t) - \sum_{j\in\mathcal{J}} Q_{ij}(t)a_{ij}(t)\}$ of each service when given the queuing states $H_i(t)$ and $Q_{ij}(t)$. To this end, we determine $a_i(t)$ and $a_{ij}(t)$ sequentially based on the "join the shortest queue first" heuristics [42], [43], rather than optimizing the admission and scheduling decisions simultaneously. Specifically, for the i-th service type, if $j = \text{argmin}_{j\in\mathcal{J}} Q_{ij}(t)$,

a simple scheduling strategy is to send all the admitted requests $a_i(t)$ to the resource plane that has the shortest queue, i.e., $a_{ij}(t) = a_i(t)$, otherwise, $a_{ij}(t) = 0$. Then, it reduces to maximizing $\mathbb{E}\{H_i(t)a_i(t) - Q_{ij^*}(t)a_i(t)\}$, which can be easily solved within the feasible interval of $a_i(t)$.

3) UAV Dispatching: Given the predefined parameters β, θ, and V, it is easy to obtain the optimal UAV dispatching capacity by minimizing $\mathbb{E}[\beta V C(t) + [Z_0(t) - \theta]C(t) + \sum_{j=1}^{2} Z_j^2]$ based on the instantaneous queue size $Z_0(t)$, i.e., obtaining the feasible interval $C(t) \in [0, C^{\max}]$.

4) Resource Slicing: It is sufficient to make individual allocation decisions on each resource plane in order to maximize the last term of the drift-plus-penalty bound in (16). Particularly, for the UAV resource plane ($j = 0$), the weight of the allocation $b_{ij}(t)$ is $Q_{i0}(t)+Z_0(t)-\theta$, while for the satellite and cellular resource plane ($j \in \{1, 2\}$), the corresponding weight is $Q_{ij}(t) + Z_j$. For each resource plane, one effective solution is to, first sort the queues in accordance with their weights in descending order, and then slice resources sequentially as long as the resource constraint (4) is satisfied. Noted that, queues with negative weights will not be considered in this phase, and the corresponding $b_{ij}(t)$ is set to 0.

5) Queue Updating: After achieving all the control parameters, the queuing states $Q_{ij}(t)$, $H_i(t)$, and $Z_0(t)$ can be updated based on (5), (13), and (7), respectively, which will be used for decision making in the next time slot.

Figure 4 summarizes the five steps of the dynamic slicing process. As mentioned before, the proposed algorithm requires no prior information on the statistics of the service request arrivals. All the control decisions are made in an online fashion based on instantaneous queuing states. It is also worth noting that, thanks to the decoupling feature enabled by (16), each of the control parameters can be determined independently. Hence, it is possible to determine those control parameters separately. As described earlier in Section V-B, minimizing the 4 terms (hence determining the control parameters) can be readily achieved by solving a single-variable differential equation ($\gamma_i(t)$, and $C(t)$), or a sorting-based greedy heuristic algorithm ($a_i(t)$, $a_{ij}(t)$, and $b_{ij}(t)$), which are all with low complexity and can be completed in polynomial time [44].

VI. Performance Evaluation

In this section, we build the simulation scenario and conduct extensive experiments to evaluate the performance of the proposed dynamic slicing scheme.

A. Simulation Setup

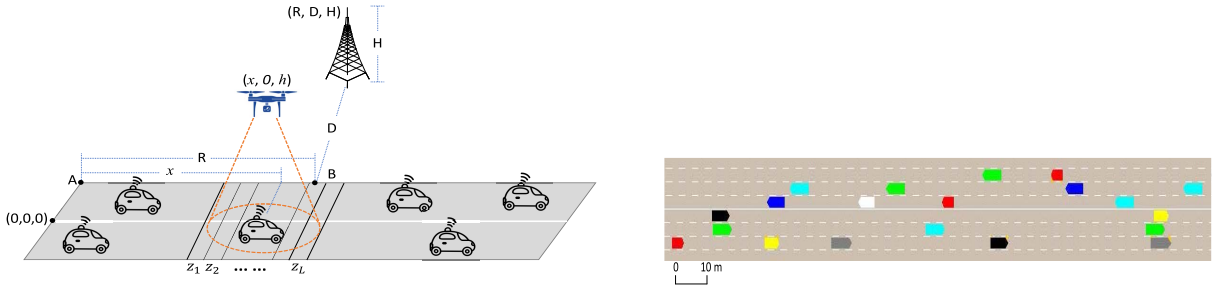
As shown in Fig. 5 (a), we create a bidirectional 8-lane highway scenario,[5] where a cellular BS locates beside the road with coordinate ($R = 2000$ m, $D = 100$ m, $H = 50$ m), and if UAVs are dispatched, they will hover at the coordinate of (x, 0 m, 5 m), i.e., $h = 5$ m. The value of x changes when different numbers of UAVs are dispatched. To emulate practical driving conditions, the simulation tool SUMO [45] is adopted to generate realistic vehicular trace. Specifically, vehicles are generated at the entrance of each lane, and the vehicle arriving process is assumed to follow a Poisson distribution with an arriving rate λ_{V}. By varying the value of λ_{V}, on-road traffic with different vehicular densities can be generated. In subsequent simulations, λ_{V} for each lane is set to be 0.25. When a vehicle is generated, a set of motor performance, including maximum speed, acceleration, deceleration, etc., will be chosen from the predefined performance configurations. For driving behaviors, the events of lane changing and overtaking happen following the widely used Krauss car-following model and LC2013 lane-changing model [45]. In addition, each lane is given a speed limit of 80 km/h, 100 km/h, 120 km/h and 140 km/h, respectively. We log the moving trace of all vehicles within a duration of 1000 s. Fig. 5 (b) shows a snapshot of the simulated scenario, where different colors indicate differentiated motor performance.

TABLE II
UAV WiFi Zone Parameters

Index	l_i (m)	r_i (Mbps)	**Index**	l_i (m)	r_i (Mbps)
1	23.8	13.5	11	0.7	162
2	21.3	27	12	2.1	135
3	7.5	40.5	13	4.5	121.5
4	10.7	54	14	2.7	108
5	5.3	81	15	5.3	81
6	2.7	108	16	10.7	54
7	4.5	121.5	17	7.5	40.5
8	2.1	135	18	21.3	27
9	0.7	162	19	23.8	13.5
10	5.8	180			

1) Spectrum Transmission Capability: To transform the spectrum resource to the transmission capability in accordance with the queueing model of packet processing, we treat each service request as a bunch of packets that need to be transmitted. The transmission capability is the number of packets that can be transmitted within the time slot duration. To obtain the data rate of the cellular downlink pipeline, similar simulation settings like [46] are adopted to set the cellular parameters, where the transmission power p_B, background noise density n_0, and total bandwidth W_B, are set to be 20 W, 1×10^{-15} W/Hz, and 20 MHz, respectively. Additionally, we set the channel power gain as $g_{Bi} = \varrho_{Bi} d_{Bi}^{-\alpha}$, where d_{Bi} is the distance between the BS and vehicle, ϱ_{Bi} is the channel fading (assumed to follow an exponential distribution with unit mean), and α is the path loss exponent (set to be $\alpha = 3$). For sub-channels, we set $K = 100$, i.e., with 100 orthogonal sub-channels that can be allocated to users, where each sub-channel has 0.2 MHz bandwidth. Note that, the data rate of each sub-channel is mainly affected by the distance d_{Bi} in accordance with (1). For each time slot, those sub-channels are assumed to be occupied by vehicles (uniformly distributed on the road) with the same probability. Therefore, the data rate of each sub-channel can be calculated by $\frac{1}{R}\int_A^B \frac{W_B}{K}\log_2\left(1 + \frac{p_{Bi}\varrho_{Bi}y^{-\alpha}}{\frac{W_B}{K}n_0}\right)d_y$. Particularly, as shown in Fig 5 (a), y is the accumulated distance when the vehicle

[5]It should be noted that, the efficacy of our dynamic slicing is independent from the underlying vehicular environments, and this simulation scenario is considered to carry out a specific performance gain.



(a) simulation scenario (b) snapshot of the simulated scenario

Fig. 5. Simulation setup.

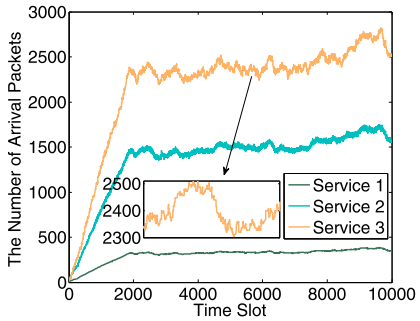


Fig. 6. The service arrival packets vs. time.

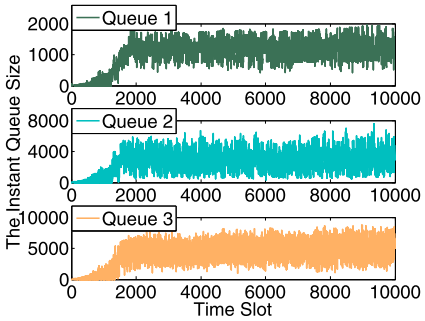


Fig. 7. Three instant service queue size vs. time.

traveling from the point A to B. Then, the initial transmission capability of the cellular resource plane can be calculated as

$$Z_1 = \frac{\tau K \int_A^B \frac{W_B}{K} \log_2 \left(1 + \frac{p_{BQBi} y^{-\alpha}}{\frac{W_B}{K} n_0}\right) d_y}{8\varsigma R}, \tag{17}$$

where ς and τ are the maximum payload of each packet and the duration of each time slot, being as 1500 bytes and 100 ms, respectively.

To mimic the WiFi-enabled UAV-V communications, the 802.11n (HT) protocol is adopted, where we set the total bandwidth W_U as 40 MHz and the band f_c operates at 2.4 GHz [34]. The length of each zone (i.e., l_j) and the associated data rate (i.e., r_j) under the free space path loss model are calculated in accordance with the WiFi MCS and SNR data sheet [36]. Table II shows the calculated WiFi zone parameters. Likewise, we can calculate the transmission capability of UAV (denoted by T_{U}) as

$$T_{\mathrm{U}} = \frac{\tau \rho \Sigma_{j=1}^{L} l_j r_j}{8\varsigma \Sigma_{j=1}^{L} l_j}, \tag{18}$$

where ρ is set as 0.9.

For Satellite-V communications, 20 sub-channels (i.e., $M = 20$) are considered, where each sub-channel can provide the data rate of 0.5 Mbps (i.e., $r_{\mathrm{S}} = 0.5 \times 10^6$ bps). Therefore, the transmission capability of the satellite resource plane Z_2 is equivalent to $\frac{\tau M r_{\mathrm{S}}}{8\varsigma}$.

2) Vehicular Service Requests: A wide spectrum of vehicular services have been discussed [2], from which we extract three typical and network-behavior differentiated services and mimic their behaviors under our dynamic slicing framework. Specifically, to assist safely driving or enable tele-operated driving, each vehicle is required to periodically download the control packet from the road-side server, where each packet contains the real-time surrounding environment status or steering instruction. To this end, a *URLLC* service (denoted by service 1) is launched for each vehicle, where they have to download one packet every time slot. To support the automated services such as high-precision map downloading or computation-intensive task offloading, a *Streaming* service (denoted by service 2) is launched for each vehicle, where the network connection is always required and they keep downloading contents from the server with a relatively stable throughput T_{strm} (set as 0.5 Mbps). To mimic the vehicular entertainment, an *Opportunistic* service (denoted by service 3) is also launched for each vehicle, where they can randomly generate a download request with a file size being stochastic. Particularly, each user launches a download request under a Poisson process with a rate $\lambda_{\mathrm{oppt}} = 0.2$ (a new request can be launched only after the previous one is finished). The download size (in bytes) is randomly chosen from the set $\{10^5, 10^6, \ldots, 10^{10}\}$, which ranges from small-size emails, to medium-size musics and large-size videos [47].

B. Performance Results

For service activities, Fig. 6 shows the arrival packets with time, where the number of *URLLC* packets also indicates the number of vehicles since only one packet is requested by each vehicle in the service. It can be seen that the number of vehicles increases first within the time slot 2000, and then becomes stable and oscillates around 340, which can emulate the traffic from the off-peak time to rush hour. Three types of network services are simulated where the *URLLC* service is small and deterministic, the *Streaming* service is medium with

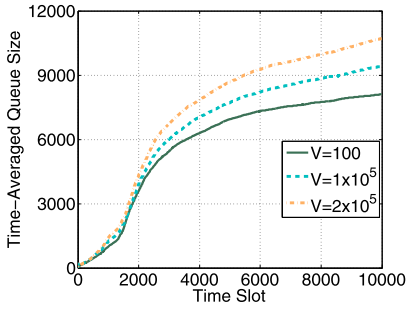


Fig. 8. Time-averaged service queue size.

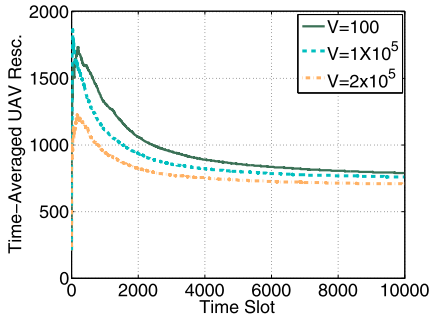


Fig. 9. Time-averaged UAV resource.

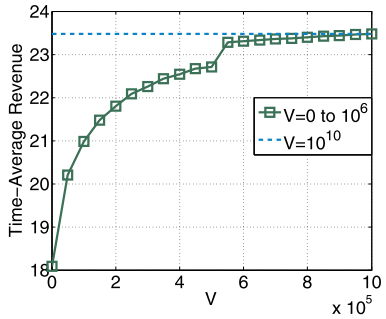


Fig. 10. Time-averaged revenue vs. V.

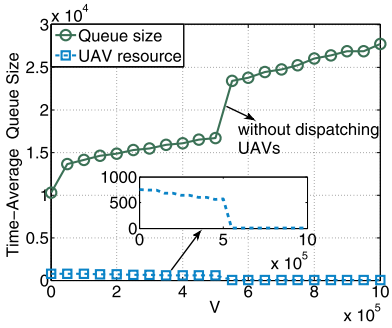


Fig. 11. Time-averaged queue size vs. V.

stable throughout requirement, and the *Opportunistic* service is heavy and more stochastic.

1) System Stability Examination: We first check the system stability, i.e., how the service queues evolve. To this end, we fix $V = 100$, $\beta = 0.002$, $\lambda_1 = \lambda_2 = \lambda_3 = 5$, $\theta = 5 \times T_{\mathrm{U}}$, to investigate the simulation results. Fig. 7 shows the instant queue size (in packets) of three services, i.e., $\Sigma_{j=1}^{3}\ Q_{0j}(t)$, $\Sigma_{j=1}^{3}\ Q_{1j}(t)$, and $\Sigma_{j=1}^{3}\ Q_{2j}(t)$, respectively. We can achieve the following two major observations. First, for three applications, their service queues are well stabilized. Second, the queue length of each service is positively correlated with the request arrivals. For instance, the queue size of three services oscillates around 1000, 3000 and 5000 packets, respectively, which are in accordance with their request arrivals shown in Fig. 6. It means that in the proposed dynamic slicing, resource would be fairly orchestrated to each slice, which is able to guarantee slice-specific performance. Fig. 8 shows the time-averaged service queue size (i.e., $\overline{Q}$), and we can see that even though the request arrival is far more than that the system can process, the time-averaged queue size can be well stabilized due to the effectiveness of the admission control. Particularly, when the time slot reaches 4000, the increase in time-averaged queue size is marginal.[6] In addition, with larger V, the time-averaged queue size becomes larger since according to (15), larger V means the system cares more about the system revenue rather than the queueing stability.

In addition, Fig. 9 shows the time-averaged UAV plane resource (in packets), i.e., $\overline{Z_0}$, and we can see that the UAV resource can be well stabilized by dispatching UAVs after the initial adaptation process. Besides, larger V can result in lower UAV resource stabilization, to explain which we can revisit to the optimal slicing algorithm of UAV dispatching. In particular, when the remaining UAV resource is smaller than $\theta - V\beta$, additional UAVs will be dispatched, making the UAV resource stabilizing around $\theta - V\beta$. It can well explain the observation in Fig. 8 that a larger V violates the queueing stability since less UAV resource is provisioned to deal with the service requests.

2) Impact of Control Parameters: To further understand the impact of parameter V, Fig. 10 and Fig. 11 shows the time-averaged revenue and queue size, respectively, when varying the value of V. From Fig. 10, we can have the following two major observations. First, with the increase of the value of V, the time-averaged revenue can increase accordingly. Second, there is an upper bound for the revenue no matter how to increase the value of V, following the stability constraint. Specifically, when we range the value V from 0 to 10^6, the time-averaged revenue increases and approaches 23.4, while even with excessively high value of V (e.g., 10^{10}), the revenue is also about 23.4. Likewise, when increasing the value of V, the time-averaged queue size is also enlarged accordingly. It happens since the queueing stability is sacrificed, conforming with (15). However, when the value of V increases from 5×10^5 to 5.5×10^5, we can find that there is an abrupt rise in both figures. To explain it, we can see from the Fig. 11, where the provisioned UAV resource is reduced to save the cost when the value of V is smaller than 5×10^5, and no additional UAV resource will be provisioned when the value of V is set as 5.5×10^5, leading to an abrupt rise for the time-averaged revenue and queue size.

[6]Note that, due to the limited observation time, the increase in time-averaged queue size is reasonable as the new time-step queue size is always larger than the previous time-averaged queue size, pulling the new time-averaged queue size to a larger one.

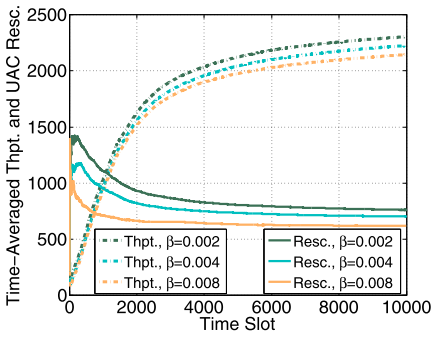


Fig. 12. Time-averaged system throughput and UAV resource vs. β.

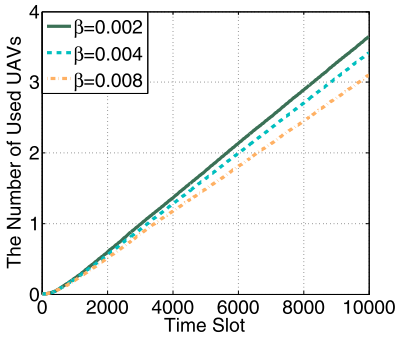


Fig. 13. The normalized number of used UAVs vs. time.

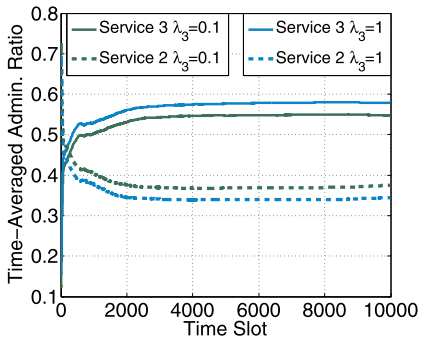


Fig. 14. The request admission percentage.

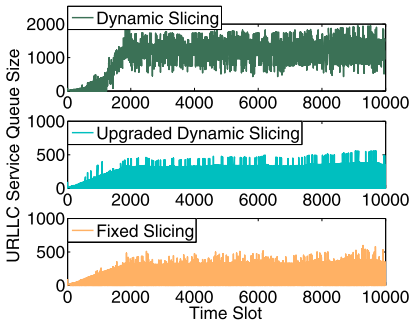


Fig. 15. The instant *URLLC* service queue size under different schemes.

We then investigate the impact of parameter β. According to (11), larger β means that the UAV dispatching cost is more considerable. We plot the time-averaged system throughput and UAV resource in Fig. 12 by ranging β from 0.002 to 0.008 while fixing $V = 10^5$. As the cost of dispatching UAV becomes more expensive, the time-averaged system throughput will decrease since less UAV resource will be leveraged, and the time-averaged UAV resource will also decrease. Assuming each UAV's battery supports at most 15 mins operation, we calculate the normalized number of used UAVs for different values of β and plot their results in Fig. 13. It can be seen that with the increase of β, the number of consumed UAVs decreases accordingly. The parameter θ means the maximum available UAVs in the environment, i.e., the UAV resource is expected to be stabilized around the value, which could similarly affect the system performance. Specifically, according to (8), larger θ results in larger UAV resource stabilization and more UAVs need to be dispatched to keep the stability.[7]

C. Differentiated Service Provisioning

Another merit of the dynamic slicing is that differentiated service provisioning can be guaranteed as each service could have distinct and independent utility functions. Specifically, we fix $V = 100$, $\beta = 0.002$ and $\lambda_1 = \lambda_2 = 1$, and then vary the value λ_3 from 0.1 to 1. We plot the request admission percentage of *Streaming* and *Opportunistic* services for two settings in Fig. 14. We can observe that with increasing the value of λ_3, the admission percentage for *Opportunistic* service increases accordingly while the admission percentage for *Streaming* service is more inclined to reduce. It implies that by adopting the proposed dynamic slicing scheme, differentiated service provisioning could be enabled by differentiating the service utility function.

Upgraded Dynamic Slicing. As the goal of the dynamic slicing is to maximize the system throughput with queueing stabilization, it is still hard to guarantee low latency transmission for *URLLC* service. To this end, we propose an upgraded dynamic slicing, in which during the fourth step of the algorithm, the *URLLC* request will be processed with the highest priority. Besides, we also propose a fixed slicing as a benchmark scheme, in which during the fourth step, the resource will be proportionally sliced for each type of services.

Figure 15 shows the instant queue size of *URLLC* service in different slicing schemes, and it can be seen that the queue size can be strictly bounded in both the upgraded dynamic slicing and fixed slicing schemes. The reason is that *URLLC* service requests are processed with the first priority in the upgraded dynamic slicing, while abundant resource is sliced for *URLLC* service in the fixed slicing. For instance, the queue size oscillates around 1000 in the dynamic slicing, indicating that *URLLC* packets would accumulate without real-time processing while in the other two slicing schemes, the queue sizes are below 500 (i.e., just the same as the number of requests), indicating that all *URLLC* packets are processed within the current time slot. Fig. 16 and Fig. 17 show the time-averaged system throughput and time-averaged queue size, respectively,

[7]Due to the similar observations and space limitation, the impact of θ is omitted.

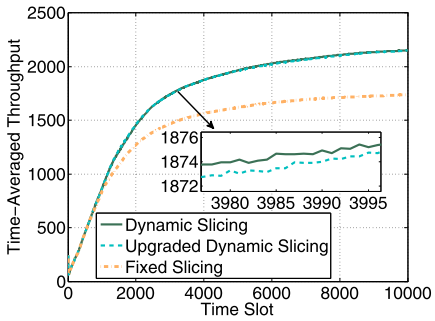


Fig. 16. Time-averaged throughput under different schemes.

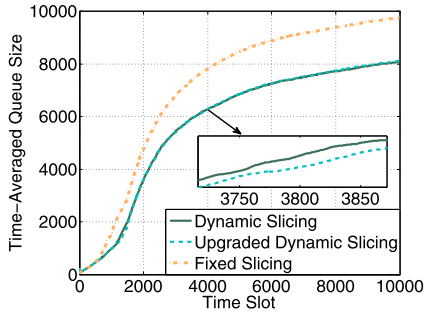


Fig. 17. Time-averaged queue size under different schemes.

and we can make the following two major statements. First, compared to the dynamic slicing, the fixed slicing fails to catch up with the vehicular dynamics and is inefficient in resource management as the system throughput degrades dramatically and the queue size accumulates rapidly. For instance, the fixed slicing provides the throughput around 1750 packets per time slot while the proposed dynamic slicing is able to improve it up to 2200 packets per time slot, with about 26% improvement. Second, the upgraded dynamic slicing can achieve the same level performance of throughput and queueing stability with the dynamic slicing as there is only negligible performance degradation in both figures. It is reasonable as compared to other two services requests, *URLLC* requests are relatively lightweight, processing which preferentially has little impact on the overall performance.

In summary, the proposed dynamic slicing can provide the superior overall performance and support differentiated service provisioning. Moreover, with the upgraded dynamic slicing, the ultra-low-latency of *URLLC* service can be also guaranteed with negligible throughput performance degradation.

VII. Conclusion and Future Work

In this paper, we have proposed an online control framework to dynamically slice the SAG spectrum resource for isolated vehicular service provisioning. Specifically, to maximize the system long-term revenue (regarding system throughput and UAV cost) and guarantee service queueing stabilization, we have decoupled the dynamic SAG resource slicing problem into several independent subproblems, including request admission and scheduling, UAV dispatching, and resource slicing, which have been solved by closed-form solutions based on the Lyapunov optimization theory. Extensive simulations have been conducted and the results have demonstrated the efficacy of the proposed dynamic slicing scheme in terms of the system throughput enhancement, the controllable UAV usage, and the trade-off between the system revenue and stabilization. For our future work, we will further take computing and storage resources slicing into consideration for end-to-end service provisioning.

References

[1] F. Lyu *et al.*, "Dynamic spectrum slicing and optimization in SAG integrated vehicular networks," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, Victoria, BC Canada, Nov. 2020, pp. 1–6.

[2] *Release 16 Description: Summary of Rel-16 Work Items*, document TR21.916, 3GPP, Jul. 2020. [Online]. Available: https://www.3gpp.org/release-16

[3] F. Lyu *et al.*, "Towards rear-end collision avoidance: Adaptive beaconing for connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1248–1263, Feb. 2021.

[4] X. Cheng, R. Zhang, and L. Yang, "Wireless toward the era of intelligent vehicles," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 188–202, Feb. 2019.

[5] J. E. Siegel, D. C. Erb, and S. E. Sarma, "A survey of the connected vehicle Landscape—Architectures, enabling technologies, applications, and development areas," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2391–2406, Aug. 2018.

[6] F. Lyu *et al.*, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020.

[7] R. Zhang, X. Cheng, and L. Yang, "Flexible energy management protocol for cooperative EV-to-EV charging," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 172–184, Jan. 2019.

[8] Y. Wang *et al.*, "Joint resource allocation and UAV trajectory optimization for space–air–ground Internet of remote things networks," *IEEE Syst. J.*, early access, Sep. 10, 2020, doi: 10.1109/JSYST.2020.3019463.

[9] 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, May 2018.

[10] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[11] X. Cheng *et al.*, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[12] Y. Chen, Y. Wang, M. Liu, J. Zhang, and L. Jiao, "Network slicing enabled resource management for service-oriented ultra-reliable and low-latency vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7847–7862, Jul. 2020.

[13] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[14] T. Taleb, I. Afolabi, and M. Bagaa, "Orchestrating 5G network slices to support industrial Internet and to shape next-generation smart factories," *IEEE Netw.*, vol. 33, no. 4, pp. 146–154, Jul. 2019.

[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, Jul. 2017.

[16] Z. Zhou, J. Feng, C. Zhang, Z. Chang, Y. Zhang, and K. M. S. Huq, "SAGECELL: Software-defined space-air-ground integrated moving cells," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 92–99, Aug. 2018.

[17] N. Kato *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.

[18] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[19] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, Dec. 2020.

[20] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.
[21] X. Xu, Y. Zeng, Y. L. Guan, and R. Zhang, "Overcoming endurance issue: UAV-enabled communications with proactive caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1231–1244, Jun. 2018.
[22] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.
[23] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 38–45, Dec. 2017.
[24] C. Campolo, A. Molinaro, A. Iera, R. R. Fontes, and C. E. Rothenberg, "Towards 5G network slicing for the V2X ecosystem," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 400–405.
[25] 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.
[26] Q. Ye, W. Zhuang, S. Zhang, A.-L. Jin, X. Shen, and X. Li, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9896–9910, Oct. 2018.
[27] F. Bahlke, O. D. Ramos-Cantor, S. Henneberger, and M. Pesavento, "Optimized cell planning for network slicing in heterogeneous wireless communication networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1676–1679, Aug. 2018.
[28] D. H. Kim, S. M. A. Kazmi, and C. S. Hong, "Cooperative slice allocation for virtualized wireless network: A matching game approach," in *Proc. 12th Int. Conf. Ubiquitous Inf. Manage. Commun.*, Jan. 2018, pp. 94:1–94:6.
[29] S. Chen, J. Hu, Y. Shi, and L. Zhao, "LTE-V: A TD-LTE-based V2X solution for future vehicular network," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 997–1005, Dec. 2016.
[30] A. Nosratinia and T. E. Hunter, "Grouping and partner selection in cooperative wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 2, pp. 369–378, Feb. 2007.
[31] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, no. 6, pp. 1986–1992, Nov. 1997.
[32] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.
[33] W. Shi *et al.*, "Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks," *IEEE Access*, vol. 6, pp. 12518–12529, 2018.
[34] *IEEE 802.11 Working Group and Others, IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2012, (Revision of IEEE Std 802.11-2007), 2012, pp. 1–2793.
[35] W. Xu, W. Shi, F. Lyu, H. Zhou, N. Cheng, and X. Shen, "Throughput analysis of vehicular Internet access via roadside WiFi hotspot," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3980–3991, Apr. 2019.
[36] *IEEE 802.11 Working Group, 802.11n/HT and 802.11ac/VHT*. Accessed: Feb. 2019. [Online]. Available: https://d2cpnw0u24fjm4.cloudfront.net/wp-content/uploads/802.11n-and-802.11ac-MCS-SNR-and-RSSI.pdf
[37] Z. Ji, Y. Wang, W. Feng, and J. Lu, "Delay-aware power and bandwidth allocation for multiuser satellite downlinks," *IEEE Commun. Lett.*, vol. 18, no. 11, pp. 1951–1954, Nov. 2014.
[38] S. A. Kanellopoulos, C. I. Kourogiorgas, A. D. Panagopoulos, S. N. Livieratos, and G. E. Chatzarakis, "Channel model for satellite communication links above 10 GHz based on weibull distribution," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 568–571, Apr. 2014.
[39] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, Feb. 2020.
[40] Z. Li *et al.*, "Energy efficient resource allocation for UAV-assisted space-air-ground Internet of remote things networks," *IEEE Access*, vol. 7, pp. 145348–145362, 2019, doi: 10.1109/ACCESS.2019.2945478.
[41] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.
[42] W. Fang, X. Yao, X. Zhao, J. Yin, and N. Xiong, "A stochastic control approach to maximize profit on service provisioning for mobile cloudlet platforms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 522–534, Apr. 2018.
[43] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in SaaS clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2648–2658, Oct. 2014.
[44] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 125:1–125:36, Oct. 2019.
[45] DLR Institute of Transportation Systems. *SUMO: Simulation of Urban MObility*. Accessed: Jan. 2019. [Online]. Available: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-1213/
[46] Y. Wu, Y. He, L. P. Qian, J. Huang, and X. Shen, "Optimal resource allocations for mobile data offloading via dual-connectivity," *IEEE Trans. Mobile Comput.*, vol. 17, no. 10, pp. 2349–2365, Oct. 2018.
[47] YouTube Help Community. *Live Encoder Settings, Bitrates, and Resolutions*. Accessed: Feb. 2019. [Online]. Available: https://support.google.com/youtube/answer/2853702?hl=en

Feng Lyu (Member, IEEE) received the B.S. degree in software engineering from Central South University, Changsha, China, in 2013, and the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2018. From October 2016 to October 2017 and from September 2018 to December 2019, he worked as a Post-Doctoral Fellow and a Visiting Ph.D. Student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a Professor with the School of Computer Science and Engineering, Central South University. His research interests include vehicular networks, beyond 5G networks, big data measurement and application design, and could/edge computing. He is also a member of the IEEE Computer Society, the Communication Society, and the Vehicular Technology Society. He has served as a TPC member for many international conferences. He was a recipient of the Best Paper Award of IEEE ICC 2019. He also serves as an Associate Editor for IEEE SYSTEMS JOURNAL and a leading Guest Editor for *Peer-to-Peer Networking and Applications*.

Peng Yang (Member, IEEE) received the B.E. degree in communication engineering and the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013 and 2018, respectively. He was with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, as a Visiting Ph.D. Student from September 2015 to September 2017, and a Post-Doctoral Fellow from September 2018 to December 2019. Since January 2020, he has been a Faculty Member with the School of Electronic Information and Communications, HUST. His current research interests include mobile edge computing, and video streaming and analytics.

Huaqing Wu (Graduate Student Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her current research interests include vehicular networks with emphasis on edge caching, wireless resource management, space-air-ground integrated networks, and application of artificial intelligence (AI) for wireless networks.

Conghao Zhou (Graduate Student Member, IEEE) received the B.E. degree from Northeastern University, Shenyang, China, in 2017, and the M.S. degree from the University of Illinois at Chicago, Chicago, IL, USA, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include space-air-ground integration networks and machine learning in wireless networks.

Ju Ren (Member, IEEE) received the B.Sc., M.Sc., Ph.D. degrees from Central South University, China, in 2009, 2012, and 2016, respectively, all in computer science.

He is currently a Professor with the School of Computer Science and Engineering, Central South University. His research interests include the Internet-of-Things, edge computing, big data, and security and privacy. He received many best paper awards from IEEE flagship conferences, including IEEE ICC'19 and IEEE HPCC'19, and the IEEE TCSC Early Career Researcher Award in 2019. He is recognized as a Highly Cited Researcher by Clarivate. He has also served as the General Co-Chair for IEEE BigDataSE'20, the TPC Co-Chair for IEEE BigDataSE'19, a Poster Co-Chair for IEEE MASS'18, and a Track Co-Chair for IEEE/CIC ICCC'19, IEEE I-SPAN'18, and VTC'17 Fall, and an active reviewer for over 20 international journals. He also serves/has served as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and *Peer-to-Peer Networking and Applications*, a Guest Editor for IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE NETWORK, and a TPC member for many international conferences, including IEEE INFOCOM'21/20/19/18, and ICDCS'21.

Yaoxue Zhang (Senior Member, IEEE) received the B.S. degree from the Northwest Institute of Telecommunication Engineering, China, in 1982, and the Ph.D. degree in computer networking from Tohoku University, Japan, in 1989. He is currently a Professor with the School of Computer Science and Engineering, Central South University, China, and a Professor with the Department of Computer Science and Technology, Tsinghua University, China. He has published over 200 technical articles in international journals and conferences, as well as nine monographs and textbooks. His research interests include computer networking, operating systems, ubiquitous/pervasive computing, transparent computing, and big data. He is also a fellow of the Chinese Academy of Engineering, China, and the Editor-in-Chief of *Chinese Journal of Electronics*.

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, Canada. His research interests include network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. He is also 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 Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, 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, and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He also received the Excellent Graduate Supervision Award in 2006 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 IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is also the President Elect of the IEEE Communications Society. He was the Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a Member of IEEE Fellow Selection Committee of the ComSoc. He has also served as the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*.