Energy-Efficient Virtual Resource Allocation of Slices in Vehicles-Assisted B5G Networks

Haotong Cao, *Member, IEEE*, Haitao Zhao, *Member, IEEE*, Anish Jindal, *Member, IEEE,* Gagangeet Singh Aujla, *Senior Member, IEEE* and Longxiang Yang

***Abstract*—Academia community started the research beyond 5G (B5G) while 5G systems and networks are still being landed for large-scale commercial applications. In order to enhance the agility and flexibility attributes of B5G networks, network function virtualization (NFV) and network slicing (NS) are attracting extensive research attention. Meanwhile, vehicles are promised to connect to the B5G networks so as to expand the service coverage and reach the 'last one mile'. In this paper, we research the virtual resource allocation of slices in vehicles-assisted B5G networks. We aim at saving total energy cost of deployed slices while ensuring high slice acceptance ratio. We firstly present the system model of vehicles-assisted B5G networks, supporting both virtualization and slicing schemes. Then, we present the energy cost of vehicles-assisted B5G networks. Afterwards, we propose one energy efficient algorithm, abbreviated as *Ener-Eff-Slice*, to solve the virtual resource allocation of slices in vehicles-assisted B5G networks. Numerical results are recorded, plotted and discussed, which prove the efficacy of our scheme. Finally, we do the conclusion marks and discuss the next-step work.**

***Index Terms*—B5G; vehicles; network slicing; NFV; virtual resource allocation; energy efficiency.**

I. Introduction

5G mobile communication systems and networks are introduced to large-scale commercial applications all over the world so as to fulfill the continuous emergence of new service patterns and explosive traffic increase. Especially in China, the number of 5G terminals is more than 450 million till the end of Sept. 2021 [1], regarded as one breakthrough in the first year of 'fourteenth five'. Since 2020, the academia community has started the emphasis on research beyond 5G (B5G) and 6G networks [2]. Moreover, various edge devices and terminals (such as vehicles, drones)[2] have evolved as key supplementary parts of B5G networks so as to expand the coverage and strengthen the processing abilities of the whole B5G networks. Traditional method of installing dedicated hardware appliances (middleboxes) for deploying new services is too costly. Hence, network function virtualization (NFV)[3] and network slicing (NS)[4] technologies have emerged to ease this burden. By adopting NFV and NS, network functions (e.g. firewall, load balancer) and resources (e.g. radio spectrum, CPU) of existing deployed hardware appliances can be virtualized, managed and allocated in an agile and flexible manner. Correspondingly, NFV and NS are widely accepted as the dominant enablers of the fundamental architecture of B5G networks.

Since NFV and NS for B5G have not been standardized, multiple technical issues are required to be solved. One dominant technical issue is the resource allocation in B5G networks, supporting both NFV and NS. Till 2021, abundant publications [5-12] exist in the literature. The exact algorithms, such as the integer linear programming (ILP) based algorithm, are proposed so as to get the optimal resource allocation per slice. The heuristics, such as the greedy method based, Markov model based, and topology attributes and resources method based [8], are proposed to calculate the feasible and sub-optimal resource allocation per slice within polynomial time. Though the resource allocation of virtual slices attracts extensive research attention from academia community, most existing publications focus on maximizing the slice acceptance and making the most use of physical resources. Existing research ignored the energy saving aspect. Known to all, minimizing total energy cost is vital and has a positive effect on maximizing the net profit of service providers. In recent years, some researchers [13-18] study the energy aspect of slices. For instance, Jang et al. [13] studies how to minimize total energy consumption of allocating resources to multiple slices. An ILP algorithm and a rounding-based heuristic algorithm are proposed. While in ref. [14], Kar et al. proposes an exact ILP and an efficient heuristic so as to deal with the energy-aware problem. Huang et al. [15] focuses on studying the placement and routing of virtual services in hybrid NFV-enabled networks. The object is to maximize the profit of total admitted traffic minus the energy cost and routing cost. A Markov Approximation based algorithm is proposed in ref. [15]. Eramo et al. [16] studies how to consolidate virtual nodes and shut down unused servers so as to minimize the total operation cost, including the energy consumption. The authors propose one exact algorithm (ILP based) and another heuristic to solve the virtual nodes placement. In ref. [17], Eramo et al. formulates the energy-aware resource allocation problem. Eramo et al. proves that the resource allocation problem is NP-hard. Hence, the heuristic is proposed instead. In the heuristic

The work of this paper was partly supported by National Natural Science Foundation of China under Grant 62071246 and 92067201, Key Research and Development Program of Jiangsu Province under Grant BE2020084-5 and BE2020084-1. *(Corresponding author: Longxiang Yang.)*

Haotong Cao is with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR 999077, China. Email: haotong.cao@polyu.edu.hk.

Haitao Zhao and Longxiang Yang are with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. Email: zhaoht@njupt.edu.cn, yanglx@njupt.edu.cn.

Anish Jindal and Gagangeet Singh Aujla are with Department of Computer Science, Durham University, United Kingdom. Email: anish.jindal@durham.ac.uk, gagangeet.s.aujla@durham.ac.uk.

algorithm, switching off extra idle servers is included. In ref. [18], the consolidation based method is proposed so as to minimize the energy consumption. Though of interest, existing energy related studies concentrate on doing the slice allocation in either core network or computer network. The proposed algorithms belong to the static type and cannot deal with the dynamically coming slices. These publications ignore the random access network (RAN) part while doing the slice allocation. In the RAN part, the effect of terminal mobility in slice allocation must be considered. The terminal/edge node can be a vehicle, drone and so on. The consideration of terminal can expand the coverage of services and applications. Hence, it is vital to consider the joint parts of the whole network while doing the slice resource allocation.

On the above basis, we research the virtual resource allocation of slices in vehicles-assisted B5G networks in this paper, having the goals of saving total energy cost and achieving high slice acceptance. Since we concentrate on researching the virtual resource allocation of slices, the network functions (e.g. firewall, load balancer, address transformer) of both B5G networks and virtual slices are omitted in this paper. We firstly present the system models of B5G networks and virtual slices. Then, we model the energy cost of vehicles-assisted B5G networks. In order to realize the efficient virtual resource allocation per slice, we propose one energy efficient algorithm, abbreviated as *Ener-Eff-Slice* in this paper. When receiving the virtual slice request, our *Ener-Eff-Slice* will select the suitable physical nodes in each part of B5G networks to do the resource allocation. Especially, the active physical nodes are selected in priority. Take note that the types of allocated resources include both wireless and wired while most of previous publications consider the wired type [5]. To validate the merits of our *Ener-Eff-Slice*, we do the evaluation work. Two derived counterparts of *Ener-Eff-Slice* are selected for performance comparison. Numerical results are plotted and discussed so as to demonstrate the energy efficiency of our *Ener-Eff-Slice* algorithm.

Major contributions of this paper are presented below:

1) System models of vehicles-assisted B5G networks and virtual slices are presented in this paper. Previous research of NFV and NS did not research the system model when end vehicles are incorporated. In addition, the energy cost model of vehicles-assisted B5G networks is formulated. Most of previous researchers [5-7] ignored saving the energy cost while doing virtual resource allocation of slices.

2) Types of allocated resources include both wireless type (radio spectrum) and wired type (CPU, storage) in this paper. In addition, one important QoS performance metric, delay, is considered and optimized in this paper. Previous researchers mainly focused on allocating wired resources and ignored considering the QoS performance per allocated slice [5].

3) An energy efficient algorithm, abbreviated as *Ener-Eff-Slice*, is proposed so as to allocate virtualized resources of slices in vehicles-assisted B5G networks. When receiving per slice, our *Ener-Eff-Slice* can select the suitable physical nodes, having abundant resources, to accommodate the virtual nodes per slice. In addition, our *Ener-Eff-Slice* can make the most use of active physical nodes and activate inactive nodes as few as possible.

4) The evaluation work of *Ener-Eff-Slice* is done in this paper. Two derived counterparts, abbreviated as *DerivedOne* and *DerivedTwo*, are selected for performance comparison. By analyzing the numerical results, our *Ener-Eff-Slice* outperforms two counterparts, in terms of the energy cost.

The rest of this paper is organized as follows. The system model and energy cost model for virtual slice and vehicles-assisted B5G networks are presented in Section II. In Section III, technical details of the *Ener-Eff-Slice*algorithm are presented. The evaluation work is conducted in Section IV. In Section V, the conclusion marks are presented.

II. System Model and Energy Cost Model for Vehicles-Assisted B5G Networks

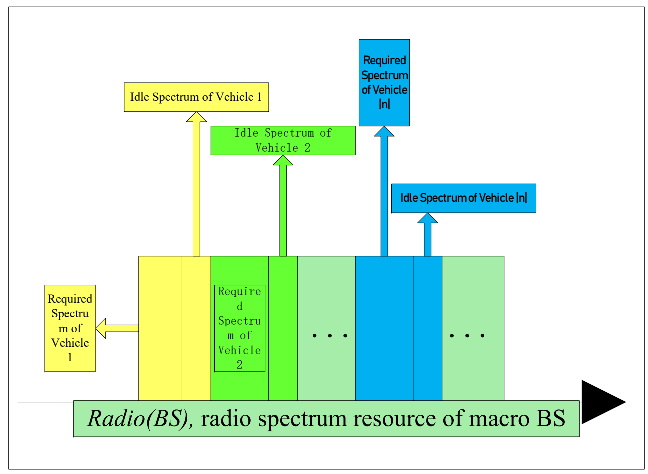

Fig. 1: Spectrum Resource Blocks Illustration of Macro BS

A. System Models for Vehicles-Assisted B5G Networks and Virtual Slices

In this paper, we adopt the graph theory [19] to model the vehicles-assisted B5G networks, abbreviated as $B5GNetworks$. $B5GNetworks$ is composed of three major parts: RAN part, transmission network (TRN) part, and core network (CN) part. Hence, we further model it as $B5GNetworks = (RAN(B5G), TRN(B5G), CN(B5G))$. With respect to the $RAN(B5G)$, only one cellular network is considered in this paper. Within the cellular network, one macro base station (BS) is included so as to be limited to one converged area. On this basis, multiple cellular networks can be further extended. With respect to the macro BS [20], it adopts the single input single output (SISO) scheme in this paper. Other schemes will be researched in the future study. That is to say, the spectrum efficiency is not taken into account. The radio resource allocation in the downlink of the cellular network is considered to be allocated in this paper. Hence, the concrete spectrum band is not considered. With respect to the macro BS, its available radio spectrum resource is labeled as $Radio(BS)$. Within the macro BS,

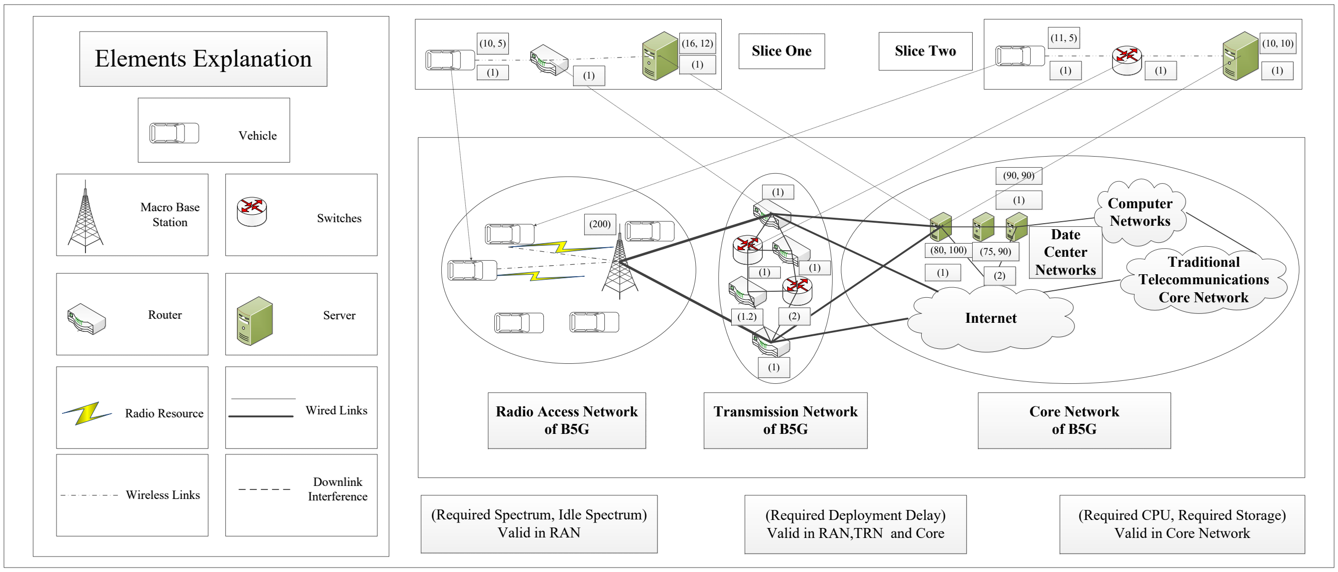

Fig. 2: System Models of Vehicles-Assisted B5G Networks and Two Virtual Slices Examples

multiple terminals are usually included. In this paper, mobile vehicles are considered as terminals. Concrete trajectories of vehicles are not considered. All vehicles run in low-speed and are within the coverage of the macro BS all the time. The spectrum resource of the macro BS $Radio(BS)$ will be divided into multiple radio resource blocks (Fig. 1), according to the tailored demands of its connected vehicles. With respect to each mobile vehicle, its assigned block consists of two parts: required spectrum part, idle spectrum part. The major function of the idle spectrum part is to avoid the interference between different vehicles. With respect to the $TRN(B5G)$, it consists of multiple routers and switches. Within the $TRN(B5G)$, there exist multiple inner physical links connecting the routers and switches. Take note that the whole $TRN(B5G)$ just has the function of forwarding and traversing data and traffic flow in this paper. We do not consider allocating the computing and storage resources of routers and switches in this paper. With respect to the $CN(B5G)$, it consists of multiple wired networks, such as the Internet, data center networks, computer networks, and telecommunications core network and so on [2]. These wired networks are composed of multiple abstracted physical nodes. These physical nodes are the abstraction of physical elements, such as servers, computers. With respect to each physical node in $CN(B5G)$ (e.g. M), it has CPU $CPU(M)$ and storage $Storage(M)$ resources to be allocated to virtual slices. Within the $CN(B5G)$, abstracted physical nodes are connected by inner physical links. In this paper, $RAN(B5G)$ part connects to $TRN(B5G)$ part via inter physical links, also known as backbone links. The backbone links are usually equipped with abundant bandwidth resources so as to transmit the traffic flow. $TRN(B5G)$ connects to $CN(B5G))$ via backbone links, too. In this paper, we do not consider the link bandwidth resource allocation. In addition, the deployment delay of vehicles-assisted B5G networks is considered. The deployment delay of one vehicle (e.g. A) is abbreviated as $Delay(A)$. The deployment delay of one transmission node B is abbreviated as $Delay(B)$ while the deployment delay of one core node M is abbreviated as $Delay(M)$.

With respect to the virtual slices, we adopt the graph theory to model them. Slices are usually requested dynamically and individually by contracted users. In virtualization and slicing research, slices arrive, following the known Poisson distribution. The arriving rate is labeled as α. All slices are set to have the line topology in this paper. With respect to the ith slice, it is labeled as $Slice(i)$. The $Slice(i)$ consists of three virtual nodes $rannode(Slice(i))$, $transnode(Slice(i))$ and $corenode(Slice(i))$. There exists one logical link connecting $rannode(Slice(i))$ and $transnode(Slice(i))$. The logical link is labeled as $rantran(i)$. There exists one logical link connecting $transnode(Slice(i))$ and $corenode(Slice(i))$. The logical link is labeled as $trancore(i)$. With respect to the $rannode(Slice(i))$, it is deployed and runs on top of a mobile vehicle which supports the virtualization and slicing schemes. Its required radio spectrum resource of $rannode(Slice(i))$ is labeled as $Radio(rannode(Slice(i)))$, having the function of transmitting data and traffic flow. With respect to the $transnode(Slice(i))$, it does not have any required resource in this paper. The function of $transnode(Slice(i))$ is to forward and traverse the traffic flow. With respect to $corenode(Slice(i))$, its required resources are CPU $CPU(corenode(Slice(i)))$ and storage $Storage(corenode(Slice(i)))$. In addition, we select the service delay of slice ($Delay(Slice(i))$) as the QoS parameter [5] in this paper. $Delay(Slice(i))$ is the sum of deployment of three virtual nodes, labeled as $Delay(rannode(Slice(i)))$, $Delay(transnode(Slice(i)))$ and $Delay(corenode(Slice(i)))$. In order to well understand the system model of vehicles-assisted B5G networks, we plot Fig. 2. Within the Fig. 2, one underlying vehicles-assisted

B5G network and two virtual slices are included. Resources and QoS attributes are highlighted, too. We do not describe the allocation results of both slices. Readers can refer to Fig. 2 and can easily find that all resource and QoS requests of both slices are satisfied.

B. Energy Cost Model for Vehicles-Assisted B5G Networks

In this sub-section, we will present the energy cost model for vehicles-assisted B5G networks. The energy cost model consists of three major parts: RAN energy cost part, TRN energy cost part, and CN energy cost part. In this paper, we focus on researching the algorithm performance. Hence, we do not consider the energy prices of different regions and cities. The energy price is set to be a constant value in the evaluation section. In further research, the energy prices of different regions will be considered. In this paper, we do not consider the temperature related factors in the energy cost. We focus on formulating and quantifying the energy cost of deploying the virtual slice (virtual end vehicle - virtual transmission node - virtual core node) onto the B5G networks supporting the slicing and virtualization schemes. The temperature related factors have little effect on the slice deployment and resource allocation [21]. Hence, temperature related factors are not included in this paper. In addition, we aim at optimizing the total energy cost for deploying slices and maintaining slices acceptance high. If extending the optimization goal, such as constructing the net profit model, the temperature related factors are necessary to be considered.

With respect to the energy cost model of RAN part, it concentrates on the energy cost of the macro BS that connects to vehicles. Though connected vehicles consume energy, they are owned by users. The service providers, responsible for realizing the slices, do not need to afford the energy cost of connected vehicles. With respect to the macro BS, it mainly focuses on providing spectrum resources and help transmitting data of virtual slices. Derived from ref. [21], the energy cost of RAN part can be formulated below:

$$\boldsymbol{P} = \begin{cases} \boldsymbol{P_{base}} + Constant \cdot Data, \\ \text{macro BS is powered up and utilized;} \\ \boldsymbol{P_{base}}, \\ \text{macro BS is powered up and idle;} \\ 0, \quad \text{macro BS is not powered up.} \end{cases} \tag{1}$$

where P_{base} is the baseline energy cost of the macro BS. $Constant$ is a constant that is adopted to record the unit energy cost of transmitting data. $Data$ is a variable that records the amount of occupied data. In this paper, we simplify it as the required spectrum. In the future, we will consider introducing the Shannon expression [22] that can be used to express the relationship between the occupied spectrum and maximum achievable data rate.

With respect to the energy cost model of the TRN part, it focuses on the energy cost of routers and switches. In this paper, the TRN part simply has the function of forwarding data and traversing the traffic flow. Hence, we will record the energy cost of forwarding data and traffic flow. Derived from ref. [21], we formulate the Expression (2) below:

$$\boldsymbol{Pow} = \begin{cases} \boldsymbol{Pow_{base}} + Const \cdot Number, \\ \text{transmission node is powered up and utilized;} \\ \boldsymbol{Pow_{base}}, \\ \text{transmission node is powered up and idle;} \\ 0, \quad \text{core node is not powered up.} \end{cases} \tag{2}$$

where Pow_{base} is the baseline energy cost of the transmission node (router or switch). $Const$ is a constant that is adopted to record the unit energy cost of transmitting the data and flow. $Number$ is a variable that is adopted to record the times of the transmission node being by the slices. Derived from ref. [21], this $Number$ variable is not related to the processing or transmitting time, such as the data package transmitting time. In addition, considering the slice lifetime, the occupied node will be released to be idle. Unitil new slice is deployed upon the node, it will only have the baseline energy cost. Thus, the value of $Number$ will not be infinite.

With respect to the energy cost model of CN part, it concentrates on the energy cost of core nodes. In this paper, we adopt the energy cost of server as given in [23]. It is owing to the fact that core nodes are the abstraction of servers. Derived from ref. [23], CPU utilization is the dominant factor of energy cost variations of a core node. If the physical core node is not powered up, its energy cost is 0. If the physical core node is powered up and in idle state, its energy cost is at its baseline level, labeled as $Power_{base}$. If the physical core node is powered up and accommodates certain one virtual core node, its energy cost is formulated in Expression (3) below:

$$\boldsymbol{Power} = \begin{cases} \boldsymbol{Power_{base}} + P^{'} \cdot CPU(corenode(Slice(i))), \\ \text{core node is powered up and utilized;} \\ \boldsymbol{Power_{base}}, \\ \text{core node is powered up and idle;} \\ 0, \quad \text{core node is not powered up.} \end{cases} \tag{3}$$

where $Power_{base}$ is the baseline of the core node. $P^{'}$ is the unit energy cost of the core node and $P^{'} = (Power_{max} - Power_{base})/CPU(M)$. M is the selected core node. Take note that the core node M must have abundant CPU resource to fulfill the required CPU of $corenode(Slice(i))$. With respect to other core node in the core network, the above energy cost model can be adopted. In the core network part, the storage resource is required to be allocated. Since storage resource has very little effect on the energy cost [21], it is not considered in Expression (2).

As introduced above, we present the energy cost model for vehicles-assisted B5G networks. Take note that we formulate the energy cost of network elements in RAN, TRN and CN parts of B5G networks. The energy cost for B5G is formulated from the side of the telecommunications providers and service providers who are responsible for constructing the infrastructure and developing the network services. Energy costs of the inserted users, such as the vehicles and end nodes, do not account in our formulated energy cost model. These inserted users are responsible for their own energy costs.

C. Major Performance Metrics

In this sub-section, we discuss the major performance metrics that will be used to quantify the proposed algorithm's allocation ability.

At first, it is the virtual slice acceptance ratio, abbreviated as $SliceRatio(Alg)$. Within the brackets, the name of the evaluated algorithm is included. As usual, this metric $SliceRatio(\ )$ can be classified into two types: one is the long-term type while the other is the short-term type. With respect to the long-term type, it measures the success rate of a certain one algorithm deploying slices in a long time period. With respect to the short-term type, it measures the success rate of a certain one algorithm deploying slices in one batch.

Secondly, it is the resource utilization, abbreviated as $ResouUtil(Alg_{type})$. Within the brackets, the name of the evaluated resource type is included. In this paper, utilizations of wired resources (CPU, storage) and wireless resource (spectrum resource) will be recorded and plotted. As usual, if the value of resource utilization is high, it can reveal that more slices are deployed successfully. Consequently, the slice acceptance ratio will be at a high level.

Thirdly, it is the energy cost, abbreviated as $EnergyCost(Alg_{Slice(i)})$, where the name of the evaluated algorithm is included in the brackets. This metric aims at quantifying the energy cost of the evaluated algorithm for accommodating $Slice(i)$. We formulate Expression (4).

$$\boldsymbol{EnergyCost(Alg_{Slice(i)})} = Price(t) \cdot (\int_{t_{start}^{Slice(i)}}^{t_{expiry}^{Slice(i)}} \mathbf{P} + \int_{t_{start}^{Slice(i)}}^{t_{expiry}^{Slice(i)}} \sum_{B \in TRN(B5G)} \sum_{transnode(i) \in Slice(i)} \mathbf{X}_B^{transnode(Slice(i))} \cdot \mathbf{Pow} + \int_{t_{start}^{Slice(i)}}^{t_{expiry}^{Slice(i)}} \sum_{M \in CN(B5G)} \sum_{corenode(Slice(i)) \in Slice(i)} \mathbf{Y}_M^{corenode(Slice(i))} \cdot \mathbf{Power}) \tag{4}$$

where $Price(i)$ is the function revealing the relationship between energy price and time. As usual, it is a time-varying function. In this paper, we set the energy price a constant (Section IV). X is the binary variable, revealing the relationship between virtual transmission node and physical transmission node. If the virtual node $transnode(i)$ is assigned to the physical node B, the value is 1, and vice versa With respect to Y, it is a binary variable, revealing the relationship between the virtual core node and physical core node.

With respect to the deployment revenue and deployment cost of $Slice(i)$, we do not introduce and formulate both metrics in this paper. In ref. [24], the deployment revenue and deployment cost are detailed.

III. The Proposed *Ener-Eff-Slice* Algorithm

In this section, we detail the proposed *Ener-Eff-Slice* algorithm. For easy understanding, we select $B5G$ and $Slice(i)$ as example descriptions. A flow chart is also plotted in Fig. (3). Since $Slice(i)$ consists of three nodes: $rannode(Slice(i))$, $transnode(Slice(i))$ and $corenode(Slice(i))$, its energy efficient resource allocation will be divided into three parts. Hence, three sub-sections will be described in order. In the fourth sub-section, we will further discuss the *Ener-Eff-Slice* algorithm.

A. Energy Efficient Resource Allocation of RAN Part

This sub-section details the resource allocation of $rannode(Slice(i))$. As introduced in Section II-A, $rannode(Slice(i))$ will be deployed on top of a selected physical vehicle, supporting both NFV and NS. Hence, the deployment delay of the selected vehicle must not more than the required deployment delay of $rannode(Slice(i))$, $Delay(rannode(Slice(i)))$. If no suitable vehicle can fulfill the $Delay(rannode(Slice(i)))$, the deployment of $rannode(Slice(i))$ fails. Hence, $Slice(i)$ will be rejected to serve. The $(i+1)$th slice will continue to be served.

If $Delay(rannode(Slice(i)))$ can be fulfilled and $randnode(Slice(i))$ connects to the selected vehicle successfully, the resource allocation of $rannode(Slice(i))$ continues. In this paper, the need for the radio spectrum resource of $rannode(Slice(i))$, $Radio(rannode(Slice(i)))$, is required to be fulfilled. We will check the available spectrum resource of macro BS $Radio(BS)$. If the sum of $Radio(rannode(Slice(i)))$ and fixed idle of $rannode(Slice(i))$ is not exceeding macro BS's available spectrum resource, the $randnode(Slice(i))$ can be allocated successfully. Otherwise, the resource allocation of $randnode(Slice(i))$ fails. The $Slice(i)$ will be rejected. With respect to the energy cost of RAN part, we will discuss it from two aspects. If $i = 1$, the macro BS is powered up to supply spectrum resources. Hence, the energy cost of $rannode(Slice(i))$ consists of the P_{base} and data transmission energy cost. If $i >= 2$, the energy cost of $rannode(Slice(i))$ is data transmission energy cost (Expression (1)).

By connecting to the macro BS and allocating the allocated spectrum resources, the resource allocation of $randnode(Slice(i))$ is done. We will turn to the resource allocation of $corenode(Slice(i))$. Since the allocation of $transnode(Slice(i))$ does not include concrete wired and wireless resources, we conduct the $transnode(Slice(i))$ allocation in the third sub-section.

B. Energy Efficient Resource Allocation of CN Part

This sub-section concentrates on selecting the suitable core node from $CN(B5G)$ and allocating wired resources (CPU, storage) to fulfill the demands of $corenode(Slice(i))$. In addition, the deployment delay of $corenode(Slice(i))$ must be guaranteed.

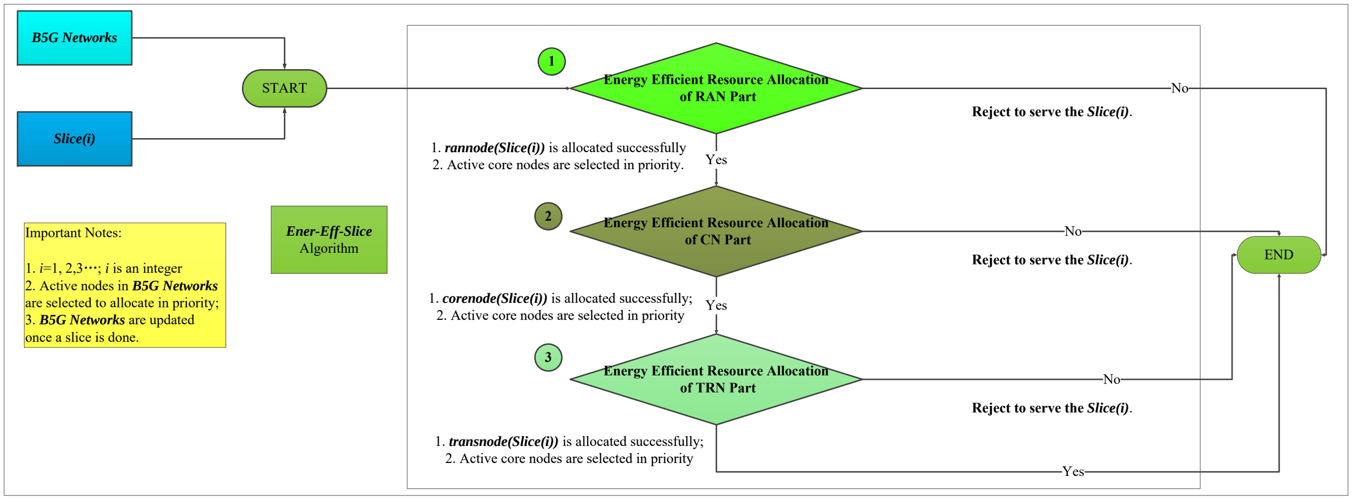

Fig. 3: Flow Chart of The Proposed *Ener-Eff-Slice* Algorithm

At first, we will find all active core nodes in $CN(B5G)$. We define one node set, labeled as SET. The SET is used to store all active core nodes in $CN(B5G)$. Then, we will adopt the Markov random method [24] to calculate the stable values of all active core nodes. We aim at scoring all active node's resource allocation abilities. As usual, the direct product of resource sum (CPU and storage) and node degree (Expression (5))[24] can be used to score node resource ability. However, in many cases, such as the sparse networks [23], the direct product method does not work and is not accurate. Hence, we adopt the Markov random method [24] in this paper.

$$\boldsymbol{Value\,(M)} = (CPU(M) + Storage(M)) \cdot Degree(M) \tag{5}$$

were $Degree(M)$ refers to the node degree attribute of M, revealing the total number of direct links of M [24]. $Degree(M)$ can directly reveal the 'connectivity' degree of node M. The sum of $CPU(M)$ and $Storage(M)$ can directly reveal the equipped and available resources of M that can be used to accommodate the virtual core slice node. On these basis, we select these direct attributes to calculate the node values. In addition, the known Markov method can further converge core node M's resource allocation accurately through the calculation. In next-step research, more direct attributes can be inserted to reveal resource allocation abilities, but not within the scope of this paper.

As we aim at calculating resource values of active nodes in limited time, we further adopt an iterative-based method and control the iteration number. With respect to the concrete procedures of iterative method, please refer to our previous publication [25]. By a number of calculations, resource values of all active nodes in $CN(B5G)$ can be achieved.

Afterwards, we will re-sort all active core nodes in SET, following the descending order. Next, we will start to select the suitable core node to accommodate $corenode(Slice(i))$. The core node M, having highest resource value, is selected in priority. If M has abundant CPU and storage to fulfill the demands of $corenode(Slice(i))$, we will further compare the deployment delay. If $Delay(M)$ is not more than $Delay(corenode(Slice(i)))$, $corenode(Slice(i))$ will be deployed on top of M. If certain one demand of $corenode(Slice(i))$ (CPU, storage, delay) is not fulfilled by M, we will select the core node, having second highlight value, to attempt. Repeat this scheme until $corenode(Slice(i))$ is deployed successfully on top of an active core node.

If no active core node can accommodate the $corenode(Slice(i))$, we will select the most suitable inactive core node to accommodate the $corenode(Slice(i))$. We define another node set $SET1$ for storing all inactive core nodes. We will repeat the Markov random method to calculate resource values of all inactive nodes. With respect to remaining procedures, they are the same as the procedures of the active nodes. We adopt the scheme until it finds one suitable inactive core node to accommodate and allocate resources to $corenode(Slice(i))$. If no inactive node in $SET1$ fulfills the resource and delay demands of $corenode(Slice(i))$, the $Slice(i)$ will be rejected to serve.

Take note that we consider all active nodes in priority. We prefer making the most use of active nodes and avoid powering up new inactive core nodes. In addition, we present the pseudo code of the energy efficient resource allocation of $corenode(Slice(i))$ (**Algorithm 1**).

C. *Energy Efficient Resource Allocation of TRN Part*

In this sub-section, we focus on doing the resource allocation of $transnode(Slice(i))$ and saving energy cost of this TRN part.

In the first two sub-sections, the $rannode(Slice(i))$ and $corenode(Slice(i))$ are deployed on top of suitable vehicle node and core node, fulfilling their resource and delay demands and saving energy cost. There must exist loop-free physical paths between the macro BS and core node. We will adopt the known shortest path (SP) method [26] to find the

Algorithm 1 Energy Efficient Resource Allocation of $corenode(Slice(i))$

Input: $CN(B5G)$, $corenode(Slice(i))$
Output: Updated $CN(B5G)$, Allocation Results of $corenode(Slice(i))$.

```
Find all active core nodes in CN(B5G) and define one node set SET for storing them;
Adopt the Markov random method [24][25] to calculate stable values of all active nodes in SET;
Re-sorting all active node in SET, following the descending order of stable values;
Define a variable K, and set K = 0;
Define a variable Num, and the initial value of Num is the number of core nodes in SET;
while Num ≥ K do
    Select the core node which has the highest value in SET to compare with the corenode(Slice(i)), in terms of CPU, storage, and delay requirements;
    if all requirements of corenode(Slice(i)) are fulfilled then
        Allocate the core node to corenode(Slice(i)), output the allocation results of corenode(Slice(i));
        if certain one requirement of corenode(Slice(i)) cannot be fulfilled by the core node then
        Remove the active core node from SET;
    end if
    K++;
end while
Store all inactive core nodes in CN(B5G) in a new node set SET1 for storing them;
Repeat the procedures to find suitable core node to accommodate corenode(Slice(i));
Update CN(B5G) and output allocation results of corenode(Slice(i)).
```

shortest path. The selection criterion of selecting the shortest is the minimum number of intermediate nodes.

With selecting the shortest physical path, we will define another node set $SET2$ for storing all transmission nodes in the shortest path. Then, we will select the nodes one by one and compare with the $transnods(Slice(i))$. If the $Delay(transnode(Slice(i)))$ can be fulfilled by one selected transmission node, the deployment of $transnode(Slice(i))$ is done. If no transmission node in the shortest path can accommodate $Delay(transnode(Slice(i)))$, we will select the second shortest path to attempt. Remaining procedures are same as the above procedures.

Until the $transnode(Slice(i))$ is deployed successfully, the deployment and resource allocation of $Slice(i)$ is done. We will continue to serve the next $(i + 1)$th slice.

D. Extra Discussion of Ener-Eff-Slice Algorithm

In this sub-section, the complexity of *Ener-Eff-Slice* is briefly discussed. As discussed in above three sub-sections, the complexity of *Ener-Eff-Slice* consists of three part. With respect to the first part, its complexity is determined by the number of vehicles within the coverage of macro BS. Hence, this part can be completed within polynomial time. With respect to the second part, its complexity lies in the iterative method. Derived from ref. [26], this part is guaranteed to be completed within polynomial time. With respect to the third part, it can be seen as a procedure of finding SP between two fixed nodes. This procedure is promised to be completed within polynomial time [26]. Hence, our *Ener-Eff-Slice* algorithm can do the energy resource allocation of slices within polynomial time.

In addition, the processing slices strategy of our *Ener-Eff-Slice* is discussed. Two main principles constitute the processing slices strategies. The first principle is the slice arriving time. The processing orders of slices are determined by their arriving time. The slice arriving earlier has the priority to be deployed in priority. The second principle is the maximum waiting time. Known to all, each slice is generated by the user and has its maximum waiting time. If the waiting time violates the maximum value, the slice will expire. Though sacrificing some slices and earning benefits, all slices have the equal right to be served. It indicates that each slice being deployed and allocated will not be interrupted. Take note that another slice set is constructed so as to store the waiting slices. When the previous slice is deployed, no matter successful or failed, the next one continues. If a certain slice's waiting time is violated, it will be cleaned from the set.

IV. Evaluation of Our *Ener-Eff-Slice* Algorithm

A. Evaluation Parameters Settings

The $B5G$ networks consists of three parts: $RAN(B5G)$, $TRN(B5G)$, and $CN(B5G)$. Within the $RAN(B5G)$, only one macro BS is included. The available spectrum resource of the BS is set to be 100. Ten vehicles in total are within the coverage of the macro BS. The deployment delay of each vehicle is an integer with the uniform distribution $[1, 3]$. The idle spectrum is for each connected vehicle node set to be 2. With respect to the $TRN(B5G)$, the number of transmission nodes is 20. With respect to each pair of transmission nodes, the connecting possibility is 0.5. With respect to each transmission node, its deployment delay is an integer, following the uniform distribution $[1, 3]$. With respect to $CN(B5G)$, the number of core nodes is set to be 80. CPU and storage of each core node are integers, both uniformly distributed between 80 and 100. The connecting possibility per node pair is 0.5.

With respect to each virtual slice, the arriving rate α is set to be 4 per 100 unit time. The evaluation will last up to 10000 unit time. In total, 400 slices will be required to be processed. With respect to each slice, it consists of three nodes. Virtual RAN node connects to virtual transmission node while virtual transmission node connects to virtual core node. The required spectrum of a virtual RAN node is an integer, following the uniform distribution between 5 and 8. The required deployment delay of virtual RAN node is an integer, following the uniform distribution $[2, 5]$. With respect to the virtual transmission node and virtual core node, their required deployment delay are integers, following the uniform distribution $[2, 5]$. The required CPU and storage of virtual core node are integers, both uniformly distributed between 20 and 40. The average lifetime per slice is set to be 100 unit time. With respect to energy cost settings, such as P_{base}, they are same to the settings in ref. [27]. With respect to resource values calculation, the settings are same to ref. [25].

With respect to the selected algorithms for comparison, they are derived from our *Ener-Eff-Slice*. They are abbreviated as *DerivedOne* and *DerivedTwo*. The difference between *Ener-Eff-Slice* and *DerivedOne* is the value calculation in the core part. *DerivedOne* uses Expression (5) directly. The difference between *Ener-Eff-Slice* and *DerivedTwo* is the allocation order. *DerivedTwo* algorithm does the TRN allocation before doing the CN allocation. Since the energy related research of deploying virtual slices in vehicles-assisted B5G networks is at its early stage, there are not abundant algorithms in the literature. We have to use the derived versions as the compared algorithms in this paper.

Owing to the fact that energy related research in vehicles-assisted B5G networks for deploying virtual slices is in its infancy, no prototype has been developed or open by both research community and industry [2][4][5]. Hence, we have to evaluate our proposed *Ener-Eff-Slice* algorithm in the ad-hoc simulator. With extending research in this direction, a prototype would soon be developed in the foreseeable future.

B. Numerical Evaluation Results

In this sub-section, we plot and discuss the dominant numerical results (Fig. (4) and Fig. (5)).

In Fig. 4 (a), the slice acceptance ratio results of three compared algorithms are plotted. As described in Section II-C, slice acceptance ratio is the dominant performance metric to evaluate the resource allocation ability of the algorithm. If the value of acceptance ratio of certain one algorithm is high, it indicates that the algorithm has strong allocation ability. By analyzing Fig. 4 (a) carefully, we can get two conclusions. The first conclusion is that all algorithms will achieve the balance between new arriving slice requests and finite physical resources in the long term. The reason is very apparent. In the beginning stage of evaluation, the available resources (wired and wireless) are abundant. Hence, most of the new arriving slices will be allocated successfully. The slice acceptance ratio will remain at a high level. With evaluation extending, the available resources will approach the shortage. Hence, the new arriving slices cannot be served and allocated. The acceptance ratio will decrease. Eventually, the balance between the available resources and new arriving slices will be achieved. Derived from Fig. 4 (a), three algorithms undergo the similar behaviors.

With respect to the second conclusion, it is that our *Ener-Eff-Slice* performs much better than two derived algorithms. By comparing *Ener-Eff-Slice* with *DerivedOne*, the cause of the performance advantage is the method of calculating core nodes' values. Our *Ener-Eff-Slice* adopts the iterative and stable method while *DerivedOne* adopts the direct product method. Our *Ener-Eff-Slice* enables to select the core nodes having accurate resource values to allocate resources to slices. However, *DerivedOne* uses the nodes having high resource

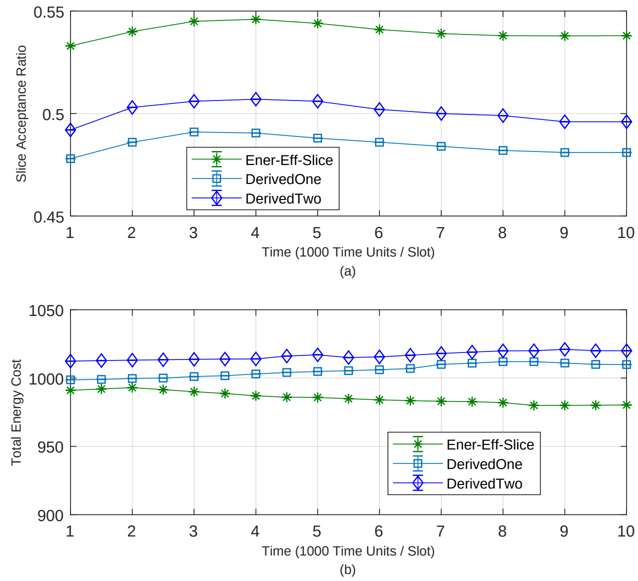

Fig. 4: Slice Acceptance Ratio and Total Energy Cost Results of Three Compared Algorithms

sum to deploy slices. When more slices come, the resource shortage comes earlier than the *Ener-Eff-Slice*. Hence, the acceptance of *DerivedOne* will be lower than that of our *Ener-Eff-Slice*. Comparing *Ener-Eff-Slice* with *DerivedOne*, the allocation order is the main difference. *DerivedTwo* does the TRN allocation before doing CN allocation. Our *Ener-Eff-Slice* does the CN allocation before doing the TRN allocation. The strategy of *DerivedTwo* will lead to more resource consumption and extra energy cost. Slices considered in this paper are made of our three nodes and have a line topology. If we process the first two ordered nodes, there will have more space for the third node. More extra resources will be consumed.

In Fig. 4 (b), we plot the energy cost results of three algorithms. By analyzing Fig. 4 (a), we can draw two inferences. The first inference is that the energy cost of all three algorithms are similar in the early stage of the evaluation and the second inference is that the energy cost of our *Ener-Eff-Slice* is at the lowest level in the long term. The reason for both conclusions is that all physical nodes in $B5G$ networks are inactive at the beginning of the evaluation. When starting to accept slices, inactive physical nodes will be powered up. Hence, all three algorithms consume a similar amount of energy. As evaluation time extends, more slices will be requested. Our *Ener-Eff-Slice* can make the most use of active nodes, especially to the active core nodes. When no active node has enough CPU and storage, new inactive nodes will be powered up. If powered, new active nodes will be made full use of. Unlike the remaining two algorithms, more inactive nodes are usually powered up in priority. Hence, two derived algorithms consume more energy than our *Ener-Eff-Slice*.

CPU utilization and storage utilization results of three algorithms are plotted in Fig. 5 (a) and Fig. 5 (b), respectively. Derived from both figures, we can easily find that resource utilizations of three selected algorithms increase throughout

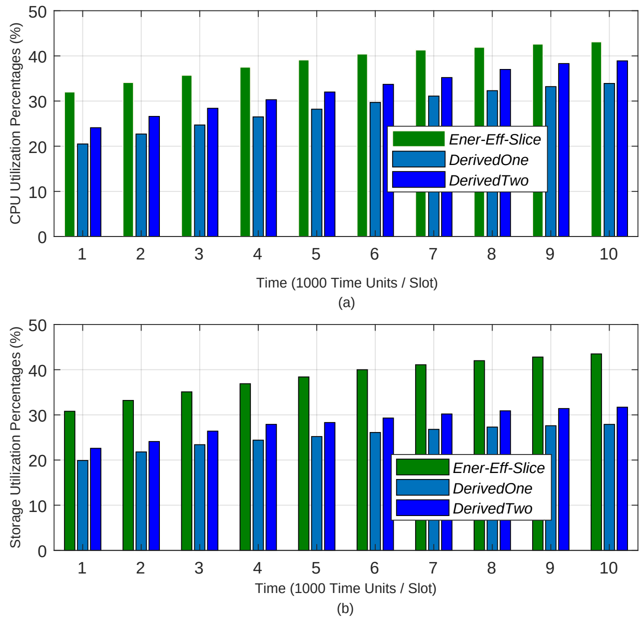

Fig. 5: CPU and Storage Utilization Results of Three Compared Algorithms

the whole evaluation. With evaluation work extending, the speed of resource utilization will decrease to be stable. Derived from both figures, we can easily find that our *Ener-Eff-Slice* consumes the largest amount of resources to deploy and allocate virtual slices. The reason for this finding is very simple: deploying more slices will lead to more resource consumption. Hence, Fig. 5 (a) and Fig. 5 (b) directly prove *Ener-Eff-Slice*'s stronger ability of deploying and allocating resources to slices.

V. Related Work

Currently, 5G communications systems and networks are being standardized for large-scale commercial application. B5G [2] is continuing to advance by academia and industry simultaneously. Edge devices and terminals, such as the vehicles, are inserted so as to expand the network coverage [28] and provide the seamless service [29][30] in 6G. Vehicles can be regarded as the integration of multiple computation and communication modules, aiming at breaking the restriction of 'last one mile' [31]. Generally speaking, the vehicles-assisted B5G networks can provide more tailored and novel services and applications. To get rid of the shortcomings of traditional private networks, NFV and NS [4] will be adopted by the vehicle-assisted B5G networks. In that case, specific functions and resources of the B5G networks can be managed and maintained in a generalized manner. However, multiple technical issues stand in the way of its successful implementation. One dominant technical issue is the resource allocation of virtual services [5], usually modeled by slices, in vehicles-assisted B5G networks.

Though abundant technical publications [5][6][7] exist in the literature, most of them are proposed, having the goal of maximizing the number of successfully deployed slices [8][9][10] or minimizing the consumed time of acquiring one virtual service solution [11][12]. For instance, the exact (ILP)[6], the heuristic [8] and the meta-heuristic [9] methods are adopted to do the virtual resource allocation. In 2020 and beyond, energy conservation is crucial in all fields, including the information and communications technology (ICT) field. Thus, it is vital to research the energy efficient virtual resource allocation of slices in vehicles-assisted B5G networks. However, there is a lack of related studies and publications in the literature. Though relevant energy related studies [13-18][32][33][34][35] have published in recent years, they have three major shortcomings. With respect to the first shortcoming, it is the limitation of allocated resources. Existing studies focus on either wireless (e.g. spectrum, power) or wired (e.g. CPU, bandwidth) resources. None considers the joint wireless and wired resources. With respect to the second shortcoming, it is the lack of energy cost model of vehicles-assisted B5G networks. With respect to the third shortcoming, it is the restriction of application scenario. Existing studies are limited in either the RAN part or the CN part. None considers the joint parts. Concerning RAN part survey is presented in ref. [35]. Hence, it is of great essence to research the energy efficient slice resource allocation in vehicles-assisted B5G networks.

To address these problems and shortcomings, we research the virtual resource allocation of slices in vehicles-assisted B5G networks. We firstly propose and formulate the energy cost model of deploying slices in vehicles-assisted B5G networks. Within the B5G networks, energy cost model of three parts are carefully formulated and presented in sequence. Then, we propose one *Ener-Eff-Slice* algorithm, having the goal of minimizing the energy cost of deploying slices and maintaining acceptance high. When receiving one slice, our *Ener-Eff-Slice* will select the strongest active physical nodes and make the most of active physical nodes to deploy the slice. Meanwhile, the wireless and wired requests of the slice will be fulfilled. If no suitable active nodes have available resources, inactive nodes will be powered to do resource allocation. This procedure aims at minimizing extra energy cost, especially for avoiding the extra baseline cost. To validate the *Ener-Eff-Slice* efficiency and merits, we conduct the evaluation in the simulation form. We derive two major counterparts of *Ener-Eff-Slice* as the compared algorithms. By doing the simulation, our *Ener-Eff-Slice* achieves the efficient energy cost while keeping the slice acceptance advantage. In addition, we illustrate the resource consumption results, aiming at highlighting our *Ener-Eff-Slice* merits. As we know, this is the first attempt to research the energy cost of slice allocation in vehicles-assisted B5G networks in the literature [2][36].

VI. Conclusions

In this paper, we research the virtual resource allocation of slices in vehicles-assisted B5G networks. We propose one energy-efficient virtual resource allocation algorithm, abbreviated as *Ener-Eff-Slice*. The goals of *Ener-Eff-Slice* algorithm are not only minimizing the total energy cost, but also guaranteeing the slice acceptance ratio high. We do the evaluation work so as to validate the energy efficiency of *Ener-Eff-Slice*.

Two derived counterparts of *Ener-Eff-Slice* are selected for performance comparison. Evaluation results vividly reveal that our *Ener-Eff-Slice* outperforms two compared algorithms, in terms of total energy cost and resource utilization.

In the future, we will further consider the broadcasting nature of radio spectrum resources in the RAN part of B5G networks. We will consider the effect of mobility of vehicles while doing slice resource allocation, too. In this paper, all vehicles move at a low speed and within the coverage of its connected BS. While in the real environment, vehicles usually move at different speeds and direction and switch connected BSs or access points (APs) from time to time [37]. In addition, we intend to expand the scenario, such as the energy efficiency in agriculture 4.0 [38]. The research of virtual slices in the B5G network is in its infancy, worthy of more research attention.

REFERENCES

[1] CAICT, China's 5G Development and Economic and Social Impact, White Paper, 2021, December. http://www.caict.ac.cn/kxyj/qwfb/bps/202112/P020211207585963474968.pdf

[2] X. You, C. Wang, et al., "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," Sci. China Inf. Sci., vol. 64, no. 1, 110301, Jan. 2021.

[3] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. Turck and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 236-262, 2016.

[4] A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," Computer Networks, vol. 167, pp. 1-40, 2020.

[5] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, "Recent advances of resource allocation in network function virtualization," IEEE Trans. Para. Distr. Sys., vol. 32, no. 2, pp. 295-314, Feb. 2021.

[6] 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, 2018.

[7] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," IEEE Commun. Surveys Tuts., vol. 21, no. 2, pp. 1409-C1434, Second Quarter 2019.

[8] H. Cao et al., "Resource-ability assisted service function chain embedding and scheduling for 6G networks with virtualization," IEEE Trans. Veh. Tech., vol. 70, no. 4, pp. 3846-3859, April 2021.

[9] X. Fu, F. Yu, J. Wang, Q. Qi and J. Liao, "Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning," IEEE Commun. Magaz., vol. 57, no. 9, pp. 102-108, Sep. 2019.

[10] S. Kulkarni et al., "NFVnice: Dynamic backpressure and scheduling for NFV service chains," IEEE/ACM Trans. Netw., vol. 28, no. 2, pp. 639-652, Apr. 2020.

[11] Y. Hu, Y. Guo, "Survivable service function chain mapping in NFV-enabled 5G networks," in Proc. of IEEE NetSoft 2021, pp. 375-380, Jul. 2021.

[12] H. Cao, A. Xiao, Y. Hu, P. Zhang, S. Wu and L. Yang, "On virtual resource allocation of heterogeneous networks in virtualization environment: A service oriented perspective," IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 4268-4280, Dec. 2020.

[13] I. Jang, D. Suh, S. Pack, and G. Dcan, "Joint optimization of service function placement and flow distribution for service function chaining," IEEE J. Sel. Areas Commun., vol. 35, no. 11, pp. 2532-2541, Nov. 2017.

[14] B. Kar, E. H.-K. Wu, and Y.-D. Lin, "Energy cost optimization in dynamic placement of virtualized network function chains," IEEE Trans. Netw. Service Manag., vol. 15, no. 1, pp. 372-386, Mar. 2018.

[15] H. Huang, S. Guo, J. Wu, and J. Li, "Service chaining for hybrid network function," IEEE Trans. Cloud Comput., vol. 7, no. 4, pp. 1082-1094, Fourth Quarter 2019.

[16] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," IEEE/ACM Trans. Netw., vol. 25, no. 4, pp. 2008-2025, Aug. 2017.

[17] V. Eramo, M. Ammar, and F. G. Lavacca, "Migration energy aware reconfigurations of virtual network function instances in NFV architectures," IEEE Access, vol. 5, pp. 4927-4938, 2017.

[18] A. Marotta, F. D, Andreagiovanni, A. Kassler, and E. Zola, "On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures," Comput. Netw., vol. 125, pp. 64-75, 2017.

[19] A. F. Molisch, Wireless Local Area Networks, New York, NY, USA: Wiley-IEEE , 2012.

[20] C. Liang, F. R. Yu, et al., "Virtual resource allocation in information-centric wireless networks with virtualization," IEEE Trans. Veh. Technol., vol. 65, no. 12, pp. 9902-9914, Dec. 2016.

[21] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 732–794, 1st Quart., 2016.

[22] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge. U.K.: Cambridge Univ. Press, 2009.

[23] B.-G. Chun, G. Iannaccone, G. Iannaccone, R. Katz, G. Lee, and L. Niccolini, "An energy case for hybrid datacenters," ACM SIGOPS Oper. Syst. Rev., vol. 44, no. 1, pp. 76–80, 2010.

[24] H. Cao, S. Wu, Y. Hu, Y. Li, and L. Yang, "A survey of embedding algorithms for virtual network embedding," China Commun., vol. 16, no. 12, pp. 1-33, Dec. 2019.

[25] H. Cao, L. Yang, and H. Zhu, "Novel node-ranking approach and multiple topology attributes-based embedding algorithm for singledomain virtual network embedding," IEEE Internet Things J., vol. 5, no. 1, pp. 108–120, Feb. 2018.

[26] T. H. Cormen, C. Stein, R. Rivest, and C. Leiserson, Introduction to Algorithms, 2nd ed. McGraw-Hill Higher Education, 2001.

[27] H. Cao, S. Wu, Y. Hu, R. Mann, Y. Liu, L. Yang, and H. Zhu, "An efficient energy cost and mapping revenue strategy for inter-domain NFV-enabled networks," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5723-5736, Jul. 2020.

[28] A. Perveen, R. Abozariba, M. Patwary, A. Aneiba, and A. Jindal, "Clustering-based redundancy minimization for edge computing in future core networks," in Proc. of IEEE 4th 5G World Forum (5GWF), pp. 453-458, 2021.

[29] R. Gunasekaran, et al., "AI-powered blockchain-a decentralized secure multiparty computation protocol for IoV," in Proc. of IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 865-870, 2020.

[30] W. Wang et al., "Secure-enhanced federated learning for AI-empowered electric vehicle energy prediction," IEEE Cons. Elec. Mag., vol. pp, no. 99, pp. 1-1, Sept. 2021.

[31] S. B. Prathiba, G. Raja, K. Dev, N. Kumar, and M. Guizani, "A hybrid deep reinforcement learning for autonomous vehicles smart-llatooning," IEEE Trans. Veh. Tech., vol. 70, no. 12, pp. 13340-13350, Dec. 2021.

[32] B. Matthiesen, O. Aydin, and E. A. Jorswieck, "Throughput and energy-efficient network slicing," in Proc. of 22nd International ITG Workshop on Smart Antennas, pp. 1-6, 2018.

[33] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Energy-efficient deep reinforcement learning assisted resource allocation for 5G-RAN slicing," IEEE Trans. Veh. Tech., vol. 71, no. 1, pp. 856-871, Jan. 2022.

[34] K. Ramasamy, M. H. Anisi, and A. Jindal, "E2DA: Energy efficient data aggregation and end-to-end security in 3D reconfigurable WSN," IEEE Trans. Green Commun. Netw., vol. pp, no. 99, pp. 1-1, Nov. 2021.

[35] F. Marzouk, J. P. Barraca, and A. Radwan, "On energy efficient resource allocation in shared RANs: Survey and qualitative analysis," IEEE Commun. Surv. Tutor., vol. 22, no. 3, pp. 1515-1538, 2020.

[36] 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. Sur. Tut., vol. 20, no. 3, pp. 2429-2453, 2018.

[37] H. Cao, H. Zhao, D. X. Luo, N. Kumar and L. Yang, "Dynamic virtual resource allocation mechanism for survivable services in emerging NFV-enabled vehicular networks," IEEE Trans. Intell. Trans. Syst., vol. pp, no. 99, pp. 1-13, Oct. 2021.

[38] A. Ranjha, G. Kaddoum, and K. Dev, "Facilitating URLLC in UAV-assisted relay systems with multiple-mobile robots for 6G networks: A prospective of agriculture 4.0," IEEE Trans. Indus. Infor., vol. pp, no. 99, pp. 1-1, Nov. 2021.