Traffic Steering Based Anomaly Prevention for User Request Provision in 6G Network Slices

Zhao Ming*, Kai Dong*, and Tarik Taleb†
*University of Oulu, Oulu, Finland.
†Ruhr University Bochum, Bochum, Germany.
Email: {zhao.ming, kai.dong}@oulu.fi, tarik.taleb@rub.de

***Abstract*—Network slices face challenges in supporting dynamic requests from user equipments (UEs) due to the resource constraints at the edge devices. This may result in potential anomalies due to resource unavailability or latency spikes. Existing schemes are hard to apply due to the lack of flexible steering of UE requests and multiplexed provisioning strategies. In this paper, we present a novel request provisioning model and propose a traffic steering framework to prevent anomalies and reduce the cost of serving normal UEs. Specifically, we categorize the UE requests into stateful and stateless types and utilize a multi-route resource provisioning strategy to address the UE anomalies. Additionally, the framework incorporates bandwidth-aware route selection and load balancing across multiple routes to improve the service bandwidth for UEs. Simulation results demonstrate that the proposed framework effectively reduces both the number of anomaly UEs and cost compared to existing baselines.**

***Index Terms*—Network Slicing, User Traffic Steering, Anomaly Prevention, Load Balancing.**

I. Introduction

In recent years, rapid developments in network technologies have facilitated the integration of network slicing technology into 5G/6G systems, supporting diverse use cases such as Internet of Things, smart city, and edge computing [1]–[5]. Built on top of the physical networks, network slicing transforms the single physical network infrastructure into multiple independent virtual networks and enables network functions to be decoupled from dedicated facilities. Network slices can be flexibly configured and adjusted to support customized service quality assurance, achieve resource isolation for different business types, and cope with the dynamic and heterogeneous requests from users [6].

In supporting the requests from user equipments (UEs), network slices initialize virtual network functions (VNFs) on physical facilities such as edge servers (ESs) or routers, and UEs associated with the slices obtain resources from these virtual nodes. Unlike services provided directly by physical ESs, VNFs can be dynamically migrated to different physical hosts through customized slice settings, and elastically scale resources according to changes in user behavior [7]–[9]. However, the supply of requests from numerous users also faces challenges. The users' requested resources may be difficult to meet within the specified delay requirements, especially in the resource-constrained scenario of edge-cloud integration. Under these circumstances, it is vital to investigate efficient resource provisioning strategies to improve the overall quality of service (QoS) to UEs and cope with potential anomalies.

There has been lots of research about resource provisioning in network slices, for instance, Masoudi *et al.* proposed an energy-optimal end-to-end network slicing framework for cloud-based 5G networks [10]. They developed a slice-based resource allocation algorithm that jointly optimizes bandwidth and processing resources to minimize the total network energy consumption while meeting slice-specific delay requirements. Besides, the authors in [11] studied UE resource provisioning in 5G network slicing and proposed a two-phase framework to allocate bandwidth and computing resources for UEs from the infrastructure provider's perspective. Additionally, in [8], they proposed to reserve resources to meet the slice QoS when the UE number is uncertain and the resource demand fluctuates randomly, and demonstrated the advantages in improving revenue, resource utilization, and interference control. However, in real-world network environments, there can be multiple requests from multiple UEs in each time slot, these studies didn't consider the load balancing among multiple user requests. The requests can be more evenly distributed to different facilities, which helps to balance the allocation of system resources and thus improve the overall robustness to cope with the UE anomalies due to resource unavailability.

On the other hand, Qiao *et al.* discussed resource allocation in open radio access network slicing and proposed a multidimensional framework to improve heterogeneous QoS and reduce the resource cost of network service providers [12]. The framework implicitly dealt with the load-balance problem by considering the fair utilization of resources to improve the allocation efficiency in the spatiotemporal action scale. In our previous work [13], we proposed a prediction-based anomaly detection and resource provisioning method to detect the potential anomalies in network slices and improve the quality of experience of UEs by deciding the resource allocation strategies with UE information prediction. However, to prevent anomalies, UEs' requests can be segregated to be stateful/stateless to enable slices with more flexible resource provisioning strategies, as stateful requests associate with specific VNFs over time slots while stateless does not. Additionally, even for each single VNF that provides resources to UEs, it is still possible to introduce multiple routes formed by different routing nodes to improve the serving bandwidth,

which are rarely considered.

To address these issues, in this paper, we investigate more flexible traffic steering of resource provisioning in network slices, consider a novel model that combines stateful and stateless requests segregation, multi-route resource provisioning, and load balancing among different routes to prevent anomalies. We formulate the optimization problem of minimizing the number of anomaly UEs and the cost of mobile network operators (MNOs) for serving normal UEs. To solve this problem, we propose a traffic steering framework to generate candidate VNFs and routes for UEs with their requested resources, latency, and mobility pattern considered. Particularly, we utilize a bandwidth-aware route selection strategy to improve the serving bandwidth for UEs using a modified Dijkstra algorithm, which performs a different goal as finding the route with maximum bandwidth [14]. After that, we dynamically allocate resources and bandwidth from routes that satisfy the UEs' latency requirements in a scaled load-balancing manner to mitigate the bandwidth unavailability in specific nodes. Based on extensive simulations, we demonstrate the advantages of our proposed framework in terms of reducing the number of anomaly UEs and the cost of MNOs. The contributions of this paper are as follows:

- We consider a novel model for UEs' request provisioning, segregate the requests to be stateful and stateless, and utilize multi-route resource provisioning and dynamic load balancing strategies to cope with the UE anomalies caused by resource unavailability.
- We propose a flexible traffic steering framework to generate candidate VNFs and routes for UEs based on their requests. Particularly, the framework utilizes a modified Dijkstra algorithm to achieve bandwidth-aware route selection to improve the serving bandwidth for UEs.
- Simulation results demonstrate the advantages of our proposed framework in terms of reducing the number of anomaly UEs and the cost of MNOs.

The remainder of this paper organizes as follows. Sect. II shows the model and formulates the problem. In Sect. III, we propose the algorithm to solve this problem. Simulation results are in Sect. IV, and Sect. V concludes this paper.

II. System Model and Problem Formulation

A. System Model

We consider a general system model in an area for user traffic steering in 6G network slicing, as shown in Fig. 1. Specifically, at the bottom of the figure, there are multiple UEs geographically distributed in different areas and are connected to the base stations (BSs) or routers through cellular links as access points (APs). Here, the BSs are serving outdoor devices, and the routers are for indoor devices. Each BS is equipped with a small ES with limited resources like CPU, storage, and bandwidth, while the routers have similar resource constraints. These APs are connected to each other via high-speed optical links and to the core network by backhaul links, while the core network exchanges the data of multiple

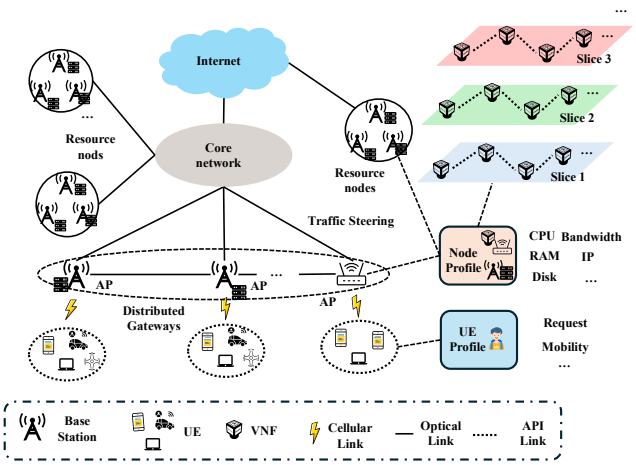


Fig. 1. The considered network architecture for user traffic steering.

BSs/routers. Particularly, for the BSs that are extremely far away or belong to different domains, the data needs to be routed by the Internet. We uniformly define the ESs and routers in the system as physical nodes.

To serve the heterogeneous requests from UEs, multiple VNFs are initialized on top of the physical nodes and are partially connected via virtual links built on top of the physical optical links to form network slices. The connected AP of each UE collects the information from UEs, which includes the requested amount of resources, the latency preferences, and the position of UEs. Afterward, all APs collaboratively decide the target VNFs and routing strategies for the UEs and steer the requests based on the decisions. Over time slots, the UEs change their request information, and the system tries to satisfy the new requests based on the slice configuration and decides the new policies for resource provisioning. When UEs' requests cannot be fully supported, anomalies may occur, and the slices need to be adjusted partially or globally [7]. We assume each UE can only be served by one slice during each time slot and neglect the details for slice initialization.

We denote the set of UEs in the system as $\mathcal{U}$, the requested resources, latency, and position of UE $u \in \mathcal{U}$ at time slot t as $r_{u,f}^t$, $l_{u,f}^t$, and (x_u^t, y_u^t), respectively, where f indicates the type of resources like CPU cycles and disk space. UEs may request bandwidth resources that are slightly different from the general resources, as it not only associated with specific nodes but also the links between nodes in the overall streaming chains. Thus, we separately denote the requested bandwidth of UE u at time slot t as b_u^t. Let $\mathcal{N}_{all} = \mathcal{N} \cup \mathcal{N}_I$ denote the set of physical nodes in the system, where $\mathcal{N}$ indicates the nodes can be directly accessed by the core network and $\mathcal{N}_I$ indicates the other nodes that need to be accessed by the Internet, i.e., $\mathcal{N} \cap \mathcal{N}_I = \emptyset$. We use (x_n, y_n) to indicate the position of n, the total resources for resource f and bandwidth of node $n \in \mathcal{N}_{all}$ are denoted as $r_{n,f}$ and b_n, and the remaining resources and bandwidth are denoted as $r_{n,f}^{rem}$ and b_n^{rem}.

The nodes initialize VNFs with varying resources to build multiple slices, the set of slices in the system can be denoted

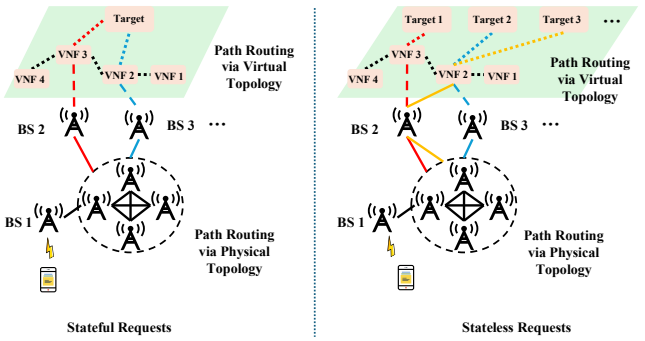


Fig. 2. The illustration of multi-route resource provisioning.

as $\mathcal{S}$, where the UEs of slice s are denoted as $\mathcal{U}_s$. We focus on anomaly prevention before anomaly occurrence, during which the VNFs don't change their host and allocated resources intermittently, thus, we denote the VNFs hosted by node n as $\mathcal{V}_n$. Besides, let S_u denote the slice that UE u belongs to, and $\mathcal{V}_s$ denote the set of VNFs in slice s. For the VNF $v \in \mathcal{V}_s$, we denote its physical host as $H(v)$, where $H(v) \in \mathcal{N}_{all}$. The allocated amount for resource f and bandwidth of v can be expressed as $r_{v,f}$ and b_v, respectively, and the remaining resources and bandwidth are denoted as $r_{v,f}^{rem}$ and b_v^{rem}.

Additionally, we denote the average latency for UEs to access the nearest AP as η^W, and the latency for BSs' optical connections as η^O, respectively. The slices can be configured with different latency performances of their virtual links to support different preferences of users, which can be realized by allocating better network channels to support latency-sensitive services based on the physical links [15]. Thus, for slice s, we denote the latency among directly connected VNFs as η_s^V. Besides, accessing the Internet generally has much higher latency than the local network, thus, we uniformly denote the latency for the UE accessing the VNFs via the Internet as η_s^I.

B. Request Provisioning Model

Anomalies in the system generally occur due to unfulfilled requests, one key issue is that all UEs' requests are implicitly treated as stateful. Another issue is that the possible multi-route resource provisioning for UEs' requests and the potential load balancing among routes, which can help to improve the utilization of bandwidth resources. We assume that each stateful request can only be provisioned by one target VNF for general resources in each time slot, and each physical node can only serve one of the VNFs in the same slice.

To achieve more flexible resource provisioning, we segregate the requests to be stateful and stateless according to their properties and consider the requests from UEs are provisioned via a multi-route strategy, as shown in Fig. 2. Specifically, for stateful requests that are assigned with a specific VNF, each request can be provisioned separately by lots of routes, but finally all routes need to fall into the assigned VNF as the target to ensure service continuity. For instance, as shown on the left side of Fig. 2, we illustrate two routes by red and blue lines, the requests of UE u should first connect to the nearest physical node as the AP (BS 1), and then the requests can be routed via physical links to the BSs that host the VNFs of the corresponding slice (BS 2 hosts VNF 3 and BS 3 hosts VNF 2). Finally, the requests can be routed to the target VNF by the virtual links that belong to the same slice with UE u, as shown by the blue and red dashed lines. Similarly, for stateless requests, the requests can be routed by multiple routes to VNFs that can be different from the previously assigned VNFs.

We denote the request type of UE u as $\delta_u \in \{0, 1\}$, where "1" means stateful requests and "0" means stateless. Let $\mathcal{R}_u^t$ denote the set of routes to serve the requests of u at time slot t, where $\mathcal{R} \in \mathcal{R}_u^t$ denotes one of the routes and can be expressed as $\mathcal{R} = (\mathcal{R}_P, \mathcal{R}_V)$. Here, $\mathcal{R}_P$ denotes the physical nodes in $\mathcal{R}$ and the others denote the VNFs in slice S_u. Moreover, we use $\mathcal{R}_{P,0}$ to denote the first item in $\mathcal{R}$, which should be the AP of UE u. Besides, $\mathcal{R}_{P,-1}$ and $\mathcal{R}_{V,0}$ denote the last physical node and the first VNF in $\mathcal{R}$, respectively, where $\mathcal{R}_{P,-1}$ hosts $\mathcal{R}_{V,0}$. Finally, $\mathcal{R}_{V,-1}$ denotes the target VNF of route $\mathcal{R}, \forall \mathcal{R} \in \mathcal{R}_u^t$. Each route serves part of the UE's request, where the resources are provided by the target node $\mathcal{R}_{V,0}$ of route $\mathcal{R}$, and the bandwidth is provided by all the nodes. For each route $\mathcal{R}$, we denote the resources that $\mathcal{R}$ can provide with u for f as $r_{u,f,\mathcal{R}}$, and the bandwidth as $b_{u,\mathcal{R}}$.

C. Problem Formulation

When UE u accesses the resources from the slices, each route's latency can be obtained as the sum of wireless communications to the AP, latency among physical nodes to the slice, and the latency among VNFs in the slice. The AP of UE u at time slot t should be the nearest physical node accessible from core networks, denoted as N_u^t, and can be obtained by

$$N_u^t = \arg\min_{n \in \mathcal{N}} \left(\left(x_n - x_u^t\right)^2 + \left(y_n - y_u^t\right)^2 \right), \forall u \in \mathcal{U}. \quad (1)$$

Besides, we denote the latency of $\mathcal{R}$ as $l_{\mathcal{R}}^t$, calculated by

$$\begin{aligned} l_{\mathcal{R}}^t = \eta^W + \eta^O(|\mathcal{R}_P| - 1) + \Pi_{(\mathcal{R}_{V,-1} \in \mathcal{N})}\ \eta_s^V(|\mathcal{R}_V| - 1) \\ + \Pi_{(\mathcal{R}_{V,-1} \in \mathcal{N}_I)} \eta_s^I, \forall \mathcal{R} \in \mathcal{R}_u^t, \forall u \in \mathcal{U}_s, \forall s \in \mathcal{S}, \end{aligned} \quad (2)$$

where $\Pi_{(\cdot)}$ is an indicator variable that equals to 1 when "$\cdot$" holds otherwise equals to 0.

The UEs' requests can be fulfilled while all the requested resources and bandwidth can be accessed with preferred latency, indicating that $\sum_{\mathcal{R} \in \mathcal{R}_u^t} r_{u,f,\mathcal{R}} \geq r_{u,f}^t$ and $\sum_{\mathcal{R} \in \mathcal{R}_u^t} b_{u,\mathcal{R}} \geq b_u^t$. Intuitively, when the latency of a route cannot satisfy the preferred latency from UEs, the resources and bandwidth cannot be provided, and we have $r_{u,f,\mathcal{R}} = 0, b_{u,\mathcal{R}} = 0$ when $l_{\mathcal{R}}^t > \min\{l_{u,f}^t\}_{f \in \mathcal{F}}$ holds. We denote the anomaly indicator of UE u at time slot t as $\Upsilon_u^t \in \{0, 1\}$, where "0" indicates that the request of u can be fulfilled and "1" indicates not, thus, Υ_u^t can be expressed as

$$\Upsilon_u^t = \begin{cases} 0, \sum\limits_{\mathcal{R} \in \mathcal{R}_u^t} r_{u,f,\mathcal{R}} \geq r_{u,f}^t, \forall f, \sum\limits_{\mathcal{R} \in \mathcal{R}_u^t} b_{u,\mathcal{R}} \geq b_u^t; \\ 1, \text{ otherwise}; \end{cases} \quad (3)$$

To serve the requests of UEs, the cost from MNOs can be related to the provisioned resources/bandwidth and latency that reflects the quality of routes, where the routes with lower

latency will introduce higher cost. For each node, let α_f and β denote the price of a unit of resource f and bandwidth, respectively, the cost for serving u can be calculated by

$$C_u^t = (1 - \Upsilon_u^t) \sum_{\mathcal{R} \in \mathcal{R}_u^t} ((\sum_{f \in \mathcal{F}} \alpha_f r_{u,f,\mathcal{R}} + \beta |\mathcal{R}| b_{u,\mathcal{R}})/l_{\mathcal{R}}^t). \quad (4)$$

We aim to reduce the number of anomaly UEs and the average cost for provisioning normal UEs, to this end, the optimization problem can be formulated as

$$\min_{\{\mathcal{R}_u^t, r_{u,f,\mathcal{R}}, b_{u,\mathcal{R}}\}_{\forall \mathcal{R} \in \mathcal{R}_u^t}} \sum_{u \in \mathcal{U}} \Upsilon_u^t + \frac{\sum_{u \in \mathcal{U}} C_u^t}{|\mathcal{U}| - \sum_{u \in \mathcal{U}} \Upsilon_u^t}, \quad (5a)$$

$$s.t. \sum_{u \in \mathcal{U}} \sum_{\mathcal{R} \in \mathcal{R}_u^t} \Pi_{(v \in \mathcal{R})} b_{u,\mathcal{R}} \leq b_v, \forall v \in \mathcal{V}_n, \forall n, \quad (5b)$$

$$\sum_{u \in \mathcal{U}} \sum_{\mathcal{R} \in \mathcal{R}_u^t} \Pi_{(v = \mathcal{R}_{V,-1})} r_{u,f,\mathcal{R}} \leq r_{v,f}, \forall v \in \mathcal{V}_n, \forall n, \forall f, \quad (5c)$$

$$\sum_{u \in \mathcal{U}} \sum_{\mathcal{R} \in \mathcal{R}_u^t} (\Pi_{(n \in \mathcal{R})} + \sum_{v \in \mathcal{V}_n} \Pi_{(v \in \mathcal{R})}) b_{u,\mathcal{R}} \leq b_n, \forall n, \quad (5d)$$

$$\sum_{v \in \mathcal{V}_n} r_{v,f} \leq r_{n,f}, \sum_{v \in \mathcal{V}_n} b_v \leq b_n, \forall n, \forall f, \quad (5e)$$

$$\Pi_{(\cdot)} \in \{0,1\}, \Upsilon_u^t \in \{0,1\}, N_u^t \in \mathcal{N}, \forall u \in \mathcal{U}, \forall t. \quad (5f)$$

Here, (5b) and (5c) indicate the total provided bandwidth and resources from VNF v to all UEs should not exceed the allocated resources/bandwidth of v; (5d) ensures the total bandwidth provided by node n and its hosting VNFs for all UEs should not exceed the total bandwidth of node n; (5e) means the resources and bandwidth of all the VNFs of node n should not be more than the total resources of node n.

III. User Traffic Steering Framework

We aim to reduce the system's abnormal UEs and provide resources for normal UEs, thus, the problem tends to be deciding the nodes and routes for serving each UE $u \in \mathcal{U}$ according to its requests and mobility information, we elaborate this process in Algorithm 1. Specifically, we first initialize the anomaly flag of u as $\Upsilon_u^t = 1$, and set a resource pool $\mathcal{P}_u^r$ and bandwidth pool $\mathcal{P}_u^b$ to store the provided resources/bandwidth for u, which will be dynamically updated. Afterward, we aim to obtain the candidate nodes for u, denoted as $\mathcal{V}_u^{t*}$. Here, for the stateless requests that don't need to be associated with specific VNFs, each VNF can be a candidate to provide resources for u, thus, we have $\mathcal{V}_u^{t*} = \mathcal{V}_{S_u}$ when $\delta_u = 0$. On the other hand, for stateful requests, the candidate nodes should be those that have been serving the UEs to keep service continuity, we use $\mathcal{V}_u^{Serve}$ to denote the VNFs that have served UE u, and can derive $\mathcal{V}_u^{t*} = \mathcal{V}_u^{Serve}$ when $\delta_u = 1$ (Lines 1-2).

After $\mathcal{V}_u^{t*}$ is obtained, we try to find the possible routes to satisfy the requests of u, to this end, UE u first gets its AP N_u^t and generates routes from N_u^t to each VNF in $\mathcal{V}_u^{t*}$, the requests should first route to one of the physical nodes that hosts the VNFs of S_u, the possible nodes can be obtained as

$$\mathcal{N}_{u,p} = \{n | n \in \mathcal{N}_{all}, \mathcal{V}_n \cap \mathcal{V}_{S_u} \neq \emptyset\}. \quad (6)$$

To achieve high bandwidth utilization, we iterate the physical and virtual routes with maximum bandwidth by a modified Dijkstra algorithm, which performs a similar iteration process to the Dijkstra algorithm with a different goal as finding the route with maximum bandwidth [14]. Thus, we set the weights of edges of the physical topology among ESs (denoted as $\mathcal{G}$) and the virtual topology among VNFs in slice S_u (denoted as $\mathcal{G}_{S_u}$) as the minimum remaining bandwidth of each ES pair and each VNF pair (Lines 3-4). The physical route $\mathcal{R}_P$ is generated from N_u^t to $n \in \mathcal{N}_{u,p}$, and the virtual route $\mathcal{R}_V$ is generated from VNF $v_{start} = \mathcal{V}_n \cap \mathcal{V}_{S_u}$ to the target VNF $v \in \mathcal{V}_u^{t*}$. The overall route $\mathcal{R}$ then is obtained and will be saved only if the total latency $l_{\mathcal{R}}^t$ satisfied $l_{u,f}^t, \forall f \in \mathcal{F}$ (Lines 5-11).

After obtaining the route $\mathcal{R}$, we add the resources and bandwidth that v and $\mathcal{R}$ can provide to $\mathcal{P}_u^r$ and $\mathcal{P}_u^b$, where the provided resources $r_{v,f}^{prov}$ is set as the remaining resources of v, and the provided bandwidth $b_{\mathcal{R}}^{prov}$ is the minimum remaining bandwidth among all physical/virtual nodes in route $\mathcal{R}$. The remaining resources/bandwidth of nodes are then accordingly updated (Lines 12-18). Notice that the provided resources/bandwidth in $\mathcal{P}_u^r$ and $\mathcal{P}_u^b$ may exceed the requests of UEs, to save the resources and achieve load-balancing among nodes in the route, we calculate the scaled ratios $\alpha_f, \forall f$ and β based on the requests of UEs and the provided resources/bandwidth. Then, all the VNFs update their remaining and provided resources based on the scaled resource, and each route's nodes update their remaining and provided bandwidth, and the edge weights of $\mathcal{G}$ and $\mathcal{G}_{S_u}$ are updated (Lines 19-29).

At last, we set a limit of iterations to avoid excessive iteration in finding all the routes, as in our considered multi-route provisioning framework, u can have lots of routes from N_u^t to a target $v \in \mathcal{V}_u^{t*}$ to increase the provisioned bandwidth. In this paper, we aim to evaluate the system performance in reducing the anomaly UEs but not detecting the anomalies by searching all the routes. Thus, we denote k as the maximum number of iterations for generating routes from N_u^t to each VNF in $\mathcal{V}_u^{t*}$, and when the iteration i gets close to k, the program will exit all the loops. Finally, the requests of u are checked to be satisfied or not to obtain Υ_u^t, and routes and provided resources/bandwidth from each route can be obtained from $\mathcal{P}_u^r$ and $\mathcal{P}_u^b$ (Lines 30-35). As $|\mathcal{N}_{u,p}|$ is generally much smaller than $|\mathcal{N}_{all}|$, the computation complexity of the algorithm can be approximated as $\mathcal{O}\left(|\mathcal{E}| \log |\mathcal{N}_{all}| + k^2 |\mathcal{N}_{all}| + k|\mathcal{E}|\right)$, where $\mathcal{E}$ denotes the edges of $\mathcal{G}$.

IV. Simulation Results

A. Simulation Settings

To evaluate the proposed traffic steering and resource allocation framework, we consider a scenario consisting of 15 ESs geographically distributed in a 100×100 m^2 area, with 5 routers facilitating network connectivity, and 50 users dynamically requesting various services. Additionally, 30 ESs are positioned outside this area and can only be accessed by the Internet. The total resources of each ES/router are uniformly set as 4 CPUs, 8 GB RAM, and 128 GB storage, with 10 Gbps bandwidth.

Algorithm 1: Proposed Framework

Input: u, $l^t_{u,f}$, (x^t_u, y^t_u), $\mathcal{V}_{S_u}, \mathcal{V}^{Serve}_u, \mathcal{G}, \mathcal{G}_{S_u}$, k.

Initialize: $\Upsilon^t_u = 1$, resource pool $\mathcal{P}^r_u = \emptyset$, bandwidth pool $\mathcal{P}^b_u = \emptyset$, iteration index $i = 0$.
Set $\mathcal{V}^{t*}_u \leftarrow \mathcal{V}^{Serve}_u$ if $\delta_u = 1$ else $\mathcal{V}^{t*}_u \leftarrow \mathcal{V}_{S_u}$.
Determine AP N^t_u by (1) and obtain $\mathcal{N}_{u,p}$ by (6).
Set the weight of edge $(n, n') \in \mathcal{G}$ and $(v, v') \in \mathcal{G}_{S_u}$ as $\min\{b^{rem}_n, b^{rem}_{n'}\}$ and $\min\{b^{rem}_v, b^{rem}_{v'}\}$.
for Each physical node $n \in \mathcal{N}_{u,p}$ **do**
  Generate route $\mathcal{R}_P$ from N^t_u to n by maximum bandwidth Dijkstra algorithm [14].
  Obtain $v_{start} = \mathcal{V}_n \cap \mathcal{V}_{S_u}$.
  for Each VNF $v \in \mathcal{V}^{t*}_u$ **do**
    Generate route $\mathcal{R}_V$ from v_{start} to v [14], set $\mathcal{R} = (\mathcal{R}_P, \mathcal{R}_V)$, calculate $l^t_{\mathcal{R}}$ based on (2).
    if $l^t_{\mathcal{R}} > \min\{l^t_{u,f}\}_{f\in\mathcal{F}}$ **then**
      Continue.
    Set $b_{\mathcal{R}} = \min\{\{b^{rem}_n\}_{n\in\mathcal{R}_P}, \{b^{rem}_v\}_{v\in\mathcal{R}_V}\}$.
    Set $r^{prov}_{v,f} = r^{rem}_{v,f}, \forall f \in \mathcal{F}, b^{prov}_{\mathcal{R}} = b_{\mathcal{R}}$.
    $\mathcal{P}^r_u \leftarrow \mathcal{P}^r_u + (v, r^{prov}_{v,f})$, $\mathcal{P}^b_u \leftarrow \mathcal{P}^b_u + (\mathcal{R}, b^{prov}_{\mathcal{R}})$.
    Update $r^{rem}_{v,f} \leftarrow r^{rem}_{v,f} - r^{prov}_{v,f}$.
    for $n \in \mathcal{R}_P$ and $v \in \mathcal{R}_V$ **do**
      Update $b^{rem}_n \leftarrow b^{rem}_n - b^{prov}_{\mathcal{R}}$.
      Update $b^{rem}_v \leftarrow b^{rem}_v - b^{prov}_{\mathcal{R}}$.
    Set $\alpha_f = \min\{1, r^t_{u,f} / \sum_{(v,r^{prov}_{v,f})\in\mathcal{P}^r_u} r^{prov}_{v,f}\}$.
    Set $\beta = \min\{1, b^t_u / \sum_{(\mathcal{R},b^{prov}_{\mathcal{R}})\in\mathcal{P}^b_u} b^{prov}_{\mathcal{R}}\}$.
    for $(v, r^{prov}_{v,f}) \in \mathcal{P}^r_u$ **do**
      Update $r^{rem}_{v,f} \leftarrow r^{rem}_{v,f} - (\alpha_f - 1) r^{prov}_{v,f}$.
      Update $r^{prov}_{v,f} \leftarrow \alpha_f r^{prov}_{v,f}$.
    for $(\mathcal{R}, b^{prov}_{\mathcal{R}}) \in \mathcal{P}^b_u$ **do**
      for $n \in \mathcal{R}_P$ and $v \in \mathcal{R}_V$ **do**
        Update $b^{rem}_n \leftarrow b^{rem}_n - (\beta - 1) \times b^{prov}_{\mathcal{R}}$.
        Update $b^{rem}_v \leftarrow b^{rem}_v - (\beta - 1) \times b^{prov}_{\mathcal{R}}$.
      Update $b^{prov}_{\mathcal{R}} \leftarrow \beta b^{prov}_{\mathcal{R}}$.
    Update edge weights in $\mathcal{G}$ and $\mathcal{G}_{S_u}$.
    $i \leftarrow i + 1$.
    if $i \geq k$ **then**
      Exit all loops.
if $\{\sum_{(v,r^{prov}_{v,f})\in\mathcal{P}^r_u} r^{prov}_{v,f} \geq r^t_{u,f}\}_{f\in\mathcal{F}}$, $\sum_{(\mathcal{R},b^{prov}_{\mathcal{R}})\in\mathcal{P}^b_u} b^{prov}_{\mathcal{R}} \geq b^t_u$ **then**
  Set $\Upsilon^t_u = 0$.

Output: $\Upsilon^t_u, \mathcal{P}^r_u, \mathcal{P}^b_u$.

Additionally, to mimic realistic network conditions, we consider the request of UEs forms a normal distribution over time slots, where the mean of the requested resources and bandwidth are randomly set from 0.01 to $r_{n,f}/2$ and $b_n/10$, where the variances are set as the half of the mean value. Besides, we consider four different service slices, where the mean and variance of latency preferences of UEs are randomly set as $40 - 150$ ms and $20 - 75$ ms, respectively. The proportion of users with stateful requests is set as 0.5, and the maximum iteration number k for route selection is set as 4. For network connections, the average wireless latency η^W from users to APs is set as 2 ms, while the average latency between physical nodes η^O is 3 ms. The latency among VNFs η^V_s ranges among $1 - 5$ ms, and the latency for accessing the Internet η^I_s ranges from 100 ms to 200 ms, which depends on the characteristics of the slice configurations. We conduct experiments over 100 time slots to evaluate the performance consistency.

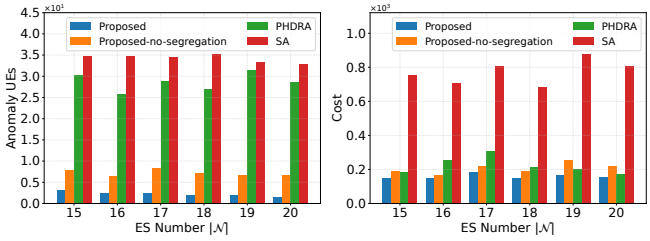
To validate the effectiveness of the proposed scheme, we consider the baselines and performance indicators as follows: *1)* **Proposed-no-segregation** method, in which all the requests are treated as stateful and can only be provisioned by the VNFs that have served the corresponding UEs in previous time slots; *2)* **Parallel Hierarchical DRL-based Resource Allocation (PHDRA)** method [12], in which the segregation of UEs' requests and the resource allocation in multiple routes provisioning are not considered; *3)* **Sequential Approaches (SA)** [8], in which the resource allocation didn't consider the load balancing of multiple routes. Moreover, to demonstrate the advantages of our proposed framework in preventing the anomaly UEs and reducing the cost of MNOs, the performance indicators are set as the number of abnormal UEs and the cost of MNOs for serving normal UEs, i.e., $\sum_{u\in\mathcal{U}} \Upsilon^t_u$ and $\sum_{u\in\mathcal{U}} (C^t_u / (|\mathcal{U}| - \sum_{u\in\mathcal{U}} \Upsilon^t_u))$.

B. Evaluation Results in Various ESs

Fig. 3 illustrates the performance comparison between our proposed framework and the baseline approaches with varying $|\mathcal{N}|$. In Fig. 3(a), we observe the number of abnormal UEs across different ES deployments. Our proposed approach consistently maintains the lowest number of anomalies across all ESs. Specifically, our solution achieves a significantly lower anomaly rate, with an average reduction of 2.33 times compared to the Proposed-no-segregation method, 12.53 times compared to the PHDRA approach, and 15.03 times compared to the SA approach. This substantial reduction in anomalies highlights the effectiveness of our multi-route provisioning strategy combined with node load balancing. Here, we can see that only the proposed scheme with request segregation can reduce the anomalies while $|\mathcal{N}|$ increases, as more ESs provide more candidate VNFs for stateless requests. Fig. 3(b) presents the cost comparison for serving normal UEs. Our approach achieves remarkable cost efficiency, showing an average reduction of 30.87% compared to the Proposed-no-segregation method, 40.39% compared to the PHDRA approach, and 3.86 times compared to the SA approach. This demonstrates that our framework not only prevents anomalies but also significantly reduces operational expenses for MNOs.

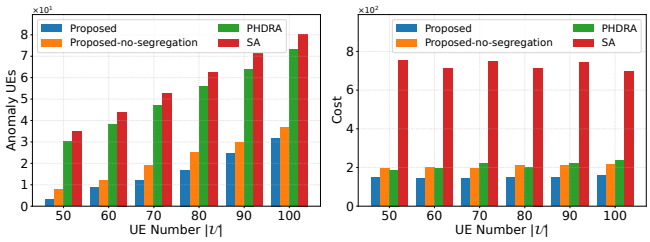
C. Evaluation Results in Various UEs

Fig. 4 demonstrates how our framework performs when $|\mathcal{U}|$ increases from 50 to 100. In Fig. 4(a), we observe that as the UE number increases, all approaches show an upward trend in



(a) Anomalies versus $|\mathcal{N}|$ (b) Cost versus $|\mathcal{N}|$

Fig. 3. The number of abnormal UEs and cost versus $|\mathcal{N}|$.



(a) Anomalies versus $|\mathcal{U}|$ (b) Cost versus $|\mathcal{U}|$

Fig. 4. The number of abnormal UEs and cost versus $|\mathcal{U}|$.

anomaly count. However, our proposed framework consistently maintains the lowest number of anomalies across all UE densities. Overall, our approach achieves an average reduction of 54.94% compared to the Proposed-no-segregation method, 3.35 times compared to the PHDRA method, and 3.92 times compared to the SA method. This demonstrates the scalability of our solution even under increasing network load. Fig. 4(b) illustrates the cost efficiency with varying UE numbers. The average cost of normal UEs does not increase or decrease with $|\mathcal{U}|$ (neither $|\mathcal{N}|$), as this indicator is only related the requests from UEs and the bandwidth/latency of selected routes. Our proposed approach maintains significantly lower costs across most UE configurations, achieves an average reduction of 37.97%, 41.35%, and 3.89 times compared to the Proposed-no-segregation, PHDRA, and the SA approach, respectively. These results collectively demonstrate that our proposed traffic steering framework with multi-route provisioning and load balancing effectively prevents anomalies while substantially reducing operational costs for MNOs across varying network configurations and user densities.

V. Conclusion

In this paper, we investigate anomaly prevention and resource allocation for UE request provisioning in 6G network slices. A novel request provisioning model and a traffic steering framework are proposed to address the key challenges posed by limited edge device resources in supporting dynamic UE requests. By incorporating load balancing across different routes to enhance the overall service bandwidth capacity of each node, the proposed scheme achieves significant performance improvements in terms of anomaly prevention and resource utilization efficiency. Extensive simulation results validate the effectiveness of our framework in mitigating the potential anomalies and reducing the cost of MNOs.

Acknowledgments

This work is partially conducted at ICTFICIAL Oy, Finland. It is also partially supported by the Business Finland 6Bridge 6Core project (Grant No. 8410/31/2022), the European Union's Horizon Europe programme for Research and Innovation through the 6G-SANDBOX project (Grant No. 101096328), and the 6G-Path project (Grant No. 101139172). The paper reflects only the authors' views. The European Commission and the Spanish Ministry are not responsible for any use that may be made of the information it contains.

References

[1] A. M. Alwakeel and A. K. Alnaim, "Network slicing in 6G: a strategic framework for IoT in smart cities," *Sensors*, vol. 24, no. 13, p. 4254, July 2024.

[2] C. Wang, R. Li, X. Wang, T. Taleb, S. Guo, Y. Sun, and V. C. M. Leung, "Heterogeneous edge caching based on actor-critic learning with attention mechanism aiding," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3409–3420, Nov. 2023.

[3] C. Sun, X. Wu, X. Li, Q. Fan, J. Wen, and V. C. M. Leung, "Cooperative computation offloading for multi-access edge computing in 6G mobile networks via soft actor critic," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 6, pp. 5601–5614, Nov. 2024.

[4] N. Ghafouri, J. S. Vardakas, A. Ksentini, and C. Verikoukis, "High-level service type analysis and MORL-based network slice configuration for cell-free-based 6G networks," *IEEE Trans. Veh. Technol.*, May 2025.

[5] Z. Ming, Q. Guo, H. Yu, and T. Taleb, "Deep reinforcement learning-based task offloading over in-network computing and multi-access edge computing," in *Proc. International Conference on Networking and Network Applications (NaNA)*, Aug. 2023, pp. 281–286.

[6] H. Yu, Z. Ming, C. Wang, and T. Taleb, "Network slice mobility for 6G networks by exploiting user and network prediction," in *Proc. IEEE International Conference on Communications (ICC)*, May 2023, pp. 4905–4911.

[7] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Netw.*, vol. 34, no. 1, pp. 84–93, Jan. 2020.

[8] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Uncertainty-aware resource provisioning for network slicing," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 1, pp. 79–93, Mar. 2021.

[9] Z. Ming, H. Yu, and T. Taleb, "Federated deep reinforcement learning for prediction-based network slice mobility in 6G mobile networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 11 937–11 953, Dec. 2024.

[10] M. Masoudi, Ö. T. Demir, J. Zander, and C. Cavdar, "Energy-optimal end-to-end network slicing in cloud-based architecture," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 574–592, Mar. 2022.

[11] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "A coverage-aware resource provisioning method for network slicing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, pp. 2393–2406, Dec. 2020.

[12] K. Qiao, H. Wang, W. Zhang, D. Yang, Y. Zhang, and N. Zhang, "Resource allocation for network slicing in Open RAN: A hierarchical learning approach," *IEEE Trans. Cogn. Commun. Netw.*, May 2025.

[13] Z. Ming, H. Yu, and T. Taleb, "User request provisioning oriented slice anomaly prediction and resource allocation in 6G networks," in *Proc. IEEE International Conference on Communications (ICC)*, June 2024, pp. 3640–3645.

[14] K. Wei, Y. Gao, W. Zhang, and S. Lin, "A modified Dijkstra's algorithm for solving the problem of finding the maximum load path," in *Proc. IEEE International Conference on Information and Computer Technologies (ICICT)*, Mar. 2019, pp. 10–13.

[15] Q. Guo, Z. Ming, H. Yu, Y. Chen, and T. Taleb, "Profit-aware proactive slicing resource provisioning with traffic uncertainty in multi-tenant flexe-over-wdm networks," in *Proc. IEEE International Conference on Communications (ICC)*, June 2024, pp. 3059–3064.