Mobility-Aware Satellite Backhaul Orchestration for 5G-Integrated Non-Terrestrial Networks

Ruoting Xiong, Kit-Lun Tong, Yi Ren, and Wei Ren

***Abstract*—Satellite-based non-terrestrial networks integrate satellites as backhauls to extend the coverage of terrestrial 5G networks. With multiple types of satellites available as backhaul links, we consider the satellite selection problem for traffic orchestration, aiming to satisfy diverse quality-of-service (QoS) requirements while minimizing monetary costs. Nevertheless, satellite mobility introduces new challenges to traffic orchestration in 5G satellite–terrestrial integrated networks. Existing solutions often fail to account for the time-varying latency, bandwidth constraints, and cost fluctuations of satellite backhaul links, which are inherently induced by satellite mobility. To address this gap, this paper proposes two satellite selection methods: a static approach based on predefined rules and a dynamic approach driven by reinforcement learning (RL). The static method first classifies incoming traffic into different application types and user behavior patterns using deep learning, then orchestrates the traffic to appropriate satellite links based on established rules. The dynamic method leverages multi-armed bandit (MAB) models within an RL framework to balance latency performance and monetary costs, thereby enabling adaptive satellite selection. Experimental results show that the static method satisfies 98% of the QoS requirements for various traffic types and reduces costs by a factor of 1.5–2. Moreover, the MAB-based models achieve a good balance between delay and price, yielding low average regret and reasonable traffic allocation even under price-surge scenarios.**

***Index Terms*—6G, satellite backhaul selection, QoS, encrypted data analysis, reinforcement learning**

I. Introduction

SATELLITE-based non-terrestrial networks (NTNs) [1], [2] are increasingly integrated into 5G and beyond mobile systems to provide backhaul connectivity for remote, rural, and disaster-affected areas. By leveraging low Earth orbit (LEO), medium Earth orbit (MEO), and geostationary Earth orbit (GEO) satellites, mobile network operators (MNOs) can significantly extend network coverage beyond the reach of terrestrial infrastructure. Fig. 1 illustrates a 5G system with satellite backhaul (5G-satellite) [3], which consists of a 5G core network (CN), base stations (BSs), and user equipment (UE). BSs are equipped with antennas that transmit information to satellites via different signal frequencies. Multiple satellites provide backhaul links between BSs in remote areas and the land-based CN, with ground stations (GSs) receiving satellite signals. Different types of traffic from UEs are transmitted through BSs and satellite backhauls, and finally arrive at the CN.

Ruoting Xiong, Kit-Lun Tong, and Yi Ren are with the School of Computing Sciences, University of East Anglia (UEA), Norwich, NR4 7TJ, U.K. (email: {ruoting.xiong, k.tong, e.ren}@uea.ac.uk, Corresponding author: Yi Ren).

Wei Ren is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China. (email:weirencs@cug.edu.cn)

However, satellite mobility introduces new challenges to traffic orchestration in 5G-satellite networks. In practice, satellites are launched in "batches" or "generations". Each batch leverages the latest available commercial off-the-shelf (COTS) components (e.g., processors, memory) at the time of manufacturing, leading to incremental increases in backhaul capacity over a 2–3-year technology cycle (*temporal heterogeneity*). As shown in Fig. 1, the satellite capacity available to a fixed BS varies over time in both LEO and MEO systems due to their continuous orbital motion (*spatial dynamics*). MEO satellites exhibit relatively slow capacity fluctuations because of their longer visibility periods, whereas LEO satellites, moving at much higher velocities, introduce more frequent capacity variations. In contrast, GEO satellites remain effectively stationary relative to the Earth, resulting in an almost constant capacity profile over time.

While prior studies are designed for static environments in which link characteristics, such as delay and price, are assumed to remain constant over time, these assumptions do not hold in satellite-assisted networks. Other researchers have investigated mobility-aware routing and resource management in satellite networks, they mainly optimize network-layer paths or resources based on explicitly available network-state information and predefined traffic requirements. Although satellite mobility and potential handover events can be predicted from orbital information, internal network conditions cannot be fully known before a backhaul is selected. As a result, satellite mobility creates a non-stationary online decision problem that cannot be directly addressed by conventional static selection or one-shot optimization methods.

To address this problem, several issues must be considered. First, from the perspective of a 5G network operator, accurately identifying traffic types is essential, as it is a prerequisite for steering traffic toward the most appropriate satellite links. Second, given the geolocation of a BS, it is necessary to estimate real-time information about currently connectable satellites, including their orbital positions, available bandwidth, and usage costs. Such estimates are critical because inaccurate or delayed satellite information can render traffic orchestration strategies ineffective. Finally, under predictable satellite mobility, an optimal traffic steering scheme must simultaneously satisfy user quality-of-service (QoS) requirements and minimize the operator's total satellite usage cost.

In this paper, we design a revenue- and QoS-driven traffic orchestration scheme for 5G operators with diverse types of satellite backhauls. We first propose a static satellite selection method based on deep learning (DL), which identifies traffic types and then orchestrates traffic according to predefined QoS rules. Our previous work [9] deployed convolutional neural

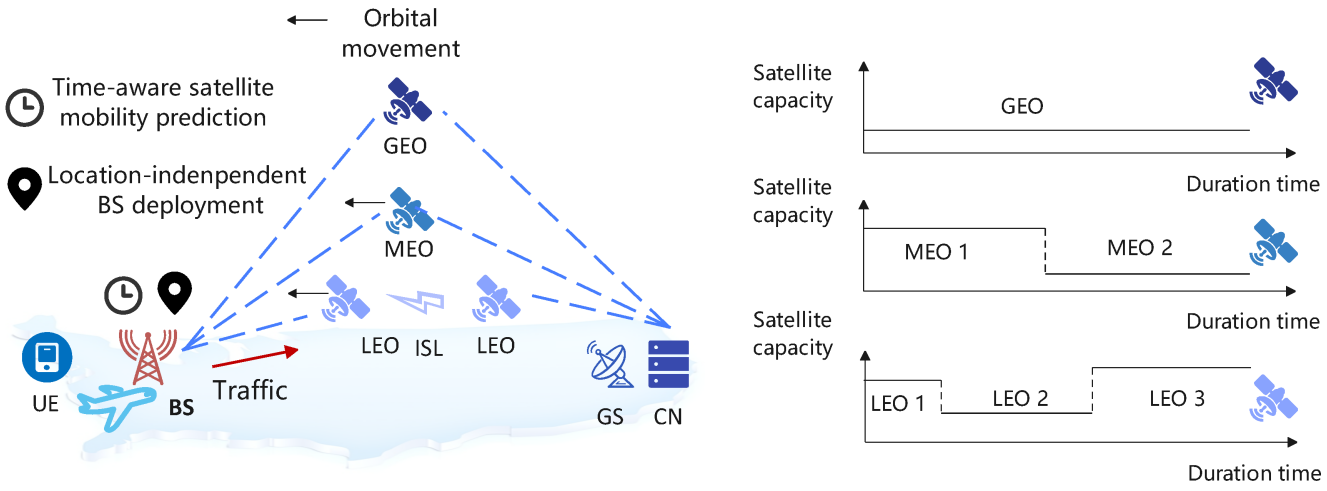


Fig. 1. 5G system with satellite backhauls: With the integration of satellites, different types of traffic from users are transmitted through BSs and satellite backhauls, and finally arrive at the land-based CN. The satellite capacity available to a fixed BS varies over time in both LEO and MEO systems due to their continuous orbital motion, while GEO satellites remain effectively stationary relative to the Earth, resulting in an almost constant capacity profile over time.

network (CNN) models for traffic classification, trained using data collected from our 5G private testbed. All algorithms and models are deployed on edge servers at the BS side. We use the Systems Tool Kit (STK) to simulate satellite mobility, enabling the prediction of real-time satellite locations. Given that the available satellite backhauls are known for a specific BS location, we can accurately estimate the connection duration between each BS–satellite pair, thereby inferring cost variations and delay fluctuations induced by satellite mobility. Furthermore, we propose a dynamic satellite selection method based on reinforcement learning (RL), which leverages a multi-armed bandit (MAB) model to achieve a trade-off between delay and price in dynamic environments. The contributions of this paper are as follows:

- *Mobility-Aware Satellite Backhaul Orchestration Formulation:* We identify and formulate satellite backhaul selection in 5G-integrated NTNs as a mobility-induced, time-varying cost–delay optimization problem. By explicitly considering satellite visibility dynamics and temporal heterogeneity, we show that satellite mobility fundamentally reshapes the tradeoff between QoS satisfaction and backhaul cost, which is not captured by existing static orchestration models.
- *Encrypted Traffic–Driven Static Satellite Selection Baseline:* We develop a practical static satellite selection baseline that integrates encrypted traffic classification with QoS-aware orchestration rules. CNN-based models are employed to identify application types and user behaviors from encrypted traffic, achieving accuracies of 95.3% and 88.7%, respectively, with an average inference latency below 18 ms. The proposed method satisfies up to 98% of QoS requirements while reducing backhaul cost by 1.5–2× compared with single-backhaul strategies.
- *Dynamic Satellite Selection Method:* We propose dynamic satellite selection methods based on MAB models to orchestrate traffic from a BS to the CN via satellite backhauls. Compared with static rules, these methods achieve higher rewards in satellite selection while maintaining a good balance between delay and cost.

The rest of the paper is organized as follows. In Section II, related work is reviewed. We present the background in Section III. Section IV describes the proposed scheme, followed by Section V performance evaluation. We conclude the paper in Section VI.

II. Related Work

The integration of satellite systems into 5G and beyond mobile networks has attracted significant research interest in recent years, covering system architecture [10], [11], protocol design [12], [13], and practical deployment. Early satellite–terrestrial measurements were conducted using GEO satellites and later extended to MEO and LEO satellites.

5G-satellite co-frequency interconnection: Numerous examples demonstrate collaboration between 5G mobile network operators and satellite backhaul service providers [14], [15]. For instance, China Telecom has verified the interoperability between MEO satellite networks and terrestrial 4G/5G networks at Antarctic research stations. Users can access China Telecom's 4G/5G BSs via mobile terminals, achieving network speeds exceeding 100 Mbps with latencies around 300 ms. Additionally, CubeSats and ground BSs have formed a collaborative network in the Qiangtang no-man's land, enabling ordinary mobile terminals to directly access the satellite-based 5G network with video-call latencies as low as 28 ms. An important future direction is 5G-satellite hybrid communications, which integrate LEO, MEO, and GEO satellites in a single scenario to provide multi-satellite backhaul. This hybrid architecture fully leverages the complementary advantages of satellite constellations in different orbits, overcoming the limitations of single-satellite systems to achieve more reliable, high-performance, and seamless connectivity.

From a traffic orchestration perspective, existing solutions can be broadly categorized into RAT selection [16]–[18], network slicing selection [19], [20], and network-layer routing/resource management in NTN environments.

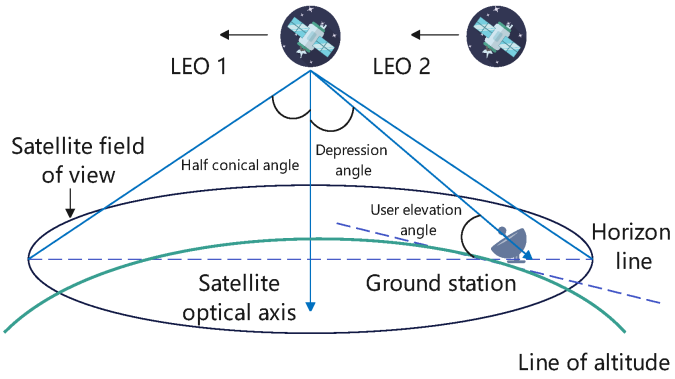


Fig. 2. Visibility relationship between satellite and ground station.

(1) RAT selection studies: Current RAT selection methods mainly focus on selecting among heterogeneous terrestrial or hybrid access networks to improve QoS, throughput, or load balancing. For example, ML-based RAT selection has been proposed for dynamic access selection in 4G/5G NSA networks [22], while DQN-based methods have been applied to heterogeneous 4G/5G/satellite access selection [24].

(2) Network slicing studies: Existing network slicing approaches mainly focus on intelligent slice selection and resource allocation in 5G core networks [25]–[28], aiming to improve resource utilization, provider profit, or service reliability. These methods typically assume explicit knowledge of service requirements and relatively stable network infrastructure, making them unsuitable for highly dynamic satellite backhaul orchestration.

(3) NTN routing and resource management studies: Recent NTN studies have investigated mobility-aware routing, delay-aware traffic engineering, and dynamic resource orchestration under time-varying satellite topologies [29]–[31]. However, these works mainly operate at the network layer, relying on explicit network-state information such as topology, link conditions, or traffic demand estimates for path optimization and resource scheduling.

In contrast, our work addresses a distinct application-layer satellite backhaul orchestration problem, where traffic characteristics must first be inferred from encrypted traffic observations. Meanwhile, due to orbital mobility, the candidate satellite set, link delay, transmission cost, and handover risk evolve continuously over time, resulting in a non-stationary decision environment.

III. Background

A. Satellite Visibility

The satellite-to-ground communication link is typically a line-of-sight (LoS) link. Due to the extremely long distance between satellites and ground stations, signals are affected by the atmosphere during propagation. However, as long as the LoS between satellites and ground users is not obstructed, a LoS communication link can be established. Considering that the mobility of ground users is relatively low compared to that of satellites, we assume that ground users are stationary relative to the Earth.

As shown in Fig. 2, the viewing field of a LEO communication satellite is a conical field of view. The straight line connecting the satellite to the bottom center of its conical viewing field is called the satellite optical axis. The half-cone angle (HCA) refers to the angle between the satellite's optical axis and the generatrix of the conical viewing field. The user elevation angle (UEVA) refers to the angle between the line of sight from the ground user to the LEO satellite and the horizon, and the satellite depression angle (SDA) refers to the angle between the line connecting the satellite's center of mass and a target point on the ground, and the tangential direction of the orbit where the satellite is located. The visibility between LEO satellites and ground users is determined by both the SDA and UEVA.

To minimize multipath fading and shadowing effects as much as possible, thereby improving communication quality and establishing a stable satellite-to-Earth communication link, A satellite handover is triggered when (1) the depression angle exceeds the half-cone angle (SDA $>$ HCA), or (2) the UEVA falls below the minimum threshold (UEVA $< 20°$).

B. Satellite-Ground Connectivity Time

Communication satellites operate in three primary orbits: LEO, MEO, and GEO, each with a distinct altitude, velocity, and orbital dynamics. GEO satellites maintain a geostationary position relative to Earth's surface due to their orbital period matching Earth's rotational cycle (approximately 24 hours), while LEO and MEO satellites undergo continuous relative motion with respect to any fixed terrestrial point. Usually, LEO satellites have orbital periods of only 90 minutes to 2 hours.

Satellite–ground connectivity time is typically shorter than the satellite coverage time because of constraints on both the UEVA and SDA, as well as the presence of blockages and obstacles. The satellite coverage time depends on the satellite's conical angle: the larger this angle, the longer the coverage time. It is also related to the satellite's velocity: higher satellite velocity results in shorter coverage time. Additionally, the satellite's velocity is determined by its orbital inclination, defined as the angle between the satellite's orbital plane and the Earth's equatorial plane. In general, a larger orbital inclination requires a higher velocity to maintain orbital stability.

Satellite mobility exhibits inherent predictability. As shown in Fig. 2, the currently available satellite is LEO 1, and the next satellite to connect will be LEO 2. After maintaining a connection with a ground station for a period, LEO 1 will move away and disconnect. The next LEO and MEO satellites connecting to the BS are both predictable. Therefore, critical satellite communication metrics, including latency and operational costs, can be precisely estimated. Leveraging this predictable orbital characteristic, our solution can acquire real-time satellite availability information, unaffected by BS mobility or changes in ground location.

C. Encrypted Data Analysis Methods

Encrypted traffic analysis is a prerequisite for directing traffic to the most appropriate satellite link. Recent studies also demonstrate real-time DL inference even on highly resource-constrained edge platforms, suggesting that CNN-based traffic

TABLE I
NOTATIONS

Notation	Meaning
$\mathbb{V}$	Node sets
$\mathbb{E}(t)$	Link sets
$\mathbb{B}$	Set of base stations
$\mathbb{S}$	Set of satellites
T	The observation time
B_b	b-th ground base station
l_b	Geographical position of the b-th BS
$e_{b,k}^{(t)}$	Communication link between B_b and S_k at time t
$t_k^{(b)}$	Connection duration of link $e_{b,k}^{(t)}$
I	Total number of 5G traffic flows forwarded by BS
$C_i(e_{b,k}^{(t)})$	Transmission cost of i-th traffic via link $e_{b,k}$ at time t
$D_i(e_{b,k}^{(t)})$	Traffic delay of i-th traffic via link $e_{b,k}$ at time t
$\hat{x}_n$	n-th one-channel figure from preprocessed traffic data
N	Total number of one-channel figures
f_Θ	DL-based classification function (parameterized by Θ)
$\hat{c}$	Total number of classification classes
p	Number of packets per data chunk (chunk size)

classification is practically feasible on comparatively resource-rich maritime and UAV edge nodes [32]. Existing studies mainly focus on classification tasks such as application, use-case, and device-type identification using DL techniques [38]. For example, [36] proposes a device classification framework capable of identifying multiple device types with high accuracy, while [37] employs LSTM and CNN models for encrypted 5G voice application identification.

The proposed framework can also support mobile-network and digital-twin applications. Learning-aided scheduling studies highlight the importance of jointly considering service demands, mobility, computing resources, and energy constraints [33], [34]. Our framework can provide traffic and QoS information for edge-resource scheduling, while the identified traffic demands and predicted satellite states can update digital twins for QoE-aware resource allocation and orchestration [35].

IV. SATELLITE BACKHAUL SELECTION FRAMEWORK

A. System Model

We consider a 5G-satellite integrated network comprising ground communications, GEO, MEO, and LEO satellites. Within each time slot, this network can be modeled as an undirected graph Φ, defined as:

$$\Phi = (\mathbb{V}, \mathbb{E}(t)) \tag{1}$$

where $\mathbb{V}$ denotes the set of nodes and $\mathbb{E}(t)$ represents the set of time-varying communication links between nodes. The node set $\mathbb{V}$ consists of both satellites and ground BSs, i.e., $\mathbb{V} = \mathbb{S} \cup \mathbb{B}$, where $\mathbb{S}$ denotes the set of all satellites, which is the union of the disjoint subsets $\mathbb{G}$, $\mathbb{M}$, and $\mathbb{L}$, representing the sets of GEO, MEO and LEO satellites, respectively. $\mathbb{B}$ is the set of ground BSs, and each BS $B_b \in \mathbb{B}$ is deployed at a specific terrestrial location l_b. For ease of reference, the key notations used in this paper are summarized in Table I.

Due to satellite mobility, only short-term connectivity can be maintained between satellites and BSs. The visibility of the k-th satellite from a BS at location l and time t is described by a binary function $P_k(t, l)$. Accordingly,

$$P_k(t, l_b) = \begin{cases} 1, & \mathrm{SDA}_k(t, l_b) \le \mathrm{HCA}_k, \mathrm{UEVA}_k(t, l_b) \ge 20^\circ, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathrm{SDA}_k(t, l_b)$ denotes the satellite depression angle, HCA_k is the half-cone angle of the satellite coverage region, and $\mathrm{UEVA}_k(t, l_b)$ denotes the user elevation angle, which should be greater than 20° (see Section III-A). Specifically, $P_k(t, l_b) = 1$ indicates that the k-th satellite is visible to BS B_b, while $P_k(t, l_b) = 0$ indicates that the satellite is not visible.

At the beginning of the protocol, each BS B_b is provided with the predicted visibility values $P_k(t, l_b)$ for all satellites over a given prediction horizon[1]. In our STK-based simulation, the access report is generated according to these geometric visibility constraints, including elevation-angle and sensor coverage conditions. Therefore, the binary visibility variable used in the optimization represents the outcome of this physical visibility model.

We denote $e_{b,k}^{(t)}$ as the identifier of the communication link between BS B_b and satellite S_k at time slot t, where the index b corresponds to the BS, and k corresponds to the satellite. The set of all links at time t is then given by $\mathbb{E}(t) = \left\{ e_{b,k}^{(t)} \;\middle|\; B_b \in \mathbb{B},\ S_k^{(b)} \in \mathcal{S}(t, l_b) \right\}$. During a given time interval T, each BS B_b may communicate with multiple transient satellites selected from its visible satellite set $\mathcal{S}(t, l_b)$. Each link $e_{b,k}^{(t)}$ is available only for a finite connection duration $t_k^{(b)}$.

The BS B_b forwards 5G traffic flows η_i ($i \in [1, \ldots, I]$) to the terrestrial CN via satellite backhauls. Each traffic flow η_i is characterized by $\eta_i = \{aT_i, z_i, \sigma_i, \zeta_i, Q_i\}$, where aT_i denotes the packet arrival time, z_i is the data size in bytes, σ_i is the application type, ζ_i represents the user behavior type, and Q_i is the QoS requirement. At the current time t, based on the predicted σ_i and ζ_i, the BS B_b selects the k-th satellite for traffic orchestration. Since satellite availability and link characteristics are predictable, the unit transmission cost and delay of the k-th satellite can be estimated in advance, which facilitates the accurate calculation of the transmission cost $C_i(e_{b,k}^{(t)})$ and delay $D_i(e_{b,k}^{(t)})$ of the i-th traffic over the link $e_{b,k}^{(t)}$ at time t.

B. Problem Formulation

From the perspective of 5G mobile network operators (MNOs), utilizing diverse satellite backhaul links can enhance network service reliability. However, balancing satellite usage costs with the need to guarantee user experience (i.e., meeting traffic QoS requirements) is a key challenge, especially considering the dynamic changes in satellite latency, bandwidth, and cost induced by satellite mobility.

The satellite selection strategy relies on accurate traffic classification. Therefore, we first optimize the deep learning

[1]Information regarding currently available satellites, including their positions, velocities, and elevation angles, can be obtained using standard satellite-tracking libraries such as SGP4-based toolkits [39].

classifier and then formulate the satellite selection problem as a constrained optimization problem. Finally, we design an optimal satellite selection strategy for traffic forwarding.

a) First Objective: Optimization of Traffic Classification Accuracy (Prerequisite for Satellite Selection): The primary goal of 5G MNOs is to accurately identify traffic attributes (application type and user behavior) from encrypted 5G traffic. To ensure high classification reliability, the objective is to minimize the prediction error of the DL-based traffic classifier, which is defined as follows:

$$\mathbf{P1}: \min_{\Theta} \frac{1}{N} \sum_{n=1} \xi(\Theta(\hat{x_n}), y_n), \mathbb{D} = \{\hat{x_n}, y_n\}_{n=1}^{N} \quad (3)$$

where $\Theta(\hat{x_n})$ is the conditional probability distribution over all labels predicted for $\hat{x_n}$, and $\mathbb{D}$ is the dataset consisting of N items $\hat{x_n}$ with labels y_n, where $N = \frac{I}{p}$ and p is the number of flows per data chunk (chunk size). The cross-entropy loss function ξ for $\hat{x_n}$ is depicted as:

$$\xi(\Theta(\hat{x_n}), y_n) = -\sum_{j=1}^{\hat{c}} y_j * log(\Theta_j(\hat{x_n})) \quad (4)$$

where $\Theta_j(\hat{x_n})$ is the softmax activation function on the final fully-connected layer, which computes the probability of assigning point $\hat{x_n}$ to class j, and y_n is the true label represented as a one-hot encoded vector, where the correct class is assigned a value of 1 and all other classes are assigned 0. y_j is the j-th element of the one-hot label vector y_n. Solving **P1** is a conventional supervised learning task, which can be performed offline by minimizing the cross-entropy loss using stochastic gradient descent and its variants.

b) Second Objective: Minimization of Total Satellite Backhaul Traffic Cost and Delay (Under QoS Constraints): Due to satellite mobility, the set of available satellite backhauls and their associated performance metrics varies over time. The objective is to design an online satellite-selection strategy that minimizes long-term backhaul cost while satisfying the heterogeneous QoS requirements of traffic flows.

At time t, each traffic flow η_i must be forwarded via one satellite selected from the set of visible satellites $\mathcal{S}(t, l_b)$, where l_b denotes the BS location. We define a binary decision variable

$$\beta_{i,k}^{(t)} = \begin{cases} 1, & \eta_i \text{ is forwarded via satellite } k \text{ at time } t, \\ 0, & \forall i, \forall k \notin \mathcal{S}(t, l_b), \forall t \end{cases} \quad (5)$$

For a given satellite $k \in \mathcal{S}(t, l_b)$ at time t, the cost and delay incurred by forwarding traffic η_i via satellite k are given by

$$C_i(e_{b,k}^{(t)}) = \bar{C}_k(t)\, z_i, \quad (6)$$

$$D_i(e_{b,k}^{(t)}) = \bar{D}_k^{(prop)}(t) + \bar{D}_k^{(proc)}(t) + \bar{D}_k^{(queue)}(t) + \bar{D}_k^{(ho)}(t), \quad (7)$$

where z_i denotes the data size of the traffic flow. $\bar{C}_k(t)$, $\bar{D}_k^{(prop)}(t)$, $\bar{D}_k^{(proc)}(t)$, $\bar{D}_k^{(queue)}(t)$ and $\bar{D}_k^{(ho)}(t)$ denote the unit transmission cost, propagation delay, processing delay, queuing delay and handover delay associated with satellite link k at time t, respectively. The handover delay term is incurred only when a satellite switch occurs, mainly for LEO and MEO satellites due to their orbital mobility and limited visibility.

Unlike terrestrial backhaul selection, the candidate satellite set is inherently time-varying due to orbital mobility. Therefore, the feasible action space dynamically changes with both time and BS location $\mathcal{S}(t, l_b)$. This mobility-induced visibility variation fundamentally differentiates NTN orchestration from conventional static backhaul selection problems.

The satellite backhaul selection problem can then be formulated as the following constrained optimization problem:

$$\begin{aligned} \mathbf{P2:} \quad \min_{\{\beta_{i,k}^t\}} \quad & \sum_{i=1}^{I} \sum_{k \in \mathcal{S}(t,l_b)} \beta_{i,k}^{(t)}\, C_i(e_{b,k}^{(t)}) \\ \text{s.t.} \quad & \sum_{k \in \mathcal{S}(t,l_b)} \beta_{i,k}^{t} = 1, \quad \forall i, \\ & D_i(e_{b,k}^{(t)}) \le Q_i, \quad \forall i \text{ such that } \beta_{i,k}^{t} = 1, \\ & \beta_{i,k}^{(t)} \in \{0,1\}, \quad \forall i, k, t. \end{aligned} \quad (8)$$

where the objective function minimizes the total cost incurred by assigning each traffic flow to a satellite backhaul. The first constraint ensures that each traffic flow is assigned to exactly one eligible satellite backhaul at each time slot. The second constraint requires that the performance metric $D_i(e_{b,k}^{(t)})$ of the selected backhaul satisfies the QoS threshold Q_i. The third constraint restricts the assignment variables to binary values for all traffic, backhauls, and time slots. Due to the time-varying and stochastic nature of satellite cost and delay induced by orbital mobility, solving **P2** optimally requires future knowledge of satellite availability and traffic arrivals. We focus on designing practical online satellite selection strategies, which do not guarantee the QoS constraint at every exploratory decision. Instead, they provide a soft online approximation of P2.

C. Proposed Framework

With multiple satellite backhauls in NTNs, practical schemes are needed to orchestrate different types of traffic across multiple satellites while meeting their QoS requirements. Fig. 3 outlines the system framework, which includes the following components: app and user-behavior identification, satellite mobility simulation, and satellite selection and traffic orchestration.

Specifically, *traffic* can be either synthetically generated or collected from real devices. After training, the model is frozen for inference. *Model prediction* identifies the application and user-behavior types of incoming traffic at the BS side before the traffic is forwarded to the CN via a satellite backhaul. The results are evaluated in terms of the F1-score, accuracy, recall, and precision.

Satellite selection aims to choose different satellites as backhauls for *traffic orchestration*. In addition, *satellite mobility simulation* provides information about currently available backhauls, which is then used to derive satellite prices and delays. There are several policies for backhaul selection: i)

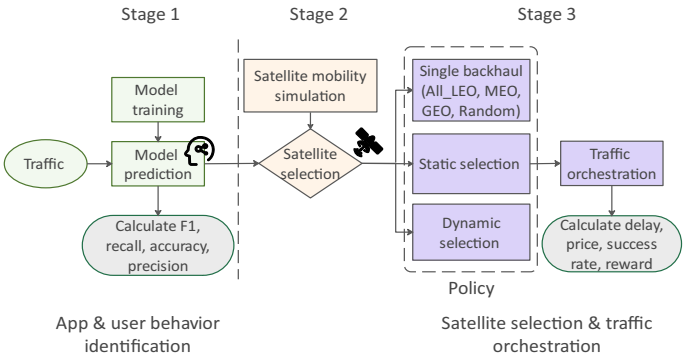


Fig. 3. System framework with different satellite selection policies.

Single-backhaul policy: assigns a single satellite backhaul to all UE traffic at a given time (All_LEO, All_MEO, or All_GEO). We also use random selection (Random) as a baseline and compare the delay, price, and success rate of these methods. ii) *Static-selection policy*: establishes fixed rules (static tables) for different traffic types based on their QoS requirements. iii) *Dynamic-selection policy*: leverages RL methods to perform real-time traffic orchestration. The RL agent selects a satellite for each traffic flow based on the estimated delay and price. We compare the rewards of the static and three dynamic methods.

The key advantage of the proposed satellite selection framework lies in its independence from BS deployment location, ensuring versatility and effectiveness across diverse scenarios. By exploiting the inherent predictability of satellite orbital dynamics, the framework can predict real-time satellite trajectories relative to any BS. This capability enables accurate estimation of critical backhaul metrics, including satellite–BS connection duration, end-to-end latency variations, and time-varying unit transmission costs. Consequently, the location-independent design allows 5G mobile network operators to seamlessly deploy the framework across diverse geographical environments while maximizing cost efficiency and satisfying traffic QoS requirements.

We present the algorithm in Alg. 1, which inputs a ground station or user terminal B_b, its current geographic location l_b, three satellite attribute sets, current time instant t, a pre-trained CNN model f_Θ deployed at the edge server dedicated to B_b, and a traffic dataset $\mathbb{D}$ comprising multiple traffic flows to be orchestrated.

In the initialization phase, satellite sets $\mathbb{G}$, $\mathbb{M}$, and $\mathbb{L}$ are retrieved from a public satellite library to access up-to-date orbital parameters (e.g., inclination, altitude, velocity) and connectivity capabilities, while the current ground location l_b of B_b is acquired and the pre-trained CNN model f_Θ is loaded into the edge server to enable real-time traffic flow identification. Our motivation for encrypted traffic classification is threefold. First, application semantics may not always be directly visible due to encrypted transport mechanisms such as VPNs. Second, standardized QoS identifiers provide only coarse service-level differentiation and may not capture finer-grained user behavior patterns that influence satellite selection decisions. Third, in some deployment scenarios, operator QoS metadata may not be directly accessible to edge-side or third-party traffic orchestration entities.

Algorithm 1: System input and output

Input: B_b, l_b, $\mathbb{G}$, $\mathbb{M}$, $\mathbb{L}$, t, f_Θ, $\mathbb{D}$;
Output: Traffic orchestration log file;
Load satellite information from the public satellite library to acquire $\mathbb{G}$, $\mathbb{M}$, and $\mathbb{L}$;
Obtain the current ground location l_b;
Load the pre-trained CNN model f_Θ at edge server for B_b;
Infer the available GEO satellite for B_b: $S_0^{(b)}$ with infinite connection time;
Infer the set of available MEO satellites for B_b: $\{S_1^{(b)}, \ldots, S_{s'}^{(b)}\} \subseteq \mathbb{M}$;
Infer the set of available LEO satellites for B_b: $\{S_{s'+1}^{(b)}, \ldots, S_s^{(b)}\} \subseteq \mathbb{L}$;
Identify the i-th traffic η_i using CNN models f_Θ;
For static method: load the QoS table for traffic orchestration;
For dynamic method: load the RL agent for traffic orchestration;
Select the k-th satellite and establish the corresponding link $e_{b,k}^{(t)}$ at time t;
Execute traffic orchestration;
Calculate the cost $C_i(e_{b,k}^{(t)})$ and delay $D_i(e_{b,k}^{(t)})$ of the selected satellite link;
Output log file: (t, η_i, $e_{b,k}^{(t)}$, $C_i(e_{b,k}^{(t)})$, $D_i(e_{b,k}^{(t)})$);

The static orchestration method loads a pre-configured QoS table, whereas the dynamic orchestration method initializes an RL agent that takes real-time environmental feedback to learn optimal policies. Following strategy selection, the k-th satellite is selected from the inferred available sets based on the orchestration policy (from either the QoS table or RL agent), the corresponding satellite link $e_{b,k}^{(t)}$ is established at time t, and traffic orchestration is executed by routing η_i through $e_{b,k}^{(t)}$. Two critical performance metrics are computed post-execution: link cost $C_i(e_{b,k}^{(t)})$ and traffic delay $D_i(e_{b,k}^{(t)})$ (including propagation, processing, and queuing delay). Finally, a log file is generated and output, recording the time instant t, the identified traffic flow η_i, and the established link $e_{b,k}^{(t)}$ to support later analysis.

Next, we present the following parts in detail: Apps and user behavior identification, satellite mobility simulation, static and dynamic selection methods.

D. Stage 1: Apps and User Behavior Identification

App identification aims to identify application names, such as Amazon and Google. Similarly, user behavior identification aims to recognize whether users are sending text, image, or audio/video data. Fig. 4 shows the process of app and user-behavior identification, including data collection and processing, model training, and model prediction. The initial setup of the 5G testbed is not depicted in the figure. The following sections introduce these steps in detail.

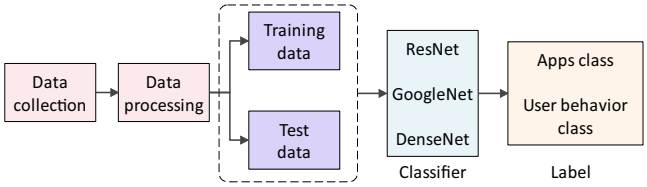


Fig. 4. Apps and user behavior identification process.

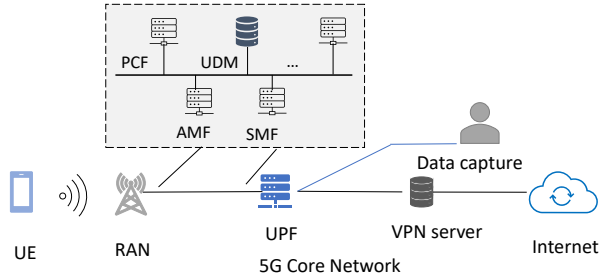


Fig. 5. 5G network architecture with VPN server.

1) Step1: 5G Testbed Setup: As shown in Fig. 5, the 5G network architecture mainly consists of three parts: UEs, the radio access network (RAN), and the CN. UEs can be mobile phones, laptops, or other IoT devices. RAN technologies include 4G/5G BSs, LoRa, and Wi-Fi. The 5G CN contains several network functions (NFs), such as the access and mobility management function (AMF), the session management function (SMF), and the user plane function (UPF).

The 5G CN decouples the control plane (CP) from the user plane (UP) to provide low-latency network services. Specifically, the UP connects to the Internet and handles the actual data traffic, whereas NFs such as the AMF and SMF in the CP exchange signalling data for access control and session management. When a UE requests access to the CN or updates a session, it communicates with the CP, which processes signalling and control data. After UEs successfully attach to the network, they communicate with the UP in the CN, namely, the UPF. Both the UPF and the BS use GTP to encapsulate the data.

2) Step2: Data Collection and Processing: We captured data at the UPF (on the Ethernet interface from the CN to the BS) using Wireshark, as shown in Fig. 5. We also set up a VPN on the UEs. For app traffic, we collected data from 10 widely used applications and several other applications classified as unknown. For each application, we collected 30,000 packet records. For user-behavior traffic, we defined seven categories: Audio, Bigfile, Image, Location, Text, Others, and Video. "Others" refers to user behaviors such as voting and sharing contact cards or links. We collected 30,000 packet records for each user behavior within each application and then mixed data corresponding to the same user behavior, regardless of application type. All data were encapsulated using GTP and contained encrypted payloads, representing traffic transmitted through satellite backhauls.

We preprocessed the .pcap files into images [2]. We divided the dataset into smaller chunks containing p packets using Splitcap. After processing, all data were converted into single-channel images, each with a sample size of 784 bytes. A larger value of p allows each sample to contain more information.

3) Step 3: Traffic Classification: Let $\hat{x}_n$ ($n \in \{1, 2, ..., N\}$) denote the n-th image, where N is the total number of images. The traffic classification process maps each image $\hat{x}_n$ to a corresponding class label, which is formalized as

$$f_\Theta : \hat{x}_n \rightarrow \{1, 2, ..., \hat{c}\}, \tag{9}$$

where $\hat{c}$ is the total number of classes (11 classes for applications, and 7 classes for user behaviors), and f_Θ is the classification function parameterized by Θ. After classification, the predicted traffic packet $\eta_i' = \{aT_i, z_i, \sigma_i', \zeta_i', Q_i'\}$ is used for subsequent satellite selection.

We choose three CNN models, f_Θ: GoogLeNet, ResNet, and DenseNet, to classify 5G GTP traffic into 11 application types and 7 user behavior types. This design prioritizes computational efficiency, robustness, and transferability. First, CNNs are well suited to the short, fixed-size representation of p packets, where long-term sequential dependencies are less important. Second, convolutional operations can be parallelized, making CNNs suitable for low-latency edge inference. Third, the image-based representation enables transfer learning from mature pretrained CNN models, reducing the data and training effort required for new traffic classes and deployment environments.

E. Stage 2: Satellite Mobility Simulation

We use STK to simulate satellite mobility. STK is a satellite simulator developed by Analytical Graphics, Inc. Given the orbital altitude and inclination, STK can calculate the position of a satellite and its coverage area at any time. Orbital altitude and inclination are important satellite parameters that influence orbital speed. In general, a higher orbital altitude corresponds to a lower orbital speed.

We use commercial satellite orbit data as inputs. Specifically, the LEO constellation data are provided by Starlink, and the MEO data are obtained from GLONASS [3]. We set the ground station at a fixed location in Wuhan, China. We assume clear-sky LoS conditions, with satellite coverage determined by the satellite's sensor angle and orbital altitude. A single LEO satellite typically covers a ground area with a diameter ranging from 100 to 1000 km, corresponding to a coverage area of several hundred to several thousand km^2. In contrast, MEO satellites have a much larger coverage range. A single MEO satellite can cover an area ranging from several hundred thousand to more than one million km^2, and some sources indicate that its coverage may reach a diameter of 7000–25,000 km. The output is a list of ground-to-satellite connection durations.

Given a fixed ground station location, our objective is to identify the currently available satellite backhauls and obtain their corresponding delay and pricing information. The steps are as follows:

1) Download the LEO and MEO satellite orbital data in the two-line element (TLE) format. We use 1,365 LEO satellites and 24 MEO satellites in the experiment.

[2]https://github.com/yungshenglu/USTC-TK2016.

[3]https://celestrak.org/

TABLE II
DIFFERENT SATELLITE SYSTEMS

Satellites	Orbit height	Communication delay	Period
LEO	500-2000 km	20-50 ms	1.5 h
MEO	2000-35786 km	100-150 ms	12 h
GEO	35786 km	500-700 ms	Stationary

2) Configure the satellite network coverage range by adjusting the sensor angles: set the LEO sensor angle to 45° and the MEO sensor angle to 10°.
3) Set the location of the ground station as Wuhan, China. This location is flexible.
4) Insert the chain module, link the constellation and the fixed location.
5) Generate the access report. The report has the format [satellite name, connection start time, connection end time, and duration]. This format provides the start time, end time, and duration of each satellite–ground connection.

Finally, we obtain a list of satellite–ground connection durations in the format [LEO_1: $time_1$, LEO_2: $time_2$, ...], with an analogous format for MEO satellites. This means that LEO_1 maintains a connection with the ground station for $time_1$. After $time_1$, LEO_1 moves away, and LEO_2 approaches the ground station to provide communication services for $time_2$.

In this work, LEO, MEO, and GEO satellite links are treated as alternative backhaul options for traffic orchestration. Physical-layer effects, such as interference, are not explicitly modeled, but are assumed to be reflected in the overall service performance metrics, including delay, availability, and cost.

F. Stage 3: Static Satellite Selection based on Deep Learning

The static satellite selection method begins by analyzing network traffic and categorizing it according to application types and user behaviors. The method then selects the appropriate satellite backhaul based on predefined QoS rules. We first present the communication delays of different satellites and then introduce our customized satellite-selection rules.

1) Satellite Communication Delay: The satellite system uses LEO, MEO, and GEO satellites for communication [41]. Because their orbital altitudes differ, their communication delays and mobility characteristics also differ. Table II lists the altitude, communication delay, and orbital period of the different satellite systems.

As shown, LEO satellites have the shortest communication delay, ranging from 20 to 50 ms. In terms of mobility, LEO satellites take approximately 1.5 hours to orbit the Earth, and they maintain a connection with the ground station for only a few minutes at a time. MEO satellites, on the other hand, offer a communication delay of 100 to 150 ms and can maintain a connection with the ground station for about 30 minutes at a time. GEO satellites, being positioned at the highest altitude, provide the broadest network coverage. In theory, just three GEO satellites can cover the entire Earth. However, they have the highest communication delay, ranging from 500 to 700 ms.

2) Satellite Handover Delay: The communication delay consists of propagation, processing, and queuing delays, which are continuously incurred during normal data transmission and jointly determine the baseline end-to-end latency. In contrast, handover delay is event-driven and only introduced when the serving satellite changes, rather than for every transmitted packet. Therefore, its impact depends on both the frequency of satellite switching and the orchestration policy.

Due to their high orbital velocity and short visibility windows, LEO satellites experience more frequent handovers, making handover latency a non-negligible factor. MEO satellites provide longer connection durations and thus lower switching overhead. GEO satellites are stationary relative to the ground and typically maintain persistent connectivity, so handover delay can be neglected. Based on representative NTN handover latency measurements reported in [40], the handover delay is modeled as 50–120 ms for LEO satellites, 10–30 ms for MEO satellites, and negligible for GEO satellites.

3) Static Rules for Traffic Orchestration: There are four main QoS classes for different types of traffic: "Conversational," "Interactive," "Streaming," and "Background." The "Conversational" class should provide network services with a delay of less than 1 s. The "Interactive" class has a delay of approximately 1 s and includes services such as e-commerce and web browsing. The "Streaming" class has a delay requirement of less than 10 s and includes streaming audio and video, FTP, and still images. The "Background" class has a delay requirement of more than 10 s and includes services such as fax. Specifically, interactive gaming requires a delay of less than 75 ms, conversational audio and video require delays of less than 150 ms (with a maximum of 400 ms), and Telnet requires a delay of less than 250 ms.

According to the 3GPP specification on 5G services and service capabilities, an application can specify its QoS requirements. This allows QoS requirements to be customized for different services. Network operators can then manage traffic across different satellites to satisfy the specific QoS needs of each application and user behavior. Suppose there are I traffic flows and K satellites. The i-th traffic is assigned to the k-th satellite as follows:

$$S_k = \mathcal{R}(\sigma_i, \zeta_i) \tag{10}$$

where $\mathcal{R}$ is the predefined QoS rule, and σ_i and ζ_i denote the application type and user-behavior type, respectively, of the i-th traffic flow. The detailed rules $\mathcal{R}$ are presented in Table III.

As shown, all BBC and Google traffic should be forwarded to GEO satellites, all Amazon, TikTok, and YouTube traffic should be forwarded to MEO satellites, and all unknown traffic should be forwarded to LEO satellites. For chat applications, different user behaviors are mapped to satellites based on their specific QoS requirements (e.g., audio and video are mapped to LEO for low latency). For example, text traffic in WhatsApp is forwarded to MEO satellites, image traffic is forwarded to GEO satellites, and audio and video traffic are mapped to LEO satellites.

G. Stage 3: Dynamic Satellite Selection based on Reinforcement Learning

Reinforcement learning is a method for making real-time optimal decisions. In this section, we introduce how to use

TABLE III
THE STATIC RULES OF SATELLITE SELECTION

App types	User behavior types	Satellites
BBC, Google	None	GEO
Amazon, TikTok, YouTube	None	MEO
Unknown	None	LEO
WhatsApp	Text	MEO
	Image	GEO
	Audio	LEO
	Video	LEO
Facebook	Text	GEO
	Image	MEO
	Audio	MEO
	Video	LEO
IG	Text	LEO
	Image	MEO
	Audio	LEO
	Video	LEO
Snapchat	Text	GEO
	Image	LEO
	Audio	MEO
	Video	MEO
Microteams	Text	GEO
	Image	GEO
	Audio	MEO
	Video	MEO

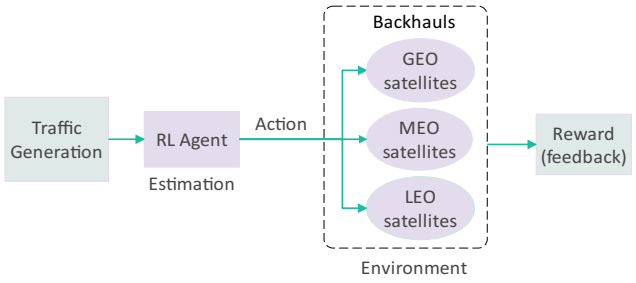


Fig. 6. Multi-armed bandit model for satellite selection (K=3).

MAB models for dynamic satellite selection. The main design issue is to define a reward function that enables the RL agent to make effective satellite-backhaul decisions.

1) Multi-armed bandit model: The multi-armed bandit model is a stateless online decision-making model. In RL, the main components are the agent and the environment. At each step, the agent selects an action based on the current estimate and receives a reward from the environment. Fig. 6 shows the 3-armed bandit model for satellite selection.

As shown in Fig. 6, the agent deployed at the BS selects satellites (actions) for incoming traffic based on the current estimates. The environment consists of three satellite tiers: LEO, GEO, and MEO. After the agent makes a decision, the reward is calculated based on the QoS performance and price of the selected satellite.

2) Traffic Characteristic: We consider a total of I network packets. Each network packet is characterized as $\eta_i = \{aT_i, z_i, \sigma_i, \zeta_i, Q_i\}$. We take model-prediction errors into account. When traffic from user requests passes through the CNN models, it is converted into predicted network packets, each characterized as $\eta_i' = \{aT_i, z_i, \sigma_i', \zeta_i', Q_i'\}$, where σ_i' is the predicted application type of the i-th network packet (output by the CNN classifier), ζ_i' denotes the predicted user-behavior type, and Q_i' is the inferred QoS requirement derived from the predicted σ_i' and ζ_i', reflecting the service-quality demand of the i-th traffic flow.

3) Action Space: For a BS B_b located at geographical position l_b, its satellite selection action space $\bar{A}^{(b)}$ is defined as:

$$\bar{A}^{(b)} = [S_0^{(b)}, S_1^{(b)}, S_2^{(b)}] \tag{11}$$

where $\bar{A}^{(b)}$ consists of three categories of satellites available for communication. Specifically, $S_2^{(b)}$ denotes the LEO satellite accessible to B_b at time t, $S_1^{(b)}$ represents the MEO satellite, and $S_0^{(b)}$ denotes the GEO satellite. Unlike LEO and MEO satellites, GEO satellites are geostationary and thus maintain a stable connection with B_b over time.

4) State Space: We cannot model CPU utilization and queue occupancy for individual satellites because they may be affected by other tasks, such as navigation and telemetry. Therefore, we use the MAB model, which eliminates the need for explicit state-space modeling and is well suited to this scenario. Specifically, LEO and MEO satellites are not geostationary: a LEO satellite can maintain a connection with a terrestrial BS for approximately 10 minutes, whereas an MEO satellite can maintain a connection for approximately 30 minutes. Therefore, we simulate changes in the unit cost and delay of LEO and MEO satellites in the RL experiment.

Assuming that the k-th satellite has been selected for traffic forwarding, the cost of the i-th traffic $C_i(e_{b,k}^{(t)})$ using link $e_{b,k}^{(t)}$ is defined as follows:

$$C_i(e_{b,k}^{(t)}) = \bar{C}_k(t) * z_i \tag{12}$$

where $\bar{C}_k(t)$ is the unit cost of the k-th satellite at time t, and z_i is the data size, in bytes, of the current traffic.

5) Reward Function: We aim to minimize the total cost of satellite backhauls while maintaining QoS requirements. Our model considers both the price and delay of user requests when calculating the overall reward. The agent receives lower rewards when selecting satellites with higher costs and longer delays. Accordingly, the reward r_i for the i-th traffic flow is formulated as a negative value, as follows:

$$r_i = -\tau * C_i(e_{b,k}^{(t)}) - (1-\tau) * D_i(e_{b,k}^{(t)}) \tag{13}$$

where τ is the weight assigned to $C_i(e_{b,k}^{(t)})$, and $D_i(e_{b,k}^{(t)})$ is the delay experienced by traffic η_i when using the k-th satellite at time t. When a handover is triggered based on the satellite access report, the total delay is modeled as the sum of the communication and handover delays, otherwise, it consists only of the communication delay. The communication delay includes propagation, processing, and queueing delays and is simulated using a uniform distribution over the representative delay range of each satellite tier (e.g., 20–50 ms for LEO satellites). The handover delay ranges from 50 to 120 ms for LEO satellites and from 10 to 30 ms for MEO satellites, whereas the handover delay for GEO satellites is considered negligible. Since delay and cost have different units, we normalize them using min–max scaling before incorporating them into the reward calculation.

For all traffic flows, the agent aims to maximize the cumulative reward. The optimization objective Ω_I is defined as follows:

$$argmax \quad \Omega_I = \sum_{i=1}^{I} r_i \tag{14}$$

6) Estimation Update: We adopt three online learning algorithms for estimation update, namely classic MAB, sliding-window upper confidence bound (SW-UCB), and discounted upper confidence bound (D-UCB).

For classic MAB, after selecting the k-th satellite and obtaining the corresponding reward, the estimation $\bar{E}_k$ is updated as:

$$\bar{E}_k \leftarrow \bar{E}_k + \frac{1}{\bar{H}_k + 1}(r_i - \bar{E}_k) \tag{15}$$

where $\bar{H}_k$ denotes the total number of times the k-th satellite has been selected, and the reward r_i is calculated according to (13).

For SW-UCB, only the most recent observations within a sliding window of size W are retained, enabling faster adaptation to changing satellite conditions. The reward estimation is updated as:

$$\bar{E}_k \leftarrow \bar{E}_k + \frac{1}{\bar{H}_k^{(W)} + 1}(r_i - \bar{E}_k) + 2\sqrt{\frac{\ln t}{\bar{H}_k^{(W)} + 1}} \tag{16}$$

where $\bar{H}_k^{(W)}$ denotes the number of times the k-th satellite has been selected within the sliding window, and the exploration coefficient is set to 2. t denotes the current decision epoch, and $\ln t$ provides a logarithmically increasing exploration term that balances exploration and exploitation.

For D-UCB, historical rewards are exponentially discounted such that recent observations contribute more strongly than stale ones. The estimation is updated as:

$$\bar{E}_k \leftarrow \gamma \bar{E}_k + (1-\gamma) r_i + 2\sqrt{\frac{\ln t}{\bar{H}_k^{(\gamma)} + 1}} \tag{17}$$

where $\gamma \in (0, 1)$ is the discount factor, and $\bar{H}_k^{(\gamma)}$ denotes the discounted effective selection count. A smaller discount factor (e.g., $\gamma = 0.99$) causes historical observations to lose influence more quickly, enabling faster adaptation to environmental changes, whereas a larger discount factor (e.g., $\gamma = 0.999$) preserves historical information for longer, resulting in more conservative adaptation to non-stationary dynamics.

The agent repeatedly chooses from a set of actions based on the current estimate. The action selection strategy follows the epsilon-greedy algorithm. With probability ϵ, the agent chooses a random action; with probability $1 - \epsilon$, it chooses the action with the highest empirical mean estimate.

V. Evaluation

A. Experiment Setup

We use free5GC as the 5G CN on a computer equipped with an Intel Core i5-10500 CPU and running Ubuntu 18.04, and connect it to the Gemtek 4G base station via an Ethernet cable. The UEs are commercial mobile phones from Samsung, Huawei, and Apple. We design the experiments as follows:

Exp1: The main purpose of Exp1 is to train the models for app and user-behavior identification. The training data account for 80% of the dataset, and the test data account for the remaining 20%. We deploy three classifiers, namely ResNet50, GoogLeNet, and DenseNet121. The experiments are based on the open-source PyTorch library. The learning rate is 0.0001, the optimizer is SGD, and the number of training epochs is 10. We evaluate the accuracy, precision, recall, and F1-score of the models, as well as the time overhead of model prediction.

Exp2: The main purpose of Exp2 is to simulate satellite mobility and obtain the ground-to-satellite connection durations. We use data for constellations of 1,365 LEO satellites and 24 MEO satellites, and fix the ground-station location in Wuhan. During the simulation, the prices and latencies of the currently available satellites fluctuate over time because of satellite mobility.

Exp3: We simulate traffic forwarding based on the static QoS rules detailed in Table III. The delays and prices of the currently available satellites fluctuate over time. We set the prices to 7.5–10 £/MB for LEO constellations, 4.5–6.5 £/MB for MEO satellites, and 2–3.5 £/MB for GEO satellites. We compare the proposed method with single-backhaul methods (All_LEO, All_MEO, and All_GEO) and a random-selection method (Random) in terms of the QoS success rate, price, and average delay.

Exp4: This experiment uses the CNN models from Exp1 and the same settings as Exp3. We perform dynamic selection among three satellite tiers, and the reward is defined in (13). The multi-armed bandit model employs the ϵ-greedy exploration strategy, with $\epsilon = 0.01$. For the SW-UCB method, we set the sliding-window size W to 1000, and for the D-UCB method, we set the discount factor γ to 0.99. Given the generated traffic, we compare the reward changes in Exp3 and Exp4.

B. Performance of Apps and User Behavior Identification

The evaluation metrics used in this paper include accuracy, precision, recall, and F1-score. The metrics are defined as follows: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, and $F1\text{-score} = \frac{2TP}{2TP+FP+FN}$, where TP denotes a true positive, i.e., a sample of class X correctly identified as X; FP denotes a false positive, i.e., a sample not belonging to class X incorrectly identified as X; TN denotes a true negative, i.e., a sample not belonging to class X correctly identified as not X; and FN denotes a false negative, i.e., a sample of class X incorrectly identified as not X.

1) Apps Identification: Fig. 7 shows the model performance for 11 application classes: Amazon, BBC, Facebook, Google, IG, Microteams, Snapchat, TikTok, WhatsApp, YouTube, and one unknown class. Fig. 7a shows the models' accuracy for app identification, while Figs. 7b–7d show their precision, F1-score, and recall, respectively.

As shown in Fig. 7a, the classification accuracies of the different models are all above 84%. The average accuracies of GoogleNet and ResNet50 are 94.2% and 94.8%, respectively. DenseNet achieves the best average accuracy of 96.9%. GoogleNet has relatively low accuracies for Google

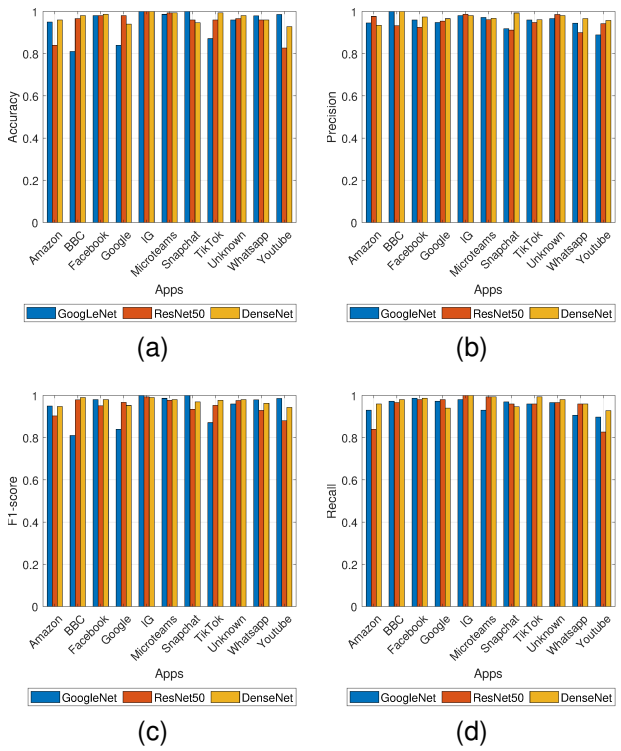


Fig. 7. Apps model performance in terms of accuracy, precision, F1-score and recall. (a) Accuracy. (b) Precision. (c) F1-score. (d) Recall.

and TikTok, at 84% and 87.1%, respectively. In addition, ResNet50 has difficulty identifying Amazon and YouTube, with accuracies of 84% and 82.7%, respectively. The recall results show similar trends to the accuracy results (Fig. 7d).

In Fig. 7b, the classification precision results for the different models are all above 89%. GoogleNet has relatively low precision for YouTube, at 89%. DenseNet achieves the best average precision of 97.1%, while GoogleNet and ResNet50 achieve 95.3% and 94.8%, respectively. Fig. 7c shows that the classification F1-scores of the different models are all above 81%. The average F1-scores of GoogleNet and ResNet50 are 94.2% and 94.9%, respectively. GoogleNet has relatively low F1-scores for Google and BBC, at 81% and 84%, respectively. DenseNet achieves the best average F1-score of 97%. In addition, the F1-scores are above 94% for most applications.

Summary: DenseNet outperforms the other models across the different metrics, whereas GoogleNet performs less well on this task. Therefore, DenseNet is selected for app identification.

In addition, we evaluate the model-prediction performance. We show the confusion matrix of the GoogleNet model for app identification in Fig. 8.

As shown in Fig. 8, the test accuracies of the classification models are all above 90%. For applications such as BBC, Snapchat, and TikTok, the test accuracy is approximately 98%. For Google, 4% of the traffic is misclassified as Facebook and 3% is misclassified as TikTok. Overall, the models achieve high test accuracy, indicating that they are well trained for app identification.

Finally, we show the F1-score results of the ResNet50 model for different values of p. Here, p represents the number of

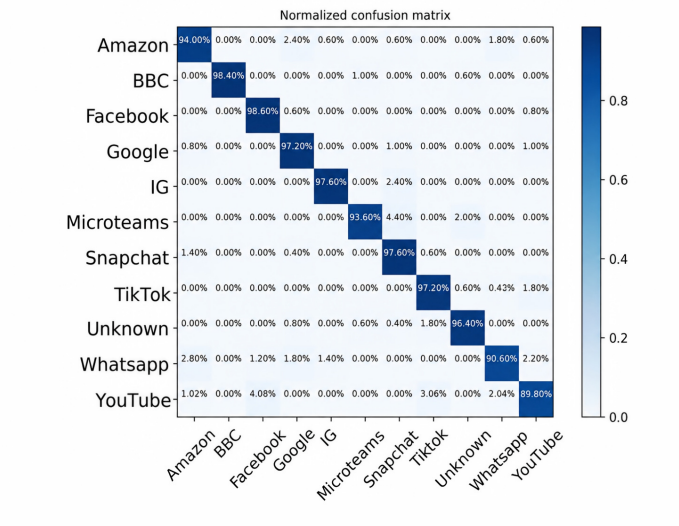


Fig. 8. The confusion matrix of Apps identification using GoogleNet model.

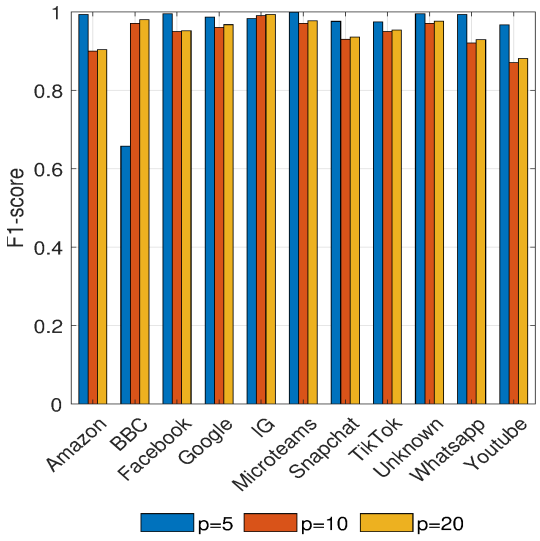


Fig. 9. The F1-score of the ResNet50 model for app identification across different p values.

network packets in each data chunk. We compare the results for $p = 5$, 10, and 20. The F1-score results for the different p values are presented in Fig. 9.

As shown in Fig. 9, the F1-score results are nearly the same when p is 10 or 20. Setting $p = 5$ negatively affects the F1-score because each sample contains less information. In addition, as the number of test samples increases, the model-training process becomes longer. Therefore, we select $p = 20$ for data processing because it accelerates model training without undermining performance. Overall, the models achieve high accuracy, and the prediction time is within the millisecond range, from 8 to 18 ms.

Summary: For app identification, even the worst-performing model can accurately classify the majority of network traffic. Additionally, during data preprocessing, setting p to 20 for a single data sample results in a model prediction time of only 18 ms.

2) User Behavior Identification: Fig. 10 shows the model performance for seven types of user behavior: Audio, Bigfile,

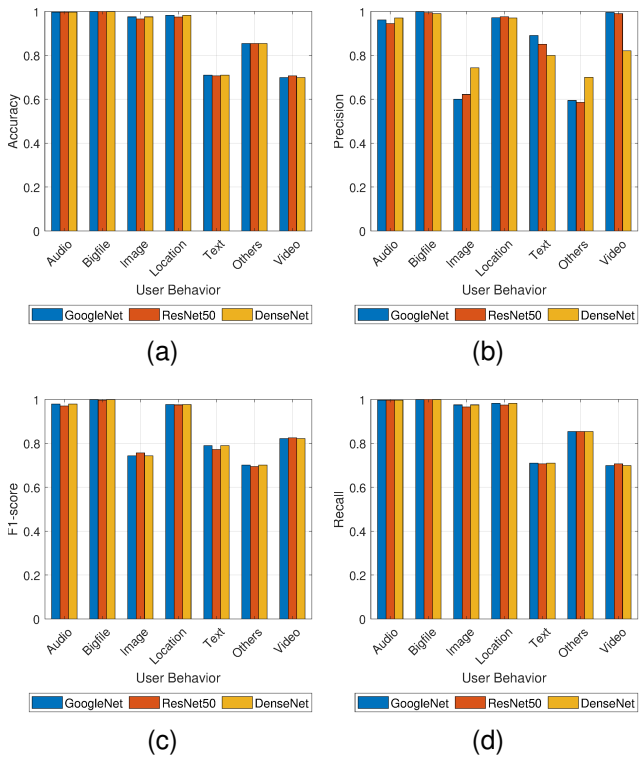


Fig. 10. User behavior model performance in terms of accuracy, precision, F1-score and recall. (a) Accuracy. (b) Precision. (c) F1-score. (d) Recall.

Image, Location, Text, Others, and Video. Fig. 10a shows the models' accuracy for user-behavior identification, while Figs. 10b–10d show the precision, F1-score, and recall, respectively.

As demonstrated in Fig. 10a, the classification accuracies for Audio, Bigfile, Image, and Location are high, all exceeding 97%. However, the models have difficulty identifying Text and Video, with the accuracy for Video classification reaching only 70%. Overall, GoogleNet achieves the best average accuracy, at approximately 88.9%. The recall results show similar trends to the accuracy results (Fig. 10d).

In Fig. 10b, the models achieve high precision in identifying Audio, Bigfile, Location, Text, and Video. However, they have difficulty identifying "Others." Specifically, GoogleNet and ResNet50 achieve only 60% precision, while DenseNet achieves 70%. In addition, the models have low precision for "Image" identification: GoogleNet and ResNet50 achieve 62%, while DenseNet achieves 74%.

In Fig. 10c, the models achieve high F1-scores for Audio, Bigfile, and Location identification, all exceeding 97.7%. However, they have relatively low F1-scores for "Image" and "Others," at 74% and 70%, respectively. Overall, all models achieve F1-scores above 85%.

Summary: For user-behavior identification, GoogleNet achieves the best average accuracy (88.9%), whereas DenseNet achieves the best precision (74%). All models achieve F1-scores above 85%.

To further examine the test accuracy of the models across different user behaviors, we show the confusion matrix of the GoogleNet model for user-behavior identification in Fig. 11.

In Fig. 11, the GoogleNet model achieves test accuracies above 98% for Audio, Bigfile, Image, and Location. For

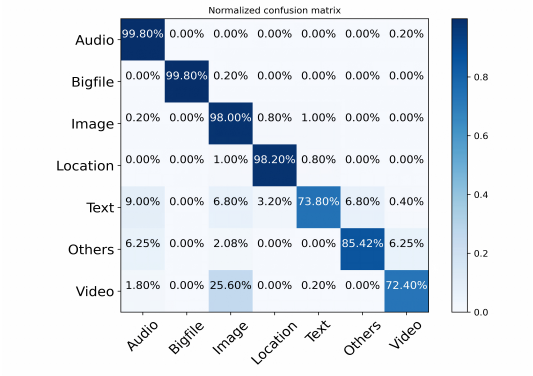


Fig. 11. The confusion matrix for user-behavior identification using the GoogleNet model.

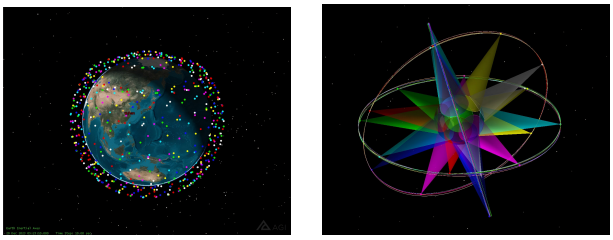


Fig. 12. The three-dimensional orbits of the LEO and MEO constellations. (a) LEO orbit data provided by Starlink. (b) MEO orbit data provided by GLONASS.

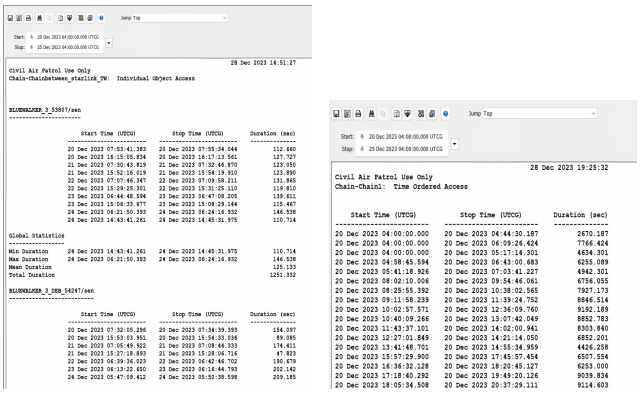
"Others," 85% of the traffic is classified correctly, while 6.25% is classified as Audio and another 6.25% as Video. In addition, the model has lower test accuracy for "Text": 9% of the traffic is classified as Audio, 6.8% as Others, and 6.8% as Image. We also observe that 25.6% of Video traffic is misclassified as Image because video consists of many image frames.

Summary: All models have difficulty distinguishing between "Image" and "Video." They also perform poorly in "Text" classification. However, recognizing "Video" as "Image" is unlikely to decrease the QoS success rate because these two categories are often orchestrated at the same service level.

C. Performance of Satellite Mobility

We use STK to predict satellite mobility. The experimental steps are described in Section IV-E. Here, we first show the orbit data of the LEO and MEO constellations on a three-dimensional panoramic view of the Earth, and then show the summary report of the connection durations. Fig. 12a shows the three-dimensional orbit data of the LEO constellation, and Fig. 12b shows that of the MEO constellation.

As shown, because the coverage area of LEO satellites is smaller than that of MEO satellites, a larger number of LEO satellites are typically deployed in a constellation to provide continuous service. Specifically, when LEO satellites orbit the Earth, they do not return to exactly the same ground location because of the Earth's rotation. Hence, even if a LEO satellite orbits the Earth 10 times per day, its number of connection periods with a given ground station is lower than 10. Therefore, to ensure 24-hour coverage of fixed ground stations, our experiment simulates a constellation consisting of



(a) (b)

Fig. 13. Access summary reports for LEO and MEO satellites, showing the connection durations between the ground station and the satellites. (a) LEO satellite access report. (b) MEO satellite access report.

1,365 LEO satellites (with the flexibility to scale to more) and 24 MEO satellites.

We also show the access summary reports for the LEO and MEO satellites obtained from STK in Fig. 13.

As shown in Fig. 13a, the connection times of LEO satellites with the ground station mostly range from 80 to 200 s, reflecting their high orbital speed. In addition, because satellite orbital data are publicly available, we can identify the names and orbital parameters of the satellites. For example, Fig. 13a shows the satellite names "BLUEWALKER_3_53807" and "BLUEWALKER_3_DEB_54247." Both satellites cover Wuhan several times within 24 h because their orbital speeds are much higher than the Earth's rotational speed. This speed difference causes the satellite ground tracks to pass periodically over the same area, such as Wuhan, resulting in multiple coverage periods within a single day. In contrast, Fig. 13b shows that MEO satellites maintain longer connections with the ground station, with connection durations ranging from 3,000 to 9,000 s. From the access reports, we can obtain information about the currently available satellite backhauls.

D. Performance of Static Satellite Selection

We compare our scheme with other single backhaul schemes, namely All_LEO, All_MEO, All_GEO and Random. After training, the models are frozen and used for prediction. We deploy a real-time monitoring system and use mobile phones to generate traffic with varying application-use ratios and user behaviors. The incoming traffic is recorded in log files. In the simulation, we replay each log file as incoming traffic and forward the traffic based on the QoS table. We then evaluate the QoS success ratio, average delay, and cost for different log files, each containing a different ratio of user behaviors (text:image:audio/video). The QoS success ratio indicates the percentage of traffic that successfully meets the predefined QoS requirements. The results are shown in Fig. 14.

As shown in Fig. 14, our method meets the QoS requirements of 98% of user requests. For the traffic proportion (5:3:2), the compared schemes achieve QoS success rates of 76.88%, 9.2%, and 50.22%, respectively. As the proportion of image and video traffic increases, the All_MEO scheme achieves a QoS success rate of 90.9%, the Random scheme achieves 83.2%, and the All_GEO scheme achieves a success rate below 15%. These schemes therefore fail to satisfy most QoS requirements for mobile-network users.

The communication results are shown in the second subfigure of Fig. 14. The average delays of the All_LEO, All_MEO, All_GEO, and Random schemes are 32.7 ms, 127.9 ms, 581.5 ms, and 320 ms, respectively, under the different traffic proportions. When the "Text" traffic ratio is high, the average delay of our method is only 140 ms. Overall, the average delay of our method under the different user-behavior proportions is below 180 ms.

In the third subfigure, the price of the All_LEO scheme is always the highest, ranging from 6,080 to 6,293£, whereas the All_GEO scheme has the lowest price, ranging from 1,710 to 1,770£. The prices of our method, random selection, and the All_MEO scheme are nearly half those of All_LEO. We omit the All_GEO scheme from the cost comparison because it cannot guarantee the QoS requirements. As the proportion of image and video traffic increases, the total price also increases because more large-sized data is generated. Specifically, when the proportion of video traffic exceeds 50%—a situation that is likely to occur in practice—the advantage of our method becomes evident, as it achieves the lowest cost, at approximately 3,096£.

Summary: The All_MEO, All_GEO, and Random schemes cannot meet the QoS requirements of most user requests, whereas our scheme achieves a 98% QoS success rate. Compared with the All_LEO scheme, our method reduces the total price by a factor of 1.5–2 while keeping the communication delay below 180 ms, thereby providing satisfactory performance for applications requiring sub-200-ms latency.

E. Performance of Dynamic Satellite Selection

We account for mobility-induced variations in satellite unit cost and transmission delay, and employ different MAB models to orchestrate traffic across heterogeneous satellites, achieving an optimal trade-off between QoS and cost. We compare the cumulative regret of the proposed static method with those of the dynamic methods. The regret is calculated using the absolute value of the reward, which is defined in Equation (13). The default weight τ is 0.5. The total number of traffic flows is approximately 865K, with a traffic-distribution ratio of 5:3:2.

1) Regret comparison of different MAB models: We compare the average regret of four algorithms, calculated by dividing the cumulative regret by the total number of traffic flows. The average regrets are shown in Fig. 15.

As shown in Fig. 15, the performance remains largely consistent across different user behavior distributions. In all scenarios, the static baseline performs the worst, while SW-UCB consistently achieves the lowest average regret. Specifically, the static baseline has an average regret of approximately 0.38, showing that fixed decision policies are ineffective under changing network conditions. The classic MAB improves performance through random exploration, reducing regret to around 0.34, but remains limited by inefficient exploration

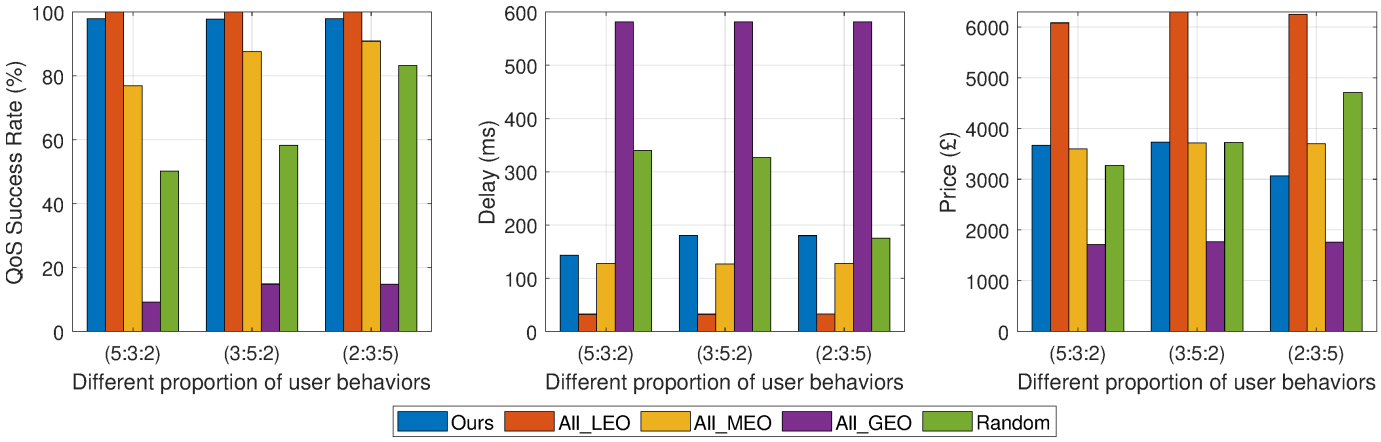


Fig. 14. Performance by varying the proportion of user behaviors: Success rate, delay and price.

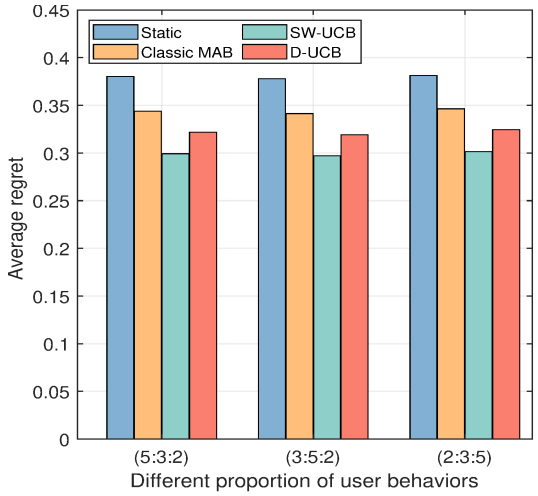


Fig. 15. Average regret comparison across different user behavior distributions.

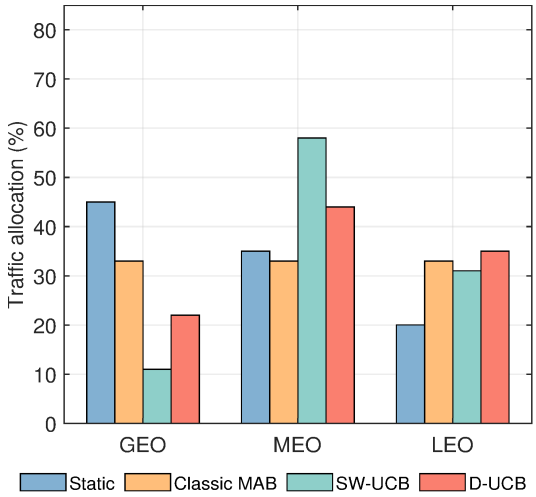


Fig. 16. Traffic allocation distribution across GEO, MEO, and LEO satellite tiers under different algorithms.

and outdated observations. D-UCB further reduces regret to approximately 0.32 by applying exponential discounting to historical rewards, enabling faster adaptation. SW-UCB achieves the best performance, with the lowest average regret of approximately 0.30. Compared with the Static baseline, SW-UCB reduces regret by about 21%, with additional gains of approximately 13% over classic MAB and 7% over D-UCB. This advantage comes from its sliding-window mechanism, which emphasizes recent observations and discards obsolete information, enabling more accurate reward estimation in time-varying environments.

2) Traffic allocation distribution under different MAB models: Fig. 16 illustrates the traffic allocation distribution across GEO, MEO, and LEO satellite tiers for different decision algorithms. The Static baseline allocates the largest proportion of traffic to GEO (approximately 45%), reflecting its conservative fixed policy and limited adaptability to changing network conditions. Classic MAB produces a more balanced allocation, distributing traffic relatively evenly between GEO, MEO, and LEO, but without strongly exploiting the most efficient satellite tier. In contrast, SW-UCB shows a clear preference for MEO, allocating nearly 58% of traffic to this tier while significantly reducing GEO usage. This suggests that SW-UCB effectively identifies the most cost-effective satellite option through rapid adaptation to recent reward observations. D-UCB also shifts traffic away from GEO and toward MEO/LEO, although its allocation remains more distributed than SW-UCB.

Summary: While the static method requires a well-designed QoS table, the RL-based dynamic method can achieve a trade-off between delay and cost during traffic orchestration. Among the dynamic methods, SW-UCB achieves the lowest regret by focusing on recent observations, whereas D-UCB demonstrates stronger robustness by discounting outdated rewards.

F. Robustness under Dynamic Pricing

To evaluate algorithm robustness under highly volatile pricing conditions, we introduce a dynamic price surge scenario in which the transmission cost of all satellite tiers (LEO, MEO, and GEO) changes periodically throughout the simulation. Specifically, every 100 decision steps, a price jump event is triggered, and the transmission cost of each satellite tier is independently rescaled by a random multiplier uniformly sampled from 1 to 10.

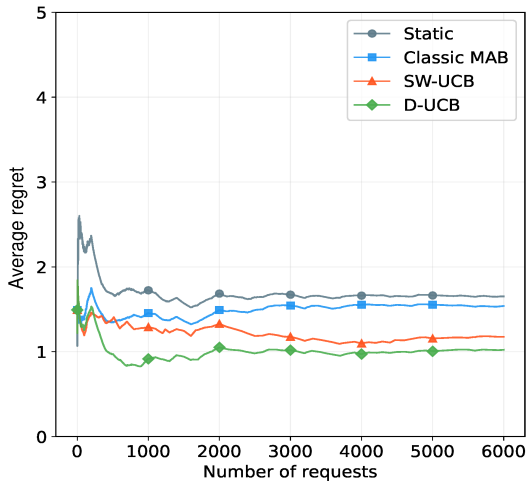


Fig. 17. Average regret under periodic price fluctuations.

1) Regret comparison of different MAB models: Fig. 17 shows the evolution of the average regret under these periodic price fluctuations. The Static baseline consistently exhibits the highest regret, confirming that fixed policies cannot adapt to rapidly changing cost conditions. Classic MAB improves performance through exploration, but its regret remains relatively high because of the persistent influence of outdated reward observations.

Among the adaptive methods, D-UCB achieves the best performance, consistently maintaining the lowest regret throughout the simulation, while SW-UCB performs slightly worse. This behavior can be explained by the relationship between the price-jump frequency and the SW-UCB window size. Since price fluctuations occur every 100 steps, while the SW-UCB sliding window spans $W = 1000$ observations, each window contains data from approximately ten different pricing regimes, making reward estimation unstable and less representative of the current environment. In contrast, D-UCB applies exponential discounting with $\gamma = 0.99$, which aggressively reduces the impact of historical observations while continuously adapting to recent changes. This enables faster tracking of rapidly varying pricing conditions, explaining its superior performance under highly volatile environments.

2) Impact of price surge intervals on average regret: Table IV reports the average regret per request under five price jump intervals ranging from 200 to 2000 decision steps, corresponding to approximately 20–200 seconds of simulation time under a traffic arrival rate of $\lambda = 10$ requests per second.

TABLE IV
AVERAGE REGRET UNDER DIFFERENT PRICE JUMP INTERVALS

Price surge (steps)	Static	Classic MAB	SW-UCB	D-UCB
200	1.6973	1.6124	1.2178	**1.0323**⋆
500	1.6961	1.6259	1.0556	**0.9091**⋆
1000	1.6798	1.6063	0.8979	**0.8864**⋆
1500	1.6933	1.5660	**1.0752**⋆	1.0777
2000	1.5928	1.5383	**0.9092**⋆	0.9653

The results reveal a clear shift in algorithm dominance depending on the pricing dynamics. When the price jump interval is shorter than or comparable to the SW-UCB sliding window size ($W = 1000$), i.e., 200, 500, and 1000 steps, D-UCB consistently achieves the lowest regret. This is because SW-UCB relies on a fixed observation window, and under frequent price fluctuations, the window still contains a substantial amount of outdated reward information from multiple historical pricing regimes, leading to less accurate reward estimation. In contrast, D-UCB continuously suppresses the influence of older observations and enables faster adaptation to rapidly changing environments. However, when the jump interval exceeds the SW-UCB window size (1500 and 2000 steps), SW-UCB slightly outperforms D-UCB. These results highlight the sensitivity of SW-UCB to window-size configuration, whereas D-UCB provides stronger robustness under highly frequent non-stationary pricing changes.

Summary: When price changes occur rapidly, D-UCB outperforms SW-UCB due to faster adaptation to abrupt reward shifts. However, when price variations become less frequent and exceed the SW-UCB observation window, SW-UCB regains its advantage through more accurate reward estimation within the current pricing regime.

VI. CONCLUSION

This paper proposes a mobility-aware satellite backhaul orchestration framework for 5G-integrated NTNs that combines encrypted traffic identification, satellite mobility prediction, and adaptive traffic steering across heterogeneous satellite tiers. A static QoS-rule-based method is introduced as a simple baseline for satellite selection. To address the non-stationary nature of NTN environments, we further investigate dynamic traffic orchestration using several MAB-based online learning algorithms, including conventional MAB, SW-UCB, and D-UCB. Overall, the proposed framework enables adaptive and cost-effective backhaul selection while maintaining application-specific QoS under mobility-induced network dynamics. Future work will incorporate real-world satellite network measurements, realistic traffic and pricing traces, weather- and blockage-aware link reliability models, and large-scale testbed validation.

ACKNOWLEDGMENT

The work of Yi Ren was supported in part by the EPSRC DIGIT Project Ref: EP/T022566/1 and web: https://digit.ac.uk/. The work of Wei Ren was supported in part by the provincial key research and development program of Hubei (No.2026BAB087).

REFERENCES

[1] Saad, Walid, Mehdi Bennis, and Mingzhe Chen. "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems." *IEEE Netw.*, vol. 34, no. 3, pp. 134-142, 2019.
[2] De Gaudenzi, Riccardo, Marco Luise, and Luca Sanguinetti. "The open challenge of integrating satellites into (beyond-) 5G cellular networks." *IEEE Netw.*, vol. 36, no. 2, pp. 168-174, 2022.
[3] 3GPP TR 23.700-27, "Study on 5G system with satellite backhaul", V18.0.0, 2022.
[4] C. Ge et al., "QoE-assured live streaming via satellite backhaul in 5G networks," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 381-391, 2019.
[5] TS 22.105, "Services and service capabilities", V17.0.0, 2022.

[6] Ma, Sami, et al., "Network characteristics of LEO satellite constellations: A starlink-based measurement from end users," *in Proc. IEEE INFOCOM*, 2023, pp. 1-10.
[7] https://www.tec.gov.in/STUDY_PAPER_MEO_Radio_Division.eps
[8] Jayaraman, Ramkumar, et al. "Effective resource allocation technique to improve QoS in 5G wireless network." *Electronics*, vol. 12, no. 2, pp. 451-461, 2023.
[9] Xiong, Ruoting, et al. "From 5G to 6G: It is time to sniff the communications between a base station and core networks." *in Proc. ACM Mobicom*, 2023, pp. 1-2.
[10] Lin, Xingqin, et al. "5G from space: An overview of 3GPP non-terrestrial networks." *IEEE Commun. Standards Mag.*, vol. 5, no. 4, pp. 147-153, 2021.
[11] T. Ma, B. Qian, X. Qin, X. Liu, H. Zhou and L. Zhao, "Satellite-terrestrial integrated 6G: An ultra-dense LEO networking management architecture," *IEEE Wireless Commun.*, early access.
[12] Zhen, Li, et al. "Energy-efficient random access for LEO satellite-assisted 6G internet of remote things." *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5114-5128, 2020.
[13] Wang, Zhifei, et al. "Random access optimization for initial access and seamless handover for 5G-satellite network." *Comput. Netw.*, vol. 214, pp. 109176, 2022.
[14] Caini, Carlo, et al. "Analysis of TCP live experiments on a real GEO satellite testbed." *Perform. Eval.*, vol. 66, no. 6, pp. 287-300, 2009.
[15] Li, Yuanjie, et al. "A networking perspective on starlink's self-driving LEO mega-constellation." *in Proc. ACM Mobicom*, 2023, pp. 1-16.
[16] Priya, Bhanu, and Jyoteesh Malhotra. "5GhNet: An intelligent QoE aware RAT selection framework for 5G-enabled healthcare network." *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 7, pp. 8387-8408, 2023.
[17] Habib, Md Arafat, et al. "Traffic steering for 5G multi-RAT deployments using deep reinforcement learning." *in Proc. IEEE CCNC*, 2023, pp. 164-169.
[18] Passas, Virgilios, et al. "Dynamic RAT selection and pricing for efficient traffic allocation in 5G HetNets." *in Proc. IEEE ICC*, 2019, pp. 1-6.
[19] Singh, Mohit Kumar, Shwetha Vittal, and A. Antony Franklin. "Serens: Self regulating network slicing in 5G for efficient resource utilization." *in Proc. IEEE 5GWF*, 2020, pp. 590-595.
[20] Sajjad, Muhammad Mohtasim, et al. "Inter-slice mobility management in 5G: Motivations, standard principles, challenges, and research directions." *IEEE Commun. Standards Mag.*, vol. 6, no. 1, pp. 93-100, 2022.
[21] Abidi, Mustufa Haider, et al. "Optimal 5G network slicing using machine learning and deep learning concepts." *Comput. Stand. Interfaces*, vol. 76, pp. 103518, 2021.
[22] Salau, Nurudeen Oladehinbo, and Muhammad Zeeshan Shakir. "Machine learning analysis of multi-radio access technology selection in 5G NSA network." *in Proc. IEEE WCNC*, 2023, pp. 1-6.
[23] A. Roy, P. Chaporkar, A. Karandikar and P. Jha, "Online radio access technology selection algorithms in a 5G multi-RAT network," in *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 1110-1128, 2023.
[24] De Santis, Emanuele, et al. "Satellite integration into 5G: Deep reinforcement learning for network selection." *Mach. Intell. Res.*, vol. 19, no. 2, pp. 127-137, 2022.
[25] Thantharate, Anurag, et al. "DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks." *in Proc. IEEE UEMCON*, 2019, pp. 762-767.
[26] Mohammedali, Noor Abdalkarem, et al. "Traffic classification using deep learning approach for end-to-end slice management in 5G/B5G." *in Proc. IEEE ICTC*, 2022, pp. 357-362.
[27] Bakmaz, Bojan, Zoran Bojkovic, and Miodrag Bakmaz. "TOPSIS-based approach for network slice selection in 5G mobile systems." *Int. J. Commun. Syst.*, vol. 33, no. 11, pp. e4395, 2022.
[28] Villota-Jacome, William F., Oscar Mauricio Caicedo Rendon, and Nelson LS da Fonseca. "Admission control for 5G core network slicing based on deep reinforcement learning." *IEEE Sys. J.*, vol. 16, no. 3, pp. 4686-4697, 2022.
[29] Morgado, António J., Firooz B. Saghezchi, Pablo Fondo-Ferreiro, Felipe Gil-Castiñeira, and Jonathan Rodriguez. "Intelligent backhaul link selection for traffic offloading in B5G networks." *IEEE Access*, vol. 12, pp. 106757-106769, 2024.
[30] Alam, Henri, Antonio De Domenico, David López-Pérez, and Florian Kaltenberger. "Optimizing integrated terrestrial and non-terrestrial networks performance with traffic-aware resource management." *arXiv preprint arXiv:2410.06700*, 2024.
[31] Borsatti, Davide, Arcangela Rago, Giuseppe Piro, Luigi Alfredo Grieco, Giuseppe Araniti, Sara Pizzi, Federica Rinaldi *et al.* "Resource orchestration in a terrestrial and non-terrestrial framework." *in Proc. AEIT*, 2025, pp. 1–6.
[32] Zhai, Liuqun, et al. "Identify what you are doing: Smartphone apps fingerprinting on cellular network traffic." *in Proc. IEEE ISCC*, 2021, pp. 1-7.
[33] J. Chen, C. Yi, R. Wang, K. Zhu and J. Cai, "Learning aided joint sensor activation and mobile charging vehicle scheduling for energy-efficient WRSN-based industrial IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5064-5078, 2023.
[34] J. Chen, C. Yi, J. Li, K. Zhu and J. Cai, "A triple learner based energy efficient scheduling for multi-UAV assisted mobile edge computing," *in Proc. IEEE ICC*, 2023, pp. 3240-3245.
[35] J. Chen et al., "Generative AI-aided QoE-aware resource allocations for RIS-assisted digital twin interaction with uncertain evolution," *IEEE Trans. Mob. Comput.*, vol. 25, no. 6, pp. 7888-7905, 2026.
[36] G. Soós, D. Ficzere and P. Varga, "User group behavioural pattern in a cellular mobile network for 5G use-cases," *in Proc. IEEE/IFIP NOMS*, 2020, pp. 1-7.
[37] Qiao, Zhuang, et al. "Encrypted 5G over-the-top voice traffic identification based on deep learning." *in Proc. IEEE ISCC*, 2021, pp. 1-7.
[38] R. Xiong, M. Xiong, Y. Ren, W. Ren and G. Parr, "Securing FL-enabled NWDAF in private 5G networks against hybrid attacks." *IEEE Trans. Green Commun. Netw.*, vol. 10, pp. 4392-4407, 2026.
[39] Kassas, Zaher M., and Joe Saroufim. "LEO PNT frameworks for non-cooperative satellites with poorly known ephemerides: Open-loop SGP4, tracking, and differential." *IEEE Aerosp. Electron. Syst. Mag.*, vol. 40, no. 9, pp. 28-45, 2025.
[40] Seeram, Siva Satya Sri Ganesh, Luca Feltrin, Mustafa Ozger, Shuai Zhang, and Cicek Cavdar. "Handover delay minimization in non-terrestrial networks: Impact of open RAN functional splits." *in Proc. IEEE ASMS/SPSC*, 2025, pp. 1–8.
[41] Yaacoub, Elias, and Mohamed-Slim Alouini. "A key 6G challenge and opportunity—Connecting the base of the pyramid: A survey on rural connectivity." *in Proc. IEEE*, 2020, pp. 533-582.

VII. Biography Section



Ruoting Xiong received the master's degree in computer science from Huazhong University of Sciences and Technology, Wuhan, China, in 2022. She is currently pursuing the Ph.D. degree with the School of Computing Sciences, University of East Anglia, Norwich, U.K. Her research topics are blockchain, 5G, and artificial intelligence.



Kit-Lun Tong received the master's degree in computer science from the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan. He is currently pursuing the Ph.D. degree with the School of Computing Sciences, University of East Anglia, Norwich, U.K. His research topics are IoT and artificial intelligence.



Yi Ren was with the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, Taiwan, as an assistant research fellow. He is currently an Associate Professor with the School of Computing Sciences, University of East Anglia (UEA), Norwich, U.K. His current research interests include the Internet of Things and 5G mobile technology.



Wei Ren (Member, IEEE) is a full Professor at the School of Computer Science, China University of Geosciences, Wuhan, China since 2013. He obtained Ph.D. degree in Computer Science from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2006. He has published more than 150 refereed papers, 1 monograph, and 3 textbooks. He has obtained 40 patents and 6 innovation awards. He is a Distinguished Member of China Computer Federation.