Sparse Bandit Learning Based Location Management for Space-Ground Integrated Networks

Huasen He, *Member, IEEE,* Changkun Qin, Shuangwu Chen, *Member, IEEE,* Xiaofeng Jiang, *Member, IEEE,* Jian Yang, *Senior Member, IEEE,* and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—The Space-Ground Integrated Network (SGIN) concept constitutes a promising solution for providing seamless global coverage. However, the mobility of satellites and wireless terminals imposes unprecedented challenges on the location management in SGIN. We tackle this challenge by conceiving a split identifier (ID) and Network Address (NA) based design for providing natural mobility support, and characterize the ID-NA mapping allocation problem by exploiting the storage capacity of both Geostationary Earth Orbit Satellites (GEOSs) and Low Earth Orbiting Satellites (LEOSs) to form a spatially distributed binding resolution system and optimize the caching reward in each LEOS. By considering the large quantity of ID-NA mapping and the sparsity of popular mapping having positive mean caching rewards, we formulate the mapping allocation problem as a sparse Multi-Armed Bandit (MAB) learning procedure, where the mappings are treated as the arms and the LEOSs act as the players. A distributed learning algorithm, namely the Sparse Upper confidence bound based Learning aided Caching algorithm (SULC), is proposed for estimating the mean caching rewards of mappings and selecting the optimal mappings for caching. Moreover, we derive a sub-linear upper bound of the cumulative learning regret to prove the learning efficiency of the proposed SULC. Extensive simulations have been conducted to show that the proposed SULC can quickly identify the popular mappings and provide near-optimal content hit rates. In contrast with the existing solutions, SULC has higher caching rewards and can significantly reduce the cumulative regret after a short period of learning.**

***Index Terms*—Bandits theory, Mapping allocation, Location management, Reinforcement learning.**

This work was supported by National Key R&D Program of China (2022YFB3902800), by the National Natural Science Foundations of China (Grant No. 62173315, 62101525, 62201543 and 62021001), by the Youth Innovation Promotion Association CAS (Grant No. 2020450), by the Strategic Priority Research Program of CAS (Grant No. XDC07020200), by the Fundamental Research Funds for the Central Universities, and by the China Environment for Network Innovations (CENI).

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/W016605/1 and EP/X01228X/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028). *(Corresponding author: Lajos Hanzo)*

H. He, C. Qin, S. Chen, X. Jiang and J. Yang are with the Department of Automation, University of Science and Technology of China, Hefei, Anhui, 230027, China and are also with Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui, 230026, China. E-mail: hehuasen@ustc.edu.cn; sa20218128@mail.ustc.edu.cn; chensw@ustc.edu.cn; jxf@ustc.edu.cn; jianyang@ustc.edu.cn.

L. Hanzo is with the Department of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK. E-mail:lh@ecs.soton.ac.uk.

I. Introduction

The terrestrial networks have been developed to cope with the explosive growth of data traffic generated by smart devices including smart vehicles, Unmanned Aerial Vehicles (UAVs), wearable devices and so on. However, the existing terrestrial networks have limited coverage as well as high construction cost, hence covering remote rural areas, deserts and oceans is economically unviable [1]–[3]. Inspired by the accelerating globalization, there is an urgent need for developing new low-latency, high-speed communication networks having seamless global coverage.

Given their large footprint, satellite networks constitute a promising complement to terrestrial networks. Hence, the integration of space and ground networks has attracted substantial attention from researchers. However, in contrast to terrestrial networks, the mobility of satellites and mobile terminals (MTs) results in dynamically time-varying topology and frequent handovers, which makes mobility management a critical issue in Space-Ground Integrated Networks (SGINs) [4], [5]. Several authors have tackled this challenge by focusing on the mobility management of IP-based satellite networks, where the IP protocol or its enhanced variants (e.g. Mobile IP, MIPv6 and the Seamless IP Diversity-based General Mobility Architecture (SIGMA) [6]) originally designed for terrestrial networks were adopted [4], [5], [7]–[9]. A handover-independent IP mobility management scheme was proposed for Low Earth Orbit Satellite (LEOS) networks in [7], where the geographical location information was exploited for reducing satellites' binding update frequency, which may be viewed as the counterpart of the terrestrial handover update frequency. In [8], the satellite networks considered were divided into multiple virtual agent domains, so as to support distributed mobility management. By deploying an augmented reconfigurable management plane on non-LEO satellites, the authors of [4] proposed a space-distributed mobility management architecture for achieving flexible mobility management function (MMF) configuration. In [9], the pros and cons of applying the current IP-based mobility management methods to LEOS networks have been analyzed, while it was argued that reducing the management cost should be a future challenge. The authors of [5] have reviewed the current developments in location management of IP-based LEOS networks with specific emphasis on the extensions of the IETF location management techniques.

In IP protocols, the devices are identified by their IP addresses, which are also used for routing [10]. The IP address of a node is configured according to its logical network

location. In the existing IP protocols, the IP address will be changed once a handover occurs and the updated binding information should be reported to the network. However, given the frequent handovers in SGINs, adopting the existing IP-based protocols would result in frequent transmission failure and severe overhead required for location update, which would waste precious onboard resources and significantly limit the scale of the SGIN. The dual-role of the IP number as a device identifier (ID) and network address (NA) has however intrinsic limitations in the context of mobility support. Indeed, the dynamic SGIN topology, frequent handovers and intermittent connections, make the existing IP-based solutions unsuitable [11].

These drawbacks of IP protocols have inspired the development of new mobile network architectures, and there is a broad consensus behind completely splitting the role of ID and NA in these new protocols [12]–[16]. The decoupling of ID and NA is eminently suitable for natural mobility support, since a node can be identified by its unique time-invariant ID throughout the whole network. A typical emerging network architecture is that of MobilityFirst (MF) [16], which was specifically designed for coping with the nodes' mobility. Again, MF splits the ID and NA, while a globally unique identifier (GUID) is assigned to each device. Additionally, a Global Name Resolution Service (GNRS) center is set up in the MF architecture to store and update the mapping assigning the GUID to its NA. The MF packet contains the GUID of the destination node, and the corresponding NA is obtained by querying the GNRS. The centralized GNRS has substantial computing and storage requirements, thus it cannot be deployed on resource-limited satellites. Moreover, to apply the MF architecture in satellite networks, the GNRS has to accommodated by ground-based nodes, which however imposes long query delay and high updating cost.

Mobility management in IP networks can be divided into two parts, i.e. location management [17] and handover management [18]. However, the splitting of ID and NA converts mobility management into a pair of procedures including location update and binding resolution [19]. Several authors have focused their attention on the mobility management of SGINs, where the ID-NA separation based design principle was adopted. An overview of mapping resolution systems conceived for satellite networks was presented in [12], where it was argued that a distributed design and a dynamic replica placement algorithm should be adopted for reducing the location update cost and lookup latency. The authors of [20] proposed a network architecture, where the ID and NA split was adopted for improving the network's scalablity and flexibility. In [19], the authors aimed for making the movement of satellites invisible to users and proposed an indirect binding scheme, where multiple logically fixed virtual attachment points were employed for the terrestrial users. The authors of [21] proposed another architecture for mobility management in satellite networks, where the terrestrial gateways were exploited as regional mobility management entities. By deploying ground-based identifier management servers and satellite-based identifier switch routers in a SGIN, a mobile handover process was proposed in [22] for reducing the handover delay. The authors of [23] divided the Space-Air-Ground Integrated Network (SAGIN) into multiple layers including GEOS layer, LEOS layer and ground-based network layer, while an intra-domain identifier switch router was deployed in each layer for achieving mapping resolution and an inter-domain identifier management server was deployed on the ground. It was demonstrated that the mapping resolution delay can meet the low-latency requirements of SAGIN. In [24], both ground-based and satellite-based location resolvers were implemented for improving the flexibility of heterogeneous SGINs, where multiple naming spaces were adopted for locating a network device. The ground-based location resolver adopted in [21], [24] suffered from long Satellite-to-Ground (S2G) transmission delay. It has become a broad consensus to deploy location resolvers in satellites [12], [19], [20], [22], [23]. However, the existing satellite-based solutions dynamically deployed the location resolvers in different satellites, which resulted in non-negligible communication overhead for the inter-satellite links. Moreover, the distributed onboard caching resources and the historical lookup information of ID-NA mapping were not exploited. Considering the high-mobility of SGINs and the limitations of existing solutions, an efficient location management system should be designed for achieving low update cost, low lookup latency and distributed binding resolution services.

Inspired by the availability of onboard storage distributed in LEOSs, caching some popular ID-NA mappings in each LEOS is capable of facilitating prompt lookup responses. However, given the limited storage, the cached mappings should be carefully selected. Yet, this specific location management problem of SGINs has not been addressed in the open literature - most of the content allocation research in SGINs is focused on file caching. In [25], a three-layer cooperative caching model was proposed for minimizing the average content acquisition delay of SGINs, where two caching strategies were conceived for exploiting base station (BS) aided, satellite and gateway based caching. By jointly formulating the content placement and multi-hop delivery as an optimization problem for satellite download based resource allocation, the authors of [26] provided a popularity-driven content placement algorithm, where a distance-sensitive popularity parameter was relied upon for multi-hop delivery. The authors of [27] considered a multi-layer satellite network, where the limited storage of Geostationary Earth Orbit Satellites (GEOSs) was exploited for balancing the load across LEOSs. By exploiting the popular Stackelberg game, a load balancing scheme was proposed, while a popularity matching algorithm was conceived for caching aided resource allocation. By dividing the satellite-integrated content-centric network into different regions in conjunction with virtual locations, the authors of [28] exploited a so-called DeepHawkes framework to predict the popularity of files and proposed a delay minimization caching replacement algorithm. The cache placement and content delivery strategies of SGINs were jointly optimized in [29] for minimizing the delivery delay, while deep Q-learning was leveraged to learn the optimal policies. In [30], both cache placement and cooperative beamforming were optimized jointly to enhance the network performance of a

integrated satellite-terrestrial network. The existing content allocation solutions [25]–[30] rely on the interaction between satellites or satellites and the ground BSs for synchronizing the caching state or request information. In contrast to the traditional contents, the ID-NA mapping information should be updated frequently due to the frequent handovers in SGINs. Adopting the existing content allocation strategies for mapping information caching may impose severe synchronization overhead. In addition, existing research contributions have demonstrated that the accurate prediction of file popularity can significantly improve the caching efficiency. However, the popularity of mapping contents may vary in different network coverage scenarios and the time-varying satellite coverage makes it difficult to estimate the content popularity in a timely manner. Moreover, the existing deep-learning based popularity estimating algorithms [26], [28], [29] are difficult to implement in location management, since long training time is required for a large number of content items.

Motivated by the drawbacks of the existing location management solutions and content allocation strategies, we propose a multi-layer binding resolution system, where both GEOS and LEOSs are employed for supporting a distributed mapping resolution service. By dividing the SGIN into multiple domains according to the coverage of GEOSs, we consider each GEOS to be the regional binding resolution service center. Compared to GEOSs, LEOSs are closer to the ground and can provide low-latency services. Inspired by the availability of onboard caching and processing capabilities of LEOSs, we exploit the historical mapping lookup information for caching the most popular ID-NA mappings at each LEOS for supporting a prompt lookup response. Against this backdrop, the main contributions of this paper are summarized as follows.

1) The frequent updates of mappings and the time-varying resolvers impose synchronization overhead for existing satellite-based solutions. For achieving distributed location management at a low synchronization overhead and prompt lookup response, we propose a multi-layer binding resolution system for ground-based mobile terminals (MTs), where our SGIN is divided into multiple GEOS-based domains and each GEOS acting as the domain-head provides regional binding resolution service. Then, each domain is further broken down into different clusters consisting of a single LEOS and multiple associated MTs, where replicas of the most popular ID-NA mappings are stored in the LEOSs for providing prompt lookup response.
2) The rapidly fluctuating satellite coverage makes it critical to identify the popular mappings in a timely manner. However, the large quantity of ID-NA pairs and the potential change of unknown mapping popularity impose unprecedented challenges on popularity-aware caching strategies. For maximizing the caching reward in each LEOS, we exploit the sparsity of the popular mappings and formulate the replica allocation procedure as a sparse bandit learning problem for promptly identifying the popular mapping contents. By modeling each mapping as an arm in the multi-armed bandit (MAB) procedure and treating the LEOSs as the players, we propose a Sparse Upper confidence bound based Learning aided Caching algorithm (SULC) for estimating the expected caching rewards of mappings and dynamically allocating the replicas at each LEOS.
3) For ensuring the convergence guarantee of the proposed SULC, we derive a sub-linear upper bound of the cumulative regret, which represents the difference between the optimal reward associated with all the popularity distributions of the mappings known in advance and the actual reward of SULC. In contrast to the classical MAB algorithms, where the regret typically scales sub-linearly with the total number of arms M, by leveraging the knowledge that there are K ($K < M$) mappings having popularity higher than a pre-defined value and each LEOS can cache γ ($\gamma \leq K$) mappings, the regret of SULC scales sub-linearly with $(K - \gamma)$ instead of M, which demonstrates the efficiency of our proposed algorithm.
4) Extensive simulations have been conducted for validating the efficiency of the proposed SULC algorithm. We show that the proposed SULC provides higher caching reward and lower cumulative regret than the existing algorithms after a short period of learning, since it converges promptly by effectively identifying the most popular mappings for providing near-optimal content hit rates.

The remainder of this paper is organized as follows. Section II presents our system overview. The formulation of the mapping allocation problem is provided in Section III, while Section IV introduces our distributed learning based location management. Then, Section V presents our simulation results and discussions. Finally, Section VI concludes the paper and summarizes our key contributions.

II. System Overview

A. System Model

In this paper, we consider the SGIN shown in Fig. 1 which is consisted by satellites (including GEOSs and LEOSs), Terrestrial Gateways (TGs) and Mobile Terminals (MTs). The set of GEOSs is denoted by $\mathcal{S}_{GEO} = \{s_1^G, s_2^G, ..., s_g^G\}$, where g represents the number of GEOSs. We assume that there are n_o LEO orbits, and each LEO orbit has n_s satellites. The LEOSs are represented by $\mathcal{S}_{LEO} = \{s_1^L, s_2^L, ..., s_{n_o \times n_s}^L\}$. We use $\mathcal{S}_{gw} = \{s_1^{gw}, s_2^{gw}, ..., s_w^{gw}\}$ to denote TGs, while the set of MTs is expressed as $\mathcal{M} = \{1, ..., m, ..., M\}$.

For supporting ubiquitous mobility in our SGIN, the ID-NA split design is adopted. Each node has a fixed unique ID and a changeable NA. It is assumed that MTs can be served either by Terrestrial Networks (TNs) or satellites. Due to the limited coverage of TNs, the MTs, which cannot be served by Terrestrial Networks (TNs), access the SGIN through LEOSs, where each LEOS can communicate with its nearest GEOS. GEOSs are connected to form a space-based backbone network, which connects with the terrestrial backbone network through TGs. Furthermore, the TGs are connected to a ground-based GNRS center. Thanks to the recent advances in satellite

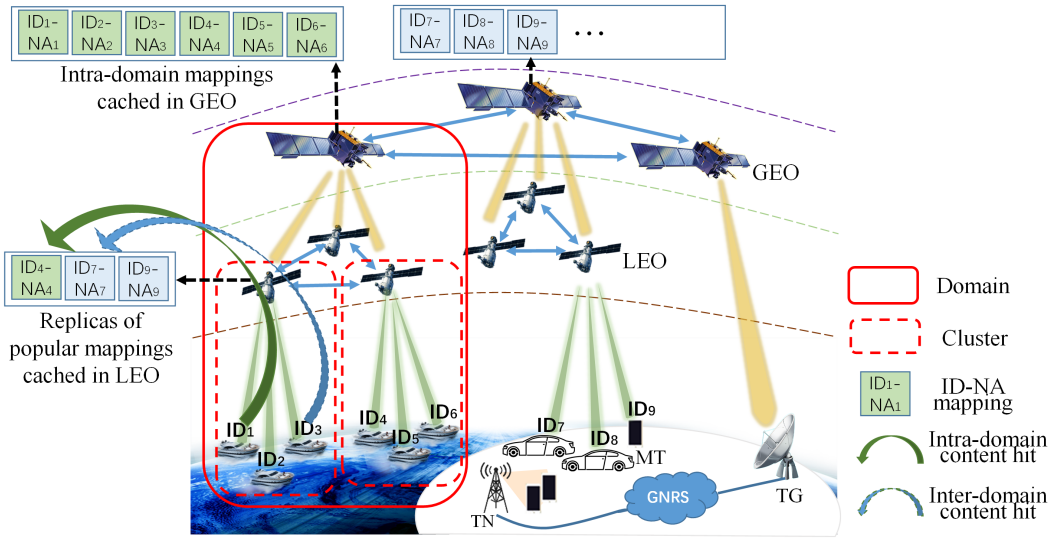


Fig. 1. System model of a space-ground integrated network.

technologies, the onboard processing and storage capabilities of both GEOSs and LEOSs have been significantly improved. Thus, we assume that both GEOSs and LEOSs are equipped with onboard processing and caching resources.

B. Multi-layer SGIN architecture

The global coverage of SGIN makes it infeasible to have a centralized binding resolution service, since the long enquiry delay and high updating cost may be unacceptable. A promising solution is to divide the network into different regions and perform mobility management within each region. A representative example is the Proxy Mobile IPv6 (PMIPv6) [31], in which the network is partitioned into multiple regions and a TG is set as the Local Mobile Anchor (LMA) for local mobility management. However, TGs can only provide limited coverage distance. Moreover, the performance of TG-based location management algorithms is degraded by the intermittency of S2G links. Motivated by PMIPv6, in this treatise we divide SGIN into multiple *domains*, according to the coverage of GEOSs. As shown in Fig. 1, each domain includes a GEOS satellite as the domain-head, multiple LEOSs connected to the GEOS, and MTs served by these LEOSs. Furthermore, the nodes in each domain are divided into multiple *clusters*, where each cluster is headed by a LEOS and it consists of the LEOS and the MTs served by it. Specifically, we group the MTs served by a TN into a single domain. By grouping the nodes in the SGIN into multiple domains and clusters, we form a multi-layer SGIN architecture.

C. Spatially distributed binding resolution system

For providing prompt lookup of the ID-NA mapping information, the mappings should be appropriately allocated. In this work, we exploit the storage of LEOSs and GEOSs for constructing a spatially distributed binding resolution system. As shown in Fig. 1, both the LEOSs and GEOSs are equipped with onboard processing and storage resources for storing the ID-NA mapping and for processing the mapping lookup requests. However, given the limited onboard storage, each GEOS only caches and maintains the ID-NA mapping information for the nodes within the same domain. In addition to maintaining the mapping information of its serving MTs, each LEOS reserves a fixed storage capacity for caching the replicas of popular mappings. When a MT intends to communicate with a destination node (DN), it will first look up the destination NA. A binding resolution request will be sent to its serving LEOS. If the mapping of the DN is cached by the serving LEOS, the destination NA can be obtained locally and sent back to the MT for triggering the routing procedure. Otherwise, the serving LEOS will send a lookup request to its domain-head (i.e. the GEOS). If the destination is an intra-domain node, a response message containing the destination NA will be returned to the MT through its serving LEOS. Otherwise, the lookup request will be forwarded to the neighboring domains until the destination ID is resolved. Thus, the lookup latency can be significantly reduced, once the mapping is cached locally. Specifically, it is assumed that the mapping of the MTs served by TN is stored in the ground-based GNRS, while the TGs are connected with the GNRS for supporting mapping lookup. By storing part of mapping in GEOSs and LEOSs, we form a spatially distributed binding resolution system.

D. Popularity distribution of mapping

For each MT, we model its lookup requests as an independent Poisson arrival process with density $\lambda > 0$. The relative frequency of requesting the ID-NA mapping of a MT can be modeled as a popularity distribution. Following the setups in [25], we use $\mathcal{F} = \{f_1, f_2, ..., f_K, f_{K+1}\}$ to denote the set of all mappings, while f_k $(k \leq K)$ represents the k^{th} most popular mapping and the set of the remaining $(M-K)$ mappings is expressed as f_{K+1}. In the t^{th} time slot, we use $d_{l,m}(t) \in [0,1]$ to denote the normalized number of requesting the mapping of MT m in the l^{th} cluster, which is normalized by the maximum number of requests. Then, we have the popularity distributions of mappings denoted by $\Theta = \{\theta_{l,m}, \forall l \in \mathcal{S}_{LEO}, \forall m \in \mathcal{M}\}$, where $\theta_{l,m}$ is the mean of

$d_{l,m}(t)$ and is distributed according to a Zipf-like distribution [32], which is given by

$$\theta_{l,m} = \begin{cases} \frac{1}{|\Upsilon_{l,m}|^{\delta_l} \sum_{k\in\mathcal{F}} k^{-\delta_l}} \ \forall l \in \mathcal{S}_{LEO}, \forall m \in \{f_1, f_2, ..., f_K\} \\ \frac{1}{(M-K)|\Upsilon_{l,K+1}|^{\delta_l} \sum_{k\in\mathcal{F}} k^{-\delta_l}} \\ \qquad\qquad \forall l \in \mathcal{S}_{LEO}, \forall m \in f_{K+1}. \end{cases} \tag{1}$$

Here, for simplicity, we assume that all the mappings in f_{K+1} have the same popularity. In, (1), δ_l represents the Zipf exponent, which reflects the skewness of popularity distribution, and a larger δ_l means that the majority of requests is concentrated on a few MTs. Furthermore, $\Upsilon_{l,m}$ is given by

$$\Upsilon_{l,m} = \begin{cases} \sigma_l - m, \ when \ \sigma_l \neq m \\ 1, \ when \ \sigma_l = m \end{cases}, \tag{2}$$

where σ_l denotes the shift of popularity distribution in the l^{th} cluster, which is used for reflecting the diverse preferences of MTs in different clusters. Regarding the practical limitations, LEOSs have no knowledge of σ_l and δ_l in advance.

III. Problem Formulation

In this paper, we focus on the allocation of ID-NA mapping in SGINs, and formulate it as an optimization problem, which is solved by exploiting an online learning method for maximizing the caching reward in each LEOS.

A. Problem formulation for caching reward maximization

Considering the limited onboard storage, only part of the ID-NA mapping can be distributively allocated in LEOS and GEOSs, while the GNRS maintains the mapping information of MTs served by the TN. The lookup latency of a binding resolution request can be significantly reduced if a *content hit* event occurs, i.e. the mapping can be retrieved in a MT's serving satellite. Therefore, one of our objectives is to maximize the content hit rate of the cached mapping in LEOSs. Specifically, if the DN is located in the same domain as the source node (SN), this event is termed as an *intra-domain content hit*, otherwise, the event is classified as an *inter-domain content hit*. In case the requested mapping cannot be found in the cluster-head, the serving LEOS will forward the lookup request to the domain-head.

For the t^{th} time slot (TS), we use $B(t) = \{B_{l,m}(t), \forall l \in \mathcal{S}_{LEO}, \forall m \in \mathcal{M}\}$ to represent the caching strategy. We have $B_{l,m}(t) = 1$ if the mapping of MT m is cached in the LEOS s_l^L. Otherwise $B_{l,m}(t) = 0$. Considering the limited storage in s_l^L, we have

$$\delta_s \sum_{m\in\mathcal{M}} B_{l,m}(t) \leq S_l, \tag{3}$$

where δ_s is the size of each mapping. Then, the maximum number of mappings that can be cached by s_l^L is denoted by $\gamma_l = \left\lfloor \frac{S_l}{\delta_s} \right\rfloor$. We use a random variable $h_l(t)$ to denote the number of content hits in s_l^L at TS t under the ID-NA mapping allocation strategy $B_l(t)$. Then, $h_l(t)$ is given by

$$h_l(t) = \sum_{m\in\mathcal{M}} U_l d_{l,m}(t) B_{l,m}(t). \tag{4}$$

Here, U_l represents the maximum number of binding lookup events in the l^{th} cluster and $U_l d_{l,m}(t)$ represents the number of times that the mapping of m is requested.

It is worth noticing that inter-domain binding lookup events impose longer delay and higher cost than intra-domain events. Therefore, we further partition the content hits into intra-domain and inter-domain content hits, which are given by

$$h_l^{intra}(t) = \sum_{m\in\mathcal{M}} D_{l,m} U_l d_{l,m}(t) B_{l,m}(t), \tag{5}$$

$$h_l^{inter}(t) = \sum_{m\in\mathcal{M}} (1 - D_{l,m}) U_l d_{l,m}(t) B_{l,m}(t), \tag{6}$$

where $D_{l,m} = 1$ if the MT m is located in the same domain of s_l^L. Otherwise, $D_{l,m} = 0$. Then, we capture the reward of s_l^L upon adopting the caching strategy $B_l(t)$ in TS t as

$$\begin{aligned} R_l(t) &= g_{inter} \times h_l^{inter}(t) + g_{intra} \times h_l^{intra}(t) \\ &= \sum_{m\in\mathcal{M}} U_l d_{l,m}(t) G_{l,m} B_{l,m}(t), \end{aligned} \tag{7}$$

where the constants g_{inter} and g_{intra} represent the caching gains of inter-domain and intra-domain content hits, respectively. Briefly, the caching gains are used for quantifying the benefit of mapping caching and can be interpreted as a general expression of the traffic offloading, energy saving or delay reduction. Since all the intra-domain contents can be retrieved from the corresponding GEOS, the intra-domain content hits have the same caching gain g_{intra}. Moreover, global coverage can be achieved by deploying as few as 3 GEOSs. Thus, we assume that the inter-domain contents can be retrieved from a neighboring domain-head and the caching gains of inter-domain content hits can also be set as a constant g_{inter}. $G_{l,m} = g_{inter} D_{l,m} + g_{intra}(1 - D_{l,m})$ is the caching gain of the mapping of m, and different caching strategies may be constructed for inter- and intra-domain mappings by adjusting the values of g_{inter} and g_{intra}. Similarly, we define the caching cost of fetching the mapping as

$$C_l(t) = \sum_{m\in\mathcal{M}} \beta G_{l,m} B_{l,m}(t), \tag{8}$$

where the constant β is a weighting factor. Thus, the total reward of adopting the caching strategy $B_l(t)$ can be written as

$$\begin{aligned} R_l^N(t) &= R_l(t) - C_l(t) \\ &= \sum_{m\in\mathcal{M}} B_{l,m}(t) r_{l,m}(t). \end{aligned} \tag{9}$$

Here, $r_{l,m}(t)$ denotes the net reward of caching the mapping of m at TS t, which is given by

$$r_{l,m}(t) = G_{l,m}(U_l d_{l,m}(t) - \beta). \tag{10}$$

In this work, we aim for finding an optimal allocation strategy for maximizing the total reward in each LEOS. The total caching reward maximization problem is defined as

$$\begin{aligned} &\max_{B_l(t)} \left\{ \sum_{t=1}^{T} R_l^N(t) \right\} \\ &s.t. \qquad (3) \\ &\qquad B_{l,m}(t) \in \{0,1\}, \quad \forall l \in \mathcal{S}_L, \forall m \in \mathcal{M} \ , \end{aligned} \tag{11}$$

where $B_l(t) = B_{l,m}(t), \forall m \in \mathcal{M}$ is the allocation strategy of the LEOS s_l^L. According to (10), given the popularity distribution of mapping contents, the caching reward distributions are known. Then, (11) can be decomposed into T independent optimization problems and optimized in each TS. However, the caching reward distributions are unknown in advance, making it difficult to solve the optimization problem. Thus, in each TS we cache some mappings and observe the requests to improve the estimate of caching reward distribution. Moreover, it can be readily observed that the maximization problem has a nonlinear objective function (OF), which has 0-1 integer variables and linear constraints. It has been proved that this kind of problems are NP-complex [33], [34]. It is not trivial to solve this problem by 'polynomial-time' algorithms. Therefore, we provide an iterative, distributed and online learning based mapping allocation algorithm that maximizes the caching reward at each TS to asymptotically solve this problem.

B. Problem Analysis

Since the reward distributions are unknown in advance, for the sake of maximizing the caching reward, we have to strike a tradeoff between exploitation (storing the ID-NA mappings with empirically optimal rewards by leveraging historical observations) and exploration (aiming for learning the unknown caching reward distributions of the under-sampled mappings to minimize the probability of missing popular mappings). Thus, the popular Multi-Armed Bandit (MAB) algorithms may be adopted for solving this problem. However, a critical limitation of applying the traditional MAB algorithms is that the cumulative regret, capturing the difference between the optimal rewards (with all the expected rewards of arms known in advance) and the actual rewards, scales sub-linearly with the number of arms [35], [36]. Thus, an excessive number of arms will impose ultra-long convergence time for the traditional MAB algorithms. It can be observed in (10) that only a subset of mappings have positive expected rewards, since a MT is likely to communicate with a small set of popular ones. Motivated by the sparsity of mappings having positive expected rewards, the total caching reward maximization problem can be modeled as a Sparse Bandit Problem (SBP) [37], which can be solved by employing online learning for exploiting the instant reward as well as exploring the under-sampled mappings. In this context, LEOSs are treated as the players, while the mappings of ID-NA pairs are viewed as the arms in the Sparse Bandit Learning (SBL).

C. The Sparse Bandit Problem

As a variant of the classical bandit problem, the SBP is designed for sparse scenarios, in which only a small set of arms have positive rewards. In a MAB algorithm, there are M arms and N players. It is assumed that the reward of the i^{th} arm ($i \in 1, ..., M$) is randomly distributed with an unknown mean. At each TS, the players obtain random rewards by playing the arms, which were decided by the bandit algorithm in the previous TS according to the preceding observations. By exploiting the historical information, the caching reward obtained will approach that of the optimal arm. In contrast to the classical bandit problem, the SBP further assumes that there are K unknown arms having positive expected rewards and $K < M$. The core idea of the SBL algorithm is to promptly identify the K arms having positive expected rewards and then activate the bandit algorithm among these arms. Thus, the regret of the SBL algorithm increases with K instead of M in the classical bandit algorithms [37]. In light of this, we will conceive a sparse bandit based online learning algorithm for solving the mapping allocation problem.

IV. Distributed Learning Based Location Management

In this section, we exploit the SBL method for solving our mapping allocation problem. We first propose a sparse Upper Confidence Bound (UCB) based learning algorithm for maximizing the caching reward of each LEOS. Then, the performance of the sparse UCB learning algorithm is analyzed in terms of regret.

A. Sparse UCB Learning on Caching Reward Distribution

Recall from (11), that we are interested in maximizing the caching reward in each LEOS, since LEOSs have limited onboard storage and could learn the served users' requests locally. Again, we formulate the caching reward maximization problem as a SBP, where the mappings of ID-NA pairs correspond to the arms and the learning agent in each LEOS acts as the player. In contrast to the classical bandit problems, we assume that each LEOS has a *priori* information[1] that there are K_l unknown MTs whose popularity is above a pre-defined threshold $\frac{\beta}{U_l}$. For simplicity, we refer to these K_l unknown MTs as popular MTs, while the rest of MTs are referred to as unpopular MTs. Similarly, the corresponding mappings are divided into popular and unpopular mappings, respectively.

In each TS, the LEOS s_l^L will play with γ_l arms simultaneously (i.e. caching the mapping of γ_l MTs) according to the learning results. Then, the estimated caching reward for the mapping of MT m can be written as

$$\bar{r}_{l,m}\left[N_m(t)\right] = \frac{1}{N_m(t)} \sum_{\tau=1}^{t-1} B_{l,m}(\tau) r_{l,m}(\tau), \tag{12}$$

where $N_m(t)$ denotes how many times the mapping of m was cached up to the $(t-1)$ TS. $N_m(t)$ is given by

$$N_m(t) = \sum_{\tau=1}^{t-1} B_{l,m}(\tau). \tag{13}$$

By leveraging the sparsity of the popular mappings, we propose the SULC procedure of Algorithm 1, which consists of three different phases, including the Round-robin phase, the Force-log phase and the UCB phase. In each TS, the learning agent pulls γ_l arms simultaneously (i.e. caching γ_l mappings) and updates the estimated caching rewards of the corresponding arms according to the requests observed. For identifying the popular mappings and activating the transition

[1]This *priori* information can be obtained by leveraging the historical observations of users' requests.

between the three phases, we create the active MT set $A(t)$ as well as the active and sufficiently sampled MT set $S(t)$, which are defined as [37]

$$A(t) = \left\{ m \in \mathcal{M} \mid \overline{r}_{l,m}(N_m(t)) \geq \sqrt{\frac{\alpha \log\left(N_m(t)\right)}{N_m(t)}} \right\}, \tag{14}$$

$$S(t) = \left\{ m \in \mathcal{M} \mid \overline{r}_{l,m}(N_m(t)) \geq \sqrt{\frac{\alpha \log(t)}{N_m(t)}} \right\}, \tag{15}$$

where α is a positive constant no less than 1.5 [35].

As shown in Algorithm 1, the learning procedure starts with a Round-robin phase, where the learning agent sequentially caches γ_l different mappings in a TS until all the mappings have been cached once. After that, the active MT set as well as the active and sufficiently sampled MT set will be updated. If the number of active MTs (denoted as $|A(t)|$) is fewer than K_l, the learning algorithm will enter the Round-robin phase. Otherwise, if there are fewer than K_l MTs in $S(t)$, the Force-log phase will be activated and the learning agent will randomly select γ_l MTs in the different set of $A(t)$ as well as $S(t)$, and their mappings will be cached in the next iteration. When the number of active and sufficiently sampled MTs is no fewer than K_l, the UCB policy is employed. The UCB policy assigns an upper confidence index to each arm in the active and sufficiently sampled mapping set, which denotes the biased evaluation of that arm based on the previous observations of the rewards obtained in the learning procedure [35]. The upper confidence index of the mapping of m at time slot t is used to represent the estimated mean reward of m, and is defined as

$$\hat{r}_{l,m} = \overline{r}_{l,m}(N_m(t)) + \sqrt{\frac{\alpha \log(t)}{N_m(t)}}. \tag{16}$$

In the UCB phase, the mappings of the optimal γ_l MTs having the highest upper confidence indices will be cached in the next iteration.

The SBL algorithm aims for promptly identifying the K_l popular mappings, and then employing the UCB policy for selecting γ_l optimal mappings for caching. The Round-robin phase is set to guarantee that all the popular mappings can be active after performing a finite number of Round-robin phases. In case the potential optimum is inactive, there are active but unpopular mappings. If an unpopular mapping is in $S(t)$, the UCB policy ensures that it will be sufficiently sampled. Then, the unpopular mappings will be removed from $S(t)$, making $S(t) < K_l$ and the learning procedure enters the Force-log phase. For the active but unpopular mappings, the Force-log phase is designed to ensure that all the active mappings can be sufficiently sampled and the unpopular mappings will become inactive after a finite number of iterations. Therefore, if the potential optimum happened to be inactive, the number of active mappings would drop below K_l, and the Round-robin phase will be performed to quickly make it active again.

To implement the proposed SULC in a SGIN having distributed caching resources, SULC operates on a time slot by TS manner and the period of time slot is set to be much higher than the maximum binding resolution delay, so that the LEOS can obtain the binding resolution results in the same time slot. Before performing SULC, MTs access their serving LEOSs and are assigned NAs. Each LEOS will store the ID-NA mappings of its serving MTs and will forward these mappings to its serving GEOS, which acts as the local binding resolution center. At the beginning of each time slot, the LEOS performs SULC to cache a set of selected mappings, which may be obtained locally or retrieved from GEOSs. In

Algorithm 1: Sparse UCB Learning based Caching Algorithm

Input: the total number of MTs (i.e. M), the number of mappings with positive mean rewards (i.e. K_l), the maximum number of mappings can be cached by a LEOS (i.e. γ_l), and the number of iterations T;
Output: the set of MTs selected in each TS (i.e. $s(t)$), whose mappings will be cached;
Initialization: $t \leftarrow 1$;
for $\tau = 1, 2, ..., \lceil \frac{M}{\gamma_l} \rceil$ **do**
  $\%Round-robin\ phase$;
  $I_{max} = \min(M, (\tau+1)\gamma_l - 1)$;
  $s(t) \leftarrow [\tau\gamma_l, \tau\gamma_l + 1, ..., I_{max}]$;
  $t \leftarrow t + 1$
end
Iteration: **while** $t \leq T$ **do**
  Update the active MT set $A(t)$, and the active and sufficiently sampled MT set $S(t)$ according to (14) and (15) respectively.
  if $|A(t)| < K_l$ **then**
    for $\tau = 1, 2, ..., \lceil \frac{M}{\gamma_l} \rceil$ **do**
      $\%Round-robin\ phase$;
      $I_{max} = \min(M, (\tau+1)\gamma_l - 1)$;
      $s(t) \leftarrow [\tau\gamma_l, \tau\gamma_l + 1, ..., I_{max}]$;
      $t \leftarrow t + 1$
    end
  end
  else if $|S(t)| < K_l$ **then**
    $\%Force-log\ phase$;
    if $|A(t) \backslash S(t)| \leq \gamma_l$ **then**
      $s(t) \leftarrow A(t) \backslash S(t)$;
    end
    else
      $s(t) \leftarrow$ $randomly\ select\ \gamma_l\ MTs\ in\ A(t) \backslash S(t)$;
    end
    $t \leftarrow t + 1$;
  end
  else
    $\%UCB\ phase$
    for $m \in S(t)$ **do**
      Calculate the uppder confidence indice $\hat{r}_{l,m}$;
    end
    $s(t) \leftarrow$ select γ_l optimal mappings with the highest uppder confidence indices;
    $t \leftarrow t + 1$;
  end
end

each time slot, MTs may send binding resolution requests to their serving LEOS, and the LEOS will record the local mapping content hit events, which will be used to update both the active MT set as well as the active and sufficiently sampled MT set in the next time slot. In this way, each LEOS selects the caching contents independently according to its local observation. Thus, imposing an excessive inter-satellite synchronization overhead is avoided making it practical to implement SULC in SGIN.

B. Performance analysis of SULC

Based on the definition of the net reward for caching a mapping presented in (10), we reorder the mappings of MTs in non-increasing order according to their expected caching rewards at the LEOS s_l^L. Then, we have

$$\mu_{l,1} \geq \mu_{l,2} \geq \cdots \mu_{l,\gamma_l} \geq \cdots \mu_{l,K_l} > 0 \geq \mu_{l,K_l+1} \cdots \geq \mu_{l,M}, \tag{17}$$

where $\mu_{l,m} = G_{l,m}\left(U_l\theta_{l,m} - \beta\right)$ is the expected caching reward of the mapping of MT m. For the sake of simplicity, we denote the first γ_l MTs as Γ, while the first K_l MTs are represented by $\mathcal{K}$. In each TS, the LEOS will cache γ_l mappings selected by the learning algorithm, and observe the users' requests for these mappings for updating their estimated reward distributions.

For evaluating the performance of the proposed SULC, we define the cumulative $regret$ as the performance metric, which is given by

$$Reg(T) = \bar{Re}(T) - Re(T), \tag{18}$$

where $Re(T)$ is the cumulative reward obtained in the learning procedure and $\bar{Re}(T)$ represents the caching reward gathered by always selecting the γ_l optimal mappings having the highest mean caching rewards, with $\bar{Re}(T)$ and $Re(T)$ given by

$$\bar{Re}(T) = \sum_{t=1}^{T}\sum_{m\in\Gamma}\mathbb{E}[\bar{r}_{l,m}(N_m(t))], \tag{19}$$

$$Re(t) = \sum_{t=1}^{T}\sum_{m\in\mathcal{M}} B_{l,j}(t) r_{l,m}(t). \tag{20}$$

Thus, the expected cumulative regret can be written as

$$\mathbb{E}[Reg(T)] = T\sum_{i\in\Gamma}\mu_{l,i} - \sum_{m\in\mathcal{M}}\mu_{l,m}\,\mathbb{E}[N_m(T+1)]. \tag{21}$$

Upon substituting $T = \frac{\sum_{m\in\mathcal{M}} E[N_m(T+1)]}{\gamma_l}$ into the above equation, we have

$$\mathbb{E}[Reg(T)] = \sum_{m\in\mathcal{M}} E[N_m(T+1)]\Delta_m, \tag{22}$$

where $\Delta_m = \mu^* - \mu_{l,m}$ and $\mu^* = \frac{\sum_{i\in\Gamma}\mu_{l,i}}{\gamma_l}$.

Since μ^* is a fixed constant, it can be observed from (22) that the expected cumulative regret can be interpreted as the summation of the regrets imposed by each mapping. Thus, we decompose the regret according to the different phases. For a MT $m \in \mathcal{M}$, the event $\{B_{l,m}(t) = 1\}$ can be decomposed into three cases, which are defined as

$$\mathcal{R} = \{B_{l,m}(t) = 1 | Round-robin\ phase\} \tag{23}$$

$$\mathcal{F} = \{B_{l,m}(t) = 1 | Force-log\ phase\} \tag{24}$$

$$\mathcal{U} = \{B_{l,m}(t) = 1 | UCB\ phase\}. \tag{25}$$

Then, the expected cumulative regret can be decomposed as

$$\mathbb{E}[Reg(T)] = Reg_{\mathcal{R}}(T) + Reg_{\mathcal{F}}(T) + Reg_{\mathcal{U}}(T) \tag{26}$$

where $Reg_{\mathcal{R}}(T)$, $Reg_{\mathcal{F}}(T)$ and $Reg_{\mathcal{U}}(T)$ are the expected cumulative regrets in the three different phases, which are given by

$$Reg_{\mathcal{R}}(T) = \sum_{m\in\mathcal{M}}\Delta_m\,\mathbb{E}\left[\sum_{t=1}^{T} B_{l,m}\left(t|\mathcal{R}\right)\right], \tag{27}$$

$$Reg_{\mathcal{F}}(T) = \sum_{m\in\mathcal{M}}\Delta_m\,\mathbb{E}\left[\sum_{t=1}^{T} B_{l,m}\left(t|\mathcal{F}\right)\right], \tag{28}$$

$$Reg_{\mathcal{U}}(T) = \sum_{m\in\mathcal{M}}\Delta_m\,\mathbb{E}\left[\sum_{t=1}^{T} B_{l,m}\left(t|\mathcal{U}\right)\right]. \tag{29}$$

It can be observed from (27), (28) and (29) that the expected regrets in the three different phases are independent. The upper bound of the expected cumulative regret in (22) can be obtained by upper bounding (27), (28) and (29) independently. Thus, in the following lemmas, we provide upper bounds for the expected regrets in the three different phases, respectively.

Lemma 1. *For the Round-robin phase, the expected cumulative regret can be upper-bounded by*

$$Reg_{\mathcal{R}}(T) \leq \left(1 + \sum_{k\in\mathcal{K}}\left(\hat{n}_k + \frac{2}{\mu_{l,k}^2}\right)\right)\sum_{m\in\mathcal{M}}|\Delta_m|. \tag{30}$$

Here, $\hat{n}_k$ is the minimum integer which ensures $\frac{\log(n)}{n} \leq \frac{\mu_{l,k}^2}{4\alpha}$ for all $n \geq \hat{n}_k$.

Proof. See Appendix A. □

It can be observed in Lemma 1 that the regret imposed in the Round-robin phase is constrained by a time-independent constant, which implies that only a finite number of Round-robin phases will be performed.

Lemma 2. *As for the Force-log phase, the upper bound of the expected regret is given by*

$$\begin{aligned} Reg_{\mathcal{F}}(T) \leq & \log(T) 4\alpha \sum_{k\in\mathcal{K}\setminus\Gamma}\frac{\Delta_{1,k}}{\mu_{l,k}^2} \\ & + 2\left(\sum_{k\in\mathcal{K}\setminus\Gamma}\frac{\Delta_{1,k}}{\mu_{l,k}^2} + \sum_{m\in\mathcal{M}\setminus\mathcal{K}}\Delta_{1,m}\right) \end{aligned} \tag{31}$$

where $\Delta_{1,k} = \mu_{l,1} - \mu_{l,k}$. Specifically, the regret imposed by the unpopular mappings in the Force-log phase is upper-bounded by $2\sum_{m\in\mathcal{M}\setminus\mathcal{K}}\Delta_{1,m}$.

Proof. See Appendix B. □

Lemma 2 shows that the regret imposed by unpopular mappings is upper-bounded by a time-invariant constant, which means that the unpopular MTs will become inactive after a finite number of Force-log phases.

Lemma 3. *With respect to the UCB phase, we upper bound the expected regret as*

$$Reg_{\mathcal{U}}(T) \leq \log(T) 4\alpha \sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k} \sum_{i\in\Gamma} \left(\frac{1}{\Delta_{i,k}^2}\right) + 2\left(\sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k}\left(\gamma_l + \sum_{i\in\Gamma}\frac{1}{\Delta_{i,k}^2}\right) + \sum_{m\in\mathcal{M}\setminus\mathcal{K}} \Delta_{1,m}\right). \tag{32}$$

Proof. See Appendix C. □

With the above Lemmas in place, an upper bound of the expected cumulative regret for adopting the SULC algorithm is presented in the following theorem, which guarantees the convergence of the proposed SULC.

Theorem 1. *The expected cumulative regret of SULC is upper-bounded by*

$$\mathbb{E}[Reg(T)] \leq \log(T) 4\alpha \sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k}\left(\frac{1}{\mu_{l,k}^2} + \sum_{i\in\Gamma}\frac{1}{\Delta_{i,k}^2}\right) + 2\left(\sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k}\left(\gamma_l + \frac{1}{\mu_{l,k}^2} + \sum_{i\in\Gamma}\frac{1}{\Delta_{i,k}^2}\right) + 2\sum_{m\in\mathcal{M}\setminus\mathcal{K}} \Delta_m\right) + \left(1 + \sum_{k\in\mathcal{K}}\left(\hat{n}_k + \frac{2}{\mu_{l,k}^2}\right)\right)\sum_{m\in\mathcal{M}} |\Delta_m|, \tag{33}$$

which scales sub-linearly with $(K_l - \gamma_l)$.

Proof. The upper bound in (33) can be readily obtained by combining the results in Lemma 1, 2 and 3. By omitting the multiplicative constants and the additive constants in (33), we have

$$\mathbb{E}[Reg(T)] \lesssim \log(T) \sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k}\left(\frac{1}{\mu_{l,k}^2} + \sum_{i\in\Gamma}\frac{1}{\Delta_{i,k}^2}\right) \lesssim \log(T) \sum_{k\in\mathcal{K}\setminus\Gamma} \max\left\{\frac{\Delta_{1,k}}{\mu_{l,k}^2}, \frac{1}{\Delta_{1,k}}, \frac{\Delta_{1,k}}{\Delta_{2,k}^2}, \ldots, \frac{\Delta_{1,k}}{\Delta_{\gamma_l,k}^2}\right\}. \tag{34}$$

Since (34) scales sub-linearly with $K_l - \gamma_l$, Theorem 1 is proved. □

C. Complexity analysis of SULC

In this subsection, we analyze the computational complexity of SULC in Algorithm 1, while the computational complexity represents the total number of operations executed in a single round. After the initialization, both the active MT set and the active and sufficiently sampled MT set will be updated in each time slot. Since the estimated caching reward for a mapping defined in (12) should only be updated if the mapping was cached in the previous time slot, there are a maximum of γ_l mappings that should be updated. Thus, updating (14) and (15) requires $O(\gamma_l + \mathcal{S}_t)$ operations, while $\mathcal{S}_t$ represents the number of mappings in $S(t)$. It can be observed from Algorithm 1 that the computational complexity for each Round-robin phase is $O(\gamma_l + \mathcal{S}_t + 2)$. As for the Force-log phase, its computational complexity is given by $O(\gamma_l + \mathcal{S}_t + 1)$. In the UCB phase, the upper confidence indices of the mappings in $S(t)$ should be calculated. Then, the γ_l optimal mapping contents with maximum upper confidence indices are selected for caching. Thus, the computational complexity of the UCB phase can be written as $O(\gamma_l + 2\mathcal{S}_t + 1)$. It has been proved in Subsection IV-B that SULC will perform a finite number of Round-robin and Force-log phases, and converges to the UCB phase. Moreover, the number of mappings in $S(t)$ will converge to K_l and $K_l \geq \gamma_l$. Therefore, the computational complexity of the proposed SULC is equivalent to $O(K_l)$. Since $K_l << M$, we can conclude that SULC maintains a low computational complexity.

V. Simulation Results and Discussions

In this section, we conduct extensive simulations for evaluating the performance of the proposed SULC algorithm. Moreover, SULC is compared to the existing caching strategies for demonstrating its benefits.

A. Simulation Setup

In our simulations, we assume that 3 GEOSs are equidistantly positioned for providing global coverage, while the Iridium constellation [38] is adopted for LEOSs which has $n_o = 6$ orbits and each orbit has $n_s = 11$ satellites. Moreover, we assume there are $M = 1000$ MTs randomly distributed on the Earth and their popularity distributions are generated according to a Zipf-like distribution. Without loss of generality, the MTs are served by their nearest LEOSs and the performance of a generic LEOS is analyzed in this section. Because the mapping allocation problem has not been addressed in the open literature, we follow similar setups to those in [25], [32], [35] to set the parameters of Zipf distribution, user request arrival process and UCB policy, respectively. To reflect the sparsity of popular mappings, the weight control parameter of the caching cost is set to ensure that there are 200 popular mappings, while the maximum number of cached mappings in each LEOS should be lower than that of popular mappings. As for the caching gains, we assume that the inter-domain caching gain is twice as much as the intra-domain caching gain. For characterizing the impact of different algorithms on networks, we estimate the one-hop binding resolution delay of a generic request as $t = 2t_p + t_c$ [23], where t_p is the distance-based propagation delay and t_c represents the processing delay. According to [23], we set $t_c = 11.4ms$. The parameters adopted are presented in Table I.

For demonstrating the benefits of the sparse bandit learning based location management, the following caching strategies will be compared to SULC.

- We refer to the Classical UCB based MAB algorithm [35] as the *CUM algorithm*. By adopting the CUM algorithm, the LEOS acting as a player learns the caching reward distributions of the mappings and implements the UCB policy for all mappings for selecting γ_l mappings with optimal UCB indices.

TABLE I
LIST OF PARAMETERS

Total number of mappings	M=1000
Zipf exponent	δ_l=0.5
Shift of popularity distribution	σ_l=0
Maximum number of lookup requests in a LEOS	U_l=75
Density of Poisson distributed request arrival process	$\lambda = 5$
Number of popular mappings	$K = 200$
Weight control parameter for caching cost	$\beta = 0.05$
Maximum number of cached mappings in each LEOS	γ_l=100
UCB parameter	$\alpha = 1.5$
Caching gain for inter-domain content hits	$g_{inter} = 10$
Caching gain for intra-domain content hits	$g_{intra} = 5$
Processing delay of a binding request	$t_c = 11.4ms$

- The deep learning-enabled algorithm adopted in [28] is referred to as the *DeepHawkes algorithm*. Since DeepHawkes requires offline training, we collected the requests in 100000 time slots to train the DeepHawkes algorithm. The batch size and learning rate of DeepHawkes are set to 32 and 0.02, respectively. After 10 epochs's training, the contents with the highest estimated popularity will be cached.
- The *Least Recently Used (LRU) algorithm* of [39] monitors the time interval between the current and the previous requests of each mapping. The least recently used mapping will be replaced once a binding lookup request cannot be resolved locally.
- The *Least Frequently Used (LFU) algorithm* of [39] counts the content hit events for each cached mapping. Once a binding resolution request cannot be satisfied locally, the mapping having the minimum number of hits will be replaced.
- In the *Optimal solution (OPT)*, we assume that the popularity distributions of all mappings are known in advance. Each LEOS will always cache the optimal γ_l mappings, which have the highest expected caching rewards.

B. Performance Evaluation

Fig. 2 presents the average content hit rates in every 100 TSs for different algorithms. Here, we define the average content hit rate as the ratio of the number of events that a request is satisfied by the serving LEOS to the total number of binding resolution requests. A higher content hit rate indicates that more binding resolution requsts can be satisfied locally at a low lookup latency. As it can be observed in Fig. 2, DeepHawkes provides the highest content hit rate. This is because a large data set has been used to train the DeepHawkes algorithm for providing good estimate of content popularity, and the mappings having the highest popularity are selected without distinguishing the different caching rewards for intra-domain and inter-domain mappings. Besides, the proposed

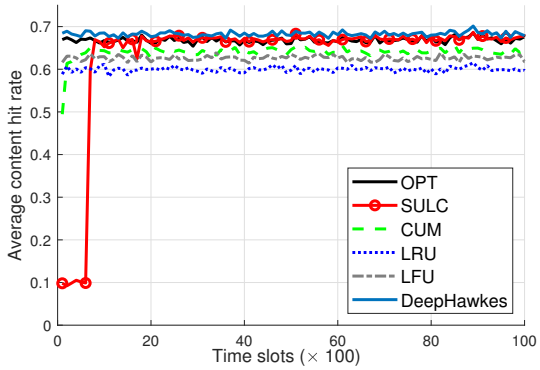


Fig. 2. Average content hit rates for different algorithms

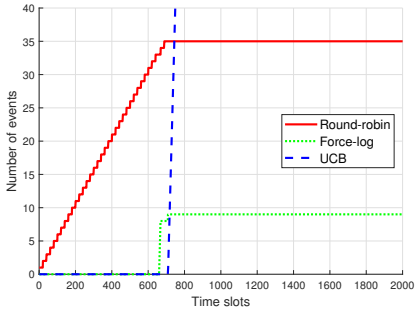


Fig. 3. Numbers of times for performing different phases in SULC

SULC outperforms the other solutions and approaches the optimal solution after a short period of learning. CUM has higher content hit rate than LFU, while LRU exhibits the worst performance. This is because the performance of a classical MAB algorithm will degrade with the number of arms [37], and it is inefficient to use an excessive number of arms. Thus, there is a gap between the optimal solution and CUM. Moreover, the observed popularity of mappings was not exploited in LRU, while the other algorithms take advantage of the historical binding lookup requests. Fig. 2 demonstrates the learning efficiency of the proposed SULC, which attains a near-optimal content hit rate after a short period of learning.

Fig. 2 shows that SULC provides low content hit rates at the beginning, while however increases dramatically after about 700 TSs. Since the proposed SULC consists of three different phases, to explore the learning procedure of SULC, Fig. 3 displays how many times the different phases are entered. It can be observed that SULC starts with a few Round-robin phases for making the popular mappings active. Then, the Force-log phase may be triggered for ensuring that the popular mappings are sufficiently sampled. After identifying the popular mappings, SULC will activate the UCB phases among these mappings.

Fig. 4 depicts the cumulative caching rewards for different algorithms, which quantity the benefits of mapping caching either in terms of energy saving or delay reduction. It can be observed that all the caching rewards tend to linearly increase with time, and SULC provides the closest performance to the optimal solution. In contrast to the existing algorithms, SULC provides poor rewards at the beginning. This is because

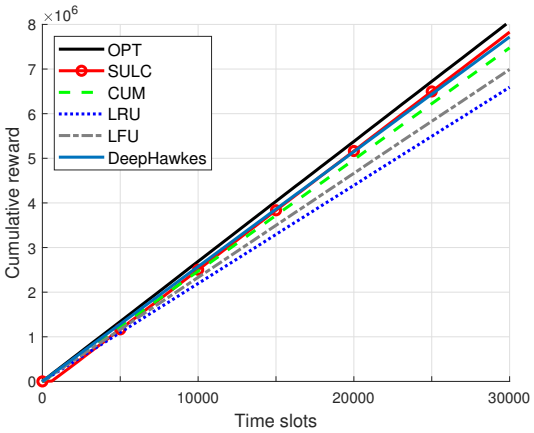


Fig. 4. Cumulative reward comparison of different algorithms

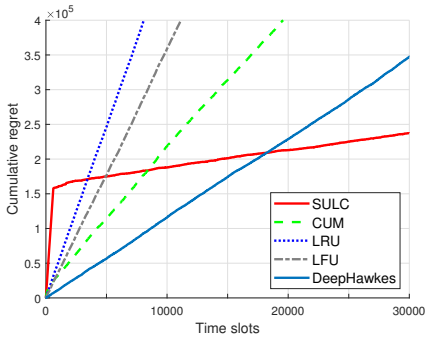


Fig. 5. Regret comparison of different algorithms

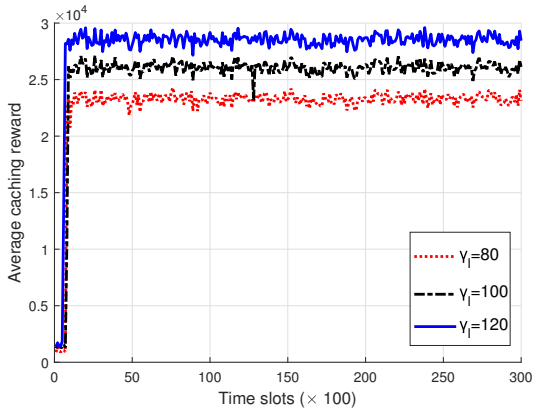


Fig. 6. Average caching reward of SULC with different γ_L

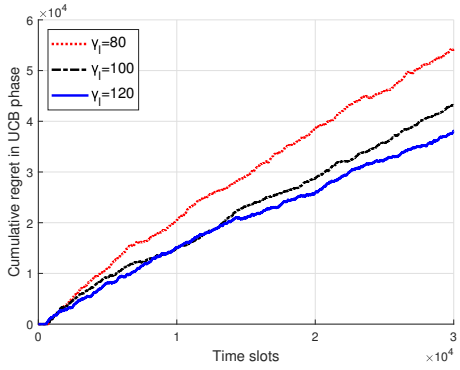


Fig. 7. Cumulative regret of SULC in UCB phase with different γ_L

SULC has to activate the Round-robin and Force-log phases for rendering the popular mappings active and sufficiently-sampled. The caching reward will be degraded, especially in the Round-robin phase, where all the mappings will be cached once. After the proposed learning algorithm converged, the gaps between SULC and the other algorithms will increase with time, since DeepHawkes ignores the different caching rewards for intra-domain and inter-domain contents and SULC attains higher content hit rates than the others.

In Fig. 5, we present the regrets of different algorithms, which represent the reward gaps against the OPT algorithm. We can observe from Fig. 5 that LRU has the worst performance, since it does not exploit the knowledge of the request frequency of mappings. By contrast, LFU exploits the historical lookup information, and caches the most frequent mappings, hence reduces the regret of LRU. However, the different caching rewards of intra-domain and inter-domain content hits are not considered in LFU, which imposes a linearly increased regret. By considering the inter-domain and intra-domain caching rewards, CUM provides much lower regret than LFU. Nonetheless, the CUM algorithm requires a long time for exploring all the mappings, and it converges slowly. After performing offline training, DeepHawkes provides the lowest regret at the beginning. However, its cumulative regret increases linearly due to the lack of distinguishing intra-domain and inter-domain contents. In contrast to the existing solutions, SULC aims for learning the unknown caching reward distributions and for exploiting the sparsity of popular mappings for speeding up the learning procedure. Therefore, SULC provides the lowest regret after a short period of learning as a benefit of identifying the popular mappings.

In Fig. 6, we compare the average caching rewards of SULC in every 100 TSs, when different cache sizes are considered. It can be observed that the average caching reward increases with γ_l. This is because a larger cache size will increase the content hit rates and result in higher caching rewards. Moreover, Fig. 6 demonstrates the efficiency of the proposed SULC, since it converges quickly for different γ_l.

To further explore the effect of the cache size on the performance of SULC in Fig. 7, we present the cumulative regret of SULC in the UCB phase (SULC-UCB) , since it has been demonstrated that SULC will converge in the UCB phases after performing a finite number of the Round-robin and Force log phases. Observe in Fig. 7 that the regret of SULC-UCB decreases with γ_l, because after identifying the K_l popular mappings, the regret is imposed by caching the mappings in $\mathcal{K}\backslash\Gamma$, which scales sub-linearly with $K_l - \gamma_l$.

In Fig. 8, we compare the regret of CUM and SULC-UCB when different number of mappings are considered. For CUM, a larger number of mappings result in increased rate-loss, since the classical bandit algorithm requires more time for exploring all the mappings and the performance of CUM degrades with M. In contrast to CUM, the increase

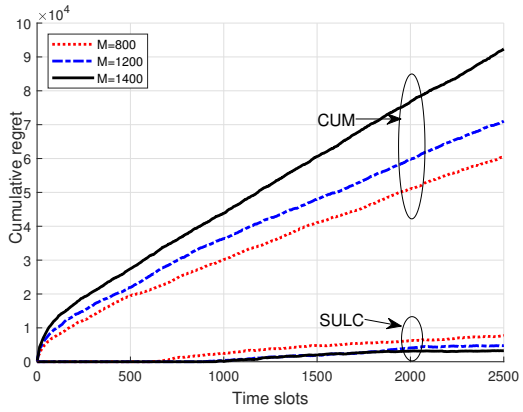


Fig. 8. Regret comparison of CUM and SULC-UCB with different M

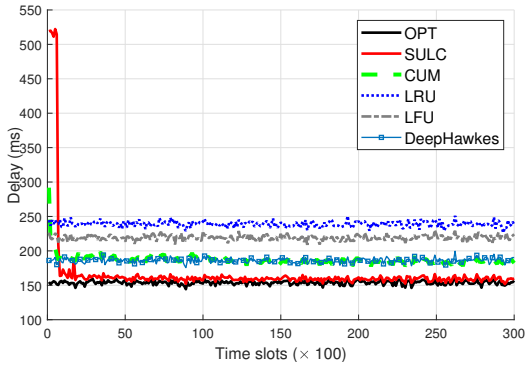
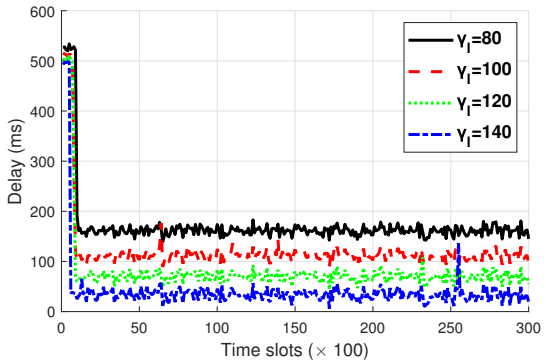
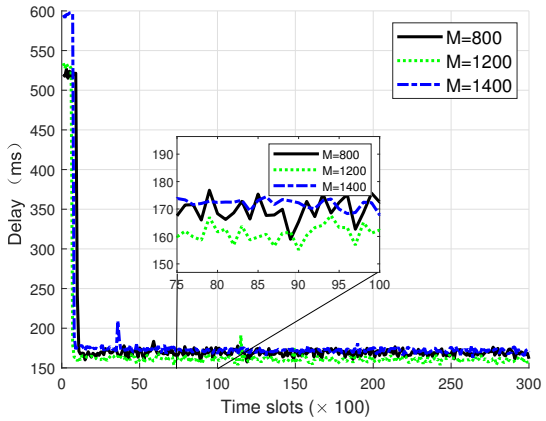


Fig. 9. Average binding resolution delay of different algorithms



(a) Binding resolution delay for different γ_l



(b) Binding resolution delay for different M

Fig. 10. Binding resolution delay comparison of SULC with different parameters

of the number of mapping has little effect on the regret of SULC-UCB. This is because the proposed SULC can promptly identify the popular mappings and apply the UCB policy for these popular mappings.

For characterizing the impact of caching algorithms on networks, we compare the average binding resolution delay of different algorithms in Fig. 9. It can be observed that SULC provides near-optimal performance after a short period of exploration. Since inter-domain binding resolution has a higher delay than intra-domain lookup, SULC tends to cache the popular inter-domain mappings for reducing the average resolution delay. By contrast, DeepHawkes caches the contents having the highest estimated popularity, resulting in longer average delay than SULC. Moreover, CUM provides similar performance to DeepHawkes, while LRU has the highest average binding resolution delay.

In Fig. 10, we analyze the impact of different network parameters on the average binding resolution delay of SULC. It can be readily observed from Fig. 10 (a) that the average binding resolution delay can be significantly reduced by increasing the cache size. This is because mapping lookup requests are more likely to be resolved locally when each LEO can cache more popular mappings. For different total numbers of mappings, Fig. 10 (b) shows that SULC converges promptly and the change of M has a low impact on the average binding resolution delay. Because the Zipf-like distribution of mapping popularity results in the sparsity of popular mappings, the majority of requests is concentrated on a few MTs. Moreover, SULC exploits the Round-robin and the Force-log phases to quickly identify the popular mappings and employs the UCB policy for caching the optimal mappings. Thus, SULC is not sensitive to the change of M.

VI. Conclusions

In this paper, we have studied the location management problems of SGINs from the perspective of ID-NA mapping allocation. By exploiting the distributed storage on board of GEOSs and LEOSs for caching part of the ID-NA mappings, we have proposed a multi-layer binding resolution system for providing prompt lookup response. By considering the limited storage of LEOSs, we formulated the mapping allocation procedure as a caching reward maximization problem. By leveraging the sparsity of the popular mappings, a sparse bandit learning based algorithm, namely SULC, has been proposed, which exploited the historical lookup information for estimating the expected caching rewards of mappings. Moreover, a sub-linear upper bound of the cumulative regret imposed by employing SULC has been provided for guaranteeing the convergence of the proposed algorithm. Extensive simulations have been conducted for demonstrating that the proposed algorithm succeeds in promptly identifying the popular mappings and attains near-optimal content hit rate.

Compared to the existed solution, we have shown that SULC provides the best binding resolution delay performance, while maintaining the lowest regret after a short period of learning.

APPENDIX

A. Proof of Lemma 1

Before providing the detailed proof of Lemma 1, we first present the following Lemma based on the Chernoff-Hoeffding bound [40] which will facilitate the derivation of the regret upper bound.

Lemma 4. *Upon assuming that the random variables $X_1, ..., X_n$ are the rewards obtained by pulling the same arm, all the random variables belong to [0,1] and have the same mean value μ. By letting $\bar{X}_n = \frac{\sum_{i=1}^{n} X_i}{n}$, for all $a \geq 0$, we have $\mathbb{P}\left\{\bar{X}_n - \mu \geq a\right\} \leq e^{-2na^2}$ and $\mathbb{P}\left\{\bar{X}_n - \mu \leq -a\right\} \leq e^{-2na^2}$ [32].*

Let us now consider the Round-robin phase. Since the mapping of each MT is cached once during the first $\left\lceil \frac{M}{\gamma_l} \right\rceil$ TSs, we have

$$\sum_{t=1}^{\left\lceil \frac{M}{\gamma_l} \right\rceil} B_{l,m}\left(t|\mathcal{R}\right) = 1, \forall m \in \mathcal{M}. \tag{35}$$

When $t \geq \left\lceil \frac{M}{\gamma_l} \right\rceil + 1$, the event $\{B_{l,m}\left(t|\mathcal{R}\right) = 1\}$ occurs only if the learning algorithm entered the Round-robin phase, which means $|A\left(\tau\right)| \leq \gamma_l$ for $\tau = t - \left\lceil \frac{m}{\gamma_l} \right\rceil$. In this case, some of the popular mappings are inactive. Thus, we have

$$\begin{aligned}&\mathcal{R} \subset \{|A\left(\tau\right)| \leq \gamma_l\} \\ &\subset \cup_{k=1}^{K_l} \left\{ \bar{r}_{l,k}\left(N_k\left(\tau\right)\right) \leq \sqrt{\frac{\alpha \log\left(N_k\left(\tau\right)\right)}{N_k\left(\tau\right)}} \middle| B_{l,m}(t) = 1, \mathcal{R} \right\}.\end{aligned} \tag{36}$$

The expectation term in (27) can be upper-bounded by

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} B_{l,m}\left(t|\mathcal{R}\right)\right] \leq 1 + \mathbb{E}\left[\sum_{t=\left\lceil \frac{M}{\gamma_l} \right\rceil + 1}^{\infty} B_{l,m}\left(t|\mathcal{R}\right)\right] \\ &\leq 1 + \sum_{k\in\mathcal{K}} \sum_{n=1}^{\infty} \mathbb{P}\left\{ \bar{r}_{l,k}\left(n\right) \leq \sqrt{\frac{\alpha \log\left(n\right)}{n}} \right\} \\ &= 1 + \sum_{k\in\mathcal{K}} \sum_{n=1}^{\infty} \mathbb{P}\left\{ \bar{r}_{l,k}\left(n\right) - \mu_{l,k} \leq \sqrt{\frac{\alpha \log\left(n\right)}{n}} - \mu_{l,k} \right\},\end{aligned} \tag{37}$$

where we replace $N_k(\tau)$ by n. When $\frac{log(n)}{n} \leq \frac{\mu_{l,k}^2}{4\alpha}$, we have $\sqrt{\frac{\alpha \log(n)}{n}} - \mu_{l,k} \leq -\frac{\mu_{l,k}}{2}$. One can readily check that there exists a minimum integer $\hat{n}_k$ and $\forall n \geq \hat{n}_k$, when we have $\frac{log(n)}{n} \leq \frac{\mu_{l,k}^2}{4\alpha}$. Thus, applying Lemma 4, (37) can be further upper-bounded by

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} B_{l,m}\left(t|\mathcal{R}\right)\right] \\ &\leq 1 + \sum_{k\in\mathcal{K}} \left(\hat{n}_k + \sum_{n=\hat{n}_k+1}^{\infty} \mathbb{P}\left\{ \bar{r}_{l,k}\left(n\right) - \mu_{l,k} \leq -\frac{\mu_{l,k}}{2} \right\} \right) \\ &\leq 1 + \sum_{k\in\mathcal{K}} \left(\hat{n}_k + \sum_{n=\hat{n}_k+1}^{\infty} e^{-\frac{\mu_{l,k}^2}{2} n} \right) \\ &\leq 1 + \sum_{k\in\mathcal{K}} \left(\hat{n}_k + \frac{2}{\mu_{l,k}^2} \right),\end{aligned} \tag{38}$$

and Lemma 1 is proved.

B. Proof of Lemma 2

Before providing the detailed proof of Lemma 2, we present a generic upper-bound of the expected cumulative regret. Recalling (21), the expected cumulative regret can be rewritten as

$$\begin{aligned}\mathbb{E}[Reg(T)] = &\sum_{i\in\Gamma} \mu_{l,i} \left(T - \mathbb{E}[N_i(T+1)]\right) \\ &- \sum_{m\in\mathcal{M}\setminus\Gamma} \mu_{l,m}\, \mathbb{E}[N_m(T+1)].\end{aligned} \tag{39}$$

Since $\mu_{l,i} \leq \mu_{l,1}$, we have

$$\begin{aligned}&\mathbb{E}[Reg(T)] \leq \mu_{l,1} \left(\gamma_l T - \sum_{i\in\Gamma} \mathbb{E}[N_i(T+1)] \right) \\ &- \sum_{m\in\mathcal{M}\setminus\Gamma} \mu_{l,m}\, \mathbb{E}[N_m(T+1)] \\ &\leq \sum_{m\in\mathcal{M}\setminus\Gamma} \mathbb{E}[N_m(T+1)] \Delta_{1,m},\end{aligned} \tag{40}$$

where $\Delta_{1,m} = \mu_{l,1} - \mu_{l,m}$.

For the Force-log phase, we decompose the regret into two parts imposed by the popular and unpopular mappings, respectively. Since the learning algorithm starts with a Round-robin phase, $\sum_{t=1}^{\left\lceil \frac{M}{\gamma_l} \right\rceil} B_{l,m}(t|\mathcal{F}) = 0$ and $Reg_{\mathcal{F}}(T)$ can be upper-bounded by

$$\begin{aligned}Reg_{\mathcal{F}}(T) \leq &\underbrace{\sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k}\, \mathbb{E}\left[\sum_{t=\left\lceil \frac{M}{\gamma_l} \right\rceil}^{T} B_{l,k}\left(t|\mathcal{F}\right)\right]}_{\mathcal{F}_1} \\ &+ \underbrace{\sum_{m\in\mathcal{M}\setminus\mathcal{K}} \Delta_{1,m}\, \mathbb{E}\left[\sum_{t=\left\lceil \frac{M}{\gamma_l} \right\rceil}^{T} B_{l,m}\left(t|\mathcal{F}\right)\right]}_{\mathcal{F}_2}.\end{aligned} \tag{41}$$

According to the definition of the Force-log phase, the mapping of MT m is cached, which implies that $\bar{r}_{l,m}(N_m(t)) \leq \sqrt{\frac{\alpha \log(t)}{N_m(t)}}$. Thus, $\mathcal{F}_1$ can be upper-bounded by

$$\mathcal{F}_1 \leq \sum_{k\in\mathcal{K}\setminus\Gamma} \Delta_{1,k} \sum_{t=\left\lceil \frac{M}{\gamma_l} \right\rceil}^{T} \mathbb{P}\left\{ \bar{r}_{l,k}(N_k(t)) \leq \sqrt{\frac{\alpha \log(t)}{N_k(t)}} \right\}. \tag{42}$$

Upon leveraging $t \leq T$, we have

$$\mathcal{F}_1 \leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{\infty} \mathbb{P}\left\{\bar{r}_{l,k}(N_k(t)) \leq \sqrt{\frac{\alpha\log(T)}{N_k(t)}}\right\}$$
$$\leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{n=1}^{\infty} \mathbb{P}\left\{\bar{r}_{l,k}(n) - \mu_{l,k} \leq \sqrt{\frac{\alpha\log(T)}{n}} - \mu_{l,k}\right\}. \tag{43}$$

Since $\sqrt{\frac{\alpha\log(T)}{n}}$ decreases with n, when $n \geq \left\lceil \frac{4\alpha\log(T)}{\mu_{l,m}^2}\right\rceil$, we have $\sqrt{\frac{\alpha\log(T)}{n}} - \mu_{l,m} \leq -\frac{\mu_{l,m}}{2}$. We further upper-bound $\mathcal{F}_1$ by

$$\mathcal{F}_1 \leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k}\left(\frac{4\alpha\log(T)}{\mu_{l,k}^2} + \sum_{n=1}^{\infty}\mathbb{P}\left\{\bar{r}_{l,k}(n) - \mu_{l,k} \leq -\frac{\mu_{l,k}}{2}\right\}\right)$$
$$\leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k}\left(\frac{4\alpha\log(T)}{\mu_{l,k}^2} + \frac{2}{\mu_{l,k}^2}\right). \tag{44}$$

When the mapping of an unpopular MT is cached in the Force-log phase, the MT should be active. Hence, $\mathcal{F}_2$ can be upper-bounded by

$$\mathcal{F}_2 \leq \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m} \times \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} \mathbb{P}\left\{\bar{r}_{l,m}\left(N_m(t)\right) \geq \sqrt{\frac{\alpha\log(N_m(t))}{N_m(t)}}\right\}. \tag{45}$$

By exploiting the fact that $\mu_{l,m} \leq 0$ for all $m \in \mathcal{M}\backslash\mathcal{K}$, we have

$$\mathcal{F}_2 \leq \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m} \sum_{n=1}^{\infty} \mathbb{P}\left\{\bar{r}_{l,m}\left(n\right) - \mu_{l,m} \geq \sqrt{\frac{\alpha\log(n)}{n}}\right\}$$
$$\leq \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m} \sum_{n=1}^{\infty} \frac{1}{n^{2\alpha/\ln(10)}}$$
$$\leq 2\sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m}. \tag{46}$$

By substituting the upper bounds of $\mathcal{F}_1$ and $\mathcal{F}_2$ into (41), (31) is obtained.

C. Proof of Lemma 3

Following similar steps to those in the proof of Lemma 2, the cumulative regret in the UCB phase can be upper-bounded by

$$Reg_{\mathcal{U}}(T) \leq \underbrace{\sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k}\,\mathbb{E}\left[\sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} B_{l,k}\left(t|\mathcal{U}\right)\right]}_{\mathcal{U}_1}$$
$$+ \underbrace{\sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m}\,\mathbb{E}\left[\sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} B_{l,m}\left(t|\mathcal{U}\right)\right]}_{\mathcal{U}_2}. \tag{47}$$

Here, $\mathcal{U}_1$ and $\mathcal{U}_2$ represent the upper bounds of the regrets for caching the popular and unpopular mappings respectively.

The mapping of $m \in \mathcal{K}\backslash\Gamma$ is cached only if its estimated mean reward is higher than that of a MT in Γ. Hence, we further upper-bound $\mathcal{U}_1$ by

$$\mathcal{U}_1 \leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} \mathbb{P}\left\{\bar{r}_{l,k}(N_k(t)) + \sqrt{\frac{\alpha\log t}{N_k(t)}} \geq \min_{i\in\Gamma}\left(\bar{r}_{l,i}(N_i(t)) + \sqrt{\frac{\alpha\log t}{N_i(t)}}\right)\right\}$$
$$\leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} \sum_{i\in\Gamma} \mathbb{P}_{k,i}(t), \tag{48}$$

where $\mathbb{P}_{k,i}(t)$ can be expressed as

$$\mathbb{P}_{k,i}(t) = \mathbb{P}\left\{\bar{r}_{l,k}(N_k(t)) + \sqrt{\frac{\alpha\log t}{N_k(t)}} \geq \bar{r}_{l,i}(N_i(t)) + \sqrt{\frac{\alpha\log t}{N_i(t)}}\right\}$$
$$= \mathbb{P}\left\{\bar{r}_{l,k}(N_k(t)) - \mu_{l,k} \geq \frac{\Delta_{i,k}}{2} + \frac{\Delta_{i,k}}{2} - \sqrt{\frac{\alpha\log t}{N_m(t)}} + \bar{r}_{l,i}(N_i(t)) - \mu_{l,i} + \sqrt{\frac{\alpha\log t}{N_i(t)}}\right\}. \tag{49}$$

Here, $\Delta_{i,k} = \mu_{l,i} - \mu_{l,k}$. It can be observed that $\frac{\Delta_{i,k}}{2} - \sqrt{\frac{\alpha\log t}{N_k(t)}} \geq 0$ if $N_k(t) \geq \left\lceil \frac{4\alpha\log(t)}{\Delta_{i,k}^2}\right\rceil$. Moreover, one can readily check that

$$\mathbb{P}\{x \geq a+b\} = \mathbb{P}\{x \geq a+b|b \geq 0\}\mathbb{P}\{b \geq 0\} + \mathbb{P}\{x \geq a+b|b \leq 0\}\mathbb{P}\{b \leq 0\}. \tag{50}$$

Since $0 \leq \mathbb{P}\{b \geq 0\} \leq 1$, the first term on the left side of (50) can be upper-bounded by

$$\mathbb{P}\{x \geq a+b|b \geq 0\}\mathbb{P}\{b \geq 0\} \leq \mathbb{P}\{x \geq a+b|b \geq 0\} \leq \mathbb{P}\{x \geq a\}. \tag{51}$$

Similarly, $0 \leq \mathbb{P}\{x \geq a+b|b \leq 0\} \leq 1$, the second term on the left side of (50) can be upper-bounded by

$$\mathbb{P}\{x \geq a+b|b \leq 0\}\mathbb{P}\{b \leq 0\} \leq \mathbb{P}\{b \leq 0\}. \tag{52}$$

Combining (51) and (52), we have

$$\mathbb{P}\{x \geq a+b\} \leq \mathbb{P}\{x \geq a\} + \mathbb{P}\{b \leq 0\}. \tag{53}$$

Then $\mathcal{U}_1$ can be further bounded by

$$\mathcal{U}_1 \leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{i\in\Gamma}\left(\frac{4\alpha\log(t)}{\Delta_{i,k}^2} + \sum_{n\geq\left\lceil \frac{4\alpha\log(t)}{\Delta_{i,k}^2}\right\rceil}^{\infty} \mathbb{P}\left\{\bar{r}_{l,k}(n) - \mu_{l,k} \geq \frac{\Delta_{i,k}}{2}\right\} + \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} \mathbb{P}\left\{\bar{r}_{l,i}(N_i(t)) - \mu_{l,i} \leq -\sqrt{\frac{\alpha\log t}{N_i(t)}}\right\}\right). \tag{54}$$

By leveraging $t \leq T$ and Lemma 4, we have

$$\mathcal{U}_1 \leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{i\in\Gamma} \left(\frac{4\alpha\log(T)}{\Delta_{i,k}^2} + \sum_{n\geq\left\lceil \frac{4\alpha\log(t)}{\Delta_{i,k}^2}\right\rceil}^{\infty} e^{-n\frac{\Delta_{i,k}^2}{2}} + \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} t^{-2\alpha/\ln 10} \right)$$
$$\leq \sum_{k\in\mathcal{K}\backslash\Gamma} \Delta_{1,k} \sum_{i\in\Gamma} \left(\frac{4\alpha\log(T)}{\Delta_{i,k}^2} + \frac{2}{\Delta_{i,k}^2} + 2 \right). \quad (55)$$

For $m \in \mathcal{M}\backslash\mathcal{K}$, its mapping is cached only if $\bar{r}_{l,m}(N_m(t)) \geq \sqrt{\frac{\alpha\log(t)}{N_m(t)}}$. Then $\mathcal{U}_2$ can be upper-bounded by

$$\mathcal{U}_2 \leq \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m} \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} \mathbb{P}\left\{ \bar{r}_{l,m}(N_m(t)) \geq \sqrt{\frac{\alpha\log(t)}{N_m(t)}} \right\}$$
$$\leq \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m} \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} \mathbb{P}\left\{ \bar{r}_{l,m}(N_m(t)) - \mu_{l,m} \geq \sqrt{\frac{\alpha\log(t)}{N_m(t)}} \right\}$$
$$\leq \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m} \sum_{t=\left\lceil \frac{M}{\gamma_l}\right\rceil}^{T} t^{-2\alpha/\ln 10} \leq 2 \sum_{m\in\mathcal{M}\backslash\mathcal{K}} \Delta_{1,m}. \quad (56)$$

Upon combining the upper-bounds of $\mathcal{U}_1$ and $\mathcal{U}_2$, Lemma 3 is proved.

REFERENCES

[1] H. D. Le, P. V. Trinh, T. V. Pham, D. R. Kolev, A. Carrasco-Casado, T. Kubo-Oka, M. Toyoshima, and A. T. Pham, "Throughput analysis for TCP over the FSO-based satellite-assisted Internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1875–1890, 2022.

[2] H. Yao, L. Wang, X. Wang, L. Zhou, and Y. Liu, "The space-terrestrial integrated network (STIN): An overview," *IEEE Commun. Mag.*, vol. PP, no. 99, pp. 2–9, 2018.

[3] Kawamoto, Y., Nishiyama, H., Kato, N., Kadowaki, and N., "A traffic distribution technique to minimize packet delivery delay in multilayered satellite networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3315–3324, 2013.

[4] S. Ji, M. Sheng, D. Zhou, W. Bai, Q. Cao, and J. Li, "Flexible and distributed mobility management for integrated terrestrial-satellite networks: Challenges, architectures, and approaches," *IEEE Network*, vol. 35, no. 4, pp. 73–81, 2021.

[5] T. Darwish, G. K. Kurt, H. Yanikomeroglu, G. Lamontagne, and M. Bellemare, "Location Management in Internet Protocol-Based Future LEO Satellite Networks: A Review," *IEEE Open J. .Commun. Soc.*, vol. 3, pp. 1035–1062, 2022.

[6] S. Fu, M. Atiquzzaman, L. Ma, and Y. J. Lee, "Signaling cost and performance of SIGMA: A seamless handover scheme for data networks," *Wireless Commun. Mobile Comput.*, vol. 5, no. 7, pp. 825–845, 2010.

[7] H. Tsunoda, K. Ohta, N. Kato, and Y. Nemoto, "Supporting IP/LEO satellite networks by handover-independent IP mobility management," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, pp. 300–307, 2004.

[8] X. Zhang, K. Shi, S. Zhang, D. Li, and R. Xia, "Virtual agent clustering based mobility management over the satellite networks," *IEEE Access*, vol. 7, pp. 89 544–89 555, 2019.

[9] P. Xie, Q. Wang, and J. Chen, "A survey of mobility management for mobile networks supporting LEO satellite access," in *IEEE/CIC Int. Conf. Commun. China, ICCC Workshops*, 2022, pp. 59–64.

[10] K. Leung, D. Shell, W. D. Ivancic, D. H. Stewart, T. L. Bell, and B. A. Kachmar, "Application of mobile-IP to space and aeronautical networks," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 16, no. 12, pp. 13–18, 2001.

[11] A. Z. M. Shahriar, M. Atiquzzaman, and S. Rahman, "Mobility management protocols for next-generation all-IP satellite networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 2, pp. 46–54, 2008.

[12] Z. Zhang, B. Zhao, Z. Feng, W. Yu, and C. Wu, "Design overview of a rapid mapping resolution system for enabling identifier/location separation in satellite network," in *Int. Conf. Optical Commun. Networks (ICOCN)*, 2017, pp. 1–3.

[13] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 343–352, 2004.

[14] D. Han, A. Anand, F. Dogar, B. Li, H. Lim, M. Machado, A. Mukundan, W. Wu, A. Akella, D. G. Andersen, J. W. Byers, S. Seshan, and P. Steenkiste, "XIA: Efficient support for evolvable internetworking," in *Proc. NSDI: USENIX Symp. Networked Syst. Des. Implement (NSDI)*, no. 309-322, 2012, pp. 1–6.

[15] P. Nikander, A. Gurtov, and T. R. Henderson, "Host identity protocol (HIP): Connectivity, mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 186–204, 2010.

[16] A. Venkataramani, J. F. Kurose, S. Banerjee, D. Raychaudhuri, K. Nagaraja, and Z. M. Mao, "Mobilityfirst: A mobility-centric and trustworthy Internet architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 74–80, 2014.

[17] Y. Fang and Y.-B. Lin, "Mobility management and signaling traffic analysis for multi-tier wireless mobile networks," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1843–1853, 2005.

[18] K. Gao, C. Xu, P. Zhang, J. Qin, L. Zhong, and G.-M. Muntean, "GCH-MV: Game-enhanced compensation handover scheme for multipath TCP in 6G software defined vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16 142–16 154, 2020.

[19] Z. Zhang, B. Zhao, W. Yu, and C. Wu, "Supporting location/identity separation in mobility-enhanced satellite networks by virtual attachment point," *Pervasive Mob. Comput.*, vol. 42, pp. 1–14, 2017.

[20] B. Feng, H. Zhou, G. Li, H. Li, and S. Yu, "SAT-GRD: An ID/Loc split network architecture interconnecting satellite and ground networks," in *IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–6.

[21] W. Han, B. Wang, Z. Feng, B. Zhao, W. Yu, and Z. Tang, "GRIMM: A locator/identifier split-based mobility management architecture for LEO satellite network," in *Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, 2016, pp. 605–608.

[22] M. Liu, H. Wang, H. Zhou, and Y. Xiang, "A mobility management method for space-earth integration network based on identity mapping system," in *IEEE World Conf. Comput. Commun. Technol. (WCCCT)*, 2021, pp. 52–57.

[23] H. Ji, H. Wang, D. Yang, Y. Xiang, and X. Feng, "A distributed identifier mapping resolving system for space-air-ground integrated network," in *IEEE Adv. Inf. Tech. Electron. and Autom. Control Conf. (IAEAC)*, vol. 5, 2021, pp. 2667–2671.

[24] B. Feng, H. Zhou, H. Zhang, G. Li, H. Li, S. Yu, and H. Chao, "HetNet: A flexible architecture for heterogeneous satellite-terrestrial networks," *IEEE Netw.*, vol. 31, no. 6, pp. 86–92, 2017.

[25] X. Zhu, C. Jiang, L. Kuang, and Z. Zhao, "Cooperative multilayer edge caching in integrated satellite-terrestrial networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2924–2937, 2022.

[26] Z. Ji, S. Wu, C. Jiang, and W. Wang, "Popularity-driven content placement and multi-hop delivery for terrestrial-satellite networks," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2574–2578, 2020.

[27] E. Wang, H. Li, and S. Zhang, "Load balancing based on cache resource allocation in satellite networks," *IEEE Access*, vol. 7, pp. 56 864–56 879, 2019.

[28] L. Liu, Y. Li, Y. Xu, Q. Zhang, and Z. Yang, "Deep learning-enabled file popularity-aware caching replacement for satellite-integrated content-centric networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4551–4565, 2022.

[29] M. He, C. Zhou, H. Wu, and X. Sherman Shen, "Learning-based cache placement and content delivery for satellite-terrestrial integrated networks," in *IEEE Glob. Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

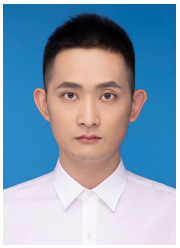
[30] D. Han, W. Liao, H. Peng, H. Wu, W. Wu, and X. Shen, "Joint cache placement and cooperative multicast beamforming in integrated satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3131–3143, 2022.

[31] V. Devarapalli, K. Chowdhury, S. Gundavelli, B. Patil, and K. Leung, "Proxy mobile IPv6," 2008.

[32] J. Song, M. Sheng, T. Q. S. Quek, C. Xu, and X. Wang, "Learning-based content caching and sharing for wireless networks," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4309–4324, 2017.

[33] R. M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations*, 1972.

[34] A. Chattopadhyay, B. Błaszczyszyn, and H. P. Keeler, "Gibbsian on-line distributed content caching strategy for cellular networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 969–981, 2018.
[35] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach learn*, vol. 47, no. 2-3, pp. 235–256, 2002.
[36] S. Vakili, K. Liu, and Q. Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 759–767, 2013.
[37] J. Kwon, V. Perchet, and C. Vernade, "Sparse stochastic bandits," in *Conf. Learn. Theory (COLT)*, 2017, pp. 1–24.
[38] H. Nishiyama, Y. Tada, N. Kato, N. Yoshimura, M. Toyoshima, and N. Kadowaki, "Toward optimized traffic distribution for efficient network capacity utilization in two-layered satellite networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 3, pp. 1303–1313, 2013.
[39] D. Lee, J. Choi, J.-H. Kim, S. Noh, S. L. Min, Y. Cho, and C. S. Kim, "LRFU: a spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE Trans. Comput.*, vol. 50, no. 12, pp. 1352–1361, 2001.
[40] B. Doerr, *Analyzing Randomized Search Heuristics: Tools from Probability Theory*. Theory Of Randomized Search Heuristics:Foundations and Recent Developments, 2014.



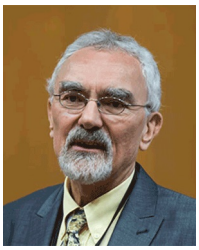
Huasen He (Member, IEEE) received the B.S. degree in automation from the University of Science and Technology of China, Hefei, China, in 2013, and the M.S. degree in signal processing and communications and the Ph.D. degree in digital communications from the University of Edinburgh, Edinburgh, U.K., in 2014 and 2018, respectively. He is currently an Associate Research Fellow with the School of Information Science and Technology, USTC. His research interests include future networks, network modeling and optimization.

Changkun Qin received the B.S. from Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently pursuing the M.Eng in the Institute of Advanced Technology from the University of Science and Technology of China. His research interests include satellite networks and mobility management.

Shuangwu Chen received the B.S. and Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2011 and 2016, respectively. He is currently an associate professor in the School of Information Science and Technology, USTC. His research interests include multimedia communications, future network and stochastic optimization.

Xiaofeng Jiang (Member, IEEE) received the B.E. and Ph.D. in information science and technology from University of Science and Technology of China (USTC), Hefei, China, in 2008 and 2013. He is currently an associate professor in the School of Information Science and Technology, USTC. His recent research interests include discrete event dynamic system, tensor analysis and big data, future network and cognitive communications.

Jian Yang (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2001 and 2006, respectively. He is currently a professor in the School of Information Science and Technology, USTC. His research interests include future network, distributed system design, modeling and optimization, multimedia over wired and wireless and stochastic optimization. Dr. Yang received Lu Jia-Xi Young Talent Award from Chinese Academy of Sciences in 2009.



Lajos Hanzo (Life Fellow, IEEE) received the master's and doctorate degrees from the Technical University (TU) of Budapest, in 1976 and 1983, respectively, the D.Sc. degree from the University of Southampton, in 2004, and the joint Honorary Doctorate degree from the TU, in 2009 and the University of Edinburgh, in 2015. He is currently a Foreign Member of the Hungarian Academy of Sciences and the Former Editor-in-Chief of the IEEE Press. He has served several terms as a Governor of both the IEEE ComSoc and VTS. He has published more than 2000 contributions at IEEE Xplore, 19 Wiley-IEEE Press books, and has helped the fast-track career of 123 Ph.D. students. Over 40 of them are a professors at various stages of their careers in academia and many of them are a leading scientists in the wireless industry. He is also a fellow of the Royal Academy of Engineering (FREng), the IET, and EURASIP. He was a recipient of the 2022 Eric Sumner Field Award.