

# QoE-Aware Efficient Content Distribution Scheme For Satellite-Terrestrial Networks

Dingde Jiang<sup>ID</sup>, Member, IEEE, Feng Wang<sup>ID</sup>, Zhihan Lv<sup>ID</sup>, Shahid Mumtaz<sup>ID</sup>, Senior Member, IEEE, Saba Al-Rubaye<sup>ID</sup>, Senior Member, IEEE, Antonios Tsourdos<sup>ID</sup>, and Octavia Dobre<sup>ID</sup>, Fellow, IEEE

**Abstract**—The satellite-terrestrial networks (STN) utilize the spacious coverage and low transmission latency of the Low Earth Orbit (LEO) constellation to transfer requested content for subscribers especially in remote areas. With the development of storage and computing capacity of satellite onboard equipment, it is considered promising to leverage in-network caching technology on STN to improve content distribution efficiency. However, traditional caching and distribution schemes are not suitable in STN, considering dynamic satellite propagation links and time-varying topology. More specifically, the unevenness of user distribution heightens difficulties for assurance of user quality of experience. To address these problems, we first propose a density-based network division algorithm. The STN is divided into a series of blocks with different sizes to amortize the data delivery costs. To deploy the caching satellites, we analyze the link connectivity and propose an approximate minimum coverage vertex set algorithm. Then, a novel cache node selection algorithm is designed for optimal subscriber matching. On the basis of time-varying network model, the STN cache content updating mechanism is derived to enable a stable and sustainable quality of user experience. The simulation results demonstrate that the proposed user-oriented STN content distribution scheme can obviously reduce the average propagation delay and network load under different network conditions and has better stability and self-adaptability under continuous time variation.

**Index Terms**—Satellite-terrestrial networks, content distribution, in-network caching, quality of experience, time-varying network

## 1 INTRODUCTION

THE terrestrial networks have been cultivated in depth [1]. However, terrestrial networks mostly cover the crowded area of human life [2], [3]. A study from UNESCO shows that about half the world region could not access the Internet by the end of 2020 [4]. For instance, some rural or disaster-stricken areas lack enough broadband infrastructures because of high construction and maintenance costs [5]. For the growing content service demands in these regions, novel network architectures are expected [6], [7]. A reliable solution is the construction of satellite-terrestrial networks (STN) [8]. The STN can utilize the spacious coverage and low transmission latency of the Low Earth Orbit (LEO) constellation to transfer requested content anytime and anywhere for regions with limited or no broadband Internet service [9]. The capital and

operational expenditures of the LEO constellation are more acceptable than the costs of base stations in remote areas [10].

Typically, the digital broadcasting of the LEO satellite can facilitate more satellite-Internet-of-things (SloTs) applications [11]. The inter-satellite links (ISLs) form the space backbone that connects subscribers to the content originator [12]. However, when massive subscribers in a region issue similar content query requests, repetitive end-to-end ISL paths will incur unbalanced load pressure and congestion in STN [13]. This is a waste of onboard transmission resources. The quality of experience (QoE) of the user is also affected. Besides, the motion of STN segments results in the time-varying of ISLs and network topology [14]. The traditional delay tolerance network (DTN) utilizes a storage-forward framework to transmit content. Novel spatial routing technologies, using time-varying links, may aggravate packet loss in long-distance content delivery [15]. These excessively repeated transport modes with long end-to-end ISLs bring unbearable latency and overhead. As the number of STN subscribers grows, redundant content delivery needs to be addressed.

With the development of storage and computing capacity of satellite onboard equipment, it is considered promising to leverage satellite caching in STN to improve content delivery efficiency [16], [17]. Instead of establishing end-to-end ISLs, placing the specified contents on LEO cache satellites will reduce delivery latency and minimizes the required bandwidth in STN [18]. In this direction, the wide coverage and native broadcasting of LEO satellite networks can be utilized for content distribution without additional cost [19]. Two kinds of contents are more suitable to be cached by LEO satellites. The first is regional alarm information (RAI), like disaster alerts and pre-warning messages. The RAIs need to be

- Dingde Jiang and Feng Wang are with the University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China.  
E-mail: jiangdd@uestc.edu.cn, uestc\_wf@126.com.
- Zhihan Lv is with the School of Data Science and Software Engineering, Qingdao University, Qingdao, Shandong 266071, China.  
E-mail: lzhihan@gmail.com.
- Shahid Mumtaz is with the Instituto de Telecomunicações, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal. E-mail: smumtaz@av.it.pt.
- Saba Al-Rubaye and Antonios Tsourdos are with the School of Aerospace Transport and Manufacturing, Cranfield University, MK43 0AL Bedford, U.K. E-mail: {s.alrubaye, a.tsourdos}@cranfield.ac.uk.
- Octavia Dobre is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada.  
E-mail: odobre@mun.ca.

Manuscript received 5 October 2020; revised 15 April 2021; accepted 18 April 2021. Date of publication 22 April 2021; date of current version 5 December 2022.

(Corresponding authors: Dingde Jiang, Feng Wang, and Zhihan Lv.)  
Digital Object Identifier no. 10.1109/TMC.2021.3074917

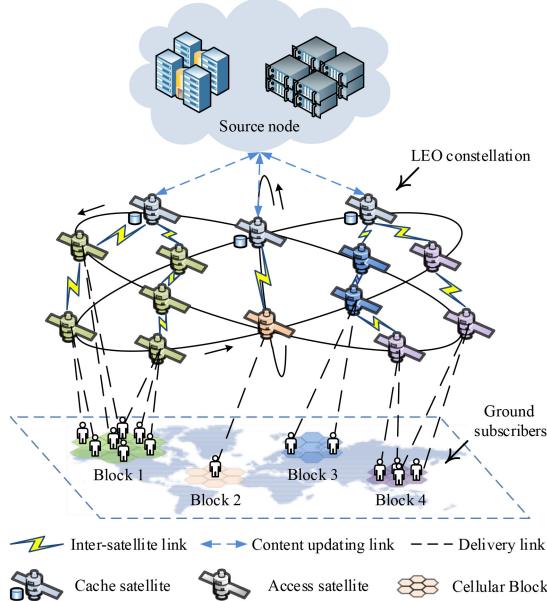


Fig. 1. In-network caching and content distribution framework for STN.

broadcast to involved users at a specific time. The second is intelligent application data (IAD), like high definition maps and files of common interests. The IADs are usually fixed or updated regularly. Both kinds are push-enabled considering the widespread and predictable requests in remote regions. Besides, they have more strict delay requirements. The QoE should be satisfactory for subscribers [20].

Different from terrestrial networks, the distribution process in STN turns into a multi-hop ISL delivery, from caching satellites to access satellites. If users have to send inquiries to a cache satellite far away, the time-sensitive content services will be meaningless. Some works investigated the content delivery with scattered cache satellites [21], [22]. But these studies do not take the distribution of subscribers into account, resulting in suboptimal schemes. The deployment of caching satellites must consider the sub-average and uneven requests to improve user QoE. Moreover, the dynamic topology of STN affects the design of cache node deployment. Some studies utilize the time-varying graph to describe the topology changes [23]. But the update of caching satellites and ISLs has not been analyzed. Therefore, how to design a QoE-aware cache satellite deployment scheme in the time-varying STN needs more study. To promote the user QoE in STN content distribution, reasonable network architecture, cache satellite deployment, and cache content updating should be considered jointly.

In this paper, we identify the key role of in-network caching in STN for subscribers in remote or disaster-stricken regions. The framework is displayed in Fig. 1. It is observed that different users access STN from different blocks. The most intuitive factor affecting user QoE is latency. Considering relatively large propagation delay from inquiry to fetch, the insight of a more user-friendly dynamic distribution solution is investigated. We first present a novel network hierarchy strategy. The entire STN network is divided into several blocks of different sizes according to ground user density. To deploy the caching satellites, we analyze the link usability and propose an approximate minimum coverage vertex set

algorithm. Then, a novel cache node selection algorithm is designed for optimal user block matching. Further, the time-varying STN model is established to determine the caching update rules, including the update frequency and ISL changing. Overall, the proposed in-network caching methods aim to achieve QoE-aware efficient content distribution in STN, jointly considering ground user distribution, cache node deployment, and network update. The main contributions of this paper are as follows:

- We propose an innovative density-based STN division (DBD) framework. The STN is divided into several blocks of different sizes according to subscriber density. The aim is to identify the dense and sparse STN regions to make the users in each block uniform. The cache satellite in each block provides subscribers with the specified content to achieve reasonable and efficient STN-based content distribution.
- In this paper, a complete cache node deployment process is designed. We first look for the minimum coverage vertex set in LEO satellite networks and prove that it is an NP-complete problem. Then, we analyze the link usability and propose an approximate minimum coverage vertex set (AMCV) algorithm. Finally, the cache node selection (CNS) algorithm is proposed for optimal block matching.
- We design the density-based time-varying graph (DTG) to realize efficient STN topology update. The updating rules are derived to enable a stable and sustainable user experience. Then, the density-based content distribution scheme (DCD) is orchestrated to complete the closed-loop autonomy. The continuity and stability of proposed QoE-aware STN content distribution are guaranteed.

The rest of this paper is organized as follows. In Section 2, we review state-of-the-art research on content distribution in STN. Section 3 constructs the scenario model. In Section 4, we propose the DBD, AMCV, and CNS algorithms to deploy cache satellites and access satellites in STN. In Section 5, the DTG is proposed to realize stable configuration transfer during topology updating. In Section 6, the algorithms are simulated and the analysis of results is presented. We conclude our work in Section 7.

## 2 RELATED WORK

Nowadays, researchers and system designers are gradually realizing that the satellite network layer and terrestrial network layer in STN are mutually complementary. User demand will affect the choreography and quality-of-service (QoS) of satellite networks all the time [6], [24]. For every ground user making rapid and efficient content retrieval, the distributed STN architecture is essential. Authors in [25] thought that the deployment of satellite gateway was a vital part of the design of STN architecture. They modified two heuristic algorithms and proposed a particle swarm-based optimization approximate algorithm (PSOA) to distribute satellite gateways around the world. However, the feasibility of deployment in remote areas was not considered. Deng *et al.* [26] proposed a multilevel clustering-based grouping method to divide LEO satellites and ground users into small

cells for quick matching. The purpose was to improve the utilization efficiency of STN resources and ensure QoS when massive users access. In [27], authors emphasized the importance of distributed data delivery in multi-satellite-ground networks. They investigated a data offloading model in STN and divided users into two categories for service provision to modify the backhaul link efficiency. Authors in [28] proposed a three-layer STN architecture to manage the satellite-user links in a distributed manner. Each layer was assigned to different functionalities for cooperative caching. Considering the difference between STN and ground network, authors in [23] established the time-varying STN model for centrality-balanced traffic management. In [29], the authors designed a distributed data transmission method in STN. All the requests are published to divided sub-regions. They selected several satellites as relay segments in each sub-region to establish the shortest data delivery path. Michele *et al.* [30] analyzed the STN networking mode and data transmission technology to modify the configuration of content delivery in STN. They learned from the architecture of ground content distribution network (GCDN) and tried to employ IP streaming cache service in time-varying STN.

STN routing is an important part of content delivery. LEO satellites are divided into source nodes, destination nodes, and relay nodes to jointly complete data transmission. Appropriate path planning enables the cost decreasing in switching calculation and routing overhead. Authors in [31] designed a routing scheme in the LEO satellite network layer with the minimum hops. To improve the network scalability, Sai *et al.* [32] proposed a network mapping method based on greedy-hyperbolic-geometry-algorithm to modify the relay satellite selection. However, these researches of satellite network traffic control need to consider the demand of ground subscribers for user-friendly data transmission. Therefore, the authors in [33] analyzed the end-to-end routing mechanism, using an elastic-heuristic optimization algorithm to select the appropriate path based on the business requirement. Zhang *et al.* [34] analyzed the constraints of satellite-user links and designed a dynamic-static alternative routing for backhaul link load optimization. To improve the adaptability of routing scheme, Kato *et al.* [35] utilized artificial intelligence to learn the features of STN traffic in each time slot. A deep-learning-based pathfinding scheme was proposed to modify STN routing performance. It is noted that the design of routing in STN gradually focuses on network stability and adaptability and user QoE, not only on the shortest paths.

With the development of storage and computing capacity of satellite onboard equipment, the effect of satellites as relay nodes can be extended. There have been studies to optimize the efficiency of content delivery using in-network caching in cloud-fog computing networks [19]. It is considered promising to leverage in-network caching in STN to improve content distribution efficiency [36]. Authors in [37] incorporated the wireless content caching into STN to expand the spectral coverage of in-network caching. Specific content was deployed in selected LEO satellites for broadcasting to improve content delivery performance. Garg *et al.* [38] analyzed the STN computing power when massive users accessed and proposed an action-coded deep deterministic policy gradient (AC-DDPG)

algorithm for the minimum content delivery delay. Similarly, authors in [21] proposed a content distribution strategy by optimizing the device access delay in STN. The aim of the model was to optimize the deployment of cache satellites. Wu *et al.* [20] considered that the caching contents need to be jointly placed in the ground stations and LEO satellites. A joint deployment strategy was proposed to optimize the backhaul bandwidth consumption. The cached file has to deal with the ISL change and routing strategy update during content delivery. Authors in [22] analyzed the link-switch consumption as content delivery by cache satellites. A time-varying STN cache node selection scheme was proposed to improve the network continuous service capability. Authors in [39] designed a testbed for STN in-network caching and content distribution. The secure multimedia data distribution framework was described with a hybrid STN topology and subscriber random accessing model.

However, the above studies only considered the dispersion of in-network caching in STN. The effect of uneven user distribution on content delivery was not mentioned. We note that the content distribution under the uneven environment is an important issue that deserves more attention. Therefore, we propose a density-based content distribution approach for STN to optimize data delivery efficiency.

### 3 SCENARIO FORMULATION

In this paper, we consider a content distribution scenario for ground subscribers. In STN, massive LEO satellites, denoted by  $\mathbb{S} = \{s_i | 1 \leq s_i \leq S\}$ , form the satellite network and achieve global coverage. The ground content center, as the source node  $gs_0$ , generates the encapsulated RAI and IAD information as content to be pushed. Based on a certain update frequency, the  $gs_0$  publishes subscribed content to  $\mathbb{S}$  for global spread, using the file cache ISLs. Note that it is impractical for all satellites to store subscribed contents. The dynamic topology will cause excessive delivery resources as content updating. Therefore,  $\mathbb{S}$  is divided into two parts for efficient content distribution. A certain proportion of  $\mathbb{S}$ , denoted by  $\mathbb{CS} = \{s_j | 1 \leq s_j \leq CS\}$ , is designed as caching satellites to store the backup of content file for convenient fetch by nearby satellites or users in coverage. The rest of  $\mathbb{S}$ , denoted by  $\mathbb{AS} = \{s_k | 1 \leq s_k \leq AS\}$ , is defined as access satellites to relay the content from the  $\mathbb{CS}$  to accessing users, denoted by  $\mathbb{U} = \{u_n | 1 \leq u_n \leq U\}$ . The STN consists of  $\mathbb{N} = \mathbb{S} + \mathbb{U}$  nodes. The distribution scenario is shown in Fig. 2. Each  $s_j \in \mathbb{CS}$  is surrounded by several  $s_k \in \mathbb{AS}$ , dividing massive satellites and covered users into several blocks. The subscriber  $u_n$  can utilize subordinate  $\mathbb{CS}$  and  $\mathbb{AS}$  to fetch the content file. If the desired content is not cached on the subordinate  $\mathbb{CS}$ , the  $u_n$  sends the inquiries to  $gs_0$  for content delivery via the ISL paths. Hence, the  $\mathbb{CS}$  set is regarded as the representative of  $gs_0$  in each block, achieving less propagation delay. The effects of resource waste due to redundant transmissions are minimized. In Fig. 2, the STN is provided with a clear, hierarchical division. The  $gs_0$  utilizes ISL paths to connect the  $\mathbb{CS}$  set for content caching and updating. For the management of  $\mathbb{CS}$  resources, the file cache paths and content update frequency need to be uniformly orchestrated. Each  $\mathbb{CS}$  distributes content to block users utilizing subordinate  $\mathbb{AS}$ . Since the satellite movement, it occurs  $\mathbb{CS}$

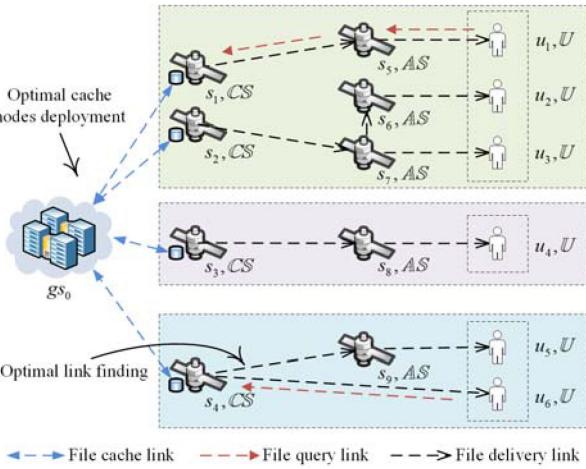


Fig. 2. System formulation of file distribution in STN.

and AS handover in each block when the assigned  $s_j \in \text{CS}$  moves out of the region. The time-varying schemes of cache updating and segment switching should be designed for service continuity. In the operation of the system, the focus of this paper is to optimally distribute content, as distance matching as possible to subscribers, regularly updating as appropriate, in order to enable the most scalable and efficient solution. The main problem is turned into how to divide the blocks and how to match AS and CS of each time slot. The user QoE will be improved under appropriate block division and node deployment. It should be noted that users are allowed to inquire the content directly when covered by CS. Besides, all the ground users are assumed stationary relative to satellites moving rapidly. In particular, the frequently utilized notations in this work are displayed in Table 1.

#### 4 CACHE SATELLITE SELECTION

The basic premise of efficient STN distribution is that subscribers of each block are expected to own cache satellites for content inquiry. The cache node set should reduce the negative impact of long propagation delays. This demand can be converted into a cache node set selection problem. In this section, we explore how to obtain the optimal cache node set. We first propose the QoE-aware STN hierarchy based on user density. Then, the STN link usability model is described to analyze the node connectivity. Based on these two aspects, the user-oriented cache node selection algorithm is proposed.

##### 4.1 Density-Based STN Division

The primary focus of STN content distribution is LEO network hierarchy. The CS and AS should be precisely selected from massive satellites. Previous literature, i.e., [21], [22], assumes that CS is evenly distributed when dividing up the blocks, and users in each cell have similar densities and requirements. These are not the cases in a real-world scenario. Obviously, the subscriber number is much higher on the ground than on the sea. Partitioning the nodes into a uniform grid would result in sizeable errors. Therefore, we improve the block partitioning method based on subscriber density. The aim is to identify the dense and sparse STN

TABLE 1  
Notations

Notation	Description
$s_i$	LEO satellite node
$u_n$	Ground user node
$gs_0$	Source node
$\mathbb{S}$	LEO satellite node set
$\mathbb{U}$	User node set
$b_i$	Subdivided block
$q$	Division point
$A_{re}$	Region area
$A_{\min}$	Minimum block area
$den(b_i)$	Density of block $b_i$
$\beta$	Division threshold
$T_{\text{LEO}}$	Satellite orbital period
$\mu$	Kepler constant
$T^S$	Remaining coverage time
$\circ(s_k)$	The degree of node $s_k$
$\alpha_O$	Geocentric angle
$\lambda$	Satellite longitude
$\varphi$	Satellite latitude
$\mathbb{T}$	Time slot set
$V_{\min}$	The minimum cover vertex set
$\mathbb{P}$	The path that contains sublinks
$D(e_i, t_i)$	The delay on link $e_i$ in time $t_i$
$s_{b_i}^{\tau}$	Time slot $\tau$ in block $b_i$
$s_{b_i}^{t_{\min}}$	The minimum time slot of $b_i$

regions to make the users in each block uniform. Considering the dynamic block division, there are two kinds of schemes. The first class, represented by GeoHash, iteratively divides the map into required cells. The second class, represented by Uber H3 and Google S2, divides the map into fine-grained subcells then aggregates similar subcells to form the desired division. Since generating massive initial subcells, the second class has more clustering randomness. For a more controllable division, the quadtree partition is modified to the DBD algorithm. The utility of DBD is controllable based on STN size and number of user requests. The modified DBD scheme is adaptive to provide a good trade-off between precision and efficiency.

Fig. 3a shows the traditional quadtree partitioning. The selected block is recursively divided into four equal sub-blocks until all the blocks meet the stop condition. The traditional method is efficient but data-independent. It may occur that a large number of subscribers are divided into a small block with only one cache satellite. This could lead to traffic imbalance. We improve the method to a data-dependent partitioning, as Fig. 3b shows. First, some initial division points are selected to divide the entire region. The subscriber density for each partitioned block is calculated. Blocks whose density exceed the threshold are selected as new regions to be divided. The process is repeated until all subblocks meet the stop condition. In the process, two conditions affecting the partition precision. The first condition is the selection of initial division points and the second is the stop condition.

*Initial division points:* the computational complexity and accuracy of the algorithm will be affected by the selection of initial division points directly. Therefore, we randomly generate division points  $\{q_i\}$  for the initial region.  $|\{q_i\}| = q$  and  $q$  is related to the region area and the minimum block area,

which is calculated as  $q = A_{re}/A_{min}$ . Then, we calculate the maximum density difference for each  $q_i$  partition:

$$\Delta den(B, q_i) = \max_{\forall b_i, b_j \in B} \{|den(b_i) - den(b_j)|\}, \quad (1)$$

where  $den(b_i)$  is the density of block  $b_i$ .  $B$  denotes the divided block set. Then, the largest  $\Delta den(B, q_i)$  is selected to compare with the threshold  $\beta$ .  $\beta$  is a constant to determine the division precision. If  $\Delta den(B, q_i)$  is greater than  $\beta$ ,  $q_i$  will be the next partition point. The quadtree partitioning is carried out on point  $q_i$ . The above process is repeated until no further subblocks can be divided. Otherwise, the current block will not be divided because the subscriber density is nearly uniform already. Especially, if each  $\Delta den(B, q_i)$  is equal, the current block is divided into  $q \times q$  subblocks with equal size to complete the partition.

### Algorithm 1. The DBD Algorithm

```

Input: STN node dataset  $\mathbb{S}, \mathbb{U}$ ;
Input: Initial block  $B$ ;
Output: Block division  $BD$ ;
1:  $block = B$ ;
2:  $BD = \emptyset$ ;
3:  $q = B/A_{min} // A_{min}$  has been preset;
4:  $u \leftarrow$  numbers of subscribers in  $block$ ;
5: if  $u < 1 || q \leq 1$  then
6:    $BD = BD \cup block$ ;
7: else
8:   Randomly generate division points  $q \in \{q_1, q_2, \dots, q_n\}$ ;
9:   for  $i = 1$  to  $n$  do // Select the optimal division point
10:    Divide the  $block$  by  $q_i$  with quadtree algorithm;
11:    for each subblock  $b \in block$  do
12:      Calculate the density  $den(b)$  of each subblock;
13:    end for
14:    Calculate  $\Delta den(B, q_i) = \max \{|den(b_i) - den(b_j)|\}$ ;
15:  end for
16:  if  $\Delta den(B, q_i) \leq \beta$  then
17:    divide  $block$  into  $q \times q$  subblocks  $\{b_i\}$ ;
18:     $BD = BD \cup \{b_i\}, 1 \leq i \leq n^2$ ;
19:  else
20:    Divide the  $block$  by  $q_i$  with quadtree algorithm;
21:    Obtain 4 subblocks  $B_{q_i} = \{b_1, b_2, b_3, b_4\}$ ;
22:    for each subblock  $b_i \in B_{q_i}$  do
23:      if  $A(b_i) \leq A_{min}$  then
24:         $BD = BD \cup \{b_i\}$ ;
25:      else
26:         $block = b_i$ ;
27:        Go back and execute Step 3;
28:      end if
29:    end for
30:  end if
31: end if
32 : return  $BD$ ;

```

**Stop condition:** The completeness of the stop condition has a great influence on the division precision and adaptivity of the algorithm. The traditional quadtree method determines the stop condition according to whether the minimum block area is reached. It is not efficient in the non-uniform division. Therefore, we set 3 stop conditions for the DBD algorithm:

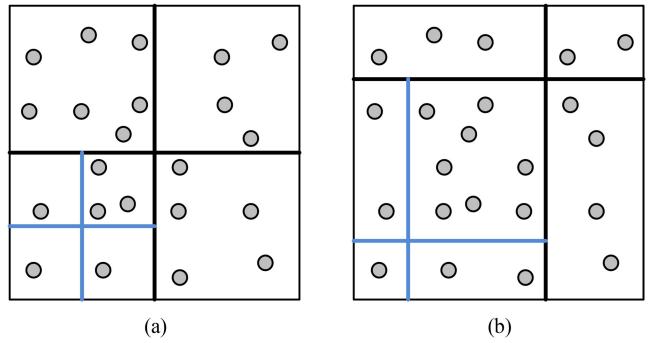


Fig. 3. Density-based quadtree division model.

- If the block to be partitioned has no subscribers, the partition algorithm is terminated. There is no value to divide an empty block. The block is labeled as the last leaf node.
- If the block to be partitioned is already uniform, the stop condition is triggered. There is no need to repeat the algorithm for this block. Divide the current block into  $q \times q$  subblocks with equal size to complete the partition.
- If the block to be divided has reached the minimum block area  $A_{min}$ , the stop condition is satisfied. Considering that there should be a certain number of accessing satellites  $\mathbb{AS}$  in each block to deliver content to subscribers, the area of  $A_{min}$  should not be too small. In fact,  $A_{min}$  is jointly determined by the LEO constellation density, the content of distributed files, and some other practical factors.

Based on the above two conditions, the DBD algorithm is adaptive for user-oriented STN division. The details of the process are shown in Algorithm 1.

## 4.2 STN Link Usability Model

In this subsection, the satellite coverage and link connectivity are analyzed for the interconnection of  $\mathbb{CS}$ ,  $\mathbb{AS}$ , and  $\mathbb{U}$ . The STN link usability model is first described in Fig. 4. We assume that satellites fly on a circular orbit and the Earth radius is represented by  $R_e$ .  $\{s_i\}$  denotes the LEO node set and  $L$  is the altitude. The distances of ISL and user access link are denoted by  $\{D_i\}$ .  $\{H_i\}$  represents the height from the Earth center  $O$  to  $\{D_i\}$ .  $\{\alpha_{ij}\}$  is the angle between sides  $os_i$  and  $s_is_j$ . We summarize the visibility of ISL and user access links as follows:

It is observed that  $D_1$  is the distance between  $s_1$  and  $s_2$ .  $H_1$  is the height from  $O$  to  $s_1s_2$ . If the following formula satisfies

$$H_1 \geq R_e, \quad (2)$$

the two satellites are physically visible to form an inter-satellite communication link.

It is observed that  $D_2$  is the distance between  $s_2$  and  $s_3$ , and  $H_2$  is the height from  $O$  to  $s_2s_3$ . If the following condition is satisfied

$$(H_2 < R_e) \cap (\alpha_{23} < 90^\circ) \cap (\alpha_{32} < 90^\circ), \quad (3)$$

the two satellites are physically invisible to form an inter-satellite communication link.

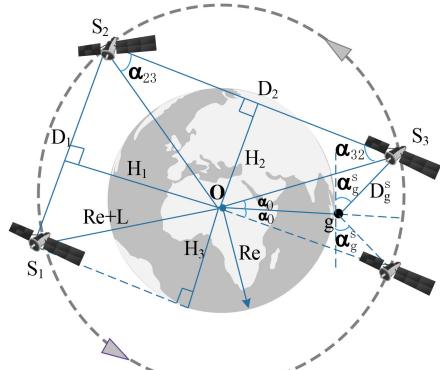


Fig. 4. The LEO link usability model.

For satellite-to-ground communications, the link availability and satellite coverage period need to be determined.  $g$  is assumed as the ground subscriber. Based on the study in [17], the elevation angle  $\alpha_g^s$  between  $s$  and  $g$  can be calculated as

$$\alpha_g^s = \arctan\left(\frac{\cos\theta_g^s - R_e/(R_e + L)}{\sin\theta_g^s}\right), \quad (4)$$

where  $\theta_g^s$  can be calculated by

$$\cos\theta_g^s = \cos(\psi - \lambda)\cos l \cos\varphi + \sin l \sin\varphi. \quad (5)$$

In (5),  $(\psi, l)$  and  $(\lambda, \varphi)$  are the longitude and latitude of  $s$  and  $g$  respectively. During the connection, the Earth center, satellite, and ground user remain in a triangle  $\Delta SOG$ . Thus, the geocentric angle  $\alpha_O$  can be calculated as

$$\alpha_O = \arccos\left(\cos\alpha_g^s \frac{R_e}{R_e + L}\right) - \alpha_g^s. \quad (6)$$

It can be observed that  $\alpha_O$  is negatively correlated with  $\alpha_g^s$ . When  $\alpha_g^s$  reaches the minimal elevation angle  $\alpha_{g,\min}^s$ , the  $\alpha_O$  reaches the maximum value  $\alpha_{O,\max}$ . It can be inferred that the coverage time of each  $s$  to  $g$  is  $2\alpha_{O,\max}$ . Based on the Kepler's third law [17], the period  $T_s^{LEO}$  of  $s$  flying around the Earth can be calculated as

$$T_s^{LEO} = 2\pi\sqrt{\frac{(L + R_e)^3}{\mu}}, \quad (7)$$

where  $\mu = 398,601.58 \text{ km}^3/\text{s}^2$  is the Kepler constant. Then, the satellite coverage period can be evaluated as

$$\begin{aligned} \tilde{T} &= \frac{2\alpha_O}{2\pi} T_s^{LEO} \\ &= 2\left(\arccos\left(\cos\alpha_g^s \frac{R_e}{R_e + L}\right) - \alpha_g^s\right) \sqrt{\frac{(L + R_e)^3}{\mu}}. \end{aligned} \quad (8)$$

From the spatial perspective, the critical elevation angle  $\alpha_{g,\min}^s$  corresponds to the maximum communication distance  $D_{g,\max}^s$ . It is observed that the user-user distance is much smaller than  $D_g$ . Hence, we assume that the  $\alpha_{g,\min}^s$  of all

covered users is identical, followed with the identical  $D_{g,\max}^s$ . The condition of establishing the user access link is:

$$D_g \leq D_{g,\max}^s. \quad (9)$$

From the temporal perspective, we assume the instantaneous geocentric angle of satellite  $s$  in time  $t$  is  $\theta_t^s$ . The elevation angle from the horizon where the satellite arrives is  $\alpha_{g,t}^s$ . The remaining coverage time is  $T_{re}^s$ . Thus, the relationship between  $T_{re}^s$ ,  $\alpha_{g,t}^s$ , and  $\theta_t^s$  is

$$T_{re}^s = \begin{cases} \frac{\alpha_O + \theta_t^s}{2\alpha_O} \tilde{T} & \alpha_{g,\min}^s \leq \alpha_{g,t}^s \leq \alpha_{g,\max}^s \\ \frac{\alpha_O - \theta_t^s}{2\alpha_O} \tilde{T} & \alpha_{g,t}^s > \alpha_{g,\max}^s \end{cases}. \quad (10)$$

where  $\alpha_{g,\max}^s$  is the maximum elevation angle. In (10),  $T_{re}^s$  takes values from  $\tilde{T}$  to 0. Thus,  $T_{re}^s$  can represent the remaining accessing time.

## Algorithm 2. AMCV Algorithm

---

**Input:** Connected graph  $G = (V, E)$ , struct  $V\{e_j\}$ ;  
**Output:** Vertex set  $V_{\min}$ ;

- 1: Label vertex set  $V$  from 1 to  $n$ ;
- 2:  $V_{\min} = \emptyset$ ;
- 3: **for**  $i = 1$  to  $n$  **do** // Select the initial cache node set
- 4:   **if**  $v_i\{e_j\} \neq \emptyset$  **then**
- 5:     Choose one vertex  $u$  from  $v_i\{e_j\}$ ;
- 6:     Add  $v_i$  and  $u$  into  $V_{\min}$ ;
- 7:     Delete  $e_{v_i \rightarrow u}$  from  $v_i\{e_j\}$  and  $u\{e_j\}$ ;
- 8:     **for**  $j = 1$  to  $m$  **do** // Delete the remaining edges
- 9:       **if**  $v_i\{e_j\} \neq \text{NULL}$  **then**
- 10:           $v_i\{e_j\} = \text{NULL}$ ;
- 11:       **end if**
- 12:     **end for**;
- 13:     **for**  $j = 1$  to  $m$  **do** // Delete the remaining edges
- 14:       **if**  $u\{e_j\} \neq \text{NULL}$  **then**
- 15:           $u\{e_j\} = \text{NULL}$ ;
- 16:       **end if**
- 17:     **end for**
- 18:      $BD = BD \cup \{b_i\}, 1 \leq i \leq n^2$ ;
- 19:     **else**
- 20:        $i = i + 1$ ;
- 21:       Go back and execute Step 4;
- 22:     **end if**
- 23: **end for**
- 24: **return**  $V_{\min}$ ;

---

In the above discussion, the physical basis of STN link deployment is established. Next, the transmission rate between the ground user  $g$  and the LEO satellite  $s$  is analyzed. We assume that the available bandwidth of the LEO satellite  $s$  is  $B^s$  Hz, within which  $B_g^s$  Hz is allocated to the ground user  $g$ . The available link capacity of  $s$  is  $C^s$  bps. The spectrum efficiency of the ground user  $g$  is  $E_g^s(t)$  bps/Hz at time instant  $t$ . The transmission rate  $R_g^s(t)$  between the ground user  $g$  and LEO satellite  $s$  is inferred as follows:

$$R_g^s(t) = J_g^s(t)B_g^s(t)E_g^s(t), \forall g \in \mathcal{G} \quad (11)$$

$$\text{s.t. } \sum_{g \in \mathcal{G}} R_g^s(t) \leq C^s, \forall s \in \mathcal{S}, \quad (12)$$

where  $J_g^s(t)$  is a binary variable whether or not the ground user  $g$  accesses the network through the LEO satellite  $s$ .  $J_g^s(t) = 1$  means that the ground user  $g$  accesses the network through the LEO satellite  $s$ . Otherwise,  $J_g^s(t) = 0$ . We define the transmission capacity threshold  $C^s$  to ensure that the sumtransmission rate of all ground users covered by the LEO satellite  $s$  cannot exceed the backhaul capacity.

The formulations of the link usability prove that the ISL in STN is finite and controlled, which makes the cache node deployment scheme different from terrestrial networks.

### 4.3 Cache Node Set Selection

The visibility and periodic variation of satellite-satellite and satellite-ground links are analyzed above. In this subsection, we are pursuing a user-friendly CS =  $\{s_j | 1 \leq s_j \leq CS\}$  for efficient content distribution without taking up too many data transmission resources. Therefore, we use  ${}^\circ(s_j)$  to indicate the number of forwarding links of nodes  $s_j$ . A larger  ${}^\circ(s_j)$  means that data of  $s_j$  can be transferred directly to more nodes. If such  $\{s_j\}$  are selected to form the cache node set, the copies of content may be distributed to a large number of scattered subscribers with only a few hops. In each time slot, the initial cache node set determination can be formulated as a linear programming problem:

$$\begin{aligned} \arg \min_{CS \subseteq S} Z &= \sum_{i=1}^S x_i \\ \text{s.t. } \sum_{i=1}^S x_i \cdot {}^\circ(s_i) &\geq 2S \end{aligned} \quad (13)$$

where  $x_i$  is a binary variable. If  $s_i$  is the cache node, the value of  $x_i$  is 1; otherwise, the value of  $x_i$  is 0. Equation (13) can find a node set  $Z$  that satisfies the condition of a connected graph with the minimum number of nodes. The obtained set is utilized as a basis to find the cache node set.

In order to ensure the backbone of the cache node and reduce the transfer resource waste, we further reduce the unnecessary nodes in  $Z$  to find a minimum coverage vertex set. Actually, the build of minimum coverage vertex set is an NP-complete problem.

**Theorem 1.** *Finding a minimum coverage vertex set in LEO satellite networks is an NP-complete problem.*

**Proof.** Typically, the LEO satellite network in each time slot can be treated as the static graph  $G = (V, E)$ . The minimum cover vertex set,  $V_{\min} = \{v_1, v_2, \dots, v_n\}$ ,  $v_i \in V$ , is a non-null node set with  $n$  vertexes. It has  $V_{\min} \subseteq V$  but cover all the edge of  $G$ . Obviously, the structure of  $V_{\min}$  can be thought of as a union of  $n$  non-null subsets  $V_{\min} = V_1 \cup V_2 \cup \dots \cup V_n$ , where  $V_i = \{v_i\}$ . Given a collection of subsets  $V_{\min}$ , it is checked that if the union of them is equal to  $G$ . The checking takes polynomial time. Thus, building the minimum coverage vertex set  $V_{\min}$  is an NP problem.  $\square$

Furthermore, finding the minimum coverage vertex set  $V_{\min}$  can be converted into a scheduling problem. That is, in all subsets of a given graph  $G = (V, E)$ , find the subset that satisfies the minimum coverage vertex set  $V_{\min}$ . Therefore, finding  $V_{\min}$  is equivalent to finding the minimum overhead in the travel salesman problem (TSP). As the TSP is a NP-

complete problem, building the minimum coverage vertex set  $V_{\min}$  in LEO satellite network is an NP-complete problem. Therefore, Theorem 1 is proved.

---

### Algorithm 3. The CNS algorithm.

---

**Input:** DBD Block divisionBD; AMCV vertex set  $V_{\min}$ ;  
**Input:** Coordinate set  $B^{\lambda\varphi}$  and  $S^{\lambda\varphi}$ ;  
**Output:** Cache node set CS;

```

1: for  $v_i \in V_{\min}$  do // STN node division
2:   for  $b_j \in B^{\lambda\varphi}$  do
3:     if  $v_i(\lambda_i, \varphi_i) \in b_j[(\lambda_{j-1}, \varphi_{j-1}), (\lambda_{j-2}, \varphi_{j-2})]$  then
4:       Delete  $v_i$  from  $S$ ;
5:       Delete  $b_i$  from  $B^{\lambda\varphi}$ ;
6:     end if
7:   end for
8: end for
9: for  $b_i \in B^{\lambda\varphi}$  do // Select cache node for each block
10:  for  $v_j \in S$  do
11:    if  $v_j(\lambda_j, \varphi_j) \in b_i[(\lambda_{i-1}, \varphi_{i-1}), (\lambda_{i-2}, \varphi_{i-2})]$  then
12:      if  $b_i[(\lambda_{i-1}, \varphi_{i-1}), (\lambda_{i-2}, \varphi_{i-2})] \neq \emptyset$  then;
13:        Choose the  $v_j$  closet to the center of  $b_i$ ;
14:        Match  $v_j$  and  $b_i$ ;
15:      end if;
16:      Add  $v_j$  to CS ;
17:    end if
18:  end for;
19: end for
20: return CS;
```

---

For solving an NP-complete problem, there is no deterministic polynomial time algorithm. Therefore, we design the AMCV algorithm to obtain an approximate minimum coverage vertex set in LEO satellite networks. According to (13), we obtain a set of nodes that satisfies the condition of a connected graph  $G = (V, E)$ . For each  $v_i \in V$ , we create an edge set  $E_{v_i}$  that records all edges associated with  $v_i$ . Then, the procedure of the minimum coverage vertex set building can be described as follows:

- Step 1: Traverse over the current  $\{x_k\}$ , and the first non-zero  $x_k$  is the initial vertex  $v_i$ .
- Step 2: If vertex  $v_i$  has associated edges, go to Step 3; otherwise, go to Step 5.
- Step 3: Select an associated edge  $(v_i, v_j)$  and record  $v_i$  and  $v_j$  in the vertex coverage set.
- Step 4: Iterate over two edge sets  $E_{v_i}$  and  $E_{v_j}$ . Delete all edges associated with  $v_i$  and  $v_j$ .
- Step 5:  $i = i + 1$ . If  $i < n$ , go to Step 2; otherwise, the iteration ends.

Specifically, the process of the AMCV algorithm is described by Algorithm 2. Fig. 5 shows the results of the AMCV algorithm. It is observed that  $\{a, c, f, g, i, k, l, n, o, q, r\}$  is one of the optimal minimum coverage vertex set.  $\{a, b, e, f, g, i, k, l, n, o, q, r\}$  is the AMCV result. The AMCV algorithm can obtain an approximate minimum coverage vertex set with greatly reduced computational complexity. Obviously, using this obtained vertex set as cache nodes can achieve the content delivery for ground subscribers. However, as most subscribers have an aggregation effect, the content delivery with this cache node set fails to consider the uneven distribution of subscribers. The users in high-density regions may

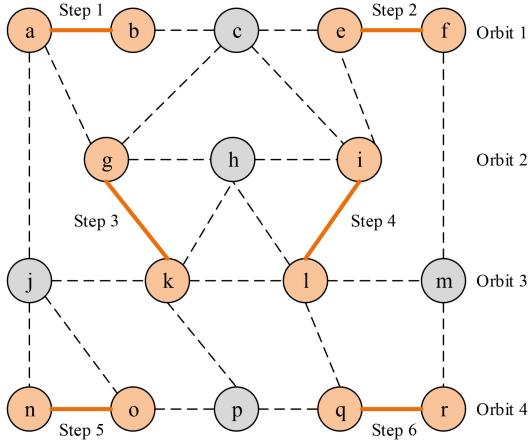


Fig. 5. The calculation results of each step of the AMCV method.

have no cache satellites overhead. Their QoE is not significantly improved, which will be proved in Section 6.

As previously mentioned, the AMCV algorithm obtains the approximate minimum coverage node set of the LEO network, but it fails to cover every user block. If a high-density block is not covered, the subscribers can only encroach on the transmission resources of satellites covering other blocks. Therefore, it is expected that every block, especially high-density block, has its own cache satellites to improve the delivery efficiency. As such, we need to check the user blocks to add more appropriate satellites to the AMCV results as cache nodes so that each cache satellite is marked to its own block. First, the traditional geocentric rectangular coordinate (GRC) is converted into geodetic longitude-latitude coordinate (GLC) for intuitive match of satellites and blocks. In the GRC, The origin is geocentric  $O$ . The  $Z$  axis points to the North Pole. The  $X$  axis points to the intersection of the geodetic meridian plane and equator. The angle between the  $Y$  axis and the  $X$  axis is  $\pi/2$ . The coordinates of satellite  $s$  in GRC can be expressed as  $(X, Y, Z)$ . Then, the latitude and longitude of satellite  $s$  are calculated as follows:

$$\begin{aligned}\lambda &= \arctan(Y/X) \\ \varphi &= \arctan[(Z + R_e \sin \varphi)/\sqrt{X^2 + Y^2}],\end{aligned}\quad (14)$$

where  $\lambda$  is the longitude and  $\varphi$  is the latitude of satellite  $s$  in GLC. Similarly, the  $(\lambda, \varphi)$  coordinates of the entire satellite node set can be obtained as  $S^{\lambda\varphi}\{s_i(\lambda_i, \varphi_i)\}$ . As for the density-based block set, the  $(\lambda, \varphi)$  coordinate intervals are expressed as  $B^{\lambda\varphi}\{b_i[(\lambda_{i-1}, \varphi_{i-1}), (\lambda_{i-2}, \varphi_{i-2})]\}$ . The range of longitude is set to  $[-180^\circ, +180^\circ]$  and the range of latitude is set to  $[-90^\circ, +90^\circ]$  for convenient calculation. In the unified frame of reference, we can directly find the blocks with no cache node and allocate cache satellites for these block. The node with the largest  $\lambda(s_k)$  in the block is selected as the cache node. If there are multiple satisfied nodes, we choose the one closest to the center of the block. The complete process of the cache node selection is described as Algorithm 3.

Actually, each subscriber will obtain a copy of the content from a cache satellite that matched the block. The scope of the block depends on the administrator's configuration, not just the subscriber's density. It is likely that content files are relayed to subscribers via multiple access satellites. To realize the efficient distribution of content files in each block, the breadth-

first-search spanning tree (BFST) algorithm is adopted. The feature of BFST is that all adjacency nodes of each vertex will be traversed in turn. In each block, this will help maximize the ISLs. If the subscriber is covered by the cache satellite, the content file can be obtained directly. Otherwise, the content file will be transferred through the ISL constructed by BFST using the access satellites.

The above constructs complete cache node deployment and content distribution strategies within each time slot. Since the satellite-ground network is a time-varying network, the connectivity of each link changes with time. Hence, we need to derive the STN caching update mechanism to improve the distribution continuity.

## 5 UPDATE OF DENSITY-BASED TIME-VARYING GRAPH

The above proposes the schemes of cache node selection and content distribution in each time slot, where the network topology is fixed. Actually, the STN is a time-varying network [40]. The relative positions and connections between nodes change over time [41]. In this section, we first build the DTG model to obtain the change rules of ISLs. Then, the cache updating strategy is designed.

### 5.1 Time-varying STN Model

Different from traditional terrestrial networks, we utilize a time-varying graph  $G$  to formulate the STN topology:

$$G = (S, E, T, R, W), \quad (15)$$

where  $S, E$  are denoted as the network nodes  $\{s_i\}$  and the inter-satellite and user access links  $\{e_{ij}\}$ .  $T$  is the time variable and is assumed to be  $\mathbb{R}^+$  for time-varying networks.  $R$  is the resource function.  $R(v_i, t)$  represents the resource usage of the selected node  $v_i$  in time  $t$ .  $W$  is the weight index.  $\{w_{ij}\}$  shows the weight of each edge. Further, we can derive the following components:

$$\begin{cases} E \subseteq S \times S \\ E \times T = \{0, 1\} \\ T \subseteq \mathbb{R}^+ \\ W = E \cdot L(t), \end{cases} \quad (16)$$

where  $L$  is the coefficient of link capability and is a function of time  $t$ .  $E \times T = \{0, 1\}$  represents whether a link  $e_{ij}$  exists at time  $t$ .  $e_{ij} \times t = 1$  means there is a link between nodes  $i$  and  $j$ . Otherwise,  $e_{ij} \times t = 0$ . Based on the definition of temporal contact proposed in [23], we can judge if the path  $\mathbb{P}$  exists in STN. Each link in the path  $\mathbb{P}$  is denoted by  $(e_k, t_k)$ , so  $\mathbb{P} = \{(e_1, t_1), (e_2, t_2), \dots, (e_n, t_n)\}$ . For  $\forall 1 \leq k \leq n$ , the existence of  $\mathbb{P}$  meet the following conditions

$$\begin{cases} \{e_i \times t_i\} = 1 \\ t_{i+1} \geq t_i + D(e_i, t_i) \\ e_i \times [t_i, t_i + D(e_i, t_i)] = 1 \\ end(e_i) = start(e_{i+1}), \end{cases} \quad (17)$$

where  $D(e_i, t_i)$  is the delay of data delivery on link  $e_i$  in time  $t_i$ . The formula  $t_{i+1} \geq t_i + D(e_i, t_i)$  guarantees the continuity of data delivery. The formula  $e_i \times [t_i, t_i + D(e_i, t_i)] = 1$  determines the minimum link existence time. Therefore, the essence of the existence of  $\mathbb{P}$  is that every component link  $(e_k, t_k)$  exists.

For each block, the existence time of the delivery path generated by BFST can be calculated by (17) to obtain the topology update interval. However, (17) cannot handle the DTG switch and cache updating since almost all block topologies have different lifetimes. Therefore, the DTG update time and the cache updating scheme are discussed next.

## 5.2 DTG Update Time

As the content initiator, the source node needs to reconfigure the cache node set in each time slot to deliver or update the content file. The cache updating time can be derived by the DTG updating. A sequence of discrete and finite time span  $T = \{1, 2, \dots, T\}$  is needed to divide the time  $T \subseteq \mathbb{R}^+$  into a series of time slots  $\mathbb{T} = \{\zeta^\tau | 1 \leq \tau \leq T\}$ . During each  $\zeta^\tau$ , the topology is regarded as a quasi-static snapshot. In DTG, the link set  $(\{e_k\}, \zeta^\tau)$  of each  $\zeta^\tau$  is jointly determined by the link usability model and the CNS algorithm. In  $\zeta^\tau$ , the links  $\{e_k^{b_i}\}$  of block  $b_i$  is generated by the BFST algorithm, where the  $s_j \in \text{CS}$  is the root node and  $s_k \in \text{AS}$  are leaf nodes. Therefore, the period of  $b_i$  in  $\zeta^\tau$  from the determination to the disappearance of  $s_j \in \text{CS}$  is regarded as  $\zeta_{b_i}^\tau$ .

Blocks are different in region size and time slot length, so the topology updates between different blocks are not synchronized. The time slot sequence of each block  $b_i$  is denoted by  $\mathbb{T}_{b_i} = \{\zeta_{b_i}^\tau | 1 \leq \tau \leq T_{b_i}\}$ . For each  $\zeta_{b_i}^\tau$ , the cache node reselection is regarded as the start time  $t_{b_i}^{\tau-0}$ , and the coordinates of the cache node are  $t_{b_i}^{\tau-0}(X_{b_i}^{\tau-0}, Y_{b_i}^{\tau-0}, Z_{b_i}^{\tau-0})$ . Similarly, the coordinates of the cache node leaving the block can be calculated as  $t_{b_i}^{\tau-1}(X_{b_i}^{\tau-1}, Y_{b_i}^{\tau-1}, Z_{b_i}^{\tau-1})$  by (14). Without loss of generality, assume that the time from  $t_{b_i}^{\tau-0}$  to  $t_{b_i}^{\tau-1}$  of the cache node is  $\zeta_{b_i}^\tau$ :

$$\zeta_{b_i}^\tau = \sqrt{(X_{b_i}^{\tau-1} - X_{b_i}^{\tau-0})^2 + (Y_{b_i}^{\tau-1} - Y_{b_i}^{\tau-0})^2 + (Z_{b_i}^{\tau-1} - Z_{b_i}^{\tau-0})^2}, \quad (18)$$

where  $c$  denotes the speed of light. This is the topology duration of block  $b_i$  at the  $\tau$  time slot. Note that  $t_{b_i}^{\tau-1} = t_{b_i}^{(\tau+1)-0}$ . The length of each time slot  $\zeta_{b_i}^\tau$  can be obtained iteratively. The topology duration set  $\{\zeta_{b_i}^\tau\}$  is obtained. The average topology duration  $\zeta_{b_i}^{\text{ave}}$  of block  $b_i$  is calculated as

$$\zeta_{b_i}^{\text{ave}} = \sum_{\tau=1}^{T_{b_i}} \zeta_{b_i}^\tau / T_{b_i}. \quad (19)$$

Similarly, the average topology duration set of the block set  $\{b_i\}$ ,  $\mathbb{T}^{\text{ave}} = \{\zeta_{b_i}^{\text{ave}} | 1 \leq i \leq N\}$ , is available. After the above calculation many times, we rely on an empirical observation that the sequence of  $\mathbb{T}^{\text{ave}}$  is similar from large to small, although the values of  $\zeta_{b_i}^{\text{ave}}$  are different in each calculation. The sequence of  $\mathbb{T}^{\text{ave}}$  is shown as follows

$$\zeta_{b_i-1}^{\text{ave}} \geq \zeta_{b_j-2}^{\text{ave}} \geq \dots \geq \zeta_{b_k-N}^{\text{ave}}, \forall \zeta_{b_i}^{\text{ave}} \in \mathbb{T}^{\text{ave}}, \quad (20)$$

where  $\mathbb{T}^{\text{ave}}$  is an almost unchanging sequence. It is observed that  $\zeta_{b_i}^{\text{ave}}$  of large block is longer, and  $\zeta_{b_i}^{\text{ave}}$  of small block is shorter. In order to ensure the consistency of the DTG update, the minimum topology duration  $\zeta_{b_i}^{\tau_{\min}}$  of the block

$b_i$  to which the  $\zeta_{b_i-N}^{\text{ave}}$  belongs is selected as the DTG update interval:

$$\zeta_{b_i}^{\text{ave}} = \min\{\zeta_{b_i-1}^{\text{ave}}, \zeta_{b_j-2}^{\text{ave}}, \dots, \zeta_{b_k-N}^{\text{ave}}\} \quad (21)$$

$$\zeta_{b_i}^{\tau_{\min}} = \min\{\zeta_{b_i}^1, \zeta_{b_i}^2, \dots, \zeta_{b_i}^{\tau}, \dots, \zeta_{b_i}^{T_{b_i}}\}. \quad (22)$$

Taking  $\zeta_{b_i}^{\tau_{\min}}$  as the DTG update time, the continuity of each distribution block is guaranteed. Every  $\zeta_{b_i}^{\tau_{\min}}$  the  $gs_0$  carry out the CNS algorithm to reconfigure CS. The cache update method should be analyzed for closed-loop autonomy.

## 5.3 Cache Content Updating

According to the update frequency  $\zeta_{b_i}^{\tau_{\min}}$ , a sequence of new DTG snapshots  $\mathbb{T} = \{\zeta^\tau | 1 \leq \tau \leq T\}$  can be calculated. The topology of each snapshot is considered invariable. During DTG switching, the new cache content and related configuration are transferred from the source node to the new cache nodes. The ISL paths of content updating should be determined. Therefore, we propose an update ISL set  $\{U_\tau\}$  to match each  $\zeta^\tau \in \mathbb{T}$ . The  $\{U_\tau\}$  is precalculated and prestored to guide ISL deployment. At present, the most common pathfinding method in a static graph is to find the shortest topology path, using an algorithm like Dijkstra [33]. In DTG, the motion of nodes gives new constraints for path finding. Considering the space communication, link reliability is more needed. The constraints like propagation delay, transmission rate, and link stability should be analyzed systematically.

**Propagation delay:** Normally the source satellite  $s$  and cache satellite  $d$  cannot be connected directly. We use  $p(s_1, s_2)$  to represent the ISL between satellite  $s_1$  and  $s_2$ . Some relay nodes are needed to connect a series of ISLs  $\{p(s, r_1), p(r_1, r_2), \dots, p(r_n, d)\}$  to build  $\text{Path}(s, d)$  and complete the data transmission. The set of all possible paths from node  $s$  to node  $d$  is denoted by  $P_{sd}$ . For fast data transfer, we define the minimum hop path (MHP):

$$\text{MHP}(s, d) = \text{Min}\{\text{Path}(s, d) | \text{Path}(s, d) \in P_{sd}, \{p\}_{sd} \neq 0\}, \quad (23)$$

where  $\{p\}_{sd} \neq 0$  means that each  $p(s_1, s_2) \in \text{Path}(s, d) \neq 0$ . The connectivity of path is first assured. The MHP can find the  $\text{Path}(s, d)$  with the lowest number of ISLs among all possible paths. According to the fundamental triangle theorem, the shortest path (SHP) certainly belongs to the MHP set. Therefore, the propagation delay of  $\text{MHP}(s, d)$  is similar to that of the SHP. We utilize MHP as the basis for pathfinding. Constraints like transmission rate and link stability are added to further satisfy the delivery stability.

**Transmission rate:** The transmission rate depends on the current resource usage of each satellite's transceiver. For example, the forwarding capability  $B_{s_1-\text{total}}$  of satellite  $s_1$  is 19 Mbps, and 4 Mbps has been used to forward tasks. The forwarding capability  $B_{s_2-\text{total}}$  of satellite  $s_2$  is 15 Mbps, and 9 Mbps has been used to forward tasks. If the ISL is established between  $s_1$  and  $s_2$ , the maximum available transmission rate  $B_{s_1s_2-\text{max}}$  is 6 Mbps, which is the smaller  $B_{\text{max}}$  for both satellites. In DTG, the available transmission rate of

ISL  $p(s_1, s_2)$  is denoted as  $B_{s_1 s_2}$ . The maximum available transmission rate of  $Path(s, d)$  is shown as follows:

$$B_{sd} = \min\{B_{sr_1}, B_{r_1 r_2}, \dots, B_{r_n d}\}. \quad (24)$$

By calculating  $B_{sd}$ , MHP can filter out ISLs below the required transmission capacity to reduce redundant calculation.

**Link stability:** High link stability can reduce jitter and signal attenuation to bring stable space data transmission. As Fig. 4 shows, satellites in the same orbital plane will keep relatively invariant positions with each other. For relay node selection, satellites in the same orbit should be given priority by MHP. We set the link weights in the DTG model to modify the pathfinding. According to formula (16), the weight matrix can be expressed as:

$$W = E \cdot L(t) = (w_{ij})_{n \times n} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \dots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}, \quad (25)$$

where  $w_{ij}$  denotes the weight of the link from satellite  $s_i$  to  $s_j$ . The initial value of each  $w_{ij}$  is the same. When MHP starts, the weight of  $\{w_{ij}\}$  in the orbital plane of the source node increases. To further simplify the pathfinding problem and reduce the computation complexity, we assume that the updated data volume of each cache satellite is uniform. The pathfinding algorithm is modified as the maximum weight MHP (MWMHP):

$$\begin{aligned} MWMHP(s, d) = \text{Min}\{Path(s, d) | Path(s, d) \in P_{sd}\} \\ \text{s.t.} \end{aligned}$$

$$\begin{aligned} W_{sd} &= \max\{w_{sr_1}, w_{r_1 r_2}, \dots, w_{r_n d}\} \\ \{B_{sr_1}, B_{r_1 r_2}, \dots, B_{r_n d}\} &\geq B_{Re} \\ \{p\}_{sd} &\neq 0 \\ \sum_{j=1}^n e_{ij} &\leq 4 \\ i - O &\leq j \leq i + O, \end{aligned} \quad (26)$$

where  $\{i, j\}$  are the subscripts of the links in the weight matrix. The maximum ISL number of each satellite is 4 to limit the search scope.  $O$  is the number of satellites in each orbit to avoid cross-orbital ISL. The modified MWMHP adds delivery rate and link stability as MHP constraints to find an express and reliable path. The efficiency of cache content updating during DTG switching is guaranteed.

Based on the above analysis, the minimum block update interval is selected as the DTG update time. Meanwhile, we design the MWMHP method to complete the DTG update. As such, the proposed time-varying STN content distribution scheme is orchestrated to complete the closed-loop autonomy.

## 6 SIMULATION RESULTS AND ANALYSIS

The computational experiments are carried out over a predictable satellite-terrestrial network simulation platform. Since popular constellations like the Starlink and the Kuiper are inclined constellations and mainly cover the earth between the latitude of  $[60^\circ S, 60^\circ N]$ , the LEO satellite network used here is a polar constellation for full earth

TABLE 2  
Parameters of Satellite Network Segments

Type	Height/km	Orbit Number
LEO satellite	895.5	6
Satellites Per Orbit	Inclination/deg	Revolution/sec
22	86.4	6027.14

coverage and content distribution. The parameters are shown in Table 2. A 6x22 LEO-polar constellation is generated using the System Tool Kit (STK) simulator to guarantee the practicality, as shown in Fig. 6. The ground users are set based on the Standard Object Database (SOD) in STK. The SOD records the geographic locations of 723 satellite terminals around the world, most of which are located on islands, mountains, rural areas, or other remote regions that can be efficiently served by satellites. Compared with an assumed user distribution, the SOD can truly represent the distribution and density of subscribers.

During the simulation, the scenarios and cache node configurations of different time slots are imported into the NS3 simulator to build corresponding network segments. In the NS3, the ISLs utilize Ka band and the channel capacity is 25 Mbps. The satellite-ground links utilize the L band and the channel capacity is 5 Mbps. Even though there are many processes in content delivery, we only simulate and evaluate the distribution process, based on the proposed STN cache deployment and update schemes. Hence, processes like how the source node allocates content with different popularities and sizes to the cache nodes to improve the cache hit rate is not considered in this paper. In the simulation, all the cache segments are assumed to distribute the desired file with the same size to the requesting users at each time slot. The requests are subject to Poisson distribution. The packet delivery rate is constant for each satellite-ground link. Besides, it is noted that a ground user can only access one satellite at a certain time. The onboard data processing time is negligible compared with a relatively large propagation delay.

Based on the platform above, five types of STN content distribution schemes are considered for comparison:

**Maximum Degree Distribution (MDD)** [22]: the CS is composed of satellites with the maximum  $\text{deg}(s_i)$ .

**Back-Tracing Caching (BTC)** [42]: Each user first searches the inquiry path to  $gs_0$  based on Minspan Tree routing. The most shared on-path satellites are selected to CS.

**Uniform Content Distribution (UCD)**: the satellites in CS are uniformly distributed in STN.

**Central Content Distribution (CCD)**: the cache satellites are selected around the source node.

**Density-Based Content Distribution (DCD)**: the CS is calculated by proposed DBD, AMCV, and CNS algorithms.

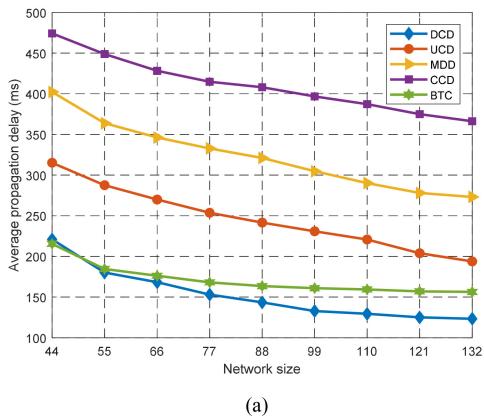
Since five schemes have different cache satellite deployment, the distribution delay and network load are first investigated under different STN conditions. Then, the stabilities of the five schemes are explored with a time-varying STN topology. From the three aspects, we assess whether the proposed scheme is more effective and user-friendly.

Different users access STN from different areas to obtain the content. The most intuitive factor affecting user QoE is

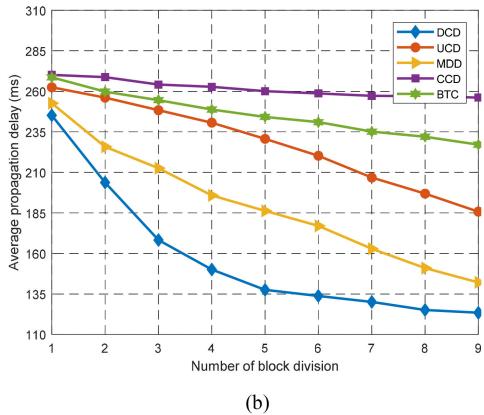


Fig. 6. The simulation model.

latency. Since the long propagation delays due to ISL distances seriously deteriorates performance, the physical distance is one of the most important indicators impacting distribution efficiency and user QoE. Considering the instantaneity and synchronicity demands of content distribution, we analyze the impact of (a) network size, (b) division accuracy, (c) number of user requests, and (d) cached file size on the average propagation delay (APD) of the five schemes to evaluate the performance. The evaluation is conducted using the control variable method. We utilize the average value by multiple calculations as result to reduce the random error.



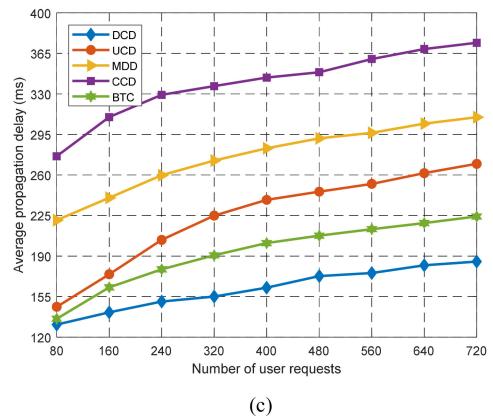
(a)



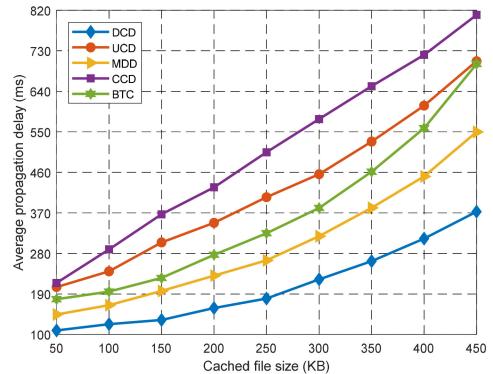
(b)

The network size is a precision reflection of LEO satellite network coverage. Fig. 7a shows the APD performance of five schemes in different network sizes. As shown in the figure, the APDs of 5 schemes already have differences as the network size is small. As the number of satellites increases, all APDs decline. It is observed that the APDs of the DCD and BTC schemes are smaller at first. This indicates that user-based cache node deployment is more efficient under coarse-grained coverage. With the increase of satellite network size, the results of 5 schemes are more accurate. In the second half, the APD decreases in all five methods. Since the pathfinding result of BTC is stable after 88, the APD stayed above 150 ms. The UCD and MDD schemes rely on the objective network location so that their improvement in APD is limited. In the CCD scheme, each cache node selection is similar, so that the APD remains high and always above 360 ms. The cache node set of the DCD scheme keeps updating as the network size increases, which benefits most from large-scale STN.

Fig. 7b displays the APD performance of five schemes in different STN division precision. For the DCD scheme, the division precision is equivalent to the iteration times of DBD algorithm. Similarly, the other schemes match the same number of cache nodes. It can be seen that all APDs of the five schemes decline at different rates, but the APD variation of the DCD scheme is the largest. In the division process of DCD, the distance between cache nodes and user will decrease with the number of DBD iterations. In contrast, the CCD method ignores the user density, and the APD decrease is limited. The APD of the MDD scheme decreases relatively fast. This is because more satellites with



(c)



(d)

Fig. 7. The performance of APD under different (a) network sizes, (b) division precision, (c) number of user requests, and (d) cached file size.  
Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on September 21, 2023 at 05:36:35 UTC from IEEE Xplore. Restrictions apply.

large degree are selected, reducing the number of ISLs. In the UCD scheme, the cache nodes are uniformly distributed. More division precision makes the space between cache nodes smaller, improving the APD performance. The additional cache nodes of BTC are deployed in regions where ISL reuse is less. The change of APD is limited.

Fig. 7c reveals the APD change of five methods with the number of user requests. Since the simulation platform is an AMD R7-2700X CPU workstation, it is reasonable to give priority to network size and block division. The capacity of each node and the cached file size are reduced proportionally, which can still demonstrate the performance of each scheme. In Fig. 7c, the maximum downlink capacity of the access satellite and the cached file size are set as 5 Mbps and 250 KB, respectively. The network size and block division are adjusted that each user is covered by more than 2 satellites. It is assumed that if one access satellite reaches saturation, the content will be relayed to the nearby available access satellite. It is observed that the APDs of five schemes increase with the number of user requests. The main reason is that non-optimal ISLs increase when content is distributed. The APD of DCD grows slowest, followed by BTC, UCD, MDD, and CCD. The BTC scheme can utilize Minspan Tree routing to obtain the minimum delay path for new access users. The APD grows slowly. The alternative ISLs of UCD, MDD, and CCD are far away from users, leading to the rapid growth of APD. Although not the optimal links, the alternative ISL selection of DCD is still based on the user density. The BFST algorithm will find suboptimal ISLs to nearby access satellites, resulting in slow APD growth.

Fig. 7d presents the APD change of five methods in different content file sizes. Similar to Fig. 7c, the file size affects the selection of ISLs and accessing links. It can be seen that the APD of all five methods increases with the file size. Since each ISL has limited transmission capacity, larger files require additional leisure paths for delivery. Enabling alternate paths increases the APD. The DCD has the slowest delay growth at first. Considering the link planning of DCD exists locally in each block, newly added links have less impact on APD. Compared with UCD, the MDD has more alternative links, and the APD performs better. The BTC pursue the reduction of redundant ISLs. The distribution performance is heavily influenced by the full occupation of each ISL. The delivery path of CCD is the longest. Its APD is always the largest as the file size increases.

The results of Fig. 7 indicate that the cache satellite deployment of the DCD scheme is more user-oriented under different STN conditions. Further, the Fig. 8 shows the traffic performance of the proposed scheme during content distribution, including the network throughput and the STN congestion. We select two sub-optimal schemes shown in Fig. 7, MDD and BTC, for comparison. The network size and block division are adjusted based on Fig. 7c. The user request and content distribution are assumed to use the same path. Specifically, the proposed DCD scheme utilizes the BFST routing for content distribution. The MDD utilizes the Dijkstra-based shortest path for content delivery. The BTC uses the Minspan Tree routing for ISL deployment, where the most shared on-path satellites are selected as the cache satellites. In Fig. 8a, it is observed that the network throughput of all 3 schemes increases linearly and is almost the same when the number of user requests is small. However, when the number of user

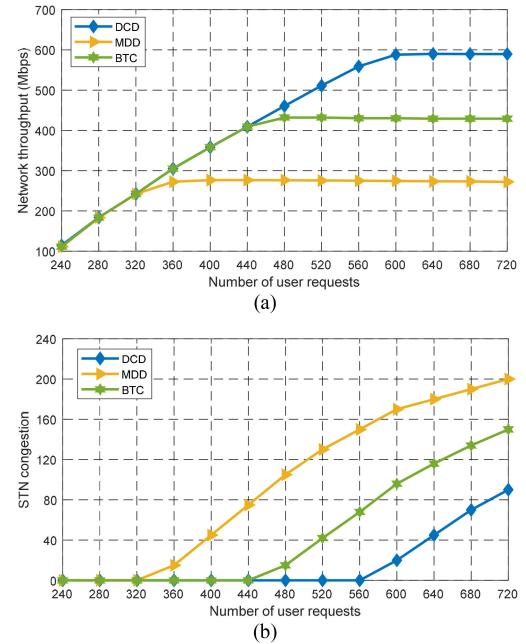


Fig. 8. The comparison of (a) network throughput and (b) STN congestion.

requests is more than 400, the network throughput of MDD stops growing and stabilizes at 270 Mbps. Similarly, when the number of users is more than 480, the BTC stabilizes at 430 Mbps. When the number of users is more than 600, the proposed DCD stabilizes at 590 Mbps. It indicates that the proposed DCD scheme has better performance in network throughput when faced with more user requests. Moreover, the STN congestion  $C_{STN}$  is explored during the experiment, which is calculated by

$$C_{STN} = \sum_{u_n \in \mathbb{U}} (q_{u_n}^{loss} / Q_{u_n}), \quad (27)$$

where  $q_{u_n}^{loss}$  and  $Q_{u_n}$  are respectively the number of packet loss and the total number of packet sent of each user  $u_n$ . The sum is utilized to represent the STN congestion degree. It is showed from Fig. 8b that when the number of user requests is small, the congestion degree of all 3 methods is 0. As the user number increases, the STN congestion of MDD begins to grow rapidly, followed by BTC and DCD. The DCD has the lowest STN congestion degree. It can be concluded from Fig. 8 that compared with MDD and BTC, the proposed DCD scheme performs better when more user requests. The density-based cache satellite deployment can alleviate the imbalance of STN load and improve the content distribution efficiency.

The above experiments are carried out with invariant time. Since the STN is time-varying, the performance of content distribution schemes needs to be investigated in dynamic scenarios. Next, we evaluate the APD performance of the proposed DCD scheme under continuous-time variation. The simulation includes the DCD, MDD, BTC, and UCD methods. The APD performance of the CCD scheme is negligible as the selected cache satellites change little with time. We equally extracted 125 moments in a 753-second interval to calculate the APD of each method. The results are shown in Fig. 9. It is observed that the curve of each method is composed of several segments. The APD of each segment varies from high to low and up again. This is because the APD of the distribution

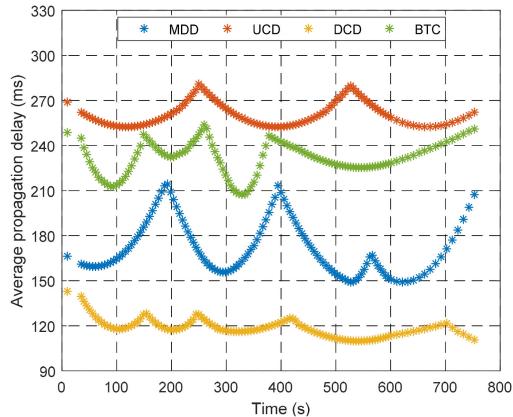


Fig. 9. The comparison of APD change with time.

link changes as the satellite moves from far and near and far away from the user. The replacement of segments represents a switch of cache nodes. Among the four methods, the DCD scheme has better APD performance. The network topology changes more times. The curve of DCD varies less than that of BTC, UCD, and MDD, which expresses stable ISLs connection based on the MWMHP algorithm. Without considering the distribution of users, the APD of MDD changes rapidly. The curve variation of each time slot is quite different. The UCD expresses stable APD changes, but the APD is too high to provide QoE-aware service. As the back-tracing node selection, the APD segments of BTC are unstable. The node deployment varies significantly for each time slot. The service continuity is not considered.

During content distribution, the link hops reflect the number of satellites forwarding the content before reaching the ground user. The increase in link hops may result in more packet loss, decreasing the user QoE. We selected 9 STN moments by equal interval sampling to calculate the average delivery hops (ADH) of each scheme. The BTC scheme is not continued to be compared because its performance is not stable at time latitude. The ADH performance of UCD, MDD, and DCD is shown in Fig. 10. It is observed that the ADH variations of the three schemes are fluctuating in different positions. The number of ADH of proposed DCD has relatively remained low, compared to the UCD and MDD. This is because more users can fetch content from nearby cached satellites in each time slot. Meanwhile, the fluctuations of 3 schemes are different. The results are similar to the curve

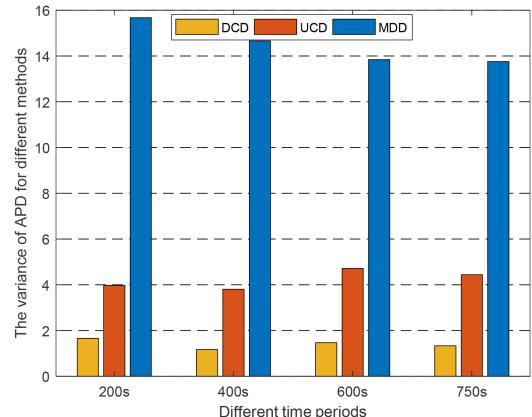


Fig. 11. The comparison of APD variance in different time periods.

variation in Fig. 9. The wave of DCD scheme reflects that it can deal with the dynamic topology well. In the time-varying STN, the DCD scheme has a better stability.

Furthermore, the variances of APD for different methods in different periods are summarized in Fig. 11. We investigate the network stability of the scheme through variance variation. It can be seen that the variance of the MDD is much larger than that of the DCD and UCD schemes. This corresponds to the amplitude of the curvature change in Fig. 9. This is because MDD uses the nodes with the highest degree as the cache node set for content distribution. However, the time-varying STN leads to frequent changing of the network topology, the cache node set, and content delivery link. Although the APD of MDD performs well in the static network (as shown in Fig. 7), it has poor performance in network stability. In contrast, both DCD and UCD schemes use network region division to improve network stability. Compared with UCD, the proposed DCD takes the user-density-based in-network caching and content distribution into consideration. The link change in each block is more refined and controllable, which improves the STN stability during long time data distribution.

We are also concerned with the performance of continuous data distribution of different schemes. The changes in the total data amount of distribution (TAD) of three schemes are demonstrated in Fig. 12. In the simulation, the number of user requests is selected as 400. All other variables are equal. The area bounded by the curve and the axis is utilized to represent the TAD difference. As shown in Fig. 12, the TAD of all 3 methods increases linearly with

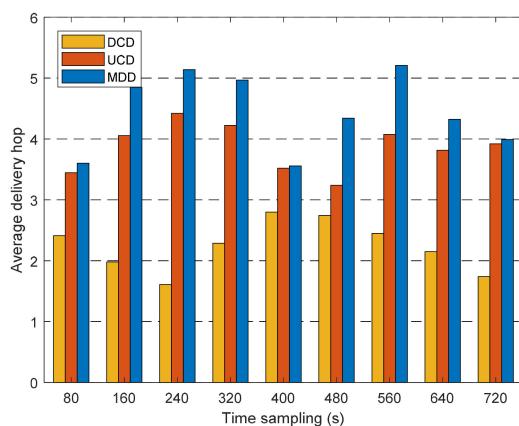


Fig. 10. The comparison of ADH change with time sampling.

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on September 21, 2023 at 05:36:35 UTC from IEEE Xplore. Restrictions apply.

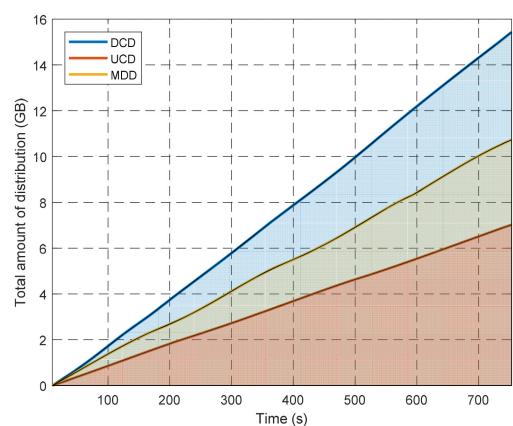


Fig. 12. The comparison of TAD change with time.

Authorized licensed use limited to: XIDIAN UNIVERSITY. Downloaded on September 21, 2023 at 05:36:35 UTC from IEEE Xplore. Restrictions apply.

time. This indicates that the distribution process is stable. It is observed that the DCD scheme performs better than UCD and MDD. The slope of the DCD curve is higher than the other two methods. This suggests that the TAD of DCD is the largest per unit of time and the TAD gaps are significant over time. Considering the large file distribution, the proposed DCD scheme is more advantageous.

The results in Figs. 9, 10, 11, and 12 indicate that the proposed DCD scheme has more stable performance in continuous time. Especially, the DCD scheme has more APD advantages for large file distribution.

## 7 CONCLUSION

In this paper, we have presented a QoE-aware efficient content distribution scheme for STN. The novel in-network caching and file distribution approach takes the unevenness of user distribution into account. Therefore, the density-based block division algorithm has been proposed to divide the STN into a series of blocks with different sizes according to user density. To deploy the caching satellites, we have analyzed the link usability and proposed an approximate minimum coverage vertex set algorithm. A novel cache node selection algorithm has been designed for optimal user block matching. Considering the time variability of STN, the cache content updating mechanism has been derived to enable a stable and sustainable user experience. Simulation results have shown that the proposed DCD method, compared with existing BTC, UCD, MDD, and CCD caching strategies, can reduce the APD and network load under different STN conditions. Furthermore, the proposed DCD method has better stability under continuous time-varying STN topology.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No. 61571104), the Sichuan Science and Technology Program (No. 2018JY0539), the Key projects of the Sichuan Provincial Education Department (No. 18ZA0219), the Fundamental Research Funds for the Central Universities (No. ZYGX2017KYQD170), the CERNET Innovation Project (No. NGII20190111), the Fund Projects (Nos. 2020-JCJQ-ZD-016-11, 61403110405, 315075802, JZX6Y202001010161), and the Innovation Funding (No. 2018510007000134), and the Innovation Funding (No. 2018510007000134), and the Key Research and Development Plan - Major Scientific and Technological Innovation Projects of ShanDong Province (2019JZZY020101).

## REFERENCES

- [1] J. Du, C. X. Jiang, H. J. Zhang, Y. Ren, and M. Guizani, "Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202–2217, Oct. 2018.
- [2] Z. J. Zhang, W. Y. Zhang, and F. H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.
- [3] G. Giambene, S. Kota, and P. Pillai, "Satellite-5G integration: A network perspective," *IEEE Netw.*, vol. 32, no. 5, pp. 25–31, Sep./Oct. 2018.
- [4] Y. H. Ruan, Y. Z. Li, C. X. Wang, and R. Zhang, "Energy efficient adaptive transmissions in integrated satellite-terrestrial networks with SER constraints," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 1, pp. 210–222, Jan. 2018.
- [5] S. Tani, K. Motoyoshi, H. Sano, A. Okamura, H. Nishiyama, and N. Kato, "An adaptive beam control technique for q band satellite to maximize diversity gain and mitigate interference to terrestrial networks," *IEEE Trans. Emerg. Top. Comput.*, vol. 7, no. 1, pp. 115–122, First Quarter 2019.
- [6] S. Zhang, W. Quan, J. L. Li, W. S. Shi, P. Yang, and X. M. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, Sep. 2018.
- [7] V. Bankey, P. K. Upadhyay, D. B. Da Costa, P. S. Bithas, A. G. Kanatas, and U. S. Dias, "Performance analysis of multi-antenna multiuser hybrid satellite-terrestrial relay systems for mobile services delivery," *IEEE Access*, vol. 6, pp. 24729–24745, Apr. 2018.
- [8] W. Abderrahim, O. Amin, M.-S. Alouini, and B. Shihada, "Latency-aware offloading in integrated satellite-terrestrial networks," *IEEE Open J. Commun. Soc.*, vol. 1, no. 2, pp. 490–500, Apr. 2020.
- [9] D. Chen, W. Wang, and T. Jiang, "New multicarrier modulation for satellite-ground transmission in space information networks," *IEEE Netw.*, vol. 34, no. 1, pp. 101–107, Jan./Feb. 2020.
- [10] S. Yang, H. Li, Z. Lai, and J. Liu, "A synergic architecture for content distribution in integrated satellite and terrestrial networks," in *Proc. IEEE/CIC Int. Conf. Commun.*, 2020, pp. 96–101.
- [11] C. Qiu, H. P. Yao, F. R. Yu, F. M. Xu, and C. L. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, Jun. 2019.
- [12] S. S. Gu, J. Jiao, Z. X. Huang, S. H. Wu, and Q. Y. Zhang, "ARMA-based adaptive coding transmission over millimeter-wave channel for integrated satellite-terrestrial networks," *IEEE Access*, vol. 6, pp. 21635–21645, Apr. 2018.
- [13] T. Liang, K. An, and S. C. Shi, "Statistical modeling-based deployment issue in cognitive satellite-terrestrial networks," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 2, pp. 202–205, Apr. 2018.
- [14] B. Li, Z. Fei, Z. Chu, F. Zhou, K.-K. Wong, and P. Xiao, "Robust chance-constrained secure transmission for cognitive satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4208–4219, May 2018.
- [15] P. Wang, J. X. Zhang, X. Zhang, Z. Yan, B. G. Evans, and W. B. Wang, "Convergence of satellite and terrestrial networks: A comprehensive survey," *IEEE Access*, vol. 8, pp. 5550–5588, Dec. 2020.
- [16] E. Wang, H. Li, and S. Zhang, "Load balancing based on cache resource allocation in satellite networks," *IEEE Access*, vol. 7, pp. 56864–56879, Apr. 2019.
- [17] S. Fu, J. Gao, and L. Zhao, "Integrated resource management for terrestrial-satellite systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3256–3266, Mar. 2020.
- [18] A. A. Dowhuszko, J. Fraire, M. Shaat, and A. Pérez-Neira, "LEO satellite constellations to offload optical terrestrial networks in placement of popular content in 5G edge nodes," in *Proc. 22nd Int. Conf. Transparent Opt. Netw.*, 2020, pp. 1–6.
- [19] Q. Li, Y. M. Zhang, Y. Y. Li, Y. Xiao, and X. H. Ge, "Capacity-aware edge caching in fog computing networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9244–9248, Aug. 2020.
- [20] H. Wu, J. Li, H. Lu, and P. Hong, "A two-layer caching model for content delivery services in satellite-terrestrial networks," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–6.
- [21] S. J. Liu, X. Hu, Y. P. Wang, G. F. Cui, and W. D. Wang, "Distributed caching based on matching game in LEO satellite constellation networks," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 300–303, Feb. 2018.
- [22] Z. H. Yang, Y. Li, P. Yuan, and Q. Y. Zhang, "TCSC: A novel file distribution strategy in integrated LEO satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5426–5441, May 2020.
- [23] Z. Q. Zhang, C. X. Jiang, S. Guo, Y. Qian, and Y. Ren, "Temporal centrality-balanced traffic management for space satellite networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4427–4439, May 2018.
- [24] J. Li, K. P. Xue, D. S. L. Wei, J. Q. Liu, and Y. D. Zhang, "Energy efficiency and traffic offloading optimization in integrated satellite-terrestrial radio access networks," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 4, pp. 2367–2381, Apr. 2020.
- [25] Y. R. Cao, Y. P. Shi, J. J. Liu, and N. Kato, "Optimal satellite gateway placement in space-ground integrated network for latency minimization with reliability guarantee," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 2, pp. 174–177, Apr. 2018.
- [26] B. Y. Deng, C. X. Jiang, J. Yan, N. Ge, S. Guo, and S. H. Zhao, "Joint multigroup precoding and resource allocation in integrated terrestrial-satellite networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8075–8090, Aug. 2019.

- [27] Z. Ji, S. Z. Cao, S. Wu, and W. B. Wang, "Delay-aware satellite-terrestrial backhauling for heterogeneous small cell networks," *IEEE Access*, vol. 8, pp. 112190–112202, Jun. 2020.
- [28] J. Li, K. P. Xue, J. Q. Liu, Y. D. Zhang, and Y. G. Fang, "An ICN/SDN-based network architecture and efficient content retrieval for future satellite-terrestrial integrated networks," *IEEE Netw.*, vol. 34, no. 1, pp. 188–195, Jan./Feb. 2020.
- [29] W. F. Shi *et al.*, "Distributed contact plan design for multi-layer satellite-terrestrial network," *China Commun.*, vol. 15, no. 1, pp. 23–34, Jan. 2018.
- [30] M. Luglio, S. P. Romano, C. Roseti, and F. Zampognaro, "Service delivery models for converged satellite-terrestrial 5G network deployment: A satellite-assisted CDN use-case," *IEEE Netw.*, vol. 33, no. 1, pp. 142–150, Feb. 2019.
- [31] Z. W. Hou, X. Q. Yi, Y. H. Zhang, Y. H. Y. Kuang, and Y. Zhao, "Satellite-ground link planning for LEO satellite navigation augmentation networks," *IEEE Access*, vol. 7, pp. 98715–98724, Jul. 2019.
- [32] S. Lv *et al.*, "Routing strategy of integrated satellite-terrestrial network based on hyperbolic geometry," *IEEE Access*, vol. 8, pp. 113003–113010, May 2020.
- [33] Q. Z. Guo *et al.*, "SDN-based end-to-end fragment-aware routing for elastic data flows in LEO satellite-terrestrial network," *IEEE Access*, vol. 7, pp. 396–410, Dec. 2019.
- [34] J. R. Zhang, S. B. Zhu, H. F. Bai, and C. Q. Li, "Optimization strategy to solve transmission interruption caused by satellite-ground link switching," *IEEE Access*, vol. 8, pp. 32975–32988, Feb. 2020.
- [35] N. Kato *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wirel. Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.
- [36] E. Wang, X. Lin, and S. Zhang, "Content placement based on utility function for satellite networks," *IEEE Access*, vol. 7, pp. 163150–163159, Nov. 2019.
- [37] K. An, Y. S. Li, X. J. Yan, and T. Liang, "On the performance of cache-enabled hybrid satellite-terrestrial relay networks," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1506–1509, Oct. 2019.
- [38] N. Garg, M. Sellathurai, and T. Ratnarajah, "In-network caching for hybrid satellite-terrestrial networks using deep reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 8797–8801.
- [39] S. P. Romano, M. Luglio, C. Roseti, and M. Zito, "The SHINE testbed for secure in-network caching in hybrid satellite-terrestrial networks," in *Proc. Eur. Conf. Netw. Commun.*, 2019, pp. 172–176.
- [40] P. Wang, X. Zhang, S. Zhang, H. Li, and T. Zhang, "Time-expanded graph-based resource allocation over the satellite networks," *IEEE Wirel. Commun. Lett.*, vol. 8, no. 2, pp. 360–363, Oct. 2019.
- [41] C. Q. Dai, L. F. Guo, S. Fu, and Q. B. Chen, "Contact plan design with directional space-time graph in two-layer space communication networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10862–10874, Dec. 2019.
- [42] Y. Li, Y. Wang, P. Yuan, Q. Zhang, and Z. Yang, "Popularity-aware back-tracing partition cooperative cache distribution for space-terrestrial integrated networks," *IET Commun.*, vol. 13, no. 17, pp. 2786–2796, Oct. 2019.



**Feng Wang** received the bachelor's degree in information and computing science from the University of Electronic Science and Technology of China. He is currently working toward the PhD degree in guidance, navigation, and control at the University of Electronic Science and Technology of China, Chengdu, China. His research interests include space-air-ground communication network and cyberspace security.



**Zhihan Lv** received the PhD degree from the University of Paris. He is currently an associate professor at Qingdao University, China. He was a research associate with University College London. He was a research engineer with CNRS, France, a postdoc research fellow with Umeå University, and experienced researcher with Fundación FIVAN, Spain. He was a Marie Curie fellow with European Union's Seventh Framework Program LANPERCEPT. He has authored or coauthored more than 200 papers in the related fields on journals including, the *IEEE Transaction on Industrial Informatics*, the *ACM Transactions on Multimedia Computing, Communications, and Applications*, and conferences including the ACM MM, ACM CHI, ACM Siggraph, ICCV, and IEEE Virtual Reality. His research interests include Internet of Things, multimedia, augmented reality, virtual reality, computer vision, 3D visualization and graphics, serious game, HCI, big data, and GIS. He is currently the guest editor of the *IEEE Transactions on Industrial Informatics*, the *Future Generation Computer Systems*, the *Neurocomputing*, and the *Neural Computing and Applications*.



**Shahid Mumtaz** (Senior Member, IEEE) received the master's and PhD degrees in electrical and electronic engineering from the Blekinge Institute of Technology, Sweden, and University of Aveiro, Portugal, in 2006 and 2011, respectively. He has more than ten years of wireless industry or academic experience and is currently a senior research scientist with Instituto de telecomunicações es Portugal. From 2005 to 2006, he was with Ericsson and Huawei, Research Labs in Sweden. He is currently a visiting professor with COM-

SATS Pakistan and a visiting researcher with Nokia Bell Labs, Belgium. He has several years of experience in 3GPP radio systems research with experience in HSPA/LTE/LTE-A and strong track-record in relevant technical field, especially physical layer technologies, LTE cell planning and optimization, protocol stack and system architecture. He has authored or coauthored four books and more than 150 publications in very high-rank IEEE transactions, journals, books, book chapters, and international conferences. His research interests include several aspects of wireless and vehicular communication, architectural enhancements to 3GPP networks including LTE-A user plan and control plan protocol stack, NAS and EPC, 5G related technologies, green communications, cognitive radio, cooperative networking, radio resource management, cross-layer design, backhaul/fronthaul, heterogeneous networks, LTE-U/WiFi, M2M and D2D communication, and baseband digital signal processing. He uses mathematical and system level tools to model and analyze emerging wireless communication architectures, leading to innovative and/or theoretically optimal new communication techniques. He is currently with leading R&D groups in the industry to transition these ideas to practice. His work is currently supported by Nokia and Huawei and Fundação para a Ciência e Tecnologia, Portugal. He has organized several workshops as the chair, Special Issues as the lead guest editor of the *IEEE Communication and Wireless Magazine*, and the *IEEE Transaction on Vehicular Technology*. He has given invited tutorials or talks in various IEEE conferences and has been invited to give lectures in different foreign universities. He is currently a scientific expert and evaluator of Research Funding Agencies, including the EU-COST, HEC-Pakistan, and Chinese NSF. In January 2017, he was nominated the chair of the IEEE new standardization on P1932.1: Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Networks. This standardization resulted from Dr. Mumtaz' novel idea on WiFi in the Licensed band. He is also involved in 3GPP standardization on LTE release 12 onwards, along with major manufacturers. He was the recipient of the "Alain Bensoussan" fellowship by ERCIM to pursue research in communication networks for one year at the VTT Technical Research Centre of Finland in 2012.



**Dingde Jiang** (Member, IEEE) received the PhD degree in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, China, in 2009. He is currently a professor at the School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research interests include network measurement, modeling and optimization, performance analysis, network management, network security in communication networks, particularly

in software-defined networks, information-centric networking, energy-efficient networks, and cognitive networks. His research is supported by the National Science Foundation of China and the Program for New Century Excellent Talents with the University of Ministry of Education of China. He was a technical program committee member for several international conferences. He is currently the editor of several international journals. He was the recipient of the best paper awards at several international conferences.



**Saba Al-Rubaye** (Senior Member, IEEE) received the PhD degree in electrical and electronic engineering from Brunel University London, U.K. She is currently a senior lecturer, a DRATeC fellow, and is leading connected system research group at the School of Aerospace, Transport and Manufacturing, Cranfield University, U.K. She is currently participating in developing industry standards by being an active voting member of IEEE P1920.2, Standard for Vehicle-to-Vehicle Communications for Unmanned Aircraft Systems, and of IEEE P1932.1 standard of License/licensed Interoperability. She has authored or co-authored many papers in IEEE journals and conferences. Her main research interests include, but not limited to UAV connectivity, communication networks, artificial intelligent, safety and security of autonomous vehicle. She has been a general co-chair, TPC co-chair, and has held other leading roles for many international conferences. She has organised and chair the 6G Network workshop in IEEEICCC2020. She is currently a chartered engineer (CEng), member of IET, Senior member of IEEE, and certified Unmanned Aircraft System Pilot. She was the recipient of the best technical paper award twice published in the *IEEE Vehicular Technology*, in 2011 and 2015, respectively.



**Octavia Dobre** (Fellow, IEEE) received the Diploma and PhD degrees from the Polytechnic Institute of Bucharest, Romania, in 1991 and 2000, respectively. Between 2002 and 2005, she was with the New Jersey Institute of Technology, USA. In 2005, she joined Memorial University, Canada, where she is currently a professor and research chair. She was a visiting professor with the Massachusetts Institute of Technology, USA, and Université de Bretagne Occidentale, France. She has coauthored more than 350 refereed

papers in her research areas. Her research interests include various wireless technologies, such as nonorthogonal multiple access and full duplex, optical and underwater communications, and machine learning for communications. She is currently the editor-in-chief of the IEEE Open Journal of the Communications Society. She is currently a fellow of the Engineering Institute of Canada. She was the editor-in-chief of the *IEEE Communications Letters*, senior editor, editor, and the guest editor of various prestigious journals and magazines. She was also the general chair, technical program co-chair, tutorial-chair, and technical co-chair of symposia at numerous conferences. She was a Fulbright scholar, Royal Society scholar, and distinguished lecturer of the IEEE Communications Society. She was the recipient of the best paper awards at various conferences, including IEEE ICC, IEEE Globecom, IEEE WCNC, and IEEE PIMRC.



**Antonios Tsourdos** received the MEng degree in electronic, control and systems engineering from the University of Sheffield in 1995, the MSc degree in systems engineering from Cardiff University in 1996, and the PhD degree in nonlinear robust flight control design and analysis from Cranfield University in 1999. He is currently a professor of autonomous systems and control at Cranfield University. He was appointed the head of the Autonomous Systems Group in 2007, head of the Centre of Autonomous and Cyber-Physical Systems in 2012, and the director of Research - Aerospace, Transport and Manufacturing in 2015. He was a member of the Team Stellar, the winning team for the U.K. MoD Grand Challenge (2008), and the IET Innovation Award (Category Team, 2009). He has been involved in a number of associated research projects including ASTRAEA, U-Space funded project EuroDrone, and DfT funded project UTM Open Access and EPSRC funded project CASCADE. He has authored or coauthored a number of papers on topics related to UTM including assured autonomy, sense and avoid, connectivity and networking UAS. Professor Tsourdos is chair of the IFAC Technical Committee on Aerospace Control, and a member of the U.K. Autonomous Systems National Technical Committee. He is currently the editorial board member of the *IEEE Transactions on Aerospace and Electronic Systems*, the Proceedings of the Institution of Mechanical Engineers, Part G: *Journal of Aerospace Engineering*, the *Aerospace Science and Technology*, the *International Journal of Systems Science*, and the *Journal of Intelligent and Robotic Systems*.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdl](http://www.computer.org/csdl).