Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8121509, 14 pages
https://doi.org/10.1155/2021/8121509

Research Article

A Collaborative Cache Strategy in Satellite-Ground Integrated Network Based on Multiaccess Edge Computing

Li Yang,[1] **Xiangguang Kong**,[2,3] **Yaowen Qi**,[1] **and Chengsheng Pan**[4]

[1] *School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *Key Laboratory of Communication Network and Information Processing, Dalian University, Dalian 116622, China*
[3] *College of Information Engineering, Dalian University, Dalian 116000, China*
[4] *Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Xiangguang Kong; kxglight@foxmail.com

Received 4 October 2021; Revised 4 November 2021; Accepted 9 November 2021; Published 3 December 2021

Academic Editor: Ming Yan

Multiaccess edge computing (MEC) provides users with a network environment and computing storage capacity at the edge of the network, ensuring a deterministic service with low delivery delay. This paper introduces a new satellite-ground integrated collaborative caching network architecture based on MEC and studies the caching strategy. On the ground side, the edge nodes (ENs) are deployed to the user side to form a hierarchical collaborative cache mode centered on the base station. On the satellite side, we utilize intelligent satellite ENs to precache and multicast the highly popular contents, reducing the initial content delivery delay. Under the constraints of the user demand and storage capacity, we study the deployment and cache scheme of ENs and establish the delivery delay minimization problem. To solve the problem, we propose a content update decision parameter for content cache update and transform the problem into improving the hit rate of ENs. Simulation results show that the proposed MEC network architecture and content caching scheme can increase the caching system hit rate to 64% and reduce the average delay by 32.96% at most.

1. Introduction

The number of mobile devices connected to the global fifth-generation mobile communication system (5G) will soon exceed billions level [1], which is far faster than the growth rate of the 4G era [2]. A wider range of intelligent devices (such as Internet of Things devices, 4K/8K video devices, and augmented reality and virtual reality devices) will continue to access the Internet [3], which will cause a large amount of traffic on the network access side. At the same time, 5G network services will put forward deterministic requirements on time delay and reliability [4]. The existing network architecture and content delivery methods will have difficulty meeting the ubiquitous and efficient network interconnection.

Multiaccess edge computing (MEC) will be an important technology to meet the requirements of 5G deterministic network services [5, 6]. Based on MEC technology, existing studies have carried out a large number of studies on edge collaborative caching [7, 8], collaborative offloading [9, 10], energy optimization [11–14], and other aspects. MEC realizes short-distance and real-time processing of user business and achieves a faster business response, effectively reducing the time delay. With the empowerment of MEC technology, edge computing and caching nodes are deployed on the user side, which not only provides users with faster and high hit rate content service responses [15] but also relieves the pressure on the cloud server and reduces the traffic on the backbone network [16]. At the same time, using the computing and storage capabilities of satellites, the deployment of MEC on satellites is considered to provide low delay and more ubiquitous network application services [17, 18].

However, the computing and storage resources in the edge servers are limited [19]. To provide a better internet experience, considering the content request traffic and content type of the served region, analyzing user request data, and perceiving user request preference information [20], the edge node (EN) deployment scheme and the collaboration scheme between them should be studied to reduce the average delivery delay and improve the request hit rate of users. For this reason, in [21], aiming to solve the problem of the collaborative cache mode of multicell scenarios on the base station side, the authors proposed a decentralized edge caching strategy via a multiagent framework with multiple actor networks and a single critic network. In [22], the authors proposed an edge network architecture based on software-defined network technology, which uses a neural network model to predict the content of user requests and precache user requests. An edge cooperative caching method based on delay and energy balance is proposed. However, the two above models will bring a large edge computing overhead to the edge server, and in the process of constantly updating model parameters, a large number of computing resources will be consumed.

The authors in [23] proposed a collaborative cache and resource allocation scheme among base station cache nodes and solved the collaborative cache problem with the goal of maximizing resource utilization. In [24], combining the cache of macrobase station and microbase station and considering the energy consumption, the minimization delay is associated with the energy consumption. Different user association strategies are proposed to minimize the delay while reducing the energy consumption. In [25, 26], the authors studied the collaborative caching scheme between users' terminals, but the collaborative sharing of storage and computing resources between terminals involves many beneficial distribution relations. The authors in [27] studied the collaborative caching scheme under the scenario of the Internet of Vehicles based on MEC technology and utilized the storage and computing resources of the Internet of Vehicles to enhance the edge caching and computing power. The authors in [28] studied cooperative cache contents in vehicles, base stations, and edge servers. In order to maximize the efficiency of content caching, reinforcement learning algorithms are used to optimize caching schemes on different devices to improve the cache hit ratio. However, in practical application scenarios, it is challenging to solve actual problems such as link disconnection and task switching caused by the high-speed movement of vehicles at all times. In [29], the authors proposed a double-layer service cache system of cloud-edge collaboration, designed a cache replacement algorithm according to the content popularity, and realized the efficient execution of tasks through the cooperation between edge nodes. However, the definition of prevalence is not sufficiently considered.

With the continuous evolution of 5G and the development of 5G-advanced and 6G network research, it has become the focus of research to use satellite and high-altitude communication platform capabilities to realize the ubiquitous global connectivity of the sky, ground, and sea [30, 31]. Therefore, to further reduce the end-to-end delay of users' content delivery [32, 33], it is necessary to fully explore the communication, storage, and computing capabilities of satellites and make use of satellite characteristics to meet more application scenarios of ground networks [34, 35]. In content storage and distribution, the satellite characteristics of wide-area connections will give more possibilities for network content delivery, which is expected to improve the quality of 5G-enhanced mobile broadband services [36]. For this reason, on the basis of studying the coordinated content distribution scheme of the ground dimension, satellite communication is used to expand the network spatial dimension.

Therefore, deploying MEC on satellite can provide lower delay and more ubiquitous network services [17, 37]. In the research of satellite-ground integrated cooperative networks based on MEC, the caching and computing nodes were deployed on the Iridium constellation in [38]. According to the characteristics of four intersatellite links for each satellite, a cooperative satellite-ground integrated network (SGIN) architecture with dual computing and caching capabilities on the satellite is proposed. In [39], the authors proposed a satellite-ground collaborative computing offloading model for edge computing at the user end, satellite end, and remote end and utilized dynamic network virtualization technology to integrate computing resources under the coverage of low-earth orbit satellites. However, this collaborative network model is difficult to apply to efficient edge caching applications. To solve the large delay caused by the long-distance transmission between the satellite and the ground, in [40], the authors added a fog computing edge node (EN) layer between the satellite and the ground base station to address different user services with different delay requirements. However, the deployment strategy of ENs below the base station is not considered, which cannot meet the higher delay requirements of 5G services. The authors in [41] proposed a time-domain separation satellite multicast and unicast cooperative transmission scheme. Multicast can satisfy the request of popular contents in most areas, while unicast can satisfy the request of unpopular contents in local areas. However, the storage resources of satellites are limited, and the caching of unpopular contents will cause a waste of resources and affect the efficient transmission of highly popular contents.

In the above studies on the ground collaborative caching scheme, the problem is mainly focused on the ENs on the base station side or below the base station. Above the base station, it directly communicates with the source server, without considering that there will still be a long communication distance between the base station and the source server, and there is still room for optimization of the delay. Besides, the author mainly focused on the formulation of cooperative caching strategy for local nodes and did not give sufficient consideration to the multilevel cooperative mode of caching nodes. In the study of satellite-ground cooperative cache, the main research is the collaboration between satellites, and the satellite-ground cooperative cache strategy between the satellite cache nodes and the ground cache nodes is not fully considered. Although these studies endowed satellite nodes with computing and storage capabilities, they failed to develop other possible potentials of satellite ENs, which restricted application scenarios of the satellite.

Therefore, this paper takes the base station as the center on the ground side, makes full use of the advantages of satellite communication in the space dimension, and proposes an edge node cache architecture of the satellite-ground integrated collaborative network. The main contributions of this paper are summarized as follows:

(1) We first propose a novel satellite-ground integrated collaborative cache network architecture with a network hierarchical cache mode centered on the base station and a large-scale precache mode of the cache node cluster through satellites

(2) We then propose a collaborative cache and content delivery scheme and build a communication model and a computing model to minimize the average delivery delay of the system. We prove that the problem model is nondeterministic polynomial hard (NP-hard)

(3) We study the content update and replacement strategy of cache nodes to solve the intractable NP-hard problem. To maximize the hit rate of the cache node, we design a content update decision parameter for content cache update

(4) We study the content caching and content distribution strategy of the collaborative network architecture, and we design a multilevel node content cocaching algorithm MNC^{3}A based on the content update decision parameter to hierarchically and coordinately cache network content and minimize the delivery delay. The time computing complexity of the algorithm is finally proven to be $O(n)$

The paper is organized as follows. Section 1 discusses the related works on the integrated network architecture and caching strategy. Section 2 presents the SGIN architecture and MEC server deployment scheme. In Section 3, we present the system model including communication mode and problem formulation. Section 4 introduces the algorithm to solve the caching problem. In Section 5, we compare the performance of our algorithms in different scenarios. Section 6 concludes the paper.

2. SGIN Architecture Empowered by MEC

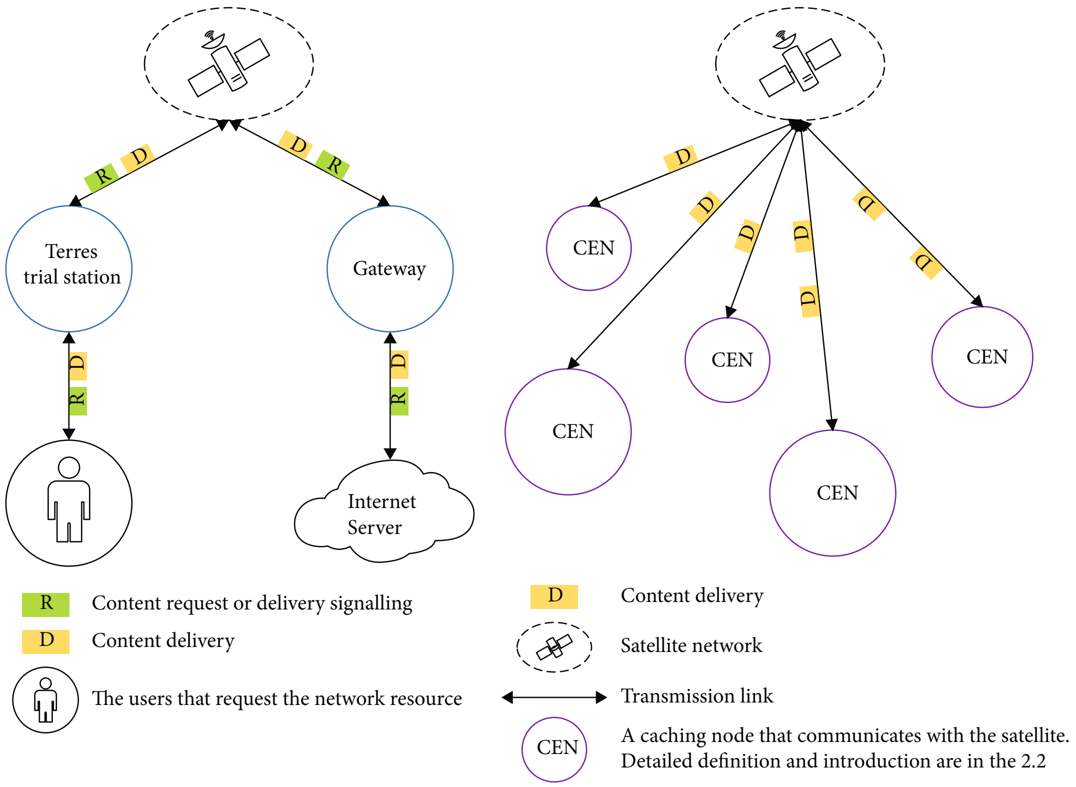
2.1. Satellite Network Communication Mode. With wide coverage and large connection characteristics, satellite networks can ignore terrain and regional interference and improve the user experience quality under specific application scenarios. This paper explores and utilizes the characteristics of satellite network communication from two aspects. First, the wide coverage of satellites is used to expand ground communication. User terminals can access the Internet through ground stations and satellite relays, as shown in Figure 1(a). When the ground network is unable to provide users with services or the service experience provided by the ground network is poor due to factors such as geographical location, communication distance, and natural disasters, the users will access the remote source server through satellite relay to obtain contents. Second, satellite communication assists the ground network by increasing the diversity of content caches, as shown in Figure 1(b). Due to the advantage of the satellite's large connection, the gateway communicates directly with the satellite, precaches regional hot contents through the satellite, and regularly broadcasts these contents to all gateways in the region covered by the satellite, realizing that hot contents in region A are cached in region B in advance. This saves the time that users in region B have to go to the source server to request contents when there is no precache, thus reducing the average delivery delay of the users.

2.2. Edge Node Deployment of SGIN. In this section, we introduce the hierarchical EN deployment design of SGIN based on MEC technology. This design makes full use of the advantages of computing resources, storage resources, wide coverage, and large connection characteristics of the satellite and solves the problems of insufficient coverage of ground resources, alleviating the load of ground links and reducing the delay of the first request and response of contents. As shown in Figure 2, under the catalysis of MEC technology, the traditional network cache nodes are sunk and split to form a hierarchical collaborative architecture including the user side EN (UEN) and base side EN (BEN), center EN (CEN) and internal EN (IEN) in the ground network, and satellite EN (SEN) in the satellite network. In the ground network, the microbase station on the user side is equipped with the computing storage edge server UEN, and nearby UENs can communicate with each other to exchange resources. All UEN contents are coordinated and managed by BEN. The IEN cluster is deployed on the metropolitan area network (MAN) side and is responsible for coordinating the caching of hot contents within certain areas. The CEN also takes the role of the autonomous gateway, which is responsible for the connectivity with the BEN and SEN and manages and dispatches the contents cached by the IEN.

Define the EN set $M = \{1, 2, 3, \cdots, m, \cdots, M\}$ and IEN set $\tilde{M} = \{1, 2, 3, \cdots, \tilde{m}, \cdots, \tilde{M}\}$. Network contents are cached, and tasks are processed through collaboration between multilevel MEC servers to improve user experience, ensure the requirements of deterministic timeliness, and increase the flexibility and stability of the network.

3. System Model

3.1. Related Variables. Define the content set $N = \{1, 2, 3, \cdots, n, \cdots, N\}$ and the content storage size set $S = \{s_1, s_2, \cdots, s_n, \cdots, s_N\}$. Define the ground cache edge node set $M = \{1, 2, 3, \cdots, m, \cdots, M\}$ and the UEN and IEN content cache 0-1 matrix $T^h = [T_1^h, T_2^h, \cdots, T_m^h, \cdots]$, where $T_m^h = [t_{1m}^h, t_{2m}^h, \cdots, t_{nm}^h, \cdots]$ and $h \in \{u, i\}$ represent the UEN and IEN, respectively. T^u is managed by BEN and T^i by CEN. Define SvEN, $v \in \{U, B, C, I, S\}$ represents the cache node storage capacity. That is, SvEN can be equal to SUEN, SBEN, SCEN, SIEN, and SSEN, which represents the storage capacity of UEN, BEN, CEN, IEN, and SEN, respectively. SNUM$_m$ represents the cached content number of cache node m.


R
D
D
R
Terres trial station
Gateway
D
R
D
R
Internet Server
R Content request or delivery signalling
D Content delivery
The users that request the network resource
D
D
D
D
D
D
D
CEN
CEN
CEN
CEN
CEN
CEN
D Content delivery
Satellite network
Transmission link
CEN A caching node that communicates with the satellite. Detailed definition and introduction are in the 2.2

(a) (b)

Figure 1: Two communication modes of a satellite network.

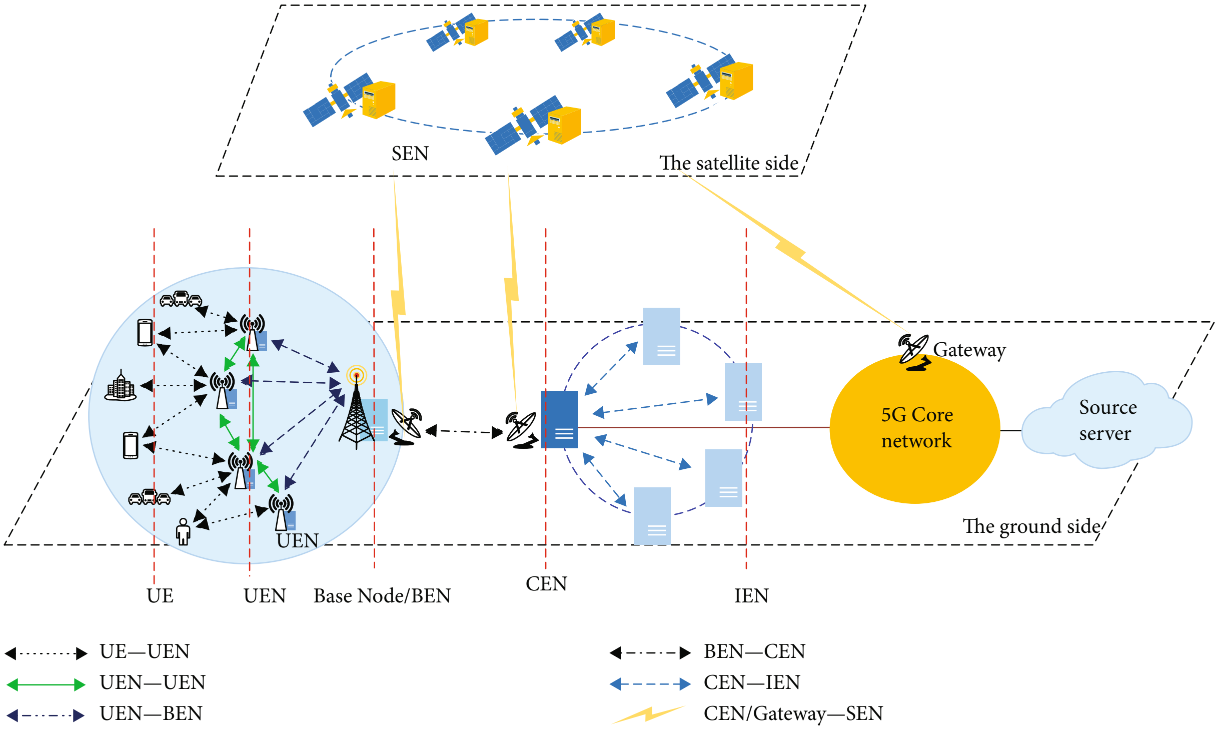


Figure 2: Architecture and edge node deployment of SGIN.

t^h_{nm} is defined by the following equation:

$$t^h_{nm} = \begin{cases} 1, & \text{if source } n \text{ in EN } m, \\ 0, & \text{others,} \end{cases} \tag{1}$$

when $t^u_{nm} = 1$, $t^i_{nm} = 0$.

Define the vector PM to record the cache priority of the cache node:

$$\mathrm{PM} = (\mathrm{PM}_1, \mathrm{PM}_2, \cdots, \mathrm{PM}_m, \cdots, \mathrm{PM}_M), \tag{2}$$

where PM_m represents the cache priority of node m.

The delivery delay should include transmission delay, propagation delay, processing delay, and queuing delay. The paper is aimed at verifying the content delivery delay minimization and edge node hit ratio maximization in multilayer converged network architecture. Processing delay and queuing delay depend on the node's computing and storage capacity and the current network load and network scheduling algorithm, which is beyond the scope of our research. We assume that the network status is good and the node's computing and storage capacity is enough. These two terms are set as constants that do not affect the final delivery delay, so they are not included in the final delay calculation model. Therefore, only transmission delay and propagation delay are considered [10]. For the calculation of propagation delay, the wireless transmission bandwidth between node m and user u is defined as d^u_m. The bandwidth between node $m1$ and node $m2$ is $l^{m2}_{m1}, \forall m \in M, \forall m1 \in M, \forall m2 \in M, m1 \neq m2$. Define B as the transmission bandwidth of the wired link and define D as the bandwidth of the satellite ground communication link.

3.2. Computation Model

3.2.1. Requested Content in UEN. The delivery delay in this case is

$$l_{(1)} = t^u_n \Delta_1, \tag{3}$$

where $\Delta_1 = s_n/d^u_m$, $t^u_n \epsilon \{0, 1\}$ indicates whether content n is in the UEN nearest to the user and can be delivered directly; if yes, $t^u_n = 1$.

3.2.2. Requested Content in BEN. In this case, BEN delivers the content to the user directly, and the delivery delay in this case is

$$l_{(2)} = (1 - t^u_n)(1 - t^u_{nm})\Delta_1. \tag{4}$$

3.2.3. Requested Content in Another UEN Connected to the Current UEN Directly. One or more of the UEN directly connected to the current UEN may cache the content requested by the user. In this case, BEN notifies all another UEN (namely, UEN-m) to send the content to the current UEN. The user selects the content reached first and discards the content reached later. Compared with $l_{(1)}$, the content delivery delay in this case increases the propagation delay from BEN to UEN-m and from UEN-m to UEN. The delivery delay in this case is

$$l_{(3)} = (1 - t^u_n)t^u_{nm}(2\Delta_2 + \Delta_1), \tag{5}$$

where $\Delta_2 = s_n/l^{m2}_{m1}$.

3.2.4. Requested Content in IEN. Compared with $l_{(2)}$, the content delivery delay in this case increases the propagation delay from the CEN to the BEN. The delivery delay in this case is

$$l_{(4)} = (1 - t^u_n)(1 - t^u_{nm})t^i_{nm}(\Delta_1 + \Delta_3), \tag{6}$$

where $\Delta_3 = s_n/B$.

3.2.5. Requested Content in SEN or Source Server. There are two cases when a user request passes through the satellite: if the content is cached in SEN, it is delivered directly; otherwise, it needs to be relayed by the satellite to the source server for the request. The delay in this case can be expressed as

$$\Delta = 2^c \Delta_4, \tag{7}$$

where $\Delta_4 = s_n/D, c \in \{0, 1\}$ represents the two cases of cached content and no cached content in SEN.

Users will select the response that arrives first from the two content acquisition methods of the ground backhaul link and satellite backhaul link and discard the response that arrives later, that is,

$$\Delta_5 = (\min(\Delta, (\Delta_1 + 2\Delta_3))). \tag{8}$$

The average delay in this case is

$$l_{(5)} = (1 - t^u_n)(1 - t^u_{nm})\left(1 - t^i_{nm}\right)\Delta_5. \tag{9}$$

3.3. Problem Formulation. Based on the above analysis, we propose the problem model of content delivery delay minimization in the hierarchical cache mode of SGIN. In Sections 3.2.1–3.2.5, content delivery delay in each case is listed, respectively, but the delay in the process of a content request can only be one of the above cases. Therefore, the time delays from 3.1 to 3.5 are added to make only one case meet the requirements through parameter control [42]. Therefore, the objective function of the delay calculation is established as

$$\begin{aligned} L(x) &= \sum_{n=1}^{N}\sum_{m=1}^{M} l_{(1)} + l_{(2)} + l_{(3)} + l_{(4)} + l_{(5)} = \sum_{n=1}^{N}\sum_{M=1}^{M}(1 - t^u_n) \\ &\quad \cdot \left(t^u_{nm}\frac{2s_n}{l^{m2}_{m1}} + (1 - t^u_{nm})(\Delta_5 + t^i_{nm}\left(\frac{s_n}{d^u_m} + \frac{s_n}{B} - \Delta_5\right)\right) + \frac{s_n}{d^u_m} \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}(1 - t^u_n)\left(2t^u_{nm}\Delta_2 + (1 - t^u_{nm})(\Delta_5 + t^i_{nm}(\Delta_1 + \Delta_3 - \Delta_5)\right) + \Delta_1 \\ &= \min\left\{\omega \cdot T_{n,m} \mid n \in N, m \in M\right\}, \end{aligned} \tag{10}$$

```
Input: content requested by the user. $L$, $F$.
Output: $\mathrm{FoHR}_n$, $\mathrm{FoRR}_n$, $\mathrm{ToL}_n$, $P_n$, PB.
Initialize: $\mathrm{FoHR}_n = 0$, $\mathrm{FoRR}_n = 0$, $\mathrm{ToL}_n = 0$.
for each requested $n \in N$
        $\mathrm{FoRR}_n + = 1$;
        $\mathrm{ToL}_n = L$;
        for each $\bar{n} \in N \&\& \bar{n} ! = n$
                $\mathrm{ToL}_{\bar{n}} - = 1$;
                if ($\mathrm{ToL}_{\bar{n}} == 0$)
                $P_n = \tau \mathrm{FoHR}_n + (1 - \tau)\mathrm{FoRR}_n$;
                if ($P_n < F$)
                        $\mathrm{PB} = P_n\%100/100$;
                        Que.push($\bar{n}$) by $(1 - PB)$‖Que.delete($\bar{n}$) by PB;
            else
                        $\mathrm{ToL}_{\bar{n}} = L$;
                        $\mathrm{FoHR}_{\bar{n}} = \mathrm{FoHR}_{\bar{n}} + \mathrm{FoRR}_{\bar{n}}$;
                        $\mathrm{FoRR}_{\bar{n}} = 0$;
            end
                end
        end for
end for
```

Algorithm 1: Content update algorithm LF.

```
Input: content request tasks that obey Zipf distribution.
Output: content response and caching strategy.
for each requested $n \in N$
/*requesting $n$ */
        if ($t_n^u == 1$)
                executing Algorithm 1 and response to user directly;
        else if ($t_n^u == 0 \&\& t_{nm}^u == 0$)
                executing Algorithm 1 and response to user via BEN;
        else if ($t_n^u == 0 \&\& t_{nm}^u == 1$)
                executing Algorithm 1 and response to user via UEN-m;
        else if ($t_n^u == 0 \&\& t_{nm}^u == 0 \&\& t_{nm}^u == 1$)
                executing Algorithm 1 and response to user via IEN;
            Else
                Requesting from source server;
/*caching $n$ */
        while ($\sum_{i=1}^{\mathrm{SNUM}} s_i + s_n > \mathrm{SvEN}$) $v \in \{U, B, C, I, S\}$
        Que.delete (Minimum_Memory_Size ($i$));
        end while
        caching $n$ in IEN with mini ($\mathrm{PM}_m$);
        executing Algorithm 1;
        caching n in vEN $v \in \{U, B, C, S\}$;
        executing Algorithm 1;
end for
for each $n \in N$ in IEN
        find several most popular content and transmit to BEN and SEN;
end for
broadcast content to CEN via satellite and caching in IEN;
executing Algorithm 1;
```

Algorithm 2: $\mathrm{MNC}^3\mathrm{A}$.

where $T_{n,m}$ is the delivery delay of content n cached in node m, ω is the indicator parameter, and $\omega \in \{0, 1\}$. If $\omega = 1$, the content is already cached in the node; otherwise, it is not cached. $\omega \bullet T_{m,n}$ means that the content requested by the user can only be delivered by the only edge node with the smallest delay. The average delay optimization problem of the system is formulated as follows:

$$P1 : \min_{\text{x}} L(x), \tag{11}$$

$$s.t. \sum_{n=1}^{N} t_{nm}^{\text{v}} s_n \leq \text{SvEN}, \forall n \in N, \forall m \in M, t_{nm}^{v} \in (0, 1), \tag{12}$$

where (11) is the target problem to be optimized and (12) is the caching storage constraint and ENs caching indicator parameter.

By reducing problem $P1$ to the generalized assignment problem, since the generalized assignment problem is an NP-hard problem, it can be proven that problem $P1$ is an NP-hard problem [43]. $P1$ is a problem model based on delay minimization. From the perspective of cache nodes, $P1$ is the problem of content delivery delay minimization. The content delivery of the cache node depends on what content the cache node caches. Therefore, the node cache optimization problem is a subproblem of the content delivery optimization problem. Caching N contents on M nodes and minimizing the time delay under this caching scheme are a classic assignment problem, so optimization problem $P1$ is a NP-hard problem.

4. Algorithm Design

4.1. Problem Description. When the delay of system content delivery decreases, it means that the content requested by the users is gradually cached in the edge cache node close to the user; that is, the cache node close to the user has a higher hit rate. Therefore, solving problem $P1$ is simplified to solving the problem of maximizing the cache hit rate of the user's edge node.

Considering the contents that are accessed cyclically and alternately, for the content update strategy of the cache node, the historical request number of content n is defined as FoHR_n, the recent request number is FoRR_n, and the lifetime is ToL_n to lay down the content update decision, where the update rules of FoHR_n and FoRR_n are as follows:

$$\begin{aligned} \text{FoHR}_n &= \text{FoHR}_n + \text{FoRR}_n, \\ \text{FoRR}_n &= 0. \end{aligned} \tag{13}$$

The update rules of FoHR_n and the relationship among FoHR_n, FoRR_n, and ToL_n are described as follows.

For each cache or update, FoRR_n of the requested content n is incremented by 1, the lifetime ToL_n is set to $L(L \in R^+)$, and the lifetime of all other content is reduced by 1. When the lifetime of a certain content becomes zero, the content update decision variable P_n is considered:

$$P_n = \tau \text{FoHR}_n + (1 - \tau)\text{FoRR}_n. \tag{14}$$

$\tau \in (0,0.5)$ is the weighted parameter used to control the influence of the historical and recent request numbers on the updating decision of contents. The value range of τ indicates that a single content update decision is greatly influenced by its recent request number. At the same time, FoHR_n and FoRR_n are update. If P_n is less than a threshold F, the content is retained with PB probability and deleted with $1 -$ PB probability. Otherwise, the content lifetime ToL is reset to L. The PB definition is as follows.

Table 1: The simulation parameters.

Parameter	Value
Content number, N	30000
Propagation rate, c (m/s)	3×10^8
LEO height (km)	700
β	0.2
γ	0.5
μ	0.3

$$\text{PB} = \frac{P_n\%100}{100}. \tag{15}$$

When content n is retained with a probability of $1 -$ PB, the content is placed in the waiting queue, Que (in this paper, the waiting queue is defined as Que for holding inactive content). If the content is requested again, the values of FoHR_n, FoRR_n, and ToL_n are reset. When the cache node is unable to continue caching contents due to insufficient memory, the inactive content is replaced from e, in the order of content storage size from small to large.

Through the above analysis, the NP-hard problem $P1$ that is difficult to solve is transformed into a problem $P2$ that maximizes the cache hit rate of the underlying cache node UEN with the content update decision determined by L and F.

$$P2 : \max_{L,F} f(L, F) = \frac{\sum_{i=1}^{U} \partial_i}{\text{UT}}, \tag{16}$$

$$s.t. \sum_{n=1}^{N} t_{nm}^{u} s_n \leq \text{SUEN}, \forall n \in N, \tag{17}$$

where $f(L, F)$ is the UEN hit rate expression under the content update decision affected by the content request number and content lifetime. UT is the total number of content requests, and $\partial_i \in \{0, 1\}$ indicates whether the requested content is in the UEN. If the request is hit, $\partial_i = 1$; otherwise, $\partial_i = 0$. (17) is the cache node storage resource constraint. The optimization problem $P2$ can be regarded as a binary optimization problem concerning variables L and F.

According to the above analysis, the content update algorithm, namely, LF, is described as Algorithm 1.

4.2. Multilevel Node Content Cocaching Algorithm. The establishment of a cache strategy should consider the joint constraints of the cache nodes' own storage capacity, busy degree, load balance, and content update decision to minimize the average delay of the system and maximize the hit rate. To ensure the load balance of the IEN server, considering the total number of existing contents of ENs in the

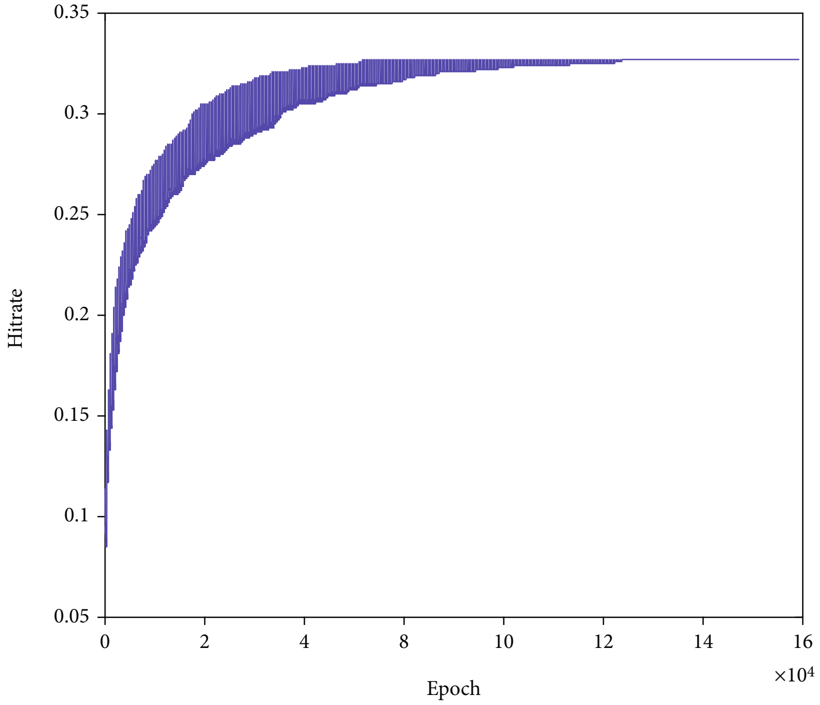


Figure 3: Process to solve L and F.

cluster, the storage resources occupied by these contents and the current busyness of the EN (the number of service requests), the storage priority PM_m of cache node M is defined as follows:

$$\mathrm{PM}_m = \beta \mathrm{SNUM}_m + \gamma \sum_{n=1}^{\mathrm{SNUM}} s_n + \mu \sum_{n=1}^{\mathrm{SNUM}} \mathrm{FoRR}_n, \qquad (18)$$

where $\beta + \gamma + \mu = 1, \beta \in (0,1), \gamma \in (0,1), \text{and}\, \mu \in (0,1)$. In this paper, we stipulate that the lower the PM value is, the higher the priority of the cache node is selected for content caching.

When the content update occurs in UEN and IEN, the t_{nm}^h corresponding position should be set to 0 after the update. At the same time, in order to ensure that contents in BEN server are the most popular in the region, it is necessary to check contents cached by BEN regularly and replace the low popular contents in BEN with the high popular contents in IEN through CEN. For SEN, considering the hot contents from local areas to ubiquitous areas with radioactive transmission characteristics, highly popular contents in certain areas for a certain period of time may also be popular in other areas, so highly popular contents in IEN are chosen to be transmitted to SEN by CEN. SEN will broadcast to all CEN clusters in the covered area on a regular basis to realize advanced caching and active pushing of highly popular contents. It reduces the initial request delay of contents and then reduces the average delivery delay of the system as a whole.

Based on the above analysis, we propose the content caching and delivery algorithm MNC^{3}A of multilevel collaborative caching nodes. The implementation of the algorithm is described as Algorithm 2.

4.3. Time Complexity Analysis. The time complexity of Algorithm 2 is on the $O(n)$ level. The proof is as follows.

Since Algorithm 2 executes Algorithm 1 internally multiple times, we first analyze the time complexity of Algorithm 1. In a single time slice of content caching and update, FORR, FOHR, ToL, and other parameters of all other n contents need to be updated. Therefore, the time complexity of Algorithm 1 is $O(n)$ in each content request process. Meanwhile, in each content delivery and caching process, Algorithm 2 needs to execute Algorithm 1 multiple times, so the time complexity of Algorithm 2 is $O(kn), k \in R^+$. In conclusion, the time complexity of Algorithm 2 is approximately on the $O(n)$ level.

5. Simulation Analysis

In this paper, we use MATLAB tools for simulation analysis. Based on the proposed edge network architecture and collaborative caching scheme of SGIN, first, we solve the value of L and F. Second, the benefits of the content update algorithm LF are verified by comparing with the traditional algorithms least recently used (LRU) [44] and least frequently used (LFU) [45]. Second, we determine the best way to deploy edge cache nodes under different traffic conditions in different regions and verify the benefits of deploying hierarchical cache nodes. Then, the MNC^{3}A algorithm proposed in this paper is compared with other verified algorithms in the literature. Finally, the effect of reducing the average delay by satellite precaching is verified. The relevant parameters of the simulation are described in Table 1, and the network contents follow a Zipf distribution.

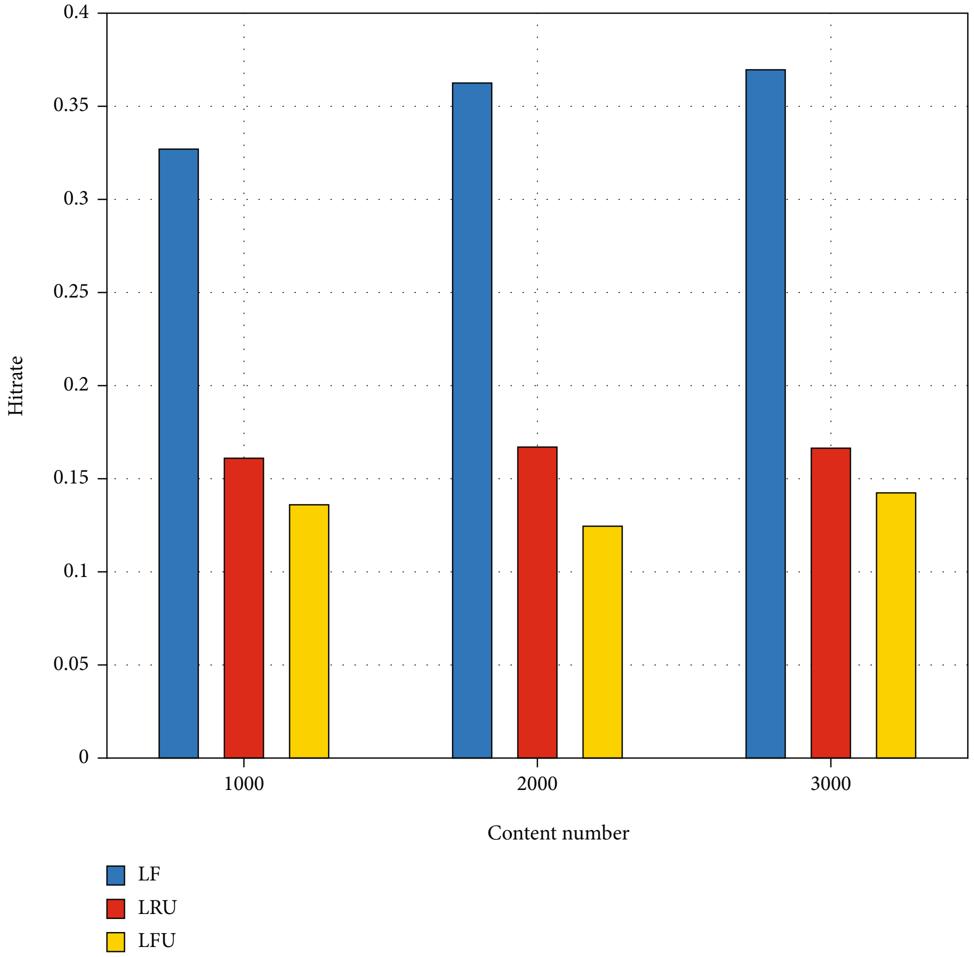


Figure 4: Hit rate of different algorithms for a single cache node.

5.1. Solving Parameters L and F. In this section, we will solve the key parameters L and F proposed in Section 4, which determine the content update strategy.

As shown in Figure 3, we find that the curve shows a gradual upward trend and then fluctuates around a maximum value. Finally, it converges to the maximum and the maximum cache hit rate of a single node are 0.327. Therefore, there are many pairs of optimal L and F values. We choose the point (149,0.1,0.327), that is, $L = 149$ and $F = 0.1$. The values will be used in the following simulations.

5.2. Content Update Algorithm Hit Rate Comparison. This section compares the hit rate of the LF content update algorithm proposed in Section 4 with that of the LFU algorithm and LRU algorithm. We compare the hit rate of a single cache node with 1000, 2000, and 3000 contents that obey the Zipf distribution. Figure 4 shows that the proposed cache update algorithm has a higher hit rate performance than the traditional algorithm.

5.3. The Effect of the Number of IEN on the Delay. Regardless of the connections between users and UEN, BEN, between BEN and CEN, and between CEN and SEN, this section only discusses the scenarios in which CEN requests contents from the source server and the collaborative relationship between CEN and IEN. The goal is to find the most appropriate number of IENs for a particular scenario. Each set of data is the average of 20 simulations. As shown in Figure 5, with the increase in the number of IENs, the average delay curves of Figures 5(a)–5(d) all show a continuous decline at first and then fluctuate around the lowest point after falling to a certain extent. This means that when the number of IENs is too small, the system cannot cache too much content, so that it has to request contents from the source server frequently, which will consume considerable time in the continuous content update process. As the number of IENs continues to increase, the delay decreases and tends to a constant value. With different numbers of tasks, the fluctuation starts from different starting points. The more tasks there are, the longer the curve continues to decline. When the number of tasks is $N = 10000$, the delay starts to fluctuate when the number of IENs is 3, while when the number of tasks is $N = 30000$, the delay curve starts to fluctuate when the number of IENs is 9. It can be concluded that the number of IENs should be reasonably planned according to the number of tasks in certain geographical regions to avoid problems such as insufficient storage resources and unconspicuous delay reduction caused by too little IEN and

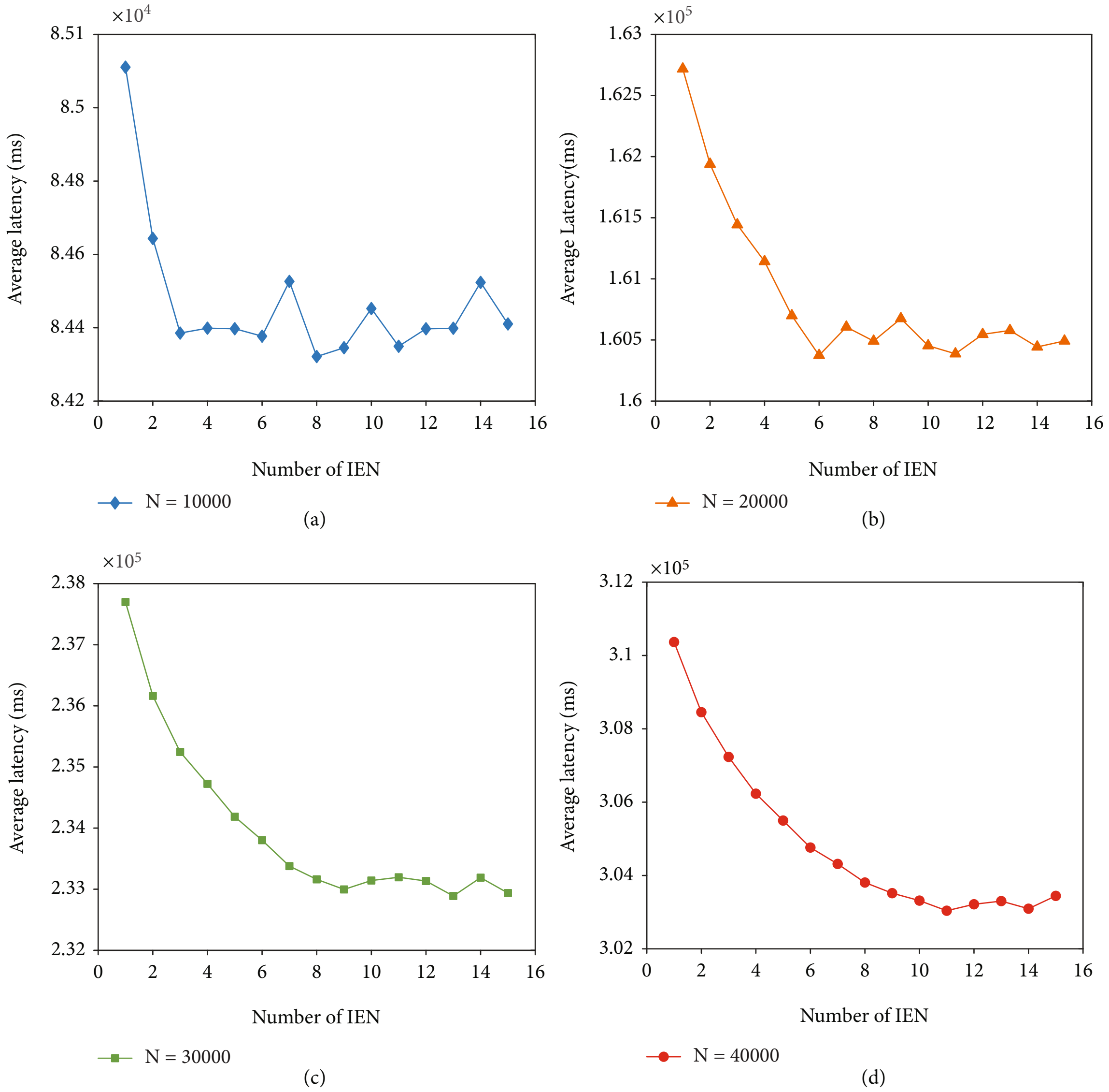


Figure 5: The effect of the number of IENs on the delay under different content numbers.

problems such as increased construction cost, management difficulties, and resource consumption caused by too much IEN.

5.4. The Effect of Cache Node on Delay on the MAN Side.

In this simulation section, the benefits of deploying cache nodes on the MAN side are verified, as shown in Figure 6. Obviously, with the center of the base station and two layers cooperative cache mode, the delay performance is superior to the single base station cooperative caching mode. As the number of user content requests increases, the two modes of the delay gap gradually increase because in the process of users constantly requesting network content, IEN is rich in contents, and the user request preferences are gradually met in the area. The average delivery delay is further reduced.

5.5. Comparison of Hit Rates of Different Caching Algorithms.

This section compares the cache hit rate between the cache strategy proposed in this article, the UCC cache scheme, and the cLRU(m) cache scheme proposed in [23], as shown in Figure 7.

Under the proposed collaborative caching strategy MN C^3A, when the base station directly obtains content from the source server, the cache hit rate is 45%, which is better than 35% for cRLU(m). However, in the deployment of the CEN+IEN collaborative architecture on the base station, the cache hit rate of MNC^{3}A finally reaches approximately 64%, which is better than those of 55% for UCC and 35% for cRLU. The average delivery time under a high hit rate is significantly reduced, and the user's QoS is improved, which proves the benefits of the hierarchical collaborative caching strategy proposed in this paper.

5.6. The Effect of Satellite Precache on Delay.

The simulation in this section will study the impact of satellite precaching of hot content in a certain area on the average delivery delay in other areas. In the multiple IEN and CEN cluster areas

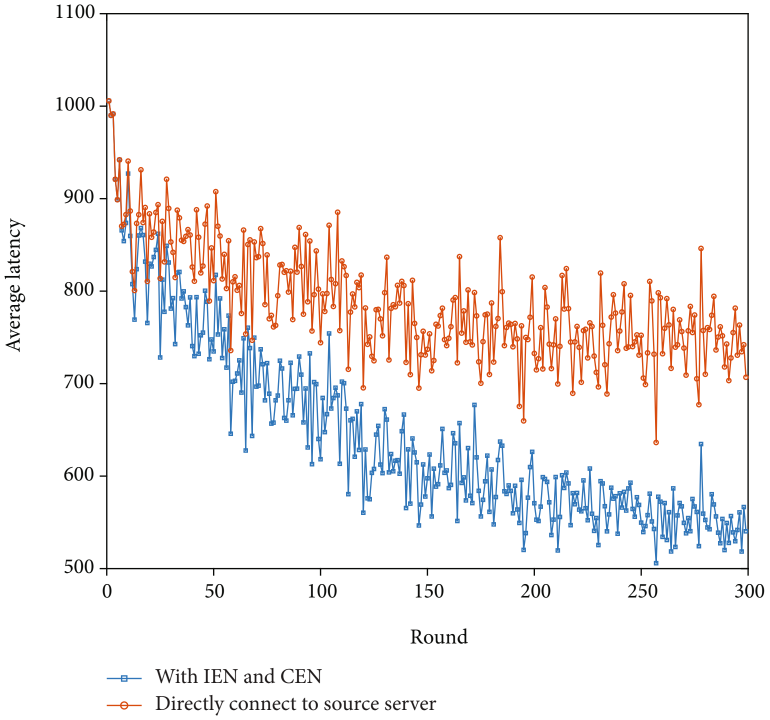


FIGURE 6: Impact of IEN and CEN multilevel collaboration on delay.

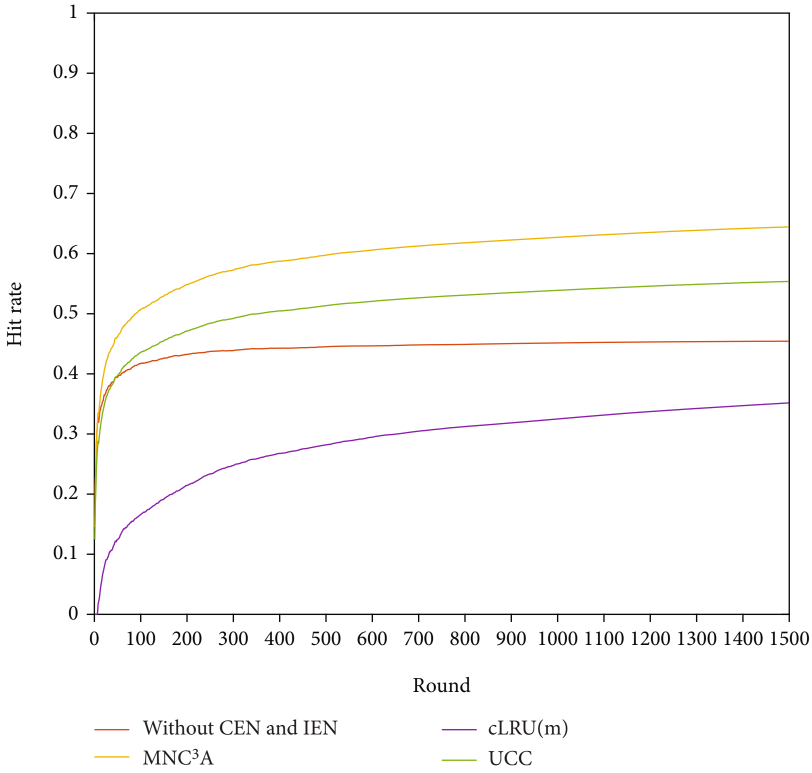


FIGURE 7: Hit rate comparison of different algorithms.

under satellite coverage, the most popular contents in area A are regularly cached by SEN and broadcast to other areas except A through satellites to achieve the precaching and active pushing of contents. By reducing the user's first request delay for a certain content, the average network delay can be further reduced. Figure 8 is a simulation

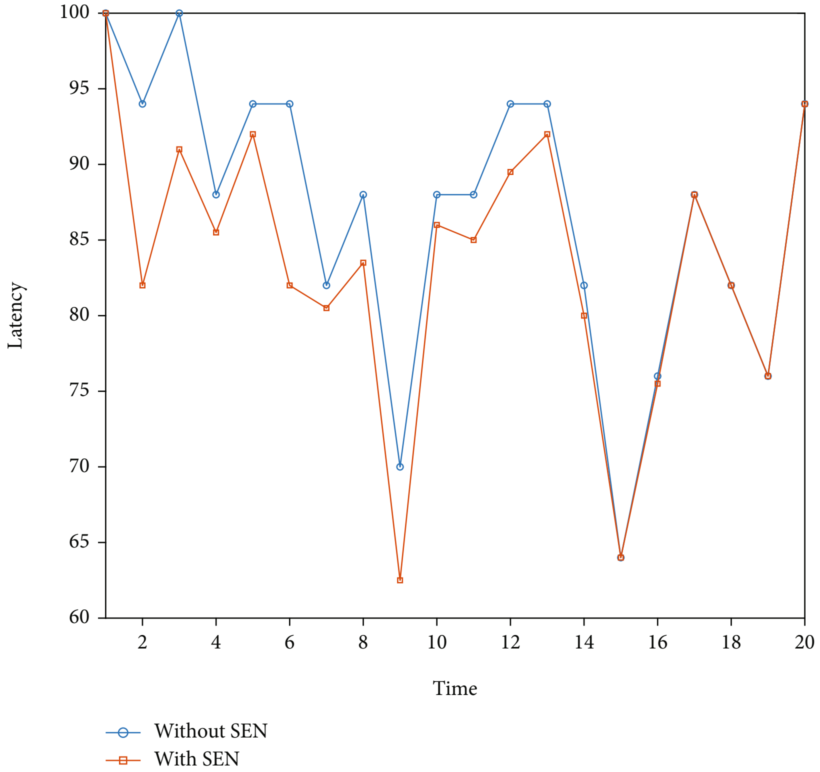


Figure 8: The effect of satellite on the delay.

comparison between satellite precaching and broadcast contents and only through terrestrial communication. As shown in Figure 8, in the first half of the time, the average delay of users in the satellite precaching mode is significantly lower than that in the case of nonsatellite precaching. This is because the satellite precaches the hot content when it is idle and broadcasts it to other areas under the satellite's coverage, saving users the time to go to the source server to request contents when there is no precaching. In the second half of Figure 8, when the content of satellite precaching is gradually cached in the IEN following the user's request, the average delay approaches gradually or even coincides.

6. Conclusion

Based on future network development trends, this article proposes a satellite-ground integrated network cache architecture based on multiaccess edge computing technology. On this basis, we establish a network content delivery delay minimization problem model and transform the problem into solving the problem of maximizing the hit rate of edge nodes. We study the collaborative cache strategy of multilevel cache nodes by finding the key decision parameters that determine the caching content update and replacement strategy. Simulation analysis shows that the proposed satellite-ground integrated edge network caching architecture and content caching strategy have a higher hit rate and significantly reduce the average delivery delay. In future work, the following aspect is still worth further study. The cache architecture studied in this paper is a static network architecture under a single time chip, but both 5G terminals and satellite nodes have strong mobility. It may be possible to consider the impact of nodes mobility on the entire cache system, and we can formulate the caching strategy under the dynamic network architecture.

Data Availability

The simulation data (media data subject to ZIPF distribution) used to support the findings of this study are included within the article. More details are available from the authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants U21B2003 and 61722105.

References

[1] G. Forecast, *Cisco visual networking index: global mobile data traffic forecast update, 2017–2022. Update*, vol. 2017, p. 2022, 2019.

[2] M. Yan, C. A. Chan, W. Li et al., "Network energy consumption assessment of conventional mobile services and over-the-top instant messaging applications," *IEEE Journal on*

Selected Areas in Communications, vol. 34, no. 12, pp. 3168–3180, 2016.

[3] M. Yan, C. A. Chan, W. Li, L. Lei, A. F. Gygax, and C. L. I, “Assessing the energy consumption of proactive mobile edge caching in wireless networks,” *IEEE Access*, vol. 7, pp. 104394–104404, 2019.

[4] T. Huang, S. Wang, Y. D. Huang, Y. Deng, J. Liu, and Y. J. Liu, “Survey of the deterministic network,” *Journal on Communications*, vol. 40, pp. 160–176, 2019.

[5] J. Andrews, S. Buzzi, W. Choi et al., “What will 5G be?,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[6] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing: a key technology to-wards 5g,” *ETSI*, vol. 11, no. 11, pp. 1–16, 2015.

[7] R. Chen, H. Lu, and P. Ma, “User-centric cooperative MEC service offloading,” in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Nanjing, China, March 2021.

[8] T. M. Ayenew, D. Xenakis, N. Passas, and L. Merakos, “Cooperative content caching in MEC-enabled heterogeneous cellular networks,” *IEEE Access*, vol. 9, pp. 98883–98903, 2021.

[9] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, “Satellite-terrestrial integrated edge com-puting networks: architecture, challenges, and open issues,” *IEEE Network*, vol. 34, no. 3, pp. 224–231, 2020.

[10] S. Zhong, S. Guo, H. Yu, and Q. Wang, “Cooperative service caching and computation offloading in multi-access edge computing,” *Computer Networks*, vol. 189, p. 107916, 2021.

[11] C. Li, P. Liu, C. Zou, F. Sun, J. M. Cioffi, and L. Yang, “Spectral-efficient cellular communications with coexistent one- and two-hop transmissions,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6765–6772, 2016.

[12] C. Li, Y. Li, K. Song, and L. Yang, “Energy efficient design for multiuser downlink energy and uplink information transfer in 5G,” *Science China Information Sciences*, vol. 59, no. 2, pp. 1–8, 2016.

[13] C. Li, S. Zhang, P. Liu, F. Sun, J. M. Cioffi, and L. Yang, “Overhearing protocol design exploiting intercell interference in cooperative green networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 441–446, 2016.

[14] M. Yan, H. Yuan, Z. Li, Q. Lin, and J. Li, “Energy savings of wireless communication networks based on mobile user environmental prediction,” *Journal of Environmental Protection and Ecology*, vol. 22, no. 1, pp. 206–217, 2021.

[15] X. Chen, L. He, S. Xu, S. Hu, Q. Li, and G. Liu, “Hit ratio driven mobile edge caching scheme for video on demand services,” in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1702–1707, Shanghai, China, July 2019.

[16] J. Sung, M. Kim, K. Lim, and J. K. K. Rhee, “Efficient cache placement strategy in two-tier wireless content delivery network,” *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1163–1174, 2016.

[17] J. Wei, J. Han, and S. Cao, “Satellite IoT edge intelligent computing: a research on architecture,” *Electronics*, vol. 8, no. 11, p. 1247, 2019.

[18] Z. Zhang, W. Zhang, and F. H. Tseng, “Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques,” *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.

[19] Y. Hao, M. Chen, L. Hu, M. S. Hossain, and A. Ghoneim, “Energy efficient task caching and offloading for mobile edge computing,” *IEEE Access*, vol. 6, pp. 11365–11373, 2018.

[20] M. Yan, S. Li, C. A. Chan, Y. Shen, and Y. Yu, “Mobility prediction using a weighted Markov model based on mobile user classification,” *Sensors*, vol. 21, no. 5, p. 1740, 2021.

[21] C. Zhong, M. C. Gursoy, and S. Velipasalar, “Deep reinforcement learning-based edge caching in wireless networks,” *IEEE Transactions on Cognitive Communi-cations and Networking*, vol. 6, no. 1, pp. 48–61, 2020.

[22] C. Li, Y. Zhang, Q. Sun, and Y. Luo, “Collaborative caching strategy based on optimization of latency and energy consumption in MEC,” *Knowledge-Based Systems*, vol. 233, p. 107523, 2021.

[23] A. Ndikumana, S. Ullah, T. LeAnh, N. H. Tran, and C. S. Hong, “Collaborative cache allocation and computation offloading in mobile edge computing,” in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 366–369, Seoul, Korea (South), September 2017.

[24] G. Vallero, M. Deruyck, M. Meo, and W. Joseph, “Base station switching and edge caching optimisation in high energy-efficiency wireless access network,” *Computer Networks*, vol. 192, p. 108100, 2021.

[25] C. Ge and K. Xu, “A hierarchical cooperative caching strategy for mobile content delivery network,” *Chinese Journal of Computers*, vol. 41, pp. 2769–2786, 2018.

[26] X. Huang, L. He, X. Chen, G. Liu, and F. Li, “A more refined mobile edge cache replacement scheme for adaptive video streaming with mutual cooperation in multi-MEC servers,” in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, London, UK, July 2020.

[27] W. Li, B. Zhang, and X. Wang, “V2X collaborative caching and resource allocation in MEC-based IoV,” *Journal of Communications*, vol. 42, pp. 1–14, 2021.

[28] L. Zhao, H. Li, N. Lin, M. Lin, C. Fan, and J. Shi, “Intelligent content caching strategy in autonomous driving toward 6G,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[29] X. Wu, Y. Wang, S. Yuan, and P. Xie, “A service caching method for mobile edge computing,” *Journal of Physics: Conference Series*, vol. 2005, no. 1, p. 012060, 2021.

[30] M. Yan, H. Yuan, J. Xu, Y. Yu, and L. Jin, “Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm,” *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, 23 pages, 2021.

[31] Z. Qu, G. Zhang, H. Cao, and J. Xie, “LEO satellite constellation for internet of things,” *IEEE Access*, vol. 5, pp. 18391–18401, 2017.

[32] O. Kodheli, A. Guidotti, and A. Vanelli-Coralli, “Integration of satellites in 5G through LEO constellations,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, December 2017.

[33] L. Boero, R. Bruschi, F. Davoli, M. Marchese, and F. Patrone, “Satellite networking integration in the 5G eco-system: research trends and open challenges,” *IEEE Network*, vol. 32, no. 5, pp. 9–15, 2018.

[34] A. Guidotti, A. Vanelli-Coralli, M. Conti et al., “Architectures and key technical challenges for 5G systems incorporating

satellites," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2624–2639, 2019.

[35] J. Zhang, X. Zhang, P. Wang, L. Liu, and Y. Wang, "Double-edge intelligent integrated satellite terrestrial net-works," *China Communications*, vol. 17, no. 9, pp. 128–146, 2020.

[36] T. de Cola and I. Bisio, "QoS optimisation of eMBB services in converged 5G-satellite networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12098–12110, 2020.

[37] S. Kota and G. Giambene, "6G Integrated Non-Terrestrial Networks: Emerging Technologies and Challenges," *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2021.

[38] Y. Wang, J. Zhang, X. Zhang, P. Wang, and L. Liu, "A computation offloading strategy in satellite terrestrial networks with double edge computing," in *2018 IEEE International Conference on Communication Systems (ICCS)*, pp. 450–455, Chengdu, China, December 2018.

[39] Z. Zhang, W. Zhang, and F. H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satel-lite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.

[40] C. Suzhi, W. Junyong, H. Hao et al., "Space edge cloud enabling network slicing for 5G satellite network," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 787–792, Tangier, Morocco, June 2019.

[41] L. Liu, J. Zhang, X. Zhang, P. Wang, Y. Wang, and L. Ouyang, "Design and analysis of cooperative multicast-unicast transmission scheme in hybrid satellite-terrestrial networks," in *2018 IEEE International Conference on Communication Systems (ICCS)*, pp. 309–314, Chengdu, China, December 2018.

[42] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "A novel mobile edge network architecture with joint caching-delivering and horizontal cooperation," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 19–31, 2021.

[43] S. Luo, X. Chen, and Z. Zhou, "F3C: fog-enabled joint computation, communication and caching resource sharing for energy-efficient IoT data stream pro-cessing," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1019–1028, Dallas, TX, USA, July 2019.

[44] F. Kurniawan, L. Yovita, and T. Wibowo, "Modified-LRU algorithm for caching on named data network," in *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 438–443, Bandung, Indonesia, July 2019.

[45] P. Aimtongkham, C. So-In, and S. Sanguanpong, "A novel web caching scheme using hybrid least frequently used and support vector machine," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–6, Khon Kaen, Thailand, July 2016.