

## Research Article

# An Intelligent Caching and Replacement Strategy Based on Cache Profit Model for Space-Ground Integrated Network

Li Yang <sup>1</sup>, Cheng Chi <sup>2,3</sup>, Chengsheng Pan <sup>4</sup> and Yaowen Qi <sup>1</sup>

<sup>1</sup>Automation College, Nanjing University of Science and Technology, Nanjing 210094, China

<sup>2</sup>Key Laboratory of Communication Network and Information Processing, Dalian University, Dalian 116000, China

<sup>3</sup>College of Information Engineering, Dalian University, Dalian 116000, China

<sup>4</sup>Nanjing University of Information Science & Technology, Nanjing 210094, China

Correspondence should be addressed to Cheng Chi; 379483223@qq.com

Received 12 August 2021; Accepted 7 October 2021; Published 19 October 2021

Academic Editor: Sang-Bing Tsai

Copyright © 2021 Li Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Compared with the stable states of the ground networks, the space-ground integrated networks (SGIN) have limited resources, high transmission delay, and vulnerable topology, which make traditional caching strategies unable to adapt to the complex space network environment. An intelligent and efficient caching strategy is required to improve the edge service capabilities of satellites. Therefore, we investigate these problems in this paper and make the following contributions. First, the content value evaluation model based on classification and regression tree is proposed to solve the problem of “what to cache” by describing the cache value of content, which considers the multidimensional content characteristics. Second, we propose a cache decision strategy based on the node caching cost model to answer “where to cache.” This strategy modified the genetic algorithm to adapt the 0-1 knapsack problem under SDN architecture, which greatly improved the cache hit rate and the network service quality. Finally, we propose a cache replacement strategy by establishing an effective service time model between the satellite and ground transmission link, which solves the problem of “when to replace.” Numerical results demonstrate that the proposed strategy in SGIN can improve the nodes’ cache hit rate and reduce the network transmission delay and transmission hops.

## 1. Introduction

In recent years, the space-ground integrated network (SGIN) has attracted much attention by its broader coverage and higher communication ability. With the improvement of satellite service capabilities, the coordinated transmission of data in SGIN will be the trend of the future network [1]. SGIN can supplement the extensive communication services, providing wider coverage area and higher reliable data transmission schemes [2–5]. At the same time, with the development of satellite computing and storage capacity, local caching and computing operations of content develop into routine works of satellites. The caching service of the SGIN can effectively reduce repetitive transmission of the large number of multimedia services, which will improve satellite network efficiency [6, 7].

However, the high transmission delay, heavy forwarding burden, and dynamic topology of satellite networks will make the traditional caching strategy not be directly applied in SGIN, decreasing the cache hit rate and network service quality [8, 9]. Therefore, it is of great practical interest to design an intelligent caching and replacement strategy for efficiently enhancing the overall distribution performance in SGIN.

The caching strategy is mainly divided into cache decision strategy and cache replacement strategy. The cache decision strategy determines whether the content be cached, and the cache replacement strategy determines whether the cached content is replaced. The computing resources of satellite nodes are limited while the content is transmitted. Among the existing cache decision strategies, the lightweight caching strategies such as leave copy everywhere [10], leave

copy down [11], move copy down [6], and probability cache [12] have low computational overhead, but they also bring a lot of cache redundancy. Those cache redundancy can be temporarily ignored on ground nodes, but it is intolerable in satellite networks where storage resources are limited.

As a widely used method on the ground, there have been many pieces of research on caching strategy. In paper [13], a Max-Gain in-Network cache gain program (MAGIC) is proposed, which uses a discriminant program to make cache decisions. Its computational complexity is high, and the main controller is completely performed, which is unsuitable for satellite networks. In paper [14], a novel caching scheme named CRCache is proposed to cache hot contents in the backbone network through network topology calculations. However, the extremely high dynamics of satellite nodes lead to a high degree of flattening, making it difficult to determine the backbone network. Paper [15] found peer nodes by establishing social relationships in the downlink to cache content and designed a cache placement algorithm based on the greedy method to configure the cache. Paper [16] proposed a network caching mechanism for time evolution coverage set indication and proposed a novel event update graph to capture topology information to efficiently distribute files in low-orbit satellite networks. Although the mentioned caching strategies greatly improve the cache hit rate, they did not consider the calculation and communication overhead, making them not completely suitable for satellite networks with limited performance.

At present, with the development of deep learning algorithms, a large number of researchers focus on using machine learning methods to predict the popularity of contents and other parameters accurately, and they make caching decisions based on the popularity of content. Paper [17] designed an average content popularity prediction method within a time window for scenarios where instantaneous content popularity may change over time. The optimal content caching probability object was found for probabilistic caching based on the average content popularity. Paper [18] proposed a weighted clustering method to consider the popularity prediction of content caching, took the loss of cache hit rate as the system regret value to express cache performance, and built a popularity prediction framework to satisfy user requirements on the cluster. Paper [19] proposed a real-time change point detector, which can accurately identify the change direction of the average content popularity by improving the heuristic algorithm of time series segmentation, hence generating a caching solution. Some of the above researches are too complex to apply on satellites, and the others do not adapt to the dynamic characteristics of satellites.

The high delay and dynamic characteristics of satellite nodes cause a large amount of cache redundancy that is no longer required after moving the satellite node. As a result, traditional cache replacement strategies such as least recently used [20] and least frequently used [21] have lag problems when used on satellite nodes. The current research on the cache replacement strategy mainly focuses on the prediction method [22–24], which cannot effectively adapt to the frequent topology switching of satellite nodes. The lack of a concise and

effective cache replacement strategy will result in redundancy of cache resources and waste of a large amount of cache space.

In this paper, we aim to establish a content caching strategy by cache profit evaluation model for SGIN. Specifically, we focus on exploring how the caching of different contents affects the caching performance and how the dynamics of satellite affect the cache profits of different contents, which involves the following focus problems:

- (i) Which to cache: With the limited capability of satellite nodes, only a few parts of contents will be cached, while the low cache hit rate will make it difficult for network resources to be used efficiently under the premise of limited node resources. It is important to improve the caching hit rate to consider which contents have a great impact and improve the ability to recognize them.
- (ii) Where to cache: In the SIGN, node resources mainly include storage, bandwidth, and computing resources. The mutual constraint relationship of the three resources determines the cost of content caching. How to choose the cache location with the smallest cache cost to achieve the largest cache profit, thereby improving cache hit rate, reducing data transmission delay, and improving overall network profit, is the key point we need to consider.
- (iii) When to replace: The traditional cache replacement algorithms will always cache the more popular contents in a certain area, but the popular contents in this area may not be required in other parts of the satellite network. The high dynamics of satellites will cause the lag of cache replacement. It is of great interest to design a concise replacement strategy to solve the influence of satellite switches.

To respond to the problems mentioned above, we proposed an intelligent caching and replacement strategy based on the cache profit model, and the main contributions are summarized as follows:

- (i) We design a caching value model for depicting the direct relationship between the caching performance and the multidimensional content features. Specifically, we found the content features that are valuable to the cache performance, and we evaluated the impact of each feature on the cache value. Finally, we used the classification and regression tree to build a cache value evaluation model. In addition, unlike the popular predictive cache method, this model will use statistical values for calculation, which makes the result accurate and verifiable.
- (ii) We propose a cache decision strategy based on the node caching cost model to calculate where the content should be cached. The cost of caching is determined by the node's remaining storage, bandwidth, and computing resources. We describe this mutual constraint relationship, normalize it to the node cache cost equation, and model the SGIN caching problem as a 0-1 knapsack problem. A

concise and effective content caching profit model is obtained by considering the relationship between the caching value and the caching cost. Finally, we modify the genetic algorithm to calculate this model under SDN architecture, and the caching decisions are made through this model, which is of great improvement to the cache hit rate.

- (iii) We redesign the satellite cache replacement strategy by establishing the effective service time model between the satellite and ground transmission link, which solves the lagging problem of the cache replacement when the satellites are switched. We calculate the service duration from the satellite to the ground through periodically moving satellite nodes and calculate the decay of content caching profit. Finally, the content will be replaced based on the current cache profit. This method improves the cache replacement lag problem caused by dynamic satellite switching while using lower computational overhead.

## 2. Content Cache Value Model

*2.1. Delay Model of SGIN.* The SGIN mainly includes satellite backbone network, ground backbone network, and mobile communication. The satellite network includes GEO and LEO satellites, and the ground network is the part to be serviced. In this paper, SDN controllers are deployed on GEO satellites to coordinate control of low orbit satellites. Low orbit satellites are deployed as intelligent computing nodes with edge computing architecture that can cache in the edge nodes.  $V = \{v_1, v_2, \dots, v_n\}$  is defined as the set of low orbit satellite nodes.  $n$  is the number of network nodes. When LEO satellite receives the content from the terrestrial content server,  $S = \{s_1, \dots, s_m\}$  represents the service content collection.  $G = \{g_1, g_2, \dots, g_k\}$  represents the ground forwarding node, which will send interest packets to satellites in the neighborhood for content requests. The network system model based on graph theory abstraction is shown in Figure 1.

Define binary vector  $c_n(s)$  as the storage state of node  $v_n$  for content  $s$ ,  $C_n(s) = 1$  means node  $v_n$  has cached content, and  $c_n(s) = 0$  means the content has not been cached. Use  $\text{Req}$  to represent the content request set, and  $\text{req}_{v_n}(s_n) \in \text{Req}$  represents the request of node  $v_n$  for content  $s_n$  in the set. At this time,  $E(n)$  can represent the content request that the satellite node  $n$  can service:

$$E(n) = \sum_{n=1}^N (\text{req}_{v_n}(s_n) \cdot c_n(s)). \quad (1)$$

The cache hit rate HR of a node can be expressed as  $\text{HR} = \sum_{\text{req}=1}^{\text{Req}} C_n(s) / |\text{Req}|$ . Assuming that the node adjacent to the ground forwarding node is  $v_i$ , the node storing the required content is  $v_j$ , and the number of hops between  $i$  and  $j$  is defined as  $H$ .  $D_h$  represents the time delay of data transmission link between two nodes; then the data transmission delay  $D_{i,j}$  of the complete data communication process of two low-orbit satellite nodes  $(v_i, v_j)$  can be defined as

$$D_{i,j} = \sum_{h=1}^{H_{i,j}} (D_h(v_{h-1}, v_h)). \quad (2)$$

Figure 2 shows the situation where the path node caches the required content. The original transmission path is from the source node  $i$  to the destination node  $j$ . While  $c_n(s) = 1$  and  $E(n) = 1$  in node  $I_2$ , it means that node  $I_2$  has cached the content and can provide services. The transmission delay will be optimized as  $D_{I_2,j} = \sum_{h=1}^{H_{I_2,j}} (D_h(v_{h-1}, v_h))$ .

Define  $D_s$  as the single path reduction of transmission delay, which can be calculated as  $D_s = D_{i,j} - D_{p,j}$ ,  $D_{\text{total}}$  denotes the overall path reductions of transmission delay, and it can be calculated by the following equation:

$$D_{\text{total}} = \sum_{n \in N} E(n) \cdot D. \quad (3)$$

Analysis of equation (3) shows that the low cache hit rate will make it difficult for network resources to be used efficiently under the premise of limited node resources. Current caching strategies are mostly based on the concept of content popularity, but it is not sufficient to fully evaluate the value of content caching when the popularity of the content is the only factor to be considered to analyze and predict. For example, highly popular but huge content will take up a lot of already limited cache space and may not be worthy of being cached. Therefore, it is necessary to use cache profit, not just content popularity, as an evaluation criterion for cache or not. We will establish a cache profit evaluation model in the subsequent chapters, of which the cache value model will be discussed in Section 2.2.

*2.2. Content Cache Value Model Based on CART.* The caching profit of content has two parts: the caching value of the content and the cost of the node caching content. This section will calculate the caching value of content to evaluate the caching profit of content, which discusses various factors that affect the value of the cache and defines and analyzes various evaluation indicators. Finally, the Classification and Regression Tree (CART) is used to solve the cache value.

We proposed six content attributes as the evaluation criteria of the cache value. The amount of storage space occupied is an important factor that affects the value of content caching. If the remaining storage space is less than the content size that needs to be cached, the content cannot be cached, or some cached content needs to be deleted. The content cached by satellite nodes is in images, text data, audio, and video. Different types of content are of different importance, and the cache profits obtained are also different. At the same time, different content request nodes lead to different content priorities. The content request of the ground base station may serve more users, and the content request of the ordinary user node may only meet their own needs.

Content popularity can be defined as the number of times the content is requested within a period, which reflects the popularity of the content. The current time content popularity of the content can be used as an important

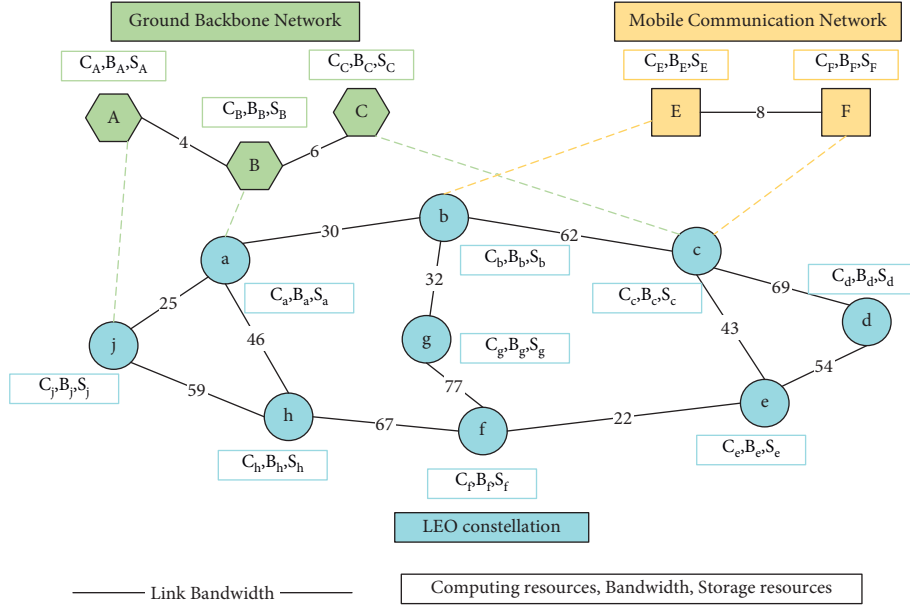


FIGURE 1: Low-orbit satellite service model of SGIN.

indicator for evaluating the value of the cache. At present, a large amount of research focuses on using machine learning methods to predict the popularity of the content and other parameters through historical data. For the satellite network, the dynamic topology and the suddenness of the content make it not completely suitable for this predictive caching scheme, and the predicted hit rate is unverifiable for actual dynamic networks. This paper uses current time content popularity instead of predicted value to do cache calculation.

Assuming a period from time  $t_0$  to  $t_1$ ,  $t_0 < t_1$ . The number of historical times content  $s_i$  is requested is  $\text{req}(s_i)$ . The current time popularity of content  $s_i$  can be defined as (4) during the period.

$$P_{s_i}^{\text{pop}} = \frac{\text{req}(s_i)}{t_1 - t_0}. \quad (4)$$

Define a data set  $D = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\}$ , where each sample  $x_i \in R^k$ ,  $y_i$  is the category label of  $x_i$ , and  $x_i = \{m_1, m_2, \dots, m_6\}$  represents a sample containing all features. These features are defined in Table 1.

The CART algorithm is used to judge the value of content caching because of its simplicity and efficiency. The CART algorithm is a bipartite recursive segmentation algorithm, which makes judgments at the branch nodes. If the judgment condition is true, it is classified as the left branch, and if the condition is false, it is classified as the right branch. Finally, a binary decision tree is formed.

Define the type of content as  $C = \{0, 1, 2, 3, 4\}$ . The CART model will divide the contents into these types. Label 0 is the type with the lowest cache value, while label 4 is the type with the highest cache value. It is necessary to select the optimal partition attribute when using the cart algorithm for decision tree generation. In this paper, the optimal attribute division method is the Gini coefficient method. Suppose that the proportion of the sample  $k$  in the current sample set  $D$  is

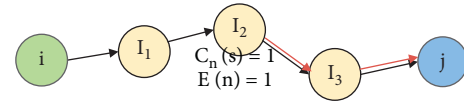


FIGURE 2: How path caching reduces the transmission delay.

TABLE 1: Data features extracted in the cache log file.

Features	Definitions
$m_1$	Content size
$m_2$	Content type
$m_3$	Request node type
$m_4$	Number of historical requests
$m_5$	Last request interval
$m_6$	Content popularity

$P_k$  ( $k = 1, 2, \dots, |y|$ ). The equation for measuring the purity of  $D$  using the Gini coefficient is as follows:

$$\text{Gini}(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} P_k P_{k'} = 1 - \sum_{k=1}^{|y|} P_k^2. \quad (5)$$

The smaller the Gini coefficient, the higher the purity of the data set  $D$ . If the attribute  $m$  is used to divide  $D$ , the equation of divided Gini index is as follows:

$$\text{Gini\_index}(D, m) = \sum_{m=1}^M \frac{|D^m|}{|D|} \text{Gini}(D^m). \quad (6)$$

Therefore, the attribute  $m_* = \arg \min_{a \in A} \text{Gini\_index}(D, a)$  with the smallest Gini index after division can be regarded as the optimal division attribute. After obtaining the optimal division attributes, CART can be used for content classification.

### 3. Caching Decision Strategy

With the support of the centralized control and global perspective of the SDN controller, it will be easy to record and calculate the multidimensional characteristics of the content. Based on these characteristics, nodes will make caching decisions that are beneficial to the whole system. We utilized these advantages to design a cache decision and cache replacement strategy architecture based on the control process of high and low orbit satellites, which is shown in Figure 3. We will introduce the modules mentioned in the following chapters.

The overall cache process based on SDN can be simply described as follows:

*Step 1.* The topology management module of the LEO satellite regularly uploads topology information to the SDN controller for utilization by the Caching Decision Maker and Routing Manager.

*Step 2.* The LEO satellite submits the received content request to the SDN controller, and the SDN controller calculates the value of the content and transmits it to the Caching Decision Maker.

*Step 3.* The routing management module of the SDN controller formulates a routing strategy based on the overall topology information and then passes it to the Caching Decision Maker. The Caching Decision Maker and Routing Manager will calculate the cache decision and input it into the Forwarding Handler.

*Step 4.* The Value Decay Time Calculator will calculate the profit decay time and join the Forwarding Handler, and the Forwarding Handler will issue the control commands of the relevant nodes.

*3.1. SGIN Cache Decision Problem.* Under the condition of limited node resources, selective caching of content is the key to improving storage resource utilization efficiency. The content caching problem with limited computing and storage resources can be described as a multiconstraint dynamic programming problem that maximizes the profit of content caching, expressed in the following equations:

$$\max D_{\text{total}}, \quad (7)$$

$$\text{s.t. } \sum L_s \cdot c_n(s) \leq Z_n \quad (n \neq 0, n \in N), \quad (8)$$

$$\sum A_s \cdot E(n) \leq A_n \quad (n \neq 0, n \in N). \quad (9)$$

Equation (8) is a constraint condition to ensure that the resource size cached by a node does not exceed its cache capacity.  $L_s$  represents the size of the resource  $s$ , and  $Z_n$  represents the maximum storage capacity of the node. Equation (9) is a constraint condition to ensure that the calculation amount does not exceed the sum of its

computing resources.  $A_s$  represents the computing resources required to transmit a single content, and  $A_n$  represents the total computing resources of the entire node.

In the SGIN, there is an optimal solution at each current moment  $t$ . Define  $r(s)$  as the profit gained from caching content  $s$ . Assuming that only content  $s$  arrives at time  $t + \Delta t$ , where  $\Delta t \rightarrow 0$ , and the profit at time  $t$  is represented by  $D_{\text{total}}^{\max}(t)$ ,  $D_{\text{total}}^{\max}(t + \Delta t) = D_{\text{total}}^{\max}(t) + r(s)$ . Due to the large number of requests for content in the network,  $D_{\text{total}}^{\max}(t) \gg g(s)$  can be obtained, so  $D_{\text{total}}^{\max}(t + \Delta t) = D_{\text{total}}^{\max}(t)$  can be approximated. Divide the time into slices, and each slot allows one request to arrive; then the current optimal decision can be obtained based on calculations based on historical data. The optimal decision at this moment can be used as the optimal cache decision for the next time slot  $\Delta t$ . By calculating the historical data and the new request for the next time slot  $\Delta t$ , the optimal solution for the next time slot can be obtained, and the best decision-making scheme for satellite caching can be obtained by repeating the above steps.

However, the planning problem of dynamic scenes is still NP-hard, and the computational complexity is extremely high. Even if a solution can be found by traversal, the time it takes is unacceptable for a dynamic network. The next section will analyze and study the caching strategy based on the network resource topology model.

*3.2. The Cache Decision Strategy by GA Method under SDN Architecture.* Because of the extremely high signaling overhead and computational complexity of the network's global dynamic delay profit maximization, an effective solution cannot be obtained according to the existing methods. Every single node in the network can obtain the network-wide resource topology model through the use of SDN. Therefore, the problem of delay profit maximization can be transformed into a single node dynamic multidimensional 0-1 knapsack problem. The general expression of the knapsack problem is how to combine to maximize the total value of the items in the backpack when the total weight of the backpack does not exceed the threshold, and each item has two attributes: weight and value. The value of content caching has been discussed in Section 3.2. The weight of items is described as the cost of content caching, which will be discussed in this section.

In the SIGN, node resources mainly include storage resources, bandwidth resources, and computing resources. The mutual constraint relationship of the three resources determines the cost of content caching. The remaining space of a node is a necessary condition for caching or not. The computing resources of the node determine the necessary waiting time for caching this content, and the bandwidth of the node determines the propagation delay of this content to other nodes. How to cache the content with the smallest cache cost and the largest cache value to achieve effective utilization of storage resources, thereby improving cache hit rate, reducing data transmission delay, and improving overall network profit, is the key that we need to consider in this section.

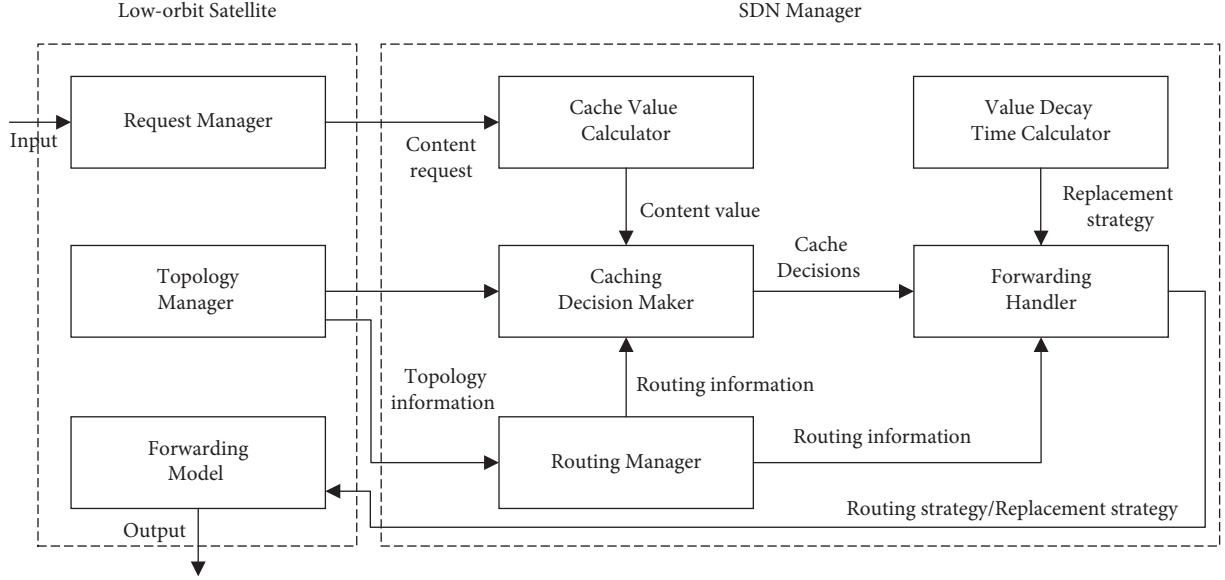


FIGURE 3: The system cache decision framework.

3.2.1. *Content Size and Remaining Cache Space.* Due to the limited storage resources of the satellite nodes in SIGN, if the remaining storage space is less than the size of the content that needs to be cached, the content cannot be cached, or some cached content needs to be deleted. Therefore, the amount of storage space occupied is also an important factor affecting content caching value. Set the storage space size of node  $v_j$  to  $H_{v_j}$ , and the size of the cache space occupied is  $H_{v_j}^{\text{occupy}}$ ; then the remaining cache space of the node can be calculated as

$$H_{v_j}^{\text{remain}} = H_{v_j} - H_{v_j}^{\text{occupy}}. \quad (10)$$

Define the size of content  $s_i$  as  $f_i$ , and use the impact factor  $\delta_{s_i}$  to represent the impact of the size of the content on the value of the cache.

$$\delta_{s_i} = \begin{cases} 1, & h_i \leq H_{v_j}^{\text{remain}}, \\ 1 - \frac{h_i - H_{v_j}^{\text{remain}}}{H_{v_j}}, & H_{v_j}^{\text{remain}} \leq h_i \leq H_{v_j}, \\ 0, & H_{v_j} \leq h_i. \end{cases} \quad (11)$$

When the remaining cache space is sufficient to cache content  $s_i$ , set  $\delta_{s_i}$  to 1. When the total capacity of the cache space  $H_{v_j}$  is less than  $h_i$ ,  $s_i$  cannot be cached; set  $\delta_{s_i}$  to 0. When some contents need to be deleted to make the content able to be cached, the larger  $h_i$  is, the smaller  $\delta_{s_i}$  will be. At the same time, the larger the node cache space  $H_{v_j}$  is, the larger  $\delta_{s_i}$  will be.

3.2.2. *Resources of Computing.* The computing resources will significantly affect the queuing delay and packet loss rate of the node, thereby affecting the cache cost of the node. The

computing resource is defined as the coupling value  $U$  between the CPU and RAM of the node. The computing resource which is the send and receives content  $s_i$  required is defined as  $u_i$ ; then the computing resource cost impact factor of content  $s_i$  cached on node  $v_j$  is defined as  $O_{s_i, v_j} = u_{s_i}/U_{v_j}$ . The more computing resources the cached content  $s_i$  occupies, the larger  $O_{s_i, v_j}$  will be.

3.2.3. *Remaining Bandwidth.* The remaining bandwidth refers to the amount of unoccupied data transmission in communication. During service transmission, the remaining bandwidth  $b_j$  of node  $v_j$  can be expressed by port data. The calculation equation is as follows:

$$b_{v_j} = B_{v_j} - \text{in}_{\text{bytes}}(v_j) - \text{out}_{\text{bytes}}(v_j), \quad (12)$$

where  $B_i$  represents the total bandwidth of node  $i$ ,  $\text{in}_{\text{bytes}}(v_j)$  represents the byte acceptance rate of node  $v_j$ , and  $\text{out}_{\text{bytes}}(v_j)$  represents the byte transmission rate of node  $i$ . The transmission delay  $t_{\text{trans}}$  of the content is used to represent its cache cost. If the size of content  $s_i$  is  $h_{v_j}$ , it can be calculated as

$$t_{i,j}^{\text{trans}} = \frac{h_{s_i}}{b_{v_j}}. \quad (13)$$

3.2.4. *Cache Cost Model Based on the Relative Storage Location of Nodes.* Due to the high dynamics of satellite nodes, considering the real-time distance will make the problem be complicated. The number of hops is used to determine the node location. The storage location is correlated with the user distance. Even if the content is highly popular, it will not generate high cache profit if it is cached in a node that is far away. The hop count index between node storage and user location is defined as  $\text{hop}_{i,j}$ , where  $i$  represents the node

caching the content,  $j$  represents the position of the user, and the hop count is used to represent the user distance abstractly. According to the network, the topology to select the Dijkstra algorithm to calculate the minimum number of hops and the transmission cost of each node is defined as  $\varphi_{i,j}$ . From this, the relative transmission cost  $t_{s_i,v_j}^{\text{spread}}$  of content  $s_i$  based on the storage location will be got. The calculation method is as follows:

$$t_{s_i,v_j}^{\text{spread}} = \sum_{j=1}^{\text{hop}_{i,j}} \varphi_{i,j}. \quad (14)$$

Based on the above equation, we can get the caching cost  $\text{cost}_{s_i,v_j}$  of content  $s_i$  at node  $v_j$ . The calculation equation is as follows:

$$\text{cost}_{s_i,v_j} = \delta_{s_i} \cdot O_{s_i,v_j} \cdot t_{i,j}^{\text{trans}} \cdot t_{s_i,v_j}^{\text{spread}} = \begin{cases} \sum_{j=1}^{\text{hop}_{i,j}} \varphi_{i,j} \cdot O_{s_i,v_j} \cdot t_{i,j}^{\text{trans}} \cdot t_{s_i,v_j}^{\text{spread}}, & h_i \leq H_{v_j}^{\text{remain}}, \\ \sum_{j=1}^{\text{hop}_{i,j}} \varphi_{i,j} \cdot O_{s_i,v_j} \cdot t_{i,j}^{\text{trans}} \cdot t_{s_i,v_j}^{\text{spread}} - \frac{h_i - H_{v_j}^{\text{remain}}}{H_{v_j}}, & H_{v_j}^{\text{remain}} \leq h_i \leq H_{v_j}, \\ 0, & H_{v_j} \leq h_i. \end{cases} \quad (15)$$

The problem of caching solution in SGIN can be expressed as how to select cache content in a node without exceeding the storage threshold of the node, to maximize the total value-cost ratio of the content cached by a single node. In traditional distributed node caching schemes, nodes make caching decisions individually, which is likely to cause cache redundancy. If the previous hop node has cached hot content, the request rate of this hot content in this node will be greatly reduced. Since the content caching cost of SDN from a global perspective considers the relative position of nodes, the solution set of a single node caching scheme can be approximated as a globally optimal solution, and the total profit  $P$  of global content caching can also be optimal. This paper proves this point in the simulation.

Assuming that the total number of contents existing in the network is  $M$ , use the binary vector  $c_n(s_i)$  defined in Chapter 2 to indicate whether to cache content  $s_i$ , and the satellite node cache knapsack problem is defined by the following equation:

$$\begin{aligned} \text{maximize } P &= \sum_{i=1}^M \frac{\text{value}_{s_i}}{\text{cost}_{s_i,v_j}} \cdot c_n(s_i), \\ \text{subject to } \sum_{i=1}^M h_i \cdot c_n(s_i) &\leq H_{v_j}, \quad j = 1, \dots, n. \end{aligned} \quad (16)$$

The knapsack problem is a classic NP-complete problem. However, people still cannot find a perfect solution for the large-scale 0-1 knapsack problem. Although the traversal method can obtain the optimal solution, the solution speed is slower. Due to the advantages of genetic algorithms in global search, this paper considers using the genetic algorithm to solve this problem. Therefore, a simplified GA optimization procedure for satellite nodes is

proposed, which further reduces the computational complexity by defining the location of characteristic genes in advance.

Define each initial gene in the population as a binary string, and each gene represents a feasible solution for a caching scheme. Use  $c_n(s_i) = x_{i_k} = 1$  or 0 to indicate whether to cache the content; the initial gene in the population can be recorded as

$$\begin{aligned} p_{\text{th}} &= \{x_{1_k}, x_{2_k}, \dots, x_{q_k}\}, \\ x_{i_k} &= 0, 1. \end{aligned} \quad (17)$$

The calculation equation of its fitness is

$$f_i = \sum_{i=1}^M \frac{\text{value}_{s_i}}{\text{cost}_{s_i,v_j}} \cdot x_{i_k}, \quad j = 1, \dots, n; i = 1, \dots, m. \quad (18)$$

For every  $p_{\text{th}}$ , if  $\forall x_{i_k} = 1$  or 0 in  $p_{\text{th}}$  making  $f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_q) \geq f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_q)$ , then  $p_{\text{th}}$  is the characteristic gene, and  $x_i = 1$  is an excellent choice. If  $p_{\text{th}}$  is the characteristic gene,  $x_i = 1$  is an excellent choice, and then the value of the optimal solution will always be  $x_i = 1$ . If there are  $q$  characteristic genes, then only  $2^{q-k}$  individuals are in the searching space.

The addition of SDN makes the caching strategy of isolated nodes derived from the genetic algorithm into a centralized solution. The content caching cost of SDN from a global perspective considers the relative position of nodes. Therefore, the set of optimal solutions for a single node cache calculated at this time can be approximated as the optimal global solution.

The optimal solution algorithm mentioned in Section 3.1 has high computational complexity and high overhead, which does not have actual engineering value, but it can be used as an evaluation index for the algorithm in this paper.

The cache hit rate is an important indicator for evaluating the efficiency of cache decision-making. The curve can be calculated by the equation  $HR = \sum_{req=1}^{Req} C_n(s) / |Req|$ . Figure 4 compares the optimal solution proposed above with the convergence of the genetic algorithm caching decision strategy based on the cache profit judgment proposed in this paper in a particular time slot  $t$ . It can be seen that the cache hit rate of Value-ga has always been in a better state, close to the optimal solution with lower computational overhead.

#### 4. Cache Replacement Strategy

The replacement strategy of the satellite cache space should be as concise as possible because the complex cache replacement strategy will affect the timeliness performance and accuracy of the cache strategy. The traditional simple cache replacement algorithm will always cache the more popular contents in a certain area, but the popular contents in this area may not be required in other parts of the satellite network. Although caching these resources improves the cache hit rate in this area, the hot resources will not be replaced for a long time and invalidated in the next area when the satellite topology is switched. In response to this problem, we introduce the concept of diminishing cache profit time based on the service duration of the satellite and ground and design a cache replacement algorithm that considers the decrease of cache profit. This method improves the cache replacement lag problem caused by dynamic satellite switching while using lower computational overhead.

Figure 5 shows how the satellite-to-ground service switches between satellites. Nodes  $A, B,$  and  $C$  are the service satellite, while nodes  $G$  and  $H$  are the ground nodes to be serviced. Since the satellite's movement is periodic, the SDN controller can cache the dynamic topology of the SIGN. Therefore, the service duration model between satellite and ground can calculate the ground service duration of each satellite and assign a fixed service satellite to each ground node. The time from the satellite entering to leaving the service distance is defined as the topology switching time. Then, the decreasing time of content cache profit can be defined as the difference between the data storage time and the next topology switching time.

Assume that the channel between the satellite node and the ground user follows the free path loss model  $P_r/P_t = 1/d^\alpha$ .  $P_r$  represents the user's received power,  $P_t$  represents the transmit power of the low-orbit satellite node,  $d$  represents the distance between the LEO satellite node and the ground user, and  $\alpha$  is the path loss factor. Considering the complexity of the power control of LEO satellite nodes and ground equipment, it is assumed that the LEO satellite node uses a constant transmission power, which is denoted as  $P_t$ .

When the ground user  $g_i$  communicates with the LEO satellite node  $v_j$ , the received signal-noise ratio (SNR)  $\gamma_{y_i}$  of the ground user can be recorded as  $\gamma_{y_i} = P_t d_{i,j}^{-\alpha} / N_0$ , where  $d_{i,j}$  is the distance between  $g_i$  and  $v_j$ , and  $N_0$  represents additive white Gaussian noise power. Equation  $d_{i,j} = (P_t / \gamma_{g_i} N_0)^{1/\alpha}$  will be got through rewriting the above

equation. In order to ensure the user's service quality, there should be  $\gamma_{G_i} \geq \gamma_{th}$ .  $\gamma_{th}$  is the threshold of SNR. Then, the maximum communication distance from satellite to the ground can be expressed as

$$d_{max} = \left( \frac{P_t}{\gamma_{th} N_0} \right)^{1/\alpha}. \quad (19)$$

Based on the above analysis, if the user equipment wants to download content from an LEO satellite node, the user equipment should be located in a circular cell with the LEO satellite node as the center and a radius less than  $d_{max}$ . Due to the regularity of the LEO satellite node movement, its service time will become calculable. The orbit of the LEO satellite is approximated as a circle, and the straight line distance between the satellite and the ground is  $H$ , and the operation period is  $T$ ; then the center angle of the satellite service for the ground station is  $\partial = 2 \cos^{-1}(d_{max}/H)$ . Considering that the azimuth angle  $\theta$  of the satellite antenna is a fixed value, the time from detecting the ground station and starting to provide service to out of service is only related to  $\theta$  and  $d_{max}$ . When  $d_{i,j} \leq d_{max}$ , the available satellite service time of the ground node can be calculated as follows:

$$T_s = \begin{cases} \frac{2T \cos^{-1}(d_{max}/H)}{2\pi}, & \theta \geq \partial, \\ \frac{T\theta}{2\pi}, & \theta < \partial. \end{cases} \quad (20)$$

When requesting content from the SDN, the node can receive the cache profit value and the cache profit decrease time returned by the SDN, which can reduce the response time of future requests and the utilization of network bandwidth. In order to avoid topology switching to invalidate hotspot contents and cause the lag in cache replacement, SDN is used to calculate the decreasing time of cache profit. The cache profit value sent by SDN and the profit diminishing time  $t$  are used as the cache weight, and the content is sorted in the cache stack according to the cache profit. When the cache profit is 0, the original file can be discarded directly when the new file arrives.

Define  $P_{s_i}^t$  as the cache profit of content  $s_i$  at time  $t$ ; the calculation method is as follows:

$$P_{s_i}^t = P_{s_i} - \frac{P_{s_i} t}{T_s}. \quad (21)$$

The specific process of the cache management algorithm of the satellite node online replacement is shown in1:

### 5. Simulation

**5.1. Simulation Environment Parameter Setting.** In order to restore the operation of the world-earth integrated network as much as possible and simulate the data stream caching process, the following work is required: simulate the real orbit of the satellite-ground switching state; the simulated satellite node needs to have calculation and cache functions, and the ground station needs to receive and send content



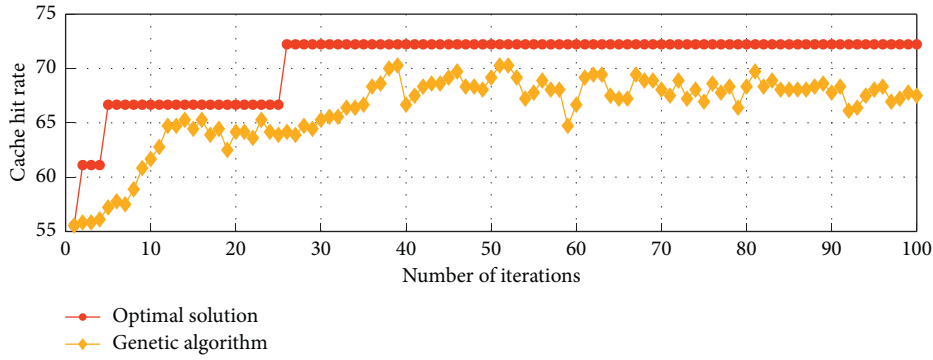


FIGURE 4: Comparison of Value-ga and optimal solution.

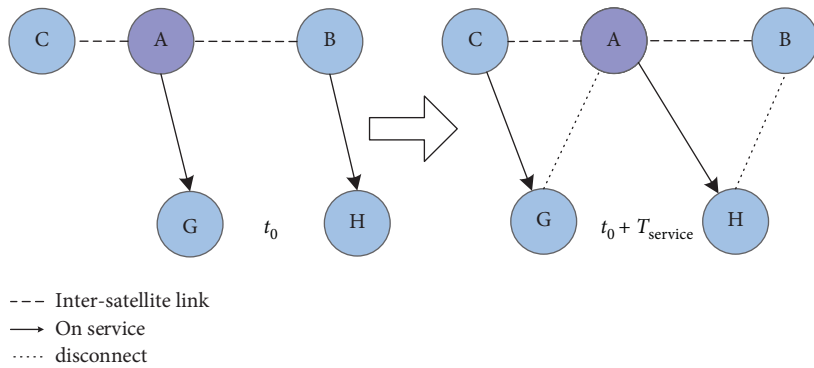


FIGURE 5: Satellite-to-ground service switching model.

```

Input: user request content  $g$ , cache profit value, time of diminishing returns  $t$ 
for every content requested by the user  $g$ 
    if  $g$  is in the cache;
        Update  $g$  information;
         $g$  cache profit update, insert the cache stack;
    else  $g$  cache miss
        request the relevant routing table of  $g$  from the SDN controller;
    while there is not enough space in the cache for  $g$ ;
        delete the object at the bottom of the cache stack  $q$ ;
    end while
    calculate cache profit of  $g$  and the rate of profit decay;
    insert the cache stack
    end if
    sort content by cache profit
end for
    
```

ALGORITHM 1: Cache replacement management.

requests of different sizes; the SDN controller would collect satellite cache logs and real-time resource status and control the satellite to cache content.

Since the official NDN simulator official simulation tool NDNSim cannot simulate dynamic satellite nodes very well, in this study, we use STK and MATLAB to build a space-ground integrated network simulation environment jointly. The satellite model built by STK includes three high-orbit satellites, 24 walker constellation low-orbit satellites, and 16 ground stations. The walker constellation satellite has an

orbital height of 1400 km and an orbital inclination of  $52^\circ$ . It is divided into three orbital planes, and each orbital plane is distributed with eight satellites. The actual operation period of the walker constellation satellite is about 120 minutes. This paper scales it in proportion to 120 seconds as a satellite operation period. The topology of the satellite remains unchanged, and the topology switching period is 10 s. The SDN controller is placed on the high-orbit satellite, and its main function is to perform log collection and global routing control. In the simulation, the interest packet is sent by 16

TABLE 2: The simulation parameter settings.

Parameters	Descriptions	Values
$n$	Total number of resource files	100
Request rate	Interest packet sending frequency value range	10, 20, 30, 40, 50, 60, 70
Cache size	Satellite node cache space size	50, 100, 150, 200, 250, 300
$T$	Simulation cycle	120
$\alpha$	Zipf distribution parameter value range	0.8, 0.9, 1.0, 1.1, 1.2, 1.3

ground stations simultaneously, the data packets are transmitted in the low-orbit satellite node, and the ground station is responsible for the last hop reception.

The parameter settings of satellite nodes are mainly obtained through STK, and the content request of the ground station is mainly set by experience. The total content request in the satellite network is modeled according to the Zipf distribution equation  $P(r) = C/r^\alpha$ , where  $P(r)$  is the requested frequency of content  $r$ , and  $\alpha$  is the Zipf distribution parameter. The 100 content files used in the design are placed on each low orbit satellite network node. A single content file size is a random updated value in the range of 1–10 profit affecting estimate, and the total size of all content files is 800 MB. In order to explore the impact of the satellite node’s cache capacity on the performance of the cache strategy, the value of the node’s cache capacity is 50–300 MB. We observed the impact of different Zipf distribution indexes on the cache hit rate through experiments. The value of Zipf distribution index is 0.8–1.3, and the default value is 1. Finally, this paper uses the interest packet sending frequency to simulate the impact of the network load on the cache hit rate. The network link bandwidth is set to 20 Mbps, and the interest packet request frequency varies within the range of 10–100/sec. The default value is 50.

The simulation parameter settings are shown in Table 2.

**5.2. Simulation Results and Analysis.** For comparison with the Value-ga caching strategy in this paper, four caching strategies are selected in this chapter. Choose the downward caching strategy LCD as the independent caching strategy, Prob as the classic scheme of the probability model in collaborative caching, CRCache as the typical algorithm considering content popularity in cooperative caching, and LCE as the basic general scheme. The replacement schemes of the four caching strategies all choose the least recently used algorithm.

In the algorithm of this paper, after the satellite network node receives the content, it will extract its features. The specific features and data settings included are shown in Table 3.

In order to explore the impact of the dynamics of satellite nodes on the algorithm performance, we recorded the transmission delay for a total of 32,000 interest packets to obtain data packets for five algorithms in a complete simulation cycle; when the interest packet transmission frequency is 20 per second, other parameters are default.

The simulation result is shown in Figure 6. It can be seen that, at the beginning of the simulation, each satellite node starts to cache in the network. As the simulation time

TABLE 3: The simulation data features.

Definitions	Descriptions
Content size	1–20 (MB)
Content type	Images, text, audio, and video
Request node type	Edge, common, and central nodes
Number of historical requests	Collected by SDN
Last request interval	Counted by LEO
Content popularity	Calculated by equation (4)

increases, the average delay of data transmission steadily decreases. In the middle and late stages of the simulation, the average delay remains stable, caused by the fact that the cache of the satellite node is full, and the ability of the cache in the network to optimize data transmission delay has reached the threshold. At the same time, due to the periodic topology switching of satellites, the hotspot contents of the previous topology become invalid on a large scale, and there is a lag in caching new hotspot contents, which makes the CRCache algorithm perform poorly at the nodes of the topology switching. Because the Value-ga algorithm in this paper introduces the concept of maximum survival time, nonhot contents can be replaced faster after topology switching. Hence, the oscillation caused by satellite topology switching is small, and the average delay can be stably maintained at a low position. The calculation results of the average delay are shown in Table 4. As can be seen from the table, the average delay of the Value-ga algorithm is the lowest.

The first comparison result obtained from the above analysis is that when the Value-ga algorithm runs in a satellite network, the average data transmission delay can be kept at a low position, and it is more stable than other algorithms.

We continue to study the performance gains of different caching schemes as the node caching capabilities change in the satellite network. Figures 7 and 8 show the average cache hit rate trend and an average number of hops as the size of the satellite node’s cache changes after the five cache schemes run for one cycle each under the default parameters. It can be seen that, with the increase of the cache size, the performance of the five schemes has been significantly improved. When the satellite cache size is only 50 MB, the average cache hit rate of the entire network is only 15%–20%. When it increases to 300 MB, the cache hit rate of the Value-ga cache strategy can be increased to 64%, and three hops reduce the average number of hops. From Figures 7 and 8, Value-ga is significantly better than the other four caching schemes regarding an average cache hit rate and the average number of hops. When the node cache size is 300 MB,

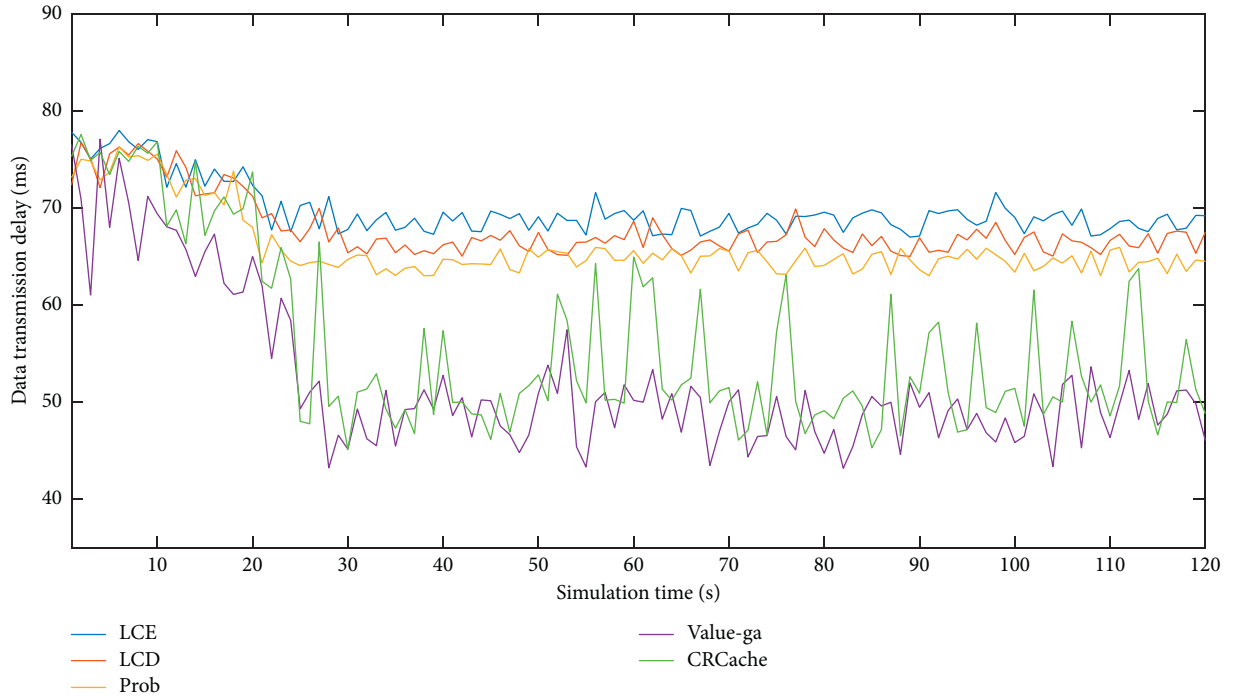


FIGURE 6: Data packet transmission delay in a single simulation cycle.

TABLE 4: Average delay of five algorithms in a single simulation cycle.

	LCE	LCD	Prob	CRCache	Value-ga
Average delay	69.71	68.03	65.98	58.30	52.96

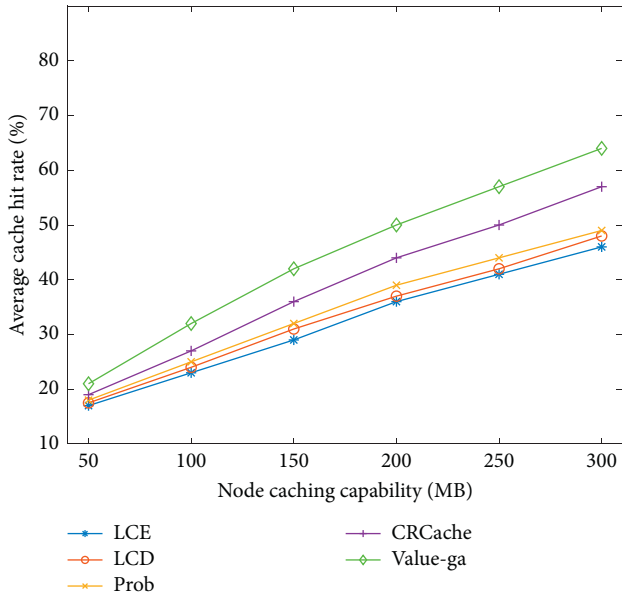


FIGURE 7: The relationship between node caching capability and average cache hit rate.

Value-ga increases the average cache hit rate by 9.58% compared with CRCache, and 0.32 hops reduce the average number of node hops. The second comparison result obtained from the above analysis is that, from the perspective

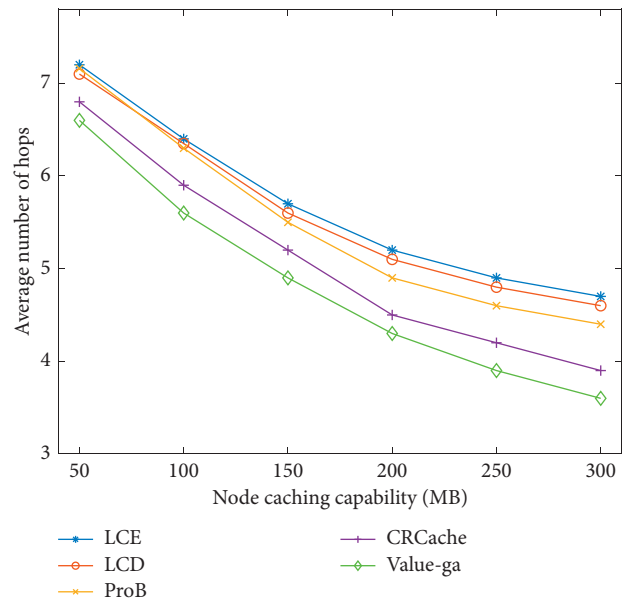


FIGURE 8: The relationship between node caching capability and average number of hops.

of the overall network, Value-ga is significantly better than the other four solutions; and as the cache size increases, the performance gap between Value-ga and the other four caches gets larger.

In order to explore the caching situation of hot content by the five algorithms, the cache size of the satellite node is set to 200 MB, and the relationship between the Zipf index and the average cache hit rate is explored. The larger the alpha index of the Zipf distribution is, the more times the hot content is requested. It can be clearly seen from Figure 9 that,

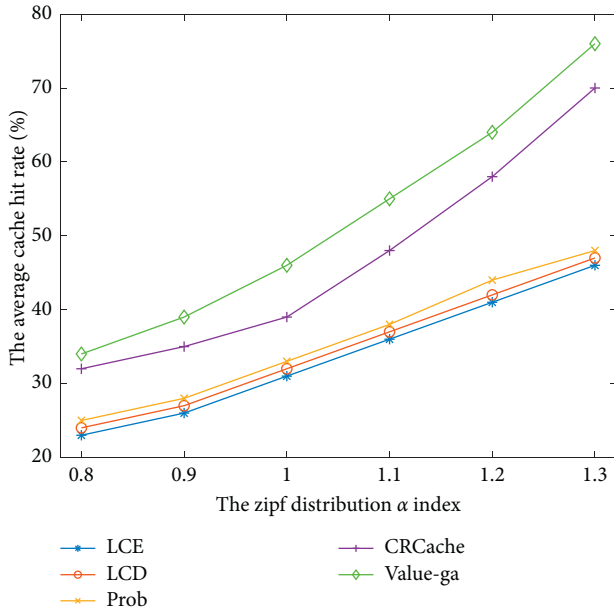


FIGURE 9: The relationship between the Zipf distribution  $\alpha$  index and the average cache hit rate.

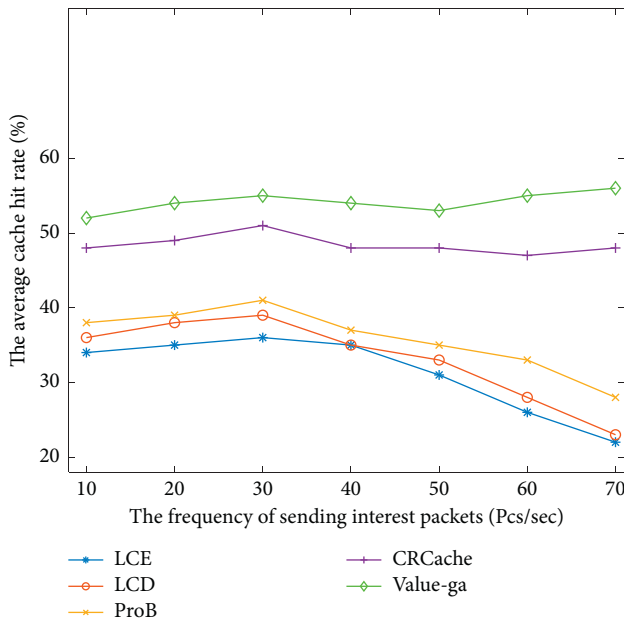


FIGURE 10: The relationship between the frequency of sending interest packets and the average cache hit rate.

with the increase of  $\alpha$ , the average cache hit rate of the five algorithms is improving, but Value-ga and CRCache are more sensitive to hot content. Due to the limitation of the dynamics of satellite nodes, the performance of CRCache cannot be fully utilized. When  $\alpha = 1.3$ , the average cache hit rate of the Value-ga algorithm is 8.57% higher than that of CRCache. Analysis shows that, in the satellite network environment, the Value-ga algorithm can predict and cache hot content more efficiently.

Figure 10 compares the cache hit rate of five caching strategies when the interest request changes. It can be seen

that as the value of the request frequency increases, the network load begins to increase, the numbers of interest packets and data packets in the network increase, and the cache hit rate of the five cache strategies is also slightly improved. When the request frequency reaches 30 per second and increases with the request frequency, the cache hit rate curve of Value-ga and CRCache remains stable, and the cache hit rate of LCE, LCD, and Prob decreases to varying degrees. This is because satellite nodes continue to perform cache replacement under high load conditions and cannot perform effective cache storage. Value-ga and CRCache can still work effectively under high load conditions due to the use of intelligent algorithms, and because Value-ga takes into account the multidimensional characteristics of data, it can cache more efficiently, thus maintaining the best performance.

## 6. Conclusions

In order to solve the problem of low cache hit rate and large data transmission delay in the space-ground integrated network, this paper proposes a Value-ga caching strategy based on the value of content caching. Through the centralized SDN controller, the profit of the content cache in the satellite network is calculated. At the same time, in order to adapt to the dynamic changes of the satellite network, a new cache replacement strategy is designed, which significantly improves the utilization efficiency of the cache space. The simulation results show that, compared with LCE, LCD, Prob, and CRCache strategies, Value-ga significantly improves the cache utilization of satellite nodes, reduces the data packet transmission delay in the network, and is more suitable for satellite networks.

## Data Availability

The data used to support the findings of this study are available from the authors upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] Cisco, *Cisco Annual Internet Report (2018–2023)*, Cisco White Paper, San Jose, CA, USA, 2020.
- [2] G. Xylomenos, C. N. Ververidis, V. A. Siris et al., “A survey of information-centric networking research,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.
- [3] T. Koponen, T. M. Chawla, B. G. Chun et al., “A data-oriented (and beyond) network architecture,” in *Proceedings of the ACM SIGCOMM*, Kyoto Japan, 2007.
- [4] X. Xu, Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and Z. A. Bhuiyan, “Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, 2020.
- [5] C. Fang, F. Richard Y u, T. Huang, J. Liu, and Y. J. Liu, “A survey of green information-centric networking:research

- issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 99, pp. 1–19, 2015.
- [6] G. Zhang, Y. Li, and T. Lin, “Caching in information centric networking: a survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.
- [7] D. Perino and M. Varvello, “A reality check for content centric networking,” in *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*, Toronto, Canada, 2011.
- [8] L. Saino, I. Psaras, and G. Pavlou, “Hashing routing schemes for information-centric networking,” in *Proceedings of the ACM Workshop on ICN*, Hong Kong, China, 2013.
- [9] L. Bertaux, S. Medjah, P. Berthou et al., “Software defined networking and virtualization for broadband satellite networks,” *IEEE Communications Magazine*, vol. 53, no. 3, pp. 54–60, 2015.
- [10] N. Laoutaris, S. Syntila, and I. Stavrakakis, “Meta algorithms for hierarchical web caches,” in *Proceedings of the 2004 IEEE International Conference on Performance, Computing, and Communications*, pp. 445–452, Phoenix, AZ, USA, 2004.
- [11] C. Bernardini, T. Silverston, and O. Festor, “A comparison of caching strategies for content centric networking,” in *Proceedings of the 2015 IEEE GLOBECOM*, pp. 1–6, San Diego, CA, USA, 2015.
- [12] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network caching for information- centric networks,” in *Proceedings of the ACM SIGCOMM Workshop ICN*, pp. 55–60, Helsinki, Finland, 2012.
- [13] J. Ren, W. Qi, C. Westphal et al., “MAGIC: a distributed MAX-Gain In-network Caching strategy in information-centric networks,” in *Proceedings of the IEEE INFOCOM*, pp. 470–475, Toronto, Canada, 2014.
- [14] W. Wang, Y. Sun, Y. Guo et al., “CRCache: exploiting the correlation between content popularity and network topology information for ICN caching,” in *Proceedings of the IEEE International Conference on Communications*, IEEE, Sydney, Australia, 2014.
- [15] C. Jiang and Z. Li, “Decreasing big data application latency in satellite link by caching and peer selection,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2555–2565, 2020.
- [16] Z. Yang, Y. Li, P. Yuan, and Q. Zhang, “TCSC: a novel file distribution strategy in integrated LEO satellite-terrestrial networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5426–5441, 2020.
- [17] J. Gao, S. Zhang, L. Zhao, and X. Shen, “The design of dynamic probabilistic caching with time-varying content popularity,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1672–1684, 2021.
- [18] Q. Chen, W. Wang, F. R. Yu, M. Tao, and Z. Zhang, “Content caching oriented popularity prediction: a weighted clustering approach,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 623–636, 2021.
- [19] S. Skaperas, L. Mamatas, and A. Chorti, “Real-time video content popularity detection based on mean change point Analysis,” *IEEE Access*, vol. 7, pp. 142246–142260, 2019.
- [20] S. Arianfar, P. Nikander, and J. Ott, “On content-centric router design and implications,” in *Proceedings of the Re-Architecting the Internet Work-Shop*, pp. 5–6, ACM, New York, NY, USA, 2010.
- [21] G. Rossini and D. Rossi, “A dive into the caching performance of content centric networking,” in *Proceedings of the IEEE International Workshop on Computer Aided Modeling & Design of Communication Links & Networks*, 2012.
- [22] E. Liu, M. Hashemi, S. Kevin, and P. Ranganathan, “Junwhan ahn proceedings of the 37th international conference on machine learning,” *PMLR*, vol. 119, pp. 6237–6247, 2020.
- [23] T. Ma, Y. Hao, W. Shen, Y. Tian, and M. Al-Rodhaan, “An improved web cache replacement algorithm based on weighting and cost,” *IEEE Access*, vol. 6, pp. 27010–27017, 2018.
- [24] A. Vakil-Ghahani, S. Mahdizadeh-Shahri, M.-R. Lotfi-Namin, M. Bakhshalipour, P. Lotfi-Kamran, and H. Sarbazi-Azad, “Cache replacement policy based on expected hit count,” *IEEE Computer Architecture Letters*, vol. 17, no. 1, pp. 64–67, 2018.