PBRS: A Content Popularity and Betweenness Based Cache Replacement Scheme in ICN-IoT

Ying Liu[1], Ting Zhi[1,2], Huachun Zhou[1], Haidong Xi[1,3]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, China
[2] CETC Big Data Research Institute Co., Ltd., China
[3] Cisco System (China) Research and Development Co., China
{yliu, 15111048, hchzhou, 15120165}@bjtu.edu.cn

Abstract

Since the Information-Centric Networking (ICN) cache can effectively reduce the requests from customers to producers and improve the efficiency of content acquisition, researchers have attempted to improve the network performance of the Internet of Things (IoT) by using the concept of ICN. In the context of ICN-IoT, an appropriate cache placement mechanism can effectively help reduce data retrieval delays, and an appropriate cache replacement mechanism is also important to maintain the cache performance. ICN generally uses the least recently used (LRU) and the least frequently used (LFU) cache replacement strategies, which may cause large storage redundancy. To improve the cache hit ratio of the network under the context of the ICN-IoT, in this paper, we propose a novel intra-network cache replacement method that integrates evaluating the popularity of the cache contents and the betweenness of the nodes. This mechanism promotes the performance of the cache replacement by weighing the importance of each node and the cache content popularity within the domain. The simulation results show that the proposed cache replacement method can achieve a higher cache hit rate than traditional cache replacement strategies.

Keywords: Information centric networking, Internet of Things, Cache replacement, Content popularity, Betweenness

1 Introduction

The development and implementation of the 5G [1] network and Internet of Things (IoT) [2] brought large-scale mobile devices into the network, and it has resulted in great challenges for Internet service providers and users. The purpose of IoT is to connect every device to the Internet so that these devices can connect to the network at any time, at any place, and on any path. When billions of IoT devices are connected to the Internet, huge amounts of data will be generated. The production, processing, and transmission of large amounts of data have brought great challenges to the traditional TCP/IP network architecture. The traditional host-centric network architecture cannot meet the increasing demand of users for real-time video, streaming media, and other high-traffic data services. In order to overcome the challenges brought by the large-scale IoT, including the inefficiencies in data transmission and processing, the difficulties in achieving an efficient utilization of network resources, and poor user experience, it is imperative to design and research a novel and practical network architecture. Among the proposed novel network architectures, Information-Centric Networking (ICN) [3] based IoT has been extensively discussed [4].

ICN separates the network resources from their locations, and points out that the Internet should focus on data contents rather than the location of contents. Its main features contain in-network caching, data content naming, mobility management, scalability, and data security and privacy. These features make ICN a promising network model in the IoT environment. To verify the feasibility and solve the problem of the actual deployment of the ICN-IoT, some studies have discussed the techniques that can act as a bridge between IP and ICN networks. For example, Shannigrahi *et al.* [5] proposed a protocol named IPoC that can enable a transition to ICN in mobile networks by encapsulating and forwarding IP traffic over an ICN core. They pointed out that IPoC/NDN can benefit 5G mobile networks by simplifying handover operations and introducing intelligent multi-path strategies. Bracciale *et al.* [6] introduced a lightweight named object solution to represent physical IoT objects in a derived name space, and they pointed out that the proposed abstraction fully exploits most of the ICN benefits and can be used to support object interaction in IoT environments. At present, research on the ICN-based IoT has explored caching mechanisms [7], naming mechanisms [8], security [9], mobility schemes [10], and many other aspects. One of the main reasons why IoT uses ICN is that ICN cache can effectively

---

*Corresponding Author: Ying Liu; E-mail: yliu@bjtu.edu.cn

DOI: 10.53106/160792642021122207004

reduce the requests from customers to producers and improve the efficiency of content acquisition. As shown in Figure 1, ICN routers can simply cache contents to satisfy future requests for the same data, which can result in a reduced usage of network resources and faster contents retrieval for IoT devices. From this perspective, we can promote the network performance through adopting effective cache placement policies, cache replacement strategies and cache coherency mechanisms.

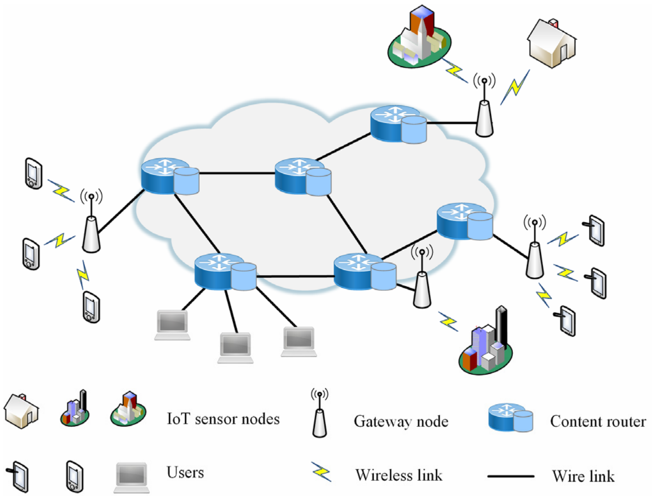

Figure 1. The sample of ICN-IoT

Named Data Networking (NDN) [11] is a representative implementation for ICN with each node keeping a Content Store (CS) [12]. The CS is responsible for caching contents, and the cached contents can satisfy the following reduplicative requests. This is one of the most important differences between ICN and the traditional TCP/IP network architecture. Compared with the huge amounts of data, the CS of the network node has limited cache capacity. As a result, how to effectively replace the content in CS and to reduce the content request miss probability as possible has become a key factor influencing the network performance [13].

In order to improve the cache efficiency in the context of the ICN-IoT, in this paper, we propose a cache replacement scheme based on content popularity and node betweenness. We set up an intra-domain resource adaptation resolving server (RARS) to maintain the cache status. There are two important parameters used to promote the performance of the cache replacement and to cache content more reasonably: the cache content popularity and the betweenness of the network node. The main contributions are as follows.

- We summarize the research status of the content cache schemes in the context of ICN-IoT.
- We propose a content popularity and betweenness based replacement scheme (PBRS), which integrates evaluating the popularity of caching contents and the betweenness of the nodes in the network topology.
- We propose to use different methods to calculate the popularity-like (popuL) value of the content according to their locations, which are classified into the edge router, the next hop router, and the complex situations.
- We conduct simulations to verify the proposed scheme. Simulation results show that the proposed cache replacement method makes the content storage more reasonable, and improves the hit ratio of the requests.

2 Related Work

2.1 The Packet Transmission in ICN

ICN focuses on the data content itself instead of the storage location of the data. It uses the name of the content as the only identifier instead of the IP address [14]. There are two types of packets in ICN - Interest packet and Data packet. When clients send out Interest packets, the Interest packets are routed to the neighbor node or the content provider, and then the Data packets are delivered back to the users along the reverse path [15]. As shown in Figure 2, when the Interest packet arrives, the intermediate node looks up the CS for the corresponding content. If the content exists, the Data packet will be returned to the requester directly. Otherwise, the node will check whether a related request entry exists in the Pending Interest Table (PIT). If the requested entry is found in the PIT, the node will add the incoming interface of the Interest packet into the list of the entry and drop the Interest packet; otherwise, the node will create a new entry and search the forward information base (FIB) for delivering the Interest packet to the next hop. When the Data packet arrives, the node will forward the packet according to the interface list of the PIT entry and cache the content in the CS. Different from the traditional client-server pattern, ICN nodes can cache data for a long term, and the Interest packets can obtain the corresponding Data packet in the intermediate node before arriving at the content server. Therefore, how to effectively replace the cached content has become one of the hotspots in ICN research.

2.2 The Cache Replacement Strategies

Generally, the ICN architecture adopts the least recently used (LRU), least frequently used (LFU), and first in, first out (FIFO) strategies as the traditional Web cache [16]. Although these strategies have been researched deeply in the traditional Web cache, Content Delivery Network (CDN), and Peer-to-peer (P2P) cache, there are some disadvantages when they are implemented in ICN. For example, they have not considered the network topology's influence on the cache performance. Muscariello *et al.* [17] pointed out that the LRU and the FIFO strategies cannot reflect the

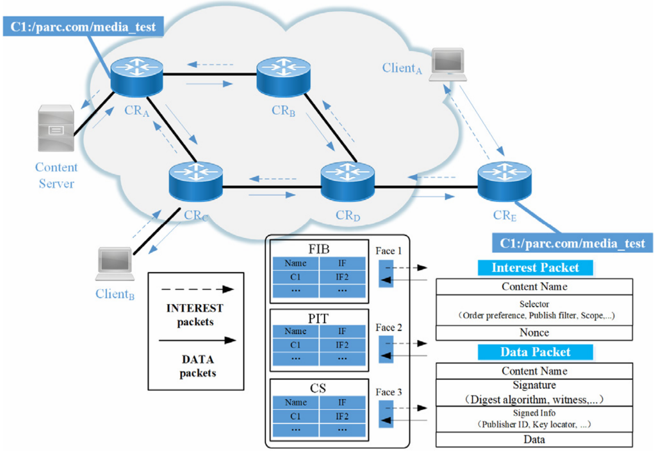

Figure 2. The transfer process of the Interest and Data packets in ICN

content popularity in real time. The LFU strategy cannot timely reflect the popularity of current requests due to its static characteristic. That is, if certain content is requested in large numbers within a period of time, leading to a high request frequency, then when the number of the requests decreases rapidly after that moment, the content will not be replaced in time because the previous high request frequency brings the content a higher cache priority. As a result, the cache space will be occupied for a long time, and thus the current high-popularity content cannot be cached. LRU-SP [18] is a cache replacement strategy designed for TCP/IP that considers the time interval of the last request and the request frequency. However, this strategy has some limitations when used in ICN, and the statistical analysis process is prone to bias. Chen *et al.* [19] proposed the least unified value (LUV)-Path strategy according to the characteristics of ICN, taking the cost of the content request into account. However, it cannot reflect the current status of the contents as it does not consider the content popularity. Rossi *et al.* [20] proposed a cache replacement strategy based on the popularity preference; it chooses two content blocks randomly and replaces the one that has the higher popularity. This design attempts to make the low-popularity content blocks stay in the cache for a longer period of time, thus ensuring that contents with different popularities in the network can be evenly distributed. However, there is the problem that the low-popularity content blocks may not be replaced for a long time, and thus the cache performance is not efficient enough. Zhang *et al.* [21] proposed to cache content according to the node utilization ratio and the popularity of content. However, the authors only compared their method with the simple random caching method, which is not novel enough. Besides, in our previous work [22], we have proposed a content popularity prediction algorithm based on an auto regressive model to promote the prediction accuracy of the content popularity. However, this work does not consider the cache replacement methods.

Researchers have also proposed many cache strategies to satisfy the demands of ICN-based IoT [23]. Vural *et al.* [24] promoted a model for the trade-off between multi-hop communication costs and the freshness of a transient data item. Results showed that the model could successfully reduce the network load, especially for frequently requested content. However, the cache replacement mechanisms were not taken into consideration. Niyato *et al.* [25] developed an analytical model to estimate the proposed cache mechanism, and used a threshold adaptation algorithm to adjust the cache parameters. The evaluation results showed a trade-off between cache and network cost. Meanwhile, Din *et al.* [7] suggested that the IoT applications well fitted to the ICN scenario. However, due to the special demands of devices, the traditional ICN cache strategies cannot be applied in the IoT environment. Hail *et al.* [26] analyzed the performance of different cache mechanisms in an ICN-IoT wireless network, and then proposed a forwarding strategy for a better content delivery performance. The replacement policy was LRU for every caching scheme. They conducted simulations to show that the designed caching and forwarding strategies can reduce the traffic volume and save the energy resources of the devices. Khodaparas *et al.* [27] proposed a novel cache mechanism named multi-criteria cooperative caching. This mechanism considers multiple attributes related to IoT and specifies their relative importance, and then decides whether each content should be cached; if so, it selects the most appropriate nodes to store the content. They also used the LRU cache replacement method. Song *et al.* [28] proposed a flexible and efficient scheme for the content sharing between mobile nodes and the local IoT network, named the tag-assisted content-centric networking (TCCN) scheme. TCCN decides whether to cache the content according to the tag filter. However, the authors did not indicate which cache replacement method was used. Meddeb *et al.* [29] considered the cache freshness requirement and proposed the least fresh first (LFF) algorithm as a cache replacement policy for ICN-based IoT networks. The LFF policy is based on time series analysis as a tool to predict future events, and therefore it can estimate the residual lifetime of cached contents. Hahm *et al.* [30] adapted the ICN protocol to new IoT scenarios, and the authors extended ICN with novel caching and replacement schemes combined with deep sleep strategies. The results showed that the cache replacement strategies could reduce energy consumption while maintaining a high availability of recent IoT contents. Since there have been many studies on the cache problem in ICN-IoT, it is important to discover effective content cache replacement methods to meet the demands of cache in ICN-IoT.

3 The Cache Replacement Scheme Based on Content Popularity and Betweenness

As a necessary constituent element in ICN, the in-network cache demands that a content router has the function of forwarding packets like traditional routers as well as caching the forwarded content. Once a content router receives Interest packets, it can reply to the Interest if the content cache has been stored in the CS. ICN emphasizes caching everywhere since the in-network caching can effectively decrease network traffic loads and transmission delays, and it can improve the service performance of future network architectures. A proper cache strategy makes the CS store the data that can better promote the cache hit ratio, reduce the stress of router nodes, and reduce the request latency of the clients. In this paper, we propose a content popularity and betweenness based cache replacement scheme. We introduce an intra-domain resource adaptation resolving server to assess the content popularity and the betweenness of the routers on different locations. Then we implement the in-network caching after weighing the importance of the network nodes and the content popularity.

3.1 Theory of Betweenness

Degree is the simplest index to evaluate the importance of complex network nodes. Generally, the node with the highest degree in the network is the most important node in the network. The betweenness is one of the measures that can quantify the importance of a node in the network topology. The value of the node betweenness can be calculated by the ratio of the number of shortest paths passing through the node and the total number of shortest paths in the network [31].

$G=(V,E)$ is an undirected graph with V nodes, and the betweenness of node v is denoted as $C_{B-SP}(v)$:

$$C_{B-SP}(v)=\sum_{s\neq v\neq t\in V}\frac{\delta_{st}(v)}{\delta_{st}}, \tag{1}$$

where δ_{st} represents the number of the shortest paths between node s and t, and $\delta_{st}(v)$ represents the number of the shortest paths that cross node v. The higher the betweenness, the more network traffic the node will take, the more contents the node will cache, and the higher the cache hit ratio.

3.2 Concept of the Content Popularity

The content popularity is an important factor in cache mechanisms. If contents with high popularity are cached on the nodes, the contents can be retrieved quickly.

Firstly, we analyze the traditional way of calculating the content popularity when using the LFU strategy in ICN. As shown in Figure 3, we assume that the requested contents are all cached in the content server, and that Router$_1$, Router$_2$, Router$_3$ and Router$_4$ do not have caches for Content$_1$ or Content$_2$ during time period T. Router$_1$ receives N Interest packets of Content$_1$ from User$_A$, User$_B$ and User$_C$, and one Interest packet of Content$_2$ from User$_A$. Router$_2$ receives one Interest packet of Content$_1$ and one of Content$_2$ from User$_1$. Since Router$_1$ and Router$_2$ cannot find Content$_1$ or Content$_2$ in the CS, the routers add new entries of Content$_1$ and Content$_2$ to the PIT and forward the Interest packets to the next hop. Then, Router$_3$ receives two Interest packets of Content$_1$ and two Interest packets of Content$_2$ in time period T. Since Router$_3$ cannot find Content$_1$ or Content$_2$ in the CS, it forwards the Interest packets to Router$_4$. Then, Router$_4$ continues to forward the Interest packets to the content server. When Router$_4$ receives Data$_1$ and Data$_2$ from the content server, it forwards the content data according to the interfaces recorded in the PIT. At the same time, Router$_4$ calculates the content popularity of Content$_1$ and Content$_2$, and makes a cache replacement decision based on the popularity values. Router$_3$ repeats the process of the content popularity calculation and the cache replacement when it receives Data$_1$ and Data$_2$.

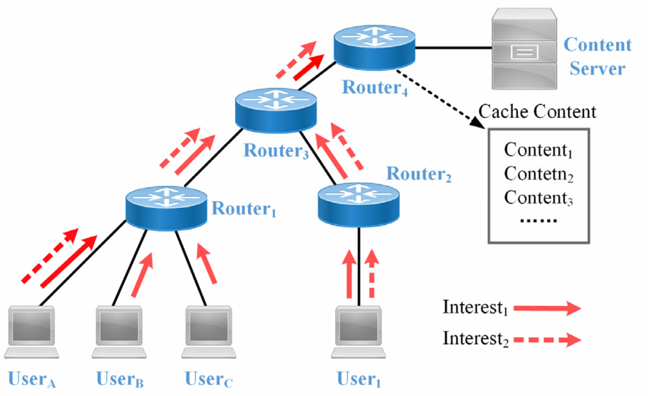

Figure 3. The sample of requesting for Content$_1$ and Content$_2$ in ICN

According to the above mechanism, Router$_3$ calculates the request frequency of Content$_1$ and Content$_2$ based on two Interest packets. However, the popularity value of Content$_1$ is much higher than that of Content$_2$ from a global perspective. Similarly, Router$_4$ calculates the request frequency of Content$_1$ and Content$_2$ based on one Interest packet, resulting in that the calculated popularity of Content$_1$ and Content$_2$ are the same. Based on the traditional LFU cache replacement strategy, if the reference value of Content$_1$ and Content$_2$ are the same, the router will replace the content randomly or replace the content according to the last request time. The reason for the difference between the calculated popularity and the actual value is that the content router executes aggregation when forwarding the Interest packets in ICN, and the PIT only records the interface information when the same Interest packet arrives rather than forwarding the

Interest packet as in traditional networks. The differences of the forwarding mechanisms lead to a different meaning of the content popularity. Specifically, the popularity based on the request frequency on Router$_3$ is restricted to itself, and it cannot be applied to the whole network.

ICN can introduce an controller to realize efficient caching and effective resource management [32-33]. In this paper, we use the resource adaptation resolving server to manage cache resources and to realize effective cache collaboration. Concretely, the content router informs the RARS of the cache information, and then the RARS maintains the cache state of the whole domain. When the RARS statistics the popularity of content, it is necessary to adopt a uniform and comparable calculation method. To solve this problem, we introduce a reference value named popularity-like. On this basis, the popularity and betweenness-based replacement scheme is proposed.

3.3 The Popularity-Like (popuL) Value

The popularity-like (popuL) is a parameter combining the betweenness and the prediction value of content popularity. Chai et al. [34] verified that the information centric networks could be more effective by caching the requested content only on a specific subset of nodes in the content distribution path. The authors put forward a cache strategy based on the betweenness centrality of nodes to cache the content on more important nodes, so as to improve the cache hit ratio. To calculate the betweenness of the nodes in Figure 3, we simplify the topology into Figure 4. Then the betweenness of nodes can be calculated as formula (2).

$$C_{B-SP}(A)=2*\left\{\begin{array}{l}\left(\frac{\delta_{U1C}(A)}{\delta_{U1C}}\right)+\left(\frac{\delta_{U1D}(A)}{\delta_{U1D}}\right)+\left(\frac{\delta_{U1S}(A)}{\delta_{U1S}}\right)+\left(\frac{\delta_{U1B}(A)}{\delta_{U1B}}\right)+\left(\frac{\delta_{U1U2}(A)}{\delta_{U1U2}}\right)\\+\left(\frac{\delta_{U2B}(A)}{\delta_{U2B}}\right)+\left(\frac{\delta_{U2C}(A)}{\delta_{U2C}}\right)+\left(\frac{\delta_{U2D}(A)}{\delta_{U2D}}\right)+\left(\frac{\delta_{U2S}(A)}{\delta_{U1S}}\right)\\+\left(\frac{\delta_{BC}(A)}{\delta_{BC}}\right)+\left(\frac{\delta_{BD}(A)}{\delta_{BD}}\right)+\left(\frac{\delta_{BS}(A)}{\delta_{BS}}\right)\\+\left(\frac{\delta_{CD}(A)}{\delta_{CD}}\right)+\left(\frac{\delta_{CS}(A)}{\delta_{CS}}\right)+\left(\frac{\delta_{DS}(A)}{\delta_{DS}}\right)\end{array}\right\}=2*(1+1+1+1+1+0+\cdots+0)=10. \quad \textbf{(2)}$$

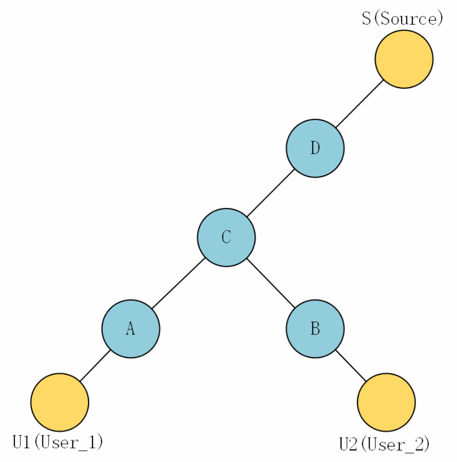

Figure 4. Simplified ICN topology

Similarly, we can obtain $C_{B-SP}(A)$, $C_{B-SP}(B)$, $C_{B-SP}(C)$ and $C_{B-SP}(D)$, as formula (3) and (4).

$$C_{B-SP}(A)=C_{B-SP}(B)=C_{B-SP}(D)=10. \quad \textbf{(3)}$$

$$C_{B-SP}(C)=24. \quad \textbf{(4)}$$

In this section, we introduce the detailed calculation method of the popuL value for nodes at different locations.

3.3.1 The Popularity-Like Value of the Edge Router

For the edge routers in Figure 5, the popularity-like value can be calculated by the ratio between the number of the Interest packets for content i and the total number of Interest packets received by the router. Then, $popuL_i$ is denoted as below:

$$popuL_i=\frac{reqs_{iT}}{\sum_{j=1}^{k}reqs_{jT}}, \quad \textbf{(5)}$$

where k is the total amount of the caching files in the edge router, $reqs_{iT}$ represents the number of the received requests for content i within time period T, and $\sum_{j=1}^{k}reqs_{jT}$ is the total number of requests within T. After obtaining the value of $popuL_i$, the edge router appends $popuL_i$ and the betweenness value to the Interest packet and forwards the packet to the next hop. When the corresponding Data packet is replied, the edge router reports the $popuL_i$ to the RARS through the cache information notification message between the edge router and the RARS.

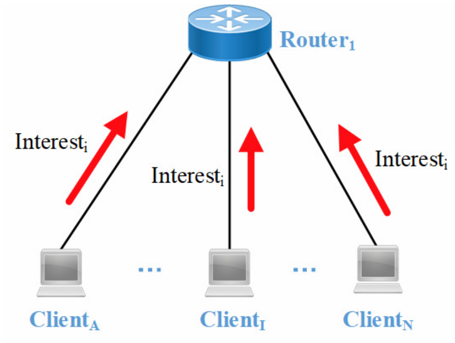

Figure 5. Calculating the popularity-like value of the edge router

3.3.2 The Popularity-Like Value of the Next-Hop Router

The Interest packet of content that the next-hop router received from the last-hop Router$_n$ contains the following information:

(a) $popuL_i(n)$: the popularity-like value of content i on Router$_n$.

(b) $C_{B-SP}(n)$: the betweenness value of Router$_n$.

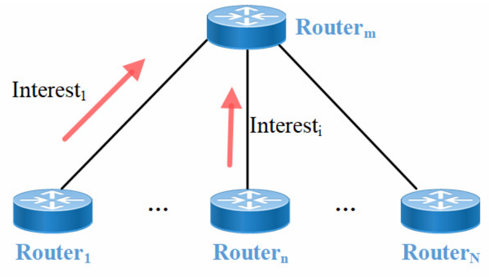

Figure 6. Calculating the popularity-like value of the next-hop router

Since there is not exist direct correlation between the value of content popularity and the betweenness, the next-hop router needs to normalize the two metrics when it calculates the popularity-like value. As shown in Figure 6, when Router$_m$ receives j Interest packets from Router$_1$ to Router$_n$, it computes the $popuL_i(m)$ of content i by (6).

$$popuL_i(m) = \frac{n}{N+1} * \frac{\sum_n popuL_i(n) * C_{B-SP}(n)}{\sum_n C_{B-SP}(n)}, \qquad (6)$$

where N represents the number of last-hop routers that connected to Router$_m$, n represents the number of routers that send Interest packets of content i to Router$_m$, and $\sum_n C_{B-SP}(n)$ represents the weighted sum of the betweenness of the routers that send Interest packets of content i to Router$_m$.

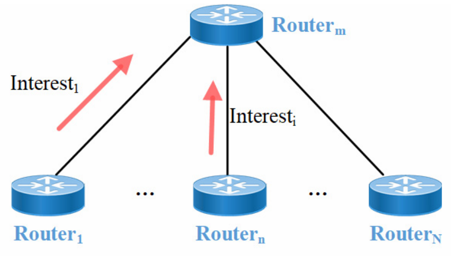

Figure 6. Calculating the popularity-like value of the next-hop router

If Router$_m$ needs to forward the Interest packet of content i to other routers after calculating the $popuL_i(m)$, it will renew the value of $popuL_i(m)$ and the betweenness of Router$_m$ into the Interest packet. By comparing the popularity-like value of the contents on different routers, it can be achieved to estimate which contents are popular and which are unpopular.

3.3.3 The Popularity-Like Value under Complex Situations

As shown in Figure 7, considering the situation that many clients connect to Router$_3$, the popularity-like value on Router$_3$ can be calculated in an equivalent way. We regard all the users connected to Router$_3$ as if they were connected to a virtual router Router$_V$, and then $C_{B-SP}(V) = C_{B-SP}(3)$, and the popularity-like value of content i on Router$_3$ can be obtained using (6).

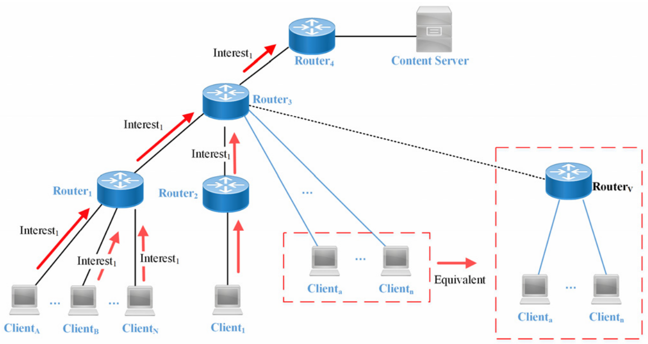

Figure 7. Calculating the popularity-like value under complex situations

4 Implementation of the PBRS

Based on the analysis above, this section introduces how to realize the proposed cache replacement method, that is, how to replace the content cache of low popularity with content of high popularity. The cache replacement strategy is detailed below.

Step 1: The RARS collects the information of the intra-domain topology, computes the betweenness values of the routers, and delivers the betweenness information to corresponding routers by the cache notification message. When the network topology changes, the RARS will update the betweenness values and inform the related routers.

Step 2: The edge router counts the number of Interest packets it receives of each content in terms of period, and stores the results in the popularity table. When the Data packets arrive, the edge router calculates the popularity-like value and stores the result in the popularity-like table. When the popularity-like table of the cache contents changes, the edge router uploads the notification information to the RARS to update the popularity-like table, to ensure that the popularity-like table on RARS is synchronized with the table in each router.

Step 3: The edge router adds the popularity-like value and its betweenness into the Interest packet when it delivers the packet to the next-hop router. The next-hop router records these values and computes the popularity-like value of content i using formula (6) when it receives the corresponding Data packet. Then, it stores the value in the popularity-like table and synchronizes with the RARS.

Step 4: The intermediate router adds the popularity-like value and its betweenness into the Interest packet when it delivers the packet to the next-hop router. The next-hop router records these values and computes the popularity-like value of the contents using (6) when it receives the Data packets; then, it stores the value in the popularity-like table and synchronizes with the RARS.

Step 5: When a router needs to replace the cache, it looks up the popularity-like table to decide the cache replacement strategy. Since the number of the entries of the popularity-like table is larger than the cache capacity, only the top-ranked content is cached in the local routers.

The content popularity and betweenness-based replacement scheme can be refined into the following four algorithms: the betweenness update algorithm in the RARS, the popularity-like value update algorithm in the RARS, the popularity-like value compute algorithm in the edge router, and the popularity-like value compute algorithm in the next-hop router.

4.1 The Betweenness Update Algorithm in the RARS

In this paper, the RARS manages the connections and load status of all intra-domain routers. Assuming that there are n routers within the domain, firstly, the RARS needs to establishe a $n*2$ size intra-domain betweenness table to store the betweenness values of the n routers. If the topology changes, the RARS recalculates the betweenness value and informs all of the intra-domain routers to refresh the betweenness value of the nodes by the cache notification massages. The form of the betweenness table is {node, betweenness}, as shown in Figure 8. The betweenness update algorithm in the RARS is detailed in Algorithm 1.

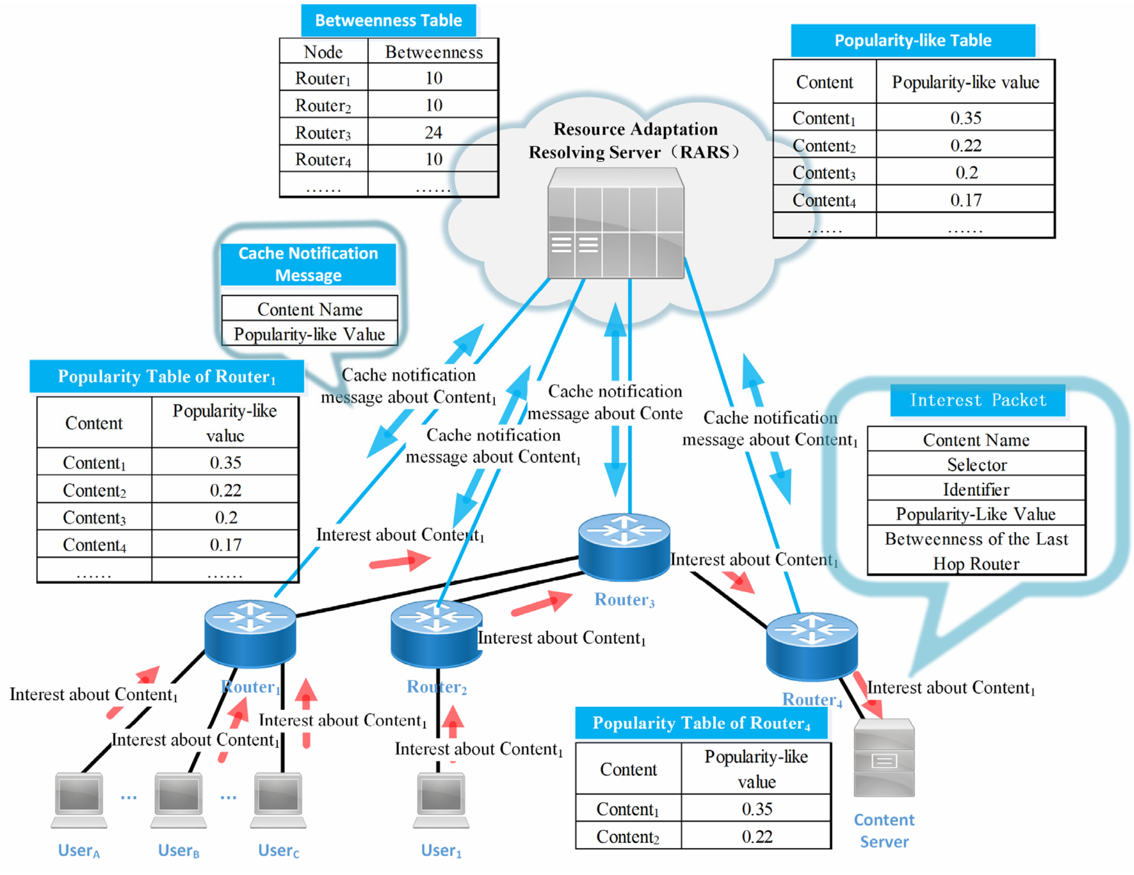

Figure 8. The content popularity and betweenness-based cache replacement strategy

Algorithm 1. The Betweenness Update Algorithm in the RARS

```
/* Establish a $n*2$ sized table in the RARS to store the betweenness values of the $n$ routers*/
Btw_Refresh():
Each node reports its connection information;
for $i=1:n$ do
    $C_{B-SP}(v)=\sum_{s\neq v\neq t\in V}\frac{\delta_{st}(v)}{\delta_{st}}$
end for
The RARS informs the betweenness value to routers
```

4.2 The Popularity-like Value Update Algorithm in the RARS

The RARS needs to establish a $m*2$ size dynamic table as a popularity-like table in the RARS to maintain all the intra-domain popularity-like values of the cache contents. Once the popularity-like value of a content in an intra-domain router changes, the router informs the RARS by the cache notification massages. When the RARS updates the popularity-like table, it will inform all intra-domain routers to keep the same value with itself. The dynamic table can be accomplished by dynamic containers such as the maximum heap, the red black tree, and the search tree to keep the {content name, popularity-like} values in order. In this paper, we recommend using the search tree since this kind of dynamic container can effectively decrease the complexity of the algorithm. The intra-domain betweenness table is shown in Figure 8. Algorithm 2 shows the popularity-like value update algorithm in the RARS.

Algorithm 2. The Popularity-like Value Update Algorithm in the RARS

```
/* Establish an $m*2$ sized table as a cache contents popularity-like table in the RARS */
/* The table keeps the {content name, popularity-like} values, using maximum heap arrangement*/
/* $m$ is greater than or equal to the maximum number of contents that can be stored by any router in the domain*/
PopuL_Refresh():
if RARS receives the popularity-like value $s$ of content $i$ from node $M$ then
    if content $i$ is in the table and the popularity-like value is $x$ then
        if $s > x$ then
            $x = s$
            RARS returns the cache notification message carrying the $\{i,s\}$ values to $M$
        else
            RARS returns the cache notification message carrying the $\{i,x\}$ values to $M$
        end if
    else
        if $s > \min\{popularity-like\ value\}$ then
            Insert the $\{i,s\}$ values to the intra-domain popularity-like table
            Issue the cache notification messages carrying the $\{i,s\}$ values to intra-domain routers
        end if
    end if
end if
```

4.3 The Popularity-Like Value Compute Algorithm in the Edge Router

When the Interest packets arrive at the edge content router, the global request counter $Counter$ in the current m_{th} period and the request counter $Counter_i$ of content i in the current period p_m need to complete an auto-increment operation. In addition, when the current period ends, that is, when the number of Interest packets in the current period reaches N, the compute process of the request frequency of content i is triggered. Then $Counter$ and $Counter_i$ are cleared and a new compute process of the request frequency begins. The popularity-like value compute algorithm in the edge router is presented in Algorithm 3.

Algorithm 3. The Popularity-Like Value Compute Algorithm in the Edge Router

```
/*In the edge router, the request frequency of content $i$ in time period m is $p_{im}$ */
Fre_Count():
    $Counter = Counter + 1$
    $Counter_i = Counter_i + 1$
    if $Counter = N$ then
        $p_{im} = Counter_i / Counter$
        $Counter = Counter_i = 0$
        $m = m + 1$
    end if
```

4.4 The Popularity-Like Value Compute Algorithm in the Next-Hop Routers

The next-hop router Router_m may receive Interest packets of content i from other routers. Router_m extracts the values of $popuL_i(n)$ and $C_{B-SP}(n)$ in the Interest packets and computes $popuL_i(m)$ using formula (6). Then it reports the cache information notification to the RARS to update the popularity-like table. The popularity-like value compute algorithm in the next-hop router is shown in Algorithm 4.

Algorithm 4. The Popularity-Like Value Compute Algorithm in the Next-Hop Routers

```
/*The next-hop router Router_m caches $n$ contents and they all have pre-popularity*/
/*The content that has the minimum popularity-like value is content $o$ and the value is $popuL_o(m)$ */
/*The maximum cache contents in Router_m is $Max\_Capacity$ and $Max\_Capacity \geq n$ */
PopuL_Calculation_NR_Proc ():
Input: Interest packet of content $i$ Router_m received
    Calculate $popuL_i(m)$ using formula (6)
    if content $i$ is one of the $n$ contents then
        Refresh $popuL_i(m)$
        Initiate a request of refreshing the popularity-like value of content $i$ to RARS by the cache notification message
    else
        if $n = Max\_Capacity$ then
            if $popuL_i(m) > popuL_o(m)$ then
                Discard content $o$
                Insert content $i$ into the popularity- like table
                Send a request of refreshing the popularity- like value of content $i$ to the RARS by the cache notification message
            else
                Discard content $i$
            end if
        end if
    end if
```

The main concept of the popularity-like based cache replacement strategy is to use an intra-domain RARS to dynamically maintain a popularity-like table. When the intra-domain routers need cache replacement, the routers decide whether to replace the cache by comparing the popularity-like value of the replaced content and the alternative content in the popularity-like table. This mechanism can achieve an efficient use of caching resources.

4.5 Complexity Analysis

For the RARS, the time complexity of the algorithm of betweenness calculation is relatively high, and the highest value is $o(n^3)$. Even with the simplified algorithm, the complexity is still up to $o(n^2)$. However, the algorithm of betweenness calculation is triggered only when the network topology changes. Specifically, the algorithm is triggered at the beginning of the establishment of the network to calculate the betweenness value of each node. If the network remains stable, the algorithm will not be triggered frequently. Therefore, when considering the time complexity of the proposed cache replacement algorithm, the time complexity of the algorithm of betweenness calculation can be ignored.

For the edge router, when the k_{th} request arrives, the running time of the algorithm contains querying the historical popularity statistics table, updating the statistics table, and the operation on the priority queue. The average time complexity of these operations is $o(\log k)$, $o(1)$ and $o(\log s)$, respectively. The popularity update algorithm adds an additional $o((s/\phi)\log k)$. time complexity per ϕ application. Therefore, for each newly arrived request, the average time complexity of the algorithm is $o((1+s/\phi)\log k + \log s + 1) \approx o(\log k)$.

The space complexity analysis of the algorithm is more complex. Two steps need large space. One is that the edge router needs to maintain two kinds of dynamic tables, the historical content popularity statistics table and the cached content popularity table. The other is that the RARS needs to maintain two tables, the in-domain cache popularity-like value table and the in-network node betweenness table. In the worst case, the space complexity of these four tables is $o(K^{d/_{d+x+y}})$. In the best case, the space complexity is close to $o(\log K)$. However, the space complexity will not increase with the number of requests, but will gradually tend to a stable range, which is ultimately limited by the size of cache space. In conclusion, it is an inevitable result that the proposed method trades space complexity for relatively low time complexity.

5 Evaluation

In this section, we evaluate our proposed mechanism using ndnSIM [35] simulator based on NS-3. The main performance parameters considered include the cache hit ratio for routers at different locations. The environment used in our experiment is an Intel(R) Core(TM)2 Duo CPU E7500@2.93GHz, with 2GB memory, and the operating system is Ubuntu 14.04 LTS 32bit. The version of ndnSIM is ndnSIM2.4.

We use the file hit ratio (FHR) to measure the performance of the different cache replacement strategies. FHR is the ratio of the number of hit requests and the total number of requests:

$$FHR = \frac{\sum_{i \in R} h_i}{\sum_{i \in R} f_i}, \tag{7}$$

where $\sum_{i \in R} h_i$ represents the number of hit requests, and $\sum_{i \in R} f_i$ represents the total number of requests. The effects of parameters adjustments on FHR are listed below.

(1) The effect of the cache size: The larger the cache size, the more contents can be cached, and the greater

the probability that the users will get the requested contents on the cache routers. Thus, the value of FHR increases with the increase of the cache size.

(2) The effect of $Zipf(\alpha)$: With the increase of α, the distribution of the popular content becomes more centralized. The contents with high popularity are more easily cached and not easily replaced.

The main objectives of the simulation contain: (1) Complete the modification of the Interest packets in ndnSIM. By adding relevant fields of the popularity-like value, the mechanism can be evaluated more practically. (2) Realize the PBRS by modifying and configuring the cache replacement strategies in each cache node, and compare the performance of the PBRS with the traditional LRU, LFU, and FIFO cache replacement strategies.

We use the linear topology, the binary tree topology and the NFSNET extension topology to assess the proposed mechanism.

5.1 Evaluation Results under the Linear Topology

The linear topology is shown in Figure 9. There are 2 clients, 1 edge router, 4 next-hop routers and 1 content server. The request arrival obeys the Poisson distribution with $\lambda = 100$, and the request probability follows the zipf distribution. Considering the demands of the clients on the contents, the number of contents, and the evaluation topologies, we configure the evaluation parameters as listed in Table 1.

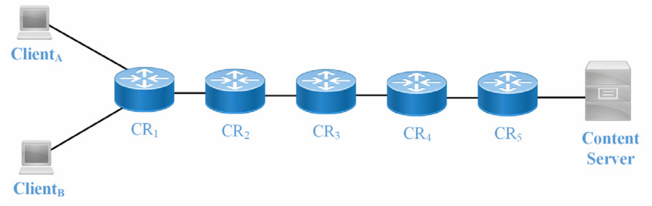

Figure 9. The linear topology

Table 1. Parameters configuration

Parameter	Value	Variation range
Number of contents	1000	200-3200
Node cache size (MB)	70	10-190
Parameter α	0.7	0.5-1.0
Evaluation time (s)	200	200

The FHR on the edge router and the next-hop router of the 5-node linear topology is shown in Figure 10 and Figure 11, respectively.

The results indicate that the proposed cache replacement strategy shows better performance than the LFU cache replacement algorithm, and far better than that of LRU and FIFO cache replacement algorithm. Moreover, as the cache capacity increases, the advantage of the PopuL-based cache replacement strategy becomes more obvious.

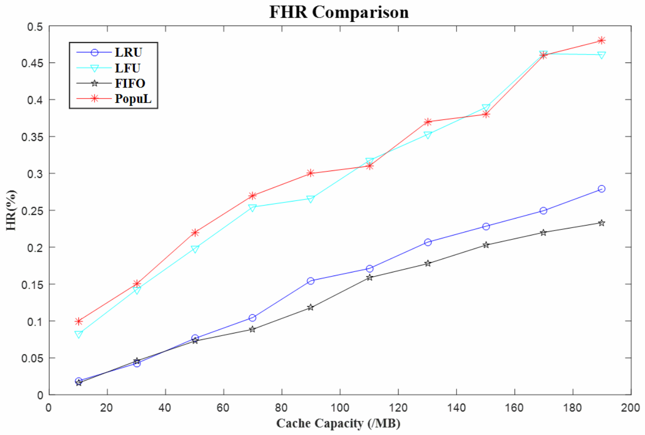

Figure 10. The FHR of the edge router in the linear topology

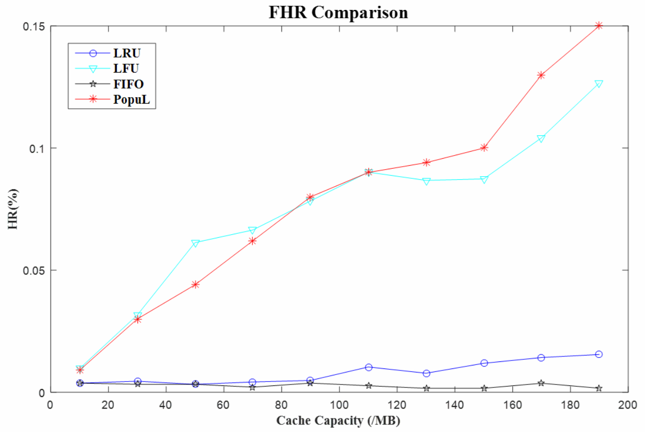

Figure 11. The FHR of the next-hop router in the linear topology

5.2 Evaluation Results under the Binary Tree Topology

The binary tree topology is shown in Figure 12. There are total 12 nodes: 4 clients, 4 edge routers, 3 next-hop routers, and 1 content server. The FHR on the edge router and the next-hop router of the binary tree topology is shown in Figure 13 and Figure 14, respectively.

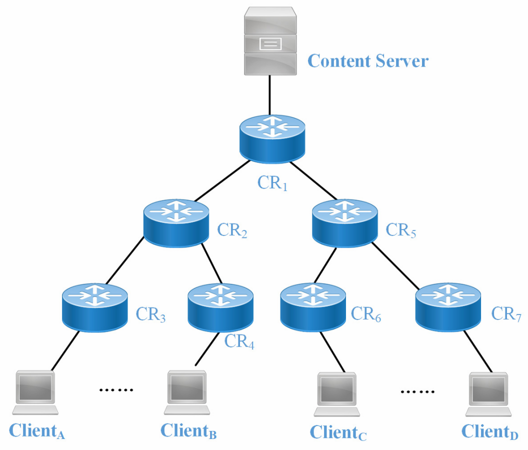

Figure 12. The binary tree topology

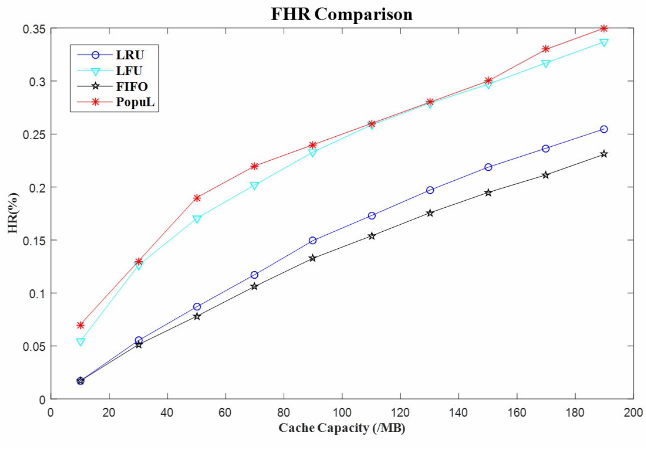

Figure 13. The FHR of the edge router in the binary tree topology

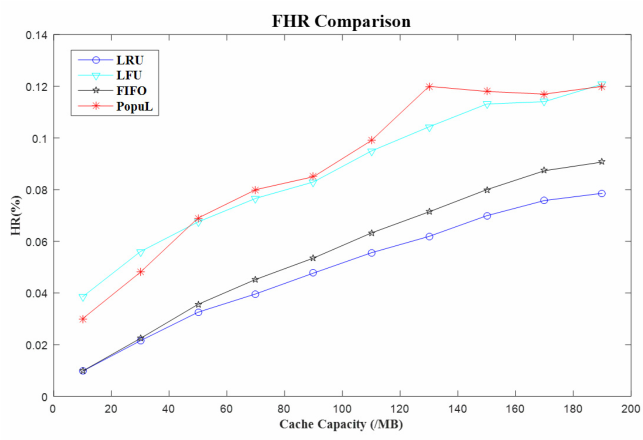

Figure 14. The FHR of the next-hop router in the binary tree topology

Similar to the linear topology, by adjusting the cache capacity of the node, we obtain a line graph of the FHR using four different cache replacement strategies. In this simulation topology, CR3, CR4, CR6 and CR7 are edge routers that only receive the Interest packet from the clients, and other nodes are next-hop routers. The figure shows that the performance of proposed strategy based on the betweenness and the content popularity is better than that of the LFU algorithm, and far better than that of LRU and FIFO cache replacement algorithm. Moreover, with the increasing cache space of network topology and the complexity of topology environment, the advantage of the proposed PBRS becomes more obvious.

5.3 Evaluation Results under the NFSNET Extension Topology

In order to better verify the cache replacement strategy designed in this paper, we adopt a more complex NFSNET topology for simulation. The NFSNET extension topology is depicted in Figure 15 with 32 nodes, which contain 14 clients, 4 edge routers, 10 next-hop routers, and 4 content servers.

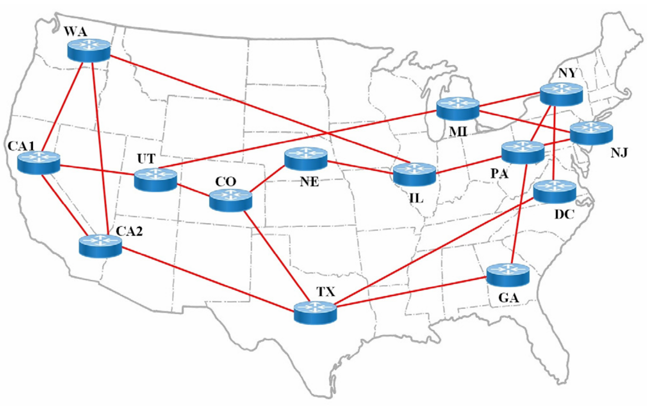

Figure 15. The NFSNET topology

In Figure 16 and Figure 17, we present the FHR on the edge router and the next-hop router under different cache replacement strategies. As the conclusion under the linear topology and the binary tree topology, simulation results indicate that the performance of the PBRS under the NFSNET extension topology is better than that of the LFU strategy, and far better than that of the LRU and FIFO strategy. Besides, with the increase in the cache capacity of the network topology, the PBRS shows a greater advantage.

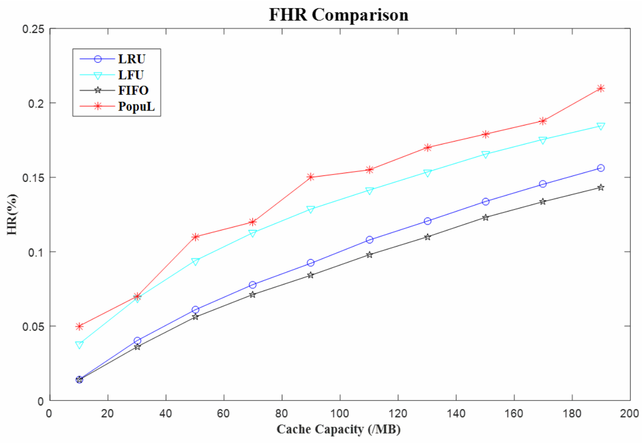

Figure 16. The FHR of the edge router in the NFSNET extension topology

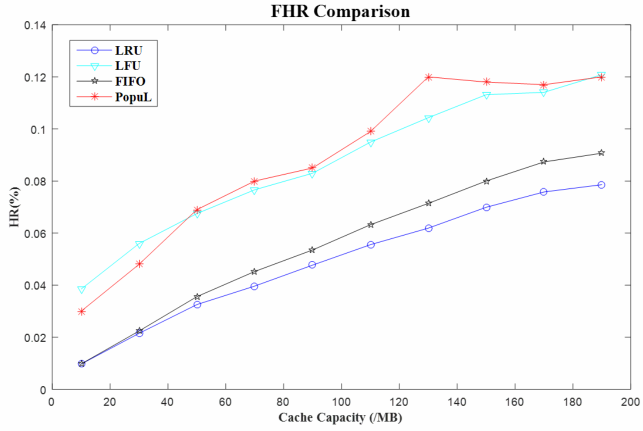

Figure 17. The FHR of the next-hop router in the NFSNET extension topology

In this paper, we propose to replace the cache according to the rank of the popularity-like value of the cache contents. The evaluation results of the popularity-like based cache replacement strategy are summarized as follows:

(1) The proposed content popularity and betweenness based cache replacement strategy uses the centralized decision-making method to effectively decrease the frequency of cache replacement. (2) The strategy sorts and analyzes the content and the betweenness of the routers to realize a reasonable resource allocation, and to promote the efficiency of the whole caching system. (3) The strategy takes the connection between time and space of the cache network into consideration, adds the local factors, and establishes the request correlation model. (4) In our proposed cache replacement strategy, the computation of the popularity has not changed too much when compared to traditional methods, and the betweenness of the routers can be obtained through precomputation after the establishment and stabilization of the network topology. The computations of these two parameters can be completed under a low-load condition and does not require too much system resources. Therefore, the utility of the caching system is overall improved.

6 Conclusion

To solve the problem that the existing cache replacement strategies have not considering the relationship between the content popularity and the network topology, in this paper, we propose an in-network cache replacement scheme based on content popularity and betweenness in the context of ICN-IoT. The cache replacement algorithm takes into account the betweenness of network nodes, and requires the routers to decide whether the cache replacement behavior occurs according to the real-time situation. The simulation results show that the strategy can achieve an effective cache replacement by introducing an intra-domain RARS to maintain the cache status while considering the popularity of the cache contents and the betweenness value of the network nodes. The proposed strategy makes the content storage more reasonable and improves the hit rate of the requests when comparing with traditional cache replacement mechanisms. In our future work, we plan to discuss the intra-domain cooperative cache deployment scheme to further improve the cache efficiency.

Acknowledgements

This work was supported by National Key R&D Program of China under Grant No. 2018YFA0701604.

References

[1] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, G. Wunder, 5G: A tutorial overview of standards, trials, challenges, deployment, and practice, *IEEE Journal on Selected Areas in Communications*, Vol. 35, No. 6, pp. 1201-1221, June, 2017.

[2] I. Ud Din, M. Guizani, B. Kim, S. Hassan, M. K. Khan, Trust management techniques for the internet of things: A survey, *IEEE Access*, Vol. 7, pp. 29763-29787, March, 2019.

[3] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, A survey of information-centric networking research, *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 2, pp. 1024-1049, Secondquarter, 2014.

[4] S. Arshad, M. A. Azam, M. H. Rehmani, J. Loo, Recent advances in information-centric networking-based internet of things (icn-iot), *IEEE Internet of Things Journal*, Vol. 6, No. 2, pp. 2128-2158, April, 2019.

[5] S. Shannigrahi, C. Fan, G. White, Bridging the icn deployment gap with ipoc: An ip-over-icn protocol for 5g networks, *Proceedings of Workshop on Networking for Emerging Applications and Technologies (NEAT'18)*, Budapest, Hungary, 2018, pp. 1-7.

[6] L. Bracciale, P. Loreti, A. Detti, R. Paolillo, N. B. Melazzi, Lightweight named object: An icn-based abstraction for iot device programming and management, *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 5029-5039, June, 2019.

[7] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, A. Habbal, Caching in information-centric networking: Strategies, challenges, and future research directions, *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 2, pp. 1443-1474, Secondquarter, 2018.

[8] M. F. Majeed, S. H. Ahmed, M. N. Dailey, Enabling push-based critical data forwarding in vehicular named data networks, *IEEE Communications Letters*, Vol. 21, No. 4, pp. 873-876, April, 2017.

[9] J. Granjal, E. Monteiro, J. S Silva, Security for the internet of things: A survey of existing protocols and open research issues, *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 3, pp. 1294-1312, Thirdquarter, 2015.

[10] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, Producer mobility support in named data internet of things network, *Procedia Computer Science*, Vol. 109, pp. 1067-1073, 2017.

[11] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, *ACM SIGCOMM Computer Communication Review*, Vol. 44, No. 3, pp. 66-73, July, 2014.

[12] M. M. S. Soniya, K. Kumar, A survey on named data networking, *Proceedings of the 2nd International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, India, 2015, pp. 1515-1519.

[13] G. M. de Brito, P. B. Velloso, I. M. Moraes, *Information-Centric Networks: A New Paradigm for the Internet*, Wiley-ISTE Press, 2013.

[14] W. Quan, C. Xu, J. Guan, H. Zhang, L. A. Grieco, Scalable name lookup with adaptive prefix bloom filter for named data networking, *IEEE Communications Letters*, Vol. 18, No. 1, pp. 102-105, January, 2014.

[15] T. Zhi, H. Luo, Y. Liu, A gini impurity-based interest flooding attack defence mechanism in ndn, *IEEE Communications Letters*, Vol. 22, No. 3, pp. 538-541, March, 2018.

[16] G. Zhang, Y. Li, T. Lin, Caching in information centric networking: A survey, *Computer Networks*, Vol. 57, No. 16, pp. 3128-3141, November, 2013.

[17] L. Muscariello, G. Carofiglio, M. Gallo, Bandwidth and storage sharing performance in information centric networking, *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*, Toronto, Canada, 2011, pp. 26-31.

[18] K. Cheng, Y. Kambayashi, LRU-SP: a size-adjusted and popularity-aware LRU replacement algorithm for web caching, *Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000*, Taipei, Taiwan, 2000, pp. 48-53.

[19] X. Chen, Q. Fan, H. Yin, Caching in information-centric networking: From a content delivery path perspective, *9th International Conference on Innovations in Information Technology (IIT)*, Al Ain, United Arab Emirates, 2013, pp. 48-53.

[20] D. Rossi, G. Rossini, *Caching performance of content centric networks under multi-path routing (and more)*, Telecom ParisTec Tech. Rep., December, 2011.

[21] M. Zhang, P. Xie, J. Zhu, Q. Wu, R. Zheng, H. Zhang, NCPP-based caching and NUR-based resource allocation for information-centric networking, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 5, pp. 1739-1745, May, 2019.

[22] Y. Liu, T. Zhi, H. Xi, X. Duan, H. Zhang, A novel content popularity prediction algorithm based on auto regressive model in information-centric iot, *IEEE Access*, Vol. 7, pp. 27555-27564, March, 2019.

[23] I. Ud Din, H. Asmat, M. Guizani, A review of information centric network-based internet of things: communication architectures, design issues, and research opportunities, *Multimedia Tools and Applications*, Vol. 78, No. 21, pp. 30241-30256, November, 2019.

[24] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, R. Tafazolli, In-network caching of internet-of-things data, *IEEE International Conference on Communications (ICC)*, Sydney, Australia, 2014, pp. 3185-3190.

[25] D. Niyato, D. I. Kim, P. Wang, L. Song, A novel caching mechanism for internet of things (iot) sensing service with energy harvesting, *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1-6.

[26] M. A. M. Hail, M. Amadeo, A. Molinaro, S. Fischer, On the performance of caching and forwarding in information-centric networking for the iot, *International Conference on Wired/Wireless Internet Communication*, Malaga, Spain, 2015, pp. 313-326.

[27] S. Khodaparas, S. Yousefi, A. Benslimane, A multi criteria cooperative caching scheme for internet of things, *IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

[28] Y. Song, H. Ma, L. Liu, TCCN: Tag-assisted content centric networking for internet of things, *IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Boston, MA, USA, 2015, pp. 1-9.

[29] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, H. Mathkour, Least fresh first cache replacement policy for ndn-based iot networks, *Pervasive and Mobile Computing*, Vol. 52, pp. 60-70, January, 2019.

[30] O. Hahm, E. Baccelli, T. C. Schmidt, M. Wahlisch, C. Adjih, A named data network approach to energy efficiency in iot, *IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, 2016, pp. 1-6.

[31] K.-I. Goh, E. Oh, B. Kahng, D. Kim, Betweenness centrality correlation in social networks, *Physical Review E: Covering Statistical, Nonlinear, Biological, and Soft Matter Physics*, Vol. 67, No. 1 Pt 2, Article No. 017101, January, 2003.

[32] B. J. Ko, V. Pappas, R. Raghavendra, Y. Song, R. B. Dilmaghani, K.-w. Lee, D. Verma, An information-centric architecture for data center networks, *Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking*, Helsinki, Finland, 2012, pp. 79-84.

[33] N. L. M. Van Adrichem, F. A. Kuipers, Ndnflow: Software-defined named data networking, *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, UK, 2015, pp. 1-5.

[34] W. Chai, D. He, I. Psaras, G. Pavlou, Cache “less for more” in information-centric networks (extended version), *Computer Communications*, Vol. 36, No. 7, pp. 758-770, April, 2013.

[35] S. Mastorakis, A. Afanasyev, I. Moiseenko, L. Zhang, *Ndnsim 2.0: A new version of the ndn simulator for ns-3*, NDN-0028, January, 2015.

Biographies

Ying Liu is a professor at the School of Electronic and Information Engineering, Beijing Jiaotong University. She received her M.S. and Ph.D. degrees from Beijing Jiaotong University in 2003 and 2012, respectively. Her current research interests include network architecture, network security, protocols optimization, wireless communications and cloud computing.

Ting Zhi received her B.S. and Ph.D. degrees from Beijing Jiaotong University in 2014 and 2020, respectively. Currently, she is working at CETC Big Data Research Institute Co., Ltd. Her research interests are network security, information centric networking and data analysis.

Huachun Zhou is a professor in National Engineering Laboratory for Next Generation Internet Interconnection Devices at Beijing Jiaotong University, China. His main research interests are in the area of mobility management, mobile and secure computing, routing protocols and network security.

Haidong Xi received the M.S. degree from Beijing Jiaotong University in 2018. Currently, he is working at Cisco System (China) Research and Development Co., Ltd. Hangzhou Branch Office. His research interests are network security and information centric networking.