

A Collaborative Cache Preloading Algorithm Based on Data Correlation in the Content Centric Networking

Wanjin Xu^{1,a}, Jianyong Sun^{1,b}, Haifei Qin^{1,c}, Chunlin Wang^{1,d}, Xiaolin Chen^{1,e,*}

¹ School of Information Science & Technology, Chuxiong Normal University, P.R.China

^axwj_cx@cxtc.edu.cn, ^bsunny@cxtc.edu.cn, ^cqhfei@cxtc.edu.cn, ^dw123@cxtc.edu.cn,
^echenxl@cxtc.edu.cn

Corresponding author: Xiaolin Chen, chenxl@cxtc.edu.cn

Keywords: Content Centric Networking, collaborative caching, preloading

Abstract. Content Centric Networking is a new type of network architecture, which is developed for the scalability and content distribution of the TCP/IP protocol over the Internet. Each node in the network has the ability of data caching, intermediate nodes and content providers can response to content request. The strategy of content caching is a core issue of the content centric networking. If the data of higher access possibility are preloaded into the cache node, the delay of data request will be reduced. In this paper, a correlation based cache preloading algorithm in Content Centric Networking is purposed. Based on the analysis of user's access behavior, the relationship among the data fragments is described by relevancy. Simulation experiment shows that the proposed algorithm can reduce the delay time of data request by about 9.57% in the case of data fragments with high correlation.

1 Introduction

With the continuous development of the Internet applications, TCP/ IP-based network exposes more and more drawbacks. Content-Centric Networking (CCN) [1] is a new network architecture, which focused on the scalability and content distribution and other issues of the TCP / IP based Internet, aimed for solve the current problems. CCN provides a possible technology roadmap in the network, has become a hot topic in the future of the network [2].

CCN redesign the methods of network routing, caching, security, etc., in content-centered style. Contents are named in the CCN network. Instead of IP address, content name is used as the network traffic identification, and it is addressable and routable. Each node in CCN network has the ability of data caching, and cached data can be reused. Contents of the cache becomes one of the core issue of CCN [2]. The research of CCN network cache technology mainly involves the allocation of cache capacity strategy, cache replacement strategy, cache utilization strategy and cache placement strategy [3-6]. At present, CCN is commonly used in the cache strategy along the way [7]. Since the nodes have data caching capabilities in the CCN, both the intermediate node and the content provider can respond to the content request, and the consumer who is interested in the same content can share the request and respond to all requests at the same time, so that the response time and the network traffic can be reduced.

Acharya first proposed the cache preloading technique. Preloading is based on speculation of the data that will be used by the users. For those who have high possibility to access, the cache node can download it in advance to its own cache for the convenient in the future; To theoretically analyze and compare the cache and preloading combined with only the performance of the cache,

the literature [8] uses stochastic Petri net to describe the combination of caching and pre-fetching, which proved that the combination of caching and pre-fetching is better than use caching technology only, can effectively reduce the average delay and improve server throughput. The pre-fetching technique is a supplement to the cache technology, which can also be called active caching technology, makes the utilization of limited network resources reasonably [9].

2 RELATED WORK

As an important application in the web, people have done more research on web caching and web preloading in web systems. Many people tend to visit the same hot information [10][11]. Based on the popular preloading algorithm, the server may have some pages that are more popular with users than other pages [12], and the hotspot based pre-load algorithm gathers users' requests through proxy server. When the user access is relatively concentrated, the web access hot, obviously by preloading pages have higher hit ratio, the effectiveness of the algorithm is better. However, the algorithm does not consider the user's specific access process, and cannot predict the next request in one visit based on the previous requests of the user, so the continuous access of the user cannot get a better response.

Preloading think based on link, the user's next request is typically from the current access page, so you can shorten the response time to the user's subsequent request by preloading some of the links on the current page [13]. However, there are more links on the web page, and each link on the preloaded web page leads to lower preload rate. However, there are more links on the web page, and each link on the preloaded web page leads to lower preload rate. Based on the preload of the link, the user's next request generally comes from the link of the current access page. Therefore, it is possible to shorten the response time rate of the user's subsequent requests by preloading some links on the current page [13], so as to shorten the response time of subsequent requests to the user utilization, but significantly increased the network traffic, literature [14] joined some historical information to help choose the preloading high possibility be user access link. This algorithm is easy to implement, and it treats every link on the page equally, but the preload utilization is low.

The method of pre-loading based on data mining is to exploit potential rules from the access path of a large number of users, based on these rules to predict the Web pages that users will visit, and preload it on this prediction. Some researchers have explored the rules of preloading based on user's access behavior, but the current study does not reach a consistent conclusion. Some studies suggest that sequential data mining is more suitable for Web preloading [15]. In other studies, sequential data mining is more suitable for Web preloading [16][17], and sequential data mining is more suitable for personalized recommendation.

Cache theory and related optimization techniques have been studied in the CDN, Web and P2P cache systems. The study shows that the caching and preloading of data with high access probability can effectively reduce the request delay time [11,18-26]. However, the cache system of CDN, Web and P2P is usually located in the application layer and in the form of a dedicated closed system, which is difficult to realize the sharing of cache objects between different applications due to the inconsistency of the sealing of communication protocol and the method of cache object naming.

Unlike traditional networks, CCN is a content oriented network structure that places content at the core of the content. It is based on unified content identification for content routing and caching decisions, which can effectively implement the sharing of cache resources between different application types in the network. The cache presents new features of closed caching systems such as

Web, P2P and CDN, which are different from traditional ones. In addition, new challenges are proposed for traditional caching theory, model and optimization methods. The new features of the cache in CCN network are mainly reflected in the transparency of the cache, the caching of the ubiquitous and the cache, and the granularity of the cache.

Cache is one of the core issues of CCN network, and if it can be effectively combined with the cache, it can improve the cache hit ratio. Preloading to research the main problem is how to choose the store to the local data, the content of the preloading is usually determined by the user's behavior. So the first thing to statistical user's access behavior, find the user access patterns, based on the forecast preloading model to guide the preloading of the next step work.

3 Cache Preloading Algorithm Based on the Data correlation

In the CCN network, there may be some logical relationship between the data fragments, for example, the logical relationship between the data slices that make up a video will be closer. When the user requests the data of the previous part of the video after the fragmentation, the subsequent data fragments are requested, and if the subsequent data can be preloaded in advance of the cache closer to the user according to the logical relationship of the data fragmentation, the delay of the data request can be reduced Time to improve the user experience.

3.1 Definition and calculation of the relevance

In order to measure the possibility that some data slices are requested by other data fragments after being requested by the same node, we use the correlation to represent, as defined below:

Definition (relevancy) : for a set of data shards $\{D|d_1, d_2, d_3, \dots, d_k, \dots, k=1, 2, 3, \dots\}$, when the data slice d_i is requested by a node, the data sub-disc d_j, \dots, d_{j+m} is the possibility of successive requests by the same node, we call this possibility the degree of correlation between the data fragments, The possibility that d_j, \dots, d_{j+m} has been requested by the same node again, we call this probability of the correlation between the data shards, with using $P(d_i, d_j, \dots, d_{j+m})$ said that this correlation can be measured by the conditional probability.

Thus, if the correlation between the data slice d_i and the data slices d_j, \dots, d_{j+m} is high (a threshold value T can be set, when $P(d_i, d_j, \dots, d_{j+m}) > T$, The data request d_j, \dots, d_{j+m} is cached in advance if the caching node caches the data slice d_i , if the data requesting node requests the data fragments d_j, \dots, d_{j+m} , The data fragment is provided near by the cache node without having to request it in the data source node, thereby reducing the delay time of the request.

In this paper, the conditional probability is used to represent the correlation between the data slices, According to the conditional probability:

$$P(D) = P(d_1, d_2, d_3, \dots, d_n) = P(d_1) \cdot P(d_2|d_1) \cdot P(d_3|d_1, d_2) \cdot P(d_4|d_1, d_2, d_3) \cdot \dots \cdot P(d_n|d_1, d_2, d_3, \dots, d_{n-1}) \quad (1)$$

Where $P(d_1)$ is the probability of occurrence of d_1 data slices, $P(d_2|d_1)$ is the probability of occurrence of data slice d_2 in the case where known data slice d_1 appears. In this paper, we approximate the behavior of the data slices according to the user's access behavior. Therefore, we need to record the user's access behavior and calculate the correlation between the data fragments according to the user's access behavior.

In order to calculate the above-mentioned conditional probability, it is necessary to establish a statistical model based on the user's request behavior for the data fragments $\{D|d_1, d_2, d_3, \dots, d_k, \dots, k=1, 2, 3, \dots\}$, but according to Formula (1) to establish the statistical model and the calculation of the conditions are more complex probability. In order to reduce the computational complexity, this paper assumes that the probability of data fragmentation d_i is only related to the previous data slice d_{i-1} , so that, according to the Markov hypothesis:

$$P(D) = P(d_1, d_2, d_3, \dots, d_n) = P(d_1) \cdot P(d_2|d_1) \cdot P(d_3|d_2) \cdot P(d_4|d_3) \cdot \dots \cdot P(d_n|d_{n-1}) \quad (2)$$

$P(d_1)$: Suppose that the number of requests for the data fragment set $\{D|d_1, d_2, d_3 \dots d_k \dots, k=1, 2, 3, \dots\}$ is always N times, d_1 the number of requests for the $f(d_1)$ times, then:

$$P(d_1) = f(d_1)/N \quad (3)$$

$P(d_i|d_j)$: If there are a large number of data fragmentation request records, the number of times $f(d_i)$ is counted separately, and the number of times d_i and d_j are requested, respectively, the number of times $f(d_i, d_j)$:

$$P(d_i|d_j) = f(d_i) / f(d_i, d_j) \quad (4)$$

The data fragmentation d_i may be related to the preceding two, three, or even more data slices. To make the degree of relevance more realistic to reflect the degree of closeness between the data slices, it can be assumed that the probability of data fragmentation d_i With the first two data fragments d_{i-1} and d_{i-2} , so that according to the Markov hypothesis there are:

$$P(D) = P(d_1, d_2, d_3, \dots, d_n) = P(d_1) \cdot P(d_2|d_1) \cdot P(d_3|d_1, d_2) \cdot \dots \cdot P(d_n|d_{n-2}, d_{n-1}) \quad (5)$$

Formula (2) corresponding to the user access behavior statistical model known as the binary model, formula (5) corresponding to the user access behavior statistical model called the ternary model. In the above calculation, the collection of user access behavior is the basis, so when the number of users access is huge, according to the ternary model to collect user access behavior than in accordance with the binary model to pay a greater cost.

3.2 Node type in the network

In order to facilitate the description of cache and preload algorithm, in this paper, if there are the following network topology:

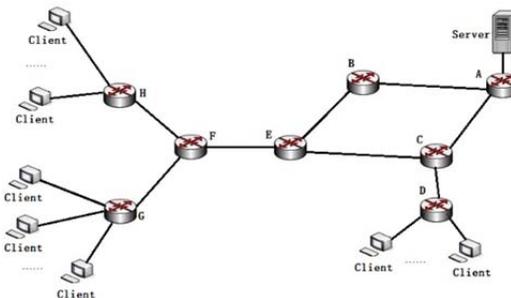


Figure 1 Network topology

In this paper, according to the role of the network nodes are divided into the following types:

3.2.1 Intermediate node

No client directly connected to the network node, as shown in Figure 1 F, E, B and so on. The main function of the intermediate node is to forward the packets in the CNN network and cache the data according to the cache and preload algorithm.

3.2.2 Network edge node

In this paper, there is a network node directly connected with the client, such as D, H, and G in figure 1. There are two main functions of network edge nodes: first, the statistics and the access behavior of directly connected clients, and the second is the function of forwarding packet message and caching. The data that is not cached by the intermediate node will be cached on the network edge node, and the associated data is preloaded according to the preload algorithm; In order to calculate the client's access behavior, the user behavior statistics table (according to the binary model statistics) is required in the network edge node.

Table 1 User Behavior Statistics

Previous Content Name	Next Content Name	Statistic times
/amazon.com/videos/mpg / _v/s0	/amazon.com/videos/mpg / _v/s1	5
.....	

The previous content name in the table and next content name are the content name information of the two Interest packets sent by the same client successively, and the statistic times is the number of times that the two data slices are successively requested. For the convenience of the statistics, on the edge of the network node, the client and for each port is directly connected to create a queue, port for each Interest received packet, put the Interest of packet Content Name into the corresponding queue, each queue can store two (dual model according to the statistics) or three (by three yuan statistics) model of the port has recently received Interest packet, the Content of a message name information, when there is new content name into the queue, according to statistics the information in the queue to find if there is a match record, the record in the statistics of Statistic times add 1, or add a record in the statistical tables and the Statistic times initialized to 1.

The network edge node periodically sends the statistical result to the data source node, which provides the basis for the data source node to calculate the correlation between the data fragments. The network edge node empties the user behavior statistics table every time the statistical result is sent to the data source node.

3.2.3 Data source node

The data source node is responsible for sending the data packet to the requester. At the same time, the user behavior statistics table is established in the data source node. When the data packet is received by the network edge node, the data node is sent to the data source node. When the statistical information is sent, the statistical information of the edge node of the network is merged with its own user behavior statistics table, and the correlation between the data fragments in the data source node is calculated according to equation (2).

3.3 Packet format

In order to distinguish different types of packets in the network, it is convenient to send the user access behavior statistics in the network, and to facilitate the cache node according to the degree of data between the pre-load, this article adds a statistical update message (Statistical Update Packet (SUP)), and the CCN in the original interest packet and data packets was modified.

3.3.1 Interest packet and data packet

Type
Content Name
Selector (order preference,publisher filter,scope,...)
Nonce
Count

Figure 2 Interest Packet

Type
Content Name
Signature (digest algorithm,witness,...)
Signed Info (publisher ID,key locator,stale time,...)
Data
Cache flag
$D=\{d_k,\dots,d_n\}$
$P(d_k,\dots,d_n)$

Figure 3 Data Packet

Interest Packet Interest message is mainly added Type field, used to distinguish the Packet type in the network, other fields and the original CCN in the same Interest Packet, the working mechanism is basically the same.

In the data packet, the type field is used to distinguish the message type in the network. The cache flag field is used to identify whether the data has been buffered. The initial value is 0, which is changed to 1 when the Data Packet is buffered. $P(d_k, \dots, d_n)$ field is the degree of correlation of the data fragment set $D = \{d_k, \dots, d_n\}$, and the D field is a data slice with a high correlation (correlation greater than the threshold) of the requested data slice and the latter two items are used to reply the interest message to the data source. At the same time, the data fragment name and the correlation degree with the highest degree of response data fragmentation are also issued, which provides the basis for the cache node to preload the data. The processing mechanism of the data packet is roughly the same as that of the original CCN. The difference is that when the node decides to cache the received data fragments, the cache node needs to set the cache flag cache flag in the data packet, The node needs to judge whether it needs to preload other data slices according to $D = \{d_k, \dots, d_n\}$ and $P(d_k, \dots, d_n)$ in the data packet. If the preload is required, the caching node according to the data provided in $D = \{d_k, \dots, d_n\}$ Information preloads the relevant data fragments.

3.3.2 Statistical update packet

In order to facilitate the transmission of user access behavior between the network edge node and the data source node, this paper designs the statistical update packet message as shown in Fig. 4, This type of message is sent by the network edge node to the data source node. The data source node is responsible for merging the received statistics with the original statistics and recalculating the correlation between the data fragments in the data source.

Type
Data source
Nonce
Time Stamp
count data

Figure 4 Statistical Update Packet(SUP)

Among them, type is the packet message type, Nonce is the random number, and Time Stamp is the time stamp, used to record the statistical update message (SUP) sending time, Data source for the data source identification, count data for the statistics.

3.4 Cooperative cache preloading algorithm based on data relevance

3.4.1 Data cache

In order to minimize the cache data between the cache nodes, save the cache space and let the cache data as far as possible from the user node, when there are multiple ports through a network node forwarding the same data fragment of the Interest packet, then the network nodes cache the requested data fragmentation then the network the node caches the requested data fragment. For example, as shown in Fig. 1, when H and G forward the Interest packet for the same data fragment by F, the node F buffers the requested data slice, and E does not cache the data. In order to simplify the design, save the cache space, in this paper simply based on the cache node PIT to determine whether there are multiple ports to receive the same data fragment of the Interest packet.

3.4.2 Cache preloading algorithm based on the data correlation

(1) The processing of Interest Packet message

The cache node handles the Interest packet: When the intermediate node receives the Interest packet, it processes the Interest packet according to the normal CCN process. If it is directly connected to the network edge node received the Interest of the client sends the Packet, in accordance with normal CCN process to deal with the Interest of the packet at the same time, the

Interest in the packet content with port name information into the corresponding queue, and according to the statistics information update user behavior in the queue.

When the data source node receives the Interest packet, it determines whether there is any other data fragment in the data source and the correlation between the data fragment and the requested data fragment is greater than a certain threshold. If the data source node receives the Interest packet, yes, the name and relevance of these data fragments are responded to the requester with the requested data along the direction of the Interest Packet message.

(2) Processing of the Statistical Update Packet (SUP)

If the intermediate node receives a statistical update packet (SUP), simply forward the packet. If the data source node receives a Statistical Update Packet (SUP), the statistics in the packet are merged with the statistics in the data source node and the data between the data fragments in the data source is recalculated relativity.

(3) The processing of Data Packet

If the intermediate node receives the Data Packet, first of all determine whether the cache algorithm need to cache the Data to the paper if you need the Data cache in the CS, and according to the packet of the relevance of $P(d_k, \dots, d_n)$ determine whether need to $D = \{d_k, \dots, d_n\}$ is preloaded, and if the relevant Data is pre-loaded to the cache, the cache flag in the data packet is set, otherwise the middle node will forward the data packet directly by PIT.

If is the network edge node receives the data packet, and the data packet is directly connected to its client, is on the edge of the network nodes forward data packet to the client at the same time, whether the newspaper article data fragmentation has been other intermediate nodes cache, if not, then the cache data in a Packet fragmentation, and according to the message of the relevance of $P(d_k, \dots, d_n)$ determine whether need to $D = \{d_k, \dots, d_n\}$ is preloaded, and the relevant data is pre-loaded to the cache of the network edge nodes.

Algorithm 1: Data source node

Input: Packet

For each Packet do

 If Packet type is Interest Packet then

 If $P(\text{Interest Packet. Content Name}, d_k, \dots, d_n) > T$ then

 Initialize the D field and the $P(d_k, \dots, d_n)$ field of the data packet with the relevance and the name of the data fragment;

 Send a response to the data packet;

 End if

 Else

 If Packet type is Statistical Update Packet then

 Update the user behavior statistics according to the Statistical Update Packet;

 Calculate the correlation between data fragments according to formula (2)

 End if

 End if

End for

Algorithm 2 cache node based on data correlation cache preloading algorithm

For each Packet do

 If Packet type is Interest Packet then

 If is the first hop of the Interest Packet then

```

    The Content Name information in the Interest packet is sent to the queue corresponding
to the port;
    Updating the user behavior statistics according to the information in the queue;
    End if
    According to the CCN normal processing process for the Interest Packet message
processing;
    End if
    Else
    If Packet type is Statistical Update Packet then
        According to the node in the FIB directly forward Statistics Update Packet, do not need
to record any information in the PIT;
    Else
    If Packet type is Data Packet message then
        If Data Packet is the last hop, Data Packet Cache flag = 0 then
            Cache the Data Packet at the last hop of the network edge node;
            If  $P(d_k, \dots, d_n) > T$  then
                Preloaded data fragments  $\{d_k, \dots, d_n\}$ ;
            End if
        End if
    End if
    Else
        According to the cache strategy to determine whether the node needs to cache the data
fragment;
        If you need to cache data
            Caching data fragments in the CS of the node;
            Change the Data Packet's Cache flag to 1;
            If  $P(d_k, \dots, d_n) > T$  then
                Preloaded data fragments  $\{d_k, \dots, d_n\}$ ;
            End if
            Data Packet according to node PIT;
        End if
    End if
    End if
    End if
    End for

```

4 Simulation and performance analysis

This section uses the NS3-based ndnSIM[27] simulation platform to simulate the cache and preloading algorithm proposed in this paper. The network topology is a random network topology composed of 40 routing nodes generated by GT-ITM. There are 1000 content objects in the data source, the content size is 20KB, the cache node CS is initially empty, the cache size is 10MB, the client requests to reach the poisson distribution, $\lambda = 20 / \text{sec}$; the user's access mode follows the Zipf distribution, the cache replacement policy is LRU.

Simulation experiments, under the same environment, the network of the cache using the algorithm of preloading and without using the algorithm of preloading, respectively in preloading method, according to the client access behavior of correlation between results calculated data

fragmentation, and according to the correlation between data fragmentation selective data preloading. In this paper, the average request delay time was used as an indicator to evaluate its performance, and the average delay time of the client request was calculated every 5 seconds. The experimental results are shown in fig.5:

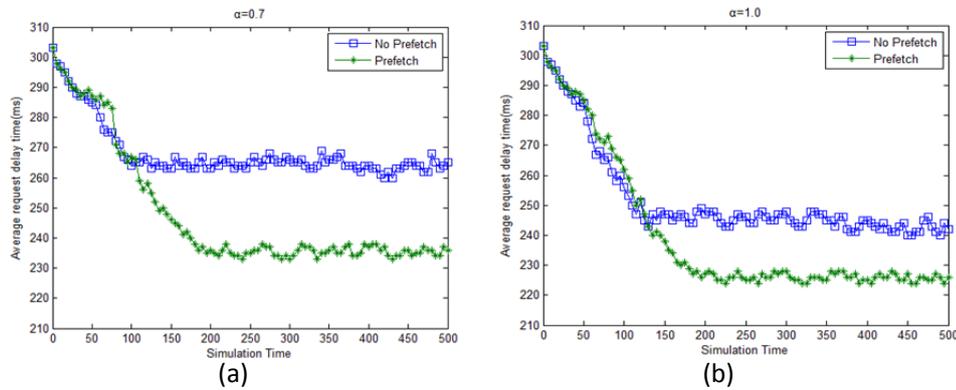


Figure 5 Average request delay time trend comparison

As can be seen from the simulation results, there is almost no difference between the preloaded cache and the no preloaded cache on the requested delay, since CS at the beginning of the cache node is empty and the client the number of requests for data is still less, according to the client's access behavior calculated data correlation can not better reflect the relationship between data fragments, so the cache node CS full before the preloaded cache and no The preloaded cache is not very different on the requested delay. When the cache node CS is filled for a period of time, there is a preloaded cache, the client requests the delay time even less than no preload time is longer, this is because at this time, the correlation can not be a good expression of the relationship between the data fragments, according to the relevance of the data at this time is not pre-installed data is required by the client, And in order to preload the data need to replace the original part of the data in the cache, so that in the case of preloaded cache, the client requests the delay time is longer than no preload. The number of client requests is increasing, according to the client access behavior calculated data relevance more objective expression of the relationship between the data fragments, according to the relevance of the data at this time is likely to be preloaded by the client when the required data, then there is preloaded cache delay than the preload cache delay is shorter.

According to the Zipf distribution formula, the larger the α , the client's request will be more hot data, and these high-calorie data is usually cached by the cache node, then no preload cache hit rate is usually higher, at this point the preload cache on the request to improve the delay time is not very obvious. when α is small, the preload cache to improve the request delay time is more obvious, this is because when α is small, the data request concentration tends to smaller, the data that the possibility of being cached is also smaller and the hit rate of the preloaded cache is reduced. If the data can be preloaded according to the relationship between the data slices, it will help to improve the cache hit rate and reduce the client request of the delay time. But when α is too small, the effect is not obvious, because when α is too small, the client request will be very scattered, then get the data between the fragments will be very low. In the cache system, it will not these low-correlation data are preloaded, even if preloaded, these preloaded data hit rate is not high. FIG. 5(a) and fig.5 (b) respectively give the average delay time of data requests in preloaded and without preloading conditions when $\alpha=0.7$ and $\alpha=1.0$, from the point of the simulation results, when $\alpha=0.7$, under the condition of using the algorithm of preloading, data request the average delay time was reduced by 7.57%, when the $\alpha=1.0$, under the condition of using the algorithm of preloading, data request the

average delay time was reduced by 6.25%.

5 Conclusion

In this paper, the cache preloading strategy in the CCN network is analyzed. According to the user's access behavior, the correlation between the data slices is described by the conditional probability, and a cache preloading algorithm based on relevance is given. Then, the simulation experiment is carried out. The experimental results show that the cache nodes in the network can reduce the delay time by filter and preloading the data of the cache by using the cache preloading algorithm described in this paper under certain conditions, thus improving the network performance. The key of this algorithm is the collection of user access behavior according to the user's access behavior to calculate the correlation between the data fragments. But when the data segmentation and the number of users access is huge, this direct collection method will become more difficult to calculate the correlation between the data fragments because it becomes time-consuming. How to establish a better statistical model to solve these problems will be the next step in the study.

Acknowledgement

This work is supported in part by the program for Innovative Research Team in University of Yunnan Province and the National Science Foundation of China under Grant No.60963021.

References

- [1] Jacobson V, Smetters D K, Thornton J D, et al. Networking Named Content [C]//Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies. New York:ACM,2009:1-12.
- [2] Ahlgren B,Dannewitz C,Imbrenda C,et al. A Survey of Information-centric Networking [J]. IEEE Communications Magazine,2012,50(7):26-36.
- [3] Psaras I,Chai W K,Pavlou G.Probabilistic In-network Caching for Information-centric Networks[C]//Proceedings of the Second Edition of the ICN Workshop on Information- centric Networking.New York:ACM,2012:55-60.
- [4] Cho K, Lee M, Park K, et al. WAVE: Popularity-based and Collaborative In-network Caching for Content-oriented Networks [C]//Conference on Computer Communications Workshops. Piscataway:IEEE,2012:316-321.
- [5] Chai W K, He D, Psaras I, et al. Cache “Less for More” in Information-centric Networks (Extended Version) [J]. Computer Communications, 2013, 36(7): 758-770.
- [6] Zhang G, Li Y, Lin T. Caching in Information Centric Network: a Survey [J]. Computer Networks, 2013, 57(16): 3128-3141.
- [7] Xylomenos G, Ververidis C, Siris V, et al. A Survey of information-centric networking research [J]. IEEE Communications Surveys & Tutorials, 2014, 16(2): 1024-1049
- [8] Shi L, Han Y, Ding X, et al. An SPN based integrated model for Web prefetching and caching [J]. Journal of Computer Science and Technology, 2006, 21(4): 482-489
- [9] Fan L, Cao P, Lin W, et al. Web prefetching between low-bandwidth clients and proxy-es: Potential and performance [C]. PPPProc of the ACM SIGMETRICS Int Conf on Measurement and Modeling of Computer Systems. New York:ACM,1999:178-187.
- [10] Lu, J., Ruan D. and Zhang G. E-Service Intelligence: An Introduction. Studies in Computational Intelligence (SCI) 37, Heidelberg, Berlin: Springer-Verlag, 2007, 1-33.

- [11] Davison, B. D. The Design and Evaluation of Web Prefetching and Caching Techniques. Doctor of Philosophy thesis, Graduate School of New Brunswick Rutgers, The State University of New Jersey, United State, 2002.
- [12] Marcatos E P, Chronaki C E. A Top-10 approach to prefetching the web [EB/OL]. Proceedings of the Eighth Annual Conference of the Internet Society. Geneva: INET, 1998 (1998-01-30) [2006-02-07]. http://www.isoc.org/inet98/proceedings/li/li_2.htm.
- [13] Chinen K, Yamaguchi S. An interactive prefetching proxy server for improvement of WWW latency [C] // Proceedings of the Seventh Annual Conference of Internet Society. Kuala Lumpur: INET, 1997: 473-478.
- [14] Davison B D. Predicting web actions from html content [C] // Proceedings of the Thirtieth ACM Conference on Hypertext and Hypermedia. Maryland: ACM, 2002: 159-168.
- [15] Nanopoulos A, Katsaros D, Manolopoulos Y. A data mining algorithm for generalized web prefetching [J]. IEEE Trans on Knowledge and Data Engineering, 2003, 15(5): 1155-1169.
- [16] Yang Q, Li T, Wang K. Building association-rule based sequential classifiers for Web document prediction [J]. Data Mining and Knowledge Discovery, 2004, 8(3): 253-273.
- [17] Mobasher B, Dai H, Luo T, et al. Using sequential and non-sequential patterns in predictive Web usage mining tasks [C] // Proc of the IEEE Int Conf on Data Mining. Los Alamitos, CA: IEEE Computer Society, 2002: 669-672
- [18] Nagaraj, S. V. Web Caching and Its Applications. Boston/Dordrecht/London: Kluwer Academic Publishers, 2004.
- [19] Acharjee, U. Personalized and Intelligence Web Caching and Prefetching. Master thesis, Faculty of Graduate and Postdoctoral Studies, University of Ottawa, Canada, 2006.
- [20] Garg, A. Reduction of Latency in the Web Using Prefetching and Caching. Doctor of Philosophy thesis, University of California, Los Angeles, United State, 2003.
- [21] Curran, K. and Duffy, C. Understanding and Reducing Web Delays. Int. J. Network Mgmt. 2005, 15, 89-102.
- [22] Saiedian, M. and Naeem, M. Understanding and Reducing Web Delays. IEEE Computer Journal. 2001, 34(12).
- [23] Kazi, T.H., Feng, W. and Hu, G. Web Object Prefetching: Approaches and a New Algorithm. 2010 11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD). 2010, 115-120.
- [24] Nigam, B. and Jain, S. Analysis of Markov model on different web Prefetching and caching schemes. 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICIC). Coimbatore, India: IEEE, 2010, 1-6.
- [25] Wang, Y. A Hybrid Markov Prediction Model for Web Prefetching. Master thesis, Department of Electrical and Computer Engineering, University of Calgary, Alberta, 2003.
- [26] Ahmad, N., Malik, O, ul Hassan, M, Qureshi, M. S. and Munir, A. Reducing user latency in web prefetching using integrated techniques. 2011 International Conference on Computer Networks and Information Technology (ICCNIT). Abbottabad, Pakistan: IEEE, 2011, 175-178.
- [27] Afanasyev A, Moiseenko I, Zhang L X. ndnSIM: NDN simulator for NS-3: NDN-0005 [R]. NDN, 2012.