Self-cleaning Breadcrumb Policy in Cache Network

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering

July 27th, 2017

By

Wuhong Niu

(5115FG18-4)

of

Nakazato Labtory

(Professor Hidenrori Nakazato)

Abstract

In-networking caching is a trend of future network, A fundamental characteristic of such systems is that local cache capacity at each node. One of the most promising systems is Content-centric Network (CCN). Considering such a system, some researchers provide a new routing policy called Breadcrumb to optimize routing efficiency based on caching history.

This paper proposes a solution to traditional Breadcrumb policy to improve the routing efficiency for in-network cache. Our evaluation shows that this modified Breadcrumb policy brings reduction in management overhead and an improvement in breadcrumb utilization comparing with the previous one.

Index Terms – In-networking cache, Content-centric Network (CCN), Breadcrumb, Routing policy

CONTENT

List of Figure

List of Table

Abbreviation

CCN	Content-Centric Networking
NDN	Named Data Networking
BC	Breadcrumbs
QoS	Quality of Service
DONA	Network Data Centric Architecture
NetInf	Network of Information
PSIRP	Publish Subscribe Internet Routing Paradigm
PURSUIT	Publish Subscribe Internet Technology
DNS	Domain Name System
RH	Resolution Handler
LCE	Leave Copy Everywhere
LCD	Leave Copy Down
SPR	Shortest Path Routing
LRU	Least Recently Used
DFQ	Download Follows Query
DFSP	Download Follows Shortest Path
CS	Content Store

Symbols

c	node
q	query/request
f	file/content/cache
T/t	time
q_f	query for content f
T_f	the time that content f passed
T_{qf}	the time that query for content f passed

Published Work

[1] Niu Wuhong, Nakazato Hidenrori, Wu Yunbin, “Self-cleaning Breadcrumb Policy in Cache Network”, IEICE, CS2017, September 2017.

1. Introduction

For several years, web caching has been used to meet the ever-increasing Web access loads [1].

Content storage (caching) systems come in many forms and flavors, one of the most promising network architectures is Content Centric Networking.

CCN is a network that is based on content. Communication focuses on contents but not hosts. It identifies each content by content name. This new architecture emphasizes content itself and redefines communication architecture with fully considering scalability and security. Communication is no more “host- to-host” but “request-content” which means that contents are decoupled with locations and thus better for mobility.

Caching in network layer but not traditional application layer cache has following differences:

- Caching is common in the network while each cache is small. Cache is embedded in each router. When the network traffic is high-cost but high-speed cache can help to reduce workload.
- Cache and network traffic interact with each other. Cache can reduce network traffic while forwarding traffic also affects the cache in a router.
- In-network cache provides unified network-layer content cache platform for different types of data.

In-networking caching capability is one basic characteristic of CCN, many works for improving such cache utilization have been proposed [4], [5], [6], [7], [8]. These studies can be classified into two kinds of approaches: cache policy and cache replacement. Cache policy is a decision made by a router whether it stores incoming content packets. When an incoming content packet is decided to be stored but the router’s local cache is fully utilized, the router should decide which cached content should be expired, which is called cache replacement policy.

And also, many studies have been published for routing in CCN, a novel routing policy is Breadcrumbs. Breadcrumbs [9] have been proposed to probe off- path caches with low network load. When a user downloads a content packet from the original content server for the first time, each intermediate router sets a pointer indicating the download direction called breadcrumbs (BCs). The chain of these BCs is called bc-trail.

Different from download history, BCs direct to the downstream of the download path, which is a reverse direction to server. See a behavior of the Breadcrumbs method in Figure 1. When a request occasionally encounters a BC stored at a router during the process it is forwarded towards sever, its forwarding will be switched from routing table–based forwarding to breadcrumb table-based forwarding. It will be sent hop-by-hop conceptually away from sever along a breadcrumb trail to a possible copy of the objective content. It is hoped that BC can improve off-path content copies' utilization. Breadcrumbs is originally proposed for general content-centric networks, but can be easily applied to CCN [10].

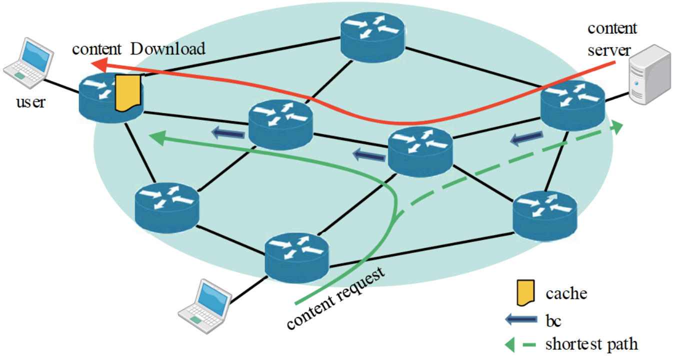

Figure 1. The structure of BC

However, multiple content downloads will lead to many obsolete BC trails in the network even though previous BC policy uses expired time to control the size of BC tables.

To counter this problem, we propose here a modified BC policy of deleting the BC once it has been used. We expect that this new solution can help to release BC table spaces and improve the hit performance of BC.

2. Research Background

2.1 IP network and what it faces

The Internet is one of the most important inventions in human history. The basic Internet principles and structure was built in 1960s-1970s in a US military network research project. After years it became the world's largest communication network for the people around the world, providing all kinds of data, applications and services. The Internet develops from small to large, from fixed network to mobile network, carrying new technologies and new applications, profoundly affecting and changing the way people work, learn and live.

Traditional Internet architecture supports the best-effort service, following the "simple core but intelligent edges" principles. Intelligent devices are deployed at edge of the network [11]. This architecture is simple, but can ensure effective connectivity and scalablity, so it is still in serve today. With the rapid development of Internet technology, Internet applications demand more and more about network performance.

First of all, with the rapid development of Internet technology and applications and the rapid growth in the number of Internet users, Internet is moving towards higher broadband, more contents and more personalized settings. The demands for contents in the network achieve a higher magnitude. It is expected that more than 90% mobile data transmissions in the world will be content requests in the future. Among them, video is expected to account to 64% of the total mobile traffic.

Secondly, with the development of network communication technology, computer technology and online business, mobile Internet, Internet of things, cloud computing, data centers and other new applications keep updating day by day. The progresses of these new technologies make existing Internet technologies have to be updated rapidly, even including custom functions. This presents a huge challenge to the scalability, mobility, security, energy consumption, quality of service and reliability of a network.

Moreover, the exponential growth of the traditional Internet also makes it hard to control or manage. It difficult to meet the needs of a wide range of applications.

Finally, the current Internet only provides Best-Effort services, that is, trying to connect different networks without guaranteeing the quality of services. The reason is that in the traditional network architecture, core network only retains limited management information but does not provide QoS guarantee. On the other hand however, the future network needs to carry a variety of new business in addition to information transmission, file sharing, service access and other technical services. In addition, a wide variety of new applications and the introduction of a variety of new technologies, as well as the dynamic network environment will greatly increase the complexity of network management and control.

Above all, the development of Internet needs far more than the Internet initially does, "Best Effort" of the transmission has been unable to meet the needs of the future network. Future network does not only have data transmission capacity, but also capacity to calculate and store data. Therefore, in the future evolution of the Internet, a variety of network systems and technologies will coexist for a long time, resulting in a large number of old and new network protocols, services and interfaces will be used in a long period. How to transfer, store and calculate in a compatible way will be a key topic for the future of network technology.

2.2 Future network and researches on it

From the technical part, first of all, computing and storing will be added as new property of network in the future; moreover, the network framework whose focal point is “named content” becomes an important developing direction of the future network. In the development trend of this kind of future network, researchers have proposed to establish the network architecture which is content-centric. The purpose of this research is to transform network architecture whose focal point is ‘where’, to network architecture which is ‘what’ centric. And ‘what’ is according to users and applications. Under the guide of this kind of idea, many researches about the content-centric network architecture have been set up, such as Network Data Centric Architecture (DONA), Named Data Networking (NDN), Content-Centric Networking

(CCN), Network of Information (NetInf), Publish Subscribe Internet Routing Paradigm (PSIRP), Publish Subscribe Internet Technology (PURSUIT) and so on. Although every research has focused on different part of the designing process of the whole architecture, all these researches are concentrating their attentions on content itself. At the same time, the nodes in the network will not only provide transferring any more. The nodes will become more intelligent and support the network. These CCN researches have identical functional module, such as naming and security of content objects, name resolution and routing, storage and caching in network. However, modules in different researches should be achieved in different ways. These researches will be introduced as below. The researches could be appropriate for different protocol stack layer. For example, data access based on name could be realized upon the existing IP architecture such as providing source name, ubiquitous caching and some corresponding transferring service. On the other hand, the researches could be regarded as packet level network technology which could change the basic framework of routing and transmitting in the network. Based on the flexibility of ICN network, many researches have been studied by more and more experts.

A. DONA

DONA have been launched in 2006 by RAD LAB and ICSI from UC Berkeley. The main researchers are Scott Shenker and Ion Stoica. The most important contribution was to propose the basic idea of DONA, and laid a foundation of designing every kinds of content-centric network architecture. The novel proposal of their researches is to change the Internet Naming structure and resolution method and come up with content-centric naming instead of Domain Name System (DNS) naming. content source in DONA publish content objects and register it to the network, and only these content could ask for achieving routing. Resolution Handler (RH) applied hierarchical structure. The structure receives request according to named routing and tries to search the nearest content from users. After that, in the process of transferring data, content objects will send back to the terminal user along the route of request.

B. CCN/NDN

Future Internet Architecture (FLA) designs safe and reliable future network architecture to solve the problem of current network which is lacking security and reliability of content transferring. The major technical challenge of CCN/NDN researches is to prove that CCN/NDN could be a reliable future architecture: routing scalability, trust model, network security, content protection and privacy, basic communication theory. The project is deployed with an end to end testing, simulated on the computer and analyzed theoretically to evaluate the proposed architecture. Moreover, the project develops standard model and achieve protocols and applications of CCN/NDN.

C. NetInf

NetInf established a data bag model which only has information of content objects and deletes the location information from the model. This project radically changes the ways in which IP data bags carried valid content and sent data with receiver's address to separate the data processing and location information. NetInf designed Name Resolution Service (NRS) for resolving information model. In NRS, content objects need be published in advance in the network and NRS is used for registering content objects corresponding to the location in the network. When a user searches for a content object, NRS would resolve many locators and choose the best one as available source to achieve data object.

D. PURSUIT/PSIRP

This project aims to establish a publishing and subscribing model which is content-centric, making sure that content is asked by users positively. The advantage of this project is to avoid malicious network attack by rejecting pushing content information. The architecture uses the globally unique network identifier to identify content. And the identifier is a scalable flat self-certified identity. At the same time, PURSUIT/PSIRP designs flat identity for identifying the extent of information which is managed by Rendezvous Point (RP). In the higher layers of the architecture, apply discrete hashing transformation to carry out polymerization function, use topological function to monitoring network diagram and establish a route from a publisher to one or more subscribers. In PURSUIT/PSIRP, the content objects need to be published in the network in advance. The published and subscribed content should be matched via RP, and resolved as transmitted ID and routing for data objects in the network.

Although every projects are focused on different process of designing architecture, these projects all focus on content without storing location, physical description. And make the nodes more intelligent instead of providing simple transferring function.

3. Content-centric Network and Technologies in CCN

3.1 Content-centric Network

The purpose of the Content-centric Network (CCN) is to develop a new Internet architecture that naturally adapts to the current communications characteristics. Instead of naming the address of data, CCN directly names data itself, the named data has become the most critical network entity. In addition, CCN is no longer concerned about the security of edge devices, but directly concerned about the security of the named content, so that the data transmission mechanism is fundamentally free to expand. The main technical challenge of CCN research is how to establish a new architecture in all directions so that it can support future network needs and can be run for a long time. It is hoped that CCN can be practically rubust from aspects of basic theory, trust mechanism, routing scalability, security, privacy and others.

Content-centric Network is a network that is based on content, communication focuses on contents but not hosts. CCN architecture has three characteristics distinct from IP networking.

First, a CCN node forward files by content names, not by addresses. This means two changes:

- Identifying addresses is replaced by identifying contents.
- The location of a content file is independent from its name.

An IP address is both the identifier and locations; hence, IP networking has mobility problems. While CCN has location independence in content naming and routing, and thus is free from mobility and multihoming problems.

Second, the publish/subscribe architecture is the main communication model in CCN: a content server publish a content with a special ID, then users request for this content by searching this ID. In IP networking, a user should know which source holds the content file of interest (spatial coupling), and the two hosts should be associated throughout the delivery (temporal coupling) [12]. With the publish/subscribe architecture, we can decouple the content and location of the server, so contents are delivered efficiently and scalably.

Third, in IP networking, a host address seen by a user is irrelevant to its content name, which results in phishing and pharming attacks. For content authentication in CCN, either a self-certifying content name [13], [15] or a signature in a packet [14] is used.

CCN architecture can be characterized by four main key technology:

- Contents naming
- Name-based Routing
- Name-based Caching
- Security of the contents

Naming is the basic characteristics disgusting CCN from IP network. Content names in CCN take the place of IP addresses in IP network. Data security mechanism and authentication have direct expressions in naming. Routing and forwarding in CCN is the process to search data object through data name, this is also data name parsing process. The paths that are chosen during routing and forwarding processes can influence which nodes to cache contents, thus, influence routing and forwarding reversely. It can be seen that these four key technologies are complementary; they reflects CCN's advantages together comparing with traditional IP network.

In the following section, the four technologies will be introduced briefly.

3.1.1 Content Naming

Naming schemes are classified into three categories: hierarchical naming and flat naming.

A. Hierarchical Naming

The representative model is CCN, under such a naming standard, a content file is named by an identifier like a web URL (/parc.com/videos/WidgetA.mpg/_V<timestamp>/_s3). This naming strategy facilitates the aggregation of the names sharing same prefix, thus reducing routing table and improving the efficiency of routers. Since the length of the content name is variable, the networking structure hence has flexibility and scalability. Also a hierarchical name can be compatible with the current URL-based applications. The hierarchical nature makes the migration from current IP network architecture easier.

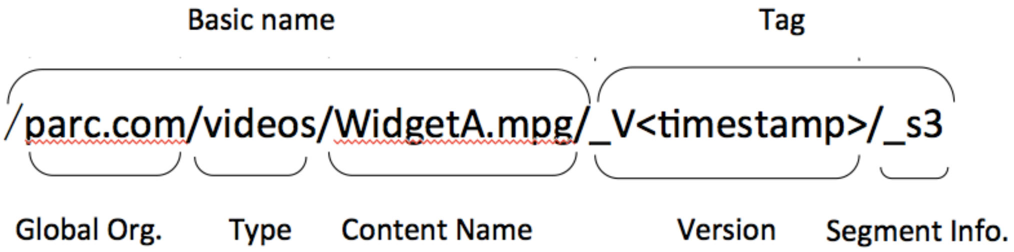

Figure 2. A hierarchical naming format

B. Flat Naming

Comparing flattening naming and hierarchical naming, flattening naming is better for security, but it is not conducive to aggregation; hierarchical naming is convenient for aggregation, but its aggregation is more difficult.

From the view of scalability, although the hierarchical naming is easy to aggregate, its aggregation will introduce a suffix problem. What’s more, hierarchical named routing table is also difficult to control. While flattening naming is not conducive to aggregation, it is better for scalability because of its address space is predictable.

3.1.2 Name-Based Routing

Routing tables are built based on the contents name in CCN, both of the forwarding of requests and contents. Since cache hit is caused by arrival of a content request at a router, content request routing is considered ad a fundamental factor of a networks' performance. Content request routing is called routing policy in CCN/NDN. Routing policies have been proposed and promote cache hit on off-paths, i.e. outside default-path. For example, Flooding [17] floods content request packets and aggressively retrieves caches from off-paths. INFORM [18] forwards requests to off-path caches with reinforcement learning based on content retrieval latency. CATT [19] calculates potential value for cache discovery based on distance between a user and a cache, and utilizes off-path caches.

3.1.3 In-Networking Caching

One of the essential differences between CCN and other network system is that routers in CCN have local caching capacity. Named content in the CCN can be automatically cached in routers along forwarding paths. Studies [20], [21], [22], [23] provide some researches in aspects of cache replacement strategy. The cache in CCN routers is similar to the buffer in IP routers, they are all used to cache data. The IP router, however, cannot re-use the cache after forwarding it out, the IP router can utilize the cached data again because the data can be matched by name-identifier in CCN architecture. For static data, CCN almost achieves never better content distribution; for dynamic data, CCN also has superior performance in the case of multicast and retransmission.

The advantages of in-network caching can be divided into two parts:

- To reduce the incoming traffic from neighbor nodes to lower the whole traffic load in the network.
- To improve the delay by caching the contents closer to their users.

In-network caching is also attractive to content providers since it can mitigate the capital load in the server nodes. Considering the characteristics of CCN, it is viable to cache popular content files and therefore improve the efficiency of routing. There are well-studied caching policies such as Least Recently Used (LRU) and Least Frequently Used (LFU) replacement policies at individual nodes, and also, by transplanting them into CCN, the performance of CCN can be further improved with coordinated CCN nodes.

For cache in CCN, lots of studies for improving cache utilization have been published [4], [5], [6], [7], [8]. These studies can be classified into two kind of approaches, cache policy and cache replacement. Cache policy is a decision a decision of whether it stores incoming content packets. When an incoming content packet is decided to be stored but the router's local cache is fully utilized, the router should decide which cached content should be evicted, which is called cache replacement policy. Routers in CCN manage local cache themselves based on these two cache management policies.

Common cache policies in CCN are introduce as following:

- LCE (Leave Copy Everywhere)

This content download policy is used in the original CCN [4]. A content Object packet is cached at all routers along a download path. It is the most popular cache decision policy and adopted by many studies [16] [18].

- Fix(p)

Fix(p) is a simple cache decision policy that each router on a download path probabilistically stores incoming content object with probability p. Because the rate of caching is decided according to the parameter p, relatively-popular contents tend to be stably cached in the network.

- LCD (Leave Copy Down)

In LCD, a content object is cached at the one-hop downward router from a cache hit router. Figure 3 shows an example of LCD behavior. After the content downloaded from the server, it is cached in the router that is one-hop downward of the server. Then, the cache is copied into the next downward router when another download occurs. The same procedure is performed thereafter. Since this behavior delivers popular contents to one-hop closer to each generated Interest packet, this method is classified to popularity-based policy.

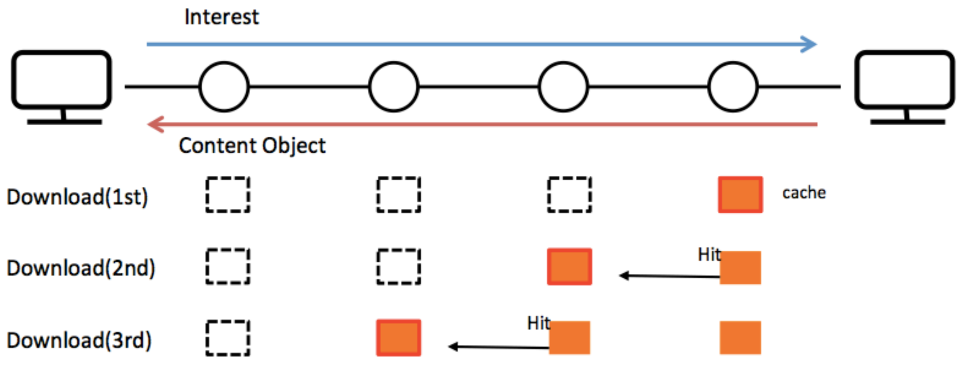

Figure 3. An example of LCD

3.1.4 Security of Content

CCN security is built on content naming itself, no matter with where or how the content is obtained. Each piece of data is bind with a mandatory signature to ensure safety. Data publishers' private signatures determine the authority of data, so that consumers do not have to pay attention to how or where data are required. It also supports more meticulous safe certification, and with specific contexts and specific data, consumers can determine whether the owner of the data is an acceptable publisher. However, since public-based encryption itself is considered ineffective, hard-to-use and difficult to deploy, it is necessary to develop new and flexible mechanisms to manage the user's trust mechanism in CCN. Studies [24], [25], [26], [27], [28] have made some researches in CCN security.

3.2 Breadcrumbs

Many studies about routing and forwarding policy have been provided. Most CCN studies default shortest path routing (SPR) for request routing. In SPR, only cached contents on default-path, i.e. the shortest path from a consumer to the content server, can be utilized. This means cached content outside this default path cannot be used even though it happens to be located closer to the consumer.

A simple content caching, location, and routing system is proposed for solving problems about caching. That method [9] is called as breadcrumb (BC). BC has adopted an implicit and transparent approach [4] which would make a big effort towards caching. In this network architecture, every router owns a local cache for storing the files which would be passed by some network like CCN. When it requests for one content, requests will be forwarded to the server of the current content as default. And the event of en-router checkers will happen, when the current request is passed by to find out whether the content has been copied and cached at this local cache. If a copy of this content is present at this local cache, then the requested content will be downloaded directly right here.

The simple introduction of BC has been shown; the basic architecture of BC will be discussed as below.

In one caching network, each node will be connected with a router and a local cache as well. And the routing history will be stored for each contents passing by, which can be called as Breadcrumb (BC). Every BC is a 5-tuple entry, which is indexed by a global file ID. The global ID contains much important information about the content. Global unique ID of the content is selected as an important part for identifying the content. Secondly, the ID must take along the identification which records node from which the content arrived and node to which the content was forwarded. Moreover, it will also record the content which passed through the node and was requested at the node in most recent moment. And it will be transformed into ID.

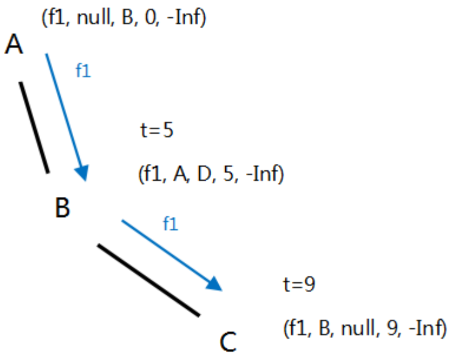

Figure 4. Breadcrumb example

In the above picture, content f_1 is downloaded and passed by from A to B and from B to D. In the final path, it will be delivered to the user node D. -Inf is used for representing for the most recent request. When the content is downloaded, it will leave a path of BC along the download path which is called as a BC trail.

The request is forwarded towards the source which is the public server of the content. On the path of routing to the server, it may encounter a router which carries breadcrumb for the current requested content. Thus, it will intercept a BC trail for this content.

Firstly, the router will check its local cache. If there is a copy of the content, the request may be satisfied to the content at this local node. On the other hand, the request should be routed upstream or downstream alternatively corresponding to the BC trail. The content will be found by following the BC trail upstream to the source at most time. However, it will be possible to be found downstream to the source much faster. Here is a similar notion of routing towards the source. The method [29] that exploiting state found at an intercepting node is used in multicast tree construction in core-base multicast routing. BC architecture chooses to forward request towards downstream in which content object are much more possible to be located. That is because, downstream caches are relatively later created then upstream.

What's more, BC policy uses a threshold value T_f to control the lifetime of BC trails. Only when the content was last cached and forwarded within time T_f in a node, requests will be sent towards downstream from this node; otherwise, requests will continue to be forwarded toward upstream which direct the source. If the value of T_f is small, the chance for requests to be forwarded upstream to the source will be higher. Therefore, less requests can take the advantages of possible nearby copies of the objective content. On the other hand, for higher values of T_f, more requests will be sent downstream, but more risks to get a hit miss.

As what's discussed above, breadcrumbs have an expiring algorithm for each BC item in each node. It is called the Best Effort Content Search (BECONS): Let c be a cache node and assume that a request q_f arrives at time t and finds that f does not exist in c. Then, for T_f – time when file passing by, and T_{qf} – time when a query for the file passing by, node c will forward q_f downstream if-and-only-if when file f is cached or refreshed by c in [t− T_f, t] or the request is sent to the downstream by c in [t−T_{qf}, t].

BECONS dose not involve any explicit communication between neighbor nodes. There is no additional maintain-messages which means traffic loads in whole networking. T_f and T_{qf} can be chosen for different networking conditions. They can be identical for all files or changed separately for each node.

After discover content object, content object will be sent back to the user. This routing process of content is also called content download policy. There are two kinds of such download policies:

- Download Follows Query (DFQ) - the file backtracks along the route the request took.
- Download Follows Shortest Path (DFSP) - the file is sent along the shortest path to the destination.

These download policies have different performances because of the characteristics of themselves, but more importantly, they determine the new locations where the file will be re-cached on its way to the destination, which mean new Breadcrumb trails. These differences are illustrated in Figure 5.

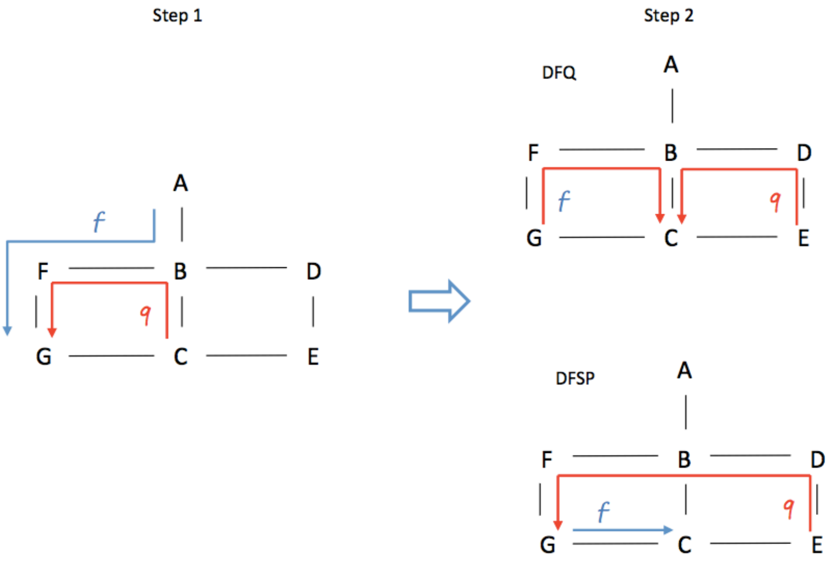

Figure 5. DFQ & DFSP. In first step, request is forwarded downstream to find a copy of the content object. Step 2 shows differences new content download paths with different download policies.

DFQ caches the file at the location in the hierarchy of the request arrives so that it prepares for other requests. This may lead to local popularity. However, in this case, only some caches will experience it because the request of a successful reply at the node will not be forwarded upstream.

DFSP, on the other hand, will deliver the file in the shortest possible manner to the user, in the given state of the system, though perhaps place file copies in locations where the file will not be requested at all. When employing DFQ, the file is cached only along nodes through which request passed, and specifically the end of the trail. From this node, the file is forwarded to the requested destination using shortest path routing. This causes a change in the forwarding table at this end node, since this new destination is now the most recent direction in which the file was sent. Thus, next time a cache-miss occurs at this node, the request will be forwarded along this fresher trail. DFSP, alternately, ensures a more stable request routing table. It can thus continue to forward requests downstream in the same direction.

Compared with using DFQ, DFSP is a better choice for Breadcrumb policy to forward content object.

4. Design Overview

The recent routing research in CCN focuses on how to utilize en-route caches in the process of interest packet routing. Some studies [31], [32] present solutions of searching en-route caches. However, these policies all result in redundant transmission of interest packets more or less. In order to obtain a better routing path, a study [33] proposed to use Ant Colony Algorithm to achieve the optimization of routing. However, it results in a heavier network burden due to depending on content collection crossing whole network too much. Another study [34] proposed a routing policy based on Ant Colony Algorithm and explicit coordinate cache information with neighbors. This strategy provides a collection of information on the adjacent node cache. There will be a large number of detection packets during detective process.

Study [35] introduced popularity into Ant Colony Algorithm Routing. It only detects and exchanges information of high-popular contents, thus searching algorithm can converge in a certain degree. These studies are mostly committed to optimize existing routing table items. Making a single outface is their way to control forwarding load and thus optimize routing. This kind of single-path routing in CCN, however, focus on this single path made by the finally chosen outface, so that reducing the utilization outside this path. Not only these possible available en-route caches, but also many edge caches are difficult to be located in routing paths, it is a challenge how to utilize these cache copies in single-path routing policy.

4.1. Research ideas

Essence of routing in CCN is actually how to build routing tables in each node. The original routing strategy in CCN tends to multicast through outfaces in routing tables, the benefit of doing so is that requests can quickly find objective contents, but will also leads to unpredictable workload for the network. Therefore, the current major work is focused on how to choose the best routing path and add corresponding faces to routing table. The main model of dealing with this problem is superimposing a searching layer to explore the best path to the content of the node.

There are two shortcomings for such an overlapping network. Firstly, the overlay usually introduces a large number of searching packets into the network and may lead to problem in consistency between layers; secondly, single-path routing cannot traverse enough nodes, especially the nodes at the edge of the network.

Study [9] designed a Breadcrumb policy as a routing strategy based on content-centric network. Trail is a "historical record" of data, it aims at tracing back to where the content has been downloaded and where has been copied. It should be defined as a linear guidance. It is now widely used in data navigation and searching engines. Breadcrumb records in each node along the Breadcrumb trail are not independent, if a breadcrumb record lost on a trail, the whole trail will be invalided.

Breadcrumb can be introduced into CCN routing strategy to help requests searching more nodes to find possible caches around. Reasonable uses of reversal paths may possibly locate objective content faster:

- More en-route caches along BC trail become the target of the request;
- There is no need for node to run complex algorithm or calculate addition information to locate possible objective sources.

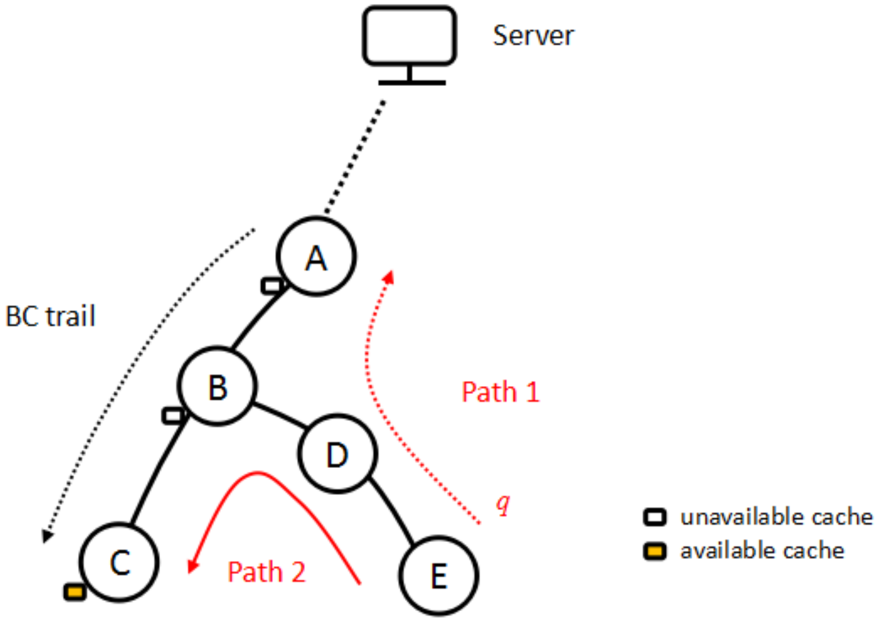

Figure 6. Utilization of edge cache in Breadcrumb policy.

When a query encounters a BC guidance in node B, it will choose path 2 but not default path 1 to find the copy of the content object in node C.

As we discussed above, each router stores a table recording BC that indicates the download direction of contents that ever pass by. When a request arrives and there is a BC corresponding to this request, then router will send it towards the BC's direction in a hop-by-hop manner [36]. However, multiple content downloads will lead to many obsolete BC trails in the network even though previous BC policy uses expired time to control the size of BC tables.

To counter this problem, we propose here a modified BC policy of deleting the BC once it has been used. We expect that this new solution can help to release BC table spaces and improve the hit performance of BC.

4.2. Self-cleaning Breadcrumb Routing Policy

BC policy uses a certain threshold time for each BC item to avoid invalid guidance. However, this expiration time is not adaptive to different networks or different popularity contents. In addition, with the content download policy DFSP (Download Follows Shortest Path), the BC trail will be extended significantly, as Figure 8 [9]. This might lead to more request-hops and more trail invalidations.

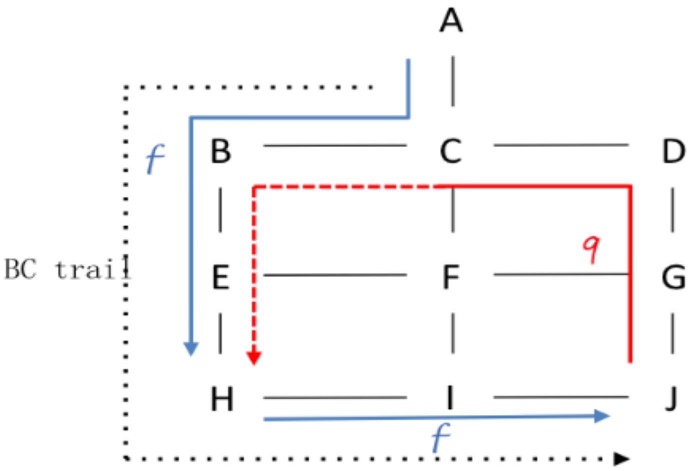

Figure 7. Lengthened BC trail in previous BC policy

To solve this defect, we propose removal of BC items once they are used. In request processing of a BC-enabled router in Figure 9, the router will check its content Store (CS) firstly, and then check BC Table if there is no content object match in CS. If there is a matched BC item, the request will be forwarded following BC guidance. Then, we suppose to delete this item immediately. This request will be forwarded along the BC trail. Thus, this BC trail will be cleared. We expect this design can help to remove overlong BC trails as well as obsolete ones, thus to improve the forwarding efficiency in the network.

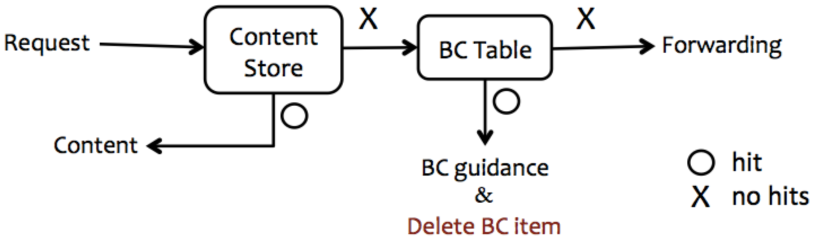

Figure 8. The structure of self-cleaning BC-enabled Router

5. Simulation model

We evaluated our proposal in the simulated networks which have 50 servers, 1000 nodes and 5000 users that are chosen randomly in the networks. The number of routers applying BC policy ranges from 100 to 1000 while the number of users applying BC policy ranges from 500 to 5000. Requests are generated by Zipf distribution (α=1.0). The number of contents is 10000. Each content is stored at a randomly selected server. The forwarding policy that we choose is SPR (Shortest Path Routing) according to the cost of links both for requests and contents. A content packet is cached at all routers along the download path, This policy is called LCE (Leave Copy Everywhere) which is used in the original CCN [30]. In the meanwhile, cache replacement policy is LRU (Least Recently Used). Simulation parameters are listed in Table 1.

Table 1. Simulation parameters

The number of Server	50
The number of Router	100~1000
The number of User	500~5000
The number of content	10000
Cache Size	2
Router cache size	5
BCLife	30
CacheLife	100

6. Result Analysis

We had several sets of parallel evaluation, and from the collected data, we got a set of diagrams.

BCmiss occurs when a request follow BC trail without finding object cache in the downstream. In Table 2 and Figure 10, we can see the number of BCmiss of the modified BC policy is less than the one of previous BC policy until the number of routers comes to 1000.

BC invalidation means that a request packet directed by BC cannot find the cache on routers along the BC trail. BC invalidation also occurs when a request packet arrives from an interface to which the corresponding BC is directed. In Table 3 and Figure 11 we measured the number of BC trail invalidations as the number of BC nodes increasing. We can see the new BC policy shows a great advantage in the number of BC invalidation. The much less invalidations mean BC trails are kept valid if they exist and request retransmission is much reduced. This may be caused by many redundant BC trails being removed promptly with applying the new BC policy. This helps to improve the BC cache hit rate and release the burden of routers.

From Table 4 and Figure 12, we evaluated the average request-hop length. We can see that the number of average hop that queries took reduces a little bit. This means the routing efficiency of request has been improved.

Table 2. The number of BCmiss

	original	modified	D-value	percentage
100	2238	2157	81	3.62%
200	8413	7954	459	5.46%
300	20760	19837	923	4.45%
400	39674	37465	2209	5.57%
500	71876	66290	5586	7.77%
600	100156	92687	7469	7.46%
700	127306	117496	9810	7.71%
800	142773	133890	8883	6.22%
900	151471	146106	5365	3.54%
1000	156136	156337	-201	-0.13%

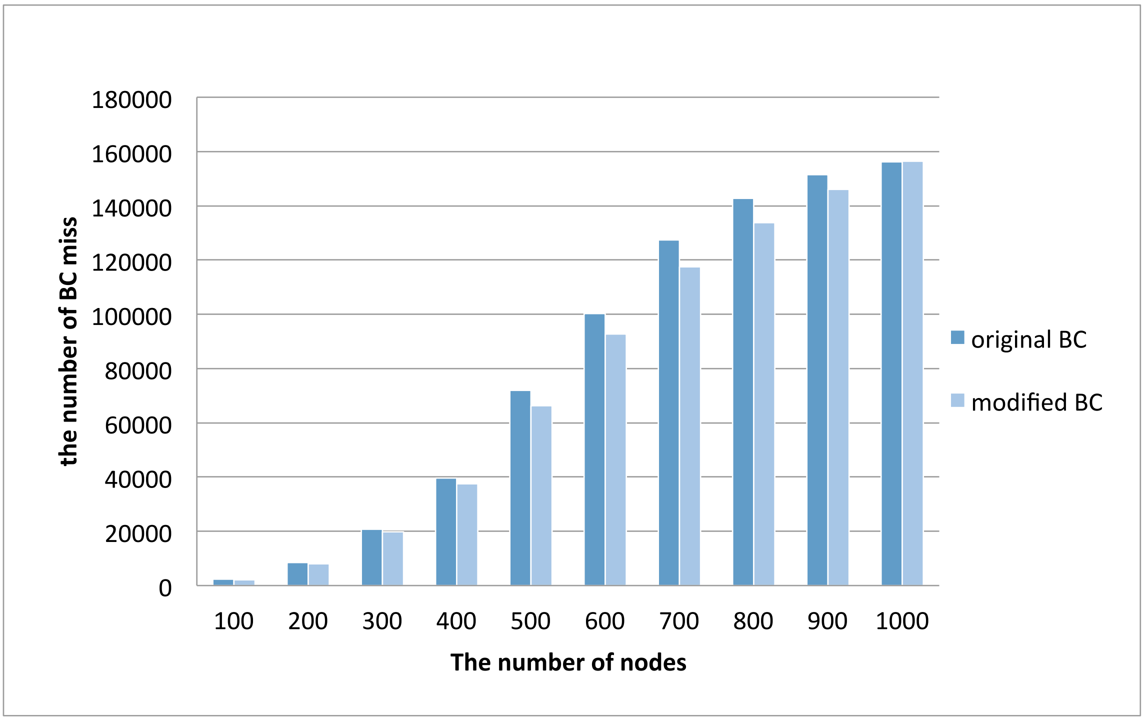

Figure 9. The number of BCmiss

Table 3. The number of invalidation

	original	modified	D-value	percentage
100	3092	831	2261	73.12%
200	13213	4089	9124	69.05%
300	36557	11943	24614	67.33%
400	72193	24942	47251	65.45%
500	143675	43392	100283	69.80%
600	221004	69435	151569	68.58%
700	304528	96016	208512	68.47%
800	415786	132457	283329	68.14%
900	518983	179703	339280	65.37%
1000	638692	244026	394666	61.79%

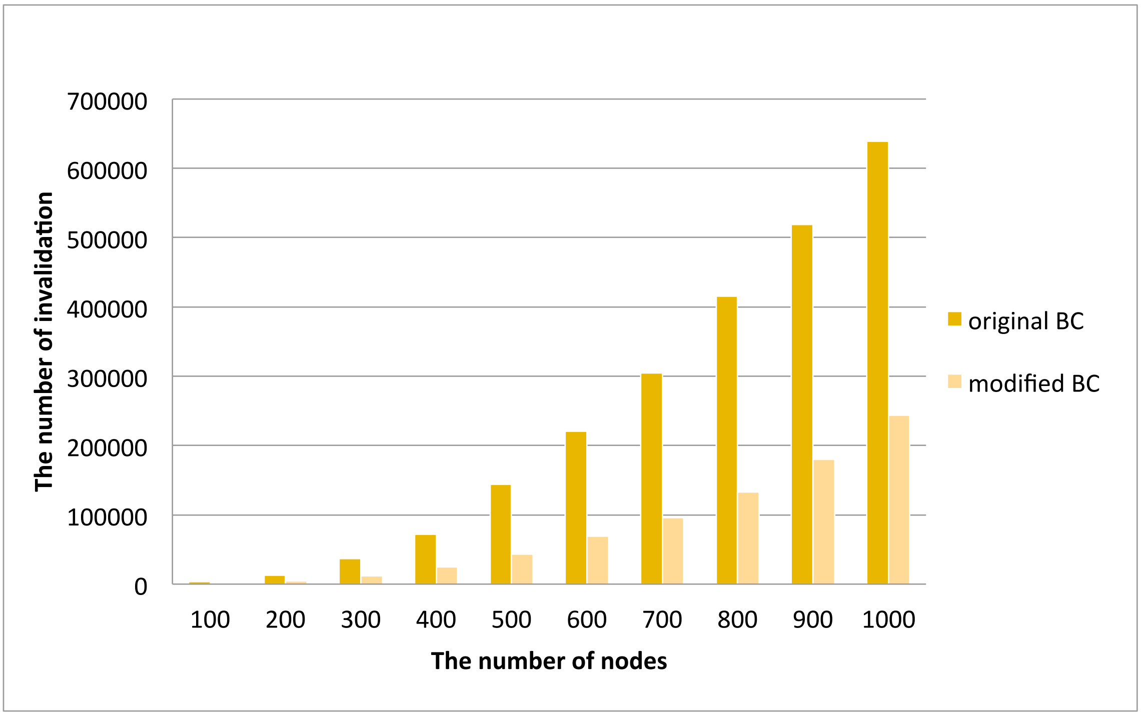

Figure 10. The number of invalidation

Table 4. The number of query-hop

	original	modified		
100	6.70637	6.70617	0.0002	0.00298%
200	6.67155	6.67043	0.00112	0.01679%
300	6.64266	6.63851	0.00415	0.06247%
400	6.58514	6.57473	0.01041	0.15808%
500	6.64623	6.60737	0.03886	0.58469%
600	6.69039	6.61281	0.07758	1.15957%
700	6.77059	6.64106	0.12953	1.91313%
800	6.98177	6.72488	0.25689	3.67944%
900	7.15824	6.77636	0.38188	5.33483%
1000	7.42475	6.87405	0.5507	7.41708%

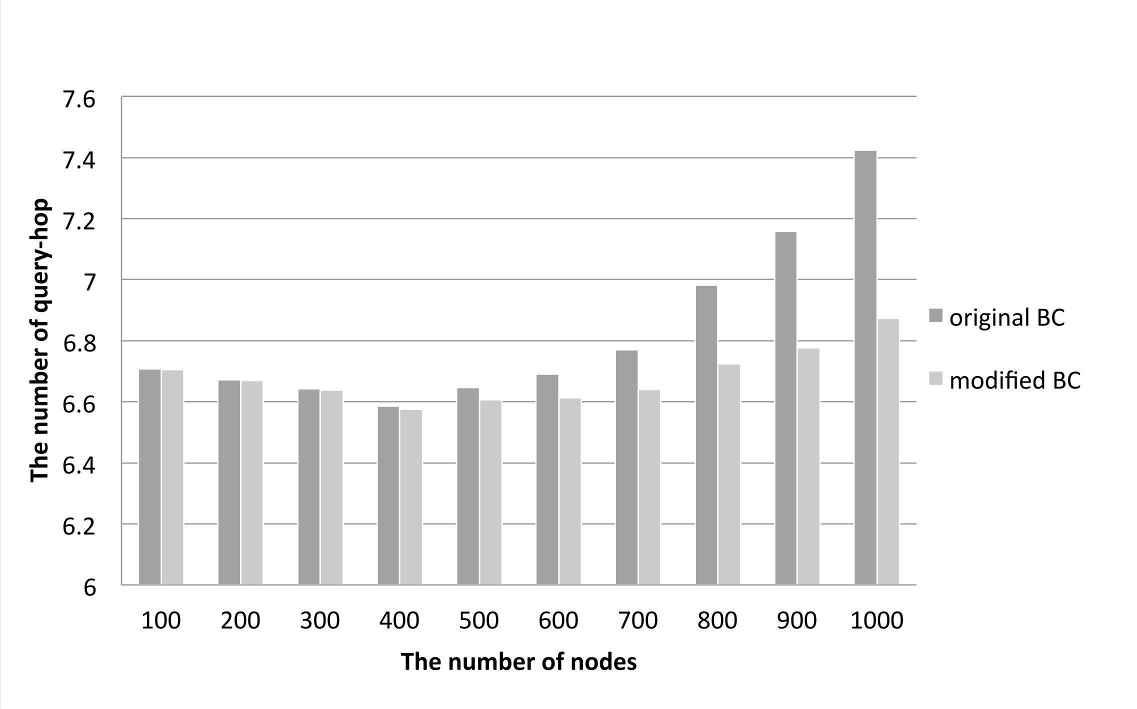

Figure 11. The number of query-hop

7. Conclusion

This paper describes the basic structure of data-oriented networks with local cache capabilities and the background of current researches, in which CCN is the most promising future network architecture. In CCN routing technology, how to improve the efficiency of routing and cache-hit ratio is the focus of the study. However, in the traditional single-path routing, the utilization of the cache near the edge of the network and the non-backbone nodes in the network have not received attention until now. In this paper, a new algorithm is proposed to improve the BC strategy for request routing in CCN, our evaluation results show that the strategy can effectively improve the cache hit rate of the edge node, improve the load condition of the server, and reduce the query-hop when user is far away from the server.

Based on the work of this paper, we can try to use different caching strategy and caching replacement strategy with BC routing strategy to find better network settings under different topologies to find a better combination to improve routing efficiency in our future work.

I. Acknowledgment

We thank Prof. Miki Yamamoto of Kansai University, Prof. Hideki Tode of Osaka City University, and Prof. Chikara Ohta of Kobe University for their help in evaluating our algorithm.

II.Reference

[1] A. Datta et. al. World Wide Wait: A Study of Internet Scalability and Cache-Based Approaches to Alleviate It, Management Science Volume 49 , Issue 10, October 2003, pp. 1425 - 1444.

[2] D. Raychaudhuri, R. Yates, S. Paul, J. Kurose, “The Cache-and-Forward Network Architecture for Efficient Mobile Content Delivery Services in the Future Interne t, ITU-T Innovations in NGN,” ITU-T Innovations in NG, N May 2008.

[3] Van Jacobson, “A New Way to Look at Networking”, http://video.google.com/videoplay?docid=-6972678839686672840

[4] Miki Yamamoto, “A Survey of Caching Networks in Content Oriented Networks.” IEICE TRANSACTIONS on Communications, Vol. E99-B, No.5, pp. 961-973.

[5] N. Laoutaris, S. Syntila and I. Stavrakakis, “Meta Algorithms for Hierarchical Web Caches.” in Proc. IEEE ICPCC’04, pp. 445-452, Bangalore, India, Aug. 2004.

[6] I. Psaras, W. K. Chai and G. Pavlou, “Probabilistic innetwork caching for information-centric networks.” in Proc. ACM ICN’12, pp. 55-60, Helsinki, Finland, Aug. 2012.

[7] W. Chai, D. He, I. Psaras and G. Pavlou, “Cache “less for more” in information-centric networks.” Computer Communications, Vol.36, No.7, pp. 758-770, Apr. 2013.

[8] S.Podlipnig and L.B.Osz, “A Survey of Web Cache Replacement Strategies,” ACM Computing Surveys, Vol.35, No.4, pp. 374-398, Dec. 2003.

[9] E. J. Rosensweig and J. Kurose, “Breadcrumbs: Effcient, best-Effort Content location in Cache Networks.” in Proc. IEEE INFOCOM’09, pp. 2631-2635, Rio de Janeiro, Brazil, Apr. 2009.

[10] T. Mori, H. Nakajima and M. Yamamoto, “A Study on In-Network Interest Guidance Method in Content-Centric Networking,” (in Japanese), B-6-42, IEICE Communications Society Conference, Sapporo, Sept. 2015.

[11] Lu Xuan, Gong Xiangyang, Cheng Ruishi, “New Internet network architecture” ZTW technology,2009(4):57-60.

[12] P. Eugster et al., “The Many Faces of Publish/Subscribe,” ACM Computing Surveys 35, 2003, pp. 114–31.

[13] T. Koponen et al., “A Data-Oriented (and Beyond) Network Architecture,” SIGCOMM ’07, 2007, pp. 181–92.

[14] V. Jacobson et al., “Networking Named Content,” CoNEXT ’09, New York, NY, 2009, pp. 1–12.

[15] K. Visala et al., “An Inter-Domain Data-Oriented Routing Architecture,” ReArch ’09: Proc. 2009 Wksp. Rearchitecting the internet, New York, NY, 2009, pp.55–60.

[16] A. Carzaniga et al., “A Routing Scheme for ContentBased Networking,” IEEE INFOCOM ’04, Hong Kong, China, Mar. 2004.

[17] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio and D. Perino, “Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN.” in Proc. ACM ICN’12, pp. 7-12, Helsinki, Finland, Aug. 2012.

[18] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi and G. Rossini, “INFORM: a dynamic INterest FORwarding Mechanism for Information Centric Networking.” in Proc. ACM ICN’13, pp. 9-14, Hong Kong, China, Aug. 2013.

[19] S. Eum, K. Nakauchi, M. Murata, Y. Shoji and N. Nishinaga, “CATT: Potential Based Routing with Content Caching for ICN.” in Proc. ACM ICN’12, pp. 49-54, Helsinki, Finland, Aug. 2012.

[20] Acs G,Conti M, Gasti P, et al. Cache privacy in named-data networking [C] . In Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on, 2013: 41-51.

[21] Carofiglio G, Gehlen V, Perino D. Experimental evaluation of memory management in content-centric networking [C] , In Communications (ICC), 2011 IEEE International Conference on, 2011:1-6.

[22] Psaras I, Clegg R G, Landa R, et al. Modelling and evaluation of CCN-caching trees [M] // Psaras I,Clegg R G,Landa R, et al. NETWORKING 2011. Springer, 2011: 2011: 78-91.

[23] Carofiglio G, Gallo M' Muscariello L, et al. Modeling data transfer in content-centric networking [C] . In Proceedings of the 23rd International Teletraffic Congress, 2011: 111-118.

[24] Compagno A, Conti M, Gasti P,et al. Poseidon: Mitigating interest flooding DDoS attacks in named data networking [C] . In Local Computer Networks (LCN), 2013 IEEE 38th Conference on, 2013: 630-638.

[25] Chaabane A, De Cristofaro E, Kaafar M A, et al. Privacy in content-oriented networking: Threats and countermeasures [J] . ACM SIGCOMM Computer Communication Review, 2013,43 (3): 25-33.

[26] Gasti P, Tsudik G, Uzun E, et al. DoS and DDoS in named data networking [C] . In Computer Communications and Networks (ICCCN), 2013 22nd International Conference on, 2013: 1-7.

[27] Afanasyev A,Mahadevan P, Moiseenko I,et al. Interest flooding attack and countermeasures in Named Data Networking [C] . In IFIP Networking Conference, 2013,2013: 1-9.

[28] Burke J,Gasti P, Nathan N, et al. Securing instrumented environments over Content-Centric Networking: the case of lighting control [J] . arXiv preprint arXiv: 1208.1336, 2012

[29] A. J. Ballardie, P. F. Francis, and J. Crowcroft (August 1993). ”Core Based Trees”, ACM SIGCOMM Computer Communication Review, 23 (4): 85 95.

[30] V. Jacobson, D. K. Smetters, J. D. Thonton, M. F. Plass, N. H.Briggs and R. L. Braynard, “Networking Named Content.” in Proc. ACM CoNEXT’09, pp. 1-12, Dec. 2009.

[31] Wong W, GiraMi M, Magalhaes M F, et al. Content routers: Fetching data on network path [C] . In Communications (ICC), 2011 IEEE International Conference on, 2011: 1-6.

[32] Daras P, Semertzidis T, Makris L, et al. Similarity content search in content centric networks [C] . In Proceedings of the international conference on Multimedia, 2010: 775-778

[33] Shanbhag S, Schwan N, Rimac I,et al. SoCCeR: Services over content-centric routing [C] , In Proceedings of the ACM SIGCOMM workshop on Information-centric networking, 2011: 62-67.

[34] Ye R, Xu M. Neighbor cache explore routing strategy in named data network [J] . Jisuanji Kexue yu Tansuo,2012, 6 (7): 593-601.

[35] He Jingjing, “Research on optimal algorithm of routing policy in Content-centric network”, [D]. [S]:Beijing University of Posts Telecommunications, 2013.

[36] Akihisa Shibuya, Yusaku Hayamizu, Miki Yamamoto, “Cache Decision Policy for Breadcrumbs in CCN.” in IEEE, 2016.