Kyushu University

Doctoral Thesis

Researches on Optimized Caching and Forwarding of Content-Centric Networking

Chengming Li

January 2015

Declaration of Authorship

I, Chengming Li, declare that this thesis titled, 'Researches on Optimized Caching and Forwarding of Content-Centric Networking' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Author:	*Supervisor:*
Chengming Li	Prof. Koji Okamura

Signed:

Date:

Excellence is a characteristic of personality.

Success in life requires focusing on and perseverance more than talent and opportunity.

KYUSHU UNIVERSITY

Abstract

Faculty of Information Science and Electrical Engineering

Department of Advanced Information Technology

Doctor of Philosophy

Researches on Optimized Caching and Forwarding of Content-Centric Networking

by Chengming LI

Content-Centric Networking (CCN) has recently attracted a lot of attention of researchers, which emerges as a novel clean slate network architecture. The core feature of CCN is changed from host-centric communication model to content-centric model. In CCN, contents are retrieved directly by their names, instead of locations. The general infrastructure of CCN is in-networking caching. Content in CCN is distributed in cost efficient and secure manner. However, in order to be deployable at a global scale, CCN must develop effective caching and forwarding engines. There are some challenge issues needed to be solve in caching and forwarding, such as scalability, intelligent forwarding strategies, power consuming and cache management.

Firstly, this thesis proposed a Greening Domain and Ant Colony based Forwarding (GDACF) scheme for CCN. One of major challenges in CCN is how to support a s-calable intelligent *Interests* forwarding strategy while considering power consuming and allowing huge name space. To address this challenge, GDACF divides the network into multiple domains and adopts the domain-based aggregation to improve its scalability. Popularity inspired caching and adaptive rate (AR) are used to improve energy-efficiency of CCN. GDACF inherently leverages parallelism and stochastic features of Ant Colony Optimization algorithm for supporting Quality of Service aware intelligent forwarding of *Interests*. It adaptively reduces impacts incited by the dynamic complex network. The evaluation results show that GDACF scheme improve the stability, robustness and energy-efficiency of Content-Centric networking.

Secondly, this thesis proposed a Game theoretical *Interest* Multiple Forwarding (GIMF) method for effective usage of cached data. It is common that the current router does not have all the information of cached data in network, because of the huge naming space and volatility of Content Store in each router. Thus, this thesis argues that it is necessary to supplement CCN with mechanisms to make Interests multiple forwarding for cached data. The goal is to maximize the residual capacity in the network so that

users can get the maximum payoff in a definite network situation. This proposals use a game theoretical method to analysis the properties of user behavior. Evaluation results prove that this method improves user's payoff in the light load case for content-centric networking.

Lastly, this thesis proposed a cluster-based in-networking caching mechanism to improve the cache hit ratio and reduce caching redundancy for CCN. One distinctive feature of CCN infrastructure is in-networking caching. As cache capacities of routers are relatively small compared with delivered data size, one challenge of in-networking caching is how to efficiently use the cache resources. This part designed the improved K-medoids cluster algorithm to cluster the whole network into k clusters and used Virtual Distributed Hash Table (VDHT) to efficiently control and manage the resources stored in each cluster. This part also proposed different policies for intra cluster routing and inter cluster routing to effectively forwarding requests. Compared with representative on-path caching schemes and hash scheme by simulation, it can be concluded that cluster-based in-networking caching mechanism can improve the cache hit ratio and reduce link load of networks.

Acknowledgements

First of all, I would like to express my sincere thanks to my supervisor Professor Koji Okamura, whose careful guidance taught me a glimpse of the esoteric field of future internet. Without his kindness and guidance, it would not be possible for me to complete this thesis. I also benefited from his patient assistance and friendly encouragement in these years. Most importantly, his critical thinking, copious knowledge, scholarly expertise and rigor to academic is worthy of me to learn. Furthermore, I would like to thank Professor Sachio Hirokawa and Professor Hirofumi Amano. Without their comments and suggestion, I can not finish the thesis well.

Secondly, I want to express my thanks to Ms. Wenjing Liu. Without her help and encouragement, I can not finish my Ph.D. study well. Needless to say, all her assistance has greatly facilitated the investigation and also enhanced the quality of this research.

Thirdly, I would like to thank all members of Professor Okamura's laboratory. Thanks a lot for your suggestion, comments and help.

In addition, I would like to thank China Scholarship Council for supporting my study in Japan. Also, I would like to thank Kyushu University giving me a warm place to grow and go ahead.

Finally, I would like dedicate this document to my beloved parents.

Contents

List of Figures

List of Tables

Abbreviations

CCN	**C**otent **C**entric **N**etworking
NDN	**N**amed **D**ata **N**etwoking
GDACF	**G**reening **D**omain and **A**nt **C**olony based **F**orwarding
GIMF	**G**ame theoretical **I**nterest **M**ultiple **F**orwarding

Chapter 1

Introduction

1.1 Background

The main role of today's internet is content delivery. Cisco Visual Networking Index 2014 shows that global IP traffic has increased more than fivefold in the past 5 years, and will increase threefold over the next 5years; Content delivery networks will carry over half of Internet traffic by 2018; Globally, IP video traffic will be 79 percent of all consumer Internet traffic in 2018, up from 66 percent in 2013. Figure 1.1 depicts the growth of different types of network traffic.

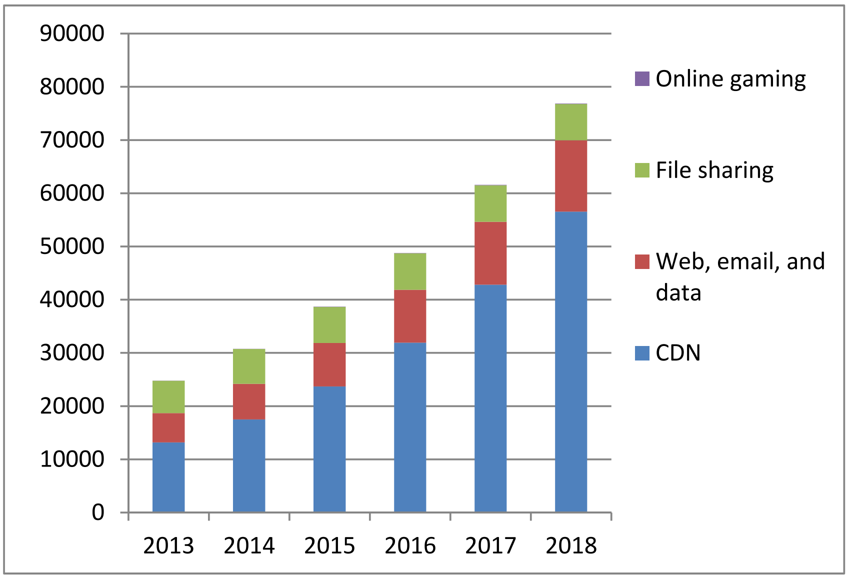

FIGURE 1.1: The growth of different types of network traffic

The Internet architecture has evolved substantially from host-centric communication model to content-centric model. The architecture of today's Internet is originally designed as a communication model that is a conversation between exactly two machines. However, it now mostly serves information-centric applications, e.g., Content Distribution Networks(CDNs) [1], P2P and Video(e.g., YouTube and Hulu). As demand for highly scalable and efficient distribution of content increases, the TCP/IP architecture may reveal its inefficiency in delivering time-sensitive multimedia traffic [2]. The Internet architecture has evolved substantially from host-centric communication model to content-centric model. Figure 1.2 illustrates the differences between host-centric communication model and content-centric model.

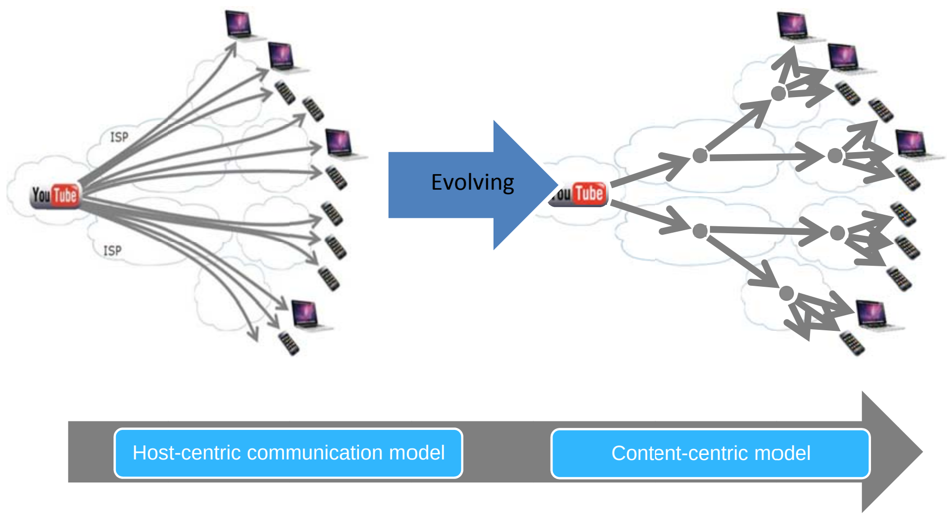

FIGURE 1.2: Host-centric communication model and content-centric model.

As the Internet architecture changing from host-centric communication model to content-centric model, there are a number of proposed architectures for Information-Centric Networking (ICN) including the Publish Subscribe Internet Routing Paradigm (PSIRP) [3], the Network of Information (NetInf) from the Design for the Future Internet (4WARD) project [4], the Cache-and-Forward Network Architecture [5], the Data Oriented Network Architecture (DONA) [6], TRIAD [7] and the Content Centric Networking (CCN) [8].

CCN is designed inherently focused on content distribution rather than host-to-host connectivity. In CCN, contents are retrieved directly by their names, instead of locations. The general infrastructure of CCN is in-networking caching. Content in CCN is distributed in a scalable, cost efficient and secure manner. The change from host-centric to content-centric has several attractive advantages, such as network load reduction, low dissemination latency, and energy efficiency [8]. While the general infrastructure of C-CN is in-networking caching, which allows any elements in the network to store content temporarily acting as servers.

1.2 Motivation and Goals

Although CCN decouples the location from network to support content delivery and has more advantages than traditional IP networking, it is incomplete and also has some challenge issues needing to be solved. The goals of this thesis are to improve the caching and forwarding performance of CCN in dynamic and complex network environment.

The first goal of this thesis is to propose a caching and forwarding strategy which delivers Interests in a scalable, QoS aware and energy-efficiency manner. Therefore, caching and forwarding strategy becomes a key component of nodes in CCN which makes them more powerful than their IP counterparts. Single shortest path can not perform well in CCN because a content object can have many destination nodes (it can be a router) by the form of replicas in CCN. Neither broadcast nor multiple path methods is efficient in CCN.

To the scalability issue, Diego Perino and Matteo Varvello take a reality check for CCN. They get the conclusion that a CCN deployment is feasible in a Content Distribution Network (CDN) and ISP scale, whereas today's technology is not yet ready to support an Internet scale deployment [9]. It is because that CCN shifts the address space from one billion IPs to at least one trillion content names [10], which causes a neat increase of forwarding table size at content routers.

To improve the forwarding performance of CCN, different routing and forwarding methods are proposed by researchers. Adaptive interest packets forwarding methods are described to improve stability of CCN [11, 12]. To reduce dissemination latency, ant colony based interest packets forwarding schemes [13–16] and Q-routing based forwarding scheme [17] were proposed. Garcia-Reinoso et al. [18] proposed scalable data replication in content-centric networking based on alias names to achieve the scalability of CCN. Abu et al. (2014) [19] focused on the interest packets retransmission in lossy CCN Networks. Hash routing scheme is proposed by Saino et al. [20] to improve the cache hit. However, none of them considered energy-efficiency issues.

There are also some researchers who focus on the energy efficiency of CCN. Hasegawa et al. [21] developed a power consumption model of a multicore software ICN router while taking into account the power consumed by power-hungry computation. Imai et al. [22]and Choi et al. [23] presented energy aware cache management focusing on cache location for CCN, which caches content appropriately in consideration of both power consumption of content caching and transmission of traffic. Kim et al. [24] presented cache capacity-aware content centric networking under flash crowds. Li et al. [25] builded the energy consumption model by translating energy optimization into that of minimizing average response hops. These methods only focus on the cache location management neglecting the power consuming by routers and links.

Therefore, to improve forwarding performance of CCN, we should not only consider the Quality of Service issues (reducing dissemination latency), but also care energy efficiency. Above all, there are three major challenges in caching and forwarding:

- Reducing the forwarding table size while allowing an unbounded name space for scalability;
- Supporting intelligent forwarding of Interests over multiple optional paths to guarantee the QoS;
- Optimal joint caching and forwarding to reduce the network traffic and improve the energy efficiency.

The second goal of this thesis is to improve the utilization of cached data in CCN. The main reason of that CCN can deal with content distribution better than end to end network model is cached data in networking. Thus, it is a challenge that how to efficiently utilize the cached data. In some cases, the content objects are so many that the CS cannot efficiently manage them, which may result in poor caching performance. Forwarding Information Base (FIB) of routers cannot contain all the content as the huge naming space; and as the content cached in Content Store of routers is changing frequently, it is very difficult to update the FIB in time for all content objects in the network. Thus, it is a problem that how to search the cached data efficiently.

Interest multiple forwarding is a key feature in CCN. This feature makes them more powerful than their IP counterparts. Routing of IP network is to calculate a single shortest path for each pair of source node and destination node. The forwarding strategy layer in a CCN node can dynamically select multiple interfaces from the FIB to forward a same Interest packet. Single shortest path can be a candidate forwarding strategy for CCN. However, it cannot perform well as it runs in end to end communication network.

In end to end communication network the destination node is definite, but in CCN a content object can have many destination nodes (it can be a router) by the form of replicas.

It is necessary to supplement CCN with mechanisms making the Interest forwarding decisions. In the case of sufficient network resources, delivering the Interest packet to multiple interfaces derived for FIB can achieve following advantages:

- The real-time decision enables nodes to fully utilize their rich connectivity and get the best users' payoff;
- It defends against route hijacking attacks (if no data returns over a particular interface for a particular name, that interface may not lead to a valid path for that name);
- It enhances the network instability (frequent oscillation of paths) while maintaining good data delivery performance.

The third goal of this thesis is to improve the cache diversity and cache hit. In-networking catching is the distinctive feature of CCN infrastructure and plays an important role in terms of system performance. In-network caching mechanism can avoid wasting network bandwidth due to the repeated delivery of popular content. Additionally, it can reduce response time for content by placing the content closer to users. The challenges surrounding in-networking caching involves cache placement, cache replacement and network cache model, etc. However, Kutscher, et al [26] define three key issues which influence the performance of in-networking caching system, i.e., cache placement, content-placement, request-cache routing. Cache placement mainly focus on deciding which nodes are supposed to upgrade for in-networking caching in a domain, which are mainly related with the whole network planning, such as, the network topology, traffic and positions [27][28]. As for content-placement, it is an issue about the distribution policy of contents across in-networking caches in a domain. However, request-cache routing solves the problem of actions took for a content request corresponding to node caches. Above all, this thesis focuses on the content placement issue and the request-cache routing issue.

As in-networking caching is so important for CCN, lots of in-networking caching strategies have been presented till now. There are mainly three caching strategies for in-networking caching. The "on-path catching" strategy is the one which allows contents to be cached temporarily at nodes on the path from content providers to consumers. This strategy reduces bandwidth consumption and content retrieval time by allowing contents closer to consumers. However, it has been demonstrated that this strategy is not optimal as it may imply a high content replication that limits the maximum number of contents that can be cached inside a domain [29][30]. Therefore, "off-path caching" is an alternative strategy that can avoid duplications and can significantly increase the overall hit radio [30]. While, the "off-path caching" limited the scalability of CCN for its per-content state required for routing. Hence, mixed techniques were proposed, like SCAN [31], which mix features of on-path and off-path techniques.

Due to content popularity, often, the same content is accessed by many users, which makes network traffic exhibit high redundancy. Furthermore, even CCN enables individual nodes to reduce redundancy by managing a local cache, redundancy can freely appear across different nodes as the default ubiquitous LRU caching scheme and the support of multi-path routing. These two aspects motive us to consider that controlling the redundancy level is a critical issue to improving the systematic caching performance of CCN.

The reminder of this thesis is organized as follows. Chapter 2 illustrates technical background and related works on Content-Centric Networking. A Scalable QoS Aware Forwarding Method for CCN is presented in Chapter 3. Chapter 4 describes a game theoretical Interest Forwarding Decision methed for cached Data in CCN. Cluster based in-networking caching scheme is proposed in Chapter 5 to improve the cache hit rate and cache diversity. At last, Chapter 6 concludes all the works.

Chapter 2

Content-Centric Networking

2.1 Architecture

CCN names are opaque to the network, i.e., routers do not know the meaning of a name (although they know the boundaries between components in a name). This allows each application to choose the naming scheme that fits its needs and allows the naming schemes to evolve independently from the network. CCN design assumes hierarchically structured names, e.g., a movie produced by Youtube may have the name /Youtube /movies/Example.rmvb.

A node in CCN consists of the Content Store (CS), the Pending Interest Table (PIT), and the Forwarding Information Base (FIB) [32]. The architecture of CCN is depicted in Figure 2.1.

- Content Store (CS): temporarily buffers Data packets that pass through this router, allowing efficient data retrieval by different consumers;

- Pending Interest Table (PIT): records currently not yet-satisfied interests and a set of corresponding incoming interfaces. Each PIT entry contains one or multiple incoming and outgoing physical interfaces; multiple incoming interfaces indicate the same data is requested from multiple downstream users; multiple outgoing interfaces indicate the same Interest is forwarded along multiple paths;

- Forwarding Information Base (FIB): maps name prefixes to one or multiple physical network interfaces, specifying directions where Interests can be forwarded.

Communication in CCN is driven by the receiving end, i.e., the data consumer. All communications in CCN are performed using two distinct types of packets: Interest

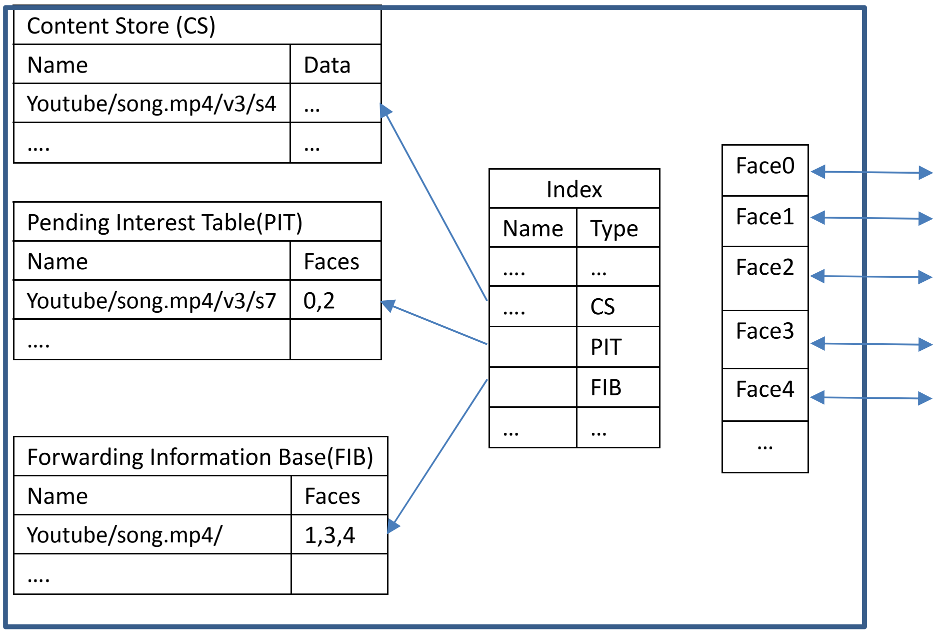

FIGURE 2.1: Architecture of Content-centric Networking.

and Data. Both types of packets carry a name, which uniquely identifies a piece of content that can be carried in one Data packet. To receive data, a consumer sends out an Interest packet which carries a name that identifies the desired data. The routing process is described in Figure 2.2.

When the Interest Packet arrives at a CCN router, the router consults the CS, PIT and FIB in that sequence. The router first checks whether the data requested have already been presented in the node's Content Store (CS) which is used to store the coming data packet by a cache replacement policy. If there is no matched data, the router will check whether the PIT has included the same Interest. In PIT, each entry contains the name of Interest and a set of interfaces from which the Interest packets have been received. If the PIT already contains the same Interest, then the router adds the Interest coming interface to the corresponding entry of PIT. Finally, the router remembers the interface from which the request comes, and then forwards the Interest packet by looking up the name in its Forwarding Information Base (FIB), which is populated by a name-based routing protocol.

Once the Interest reaches a node which contains the requested data, a Data packet, which carries both the name and the content of the data, is sent back together with a signature by the producer's key. This Data packet trace in the reverse path created by the Interest packet back to the consumer. When a Data packet is received, its name is

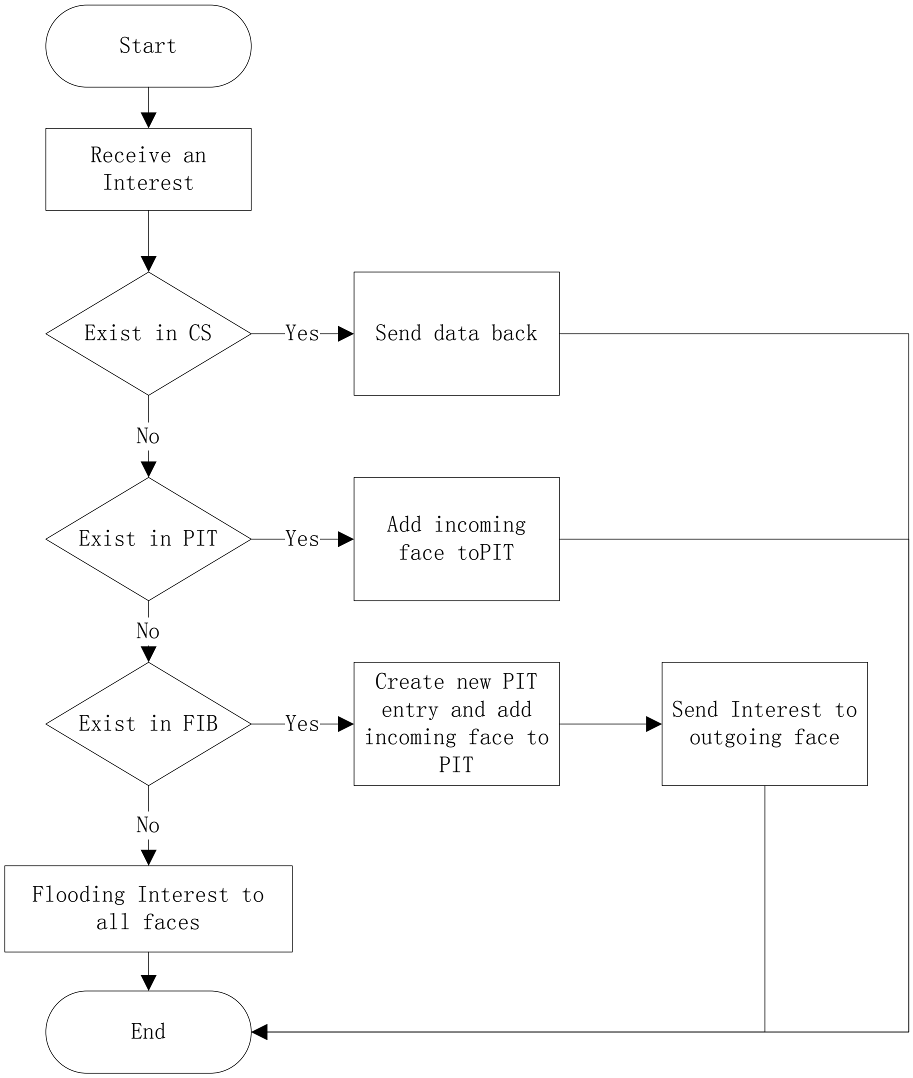

FIGURE 2.2: Routing process of Content-centric Networking.

used to look up the PIT. If a matching PIT entry is found, the router sends the Data packet to the interface(s) from which the Interest was received, caches the data in the CS, and removes the PIT entry. Otherwise, the Data packet is deemed unsolicited and is discarded. Each Interest also has an associated lifetime; the PIT entry is removed when the lifetime expires. Although the maximum lifetime is specified by users, it is ultimately a router's decision on how long it is willing to keep a PIT entry. For simplicity, we assume routers keep Interests in PIT for one second in our simulations.

2.2 Routing and Forwarding Strategies

A common feature of all the routing algorithms is the presence in every network node of a data structure, called routing table, holding all the information used by the algorithm to make local forwarding decisions. The routing table is both a local database and a local model of the global network status. The routing algorithms can be classified into centralized routing and distributed routing based on routing table managed by central controller or local router. They also can be classified into static routing and adaptive routing based on whether router concerns current network state when it routs packet. The distributed and adaptive routing is expected in the majority of network systems.

CCN routes and forwards packets on names, which eliminates four problems that addresses pose in the IP architecture: address space exhaustion, NAT traversal, mobility, and scalable address management [32]. Routing can be done in a similar fashion to today's IP routing. Instead of announcing IP prefixes, a CCN router announces name prefixes that cover the data that the router is willing to serve. This announcement is propagated through the network via a routing protocol, and every router builds its FIB based on received routing announcements.

The paper [32] proposed tackling two major challenges in routing: (a) bounding the amount of routing state while allowing an unbounded name space; and (b) supporting intelligent forwarding of Interests over multiple paths. The paper [12] took the routing as a role in CCN. They specify how CCN routing coordinates with forwarding through interface ranking and probing mechanisms. CCN routing protocols can benefit from the forwarding plane due to the relaxed requirement on timely detection of failures and convergence delay. Consequently NDN routing stability and scalability can be greatly improved.

Forwarding strategy is a key component in CCN nodes that makes them more powerful than their IP counterparts. Routing of IP network is to calculate a single shortest path for each pair of source node and destination node. The forwarding strategy layer in a CCN node can dynamically select multiple interfaces from the FIB to forward a same Interest packet. Single shortest path can be a candidate forwarding strategy for CCN. The paper [33] proposed an OSPF routing algorithm for CCN. However, it cannot perform well as it runs in end to end communication network. In end to end communication network the destination node is definite, but in CCN a content object can have many destination nodes (it can be a router) by the form of replicas.

Adaptive Interest forwarding method is presented in paper [34]. This paper uses Interest NACK to detect the Duplicate, Congestion and No data cases. To rank the interfaces for a content in a node, they classify the interfaces into three levels: Green(the interface is working), Yellow(the interface may work or not) and Red(the interface does not work). They adopt rate limitation to control congestion. However, this scheme can not rank the interfaces accurately based on the networking state.

The authors Carofiglio, et al, proposed a multi-path congestion control scheme for CCN [35]. This paper propose the first time a congestion control mechanism realizing efficient multipath communication over content-centric networks. Their proposal is based on a Remote Adaptive Active Queue Management (RAAQM) at the receiver that performs a per-route control of bottleneck queues along the paths.

To improve the cache hit rate and reduce the traffic load, some Hash related routing schemes were proposed. Lorenzo Saino, et al, presented a hash routing scheme for CCN [20]. However, it is not suitable for large scale network. Hang Liu, et al, used multi-level Distributed Hash Table routing method to get content aggregation [36]. Although it is scalable, this method mainly focuses on content aggregation. All of them can not adaptively perceive network state changing.

Thus, the CCN still need an intelligent adaptive forwarding scheme to full use of its features.

2.3 Cache Management

The researches about catching have been widely carried out in the past [37]. While catching in CCN has its own content-oriented features. First of all, caching is a native property of routers in CCN. In CCN, request catching and content catching should be handled at a same network layer. That makes content retrieval and replacement be considered at line speed [38]. More clearly, a router should check if its local content store has the requested content before it sends a request to next hop. Secondly, the placements of the ubiquitous caches are arbitrary, but not hierarchical, which make caching in CCN different from Web-catching and Content Delivery Network (CDN). At last, as content chunks in CCN are identified by the unique names, different applications could use same cache space in a router at the same time. This is the most basic feature of catching in CCN, which make it different from web, CDN and P2P.

Caching has been studied for Web system, P2P systems, network system and so on, in order to improve performance of systems by reducing bandwidth usage, server load, and response time. Generally speaking, caching schemes can be classified into two types, i.e., centralized caching and decentralized caching. Centralized caching is one whose data are only distributed by a central node and request will be responded by this central node. One typical example is Web-caching. Centralized caching mechanisms do better in controlling and managing network resources. However, it not only increases communication overhead for updating the content location, but also reduces flexibility in terms of available cache locations. Decentralized caching mechanisms cache the content at any place of network and manage the cached data by servers or routers locally, e.g., in-networking caching in CCN. In-networking caching is one of main infrastructures of CCN. As it needs no communication overhead and its operation is location-independent, in-networking caching improves system performance. In CCN, in-networking caching schemes are mainly divided into three categories: on-path caching, off-path caching [20], and hybrid techniques.

On-path caching schemes have already been studied in the past [39], which focus on the issue of cache placement [27][40]. In CCN, Data packets in on-path caching schemes are stored in any on-path nodes [8] or a subset of traversed nodes [29][41] as they travelling through the network. Interests are delivered according to the defined forwarding policies and Data packets will be returned in the inverse way of Interests. Representative on-path routing schemes are Leave Copy Everywhere (LEC) [32], Leave Copy Down (LCD) [39], ProbCache [41], Centrality-based caching [29]. The problems of the popularity-driven content caching [42], content location [43] etc. are attracted lots of attentions. Even the scalability of on-path caching is strong, it limits cache hits due to redundant caching of contents.

Data packets in network using off-path caching are cached to node according the defined rules. Interests must be forwarded in the same rules, as Data packets are traveling the reverse ways of Interests. Generally, Interests are handled by nodes in network cooperatively [44][45]. Author Rosensweig et al. [46] proposed a method named "Breadcrumbs". By additionally storing minimal information regarding caching history, they developed a content caching, location, and routing system that adopts an implicit, transparent, and best-effort approach towards caching. Hash technologies are also studied for in-networking caching of CCN. Author Saino et al. [20] designed five different hash-routing schemes which exploited in-network caches without requiring network routers to maintain per-content state information. A domain is considered as a whole by these five schemes. In contrast to on-path caching, off-path caching owns a higher cache hits, but has limited scalability due to per-content state required for routing.

Hybrid techniques of on-path caching and off-path caching are also explored. By exploiting nearby and multiple content copies for the efficient delivery, SCAN [31] exchanges the information of the cached contents using Bloom filter. Compared with IP routing, SCAN can offer reduced delivery latency, reduced traffic volume, and load balancing among links. Hybrid techniques are supposed to mix features of on-path and off-path techniques and balance the performance of scalability and cache hits.

Cache management still is one of most important challenge issue for CCN.

Chapter 3

Energy Efficient QoS Aware Forwarding Method for Content-Centric Networking

3.1 Preface

As the content traffic increases and Internet is rapidly growing, power consumption has shown a growing and alarming trend in information and communication technology (ICT) [47]. The power cost brought by hardware devices such as network devices and servers in data centers accounts for a dominant part of the operational costs of data centers [48]. Both industry and academia have been striving to improve energy efficiency of networking devices [49] and eventually to realize so called energy-proportional networking: i.e., the energy consumption is proportional to utilization of network interfaces [50][51].

Since networking devices have a wide spectrum of energy consumption for packet forwarding, reducing transit traffic in the network and saving energy used for data transport is one important way to green internet. Lee et al. [52] and Butt et al. [53] confirmed that Content-Centric Networking (CCN) is more energy efficient than conventional CDNs and P2P networks, but CCN still needs to improve energy efficiency.

The main objectives of this chapter include (1) selecting performance metrics to rank interfaces, e.g., overhead, round trip delay, bandwidth; (2) supporting intelligent forwarding of Interests over multiple paths, which is Quality of Service aware and energy efficiency; and (3) avoiding instability (frequent oscillation of paths) while maintaining good data delivery performance. Different policies for in Intra-domain forwarding case and Inter-domain forwarding case are designed to achieve the goal of scalability and mobility.

This chapter presents a Greening Domain and Ant Colony based Forwarding (G-DACF) method as the joint caching and forwarding strategy for CCN. Firstly, the Domain-based aggregation is adopted to solve the problems caused by the large name space. The basic idea is that dividing the network into multiple ISP-based domains. Each domain is an autonomous system (AS) and is managed by a specified ISP. By this method, the large name space can be reduced dramatically. The domain name is used as the first hierarchical name to distinguish the different content with the same name. Users can name the sub-name of the content arbitrarily. This method can reduce the Forwarding Information Base (FIB) table size significantly.

Secondly, popularity inspired caching and adaptive rate (AR) are used to improve energy-efficiency of CCN. These methods improve the energy efficiency from two aspects: 1) reducing average content retrieve hops; and 2) reducing power consumed by routers and links. To reduce average content retrieve hops, Popularity Heuristic Caching(PHC) scheme is proposed; To reduce the power consumed by routers and links, Adaptive Rate (AR) which is on-line power adaptation is adopted.

Lastly, a revised Ant Colony Optimization based forwarding method is designed to decide the Interest packets forwarding strategy. CCN's native multipath forwarding and its symmetric routing, data being only sent back by traversing the reversed Interest path, inherently match the natural behavior of ants while searching for the shortest path between their nest and some food source. Therefore, a GDACF algorithm is proposed as the forwarding strategy for CCN to solve intelligent forwarding issues.

The reminder of this chapter is organized as follows. Section 3.2 illustrated technical background on Ant Colony Optimization algorithm. Section 3.4 presents the design details of GDACF in Intra-domain case and in Inter-domain case. At last, the evaluation of our proposal are discussed in section 3.5.

3.2 Ant Colony Optimization

Ant Colony Optimization (ACO) principles are based on the natural behaviour of ants while searching for the shortest path between their nest and some food source. Ants communicate indirectly by laying pheromone trails and following trails with higher pheromone. Pheromone will accumulate on the shortest path [54]. The Ant Colony is an artificial swarm intelligence system. Dorigo and Stutzle developed the first ant based algorithm which was called Ant System [55]. It was used to solve the Travelling Salesman Problem (TSP), a well-known NP-Hard problem.

The Ant System (AS) has two main phases: (1) the construction of the solution and (2) the updating of pheromone. ACO algorithm has been used to solve a wide variety of different optimization problems, ranging from scheduling, network routing to assignment problems.

Ants move on the network by going from one node to another. The next hop where it moves to is probabilistically chosen based on the pheromone quantities deposited on the edge outgoing from the current node. Since each ant starts from the same source node, when all nodes have been passed through, a feasible solution has been achieved. After the ants have constructed their respective solutions, the pheromone trails are updated. On the one hand, pheromone values are decreased through the evaporation. In the evaporation progress, pheromone values are decreased by a constant decay; on the other hand, pheromone values are increased in the parts of the network which are present in the best solutions [56–58].

AntNet [59] is one of the well-known ACO based routing protocols for packet switched networks. It is an alternative routing algorithm to the well-known OSPF protocol traditionally used for packet switched networks. In AntNet, a group of mobile agents (or artificial ants) build paths between pairs of nodes by exploring the network concurrently and exchanging obtained information to update the routing tables in the nodes. Hence, using ACO for routing in dynamic network seems to be appropriate [60].

Laura Rosati et al. [61] proposed an ACO algorithm which aims at minimizing complexity of the nodes by the expenses of the optimality of the solution. It is particularly suitable for environments where fast communication establishment and minimum signal overhead are requested. However, this proposal is optimal for a less number of nodes in the cluster and also not suitable for ad hoc network.

Shashank Shanbhag et al. [15] presented SoCCeR-Services over Content-Centric Routing. SoCCeR extends CCN with integrated support for service routing decisions leveraging ant-colony optimization. SoCCeR adds a control layer on top of CCN for the manipulation of the underlying Forwarding Information Base (FIB). SoCCeR routes service requests selectively to service instances with lighter loads and is highly responsive to network and service state changes.

3.3 Problem Analysis

Joint caching and forwarding strategies are one of the most important components in CCN. It makes them more powerful than their IP counterparts. Routing of IP network is to calculate a single shortest path for each pair of source node and destination node. Differently, the forwarding strategy layer in a CCN node can dynamically select multiple interfaces from the Forwarding Information Base (FIB) to forward each Interest packet. An example of Interest forwarding in CCN is depicted in Fig.3.1. After node 1 receiving Interest which requests a.file, there are two interfaces that node 1 can choose to forward the Interest and node 1 can dynamically choose either of them.

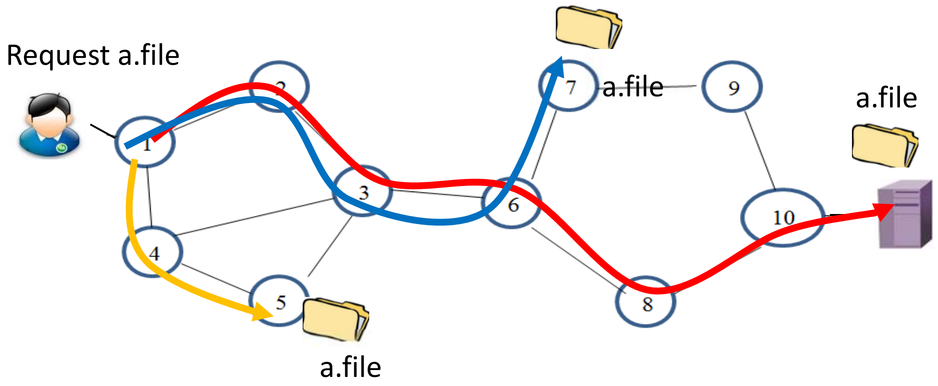

Figure 3.1: An example of Interest forwarding in CCN.

Therefore, a question is raised: which interface or interfaces should the nodes choose to forward Interests? The simplest strategy is to send an Interest on each of the interfaces in a FIB entry in sequence. If there is no response to the Interest, then try to send to next interface.

Single shortest path can also be a candidate forwarding strategy for CCN. It consumes low communication overhead and is suitable for the network with high traffic. However, as it runs in end to end communication network, volatile behavior of replicas and single point failure scenario lead it can not perform well. In the end to end communication network, the destination node is definite. However, in CCN, a content object can have many destination nodes (it can be a router) by the form of replicas. Furthermore, the huge content name space, especially in chunk level, makes it is very difficulty to update the FIB in time for all content object in the network. Thus, the shortest path record in FIB sometimes is not real shortest path for a content object. The disadvantages of single shortest path are low mobility, instability (congestion, node failure) and low round trip delay, but not minimum (a content object can have many replicas cached in the networking).

Interests also can be sent on all the interfaces at once and the node see which interfaces receive data first, scilicet broadcast. These interfaces will be used for a period of time and their performance are monitored. Even broadcast method has high mobility and ideal minimum round trip delay, sending all Interests to all interfaces can easily make the network overload and congested when the number of Interests is relatively big.

To scalability issue, D. Perino et al. [9] takes a reality check for CCN. They get the conclusion that a CCN deployment is feasible in a Content Distribution Network (CDN) and ISP scale, whereas today's technology is not yet ready to support an Internet scale deployment. It is because that CCN shifts the address space from one billion IPs to one trillion content names at least [10], which causes a neat increase of forwarding table size at content routers. Thus, global-scale CCN needs a scalable forwarding strategy.

To energy efficiency issues, in-networking caching is beneficial in view of energy-efficiency. Thus, energy efficient cache management schemes were presented to save energy by Imai et al. [22], Choi et al. [23], Li et al. [25]. These energy consumption models essential are the function of average response hops of content data. These methods only focus on the cache location management neglecting the power consuming by links. Thus, this chapter also use this energy consumption model regarding the average response hops. Therefore, there are two aspects for energy efficiency: 1) reducing average content retrieve hops; and 2) reducing power consumed by routers and links..

The main characteristics of the forwarding problem in CCN networks can be summarized in the following way:

- Intrinsically distributed: The contents and the decision system are completely distributed over all the network nodes. At each CCN node, the forwarding algorithm can only make use of local, up-to-date information, and of non-local, delayed information coming from the other nodes.

- Stochastic and time-varying: Interests arrive and content generation is non-stationary and stochastic. The status of network is also dynamic and time-varying because of congestion and node failure.

- Multi-objectives: several performance measures are usually taken into account. The most common are throughput(bit/sec) and average packet delay(sec). The former measures the quantity of service that the network is able to offer in a certain amount of time (amount of correctly delivered bits per time unit), while the latter denotes the quality of service produced at the same time.

Above all, the forwarding problems in CCN are a stochastic distributed multi-objective problem. Information propagation delays and the difficulty to completely characterize the network dynamics under arbitrary traffic patterns, make the general forwarding problem intrinsically distributed. Hence, forwarding decisions can only be made on the basis of current local information and approximate information about the future network states.

3.4 Greening Domain and Ant Colony based Forwarding Scheme

3.4.1 Domain based Network Framework

Based on previous work did in paper [1], we have known that a CCN deployment is feasible in a Content Distribution Network (CDN) and ISP scale. Whereas today's technology is not yet ready to support an Internet scale deployment. One of its main challenges is how to solve problems produced by the huge content naming space.

To solve the problems caused by the large name space, the domain-based aggregation is adopted to support a scalable forwarding method. The basic idea is dividing the networking into multiple domains based on ISP. A ISP may have a lot private domains. Based on this approach, a network can be modeled as a framework made up of domains. An example of network model is illustrated in Fig. 3.2. This method also enable CCN can get the maximum performance in a local network. This is also the reason why these ideas are taken. Each domain is an autonomous system (AS) with a center node and is managed by a specified ISP. This approach has two basic components: (a) hierarchical provider-assigned names to facilitate aggregation; and (b) a mapping service to map user-selected names to provider-assigned names [32].

According to the naming, a content name is composed of a Domain name, an ISP-assigned name and an user-selected name. The naming of CCN in this proposal is denoted in Fig. 3.3. In order to be compatible with the current Internet, this proposal can directly use the IP domain name as the ISP-assigned name of the hierarchical content name.

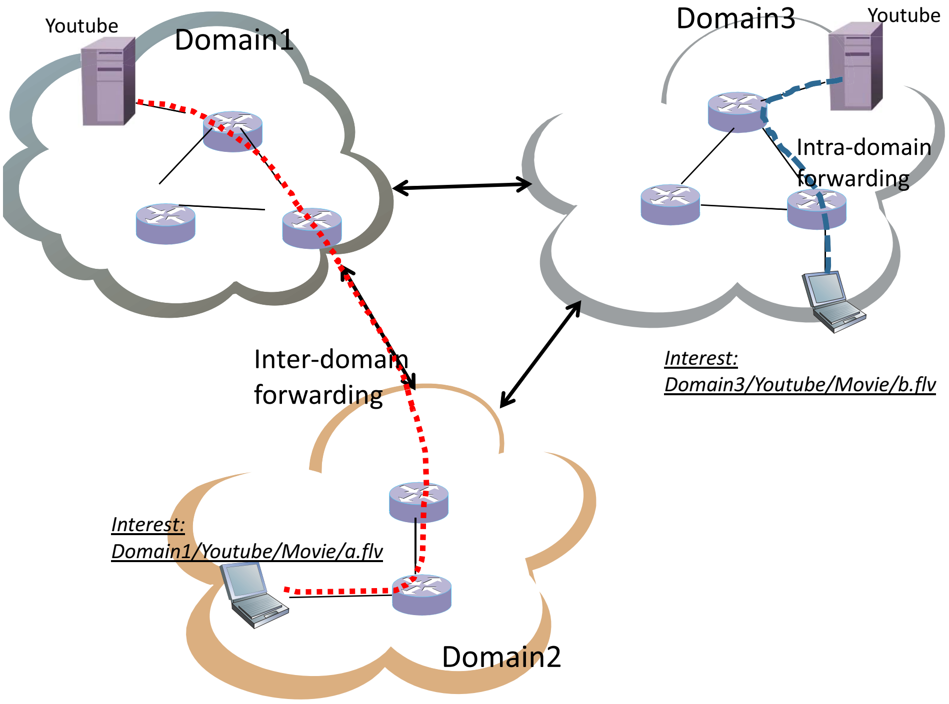

FIGURE 3.2: An example of domain based network model.

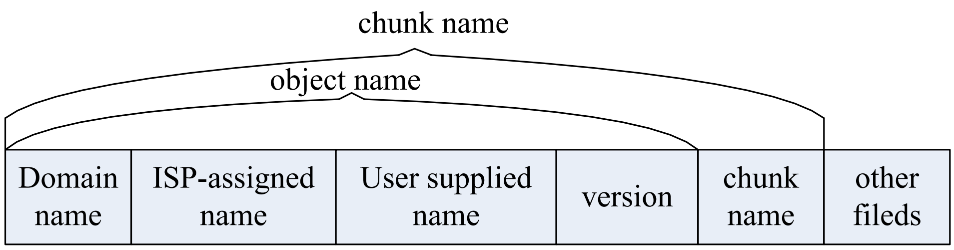

FIGURE 3.3: An example of CCN naming in DACF.

Regarding forwarding, the Interest packets should be routed to the direction pointed to the domain where the server resides in. It is because contents are spreading around the center of data repositories ideally. Thus, the closer the Interest packet is forwarded towards the server, the more possible it can be replied. Intra-domain forwarding and Inter-domain forwarding also use different strategies.

3.4.2 Overview of GDACF

In nature, ants randomly spread out from their nests in search of food. Once a food source is discovered by an ant, it returns to its nest, leaving a trail of pheromone. Nearby ants are attracted to this pheromone trail. As more and more ants use the same path, the trail of pheromone grows stronger and other ants converge onto the same path. Thus a shortest path from the nest to the food source is determined. Once the food source is used up, the path is no longer taken. No more pheromone will be left and pheromone left before are volatile. So the trail eventually vanishes and newer paths are then explored.

Inspired by the behaviors of ants, Ant Colony based approach was designed and widely used. It is very suitable for CCN forward strategy. Because this approach has the features of inherent parallelism, stochastic nature and autocatalytic in nature, which are same with the characteristics of CCN forward strategy problems analyzed before. The Ant Colony based approach uses the positive feedback to face the dynamic and complex network situations adaptively.

In these proposals, pheromone values are used to express the quality of paths. Pheromone values are defined by the function of Quality of Service metrics and updated by using revised ant colony algorithm.

To build an intelligent and adaptive forwarding plane based on the CCN routers' datagram state, a FIB component is redesigned to record pheromone values and rank faces for each content ID. There are three tables:Inter-domain Forwarding Table, Intra-domain Forwarding Table and Current Forwarding Table. The design of each table is detailed in Section 3.4.3.

In order to imitate ants, packets in the CCN are treated as ants which emerge in the pair of Interest Ant and Data Ant in these proposals. There are two types of packets, and they have different behaviors. One is Normal Ant (Normal Interest Ant and Normal Data Ant) which is generated by consumers and is used to retrieve the data; the other is Hello Ant(Hello Interest Ant and Hello Data Ant) which is generated by routers and is used to gather the routing and forwarding information [14]. The Hello Ant packet contains the information of the path load, the minimum bandwidth, the round trip delay and hops of the whole path. However, in order to reduce the packet size and drop down the router operation time, the Normal Ant packet contains less information than the Hello Ant packet does. It only contains the path overhead and the Multiple type. Fig. 3.4 represents the two kinds of packets of Data Ant. Two types of Interest Ant are similar with that of Data Ant. The details of behaviours of two kinds of packets are described in Section 3.4.6

ID	Type	Load	Timestamp	Bandwidth	Hops

(a)

ID	Type	Multi	Load	Hops	Data

(b)

FIGURE 3.4: (a) An example of Hello Data Ant packet; (b) An example of the Normal Data Ant packet

3.4.3 Node Design

In these proposals, the FIB part of CCN is redesigned. To exploit the inherent advantages of CCN and extend it by revised ACO to gather routing information, constructing three forwarding tables in redesigned FIB: Inter-domain Forwarding Table, Intra-domain Forwarding Table and Current Forwarding Table. Some parameters which will be used by GDACF algorithm are also added. In these proposals, as the first name of the hierarchical content names in the FIB, i.e., domain name, has the highest priority, FIB uses domain name to decide what kind of table, Intra-domain or Inter-domain, should be used for Interest packet forwarding.

Details about the three tables are as follows. The Intra-domain Forwarding Table consists of ISP-assigned name, associated interfaces, corresponding pheromone values, last load and the number of matched times. Fig. 3.5 shows an example of Intra-domain Forwarding Table. The Intra-domain Forwarding Table is responsible of keeping the routing information among all the nodes in the same domain. The Inter-domain Forwarding Table is similar with the Intra-domain Forwarding Table, except its first column is the domain name. It is in charge of maintaining the routing information among all the domains. Current Forwarding Table (CFT) is active items in all FIB table in CCN, and the responsibility of it is controlling overload. CFT is consist of object name, faces, remaining capacity, overhead and pheromone value. It is depicted in Fig. 3.6.

This chapter keeps original routing methods proposed in CCN and makes some extension. If there is no hit after consulting CS and PIT, an Interest packet should be forwarded based on the information of FIB. If the Interest packet is the Inter-domain packet, the node should check the Inter-domain Forwarding Table directly. Otherwise,

ISP-assigned name	**Matched**	**Interfaces**	**Load**	**Pheromone**
Youtube	n(Youtube)	A	$L_A(t)$	$\tau_{iA}^{You}(t)$
		B	$L_B(t)$	$\tau_{iB}^{You}(t)$
		C	$L_C(t)$	$\tau_{iC}^{You}(t)$
Facebook	n(Facebook)	B	$L_B(t)$	$\tau_{iB}^{Face}(t)$
		D	$L_D(t)$	$\tau_{iD}^{Face}(t)$
		F	$L_F(t)$	$\tau_{iF}^{Face}(t)$
...	...	...	...	...

FIGURE 3.5: An example of Intra-domain Forwarding Table

Object Name	**Interface**	**Capacity**	**Load**	**Pheromone**
Example.file	A	C1	$L_A(t)$	$\tau_{iA}^{You}(t)$
	B	C2	$L_B(t)$	$\tau_{iB}^{You}(t)$
	C	C3	$L_C(t)$	$\tau_{iC}^{You}(t)$
Example.video	B	C1	$L_B(t)$	$\tau_{iB}^{Face}(t)$
	D	C2	$L_D(t)$	$\tau_{iD}^{Face}(t)$
	F	C3	$L_F(t)$	$\tau_{iF}^{Face}(t)$
...	...		...	...

FIGURE 3.6: An example of Current Forwarding Table

the Interest packet is the Intra-domain packet, the node should check the Content Forwarding Table firstly. If it is still missing, then check the Intra-domain Forwarding Table.

In this proposal, every node hold a FIB control module whose responsibility is to update the FIB tables. Furthermore, each node acts independently and asynchronously in the network.

3.4.4 Energy-Efficient by Joint Caching and Forwarding

The energy consumption is mainly contributed by content caching and data transmission. Thus, energy consumption model can be expressed as $E_{tot} = E_{cache} + E_{tr}$. Since in-networking caching is beneficial in view of energy-efficiency, energy consumption models for data transmission can be essentially constructed as the function of average response hops. In this chapter, energy consumption model within t time interval prosed by [25] was used and shown as following.

$$E_{tot} = E_{cache} + E_{tr} = (P_r + P_l)H_{avg} + P_{ca}t\beta_{ca} + P_r \tag{3.1}$$

Where P_r is energy density of a router (J/bit); $P_l$is energy density of a link (J/bit); P_ca is power density of caching in content router (W/bit); E_tot is the total energy consumption; E_tr is the transmission energy consumption; E_cache is the caching energy consumption; β_{ca} is caching ratio.

As mentioned in section 3.3, these methods proposed in papers [22, 23, 25] only focus on the cache location management while neglect the power consuming by links. Therefore, the improvement of energy efficiency of this chapter in two aspects: 1) reducing average content retrieve hops; and 2) reducing power consumed by routers and links.

To reduce average content retrieve hops, Popularity Heuristic Caching(PHC) scheme was proposed. The core ideas of PHC are caching the more popular content and improving cache diversity which are used to overcome limit cache resources.

To improve cache diversity, each Normal Data packet contains a $Diversity$ field. $Diversity$ here is expressed by an integer and is initialized to 0. $Diver_{thr}$ controls the density of replica caching. When a node receives a Normal Data packet, it should check the the parameter c firstly. If the parameter $Diversity$ contained in Normal Data packet is not equal to $Diver_{thr}$, the node sets $Diversity = Diversity + 1$ and forwards it directly. Otherwise, the node sets $Diversity = 0$ and considers to cache it into CS. In the simulation section, $Diver_{thr} = 3$ is set.

The popularity is calculated by content repositories. Each repository maintains a table of requested contents which includes content name, recent requested times and popularity. Thus, popularity can be calculated by $Popu(X = i) = ReqTimes_i \div TotalReqTimes$ simply. This parameter is included in Popularity field of Normal Date packet. If the nodes want to cache and replacement a chunk of CS, it should compare the Popularity firstly and then execute the other replacement methods.

In order to save the energy of caching and reduce the power consumed by CS check, a threshold $Popu_{thr}$ was set and nodes directly discard Normal Data packets whose popularity are smaller than $Popu_{thr}$. content popularity is assumed in Zipf distributed,

$$Popu(X = i) = \frac{i^{-\alpha_{zipf}}}{\sum_{j=1}^{|F|} i^{-\alpha_{zipf}}} \tag{3.2}$$

where α is the skewness factor indicating the concentration degree of object access; i is the rank of the $i - th$ most popular file in a catalog of size $|F|$ files. Thus, $Popu_{thr}$ can be calculated by the following equation:

$$Popu_{thr} = Popu(X = \frac{\kappa \bullet C \bullet Diver_{thr}}{F}) \tag{3.3}$$

Where the C is cache size; the κ represents the multiple of popular data you want to cache. Theses proposals set $\kappa = 10$.

To reduce the power consumed by routers and links, Adaptive Rate (AR) was adopted which is on-line power adaptation, with link rates being adjusted relatively rapidly in response to changes in the traffic demand.

In a node of NDN, each $F_i \in F$ has a queue length limit on how fast Interest packets can be forwarded over an interface and experimented with a simple calculation of the Interest rate limit: $|F_i| = \epsilon \times \mathrm{C_i} \div \overline{\mathrm{S_i}}$ proposed in [11]. $|F_i|$ represents the maximum queue length of face i in node; $\mathrm{C_i}$ is the upstream link capacity of i; $\overline{\mathrm{S_i}}$ is an estimate of the size of the Data packets that have been received over i, and ϵ is a configurable parameter.

$$Rate_i = K_j \times C_i, if(K_j - 0.1)C_i \leq \frac{len_i\overline{S_i}}{C_i} \leq K_jC_i \tag{3.4}$$

$Rate_i$ represents the rate of link i; len_i expresses the pending interests length of face i; $K_j \in \{0.1, 0.2, ..., 1\}$. Thus, each node monitors the traffic flowing through length of PIT for a face and this information is used to identify traffic thresholds. When the traffic exceeds or falls below these thresholds, the link rate is adjusted to a higher or lower value accordingly, and this in turn impacts the consumed power of the device.

3.4.5 Quality of Path Measure

3.4.5.1 Quality of Service Aware Metrics

In CCN routers, the Forwarding Information Base (FIB) provides information about the face through which the content can be found. Unlike IP routers, the FIB maintains a list of faces pointing to the multiple sources that can provide the same content, which enables access to the content using multiple paths to different sources. This multipath mechanism can be exploited for improving application specific QoS.

The following introduces parameters which characterize the quality of the path. $P_{ID}(i,j)$ denotes the path from the source node i to the destination node for content name ID via the neighbor node j. The path load of the $P_{ID}(i,j)$ is defined as the average of the load of all edges and nodes in that path and can be given by:

$$L_{ID}(i,j) = \frac{\sum_{v \in P_{ID}(i,j)} l(v)}{|v|} \tag{3.5}$$

Where the $|v|$ is the number of network links along the path $P_{ID}(i,j)$. $l(v)$ is the load of node v.

The load of each node can be calculated as:

$$l(i) = \frac{N_i}{N_{iMax}} \tag{3.6}$$

Here the N_i denotes the number of the Ants in the node i (which is get from the current length of PIT), and the N_{iMax} denotes the number of the Ants in the node i when it has the full load (the maximum length of PIT).

The round trip delay of the path $P_{ID}(i,j)$ is simply the time consumed along the $P_{ID}(i,j)$:

$$D_{ID}(i,j) = t_{forwarding} - t_{receiving} \tag{3.7}$$

Where $t_{forwarding}$ is sending time and $t_{receiving}$ is receiving time.

The bottleneck bandwidth of the path $P_{ID}(i,j)$ is defined as the minimum available residual bandwidth at any link along the path:

$$B_{ID}(i,j) = min\{b(e), e \in P_{ID}(i,j)\} \tag{3.8}$$

Where e denotes a link and $b(e)$ is the bandwidth of link e.

3.4.5.2 Pheromone Definition and Update

Let $G = (V, E)$ be the graph representing the network where V are the nodes and E are links between nodes in network. $\tau_{ij}^{ID}(t)$ represents the pheromone value from the node i to the destination for content name ID via the node j. Pheromone values are normalized as following:

$$\sum_{j \in F_i^{ID}} \tau_{ij}^{ID} = 1 \tag{3.9}$$

where F_i^{ID} denotes all interfaces through which it can traverse to the node who has the content named ID.

The pheromone value for a new name (domain name or ISP-assigned name) in current time t is calculated as the normalized sum of load of path, round trip delay and bandwidth parameters given by the equation:

$$\begin{aligned}
\tau_{ij}^{ID}(t) = \alpha \frac{1}{|F_i^{ID}| - 1}&\left(1 - \frac{L_{ID}(i,j)}{\sum_{k \in F_i^{ID}} L_{ID}(i,k)}\right) \\
+\beta \frac{1}{|F_i^{ID}| - 1}&\left(1 - \frac{D_{ID}(i,j)}{\sum_{k \in F_i^{ID}} D_{ID}(i,k)}\right) \\
&+\gamma\left(\frac{B_{ID}(i,j)}{\sum_{k \in F_i^{ID}} B_{ID}(i,k)}\right)
\end{aligned} \tag{3.10}$$

Where α, β, γ satisfy the equation $\alpha + \beta + \gamma = 1$. The weighting constant α, β, γ decide the contribution of the overhead of path, round trip delay and bandwidth parameters towards the calculation of pheromone.

The pheromone update is performed according to the following update function:

$$\tau_{ij}^{ID}(t+1) = (1-\rho)\tau_{ij}^{ID}(t) + \rho \Delta \tau_{ij}^{ID}(t) \tag{3.11}$$

Where ρ represents the pheromone update rate, $\rho \in [0, 1]$. $\Delta \tau_{ij}^{ID}(t)$ is the increment of pheromone and its calculation is depicted in Section 3.4.6.

3.4.6 Forwarding Strategies

3.4.6.1 Hello Ant Packet

As introduced before, Hello Ant packets are usually generated by routers and are used to gather the routing and forwarding information in order to update the information of FIB. To improve the performance, different polices were designed for forwarding Hello Ants in the Inter-domain forwarding case and in the Intra-domain forwarding case.

In the Inter-domain forwarding progress, the Hello Interest packet is generated by a source node, usually is a router, in a fixed time interval or at the initial stage. The source node randomly chooses a domain name from the Inter-domain Forwarding Table by the roulette method. The roulette wheel is constructed by the probability of the domain name being matched. Once a domain name is selected, the Hello Interest Ant packet will be forwarded to all interfaces of the node to the center node of the destination domain.

When a middle node of the path receives a Inter-domain Hello Interest Ant, it pushes the time-stamp to time-stamp stack carried by the Hello Interest Ant and forwards the packet to one of interfaces in FIB probabilistically. The probability of the interface being selected is decided by the pheromone value and the length of the queue for that interface. If there is no information in FIB for this Hello Interest Ant, the middle node will broadcast this request and generate Hello Interest Ant for itself.

When the destination node receives a Inter-domain Hello Interest Ant, it will generate a Hello Data Ant and calculate the parameters.

When the node in the path receives a Hello Data Ant, it will update the pheromone values and rank the interfaces in the FIB by the information contained in the Hello Data Ant.

When the middle node i of the path receives a Hello Data Ant from interface j, the node updates the path load $L_{ID}(i,j)$ firstly, then calculates the $\Delta\tau_{ij}^{ID}(t)$ by the Equation 3.12 and updates $\tau_{ij}^{ID}(t+1)$ by Equation 3.11.

$$\Delta\tau_{ij}^{ID}(t) = 1 - \frac{L_{ID}(i,j)}{\sum_{k \in F_i^{ID}} L_{ID}(i,k)} \tag{3.12}$$

Then the pheromone values of other faces except j need to be updated. The $\Delta\tau_{im}^{ID}(t)$ is set to $\tau_{ij}^{ID}(t+1) - \tau_{ij}^{ID}(t)$ for calculating the evaporation function, where $m \in F_i^{ID}, m \neq j$. The evaporation function is:

$$\tau_{im}^{ID}(t+1) = \tau_{im}^{ID}(t) - \frac{\Delta\tau_{im}^{ID}(t)\tau_{im}^{ID}(t)}{\sum_{k\neq j,k\in F_i^{ID}} \tau_{ik}^{ID}(t)} \tag{3.13}$$

The evaporation function is associated with the $\Delta\tau_{ij}^{ID}(t)$ value, and it keeps the normalized pheromone values as $\sum_{j\in F_i^{ID}} \tau_{ij}^{ID}(t) = 1$.

If there is no reply for the Hello Interest Ant, the $\tau_{ij}^{ID}(t+1)$ is equated to 0 to express that the link is failure or congested at the time t+1. Then update pheromone values of other faces by Equation 3.13 with $\Delta\tau_{im}^{ID}(t) = \tau_{ij}^{ID}(t)$

When the source node i receives all the Hello Data Ants for interface j, if it is in initial stage, node i will calculate the pheromone values by Equation 3.10; otherwise, it will update the corresponding pheromone values by the following function.

$$\tau_{ij}^{ID}(t+1) = (1-\sigma)\tau_{ij}^{ID}(t) + \sigma\tau_{new}, \forall j \in F_i^{ID} \tag{3.14}$$

Where τ_{new} is calculated by equation 3.10 and σ represents the pheromone update rate and $\sigma \in [0,1]$. This progress is similar to the pheromone initial stage.

In the intra-domain forwarding progress, the intra-domain Interest Ants are only generated at the initial stage or in the packet lost case. When a middle node of the path receives a intra-domain Hello Interest Ant, it forwards the Interest Ants to the best of interfaces in the Intra-domain Forwarding Table which has the highest pheromone value and forwards one copy to another interface randomly. Other operations are some to inter-domain forwarding.

3.4.6.2 Normal Ant Packet

Normal Ant packets are drove by the consumers. When a consumer generates a request, a Normal Interest Ant is born. The forwarding policy of Normal Interest Ants is different from that of Hello Interest Ants.

In the Inter-domain forwarding case, the node forwards the Normal Interest Ant greedily to the best interface which has the highest pheromone value. In the Intra-domain forwarding case, the forwarding mode is decided by the parameter MULTI (MULTI expresses the number of paths by which the Interest is forwarded) of the Normal Interest Ant packet. The Normal Interest Ant packet will be forwarded to all interfaces only in the case that there is no forwarding information in the FIB for the content name.

The Normal Data Ant contains the load parameter. When a node receives a Normal Data Ant, it will update the pheromone values. When the node i of the path receives a Normal Data Ant from interface j, the node updates the path load $L_{ID}(i,j)$ firstly, then calculates the $\Delta\tau_{ij}^{ID}(t)$ by the Equation 3.12 and updates $\tau_{ij}^{ID}(t+1)$ by Equation 3.11.Then the pheromone values of other faces except j were updated by Equation 3.13.

3.4.6.3 Traffic Type Aware Forwarding

In these proposals, the traffic types are classified into Mobile and Non-mobile to get a favorite mobility. If the traffic type of Normal Interest Ant is the mobile, the packet are forwarded to each interface j, $j \in F_i^{ID}$, at the first hop for the mobility. The consumer may request some data but then move to a new area of base station access point. Usually, the two different base-station access points belong to the same network domain. Though the Data will arrive at the old location and be dropped, it is cached along the path. When the consumer retransmits the Interest, it will likely pull the Data from a nearby cache, making the interruption minimal [32]. Usually, it can run well in a local network.

The paper [62] classify information traffic types based on two characteristics: a)reliable vs. unreliable transfer and b) real-time vs. on-demand delivery. The combination of these two characteristics leads to three broad categories: a) channels , b) on-demand documents and c) real-time documents. Accordingly, parameter MULTI is used to represent the forwarding mode of Normal Interest Ant in the path in this chapter. The value of MULTI can be set to SINGLE, DOUBLE and ALL. It is designed to support diversity traffic type in CCN. The router of the path can decide the forwarding mode by the parameter MULTI and current network conditions.

In proposals of this chapter, real-time documents traffic has the highest priority. Interests for real-time documents will be forwarded firstly to the interface which has the minimum round trip delay. Channels has higher priority than on-demand documents. But when the CCN network is overload, channels Interest packets will be discard firstly and directly. If the router is going to discard the Interest packets of on-demand documents and real-time documents type, it will response by sending signaled data packet. The signaled data packet makes the routers and users detect the packet loss rapidly.

3.4.6.4 GDACF Approach

In GDACF, packet forwarding is QoS aware and energy efficient. the ranking of interfaces utilizes the QoS values of the path inherently. The outline of GDACF is showed in Algorithm 1. Algorithm 2 and Algorithm 3 describe Interest forwarding process and Data packet forwarding process separately.

Algorithm 1: Domain and Ant Colony based Forwarding method for CCN

Input: t := Currenttime;
t_{end}:= Time length of the simulation;
Δt := Time interval between ants generation;

1 **for** *each node i in the network* **do**
2   **while** $t < t_{end}$ **do**
3   IN-PARALLEL
4   **if** t *mod* Δt = *0* **then**
5   **if** *the node i is a router* **then**
6   Randomly select a content name ID;
7   Broadcast Hello Interest Ants for ID;
8   **end**
9   **end**
10   Forwarding(Packet P)
11   **if** P *is a Interest packet* **then**
12   InterestForward(P);
13   **end**
14   **else**
15   DataForward(P);
16   **end**
17   END IN-PARALLEL
18   **end**
19 **end**

At the initial stage or if there is no forwarding information in the FIB tables for ID, the node will generate the Hello Interest Ants, as shown in line 4 to 9 of Algorithm 1. Interest Ants are sent from each source node to all possible destination nodes in the network during the ant foraging phase (see Algorithm 2). Data Ants return along the same path with the Interest Ants but in reversed direction, as presented in 3. When the node receives all the Hello Data Ants requested by itself, they will update the corresponding pheromone values by using line 4 to line 13 of the Algorithm 3.

When a node receives a packet, it judges packet type firstly. If it is an Interest packet, it needs to make sure the forward type, Inter-domain or Intra-domain, by the

Algorithm 2: Interest Forwarding Strategy

```
   Input: Interest P
   Output: Forwarding actions
1  function InterestForward(P)
2  if the ID exists in the FIB then
3      if Intra-domain Forwarding then
4          if P is Normal Interest Ant then
5              Forward the Normal Interest ant to the interfaces js based on the MULTI,
               j ∈ F_i^ID;
6          end
7          else
8              Forward the Hello Interest Ant to the best interface j, j ∈ F_i^ID;
9          end
10     end
11     else
12         if P is Normal Interest Ant then
13             Forward the Normal Interest ant to the best interface j, j ∈ F_i^ID;
14         end
15         else
16             Forward the Hello Interest Ant to a interface j selected randomly, j ∈ F_i^ID;
17         end
18     end
19 end
20 else
21     Broadcast the Interest ants and initialize the ID;
22 end
```

domain name. In this proposal, for the Normal Interest Ants, the MULTI is set to SINGLE in the Inter-domain forwarding case and to DOUBLE in the Intra-domain forwarding case. If the traffic type of Normal Interest Ant is the mobile, the packet are forwarded to each interface j, $j \in F_i^{ID}$, at the first hop for the mobility. When the network has the ability of performing the original CCN well, it will have many advantages, e.g., mobility. For example, a consumer is watching a streaming video in a moving vehicle. If it is a data packet, the node forwards packet by PIT and updates corresponding information of FIB. The forwarding strategies are presented in Algorithm 3.

The polices of proposals of this chapter can be summaries as following:

- Intra-domain forwarding (line 3 to 10 in Algorithm 2). Hello Interest Ants are generated only when it is in the initial stage or in the packet loss case and the Hello Interest Ants are forwarded to all interfaces of the node. For the next hop selection problem, Hello Interest Ants are forwarded to the best interface of the middle nodes in the path; Normal Interest Ants will be forwarded by the parameter MULTI.

- Inter-domain forwarding (line 11 to 18 in Algorithm 2). Hello Interest Ants are generated in a fixed time interval. The Hello Interest Ants are forwarded to the each interface j, $j \in F_i^{ID}$. For the next hop selection problem, Hello Interest Ants randomly choose a interface based on the probability of interfaces being selected which is calculated by the pheromone value; while the Normal Interest Ants only select the best interface decided by the pheromone values and the current queue length of that.

Algorithm 3: Data Packet Forwarding Strategy

```
Input: Data P
Output: Forwarding actions
function DataForward(P)
if current node is not destination then
    Update interface j by the equation 3.11 and 3.12, update the other interfaces by
    equation 3.13;
    Forward the packet to the Interest coming interface;
end
else
    Wait until All possible Data packet returned;
    if it is the initial stage then
        Update the corresponding pheromone by the equation 3.10;
    end
    else
        Update the corresponding pheromone by the equation 3.14;
    end
end
```

3.4.7 Overload Control in GDACF

Although the proposed GDACF method above improves CCN network performance by selecting the best path to avoid congestion, it is not enough because that GDACF can only work well under the not full loading network (loads are supposed less than 90%). When the network is overload, no matter which path is chose, it does not work. In this case, the best choice is discarding the request. Both to ensure scalability and to preserve performance, it is necessary to implement some form of overload control.

To improve the CCN throughput, a Current Forwarding Table (CFT) which present the current status of CCN network is designed. CFT is scalable for GDACF method. In practice, content requests in progress have a mix of intrinsic rates. The analysis in [63] shows that the number of content requests that need scheduling remains less than a few hundred for any mix, even when the number of content requests in progress is counted in hundreds of thousands. The most likely mix is a very large number of low rate content requests, together counting for a large fraction of Capacity, with a handful of high rate flows dynamically sharing the residual capacity.

The parameter Capacity of CFT represents the remaining capacity of path for this content. For simple, in proposals of this chapter, the available number of request in the path was used to represent approximate capacity of path. The parameter Capacity is updated by the feedback of Data Ant packets. It is initialized by the minimum remaining buffer queue length of path. Here the buffer queue is used for each link of router, not the Content Store.

One possibility is to emulate TCP by implementing additive increase, multiplicative decrease (AIMD) congestion control. The router maintains a window W of content pending Interests (i.e., sent Interests for which the Data packet has not been received) and adjusts its size by adding $\frac{\alpha^*}{W}$ to W for each Data packet received and decreasing W by a factor β^* whenever a loss is detected. The values of $\frac{\alpha^*}{W}$ and β^* together with the round trip time between sending an Interest and receiving its Data determine how aggressively the user tracks available bandwidth, as is well-known from studies of TCP.

Signaling data packet and the timeout are used to detect packet loss. In IP, the efficiency of TCP relies on the rapid detection of packet loss through out-of-sequence packet delivery. In CCN, where flows can have multiple sources and multiple destinations, using ACK to detect loss is not possible [64]. When a router decides to discard a Interest, it will response by sending a Data packet which is signaled to discard. Current CCNx implementations use time outs to detect loss, both in the end-system and in routers that need to detect loss to remove corresponding PIT entries. It is not easy to set such time outs to a value that is small enough for timely reactions while avoiding false alerts for packets that are simply delayed. However, it is useful for node failure case.

3.5 Evaluation

In this section, the performance of DACF is evaluated by using a discrete event-driven simulator by means of chunk-level simulations. DACF forwarding scheme was implemented by extending ccnSim [57] simulator, the OMNET++ [65] based CCN simulator. The dynamic forwarding policy proposed by the NDN project was also implemented for comparison.

3.5.1 Simulation Methodology

The network topology was modeled as an graph $G(d, n, p)$, where d is the number of domains, n is the number of nodes in each domain and p is the probability that a link connecting two nodes does exist. GT-ITM [66] generates an Internet-like topology which consists of 1 transit domain, 4 stub domains, which is illustrated in Fig. 3.7. Domain 0 is the transit domain. Links are characterized by their propagation delay and bandwidth. The link bandwidth of each stub domain is set to 100 Mbps and the link bandwidth of transit domain is set to 200Mbps. Link propagation delays range from 1 to 5 ms. Each stub domain has 3 client users which are connected to its border nodes. There are two repositories which are connected to Domain 2 and Domain 4 respectively. Two repositories store the same content. Thus, the network traffic from Domain 2 to Repository 1 or from Domain 4 to Repository 2 is intra-domain traffic; The remaining traffic is inter-domain traffic.

In order to assess the performance of the DACF schemes, their performances are also compared with the following schemes:

- DACF: A node forwards the Interests by DACF method;
- Shortest: A node forwards the Interests by the shortest path algorithm;
- CCN: A node forwards the Interests by the stateful forwarding plane proposed by the NDN project .

The analysis of the performance of the DACF method focus on two specific metrics: 1) *Data retrieve time*, which represents the time elapsing in retrieving one unit data; 2) *network throughput*, which is defined as the network successfully delivered data in one time unit. The first metric is calculated as the file retrieving time divided by the file size (the number of chunks). It allows quantifying the performance in user perspective.

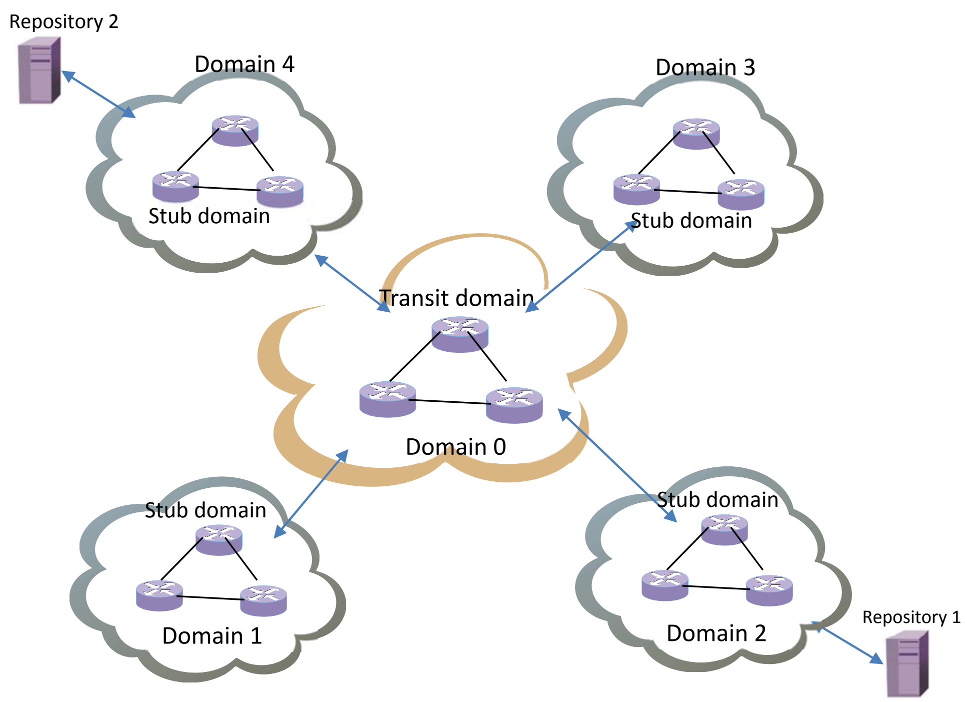

FIGURE 3.7: Simulation Topology

The second metric, which is used to assess the performance in the network perspective, is expressed by the average amount of chunks delivered successfully during the one unit simulation time.

In experiments of this chapter, nodes are mainly characterized by their cache size, cache decision policy and cache replacement policy. Every node is equipped with a cache of size C. Cache decision policies include: 1) ALWAYS: new content is always cached; 2) FIX (P): caching decisions are taken uniformly at random with a fixed probability P. Here, FIX(0.8) cache decision policy was only considered because it achieves the simplest possible level of uncoordinated and distributed cache diversification. Cache replacement policies have been long studied, e.g., Least Recently Used(LRU), First In First Out and UNIF (a chunk selected uniformly at random is replaced). LRU (the least recently used chuck is replaced) is used as cache replacement policy [38].

In the network of this chapter, a large YouTube-like catalog is assumed consisting of $|F| = 10^8$ files. Each single session is characterised by the number of requested packets(File size)and packet size(Chunk size). File size F is geometrically distributed with average 10^3 chunks and chunk size is set to 10KB. The size of content is 10 MBytes which is same to the average size of videos in YouTube [67]. Considering realistic CCN cache size and Internet catalog, the ratio $\frac{C}{|F|F}$ keep ranging from 10^{-5} to 10^{-1}. Unless otherwise specified, $C = 10GB$. The Mandelbrot-Zipf distribution model calculates the content popularity, where $\alpha = 1.5$ and $q = 0$. Users perform File-level requests according to a Poisson process with exponentially distributed arrival times at a λHz rate. In the DACF algorithm, α=0.2, β=0.6, γ=0.2, ρ=0.2 and σ=0.5. Simulation results are averaged over 10 simulation runs.

The examination of a network's packet delivery performance is under three scenarios: (1) normal traffic, in which the packet lost rate is under the threshold of 5% for shortest path routing method; (2) congestion, in which packet loss rate is higher than the threshold of 5% for shortest path routing method and some links do not have enough bandwidth to carry the network traffic; (3) link failure, in which links randomly fail by certain probability and then recover after a certain time. For each simulation cycle, a simulator is set to complete a mission of transporting a certain amount of files. A simulator stops generating *Interests* when a certain mount of files are requested, expressed in the number of chunks. Unless otherwise specified, topology is with $G(d, n, p) = G(5, 10, 0.3)$ and the YouTube like scenario with $(F, |F|, C, \alpha, q) = (10^3, 10^8, 10^7, 1.2, 0)$.

3.5.2 Normal Traffic Load

In this section, the performance comparisons of three methods use the *Data retrieve time* metric in the normal traffic scenario, where the topology is $G(d, n, p) = G(5, 10, 0.3)$ and file-level request rate is $\lambda = 1Hz$ for each user.

Firstly, the suitability of DACF is evaluated under varying conditions of cache sizes or content popularity distribution skewness. Fig. 3.8 depicts the *Data retrieve time* as function of content popularity skewness α with cache size $C = 0.01\%$. Content popularity distribution skewness is represented by Mandelbrot-Zipf distribution parameter α, ranging from 0.7 to 1.5. It can be seen that data retrieve time decreases as the content popularity distribution skewness *alpha* increases, especially when *alpha* more than 1.0, there is a sharply decline. As the *alpha* of content popularity distribution skewness increases, the number of *Interests* for popular data increases. Thus, there are more *Interests* can be satisfied by the intermediate nodes of path leading to the server, with a consequence decrease in the data retrieve time. As it can be seen from this graph DACF achieves the greatest performance for all values of *alpha* considered.

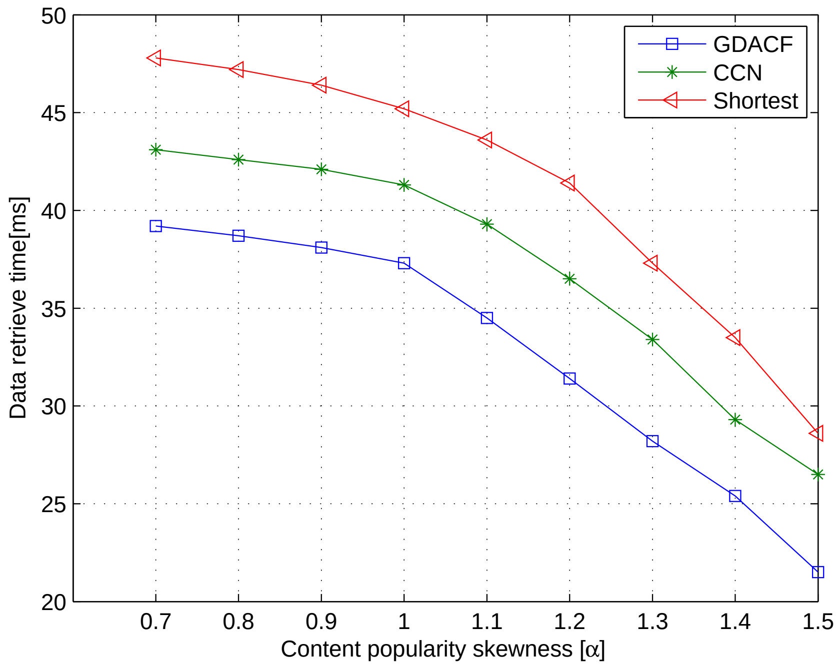

FIGURE 3.8: f
or DACF]Data retrieve time as function of content popularity skewness $[\alpha]$

Fig. 3.9 illustrates the sensitivity of *Data retrieve time* against the variations of cache over catalog size with content popularity distribution skewness $\alpha = 1.2$. The simulation varies the ratio of the cache over catalog size and keeps a single cache can hold between 10^{-1} and 10^{-5} of the whole content. It is showed that with the cache size increases, data retrieve time sharply decreases. The more the popular data can be cached in the intermediate nodes of network, the shorter path users can use to get data. When the cache size is up to $C = 1\%$, data retrieve time declining trend are becoming smaller until no change. When cache size is more than $C = 10\%$, data retrieve time of DACF has a slight decline but the other two methods have no change. It can be observed that DACF provides smallest data retrieve time among three algorithms for all cache size values. Specifically, it provides a performance improvement between 18-33% with respect to shortest path forwarding, and between 10-33% with respect to the CCN forwarding strategy.

Secondly, the analysis of the scalability of DACF is under varying conditions of topology sizes (the number of nodes) or varying conditions of topology connectivity, where cache sizes $C = 0.01\%$, content popularity distribution skewness $\alpha = 1.2$ and request rate $\lambda = 1Hz$.

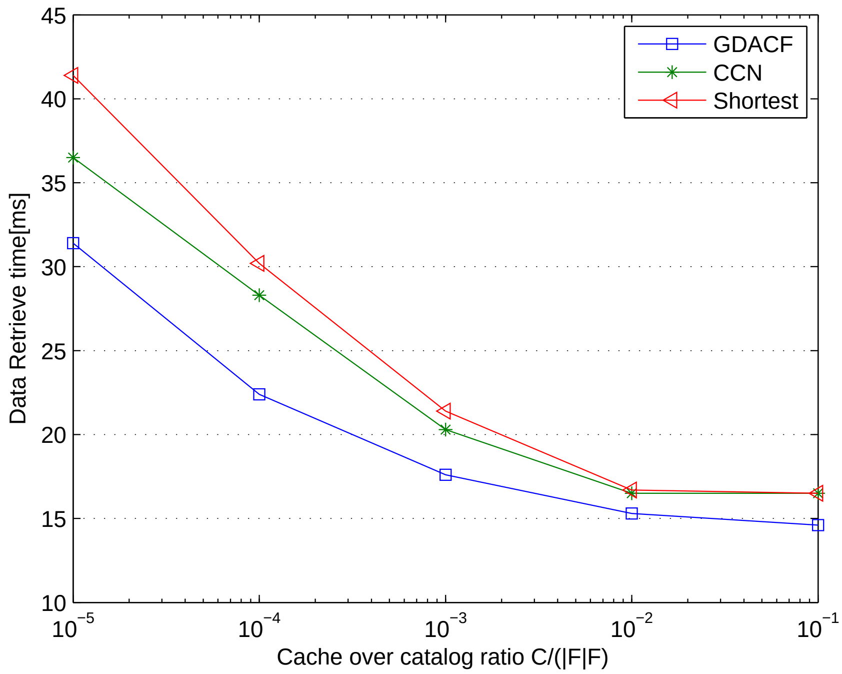

FIGURE 3.9: Date retrieve time as function of cache over catalog ratio

Fig. 3.10 shows the *Data retrieve time* as the function of connectivity p with the number of nodes in each domain $n = 10$. Connectivity p of topology represents the number of paths between two nodes. As expected, data retrieve time decreases as the network connectivity increases for all three algorithms. As the network connectivity increases, users can get data by the shorter paths or more optional multiple paths except the shortest paths. It can be observed from figure that DACF provides the greatest performance for all values of network connectivity considered. The result also shows that DACF can better utilize an increasing number of paths because the performance gap increases with the connectivity between DACF and other algorithms.

Fig. 3.11 reports the sensitivity of the *Data retrieve time* against variations of the number of nodes in each domain with the network connectivity $p = 0.3$. The number of nodes in each domain denotes the size of network topology. It shows that data retrieve time increases as the domain size increases for all three methods. When the domain size increases, the length of paths from clients to repositories increase. It also shows that DACF outperforms other algorithms for all domain sizes. Specifically, DACF outperforms 18-33% than shortest path forwarding, and 10-33% than the CCN forwarding strategy.

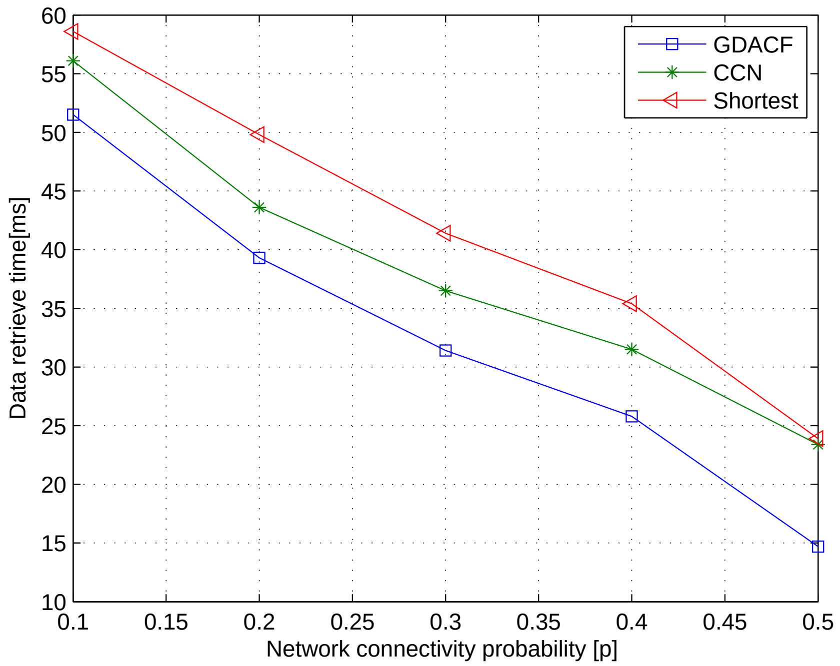

FIGURE 3.10: Data retrieve time as function of connectivity probability

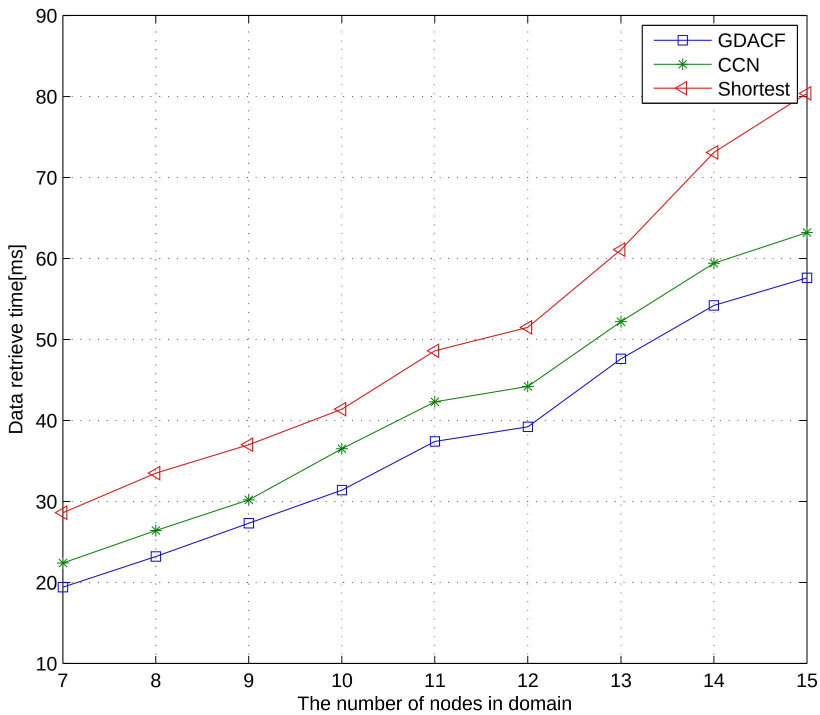

FIGURE 3.11: Date Retrieve Time as function of the number of nodes in each domain

3.5.3 Energy Efficiency

In this section, the energy saving of these proposals for normal traffic load is studied, where topology is set with $G(d,n,p) = G(5,10,0.3)$ and file-level request rate is $\lambda = 1Hz$ for each user. The devices' parameters are referred to [25], energy density of a router is in the order of $P_r = 2 \times 10^{-8}$ Jbit. Energy density of a link is $P_l = 0.15 \times 10^{-8}$ Jbit, power density of caching in content router is $P_ca = 10^{-9}$Wbit. β_{ca} is the percentage of the caching capacity to the total amount of requesting contents. $P(Shortest)/P(IP)$, $P(Shortest)/P(IP)$ and $P(CCN)/P(IP)$ represent the energy saving. $P(IP)$ is measured in same scenario but no in-networking caching.

Fig. 3.12 depicts the Power saving as function of content popularity skewness α with cache size $C = 0.01\%$. α ranges from 0.7 to 1.5. It can be seen that power saving $P/P(IP)$ decreases as the content popularity distribution skewness α increases. Specifically, GDACF saved power between 16.99% - 19.59% than shortest path forwarding, and between 11.92% - 14.46% than the CCN forwarding strategy.

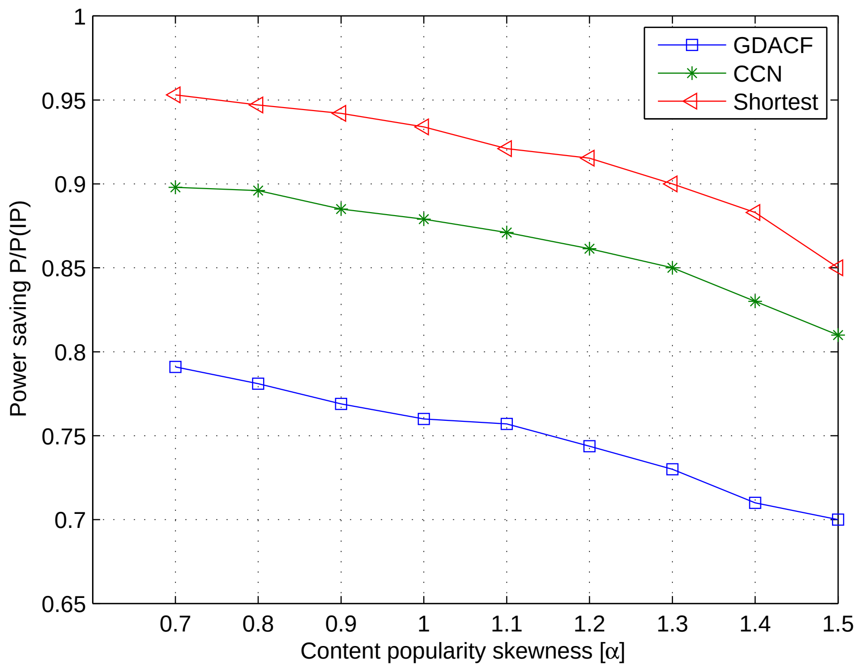

FIGURE 3.12: Power saving as function of content popularity skewness $[\alpha]$.

Fig. 3.13 illustrates the sensitivity of power saving against the variations of cache over catalog size with content popularity distribution skewness $\alpha = 1.2$. The results show that with the cache size increases, power saving $P/P(IP)$ firstly decreases, then increases when cache size is bigger than 1%. Specifically, GDACF saved power between

11.97% - 20.03% than shortest path forwarding, and between 9.66% - 15.08% than the CCN forwarding strategy.

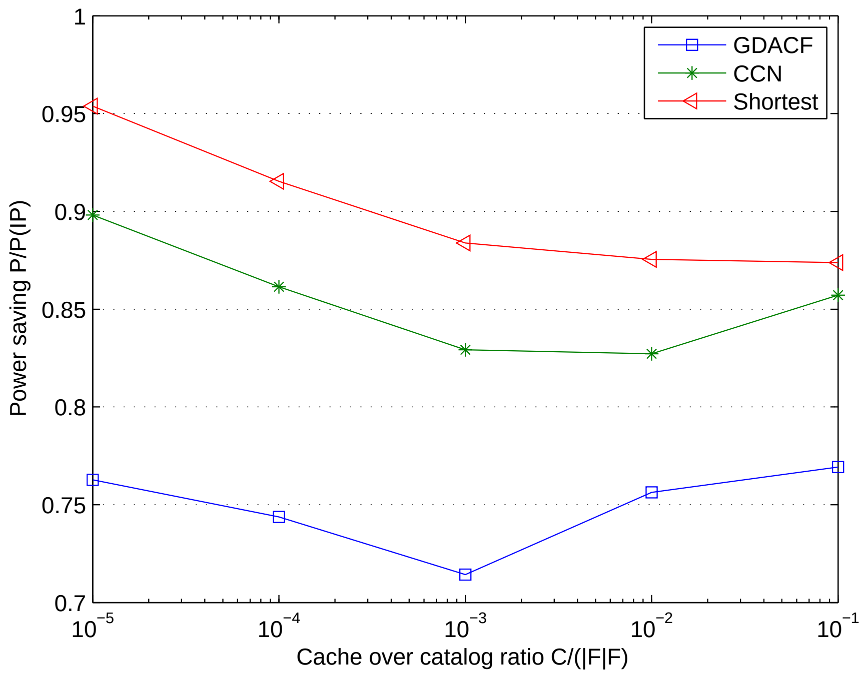

FIGURE 3.13: Power saving as function of cache over catalog ratio.

3.5.4 Congestion

In this section, the performance of the algorithms for increasing traffic load, examining the network status were studied in congestion scenario. Traffic is defined in terms of open sessions between pairs of different nodes. Each single session is characterized by the number of transmitted packets, and by their size and inter-arrival time distributions. The file-level users request rate λ is used to model the network traffic load level. In this scenario, the network topology is also set with $G(d, n, p) = G(5, 10, 0.3)$ and the YouTube like contents distribution is set with $(F, |F|, C, \alpha, q) = (10^3, 10^8, 10^7, 1.2, 0)$. Each router has buffer size(not cache size) of 50 chunks for each link. Interest packet size is set to 100 bytes. Users can request different contents simultaneously and request rate is decided by parameter λ. But for a content, at most one chunk request can be sent by any user before receiving the corresponding data.

Fig. 3.14 shows *Data retrieve time* as the function of of users request rate λ, which decides the traffic load level. Clearly, data retrieve time increases as the users request rate increases for all methods. It can be seen that under low network traffic load conditions, three methods have similar performance. The shortest paths almost have enough bandwidth to carry the offered network traffic. As the network traffic load increases, DACF slightly outperforms the other two algorithms. However, when the network traffic is high, the DACF has dramatically better performance than the other two algorithms. This is due to the fact that DACF forwards Interests by multiple available paths. When the users request rate *lamda* is greater than 2.0, data retrieve time of DACF methods also sharply rises like the other two methods. Specifically, it provides a performance improvement between 18-33% with respect to shortest path forwarding, and between 10-33% with respect to the CCN forwarding strategy.

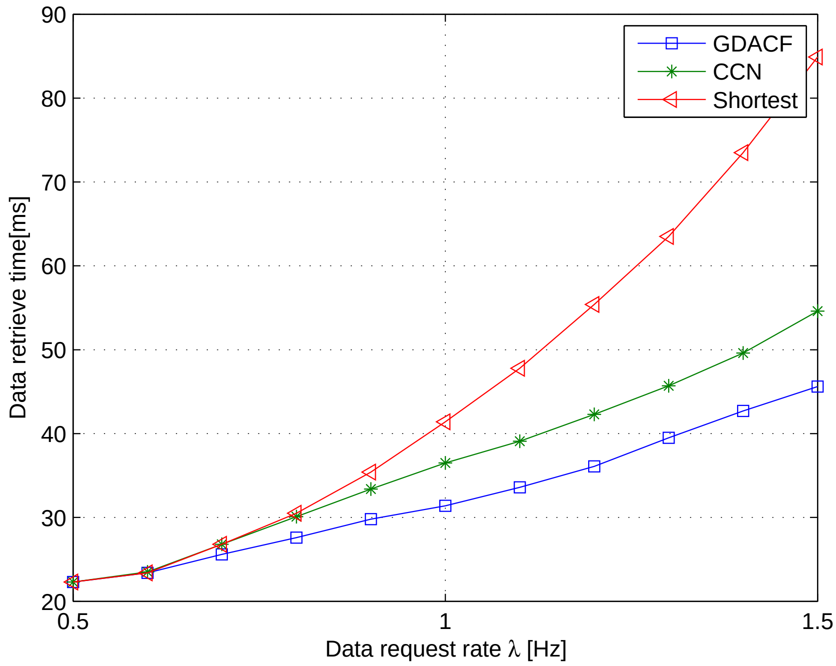

FIGURE 3.14: Date retrieve time as function of data request rate λ

Fig. 3.15 reports the performance of network throughput against varying values of users request rate. As expected, network throughput increases as the users request rate λ increases for three methods. DACF has a slight higher throughput than CCN method; DACF and CCN methods outperform shortest path forwarding method; Specially, the throughput of shortest path forwarding method almost dose not change when the users request rate λ increases up to 1.4 Hz.

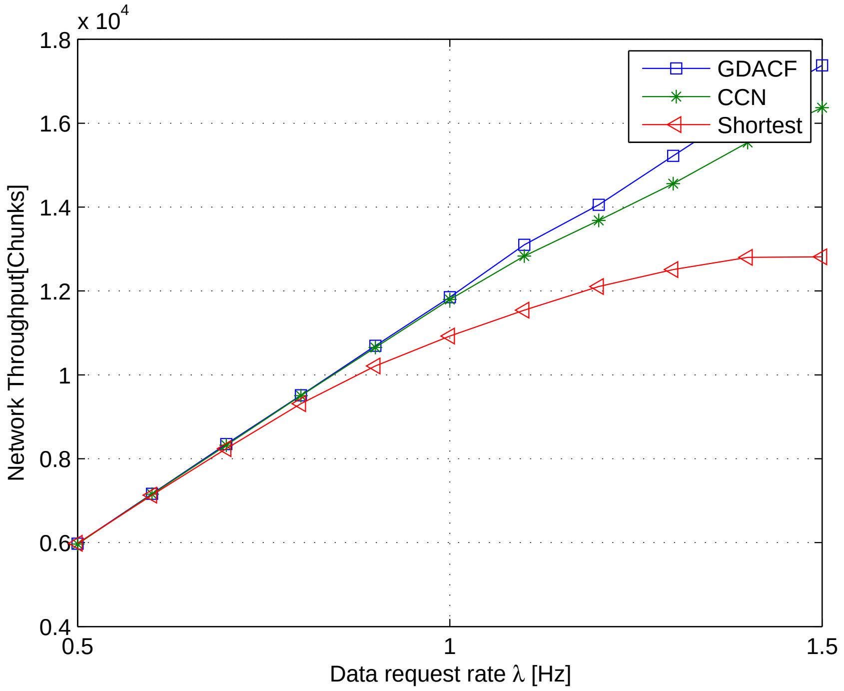

FIGURE 3.15: Network throughput as function of data request rate λ

3.5.5 Link Failure

To evaluate the robustness of DACF, the performance of the tree methods in the link failure scenario was simulated. Robustness is defined as the ability of a forwarding algorithm to continue operating despite some uncertain errors, such like link failure, node failure, topology change, etc. Firstly, the generation of the link failure scenarios is as following. Each link in the network has a uniform failure probability and randomly fails according to this probability to generate one link scenario. When the simulation finishes initial stage and becomes stable, links failed in previous scenario are set to be unavailable and all packets sent over these links are dropped. Simulations of three methods run over 100 randomly generated failure scenarios for one link failure probability.

Fig. 3.16 describes the instantaneous throughput as the function of simulation time with 5% link failure probability. Instantaneous throughput is the amount of chunks forwarded successfully for a given time slot t during an algorithm simulation. When the simulation time is up to 20s, some links are set to be unavailable based on the previous generated link failure scenarios. It is observed that three algorithms have different degrees of decline in instantaneous throughput; DACF has quicker recovering time from link failure than CCN method; DACF also provides a slight higher throughput than

CCN method after link failure. Shortest path algorithm has the worst performance. The robustness of DACF can be observed with the relationship of CCN and Shortest path algorithms.

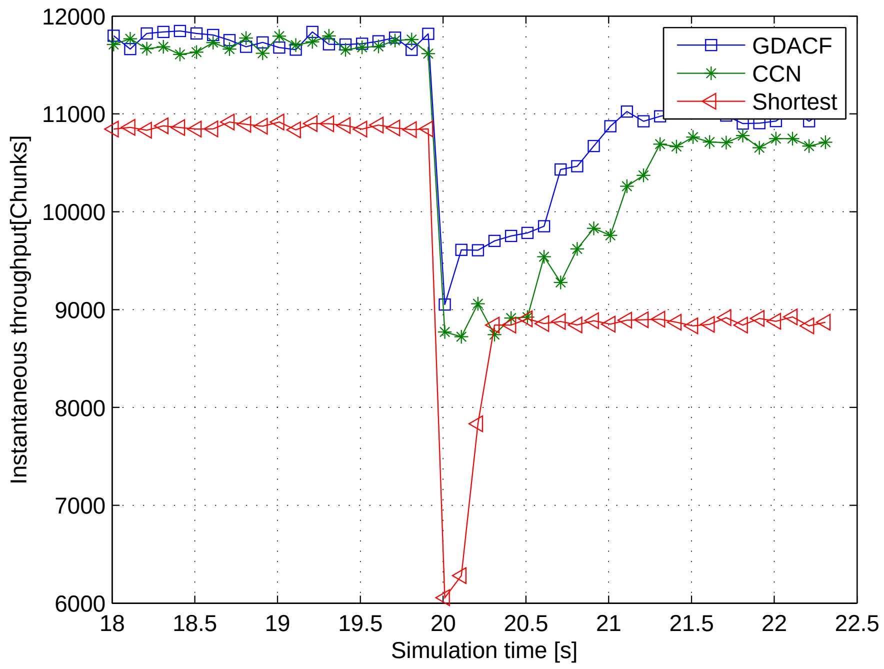

Figure 3.16: Network instantaneous throughput as function of simulation time with 5% link failure probability

Chapter 4

A Game Theoretical Interest Forwarding for Cached Data in CCN

4.1 Preface

Content-Centric Networking (CCN), decoupling content from hosts at the network layer, has several attractive advantages, such as network load reduction, low dissemination latency and energy efficiency. However, FIB table of routers cannot contain all the content as the huge naming space; and as the content cached in Content Store of routers is changing frequently, it is very difficult to update the FIB in time for all content objects in the network. Thus, it is a challenge that how to efficiently utilize the cached data.

This Chapter presents a game theoretical Interest multiple forwarding decisions for cached data method to maximize the users's payoff and network's payoff. Game players are Interest flows. Actions of a player are the number of forwarded replica Interests. Payoff function is defined to express the behaviors of players. Network uses the Potential Heuristic Allocation method to achieve global benefits of network.

The main objectives of this chapter include: 1) fully utilizing rich connectivity to get the best users' payoff; 2) Against route hijacking attacks (if no data returns over a particular interface for a particular name, that interface may not lead to a valid path for that name); Enhancing the network instability (frequent oscillation of paths) while maintaining good data delivery performance.

The rest of this chapter is organized as follows. Technical background of Game theory is given in Section 4.2. Section 4.3 presents the non-cooperative game analysis for Interest multiple forwarding problems. Potential Heuristic Allocation policy is described in Section 4.4. Section 4.5 presents simulation setting and simulation results.

4.2 Game Theory

John von Neumann and Oskar Morgenstern established game theory as a separate field of science when they published their book [68] in 1944. Since then great strides have been made in this area, mainly in the field of economics and biology. However, game theory can also be applied to many fields of science, where decision makers have conflicting interests. Thus, it comes as no surprise to read papers related to networking that adopt game theoretical concepts to analyze a protocol's performance or propose a solution that corresponds to a Nash Equilibrium (NE) set of strategies [69][70].

Game theory could be defined as "the study of mathematical models of conflict and cooperation between intelligent rational decision makers" [71].

A game consists of a principal and a finite set of players $N = \{1, 2, ..., N\}$, each of which selects a strategy $x_i \in X_i$ with the objective of maximizing his utility u_i. The utility function$u_i(s) : X \longrightarrow R$ represents each player's sensitivity to everyone's actions. People or entities (decision makers in general) that play the game are called the players.

A strategy for a player is a complete plan of actions in all possible situations in the game. The players try to act selfishly to maximize their consequences according to their preferences. The set of player i's possible actions is called the action space X_i of player i.

Two types of games are distinguished: one is non-cooperative games in which each player selects strategies without coordination with others. The other is cooperative games in which the players cooperatively try to come to an agreement, and the players have a choice to bargain with each other so that they can gain maximum benefit, which is higher than what they could have obtained by playing the game without cooperation.

In a static game, the players make their decisions simultaneously at the beginning of the game. In a dynamic or sequential game, the players interact with each other, and they do not decide simultaneously, but they follow a sequence. If the interactions are repeated in time, the game is called repeated , and each interaction corresponds to a stage of the game. In this case, the players have the opportunity to modify their strategies over time.

The equilibrium strategies are chosen by the players in order to maximize their individual payoffs. In game theory, the Nash Equilibrium is a solution concept of a game involving two or more players, in which no player has anything to gain by changing only his own strategy unilaterally. If each player has chosen a strategy and no player can benefit by changing his strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash Equilibrium.

One of the papers that applied game theory to the problem of routing was [72]. They consider a communication network shared by several selfish users. Each user seeks to optimize its own performance by controlling the routing of its given flow demand, giving rise to a non-cooperative game. For a two-node multiple links system, uniqueness of the Nash Equilibrium is proven under reasonable convexity conditions.

Based on the above models for the general network, Altman et al. in [73] provided the necessary conditions in order for the NE to be unique and make the polynomial cost structure attractive for traffic regulation and link pricing in telecommunication networks. They considered a class of polynomial link cost functions adopted originally in the context of road traffic modeling, and showed that these costs have appealed properties that lead to predictable and efficient network flows.

In contrast to previous works, authors in [74] considered the cost function in a multiplicative way and assumed that the cost function is an additive combination of the objectives of routing, namely the maximization of throughput and the reduction of the delay.

4.3 Design

4.3.1 Problem Description

Forwarding strategy is a key component of NDN nodes that makes them more powerful than their IP counterparts. Routing of IP network is to calculate a single shortest path for each pair of source node and destination node. CCN inherently supports multiple same Interests forwarding simultaneously. The forwarding strategy layer in a NDN node can dynamically select multiple interfaces from the Forwarding Information Base (FIB) to forward each Interest packet.

The simplest strategy is to send an Interest on each of the interfaces in a FIB entry in sequence. If there is no response to the Interest, then try the next interface. Single shortest path can be a candidate forwarding strategy for CCN. However, it cannot perform well as it runs in end to end communication network. In end to end communication network, the destination node is definite, but in CCN, a content object can have many destination nodes (it can be a router) by the form of replicas. Thus, sometimes the shortest path record in FIB is not real shortest path for a content object. It is very difficult to update the FIB in time for all content objects in the network because of the huge content name space, especially in chunk level.

Another forwarding strategy is sending Interests on all the interfaces at once and see which interfaces receive data first. These interfaces will be used for a period of time and their performances are monitored. If the routers do it for all the Interest packets, it can make the network overload and congested easily.

A more flexible design is each FIB entry containing a program specialized to make Interest multiple forwarding decisions. This chapter presents a Game theoretical Interest Multiple Forwarding (GIMF) decisions method to solve this problem. The goal of this proposals is fully utilizing the residual capacity in the network so that users can get the maximum payoff in a definite network situation.

4.3.2 Gaming Analysis

The hierarchical CCN naming convention described previously lends itself to the identification of flows. A CCN flow consists of packets bearing the same object name [64]. In a node of CCN, a set of flows I share a set of parallel paths represented by faces F. Each $F_i \in F$ has a queue length limit on how fast Interest packets can be forwarded over an interface and experimented with a simple calculation of the Interest rate limit: $|F_i| = \alpha \times \mathrm{C_i} \div \overline{\mathrm{S_i}}$ proposed in [11]. $|F_i|$ represents the maximum queue length of face i in node; $\mathrm{C_i}$ is the upstream link capacity of i; $\overline{\mathrm{S_i}}$ is an estimate of the size of the Data packets that have been received over i, and α is a configurable parameter. Here X (Equation 4.1) is defined as the total queue length (total available resources) in the node and X_0 as the queue utilization caused by all background traffics. $|F|$ denotes the number of total faces in node.

$$\mathrm{X} = \sum_{i=1}^{i \leq |F|} |F_i| \tag{4.1}$$

Each $I_i \in I$ aims to minimize the individual cost and maximize the utilization selfishly by deciding the multiple forwarding degree x_i. F^i is the set of faces for Interest

I , through which the Interest $I_i \in I$ can reach the repository nodes with H hops. F^i can be got from the FIB table of node. The multiple forwarding decisions problem models as I_i selecting the subset $f^i \subset F^i, f^i \neq \emptyset$ to get the best cost. The network model is described in Figure 4.1.

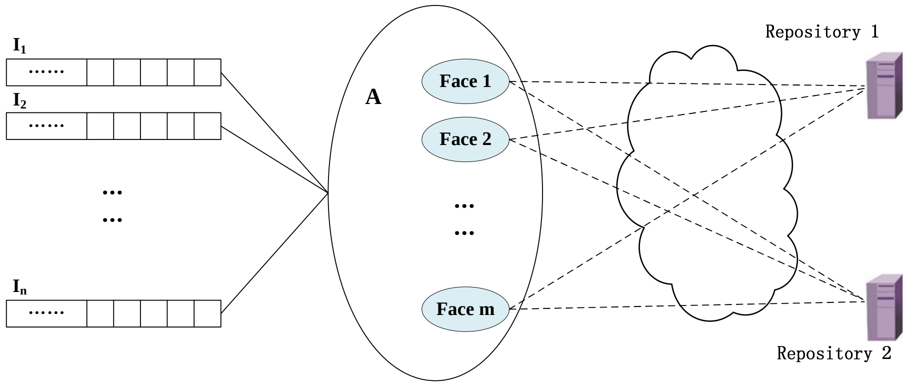

FIGURE 4.1: Network model.

In this model, the game players are considered as flow I. Set the player i using the node resources as $x_i \in X_i$. X_i is a collection of node resources may be occupied by player i. X_i is strategic space of player i. When only discuss interfaces without considering other types of node resources, x_i is the multiple interfaces f^i used by player i. This model defines the $x_i = \left\{x_i | 1 \leq x_i \leq |f^i|\right\}$ simply. For an Interest of I flow, the f^i can be got from FIB table in CCN. In FIB table, the interfaces are sorted by the hops which present the distance from the node to repository. The face for an Interest with minimum hops has the highest priority to be selected. The Interest forwarded by the shortest path is called *main Interest*, correspondingly, the Interests forwarded by longer path are called *replica Interest* in this chapter. Here, this model considers the strategic space as continuously divisible to guarantee the Nash Equilibrium existence.

4.3.2.1 Payoff function

The Payoff Function of player i specifies the total gains of player i when it takes action x_i, which is a kind of variable showing the worth achieved by players using the node resources.

The general form of Payoff Function consists of two parts: $Payoff = Benefit - Cost$ [74]. Thus, the payoff function U_i of player i is defined as Equation 4.2:

$$U_i\left(x_i, X_{-i}\right) = Benefit_i\left(x_i, X_{-i}\right) - Cost_i\left(x_i, X_{-i}\right) \tag{4.2}$$

Here, x_i denotes the multiple Interest forwarding degree. When there are no external controls, utility function stipulates the gain of player i when it takes action x_i. Due to the related form of action in the game, the utility function of player i is not just the function of x_i, but also is the function of other players.

Denote $\mathrm{X} = (x_1, \cdots, x_i, \cdots, x_n)$ as the vector constituted by all the players' actions, and $\mathrm{X}_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$ as the vector constituted by other players' actions except user i. Then the utility obtained by player i is $U_i(x_i, \mathrm{X}_{-i})$, can also be abbreviated as $u_i(\mathrm{X})$. The utility function of player i is the mapping from the set of action X_i to the set of real number R^1, $u : X_i \to R^1$, which defines the preferences of players on the set of actions. For all $x, y \in X_i$, if and only if $U(x) > U(y)$, players prefer the action x than the action y.

Here this proposals assume that $|F^I|$ is continuously divisible, and can be represented by a real number. Strategic space X_i is the real axis of a non-empty closed space and is a non-empty compact convex set. It is used to guarantee the Nash Equilibrium existence. Actually, the players do the action by the rounding of x_i in the simulation section.

The Benefit function of player i is defined as:

$$Benefit(x_i, X_{-i}) = t_m \times P_i \times (x_i - 1) \tag{4.3}$$

Here, the t_m denotes unit time gain for player i doing the action of sending replica Interest; P_i represents the probability of a replica Interest retrieving a cached data faster than main Interest. The purpose of players who send replica Interests is to more stably retrieve the data faster. I use $t_m \times P_i \times (x_i - 1)$ to denote the estimated benefits for players who send $x_i - 1$ replica Interests.

Cost function specifies the punishment given to players from the network when player i takes action x_i . The Cost function is defined as:

$$Cost(x_i, X_{-i}) = t_q \times \frac{x_i}{\sum x} \times e^{(\sum x + X_0)/X - 1} \tag{4.4}$$

Where the t_q denotes unit time cost for queuing because of player i doing the action of sending replica Interest; In this expression, the deterministic term $1/X - (\sum x + X_0)$ represents the expected congestion delay on a link for an M/M/1 delay function [75]. I use $e^{(\sum x + X_0)/X - 1}$ to express the normalized queuing time factor and adopt $\frac{x_i}{\sum x}$ to present the proportion of queuing time for player i.

From the network's perspective, the nodes adopt some mechanism to transport packets efficiently and fairly. Usually the nodes use Max-Min fair queue to implement transmission fairly. I also proposed a Potential Heuristic queue method to consider efficiency and fairness in Section 4.4.

Thus, the utility function can be described as following:

$$U_i\left(x_i, X_{-i}\right) = t_m \times P_i \times (x_i - 1) - t_q \times \frac{x_i}{\sum x} \times e^{(\sum x + X_0)/X - 1} \tag{4.5}$$

$U_i$is a increasing function of x_i and it is diminishing marginal returns. A higher x_i does not necessarily yield better performance for player i. On the condition of $\sum x \leq X$, user can get an optimal $\overline{U}_i$ to meet $\frac{\partial \overline{U}_i}{\partial x_i} = 0$ through adjusting x_i. The unilaterally optimizing behaviors of user i meet:

$$\frac{\partial Benefit_i}{\partial x_i} = \frac{\partial Cost_i}{\partial m_i} \tag{4.6}$$

Here, this method assumes the amount available resources of node is X. The resources allocation accords player's need. This model can adopt a simple resource allocation method which is denoted as following:

$$x_i = \frac{|F^i|}{\sum_{j=1}^{n} |F^j|} X \tag{4.7}$$

4.3.2.2 Nash Equilibria

A NE is a set of strategies where each player has no incentive to deviate, in other words, given the strategies of all other players, if he changes his strategy he can only decrease his utility. More specifically, if x_i is an arbitrary action of player i and X_{-i} is the set of actions of all other players, then the action profile $x^* = (x_i^*, X_{-i}^*)$ constitutes a NE if for every player i, $U_i(x_i^*, X_{-i}^*) \geq U_i(x_i, X_{-i}^*)$, $\forall\ x_i \in X_i$. The action vector $x^* = (x_1^*, \cdots, x_n^*)$ is set as Nash Equilibrium, then outcome can be got : $U_i\left(x_i^*, x_{-i}^*\right) \geq U_i\left(x_i^*, x_{-i}^*\right), \forall x_i \in X_i, \forall i \in [1, n]$.

The existence of the Nash Equilibrium [71] is constrained as following: In game $G = [n, \{x_i\}, \{u_i(\cdot)\}]$, the necessary and sufficient conditions of the existence of the Nash Equilibrium is: for all $i = 1, 2, \cdots, n$, there is: *i)* X_i is a non-empty, compact convex set on Euclidean space; *ii)* $U_i(x)$ is continuous in the x, and is quasi-concave function of x_i.

The optimal payoff of player i is recorded as $\tilde{U_i}$. $\tilde{U_i}$ can be assumed as increasing functions of x_i (Allocated more interfaces, get the greater utility), and meet diminishing marginal returns(The the speed of utility increasing reduces with the increase of the forwarding degree x_i):

$$\frac{\partial \tilde{U_i}\left(x_i\right)}{\partial x_i} > 0, \frac{\partial^2 \tilde{U_i}\left(x_i\right)}{\partial {x_i}^2} < 0 \tag{4.8}$$

$$Max \sum_{i=1}^{n} \tilde{U_i}\left(x_i\right) s.t. \sum_{i=1}^{n} x_i \leq X \tag{4.9}$$

The solution of this model can be represented as Equation 4.9 to solve the maximum value of payoff of all players. Using Lagrange Method of Multiplier for solving, suppose a Lagrangian function $L\left(x_1, x_2, \cdots, x_n\right)$ where exits:

$$L = \sum_{i=1}^{n} \tilde{U_i}\left(x_i\right) + \lambda \left(X - \sum_{i=1}^{n} x_i\right) \tag{4.10}$$

In which λ is a specific unknown constant. The optimal solution should satisfy the condition that the partial derivatives that L for all unknowns is 0:

$$\frac{\partial L}{\partial x_i^*} = \frac{d\tilde{U_i}}{dx_i^*} - \lambda = 0, i = 1, 2, \cdots, n \tag{4.11}$$

That is:

$$\frac{d\tilde{U_i}}{dx_1^*} = \cdots = \frac{d\tilde{U_i}}{dx_i^*} = \cdots = \frac{d\tilde{U_n}}{dx_n^*} \tag{4.12}$$

From Equation 4.5, it can be got that the utility function is concave function. Thus, Equation 4.12 has unique solution. This solution is the best Interest forwarding decisions.

4.4 Potential Heuristic Allocation for System

In proposed model, when the node receives a set of Interest flow I with corresponding multiple forwarding decision x_i, how to allocate the queue resources for each player $I_i \in I$ is a key issue. The allocation according to user's need and fairness allocation method are not the best method because that they do not consider the system utility.

Usually, there is no global objective function of networking outcome in proposed model or other similar models. In order to improve the efficiency of whole networking, Potential Heuristic Allocation (PHA) method is proposed for this model. The global objectives of networking are defined as 1) considering fairness of each player, 2)maximizing the player's utility and 3) improving the global networking cache hit rate.

The key idea of PHA method is that the Interest i with more potential hit has higher priority to allocate resource. For this purpose, this model redesigns the FIB table to record some metrics used to calculate the potential values. It adds a column into FIB table named *hits* which represents the number of hits for a Content ID by interface f_j^{ID}. An example of FIB table is illustrated in Figure 4.2.

In this model, k_j^i denotes the hits of player i through the interface j. The corresponding potential value ρ_j^i is defined as following:

$$\rho_j^i = \frac{k_j^i}{\sum_{x=1}^{x<=|f^i|} k_x^i} \tag{4.13}$$

The potential value ρ_j^i implies the probability of hit for Interest i through interface j. The interface list for Interest I is sorted by the value ρ. Thus, ρ_1^i has the highest priority for player i.

Content Name	**Interfaces**	**Fits**	**Potential**
Youtube	A	k_1	ρ_1^{You}
	B	k_2	ρ_2^{You}
	C	k_3	ρ_3^{You}
Facebook	B	k_1	ρ_1^{Face}
	D	k_2	ρ_2^{Face}
	F	k_3	ρ_3^{Face}
...	...	...	...

FIGURE 4.2: An example of FIB table for potential heuristic allocation.

In PHA method, 1) the node sorts x_i. The player i with smallest x_i has the highest priority. 2) the node sort the ρ_j^i. The sorting algorithm compare two ρ by priority firstly. If the priority is same, then compare the real value of two ρ. 3) The node allocate the resources for each ρ_j^i by the sorted sequence until the capacity of each interface reaches the threshold $|F_i|$ or all ρ_j^i has been allocated.

An example is described in Table.4.1. The actions of all players are $x_1 = x_2 = x_3 = x_4 = 4$. In PHA method, the priority queue is used to represent the fairness. This parameter keeps that network resources can be allocated to each user fairly. The parameter ρ denotes the network utility. Under the premise of ensuring fair, the routers consider the network efficiency. Routers allocate the network resources to the players who has great probability to get cache hit.

TABLE 4.1: An Example of PHA Method

Priority Queue	Players			
Q_1	ρ_1^1	ρ_1^3	ρ_1^2	ρ_1^4
Q_2	ρ_2^2	ρ_2^1	ρ_2^3	ρ_2^4
Q_3	ρ_3^1	ρ_3^2		
Q_4	ρ_4^2			

4.5 Evaluation

In order to assess the effectiveness of proposed scheme for CCN, I implemented the game theoretical Interest forwarding scheme by extending ccnSim [57] simulator which is the OMNET++ based CCN simulator. Simulations were run on an Intel Core 2 Duo CPU T9400 running at 2.53 GHz and 4 GB of memory.

4.5.1 Simulation Setting

In simulation, a network is modeled as a graph $G = (n, p)$, where n is the number of nodes in the network and p is the probability of a connecting link exists between two nodes. GT-ITM [76] is used to generate a topology simulating the Internet, whose $n = 50, p = 0.3$. Links between nodes are characterized by their bandwidth and propagation delay. The bandwidth of each link is set to 100Mbs and link propagation delays range from 1ms to 5ms.

In simulation scenario, the chunk size is set to 10KB; file size is about 10^3 chunks; catalog size is up to 10^7 files. I select cache sizes of 10 GB and keep the ratio of cache over catalog on the order of 10_{-4} ($Cache/Catalog = 10^{-4}$). The routers use standard replacement method LRU (evicts the least recently used packet) and decision polices ALWAYS (caches every chunk it receives) [38]. The parameters of simulation are showed in Table 4.2.

Table 4.2: Simulation parameters

Parameters	Values	Explanation
n	50	Number of nodes
p	0.3	Connectivity probability
b	100,150, 200Mbs	Link bandwidth
d	[1,5]ms	Link delay
α	0.9	content popularity distribution skewness
q	0.25	content popularity distribution skewness
Chunk size	10KB	CCN chunk size
Cache size	10GB	Cache size of each node
Catalog size	10^7 files	each file is 10^3 chunks
(Cache/Catalog) ratio	10^{-4}	$\frac{C}{\lvert F \rvert F}$

There are two repositories which store the same content. Among the nodes, two nodes connected to repository are randomly selected. The Mandelbrot-Zipf distribution model is used to calculate the content popularity, where α and q=0.25. The network has 10 client users which are connected to its border nodes. Users perform File-level requests according to a Poisson process with exponentially distributed arrival times at a 1 Hz rate.

4.5.2 Simulation Results

I did the evaluation and analyze the effectiveness of CCN with three different Interest forwarding algorithms:

- GIMF: A node forwards the Interests by game theoretical Interest Multiple Forwarding decision method;
- CCN-S: A node forwards the Interests by the shortest path algorithm;
- CCN-B: A node forwards the Interests to all interfaces through which the Data is available.

I compare the four schemes by focusing on the metric: average data retrieve time, which denotes the user's benefits directly.

Figure 4.3 shows date retrieve time as function of cache over catalog ratio with content popularity distribution skewness $alpha = 0.8$ in CCN with three different Interest forwarding methods. Abscissa is the cache over catalog ratio. Ordinate is the average data retrieve time. We can see that with the cache size increases, data retrieve time sharply decreases. When the cache size is small, the GIMF has slightly better

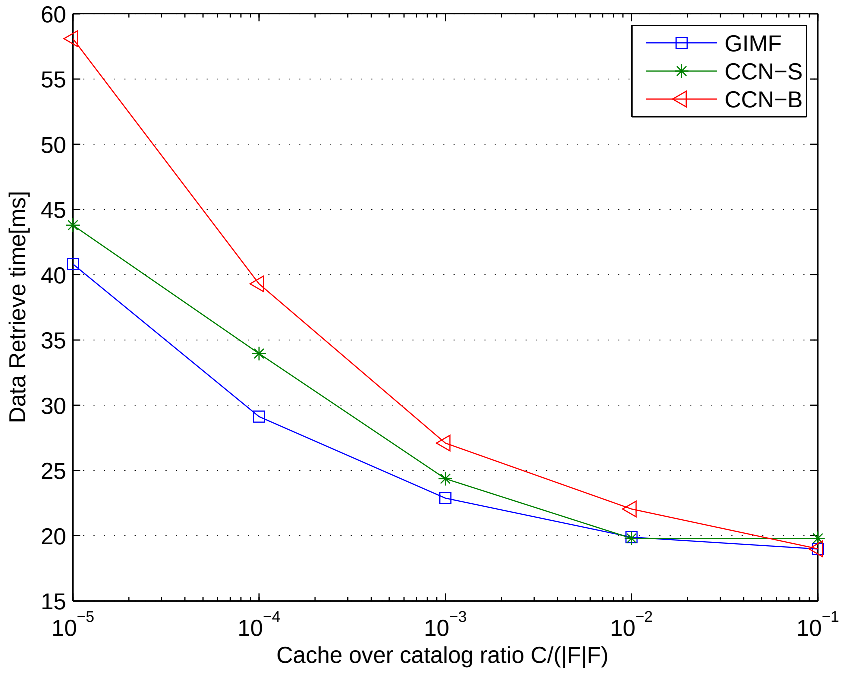

FIGURE 4.3: Date retrieve time as function of cache over catalog ratio.

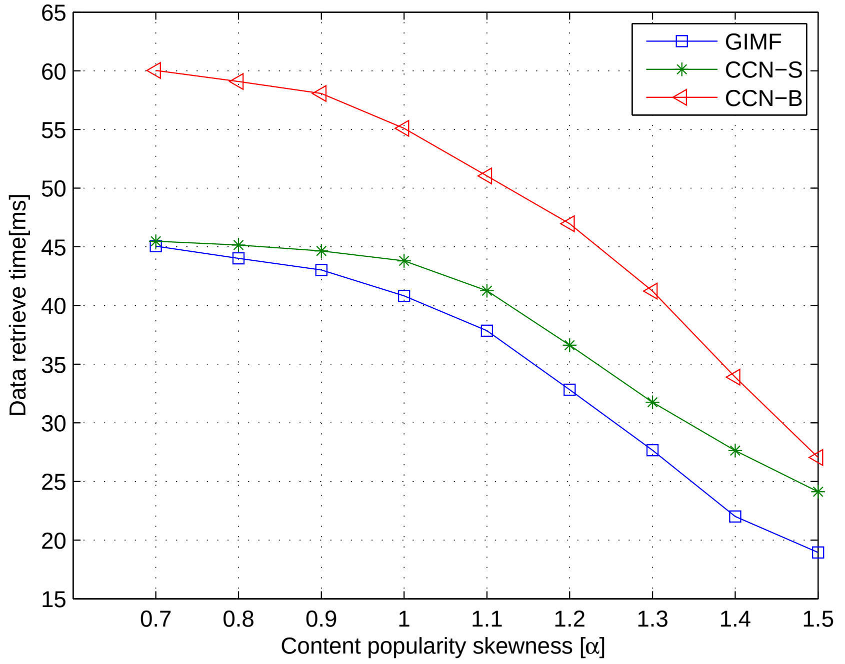

FIGURE 4.4: Data retrieve time as function of content popularity skewness [α].

performance than CCN-S. However, as the cache size increases, the gap between three forwarding mechanisms is becoming smaller until same. GIMF has dramatically better performance than original CCN-B. This is due to the fact that CCN forwards Interest to all reachable service instances, which takes up the large of bandwidth and makes the network congestion.

Figure 4.4 depicts the Data retrieve time as function of content popularity skewness α with cache size C = 10GB. It can be seen that data retrieve time decreases as the content popularity distribution skewness alpha increases, especially when alpha more than 1.0, there is a sharply decline. CCN with GIMF has similar performance with CCN-S when the skewness α is small. As skewness α increase, GIMF has better performance than CCN-S. This is because that GIMF forwards the Interest to multiple paths which can get higher cache hits than CCN-S when the popular data increase.

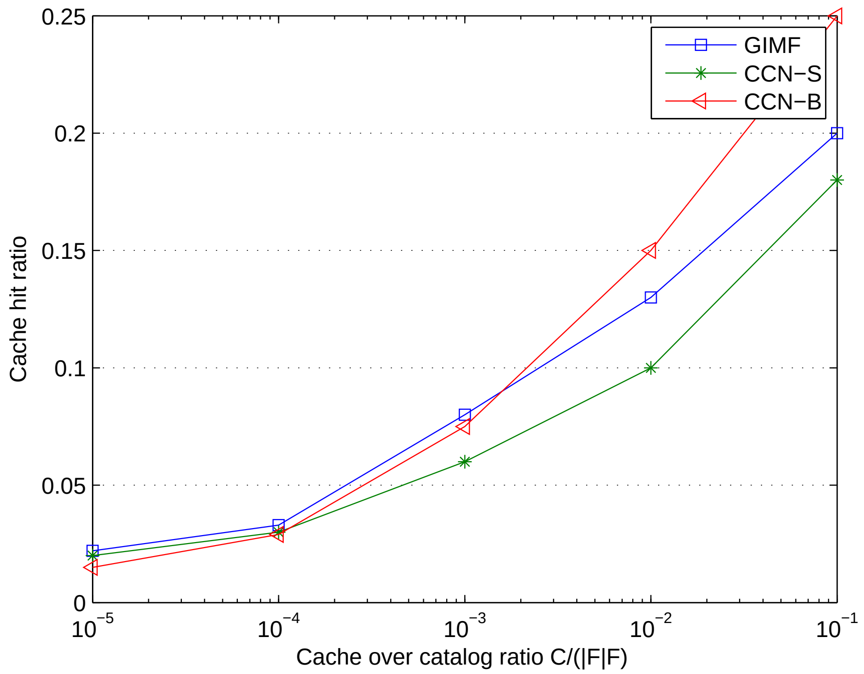

FIGURE 4.5: Cache hit as function of cache over catalog ratio.

I also evaluate the cache hit cache hit ratio as function of cache over catalog ratio for three forwarding schemes. As showed in Fig. 4.5, with the increase of cache over catalog ratio, the cache hit radio of all schemes increased. Furthermore, GIMF scheme has higher cache hit ratio than the other two schemes when cache over catalog ratio is smaller than 10^{-3}; when cache size over catalog ratio is bigger than 10^{-3}, GIMF scheme has lower cache hit than CCN-B, but better performance than CCN-S.

Chapter 5

Cluster-based In-networking Caching for CCN

5.1 Preface

In-networking catching is the distinctive feature of CCN infrastructure and plays an important role in terms of system performance. In-network caching mechanism can avoid wasting network bandwidth due to the repeated delivery of popular content. Kutscher, et al [26] define three key issues which influence the performance of in-networking caching system, i.e., cache placement, content-placement, request-cache routing. This chapter focuses on the content placement issue and the request-cache routing issue.

The proposals in this chapter are motivated by following considerations. Due to content popularity, often, the same content is accessed by many users, which makes network traffic exhibit high redundancy. Furthermore, even CCN enables individual nodes to reduce redundancy by managing a local cache, redundancy can freely appear across different nodes as the default ubiquitous LRU caching scheme and the support of multi-path routing. These two aspects motive us to consider that controlling the redundancy level is a critical issue to improving the systematic caching performance of CCN.

This chapter proposed a cluster-based in-networking caching for CCN. Through dividing nodes of network into clusters, it makes sure that no cache redundancy happens in a cluster to improve cache diversity. In order to efficiently control and manage the state of cache in a cluster, it uses a distributed hash table.

The goals of cluster-based in-networking caching scheme are improving cache hit which means reduction in bandwidth usage, reducing caching redundancy, efficient utilization of available cache resources, and balancing distribution of content among the available caches.

The rest of this chapter is organized as follows. System model and assumptions are detailed in Section 5.2 and Section 5.3 introduced the proposal, cluster-based in-networking caching for CCN. Finally, section 5.4 is the simulation evaluation.

5.2 Problem Analysis

As mentioned above, both on-path caching and off-path caching have advantages and disadvantages. Redundancy of contents caching makes on-path caching have limited cache hits even it do well in scalability. Oppositely, the shortness of off-path caching is limited scalability which is caused by per-content state required for routing. Problems of on-path caching and hash caching are depicted in Figure 5.1 and 5.2 respectively.

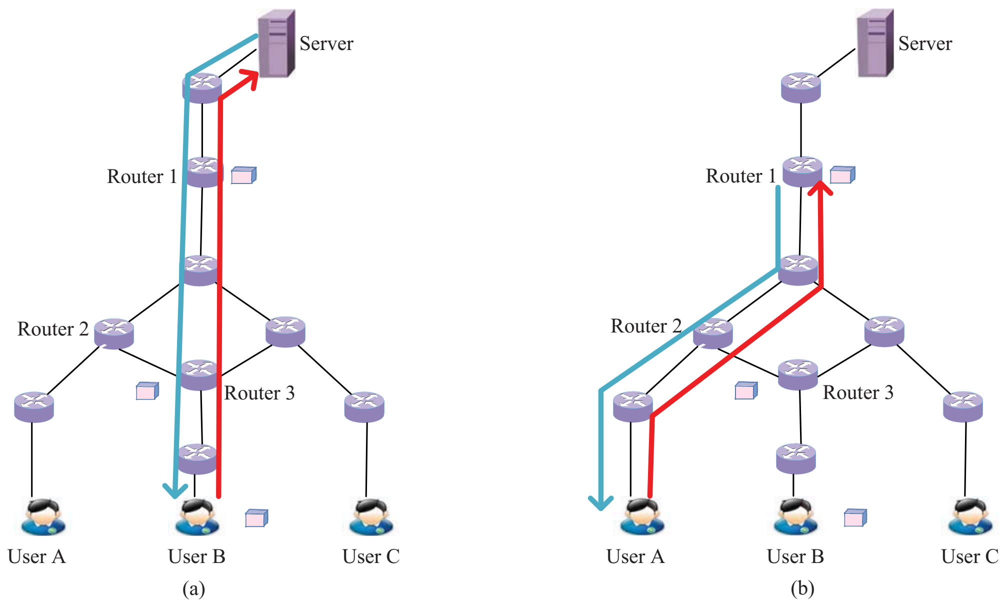

FIGURE 5.1: Problem analysis of on-path caching.

Figure 5.1 shows the problem happens in on-path caching. The red line represents the request route and the blue line indicates the data response route. In Figure 5.1(a), User B sends an Interest packet to request Data in the red line. After receiving Interest, Server returns Data packet travelling the reverse path of Interest to B. During that period, the assumption is that copies of that Data will be cached in Router 1 and

Router 3 according the on-path caching methods. User B also stores the Data packet. When another user, User A, wants to request the same Data packet as B does, problem happens, as shown in Figure 5.1(b). When the Interest sent by User A comes in Router 2, it is delivered to the direction of Router 1, and the request is responded by Router 1, as previous stored the request Data. However, we can see that, Router 3 also store the request Data packet and it is closer to Router 2 compared with Router 1. Ideally, the closer router, Router 3, is hoped to response the request, which will reduce the link load of network.

Hash caching schemes proposed by Saino et al. [20] can solve the above problem. However, it is more suitable for small scale network. When use them in large scale networks, there are also problems. Figure 5.2 demonstrates one problem happens in hash caching schemes. User sends an Interest whose request Data is previously stored in Router 2 and in Server, of course. When the Interest comes in Router 1, according to hash schemes, it is delivered to Router 2 and is responded by Router 2, rather than the Server, which is closer to Router 1.

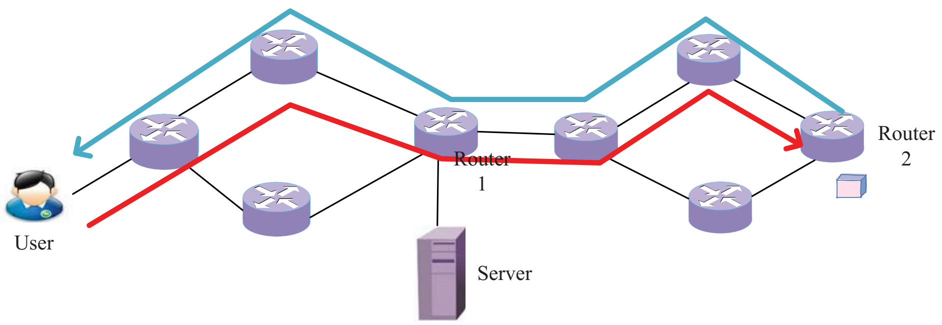

FIGURE 5.2: Problem analysis of hash caching.

To solve above problems, the goals are going to be achieved are as follows:

- Improving cache hit: increase of cache hit means reduction in bandwidth usage;
- Reducing caching redundancy: caching has been traditionally used to reduce traffic redundancy; Efficient utilization of available cache resources can improve cache diversity and routing robustness;
- Balancing distribution of content among available caches.

5.3 Cluster-based In-networking Caching

As mentioned earlier, the same content will be accessed by many users due to popularity. Thus, network traffic exhibits high redundancy. Although CCN allows individual nodes manage local caches in order to reduce redundancy, redundancy can freely appear across different nodes, i.e., different nodes store copies of the same content. This is caused by the default ubiquitous LRU caching scheme and the support of multi-path routing. Motivated by these, the proposed solution of this chapter, Cluster-based In-networking Caching, will be introduced in this section.

5.3.1 Overview

Topologies of networks in this chapter are assumed to be plane. Let $G = (V, E)$ be the graph representing the network where V ($|V| = N$, N is the number of nodes in network) are the nodes and E are the edges in the graph. By using cluster algorithm, the network is divided into K clusters. Furthermore, corresponding routing polices for inter-cluster and intra-cluster are also designed.

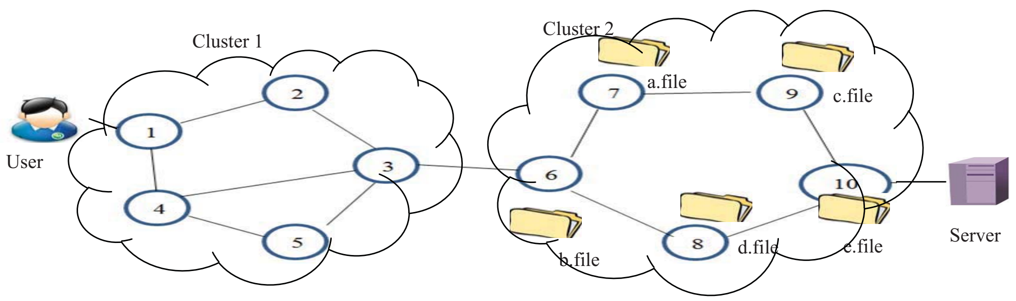

Figure 5.3: An example of cluster-based in-networking caching.

In each cluster, there is no cache redundancy. To assure that, not only contents in a node are different from each other, but content in different nodes of a cluster are not same. However, content stored in different clusters could be same. Through this rule, the cache diversity of the network is increased. As shown in Figure 5.3, there are two clusters in the network, Cluster 1 and Cluster 2. In Cluster 2, different nodes cache different content, while content cached by Cluster 1 can be same with content stored in Cluster 2.

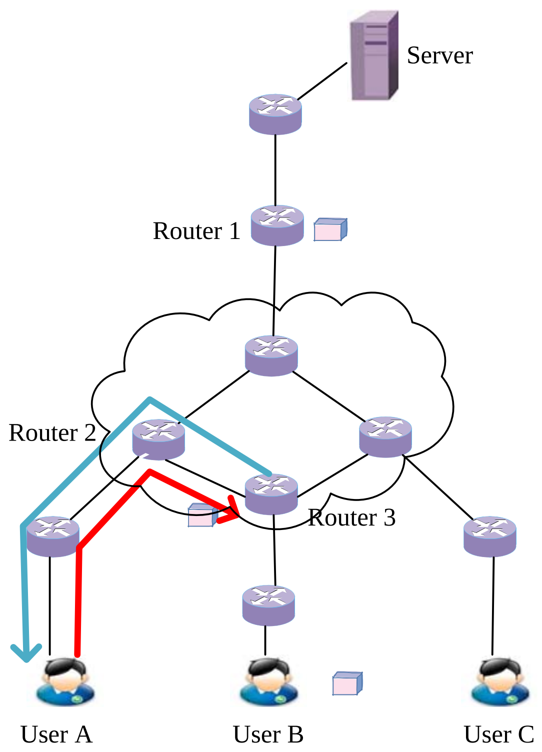

FIGURE 5.4: Cluster-based caching solves the problem of on-path caching.

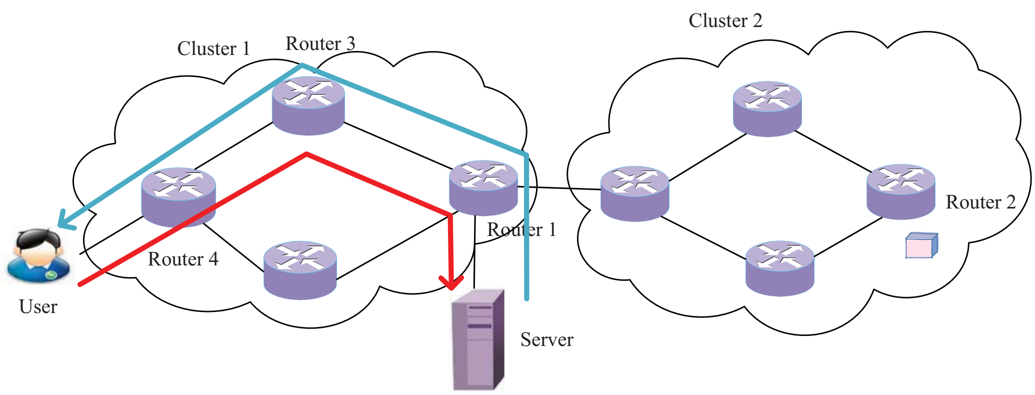

FIGURE 5.5: Cluster-based caching solves the problem of hash caching.

Virtual distributed hash table is used to effectively control and manage chunks caching in each cluster. Nodes in a cluster use a same hash function to calculate location of caching node for a specific content.

Cluster-based in-networking caching can solve the problems described in Section 5.2. The problem, which happens in Figure 5.1, can be solved as shown in Figure 5.4. Here Router 2 and Router 3 are assumed in the same cluster. After User B captured Data, same as Fig. 1(a), User A sends Interest to request the same Data. When the Interest delivered to Router 2, Router 2 forwards the Interest to Router 3 by the distributed hash routing table. Router 3 sends back the Data packet.

Figure 5.5 details how cluster-based caching solves the problem happens in hash caching schemes presented in Figure 5.2. Firstly, the network is divided into two clusters, Cluster 1 and Cluster 2. Router 1, Router 3 and Router 4 are in Cluster 1, while Router 2 is in Cluster 2. When Router 4 receives the Interest sent by User, it firstly calculates the location of caching node for the requested Data packet, which is assumed Router 3. Then Router 2 forwards the Interest to Router 3. After receiving the Interest, Router 3 check whether the Data stored locally. If it not hits, Router 3 forwards the Interest to Server by the shortest path. At last, Server returns the Data packet by reverse path of the Interest.

5.3.2 Improved K-medoids cluster algorithm

The cluster algorithm is descried in Algorithm 4, the improved K-medoids cluster algorithm. It needs two input parameters: K and G. The parameter K is currently assumed to be predefined based on the scale of the network, while the parameter G demonstrates the graph of the network,$G = (V, E)$, V is number of network nodes and E is edges of the network. The result of the algorithm is K parts division of the network, i.e., K clusters. The Euclidean distance in k-medoids clustering is replaced with new defined distance with Equation 5.1 between two nodes. The reason is that Euclidean distance cannot reflect the real relationship between nodes in networks. Equation 5.1 is used to calculate distance value of any node i to centroid node of cluster k:

Algorithm 4: The improved K-medoids cluster algorithm

Input: K, predefined according the scale of networks;$G = (V, E)$, the graph of the network,$|V| = N$, N is number of nodes of the whole network.

Output: K clusters, divide the network into K parts

1 Delete nodes with only one edge;

2 Choose K nodes who have most number of edges; Set the nodes as initial centroids of K clusters;

3 For each node i, calculate its dis_K^i from centroid node of cluster k ($k \in [1, K]$)by Equation 5.1; add node i into the cluster with the smallest distance value;

4 For each cluster k, $\forall$ node $i, j \in K$, calculate Dis_k^i by Equation 5.2; set the node with smallest Dis_k^i value as the new centroids of cluster k;

5 Repeat 3 and 4 until centroids are not change.

$$dis_k^i = \frac{d_i}{D} \times \frac{B}{b_i} \times \frac{C}{c_i} \times h_i \tag{5.1}$$

Where $i \neq k$ and D is average delay of all pair nodes in network. $d_i$is the delay from node i to node k by the shortest path; B is average bandwidth of all pair nodes in network, b_i is the average bandwidth from node i to node k by the shortest path; C is average cache size of all nodes on the path of each pair nodes in network; c_i is the cache size of nodes on the shortest path from node i to node k; $h_i$is the hops from node i to node k by the shortest path.

The following equation is used to calculate average distance from node i to other nodes in cluster k.

$$Dis_j^i = \frac{1}{|k|} \sum_{i=1}^{i<|k|} \sum_{j=i+1}^{j\leq|k|} \frac{d_{ij}}{D} \times \frac{B}{b_{ij}} \times \frac{C}{c_{ij}} \times h_{ij} \tag{5.2}$$

5.3.3 Virtual Distributed Hash Table (VDHT)

To control and manage the contents stored in each cluster, each cluster holds only one Virtual Distributed Hash Table (VDHT). The VDHT is made up of CSs of all nodes in the cluster. Each CS of the node is indicated by the node ID. In a cluster, all nodes have a same hash function. When an Interest comes in a cluster through any nodes of the cluster, the receiving node calculates the location of caching node, node ID, by the hash function according to the ID of requested chunk. Then the Interest will be delivered to the corresponding node.

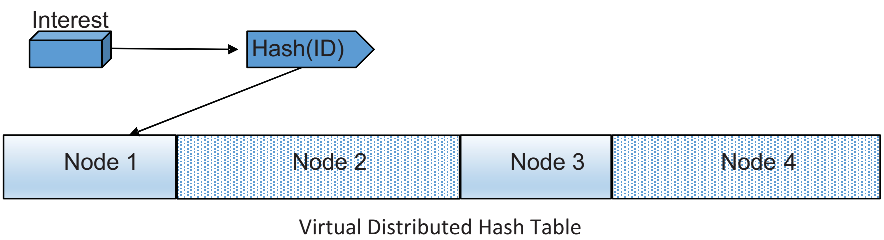

FIGURE 5.6: An example of VDHT of a cluster.

A simple VDHT of that cluster is shown in Figure 5.6. To illustrate the components and functions of VDHTs, there are four nodes in a cluster is assumed. The CSs of Node 1, Node 2, Node 3 and Node 4 construct VDHT as a whole. Each CS of the four nodes is indicated by node ID, Node 1, Node 2, Node 3 and Node 4, in the VDHT. When a Interest for chunk comes in the cluster through any of them, such as Node 3, then Node 3 calculates the node ID according the chunk ID by hash function, Node 1 for example. After that, the Interest will be forwarded from Node 3 to Node 1 and Node 1 checks the CS to make sure whether the requested chunk is cached here or not.

5.3.4 Intra-cluster and Inter-cluster forwarding

Before introducing routing polices, the components of redesigned Interest packet and Data packet are firstly presented. An Interest packets consists content name, version, sequence number, cluster ID, node ID and other filed. Content name, version and sequence number is named as chunk ID. An example of Interest packet is shown in Figure 5.7. Similarly, the header of a Data packet is similar with Interest.

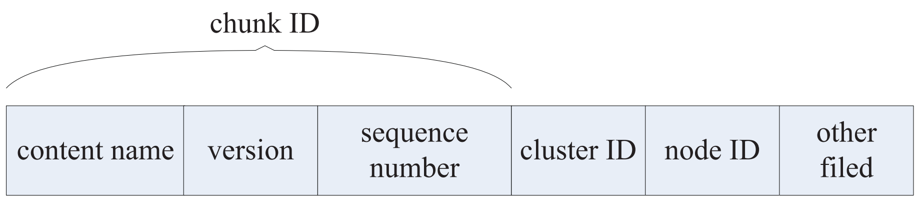

FIGURE 5.7: An example of Interest packet for Cluster Caching.

In cluster-based caching, Algorithm 5, the intra-cluster forwarding algorithm, routes Interest between nodes belonging to the same cluster.

When an Interest received by a node, the node firstly check if the cluster ID of Interest is same with its own cluster ID. If they are not same, it means that it is the first time that the Interest comes in the cluster. Then the node copies its cluster ID to *Interest.clusterID*, calculates a hash value using the Hash function according to the *Interest.chunkID* and gives it to *Interest.nodeID*, then forwards the Interest to the node with *Interest.nodeID*. However, if *Interest.clusterID* equals the cluster ID of the node, it means that this is intra-cluster forwarding. Then the node checks whether *Interest.nodeID* equals its own ID or not. This step is check whether the node is the location of the chunk that Interest request. If they are not equal, the node forwards the Interest to the node with *Interest.nodeID*. Conversely, the node is the location of the requested chunk, and then the node checks its CS to find the requested chunk. If the

Algorithm 5: The Intra-cluster Interest forwarding Algorithm

Input: Interest;

Output: Forwarding actions

```
if Interest.clusterID ≠ my.clusterID then
    Interest.clusterID ← my.clusterID;
    Interest.nodeID ← Hash(Interest.chunkID);
    Forward the Interest to the node with nodeID;
end
else
    if Interest.nodeID ≠ my.nodeID then
        Forward the Interest to the node with Interest.nodeID;
    end
    else
        if the request Data cached in CS then
            return Data;
        end
        else
            Forward the Interest to repository;
        end
    end
end
```

chunk exits, the node returns it, or the node forwards the Interest to the direction of the server.

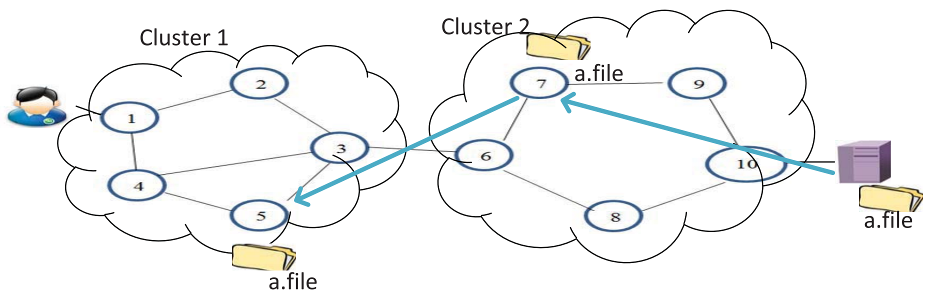

FIGURE 5.8: An example of inter cluster routing.

After Data packet generated, the Data packet is forwarded by reverse route of the Interest and stored in the node with Data.nodeID on the first cluster it passed through.

Among clusters, on-path caching is used. this part uses LCD. In inter-cluster forwarding case, when data already cached in current cluster, data packet will be cached in the next cluster data packet passed through. Figure 5.8 shows an example of inter cluster routing. After someone has requested a.file, a.file was cached in node 7 in Cluster

2. When other users request a.file through nodes in Cluster 1, a.file will be delivered to Cluster 1 and stored in node 5 as depicted.

5.4 Performance Evaluation

The performance of cluster-based in-networking caching was evaluated by extending the ccnSim [57], which is a CCN simulator based on OMNET++ [65]. For the comparison, Hash scheme, ProbCache, LCE were also implemented on the ccnSim simulator.

5.4.1 Simulation Settings

In simulation, a network is modeled as a graph $G = (n, p)$, where n is the number of nodes in the network and p is the probability of a connecting link exists between two nodes. GT-ITM [76] is used to generate a topology simulating the Internet, whose $n = 50, p = 0.3$. Links between nodes are characterized by their bandwidth and propagation delay. The bandwidth of each link is randomly chosen from a set $\{100\text{Mbs}, 150\text{Mbs}, 200\text{Mbs}\}$ and link propagation delays range from 1ms to 5ms. The network is divided into 6 clusters.

The Mandelbrot-Zipf distribution model calculates the content popularity. Unless otherwise specified, $\alpha = 0.9$ and $q = 0.25$ were set. There are two repositories which store the same content. Among the nodes, 2 nodes which are connected to repository were randomly selected. The network has 10 client users which are connected to its border nodes. Users perform File-level requests according to a Poisson process with exponentially distributed arrival times at a 2 Hz rate.

All caching schemes have been evaluated assuming that the chunk size is 10KB; file size is about chunks; catalog size is up to files. The cache sizes of 10 GB are selected and the ratio of cache over catalog on the order of ($Cache/Catalog = 10^{-4}$) is kept. The contents are evicted according to a Least Recently Used (LRU) policy (evicts the least recently used packet) and are cached by decision ALWAYS policy (caches every chunk it receives). For clarity, parameters described above and other parameters are listed in Table 5.1.

In order to assess the performance of the cluster-based caching, following schemes are implemented:

- Cluster: Cluster-based caching scheme proposed in this paper;

Table 5.1: Simulation parameters

Parameters	Values	Explanation
n	50	Number of nodes
p	0.3	Connectivity probability
b	100,150, 200Mbs	Link bandwidth
d	[1,5]ms	Link delay
α	0.9	content popularity distribution skewness
q	0.25	content popularity distribution skewness
Chunk size	10KB	CCN chunk size
Cache size	10GB	Cache size of each node
Catalog size	10^7 files	each file is 10^3 chunks
(Cache/Catalog) ratio	10^{-4}	$\frac{C}{\lvert F\rvert F}$

- Hash: Symmetric hash routing proposed in [20];
- ProCache: cache content along the path by probability;
- LCE: cache everything everywhere.

The four schemes are compared by focusing on two metrics: 1) cache hit radio, which represents the capability of the caching scheme to reduce the amount of redundant traffic; 2) link load, which is used to evaluate the efficiency of data transmission on network; 3) average data retrieve time, which denotes performance of network in user view. Cache hit radio and link load are measured for four schemes under varying content popularity distribution skewness and cache over catalog radio. Content popularity distribution skewness is represented by parameter α and q of the Mandelbrot-Zipf.

5.4.2 Simulation Results

The simulation results are depicted from Figure 5.9 to Figure 5.12. Figure 5.9 and Figure 5.10 show the changing curves of cache hit ratio of four schemes with varying cache over catalog ratio and varying content popularity skewness. The affections of changing cache over catalog ratio and varying content popularity skewness on the link load of the network are represented on Figure 5.11 and Figure 5.12.

From the Figure 5.9, it is shown that with the increase of cache over catalog ratio, the cache hit radio of all schemes increased. Further, all the off-path schemes, Cluster-based caching and Hash routing, have higher cache hit ratio than on-path schemes, ProbCahe and LCE. More specifically, Cluster-based caching scheme outperforms than ProCache and LCE, while it does worse than Hash routing scheme. This phenomenon is

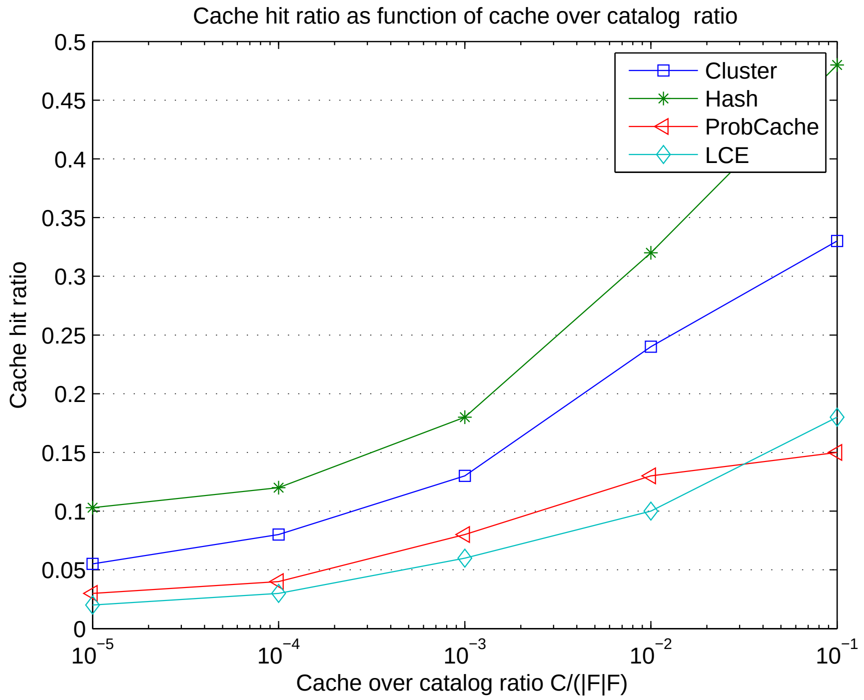

FIGURE 5.9: Cache hit ratio as function of cache over catalog ratio.

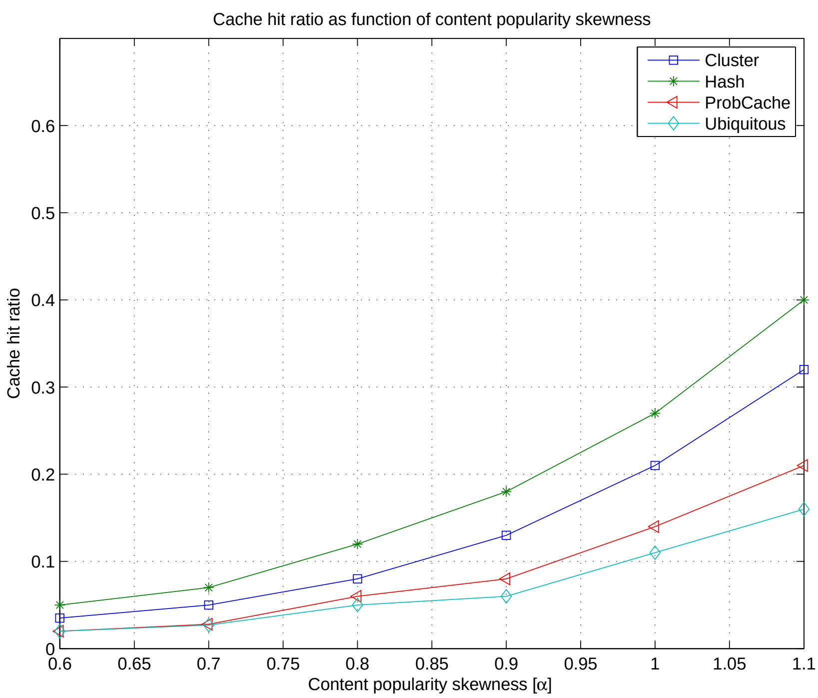

FIGURE 5.10: Cache hit ratio as function of content popularity skewness.

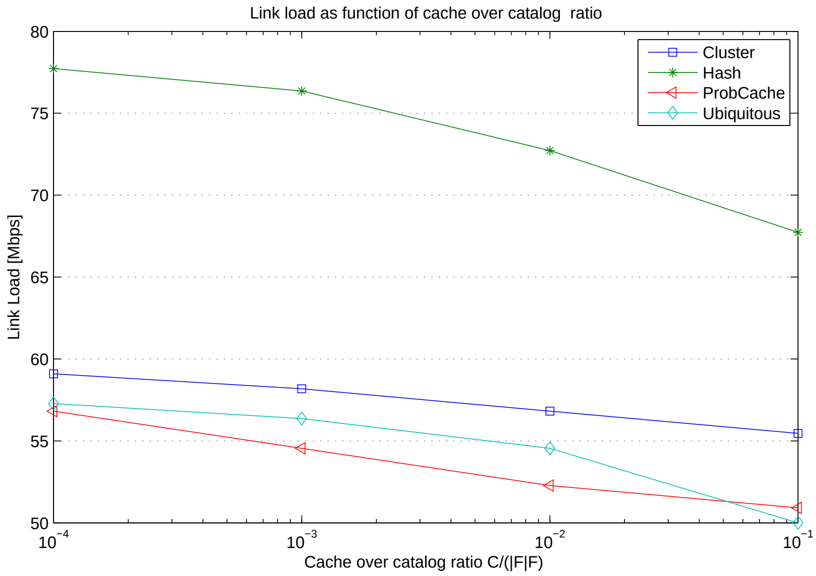

FIGURE 5.11: Link load as function of cache over catalog ratio.

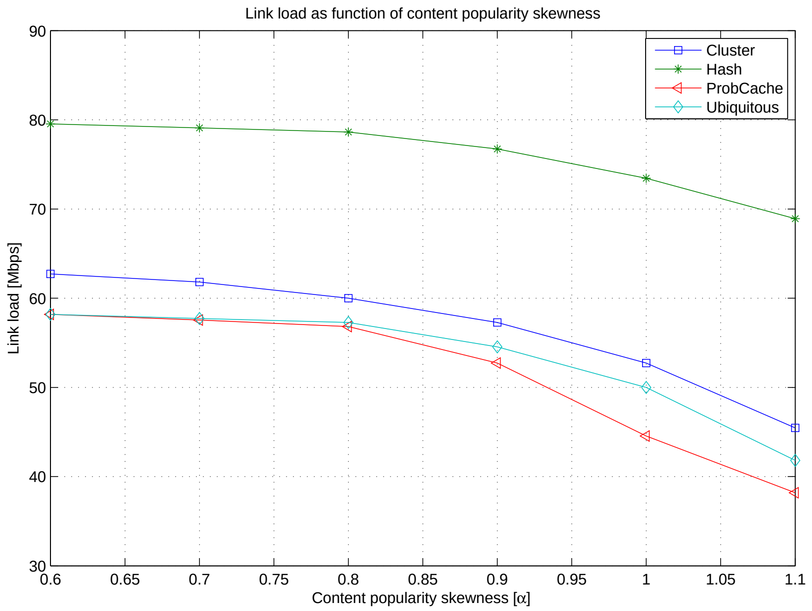

FIGURE 5.12: Link load as function of content popularity skewness.

caused by that cluster-based caching using cluster method to improve the cache diversity of the network compared with ProCache and LCE, while hash routing has higher cache diversity than that of cluster-based caching by making all contents of the network stored are different.

The results showed in Figure 5.10 are similar with Figure 5.9. However, the difference between them is that the gaps between lines in Figure 5.10 are smaller than that in Figure 5.9. From that, the conclusion that cache over catalog ratio does more effect on the cache hit ratio than content popularity skewness does can be get.

Figure 5.11 shows that link load decreasing with increasing cache over catalog ratio. Conversely to cache hit ratio, all on-path schemes has lower values than off-path schemes have. In other words, on-path caching schemes outperform than off-path schemes in terms of link load. This is because the on-path schemes use the shortest path method at cost of throughput reducing to get data, which decrease link load. Comparing the two off-path schemes, link loads of cluster-based caching are extraordinarily smaller than that of hash scheme. The reason for that is cluster reducing the redundant routes as Figure 5.2 depicted.

Link loads are varying with changing content popularity skewness is shown in Figure 5.12. As it is shown, the changing lines are similar with lines in Figure 5.11.

Chapter 6

Conclusions

Content-Centric Networking (CCN) emerges as a novel clean slate network architecture. The core feature of CCN is changed from host-centric communication model to content-centric model. In-networking caching is the general infrastructure of CCN and contents are retrieved directly by their names, instead of locations. In-networking caching and name based forwarding engine make CCN more powerful than IP network in content distribution. Thus, this thesis focused on two significant basic components of CCN: caching management and forwarding strategies.

This thesis firstly illustrated the architecture of CCN and summarized the features of CCN. Secondly, related works according to forwarding engine and cache management of CCN are reviewed. Then, the problems and challenge issues are presented. Lastly, the corresponding methods were designed to solve these problems.

In Chapter 3, GDACF adopted the Domain-based aggregation to solve the issues due to the large naming space. The network is divided into multiple domains. Popularity inspired caching and adaptive rate (AR) schemes are studied to improve energy-efficiency of CCN. GDACF algorithm utilizes two kinds of ants to complete the entire routing and forwarding optimization progress. The responsibility of Hello Interest Ants is gathering forwarding information and optimizing the path for Normal Interest Ant packets. The Normal Interest Ants reinforce the optimization of the quality of service aware path. The Normal Interstate Ants adopt the greedy method for next hop selection in the Inter-domain forwarding case and adopt multi-casting method for Intra-domain case. The Hello Interest Ants select the next hop probabilistically for the purposes that the current network states can be updated in time and the new path can be found.

The main objectives of GDACF include (1) selecting performance metrics to rank interfaces, e.g. overhead, round trip delay, bandwidth; (2) supporting intelligent forwarding of Interests over multiple paths, which is Quality of Service aware and traffic type aware; and (3) avoiding instability (frequent oscillation of paths) while maintaining good data delivery performance. Different policies in Intra-domain forwarding case and Inter-domain forwarding case are disigned for the goal of scalability and mobility.

Simulations for GDACF were did in three scenarios: normal traffic, congestion and link failure. GDACF presents better performance in data retrieve time, power saving and recover time. Simulation results show that GDACF algorithm adaptively reduces the impacts incited by the complex dynamic network, e.g. link failure, network congestion and dynamic network topology.

Chapter 4 investigated Interest forwarding strategy in Content-Centric Networking where a set of Interests sharing a multiple interfaces from which the Interest can get the response from repository. Users are assumed to be self-regarding and make their decisions with the sole goal of maximizing their perceived quality. Game theoretical Interest Multiple Forwarding decision (GIMF) method was presented to improve the users' payoff when the network is not in the high load traffic. GIMF used non-cooperative game theory to analysis the replica Interests forwarding decision. Interest flows were treated as the game player. Each game player maximizes his payoff cost. In the network perspective, Potential Heuristic Allocation (PHA) method was proposed to queue the replica Interests which considers the fairness and network efficiency simultaneously. GIMF improves the utilization rate of network resources. The simulation results show that this proposals improved the CCN performance. It can be adaptively make the Interest multiple forwarding decisions in different network traffic scenarios.

In Chapter 5, cluster-based caching mechanism was proposed to improve cache hit, reduce caching redundancy, and balance distribution of content among available caches. In the cluster-based caching mechanism, the improved K-medoids cluster algorithm is designed to cluster the whole network into clusters. Virtual Distributed Hash Table (VDHT) was used to efficiently control and manage the content stored in each cluster. Different policies for intra cluster routing and inter cluster routing were designed for effective routing. Through simulations, cluster-based in-networking caching outperforms than on-path caching schemes, ProCache and LCE in terms of cache hit radio. It also do better in link load compared with hash schemes. The main contributions of this part are: 1) Cluster-based in-networking caching for CCN is proposed to improve cache hit, reduce cache redundancy and improve efficient utilization of available cache resources, and balance distribution of content among the available caches; 2) Two type of routing policies are designed for cluster-based in-networking caching, i.e., inter-cluster routing and intra-cluster routing; 3) The effectiveness of cluster-based in-networking caching scheme is evaluated by extensive simulations. The results show that this scheme balance the cache hit radio and link load compared with other schemes.

The future works are presented as following:

- To the forwarding strategies, the terms of efficiency and fairness need to be considered based on the model of [77] which provides an analytical framework for the evaluation of average content delivery performance under statistical bandwidth and storage sharing.

- To the utilization of cached data, different game theory models need to be disscused for Interest multiple forwarding decisions in CCN. Furthermore, the multipath Interest forwarding for CCN can be considered.

- To the cluster based caching, it is worthy to be discussed that how to automatically generate the scale of clusters and the number of clusters of a network.

Bibliography

[1] G. Pallis and A. Vakali. Insight and perspectives for content delivery networks. *Communications of the ACM*, 49(1):101–106, 2006.

[2] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi. A survey on content-oriented networking for efficient content delivery. *Communications Magazine, IEEE*, 49(3): 121–127, 2011.

[3] Dmitrij Lagutin, Kari Visala, and Sasu Tarkoma. Publish/subscribe for internet: Psirp perspective. *Valencia FIA book*, 4, 2010.

[4] Christian Dannewitz. Netinf: An information-centric design for the future internet. In *Proc. 3rd GI/ITG KuVS Workshop on The Future Internet*, 2009.

[5] Snehapreethi Gopinath, Shweta Jain, Shivesh Makharia, and Dipankar Raychaudhuri. An experimental study of the cache-and-forward network architecture in multihop wireless scenarios. In *Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on*, pages 1–6. IEEE, 2010.

[6] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review*, 37(4):181–192, 2007.

[7] Mark Gritter and David R Cheriton. An architecture for content routing support in the internet. In *USITS*, volume 1, pages 4–4, 2001.

[8] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.

[9] D. Perino and M. Varvello. A reality check for content centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 44–49. ACM, 2011.

[10] The size of content naming space, 2008. URL `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`.

[11] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. A case for stateful forwarding plane. *Computer Communications*, 36 (7):779–791, 2013.

[12] Cheng Yi, Jerald Abraham, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. On the role of routing in named data networking. *Named-Data Networking Project, Tech. Rep*, 2013.

[13] Aadil Zia Khan, Shahab Baqai, and Fahad R Dogar. Qos aware path selection in content centric networks. In *Communications (ICC), 2012 IEEE International Conference on*, pages 2645–2649. IEEE, 2012.

[14] C. Li, W. Liu, and K. Okamura. A greedy ant colony forwarding algorithm for named data networking. In *Proceedings of the APAN Network Research Workshop 2012*. APAN, 2012.

[15] S. Shanbhag, N. Schwan, I. Rimac, and M. Varvello. Soccer: services over content-centric routing. In *ACM SIGCOMM workshop on Information-centric networking*, pages 62–67, 2011.

[16] Jonas Eymann and Andreas Timm-Giel. Multipath transmission in content centric networking using a probabilistic ant-routing mechanism. In *Mobile Networks and Management*, pages 45–56. Springer, 2013.

[17] Raffaele Chiocchetti, Diego Perino, Giovanna Carofiglio, Dario Rossi, and Giuseppe Rossini. Inform: a dynamic interest forwarding mechanism for information centric networking. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pages 9–14. ACM, 2013.

[18] Jaime Garcia-Reinoso, Norberto Fernández, Ivan Vidal, and Jesús Arias Fisteus. S-calable data replication in content-centric networking based on alias names. *Journal of Network and Computer Applications*, 2014.

[19] Amuda James Abu, Brahim Bensaou, and Jason Min Wang. Interest packets retransmission in lossy ccn networks and its impact on network performance. In *Proceedings of the 1st international conference on Information-centric networking*, pages 167–176. ACM, 2014.

[20] Lorenzo Saino, Ioannis Psaras, and George Pavlou. Hash-routing schemes for information centric networking. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pages 27–32. ACM, 2013.

[21] Toru Hasegawa, Yuto Nakai, Kaito Ohsugi, Junji Takemasa, Yuki Koizumi, and Ioannis Psaras. Empirically modeling how a multicore software icn router and an icn network consume power. In *Proceedings of the 1st international conference on Information-centric networking*, pages 157–166. ACM, 2014.

[22] Satoshi Imai, Kenji Leibnitz, and Masayuki Murata. Energy-aware cache management for content-centric networking. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 1623–1629. IEEE, 2013.

[23] Nakjung Choi, Kyle Guan, Daniel C Kilper, and Gary Atkinson. In-network caching effect on optimal energy consumption in content-centric networking. In *Communications (ICC), 2012 IEEE International Conference on*, pages 2889–2894. IEEE, 2012.

[24] Dabin Kim, Sung-Won Lee, Young-Bae Ko, and Jae-Hoon Kim. Cache capacity-aware content centric networking under flash crowds. *Journal of Network and Computer Applications*, 2014.

[25] Jun Li, Bin Liu, and Hao Wu. Energy-efficient in-network caching for content-centric networking. *Communications Letters, IEEE*, 17(4):797–800, 2013.

[26] Dirk Kutscher, Suyong Eum, Kostas Pentikousis, Ioannis Psaras, Daniel Corujo, Damien Saucez, Thomas Schmidt, Matthias Waehlisch, et al. Icn research challenges. *IRTF, draft-kutscher-icnrg-challenges*, 2013.

[27] P Krishnan, Danny Raz, and Yuval Shavitt. The cache location problem. *IEEE/ACM Transactions on Networking (TON)*, 8(5):568–582, 2000.

[28] Valentino Pacifici and Gyorgy Dan. Content-peering dynamics of autonomous caches in a content-centric network. In *INFOCOM, 2013 Proceedings IEEE*, pages 1079–1087. IEEE, 2013.

[29] Wei Koong Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache "less for more" in information-centric networks. In *NETWORKING 2012*, pages 27–40. Springer, 2012.

[30] Stéphane Deconinck. Minimizing bandwidth on peering links with deflection in named data networking. 2012.

[31] Munyoung Lee, Kideok Cho, Kunwoo Park, Taekyoung Kwon, and Yanghee Choi. Scan: Scalable content routing for content-aware networking. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[32] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D Thornton, Diana K Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, Christos Papadopoulos, et al. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[33] Lan Wang, AKMM Hoque, Cheng Yi, Adam Alyyan, and Beichuan Zhang. Ospfn: An ospf based routing protocol for named data networking. *University of Memphis and University of Arizona, Tech. Rep*, 2012.

[34] Cheng Yi, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. Adaptive forwarding in named data networking. *ACM SIGCOMM computer communication review*, 42(3):62–67, 2012.

[35] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, and Michele Papalini. Multipath congestion control in content-centric networks. *IEEE NOMEN*, 13, 2013.

[36] Hang Liu, Xavier De Foy, and Dan Zhang. A multi-level dht routing framework with aggregation. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 43–48. ACM, 2012.

[37] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999.

[38] Dario Rossi and Giuseppe Rossini. Caching performance of content centric networks under multi-path routing (and more). *Relatório técnico, Telecom ParisTech*, 2011.

[39] Nikolaos Laoutaris, Hao Che, and Ioannis Stavrakakis. The lcd interconnection of lru caches and its analysis. *Performance Evaluation*, 63(7):609–634, 2006.

[40] Jia Wang. A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review*, 29(5):36–46, 1999.

[41] Ioannis Psaras, Wei Koong Chai, and George Pavlou. Probabilistic in-network caching for information-centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 55–60. ACM, 2012.

[42] Jun Li, Hao Wu, Bin Liu, Jianyuan Lu, Yi Wang, Xin Wang, Yanyong Zhang, and Lijun Dong. Popularity-driven coordinated caching in named data networking. In *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, pages 15–26. ACM, 2012.

[43] Gareth Tyson, Sebastian Kaune, Simon Miles, Yehia El-khatib, Andreas Mauthe, and Adel Taweel. A trace-driven analysis of caching in content-centric networks. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pages 1–7. IEEE, 2012.

[44] Renu Tewari, Michael Dahlin, Harrick M Vin, and Jonathan S Kay. Design considerations for distributed caching on the internet. In *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*, pages 273–284. IEEE, 1999.

[45] Konstantinos Katsaros, George Xylomenos, and George C Polyzos. Multicache: An overlay architecture for information-centric networking. *Computer Networks*, 55(4): 936–947, 2011.

[46] Elisha J Rosensweig and Jim Kurose. Breadcrumbs: Efficient, best-effort content location in cache networks. In *INFOCOM 2009, IEEE*, pages 2631–2635. IEEE, 2009.

[47] Raffaele Bolla, Roberto Bruschi, Alessandro Carrega, Franco Davoli, Diego Suino, Constantinos Vassilakis, and Anastasios Zafeiropoulos. Cutting the energy bills of

internet service providers and telecoms through power management: An impact analysis. *Computer Networks*, 56(10):2320–2342, 2012.

[48] Mingwei Xu, Yunfei Shang, Dan Li, and Xin Wang. Greening data center networks with throughput-guaranteed power-aware routing. *Computer Networks*, 57(15): 2880–2899, 2013.

[49] Raffaele Bolla, Roberto Bruschi, Alessandro Carrega, and Franco Davoli. Green networking with packet processing engines: Modeling and optimization. *IEEE/ACM Transactions on Networking (TON)*, 22(1):110–123, 2014.

[50] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI*, volume 8, pages 323–336, 2008.

[51] RQ Shaddad, AB Mohammad, SA Al-Gailani, AM Al-hetar, and MA Elmagzoub. A survey on access technologies for broadband optical and wireless networks. *Journal of Network and Computer Applications*, 41:459–472, 2014.

[52] Uichin Lee, Ivica Rimac, Daniel Kilper, and Volker Hilt. Toward energy-efficient content dissemination. *Network, IEEE*, 25(2):14–19, 2011.

[53] Muhammad Rizwan Butt, Oscar Delgado, and Mark Coates. An energy-efficiency assessment of content centric networking (ccn). In *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1–4. IEEE, 2012.

[54] B. Chandra Mohan and R. Baskaran. A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert Systems with Applications*, 2011.

[55] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41, 1996.

[56] M.S.R. Monteiro, D. Fontes, and F. Fontes. An ant colony optimization algorithm to solve the minimum cost network flow problem with concave cost functions. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 139–146. ACM, 2011.

[57] Giuseppe Rossini, Dario Rossi, et al. Large scale simulation of ccn networks. *Large scale simulation of CCN networks*, pages 1–4, 2012.

[58] K. Bhaskaran, J. Triay, and V.M. Vokkarane. Dynamic anycast routing and wavelength assignment in wdm networks using ant colony optimization (aco). In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[59] G. Di Caro and M. Dorigo. Antnet: Distributed stigmergetic control for communications networks. *arXiv preprint arXiv:1105.5449*, 2011.

[60] B. Baran and R. Sosa. A new approach for antnet routing. In *Computer Communications and Networks, 2000. Proceedings. Ninth International Conference on*, pages 303–308. IEEE, 2000.

[61] Laura Rosati, Matteo Berioli, and Gianluca Reali. On ant routing algorithms in ad hoc networks with critical connectivity. *Ad Hoc Networks*, 6(6):827–859, 2008.

[62] Christos Tsilopoulos and George Xylomenos. Supporting diverse traffic types in information centric networks. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 13–18. ACM, 2011.

[63] Abdesselem Kortebi, Luca Muscariello, Sara Oueslati, and James Roberts. Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing. In *ACM SIGMETRICS Performance Evaluation Review*, volume 33, pages 217–228. ACM, 2005.

[64] Sara Oueslati, James Roberts, and Nada Sbihi. Flow-aware traffic control for a content-centric network. In *INFOCOM, 2012 Proceedings IEEE*, pages 2417–2425. IEEE, 2012.

[65] OMNet. Omnet++ network simulation framework. `http://www.omnetpp.org/`, 2013.

[66] Ellen W Zegura, Kenneth L Calvert, Samrat Bhattacharjee, et al. How to model an internetwork. In *IEEE infocom*, volume 96, pages 594–602. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1996.

[67] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 15–28. ACM, 2007.

[68] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior (Commemorative Edition)*. Princeton university press, 2007.

[69] Fotini-Niovi Pavlidou and Georgios Koltsidas. Game theory for routing modeling in communication networks: A survey. *Journal of communication and networks*, 10 (3):268–286, 2008.

[70] Dimitris E Charilas and Athanasios D Panagopoulos. A survey on game theory applications in wireless networks. *Computer Networks*, 54(18):3421–3430, 2010.

[71] Roger B Myerson. *Game theory: analysis of conflict*. Harvard University Press, 1997.

[72] Ariel Orda, Raphael Rom, and Nahum Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking (ToN)*, 1(5): 510–521, 1993.

[73] Eitan Altman, Tamer Basar, Tania Jimenez, and Nahum Shimkin. Competitive routing in networks with polynomial costs. *Automatic Control, IEEE Transactions on*, 47(1):92–96, 2002.

[74] Ismet Sahin and Marwan A Simaan. A flow and routing control policy for communication networks with multiple competitive users. *Journal of the Franklin Institute*, 343(2):168–180, 2006.

[75] Richard J La and Venkat Anantharam. Optimal routing control: Repeated game approach. *Automatic Control, IEEE Transactions on*, 47(3):437–450, 2002.

[76] EW Zegura. Gt-itm: Georgia tech internetwork topology models (software). *Georgia Tech,"http://www. cc. gatech. edu/fac/Ellen. Zegura/gt-itm/gt-itm/tar.gz"*, 1996.

[77] Luca Muscariello, Giovanna Carofiglio, and Massimo Gallo. Bandwidth and storage sharing performance in information centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 26–31. ACM, 2011.

Published Papers

1. Chengming Li, Wenjing Liu, Koji Okamura. A greedy ant colony forwarding algorithm for Named Data Networking [C]. Proceedings of the Asia-Pacific Advanced Network, 2012, 34: 17-26.

2. Chengming Li, Koji Okamura, Wenjing Liu. Ant Colony Based Forwarding Method for Content-Centric Networking [C]. Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on. IEEE, 2013: 306-311.

3. Chengming Li, Koji Okamura. Game Theory based Interest Forwarding Decisions in Content-Centric Networking [C]. Proceedings of the 8th International Conference on Future Internet Technologies. ACM, 2013.

4. Zhaolong, Ning, Koji Okamura, and Chengming Li. A Joint Network Coding and Scheduling Algorithm in Wireless Network [C]. Proceedings of the Asia-Pacific Advanced Network, 2013, 36: 49-56.

5. Zhaolong, Ning, Koji Okamura, and Chengming Li. Power allocation for two-way relay channel in wireless bidirectional networks [C]. Communications (MICC), 2013 IEEE Malaysia International Conference on. IEEE, 2013:46-50.

6. Chengming Li, Koji Okamura. Trusted Storage Virtualization in Cloud Computing[C]. Proceedings of the Asia-Pacific Advanced Network, 2014, 38:123-136.

7. Chengming Li, Koji Okamura. Cluster-based In-networking Caching for Content-Centric Networking[J]. International Journal of Computer Science and Network Security, 2014, 14(11):1-9.

8. Chengming Li, Koji Okamura. A Game Theoretical Interest Forwarding for Cached Data in Content-Centric Networking [J]. Advances in Computer Science: an International Journal, 2014, 3(6):73-81.