Performance Analysis and Optimisation of In-network Caching for Information-Centric Future Internet

Submitted by Haozhe Wang to the University of Exeter

as a thesis for the degree of

Doctor of Philosophy in Computer Science

In January 2017

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature: .....

Abstract

The rapid development in wireless technologies and multimedia services has radically shifted the major function of the current Internet from host-centric communication to service-oriented content dissemination, resulting a mismatch between the protocol design and the current usage patterns. Motivated by this significant change, Information-Centric Networking (ICN), which has been attracting ever-increasing attention from the communication networks research community, has emerged as a new clean-slate networking paradigm for future Internet. Through identifying and routing data by unified names, ICN aims at providing natural support for efficient information retrieval over the Internet. As a crucial characteristic of ICN, in-network caching enables users to efficiently access popular contents from on-path routers equipped with ubiquitous caches, leading to the enhancement of the service quality and reduction of network loads.

Performance analysis and optimisation has been and continues to be key research interests of ICN. This thesis focuses on the development of efficient and accurate analytical models for the performance evaluation of ICN caching and the design of optimal caching management schemes under practical network configurations. This research starts with the proposition of a new analytical model for caching performance under the bursty multimedia traffic. The bursty characteristic is captured and the closed formulas for cache hit ratio are derived. To investigate the impact of topology and heterogeneous caching parameters on the performance, a comprehensive analytical model is developed to gain valuable insight into the caching performance with heterogeneous cache sizes, service intensity and content distribution under arbitrary topology. The accuracy of the proposed models is validated by comparing the analytical results with those obtained from extensive

simulation experiments. The analytical models are then used as cost-efficient tools to investigate the key network and content parameters on the performance of caching in ICN.

Bursty traffic and heterogeneous caching features have significant influence on the performance of ICN. Therefore, in order to obtain optimal performance results, a caching resource allocation scheme, which leverages the proposed model and targets at minimising the total traffic within the network and improving hit probability at the nodes, is proposed. The performance results reveal that the caching allocation scheme can achieve better caching performance and network resource utilisation than the default homogeneous and random caching allocation strategy. To attain a thorough understanding of the trade-off between the economic aspect and service quality, a cost-aware Quality-of-Service (QoS) optimisation caching mechanism is further designed aiming for cost-efficiency and QoS guarantee in ICN. A cost model is proposed to take into account installation and operation cost of ICN under a realistic ISP network scenario, and a QoS model is presented to formulate the service delay and delay jitter in the presence of heterogeneous service requirements and general probabilistic caching strategy. Numerical results show the effectiveness of the proposed mechanism in achieving better service quality and lower network cost.

In this thesis, the proposed analytical models are used to efficiently and accurately evaluate the performance of ICN and investigate the key performance metrics. Leveraging the insights discovered by the analytical models, the proposed caching management schemes are able to optimise and enhance the performance of ICN. To widen the outcomes achieved in the thesis, several interesting yet challenging research directions are pointed out.

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards my supervisor Prof. Geyong Min for the inspiration and continuous support of my PhD study. I have gained immensely from his wisdom, his critical thinking, his integral view on research, and most importantly, from his hard working attitude. He taught me to work with enthusiasm and curiosity, two fundamental qualities for a researcher. Without his guidance, constant feedback and financial support, this PhD would not have been achievable. Apart from his tremendous academic support, he has given me so many wonderful opportunities and always been there for me and allowed me to grow both personally and professionally. I am really glad to be associated with a great person like Prof. Geyong Min in my life.

My sincere thanks also goes to my second supervisor and friend Dr. Jia Hu for his dedicated help to my research, especially for sharing his insightful comments and encouragement, and also for widening my research from various perspectives. All the discussion and brainstorming sessions have been invaluable help and motivation for me at various stages of writing my thesis.

I thank my fellow research colleagues and labmates, Dr. Yulei Wu, Wang Miao, Noushin Najjari, Chengqiang Huang, Yuan Zuo, Xiangle Cheng, Yujia Zhu, Zhengxin Yu, Dr. Yang Liu, Dr. Lejun Chen, Fangming Zhong, for always standing by my side and sharing a great relationship as compassionate friends. All of you have enriched my life and been a great strength to me all throughout my PhD pursuit.

Last buy not the least, I would like to thank my family, my mother and father, from the bottom of my heart for all their love and encouragement and the sacrifices that they have made on my behalf. They have the kindness and the patience to tolerate my absence from the many responsibilities to complete my PhD study.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronyms / Abbreviations

ARTT Average Round Trip Time

BS Base Station

C-RAN Cloud Radio Access Network

CCN Content-Centric Networking

CDN Content Delivery Networks

CS Content Store

DHT Distribute Hash Table

DONA Data-Oriented Network Architecture

FIB Forwarding Information Base

FIFO First-In-First-Out

ICN Information-Centric Networking

IoT Internet of Things

IP Internet Protocol

IPTV Internet Protocol Television

IRM Independence Request Model

ISP Internet Service Provider

LCE Leave Copy Everywhere

LFU Least Frequently Used

LRU Least Recently Used

MMPP Markov-modulated Poisson process

MRU Most Recently Used

NDN Named Data Networking

NetInf Network of Information

NFV Network Functions Virtualisation

OTT Over-the-Top

P2P Peer-to-Peer

PARC Palo Alto Research Centre

PIT Pending Interest Table

PSIRP Publish-Subscribe Internet Routing Paradigm

QoE Quality of Experience

QoS Quality of Service

SBS Small Base Station

SLA Service Level Agreement

TCP Transmission Control Protocol

TLS Transport Layer Security

URI Uniform Resource Identifier

VoD Video on Demand

List of Publications

[1] Haozhe Wang, Geyong Min, Jia Hu, Wang Miao, and Nektarios Georgalas. Performance Evaluation of Information-Centric Networking for Multimedia Services. In *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pages 146–151. IEEE, mar 2016. (Best Paper Award)

[2] Haozhe Wang, Geyong Min, Jia Hu, Hao Yin, and Wang Miao. Caching of Content-Centric Networking under Bursty Content Requests. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2522–2527. IEEE, apr 2014.

[3] Huijie Yang, Jia Hu, Haozhe Wang, and Hao Wu. A Content Caching Strategy Based on Demand Estimate Function in Small Cell Networks. *Accepted by IEEE GreenCom 2017*, IEEE, jun 2017.

[4] Haozhe Wang, Jia Hu, Geyong Min, Wang Miao, and Nektarios Georgalas. Cost-Aware QoS Optimisation for Caching in Information-Centric Networking. *Submitted to IEEE GLOBECOM 2017.*

[5] Haozhe Wang, Jia Hu, Geyong Min, Wang Miao, and Nektarios Georgalas. Cost-Aware QoS Optimisation for Caching in Information-Centric Networking for Multi-services. *to be submitted to IEEE Transactions on Communications.*

[6] Haozhe Wang, Geyong Min, Jia Hu, Wang Miao, Nektarios Georgalas, and Yang Liu. Performance Analysis and Optimisation of Information-Centric Networking for Multimedia Services. *to be submitted to for IEEE/ACM Transactions on Networking.*

[7] Wang Miao, Geyong Min, Yulei Wu, Haozhe Wang, and Jia Hu. Performance Modelling and Analysis of Software-Defined Networking under Bursty Multimedia Traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12(5s):1–19, sep 2016.

[8] Wang Miao, Geyong Min, Yulei Wu, and Haozhe Wang. Performance Modelling of Preemption-Based Packet Scheduling for Data Plane in Software Defined Networks. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 60–65. IEEE, dec 2015. (Best Paper Award)

Chapter 1

Introduction

The major principles and architecture of current Internet were designed back in 1970s. From the beginning, the Internet runs on top of TCP/IP (Transmission Control Protocol and Internet Protocol) suite of protocols, which is the fundamental building block for the Internet's architecture. The Internet paradigm is a host-to-host based model, aiming at resource sharing and interconnection between hosts. All the Internet information exchanges are realised by packet switching which forwards packets using the destination address contained in the packet itself. This host-centric Internet has perfectly matched the early Internet usage, because the Internet applications or protocols at that time were all end-to-end communications such as the instance massage exchange, e-mail service and FTP file download.

The advent of the World Wide Web has radically changed Internet usage from host-to-host to service access and data retrieval. Multimedia services like Video on Demand (VoD), over-the-top (OTT) video, IPTV, on-line gaming and social networking represent the predominant usage of today's Internet. These services focus on the requested contents rather than host location, since the users tend to pay less attention to where the content is obtained, and are more interested in how fast and reliable the requested content can be accessed [1]. Therefore, the usage of Internet is switching from the host-centric to a content-oriented model. In order to follow these evolutions, more and more patches or overlay are added to the current Internet protocol stack, such as Peer-to-peer (P2P), Proxy, Content-Delivery Networks (CDNs), multi-homing, multicast, Mobile IP, etc., which made the Internet

become more and more complex. Even if the Internet has been able to address the challenges raised by new applications, the host-centric Internet is becoming too heavy to offer the best performance to the end-users. Due to the mismatch between the architecture and its current usage, the Internet is inadequate to deal with the problems of scalable content distribution, Quality-of-Experience (QoE), mobility, security, etc.

Facing these shortcomings, research communities are motivated to design Future Internet architectures. Information Centric Networking (ICN) paradigm was proposed to address the above-mentioned architectural problems. ICN treats content objects as the first class entity of the network architecture instead of physical location by identifying and routing contents by unified names. ICN is a receiver-driven networking model, where end-users only express their interests for a given content, then the entire network is in charge of routing the requests based on the content names only. The requests are forwarded towards the best content containers and delivering the contents through the reverse paths to the end-users. It decouples content from location to achieve a naturally content-centric architecture [2].

The rest of this chapter is organised as follows: Section 1.1 is devoted to the motivations and challenges of this research. The research aims and major contributions of this thesis are shown in Section 1.2. Finally, the outline of this thesis is presented in Section 1.3.

1.1 Motivations and Challenges

A notable advantage of ICN is to provide native support for scalable and highly efficient information retrieval. This is achieved through transparent and ubiquitous in-network caching. ICN routers are equipped with cache memory and enabled to cache frequently requested contents passing them. By means of caching, it is possible for ICN routers to serve requests of cached contents rather than forwarding the requests to the original server. In-network caching is considered as an integral part of ICN [3] to efficiently obtain content, alleviate congestion, reduce network load, and enhance Quality-of-Experience (QoE) perceived by users. However, in-network

caching differs from Web caching in which cache is transparent to applications and cached content is finer grained. This poses new challenging issues to be addressed, such as cache management and cache placement/replacement strategies, to efficiently utilise the limited cache resources. For example,

(i) ICN is still in its early age, thus it is necessary to have clear understanding of the factors that affect its potential performance before ICN can be widely adopted in commercial systems. A majority of the existing works on ICN caching were carried out via simulations. However, analytical model of ICN cache networks is indispensable for the understanding of the intrinsic behaviours and important metrics of caching performance. Due to the new receiver-driven transmission model and finer grained caching object, it becomes a challenge to analytically characterise the performance of ICN. Even when a single ICN node employing basic cache strategy (such as Least Recently Used) is considered, it is still difficult to model the data transferring process and quantify the caching performance. The modelling issue become even harder when a network of cache nodes are taken into account [4]. Therefore, it is imperative to develop a unified and accurate analytical model based on appropriate assumption under realistic scenarios.

(ii) As the dominant services of the Internet, various multimedia applications generate a substantial amount of traffic [5]. Due to the diversity of multimedia applications, data streams generated by these applications have various impacts on the performance of ICN. The data traffic generated by multimedia services are known to exhibit a bursty nature [6]. Because the data transmission in ICN is a receiver-driven process, the content request process of multimedia services also exhibits the bursty characteristics. The constant arrival rate or Poisson process used by most of the existing ICN cache studies [7–10] are inadequate for capturing traffic burstiness of multimedia services. Therefore, traffic pattern becomes a crucial factor, and a new analytical model is needed to capture the the bursty nature of content requests when we evaluate the performance of caching.

(iii) Although ICN caching has received considerable research attention, the existing works on ICN caching are mainly focused on a single cache node or special cache network topologies, such as cascade topology or tree topology. These special network topologies simplify the interoperability between cache nodes, therefore simplify the development and analysis of cache network models. But in ICN caches are ubiquitous and transmission paths are no longer fixed, the realistic topology of cache networks should be represented by arbitrary graphs [11] rather than the fixed parent-child relationships. Consequently, it is necessary to develop a comprehensive and accurate analytical model for understanding the performance of ICN caching under arbitrary network topology.

(iv) Various services differ considerably in their request rate, content size and popularity, this heterogeneity requires ICN to efficiently share and allocate cache resources among different services. This poses new challenges to be addressed for in-network cache management and optimal caching resource allocation. Therefore, an efficient caching management scheme which optimally allocates caching resources to each node for reducing the traffic load of services and improving the service quality is in demand.

(v) In-network caching can benefit both Internet Service Providers (ISPs) and end users. It reduces the traffic load of services for ISPs and alleviate network congestions. In the mean time, it improves the Quality of Service (QoS) for end users and reduces service latency. However, the economic aspect of ICN has received marginal consideration so far [12]. But It is vital to understand the potential cost-effectiveness of ICN before its wide deployment in ISP's network. As a result, it is timely to take the economic aspect into account in order to achieve both cost-efficiency and QoS guarantee in ICN caching.

1.2 Research Aims and Contributions

The research work reported in this thesis is focused on the analysis and optimisation of the performance of ICN serving multimedia services under realistic network environments. The main objectives of the research are:

(i) To develop cost effective and versatile analytical models for the performance evaluation of ICN with bursty multimedia traffic under arbitrary network topology.

(ii) To exploit the analytical models to investigate the impact of key performance metrics of ICN and develop an optimal cache allocation scheme under practical network configurations.

(iii) To investigate the association between the cost of ICN and QoS and find the optimal trade-off between them with a novel cost-aware caching mechanism in the presence of realistic caching strategy and heterogeneous network.

To achieve these objectives, the research develops new analytical models to investigate the performance of ICN in the presence of heterogeneous bursty traffic under arbitrary network topology. The accuracy of the proposed models is validated through the extensive comparison of the analytical performance results with those obtained from ICN simulation experiments. Novel caching schemes that leverage the analytical modes are proposed to improve the performance of ICN. The major contributions of this research are summarised as follows:

(i) A new analytical model is developed to evaluate the caching performance of ICN under bursty multimedia content requests. To capture the bursty characteristics of multimedia services, the developed model adopts the Markov-modulated Poisson process (MMPP) to represent the time-varying content requesting process. A thorough analysis has been conducted to evaluate the caching performance of a tree network under Zipf-like content distribution and Least Recently Used (LRU) caching strategy.

(ii) A comprehensive analytical model for gaining the insights of caching performance under arbitrary network topology with heterogeneous cache sizes, content distributions and request intensity is developed. The model derives the cache hit ratio at any node for various services as the key performance metric. The model is used to explore the impact of the cache size, content popularity distribution and content length on the performance of ICN caching.

(iii) A caching allocation scheme is proposed by leveraging the developed analytical model to achieve optimal caching resource allocations under heterogeneous bursty content requests and content distributions in realistic network configurations. The scheme obtains a way for deploying a limited budge of caches to the nodes targeting at minimising the total traffic within the network and improving the network performance. The performance results reveal that the proposed caching allocation scheme can achieve better caching performance and network resource utilisation than the default homogeneous and random caching allocation strategy.

(iv) A cost-aware QoS optimisation caching mechanism is proposed aiming for cost-efficiency and QoS guarantee in ICN. Two new models are designed to investigate the inner association between the economic aspect of network and service level agreement (SLA). The cost model takes into account installation and operation cost of ICN under a realistic ISP network scenario. The QoS model formulates the service delay and delay jitter in the presence of multiple services and general probabilistic caching strategy. The proposed mechanism jointly considers the two models and adopts a multi-objective optimisation method to find the optimal caching resource allocation. The performance results show that the cost-aware caching mechanism can attain an economical ICN and also guarantee the service performance.

1.3 Outline of the Thesis

The remainder of the thesis is organized as follows.

Chapter 2 introduces the background of ICN including the key features, architectures and caching strategies. A detailed literature review on the performance modelling and optimisation is then presented.

Chapter 3 presents analytical models for evaluating the ICN performance under bursty multimedia content requests. Particularly, the first model analyses the edge ICN nodes with LRU and Leave Copy Everywhere (LCE) caching strategies and Zipf popularity distribution; the second one considers the caching performance in a tree network. The models are used to explore the key network and content parameters of ICN caching.

Chapter 4 proposes a comprehensive analytical model for arbitrary ICN with heterogeneous bursty requests, content sizes and popularity distribution. Then a model-based caching allocation strategy is proposed to improve the ICN performance and achieve the optimal caching resource allocation.

Chapter 5 develops a cost-aware QoS optimisation caching mechanism to study the trade-off between Quality-of-Service (QoS) and cost of ICN. A cost model and a QoS model are designed to investigate the inner association between them.

Finally, Chapter 6 concludes the thesis and outlines the future research work.

Chapter 2

Background and Literature Review

For more than a decade, the inherent drawbacks of the current Internet have been calling for its revolutionary designs. The host-oriented communication model, which was designed for special data transmission in the early age of Internet, is causing troubles everywhere in nowadays content based various multimedia services. Consequently, Information Centric Network (ICN) is proposed to solve these problems. As the most permanent clean-slate approach for next generation Internet, ICN has attracted much attention from network researchers in the passed few years. Although ICN enters now into the main stream of networking research, it is still in its early stage. Many projects have been carried on in order to propose a concrete ICN solution to deploy it in reality. This chapter gives a general background knowledge and presents an in-depth review of the related research on caching performance modelling and enchantment of ICN.

The rest of this chapter is organized as follows. The background knowledge including the ICN architectures with important features is presented in Section 2.1. One of the most promising ICN architecture, Content-Centric Networking (CCN) is introduced in Section 2.2. In-network caching as one of the most important feature of ICN is detailed in Section 2.3. A detailed literature review on modelling of ICN performance as well as the caching allocation and optimisation is then presented in Section 2.4.

2.1 Information-Centric Networking (ICN)

The ICN paradigm focusing on redesigning the Future Internet architecture, placing named data rather than content locations (IP addresses) in the centre of the network design. The primary goal of ICN is to shift the current host-centric communication model of the Internet to a content-oriented model, in order to overcome the mismatch between the current usage patterns and the original design. In this section, the fundamental features of ICN common to most proposed designs are investigated, followed by the design principles of three ICN approaches.

2.1.1 Features of ICN

Many ICN architectures have been proposed by different research groups. Although the proposed architectures of ICN differ with respect to the details, they share several objectives and common components, including location-independent naming, in-network caching, name-based routing and content security.

Naming

The fundamental concept of the ICN is to evolve the host-based Internet architecture to a named-based one. ICN requires globally unique and location independent names [13, 14]. The content name is the only identifier of each content object, a globally unique name should be used for routing at a global level. Unlike the IP address which is relevant to the geography location, the naming of content should be location independent to route to the best content sources, such as the origin publisher or the on-path caches. Various kinds of naming schemes have been proposed, in order to achieve persistent naming and provide an efficient and scalable look-up for names of billions of contents. Among all those schemes, two categories of naming structures have been largely received: the hierarchical and the flat namespaces. Each naming scheme has the advantages and disadvantages in terms of security and scalability, thus, naming remains an open research issue. The hierarchical scheme adopts the similar ideas as the IP addresses [15] and the Uniform Resource Identifier (URI) [16], which can be aggregated into the prefixes and perform the longest match or shortest

match. Furthermore, it improves the scalability of the routing system. In some architecture, such as CCN [17], the hierarchical names are unique for routing but also human-readable. The flat naming scheme consists of identifying data with a flat label which is not suitable for aggregation. However, it can perform the Distribute Hash Table (DHT) based look-ups to improve the efficiency [18].

In-network Caching

One of the most remarkable features that differs ICN from the current Internet is introduction of the in-network caching capability. Every network element of the ICN can temporarily hold a copy of the data packets that traverse it. In virtue of the unique names, ICN is aware of the names content they are delivering, therefore requests for contents can be satisfied by any node holding a copy in its cache within the network, which reduces the response delay. In this way, ICN architectures naturally offer a reliable, scalable and application-independent network-level content delivery system, instead of leveraging any external storage resource such as Content Delivery Networks (CDN) and Peer-to-Peer (P2P) systems, commonly adopted in the current Internet architecture [11]. This ubiquitous in-network caching significantly improves the content retrieving efficiency and alleviates the traffic congestion.

Routing and Delivery

There are two phases in content retrieving in ICN, the request routing phase and the content delivery phase, respectively [13]. The request routing phase is realised in two manners: the naming resolution based routing and the direct name based routing. The naming resolution requires one or several centralised servers. These servers collect the content publication information and build a global view of the contents in the network. When an ICN router forwards a request, they resolve the content name in order to find the closest copy and deliver it to the receiver. Contrarily, the name based routing is directly performed in the ICN routers. Each router has a local routing table-liked forwarding information base which is filled by the content publication messages. In this case, the router addresses the request and routes the request to one or multiple potential data sources followed its own forwarding

strategies. The request routing methods heavily depends on how contents are named and published. During the content delivery phase, the providers do not send contents directly to the end-users, and the delivery is completely a receiver-driven process. The receiver sends requests to the network, and the provider can only response to the requests. After the source has received the request message, the data is routed back to the requester.

Security

Security in today's host-centric Internet is based on the protection of the end-to-end communication channel. For example, Transport Layer Security (TLS) is used to established end-to-end connectivity, and authenticated server is used in synchronous three-way authentication. Contrarily, ICN requires location independent and self-certifying security mechanism to enable ubiquitous in-network caching system [19]. Since requests can be satisfied by any network element, the architecture must secure the content itself rather than the communication path. Security is strictly related to naming, and the name should embed the information of data integrity and signature verification.

2.1.2 ICN Architectures

In the past few years many projects have been carried on in order to propose a concrete ICN solution to deploy it in reality. For example, key projects in the United States (CCN/NDN [17]/[20], and previously DONA [21]) and Europe (NetInf from the project 4WARD [22]/SAIL [23], PSIRP/PURSUIT [24][25] and COMET [26]) have worked on redesigning core network primitives to realise ICN architectures. Since its proposal, ICN has attracted significant interest in both academia and industry. This section introduces three projects related to ICN approaches at a high level with the purpose of providing a general understanding of them before going into a detailed discussion in the next section.

Data-Oriented Network Architecture (DONA): In DONA, contents are published into the network by the sources. Nodes that are authorized to serve

data, register to the resolution infrastructure consisting of resolution handlers (RHs). Requests (FIND packets) are routed by name toward the appropriate RH. Data is sent back in response, either through the reverse RH path, enabling caching, or over a more direct route. Content providers can perform a wildcard registration of their principal in the RH, so that queries can be directed to them without needing to register specific objects. It is also possible to register contents names before the content is created and made available. Register commands have expiry times. When the expiry time is reached, the registration needs to be renewed. The RH resolution infrastructure routes requests by name in a hierarchical fashion and tries to find a copy of the content closest to the client. DONA's anycast name resolution process allows clean support for network-imposed middleboxes (e.g., firewalls, proxies).

Publish-Subscribe Internet Routing Paradigm (PSIRP): In PSIRP, contents are also published into the network by the content sources. The publication belongs to a particular named scope. Receivers can subscribe to contents. The publications and subscriptions are matched by a rendezvous system. The subscription request specifies the scope identifier (SI) and the rendezvous identifier (RI) that together name the desired content. The identifiers are input to a matching procedure resulting in a forwarding identifier (FI), which is sent to the content source so that it can start forwarding data. The FI consists of a Bloom filter that routers use for selecting the interfaces on which to forward an content. This means that routers do not need to keep forwarding state. The use of Bloom filters results in a certain number of false positives; in this case this means forwarding on some interfaces where there are no receivers.

Network of Information (NetInf): In NetInf, there are two models for retrieving contents, via name resolution and via name-based routing, thereby allowing adaptation to different network environments. In NetInf, depending on the model used in the local network, sources publish contents by registering a name/locator binding with a name resolution service (NRS), or announcing routing information in a routing protocol. A NetInf node holding a copy of a content (including in-network caches and user terminals) can optionally register its copy with an NRS, thereby

adding a new name/locator binding. If an NRS is available, a receiver can first resolve a content name into a set of available locators and can subsequently retrieve a copy of the data from the "best" available source(s). Alternatively, the receiver can directly send out a GET request with the content name, which will be forwarded toward an available content copy using name-based routing. As soon as a copy is reached, the data will be returned to the receiver. The two models are merged in a hybrid resolution/routing approach where a global resolution system provides mappings in the form of routing hints that enable aggregation of routing information.

2.2 Content-Centric Networking

Content-Centric Networking (CCN) [17], one of the most well-known ICN paradigm designed by the Palo Alto Research Centre (PARC), elaborates some ideas and principles previously introduced by other ICN proposals. CCN proposes a clean-slate Internet architecture including naming and forwarding functionalities, taking care of security, content dissemination and mobility issues. The mainly design of CCN focuses on the network layer and transport layer of the ISO Internet model, maintaining the simplicity of IP in order to become the thin waist of the future Internet Architecture. Recently, projects focused on CCN called Named Data Networking (NDN) [20] has been funded by the NSF's Future Internet Architecture Program.

The research work described in this thesis is mainly focused on the architecture proposed in CCN, but the results can broadly apply in the context of general ICN architectures and other caching-related work. In this section, the CCN architecture is described.

CCN is driven by two types of packets: *Interest* and *Data* defined in [17], shown in Fig. 2.1. *Interest* packets are sent by users to ask for the named contents. Each *Interest* packet contains a content name based on a hierarchical structure, similar to a URI, with the prefix illustrating the global and organizational routing information, and a suffix providing the details of version and segmentation. *Data* packet, each of which contains a unit of data of the request content, is transmitted in response to the matching *Interest* and consumes that *Interest*.

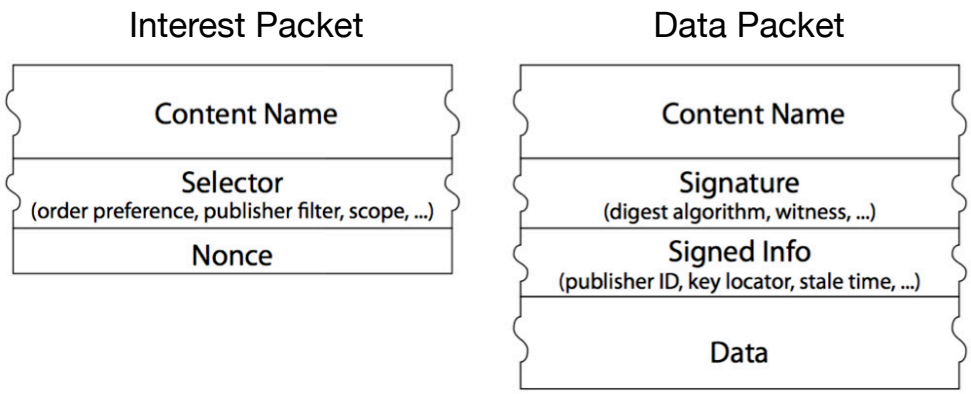


Fig. 2.1 CCN packet types. (Source: [17])

In CCN, the traditional IP routing table that is identified by IP addresses is no longer suitable and should be adapted to forward data, resolving the content names instead of the IP addresses. This adapted routing table is named as the Forwarding Information Base (FIB) in CCN and maps the names of content to the output interface(s) that should be used to forward *Interest* towards appropriated router(s). This router may have the right *Data* or knowledge about how to propagate the *Interest* to potential data sources.

The Content Store (CS), which serves as a local cache for contents, is checked to see whether the requested *Data* packet is already available. If this is the case, the CCN router will deliver that content from the CS and consume the *Interest* packet. The CS caches the *Data* passed through the router, so the subsequent content requests can be satisfied from it.

The Pending Interest Table (PIT) is a new structure in the CCN. It tracks the incoming interface(s) from which the pending *Interests* have arrived. The architecture of CCN router is illustrated in Fig. 2.2.

When one CCN router receives an *Interest* packet and the CS has no entry for the requested data, the PIT will be checked. If an *Interest* has no match in either the CS or PIT, a new PIT entry is created and the *Interest* is forwarded through one or more interfaces according to the FIB. When this CCN router receives the response *Data* packet, it looks up the PIT for the incoming interfaces and forwards the *Data* towards all the matching interfaces. After that, the entry for this content is removed from the PIT. In case there are multiple *Interests* for the same content,

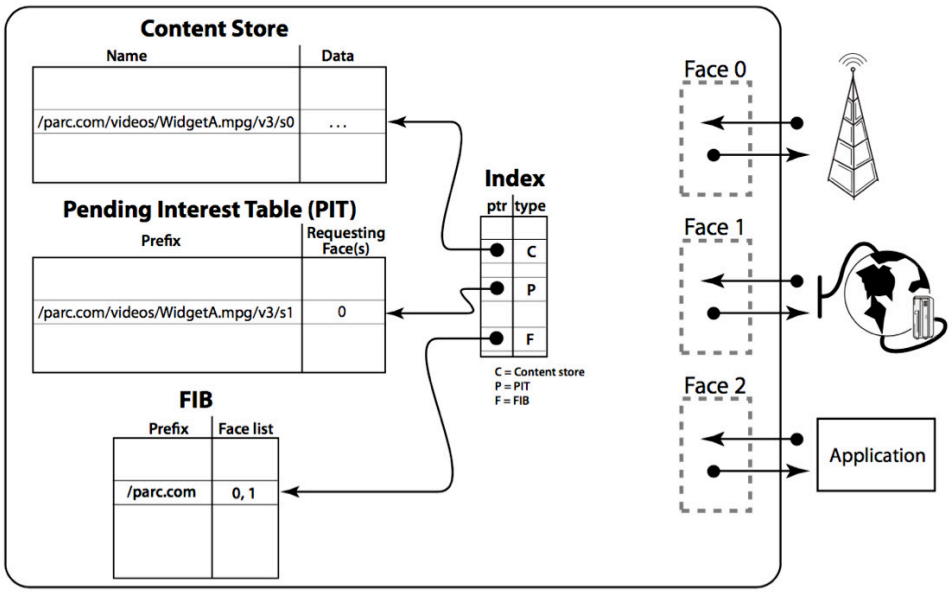


Fig. 2.2 The key components of CCN. (Source: [17])

the CCN router will only forward the first *Interest* once, and keeps track of all the interfaces from which the *Interest* packets are received. When the router receives the corresponding *Data* packet, it will forward that packet to all the interfaces tracked in the PIT. By so doing, the CCN network then naturally offers a native request aggregation and multicast function.

The working flow of CCN is illustrated in Fig. 2.3. In the content retrieving process, end-users request content by sending *Interest* packets for named data chunks to the CCN access edge router. If the requested *Data* is already stored in the CS, it is immediately sent back in reply to the user *Interest*. Otherwise, the CCN router inserts a reference to the interface from where the *Interest* came from in the PIT and forwards the *Interest* packet to certain interfaces according to the FIB and the forwarding strategy. In case multiple requests for the same content hit a CCN router that has already created a PIT entry, however which has not received the content yet, the PIT is appended with references to the latter requests. In this way, CCN naturally performs request aggregation, reducing redundant network loads. In the worst case, the requested *Data* of *Interest* packet is not cached at any CS of intermediate CCN routers, then the *Interest* packet will be forwarded to the original data repository. The *Data* packet sent in response to the *Interest* packet travels then

back according to PIT information, and PIT references are removed once interests are satisfied by sending *Data* on the corresponding interfaces.

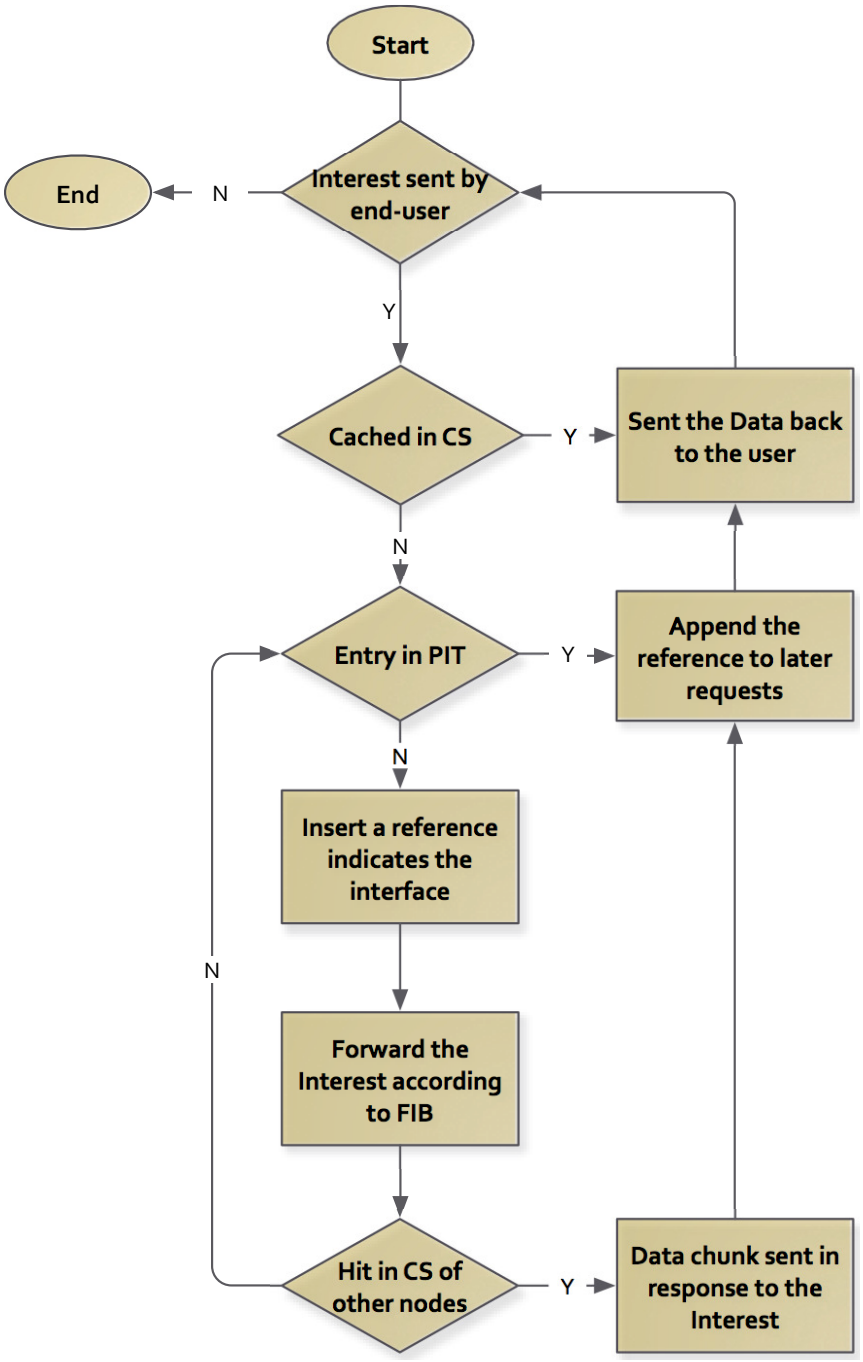


Fig. 2.3 The working flow of CCN.

2.3 In-network Caching

ICN architecture considers in-network caching as an integral part of the network. Caching plays a vital role in order to efficiently obtain the popular content, alleviate congestion, reduce redundant network load, and enhance the Quality-of-Experience

(QoE). Therefore, in-network caching of ICN has received considerable research attention in recent years [2, 3, 7, 8, 10, 11, 27–56]. Caching management that includes the cache decision and replacement polices has been in the centre of caching research. Before the surveys of related ICN caching work, the techniques and mechanism for caching performance are introduced in this section.

2.3.1 Cache decision policies

Cache decision policy determines which contents are to be stored at which cache nodes. Most of the ICNs use implicit cache placement policies, which means there is no global coordination proxy that is often used in Content delivery network (CDN).

In the existing cache of web proxies or CDN, it may be feasible to find the optimal content placement based on a priori knowledge of the number of caching nodes, network topology, routing distances, and traffic demand [57, 58]. However in ICN, caches are operated at networking layer (layer 3) and transparent to applications, so the cache solution of traditional web caches are no longer fit due to the lack of high level and aggregated vision. ICN is a flattening network, and the nodes need to exchange information as little as possible. Therefore, each cache node should be able to determine whether to cache the content by itself.

Moreover, cache nodes are arranged in an arbitrary topology and cache operation needs to obey the line-rate requirement. These factors have made traditional explicit cache decisions unsuitable for ICN, since they introduce more complexity and communication overheads. ICN needs simple yet effective cache decision policies. The simple Leave copy everywhere (LCE) [11, 17], which is the first caching policy proposed in ICN, copies the content on every node on a delivery path. There are other topology-based caching such as Leave Copy Down (LCD) [59] and Move Copy Down (MCD)[60]. When a cache hit happens, LCE caches the content on a delivery path towards to the edge of the network. Each cache node pushes the content one hop closer to the end-users, so after several requests, popular contents will be cached at the edge network. Similar to LCD, MCD also tends to cache the popular content to the edge network. The difference is that when a content cached on the current

node generates a hit, the node will delete the copy and a copy of the content will be moved to the downstream node. LCD and MCD aim to reduce the redundancy of caching.

Leave Copy Probabilistically (LCP) [31] also called cache with probability, is a classic probabilistic caching policy. A fixed probability p is given as a priori. Caching decision is individually made by each node, and the requested content is copied with probability p. LCP cannot distinguish the types of contents and treat all contents the same. Some modifications [61, 62] have been made for LCP with dynamic probability, i.e., each node has various probability p to decide whether to copy the content or not. When $p = 1$, LCP becomes to LCE. LCE is the default cache decision policy for some ICN architectures (such as CCN and NDN) [17] due to its simplicity and operating at line speed, and has been used as a benchmark for the performance evaluation of other caching policies.

2.3.2 Cache replacement policy

Limited cache sizes require the application of cache replacement policies which determine the content to be evicted from the cache when it is full. One of the fundamental operations is to determine the object to be removed from the cache when it is full. Cache replacement policies have received extensive research in the web cache literature [63]. However, the demand for low overhead operations is highlighted in ICN, which means cache replacement should perform as fast as possible, so complex replacement algorithms are not suitable for this purpose.

The replacement policies can be classified into two main categories, recency-based and frequency-based policies. The recency-based policies use recency as the main factor. The most widely analysed recency-based replacement policy is Least-Recently Used (LRU), which removes the content from the cache that has not been referenced for the longest period of time. In [64] authors provide an analytical characterisation of the miss probability of LRU under Poisson assumptions of content request arrivals. Other recency-based policies are more or less extensions of the LRU strategy. LRU policy has been successfully adopted in buffer management

and computer memory. One advantage of recency-based replacement polices is the temporal locality of reference which provides the ability to adapt to content popularity changes. Furthermore, LRU-like policies have low complexity and can be implemented at line speed. The LRU policy has been investigated in ICN caching in [9, 10, 32, 65].

Frequency-based policies use frequency as a main factor. Like the status of LRU in recency-based policies, Least-Frequently Used (LFU) is the most famous frequency-based policy. LFU is based on the fact that the popularities of different contents can be represented by the access frequency. The LFU applied in real-world is not perfect LFU that counts the global number of requests to a content. Instead, in-cache LFU that counts the request in the local cache only is more widely adopted. So in the LFU policy, a count statistics for the number of past requests to each content is recorded, and the least frequently requested content is evicted when the cache is full. The advantage of frequency-based strategies is that they can achieve an optimal cache hit for independent requests and avoid fluctuating contents storing and removing from the cache, if the popularity is monotonous to time. However, LFU is too complex to practically implement in ICN, due to the unlimited count statistics. It is worth noting that the research results in [27, 65] have shown that the performance of LRU policy is indistinguishable from some complex polices such as Most Recently Used (MRU) and Most Frequently Used (MFU), and LRU performance is sometimes as good as that of LFU.

In practical implementation, cache replacement policies and decision policies interwork with each other. In [66], a content replacement policy named Least-Value First (LVF) is proposed based on a utility function that takes into account two aspects, the popularity of content and the delay for retrieving that content, respectively. Authors of [67] argued that the caching policies should be chosen according to the location of a cache node in the network. For the edging cache nodes which are near or connect to the users, a replacement decision should be based on the number of hops to retrieve the content. For the caches within the network which connect to other cache nodes, a replacement decision should further consider the

number of router interfaces where a request for this content has arrived, in addition to the number of hops to retrieve the content.

2.3.3 Content characteristics

The key characteristics that are significant for caching performance of ICN are discussed.

Types of content. The Cisco Visual Networking Index published in 2016 classifies Internet traffic and forecasts global demand for the period 2015-2020 [5]. Some 82% of traffic is content retrieval, classified as web, email and data, file sharing, gaming and Internet video. Moreover, Internet video streaming and downloads are beginning to take a larger share of bandwidth and will grow to more than 69 percent of all consumer Internet traffic in 2017. Internet video can be further divided into short-form Internet video (for example, YouTube, video on Facebook and Twitter), long-form Internet video (for example, Hulu), live Internet video, Over-the-top video (for example, Netflix, HBO, Amazon Video), online video purchases and rentals, webcam viewing, and web-based video monitoring (excludes P2P video file downloads).

Catalog and content size The term catalog size describes the number of individual contents in a network. The total number of indexed web pages in the world is at least 4.55 billion [68]. ICN identifies contents by global unified name at chunk granularity, therefore the number of names objects with respect to ICN is estimated at 10^{23} orders of magnitude [69]. Deployability and scalability concerns may affect the performance and efficiency of caching policies [70].

User Generated Content (UGC) is dominated by YouTube. A recent study by Zhou et al. estimated that there are currently 5×10^8 YouTube videos of mean size 10 MB [71]. VoD catalogues, on the other hand, are much smaller. Inspection of various sites yields populations measured in thousands of movies, TV shows and trailers. The mean VoD object size is estimated around 100 MB.

Popularity distribution Caching performance depends crucially on the content popularity distributions. As usually assumed in the literature and pointed out in

[72] for Youtube, the content popularity can be characterised by Zipf or Zipf-like laws: the request rate $q(n)$ for the n^{th} most popular page is proportional to $1/n^{\alpha}$ for some α. The exponent parameter α of the Zipf distribution in the ICN literature varies between 1 and 2.5. Many caching strategies are based on content popularity distribution, such as LRU, LFU and more sophisticated variants derived from them.

2.4 Literature Review

Despite there has been a large amount of literature on web caching, e.g. [63], novel aspects of ICN technology (e.g., splitting content into chunks, arbitrary network topologies, line speed requirement, etc.) rekindle the research interest of in-network caching. Furthermore, the requests in ICN are highly correlated comparing to traditional caching, since a new request for a content will further triggers a series of requests for data chunks. In this section, the prior work related to performance modelling of ICN caching, cache resource management and performance optimisation, and economic implications of caching are reviewed. Furthermore, we also highlight that how this research can fill some of the most important gaps left in the existing work.

2.4.1 Caching Performance Modelling of ICN

A majority of the existing works on ICN caching were carried out via simulations [8, 27, 30, 32, 33, 44, 48, 73]. However, analytical modelling of ICN cache networks is indispensable for the understanding of the intrinsic behaviours and features of in-network caching. A unified and accurate analytical is in demand as a cost-effective tool to analyse the behaviour of ICN caching and further guide the design and optimisation of ICN.

Mechanisms for caching have been widely studied at the application level, mostly in the context of web applications. Research on current traffic patterns could shed additional light on the popularity characteristics of information today and thus to the possible benefits from widespread caching. For instance, a recent study has shown that web information popularity has changed during the past few years, affecting

application level caching performance [74]. Another issue is that when caching takes place inside the network, cache space management therefore becomes crucial for the network, and recent works, albeit based on simplified traffic models, have indicated that intelligent schemes can substantially improve performance [31, 75, 76]. However, in today's service-oriented network, different types of multimedia traffic will compete for the same caching space, therefore the simplified traffic models such as Poisson cannot be used to accurately quantify the characteristics of content requests in ICN. Most of the existing in-network cache studies consider the simplified traffic models such as constant arrival rate and Poisson process [7–10], which fail to capture the bursty nature of content requests in ICN.

Moreover, most of the ICN caching studies did not consider a realistic topology, thus ignored the correlations among ICN nodes. In [9, 10, 31, 77], the performance of in-network caching is evaluated in simple topologies such as simple cascade topology, diamond topology or tree topologies. In contrast to the aforementioned network topologies, the realistic ICN topology of cache networks should be represented by arbitrary graphs [11] or scale-free topologies [78–80] due to the mobility and on-path caching features of ICN. [81] proposes an approximate model to investigate general cache networks under Poisson content request process. The impact of topology on the performance of caching has been investigated in [75] and proposed a caching strategy exploiting the concept of betweenness centrality to select the nodes having the highest probability of getting a cache hit along the content delivery path.

The arbitrary topology also brings new constrain for performance modelling. Existing caching works [10, 82, 83] have adopted the assumption of independent content requests, following the Independent Reference Model (IRM). However, content requests in ICN may correspond to the retrieval of multiple chunks of a content, which leads to the requests being correlated to each other. Furthermore, due to the arbitrary topology, a node may receive the missing content requests forwarded from its neighbours, thus the assumption of IRM arrivals may not be justified. Authors of [84] have questioned the validity of the IRM with regard to object naming granularity. Therefore, an analytical model that can capture the bursty nature of content requests

and in the mean time can evaluate the performance of ICN caching under arbitrary network topology is in demand.

In this work, instead of the simple traffic assumption, we take into account the bursty traffic arisen by multimedia services and heterogeneous content popularity distribution to investigate the cache performance under arbitrary topology through developing a comprehensive analytical model. Furthermore, the developed analytical model is used to explore the key factors that have impact on the caching performance.

2.4.2 Caching Management and Allocation Optimisation

A good caching strategy can reduce the traffic load of services and improve the service quality. Caching strategies have been widely studied in [3, 28–30, 32, 33, 36, 37, 40, 44, 46, 47, 49, 75, 85–95] for the improvement of caching performance. Recently, cache allocation optimisation has received much attention.

Cache allocation problem has been studied in [57, 96–99] for applications of CDN. This off-path caching optimal placement problem has been solved through optimisation methods [100, 101]. However, one fundamental challenge in ICN is that under a fixed total caching resource limit, how to allocate the caching resource to on-path nodes with heterogeneous traffic and content distribution, so that the cache performance of the whole system can be optimised. This is indeed a network dimensioning problem under a given budget. Cache space allocation needs to take account of network topology and traffic demands. The results in [44] show that the cache size determined based on centrality can only have marginal effect on performance improvement, and this marginal improvement can even be achieved with a much simpler allocation scheme—degree based allocation, i.e., the cache capacity allocated to a node is proportional to its node degree. In [31], it was shown that allocating more storage resource to edge routers rather than core routers is beneficial for performance improvement. These studies together show that when network topology is considered, a resource allocation scheme should not simply be based on the node's static topological centrality, but should be based on the distance from the caching node and users as well as its serviced user population. Meanwhile,

since simple cache space allocation alone cannot achieve considerable performance improvement, it is necessary to integrate the cache space allocation with object placement and object query policies. A heuristic algorithm was proposed in [85] to find a near-optimal cache allocation. Extensive simulations are conducted to observe the key factors for caching performance. A heterogeneous cache allocation was proposed in [44] based on degree centrality which allocates caches proportionally to the number of links, and results show that only limited performance gain is achieved under heterogeneously sized caches. Because caching mechanism based on graph-related centrality properties fails to take the traffic patterns into account, a cache size optimisation scheme is presented in [49] to overcome the problem. The proposed algorithm determines the cache size based on the importance of a router by using manifold learning to analyse traffic distribution and user behaviours. [86] presented a probabilistic approach that allocates resources along the path according to the content flow, aiming at reducing the caching redundancy and improve fairness for different contents.

In this research, a cache allocation scheme, which leverages the knowledge discovered by the comprehensive analytical model, is proposed to achieve the global optimal allocation for a realistic ICN configuration with heterogeneous bursty content requests and content popularity distribution.

2.4.3 Economic Impact on Caching

The previous optimising work mainly focus on enhancing caching efficiency [8, 49, 86, 87, 102]. Despite the potential technical benefits investigated in the previous sections, ICN has so far remained in the research community, unlike CDNs or web caches that have seen a large deployment. The reason behind this is that the economic incentives are not clear enough to persuade leading ISPs to transform their network architectures. Nevertheless, the economic aspect of ICN has received marginal consideration so far [91]. Therefore, it is crucial to investigate the monetary impact on the cache deployment in ICN.

The economic issues have been studied in the caching mechanism under peer-to-peer (P2P) and web caches [12, 53, 103–105]. However in the context of ICN, caches are formed as an arbitrary network, and performance are influenced by multiple services due to caches are shared by and transparent to the applications. Therefore, the cost of cache network differs from caching of the Internet.

To address this problem, game theory is widely adopted in many works. A game-based pricing model that provides economic incentives for caching and sharing content in ICN has been proposed in [106]. A model has been developed in [12] for the assessment of the economic incentives of different network players to widely deploy storage for ICN caching. The economic feasibility of ICN has been evaluated in [107] by using the the two-sided markets analysis, and compared with client-server, peer-to-peer and CDN models. Kocak et al. [37] used game theory to study a price-convex demand-response pricing model. These works focus on the interaction between different players in the network and does not consider the cache design and performance. Therefore, the cost and performance are treated as orthogonal problems. Araldo et al. [92] show that considering the traffic costs of a network operator leads to cache allocations that are suboptimal in terms of hit rate. In [91], two models were proposed to investigate the impact of content retrieval cost on the caching design, but the authors make strong assumptions for the models such as unrealistic content requests and simplified topology.

In order to foster the practical deployment of ICN in next-generation network by service providers, it becomes crucial to understand the performance and cost bounds of ICN. To this end, this research investigates the inner association between the performance and cost of ICN. The results are useful for network operators to select the most appropriate settings for different services requirements.

2.5 Summary

In this chapter, the background knowledge of ICN, including the key features and existing architectures have been investigated. Furthermore, a survey of the in-networking caching research has been presented. More specifically, a description

of the existing caching strategies and key characteristics of caching performance has been provided. At last, a comprehensive literature review of state-of-the-art cache performance modelling and optimisation has been presented, indicating the new challenges in ICN caching and showing how this research can address these challenges.

Chapter 3

Performance Analysis of ICN Caching for Multimedia Services

3.1 Introduction

The rapid development in multimedia services has given rise to new requirements for the Internet, such as supporting billions of devices and transmitting huge amount of multimedia contents in real-time. Due to the diversity of multimedia applications, traffic generated by these applications have various impacts on the performance of ICN. Therefore, traffic pattern becomes a crucial factor when conducting performance evaluation of caching. As the dominant type of network traffic nowadays, the data streams generated by multimedia services are known to exhibit the bursty nature [6]. Because the data transmission in ICN is a receiver-driven process (i.e., the communication is driven by receivers sending requests), the content request process of multimedia services also exhibits the bursty characteristics, which needs to be taken into account when evaluating the performance of ICN. However, most of the existing ICN cache studies consider the simplified content request models such as constant arrival rate and Poisson process [7–10], which fail to capture the bursty nature of content requests in ICN. This arises an gap as how to characterise the bursty nature of ICN traffic.

In-network caching is considered as an integral part of ICN [3] to efficiently obtain content, alleviate congestion, reduce network load, and enhance users' Quality-of-

Experience (QoE). However, in-network caching differs from Web caching in which cache is transparent to applications and content to be cached is finer grained. This poses new challenging issues to be addressed, such as cache management and cache placement/replacement strategies, to efficiently utilise the limited cache.

A unified and accurate analytical model of caching can be used to evaluate the performance of ICN and to further guide the design of optimised ICN protocols. The weakness of the existing caching studies of ICN reflect in the assumption of traffic pattern. Most of the existing studies consider the simplified traffic models such as constant arrival rate and Poisson process [7–10], which fail to capture the bursty nature of content requests in ICN. To fill in this gap, this chapter aims to investigate the performance issues of caching in ICN routers under the bursty content requests. To this end, a new analytical model is developed as a cost-effective performance tool to investigate the caching of ICN, especially the *cache hit ratio* of routers. The analytical model contains two main parts, namely the model for the single router and the model for tree network. The developed analytical model adopts the Markov-modulated Poisson process (MMPP) to capture the bursty nature of the content requests. The Least-Recently-Used (LRU) replacement policy is taken into account because it has been applied successfully in many caching systems [63]. The accuracy of the analytical model is validated through extensive simulation experiments. Finally, the analytical model is used to evaluate the impact of key metrics, such as the cache size, content size and content popularity on the performance of caching in ICN.

The remainder of this chapter is organized as follows. Section 3.2 presents the system parameters and the model of bursty content requests, and then derives the analytical model for caching of ICN. Section 3.3 validates the analytical model and carries out the performance analysis. Section 3.4 concludes the chapter

3.2 Analytical Model

This section presents the system model and methodology used to develop the analytical models for caching performance of ICN under bursty multimedia traffic. Firstly, system parameters and assumptions are introduced. Secondly, MMPP is leveraged

to model the bursty content requests of multimedia services. Then, an analytical model is developed to investigate the caching performance of ICN routers in the presence of bursty content requests.

3.2.1 System parameters

The system parameters are introduced in this subsection. Table 3.1 provides a summary of the notations used in the derivation of the model, and the notations are explained in details below:

(i) A total of $\mathbb{O} = \{content_1, \ldots, content_O\}$ different contents are considered in the model. Contents are formed into K sets, with each set representing one type of services, and each set contains $m = O/K$ different contents. Within each service type, k, the m contents have the same popularity.

(ii) The popularity of contents belonged to different services follows the Zipf distribution, which is widely used for characterising the content popularity [8, 9, 32, 108], because it has been pointed out in [72] that the popularity of real web content accesses was observed following the Zipf distribution. As a result, contents in service k (i.e., the k-th most popular content) are requested with the probability $q_k = f(\alpha, k) = \frac{1/k^\alpha}{\sum_{i=1}^{K} 1/i^\alpha} = \frac{D}{k^\alpha}$, $k \geq 1$, where $\alpha \geq 1$ is the value of the exponent characterising the Zipf distribution, $1/D = \sum_{i=1}^{K} 1/i^\alpha$.

(iii) Content with file size F is segmented into chunks. The content size, F, follows a geometrical distribution with δ chunks on average, i.e. $\mathbb{P}(F = s) = \frac{1}{\delta}(1 - \frac{1}{\delta})^{s-1},\ s = 1, 2, 3, \ldots$

(iv) Since caching operation needs to be line-rate and services running on caches are diverse, complex cache coordination policy are not suitable due to the high complexity and communication overhead [11]. The model considers Lease Recently Used (LRU) policy cache replacing policy on each node, which has low complexity and can be implemented at line speed. The LRU policy has been used in [9, 10, 32, 65].

Table 3.1 Summary of system parameters

Parameter	Meaning
$\mathbb{O}$	Total number of different content items
m	Number of different contents in each class
K	Number of class
C	Content Store (Cache) size in number of chunks
δ	Average content size in number of chunks
F	File size in number of chunks (geometrically distributed)
$h_k, h_k(i)$	Cache hit ratio for contents in class k at the 1^{st}, i^{th} level
H	Global cache hit ratio
$q_k, q_k(i)$	Probability of requests for contents in class k at the 1^{st}, i^{th} level
α	Zipf exponent characterizing the distribution
$\lambda_{tot}, \lambda_{tot}(i)$	Total content request rate at the 1^{st}, i^{th} level of cache
$\lambda_k, \lambda_k(i)$	Mean request rate for contents in class k at the 1^{st}, i^{th} level of cache
λ_{sk}	Initial content arrival rate of users
Q_k	Infinitesimal generator of requests for contents in class k
Λ_k	Request rate matrix of contents in class k
$\boldsymbol{\lambda}$	Arrival rate vector for all classes. Diagonal values of Λ
N	Number of ICN routers in the network
τ_k	Time interval between two requests for the same chunk that generate a miss
L_i	Link delay between the 1^{st} level and the i^{th} level of routers

3.2.2 Modelling of Bursty Content Requests

Content requests of multimedia applications in ICN exist a bursty nature which can be represented by MMPP. The MMPP is a doubly stochastic process with the arrival rate varying according to an irreducible continuous-time Markov chain [109]. It is capable of modeling the bursty content requests because it can capture the time-varying arrival rate of different services. The arrival process of each type of multimedia services is modeled by a special case of MMPP called Interrupted Poisson Process (IPP). IPP_k is adopted to model the content requests for type k service, and is characterized by an infinitesimal generator Q_k of the underlying Markov process and a rate matrix Λ_k. Q_k and Λ_k are given by

$$Q_k = \begin{bmatrix} -\sigma_{k1} & \sigma_{k1} \\ \sigma_{k2} & -\sigma_{k2} \end{bmatrix}, \ \Lambda_k = \begin{bmatrix} \lambda_{sk} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.1}$$

where σ_{k1} denotes the probability that the state of the Markov chain transits from state 1 to 2, and σ_{k2} is the transition rate from state 2 to 1. λ_{sk} is the request

arrival rate at state 1 when a content in k is requested, while in state 2, no content is requested. According to Q_k and Λ_k, the mean arrival rate for contents in class k, λ_k, can be calculated as

$$\lambda_k = \frac{\sigma_{k1} \times 0 + \sigma_{k2} \times \lambda_{sk}}{\sigma_{k1} + \sigma_{k2}} \tag{3.2}$$

The total content request process is represented by the superposition of K input IPPs, which is again an MMPP, as the MMPP is closed under the superposition operations. The generator Q and the rate matrix Λ of the composite MMPP are calculated from the individual generators Q_k and rate matrices Λ_k as follows

$$\begin{aligned} Q &= Q_1 \oplus Q_2 \oplus \cdots \oplus Q_K, \\ \Lambda &= \Lambda_1 \oplus \Lambda_2 \oplus \cdots \oplus \Lambda_K. \end{aligned} \tag{3.3}$$

where $\oplus$ denotes the Kronecker-sum. The composite Q and Λ of the superposed MMPP are $K^2 \times K^2$ matrices and can be written as

$$Q = \begin{bmatrix} -\sigma_1 & \sigma_{12} & \cdots & \sigma_{1K^2} \\ \sigma_{21} & -\sigma_2 & \cdots & \sigma_{2K^2} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{K^21} & \sigma_{K^22} & \cdots & -\sigma_{K^2} \end{bmatrix}, \ \sigma_i = \sum_{\substack{j=1 \\ j \neq i}}^{K^2} \sigma_i j,$$

$$\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_{K^2}), \boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_{K^2})^T \tag{3.4}$$

The mean arrival rate, λ_{tot}, of the composite MMPP can be derived from the steady-state vector $\boldsymbol{\pi}$ and the arrival rate vector $\boldsymbol{\lambda}$, as

$$\lambda_{tot} = \boldsymbol{\pi}\boldsymbol{\lambda}. \tag{3.5}$$

where $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi}Q = 0$, and $\boldsymbol{\pi}\boldsymbol{e} = 1$.

In the model, contents are split into chunks that are identified by a unique name and are stored in one or more repositories in the network. The arrival requests with rate λ_k for contents in type k are equally chosen among the m different contents in the given type of service. This arrival process can be represented by Q_k and Λ_k. A content request yields the request for the first chunk of that content. Once a chunk is received, the request for the next chunk is sent continuously until the reception of the last chunk of that content.

3.2.3 Modelling of Caching Performance under Bursty Content requests

This subsection presents the analytical model of the *cache hit ratio* of requests for contents in class $k = 1, \ldots, K$ under the aforementioned bursty content requests.

Given the MMPP content request with intensity λ_k for contents in class k, the average content size, δ, and the cache size, C, the stationary *cache hit ratio*, h_k, is given by

$$h_k = h_k(1) = 1 - \beta_k e^{-u_k \tau_k} - (1 - \beta_k) e^{-v_k \tau_k} \tag{3.6}$$

where the parameters u_k and v_k are the two eigenvalues of $(\Lambda_k - Q_k)$, and can be written as

$$\begin{aligned} u_k &= \frac{\frac{\lambda_k}{m} + \sigma_{k1} + \sigma_{k2} - d_k}{2} \\ v_k &= \frac{\frac{\lambda_k}{m} + \sigma_{k1} + \sigma_{k2} + d_k}{2} \end{aligned} \tag{3.7}$$

with

$$d_k = \sqrt{\left(\frac{\lambda_k}{m} + \sigma_{k1} - \sigma_{k2}\right)^2 + 4\sigma_{k1}\sigma_{k2}}$$

β_k is the transformation parameter from the *IPP* to the hyperexponential distribution, and is given in [109]

$$\beta_k = \frac{\frac{\lambda_k}{m} - v_k}{u_k - v_k} \tag{3.8}$$

The inter-arrival time, τ_k, denotes the interval in which there are more than C requests arrived between two subsequent requests of the same chunk in class k. τ_k is determined by the cache size, C, and the mean arrival rate of chunk requests, g, between an open interval $(0, t)$. The mean arrival rate, g, can be written as

$$g = \Gamma(1 - \frac{1}{\alpha})^{\alpha}(\frac{\lambda_{tot} D}{2})m^{\alpha-1}\delta^{\alpha} \tag{3.9}$$

Thus, the inter-arrival time, τ_k, is given by

$$\tau_k = C^{\alpha}/g \tag{3.10}$$

Proof. To determine the quantity of g, τ_k, and h_k, some notations are defined for the sake of clarity.

$R^i_{jk}(u, t)$ represents the number of requests for chunk i of content j in class k in an open interval (u, t), and $R^i_{jk}(u, t) \sim \text{MMPP}(\frac{\lambda_k}{m})$. Furthermore, $N^i_{jk}(u, t) = \mathbb{1}\{R^i_{jk}(u, t) > 0\}$ is a Bernoulli variable, meaning that at least one chunk i in class k is requested in the open interval (u, t). $S(u, t)$ denotes the number of chunks that requested in the interval (u, t) over all classes, and can be written as

$$S(u, t) = \sum_{k=1}^{K}\sum_{j=1}^{m_k}\sum_{i=1}^{F_j} N^i_{jk}(u, t) \tag{3.11}$$

Similarly, $S^i_k(u, t)$ denotes the number of chunks requested in the interval (u, t) that are different from chunk i in class k, and is given by

$$S^i_k(u, t) = \sum_{k'=1}^{K}\sum_{j'=1}^{m_{k'}}\sum_{i' \neq i}^{F_{j'}} N^{i'}_{jk'}(u, t) \tag{3.12}$$

To derive the inter-arrival time τ_k, the mean arrival rate of chunk requests, g, is to be examined first. For a given chunk in class k, the expect number of requests,

$\mathbb{E}[N^i_{jk}(u,t)]$, for that chunk in the open interval (u,t), is given by

$$\begin{aligned}\mathbb{E}[N^i_{jk}(u,t)] &= \mathbb{P}(N^i_{jk}(u,t)=1) = \mathbb{P}(R^i_{jk}(u,t)>0)\\ &= \beta_k e^{-u_k(t-u)} + (1-\beta_k)e^{-v_k(t-u)} \end{aligned} \tag{3.13}$$

This is due to the fact that contents with the same popularity are requested uniformly with the probability $1/m$. Furthermore, the content request process, $\{R^i_{jk}(u,t)>0\}$, is an MMPP with rate λ_k/m and chunks are indistinguishable.

The mean arrival rate of chunk requests, g, can be derived as $g = \lim_{t\to\infty} \frac{\mathbb{E}[S(0,t)]^\alpha}{t}$. In order to compute $\lim_{t\to\infty}\mathbb{E}[S(0,t)]$, the lower bound is computed as $K \to \infty$ [108],

$$\begin{aligned}\mathbb{E}[S(0,t)] &= \sum_{k=1}^{\infty}\sum_{j=1}^{m_k}\mathbb{E}[\sum_{i=1}^{F_j} N^i_{jk}(0,t)]\\ &= m\delta\sum_{k=1}^{\infty}(\beta_k e^{-u_k t} + (1-\beta_k)e^{-v_k t})\\ &= m\delta\sum_{k=1}^{\infty}\int_k^{k+1}(\beta_k e^{-u_k t} + (1-\beta_k)e^{-v_k t})dy\\ &\geq m\delta\int_0^{\infty}(\beta_k e^{-u_y t})dy\\ &= m\delta\int_0^{\infty}(\frac{\frac{\lambda_y}{m}-v_y}{u_y-v_y}e^{-\frac{\frac{\lambda_{tot}D}{my^\alpha}+\sigma_{y1}+\sigma_{y2}-d_y}{2}t})dy\\ &= \frac{m\delta}{\alpha}(\frac{\lambda_{tot}Dt}{2m})^{\frac{1}{\alpha}}\int_0^{\frac{\lambda_{tot}Dt}{m}}(2z - t(\sqrt{\sigma_{z1}}-\sqrt{\sigma_{z2}})^2)^{-1-\frac{1}{\alpha}}e^{-z}dz\\ &\sim \Gamma(1-\frac{1}{\alpha})(\frac{\lambda_{tot}Dt}{2})^{\frac{1}{\alpha}}m^{1-\frac{1}{\alpha}}\delta,\ t\to\infty \end{aligned} \tag{3.14}$$

Similarly, the upper bound can be derived as,

$$\begin{aligned}
\mathbb{E}[S(0,t)] &= m\delta(\beta_k e^{-uDt/m} + (1-\beta_k)e^{-vDt/m}) + \\
&\quad m\delta \sum_{k=1}^{\infty} \int_{k}^{k+1} (\beta_k e^{-u_k t} + (1-\beta_k)e^{-v_k t})dy \\
&= m\delta(\beta_k e^{-uDt/m} + (1-\beta_k)e^{-vDt/m}) + \\
&\quad m\delta \int_{0}^{\infty} (\frac{\frac{\lambda_y}{m} - v_y}{u_y - v_y} e^{-\frac{\frac{\lambda_{tot}D}{my^{\alpha}} + \sigma_{y1} + \sigma_{y2} - d_y}{2} t})dy \\
&= m\delta(\beta_k e^{-uDt/m} + (1-\beta_k)e^{-vDt/m}) + \frac{m\delta}{\alpha}(\frac{\lambda_{tot}Dt}{2m})^{\frac{1}{\alpha}} \times \\
&\quad \int_{0}^{\lambda_{tot}Dt/m} (2z - t(\sqrt{\sigma_{z1}} - \sqrt{\sigma_{z2}})^2)^{-1-\frac{1}{\alpha}} e^{-z} dz \\
&\sim m\delta + \Gamma(1 - \frac{1}{\alpha})(\lambda_{tot}Dt/2)^{\frac{1}{\alpha}} m^{1-\frac{1}{\alpha}}\delta, \ t \to \infty
\end{aligned} \tag{3.15}$$

As a consequence, both the upper and lower bounds of the mean arrival rate of chunk requests, g, coincide with

$$g = \lim_{t\to\infty} \frac{\mathbb{E}[S(0,t)]^{\alpha}}{t} = \Gamma(1 - \frac{1}{\alpha})^{\alpha}(\frac{\lambda_{tot}D}{2})m^{\alpha-1}\delta^{\alpha} \tag{3.16}$$

After g is derived, the inter-arrival τ_k can be calculated as follows. By considering the miss of request for a chunk, due to the property of LRU policy, a request for a given chunk generates a cache miss when more than C different chunks are requested after its previous request. Since the number of arrival chunks is larger than the the cache size, C, the given chunk has been removed from the cache before the arrival of the new request. The requests for chunk i of a content in class k are denoted by moments $\{\tau_n^{(ik)}\}_{n\geq 0}$, with increments $\{\tau_{n+1}^{(ik)} > \tau_n^{(ik)}\}_{n\geq 0}$ and $\tau_0^{(ik)} = 0$. According to the property of MMPP, sequence $\{\tau_n^{(ik)}\}, i \geq 1, k \geq 1$ is mutually independent, thus, the inter-arrival $(\tau_n^{(ik)}, \tau_{n+1}^{(ik)})$ can be presents as $\tau^{(ik)}$, for any $n \geq 0$. For brevity, we define a sequence of indicator variables as

$$B_n^{(ik)}(x) = \mathbb{1}[\text{a miss occurs at moment } \tau_n^{(ik)}] \tag{3.17}$$

and $\{B_n^{(ik)}(x)\}$ is a Bernoulli sequence.

Due to the property of LRU, the request for chunk i is moved to the front of the LRU list at each moment, τ_n^{ik}. If more than C requests of different chunks arrive within $(\tau_n^{ik}, \tau_{n+1}^{ik})$, a miss will be generated, as a result, the Bernoulli sequence, $\{B_n^{(ik)}(x)\}$, can be rewritten as

$$B_n^{(ik)}(x) = \mathbb{1}\{S_k^i(\tau_n^{(ik)}, \tau_{n+1}^{(ik)}) \geq C\} \tag{3.18}$$

Therefore, the miss probability of a reqeust can be obtained by $\mathbb{P}[B_1^{(ik)}(x) = 1] = \mathbb{P}[S_k^i(\tau_{n-1}^{(ik)}, \tau_n^{(ik)}) \geq C] = \mathbb{P}[S_k^i(\tau^{(ik)}) \geq C]$. Thus, the probability of a miss occurring at moments $\{\tau_0^{(ik)}, \tau_1^{(ik)}, \ldots, \tau_n^{(ik)}\}$ is given by,

$$\begin{aligned}
&\mathbb{P}[B_1^{(ik)}(x) = 1, B_2^{(ik)}(x) = 1, \cdots, B_n^{(ik)}(x) = 1] \\
&= \mathbb{P}[S_k^i(\tau_0^{(ik)}, \tau_1^{(ik)}) \geq C, \cdots, S_k^i(\tau_{n-1}^{(ik)}, \tau_k^{(ik)}) \geq C] \\
&= \mathbb{P}[S_k^i(\tau^{(ik)}) \geq C]^n
\end{aligned} \tag{3.19}$$

Finally, the inter-arrival time τ_k, in which more than C chunks arrive between two subsequent requests of the same chunk in class k, can be derived from the miss probability at any arrival moment [108], $\tau_n^{(ik)}$, and is given by

$$\begin{aligned}
\lim_{C\to\infty} \mathcal{P}[S_k^i(\tau_n^{(ik)}, \tau_{n+1}^{(ik)}) \geq C] &= \lim_{C\to\infty} \mathcal{P}[\tau^{(ik)} \geq C^\alpha/g] \\
&= \beta_k e^{-u_k C^\alpha/g} + (1-\beta_k) e^{-v_k C^\alpha/g}
\end{aligned} \tag{3.20}$$

where $g = \Gamma(1-\frac{1}{\alpha})^\alpha (\frac{\lambda_{tot} D}{2}) m^{\alpha-1} \delta^\alpha$

Thus, the *cache hit ratio* of the ICN routers with the bursty (MMPP) content request is given by

$$h_k = 1 - \beta_k e^{-u_k \tau_k} - (1-\beta_k) e^{-v_k \tau_k}$$

where $\tau_k = C^\alpha/g$, which, in conjunction with Eq. (3.16), concludes the proof.

3.2.4 Modelling of Caching Performance with Tree Topology

In this subsection, the analytical model developed in the above subsection is applied to the study of the tree topology illustrated in Fig. 3.1, and to derive the closed-form expression for cache hit ratios at each level. The binary tree is a reasonable topology for ICN, because with the repository at the root, it uses the single path routing, and the shortest path is always the path linked to the parent. Moreover, the advantages of requests aggregation in ICN can be better revealed in a tree topology.

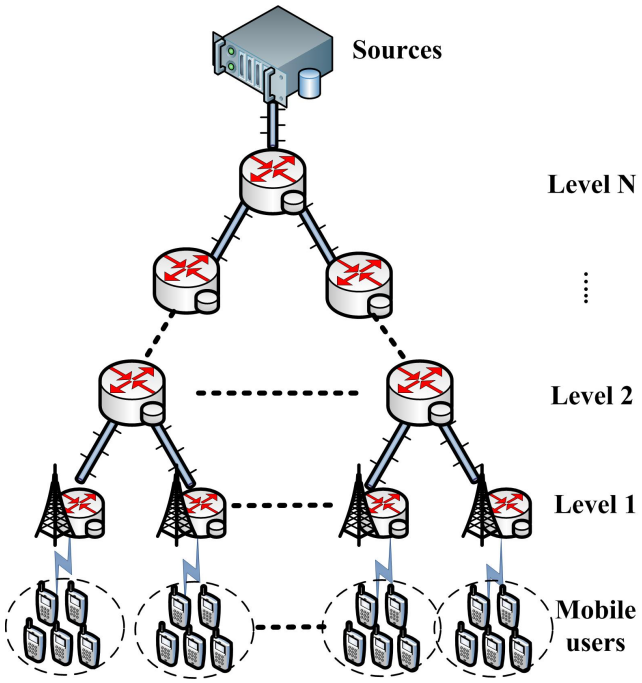


Fig. 3.1 The binary tree network topology to be investigated for caching performance.

Characteristics for cache hit ratio

Before the *cache hit ratio* of each router in network is derived, the mean arrival rate of chunk requests at level i of caches in the tree, $g(i)$, is identified first. Similar to g, $g(i)$ denotes the number of arrived chunk requests at the i^{th} level of routers in an open interval $(0,t)$.

Let the MMPP content requests represent the input of the router at the i^{th} level of cache with rate $\lambda_k(i)$ and popularity distribution $q_k(i) = \frac{\prod_{j=1}^{i-1}(1-h_k(j))q_k}{\sum_{m=1}^{K}\prod_{j=1}^{i-1}(1-h_m(j))q_m}, k = 1,\ldots,K,$. Moreover, let $S_i(0,t)$ be the number of chunks that requested in an open interval $(0,t)$ at routers of i^{th} level. As $K \to \infty, t \to \infty$, the mean arrival rate of

chunk requests at the i^{th} level of routers, $g(i)$, is given by,

$$g(i) = \lim_{t\to\infty} \frac{\mathbb{E}\left[S_i(0,t)\right]^{\alpha}}{t} = \frac{\lambda_{tot}(i)}{\mu(i-1)} g \tag{3.21}$$

where $\lambda_{tot}(i)$ is the total content request rate at the i^{th} level, and $\mu(i-1)$ is the miss rate of the requested content at the $(i-1)^{th}$ level.

The miss sequences for contents of different class are mutually independent. Thus, the cache miss process at the current level constitutes the input process for caches at the next level. For a binary tree topology, the request arrival rate $\lambda_k(i+1)$ is derived by $2\mu_k(i) = 2\lambda_k(i)(1-h_k(i))$. By superposing all the request at level i, the total arrival rate of level $i+1$ is given by $\lambda_{tot}(i+1) = 2\sum_{l=1}^{K}\mu_l(i) = 2\mu(i)$. As a result, the mean arrival rate at the i^{th} level of routers, $g(i)$, can be written as $g(i) = 2g$.

Given a binary tree network topology with N levels and $2^N - 1$ routers, let an MMPP content request process represent the input of the ICN routers at the i^{th} level with a Zipf popularity distribution of contents. Hence, $\forall 1 < i \leq N$, the *cache hit ratio* at the i^{th} level of routers, $h_k(i)$, is given by

$$h_k(i) = 1 - (1 - h_k(1))^{\prod_{m=1}^{i-1}(1-h_k(m))} \tag{3.22}$$

Proof. Given a request for a chunk of content in class k, due to Eq. (3.6), the *cache hit ratio* at the first level is represented by $h_k(1) = 1 - \beta_k e^{-u_k\tau_k} - (1-\beta_k)e^{-v_k\tau_k}$. Based on the result for the first level, the *cache hit ratio* for the second level can be derived by the modified content request rate, $g(2)$, and the modified popularity, $q(2)$, as

$$\begin{aligned}
u_k(2) &= \frac{\frac{\lambda_{tot}(2)q_k(2)}{m} + \sigma_{k1}(2) + \sigma_{k2}(2) - d}{2} \sim \frac{\lambda_{tot}(2)q_k(2)}{2m} \\
&= \frac{\lambda_{tot}(2)}{2m} \frac{q_k(1-h_k(1))}{\sum_{l=1}^{K} q_l(1-h_l(1))} \\
&= \frac{2\lambda_{tot}\sum_{l=1}^{K}(1-h_l(1))q_l}{2m} \frac{q_k(1-h_k(1))}{\sum_{l=1}^{K} q_l(1-h_l(1))} \\
&= \frac{\lambda_{tot}q_k}{m}(1-h_k(1)) \\
&\sim 2u_k(1-h_k(1))
\end{aligned} \tag{3.23}$$

Similarly, $v_k(2)$ can be derived by the same process, and is given by

$$v_k(2) \sim 2v_k(1 - h_k(1)) \tag{3.24}$$

Thus, the *cache hit ratio* for class k at the second level of routers, $h_k(2)$, can be written as

$$\begin{aligned} h_k(2) &= 1 - \beta_k e^{-u_k(2)\tau_k(2)} - (1-\beta_k)e^{-v_k(2)\tau_k(2)} \\ &= 1 - \beta_k e^{-2u_k(1-h_k(1))\frac{C^\alpha}{g(2)}} - (1-\beta_k)e^{-2v_k(1-h_k(1))\frac{C^\alpha}{g(2)}} \\ &= 1 - \beta_k e^{-u_k\frac{C^\alpha}{g}\frac{2g}{g(2)}(1-h_k(1))} - (1-\beta_k)e^{-v_k\frac{C^\alpha}{g}\frac{2g}{g(2)}(1-h_k(1))} \end{aligned} \tag{3.25}$$

Then we have,

$$1-h_k(1) = \beta_k e^{-u_k\frac{C^\alpha}{g}} - (1-\beta_k)e^{-v_k\frac{C^\alpha}{g}} \tag{3.26}$$

$$1-h_k(2) = \beta_k e^{-u_k\frac{C^\alpha}{g}\frac{2g}{g(2)}(1-h_k(1))} + (1-\beta_k)e^{-v_k\frac{C^\alpha}{g}\frac{2g}{g(2)}(1-h_k(1))} \tag{3.27}$$

Eqs. (3.26), (3.27) converge to the term with $e^{-u_k\tau_k}$ as C increases, because the term $\beta_k e^{-u_k C^\alpha/g}$ becomes dominant for large C. Recall the Eq. (3.6), and the miss probability of routers at the first level can be represented as $1 - h_k(1) = \beta_k e^{-u_k\frac{C^\alpha}{g}}$, thus, $h_k(2)$ can be written as

$$\begin{aligned} 1 - h_k(2) &\sim \beta_k e^{-\frac{u_k C^\alpha}{g}\frac{2g}{g(2)}(1-h_k(1))} \\ &= (1-h_k(1))^{1-h_k(1)} \end{aligned} \tag{3.28}$$

Therefore, the *cache hit ratio* at level i of a tree network can be derived by the iteration of the process from Eq. (3.23) to Eq. (3.28). The expression of the *cache hit ratio* at ICN nodes at level i, $h_k(i)$ that depends on the previous levels' *cache hit ratio*, is given by

$$h_k(i) = 1 - (1-h_k(1))^{\prod_{m=1}^{i-1}(1-h_k(m))} \tag{3.29}$$

which concludes the proof.

3.3 Model Validation and Performance Evaluation

The accuracy of the developed analytical model is validated by a chunk-level ICN simulator called ccnSim developed under the OMNeT++ framework. This open-source simulator focuses on the performance of caching policies and scalability [73]. It implements the Content Store (CS), Pending Information Table (PIT) and Forwarding Information Base (FIB) data structures, and content retrieve operations with different policies. This open-source simulator implements the CS, PIT and FIB data structures, and content retrieve operations of ICN. In this section we present simulative results to corroborate the above theoretical results first in the case of a single content store, then in the case of a network of content stores as in ICN.

3.3.1 Single Cache Performance under Bursty Content Requests

A population of $\|\mathbb{O}\| = 500$ distinct contents is considered. The contents are allocated into $K = 10$ classes with decreasing content popularity. The popularity distribution follows the Zipf law with the exponent parameter $\alpha = 2$. The value of the Zipf exponent, α, is abstracted from the analysis of YouTube for a realistic Internet catalog size [72]. Each class has $m = 50$ contents which are split into chunks with size of $10KB$. The size of contents is geometrically distributed with average 10^3 chunks ($10MB$). The size of chunk, $10KB$, and the average size of content, 10^3 chunks, are used in the literature [32]. The content requests generated by mobile users are modelled by MMPP with total intensity $\lambda_{tot} = 10\, contents/sec$, and the chunk transmission window size is $W = 1$ which is the default size of ICN [17]. The LRU replacement policy is implemented on each ICN node, and all the nodes are equipped with a cache of size C chunks. The parameters are summarised in Table 3.2. The results begin to be recorded until the simulation reach the steady state, and the results are the mean value gathering from over 10 simulation runs.

The developed analytical models and simulations are applied to study the topology in Fig. 3.2. For the single cache performance, the nodes at level 1 is considered. The results represent the average cache hit ratio of the 8 nodes of level 1.

Table 3.2 Parameters set for the validation

Parameter	Values
N	15
$\|\mathbb{O}\|$	500
m	50
K	10
chunk size	10KB
C	1GB, 1.2GB, 1.5GB
δ	10MB
α	2
W	1
$lambda_{tot}$	10 contents/s

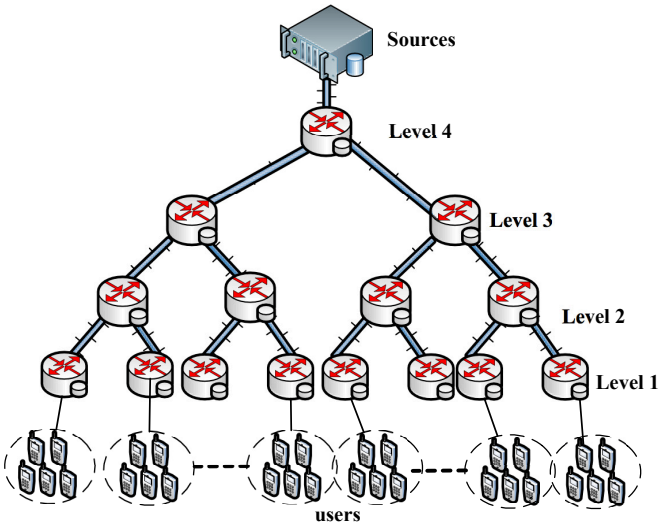


Fig. 3.2 4-level binary tree network used for the validation

Fig. 3.3 depicts the *cache hit ratio* as a function of the cache size with the Zipf exponent $\alpha = 2$ for different cache size C, with 100000 chunks ($1GB$), 120000 chunks ($1.2GB$) and 150000 chunks ($1.5GB$), respectively. The results derived from the analytical model and the simulations match closely. As expected, the cache hit ratio increased as cache size increases.

3.3.2 Network of Caches with Tree Topology under Bursty Content Requests

Considering the topology represented in Fig. 3.2, an $N = 4$ levels binary tree is built. In the tree network, all the *Data* is stored in the server which is connected to the ICN node at the root, while the leaf nodes are regarded as the edge nodes which

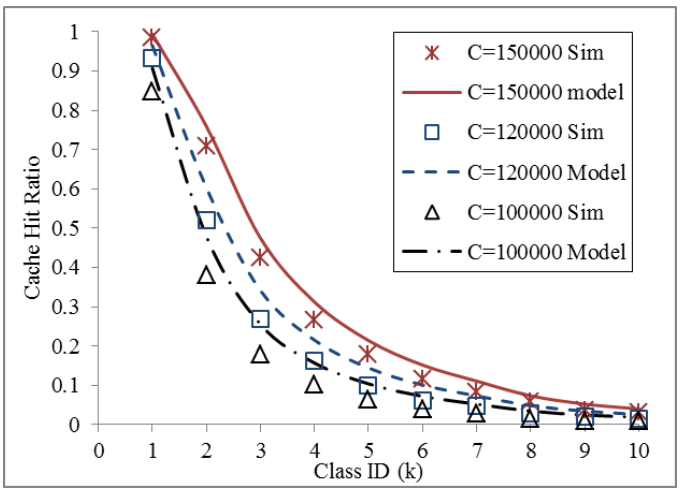


Fig. 3.3 Single cache hit ratio predicted by the model against those obtained through simulation under bursty traffic vs. contents of different classes with the different cache size $C = 100000, 1200000, 150000$ chunks.

receive content requests from end-users. In the network, all the links have the same delay $1ms$, and the same bandwidth capacity $10Gbps$. Every node is equipped with a cache with size $C = 150000$ chunks $(1.5GB)$ which implements an LRU replacement policy. The bursty content requests at the edge routers are driven by the MMPP arrival process with intensity $\lambda_{tot} = 10$ $contents/s$. The chunk transmission window size W is set to 1 chunk. In that case, at most one requested *Interest* can be sent by any user prior to receive the corresponding *Data* and issue the next *Interest*. Content population characteristics are the same as the single router model with the Zipf exponent $\alpha = 2$, average file size $10MB$ and the chunk size equals $10KB$. The results are only recorded when the system reaches the steady state. The results are the mean value of 10 times of simulation runs.

Fig. 3.4 depicts the *cache hit ratio* of ICN nodes at different levels, from level 1 to level 4. The comparison outlines a good match between the model and the simulation. From Fig. 3.4 it can be observed how content popularity changes along the path. Requests for the most popular contents are almost completely hit at the edge routers, as a consequence, the hit ratio for class $k = 1$ is very small at upper levels. The less popular classes, such as class $k = 2$ and $k = 3$, are mostly cached at the first and second level, whereas unpopular contents are rarely cached as they are hardly requested and quickly replaced by other contents in the cache.

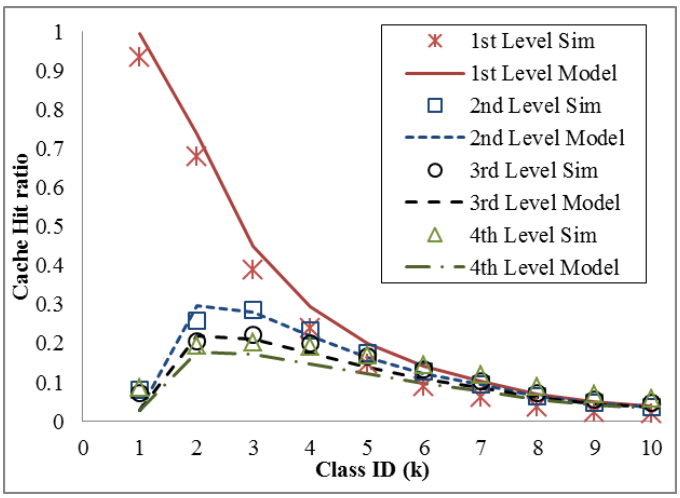


Fig. 3.4 Cache hit ratio predicted by the model against those obtained via simulation under bursty traffic for different level of routers, with $N = 4$ level binary tree, cache size $C = 150000$ chunks, and the Zipf exponent $\alpha = 2$.

The figure reveals that the analytical performance results closely match those obtained from the simulation experiments both in single model and network model, validating the accuracy of the developed analytical model.

3.3.3 Performance Evaluation

In this subsection, the developed model is used as a cost-effective tool to investigate the impact of key metrics on the performance of caching of ICN under bursty content reqeust. Caching performance is usually expressed in terms of the *cache hit ratio*. According to Eq. (3.6), the cache hit ratio is a function of the cache size C, the average content size δ, and the Zipf exponent α. Unless otherwise stated, the value of parameters in this section is the same as that in Section IV.

As far as the cache size of the ICN router is concerned in Fig. 3.3, as expected, the increase in cache size C causes the growth in *cache hit ratio*. The reason behind this can be found from the cache hit equation, i.e., Eq. (3.6). The hit ratio depends on the ratio cache size over content size that can be represented as $\frac{C}{M \times \delta}$. In fact, for a given content size, the larger C can lead to more storage space for caching chunks.

Furthermore, the content size plays an important role as well. Fig. 3.5 depicts that, as a function of the cache size, the *cache hit ratio* is also affected by the content size, δ. As the content size increases, the hit rate decreases accordingly. In other words, increasing the content size δ leads to the decrease of the ratio of cache size C

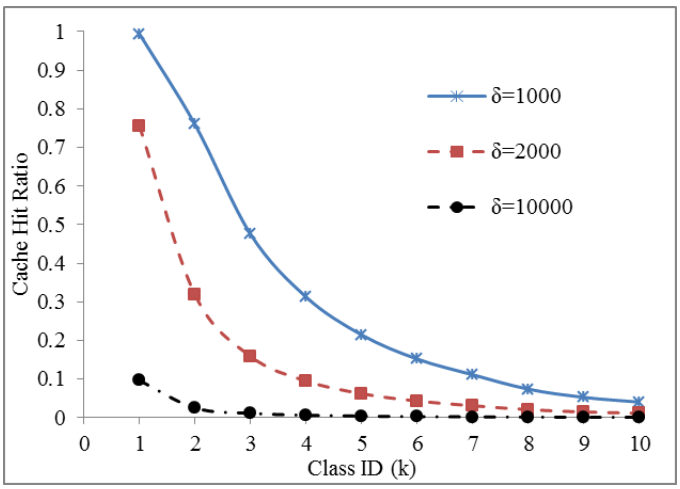


Fig. 3.5 Cache hit ratio predicted by the model for different content sizes.

to the whole content size $M \times \delta$. As illustrated in Fig. 3.5, when the content size becomes very large ($\delta = 0.1GB$), the cache hit ratio for all kinds of contents can be very low, indicating that the LRU strategy is not good at dealing with the large size of content.

To investigate the impact of Zipf exponent α on the caching performance, the global cache hit ratio, H, can be derived from the superposition of the miss sequences of different content in all popularity classes, and is given by

$$H = \sum_{k} q_k h_k \tag{3.30}$$

As depicted in Figs. 3.6 and 3.7, the global cache hit ratio, H, is a monotone increasing function of the Zipf exponent, α. This is because the smaller α leads to a flatter popularity, which means the popularity of each class is close to the others. In this case, the contents in the cache are removed more frequently than larger α, which drives down the cache hit ratio.

The figures reveal that the developed analytical model manages to predict the cache hit ratio of the router in ICN in the presence of bursty traffic.

3.4 Summary

An analytical model has been developed to investigate the caching performance of ICN under bursty content request. MMPP is leveraged to capture the bursty nature

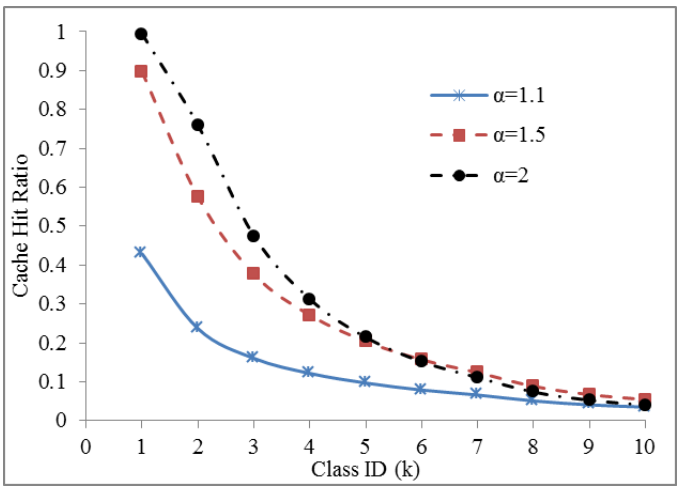


Fig. 3.6 Cache hit ratio predicted by the model for different Zipf exponent α vs. content of different classes.

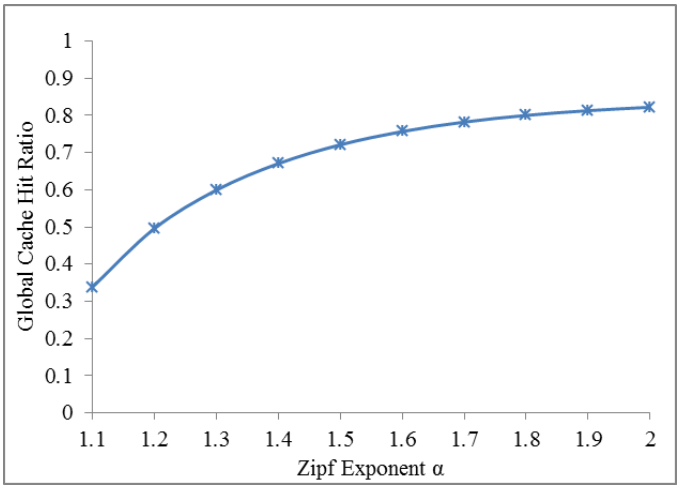


Fig. 3.7 Global cache hit ratio predicted by the model vs. different Zipf exponent α.

of content requests, and closed-form expressions of cache hit ratio has been derived in the models for single cache node and network of caches under tree topology. The accuracy of the two models has been verified through comparing the analytical results with extensive ICN simulation experiments. The models derive the expressions of the important caching performance metrics including cache size, content popularity, and content size. An investigation into the impact of the above metrics has been conducted. The efficiency of the analytical model for caching performance has also been evaluated. The analytical results have shown that the most popular contents are cached at the edge level of nodes in a tree network, so the uses can access those contents with less delay and better service quality. Furthermore, the performance results have also shown that content distribution has significant impact on the caching

performance. As the Zipf exponent becomes small, the cache hit ratio of ICN node decreases dramatically.

Chapter 4

Performance Analysis and Optimisation of Heterogeneous Caching under Arbitrary Topology

4.1 Introduction

Although ICN caching has received considerable research attention, the existing works on ICN caching are mainly focused on a single cache node or special cache network topologies, such as cascade topology or tree topology [9, 10, 27, 31, 77]. These special network topologies simplify the interoperability between cache nodes, therefore simplify the establishment and analysis of cache network models. But in ICN caches are ubiquitous and transmission paths are no longer fixed, the realistic topology of cache networks should be represented by arbitrary graphs [11] rather than the fixed parent-child relationships. Moreover, most existing analytical models for caching of ICN have been developed under the assumption of certain network topology, homogeneous cache size and content popularity distribution [9, 10].

To fill in these gaps, this chapter extents the analytical model under bursty content requests to investigate ICN caching with heterogeneous cache sizes and popularity under arbitrary topology. The performance of an arbitrary ICN is indicated and evaluated by cache hit ratio for different services at any node. The accuracy of the analytical model is validated through comparing the analytical results with those

obtained from simulation experiments. Then the analytical model is used to explore the impact of the key network and content parameters on the performance of ICN caching.

Moreover, to analyse the effects of heterogeneous cache sizes and content distributions. The developed arbitrary topology ICN model is used to design a caching allocation strategy which decides how much caching resources to be allocated to a node depending on the location and content distribution, and to achieve an optimal cache performance.

The novelty of this chapter can be summarised as follows:

(i) The proposed analytical model can derive the cache hit ratio for heterogeneous bursty content requests of multimedia service at any node in an arbitrary network.

(ii) The model is used as a cost-efficient tool to investigate the impact of key parameters of ICN, in terms of cache size, topology, content size and content popularity distribution. A new metric, average round trip time (ARTT) is defined to evaluate the delay performance.

(iii) An optimal cache allocation mechanism is proposed that leverages the proposed model and an evolutionary algorithm to find the optimal allocation of cache resources to achieve the best caching performance in ICN under different scenarios.

The remainder of the chapter is organized as follows. Section 4.2 presents the system parameters and proposes the analytical model for ICN caching with heterogeneous caching sizes and content popularity under arbitrary topology. In Section 4.3, the accuracy of the model is validated and then the model is used to carry out the performance analysis of ICN caching. Section 4.4 proposes an optimal caching allocation mechanism based on the developed model. Finally, Section 4.5 concludes the chapter.

4.2 Analytical Model

This section presents the system parameters used to develop the model firstly. Then in Section 4.2.2 an analytical model for the caching performance under heterogeneous cache sizes, content request rates and popularity distributions in an arbitrary topology is developed.

4.2.1 System parameters

Table 4.1 provides a summary of the notations used in the derivation of the model. Some parameters are with the same meaning as ones introduced in Section 3.2.1. The new parameters are presented investigate the arbitrary topology and are explained in details below:

(i) The cache network is represented by ICNet $= (V, E)$, where $V = \{v_1, \ldots, v_n\}$ denotes the cache nodes in the network, $E \subseteq V \times V$ denotes the links between nodes.

(ii) Each node v_n in the cache network contains a cache with the size of C_{v_n} chunks. Chunk is the minimum caching unit. In ICN, contents are segmented into multiple smaller pieces, called chunks, and each chunk is treated as an individually named object, aiming to allow flexible distribution and flow control. The similar idea of segmentation has also been adopted in many other content distribution systems, such as BitTorrent and eMule.

(iii) The cache on each ICN node runs the LRU cache replacement policy. The LRU policy has low complexity and has been used in [9, 10, 32, 65]. Moreover, the caching operations of LRU can be implemented at line speed, which is one important requirement of ICN.

(iv) A total of $\mathbb{O} = \{content_1, \ldots, content_O\}$ different contents are considered in the model. Contents are divided into K types of services, and each service contains $m = \|O\|/K$ different contents.

(v) The popularity of contents belonged to different services follows the Zipf distribution. Nodes at different locations have various popularity distribution. $q_{n,k}$, where $q_{n,k} = f(\alpha_n, k) = \frac{1/k^{\alpha_n}}{\sum_{i=1}^{K} 1/i^{\alpha_n}} = \frac{D_n}{k^{\alpha_n}}$, denotes the probability for the $k-th$ popular content at node n to be requested, with $1/D_n = \sum_{i=1}^{K} 1/i^{\alpha_n}$.

(vi) The size of content, $S(content_i)$ follows geometric distribution with an average of F chunks, i.e. $\mathbb{P}(S(content_i) = l) = \frac{1}{F}(1 - \frac{1}{F})^{l-1},\ i = 1, 2, \ldots, O$, and $l > 0$.

(vii) The arrivals of content requests generated is modelled by the superposition of K individual interrupted Poisson process (IPP) which is a two-state MMPP. For a content belonging class k, the content request process (content/chunk levels) with content request rate $\mathbf{\Lambda}_{k,n} = [\lambda_k(n), 0], k \in K, n \in N$.

(viii) The Average Round Trip Time ($ARTT$) of service k at node n, $ARTT_{k,n}$, denotes the average time-interval between the arrival of one request for content in service k at node n and the reception of the corresponding requested content. It acts as a similar role to the round trip time (RTT) for Transmission Control Protocol (TCP) connections in the Internet, and can be used to evaluate the performance of ICN.

4.2.2 Modelling the Performance of Arbitrary ICN

An analytical model is developed to investigate the performance of ICN caches with heterogeneous cache sizes and popularity distribution in an arbitrary topology. To evaluate the performance of caching, *cache hit ratio* is considered as the key performance metric in the model, since high cache hit ratio will result in efficiently access of contents and better quality of user experience (QoE). Furthermore, it will reduce the traffic load in the network and achieve better energy efficiency. The rest of this subsection will describe the details of the calculation of the cache hit ratio.

Generation of a cache miss

To determine the cache hit ratio, we need to know how a cache missing process is generated. If one request for $chunk_i^1$ of $content_i$ generates a miss at node v_j, due to

Table 4.1 Summary of main notations

Parameter	Meaning
V	The set of ICN caching nodes
E	Links between the nodes
$\mathbb{O}$	Total number of different content items
K	Number of different types of services
m	Number of different contents in each type of service
C_{v_n}	Cache size in number of chunks of node v_n
α	Zipf exponent characterizing the distribution
$q_{k,n}$	Probability of requests for contents of service k at node n
F	Average content size in number of chunks
S	File size in number of chunks following geometrically distributed
h_{k,v_n}	Cache hit ratio for a chunk of contents in type k at node v_n
H_k	Mean cache hit ratio for contents in type k service
H	Global cache hit ratio
$\lambda_k(n)$	Mean arrival rate of requests for contains in type k service at node v_n
$\lambda_{k_{tot}}(n)$	Actual content requests rate for type k service at node v_n
$\lambda_{tot}(n)$	Total content request rate for all kinds of services at node v_n
Q_k	Infinitesimal generator of requests for contents in class k
Λ_k	Request rate matrix of contents in class k
N	Number of ICN nodes in the network
$ARTT_{k,n}$	Average round trip time of service k at node n

the property of LRU caching policy, it means that more than C_{v_j} different chunks other than $chunk_i^1$ are requested during the interval between the current request for $chunk_i^1$ and previous request for the same chunk on node v_j. Therefore, the cache hit ratio of the request for a chunk of a content in type k at node v_n can be given by

$$h_{v_n}^k = 1 - \mathbb{P}(Req_{v_n}(\tau_n^k) \geqslant C_{v_n}) \tag{4.1}$$

where $Req_{v_n}(\tau_n^k)$ denotes that the number of different chunk requests arrived at node v_n during the inter-arrival time τ_n, between two subsequent requests of the same chunk in service type k. So the calculation of cache hit ratio has been translated into solving of the probability $\mathbb{P}(Req_{v_n}(\tau_n^k) \geqslant C_{v_n})$.

Firstly, we consider the content request process at a single node. Let $R_{j,k}^i(u, v)$ denote the number of requests for the i-th chunk of $content_j$ in service type k during

an open interval (u, v). Since the requests for the same chunk during (u, v) will not cause a replacement in the cache, we further need to calculate the number of requests for different chunks. Let $B^i_{j,k}(u, v)$ denotes a Bernoulli variable that means chunk i of $content_j$ in class k has been requested in the open interval (u, v), i.e., $B^i_{j,k}(u, v) = \mathbb{1}\{R^i_{j,k}(u, v) > 0\}$. Finally, $\forall k' \in K, \forall j' \in m'_k, \forall i' \in F_{j',k'}$ let $N^{i'}_{j',k'}(u, v)$ denote the number of distinct chunks different from $chunk^{i'}_{j',k'}$ that are requested among the interval (u, v) from all the services, and $N^{i'}_{j',k'}(u, v)$ can be calculated as

$$N^{i'}_{j',k'}(u, v) = \sum_{k=1}^{K} \sum_{j=1}^{m_k} \sum_{\substack{i=1 \\ i \neq i'}}^{F_{j,k}} B^i_{j,k}(u, v) \tag{4.2}$$

Because the bursty requesting process is captured by MMPP and the service popularities follow the Zipf distribution, the time interval between the request for a chunk and its subsequence request is various for different services. Furthermore, the content and chunks in the same service have the same probability being requested, so the cache hit ratio is calculated for different services. Under the Zipf distribution and MMPP modeled bursty requests, given the requests for contents in service type k with intensity $\lambda_{tot}(k, n)$ and MMPP transition matrix Q^k_n, $\mathbb{P}(Req_{v_n}(\tau^k_n) \geqslant C_{v_n})$ can be written as [110]

$$\begin{aligned} \mathbb{P}(Req_{v_n}(\tau^k_n) \geqslant C_{v_n}) =& \mathbb{P}(\sum_{k=1}^{K} \sum_{j=1}^{m_k} \sum_{\substack{i=1 \\ i \neq i'}}^{F_{j,k}} B^i_{j,k}(\tau^k_n) \geqslant C_{v_n}) \\ =& \beta^k_n e^{-u^k_n \tau^k_n} + (1 - \beta^k_n) e^{-v^k_n \tau^k_n} \end{aligned} \tag{4.3}$$

where $\forall chunk_{i'} \in \mathbb{O}$ and the parameters u^k_n ,v^k_n are the two eigenvalues of $(\Lambda^k_n - Q^k_n)$, and can be written as [109]

$$\begin{aligned} u^k_n &= \frac{\frac{\lambda_{tot}(k,n)}{m} + \sigma^{k1}_n + \sigma^{k2}_n - d^k_n}{2} \\ v^k_n &= \frac{\frac{\lambda_{tot}(k,n)}{m} + \sigma^{k1}_n + \sigma^{k2}_n + d^k_n}{2} \end{aligned} \tag{4.4}$$

with

$$d_n^k = \sqrt{(\frac{\lambda_{tot}(k,n)}{m} + \sigma_n^{k1} - \sigma_n^{k2})^2 + 4\sigma_n^{k1}\sigma_n^{k2}} \tag{4.5}$$

and β_n^k is the transformation parameter from the *IPP* to the hyperexponential distribution and given by

$$\beta_n^k = \frac{\frac{\lambda_{tot}(k,n)}{m} - v_n^k}{u_n^k - v_n^k} \tag{4.6}$$

As a result, the cache hit ratio Eq. (4.1) can be written as

$$h_{v_n}^k = 1 - \beta_n^k e^{-u_n^k \tau_n^k} - (1 - \beta_n^k) e^{-v_n^k \tau_n^k} \tag{4.7}$$

Calculation of the time interval τ_n^k

The inter-arrival time, τ_n^k, denotes the interval in which there are more than C_{v_n} requests arrived between two subsequent requests of the same chunk in Type k. Since the number of arrival chunks is larger than the the cache size, the requested chunk has been removed from the cache, thus it generates the event of cache miss.

From the previous analysis, τ_n^k is determined by content request rate (Λ_k), the cache size (C_{v_n}), the overall content number ($\mathbb{O}$), the types of services (K), the content size (F) and the popularity distribution (Zipf parameter α), so τ_n^k can be expressed as a function of the above variables, i.e,

$$\tau_n^k = f(\Lambda_k, C_{v_n}, \mathbb{O}, K, F, \alpha) \tag{4.8}$$

In our previous work [110], the mean arrival rate at chunk-level requests, g and the inter-arrival time, τ_k, under a single ICN node scenario has been derived as

$$\begin{aligned} g &= \Gamma(1 - \frac{1}{\alpha})^\alpha (\frac{\lambda_{tot} D}{2}) m^{\alpha-1} F^\alpha \\ \tau_k &= C^\alpha / g \end{aligned} \tag{4.9}$$

However, for an ICN node in a caching network scenario, the rate of content requests arrived at a node is the combination of two streams. One is the bursty stream of content requests arriving exogenously, and the other is the received stream of forwarding requests generated by its neighbours in the event of a miss. So the synthetic arrived requests at a caching node in an arbitrary caching network is given by

$$\lambda_{tot}(k,n) = \lambda(k,n) + \sum_{\substack{v_{n'} \neq v_n \\ E<n,n'>\neq 0}} miss(k,n') \tag{4.10}$$

The miss rate at an ICN node not only depends on the caching policy, but also relates to the time for moving a copy of the content to the cache after a miss generated. The paper follow the common practice in [81, 111, 112] that the content is downloaded and cached instantaneously after a miss occurs. So, the chunk arrival rate at node v_n, g_n, and inter-arrival time τ_n^k can be written as

$$\begin{aligned} g_n &= \Gamma(1-\frac{1}{\alpha})^{\alpha}(\frac{\lambda_{tot}(n)D_n}{2})m^{\alpha-1}F^{\alpha} \\ \tau_n^k &= (C_{v_n})^{\alpha}/g_n \end{aligned} \tag{4.11}$$

where $\lambda_{tot}(n)$ is the mean arrival rate of all the requests at node v_n, and is composed by

$$\begin{aligned} \lambda_{tot}(n) &= \sum_{k=1}^{K} \lambda_{tot}(k,n) \\ &= \sum_{k=1}^{K} \lambda(k,n) + \sum_{k=1}^{K} \sum_{\substack{v_{n'} \neq v_n \\ E<n,n'>\neq 0}} miss(k,n') \end{aligned} \tag{4.12}$$

The first part of Eq. (4.12) is the mean arrival rate of MMPP (which can be solved by Eq. (3.5). To determine the second part, $\sum_{k=1}^{K} \sum_{E<n,n'>\neq 0}^{v_{n'} \neq v_n} miss(k,n')$, Eq. (4.7) has to be solved for all types k and different nodes v_n.

Calculation of cache hit ratio under arbitrary ICN

To determine the cache hit ratio at any node in arbitrary ICN, the analytical model treats this problem as a hot-spot problem. The reason behind this is that ICN

transmit content request by hops along the path towards the content repository, in which a missing request will be forwarded to the next hop node that is nearer to repository. As a result, the caching nodes connect or near to the repositories will receive massive miss requests from the nodes that are far from the repositories, resulting in the nodes near to the repositories becoming hot-spot nodes.

In the model, we consider a realistic arbitrary ICN topology. The network contains R repositories with contents are stored in each repository. Let $st_{o_j}^{r_i}$ denote that content o_j is stored in repository r_i. So we have

$$st_{o_j}^{r_i} = \begin{cases} 0 & o_j \in r_i \\ 1 & o_j \notin r_i \end{cases} \tag{4.13}$$

Furthermore, we denote $d_{v_n}^{o_j}$ the number of hops that a request for o_j from node v_n to the nearest repository that contains o_j. The request is possible to be satisfied by the cache at each hop, we denote this probability as $p_{n,n+i}^{o_j}, 1 \leqslant i \leqslant d_{v_n}^{o_j}$, i.e., a request for o_j from v_n is hit in the cache at node v_{n+i} with probability $p_{n,n+i}^{o_j}$.

An ICN node in arbitrary network not only receives requests from the end-users connecting to it which are called the exogenous requests, but also receives requests forwarded by their neighbours which are called the missing requests. Note that missing requests will only be forwarded to the a neighbour node if that node is nearer to one of the repositories that contains the requested content. Node v_n that is d_{v_n} hops away from the nearest repository receives the missing content requests from its neighbour nodes that are more than d_{v_n} hops away to that repository, i.e.,

$$miss_n^{o_j} = \sum_{E<n,n'>} miss(o_j, n'),\ d_{v_n}^{o_j} < d_{v_{n'}}^{o_j} \tag{4.14}$$

where $miss_n^{o_j}$ denotes the requests for o_j that v_n receives from all the neighbour nodes.

According to Eq. (4.7) (4.10) (4.11) and (4.12), the cache hit ratio for service k at node n depends on the arrival request pattern and content distribution, but in a hot-spot network, it is also significantly related to the location of the node.

Therefore, we use a recursive method to calculate the cache hit ratio of ICN node in an arbitrary network.

Suppose that a request for the i^{th} chunk of o_j in service k from node v_s travels along the path and is satisfied by the node v_f that is $d_{v_f}^{o_j}(0 \leqslant d_{v_f}^{o_j} \leqslant d_{v_s}^{o_j})$ hops away from the repository, i.e., the hot-spot. Let $P_f^{i,o_j,k}$ denote the probability of the i^{th} chunk of content o_j in service k hitting in the cache of node v_f that is $d_{v_f}^{o_j}$ hops away from the nearest repository. For general cases where $0 < d_{v_f}^{o_j} < d_{v_s}^{o_j}$, the probability can be expressed as

$$\begin{aligned} P_f^{i,o_j,k} &= (1-h_{v_s}^k)(1-h_{v_{s-1}}^k)\dots(1-h_{v_{f-1}}^k)\centerdot h_{v_f}^k \\ &= \prod_{q=s}^{f-1}(1-h_{v_q}^k)\centerdot h_{v_f}^k \end{aligned} \tag{4.15}$$

Then, we consider the two boundary cases. When $d_{v_f}^{o_j} = 0$, the request is missed at all the nodes along the path, while $d_{v_f}^{o_j} = d_{v_s}^{o_j}$ means the requested content is stored in the cache of first hop node v_s, then we have

$$P_f^{i,o_j,k} = \begin{cases} \prod_{q=s}^{0}(1-h_q^k) & d_{v_f}^{o_j} = 0 \\ h_{v_s}^k & d_{v_f}^{o_j} = d_{v_s}^{o_j} \end{cases} \tag{4.16}$$

Note that contents of the same service are requested under the same probability due to the Zipf-like content distribution, so the cache hit ratio is independent of the specific object, which means $P_f^{i,o_j,k}$ is equivalent to P_f^k for request of any chunk in service k.

To calculate $h_{v_n}^k$, we leverage Eq. (4.14), (4.15) and (4.16) to get the total requests arriving at node n, so we have,

$$\begin{aligned}\lambda_{tot}(n) &= \sum_{k\in K}\lambda(k,n) + miss(n) \\ &= \sum_{k\in K}(\lambda(k,n) + \sum_{o_j\in\mathbb{O}_k} miss_n^{o_j}) \\ &= \sum_{k\in K}(\lambda(k,n) + \sum_{o_j\in\mathbb{O}_k}\sum_{E<n,n'>} miss(o_j,n')) \\ &= \sum_{k\in K}(\lambda(k,n) + \sum_{E<n,n'>}\lambda(k,n')(1-P_{n'}^k)) \end{aligned} \tag{4.17}$$

From Eq. (4.7) (4.11) and (4.17), we observe that cache hit ratio $h_{v_n}^k$, arrival pattern $\lambda_{tot}(n)$, and topology that determins the distance $d_{v_n}^k$ are dependent on each other. As a result, to solve the equations we develop an algorithm which calculates the results in a recursive manner, which is described in Alg. 1.

The for loop in Step 1 initialises the exogenous content requesting rate according to MMPP arrival and original cache hit ratio at each node without considering the forwarded missing requests. From Step 4, the cache hit ratio is calculated by a recursive manner. Step 5-11 calculate the missing request rate through recursively examining the miss request forwarded from the neighbour nodes. The stopping criterion is that the current node is the last node to be examined or it is the farthest node from the repository for the current service. As a result, this node does not receive any missing requests forwarded by other nodes, and the original hit ratio is used to compute the missing process at this node. Then, this missing requests is taken into account by the neighbours and used to compute the new missing process. After all the nodes have been examined, Eq. (4.12) (4.11) and (4.7) are used to calculate the cache hit ratio of node under different services. Step 15-17 map each service to its weight using the Zipf popularity distribution, and derive the average cache hit ratio for each node.

Algorithm 1 Calculation of cache hit ratio

Input: (V, E), N, C, K, α, Λ, Q, m, F
Output: h_{v_n}: average cache hit ratio at any node v_n

for each $n \in N$, $k \in K$ **do**
  initialise the exogenous $\lambda(k,n) \leftarrow \Lambda, Q, \alpha, K, N$;
  Calculate the original $h^k_{Ori,v_n} \leftarrow \tau^k_n \leftarrow g_n$;
end for
for each $n \in N$ **do**
  Generate neighbour set $Nei(n) \leftarrow (V, E)$
  $miss(n) \leftarrow \sum miss(k, Nei(n))$;
  while $Nei(n) \neq \phi$ **do**
    for each $n' \in Nei(n)$ **do**
      $n \leftarrow n'$ and **repeat** Step 6;
    end for
  end while
  if $(Nei(n) = \phi)$ or $(d^k_{v_n} > d^k_{v'_n}, \forall n' \in Nei(n), k \in K)$ **then**
    return $miss(n) = \sum_{k \in K} \lambda(k,n) \cdot h^k_{Ori,v_n}$;
  end if
  $\lambda_{tot}(n) = \sum_{k \in K} \lambda(k,n) + miss(n)$;
  $\tau^k_n \leftarrow \lambda_{tot}(n), \alpha, m, F, C$;
  $h^k_{v_n} \leftarrow \tau^k_n, \Lambda, Q$;
  for each $k \in K$ **do**
    $q_{k,n} = 1/k^{\alpha_n} / \sum_{i=1}^{K} 1/i^{\alpha_n}$;
    $h_{v_n} = q_{k,n} \cdot h^k_{v_n}$;
  end for
end for
return h_{v_n}

4.3 Model Validation and Performance Analysis

The accuracy of the developed analytical model is validated by the same ICN simulator, ccnSim, developed under the OMNeT++ framework. An arbitrary topology has been implemented in ccnSim.

In this section, the model developed for caching network with arbitrary topology under bursty content requests is validated in a two-dimensional 5×5 torus network as illustrated in Fig. 4.1. Torus topology has been widely used [32, 45, 81] to investigate the cache performance. Furthermore, by applying different routing methods, a torus topology can be formed into a cascade topology (deterministic routing) or a tree topology (shortest path routing). Each ICN node receives exogenous content requests from a group of users. One repository which contains all the contents is randomly

placed in the network for multiple simulation experiments. Multimedia services are requested and consumed by end-users.

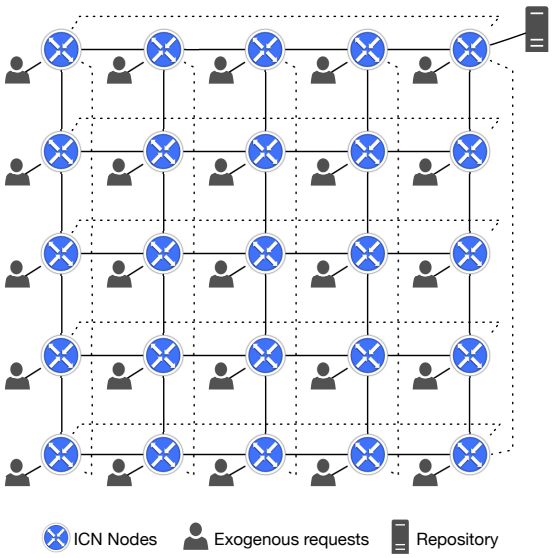


Fig. 4.1 Network topology: 5×5 two-dimensional torus network with one repository connecting to a random node.

The parameters for the validation are presented in Tab. 4.2. A total of $\|\mathbb{O}\| = 500$ different contents are considered and allocated into $K = 10$ sets with decreasing content popularity as the set number increases. The popularity of each set of contents follows the Zipf distribution with the exponent parameter $\alpha = 2$. The Zipf exponent $\alpha = 2$ is derived from the analysis of YouTube for a realistic Internet catalog size [72]. Each type of service owns $m = 50$ contents which are split into chunks of $10KB$, and the content size is geometrically distributed with the average 10^3 chunks ($10MB$). The size of chunk of $10KB$ and the average size of content of 10^3 chunks have been widely used in the literature. In the torus network, each node receives the exogenous content requests generated by end-users accessing multimedia services. The exogenous requests are generated under a bursty manner with the mean arrival intensity 10 $contents/s$, and the transmission window size of chunk is set to $W = 1$. The standard Leave Copy Everywhere (LCE) decision policy and LRU replacement policy are implemented on each cache node with the equal size C_{v_n}. Note that the transmission window with $W = 1$, LRU and LCE are default policies of CCN [17].

Table 4.2 Parameters set for the validation

Parameter	Values
ICNet	5×5 Torus
N	25
$\|\mathbb{O}\|$	500
m	50
K	10
chunk size	10KB
C_{v_n}	1GB, 1.2GB, 1.5GB
F	10MB
α	2

In the topology shown in Fig. 4.1, nodes can be classified into 5 types based on their distances to the repository. In a 5×5 torus network, the maximum number of hops from any node to the repository is 4 hops. Because there is only one repository in the network, the distance from a node with a request for any content to the repository is only determined by the location of the node, therefore, $d_{v_n}^{oj}$ is equivalent to d_{v_n}. To demonstrate a more general scenario, the random forwarding method is used in the validation, which means that in the event of a cache missing at a node, each neighbour node has the equal chance to receive the forwarded missing request. Moreover, according to the algorithm describe in Alg. 1, the nodes that are 4 hops away from the repository will not receive forwarded missing requests from their neighbours, i.e., $\forall n \in N$, if $d_{v_n} = 4$, then we have

$$\begin{aligned} \lambda_{tot}(n_{d_4}) &= \sum_{k \in K} \lambda(k, n_{d_4}) \\ miss(k, n_{d_4}) &= \lambda_{tot}(k, n_{d_4}) \cdot h^k_{v_{n_{d_4}}} \end{aligned} \tag{4.18}$$

where n_{d_i} means that the node is 4 hops away from the repository. Then for the nodes that are 3 hops away from the repository, each node has 50 percent of chance to receive the forwarded missing request. In such a situation, Eq. (4.10) can be re-written as

$$\lambda_{tot}(k, n_{d_3}) = \lambda(k, n_{d_3}) + \frac{1}{2} miss(k, n_{d_4}) \tag{4.19}$$

Similarly, we can derive the total arrival rate for n_{d_2} as

$$\lambda_{tot}(k, n_{d_2}) = \lambda(k, n_{d_2}) + \frac{1}{2} \sum_{n' \in \{n_{d_4 1}, n_{d_4 2}\}} miss(k, n') \tag{4.20}$$

The 4 nodes that are 1 hop away from the repository receive the missing requests from their 3 neighbours, and all forward the requests that are not satisfied by their caches to the node connected to the repository. The arrival rate for these nodes can be derived by

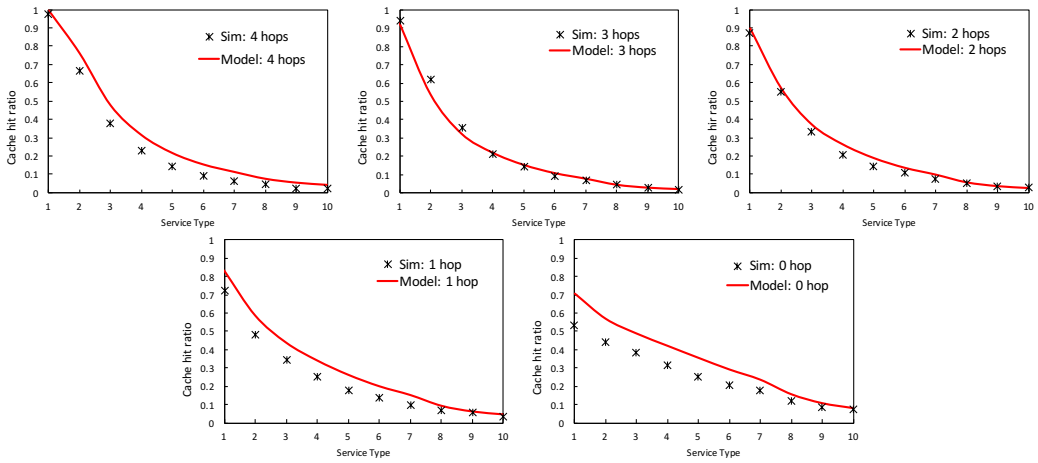
$$\lambda_{tot}(k, n_{d_1}) = \lambda(k, n_{d_1}) + \frac{1}{2} \sum_{n' \in \{v_n | d_{v_n} = 2\}} miss(k, n') \tag{4.21}$$

$$\lambda_{tot}(k, n_{d_0}) = \lambda(k, n_{d_0}) + \sum_{n' \in \{v_n | d_{v_n} = 1\}} miss(k, n') \tag{4.22}$$

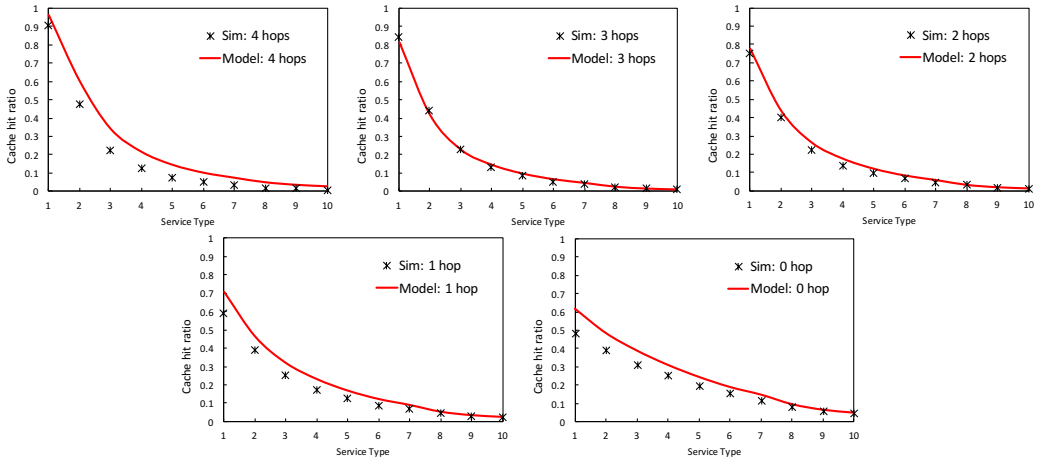
Due to the symmetry feature of torus network, nodes with the same distance from the repository have similar performance, thus can be integrated and investigated together. As a result, we define the mean cache hit ratio for contents of service k at nodes that are i hops away from the repository as H_{k,d_i}. H_{k,d_i} can be derived by a weighted average of the cache hit ratio for service k at each node that are i hops from the repository, and is given by

$$H_{k,d_i} = \sum_{v_n \in \{n | d_{v_n} = i\}} \omega_n h^k_{v_{n_{d_i}}} \tag{4.23}$$

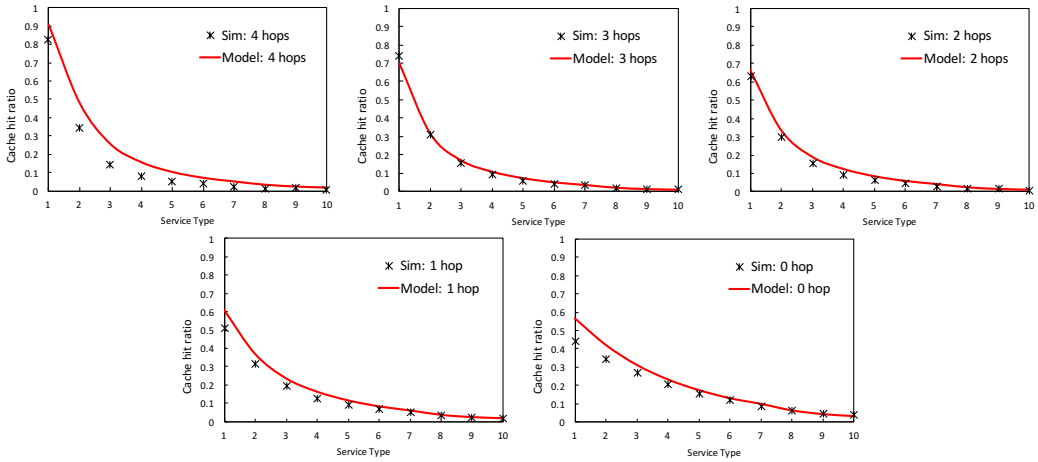
where ω_n is the weight factor of the node that depends on the traffic load through it, with $\sum \omega_n = 1$. Fig. 4.2 depicts the mean cache hit ratio, H_{k,d_i}, as a function of service type under different number of hops and cache sizes. The cache hit ratio is examined at various hops with cache size of 100000 chunks (1GB), 120000 chunks (1.2GB) and 150000 chunks (1.5GB), respectively. The figures show that the analytical performance results match well with those obtained from the simulation for nodes at different hops, which validates the accuracy of the developed analytical model.



(a) Cache hit ratio of nodes with C_{v_n}=150000 at different hops



(b) Cache hit ratio of nodes with C_{v_n}=120000 at different hops



(c) Cache hit ratio of nodes with C_{v_n}=100000 at different hops

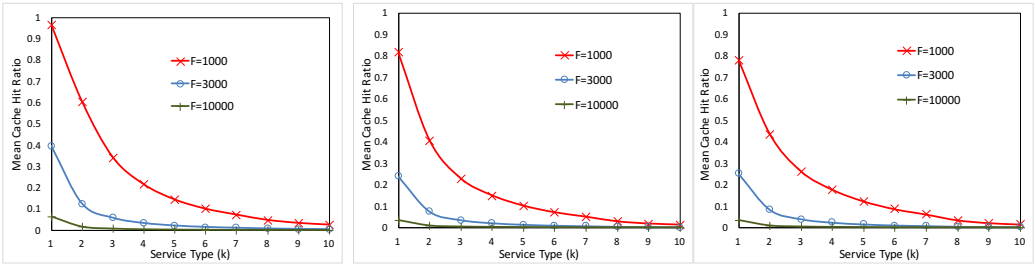
Fig. 4.2 Mean cache hit ratio H_{k,d_i} predicted by the model against those obtained through simulation under bursty traffic vs. content of different service types with various cache sizes C_{v_n}.

Next, the developed model is used as an efficient tool to investigate the impact of key network and content parameters on the caching performance in ICN. According to Eqs. (4.11) and (4.7), cache hit ratio $h_{v_n}^k$ is a function of these parameters: cache size C_{v_n}, average content size F, total number of contents O and Zipf exponent α. Unless otherwise stated, the values of the parameters in the performance evaluation is the same as those in Tab. 4.2.

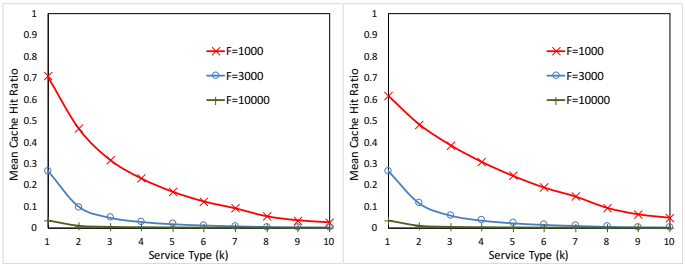
Cache size is one of the most important factor for the performance of ICN. From Fig. 4.2, we can see that the decrease of cache size C_{v_n} causes the reduction of cache hit ratio as expected. Furthermore, topology also plays an important role in the cache performance. As illustrated in all Figs. 4.2a, 4.2b and 4.2c, the edge nodes have the highest cache hit ratio in all three cases. As nodes are nearer to the repository, the cache hit ratio decreases accordingly. Because the edge nodes do not receive missing requests from other nodes, the traffic load at edge nodes is lighter compared to the core nodes. The developed analytical model takes the above parameters into account and can be used to investigate the performance of any node in an arbitrary network.

Content size, F, also has a large impact on caching performance. As shown in Fig. 4.3, the mean cache hit ratio is clearly affected by the content size, as the increase in F leading to the decrease in hit rate. As illustrated in Fig. 4.3, when average content size becomes very large ($F = 0.1GB$), the cache hit ratios for all types of contents are very low. This indicates that a different cache policy needs to be designed for services with large contents. In addition, to improve the cache hit ratio for large content, the chunk-level popularity distribution instead of the content-level one should be investigated.

To investigate the impact of content popularity distribution on the cache performance, different Zipf exponent α values are considered. Furthermore, the change of popularity distribution has influence on all types of services in the network. To illustrate this influence, we define the global cache hit ratio, H_{d_i}, for comparison. H_{d_i} can be derived by the superposition of the cache hits of different types of services



(a) 4 hops (b) 3 hops (c) 2 hops



(d) 1 hop (e) 0 hop

Fig. 4.3 Mean cache hit ratio predicted by the model for different content sizes with cache size $C_{v_n} = 120000$.

at the nodes with same hops, and is written as

$$H_{d_i} = \sum_{k \in K} q_k H_{k,d_i} \tag{4.24}$$

As depicted in Fig. 4.4, the global cache hit ratio, H_{d_i}, is a monotone increasing function of the Zipf exponent α for all nodes at different locations. The reason behind this is that smaller α values lead to a flatter popularity distribution, which means that the popularity of each service is closer to the others and the contents in each type of services are requested with a similar probability. The flat popularity narrows the gap between the most popular service and least popular service, which means that even the least popular service has more chances to be accessed by end-users. The LRU policy assumes that the content requested recently is likely to be requested again. Under diverse content requests, less popular contents also have chances to be stored in the cache and may substitute for popular contents. In this case, due to the limited size of cache, small α value results in frequent cache replacement and thus pulls down the cache hit ratio. Note that for smaller α, such as $\alpha = 1.1$ to 1.5, the global cache hit ratio of core nodes ($d_{v_n} = 0$ or 1) is higher than edge nodes (3 and 2 hops away), which is caused by the flat content popularity distribution

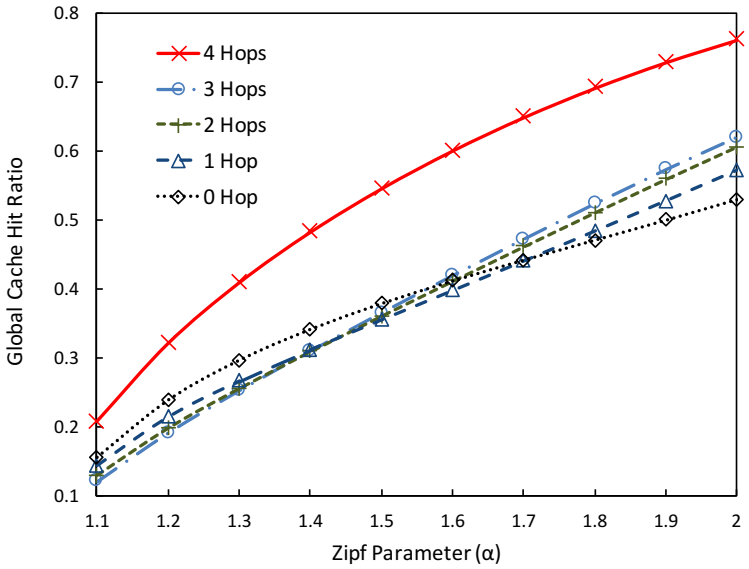


Fig. 4.4 Global cache hit ratio H_{d_i} predicted by the model vs. different Zipf exponent α with cache size $C_{v_n} = 120000$.

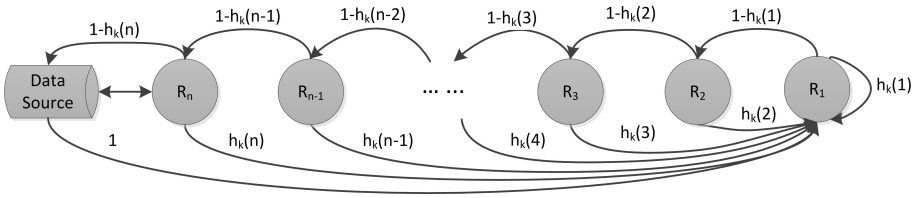


Fig. 4.5 CCN router state transition diagram

with a small α. Nodes with 3 and 2 hops receive some of the missing requests from their neighbours, because all services have closer requesting rates, the caches are frequently replaced which leads to a low cache hit ratio. However, the core nodes receive large missing requests from edge nodes, which offsets the popularity impact in this case and results in a slightly higher cache hit ratio.

To evaluate the service quality enhanced by caching, we define the Average Round Trip Time (ARTT) of a service requested from a node, as $ARTT_{k,n}$, which denotes the average time interval between the arrival of one request for service k at node n and the reception of the requested content. It acts as a similar role to the round trip time (RTT) for TCP connections in IP network. The requested content can be provided by any ICN node in the network which leads to different ARTTs. The ARTT is strongly depends on the location of node where the request arrived and the cache hit ratio along the request routing path. Therefore, $ARTT_{k,n}$ is defined as a weighted sum of link delay, where the weights correspond to the cache hit ratio at

each node along the transmission path as shown in Fig. 4.5. $ARTT_{k,n}$ is given by

$$\begin{aligned} ARTT_{k,n} &= \mathbb{E}(P_n^{i,o_j,k} \cdot L_n) \\ &= \mathbb{E}(\prod_{q=1}^{d_{v_n}-1} (1 - h^k_{v_{n+q}}) \cdot h^k_{v_n} \cdot L_n)) \\ &= \frac{1}{N} \sum_{i=1}^{N} L_i \cdot h^k_{v_{n_{d_i}}} \prod_{m=1}^{i-1} (1 - h^k_{v_{n_{d_j}}}) \end{aligned} \tag{4.25}$$

where L_i denotes the link delay between the request arrival node and the node that satisfies the request, which is determined by the number of hops between the two nodes. When the requested content is obtained from the repository, the request has generated a miss at every node along its path to the repository. In this case, to calculate the time interval of requests satisfied by the repository, we define the $ARTT^{Repo}_{k,n}$ as

$$ARTT^{Repo}_{k,n} = L_r \prod_{i=1}^{n} (1 - h^k_{v_{n_{d_i}}}) \tag{4.26}$$

where L_r is the link delay between node n and the repository.

Fig. 4.6 shows the ARTT of the nodes that are 4 hops way from the repository, $ARTT_{k,n_4}$, as a function of different types of services. The ARTT measures the average time to obtain a requested service and is closely related to the cache hit rates along the path. The most popular contents are often cached in ICN nodes, therefore, $ARTT_{1,n_4}$ is a small value of 2ms, On the contrary, the least popular contents are hardly cached within the network, therefore, most of them are accessed from the repository, with a consequent large $ARTT_{10,n_4}$ of around 11ms.

4.4 Model-based Optimal Cache Allocation

When ICN nodes are deployed in realistic networks, cache resource will be placed at each ICN node. To achieve the maximum caching performance and resource utilisation, a cache allocation strategy which is able to determine how to distribute cache resource across nodes is needed. The strategy targets at minimising the total traffic within the network and improving the network performance.

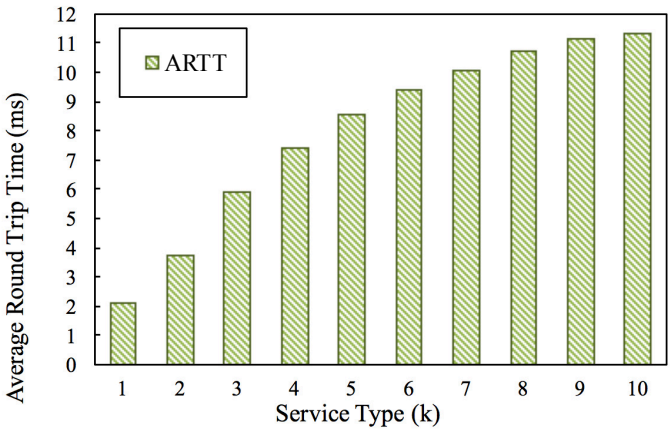
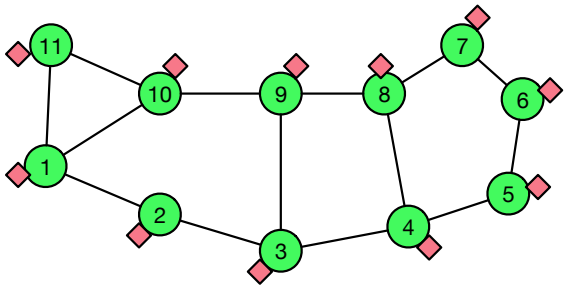


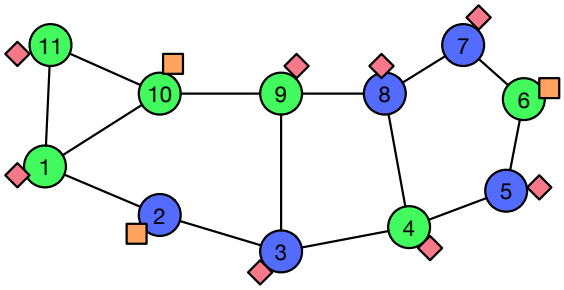
Fig. 4.6 Average round trip time of different types of services under the torus topology with link delay equals 2ms and $C_{v_n} = 120000$.

Some existing works on ICN caches considered homogeneous cache sizes, where the cache size of each node is equal. However, the cache performance is related to many parameters except the cache size, including content request rates, content distributions and network topology. In this section, as an application of the proposed analytical model, a cache allocation strategy is designed to allocate an optimal cache size to each ICN node in order to achieve the best cache performance.

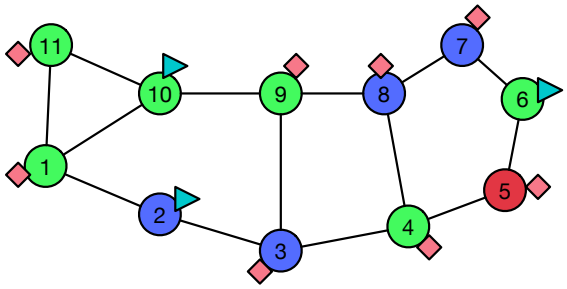
To evaluate the proposed strategy, we consider a realistic network topology called Abilene [113]. Three different network scenarios are investigated and illustrated in Fig. 4.7. In the first scenario, an ideal homogeneous content request intensity and popularity distribution is considered. The homogeneous scenario was used in [49] when it optimises the cache allocation. According to the previous analysis, different request arrival rates and content distributions exhibit various effects in cache performance. Moreover, because in practise different groups of customers are expected to have diverse interests in contents and generate different traffic volumes, we further consider a heterogeneous network with various intensity of requests and content distributions in the second scenario. Two types of arrival rates are considered, with some nodes generating triple volume of traffic than the others. Furthermore, contents are requested under two different popularity distributions. In the third scenario, we consider three request rates with large differences among the nodes. Different content distributions are considered as well, but with less skewed content popularity than that in the Scenario 2.



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

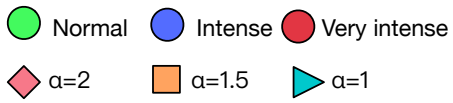


Fig. 4.7 The topology of Abilene network with three different network configuration

For all the scenarios, the topology is described as a graph $S = (V, E)$, with the network size $|V| = 11$. Each node in the network is connected to a group of users. The total number of contents provided in the network is $M = 500$ with the average content length of 100 chunks. Contents are equally distributed into 10 types of services. The popularity distribution and content request rate of each node is denoted as α_{v_n} and R_{v_n}, respectively. In the first scenario, each green node receives 20 content requests per second with the Zipf distribution exponent $\alpha = 2$. In the second scenario, the purple nodes (node 2,3,5,7,8) receive three times of requests than the green nodes, with $R_{v_{green}} = 20$ requests/sec and $R_{v_{purple}} = 60$ requests/sec, respectively. The requested contents follow popularity distributions with

$\alpha_{v_{1,3,4,5,7,8,9,11}} = 2$ and $\alpha_{v2,6,10} = 1.5$, respectively. In the third scenario, the content requests of node 5 become very intense with the highest rate of requests $R_{v_5} = 90$ requests/sec. Furthermore, a less skewed popularity distribution is considered, with a smaller $\alpha = 1$ than those in Scenario 2.

The goal of the cache allocation strategy is to find the optimal cache allocation under different network scenarios, aiming at the improvement of caching performance and reduction of traffic loads. As a result, the optimal goal is represented by two metrics, the global cache hit ratio H and the count of missing requests that propagate the traffic into network core and increase the delay. Because the missing request streams in the network is highly related to the cache hit ratio per node, it could be redundant to consider both metrics during the optimisation process. Therefore, the optimisation problem can be expressed as

$$
\begin{aligned}
&\max_{C_{tot},V}\{\sum_{v_n \in V} \omega_n H_{v_n}\} \\
&\text{s.t. } \forall v_n \in V : C_{min} \leqslant C_{v_n} \leqslant C_{max} \\
&C_{min} = 2000, \forall v_n \\
&C_{max} = \max\{C_{tot} - (|V| - 1)C_{min}, P_{v_n} \cdot C_{tot}\}, \forall v_n \\
&\text{and } \sum_{\forall v_n \in V} C_{v_n} \leqslant C_{tot}
\end{aligned}
$$

where C_{tot} is defined as the overall cache budget in the network. To facilitate the analysis, the cache size is quantised in the unit of chunk. To avoid poor fairness, the minimum cache size at each node is set to $C_{min} = 2000$ chunks. During the optimisation, if the sum of the optimised cache amount is greater than C_{tot}, the cache size should be normalised to ensure the overall cache amount is less or equal to C_{tot}. P_{v_n} is the normalised proportion of the cache size of v_n to the total cache size, C_{tot}. The cache hit ratio H_{v_n} in the objective function can be represented as a function of service cache hit ratio, $h^k_{v_n}$, which has been investigated in the model. As a result, the analytical model is employed to solve the optimisation problem by

calculating the local and global cache hit ratio under different network configurations. The objective function can be rewritten as

$$\text{Obj\_Fun} = max\{\sum_{\forall v_n \in V} \omega_n \sum_k q_{k,n} h_{v_n}^k\} \tag{4.27}$$

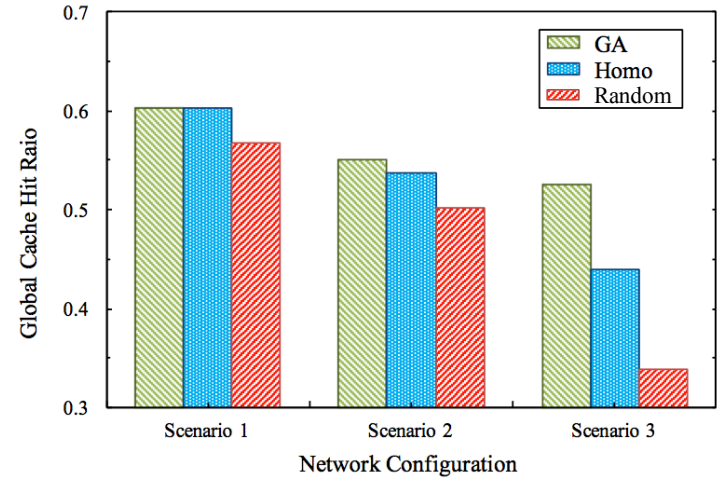
where v_n denotes a node in the Abilene network, ω_n denotes the weight of the node which depends on the intensity of requests, and q_k represents the popularity of content.

The formulated cache allocation optimisation problem is solved using genetic algorithm (GA). Since the goal of the optimisation is to yield a global maxima of cache hit ratio under different network configurations, GA as an evolutionary algorithm is able to obtain a possibly global optimum answer by a finite number of evolution steps. GA uses random search in the decision space, which is called chromosomes, via selection, crossover and mutation operators in order to reach its goal.

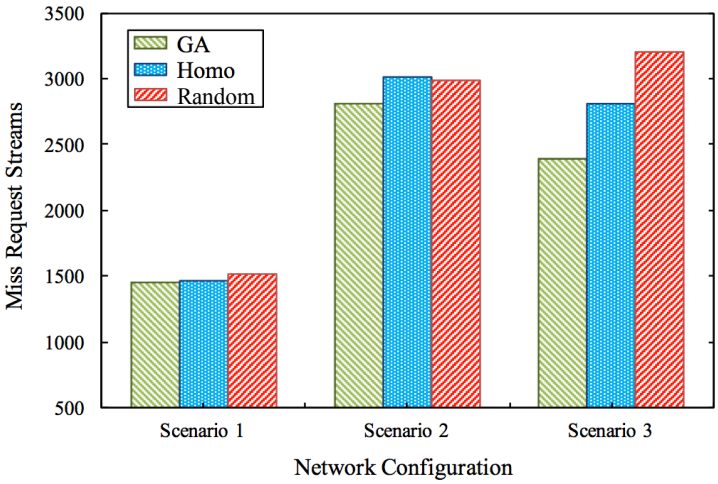
Before using the GA to solve the optimization problem, representation of a chromosome needs to be defined. The chromosome in this problem is a combination of cache allocation of each node. Real value is used to reflect the actual cache size, so the fitness values in GA can be directly computed. In this paper, the fitness function is equal to the objective function because the chromosome with the best cache allocation is the result we seek.

The optimisations are conducted under the three different network scenarios defined above, and the results are evaluated by comparing the proposed method with the homogeneous (Homo) and random allocation methods. Fig. 4.8a and 4.8b depict that the caching allocation strategy increases the global cache hit ratio and reduces the overall traffic in all three scenarios. The corresponding cache allocation results are shown in Fig. 4.9.

Some important observations are found: (i) As illustrated in Fig. 4.8, in the first scenario with the homogeneous network configuration, the improvement brought by the optimal cache allocation is little. But as the network becomes more complex, the improvement becomes significant. In particular, for the third scenario, the global



(a) The optimisation results: hit ratio



(b) The optimisation results: missing requests

Fig. 4.8 The optimal results of proposed allocation mechanism under three different scenarios.

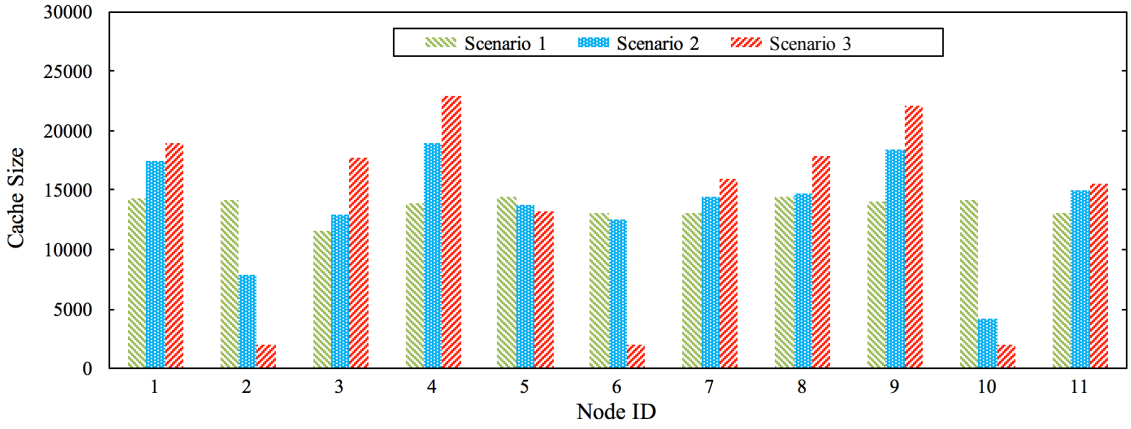


Fig. 4.9 The optimal cache size allocation of the proposed strategy under three different network scenario

cache hit ratio increases up to 55% compared to the random cache allocation, while the number of missed requests is 25.6% less than the random allocation method.

The reason behind this is that in the homogeneous network every node has similar importance, therefore, to achieve a higher cache hit ratio the cache resource allocated to each node is close, as shown in Fig. 4.9. As the network becomes more complex, the importance of each node is different, while the proposed allocation strategy is able to consider the difference and thus achieve a better performance. (ii) Under the homogeneous network (Scenario 1), according to the cache allocation result in Fig. 4.9, a larger cache allocated to the nodes in the middle of the path (Node 1,2,4,5,8,10) leads to a higher gain. This is due to the fact that the middle nodes connecting the edge nodes and core nodes have larger node degrees, which means they are more important than the others. As a result, a larger cache size allocated to these nodes will reduce the overall number of missed requests. (iii) Through observing the cache allocation in Fig. 4.9, we can see that the popularity of content has more impact on the cache allocation than the request intensity. In the third scenario, Node 5 receives the highest number of content requests, but the three nodes with the smallest cache sizes are Nodes 2, 6 and 10. Because these nodes have a flat population distribution (small $\alpha = 1$), which means that the requests for popular and unpopular contents are similar. Learning from the analysis in Section V, small α leads to a low cache hit ratio, which results in very limited gain by the increase of cache size at these nodes.

4.5 Summary

In this chapter, an analytical model has been developed to investigate the caching performance of ICN with arbitrary topology with heterogeneous cache sizes and content popularity. The cache hit ratio at any node of ICN is derived as the key performance metric. Simulation experiments have been performed to validate the effectiveness and accuracy of the analytical model. This accurate and simple model has been used as a cost-effective tool to gain the insights of the impact of key network and content parameters, such as cache size, topology, content popularity, and content size on the caching performance in ICN. As an application of the analytical model, a cache allocation strategy has been proposed to obtain optimal cache allocations under different scenarios. The results show that the model-based strategy can

achieve better caching performance than the default homogeneous and random caching allocation strategy. Some observations have been found from the results: the popularity distribution of content has more impact on the cache allocation than the request intensity; the service delay, ARTT, is mainly determined by type of services.

Chapter 5

Cost-Aware QoS Optimisation for Caching of ICN for Multi-Services

5.1 Introduction

In order to alleviate the pressure of high demands for network bandwidth and service quality posed by various services, a key feature of ICN is to provide transparent, ubiquitous in-network caching to improve network resource utilisation and reduce service latency [39]. Caching has been already deployed in present Internet, and caching theory and techniques to optimise the caching system have been extensively studied. However, in-network caching differs from traditional Web caching in which cache is transparent to applications and content to be cached is finer grained. This poses new challenges to be addressed for in-network cache management and optimal caching strategies: 1) Various services differ considerably in their request rate, content size and popularity, this heterogeneity requires ICN to efficiently share and allocate cache resources among different services. 2) Requests in traditional file-based caches are typically assumed to follow the independent reference model (IRM), which is not valid in ICN caching where correlations are common among requests in an arbitrary cache network.

Apart from the above challenges of in-network caching, ICN has been suffering the realistic deployment issue in service provide networks due to the lack of ICN capable hardware. Thanks to the emerging network technologies, Software-Defined

Networking (SDN) and Network Functions Virtualisation (NFV), which bring programmability and scalability to the network, ICN can be integrated into the current Internet without deploying new hardware. However, to manage such flexible network requires the ability to be aware of the current network status, such as traffic pattern, content distribution and topology. The lack of such ability may result in unreasonable resource allocation and low utilisation [95]. Therefore, the ICN demands dynamical network resources management, especially the caching resource.

In-network caching is considered as an integral part of ICN, and can benefit both Internet Service Providers (ISPs) and end users. It reduces the traffic load of services for ISPs and alleviate network congestions. In the mean time, it improves the Quality of Service (QoS) for end users and reduces service latency. ICN caching has been widely studied in terms of caching strategies [3, 30, 36, 40, 75] and caching performance [32, 46]. Besides, cache allocation optimisation has received much attention. [44, 85, 86] all seek to address the challenges that where should cache resources be placed and how much cache should be allocated. However, the economic aspect of ICN has received marginal consideration so far [12]. But It is vital to understand the potential cost-effectiveness of ICN before its widely deployment in ISP's network. To address this problem, [106] proposes a game-based pricing model that provides economic incentives for caching and sharing content in ICN. [37] also uses game theory to study a price-convex demand-response pricing model. Both works focus on the interaction between different players in the network and does not consider the cache design and performance. The cost and performance are treated as orthogonal problems. [92] proposes two models to investigate the impact of content retrieval cost on the caching design. But the paper makes strong assumptions for the models such as unrealistic content requests and simplified topology.

In order to foster the practical deployment of ICN in next-generation network by Internet service providers, it becomes crucial to understand the performance and cost bounds of ICN. To this end, this chapter proposes a cost-aware caching mechanism to study the performance and cost of ICN and investigates the inner association between them. The novelty of this chapter is threefold:

(i) Two new models are developed to investigate the relation between economic aspect and network performance. The cost model takes into account installation and operation cost of ICN under a realistic ISP network scenario. The QoS model considers multiple key metrics and is formulated under an arbitrary network topology with heterogeneous bursty content requests.

(ii) Arbitrary network topology, heterogeneous bursty content requests, different content popularity distributions, and probability caching are considered to provide a more practical ICN environment.

(iii) A cost-aware caching mechanism that jointly optimises two conflicting goals derived from the above models is proposed. A multi-objective evolution algorithm is adopted to find the optimal caching placement. Numerical results show the effectiveness of the proposed mechanism in achieving cost-efficiency and QoS guarantee in ICN caching.

The remainder of the chapter is organised as follows. Section 5.2 is devoted to the design of the cost model and QoS model followed by the analysis of cache performance. Section 5.3 presents the proposed cost-aware caching mechanism and solution for the formulated multi-objective optimisation problem. The effectiveness of the proposed mechanism is evaluated by numerical experiments in Section 5.4. Finally, Section 5.5 concludes the chapter.

5.2 System Model

In this section, the parameters of ICN used in the system model are described first. Then we develop two conflict models, namely the cost model and the QoS model, to investigate the connection between network cost and service quality. The cost model calculates the cost of nodes under various location-based pricing schemes while the QoS model characterises the QoS of ICN nodes on an arbitrary network topology under bursty content requests.

5.2.1 System Parameters

In a realistic service provider network, cache performance is determined by multiple parameters, such as cache size, traffic pattern, cache strategy, network topology, and content catalog and popularity distribution. Before designing the cost and QoS model, the parameters and assumptions that take into account the unique characteristics of ICN used in this chapter are introduced. Tab. 5.1 summarises the notations used in this chapter.

Caching strategy

The caching strategy includes cache decision policy and cache replacement policy. Cache decision policy determines whether and where contents are to be cached. The Leave Copy Probabilistically (LCP) where contents are placed in caches at random with a probability $q_n \in [0, 1]$, is considered in the chapter. For each node, the probability varies according to the importance of the node. LCP is chosen in light of its simplicity yet effectiveness[114]. By increasing q_n to 1, LCP becomes the Leave Copy Everywhere (LCE) policy which is the standard policy in ICN [17]. Since caching operation needs to be line-rate and services running on caches are diverse, complex cache coordination policy are not suitable due to the high complexity and communication overhead [11]. The chapter considers Lease Recently Used (LRU) policy cache replacing policy on each node, which has low complexity and can be implemented at line speed.

Traffic pattern

In the realistic network, there are multiple types of services that generate various traffic patterns. The traffic patterns have huge impact on the performance caching, since services have different characteristics that lead to different resources and QoS requirements. Multimedia services, such as video and audio services have dominated the network, which results in a bursty characteristic within network traffic. To capture the bursty nature of ICN traffic, Markov-modulated Poisson process (MMPP) is used to represent the arrivals of content requests. MMPP is a doubly stochastic

process with the arrival rate varying according to an irreducible continuous-time Markov chain [109]. It is capable of modelling the bursty content requests because it captures the time-varying arrival rate of various services via two matrices. The arrival rate matrix Λ_n represents the intensity of content requests of different states at node n, while the infinitesimal generator matrix Q_n represents the underlying Markov transition process between the states. For generalisation, we do not specify a concrete traffic pattern classification. Instead, we assume there are K types of services. The request process for every service is described by an independent MMPP process with average arrival rate λ_k.

Popularity distribution

The caching performance depends crucially on the relative popularity of different services. Therefore, the above traffic patterns are used in combination with the Zipf law of content popularity, which is frequently observed in traffic measurements of the Internet services and widely adopted for characterising the content popularity in performance evaluation studies [32, 75]. In this work, we consider a simple Zipf law where contents in the k-th most popular content are requested with the probability ρ^k that is proportional to $1/k^\alpha$. The exponent α characterises the skewness of popularity.

Topology

Like introduced in the last chapter, an ICN network with arbitrary topology [11], which is more realistic compared to specific hierarchical or flat topology is considered when building the models. In an arbitrary topology, the dependence among caches become more complex, because missing requests are forwarded to different directions that increase the degree of correlation.

5.2.2 Cost Model

ICN aims for increasing the efficiency of content distribution, but whether it can be cost-effective for service providers is a key enabler for its realistic implementation.

Table 5.1 Summary of notations

Parameter	Meaning
N	Number of ICN nodes
K	Number of different types of services
$Cost_n$	Total cost for ICN node n
$Cost_n^{\text{CAPEX}}$	Capital expenditure for deploying node n
$Cost_n^{\text{OPEX}}$	Operational expenditure for running node n
p_e^{cap}	Device cost for an ICN node
$p_{c,n}^{cap}$	Caching unit cost for ICN node n
$p_{l,n}^{cap}$	Bandwidth unit cost for ICN node n
C_n	Cache size of node n
BW_n	Amount of bandwidth supported by node n
$p_{c,n}^{op}$	Cache unit cost for operation at node n
$p_{s,n}^{op}$	Unit cost for retrieving a content at node n
$p_{t,n}^{op}$	Unit cost for forwarding a content request at node n
r_n	Number of requests satisfied at node n
$\overline{r_n}$	Number of requests forwarded from node n
λ_n^k	Arrival rate of requests for contents of service k at node n
$\lambda_{tot,n}^k$	Combined content request rate for service k at node n
$h_{v_n}^k$	Cache hit ratio for requests of contents of service k at node n
S_k	Average delay of service k
$s_{m,o}^k$	Delay of a request from node m for content o of service k
Ω	Set of contents in the network
Ω_k	Total number of different contents of service k
$\boldsymbol{P_k}$	Set of cache hit rate for service k at nodes
$\boldsymbol{L}$	Set of link delay
σ_n	Fraction of requests at node n to the total network traffic
$\boldsymbol{p_{m,n}^k}$	Set of hit probability for service k at each node along the path from m to n
$\boldsymbol{l_{m,n}}$	Link delay of each segment along the path from m to n
$J_k(n)$	Jitter of service k at node n
$D_{k,n}$	Variation of delay between two successive requests for service k at node n
$\eta_{k,n}$	Differential delay parameter for service k at node n
α_n	Zipf exponent characterizing the skewness of popularity
τ_n^k	Time interval for generating a request missing process
ρ_n^k	Fraction of requests for service k at node n
Λ_n	Arriving rate matrix for node n
Q_n	Infinitesimal generator of requests for contents arriving at node n
C_{tot}	Total cache resource budge of network

Bearing in mind the cost-efficiency issue, we develop a new model to evaluate the cost for ICN nodes. The cost includes two parts, capital expenditure (CAPEX) and operational expenditure (OPEX). Consequently, the cost for an ICN node n can be written as

$$Cost_n = Cost_n^{\text{CAPEX}} + Cost_n^{\text{OPEX}} \tag{5.1}$$

The CAPEX represents the installation expenditure of ICN devices. Assuming a fixed device cost (p_e^{cap}), the rest of cost is determined by two other specifications, i.e, the cache size and the bandwidth. We denote C as the cache size of an ICN node, and BW as the bandwidth supported by that node. The unit cost for cache and bandwidth are denoted as p_c^{cap} and p_l^{cap} that are various depending on locations. Therefore, the cost of CAPEX for one node, $Cost_n^{\text{CAPEX}}$ is expressed as

$$Cost_n^{\text{CAPEX}} = p_e^{cap} + p_{c,n}^{cap} \cdot C_n + p_{l,n}^{cap} \cdot BW_n \tag{5.2}$$

The OPEX includes the caching cost and traffic cost. The caching cost, $Cost_c^{op}$, is composed by two factors, the cost of storing that is proportional to cache size, and the cost of retrieving content from the cache when incoming requests are hit in the cache. The traffic cost, $Cost_t^{op}$, is proportional to the traffic forwarded to the neighbour nodes when requests are not satisfied by the cache. Because both costs depend on the unit cost related to the location, such as electricity tariff and rental fee, we denote the unit price for caching cost, retrieving cost and forwarding cost as p_c^{op}, p_s^{op} and p_t^{op}, respectively. Then the OPEX for node n is given by

$$\begin{aligned} Cost_n^{\text{OPEX}} &= Cost_{c,n}^{op} + Cost_{t,n}^{op} \\ &= p_{c,n}^{op} \cdot C_n + p_{s,n}^{op} \cdot r_n + p_{t,n}^{op} \cdot \overline{r_n} \end{aligned} \tag{5.3}$$

where r_n denotes the amount of content requests that are satisfied by the cache at node n, and $\overline{r_n}$ denotes the requests that are missed at the current node and forwarded to the adjacent nodes. To calculate the amount of requests that are

satisfied by the cache, we denote the arrival rate of requests for service k as λ^k, and the hit ratio for such service as $h^k_{v_n}$. Since the arrival rate and cache hit ratio are different for each service at each node in a realistic scenario, we firstly consider the cost for serving one kind of service. Let λ^k_n denote the number of requests for service k arrived at node n, and h^k_n as the proportion of those requests that are hit in the cache, then we have

$$\begin{aligned} r_n &= \sum_{k \in K} \lambda^k_n \cdot h^k_{v_n} \\ \overline{r_n} &= \sum_{k \in K} \lambda^k_n \cdot (1 - h^k_{v_n}) \end{aligned} \tag{5.4}$$

Eq. (5.4) is only valid under the independent reference model (IRM), which means that the incoming requests for the same content at a node have the same probability and does not depend on any other sources. However in a network of caches, because the miss requests at one node are forwarding and becoming part of the incoming demand of its neighbours, the effects of miss requests from neighbouring nodes should be considered, which can be given by

$$r^k_n = \lambda^k_{tot,n} \cdot h^k_{v_n} \tag{5.5}$$

$$\lambda^k_{tot,n} = \lambda^k_n + \sum_{n' \in E_{<n,n'>}} \overline{r^k_{n'}} \tag{5.6}$$

$$\overline{r^k_n} = \lambda^k_{tot,n}(1 - h^k_{v_n}) \tag{5.7}$$

Accordingly, Eq. (5.4) should take into account the additional content requests and be updated as

$$\begin{aligned} r_n &= \sum_{k \in K} (\lambda^k_n + \sum_{n' \in E_{<n,n'>}} \overline{r^k_{n'}}) h^k_{v_n} \\ \overline{r_n} &= \sum_{k \in K} (\lambda^k_n + \sum_{n' \in E_{<n,n'>}} \overline{r^k_{n'}})(1 - h^k_{v_n}) \\ &= \sum_{k \in K} (\lambda^k_n + \sum_{n' \in E_{<n,n'>}} \overline{r^k_{n'}} - r^k_n) \end{aligned} \tag{5.8}$$

From Eq. (5.8) we can see that r_n^k and $\overline{r_n^k}$ are functions of each other, and both of them relied on the exogenous arriving rate λ_n^k. In Sec. 5.2.4, we show that an iterative method can be used to solve the equations.

5.2.3 QoS Model

Delay of service and jitter are two of the most important factors to measure QoS. For delay-sensitive services, such as video and voice, the QoS is largely determined by service delay. In the mean time, jitter, variations in packet transit delay, also has significant impact on the experience of users. Therefore, we use service delay and jitter to monitor service quality. The following sections introduce how they can be calculated.

Service delay

The delay of service is measured by the average time interval for the delivery of a content in that service. Let S_k denote the delay for service k, and it can be written as a function of

$$S_k = \mathcal{F}(\boldsymbol{\Omega_k}, \boldsymbol{P_k}, \boldsymbol{L}) \tag{5.9}$$

where Ω_k is a set of content objects, o^k, contained in service k, i.e.,

$$\Omega_k = \{o_i^k | i \in \Omega, k \in K\} \tag{5.10}$$

$\boldsymbol{P_k}$ is a vector with the elements denoting the probabilities that the requested service is satisfied by the nodes on the path to content providers' servers, hence

$$\boldsymbol{P_k} = [h_{v_1}^k, h_{v_2}^k, \ldots, h_{v_N}^k] \tag{5.11}$$

We further define a vector $\boldsymbol{p_{m,n}^k}$, which denotes the probability that requests for service k arriving at node m is satisfied by node n. $\boldsymbol{p_{m,n}^k}$ can be easily generated

from $\boldsymbol{P_k}$:

$$\boldsymbol{p^k_{m,n}} = [1 - h^k_{v_m}, 1 - h^k_{v_{m+1}}, \ldots, h^k_{v_n}] \tag{5.12}$$

where v_{m+1} means the next hop towards node n from node m. L is a vector denotes the link delay along the path, which can be written as

$$\boldsymbol{L} = [l_{1,2}, \ldots, l_{N-1,N}] \tag{5.13}$$

where $l_{i,j}$ is the link delay between node v_i and node v_j. Similarly, we define the link delay of each segment of the path from v_m to v_n as $\boldsymbol{l_{m,n}} = [l_{m,m+1}, \ldots, l_{n-1,n}]$. Finally, we can further express Eq. (5.9) as

$$S_k = \sum_{m \in N} \sigma_m \cdot S_{k,m} = \sum_{m \in N} \sigma_m \cdot |\Omega_k| \cdot s^k_{m,o} \tag{5.14}$$

where σ_m is the ratio of the incoming requests at node m to that at the whole network, $|\Omega_k|$ is the number of requested objects, and $s^k_{m,o}$ denotes the average delay for retrieving a single object o in service k at node m. It can be computed as

$$\begin{aligned} s^k_{m,o} &= h^k_{v_{m,o}} + (1 - h^k_{v_{m,o}}) h^k_{v_{m+1,o}} l_{m,m+1} + \ldots \\ &\quad + (1 - h^k_{v_{m,o}})(1 - h^k_{v_{m+1,o}}) \ldots (1 - h^k_{v_{m+d-1,o}}) \\ &\quad h^k_{v_{m+d^k_m},o} \sum\nolimits_{i=1}^{m+d^k_m} \boldsymbol{l}_{m,m+d^k_m}(i) \\ &= \sum_{n=m}^{m+d^k_m} \prod_{q=1}^{n-m} \boldsymbol{p}^k_{m,n}(q) \sum_{i=1}^{n-m} \boldsymbol{l}_{m,n}(i) \end{aligned} \tag{5.15}$$

where d^k_m denotes the maximum distance (i.e., the number of hops) travelled by a request arriving at node m for service k. Note that the contents in the same service are requested under the same probability, for the sake of simplicity, we assume that the cache hit ratio is independent of the specific object, thus, $h^k_{v_m}$ is equivalent to

$h^k_{v_{m,o}}$. At last, we can obtain the average delay for service k as

$$S_k = |\Omega_k| \sum_{m \in N} \sigma_m \sum_{n=m}^{m+d^k_m} \prod_{q=1}^{n-m} \boldsymbol{p}^k_{m,n}(q) \sum_{i=1}^{n-m} \boldsymbol{l}_{m,n}(i) \tag{5.16}$$

Jitter of services

The jitter of service is derived as the variation in inter-arrival time between the sending and reception for a request in that service. In ICN, the deviation of the inter-arrival time is caused by the fact that the requested contents can be retrieved by any nodes along the path to the remote server storing the contents. Furthermore, multiple services share the network capacity and update the caches, thus content requests will be satisfied by different nodes over time.

Let J_k denote the jitter of service k, and it is highly related to the cache hit ratio and the routing path. To calculate J_k, we consider the jitter of service k at node n, denoting as $J_k(n)$. In realistic network, the value of $J_k(n)$ is continuously calculated for each request arrived at node n as

$$J_k(n) = J_k(n) + (|D_{k,n}(i-1,i)| - J_k(n))/\phi \tag{5.17}$$

where $D_{k,n}(i-1,i)$ is the difference of packet delay between two successive requests for service k arrived at node n. The parameter ϕ which provides a low pass filtering is used to remove the impact of spikes during the calculation. The different D is determined by cache hit ratio and cache replacement strategy. As a result, in the QoS model, the long-term average $D_{k,n}(i-1,i)$ can be derived as

$$\mathbb{E}(D_{k,n}(i-1,i)) = \mathbb{E}(inter_{k,n}(o_{i-1}) - inter_{k,n}(o_i)) = \eta_{k,n} \cdot s^k_{n,o} \tag{5.18}$$

The successive requests for contents belonging to the same service is highly correlated. Moreover, the model considers average cache hit ratio that is independent of specific content. Therefore, the two requests have very similar cache hit ratio at the nodes along the path to the server. The jitter is caused by the sharing of multiple services. More specifically, the cache of an ICN node is updated by multiple services

going through it, so there is a case that requested content is evicted from cache due to requests from other services, although the average cache hit ratio may be still the same. To calculate this difference, we define η as the differential parameter, which is determined by the service requesting rates at a node.

$$\eta_{k,n} = \frac{\sum_{i \in K} \lambda^i_{tot,n}(1 - h^i_{v_n}) \cdot |\Omega_k|}{C_n}(1 - \frac{\lambda^k_{tot,n}}{\sum_{i \in K} \lambda^i_{tot,n}}) \tag{5.19}$$

The left part is the proportion of the missing requests to the cache size, which indicates that how frequent a cached object is replaced. The right part shows the request intensity of the service accounts for the whole arrival requests at node n. When the proportional is large, which means that the service is popular and the requests refresh the cache to keep those contents, the difference parameter η should be small, hence we use the compliment.

To calculate $J_k(n) = 0$, we start with $J_k(n) = 0$, and then iterate Eq. (5.17) until the variation is close to 0. At last, similar to service delay S_k, the jitter of service k can be derived as

$$J_k = \sum_{n \in N} \sigma_n J_k(n) \tag{5.20}$$

5.2.4 Cache Performance Analysis

Cache performance is crucial for quantifying the metrics defined in the cost and QoS models. Cache hit ratio is one of the most important metric to determine the cost and QoS models. With the parameters and assumptions in Section 5.2, we evaluate the performance of caching in ICN under arbitrary topology and bursty traffic. We leverage and extend the model that we developed in [115] and the cache hit ratio for service k at node n can be derived as

$$h^k_{v_n} = 1 - \mathbb{P}(Req_{v_n}(\tau_{n,k}) \geqslant C_n) \tag{5.21}$$

where $Req_{v_n}(\tau^k_n)$ denotes that the number of different content requests arriving at node n between two subsequent requests of the same content in service type k during

an inter-arrival time τ_n^k. τ_n^k is the cache eviction time and the generation of a caching miss process. It largely depends on the traffic pattern, popularity distribution and caching strategy. Under the MMPP bursty content requests, Zipf content distribution, and LRU and LCE caching strategies, τ_n^k can be expressed as a function of

$$\tau_{n,k} = \mathcal{F}(C_n, \Lambda_n, Q_n, \alpha_n, q_n), \quad \forall k \in K \tag{5.22}$$

The bursty arrival rate at node n for service k, λ_n^k can be given by

$$\lambda_n = \pi_n \cdot \Lambda_n \tag{5.23}$$

$$\rho_{n,k} = \frac{1/k^{\alpha_n}}{\sum_{i=1}^{K} 1/i^{\alpha_n}}, k \in K \tag{5.24}$$

$$\lambda_n^k = \rho_{n,k} \cdot \lambda_n \tag{5.25}$$

where π_n is the steady-state vector that subjects to $\pi_n Q_n = 0, \pi_n \boldsymbol{e} = 1$, and ρ_n^k is the fraction of requests for service k at node n. In our previous work [115], the inter-arrival time, $\tau_{n,k}$ had been derived as

$$g_{n,k} = \frac{1}{2} \cdot \Gamma(1 - \frac{1}{\alpha_n})^{\alpha_n} \frac{\sum_{k \in K} \lambda_{tot,n}^k}{\sum_{k \in K} 1/k_n^{\alpha}} \cdot |\Omega_k|^{\alpha_n}$$

$$\tau_{n,k} = (C_n)^{\alpha_n} / g_{n,k} \tag{5.26}$$

The cache hit ratio for service k at node n, $h_{v_n}^k$ can be expressed in the form of

$$h_{v_n}^k = 1 - \beta_{n,k} e^{-u_{n,k}\tau_{n,k}} - (1 - \beta_{n,k}) e^{-v_{n,k}\tau_{n,k}} \tag{5.27}$$

where β_n^k, u_n^k and v_n^k are the parameters derived from MMPP [115]. We omit the details here for the reason of space limit.

In this section, the calculation of cache hit ratio is extended to a more general cache replacement strategy, LCP, where the requested content is copied with probability q_n at each cache along the returning path. Let us consider the hit process for request of service k arriving at node n. The requested content is provided by the cache meaning that during the last time interval $\tau_{n,k}$ (1) either the content has already been stored

in the cache, (2) or a request for the same content arrived and the content was copied into the cache with probability q_n. As a result, we update the $h^k_{v_n}$ following the method used in [114] as

$$\begin{aligned} h^k_{q_n,v_n} &= P(hit) \cdot P(store, copy) \\ &= h^k_{v_n}(P_{n,store}(k) + q_n(1 - P_{n,store}(k))) \end{aligned} \tag{5.28}$$

When a request is arrived, the content already in the cache will definitely generate a hit process. Therefore, the hit rate $h^k_{q_n,v_n}$ is equal to $P_{n,store}(k)$. Then we can have

$$h^k_{q_n,v_n} = \frac{q_n \cdot h^k_{v_n}}{(1 - h^k_{v_n}) + q_n \cdot h^k_{v_n}} \tag{5.29}$$

Due to the arbitrary topology, the change of cache hit ratio at node n also influences other nodes in the network. By assigning $n \in N$ to different value in Eq. (5.29), we get the initial values for the iteration. Then we consider the missing process. After a cache miss for service k is generated at node n', two cases can happen with probabilities,

$$\begin{cases} P(store|miss) & = q_{n'}(1 - h^k_{q_{n'},v_{n'}}) \\ P(no\_store|miss) & = (1 - q_{n'})(1 - h^k_{q_{n'},v_{n'}}) \end{cases} \tag{5.30}$$

Therefore, if node n receives a forwarded request for content in k from n' since the last forwarded same request, the probability of this includes two cases: (1) the requested content was cached, but was evicted from the cache during the time $\tau_{n,k}$, and (2) the requested content was not cached. So we have

$$P_{n',miss}(k) = q_{n'}(1 - h^k_{q_{n'},v_{n'}})(1 - \rho_{n',k}) + (1 - q_{n'})(1 - h^k_{q_{n'},v_{n'}}) \tag{5.31}$$

Then the equations (5)-(7) are re-calculated accordingly as,

$$r_{n,k} = \lambda^k_{tot,n} \cdot h^k_{q_n,v_n} \tag{5.32}$$

$$\lambda^k_{tot,n} = \lambda^k_n + \sum_{n' \in E_{<n,n'>}} \overline{r_{n',k}} \tag{5.33}$$

$$\overline{r_{n,k}} = \lambda^k_{tot,n} \cdot P_{n,miss}(k) \tag{5.34}$$

After the new $\lambda^k_{tot,n}$ is derived, Eq. (5.26) (5.27) are updated as well. The iteration is continuing until the value of $h^k_{q_n,v_n}$ reaches the steady state. At last, the value of $h^k_{q_n,v_n}$ is the cache hit ratio for each ICN node, then $h^k_{v_n} = h^k_{q_n,v_n}$ is used to quantify the cost and QoS model.

5.3 Cost-Aware Caching Allocation Scheme

In this section, we first provide the rational behind our method, showing that the cost efficiency and QoS guarantee are conflict goals. The classic caching strategies focusing on increasing the cache hit ratio which results in better QoS may be detrimental in terms of cost. Next, we propose a cost-aware optimal caching mechanism that integrates the cost model and the QoS model into a multi-objective optimisation framework, which is solved through an evolutionary algorithm.

5.3.1 Multi-objective Optimisation Goals

Most recent caching schemes aim at optimising the cache hit ratio, however, according to the two models presented in Section 5.2, a larger cache will improve the performance of ICN, however, it also increases both CAPEX and OPEX for service providers. Therefore, the goals improving the QoS and reducing the cost are conflicting by nature. Costs depend on locations and relationships among nodes, while QoS demands are related to services. As a result, when we allocate caching resources to ICN nodes, there is a trade-off between cost and QoS. We design two goals to investigate this

trade-off in cost-aware caching.

$$Obj_{Cost} = min\{\sum_{n \in N} Cost_n\} \tag{5.35}$$

$$Obj_{QoS} = min\{\mathbb{E}(S_k), \mathbb{E}(J_k), \forall k \in K\} \tag{5.36}$$

Obj_{Cost} aims to find the cache placement strategy that leads to the minimum cost of the whole network, while Obj_{QoS} targets at minimising the service delay and jitter for various services. The solution for either goal must yields to the following design constraints:

(i) Cache budget: the total size of cache to be deployed in the network has an upper bound, C_{tot}.

(ii) Service Level Agreement (SLA): there are requirements of QoS for various services. The maximum delay and jitter for each type of service is represented as $\widehat{S_k}$ and $\widehat{J_k}$.

(iii) Bandwidth capability: the volume of traffic being transmitted at one node should be less than the bandwidth of that node BW_n.

(iv) Cost budget: for service providers, the cost of network should be kept low to make profits and remain competitive. Therefore, a cost budget $Cost_{max}$ is set to control the total cost.

In order to guarantee the QoS, the main objective is to optimise the cache performance, which, in turn, requires the improvement of global cache hit rate. In practice, it is desirable to place large amount of caching resource at the nodes that receive high volume of requests. In such a case, the cache can be fully utilised and the delay and jitter for retrieving requested contents can be reduced. Nevertheless, more caches lead to higher cost, so it is difficult to balance the network cost and QoS performance. Therefore, we transform the trade-off into a multi-objective

optimisation problem which can be described as follows:

$$\begin{aligned} & min[Obj_{Cost}, Obj_{QoS}] \\ \text{s.t.,} \quad & \forall n \in N : \sum_{n \in N} C_n \leqslant C_{tot}, \quad C_{min} \leqslant C_n \leqslant C_{max} \\ & \forall k \in K : S_k \leqslant \widehat{S_k}, J_k \leqslant \widehat{J_k} \\ & \forall n \in N : \sum_{k \in K} \lambda_{tot,n}^{k} \leqslant BW_n \\ & \forall n \in N : Cost_{tot} = \sum_{n \in N} Cost_n \leqslant Cost_{max} \end{aligned} \tag{5.37}$$

5.3.2 Design of a Cost-aware Caching Mechanism

We propose a cost-aware caching mechanism in order to achieve significant saving in cost and also guarantee the QoS. To achieve this, we leverage the two models developed in Sec. 5.2.2 and 5.2.3 to investigate the inner connection between the two goals and try to find a balance between them.

The goal function of network cost can be written as

$$\begin{aligned} \sum_{n \in N} Cost_n &= \sum_{n \in N} (Cost_n^{\text{CAPEX}} + Cost_n^{\text{OPEX}}) \\ &= \sum_{n \in N} (p_e^{cap} + (p_{c,n}^{cap} + p_{c,n}^{op}) C_n + p_{l,n}^{cap} \cdot BW_n \\ &\quad + p_{s,n}^{op} \cdot r_n + p_{t,n}^{op} \cdot \overline{r_n} \end{aligned} \tag{5.38}$$

So the cost is determined by the cache size, location of node (network topology), intensity of requests (traffic pattern) and cache hit ratio.

While the expectation of delay and jitter can be expressed as

$$\begin{aligned} \mathbb{E}(S_k) = \sum_{k \in K} \rho_k \cdot S_k = \sum_{k \in K} |\Omega_k| \sum_{m \in N} \rho_{m,k} \cdot \sigma_m \\ \sum_{n=m}^{m+d_m^k} \prod_{q=1}^{n-m} \boldsymbol{p}_{m,n}^{k}(q) \sum_{i=1}^{n-m} \boldsymbol{l}_{m,n}(i) \end{aligned} \tag{5.39}$$

$$\mathbb{E}(J_k) = \sum_{k \in K} \rho_k \cdot J_k = \sum_{k \in K} \sum_{n \in N} \rho_{n.k} \cdot \sigma_k \cdot J_k(n) \tag{5.40}$$

Observing from equations, the delay and jitter of service are determined by the cache size, network topology, content distribution, traffic pattern, cache strategy and cache hit ratio.

However, Eq. (5.39)(5.40) are not easy to quantify the fitness of the QoS goal. To this end, we employ the modified sigmoid function [116] to integrate the two metrics and further take the tolerance for service delay and jitter into account. The value of delay and jitter calculated by QoS model is compared with the maximum delay $\widehat{S_k}$ and jitter $\widehat{J_k}$ that users could tolerate. The value of the goal is between 0 and 1, which can facilitate the representation of the QoS perceived by users, with 0 the highest levels and 1 the lowest levels. Actually, from the meaning of sigmoid function, when the value is larger than 0.5, the delay or jitter for the current service has exceeded the maximum tolerant value. Since different services have various QoS preferences, we define the delay tolerant parameter θ_{d_k} and jitter tolerant parameter θ_{j_k} to indicate the users sensitivity to the increase in delay and jitter of service k. Furthermore, the range of variation differs between delay and jitter, which may result in being prejudiced against the overall QoS. For eliminating this prejudice, two balancing factors are applied in the modified sigmoid function to normalise the variation of delay and jitter into the same range. Therefore, the goal of QoS can be expressed as

$$Obj_{QoS} = \sum_{k \in K} \rho_k \left(\frac{\theta_{d_k}}{1 + \exp(-\alpha_d(S_k - \widehat{S_k}))} + \frac{\theta_{j_k}}{1 + \exp(-\alpha_j(J_k - \widehat{J_k}))} \right) \tag{5.41}$$

where the values of two tolerant parameters, θ_{d_k} and θ_{j_k}, specify the impact weights of delay and jitter on the QoS of service k, with $\theta_{d_k} + \theta_{j_k} = 1, 0 \leqslant \theta_{d_k}, \theta_{j_k} \leqslant 1$, ρ_k is the importance of service k to the whole network and can be calculated by $\rho_k = \sum \rho_{n,k} \cdot \sigma_n$, α_d and α_j are the balancing factors, and $\widehat{S_k}, \widehat{J_k}$ are the largest delay and jitter of service k that users could tolerate.

The cache size, topology and traffic pattern have impact on both goals. By allocating different caching resource to each node under various topology and het-

erogeneous traffic, we obtain different values of Obj_{Cost} and Obj_{QoS}. However, the evaluation of all possible cache allocations to find the Pareto optimal results with respect to predefined constraints for a particular network topology with different costs, bursty content requests, and various popularity distributions is very complex, time-consuming, and in many cases exceeds the computation resources of machines. By changing either the size of network or available cache sizes, the time and resource demands for the computation surge dramatically.

To accelerate the convergence speed and maintain the diversity, we use the fast elitist multi-objective evolutionary algorithm: non-dominated sorting genetic algorithm II (NSGA-II). The algorithm uses fast non-dominated sorting, crowding distance, crowded-comparison operator and elitism mechanism, and it is very efficient for finding the Pareto optimal fronts.

The input for the mechanism consists of two parts. One is the population vectors of caching resource allocated for each node. In the case of no prior knowledge, initial population vector is generated according to the uniform random distribution. Since the sum of cache amount may be greater than the total available cache resource C_{tot}, the cache size should be normalised to ensure that the total cache amount is subject to the constraints. The other part containing information associated with the topology, request intensity of nodes, popularity distribution, location-related cost and SLAs is fed into the mechanism to evaluate the objectives.

5.3.3 Computational Complexity Analysis

The complexity of one iteration of the entire scheme is considered. THe computational complexity consists of two parts: the evaluation of the goals through models and the operations of the evolutionary algorithm.

To calculate the Obj_{Cost} and Obj_{QoS} objective, the complexity depends on calculation of cache hit ratio of various services at each node which requires $O(N \times K)$ computations, where N is the size of network and K is number of service types. Because the two goals share some key parameters, the complexity for both goals is $O(2 \times N \times K)$. For the evolutionary algorithm, the non-dominated sorting in

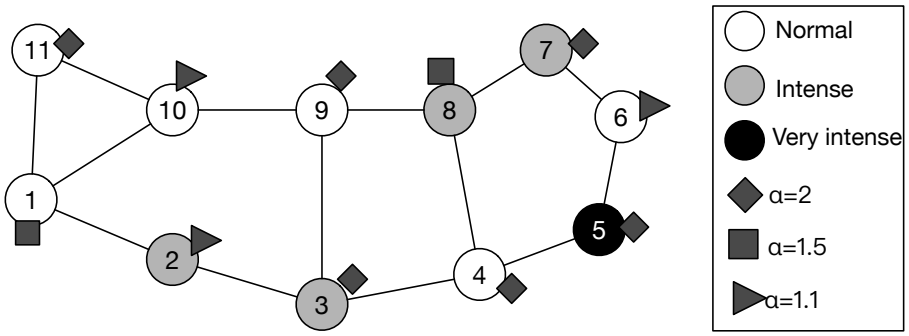


Fig. 5.1 Network topology of Abilene with different content requesting rate and content popularity distribution.

NSGA-II requires $O(2 \times S_p^2)$ comparisons, where S_p is the population size of the first non-dominated front. The select of S_p solutions for next generation using crowding-distance assignment is $O(2 \times S_p \log S_p)$. Therefore, the overall complexity of the NSGA-II is governed by the non-dominated sorting with $O(S_p^2)$. Since each comparison requires the calculation of the two goals, the overall complexity of one iteration is $O(N \times K \times S_p^2)$. So the complexity largely depends on NSGA-II and increases linearly with the size of network and variety of service, which means the proposed scheme can be scaled up for large network and complex applications.

5.4 Numerical Evaluation

In this section, we present numerical simulations using Matlab to demonstrate the effectiveness of the proposed cost-aware optimal caching mechanism. The goal is to find an optimal caching placement strategy for saving the network cost and maintaining the QoS.

5.4.1 Experiment Setup

Network topology. We consider a realistic network called Abilene [113] as the representative network topology, as shown in Fig. 5.1. The network contains $N = 11$ nodes and caching resources will be placed at each node.

Content request. All the nodes in the network receive bursty content requests that are modelled by MMPP. In order to represent a more realistic scenario, the nodes receive various intensity of content requests. As shown in Fig. 5.1, each node

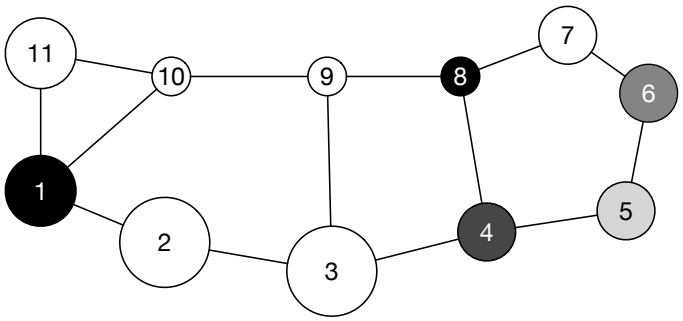


Fig. 5.2 The diversity of node cost based on location shown on the topology. Higher CAPEX nodes are darker in colour and higher OPEX nodes are larger in size.

in the network is connected to a group of users. Three levels of arriving rate are considered. 6 nodes at different locations receive a normal extraneous requesting rate of 10 contents/s. 4 nodes receive a more intense rate with 60 contents/s, while one node receives a very intense rate of 100 contents/s.

Multiple services. We consider $K = 10$ types of services and a content catalog consisting of 10^6 contents. The contents are equally allocated into the 10 different services following a Zipf distribution. Services are classified as video traffic, audio traffic and file transfer traffic with different service constraints. Since some services are more delay sensitive whilst others are jitter sensitive, $\boldsymbol{S} = [\widehat{S_1}, \widehat{S_2}, \ldots, \widehat{S_K}]$ and $\boldsymbol{J} = [\widehat{J_1}, \widehat{J_2}, \ldots, \widehat{J_K}]$ are defined as the service delay and jitter requirement vectors.

Content distribution. We set diverse content popularity at different nodes. Three types of distribution are configured with different Zipf exponent $\alpha = \{2, 1.5, 1.1\}$. α is the skewness of popularity. With a large value of α, few most popular services account for the majority of total requests.

Diversity cost. Nodes at different locations have different CAPEX and OPEX. 5 different types of prices for CAPEX and 4 types of prices for OPEX are illustrated in Fig. 5.2, where the nodes drawn with a larger size denote a higher cost for operation, while the nodes drawn with darker colour represent that these locations are more expensive for installation. Without loss of generality, the unit prices are randomly assign to every node.

The setting for variables are summarised in Tab. 5.2. Furthermore, the total cache budge, $C_{tot} = \{1.0e6, 1.2e6, 1.5e6\}$, is in the unit of content. To avoid the extreme case where the cache size allocated to the node is 0 or being too small,

the minimum cache size is set to $C_{min} = 5000$ contents, and the maximum size is $C_{max} = C_{tot} - (N-1)C_{min}$. The link delay is set to $2ms$ for all connections between ICN nodes. One repository that contains all the contents is connected to node 3. The bandwidth supported by each node and the corresponding cost, BW_n and $p_{l,n}^{cap}$, are set to same values for all nodes as 10^6 and 0.2, respectively. The probability for LCP caching placement strategy is 0.9. The low pass filtering for jitter is set as 16 [117].

Table 5.2 Settings of variables for evaluation

Node	Λ_n[1]	α_n	$p_{c,n}^{cap}$[2]	$p_{c,n}^{op}$	$p_{s,n}^{op}$	$p_{t,n}^{op}$
1	10	1.5	5	1.2	1.6	1.6
2	60	1.1	1	1.5	1.8	1.8
3	60	2	1	1.5	1.8	1.8
4	10	2	4	1.1	1.4	1.4
5	100	2	2	1.1	1.4	1.4
6	10	1.1	3	1.1	1.4	1.4
7	60	2	1	1.1	1.4	1.4
8	60	1.5	5	1	1.2	1.2
9	10	2	1	1	1.2	1.2
10	10	1.1	1	1	1.2	1.2
11	10	2	1	1.2	1.6	1.6

[1] The arrival rate is contents/second.
[2] Cost of a node is expressed as the ratio between cost of the node to that of the cheapest node.

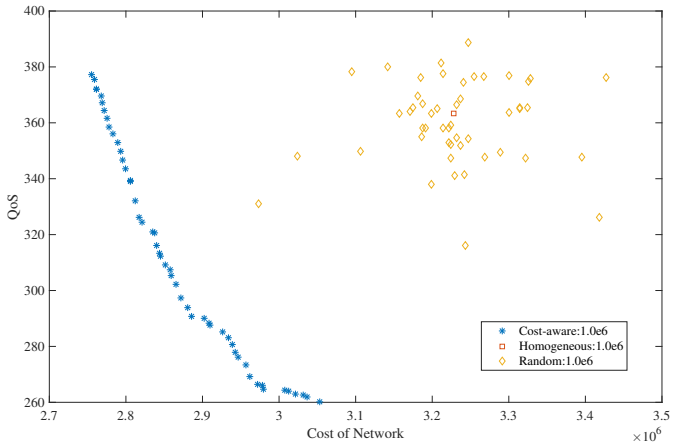
5.4.2 Performance Evaluation

This subsection demonstrates the performance of the proposed optimal caching mechanism. The achieved optimal results are compared with the equal caching allocation which is the default method for ICN and the random caching allocation to verify the effectiveness of the proposed mechanism. The population size is set as 50 and the maximum generation is set as 500 for the evolutionary algorithm. The crossover probability for NSGA-II is set to 0.9, and the mutation probability is $1/N$.

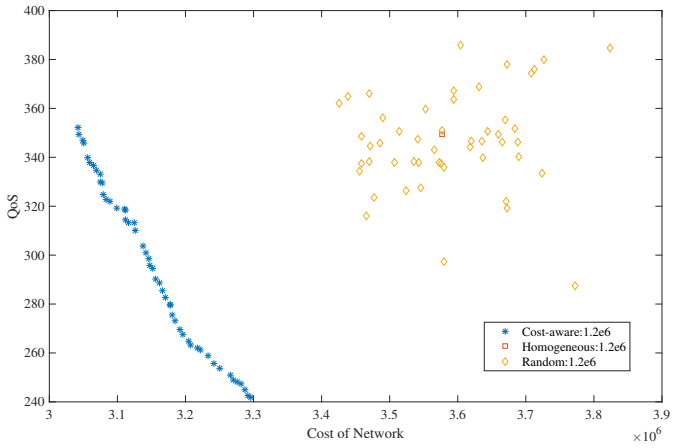
Fig. 5.3 depicts the Pareto optimal set of the Obj_{Cost} versus Obj_{QoS} trade-off under three different cache budgets. The optimal results show that the cost of network and QoS are conflicting goals. The figure illustrates that with some sacrifice in the cost of network, a significant enhancement in QoS can be achieved. For example, by increasing the total cost for 4%, a significant improvement up to 25% can be achieved in service quality. The optimal set can be leveraged by network managers to select the most appropriate solutions for different purposes. Furthermore, observing from Fig. 5.3, under all three cache budgets, a majority of the optimisation results are better than the homogeneous and random cache allocation methods, especially in terms of the network cost. Therefore, the proposed cost-aware caching scheme can attain an economical ICN and in the same time improve or guarantee the service performance.

Figs. 5.4 and 5.5 illustrate the results for each goal under different caching allocation methods. The results of the cost-aware caching scheme in two figures are the mean values of the Pareto frontier set. The results of random method are the mean values of 50 allocation cases. The figures show that the proposed cost-aware caching mechanism outperforms the homogeneous and random cache placement methods in terms of both network cost and service delay. As shown in the figures, the cost of network is around 12.5% less than the homogeneous and random methods under $C_{tot} = 1.0 \times 10^6$, 13.5% less than the homogeneous and random methods under $C_{tot} = 1.2 \times 10^6$, and 15.1% less than the homogeneous method and 14.2% less than the random method as C_{tot} increases to 1.5×10^6. As for the QoS aspect, the cost-aware caching placement has a noticeable improvement of the quality by 16.4% and 15.4% comparing to homogeneous and random methods with $C_{tot} = 1.0 \times 10^6$, 18.4% and 17.5% better under $C_{tot} = 1.2 \times 10^6$, and 18.2% and 16.1% better under $C_{tot} = 1.5 \times 10^6$.

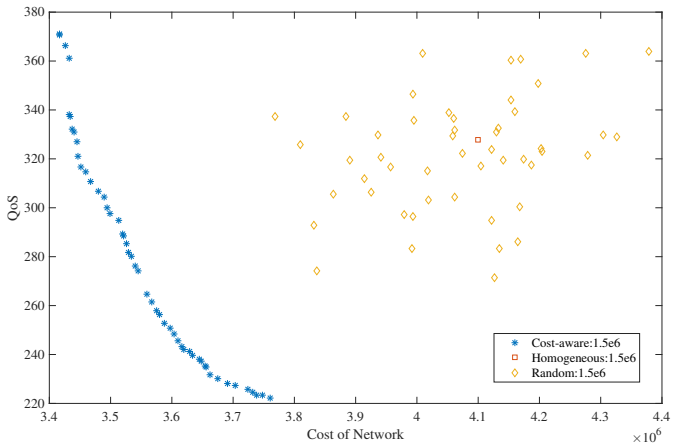
Fig. 5.6 shows the cache allocation result of the median of Pareto optimal set under $C_{tot} = \{1.0e6, 1.2e6, 1.5e6\}$. There are different factors that could impact the allocation. Under three different cache budget, node $3, 5, 7, 9, 11$ are allocated larger caching resource than others, which means that content distribution has the most



(a) Cache budget $C_{tot} = 1.0e6$



(b) Cache budget $C_{tot} = 1.2e6$



(c) Cache budget $C_{tot} = 1.5e6$

Fig. 5.3 Pareto optimal set under the goals (cost of network vs QoS) comparing with homogeneous and random cache allocation methods with cache budget $C_{tot} = \{1.0e6, 1.2e6, 1.5e6\}$.

significant impact among all the factors. Because these nodes have various request intensity and different CAPEX and OPEX, but the same content popularity with Zipf exponent $\alpha = 2$. When α is larger, the content distribution is more skewed. This implies that by allocating more caching resource to the node, the gain of caching

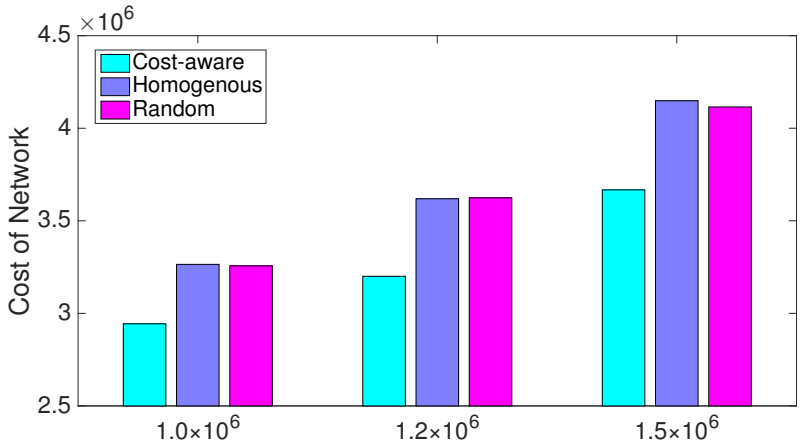


Fig. 5.4 Comparing of network cost among three cache allocation methods under caching budget $C_{tot} = \{1.0e6, 1.2e6, 1.5e6\}$.

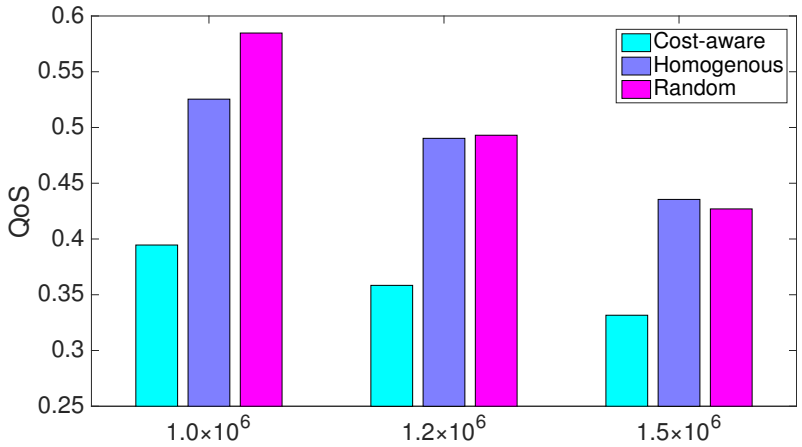
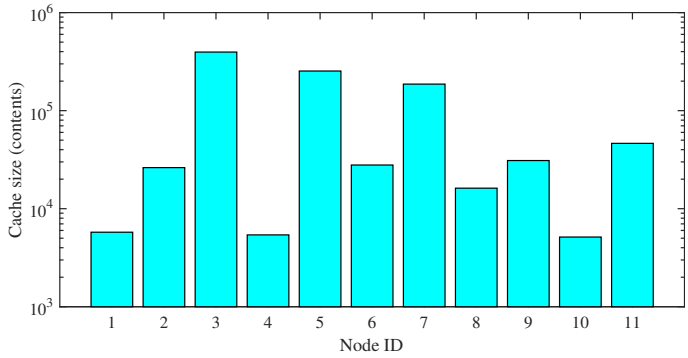
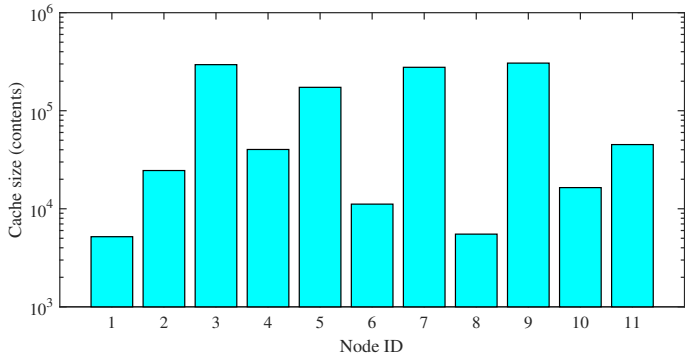


Fig. 5.5 Comparing of service quality among three cache allocation methods under caching budget $C_{tot} = \{1.0e6, 1.2e6, 1.5e6\}$.

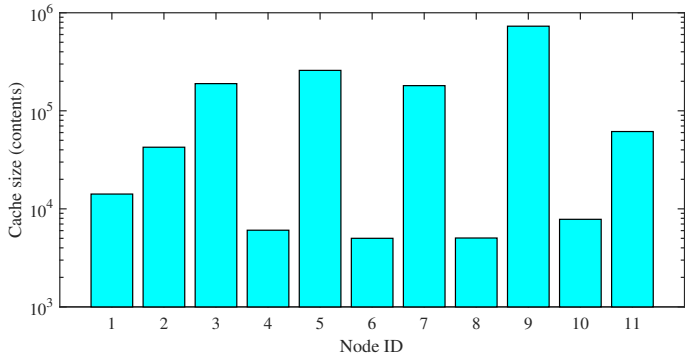
performance is more obvious, which leads to a better QoS under the same cost. Among these nodes, node 9 is allocated a larger cache size due to its lower cost and the location. Both CAPEX and OPEX at node 9 are lower and it locates at the core network. Location is another key metric. Nodes at the core network connect with multiple neighbour nodes. If we improve the performance of these nodes, missing requests forwarded from edge nodes have higher probability to be satisfied within the network instead of going to the remote servers, which reduces the delay and improves QoS. Therefore, although node 3 has high OPEX cost, it is still allocated more caching resource. Furthermore, cost also has an influence on cache placement. For nodes with a higher cost, such as node $4, 8$, even though they are located on the paths of many nodes to the repository, the sizes of their caches are small. The cache sizes of node $1, 2, 6, 10$ are the synthetic impacts of the above factors.



(a) Cache budget $C_{tot} = 1.0e6$



(b) Cache budget $C_{tot} = 1.2e6$



(c) Cache budget $C_{tot} = 1.5e6$

Fig. 5.6 Cache allocation for each node for the median of Pareto optimal set under $C_{tot} = \{1.0e6, 1.2e6, 1.5e6\}$.

To further investigate the impact of content popularity distribution on the proposed caching mechanism, different Zipf exponent α values are applied to the mechanism, varying from 1.1 to 2. Moreover, to decrease the influences of other variables, the content request rates are set to 20 contents/s with caching probability equals 0.5 for all nodes in the network. The cost distribution is using the same configuration as Fig. 5.2. The results in Fig. 5.7 show that the proposed cost-aware caching mechanism can maintain a steady cost around 2.75×10^6 under various

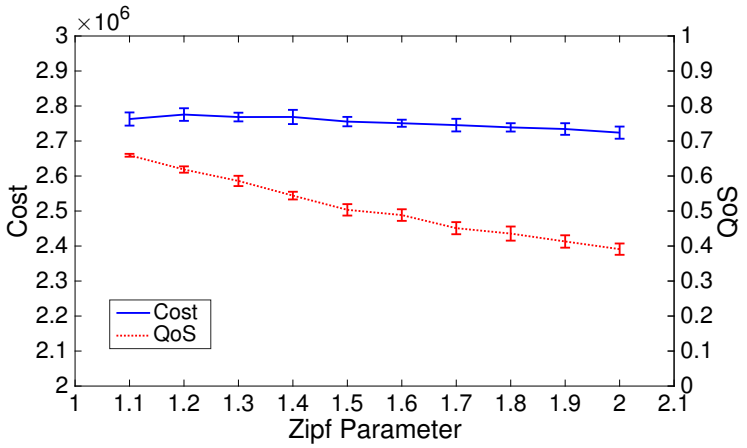


Fig. 5.7 Impact of Zipf parameter α on the cost and QoS with $q_n = 0.5$ and homogeneous content request rate.

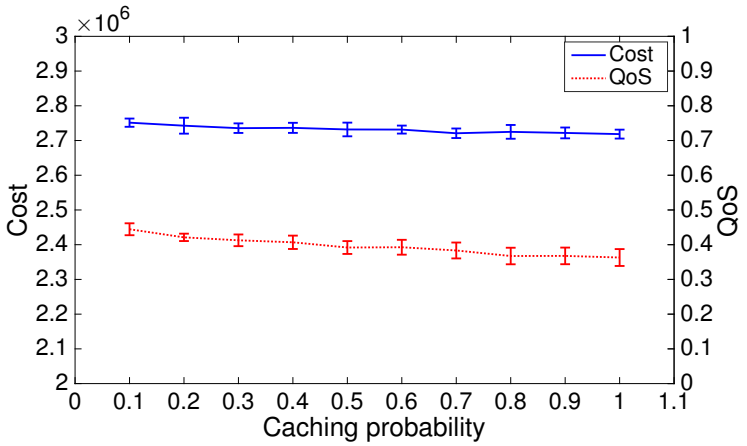


Fig. 5.8 Impact of caching probability q_n on the cost and QoS with homogeneous content request rate and popularity distribution.

content distributions. The QoS has increased as the Zipf parameter becomes larger, which is consistent with the observation from the allocation results in Fig. 5.6.

Finally the reliability of proposed mechanism is investigated under different caching probability of LCP, ranging from 0.1 to 1. When the probability $q_n = 1$, the caching replacement methods becomes LCE. The impact of caching probability q_n on the cost and QoS is analysed. The network is configured in homogeneous with $\alpha = 2$ and requesting rate as 20 contents/s. The results in Fig. 5.8 show that the proposed cost-ware QoS optimisation scheme achieve the steady results under various caching probability. The variation range of the cost is maintained within 1% and the range for the QoS is maintained within 5%, with the QoS slightly improved as the caching probability increasing.

5.5 Summary

In this chapter, a cost-aware optimal caching mechanism has been proposed to find the trade-off between cost of network and service quality in an ICN network. A cost model has been developed to capture the main parameters that contributes to the cost of ICN under an ISP network scenario, and a QoS model is derived to measure the service quality by quantifying the service delay and transmission jitter under arbitrary topology and bursty content requests. This trade-off is translated into a multi-objective optimisation problem, which aims to minimise the network cost and guarantee the QoS. Numerical experiments have been conducted to evaluate the effectiveness of the proposed mechanism. The results have shown that the proposed mechanism can achieve better QoS and lower network cost comparing to the homogeneous and random cache allocation methods. Different factors such as node location, content popularity, and unit cache cost that could impact the cache allocation strategy have been investigated. Some insights have been observed from the optimal cache allocation: content distribution has the most significant impact on the cache allocation; Node location, which determines the traffic intensity and routing path, also greatly affects the allocating decision; Unit price of CAPEX and OPEX has a moderate impact on the cache allocation.

Chapter 6

Conclusions and Future Work

The current Internet architecture was designed based on the host-to-host model, which was aimed at information exchange and communication. However, with the emerging technologies such as Internet-of-Things (IoT), mobile cloud services and the dramatic growth of various multimedia services, there comes a significant change to the main focus of the Internet, shifting from host-centric to service-oriented, and resulting in a mismatch of protocol design and current usage patterns. In fact, the Internet is becoming a content distribution platform, which motivates the origin of Information-Centric Networking (ICN) architecture. In-network caching is considered as an integral part of ICN to efficiently obtain content, alleviate congestion, reduce network load, and enhance users' QoE. The primary focus of this thesis is to investigate and enhance the performance of ICN caching in an arbitrary topology under bursty multimedia traffic. In the following, a summary of the work in this thesis is provided and some directions of future research are indicated.

6.1 Conclusions

This research work has presented new analytical models for performance evaluation of ICN caching in the presence of practical network environment for multimedia services. The accuracy of the proposed models has been validated through comparing the analytical results with those obtained from simulation experiments of an ICN simulator based on OMNET++. The developed analytical models have been used to explore the impact of the key networking and caching metrics on the performance of ICN caching. Moreover, new caching management schemes that leverage the insights

discovered in the models have been designed to optimise the performance of ICN. The major achievements in this thesis are summarised as follows:

(i) A novel analytical model has been developed to evaluate the caching performance of ICN under bursty multimedia content requests. MMPP has been adopted by the model to capture the bursty characteristics of multimedia services. The cache hit ratio as the key performance index of ICN has been derived for different services at both edge nodes and core nodes under a tree network. A thorough investigation into the caching performance under Zipf-like content distribution, LCE caching decision policy and LRU replacement strategy has been conducted. Extensive ICN simulation experiments based on the OMNET++ framework have been performed to validate the effectiveness and accuracy of the analytical model.

(ii) A comprehensive analytical model for analysis of caching performance under arbitrary network topology with heterogeneous busrty content requests and popularity has been proposed. The model has taken into the impacts of different traffic loads and content distribution on the performance and derived the cache hit ratio at any node for various services as the key performance metric. The developed model has been used to further explore the impact of key parameters of ICN, in terms of cache size, topology, content size and content popularity distribution. A new metric, average round trip time (ARTT) has been defined to evaluate the delay performance.

(iii) An optimal cache allocation mechanism that leverages the insights gained from the proposed model has been proposed targeting at minimising the total traffic within the network and improving the network performance. An evolutionary algorithm has been adopted to find the optimal allocation of cache resources under a realistic network topology called Abilene. Three different network scenarios with heterogeneous content request intensity and popularity distribution have been investigated. The performance results have revealed that, for all three scenarios, the optimal caching allocation scheme can achieve

better caching performance and network resource utilisation than the default homogeneous and random caching allocation strategy.

(iv) A cost-aware optimal caching mechanism has been proposed to find the trade-off between cost-efficiency and QoS guarantee in an ICN network. A cost model has been developed to capture the main parameters that contribute to the cost of ICN under an ISP network scenario. A QoS model has been derived to measure the service quality by quantifying the service delay and transmission jitter under arbitrary topology and heterogeneous bursty content requests. Through the two models, the inner association between the cost of network and QoS has been investigated. The trade-off between cost and QoS has been solved as a multi-objective optimisation problem, which aims to minimise the network cost while improving the QoS. Numerical experiments have been conducted to evaluate the effectiveness of the proposed mechanism. The results have shown that the proposed mechanism can achieve better QoS and lower network cost comparing to the homogeneous and random cache allocation methods. Different factors such as node location, content popularity, and unit cache cost that could impact the cache allocation strategy have also been investigated.

6.2 Future Work

The research of this thesis mainly concentrates on analysis and optimisation of the performance in ICN with heterogeneous multimedia traffic and arbitrary topology. As a panacea for content oriented Internet service, ICN provides potential solutions to various problems in the current host based communications. The main research outcomes achieved in the research can be extended to accommodate several interesting yet challenging research directions and can benefit other emerging networking technologies.

Practical ICN Architecture

While the choice of caching policies is already fairly representative in this research, it is not exhaustive. Therefore, a more practical ICN architecture with new policies can be considered. As far as the decision policies are concerned, the deterministic (e.g., based on hashing of the content as done in [118] for P2P VoD) or probabilistic distance-based policies (e.g., as done in CONIC [43]) is worth investigating. Another potential direction will be the heterogeneous cache size vs. cache placement trade-off: on the one hand, larger caches should be placed at nodes with higher betweenness centrality (i.e., nodes locate on the intersection of many shortest paths); on the other hand, this may not always be feasible due to line-speed operation and technological constraints (e.g., large memories are generally slow and thus may be suitable only at the edge of the network). Moreover, it may be worth jointly investigating routing strategies and cache replacement (e.g., similarly to [119] where historical content routing information is considered for content replacement). Finally, energy-efficient caching is also a promising topic [120], which is attracting more and more interest in ICN research. The impact of incorporating energy metric into the performance measurement and design of ICN is worth further study.

Learning-based Caching

One noticeable trend in wireless networks is to deploy cache-enabled small base stations (SBSs) to offload traffic from the macro base station (BS). Caching content at small BSs in order to increase the QoE of the users and alleviate congestion in the backhaul connection has received great attention [121–123]. However, a potential shortcoming of this heterogeneous wireless infrastructure is that, due to high-dense and cost limitations, small BSs are connected to the core network through low-capacity and unreliable backhaul links. Therefore, this small cell solutions alone will not suffice to efficiently solve the QoS requirements associated with peak traffic demands.

One promising solution is to improve the accessibility of service contents by storing the most popular contents in the local caches of SBSs. However, deciding which content should be cached in the limited storage space available at the SBSs is

a NP-hard problem. In [124], a mobility-aware heuristic solution has been proposed to address the NP-hard optimisation problem of maximising the caching hit ratio of mobile users. Furthermore, the popularity distribution is assumed to be known perfectly in current works. In practice, such assumption is not justified, thus learning-based approaches have been proposed to estimate the popularity distribution in [121, 125, 126]. New challenges are raised by the estimation processes, due to the computational intensity and real-time requirement. Therefore, intelligent caching placement strategy leveraging learning-based approaches for improving the QoS is a noteworthy direction.

ICN for 5G Network and Clouds

The future 5G network is designed to cater the surging bandwidth demand of skyrocketing mobile Internet traffic. ICN as a novel network architecture is receiving a lot of attention in the 5G networks [127–129]. The integration of ICN in the 5G network includes content caching and support for mobility.

Cloud Radio Access Network (C-RAN) is one of the evolving architectures for 5G networks [130, 131]. It aims to simplify a radio access network by centralising signal processing in cloud servers and data centres, while maintaining similar coverage to reduce both the capital and operating expenditures. This novel architecture aggregates all BS computational resources into a central pool, called virtual baseband unite (vBBU) pool, and decouples the antennas into geographically distributed remote radio heads (RRHs). By adopting the centre processing of C-RAN architecture, operators will be able to meet the diverse demands for high resource utilisation, scalability, flexibility and spectral efficiency.

The in-network caching of ICN brings opportunities of cooperative and aggregation communication in the network. Introducing ICN communication model into high-density deployed C-RAN can achieve efficient information distribution. The ubiquitous cache-enabled devices can reduce the redundant access and duplicated transmission. Yang et al. [128] proposed a new network architecture that combines ICN, SDN and C-RAN for high-density wireless heterogeneous network. Two types of deployment of the architecture, offload traffic and cache traffic, are investigated to

benefit from the in-network caching of ICN for efficient content delivery and multicasting. However, C-RAN solely focuses on the radio air interface regardless of the transmission in the core network. Therefore, there is a challenge remaining on how to seamlessly integrate C-RAN and ICN. This requires new performance analytical tools and management schemes for resource optimisation, efficient multicast and native mobility in cache-enabled 5G network.

Bibliography

[1] Athanasios V. Vasilakos, Zhe Li, Gwendal Simon, and Wei You. Information centric network: Research challenges and opportunities. *Journal of Network and Computer Applications*, 52:1–10, 2015.

[2] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36, jul 2012.

[3] Andriana Ioannou and Stefan Weber. A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking. *IEEE Communications Surveys & Tutorials*, 18(4):2847–2886, 2016.

[4] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, and Diego Perino. Modeling data transfer in content-centric networking. *2011 23rd International Teletraffic Congress (ITC)*, pages 111–118, 2011.

[5] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. Technical report, 2016.

[6] Xiaolong Jin and Geyong Min. Performance analysis of priority scheduling mechanisms under heterogeneous network traffic. *Journal of Computer and System Sciences*, 73(8):1207–1220, 2007.

[7] Somaya Arianfar, Pekka Nikander, and Jörg Ott. Packet-level caching for information-centric networking. In *Proc. ACM SIGCOMM, ReArch Workshop*, 2010.

[8] Wei Koong Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache "Less for More" in Information-Centric Networks. In *NETWORKING*, volume 7289, pages 27–40. Springer, 2012.

[9] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, and Diego Perino. Modeling data transfer in content-centric networking. In *Proc. International Teletraffic Congress*, pages 111–118, 2011.

[10] Ioannis Psaras, Richard G Clegg, Raul Landa, Wei Koong Chai, and George Pavlou. Modelling and evaluation of CCN-caching trees. In *NETWORKING 2011*, pages 78–91. Springer, 2011.

[11] Guoqiang Zhang, Yang Li, and Tao Lin. Caching in information centric networking: A survey. *Computer Networks*, 57(16):3128–3141, 2013.

[12] Patrick Agyapong and Marvin Sirbu. Economic incentives in information-centric networking: implications for protocol design and public policy. *IEEE Communications Magazine*, 50(12):18–26, dec 2012.

[13] Md. Bari, Shihabur Chowdhury, Reaz Ahmed, Raouf Boutaba, and Bertrand Mathieu. A survey of naming and routing in information-centric networks. *IEEE Communications Magazine*, 50(12):44–53, dec 2012.

[14] Ali Ghodsi, Teemu Koponen, Jarno Rajahalme, Pasi Sarolahti, and Scott Shenker. Naming in content-oriented architectures. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*, page 1, New York, New York, USA, 2011. ACM Press.

[15] Vince Fuller, Tony Li, Jessica Yu, and Kannan Varadhan. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. *Request for Comments*, pages 1–24, 1993.

[16] Masinter Larry, Tim Berners-Lee and Roy T. Fielding. Uniform Resource Identifier (URI): Generic Syntax Status. *Request for Comments: 3986*, 2005.

[17] 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 - CoNEXT '09*, page 1, New York, New York, USA, 2009. ACM Press.

[18] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, oct 2001.

[19] Mihaela Ion, Jianqing Zhang, and Eve M. Schooler. Toward content-centric privacy in ICN. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking - ICN '13*, page 39, New York, New York, USA, 2013. ACM Press.

[20] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D Thornton, Diana K Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, Christos Papadopoulos, and Others. Named data networking (ndn) project. Technical report, 2010.

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

[22] Bengt Ahlgren, Matteo D'Ambrosio, Marco Marchisio, Ian Marsh, Christian Dannewitz, Börje Ohlman, Kostas Pentikousis, Ove Strandberg, René Rembarz, and Vinicio Vercellone. Design considerations for a network of information. In *Proceedings of the 2008 ACM CoNEXT Conference*, page 66. ACM, 2008.

[23] Bengt Ahlgren, Pedro A Aranda, Prosper Chemouil, Sara Oueslati, Luis M Correia, Holger Karl, Michael Sollner, and Annikki Welin. Content, connectivity,

and cloud: ingredients for the network of the future. *Communications Magazine, IEEE*, 49(7):62–70, 2011.

[24] Petri Jokela, András Zahemszky, Christian Esteve Rothenberg, Somaya Arianfar, and Pekka Nikander. LIPSIN: line speed publish/subscribe inter-networking. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 195–206. ACM, 2009.

[25] Nikos Fotiou, Dirk Trossen, and George C Polyzos. Illustrating a publish-subscribe internet architecture. *Telecommunication Systems*, 51(4):233–245, 2012.

[26] Wei Chai, Ning Wang, Ioannis Psaras, George Pavlou, Chaojiong Wang, Gerardo Garcia de Blas, Francisco Ramon-Salguero, Lei Liang, Spiros Spirou, Andrzej Beben, and Eleftheria Hadjioannou. Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services. *IEEE Communications Magazine*, 49(3):112–120, mar 2011.

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

[28] Yanhua Li, Haiyong Xie, Yonggang Wen, and Zhi-Li Zhang. Coordinating In-Network Caching in Content-Centric Networks: Model and Analysis. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 62–72. IEEE, jul 2013.

[29] Yanhua Li, Haiyong Xie, Yonggang Wen, Chi-Yin Chow, and Zhi-Li Zhang. How Much to Coordinate? Optimizing In-Network Caching in Content-Centric Networks. *IEEE Transactions on Network and Service Management*, 12(3):420–434, sep 2015.

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

[31] 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 - ICN '12*, ICN '12, page 55, New York, New York, USA, 2012. ACM Press.

[32] Giuseppe Rossini and Dario Rossi. A dive into the caching performance of Content Centric Networking. In *2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 105–109. IEEE, sep 2012.

[33] Wei Wang, Yi Sun, Yang Guo, Dali Kaafar, Jiong Jin, Jun Li, and Zhongcheng Li. CRCache: Exploiting the correlation between content popularity and network topology information for ICN caching. *2014 IEEE International Conference on Communications, ICC 2014*, pages 3191–3196, 2014.

[34] Ammar Gharaibeh, Abdallah Khreishah, Issa Khalil, and Jie Wu. Distributed Online En-Route Caching. *IEEE Transactions on Parallel and Distributed Systems*, 27(12):3455–3468, dec 2016.

[35] Daphne Tuncer, Marinos Charalambides, Raul Landa, and George Pavlou. More control over network resources: An ISP caching perspective. *2013 9th International Conference on Network and Service Management, CNSM 2013 and its three collocated Workshops - ICQT 2013, SVM 2013 and SETM 2013*, pages 26–33, 2013.

[36] Kideok Cho, Munyoung Lee, Kunwoo Park, Ted Taekyoung Kwon, Yanghee Choi, and Sangheon Pack. WAVE: Popularity-based and collaborative in-network caching for content-oriented networks. *Proceedings - IEEE INFOCOM*, pages 316–321, 2012.

[37] F. Kocak, George Kesidis, T.-M. Pham, and S. Fdida. The Effect of Caching on a Model of Content and Access Provider Revenues in Information-centric Networks. In *2013 International Conference on Social Computing*, pages 45–50. IEEE, sep 2013.

[38] Yusung Kim and Ikjun Yeom. Performance analysis of in-network caching for content-centric networking. *Computer Networks*, 57(13):2465–2482, 2013.

[39] Meng Zhang, Hongbin Luo, and Hongke Zhang. A Survey of Caching Mechanisms in Information-Centric Networking. *IEEE Communications Surveys & Tutorials*, 17(3):1473–1499, 2015.

[40] Sumanta Saha, Andrey Lukyanenko, and Antti Yla-Jaaski. Cooperative caching through routing control in information-centric networks. *Proceedings - IEEE INFOCOM*, pages 100–104, 2013.

[41] Pavlos Sermpezis and Thrasyvoulos Spyropoulos. Effects of Content Popularity on the Performance of Content-Centric Opportunistic Networking : An Analytical Approach and Applications. pages 1–12, 2014.

[42] Valentino Pacifici and Gyorgy Dan. Coordinated Selfish Distributed Caching for Peering Content-Centric Networks. *IEEE/ACM Transactions on Networking*, pages 1–12, 2016.

[43] Yuncheng Zhu, Maoke Chen, and Akihiro Nakao. Conic: Content-oriented network with indexed caching. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–6. IEEE, 2010.

[44] Dario Rossi and Giuseppe Rossini. On sizing CCN content stores by exploiting topological information. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 280–285. IEEE, mar 2012.

[45] V Sourlas, L Gkatzikis, P Flegkas, and L Tassiulas. Distributed Cache Management in Information-Centric Networks. *IEEE Transactions on Network and Service Management*, 10(3):286–299, sep 2013.

[46] Michele Mangili, Fabio Martignon, and Antonio Capone. A comparative study of content-centric and content-distribution networks: Performance and bounds. *GLOBECOM - IEEE Global Telecommunications Conference*, pages 1403–1409, 2013.

[47] Vasilis Sourlas, Lazaros Gkatzikis, Paris Flegkas, and Leandros Tassiulas. Distributed Cache Management in Information-Centric Networks. *IEEE Transactions on Network and Service Management*, 10(3):286–299, sep 2013.

[48] James Roberts and Nada Sbihi. Exploring the memory-bandwidth tradeoff in an information-centric network. In *Proceedings of the 2013 25th International Teletraffic Congress (ITC)*, pages 1–9. IEEE, sep 2013.

[49] Yuemei Xu, Yang Li, Tao Lin, Zihou Ang, Enjia Niu, Hui Tang, and Song Ci. A novel cache size optimization scheme based on manifold learning in Content Centric Netorking. *Journal of Network and Computer Applications*, 37(1):273–281, 2014.

[50] Zhongxing Ming, Mingwei Xu, and Dan Wang. Age-based cooperative caching in Information-Centric Networks. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 268–273. IEEE, mar 2012.

[51] Zhe Li and Gwendal Simon. Time-Shifted TV in Content Centric Networks: The Case for Cooperative In-Network Caching. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, jun 2011.

[52] Jason Min Wang, Jun Zhang, and Brahim Bensaou. Intra-AS cooperative caching for content-centric networks. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking - ICN '13*, page 61, New York, New York, USA, 2013. ACM Press.

[53] Gyorgy Dan. Cache-to-Cache: Could ISPs Cooperate to Decrease Peer-to-Peer Content Distribution Costs? *IEEE Transactions on Parallel and Distributed Systems*, 22(9):1469–1482, sep 2011.

[54] Giuseppe Rossini and Dario Rossi. Coupling caching and forwarding. In *Proceedings of the 1st international conference on Information-centric networking - INC '14*, pages 127–136, New York, New York, USA, 2014. ACM Press.

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

[56] Mengjun Xie, Indra Widjaja, and Haining Wang. Enhancing cache robustness for content-centric networking. In *2012 Proceedings IEEE INFOCOM*, pages 2426–2434. IEEE, mar 2012.

[57] Lili Qiu, Venkata N. Padmanabhan, and Geoffrey M. Voelker;. On the placement of Web server replicas. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, volume 3, pages 1587–1596. IEEE.

[58] Sem Borst, Varun Gupta, and Anwar Walid. Distributed Caching Algorithms for Content Distribution Networks. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, mar 2010.

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

[60] N. Laoutaris, S. Syntila, and I. Stavrakakis. Meta algorithms for hierarchical Web caches. In *IEEE International Conference on Performance, Computing, and Communications, 2004*, pages 445–452. IEEE.

[61] Suyong Eum, Kiyohide Nakauchi, Masayuki Murata, Yozo Shoji, and Nozomu Nishinaga. CATT: potential based routing with content caching for ICN. In *Proceedings of the second edition of the ICN workshop on Information-centric networking - ICN '12*, page 49, New York, New York, USA, 2012. ACM Press.

[62] Andriana Ioannou and Stefan Weber. Towards on-path caching alternatives in Information-Centric Networks. In *39th Annual IEEE Conference on Local Computer Networks*, pages 362–365. IEEE, sep 2014.

[63] Stefan Podlipnig and Laszlo Böszörmenyi. A survey of web cache replacement strategies. *ACM Computing Surveys*, 35(4):374–398, 2003.

[64] Predrag R Jelenković and Xiaozhu Kang. Characterizing the miss sequence of the lru cache. *ACM SIGMETRICS Performance Evaluation Review*, 36(2):119–121, 2008.

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

[66] Fadi M. Al-Turjman, Ashraf E. Al-Fagih, and Hossam S. Hassanein. A value-based cache replacement approach for Information-Centric Networks. In *38th Annual IEEE Conference on Local Computer Networks - Workshops*, pages 874–881. IEEE, oct 2013.

[67] Jason Min Wang and Brahim Bensaou. Progressive caching in CCN. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 2727–2732. IEEE, dec 2012.

[68] The size of the World Wide Web (The Internet), http://www.worldwidewebsize.com/, 2017.

[69] Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox. Information-centric networking: seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, page 1. ACM, 2011.

[70] Matteo D'Ambrosio, Christian Dannewitz, Holger Karl, and Vinicio Vercellone. MDHT: a hierarchical name resolution service for information-centric networks. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*, page 7, New York, New York, USA, 2011. ACM Press.

[71] Jia Zhou, Yanhua Li, Vijay Kumar Adhikari, and Zhi-Li Zhang. Counting youtube videos via random prefix sampling. In *Proceedings of the 2011 ACM*

SIGCOMM conference on Internet measurement conference, pages 371–380. ACM, 2011.

[72] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc. ACM SIGCOMM on Internet measurement*, pages 1–14, 2007.

[73] Michele Tortelli, Dario Rossi, Gennaro Boggia, and Luigi Alfredo Grieco. ICN software tools: Survey and cross-comparison. *Simulation Modelling Practice and Theory*, 63:23–46, apr 2016.

[74] Sunghwan Ihm and Vivek S. Pai. Towards understanding modern web traffic. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference - IMC '11*, page 295, New York, New York, USA, 2011. ACM Press.

[75] Wei Koong Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache "less for more" in information-centric networks (extended version). *Computer Communications*, 36(7):758–770, 2013.

[76] Giovanna Carofiglio, Vinicius Gehlen, and Diego Perino. Experimental Evaluation of Memory Management in Content-Centric Networking. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, jun 2011.

[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.

[78] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, oct 1999.

[79] B´ela Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290, may 2001.

[80] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of Growing Networks with Preferential Linking. *Physical Review Letters*, 85(21):4633–4636, nov 2000.

[81] Elisha J Rosensweig, Jim Kurose, and Don Towsley. Approximate models for general cache networks. In *Proc. IEEE INFOCOM*, pages 1–9, 2010.

[82] Vasilis Sourlas, Georgios S. Paschos, Paris Flegkas, and Leandros Tassiulas. Caching in Content-Based Publish/Subscribe Systems. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pages 1–6. IEEE, nov 2009.

[83] Jason Min Wang and Brahim Bensaou. Progressive caching in CCN. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 2727–2732. IEEE, dec 2012.

[84] Ali Dabirmoghaddam, Maziar Mirzazad Barijough, and J.J. Garcia-Luna-Aceves. Understanding optimal caching and opportunistic caching at "the edge" of information-centric networks. In *Proceedings of the 1st international conference on Information-centric networking - ICN '14*, pages 47–56, New York, New York, USA, 2014. ACM Press.

[85] Yonggong Wang, Zhenyu Li, Gareth Tyson, Steve Uhlig, and Gaogang Xie. Design and Evaluation of the Optimal Cache Allocation for Content-Centric Networking. *IEEE Transactions on Computers*, 65(1):95–107, 2016.

[86] Ioannis Psaras, Wei Koong Chai, and George Pavlou. In-network cache management and resource allocation for information-centric networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):2920–2931, 2014.

[87] Yonggong Wang, Zhenyu Li, Gareth Tyson, Steve Uhlig, and Gaogang Xie. Optimal cache allocation for Content-Centric Networking. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, oct 2013.

[88] Mohammad Hajimirsadeghi, Narayan B. Mandayam, and Alex Reznik. Joint Caching and Pricing Strategies for Information Centric Networks. In *2015*

IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE, dec 2014.

[89] Byung-Gon Chun, Kamalika Chaudhuri, Hoeteck Wee, Marco Barreno, Christos H. Papadimitriou, and John Kubiatowicz. Selfish caching in distributed systems: a game-theoretic analysis. *Proc. of ACM symposium on Principles of Distributed Computing (PODC)*, pages 21–30, 2004.

[90] Valentino Pacifici and Gyorgy Dan. Convergence in player-specific graphical resource allocation games. *IEEE Journal on Selected Areas in Communications*, 30(11):2190–2199, 2012.

[91] Andrea Araldo, Dario Rossi, and Fabio Martignon. Cost-Aware Caching: Caching More (Costly Items) for Less (ISPs Operational Expenditures). *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1316–1330, 2016.

[92] Andrea Araldo, Michele Mangili, Fabio Martignon, and Dario Rossi. Cost-aware caching: Optimizing cache provisioning and object placement in ICN. *2014 IEEE Global Communications Conference, GLOBECOM 2014*, pages 1108–1113, 2014.

[93] Andrea Araldo, Dario Rossi, and Fabio Martignon. Design and evaluation of cost-aware information centric routers. *Proceedings of the 1st international conference on Information-centric networking - INC '14*, pages 147–156, 2014.

[94] Jing Ren, Kejie Lu, Sheng Wang, Xiong Wang, Shizhong Xu, Lemin Li, and Shucheng Liu. VICN: A versatile deployment framework for information-centric networks. *IEEE Network*, 28(3):26–34, 2014.

[95] Jia Ru, Chen Zhe, Luo Hongbin, and Zhang Hongke. Status-aware resource adaptation in information-centric and software-defined network. *China Communications*, 10(12):66–76, 2013.

[96] Jie Dai, Zhan Hu, Bo Li, Jiangchuan Liu, and Baochun Li. Collaborative hierarchical caching with dynamic request routing for massive content distribution. In *2012 Proceedings IEEE INFOCOM*, pages 2444–2452. IEEE, mar 2012.

[97] David Applegate, Aaron Archer, Vijay Gopalakrishnan, Seungjoon Lee, and K. K. Ramakrishnan. Optimal content placement for a large-scale VoD system. In *Proceedings of the 6th International COnference on - Co-NEXT '10*, page 1, New York, New York, USA, 2010. ACM Press.

[98] Syed Hasan, Sergey Gorinsky, Constantine Dovrolis, and Ramesh K. Sitaraman. Trade-offs in optimizing the cache deployments of CDNs. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 460–468. IEEE, apr 2014.

[99] Nikolaos Laoutaris, Orestis Telelis, Vassilios Zissimopoulos, and Ioannis Stavrakakis. Distributed Selfish Replication. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1401–1413, dec 2006.

[100] Moses Charikar and Saikat Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 378–388. IEEE Comput. Soc.

[101] Samee Ullah Khan and Ishfaq Ahmad. Comparison and analysis of ten static heuristics-based Internet data replication techniques. *Journal of Parallel and Distributed Computing*, 68(2):113–136, feb 2008.

[102] Chadi Barakat, Anshuman Kalla, Damien Saucez, and Thierry Turletti. Minimizing bandwidth on peering links with deflection in named data networking. In *2013 Third International Conference on Communications and Information Technology (ICCIT)*, pages 88–92. IEEE, jun 2013.

[103] Haiyong Xie, Y. Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. P4p: provider portal for applications. *ACM SIGCOMM Computer Communication Review*, 38(4):351, oct 2008.

[104] Murtaza Motiwala, Amogh Dhamdhere, Nick Feamster, and Anukool Lakhina. Towards a cost model for network traffic. *ACM SIGCOMM Computer Communication Review*, 42(1):54, jan 2012.

[105] Ignacio Castro, Rade Stanojevic, and Sergey Gorinsky. Using Tuangou to Reduce IP Transit Costs. *IEEE/ACM Transactions on Networking*, 22(5):1415–1428, oct 2014.

[106] Tuan-minh Pham, Serge Fdida, and Panayotis Antoniadis. Pricing in Information-Centric Network Interconnection. In *IFIP Networking Conference*, pages 1–9, 2013.

[107] Nan Zhang, Tapio Levä, and Heikki Hämmäinen. Value networks and two-sided markets of Internet content delivery. *Telecommunications Policy*, 38(5-6):460–472, jun 2014.

[108] Predrag R Jelenković and Ana Radovanović. Least-recently-used caching with dependent requests. *Theoretical computer science*, 326(1):293–327, 2004.

[109] Wolfgang Fischer and Kathleen Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18(2):149–171, 1992.

[110] Haozhe Wang, Geyong Min, Jia Hu, Hao Yin, and Wang Miao. Caching of Content-Centric Networking under Bursty Content Requests. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2522–2527. IEEE, apr 2014.

[111] Asit Dan and Don Towsley. An approximate analysis of the LRU and FIFO buffer replacement schemes. In *Proc. ACM SIGMETRICS*, pages 143–152, 1990.

[112] Hao Che, Z Wang, and Ye Tung. Analysis and design of hierarchical Web caching systems. In *Proc. IEEE INFOCOM*, volume 3, pages 1416–1424, 2001.

[113] Abilene network, https://www.internet2.edu/presentations/fall02/20021028-Abilene-Corbato.pdf., 2017.

[114] Michele Garetto, Emilio Leonardi, and Valentina Martina. A Unified Approach to the Performance Analysis of Caching Systems. *ACM Transactions on*

Modeling and Performance Evaluation of Computing Systems, 1(3):1–28, may 2016.

[115] Haozhe Wang, Geyong Min, Jia Hu, Wang Miao, and Nektarios Georgalas. Performance Evaluation of Information-Centric Networking for Multimedia Services. In *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pages 146–151. IEEE, mar 2016.

[116] Jia Hu, Geyong Min, Weijia Jia, and Mike E. Woodward. Admission Control in the IEEE 802.11e WLANs Based on Analytical Modelling and Game Theory. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pages 1–6. IEEE, nov 2009.

[117] Henning Schulzrinne, S. Casner, R. Frederick, and Van Jacobson. RTP: A Transport Protocol for Real-Time Applications (RFC 1889). *IETF Request for Comments (RFC)*, 1996.

[118] Yaning Liu and G Simon. Peer-Assisted Time-Shifted Streaming Systems: Design and Promises. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, 2011.

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

[120] Chao Fang, F. Richard Yu, Tao Huang, Jiang Liu, and Yunjie Liu. A survey of energy-efficient caching in information-centric networking. *IEEE Communications Magazine*, 52(11):122–129, 2014.

[121] Pol Blasco and Deniz Gunduz. Learning-based optimization of cache content in a small cell base station. In *2014 IEEE International Conference on Communications (ICC)*, pages 1897–1903. IEEE, jun 2014.

[122] B. N. Bharath, K. G. Nagananda, and H. Vincent Poor. A Learning-Based Approach to Caching in Heterogenous Small Cell Networks. *IEEE Transactions on Communications*, 64(4):1674–1686, apr 2016.

[123] Sabrina Muller, Onur Atan, Mihaela van der Schaar, and Anja Klein. Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks. *IEEE Transactions on Wireless Communications*, 16(2):1024–1036, feb 2017.

[124] Yang Guan, Yao Xiao, Hao Feng, Chien Chung Shen, and Leonard J. Cimini. MobiCacher: Mobility-aware content caching in small-cell networks. *2014 IEEE Global Communications Conference, GLOBECOM 2014*, pages 4537–4542, 2014.

[125] Pol Blasco and Deniz Gunduz. Multi-armed bandit optimization of cache content in wireless infostation networks. *IEEE International Symposium on Information Theory - Proceedings*, pages 51–55, 2014.

[126] Ejder Bastug, Mehdi Bennis, and Merouane Debbah. A transfer learning approach for cache-enabled wireless networks. In *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 161–166. IEEE, may 2015.

[127] A Hakiri and P Berthou. Leveraging SDN for The 5G Networks: Trends, Prospects and Challenges. *arXiv preprint arXiv:1506.02876*, pages 1–23, 2015.

[128] Chenchen Yang, Zhiyong Chen, Bin Xia, and Jiangzhou Wang. When ICN meets C-RAN for HetNets: An SDN approach. *IEEE Communications Magazine*, 53(11):118–125, 2015.

[129] 4G Americas. 5G Technology Evolution Recommendations. Technical Report White Paper, 2015.

[130] Aleksandra Checko, Henrik L. Christiansen, Ying Yan, Lara Scolari, Georgios Kardaras, Michael S. Berger, and Lars Dittmann. Cloud RAN for Mobile Networks-A Technology Overview. *IEEE Communications Surveys & Tutorials*, 17(1):405–426, 2015.

[131] Jun Wu, Zhifeng Zhang, Yu Hong, and Yonggang Wen. Cloud radio access network (C-RAN): A primer. *IEEE Network*, 29(1):35–41, 2015.