

Content-Oriented Approaches for Efficient Data Transmission in Future Internet

Saran Tarnoi

A dissertation submitted to the Department of Informatics

School of Multidisciplinary Sciences
in partial fulfillment for the degree of

Doctor of Philosophy

at

The Graduate University for Advanced Studies (SOKENDAI)

© 2015

To my beloved family

Abstract

This dissertation presents several content-oriented approaches for efficient data transmission in the future Internet. These approaches leverage known properties of contents in various aspects to facilitate content delivery. Our proposed approaches are grounded in two main techniques: 1) Content-Centric Networking (CCN) architecture and 2) QoS-aware routing with network coding. They are used to address problems in particular parts of the Internet, which can be categorized by location into the core and edge areas.

An increasing amount of content retrieval traffic raises concern about network congestion in the core area of the modern Internet. An effective solution to network congestion is network caching, whose basic principle is to store some contents in the storage close to content requesters so that some requests can be solved locally. Available implementation of network caching is often unnecessarily complicated. It requires complex middleware for content-to-location translation since contents are not natively identifiable to the data plane of routers. Content-Centric Networking (CCN), which is also called Named Data Networking (NDN), is a new architecture for the future Internet that solves this issue by substituting host addresses in packets with content names and using caches of routers as in-network caches. One of major challenges in CCN is how to efficiently utilize the in-network caches which have limited storage. We address this challenge with two approaches: an Optimal Cooperative Routing Protocol (OCRP) and a probabilistic caching scheme.

The OCRP is an intra-domain routing protocol that utilizes the content retrieval statistics offered by CCN routers to adjust transmission routes. The OCRP consists of three main processes: (1) Prefix Popularity Observation; (2) Prefix Group (Un)Subscription; and (3) Forwarding Information Base (FIB) Reconstruction. In Prefix Popularity Observation, each CCN router observes popularly cited prefixes to activate Prefix Group (Un)Subscription. Prefix Group (Un)Subscription notifies a central routing controller that which CCN router wants to join or leave which prefix group. A prefix group is the group of the CCN routers that frequently forward requests for the same contents. Based on prefix groups, a central routing controller computes an optimal cooperative path by

solving an integer linear programming. Finally, FIB Reconstruction adjusts the routes by updating the routing tables of the CCN routers involved in a newly computed optimal cooperative path. Simulation results show that the OCRP offers better reduction of server load and round-trip hop distance than the shortest path routing. In addition to the routing scheme, we investigate cache management for CCN because it is another important factor that affects the utilization of in-network caches. The cache management schemes of interest to us are combinations of a probabilistic caching scheme and different cache replacement policies. The cache replacement policies include Least Frequently Used (LFU), Least Recently Used (LRU), Random Replacement (RR), and First In First Out (FIFO) policies. Computer simulations are organized to evaluate the performance of the cache management schemes by using several network topologies. Simulation results show that the performance of in-network caches can be improved by using a probabilistic caching scheme along with LRU. Furthermore, we develop a new analytical model to gain deeper understanding of a probabilistic caching scheme. By using this model, several important properties of the probabilistic caching scheme are established as a function of cache replacement policies and network topologies. We have found that a combination of a probabilistic caching scheme and LRU offers the best performance among considering cache management schemes. However, if a CCN router cannot afford LRU due to a complexity constraint, RR is preferred to FIFO since the former yields better performance than the latter.

We proceed to address data transmission problems in the edge area of the Internet. Specifically, we facilitate reliable data transmission in multi-hop wireless networks, which are increasingly often the front-end accesses to the Internet. Contents can be encoded into multiple data layers to support heterogeneous users' demands, devices, and network capacities. For acceptable qualities of contents at end-users, data transmission requires quality-of-service (QoS) such as a data rate and a tolerable packet loss rate. Achieving QoS in multi-hop wireless networks is challenging due to unreliable wireless links and scarce link bandwidth. To address the challenge, we introduce QoS-aware routing schemes for unicast and multicast transmissions, with the help of network coding techniques.

We propose a new QoS-aware routing scheme to enable QoS guarantee for unicast transmission in multi-hop wireless networks. This scheme employs cooperative network

coding (CNC) to improve wireless channel usage and consists of two main steps. First, this scheme uses an integer linear optimization to obtain optimal routes of all unicast flows. The constraints of this optimization problem, such as the transmission rate and tolerable error rate of each data layer, are derived for QoS guarantee. Second, the scheme decides whether or not CNC will be applied to different unicast flows at intermediate nodes. The decision criteria are based on the network topology and QoS requirements. In addition to the unicast transmission, we propose a new QoS-aware routing scheme for reliable multicast transmission in multi-hop wireless networks. This scheme solves an integer linear optimization problem for an optimal route of multicast transmission. When packet loss rates of wireless links are high, a multi-source technique is exploited to enable path diversity which improves the reliability of transmitted data. Furthermore, inter-source network decoding is utilized to improve an achievable data rate at client, where a data layer can be recovered by using network-coded data that are not necessarily from the same source. Simulation results show that both of the proposed schemes yield better reliability of data transmission in multi-hop wireless networks than several QoS-oblivious routing schemes.

Acknowledgements

I would like to express my deepest appreciation to my advisor Prof. Yusheng Ji for her guidance, carefulness, patience, and profound vision regarding the research and my future career. Her supervision has allowed me to learn being an independent researcher. I would also like to thank Prof. Shigeki Yamada, Assoc. Prof. Shunji Abe, Assoc. Prof. Kensuke Fukuda, Assoc. Prof. Michihiro Koibuchi, and Asst. Prof. Celimuge Wu for their valuable comments on this research and suggestions for the development of this work.

My sincere thanks are given to Assoc. Prof. Wuttipong Kumwilaisuk from King Mongkut's University of Technology Thonburi, Assoc. Prof. Poompat Saengudomlert from Bangkok University, and Prof. C.-C. Jay Kuo from University of Southern California for their continuous supports. Their expertise, attitude, and intelligence have contributed to the deepness and thoroughness of this work. Moreover, I would like to thank Dr. Kalika Suksoomboon from KDDI R&D Laboratories Inc., Asst. Prof. Vorapong Suppakitpaisarn from The University of Tokyo, and Dr. Kriangkrai Limthong from Bangkok University for their countless pieces of advice on research methodology and analytical techniques. I would also like to thank students, researchers, and internship students at National Institute of Informatics (NII) for their helpful comments on this work.

My appreciation is also given to NII and the Japan Student Services Organization (JASSO) for their scholarships and research funding. I would also like to thank the secretaries of Klab, the administration staff members of NII, and those of The Graduate University for Advanced Studies (SOKENDAI) for their supports.

Last but not least, my special thanks are given to my family and friends. They gave me spiritual power and encouragement when I struggled with hard problems. I obtained from them many precious things that I hardly find elsewhere.

Contents

Contents	i
List of Figures	iv
List of Tables	vii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	8
1.2.1 Routing and Caching Schemes for Content-Centric Networking . . .	8
1.2.2 QoS-aware Routing with Network Coding in Multi-hop Wireless Networks	9
1.3 Dissertation Organization	10
2 Background	11
2.1 Content-Centric Networking	11
2.1.1 Routing Schemes for Content-Centric Networking	14
2.1.2 Cache Management for Content-Centric Networking	15
2.1.3 Performance Analysis of Probabilistic Caching Systems	16
2.2 Network Coding	18
2.2.1 Unicast Transmission with Cooperative Network Coding	20
2.2.2 Multicast Transmission with Network Coding	23
3 Cooperative Routing for Content-Centric Networking	26
3.1 Introduction	26
3.2 Optimal Cooperative Routing Protocol	27
3.2.1 Prefix Popularity Observation	31
3.2.2 Prefix Group (Un)Subscription	32
3.2.3 FIB Reconstruction	34
3.3 Optimal Cooperative Path Selection	36
3.3.1 Network Model	36
3.3.2 Objective Function	36
3.3.3 Flow Conservation Constraint	38
3.3.4 Cache Contention Mitigating Constraint	38
3.3.5 Path Length Constraint	39
3.3.6 Problem Formulation	39
3.4 Performance Evaluation	41
3.4.1 Simulation Set-up	41
3.4.2 Effects of Caching and Cache Replacement Policies	43
3.4.3 Effects of the Number of Prefix Groups	45

3.4.4	Effects of the Number of Requester Routers in Prefix Group	47
3.4.5	Effects of Content Popularity Distribution and the CS Size of CCN Router	47
3.4.6	Effects of Node Degree	50
3.4.7	Computational Complexity	52
3.5	Conclusion	54
4	Cache Management in Content-Centric Networks	55
4.1	Introduction	55
4.2	Cache Management Schemes	57
4.2.1	Caching Schemes	57
4.2.2	Cache Replacement Policies	59
4.3	Comparative Study of Cache Management Schemes	60
4.3.1	Simulation Set-up	60
4.3.2	Evaluation Metrics	61
4.3.3	Network Topologies	61
4.3.4	Results and Discussions	63
4.4	Conclusion	73
5	Performance Analysis of Probabilistic Caching Scheme	74
5.1	Introduction	74
5.2	System Model and Assumptions	75
5.2.1	Caching Scheme and Cache Replacement Policies	76
5.2.2	Request Traffic Model	77
5.2.3	Ergodicity of Markov Chains	77
5.3	Performance Analysis of Probabilistic Caching Scheme and Random Replacement Policy	78
5.4	Performance Analysis of Probabilistic Caching Scheme and First In First Out Policy	83
5.5	Performance Analysis of Probabilistic Caching Scheme and Least Recently Used Policy	90
5.6	Experimental Results	93
5.6.1	Model Validation and Insights	93
5.6.2	Experiments with Real Internet Topologies	95
5.7	Conclusion	97
6	Content Importance-based Routing Using Cooperative Network Coding	99
6.1	Introduction	99
6.2	System Model and Problem Description	100
6.2.1	Network Model	100
6.2.2	QoS Guarantee	102
6.3	Optimal Path Selection for Layered Unicast	103
6.3.1	Objective Function	103
6.3.2	Flow Conservation Constraint	106
6.3.3	Reliability Constraint	107
6.3.4	Wireless Link Scheduling Constraint	107
6.3.5	Problem Formulation: A Summary	109
6.4	QoS-Aware CNC for Layered Unicasts	111
6.4.1	Three Basic Local Structures	111

6.4.2	Coding Rules and Opportunities	112
6.4.3	Reliability Computation	113
6.4.4	CNCE Protocol	116
6.5	Experimental Results	118
6.5.1	Experimental Set-Up	119
6.5.2	Effects of Link Transmission Probabilities	120
6.5.3	Effects of Node Densities	125
6.5.4	Effects of Traffic Demands	125
6.5.5	Effects of Wireless Interference: A Case of Single Wireless Channel	129
6.6	Conclusion	131
7	Content Importance-based Multicast Using Inter-source Network Coding	133
7.1	Introduction	133
7.2	System Model and Problem Description	136
7.2.1	Scalable Video Coding	136
7.2.2	Network Model	137
7.2.3	QoS Guarantee	138
7.2.4	Optimum Path Selection for Layered Multicast	138
7.2.5	Objective Function	139
7.2.6	Problem Formulation	141
7.3	Reliable Layered Multicast with Multiple Sources and Network Coding	143
7.3.1	Transmission Reliability of Layered Multicast with Multi-Source	143
7.3.2	QoS-aware Multicast with Multiple Sources and Network Coding	144
7.3.3	Inter-source Network Decoding	149
7.4	Experimental Results	151
7.4.1	Comparison of Achievable Data Rate and Reliability	154
7.4.2	Comparison of Objective and Subjective Qualities	157
7.5	Conclusion	159
8	Conclusion	162
8.1	Discussion	162
8.2	Conclusion	165
8.3	Future Work	166
8.3.1	Network Coding in Content-Centric Networks	166
8.3.2	Scalable Video Streaming in Content-Centric Networks	168
8.3.3	Cache Management Schemes for Video Workloads	169
	Bibliography	170
	A Publication List	179

List of Figures

1.1	The network core and wireless access networks in the Internet.	3
2.1	Packets types of CCN.	12
2.2	Components of CCN router.	12
2.3	Data transmissions in a butterfly network.	18
2.4	Data transmissions in a wireless network.	20
3.1	Routing path selection for multiple flows of Interest packets based on cooperative routing for CCN.	28
3.2	The overview of OCRP.	29
3.3	Process of handling Interest packets and tracking of popularity of each FIB entry.	33
3.4	Format of each signaling packet.	34
3.5	FIB reconstruction process of involved CCN router.	35
3.6	Impact of the caching and cache replacement policies.	44
3.7	Impact of the number of prefix groups.	46
3.8	Impact of the number of requester routers in each prefix group.	48
3.9	Impact of content popularity distribution.	49
3.10	Impact of node degree.	51
3.11	Computational time for the optimal cooperative path.	53
3.12	Example of multiple areas in an OCRP network.	54
4.1	Cascading network used in our simulations.	62
4.2	The network topology of SINET4.	63
4.3	The server load for different CS sizes in the cascading network.	64
4.4	The round-trip hop distance for different CS sizes in the cascading network.	66
4.5	Hit rate with respect to the node level in the cascading network.	67
4.6	Instantaneous behavior of different caching schemes and replacement policies in the cascading network.	69
4.7	The server load for different CS sizes in SINET4.	70
4.8	The round-trip hop distance for different CS sizes in SINET4.	71
4.9	Hit rate towards the node level in SINET4.	71
4.10	Instantaneous behavior of different caching schemes and replacement policies in SINET4.	72
5.1	State transition probability in the Markov chain for a discrete cache whose cache management is a combination of $Prob(q)$ and RR.	79
5.2	A content-centric network forming a two-level caching hierarchy.	81
5.3	State transition probability in the Markov chain of a discrete cache whose cache management is a combination of $Prob(q)$ and FIFO.	84

5.4	State transition probability in the Markov chain of a discrete cache whose cache management is a combination of $Prob(q)$ and LRU.	90
5.5	Hit rate versus item population for the discrete cache controlled by RR. . .	93
5.6	Hit rate versus item population for the discrete cache controlled by FIFO. . .	94
5.7	Hit rate versus item population of the discrete cache controlled by LRU. . .	94
5.8	Miss rate of a cache network in Figure 5.2 versus a caching probability. . .	95
5.9	Server load versus the cache size in the SINET4 topology.	96
5.10	Server load versus the cache size in the GÉANT topology.	96
5.11	Server load versus the cache size in the Internet2 topology.	97
6.1	An example of defining the information value of each sublayer based on its original layer.	104
6.2	An exemplary wireless network.	109
6.3	Coding rules with XOR (\oplus) operation.	112
6.4	An extended A-B structure which has more than one intermediate node. . .	115
6.5	Three basic local structures for the CNCE protocol: (a) the A-B structure, (b) the Y structure and (c) the X structure. The dashed and regular arrows represent the overhearing and direct transmissions, respectively.	116
6.6	Comparison of the throughput for various link qualities in the networks having 15 nodes.	121
6.7	Comparison of the number of channel uses for various link qualities in the networks having 15 nodes.	121
6.8	Comparison of the throughput per channel use for various link qualities in the networks having 15 nodes.	122
6.9	Comparison of the throughput for various link qualities in the networks having 20 nodes.	122
6.10	Comparison of the number of channel uses for various link qualities in the networks having 20 nodes.	123
6.11	Comparison of the throughput per channel use for various link qualities in the networks having 20 nodes.	123
6.12	Comparison of the throughput for different routing schemes as a function of node density.	126
6.13	Comparison of the number of channel uses for different routing schemes as a function of node density.	126
6.14	Comparison of the throughput per channel use for different routing schemes as a function of node density.	127
6.15	Comparison of the throughput for different routing schemes as a function of the number of s-d pairs in each network.	127
6.16	Comparison of the number of channel uses for different routing schemes as a function of the number of s-d pairs in each network.	128
6.17	Comparison of the throughput per channel use for different routing schemes as a function of the number of s-d pairs in each network.	128
6.18	Comparison of the throughput for various link qualities in single-channel networks having 15 nodes.	130
6.19	Comparison of the number of channel uses for various link qualities in single-channel networks having 15 nodes.	130
6.20	Comparison of the throughput per channel use for various link qualities in single-channel networks having 15 nodes.	131
7.1	Layered multicast with multiple sources and inter-source network decoding. . .	134

7.2	Different bitstreams enabling different spatial, temporal, and SNR scalabilities.	137
7.3	A flowchart demonstrating the QoS-aware multicast with multiple sources and network coding.	145
7.4	A network with two source nodes.	146
7.5	A set of paths for each destination obtained from the primary source node S2: (a) a set of paths for destination node 3, (b) a set of paths for destination node 4, and (c) a set of paths for destination node 5.	147
7.6	The network when a set of links has been removed in Step 3	148
7.7	A set of paths for each destination obtained from the secondary source node S1: (a) a set of paths for destination node 3 and (b) a set of paths for destination node 5.	148
7.8	A set of paths from multiple sources.	149
7.9	An example network using inter-source network coding.	150
7.10	Comparison of achievable data rate of all layers for various link conditions.	153
7.11	Comparison of reliability of data transmission of layer 0 for various link conditions.	154
7.12	Comparison of reliability of data transmission of layer 1 for various link conditions.	155
7.13	Comparison of reliability of data transmission of layer 2 for various link conditions.	155
7.14	Comparison of the number of links used for each transmission scheme.	156
7.15	Comparison of the average PSNR of video sequence “Paris” for various link conditions.	157
7.16	Comparison of the average PSNR of video sequence “Bus” for various link conditions.	158
7.17	Comparison of the average PSNR of video sequence “Football” for various link conditions.	158
7.18	Comparison of the average PSNR of video sequence “Mobile” for various link conditions.	159
7.19	Visual example selected from video sequence “Paris” from various routing schemes : (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.	159
7.20	Visual example selected from video sequence “Bus” from various routing schemes: (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.	160
7.21	Visual example selected from video sequence “Football” from various routing schemes: (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.	160
7.22	Visual example selected from video sequence “Mobile” from various routing schemes: (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.	160
8.1	Example of network coding in a content-centric network.	167
8.2	RTT variability of scalable video transmissions in a content-centric network.	168

List of Tables

5.1	Summary of notations in our analytical model for cache analysis	78
5.2	Topological properties of the considered networks	95
6.1	Summary of notations for QoS-aware routing in multi-hop wireless networks	101
6.2	Distance thresholds for different transmission data rates.	119
7.1	Summary of notations for the optimal path selection for layered multicast .	139
7.2	Summary of notations for QoS-aware multicast with multiple sources and network coding	144
7.3	Comparison of the reliability obtained by single source and multi-source multicast.	149
7.4	Distance thresholds for different transmission data rates	153

Chapter 1

Introduction

This chapter provides the research motivation which is based on problems associated with data transmission in the Internet. The contributions of this dissertation are summarized and the dissertation organization is given at the end of chapter.

1.1 Motivation

Overlooking properties of transmitted data could lead to inefficient data transmission in the Internet. The cause of this issue is rooted in a conflict between a host-to-host communication model of the Internet and what it is used for. The principles of data transmission in the current Internet are based on an architecture which mainly relies on communication between two distant nodes. Data transmission aims to send a bunch of data from one place to another place in a fast, reliable, and secure manner. To achieve the goal, many techniques were implemented on top of the host-to-host communication model. While being effective nowadays, the add-on techniques inevitably impose difficulties of implementation and maintenance, leading to excessive overheads and costs. Properties of data, including the relation between them, are not often taken into account when they are transmitted. The forwarding engine of a router, which is key equipment used in data transmission, merely treats the carriers of data as anonymous packets. It mostly pays attention to only the destination addresses of packets, even though exception holds for some mechanisms that can roughly prioritize packet forwarding based on the data types.

In fact, the data being delivered in the Internet are partly meaningful by themselves. They are often a part of content having some specific properties. In some cases, these properties are highly useful for data transmission.

For instance, data can belong to chunks of a video. If the video is encoded by using a hierarchically structured coding, different pieces of the data may differently contribute to video quality at a client. When network resources are so limited that all data cannot be transmitted to destination equally well, their transmissions should be provided with different priorities based on their importance. In addition, when the video is public and accessible from a group of clients, it may become popular and is accordingly watched by many clients in the group. When a host-to-host communication model is used, the same data are repeatedly transmitted from a video server to these clients one after another, resulting in a large amount of redundant traffic in networks. It is so because the data plane does not realize that it repeats data transmission for the same content. On the other hand, if packets are identifiable by the content names, the data plane will recognize data and could manipulate them more precisely. Network caching could become an inherent part of data transmission, as the individual data can be cached and discoverable to associated content requests directly on the data plane. This approach simplifies a content delivery process and efficiently eliminates redundant traffic. From the above example, we can see that many new opportunities will be open for more efficient data transmission when the properties of content are taken into account.

In this dissertation, we propose several content-oriented approaches for efficient data transmission in the future Internet, by leveraging some known properties of content in various aspects. Our approaches are grounded in two main techniques: 1) Content-Centric Networking (CCN) architecture and 2) QoS-aware routing with network coding. We use them to solve data transmission problems in particular parts of the Internet, which can be categorized by location into the core and edge areas.

In the first half of the dissertation, we address inefficient data transmission in the core area of the Internet. The majority of traffic in the Internet is from content retrieval [1, 2, 3, 4, 5]. Content retrieval broadly denotes activities pertaining to downloading contents on the Internet. Contents are pieces of information stored in various electronic forms such as web objects, documents, images, audio and video files. An increasing amount

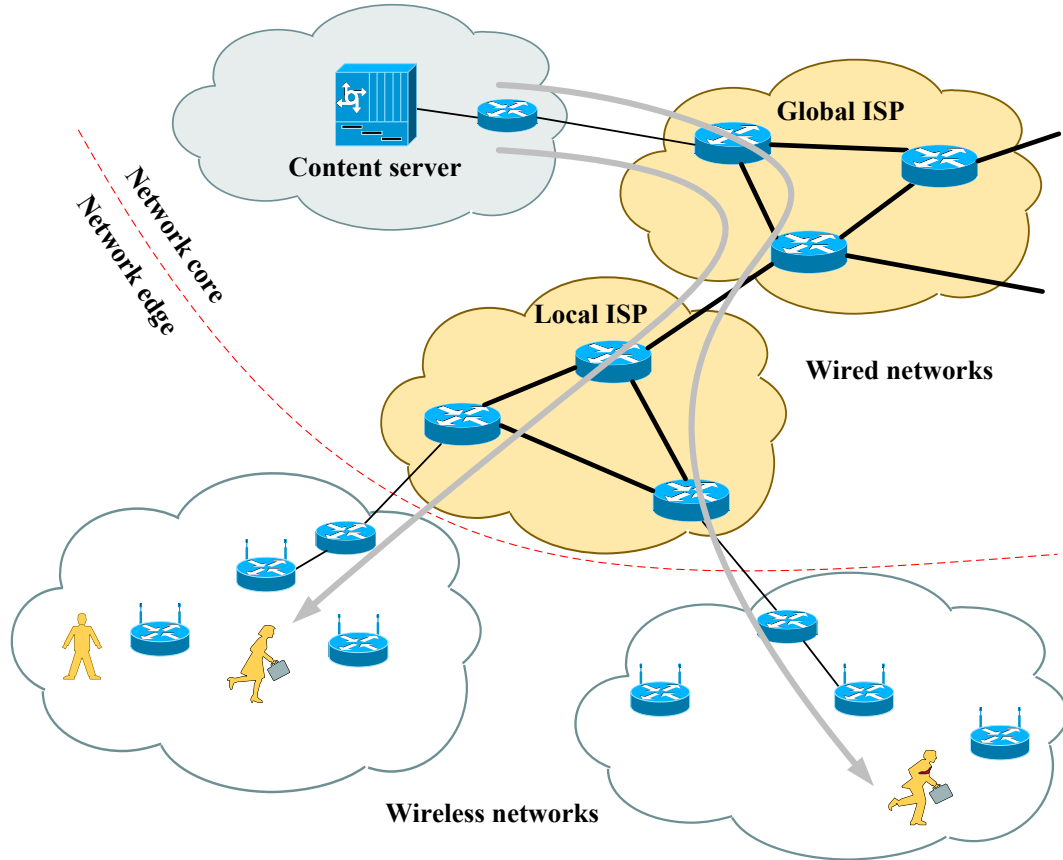


Figure 1.1: The network core and wireless access networks in the Internet.

of content retrieval traffic raises concern about network congestion, especially at fragile interconnection points between multiple networks as shown in Figure 1.1. When content consumers heavily retrieve contents from a content server, the demand for traffic on a link between networks may exceed the link capacity and network congestion would occur. Network congestion is a main cause of undesired effects such as delay, packet drops, and connection blocking. An overloaded content server could also develop the similar effects.

Provisioning the entire links in the Internet with redundant capacities is an ideal solution to network congestion. However, this approach is impractical, considering the gigantic scale and complexity of the Internet. In content retrieval paths, data often travel across multiple networks. While ones do their best to ensure sufficient network capacity, network congestion would still occur if others do not.

A practical solution to this problem is network caching. Network caching reduces the traffic in networks by fulfilling the requests of content locally. This approach makes use of a fact that many end-users often retrieve the same contents from the Internet.

To enable network caching, in-network caches are placed in networks close to end-users. In-network caches are responsible to temporarily storage, keeping the content that has recently traversed the networks. When clients demand a content recently requested by others, they can retrieve this content from in-network caches, instead of distant original content servers. Efficient in-network caches can reduce not only the traffic in networks, but also the content retrieval time and the content server load.

Existing systems of in-network caches include Web caches (also called proxy servers) and Content Delivery Networks (CDNs) [6]. They have been playing an important role in delivering contents on the Internet to end-users [5]. Nevertheless, their usage is typically restricted to a handful of applications. They also require complicated implementation that involves large infrastructure, middleware for content-to-location translation, and engines for intercepting and redirecting content requests. Importantly, their protocols are based on the application layer, thus causing considerable overheads and latencies. These factors unnecessarily complicate the processes of content retrieval [7, 8].

In this work, we consider Content-Centric Networking (CCN) architecture [7], which is a new architecture for the future Internet, to be a new solution to data transmission in the core area area. CCN is a root of a more recent project called Named Data Networking (NDN)[9]. Hence the terms CCN and NDN are used interchangeably. In CCN, packets are distinguished by unique names and the caches of routers are used as in-network caches. CCN data plane uses the names of packets in packet forwarding and network caching. From a content retrieval point of view, CCN is more efficient than existing host-centric communication model, as the network caching is performed directly in the network layer. However, the caches of CCN routers tend to be small in comparison to the amount of contents on the Internet [10]. Efficient utilization of the in-network caches is therefore a key to efficient data transmission in content-centric networks.

We see opportunities to improve the cache utilization of CCN from the routing and caching points of view, through deliberating over some properties of content retrieval.

CCN packets are typically forwarded in the shortest-path basis [11, 12], which may not fully exploit the caching benefits of CCN routers. Different CCN routers may use disjoint paths to forward their packets, even though these packets contain the same content. In this case, each CCN router would benefit from the cached content that are subject to only

its previously transmitted data. Moreover, when many unique contents arrive at a CCN router, the cache of the CCN router must handle these diverse contents. The contents may be so many that the cache cannot manage them efficiently and would result in poor caching performance. To solve the problem, we introduce cooperative routing scheme for CCN. The CCN routers frequently forwarding requests for the same set contents can cooperatively share a common path. The contents cached by the CCN router on this path are not only useful for a given CCN router but all the participants. In addition, the CCN routers may frequently forward requests of content from the same content publisher. Some contents requested by a CCN router have a high probability that they would be requested by the other CCN routers. In practice, this scenario occurs when a group of end-users tries to access the same website, such as YouTube [4]. Once a popular content is viewed by a client, it is often viewed again by other clients who regularly access the considering website.

Cache management also contributes to the cache utilization. CCN suggests caching all contents that traverse a CCN router, which is called a universal caching scheme, so that the cached contents will fulfill some future requests. Nevertheless, the universal caching scheme does not work well in CCN, as it causes many redundant copies of the same content in networks [13, 14, 15, 16, 17]. Several sophisticated caching schemes are proposed to replace the universal caching scheme [13, 18, 19, 20, 21]. Unfortunately, they often add complexity and overheads to caching systems, which are not suitable for high-speed CCN routers. Among available caching schemes, we observe that randomly caching contents at a certain probability, which is called a probabilistic caching scheme, can overcome some main drawbacks of the universal caching scheme. The probabilistic caching scheme was used as a benchmark policy by several works in literature [13, 14, 15]. However, the insights into the behavior of the probabilistic caching scheme have never been studied. It has been roughly evaluated when it is used with a cache replacement policy, i.e., Least Recently Used (LRU). There has never been a criterion that suggests a decent value of a caching probability as well as its practical limitation. Therefore, we investigate the behavior of the probabilistic caching scheme, aiming to justify the feasibility of using it in cache management for CCN.

In the second half of the dissertation, we focus on data transmission in the edge of

the Internet. In a content retrieval path, content is delivered to end-users through the edge area of the Internet, where wireless networks have become common. This scenario is illustrated in Figure 1.1. Among available wireless networks, our interest is centered in multi-hop wireless networks [22], such as wireless mesh networks [23]. Multi-hop wireless networks offer a multi-hop wireless backbone that interconnects isolated local area networks (LANs) and extends backhaul access to the users who are out of the range of typical access points. Nodes in multi-hop wireless networks are usually stationary and can be permanently power-supplied. Consequently, the main factors affecting data transmission are the quality and capacity of wireless links.

Transmitting data in multi-hop wireless networks without considering the properties of contents may cause end-users dissatisfaction and frustration. Contents can be encoded with *layered coding* to serve heterogeneous users' needs, device capabilities, and network capacities. In layered coding, a content is encoded into multiple data layers with different data rates. The quality of content at a client increases with the number of data layers received. Due to the hierarchical structure of layered coding, a lower data layer is more important to decoder than a higher layer data. Decoding of a higher data layer requires that the entire lower data layers must be correctly received [24]. Therefore, quality-of-service (QoS) for layered data transmission is required to ensure acceptable quality of content at a client. QoS can be a combination of the data rate and tolerable packet loss rate of each data layer. However, achieving QoS in multi-hop wireless networks is a daunting task, considering unreliable quality and limited bandwidth of wireless links [25].

Forward Error Correction (FEC) [26, 27] is a coding technique performed by sources to control errors of data transmission over noisy channels. In FEC, a source transmits data encoded with an error-correcting code (ECC) to a destination. The encoded data are larger than the original data as a result of additional redundant bits. The redundant bits allow for a recovery of the original data from received data, whose parts may be lost or corrupted during data transmission. More redundant bits are added when the channel quality becomes worse to guarantee certain data reliability. Therefore, FEC may not be suitable for QoS-sensitive data transmission over unreliable wireless links with limited bandwidth.

On the other hand, QoS-aware routing can be used for providing QoS guarantee to data

transmission. QoS-aware routing, which is also called content importance-based routing from now, considers the importance of contents when it selects transmission routes. This approach rather selects a more reliable route for data transmission that demands stricter QoS, instead of sending data with redundant bits over arbitrary routes. Furthermore, network coding can be employed to improve the utilization of network resources [28]. In this work, we utilize both QoS-aware routing and network coding to achieve QoS guarantee for unicast and multicast transmissions in multi-hop wireless networks.

Efficient unicast transmission in wireless networks can be achieved by leveraging wireless broadcast and cooperative network coding (CNC) [29, 30]. When multiple unicast flows are delivered in a multi-hop wireless networks, CNC can be used to improve the channel utilization and total network throughput. Depending on the network topology, CNC is applied to unicast flows at intermediate nodes in multi-hop wireless networks. However, performing CNC may affect the QoS of data in each unicast flow. This is because the success of a unicast session depends on the other unicast session involved in CNC. Encoding and decoding nodes require that all CNC packets must be delivered successfully. Therefore, achieving high network throughput, data quality guarantee, and efficient channel utilization under unreliable wireless links is challenging.

For reliable multicast transmission, when there is more than one source, data from multiple sources can be used to improve the reliability of transmitted data at terminal nodes. However, if this approach is directly applied to multi-hop wireless networks, it may affect the efficiency of network resource usage. This is because network resource consumption increases with the number of sources participating in the multicast transmission. Network coding can be used to enable inter-source network decoding for more efficient network resource utilization. Inter-source network decoding allows each client to use network-coded packets from multiple sources to recover the original data. The targets of multicast transmission include videos encoded with *scalable video coding* (SVC) [24]. By using layer coding, SVC encodes a video into a base layer and several enhancement layers [31] to enable scalability for heterogeneous users, devices, and networks. Data of a lower layer contribute to the end-quality of a video more than those of a higher layer. In multi-hop wireless networks, available network resources may not meet the QoS requirements of all data layers. Therefore, prioritizing transmission based on the importance of

data layers is essential.

1.2 Contributions

We propose several content-oriented approaches for efficient data transmission in the core and edge areas of the Internet. Specifically, CCN is used to enable network caching, aiming to eliminate redundant traffic caused by content retrieval in the core area. In the edge area, QoS-aware routing is used along with network coding techniques to enable reliable and efficient data transmission in multi-hop wireless networks. Our main contributions are as follows.

1.2.1 Routing and Caching Schemes for Content-Centric Networking

We introduce a new cooperative routing scheme and a probabilistic caching scheme for CCN. Our ultimate goal is to improve the utilization of in-network caches in content-centric networks.

Cooperative Routing Scheme

We propose a cooperative routing scheme for CCN, focusing on a route optimization based on content retrieval statistics. This routing scheme selectively aggregates multiple data flows onto the same path with a goal to improve the utilization of in-network caches while mitigating cache contention of the CCN routers. We propose an Optimal Cooperative Routing Protocol (OCRP) to sketch an implementation of the cooperative routing scheme on the control plane of CCN. Moreover, we provide a new method for content popularity observation at CCN routers, leveraging a fact that CCN names packets with hierarchical prefix names. An integer linear optimization problem is formulated for calculating an optimal cooperative path for OCRP. Computer simulation is organized to evaluate the performance of OCRP with regard to the reductions of server load and round-trip hop distance, when it is used in mesh network topologies.

Probabilistic Caching Scheme

We propose to use a probabilistic caching scheme in cache management for CCN. We evaluate the performance of probabilistic caching systems with various cache replacement policies by means of computer simulation and mathematical analysis. The cache replacement policies consist of Random Replacement (RR), First In First Out (FIFO), Least Recently Used (LRU), and Least Frequently Used (LFU) policies. Computer simulation is conducted by using network topologies derived from several real academic networks and various parameters which include the cache size, content population, and content popularity distribution. We develop a new analytical model to mathematically analyze the probabilistic caching scheme. This model is based on Markov chains under Independent Reference Model (IRM) and Zero Download Delay (ZDD) assumption. By analyzing the results from the computer simulation and the analytical model, we can establish several important properties of a probabilistic caching scheme, drawing up guidelines for lightweight and efficient cache management for CCN.

1.2.2 QoS-aware Routing with Network Coding in Multi-hop Wireless Networks

We propose novel QoS-aware routing schemes for unicast and multicast transmissions in multi-hop wireless networks. Network coding is utilized in these routing schemes to improve the utilization of wireless channels while retaining QoS of data transmission.

Unicast Transmission

We propose a QoS-aware routing scheme for efficient unicast transmission in multi-hop wireless networks with cooperative network coding. We formulate an integer linear optimization problem for calculating an optimal route for all unicast flows based on QoS requirements. We investigate the effects of performing CNC in lossy wireless networks on the reliability of data transmission. A Cooperative Network Coding Establishment (CNCE) algorithm is proposed for selectively applying CNC to particular unicast flows without QoS violation. We conduct computer simulation to evaluate this routing scheme in terms of throughput and channel utilization, when it is used in different network topologies

and various parameters.

Multicast Transmission

We propose a QoS-aware routing scheme for scalable video multicast in multi-hop wireless networks. The routing scheme selects an optimal path for scalable video multicast transmission based on QoS requirements of each data layers and network resources. Random linear network coding is applied at intermediate nodes for efficient channel utilization. In addition, we propose a more robust and efficient multicast transmission by using source diversity and inter-source network decoding technique. The multiple sources and inter-source network decoding are employed when the reliability of data obtained from the optimal path does not meet QoS requirement. Computer simulation is organized to evaluate the routing scheme in terms of the achievable data rate as well as the objective and subjective qualities of scalable videos at clients.

1.3 Dissertation Organization

The rest of the dissertation is organized as follows. Chapter 2 gives some background of architectures and techniques used in this dissertation. The introduction to the Content-Centric Networking (CCN) architecture and network coding techniques is provided along with related work. Chapter 3 presents a cooperative routing scheme and the details of an Optimal Cooperative Routing Protocol (OCRP) for CCN. Chapter 4 presents a comparative study of cache management schemes for content-centric networks. Chapter 5 presents a performance analysis of a probabilistic caching scheme. Chapter 6 presents content importance-based routing in multi-hop wireless networks using cooperative network coding. Chapter 7 presents a framework for robust scalable video multicast transmission in multi-hop wireless networks using inter-source network coding. We provide discussion, conclusion, and future work in Chapter 8.

Chapter 2

Background

This chapter gives some background knowledge about the Content-Centric Networking architecture and network coding. The related work is also provided.

2.1 Content-Centric Networking

Content-Centric Networking (CCN) [7] is a new architecture for the Internet. It is one of the information-centric networking (ICN) architectures [32, 33, 34] proposed for tailoring the Internet to meet requirements of modern applications emerging on the Internet. Note that Named Data Networking (NDN) architecture [9] is a derivative work of CCN and shares the same communication model. For generality, the terms NDN and CCN are used interchangeably in this dissertation.

CCN communication model is centered around the content rather than the locations of hosts. CCN identifies and routes packets with unique and hierarchical prefix names. The prefix name can be based on the Uniform Resource Identifier (URI) Representation [7, 11, 12]. There are two types of CCN packets, *Interest* and *Data*. Figure 2.1 shows the formats of Interest and Data packets. An Interest packet contains Content Name, Selector, and Nonce. Content Name is the name of a desired content object. Selector is an optional field for selecting a correct version of interested content object. Nonce is a random number that is used with Content Name to detect looping Interest packets. A Data packet contains Content Name, Signature, Signed Info, and Data. Content Name is the name of Data which is the requested content object. Signature and Signed Info

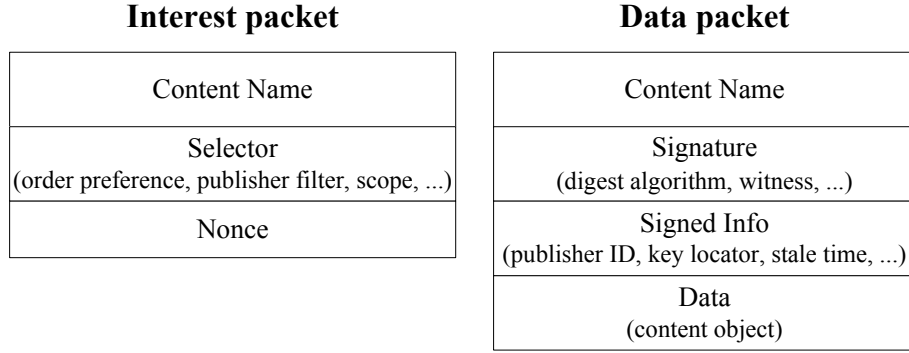


Figure 2.1: Packets types of CCN.

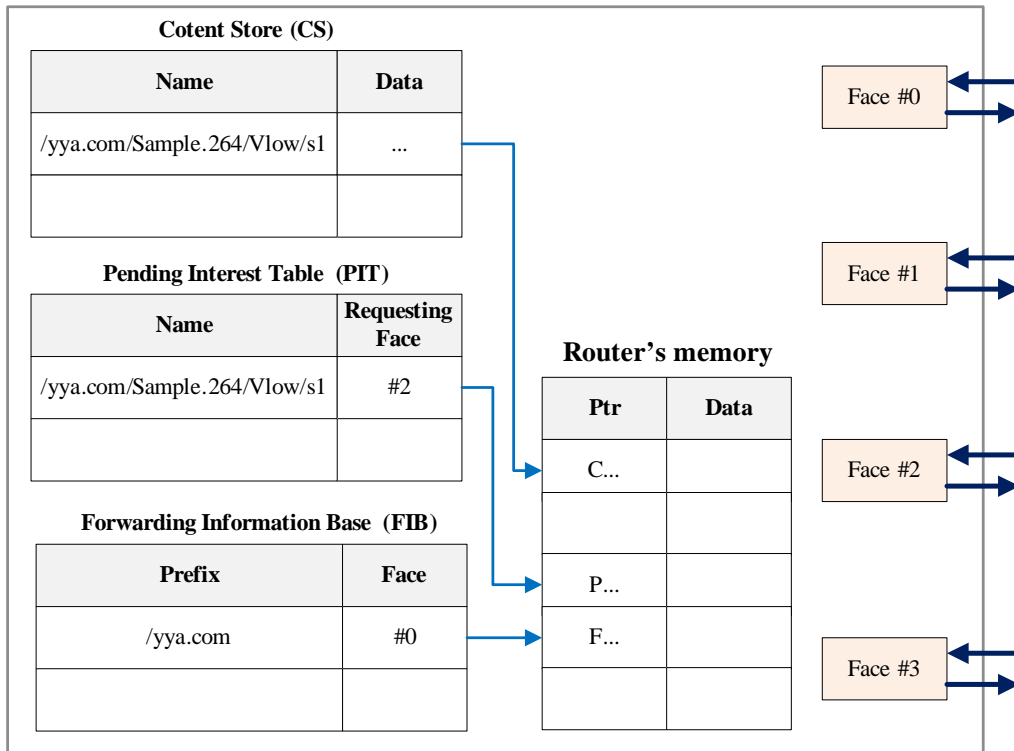


Figure 2.2: Components of CCN router.

are security information used for data validation. Having these components in the Data packet, the protection and trust of communication in CCN are the properties of content itself, not the connections on which it travels.

The architecture of a CCN router can be described as follows. Figure 2.2 shows three data structures maintained by a CCN router. The data structures consist of: Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT).

- FIB of a CCN router is partly similar to the routing table of a conventional IP router whose destination field is changed into the prefix field of content names. It is used

for determining where an Interest packets is forwarded to.

- CS of a CCN router is similar to the buffer memory of an IP router but has different objective. It is used for caching content objects carried by arriving Data packets so that the CCN router can satisfy some successive Interest packets.
- PIT is responsible for tracking the pending Interest packets that have been forwarded toward content sources. It collects the breadcrumbs left by forwarded Interest packets in a form of PIT entries. The PIT entries give breadcrumb trails for Data packets to reach all of concerning content requesters.

Communication in CCN is driven by exchange of Interest and Data packets. It is initiated by a content requester inserting an Interest packet into a content-centric network. Upon an arrival of the Interest packet at a CCN router, the CCN router searches its CS for Data matching the Content Name of the Interest packet. If there is a matching Data, the CCN router will send a Data packet containing the Data to the content requester. Otherwise, the CCN router checks its PIT whether any Interest packet having the same Content Name has been forwarded. If so, a “Face” where the Interest packet arrives on, denoted by a requesting face, is appended to an existing PIT entry in the PIT. The Face broadly denotes a hardware interface or an application process within a machine. Otherwise, a longest prefix match is performed in FIB to determine where to forward the Interest packet. Then, the CCN router forwards the Interest packet to the next hop and inserts a new PIT entry with the Content Name and the requesting face into its PIT.

When the Interest packet meets a matching Data, in either a CCN router or a content server, a Data packet will be sent to the content requester in a reverse path of the Interest packet. This path is pointed by PITs of the CCN routers involved in transmission of the Interest packet. When the Data packet arrives at a CCN router, the CCN router forwards the Data packet to the next hop through recorded requesting face(s). The associated PIT entry is also erased to indicate that the pending Interest packet has been satisfied. The Data in the Data packet can be stored in the CS of each CCN router that it traverses, depending on a caching scheme. When the CS is in a full state, a cache replace policy selects a content object currently in the CS to be replaced by an arriving one.

A few works comparing CCN to existing content delivery approach are present in

literature [8, 35]. Carofiglio et al. [8] gave a comparison between ICN architectures and Content-Delivery Networks (CDN). The comparison is based on the following metrics: heterogeneity, efficiency, reactivity, and granularity. This work points out that ICN offers many new opportunities of doing more efficient content delivery than CDN, since the limitations of CDN in the mentioned metrics does not apply to ICN architectures. A comparative study of the performance of content-centric and content-delivery networks was carried out by Mangili et al. [35]. This study focuses on the performance in terms of a reduction of the total bandwidth consumed in networks. Interestingly, the study results show that CDN provides slightly better performance than CCN, even though CDN in practice requires a much higher degree of complexity in operation and management than CCN.

2.1.1 Routing Schemes for Content-Centric Networking

Several shortest-path routing schemes for CCN were proposed in [36, 37, 38]. By using different techniques, these routing schemes try to route each content request to a closest content repository in the shortest path basis while addressing the scalability and overhead issues in CCN. They all succeed in offering a robust and efficient mechanism for the shortest path discovery. However, the in-network cache utilization is not considered by these routing schemes.

A popularity-based routing scheme for CCN has been recently proposed in [39]. This routing scheme performs load balancing based on the popularity of the content retrieval. This approach categorizes content objects with their popularity and lets CCN routers implement different cache replacement policies for different content categories. This routing scheme improves the cache utilization but it requires a system resource to observe the popularity of all content objects in a real-time basis, which may not be applicable in practice.

The other work that uses content retrieval statistics in a routing scheme was presented in [12]. In the first step, this scheme lets each CCN router broadcast arriving Interest packets to all faces and wait for corresponding Data packets from a CCN router connected to a content provider, which is called a provider router from now. This process continues for a period of time. In the second step, after the provider router has learned the content

popularity, it floods the prefix announcement messages in the considering content-centric network to construct the FIB tables of other CCN routers. This method effectively addresses a problem of FIB explosion since only the popular prefixes are maintained by FIBs. Nevertheless, the scheme ignores the caching functionality in the CCN architecture.

2.1.2 Cache Management for Content-Centric Networking

Evaluation of caching schemes and replacement policies has been extensively studied in literature. However, caching and replacement methods, which cooperatively manage a caching system, were often evaluated as separate studies.

A number of caching strategies for content-centric networks were proposed in [18, 19, 13, 20, 21, 40]. A content-popularity-based caching scheme was presented in [18]. In this scheme, each node maintains the local content popularity by counting the number of received requests for each content names. When the number of received requests for a content name at a node reaches a threshold, this content is considered a popular content, and this node suggests to neighbor nodes that they should cache the popular content. This scheme has been shown to be more efficient than a universal caching scheme in terms of cache hit rate and content diversity in network. However, this scheme requires that each CCN router maintains the popularity indexes of all contents, introducing large overheads and complexity.

A number of cooperative caching strategies were presented in [19, 13, 20, 21]. In these strategies, the entire nodes in a content-centric network maintain the information of content popularity through synchronization and make a cooperative caching decision based on this information. While being efficient, these sophisticated caching schemes can be hardly implemented in practice because of the large scale of networks, link latency, and communication overheads.

Chai et al. [40] proposed a centrality-based caching algorithm whose caching decision is based on the concept of betweenness centrality. The performance of their caching method was well supported by simulation results but it was evaluated only with a cache replacement policy (LRU).

2.1.3 Performance Analysis of Probabilistic Caching Systems

There are a number of analytical models for performance evaluation of caching systems in literature. King [41] used Markov chains to model caching systems with LRU and FIFO policies. They derived the steady-state performance of the caching systems from the Markov chains. Markov chains were also used by Gelenbe [42] to model caching systems with RR policy. He used his model to show that RR and FIFO policies yield that same steady-state performance of caching systems for an IRM request traffic. By using the Markov chain-based approaches, the performance of caching systems are evaluated at a computational cost that grows exponentially with the cache size and item population. To improve the scalability of the analytical model proposed by [41], Dan and Towsley [43] proposed an approximate analysis for LRU and FIFO policies. Their approach offers accurate estimations of caching performance at an affordable computational cost.

Garcia-Reinoso et al. [44] used Markov chains to model the behavior of a discrete cache running a probabilistic caching scheme and LRU policy. They developed an iterative algorithm to be used for estimating the cache hit ratio for each item. They showed that a cache hit ratio for most popular items can be improved by decreasing a caching probability. Unlike ours, this analytical model did not address the impacts of a caching probability on RR and FIFO policies.

The in-network caches of content-centric networks can be viewed as cache networks, which are constituted by the caching systems of interconnected CCN routers. Therefore, investigation of the cache networks' behaviors is useful and necessary. In [45], an algorithm for multi-cache approximation, named as *a-NET*, was proposed to investigate the behavior of cache networks. The *a-NET* algorithm is based on a single cache approximation for LRU policy, which was proposed by Dan and Towsley [43]. Nevertheless, the results obtained from *empha-NET* exhibit nontrivial errors (almost 16%). Authors have concluded that these errors are caused by non-IRM miss streams between caches. Wang et al. [46] developed a multi-cache with aggregation approximation (MCAA) algorithm to obtain the cache miss rate and virtual routing-trip time (VRTT) in content-centric networks. The MCAA algorithm considers the effects of Interest packet aggregations in content-centric networks and offers more accurate results than *a-NET*. However, both *a-NET* and

MCAA do not take into account the impacts of a probabilistic caching scheme.

By using Markov chains, Resensweig et al. [47] carried out a study of the ergodicity of cache networks. They pointed out that the steady-state performance of a caching system depends on the initial state of the system for some cache replacement policies such as FIFO policy. The findings address a question of whether or not ones must vary the initial state of the caching systems in their experiments for a valid evaluation.

In addition to the approaches based on Markov chains, other analytical modelling techniques have been proposed for evaluating caching systems. Che et al. [48] developed an analytical model for characterizing an uncooperative two-level hierarchical caching system, where each cache locally and independently runs LRU policy. This analytical model is later regarded as Che's approximation. In the Che's approximation, the cache miss rate of a caching system can be estimated by assuming exponential sojourn times of items in the caching system. Despite several assumptions, the Che's approximation can accurately predict the performance of caching systems. Fricker et al. [49] provided a mathematical explanation for the success of the Che's approximation and derived an approximate model for caching systems with RR policy. However, none of these models consider the caching systems that run a probabilistic caching scheme.

By extending the Che's approximation, Martina et al. [50] proposed a unified approach to performance analysis of caching systems. Several combinations of caching schemes and cache replacement policies are considered by using this approach. Importantly, the approach can be used to accurately approximate the performance of caching systems with a probabilistic caching scheme and LRU policy. Unfortunately, this work did not address the interplay between a probabilistic caching scheme and other cache replacement policies such as RR and FIFO.

Another work closely related to performance evaluation of probabilistic caching systems was carried out by Bianchi et al. [51]. In this work, a new analytical model is developed to evaluate the performance of caching systems that randomly cache just a fraction of content objects arriving at CCN routers. The motivation of this work is grounded in a security point of view in the CCN architecture. To ensure that CCN routers store only valid content object, content integrity and provenance verification is required. The speed of this verification might be much slower than the data forwarding speed of CCN routers,

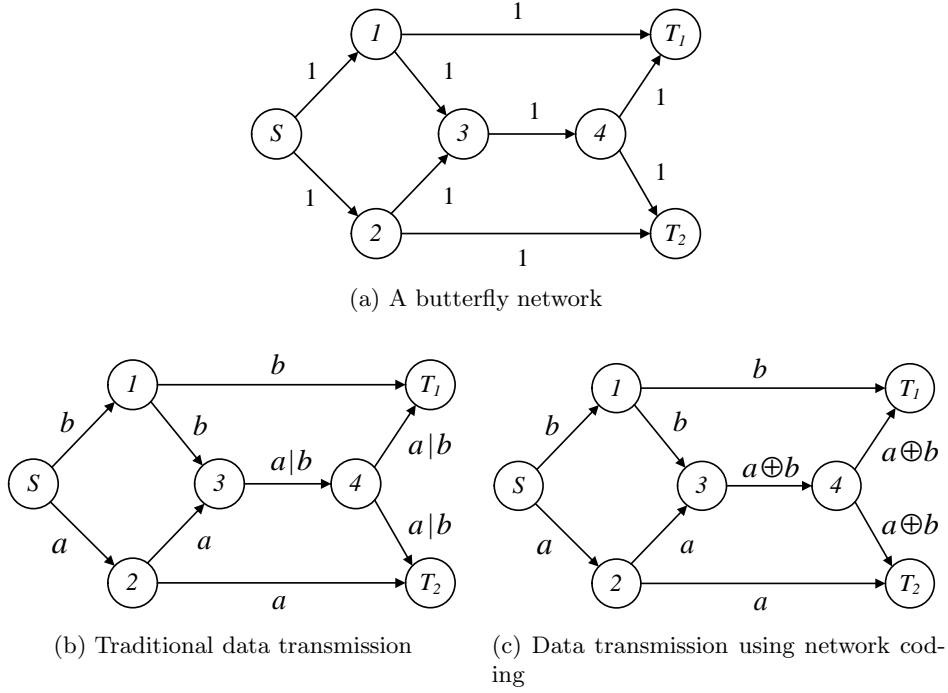


Figure 2.3: Data transmissions in a butterfly network.

thus limiting caching verified content object to a fraction of forwarded ones. They pointed out that a reduction of caching probability may increase or decrease the cache hit rate depending on the amount of temporal locality of the requests. Nevertheless, this analysis focuses only on the caching systems with LRU policy, whereas RR and FIFO policies are out of their study scope.

2.2 Network Coding

Network coding was introduced along with max-flow min-cut theorem [28] as a new approach to perform data transmissions in modern computer networks. Many studies have been conducted in various directions to extend this seminal work. The term network coding in this dissertation refers to random linear network coding [52], which is a randomized coding method that maintains a vector of coding coefficients for each of the packets sent from sources. The vector of coding coefficients is updated by each coding node along transmission paths. An example of a vector of coefficients is a vector containing 0's and 1's elements, which represent an XOR operation of inputs.

Consider a butterfly network in Figure 2.3(a). Each communication link of the network

can accommodate a packet per transmission time slot. Source node S wants to multicast two packets, a and b , to terminal nodes T_1 and T_2 . With traditional transmission scheme in Figure 2.3(b), nodes 3 and 4 must transmit either packet a or packet b to node T_1 or node T_2 per transmission time slot, as a result of a bottleneck link between nodes 3 and 4. Figure 2.3(c) demonstrates data transmission using network coding. Node 3 produces a new packet by taking an XOR (\oplus) operation of packets a and b and transmits the new packet to both terminals through node 4. Taking an XOR operation, node T_1 decodes packet $a \oplus b$ using packet b to obtain packet a . With the same method, T_2 can obtain packet b . This example shows an advantage of the network coding technique in terms of throughput improvement. In addition, network coding can be used to improve the robustness to packet losses and link failures, as well as the security of data transmissions [53, 52]. In Chapter 7, we exploit the network coding technique to improve the robustness of multimedia transmissions in multi-hop wireless networks.

Moreover, network coding can be used along with broadcast nature of wireless signal transmitters to improve channel utilization. Katti et al. [29] proposed COPE, which is also regarded as cooperative network coding (CNC) [30], to demonstrate a new concept for utilizing network coding in wireless networks. The design of CNC is based on the theory of network coding, focusing on taking the XOR operation of multiple independent data in certain network topologies. An example of CNC is shown in Figure 2.4. The considering network consists of three wireless nodes, where all nodes have the same transmission range. Specifically, nodes R_1 and R_3 are in the transmission range of node R_2 but node R_1 is out of the transmission range of node R_3 . Node R_1 wishes to send packet a to node R_3 while node R_3 wishes to transmit packet b to node R_1 . Packets a and b are the same in terms of size.

Define $R_i \xrightarrow{x} R_j$ as a broadcast transmission of packet x from R_i to R_j . With a traditional transmission scheme shown in Figure 2.4(a), the two packets are relayed and forwarded to the next terminals by router R_2 . In other words, we need four broadcast transmissions: $R_1 \xrightarrow{a} R_2$, $R_3 \xrightarrow{b} R_2$, $R_2 \xrightarrow{a} R_3$, and $R_2 \xrightarrow{b} R_1$. To reduce the number of transmissions, we can employ CNC at node R_2 as in Figure 2.4(b). That is, node R_2 performs an XOR operation of packets a and b to produce packet $a \oplus b$. Packet $a \oplus b$ can be transmitted to nodes R_1 and R_3 with a single broadcast transmission because

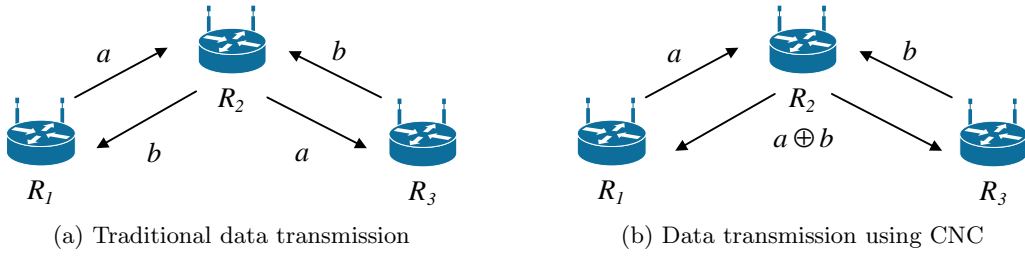


Figure 2.4: Data transmissions in a wireless network.

both nodes R_1 and R_3 are in the transmission range of R_2 . Nodes R_1 and R_3 decode their received packets to obtain packets b and a , respectively, by using an XOR operation. We can achieve the same goal with just three transmissions: $R_1 \xrightarrow{a} R_2$, $R_3 \xrightarrow{b} R_2$, and $R_2 \xrightarrow{a \oplus b} R_1, R_3$. In this example, CNC reduces the total number of transmissions in the network from four to three, producing a coding gain of $4/3 = 1.33$ [29]. This CNC topology is called an A-B structure.

In addition to the A-B structure shown in this example, there are other topologies in which CNC can be used, such as X, Cross, and Wheel structures [29]. In Chapter 6, we carry out an investigation of the interplay between the reliability of data encoded with CNC and packet loss ratio of lossy channels.

2.2.1 Unicast Transmission with Cooperative Network Coding

A number of studies were carried out to show the benefits of network coding in unicast transmissions over wireless channels. Traskov et al. [54] studied network coding for multiple unicast sessions. They proposed code construction techniques for certain connection points that are feasible with a network coding technique called the poison-antidote concept. Li et al. [55] investigated the benefit of network coding in the routing of multiple independent unicast transmissions. They pointed out that the coding advantage is not finitely bounded in directed networks. In undirected networks, they showed that the potential for network coding to increase achievable throughput is equivalent to its potential to increase bandwidth efficiency.

Based on COPE introduced by [29], a study on network coding-aware routing for unicast sessions in wireless networks was carried out by Sengupta et al. [56, 57]. They demonstrated that network coding-aware routing yields better throughput than network

coding-oblivious routing. In [30], a study on CNC-aware routing in multi-rate networks was carried out. This work extends COPE by exploiting spatial diversity in packet routing to improve the opportunity for using CNC. The bound on the throughput gain of network coding and broadcasting in wireless networks was studied in [58]. The authors showed analytically that the benefit was upper bounded by a constant for both the protocol model and the physical model of wireless transmissions. Wei et al. [59] proposed a network-coding-aware routing protocol in lossy wireless networks. This method offers throughput improvement via the structure selection of CNC.

All of the aforementioned studies use optimization problems to obtain their routing solutions. However, they do not consider QoS guarantee for layered data transmissions.

Due to the limited transmission range of a wireless node, it is typical for a source node to transmit data to a destination node by going through several intermediate nodes. Layered video transmission in wireless networks using relay nodes was proposed by Alay et al. [60]. Layered video transmission employs successive refinement of information or scalable coding was considered in [31]. Video streaming using network coding over wireless networks was proposed by Seferoglu et al. [61]. The proposed video-aware opportunistic network coding scheme considers the decodability of network codes by multiple receivers as well as the relative importance and delay of video packets. However, the QoS guarantee issue has not yet been examined in depth in these papers.

Mahapatra et al. [62] proposed a QoS-and-energy-aware routing scheme for real-time traffic in wireless sensor networks. The scheme employs an adaptive prioritized Medium Access Control (MAC) to provide a differentiated service model for real-time packets. However, network coding was not considered in [62]. More recently, Supittayapornpong et al. [63] proposed a framework of layered data multicasting with QoS guarantee, which includes network code assignment to each node in the network. In addition, a practical algorithm which calculates the optimal network code length providing QoS guarantee for wireless multicast was proposed in [64]. However, their framework did not address the issue of network contention due to the co-existence of multiple unicast video streams.

Greco et al. [65] proposed a framework for reliable video streaming in lossy wireless networks using Expanding Window Network Coding (EWNC), Multiple Description Coding (MDC), and a novel Rate-Distortion Optimised (RDO) scheduling algorithm. However,

they assumed that multiple sources transmit the same video to a single receiver. In addition, their framework cannot be applied to the streaming of layered videos. Oh et al. [66] proposed a practical online scheduling algorithm for mobile video streaming to multiple clients. In their work, an access point (AP) constructs and broadcasts the best network code, which is based on the packets of I-frames with high Peak Signal-to-Noise Ratio (PSNR) during a Group of Picture (GoP), to all clients. Their framework well addressed a problem of single-hop video transmissions from an access point to mobile users in lossy wireless networks. However, they did not consider multi-hop transmissions in multi-hop wireless networks.

Yang et al. [67, 68] proposed a network coding-based multipath routing (NCMR) scheme for wireless sensor networks. They used random linear network coding to improve the reliability of the data transmission on braided multiple paths. Their approach was proven to be efficient in terms of energy consumption, which can be shown by a reduced number of transmissions in wireless sensor networks. Nevertheless, the QoS requirement and transmission rate were not primarily considered in their works.

Ning et al. [69] studied the relation between network coding and spatial reuse in wireless mesh network. They pointed out that greedy network coding (i.e., establishing network coding as much as possible) may reduce network throughput because of a decrease in spectrum spatial reuse. To solve the problem, they proposed an optimal network coding-aware scheduling scheme which includes a power control mechanism, taking into account both adaptive relaying method selection and spatial reuse. They formulated an integer linear programming to solve the link scheduling problem. An adaptive power control method with a network coding-aware link scheduling scheme was developed to further improve wireless spectrum utilization. Even though their proposed scheme offers a significant improvement in network throughput, the importance of transmitted data and QoS guarantee were out of their study scope.

Uddin et al. [70] carried out a study of a cross-layer design in random-access-based fixed wireless multihop networks with a physical interference model. They considered ALOHA medium access control protocol for link-layer operation. An optimization problem was formulated for obtaining the maximal throughput, which is provided by an optimal configuration of the routing, access probability, and transmission rate in a slotted ALOHA

system. They also modified the mentioned problem so that it enables XOR-based network coding without opportunistic listening. Because their primary goal is to achieve the maximal throughput, QoS of transmitted data cannot be guaranteed by using their proposed optimization frameworks.

2.2.2 Multicast Transmission with Network Coding

A number of single-source routing schemes for multicast transmission with network coding were proposed in literature. Zhu et al. [71] presented a set of distributed algorithms to improve the achievable data rate of an end-to-end multicast session using network coding. They used redundant paths from a single source to multiple sinks to achieve a better achievable data rate. Sundaram et al. [72] proposed a polynomial-time algorithm for multicasting layered data to heterogeneous receivers using network coding. The algorithm gives a transmission rate equaling max flows of all receivers. Supittayapornpong et al. [63] proposed a framework for multicasting layered data with QoS guarantee. They formulated an optimization problem to select reliable paths for layered data transmission. A heuristic algorithm for selecting static network-codes was also proposed. All above-mentioned works are limited to the case that there is only one source in the network. When the source fails to transmit data, destinations do not have alternative sources to receive the transmitted data. This will cause a service interruption at receivers.

Kim et al. [73] proposed the *pushback algorithm* for multi-resolution multicast. The goal is to maximize the total rate achieved by all destinations, while guaranteeing the decoding probability of the base layer at each receiver. However, QoS guarantee and packet loss were not considered in their work, which may not reflect the real-world situation. Xu et al. [74] proposed a layered separated network coding scheme that aims at maximizing total layers received by all receivers. They did not consider packet loss, which is an important characteristic of wireless network. Zhao et al. [75] proposed a solution to improve the throughput of an overlay multicast session with slightly higher delay and network resource consumption. However, this work was applied only in lossless environment.

A network may have multiple sources, servers, or even buffered storage units containing replicated contents. To give an example, when a client initiates a data request, it can receive data not only from the servers but also from the surrounding peers that buffer

the same data [76]. The recent work in [7] described an efficient caching mechanism that can improve the transmission of streaming data in multi-source environments. Nodes can cooperate to exploit the diversities of sources and paths to access data in order to improve the reliability of data transmission. Han [77] presented necessary and sufficient conditions for reliable transmission over a network for multicasting multiple correlated sources to multiple sinks in lossy channels. When caching at nodes in a multi-hop wireless network is possible, the video file can have multiple replicas from previous request. Consequently, the newly requesting client might have multiple sources for accessing the video file.

Li et al. [55, 78] set up a route selection method for Video-on-Demand streaming system in a multi-hop wireless networks using the rate/interference-distortion optimization. They proved that the source diversity can improve wireless video streaming quality. Their work aimed at streaming video clips having multiple descriptions generated using multiple description coding (MDC), which is different from scalable video coding in terms of prioritizing hierarchy of data. However, MDC is not widely used because it is not a part of video coding standards. A peer-to-peer (P2P) streaming scheme for scalable video was investigated to enhance video transmission over the Internet [79, 80, 81]. Xu et al. [80] proposed a peer-to-peer video-on-demand system using multiple descriptions and source diversity, where multiple ordinary computers (peers) were used as servers and the client can retrieve different layers of the video file from these peers. Although the transmission scheme could deliver high quality services to users, it consumes a large amount of network resources. Selvam et al. [81] presented a cluster-based approach for multi-source multicast routing protocol in a mobile ad-hoc network. However, reliability was not included in their path selection algorithm. Although path diversity was taken into account, the bandwidth may not be used efficiently since redundant data may be transmitted through paths with a higher packet loss rate.

Ho et al. [52, 82] presented a distributed random linear network coding for transmission and compression of information in a multi-source multicast network. Each node in a network independently and randomly selects linear network codes, which indicate the correlation of input and output information. They proposed a general bound on successful transmission probabilities of such codes for arbitrary networks, showing that the successful decoding probability increases exponentially with the code length. However, this work

considers only lossless environments and uses the exact decoding probability of network coding instead of a bound of probability in designing a transmission system to give more accurate results. Trullols-Cruces et al. [83] proposed an exact decoding probability of random network coding. They computed the exact decoding probability that a receiver obtains N linearly independent packets over $K \geq N$ received packets of random network coding over a Galois Field, where K is the number of transmitted packets from sources. Zhao [84] gave a less complicated derivation of the exact decoding probability. We use the results of [83, 84] to calculate an exact decoding probability of random network coding for our proposed routing scheme in Chapter 7.

Chapter 3

Cooperative Routing for Content-Centric Networking

This chapter presents a cooperative routing scheme for CCN along with an Optimal Cooperative Routing Protocol (OCRP). The OCRP selectively aggregates flows of Interest packets onto a common path to improve the cache utilization while mitigating cache contention inside CCN routers. Simulation results show that OCRP offers an improvement in server load and round-trip hop distance in comparison to the shortest path routing scheme.

3.1 Introduction

Typically, CCN packets are forwarded in the shortest-path manner [11, 12]. Although this approach is widely used, it may not fully utilize the caching benefits of CCN routers. Different CCN routers may use disjoint paths to forward their packets, even though these packets contain the same content. As a result, each CCN router would benefit from the cached content that are subject to only its previously transmitted data. Moreover, when different contents arrive at a CCN router, the CS of the CCN router must handle all of the diverse contents. The contents may be so many that the cache cannot manage them efficiently and would result in poor caching performance.

In this chapter, we present an Optimal Cooperative Routing Protocol (OCRP) for content-centric networks to improve the utilization of in-network caches in content-centric

networks. The objective of OCRP is to selectively aggregate the multiple flows of Interest packets onto the same path to improve the cache utilization while mitigating the cache contention of the CSs of CCN routers on the routing path. The OCRP consists of three main processes: (1) Prefix Popularity Observation; (2) Prefix Group (Un)Subscription; and (3) FIB Reconstruction. Prefix Popularity Observation lets each CCN router observe the popular prefixes to activate the Prefix Group (Un)Subscription. Prefix Group (Un)Subscription lets the designate router (DR) know which CCN router demands to enter or leave which prefix group. The DR selects an optimal cooperative path based on the prefix groups. FIB Reconstruction updates the FIB entries of the CCN routers involved in the newly computed optimal cooperative path. The optimal cooperative path is obtained by solving an integer linear optimization under flow conservation constraint, cache contention mitigating constraint, and path length constraint. The server load and round-trip hop distance are used to evaluate the performance of OCRP in comparison to the shortest path routing.

The rest of this chapter is organized as follows. Section 3.2 describes OCRP and network requirements. Section 3.3 introduces the integer linear optimization problem that is used for the optimal cooperative path calculation. Section 3.4 presents the performance evaluation of OCRP as well as the discussion on the simulation results. Finally, the conclusion are given in Section 3.5.

3.2 Optimal Cooperative Routing Protocol

The goal of our Optimal Cooperative Routing Protocol (OCRP) is to selectively aggregate multiple flows of Interest packets onto the same path to improve the cache utilization, aiming to mitigate the cache contention of the CSs of CCN routers. Figure 3.1 shows an example of routing path selection for multiple flows of Interest packets based on cooperative routing for CCN.

Two logical types of CCN routers in OCRP are defined as follows.

- **Requester Router** is a CCN router that generates or receives Interest packets from the other CCN routers. A requester router forwards the Interest packets to the next-hop CCN routers if a matching content object is not in its CS table.

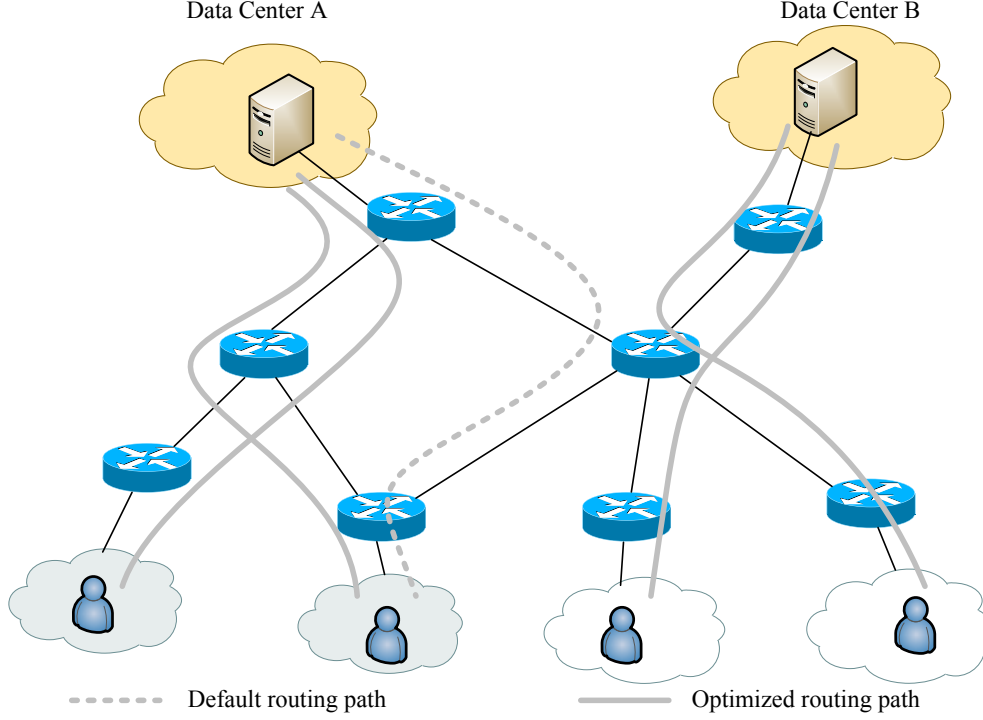


Figure 3.1: Routing path selection for multiple flows of Interest packets based on cooperative routing for CCN.

- **Provider Router** is a CCN router that originates some name prefixes. The provider router is logically connected to the server of content publisher that generates the content objects corresponding to the name prefixes.

One of highly desirable functions of CCN router is a traffic monitoring capability. Most previous work generally assumed that a CCN router can observe the traffic characteristics at a content chunk-level and used this information to perform either the cooperative caching [13, 18] or the optimal content assignment [85]. However, practical limitations of this approach are from the fact that the number of unique content objects transited by a CCN router is massive. A large memory of each CCN router is necessary for keeping the statistics of the whole content objects. In addition, updating such statistics at high speed brings excessive load to the processor unit of a CCN router. Therefore, we consider another approach, which demands less memory and lower processing cost, to observe the popularity distribution of the content in a flow-level. We propose to observe the popularity of FIB entries, instead of targeting content objects. This approach can be implemented by counting the referencing times of each FIB entry within a certain time interval.

Figure 3.2 depicts an overview of the OCRP. Once the Interest packets that re-

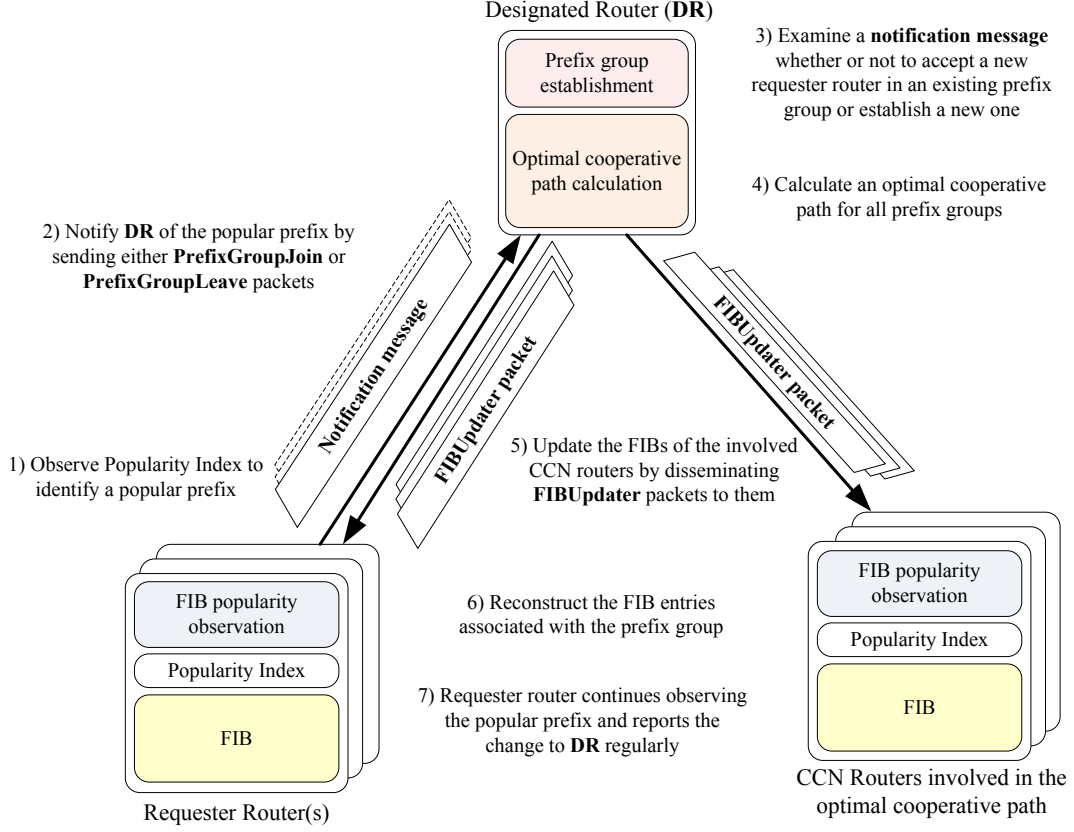


Figure 3.2: The overview of OCRP.

quest for **Similar Content** are frequently sent out from a CCN router, the router will send a notification message to the **Designated Router (DR)**, which is the central controller of the OCRP, to subscribe to an existing **Prefix Group** or to request the formation of a new one. The Similar Content refers to the different content objects that are served by the same content publisher. Therefore, their prefix names are in general similar and share some components. For example, *com/youtube/sports/8aa994*, *com/youtube/news/dssf5*, and *com/youtube/sports/552ql* are similar content objects whose prefixes all share *com/youtube/*. Fundamentally, the FIB entry of *com/youtube/* is referenced whenever these content objects are sent from the considering CCN router. The DR can be selected from the CCN router that have the highest graph-related metric, such as Degree Centrality, Betweenness Centrality, or Graph Centrality [86]. A **Backup Designated Router (BDR)** can be selected for decreasing the load of DR or to be a back up resource. The insights into collaboration between DR and BDR remains an open issue for future investigations. The DR examines the received notification messages and selects the optimal cooperative path for all active Prefix groups. We will discuss the op-

timal cooperative path calculation in Section 3.3. A Prefix Group consists of a provider router, which originates the prefix, and a number of the requester routers that frequently send out the Interest packets carrying the corresponding prefixes.

The optimal cooperative path should yield the following advantages.

- **Cache-Hit Rate:** The requester routers send the Interest packets that are requesting the content objects from the same content publisher on the same path. As a result, the possibility that CSs of the CCN routers on that path satisfy an Interest packet increases, thus improving the cache-hit rate.
- **Total Network Traffic:** According to the forwarding process of CCN, the state of an unsatisfied Interest packet is inserted into the PIT for the routing of a corresponding Data packet. If the PIT has already an entry for the prefix name, the requesting face of the Interest packet will be added to the existing entry. Therefore, one Interest packet sent out from the considering CCN router can be responsible for many successive Interest packets. Putting similar Interest packet flows on the same path increases the number of mentioned events and saves the bandwidth used for repeatedly sending the same Data packets.
- **Server Load:** The number of Data packets served by the provider router can be reduced by improving the CS and PIT utilizations.
- **Content Retrieval Time:** An Interest packet can meet its desired content in a closer CS more frequently as a result of the improved cache utilization.

Note that aggregating traffic onto the same path may cause network congestion in an IP network. However, the CCN architecture removes some of data transmission redundancy by storing data in the CSs of CCN routers while supporting multicast transmission using the redesigning forwarding process [7, 36, 87].

Finally, the FIB entries of the CCN routers that are involved in the optimal cooperative path will be updated by the update messages sent from the DR. Note that a CCN router in a content-centric network can be both the requester and provider routers of different prefix groups at the same time.

The OCRP requires two basic principles in a considering content-centric network: (1) the information of the network topology; and (2) default FIB of CCN routers.

First, each CCN router has the knowledge about the network topology it belongs to. To distinguish a CCN router from the others, each router is identified by a unique router ID [12]. For simplicity, each CCN router is uniquely named by using an integer. There is no direct relation between the router ID and content in this scenario. Each CCN router has several faces (interfaces), which are identified and locally referred to by using unique integers. Other network properties, e.g., the link bandwidth and link status, are globally exchanged throughout the network. The system can adopt **Open Shortest Path First** (OSPF) [6] to maintain the network topology and calculate the shortest path for handling the signaling processes. **Link State Advertisement** (LSA) packets, which contain the status of each face and link, are exchanged among the CCN routers. Each router eventually learns the network topology and maintains the current status by updating its **Link State Database** (LSDB) based on the exchanged LSA packets. Importantly, the DR must know the CCN routers that advertise prefixes because they are prospective provider routers.

Second, each CCN router contains a default FIB. The network system can adopt the OSPF for Named-data (OSPFN) protocol [11] to provide a name-based routing table, which is later summarized into a default FIB for each CCN router. The OSPFN protocol uses OSPF's Opaque LSAs (OLSA) to announce the name prefixes while flooding the regular LSA packets to maintain the topology database. This protocol has already been deployed at all working groups participating in the NDN testbed. We will omit the details of this protocol and focus on our proposed cooperative routing protocol from now. The proposed OCRP consists of three processes: (1) Prefix Popularity Observation; (2) Prefix Group (Un)Subscription; and (3) FIB Reconstruction.

3.2.1 Prefix Popularity Observation

The prefix popularity observation function is conducted in each CCN router. The objectives of this process are to observe the popularly cited prefixes and to activate a prefix group (un)subscription function, which lets the DR router know which requester router wants to either join or leave a prefix group.

Once an Interest packet confirms that its prefix is not in the PIT or the desired content

object is not in the CS, the FIB is checked for the longest prefix match. After the longest prefix match is found, the relevant face (interface) is selected to forward the Interest packet. We can observe the popularity of a prefix citation by tracking the number of times that each FIB entry is referred to during a period of time. **Exponentially Weighted Moving Average** (EWMA) can be exploited to estimate the popularity of the prefix citation and to generate the *popularity index*. The popularity index, therefore, practically represents the popularity of the prefix. It is interesting to note that the memory used for tracking the prefix popularity may not increase proportionally to the total number of content objects. In contrast, it depends on the number of FIB entries, which can be regulated by using a FIB construction policy. This function does not lead to the limitation of an individual observation of each content object, i.e., content popularity observation at a content object-level. The observation process is independent of the forwarding process of a CCN router, so it does not cause additional forwarding complexity. Figure 3.3 demonstrates the process of handling an Interest packet while tracking the popularity of each FIB entry. When the popularity index of an observed FIB entry reaches a given threshold, which is equal to 50 in this example, the protocol activates the prefix group (un)subscription function.

The time interval used for sampling the prefix popularity is a significant factor for optimizing the system performance. A CCN router can capture a popular prefix sooner when the designated time interval is shorter. As a result, the optimal cooperative path is computed based on recent traffic information. The next Interest packet transmissions gain more benefit from the newly computed optimal cooperative path than that of the outdated ones. However, a shorter time interval may increase the computational cost of the popularity observation process as well as increasing the signaling overheads, which is harmful for routing stability and scalability.

3.2.2 Prefix Group (Un)Subscription

A CCN router calls the prefix group (un)subscription after the popularity index of an observing prefix citation has reached a given threshold. The objective of this process is to let the DR know which requester router demands to join which prefix group. On the other hand, a requester router may want to leave a prefix group when the popularity index of the prefix citation becomes lower than the threshold. There are two signaling

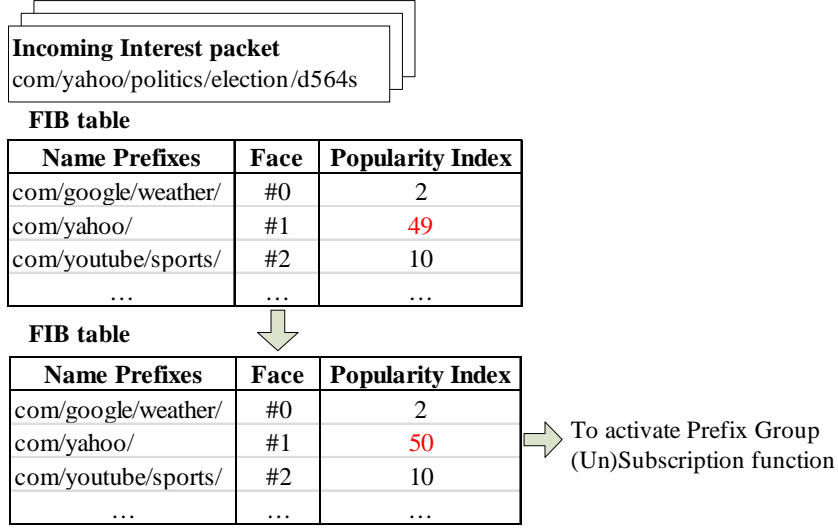


Figure 3.3: Process of handling Interest packets and tracking of popularity of each FIB entry.

packets related to this function, i.e., *prefixGroupJoin* and *prefixGroupLeave* packets shown in Figure 3.4. A requester router sends a *prefixGroupJoin* packet to the DR to join a prefix group if it already exists. Otherwise, a new prefix group can be established. When the *prefixGroupJoin* packet has arrived at the DR, the DR decides whether or not it can accept the *prefixGroupJoin*. The criteria for the request acceptance are as follows.

- During a time interval, there is more than one requester router that is willing to join the prefix group.
- The provider router that advertises the prefix still has available link capacity for the new requester router. The request message is denied if a bottleneck link will occur to any outgoing link from the provider router due to the added traffic. Nevertheless, the Interest packets sent from the anticipated requester router can access the provider router by using the default route despite the denial of its *prefixGroupJoin*.

The last requirement is necessary for bottleneck avoidance, which can be caused by putting too much traffic on the optimal cooperative path. If these requirements are satisfied, the optimal cooperative path selection for all prefix groups, which will be explained later in section 3.3, is conducted at the DR. Otherwise, the *prefixGroupJoin* packet is discarded. The anticipated requester router needs to wait for a random period of time to resubmit the *prefixGroupJoin* packet to the DR.

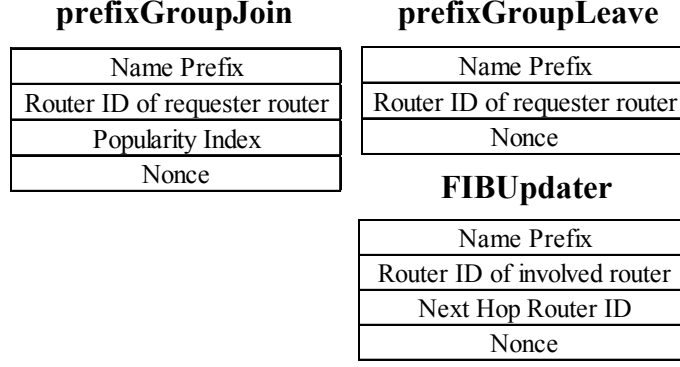


Figure 3.4: Format of each signaling packet.

When the popularity index of the prefix citation becomes lower than the threshold, a requester router will send a *prefixGroupLeave* packet to the **DR**. In contrast, there is no acceptance evaluation for the *prefixGroupLeave* packet. The optimal cooperative path is calculated whenever a *prefixGroupLeave* packet arrives at the DR so that the newly computed optimal cooperative path can offer a better performance to the remaining members of the prefix group. After the optimal cooperative path calculation has been completed, the DR then calls the FIB reconstruction function in Section 3.2.3.

3.2.3 FIB Reconstruction

The objective of this process is to reconstruct the FIB entries of the CCN routers involved in the newly computed optimal cooperative path of all prefix groups. The DR creates a number of *FIBUpdater* packets and sends each of them to the involved routers. A *FIBUpdater* packet contains the necessary information used by a CCN router to reconstruct the FIB entries associated with the prefix. Figure 3.4 demonstrates the format of the *FIBUpdater* packet. The disseminations of the *FIBUpdater* packets exploit the knowledge of the network topology from the LSDB. The FIB reconstruction starts taking action from the CCN router being the nearest to the provider routers to the requester routers in an orderly fashion, so that the flows of the Interest packets proceed without interruption.

The FIB reconstruction process is shown in Figure 3.5. When a *FIBUpdater* packet arrives at its relevant CCN router, the FIB entries are updated according to the information residing in the *FIBUpdater* packet. The name prefix and its next hop router ID, which is mapped to the corresponding face, are inserted into the FIB table with the highest prefer-

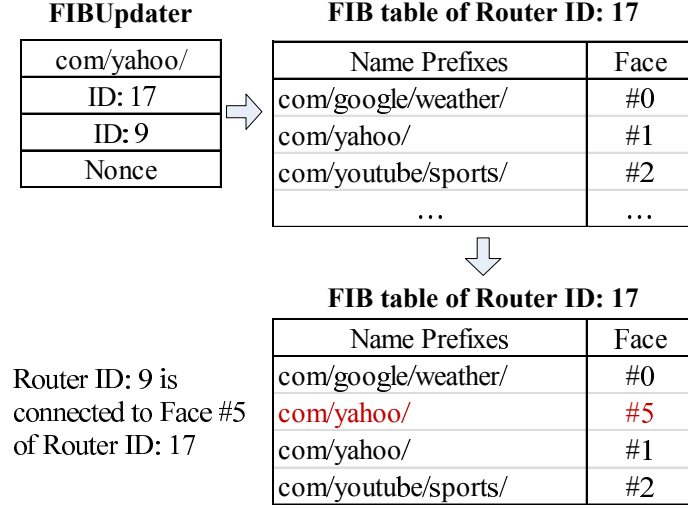


Figure 3.5: FIB reconstruction process of involved CCN router.

ence in comparison with those of the existing entries. This updating process preserves the conventional benefit of CCN which inherently supports a multipath configuration. The backup paths of the prefix remain prompt when there is link failure in the preferable path. We follow the basic multiple paths configuration of CCN found in [11], i.e., the face having the highest preference, which can be defined by the lowest path cost, is selected to send the Interest packet associated with that prefix.

The forwarding process at each CCN router can continue while the system is deploying the optimal cooperative path. The transmission path gradually changes during the FIB reconstruction step. Eventually, the next Interest packets, which match the prefix, will travel down the optimal cooperative path. Each requester router continues tracking the popularity of the prefix and update its status to the DR. Importantly, to prevent the network from the overwhelming signalling messages caused by the OCRP, each requester router should send an update message periodically after it waits for a decent time interval. The popularity of prefix in general tends to continue for a long time (from a few hours up to several day) [4]. We, therefore, adjust the interval over a relatively long-term period to compromise on the signalling overheads with the accuracy of the optimal cooperative path. The DR can conduct an optimal cooperative path selection and update the associated routing paths when the change of network topology is detected.

3.3 Optimal Cooperative Path Selection

In this section we describe an essential component of our cooperative routing protocol, which is the optimal cooperative path selection. The DR has established a number of unique prefix groups as a result of receiving a number of *prefixGroupJoin* and *prefixGroupLeave* messages from requester routers. Each prefix group contains a provider router and a number of associated requester routers. The DR performs the optimal cooperative path selection for all registered prefix groups by solving an integer linear optimization problem. The objectives of the optimal cooperative routing are: (1) to encourage similar Interest packets to travel on the same path so that the relevant content objects, which are carried by associated Data packets, are possibly cached in the same set of CSs; and (2) to separate the path used by a prefix group from those by the others to minimize the set of the CSs of CCN nodes that are commonly used by different prefix groups.

3.3.1 Network Model

A network is modeled as a directed graph $G(N, E)$, where N and E are the sets of CCN routers and directed links in the network, respectively. Let i be the index of an established prefix group in the network. A set of the indices of prefix groups is defined by I , where $i \in I$. Define (s, d, i) as an approved transmission session in prefix group i , where s and d are the provider router and a requester router, respectively. A set of all (s, d, i) in the network is defined by Γ , i.e., $(s, d, i) \in \Gamma$. For link $l \in E$, let $\mathbf{t}(l)$ and $\mathbf{r}(l)$ be the transmitter and receiver routers of link l , respectively. For CCN router $n \in N$, let $T_O(n) = \{l \in E | n = \mathbf{t}(l)\}$ and $T_I(n) = \{l \in E | n = \mathbf{r}(l)\}$ be the sets of outgoing and incoming links of router n , respectively. We characterize the network properties as follows. CCN router n has a fixed size of CS whose capacity is equal to b_n while the bandwidth and average delay of link l are defined by c_l and d_l , respectively.

3.3.2 Objective Function

We have two objectives to optimize: (1) maximizing the number of the transmission sessions participating in the optimal cooperative path; and (2) minimizing the cost of using this path. Define $x_{(s,d,i)}$ as a binary variable that indicates whether or not transmission

session (s, d, i) joins the optimal cooperative path. If $x_{(s,d,i)} = 1$, transmission session (s, d, i) participates the optimal cooperative path. Otherwise, $x_{(s,d,i)} = 0$. Define $w_{(s,d,i)}$ as a weighting parameter used to prioritize a particular term in the multi-objective function. Define t_l^i as a binary variable indicating whether or not link l is used in the optimal cooperative path by some transmission sessions in prefix group i . If $t_l^i = 1$, link l is used in the optimal cooperative path by at least one transmission session in prefix group i . Otherwise, $t_l^i = 0$. The cost of using link l is defined by p_l , which can come from a weighted combination of link bandwidth c_l , delay d_l , and the CS size of its receiver router of the link $b_{r(l)}$. In other words, $p_l = \beta_1/c_l + \beta_2 d_l + \beta_3/b_{r(l)}$, where β_1 , β_2 , and β_3 are weighting parameters and $\beta_1, \beta_2, \beta_3 \in \mathbb{R}$, where \mathbb{R} is a set of real numbers. The multi-objective function can be written as

$$\sum_{(s,d,i) \in \Gamma} w_{(s,d,i)} x_{(s,d,i)} - \sum_{i \in I} \sum_{l \in E} p_l t_l^i, \quad (3.1)$$

where $w_{(s,d,i)} = TTL_{(s,d,i)} \times \max_{l \in E}(p_l)$, $TTL_{(s,d,i)}$ is the initial *time to live* (TTL) or a hop limit of a message belonging to transmission session (s, d, i) in the network.

The first term of function (3.1) implies the number of the transmission sessions that participate in the optimal cooperative path, whereas the second term represents the cost used by this path. We attempt to prioritize the first objective (i.e., the first term of function (3.1)) since it directly contributes to the benefits of the optimal cooperative path described in Section 3.2. We therefore set $w_{(s,d,i)}$ to the possible maximum path cost for transmitting session (s, d, i) . For instance, if a number of transmission sessions in prefix group j participate in an optimal cooperative path, the cost of using this path is equal to $\sum_{l \in E} p_l t_l^j$. Although the most expensive path is selected to be the optimal cooperative path for these transmission sessions, we still obtain $\sum_{(s,d,i) \in \Gamma_j} w_{(s,d,i)} x_{(s,d,i)} \geq \sum_{l \in E} p_l t_l^j$, where $\Gamma_j = \{(s, d, i) \in \Gamma | i = j\}$. To maximize function (3.1), $x_{(s,d,i)}$ and t_l^j should be set to 1 and 0 as many times as possible, respectively. As a result, maximizing this multi-objective function is equivalent to maximizing the number of transmission sessions participating in the optimal cooperative path while minimizing the cost used by this path.

In practice, TTL is typically determined by the Operating System (OS) of devices or the underlying routing protocol. The role of TTL is to prevent a routing loop in TCP/IP networks, whereas it has not been studied in the case of content-centric networks.

Fundamentally, $TTL_{(s,d,i)}$ of all transmission sessions should be identical to guarantee the fairness among different transmission sessions in an OCRP network. For example, $TTL_{(s,d,i)} = 64$ for all transmission sessions as it was the default TTL recommended for the Internet Protocol (IP) [88].

3.3.3 Flow Conservation Constraint

This constraint provides the connectivity of the path for each transmission session $(s, d, i) \in \Gamma$. Define $f_l^{(s,d,i)}$ as a binary variable indicating whether or not link l is used for transmission session (s, d, i) . If $f_l^{(s,d,i)} = 1$, link l is used in the optimal cooperative path by transmission session (s, d, i) . Otherwise, $f_l^{(s,d,i)} = 0$. The information flow of transmission session (s, d, i) in prefix group i from provider router s to requester router d is equal to one. In addition, the total flow into an intermediate router is equal to the total flow out of the router. Thus, the constraint on the flow conservation can be expressed mathematically as

$$\sum_{l \in T_O(n)} f_l^{(s,d,i)} - \sum_{l \in T_I(n)} f_l^{(s,d,i)} = \begin{cases} x_{(s,d,i)}, & n = s \\ -x_{(s,d,i)}, & n = d \\ 0, & \text{otherwise,} \end{cases} \quad (3.2)$$

for all $(s, d, i) \in \Gamma$ and $n \in N$. Notice that when $x_{(s,d,i)} = 0$, the total flow of transmission session (s, d, i) in prefix group i out of source s or into destination d must be zero. In other words, transmission session (s, d, i) cannot take part in the optimal cooperative path.

3.3.4 Cache Contention Mitigating Constraint

The requester routers in the same prefix group are encouraged to share the same path, whereas those in different prefix groups should not share the same path to mitigate the potential cache contention occurring in the CSs of CCN routers. The constraint satisfying the above properties can be written as

$$f_l^{(s,d,i)} \leq t_l^i, \quad (3.3)$$

for all $l \in E$ and $(s, d, i) \in \Gamma$ and

$$\sum_{i \in I} t_l^i \leq \gamma_l, \quad (3.4)$$

where γ_l is a positive integer number, for all $l \in E$. Practically, γ_l can be determined equalling a weighted combination of link capacity c_l and the size of CS of the receiver router $b_{r(l)}$. Constraint (3.3) encourages different requester routers in the same prefix group to use the same set of links. On the other hand, constraint (3.4) controls the number of prefix groups that share the same link. In this work, we let $\gamma_l = 1$, for all $l \in E$, to allow at most one prefix group to use any link. In other words, the different prefix groups all use disjoint paths.

However, setting $\gamma_l = 1$ may not be suitable in the case that the number of prefix groups is large because a number of transmission sessions (s, d, i) may not be able to join the optimal cooperative path. In that case, γ_l should be set to be larger than one.

3.3.5 Path Length Constraint

We formulate this constraint to ensure that the path length from requester router d to provider router s of transmission session (s, d, i) is not longer than that when using the shortest path whose length is equal to $h_{(s,d)}$. It can be mathematically written as

$$\sum_{l \in E} f_l^{(s,d,i)} \leq h_{(s,d)}, \quad (3.5)$$

for all $(s, d, i) \in \Gamma$. The constraint narrows down the feasible solutions of the optimal cooperative path. To allow more routes to become the candidates of the optimal cooperative path, (3.5) can be rewritten as

$$\sum_{l \in E} f_l^{(s,d,i)} \leq h_{(s,d)} + k, \quad (3.6)$$

where $k \in \mathbb{I}^+$ and \mathbb{I}^+ is a set of non-negative integer numbers.

3.3.6 Problem Formulation

To compute the optimal cooperative path for all transmission sessions in all prefix groups, an integer linear optimization is established as follows.

Maximize

$$\sum_{(s,d,i) \in \Gamma} w_{(s,d,i)} x_{(s,d,i)} - \sum_{i \in I} \sum_{l \in E} p_l t_l^i \quad (3.7a)$$

subject to

$$\sum_{l \in T_O(n)} f_l^{(s,d,i)} - \sum_{l \in T_I(n)} f_l^{(s,d,i)} = \begin{cases} x_{(s,d,i)}, & n = s \\ -x_{(s,d,i)}, & n = d \\ 0, & \text{otherwise,} \end{cases} \quad (3.7b)$$

$$\forall (s, d, i) \in \Gamma, \forall n \in N,$$

$$f_l^{(s,d,i)} \leq t_l^i, \forall (s, d, i) \in \Gamma, \forall l \in E, \quad (3.7c)$$

$$\sum_{i \in I} t_l^i \leq \gamma_l, \forall l \in E, \quad (3.7d)$$

$$\sum_{l \in E} f_l^{(s,d,i)} \leq h_{(s,d)} + k, \forall (s, d, i), \quad (3.7e)$$

$$f_l^{(s,d,i)}, t_l^i, x_l^{(s,d,i)} \in \{0, 1\}, \forall l \in E, \forall (s, d, i) \in \Gamma, \forall i \in I. \quad (3.7f)$$

An optimal solution to the problem can be obtained by various mathematical programming tools. We select CPLEX Optimizer [89] to be the solver for linear programming. An optimal solution of the problem is a set of $t_l^i = 1$, for all $l \in E$ and $i \in I$, which represents the optimal cooperative path for all prefix groups.

A set of transmission sessions $\hat{\Gamma}$, which mathematically refer to those having $x_{(\hat{s}, \hat{d}, \hat{i})} = 0$, may not be able to join the optimal cooperative path due to the lack of link availability, where $(\hat{s}, \hat{d}, \hat{i}) \in \hat{\Gamma}$. On one hand, these transmission sessions can simply use their default paths to proceed. On the other hand, a suboptimal cooperative path can be calculated for each of them. The suboptimal cooperative path of each transmission session is the shortest path between provider router \hat{s} to requester router \hat{d} that maximally shares its links with the optimal cooperative path of prefix group \hat{i} . Recall that the optimal cooperative path is an optimal solution to Problem (3.7). By using the above variables together with the set of $t_l^{\hat{s}}$, for all $l \in E$, we formulate an integer linear optimization for the suboptimal cooperative path for all $(\hat{s}, \hat{d}, \hat{i}) \in \hat{\Gamma}$ as follows.

Maximize

$$\sum_{l \in E} p_l f_l^{(\hat{s}, \hat{d}, \hat{i})} (t_l^{\hat{i}} - 1) \quad (3.8a)$$

subject to

$$\sum_{l \in T_O(n)} f_l^{(\hat{s}, \hat{d}, \hat{i})} - \sum_{l \in T_I(n)} f_l^{(\hat{s}, \hat{d}, \hat{i})} = \begin{cases} 1, & n = \hat{s} \\ -1, & n = \hat{d} \\ 0, & \text{otherwise,} \end{cases} \quad (3.8b)$$

$$\forall (\hat{s}, \hat{d}, \hat{i}) \in \hat{\Gamma}, \forall n \in N.$$

The objective function is expressed by function (3.8a) whereas constraint (3.8b) is the flow conservation constraint for the problem. The Problem (3.8) is solved for transmission session $(\hat{s}, \hat{d}, \hat{i})$ by using the same approach used for solving Problem (3.7). An optimal solution to Problem (3.8) is a set of $f_l^{(\hat{s}, \hat{d}, \hat{i})} = 1$, for all $l \in E$. Finally, the optimal and sub-optimal cooperative paths are combined and then transferred to the FIB Reconstruction process in Section 3.2.3.

3.4 Performance Evaluation

3.4.1 Simulation Set-up

We compare under various networking scenarios and diverse parameter settings the performance of our optimal cooperative routing protocol (*OCRP*) to the two following routing schemes.

1. **Shortest path routing (SP)** [7, 11] which is the default routing scheme for CCN.
2. **Cooperative routing scheme (CP)** which was proposed in our previous work [90].

In short, the cooperative routing scheme has two major differences from our optimal cooperative path routing scheme. First, the cooperative path routing protocol does not require DR nor BDR. The prefix group establishment process and the cooperative path calculation are separately conducted by each provider router. Second, the cache contention among the cooperative paths of different prefix groups is not considered, i.e., the cooperative path of each prefix group is calculated at an absence of the cache contention mitigating constraint.

We use ndnSIM [91], which is a NS-3 based network simulator dedicated to named data networking research, to conduct our simulations. The performance of each routing scheme is evaluated by observing the server load and round-trip hop distance.

- **Server load** represents a prospective benefit of using CCN from the content publisher standpoint. It directly reports the volume of traffic that a server must generate in response to its received content requests. At an absence of cache in a network, the volume of server load is proportional to that of aggregated requests from all content requesters. An increasing server load pushes content providers to upgrade their facilities to provide an acceptable quality of service to all content consumers. In addition, if the servers are located outside the network, the server load also implies the volume of inter-domain traffic.
- **Round-trip hop distance** is the sum of hops used by an Interest packet and its corresponding Data packet in a trip of content retrieval. The round-trip hop distance is shorter if a requester can find its desired content object in nearby CCN routers. It implies the waiting time of a content requester and partially estimates the network load.

We evaluate the performance of the OCRP when it is used in mesh network topology. The topologies of backbone networks [92, 93, 94] are often mesh networks to provide path diversity, which improves the robustness and availability of the networks. Because there are various possible transmission paths in a mesh network, the path selection is important and could affect the utilization of network resources. It is worth noting that OCRP may be applicable to other network topologies that also offer path diversity. However, this subject is out of the scope of this experiment.

Each network used in our simulation is randomly generated by using the *igraph* package [95]. Each geographic random topology contains 60 nodes where each of them represents a CCN router. Differently from the other works that assumed a homogeneous content retrieval throughout a network, we model the volume and characteristics of the Interest packet traffic based on the geographical popularity, which was recently studied in [4]. Each generated Interest packet asks for one content object. The size of all content objects is identical. A number of nodes is randomly selected to represent the provider nodes connected to content publishers. We randomly select a number of nodes to be the requester routers that subscribe to a number of prefix groups. Each selected requester router generates random Interest packets asking for the similar content corresponding to its subscribed

prefix group(s).

The profile of the Interest packet traffic is modelled by using the Zipf's distribution which describes the popularity of each content object. Let M denote content catalog cardinality and $1 \leq m \leq M$. The probability of requesting a content object with rank m is $\frac{1}{C \times m^\alpha}$ with $C = \sum_{j=1}^M 1/j^\alpha$, where α is an exponent parameter that shapes the content requests.

Unless otherwise specified, the simulations run with the following settings. Each simulation run begins with all CSs being empty (i.e., cold start). The bandwidth as well as delay of each link is identical whereas lossless transmissions are assumed on all links. The CCN routers in each network all have an identical size of CS that can cache content up to 16 units, where a unit is equal to the size of a content object. As a result, the cost of using each link is the same, so the values of β_1 , β_2 , and β_3 do not impact the results. Each content provider serves 100 uniquely popular content objects. We conduct simulations on 30 networks to obtain average results while the simulation time of each run is equal to 1,000 seconds with 400 second warm-up period. Each requester router requests content objects following the Poisson process whose mean is equal to 50 requests per second. The *TTL* is set to 64 for all messages. The Zipf's exponent parameter (α) is set to one, i.e., $\alpha = 1.0$.

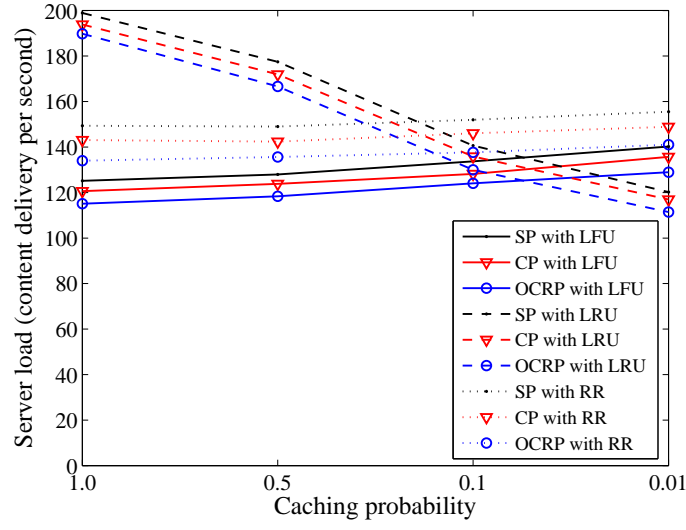
3.4.2 Effects of Caching and Cache Replacement Policies

In this section, we investigate the impacts of the caching and cache replacement policies deployed in the CS of each CCN router. We select two widely used caching policies as follows.

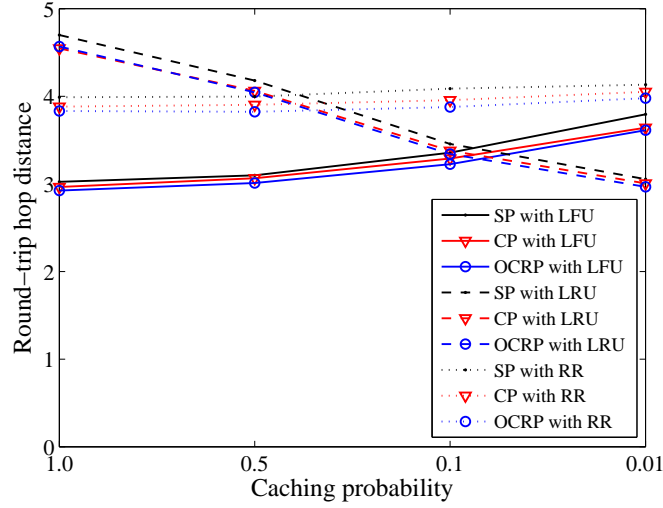
1. *Always* which can be alternatively referred to as a *universal caching scheme* [7].

The caching policy lets each CCN router cache all the content objects traversing it in the CS.

2. *Prob(q)* [14, 40] which requests each CCN router to randomly cache a traversing content objects at a certain probability p . We vary the value of p from 0.01 to 1.0, i.e., $p \in \{0.01, 0.1, 0.5, 1.0\}$. We note that *Always* is a special case of *Prob(1.0)*.



(a) Server load



(b) Round-trip hop distance

Figure 3.6: Impact of the caching and cache replacement policies.

We implement **Least Frequently Used** (LFU), **Least Recently Used** (LRU), and **Random Replacement** (RR) as the cache replacement policy of the CS of CCN router. Three prefix groups are established in each network while each group contains a content provider router and ten requester routers. In addition, the path length constraint is strictly governed by setting $k = 0$.

The average results of each routing scheme when it works with particular caching and cache replacement policies in terms of the server load and round-trip hop distance are shown in Figs 3.6(a) and 3.6(b), respectively. We find that *OCRP* consistently gives the

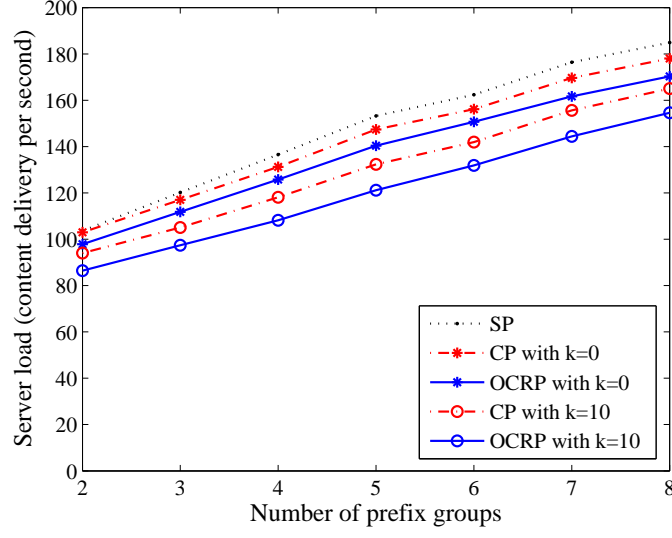
lowest server load compared to *CP* and *SP* despite the differences in the performances of different cache replacement policies. Interestingly, *OCRP* also show its advantage by achieving the lowest round-trip hop distance for all cases. From the results, we draw a conclusion that *OCRP* can improve the server load and round-trip hop distance regardless of cache replacement policies deployed in each CCN router. In addition, we observe that using LRU and *Prob*($p = 0.01$) as the cache replacement and caching policies gives slightly better results than using LFU along with *Always*. It is obvious that LRU and *Prob*($p = 0.01$) are more preferable than the other in practice, considering the complexity and caching cost of these cache replacement and caching policies. Therefore, we hereafter show only the results of the simulations that deploy LRU and *Prob*($p = 0.01$) as the cache replacement and caching policies of each CCN router.

3.4.3 Effects of the Number of Prefix Groups

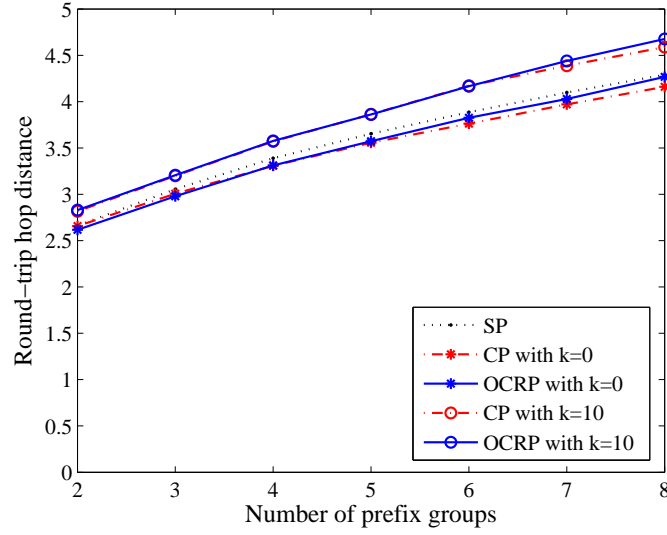
We next investigate the performance of our *OCRP* for various numbers of the prefix groups in each network. The number of prefix groups in each network is varied from two to eight, where each group contains a provider router and ten requester routers. We set both strict path length constraint (i.e., $k = 0$) and its loosely controlled counterpart (i.e., $k = 10$) in our simulations.

The average server load and round-trip hop distance of each routing scheme versus the number of prefix groups in each network are shown in Figs. 3.7(a) and 3.7(b). From the results, we find that both *OCRP* and *CP* all give better server loads than *SP* for all cases. *OCRP* gives better server load than *CP* for all simulated numbers of prefix groups. In addition, *OCRP with $k = 10$* and *CP with $k = 10$* yield better server loads than *OCRP with $k = 0$* and *CP with $k = 0$* , respectively. On one hand, we find that the performance gain in terms of server load provided by *OCRP* becomes steady when the number of prefix groups exceeds five. On the other hand, the gap between the round-trip hop distance of *OCRP with $k = 10$* and that of *SP* still gradually grows when the number of prefix groups surpasses five.

The results come from the fact that loosening the path length constraint (i.e., a large value of k) allows our optimal cooperative path selection to more effectively find the optimal cooperative path by extending the length of feasible paths. The gains are more



(a) Server load



(b) Round-trip hop distance

Figure 3.7: Impact of the number of prefix groups.

significant for large numbers of prefix groups which show the benefits of using *OCRP* in the network with dense and heterogeneous traffic. The inferior reduction of the server load given by *SP* comes from the presence of high cache contention at some CCN routers, which is a result of many irrelevant transmission sessions being transferred in the populated networks. On the contrary, we observe that *OCRP with k = 10* and *CP with k = 10* all yield longer round-trip hop distances than *SP* whereas *OCRP with k = 0* and *CP with k = 0* all give the shorter ones. By using a larger value of k , which allows the paths longer

than the shortest paths to become feasible solutions, the optimal cooperative path may be longer than the shortest path in some cases. This is a vital trade-off between the reductions of the server load and round-trip hop distance. Nevertheless, we observe that the gap between the round-trip hop distance of *OCRP with $k = 10$* and that of *SP* is relatively small compared to the significant improvement in the server load reduction.

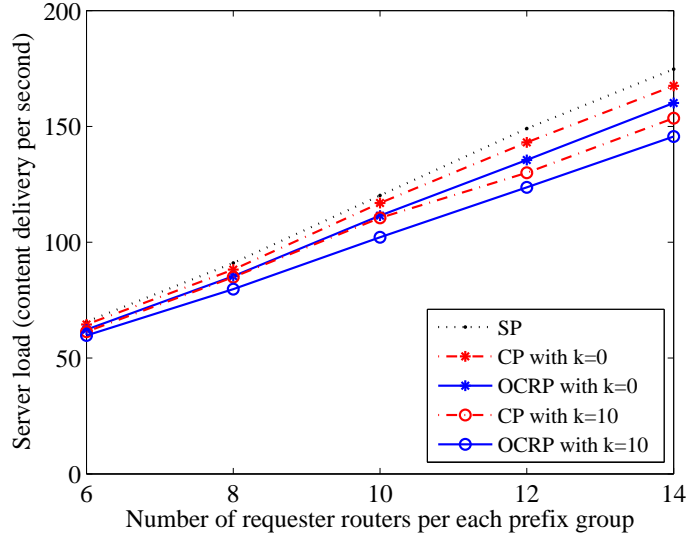
3.4.4 Effects of the Number of Requester Routers in Prefix Group

In this section, we study the impact of the number of requester routers in each prefix group on the performance of the OCRP. We vary the number of requester routers in each prefix group from six to 14 while three prefix groups are available in each network.

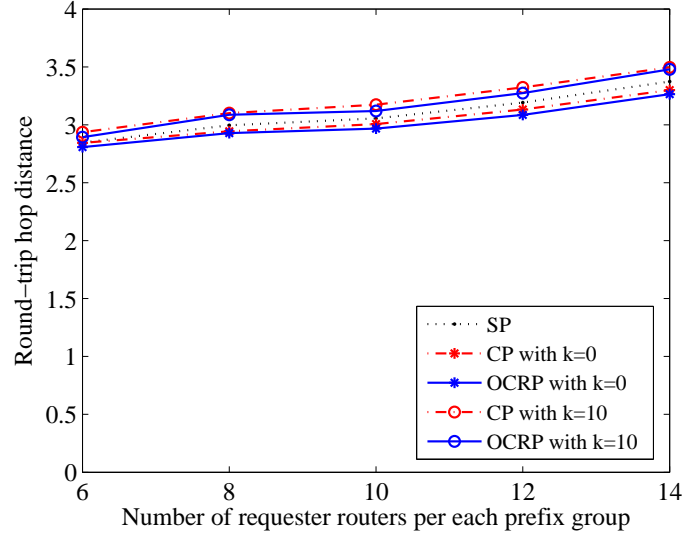
Figures 3.8(a) and 3.8(b) show the average server load and round-trip hop distance of each routing scheme as a function of the number of requester routers in each prefix group. From the results, we observe that *OCRP* and *CP* all yield better server loads than *SP* for all the number of requesters. In addition, *OCRP with $k = 10$* and *OCRP with $k = 0$* give better server loads than *CP with $k = 10$* and *CP with $k = 0$* , respectively. We find that *OCRP with $k = 10$* and *CP with $k = 10$* again give better server loads than their counterparts that operate with $k = 0$. The gains are more significant for large numbers of requester routers in each prefix group. This comes from the fact that more requester routers can possibly join the optimal cooperative path when more requester routers subscribe to the same prefix group. However, we observe a possible limitation of the optimal cooperative routing scheme when round-trip hop distance is a major concern. We find that *OCRP with $k = 10$* and *CP with $k = 10$* all yield slight longer round-trip hop distance than *SP*. On the contrary, the round-trip hop distances provided by *OCRP with $k = 0$* and *CP with $k = 0$* are slightly shorter than that of *SP* for all cases. Nevertheless, the gaps between the round-trip hop distance of *SP* and those of the variants of *OCRP* are relatively small compared with the gaps in terms of their achievable server loads.

3.4.5 Effects of Content Popularity Distribution and the CS Size of CCN Router

The Zipf's exponent parameter (α) and the CS size of each CCN router are varied to investigate the impact of those parameters on the performance of the OCRP. The value



(a) Server load

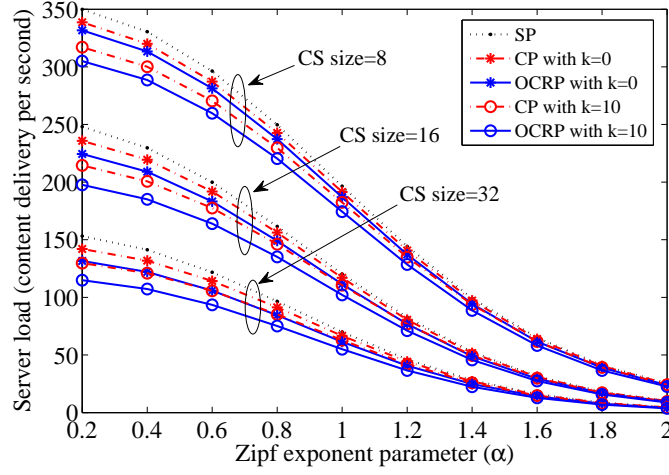


(b) Round-trip hop distance

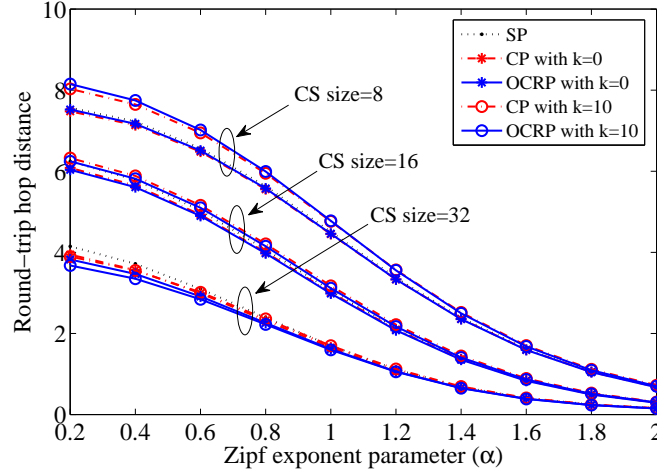
Figure 3.8: Impact of the number of requester routers in each prefix group.

of α is varied from 0.2 to 2.0 while the size of the CS of each CCN router is varied from eight to 32 units. There are three unique prefix groups in the network where each group has one provider router and ten requester routers.

Figures 3.9(a) and 3.9(b) show the average server load and round-trip hop distance of each routing scheme in the network with various CS sizes of CCN routers versus the value α . From the results, we find that the content popularity distribution drastically affects the performance of CCN. We observe that *OCRP* and *CP* all give significant improvement of



(a) Server load



(b) Round-trip hop distance

Figure 3.9: Impact of content popularity distribution.

server load upon *SP*. The server load gains are more significant for the low values of α . To be more specific, we find that *OCRP with $k = 10$* yields more than 20% improvement of the server load upon that given by *SP* for $\alpha = 0.2$ while observing the similar round-trip hop distances between the two when $\text{CS size} \in \{16, 32\}$. In fact, the similar round-trip hop distances provided by the different routing schemes are observable for all values of α . However, we observe that *OCRP with $k = 10$* and *CP with $k = 10$* all yield longer round-trip hop distance than *SP* when $\text{CS size} = 8$.

The results come from the fact that when the value of α is low, the popularity distribution of content objects tends to be uniform. Therefore, the cache-miss rate of the CS of each CCN router increases, so Interest packets in general travel in a longer distance where

the optimal cooperative path shows its benefit. The cache contention in the CS of a CCN router, which depends on the diversity of the content objects to be cached in the CCN router, is proportional to the diversity of the Interest packets traversing it. By using the optimal cooperative path, the Interest packets sent from the requester routers in the same prefix group travel on the same path. Therefore, the associated content objects are cached in the CS of the CCN router on the same path where the subsequent Interest packets can frequently find them. The other prefix groups also adopt the same approach but they all use disjoint paths, so the cache contention in some CSs of CCN routers are accordingly mitigated. In other words, the CCN routers on the optimal cooperative path handle the content objects of fewer prefix groups compared with those on a non-optimized, native shortest path.

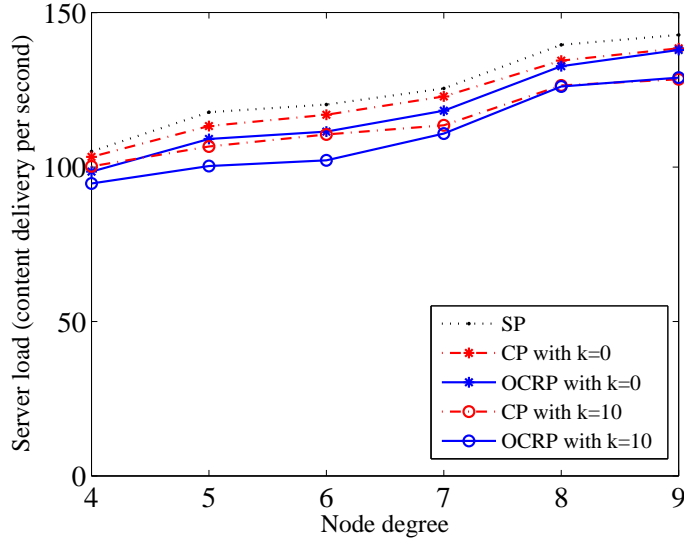
Nevertheless, when the CS size is so small that the miss-rate of the CSs of the CCN routers on the optimal cooperative path is high, many Interest packets may inevitably access their associated provider routers through long transmission paths and accordingly result in longer round-trip hop distances.

3.4.6 Effects of Node Degree

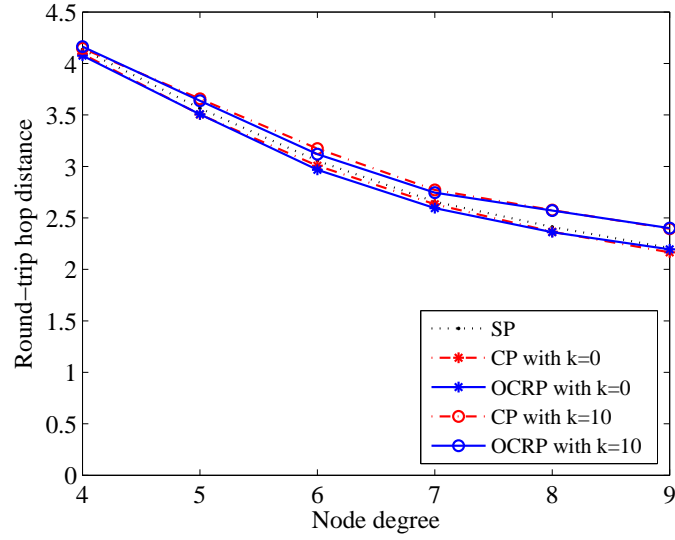
In this section, we compare the performance of the OCRP to those of the other routing schemes for the networks with various node degrees. Three prefix groups are available in each network where each prefix group consists of a provider router and ten associated requester routers. The average node degree of each network is varied from four to ten.

The average server load and round-trip hop distance of each routing scheme are shown in Figs. 3.10(a) and 3.10(b). From the results, we find that the node degree variation results in the different performances of the evaluated routing schemes. We observe that the server loads of all routing schemes increase when the node degree increases. The results can be explained as follows. Given the same number of CCN routers, the number of links increases when the node degree increases. In general, the increase of the number of links results in the shorter path connecting a particular pair of the nodes. When the shorter path is used by a transmission session, the lesser capacity of aggregated cache on this path degrades the on-path caching performance and leads to the increase of the server load.

In addition, we find that *OCRP* and *CP* all give better server loads than *SP* for



(a) Server load



(b) Round-trip hop distance

Figure 3.10: Impact of node degree.

all node degrees. The server load reduction gains are more significant for large node degrees. We observe that *OCRP with $k = 10$* and *CP with $k = 10$* yield better server loads than *OCRP with $k = 0$* and *CP with $k = 0$* , respectively. However, the gap between *OCRP* and *CP* becomes small for large node degrees, for those with $k = 0$ and $k = 10$. This comes from the fact that the number of feasible routing paths also increases when the node degree increases. Consequently, the possibility that the cooperative paths of different prefix groups given by *CP* share the same set of links decreases. As a result,

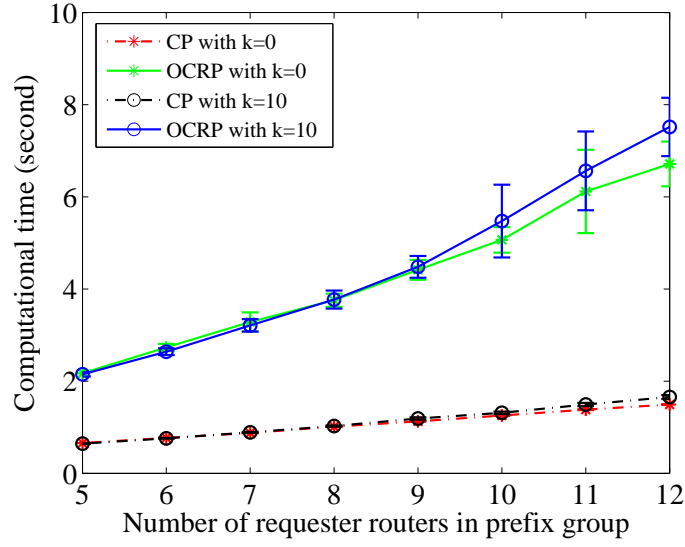
the cooperative paths given by *CP* converge to the optimal cooperative path provided by *OCRP* when the node degree of each network is large.

We find some small differences between the round-trip hop distances given by different routing schemes for all node degrees. To be more specific, *OCRP with $k = 0$* and *CP with $k = 0$* all give slightly better round-trip hop distance than *SP* for all node degrees. On the other hand, *OCRP with $k = 10$* and *CP with $k = 10$* all yield longer round-trip hop distance than *SP*, especially, when the node degrees are large. The gaps between the round-trip hop distances of *SP* and *OCRP* are relatively small compared with those of server loads provided by the considering routing schemes.

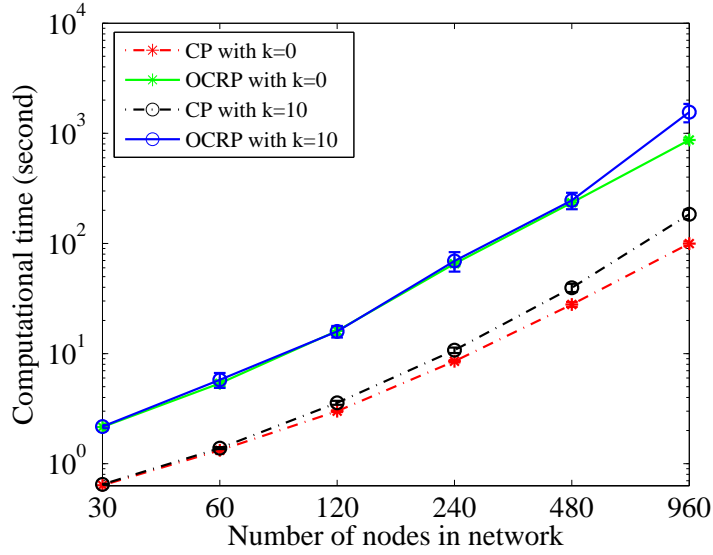
3.4.7 Computational Complexity

In this section we compare the computational time of the optimal cooperative path selection problem of *OCRP* to that of *CP*. The computation is conducted on the computer using a CPU Intel Core i5 M520 clocked at 2.4 GHz in conjunction with 8 GB of RAM. We scrutinize the computational time with two cases: (1) we vary the number of requester routers per prefix group from five to 12 while there are 60 nodes in each network; and (2) the number of nodes in the network is varied from 30 to 960 while there are 10 requester routers per prefix group. In both cases, there are three unique prefix groups in network, where the average node degree is equal to five. The average results are reported with 95% confidence interval.

Figures 3.11(a) and 3.11(b) show the computational time of *OCRP* compared to that of our previously proposed *CP*, versus the number of requester routers per prefix group and the number of nodes in the network, respectively. From the results, we find that *OCRP* takes longer computational time than *CP* in all considered cases. The gap between the computational times of the two becomes more significant when the numbers of requester routers per prefix group and nodes per network increase. A larger value of k results in longer computational time of *OCRP* but marginally affects the computational time of *CP*. This comes from the fact that *OCRP* lets DR alone compute the optimal cooperative path for all prefix groups, whereas *CP* lets the provider router of each prefix group do this task only for the prefix group it belongs to. As a result, the optimal cooperative path selection problem of *OCRP* is more complex than that of *CP* due to the larger number of



(a) Impact of the number of requester routers per prefix group



(b) Impact of the number of nodes per network

Figure 3.11: Computational time for the optimal cooperative path.

variables and constraints. The results show an essential trade-off between an improvement in the in-network caching performance given by OCRP and the complexity of the optimal cooperative path selection problem.

Our OCRP uses the CPU-intensive path selection algorithm, which may raise a scalability issue. However, we can practically handle this issue by adopting a concept of network partitioning, which is a fundamental key of OSPF protocol [6]. For instance, a large network can be partitioned into a number of connected areas shown in Figure3.12.

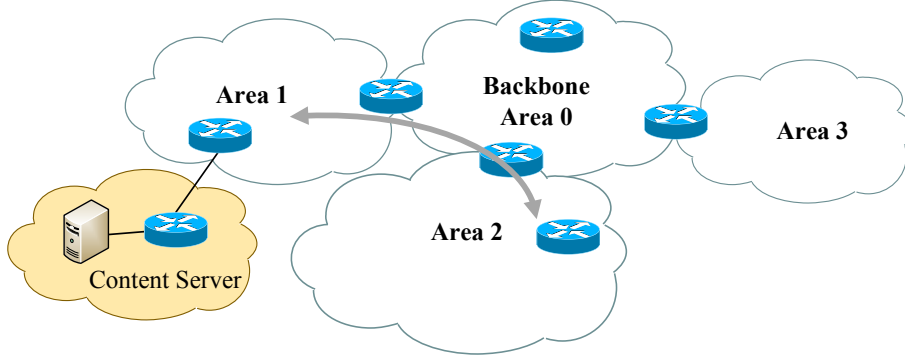


Figure 3.12: Example of multiple areas in an OCRP network.

Each area has its DR which handles the optimal cooperative path selection for the area. As a result, the optimal cooperative path selection problem becomes tractable when we keep the number of routers per area small. Nevertheless, the popularity of a popular prefix in general tends to continue for a long period of time, whereas the topology of a core network does not change frequently. The reasonable computational time of OCRP could be applicable in real networks.

3.5 Conclusion

We propose an optimal cooperative routing protocol (OCRP) for CCN; it employs FIB reconstruction based on the content retrieval statistics to improve the cache utilization of CCN. There are three main components in the proposed routing protocol: (1) Prefix Popularity Observation; (2) Prefix Group (Un)Subscription; and (3) FIB Reconstruction. An integer linear optimization is formulated to calculate the optimal path for the cooperative routing. The constraints of the optimization framework are flow conservation constraint, cache contention mitigating constraint and path length constraint. We compare the performance of our proposed scheme to those of the shortest path routing and our previously proposed cooperative path routing schemes when they are used with different caching and cache replacement configurations in various networks. By using the proposed optimal cooperative routing scheme, the simulation results show an improvement in the server load and round-trip time compared with those of the other routing schemes. The complexity of OCRP is evaluated by measuring the computational time of the optimal cooperative path selection problem.

Chapter 4

Cache Management in Content-Centric Networks

This chapter presents the performance evaluation of various cache management schemes for CCN by means of computer simulation. Each cache management scheme is a combination of a caching scheme and a cache replacement policy. The caching schemes include a universal caching scheme and a probabilistic caching scheme. The cache replacement policies consist of Least Frequently Used (LFU), Least Recently Used (LRU), and Random Replacement (RR) policies. The characteristics of these cache management schemes are concluded at the end of this chapter.

4.1 Introduction

A content-centric network consists of a number of interconnected CCN routers, thus they are a network of caches. A traditional IP network that deploys caching systems, e.g., Web caches and Content Distribution Networks (CDNs), is also a network of caches. However, the content-centric and IP networks are different in terms of their caching granularity. An IP network may have several caches or data centers that are monitored by system administrators. In the case of CDN, the content stored in the data centers are often deliberately selected in advance by content providers [6]. In contrast, the number of CCN routers in a content-centric network, which is seen as the number of independent caches,

can vary from a few to several thousand nodes, depending on where the CCN architecture is deployed. A large number of nodes in a content-centric network as well as the link latency between them could make the cooperative caching and centralized management infeasible or ineffective.

In this chapter, we evaluate the performance of various combinations of a probabilistic caching scheme and different cache replacement policies in search of an efficient cache management scheme for content-centric networks. Extensive computer simulation is organized to study the behavior of the probabilistic caching scheme when it works with different cache replacement policies, i.e., Least Recently Used (LRU), Least Frequency Used (LFU), and Random Replacement (RR). We conduct the computer simulation by using various network topologies and a set of parameter settings. Our simulation results show different yet interesting behavioral characteristics of the probabilistic caching scheme as a function of a cache replacement policy. The probabilistic caching scheme gives the improvement in the server load, round-trip hop distance, and cache hit rate compared with a universal caching scheme only when it works with LRU. The improvement increases as an inverse function of the caching probability assigned to the probabilistic caching scheme. On the contrary, the probabilistic caching scheme does not work well when each router implements LFU. In addition, varying the caching probability of the probabilistic caching scheme does not impact the in-network caching performance when RR is deployed in content-centric networks. Last but not least, the initial state of a network of caches is long for a small caching probability of the probabilistic caching scheme regardless of the deployed cache replacement policy.

The rest of this chapter is organized as follows. Section 4.2 describes the model of cache management schemes used in this study. We include the definition of several caching and cache replacement policies as well as enumerating their important properties in this section. Section 4.3 presents the comparative study of cache management schemes, discussion on simulation results, and conclusive remarks. Finally, we conclude this work in Section 4.4.

4.2 Cache Management Schemes

There are two important algorithms that jointly manage a caching system: a caching scheme and a cache replacement policy. The capacity of a cache is limited by the embedded physical memory whose size is in general smaller than the item population. A *caching scheme* decides whether a caching system stores an item in its cache. An empty cache becomes full as a result of storing such items. A *cache replacement policy* is then needed when a new item enters the cache system and requires some space. A currently cached item is selected to be a victim for an eviction. The victim selection is governed by the cache replacement policy. The objective of a sophisticated cache replacement policy is to keep the items that are likely to be requested in the future by replacing an item that tends to be useless for future requests. A commonly used criterion for evaluating a caching system is its *hit ratio*. The hit ratio of a caching system is the ratio of requests that meet their desired items in the cache to the requests that do not. The caching schemes and cache replacement policies considered in this chapter as well as their properties are described as follows.

4.2.1 Caching Schemes

Routers in content-centric networks should be allowed to operate in a fast and distributed manner to fully exploit the benefit of CCN. Otherwise, the CCN architecture itself may cause a bottleneck in the network. We, therefore, focus on the two most straightforward and commonly known caching schemes, the universal and probabilistic caching schemes, which are considerably practical and scalable to a wide range of network sizes.

Universal Caching Scheme (Always)

The default CCN architecture suggests that a router should exploit a universal caching strategy, which is referred to as *Always* hereafter, as a caching decision policy of each CCN router [7]. A CCN router that deploys *Always* as its caching policy always caches the content object extracted from a valid Data packet. The approach can quickly distribute content in a content-centric network. However, there are evidences pointing out that *Always* can put the replicas of the same content objects in multiple CCN routers and

thus degrades the overall performance of in-network caching, which is indicated by low cache hit rates at intermediate routers [14]. However, the poor performance of *Always* has been confirmed when it is merely used with a particular cache replacement scheme, i.e., Least Recently Used (LRU). Therefore, we further investigate the behavior of *Always* when it is used with other replacement schemes in the next section.

Probabilistic Caching Scheme ($Prob(q)$)

The probabilistic caching scheme, which is referred to as $Prob(q)$ from now, was used as a benchmark scheme in the literature [13, 14, 15, 96]. The key idea is that each CCN router randomly caches a content object that traverses it at a certain caching probability, which is defined by q , where $0 < q < 1$. *Always* is a special case of $Prob(q)$, where $q = 1$. To the best of our knowledge, there has never been an unambiguous criterion that suggests a decent value of q and we first focus on this issue. Interestingly, the performance of $Prob(q)$, where $q < 1$, is better than that of *Always*, which can be inferred from the improvement in server hit rate and hop distance [13, 14, 19]. However, the $Prob(q)$ has been tested only with a cache replacement policy, LRU. Note that $q = 0.1$ is the lowest value of q that has ever been used [13]. We propose that the value of q could be further decreased to improve the caching performance while its limit is constrained by an acceptable duration of the transit state of caching systems.

From observation, the important properties of the $Prob(q)$ can be summarized as follows: 1) Decreasing the caching probability q in $Prob(q)$ reduces the probability that multiple CCN routers on a delivery path cache the same content object in a content delivery; and 2) Decreasing the caching probability q of $Prob(q)$ results in a longer duration of the initial state of caching systems, given a static request pattern.

The first property suggests that a small value of q should be assigned to $Prob(q)$ in order to effectively distribute multiple content objects in a content-centric network and to efficiently utilize the in-network caching ability of CCN. In other words, the diversity of the content objects cached in the network can be improved by decreasing the value of q . However, setting a small value of q may result in a long duration of the initial state of caching systems according to the second property, which leads to a poor performance of capturing a high variation of access patterns.

4.2.2 Cache Replacement Policies

The capacity of a cache is generally smaller than the population of items, so all of such items cannot simultaneously reside in the cache. If the cache is full, the caching system must discard one of currently cached items before it can store a new item. A cache replacement policy determines which item is evicted. We consider three commonly known algorithms: Least Recently Used (LRU), Least Frequency used (LFU), and Random Replacement (RR).

- LRU tries to keep recently active items in the cache by discarding the item that is least-recently-used. LRU is simple to implement and operates fast since its running-time per request is $O(1)$. However, if the capacity of cache is not large enough, LRU poorly performs when items are requested in a round robin fashion. Items will consistently enter and leave the cache without cache hit occurs.
- LFU replaces the least-frequently-used item with a new one. LFU is optimal when the requests received at different times are stochastically independent [97]. However, the running-time per request is logarithmic in the cache size ($O(\log(n))$), where n is the cache size. In addition, it adapts poorly to variable access patterns by accumulating stale items with past high-frequency counts.
- RR is the simplest replacement policy, where one of currently cached items is randomly evicted whenever a replacement is invoked. It does not keep past information of access patterns and thus requires the minimal system requirements to operate. We use RR as a reference for the aforementioned policies, i.e., LRU and LFU.

To the best of our knowledge, most of previous studies used LRU and RR as replacement policies of CSs in content-centric networks when $Prob(q)$ was deployed [14, 15, 19]. We, therefore, evaluate our caching schemes of interest, i.e., *Always* and $Prob(q)$, when they work with LRU, LFU, and RR by means of simulation in the next section.

4.3 Comparative Study of Cache Management Schemes

4.3.1 Simulation Set-up

We use ndnSIM [91], which is a NS-3 based network simulator dedicated to named data networking study, to conduct our simulations. All basic structures of CCN, FIB, PIT, and CS, are reproduced by ndnSIM. For the time being, ndnSIM models the routing mechanism of CCN which is driven by exchanging an interest packet and a Data packet. However, it does not allow variable size of content object, so we ignore the content segmentation in our simulations and assume an identical size of content objects.

We assign various values of the caching probability q , where $q \in \{1.0, 0.7, 0.3, 0.01\}$. As a result, our simulations take into account *Always*, *Prob(0.7)*, *Prob(0.3)*, and *Prob(0.01)*. We use LRU, LFU, and RR as a replacement policy of the CS of each CCN router. The CS size of each node is varied from 1% to 10% of the content population. We use the Dijkstra's algorithm to calculate the shortest path (in terms of the number of hops) to reach every content provider. Then we translate the calculated path to the FIB of each node.

The profile of content requests is modelled by using the Zipf's distribution which describes the popularity of each content object. Let M denote content catalog cardinality and $1 \leq i \leq M$. The probability of requesting a content with rank i is $\frac{1}{C \times i^\alpha}$ with $C = \sum_{j=1}^M 1/j^\alpha$, where α is an exponent parameter that shapes the content requests. A decent value of α has been being ambiguous in recent studies focusing on the caching performance of content-centric networks [40, 15, 98]. The value of α directly depends on the types of particular contents [98]. Equally important, different values of α can be derived from different geographical locations [4]. We, therefore, use $\alpha = 1.0$ as it was used in [40]. From our simulation results, the order of the performance among the caching and cache replacement policies are unchanged for $0.4 \leq \alpha \leq 1.6$, so the results presented in this chapter are valid for these values of α .

Each simulation run begins with all CSs being empty (i.e., cold start). Unless otherwise specified, the simulations run with the following parameters. The total simulation time is equal to 10,000 seconds with 4,000 second warm-up period. Each content requester requests content objects following the Poisson process whose mean is equal to 50 requests/s.

Each content provider serves 1,000 different content objects. The uniform size of CS is varied from 1%, 2%, 5%, and 10% of the total content population. The results are reported at 95% confidence interval.

4.3.2 Evaluation Metrics

We evaluate the performance of each caching scheme when it works with the particular cache replacement policy by observing four metrics: the server load, round-trip hop distance, hit rate, and instantaneous behavior.

Server load represents a prospective benefit of using CCN from the content provider standpoint. It directly reports the volume of traffic that a server must generate in response to its received requests. At an absence of cache in a network, the server load is equal to aggregated requests from all content requesters. An increasing server load pushes content providers to upgrade their facilities to provide an acceptable quality of service to all content requesters.

Round-trip hop distance is the sum of hops used by an interest packet and corresponding Data packet in a trip of content retrieval. The round-trip hop distance is shorter if a requester can find its desired content object in nearby CCN routers. It also implies the **access latency** and partially estimates the **network load**.

Cache hit rate is commonly used to evaluate caching system. A high hit rate of a caching system implies its good performance. The hit rate of a CCN router is the probability that a content request finds its desired content objects in the CS of a CCN router. However, the hit rates of different routers may differently contribute to the overall performance of a content-centric network.

Instantaneous behavior is observed to illustrate how fast or slow a caching scheme can adapt caching systems to an access pattern. We can observe the duration of the initial state of a network of caches from the instantaneous behaviors of server load, round-trip hop distance, or hit rate. For brevity the dissertation, we show only the instantaneous behavior of the server load in this chapter.

4.3.3 Network Topologies

We conduct our simulations on two network topologies, which are described as follows.

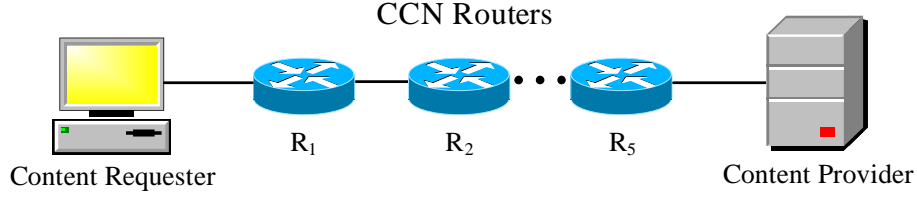


Figure 4.1: Cascading network used in our simulations.

Cascading Network

We use a fixed length cascading network in our simulations. A cascading network contains five CCN routers as shown in Figure 4.1. We consider two study cases of using the cascading network in order to cover its practical use.

- **One content requester:** The first study case is that request traffic accesses the cascading network at one CCN router. A content requester is connected to the CCN router at one end of network (R_1), whereas a corresponding content provider is connected to the CCN router at the other end (R_5).
- **Multiple content requesters:** The second study case considers that requests enter the network through multiple CCN routers. More specifically, four content requesters are connected to routers R_1 , R_2 , R_3 , and R_4 . These content requesters request content objects from the content provider that is connected to the end of the network (R_5).

SINET4: A Real Internet Topology

Another network topology of interest is based on the Science Information NETwork 4 (SINET4) [92]. SINET4 is providing a unified network connection to 700 universities and research institutes in Japan. The network topology of SINET4 is shown in Figure 4.2. SINET4 is constituted of eight core nodes and 42 edge nodes that are geographically distributed around Japan. Some routers are connected to foreign academic networks and commercial Internet service providers to link SINET4 to the globe. SINET4 forms a hybrid topology that consists of a mesh network and a number of star topology networks. The eight core nodes constitute a mesh network, whereas each core node is connected to a number of edge nodes to form a star topology network. We conduct our simulations

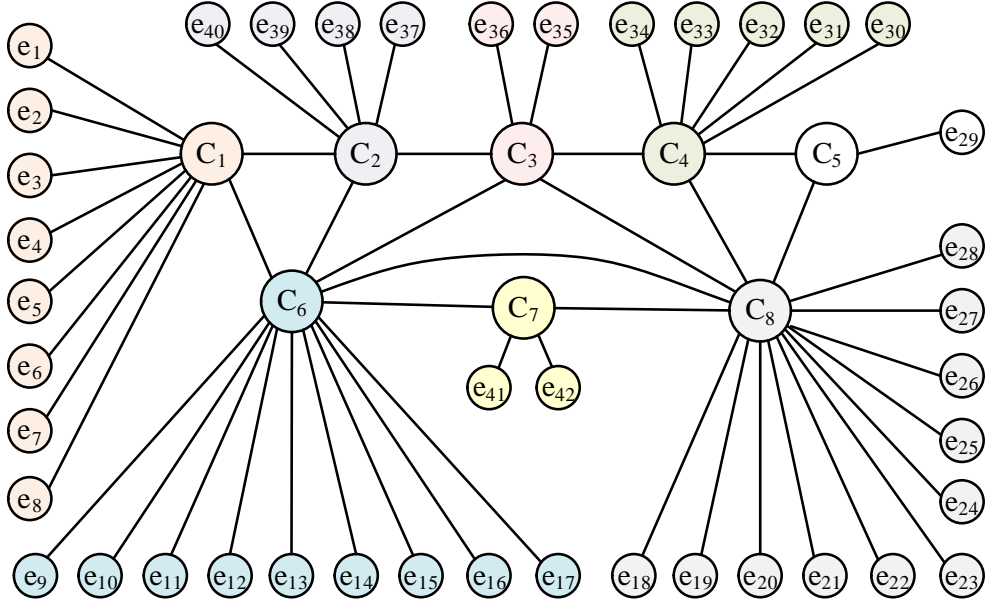


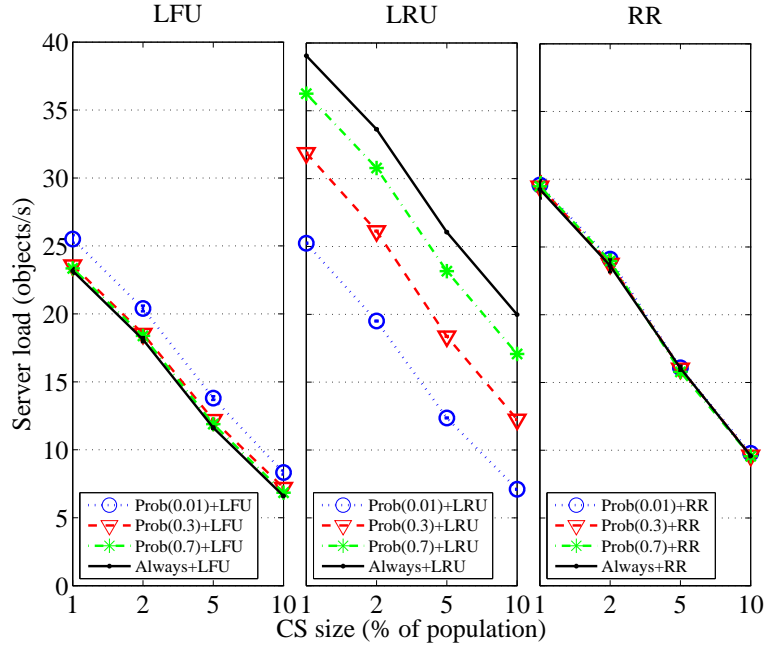
Figure 4.2: The network topology of SINET4.

using the network topology of SINET4 as follows. All nodes in the network represents the identical CCN routers. We link a unique content provider to each core node. Each content provider provides 1,000 unique content objects so that 8,000 content objects are available in total. A content requester is connected to each edge node so that there are 42 content requesters in the network. Each content requester equally requests content objects from all content providers.

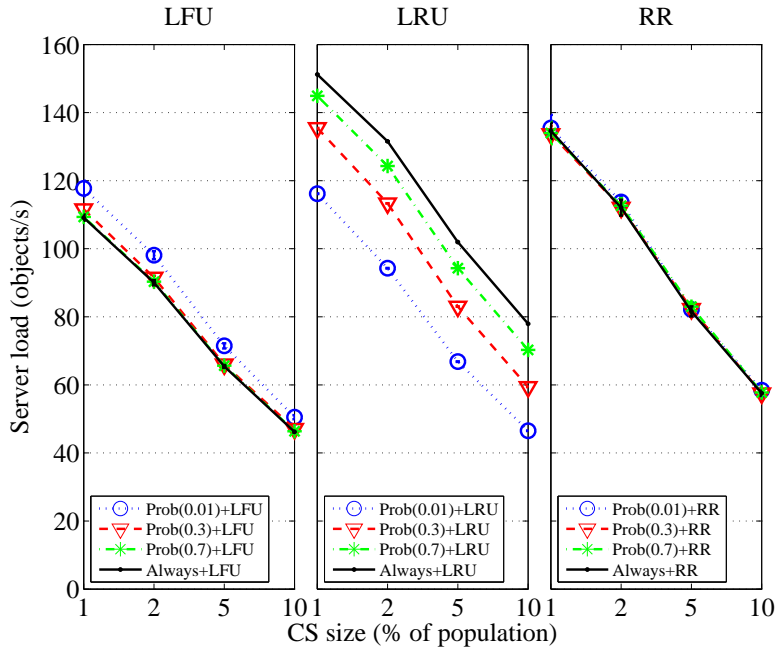
4.3.4 Results and Discussions

Experiments with Cascading Network

We report the average server load of the content provider for each caching scheme that works with different cache replacement policies in Figure 4.3. The results for the first and the second study cases of a cascading network are shown in Figs 4.3 (a) and (b), respectively. We find that the performance of server load reduction varies as functions of the caching scheme, cache replacement policy, and the size of CS. We observe that the server load decreases as a function of CS size for all cases. Surprisingly, each replacement policy shows its unique behavior given the different caching schemes. $Prob(q) + LRU$ gradually reduces the server load when the value of q is decreased. Specifically, $Prob(0.01) + LRU$ yields 20% improvement in the server load reduction over $Always + LRU$. The improve-



(a) One content requester



(b) Multiple content requesters

Figure 4.3: The server load for different CS sizes in the cascading network.

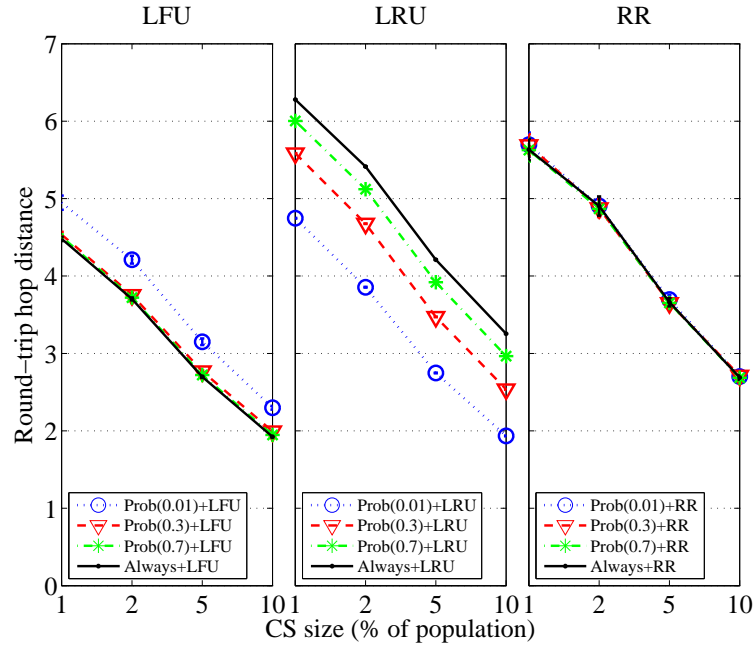
ment comes from the reduced replicas of the same content in multiple routers. These results also apply to the second study case.

In contrast, we obtain the opposite results for $Prob(q) + LFU$ and $Always + LFU$.

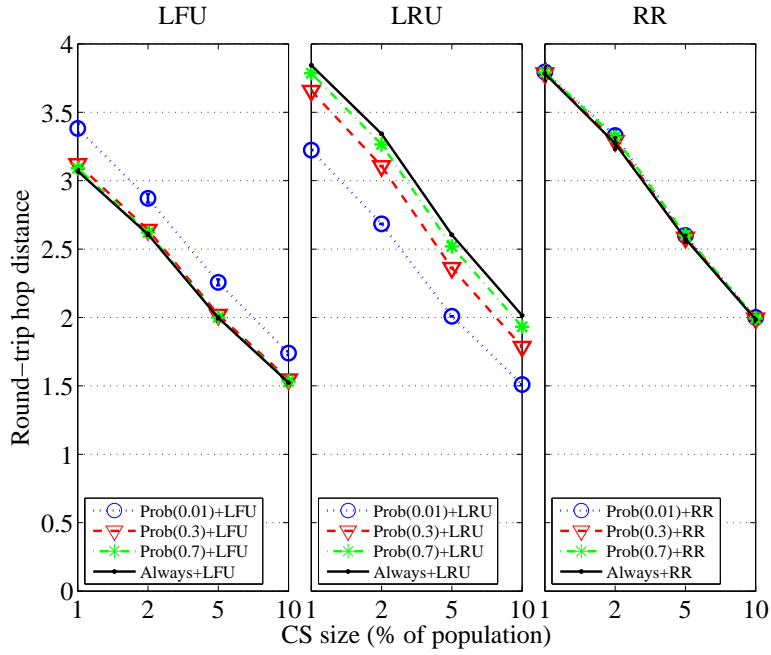
$Prob(q) + LFU$ increases the server load when the value of q decreases. This is due to the poor performance of LFU against variable access patterns. Specifically, each CCN router may accumulate stale items with past high-frequency counts. For instance, a popular content object may be cached by a CCN router at the initial state of the caching system as a result of $Prob(q)$, where $0 < q < 1$. The cached content object then starts to accumulate the reference frequency and continues to reside in the CCN router if it is popular at that time. The accumulated reference frequency of the content object is a nondecreasing function as long as the content object can continuously stay in the CCN router. When the cache enters its steady state, this content object may become less popular than other content objects from the router's standpoint. This is because the interest packets corresponding to this content object are already satisfied by another router nearby the content requester. However, the accumulated reference frequency of the content object may be higher than that of currently popular content objects. As a result, the stale content object pollutes the cache when LFU is used. In addition, $Prob(q)$ intensifies the issue when the value of q is smaller, since the stale content object may stay in the cache for a longer period. This comes from the fact that the CCN router randomly caches fewer incoming content objects due to a lower caching probability.

We find that $Prob(q) + RR$ does not yield any significant change to the results in terms of the server load reduction, although we alter the values of q . The results for $Always + RR$ and $Prob(q) + RR$ are almost identical. It is because RR naturally distributes content objects in the network regardless of the caching scheme used in our simulations. The above statement applies to the results of the second case study (i.e., content requests enter the network through multiple routers) as shown in Figure 4.3. It is worth to note that the performance of $Always + LRU$ regarding the server load reduction is inferior to that of $Always + RR$ in both study cases. However, we find that the gap between the performance capabilities of different caching schemes and cache replacement policies for the second study case is smaller than that of the first case.

In essence, $Always + LFU$, gives the best performance as a reward for its complexity. On the contrary, $Always + LRU$ performs the worst in all cases. The performance of $Prob(q)$ obviously depends on the value of q when it works with LRU. This is because LRU tends to leave the same content object in neighbor nodes, and $Prob(q)$ with a smaller



(a) One content requester

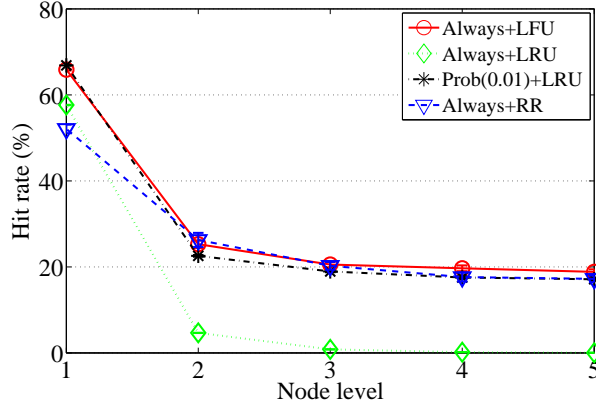


(b) Multiple content requesters

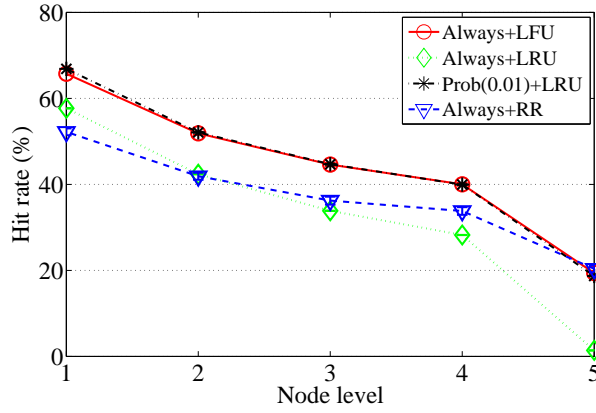
Figure 4.4: The round-trip hop distance for different CS sizes in the cascading network.

value of q prevents this from happening.

The results in terms of round-trip hop distance are shown in Figs. 4.4 (a) and (b) for the first and second study cases, respectively. The order of performance among the



(a) One content requester



(b) Multiple content requesters

Figure 4.5: Hit rate with respect to the node level in the cascading network.

caching and cache replacement policies in terms of round-trip hop distance is similar to that in terms of server load. On the other hand, the performance of *Always + LRU* in terms of round-trip hop distance is significantly inferior to that of *Always + RR* in the first study case, whereas it is almost identical to that of *Always + RR* in the second case. The results come from the fact that the performance of *Always + LRU* highly depends on where the requests enter the cascading network, which can be observed from the results in terms of cache hit rate as follows.

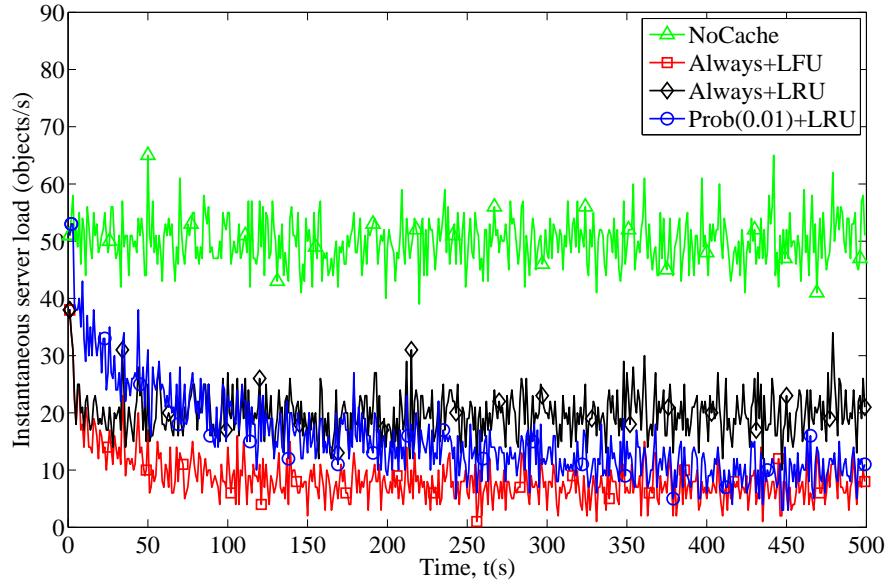
We measure the cache hit rate of CCN routers towards the node levels and report them in Figs 4.5 (a) and (b) for the first and second study cases, respectively. We show a comparison of *Always+LFU*, *Always+LRU*, *Prob(0.01)+LRU*, and *Always+RR* when the CS size of each router is equal to 10% of population due to limited space. For the first study case where the requests access the network at the node level 1, *Always + LFU* gives

the best hit rates for all node levels in comparison to the other schemes. Interestingly, $Prob(0.01) + LRU$ remarkably overcomes $Always + LRU$ for all node levels. We find that $Always + LRU$ gives a high hit rate for the node level 1 whereas the other nodes all suffer limited hit rates. In fact, $Always + RR$ achieves higher hit rate than $Always + LRU$ for every node level except the node level 1. For the second study case, the requests enter the network through multiple routers, so these routers become the first hop routers of some requests. $Always + LRU$, in essence, performs well for the first hop router, so its performance for the second case is better than that for the first case. The hit rates of all node levels for $Always + LFU$ and $Prob(0.01) + LRU$ are almost identical. $Prob(0.01) + LRU$ is easier to implement than $Always + LFU$, considering their running-time per request and caching-cost per transmission. In other words, although $Prob(0.01)$ lets each router cache fewer content objects than $Always$ in a transmission, it still gives the comparable performance.

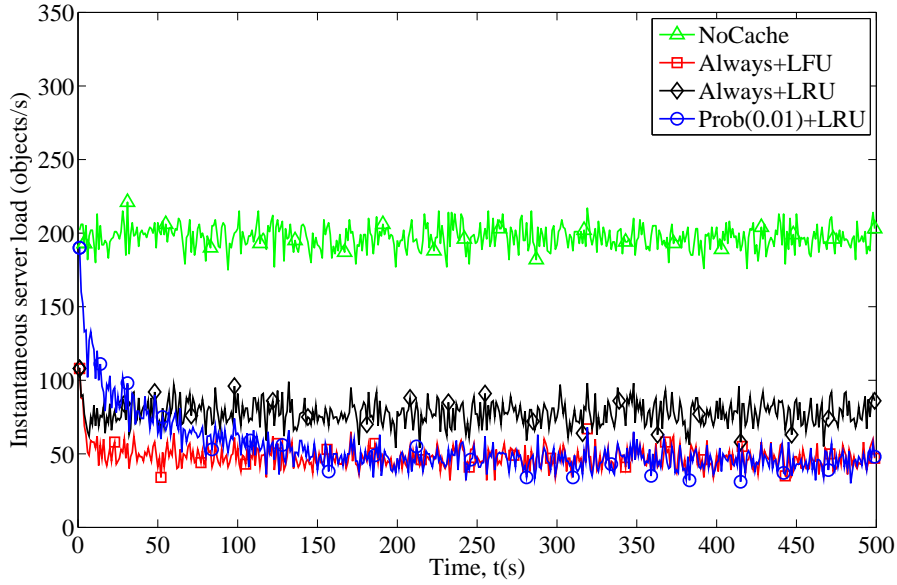
We next show in Figs. 4.6 (a) and (b) the instantaneous behaviors of the different caching schemes and cache replacement policies in comparison to the network without cache (*NoCache*). We show only the tracks of the server load for $Always + LFU$, $Always + LRU$, and $Prob(0.01) + LRU$ when the CS size is equal to 10% of the population, since the order of the server loads provided by different cache management schemes is unchanged for all considered CS sizes. We do not show the results for $Prob(0.01) + LFU$ since they are almost indistinguishable from that of $Prob(0.01) + LRU$. We find that both $Always + LFU$ and $Prob(0.01) + LRU$ perform better than $Always + LRU$ in their steady states for both study cases. However, we observe that $Prob(0.01) + LRU$ takes much longer duration of the initial state than the others. The long initial state of a network of caches caused by $Prob(0.01) + LRU$ may not be suitable to a request pattern with a high variation.

Experiments with Real Internet Topology

We report the results for SINET4 topology in terms of the server load reduction, round-trip hop distance, hit rate, and instantaneous behavior in Figs. 4.7, 4.8, 4.9, and 4.10, respectively. Figure 4.7 shows the average server load for all content providers that are linked to the network. The results partly resemble that of the cascading network in the previous section. $Prob(q) + LRU$ significantly improves the server load reduction when the



(a) One content requester



(b) Multiple content requesters

Figure 4.6: Instantaneous behavior of different caching schemes and replacement policies in the cascading network.

value of q is decreased. The improvement is a result of the probabilistic caching scheme that minimizes the probability that multiple CCN routers cache the same content object. In addition, $Prob(q)$ alleviates the downside of LRU that occurs when request patterns follow a round-robin fashion. $Prob(q)$ allows a cached content object to longer stay in a CCN router until the router caches a new content object. In contrast, we observe that

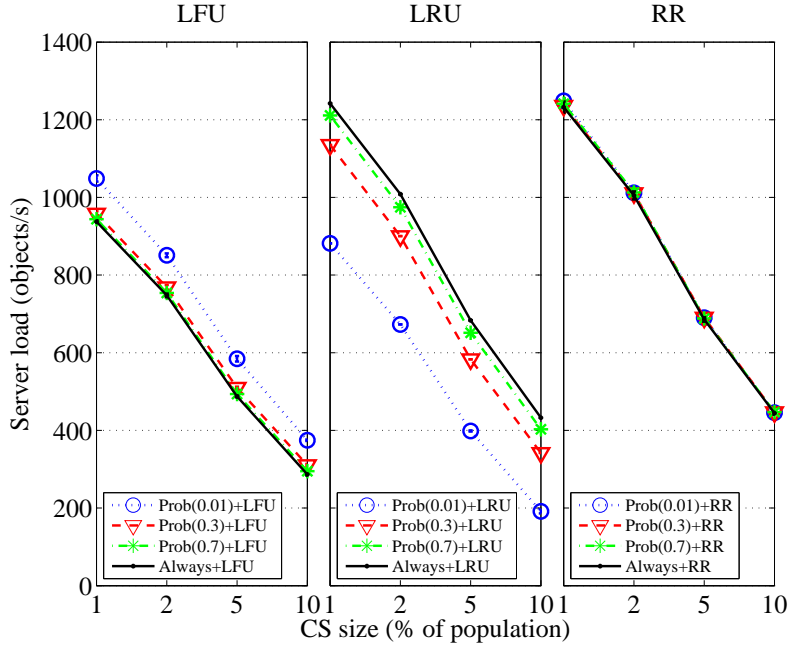


Figure 4.7: The server load for different CS sizes in SINET4.

$Prob(q) + LFU$ performs worse than $Always + LFU$ when $q \in \{0.01, 0.3, 0.7\}$ which is reflected by the increased server load. Thus, we infer from the results that *Always* is the best caching scheme given the LFU as a replacement policy. The results show the incompatibility of using $Prob(q)$ with LFU and reiterate the importance of matching the particular caching scheme to a cache replacement policy. Interestingly, $Prob(0.01) + LRU$ even overcomes $Always + LFU$ in the SINET4 topology. It is because LFU suffers from accumulating the content objects with past high-frequency counts whereas LRU can benefit from the content objects distributed in the network. As we expect, $Always + RR$ and the variants of $Prob(q) + RR$ all give indistinguishable results.

We report in Figure 4.8 the round-trip hop distance for different caching schemes and replacement policies at various sizes of CS. The results in terms of the round-trip hop distance follow the trend of server load for all cases. We observe that $Prob(0.01) + LRU$ even gives a better round-trip hop distance than that of $Always + LFU$ for a few percentages. RR consistently yields the worst round-trip hop distance among those of the other replacement policies regardless of the exploited caching policies.

We next show in Figure 4.9 the average hit rate of CCN routers whose CS sizes are equal to 10% of the content population as a function of the node levels, i.e., core and edge

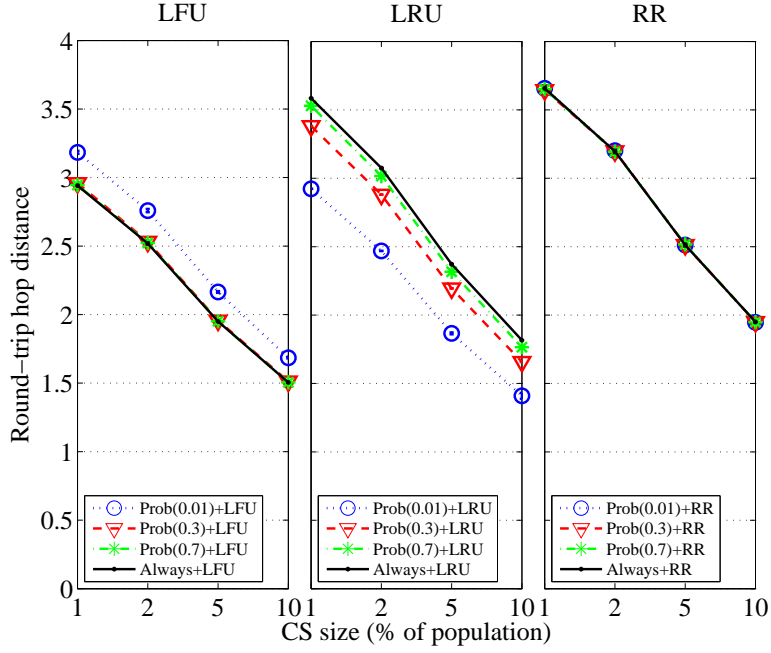


Figure 4.8: The round-trip hop distance for different CS sizes in SINET4.

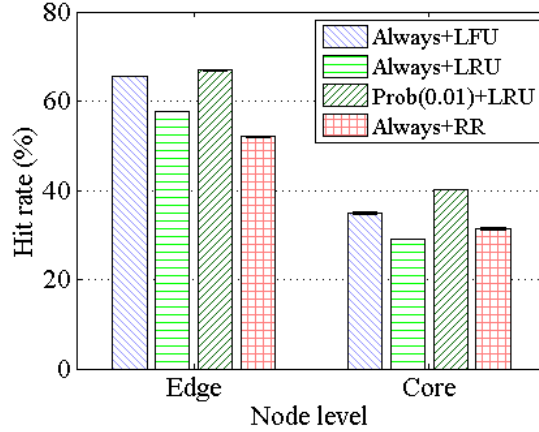


Figure 4.9: Hit rate towards the node level in SINET4.

nodes. We find that $Prob(0.01) + LRU$ gives better hit rates than $Always + LFU$ and the other schemes. $Prob(0.01) + LRU$ significantly improves the hit rates of both core and edge nodes (up to 10%) over that of $Always + LRU$.

Figure 4.10 shows the instantaneous behaviours of $Always$ and $Prob(0.01)$ when they work with LFU, LRU, and RR in the SINET4 topology where the CS size of each router is equal to 10% of content population. The results show that the initial state of the caching system for $Prob(0.01) + LRU$ is longer than those for $Always + LFU$, $Always + LRU$, and

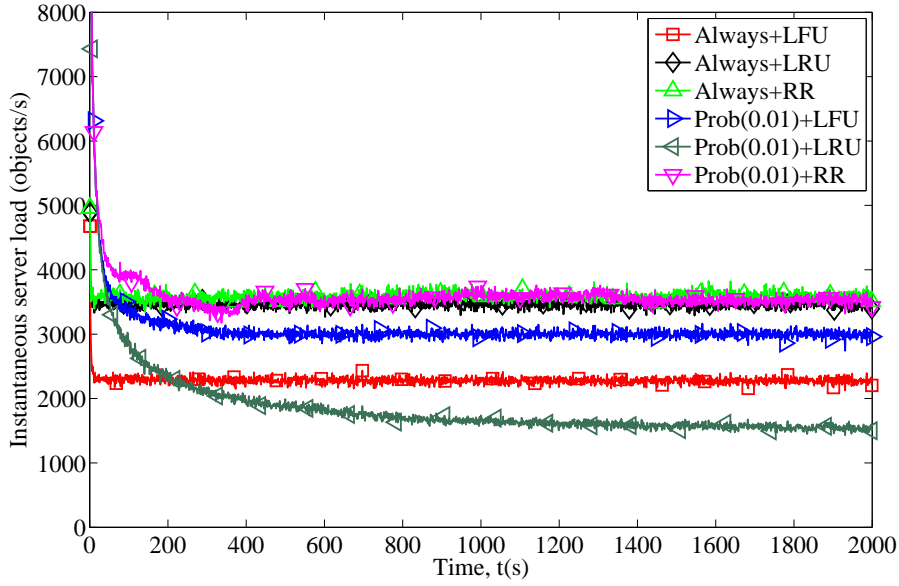


Figure 4.10: Instantaneous behavior of different caching schemes and replacement policies in SINET4.

Always + RR. The instantaneous behaviors of $Prob(0.01) + LFU$ and $Prob(0.01) + RR$ also follow that of $Prob(0.01) + LRU$. Nevertheless, $Prob(0.01) + RR$ results in a shorter duration of the initial state of caching systems than $Prob(0.01) + LFU$ and $Prob(0.01) + LRU$. However, it gives the poorest caching performance in the steady state of the caching systems compared with the others.

Conclusive Remarks

The behavior of a probabilistic caching scheme explicitly varies as a function of a cache replacement policy. The probabilistic caching scheme gives the improvement in the server load, round-trip hop distance, and cache hit rate compared with a universal caching scheme only when it works with LRU. The improvement increases as an inverse function of the caching probability assigned to the probabilistic caching scheme. When LFU is deployed in a content-centric network, a universal caching scheme is a policy of choice since it gives a better performance than the probabilistic caching scheme. On the contrary, the probabilistic caching scheme even magnifies the issue of LFU by letting CCN routers accumulate stale content objects with past high-frequency counts. The probabilistic and universal caching schemes have an identical behavior when RR is deployed in content-centric networks. The initial state of a network of caches is longer when the caching

probability of a probabilistic caching scheme is decreased regardless of the deployed cache replacement policy.

4.4 Conclusion

We study the behavioral characteristics of a probabilistic caching scheme by means of computer simulation. We evaluate the probabilistic caching scheme when it works with different cache replacement policies. The evaluation metrics consist of the server load, round-trip hop distance, cache hit rate, and instantaneous behaviour. The simulation results show that the behavior of a probabilistic caching scheme explicitly varies as a function of a cache replacement policy. The performance of probabilistic caching scheme and the duration of the initial state of a network of caches are inverse functions of a caching probability. The probabilistic caching scheme works well only in the caching system that implements Least Recently Used (LRU) as a cache replacement policy, whereas the limit of its performance comes from the increased duration of the initial state of the caching system.

Chapter 5

Performance Analysis of Probabilistic Caching Scheme

This chapter presents a new analytical model for evaluating the performance of a probabilistic caching scheme with various cache replacement policies. The considering cache replacement policies consist of Random Replacement (RR), First In First Out (FIFO), and Least Recently Used (LRU). Several important properties of the probabilistic caching scheme are stated in this chapter.

5.1 Introduction

The behavior of a probabilistic caching scheme in content-centric networks has been studied by using computer simulations [14, 15, 99, 100]. However, not many works in literature investigate its logical behavior by means of mathematical analysis. The deep understanding of a probabilistic caching scheme in content-centric networks has not been fully gained. To address this issue, Martina et al. [50] developed an analytical model to study a probabilistic caching scheme with Least Recently Used (LRU) policy. Their model is based on the Che's approximation [48], which estimates a cache miss rate by assuming exponential sojourn times of items in a cache. Other analytical approaches to the same objective are found in [51, 44]. Nevertheless, these previous works did not address the results when a probabilistic caching scheme is used with other cache replacement policies such as Random Replacement (RR) and First In First Out (FIFO). In the context of content-centric

networks, RR and FIFO policies might be a better cache replacement policy than LRU policy to be used by CCN routers, since they are simpler to implement [98, 50].

In this chapter, we develop a new analytical approach to behavioral analysis of a probabilistic caching scheme. An independent factor of this analysis is the cache replacement policies that a probabilistic caching scheme is used with. The cache replacement policies consist of RR, FIFO, and LRU policies. Our model is based on Markov chains under Independent Reference Model (IRM) and Zero Download Delay (ZDD) assumption. By using this model, we can establish several important properties of the probabilistic caching scheme: 1) The hit rate of a discrete cache whose cache replacement policy is either RR or FIFO is independent of a caching probability; 2) A caching probability impacts the hit rate of a cache hierarchy whose cache replacement policy is either RR, FIFO, or LRU; and 3) The hit rate of a cache hierarchy with RR bounds that with FIFO for any caching probability. To the best of our knowledge, our model is the first analytical work to show that the performance of a probabilistic caching scheme is readily affected by a companion cache replacement policy. The findings become an essential part in the guidelines for effectively using a probabilistic caching scheme in content-centric networks.

The rest of this chapter is organized as follows. Section 5.2 discusses the caching system model and assumptions used by the analysis. The definitions of a probabilistic caching scheme and cache replacement policies are also addressed in this section. Sections 5.3, 5.4, and 5.5 model the behavior of a probabilistic caching system with RR, FIFO, and LRU policies by using Markov chains, respectively. The validation of the proposed analytical model with computer simulation is provided in Section 5.6. Finally, conclusion remarks are given in Section 5.7.

5.2 System Model and Assumptions

We characterize a cache management scheme of a caching system by a **caching scheme** and a **cache replacement policy**. A **caching scheme** decides whether a caching system stores an item in its cache. When a cache is in a full state, a **cache replacement policy** decides which currently cached item is discarded before a new item enters the cache. Hit rate (or hit ratio) is commonly used to evaluate a caching system. It indicates a likelihood

that a caching system can provide the requested items to requesters. The higher a hit rate, the higher performance of a caching system.

5.2.1 Caching Scheme and Cache Replacement Policies

A caching system in each CCN router must keep up with the packet forwarding speed to fully exploit the benefit of CCN. Otherwise, the in-network caches may become a bottleneck in the network. This requirement constrains the caching scheme and cache replacement policy of a CCN router to be noncomplex. We therefore focus on a probabilistic caching scheme ($Prob(q)$) which is simple and scalable to a wide range of network sizes. $Prob(q)$ lets each CCN router randomly cache a fraction of content objects carried by arriving Data packets. A caching event happens at caching probability q , where $0 < q < 1$. $Prob(q)$ is simple to implement compared to other existing caching schemes. Unlike the popularity-based caching schemes [13, 19] and collaborative caching schemes [20, 21], $Prob(q)$ requires neither the past access information nor a synchronization among routers, and thus becomes a promising candidate of a cache management scheme for content-centric networks.

We consider three simple and fast cache replacement policies in our analysis: Random Replacement (RR), First In First Out (FIFO), and Least Recently Used (LRU).

- RR is the simplest replacement policy, where one of the items currently in the cache is randomly evicted in the uniform distribution whenever a replacement is invoked. While RR has been rarely used in practice, its simplicity would be beneficial for the implementation of fast CCN routers.
- FIFO replaces the cached item that was first to enter a cache among the items currently in the cache. It has been widely used in data buffer of conventional routers.
- LRU tries to keep recently active items in a cache by discarding the least-recently-used item. It has been used by various caching systems such as RAM, web browser and web proxies, and databases. We consider LRU as an upper bound for the complexity of the policies that can be implemented at line speed.

5.2.2 Request Traffic Model

We assume that the requests of items follow the Independent Reference Model (IRM) [99, 47], where a request of item a occurs with probability $p(a)$ and is independent of previous requests for all items. Although the requests from an individual may exhibit a strong temporal locality, the IRM is a reasonable qualitative model in representing accesses to the router which provides services to a large number of users. We also assume Zero Download Delay (ZDD) [47]. Under a ZDD assumption, when a cache miss of an item request occurs at a cache, the requested item is assumed to be instantly downloaded to the cache. In practice, this assumption is possible if every request has been satisfied before the next one is generated.

5.2.3 Ergodicity of Markov Chains

We utilize Markov chains to model the behavior of a caching system. One of the properties of Markov chains can be stated as follows.

Definition 1. [47] *An **ergodic set** of a Markov chain is a set of the states that every state can be reached from every other state through one transition or more, and that cannot be left once it is entered.*

A Markov chain is quasi-ergodic if its state space consists of an ergodic set. Otherwise, the Markov chain is non-ergodic. The steady-state of a quasi-ergodic Markov chain does not depend on its initial-state but that of a non-ergodic one does. The ergodicity of the Markov chain of a caching system is subject to its replacement policy. Specifically, both RR and LRU characteristics correspond to quasi-ergodic Markov chains, whereas FIFO yields a non-ergodic one [47]. We will adopt the characteristics of cache replacement policies to establish several important properties of $Prob(q)$.

The notations used in this chapter are summarized by Table 5.1.

Table 5.1: Summary of notations in our analytical model for cache analysis

$p(a)$: probability of requesting item a
q	: caching probability
X	: set of items available for download
C	: caching capacity of a CCN router
$\pi(\star)$: stationary distribution of state \star in a Markov chain
Caching systems with RR	
Γ	: state space of the Markov chain of a discrete cache with RR
A	: state of a discrete cache with RR, $A \in \Gamma$
$S_i(A)$: set of states that have one item different from A
$\gamma(a)$: set of states that the cache currently stores item a , $\gamma(a) \subset \Gamma$
R_{RR}	: hit rate of a discrete cache with RR
R_i	: the i^{th} hop router in the cache hierarchy, $i \in \{1, 2\}$
$A_{(i)}$: state of the cache of router R_i , $A_{(i)} \in \Gamma$
Δ	: state space of the cache hierarchy with RR
B	: state of the cache hierarchy with RR, $B = [A_{(1)}, A_{(2)}]$, $B \in \Delta$
R_{RR}^H	: hit rate of the cache hierarchy with RR
Caching systems with either FIFO or LRU	
Φ	: state space of the Markov chain of a discrete cache
\mathbf{A}_k	: array of C distinct items representing a state of a discrete cache
$\mathbf{a}_i^{(k)}$: item at the i^{th} position of a cache in state \mathbf{A}_k
$M(\mathbf{A}_k)$: set of items when a cache is in state \mathbf{A}_k
$\mathbf{S}'(\mathbf{A}_k)$: set of states where: <ul style="list-style-type: none"> • item $\mathbf{a}_1^{(k)}$ is not in the cache • items $\mathbf{a}_2^{(k)}, \dots, \mathbf{a}_C^{(k)}$ are at the $1^{st}, \dots, (C-1)^{th}$ positions of the cache • item e is at the C^{th} position of the cache, where $e \in X \setminus M(\mathbf{A}_k)$
$\beta(a)$: set of states when the cache currently stores item a
R_{FIFO}	: hit rate of a discrete cache with FIFO
P_A	: set of permutations of A
$\mathbf{A}_{(i)}$: cache state of router R_i in the cache hierarchy for $i \in \{1, 2\}$
Λ	: state space of the Markov chain of the cache hierarchy
\mathbf{B}	: state of the cache hierarchy, $\mathbf{B} = [\mathbf{A}_{(1)}, \mathbf{A}_{(2)}]$, $\mathbf{B} \in \Lambda$
R_{FIFO}^H	: hit rate of the cache hierarchy with FIFO
$\mathbf{S}''(\mathbf{A}_k)$: set of states where: <ul style="list-style-type: none"> • the items in the cache are the same as in \mathbf{A}_k • the order of these items are particularly different from \mathbf{A}_k such that $\mathbf{S}''(\mathbf{A}_k) = \{\mathbf{A}_{k''} \in \Phi \mathbf{A}_{k''} = [\mathbf{a}_2^{(k)}, \dots, \mathbf{a}_m^{(k)}, \mathbf{a}_1^{(k)}, \mathbf{a}_{m+1}^{(k)}, \dots, \mathbf{a}_C^{(k)}] \text{ for } 2 \leq m \leq C\}$
R_{LRU}	: hit rate of a discrete cache with LRU
R_{LRU}^H	: hit rate of the cache hierarchy with LRU
\mathbf{a}_k	: arbitrary item in X , where $k \in \{1, 2, 3\}$

5.3 Performance Analysis of Probabilistic Caching Scheme and Random Replacement Policy

In this section, we analyze the performance of a caching system whose cache management scheme is a combination of $Prob(q)$ and RR in the contexts of a discrete cache and a cache hierarchy. We first show that the hit rate of a discrete cache with RR policy does not depend on a caching probability q by formulating a Markov chain of a discrete cache.

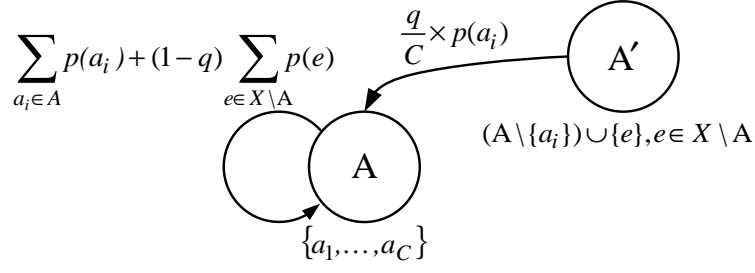


Figure 5.1: State transition probability in the Markov chain for a discrete cache whose cache management is a combination of $Prob(q)$ and RR.

Moreover, we show that the caching probability affects the hit rate of a cache hierarchy with RR policy.

Let us start with a discrete cache. Define X as the set of items available for download. The capacity of the considered cache is equal to C , where $C \leq |X|$. The interplay between a $Prob(q)$ and RR can be shown through the following theorem.

Theorem 1. *A hit rate of a discrete cache with RR is independent of caching probability q , where $0 < q \leq 1$.*

Proof. Define $A = \{a_1, a_2, \dots, a_C\}$ as an unordered collection of items that represents a cache state. Define Γ as the state space of the Markov chain. Let $\pi(A)$ be a stationary distribution of cache state A , where $A \in \Gamma$.

Lemma 1. *The stationary distribution of cache state A can be expressed by*

$$\pi(A) = \frac{\prod_{a_i \in A} p(a_i)}{\sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)}, \quad (5.1)$$

where \tilde{A} is a cache state in Γ .

Proof. Let us first verify that the function $\pi(A)$ in (5.1) is a probability distribution. It can be shown that $\pi(A) \geq 0$ for all $A \in \Gamma$ and

$$\sum_{A \in \Gamma} \pi(A) = \sum_{A \in \Gamma} \frac{\prod_{a_i \in A} p(a_i)}{\sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)} = 1. \quad (5.2)$$

By using $Prob(q)$, the probability that a CCN router inserts an item into its cache is equal to q . The transition probability among the cache states is shown in Figure 5.1. There are two specific patterns of the cache states that can become state A in one state transition: state A and the states that have one item different from state A . We denote the set of the states satisfying the latter by $S_i(A)$, where $S_i(A) = \{(A \setminus \{a_i\}) \cup \{e\} \mid e \in X \setminus A\}$.

The state transition probability from A to itself is equal to the probability that any item in A is requested or item $e \in X \setminus A$ is requested but the cache decides not to cache the item e . It is equal to

$$\sum_{a_i \in A} p(a_i) + (1 - q) \sum_{e \in X \setminus A} p(e).$$

On the other hand, the state transition probability from $A' \in S_i(a)$ to A is equal to the probability that item a_i is requested and the element that is in A' but not in A is a target to be discarded as well as the cache decides to store the item a_i . The state transition probability is thus equal to

$$\frac{1}{C} \times q \times p(a_i).$$

The stationary distribution of cache state A for a caching probability q can be written as

$$\begin{aligned} \pi(A) &= \left(\sum_{a_i \in A} p(a_i) + (1 - q) \sum_{e \in X \setminus A} p(e) \right) \pi(A) + \frac{q}{C} \sum_{a_i \in A} \left(p(a_i) \sum_{A' \in S_i(A)} \pi(A') \right) \\ &= \left(1 - q \sum_{e \in X \setminus A} p(e) \right) \pi(A) + \frac{q}{C} \sum_{a_i \in A} \left(p(a_i) \sum_{A' \in S_i(A)} \pi(A') \right). \end{aligned} \quad (5.3)$$

By substituting $\pi(A)$ in (5.3) with the term on the right side of (5.1), we obtain

$$\begin{aligned} \prod_{a_i \in A} p(a_i) &= \left(1 - q \sum_{e \in X \setminus A} p(e) \right) \left(\prod_{a_i \in A} p(a_i) \right) \\ &\quad + \frac{q}{C} \sum_{a_i \in A} \left(p(a_i) \sum_{A' \in S_i(A)} \prod_{a_j \in A'} p(a_j) \right) \\ &= \left(\prod_{a_i \in A} p(a_i) \right) - q \left(\sum_{e \in X \setminus A} p(e) \right) \left(\prod_{a_i \in A} p(a_i) \right) \\ &\quad + \frac{q}{C} \sum_{a_i \in A} \left[p(a_i) \left(\prod_{a_j \in A \setminus \{a_i\}} p(a_j) \right) \sum_{e \in X \setminus A} p(e) \right] \\ &= \left(\prod_{a_i \in A} p(a_i) \right) - q \left(\sum_{e \in X \setminus A} p(e) \right) \left(\prod_{a_i \in A} p(a_i) \right) \\ &\quad + q \left(\prod_{a_i \in A} p(a_i) \right) \left(\sum_{e \in X \setminus A} p(e) \right) \\ &= \prod_{a_i \in A} p(a_i). \end{aligned} \quad (5.4)$$

Thus, (5.1) is the stationary distribution of cache state A , $\pi(A)$. \square

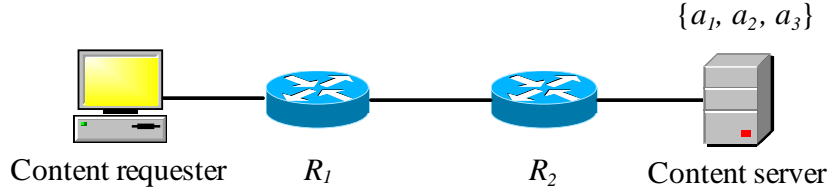


Figure 5.2: A content-centric network forming a two-level caching hierarchy.

The stationary distribution of any cache state remains unchanged for $0 < q \leq 1$ according to (5.4). The hit rate of a discrete cache is equal to the probability that a request of an item arrives at the cache while it has the item. Let $\gamma(a)$ be the set of states that the cache stores item a , i.e., $\gamma(a) = \{A \mid A \ni a_i\}$, where $\gamma(a_i) \subset \Gamma$. Given RR as the cache replacement policy, the hit rate of the cache can be written as

$$\begin{aligned} R_{RR} &= \sum_{a \in X} p(a) \sum_{A \in \gamma(a)} \pi(A) \\ &= \sum_{a \in X} p(a) \sum_{A \in \gamma(a)} \frac{\prod_{a_i \in A} p(a_i)}{\sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)}. \end{aligned} \quad (5.5)$$

The hit rate of a discrete cache controlled by RR is independent of a caching probability assigned to a probabilistic caching scheme. \square

Next, let us consider a behavior of $Prob(q)$ when it is used with RR in a network of caches forming a cache hierarchy. While the hit rate of a discrete cache with a combination of $Prob(q)$ and RR is independent of a caching probability q as proven in Theorem 1, it can be demonstrated that the hit rate of a cache hierarchy depends on a caching probability q .

Consider a network in Figure 5.2 which establishes a cache hierarchy. There are two routers, R_1 and R_2 , both are individually managed by a combination of $Prob(q)$ and RR. Requests of items access a network through R_1 , and only the requests that are not satisfied by R_1 enter R_2 . The cache of each router can store at most two items, $C = 2$. The content server stores three unique items, $X = \{a_1, a_2, a_3\}$.

Proposition 1. *The hit rate of the cache hierarchy with a combination of $Prob(q)$ and RR varies inversely with a caching probability q , for $0 < q \leq 1$.*

Proof. Let Δ be the state space of a Markov chain of the cache hierarchy. Define B as a state of the cache hierarchy, where $B = [A_{(1)}, A_{(2)}]$, $A_{(i)}$ is a cache state of router R_i for

$i \in \{1, 2\}$, and $A_{(i)} \in \Gamma$.

Lemma 2. *Given a caching probability q , the stationary distribution of cache state $[A, A']$ can be expressed by*

$$\pi([A, A']) = \begin{cases} \frac{\prod_{a_i \in A} p(a_i)}{(5-3q) \sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)}, & \text{if } A' = A \\ \frac{(4-3q) \prod_{a_i \in A} p(a_i)}{(10-6q) \sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)}, & \text{otherwise.} \end{cases} \quad (5.6)$$

Proof. π in (5.6) is a probability distribution as $\pi([A, A']) \geq 0$ for all $A, A' \in \Gamma$ and $\sum_{A, A' \in \Gamma} \pi([A, A']) = 1$.

Let A, A', A'' be distinct elements in Γ . Without loss of generality, we assume that $A = \{a_1, a_2\}$, $A' = \{a_1, a_3\}$ and $A'' = \{a_2, a_3\}$. For simplicity, we refer to $p(a_i)$ as p_i for $i \in \{1, 2, 3\}$. The stationary distribution of a state in Δ can be expressed by

$$\begin{aligned} \pi([A, A]) &= (p_1 + p_2 + (1-q)^2 p_3) \pi([A, A]) \\ &\quad + \frac{p_1 q}{2} \pi([A'', A]) + \frac{p_2 q}{2} \pi([A', A]) \\ &\quad + \frac{p_2 q^2}{4} \pi([A', A']) + \frac{p_1 q^2}{4} \pi([A'', A'']), \end{aligned} \quad (5.7)$$

$$\begin{aligned} \pi([A, A']) &= (p_1 + p_2 + (1-q)p_3) \pi([A, A']) \\ &\quad + \frac{p_3(1-q)q}{2} \pi([A, A]) + \frac{p_2 q(1-q)}{2} \pi([A', A']) \\ &\quad + \frac{p_1 q}{2} \pi([A'', A']) + \frac{p_1 q^2}{4} \pi([A'', A'']) \end{aligned} \quad (5.8)$$

$$\begin{aligned} \pi([A, A'']) &= (p_1 + p_2 + (1-q)p_3) \pi([A, A'']) \\ &\quad + \frac{p_3(1-q)q}{2} \pi([A, A]) + \frac{p_2 q^2}{4} \pi([A', A']) \\ &\quad + \frac{p_2 q}{2} \pi([A', A'']) + \frac{p_1 q(1-q)}{2} \pi([A'', A'']). \end{aligned} \quad (5.9)$$

π in (5.6) is a stationary distribution in the considering Markov chain if and only if it satisfies (5.7), (5.8), and (5.9). Let us begin with state $[A, A]$. By substituting π in (5.7) with that in (5.6), we obtain

$$\begin{aligned} 2p_1 p_2 &= 2(p_1 + p_2 + (1-q)^2 p_3) p_1 p_2 + \frac{p_1 q}{2} (4-3q) p_2 p_3 \\ &\quad + \frac{p_2 q}{2} (4-3q) p_1 p_3 + \frac{p_2 q^2}{4} 2p_1 p_3 + \frac{p_1 q^2}{4} 2p_2 p_3 \\ &= 2(p_1 + p_2 + (1-q)^2 p_3) p_1 p_2 + (4q - 2q^2) p_1 p_2 p_3 \\ &= 2p_1 p_2. \end{aligned}$$

It can be demonstrated that π in (5.6) satisfies (5.8) and (5.9) by repeating the same

process. □

With (5.6), the hit rate of the cache hierarchy can be computed as

$$\begin{aligned} R_{RR}^H &= 1 - p(e)\pi([A, A]) - p(e')\pi([A', A']) - p(e'')\pi([A'', A'']) \\ &= 1 - \frac{3 \prod_{a_i \in X} p(a_i)}{(5 - 3q) \sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)}, \end{aligned} \quad (5.10)$$

which increases inversely with a caching probability q . □

5.4 Performance Analysis of Probabilistic Caching Scheme and First In First Out Policy

In this section, the performance of a caching system with a combination of $Prob(q)$ and FIFO is analyzed. The state transitions of this cache model rely on the order of items in the cache. When the cache is full, the item at the last position is evicted before a new item is inserted into the first position of a cache. The probability of item eviction and insertion is equal to caching probability q .

To study the behavior of this caching system, we first prove the independence between the hit rate of a discrete cache using FIFO and a caching probability q . Next, we show that the performance of FIFO and that of RR are the same, when they are used to control a discrete cache regardless of caching probability. Moreover, we show that the hit rate of a cache hierarchy with FIFO can be improved by reducing a caching probability q . Finally, we show that the hit rate of a cache hierarchy with RR bounds that with FIFO.

Theorem 2. *The hit rate of a discrete cache with FIFO is independent of a caching probability q , where $0 < q \leq 1$.*

Proof. Define \mathbf{A}_k as an array of C distinct items representing a cache state, where $\mathbf{A}_k = [\mathbf{a}_1^{(k)}, \dots, \mathbf{a}_C^{(k)}]$ and $\mathbf{a}_i^{(k)}$ is the item at the i^{th} position of a cache in state \mathbf{A}_k . Let $M(\mathbf{A}_k)$ be a set of items, when a cache is in state \mathbf{A}_k . Let Φ be the state space of a Markov chain of the considering caching system. Φ expands over all C -permutations of X , where $\Phi := \{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{\left(\frac{X!}{(X-C)!}\right)}\}$. The stationary distribution of the state of Φ can be computed based on the following lemma.

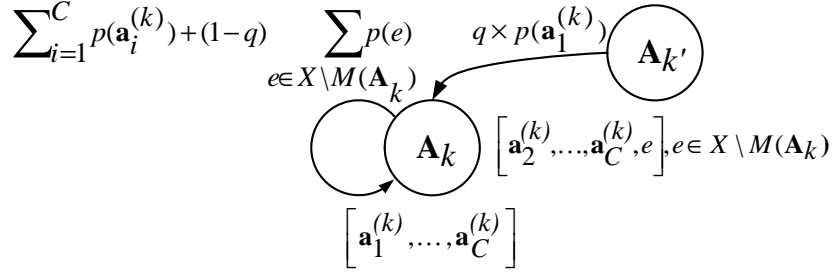


Figure 5.3: State transition probability in the Markov chain of a discrete cache whose cache management is a combination of $Prob(q)$ and FIFO.

Lemma 3. *Given a caching probability q , the stationary distribution of state \mathbf{A}_k can be expressed by*

$$\pi(\mathbf{A}_k) = \frac{\prod_{i=1}^C p(\mathbf{a}_i^{(k)})}{\sum_{\mathbf{A}_l \in \Phi} \prod_{j=1}^C p(\mathbf{a}_j^{(l)})}. \quad (5.11)$$

Proof. π in (5.11) is a probability distribution as $\pi(\mathbf{A}_k) \geq 0$ for all $\mathbf{A}_k \in \Phi$ and $\sum_{k=1}^{|\Phi|} \pi(\mathbf{A}_k) = 1$. We next show that $\pi(\mathbf{A}_k)$ is the stationary distribution of a cache state \mathbf{A}_k in Φ . Define $\mathbf{S}'(\mathbf{A}_k)$ as a set of the cache states, where item $\mathbf{a}_1^{(k)}$ is not in a cache, items $\mathbf{a}_2^{(k)}, \dots, \mathbf{a}_C^{(k)}$ are at the $1^{st}, \dots, (C-1)^{th}$ positions of a cache, and item e is at the C^{th} position of a cache, where $e \in X \setminus M(\mathbf{A}_k)$. In other words, $\mathbf{S}'(\mathbf{A}_k) = \{\mathbf{A}_{k'} \in \Phi | \mathbf{A}_{k'} = [\mathbf{a}_2^{(k)}, \dots, \mathbf{a}_C^{(k)}, e], e \in X \setminus M(\mathbf{A}_k)\}$.

Consider the state transition probability of a Markov chain in Figure 5.3. Either state \mathbf{A}_k or $\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)$ transits to state \mathbf{A}_k . A transition probability from state \mathbf{A}_k to itself is equal to the probability of either one of the two following events. First, it is equal to the probability that any item in \mathbf{A}_k is requested. Second, it is equal to the probability that a request of the item that is not in \mathbf{A}_k arrives at the cache (a cache miss occurs) but the cache decides not to cache the corresponding item. This probability is equal to

$$\sum_{i=1}^C p(\mathbf{a}_i^{(k)}) + (1-q) \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e).$$

The transition probability from state $\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)$ to state \mathbf{A}_k is equal to the probability that a request of item $\mathbf{a}_1^{(k)}$ arrives at a cache and a cache decides to cache the corresponding item. This probability is equal to

$$q \times p(\mathbf{a}_1^{(k)}).$$

With the above transition probabilities, the stationary distribution of $\pi(\mathbf{A}_k)$ can be

expressed as

$$\begin{aligned}
\pi(\mathbf{A}_k) &= \left(\sum_{i=1}^C p(\mathbf{a}_i^{(k)}) + (1-q) \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e) \right) \pi(\mathbf{A}_k) \\
&\quad + q \times p(\mathbf{a}_1^{(k)}) \left(\sum_{\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)} \pi(\mathbf{A}_{k'}) \right) \\
&= \left(1-q \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e) \right) \pi(\mathbf{A}_k) \\
&\quad + q \times p(\mathbf{a}_1^{(k)}) \left(\sum_{\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)} \pi(\mathbf{A}_{k'}) \right). \tag{5.12}
\end{aligned}$$

π in the lemma statement is the stationary distribution of state in Φ if and only if it satisfies (5.12). We prove this lemma by substituting $\pi(\mathbf{A}_k)$ with the term on the right side of (5.11). Then, we obtain

$$\begin{aligned}
\prod_{i=1}^C p(\mathbf{a}_i^{(k)}) &= \left(1-q \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e) \right) \left(\prod_{i=1}^C p(\mathbf{a}_i^{(k)}) \right) \\
&\quad + q \times p(\mathbf{a}_1^{(k)}) \sum_{\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)} \left(\prod_{j=1}^C p(\mathbf{a}_j^{(k')}) \right) \\
&= \left(1-q \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e) \right) \left(\prod_{i=1}^C p(\mathbf{a}_i^{(k)}) \right) \\
&\quad + q \times p(\mathbf{a}_1^{(k)}) \sum_{e \in X \setminus M(\mathbf{A}_k)} \left(\prod_{i=2}^C p(\mathbf{a}_i^{(k)}) p(e) \right) \\
&= \prod_{i=1}^C p(\mathbf{a}_i^{(k)}) - q \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e) \left(\prod_{i=1}^C p(\mathbf{a}_i^{(k)}) \right) \\
&\quad + q \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e) \left(\prod_{i=1}^C p(\mathbf{a}_i^{(k)}) \right) \\
&= \prod_{i=1}^C p(\mathbf{a}_i^{(k)}).
\end{aligned}$$

Therefore, $\pi(\mathbf{A}_k)$ in the lemma statement is the stationary distribution of cache state \mathbf{A}_k . \square

Define $\beta(a) \subseteq \Phi$ as a set of the cache states containing item a , where $\beta(a) = \{\mathbf{A}_k | M(\mathbf{A}_k) \ni a\}$. Based on Lemma 3, the hit rate of a discrete cache with a combi-

nation of $Prob(q)$ and FIFO can be expressed by

$$\begin{aligned}
R_{FIFO} &= \sum_{a \in X} p(a) \sum_{\mathbf{A}_k \in \beta(a)} \pi(\mathbf{A}_k) \\
&= \sum_{a \in X} p(a) \sum_{\mathbf{A}_k \in \beta(a)} \frac{\prod_{i=1}^C p(\mathbf{a}_i^k)}{\sum_{\mathbf{A}_l \in \Phi} \prod_{j=1}^C p(\mathbf{a}_j^{(l)})}.
\end{aligned} \tag{5.13}$$

Therefore, the hit rate is independent of a caching probability. \square

We can observe from Lemma 3 that the stationary distribution of state \mathbf{A}_m is equal to that of \mathbf{A}_n , if $M(\mathbf{A}_m) = M(\mathbf{A}_n)$. With Lemma 1 and Lemma 3, we set up a theorem as follows.

Theorem 3. *The hit rate of a discrete cache with a combination of $Prob(q)$ and RR is equal to that with a combination of $Prob(q)$ and FIFO, regardless of a caching probability q . In other words, $R_{RR} = R_{FIFO}$, for $0 < q \leq 1$.*

Proof. Consider that X is the set of all items, Γ is a set of the subsets of X with the size of C , and $\gamma(a)$ is a set of such subsets that contain item a , where $\gamma(a) \subset \Gamma$. Define P_A as a set that contains $C!$ permutations of A , where $A \in \Gamma$. The hit rate of a discrete cache with a combination of $Prob(q)$ and FIFO can be written as

$$R_{FIFO} = \sum_{a \in X} p(a) \sum_{A \in \gamma(a)} \sum_{\mathbf{A}_k \in P_A} \pi(\mathbf{A}_k). \tag{5.14}$$

Since $\mathbf{A}_m, \mathbf{A}_n \in P_A$, $\pi(\mathbf{A}_m) = \pi(\mathbf{A}_n)$. Moreover, $\prod_{i=1}^C p(\mathbf{a}_i^{(k)}) = \prod_{a_i \in M(\mathbf{A}_k)} p(a_i)$. Therefore, we obtain

$$\begin{aligned}
\sum_{\mathbf{A}_k \in P_A} \pi(\mathbf{A}_k) &= (C!) \pi(\mathbf{A}_k) \\
&= \frac{(C!) \prod_{a_i \in M(\mathbf{A}_k)} p(a_i)}{\sum_{\mathbf{A}_l \in \Phi} \prod_{a_j \in M(\mathbf{A}_l)} p(a_j)} \\
&= \frac{(C!) \prod_{a_i \in A} p(a_i)}{\sum_{\tilde{A} \in \Gamma} \left((C!) \prod_{a_j \in \tilde{A}} p(a_j) \right)}.
\end{aligned}$$

Finally, (5.14) can be rewritten as

$$R_{FIFO} = \sum_{a \in X} p(a) \sum_{A \in \gamma(a)} \frac{\prod_{a_i \in A} p(a_i)}{\sum_{\tilde{A} \in \Gamma} \prod_{a_j \in \tilde{A}} p(a_j)}, \tag{5.15}$$

which is equal to R_{RR} in (5.5) for $0 < q \leq 1$. \square

Next, let us consider another property of $Prob(q)$ when it is used with FIFO by a cache hierarchy. Consider a network in Figure 5.2 which establishes a cache hierarchy.

The server, requester, and routers in the network are configured as in the previous section except that the cache replacement policy of each router is FIFO.

Proposition 2. *The hit rate of the cache hierarchy with a combination of $\text{Prob}(q)$ and FIFO varies inversely with a caching probability q .*

Proof. Let \mathbf{B} denote a state of a Markov chain of this cache hierarchy, where $\mathbf{B} := [\mathbf{A}_{(1)}, \mathbf{A}_{(2)}]$ and $\mathbf{A}_{(i)}$ is a cache state of router R_i for $i \in \{1, 2\}$. Define Λ as the state space of this Markov chain. We shall demonstrate that Λ is non-ergodic. Specifically, Λ consists of four disjoint ergodic sets: $\Lambda_{i,j} = \{[\mathbf{A}_{(1)}, \mathbf{A}_{(2)}] \mid \mathbf{A}_{(1)} \in \Phi_i, \mathbf{A}_{(2)} \in \Phi_j\}$ for $i, j \in \{1, 2\}$, where $\Phi_1 = \{[a_1, a_2], [a_2, a_3], [a_3, a_1]\}$ and $\Phi_2 = \{[a_2, a_1], [a_1, a_3], [a_3, a_2]\}$.

Define \mathbf{B}_0 as the initial-state of the cache hierarchy. The steady-state of this Markov chain depends on \mathbf{B}_0 since a state cannot transit to the states outside its own ergodic set. Let us consider the following example. Suppose that $\mathbf{B}_0 = [[a_1, a_2], [a_1, a_2]]$ and $\mathbf{B}_i = [[a_1, a_2], [a_2, a_1]]$. There are no requests that change \mathbf{B}_0 to \mathbf{B}_i . In other words, $[a_1, a_2]$ will never transit to $[a_2, a_1]$.

For simplicity, we refer to $p(\mathbf{a}_i)$ as p_i for $i \in \{1, 2, 3\}$, and let $p_{\text{all}} := p_1 p_2 + p_1 p_3 + p_2 p_3$. Moreover, let \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 be distinct elements in X .

Lemma 4. *Given $\mathbf{B}_0 \in \Lambda_{i,i}$, for $i \in \{1, 2\}$, $\pi(\mathbf{B}) = 0$ if $\mathbf{B} \notin \Lambda_{i,i}$. Otherwise,*

$$\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_2]]) = \frac{p_1 p_2}{(4 - 3q)p_{\text{all}}}, \quad (5.16)$$

$$\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_1]]) = \frac{(2 - 2q)p_1 p_2}{(4 - 3q)p_{\text{all}}}, \quad (5.17)$$

$$\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_3]]) = \frac{(1 - q)p_1 p_2}{(4 - 3q)p_{\text{all}}}, \quad (5.18)$$

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in X$.

Proof. π in the lemma statement is a probability distribution as $\pi(\mathbf{B}) \geq 0$ for all $\mathbf{B} \in \Lambda_{i,i}$, and $\sum_{\mathbf{B} \in \Lambda_{i,i}} \pi(\mathbf{B}) = 1$.

Let $\mathbf{A}_1 := [\mathbf{a}_1, \mathbf{a}_2]$, $\mathbf{A}_2 := [\mathbf{a}_3, \mathbf{a}_1]$, and $\mathbf{A}_3 := [\mathbf{a}_2, \mathbf{a}_3]$, where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in X$. The

stationary distribution of state in $\Lambda_{i,i}$ for $i \in \{1, 2\}$ can be expressed by

$$\begin{aligned}\pi([\mathbf{A}_1, \mathbf{A}_1]) &= (p_1 + p_2 + (1 - q)^2 p_3) \pi([\mathbf{A}_1, \mathbf{A}_1]) \\ &\quad + qp_1 \pi([\mathbf{A}_3, \mathbf{A}_1]) + q^2 p_1 \pi([\mathbf{A}_3, \mathbf{A}_3]),\end{aligned}\tag{5.19}$$

$$\begin{aligned}\pi([\mathbf{A}_1, \mathbf{A}_2]) &= (p_1 + p_2 + (1 - q)p_3) \pi([\mathbf{A}_1, \mathbf{A}_2]) \\ &\quad + (1 - q)qp_3 \pi([\mathbf{A}_1, \mathbf{A}_1]) \\ &\quad + q \times p_1 \pi([\mathbf{A}_3, \mathbf{A}_2]),\end{aligned}\tag{5.20}$$

$$\begin{aligned}\pi([\mathbf{A}_1, \mathbf{A}_3]) &= (p_1 + p_2 + (1 - q)p_3) \pi([\mathbf{A}_1, \mathbf{A}_3]) \\ &\quad + q(1 - q) \times p_1 \pi([\mathbf{A}_3, \mathbf{A}_3]).\end{aligned}\tag{5.21}$$

π in the lemma statement is the stationary distribution of state in $\Lambda_{i,i}$ for $i \in \{1, 2\}$ if and only if it satisfies (5.19), (5.20), and (5.21). We prove the lemma by substituting π in (5.19), (5.20), and (5.21) with the terms on the right side of (5.16), (5.17), and (5.18). We then follow the same steps used to proof Lemma 2 to complete this proof. \square

Lemma 5. *Given $\mathbf{B}_0 \in \Lambda_{i,j}$, where $i \neq j$ for $i, j \in \{1, 2\}$, $\pi(\mathbf{B}) = 0$, if $\mathbf{B} \notin \Lambda_{i,j}$. Otherwise,*

$$\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]) = \frac{p_1 p_2}{(5 - 3q)p_{\text{all}}},\tag{5.22}$$

$$\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_2]) = \frac{(2 - 2q)p_1 p_2}{(5 - 3q)p_{\text{all}}},\tag{5.23}$$

$$\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_3]) = \frac{(2 - q)p_1 p_2}{(5 - 3q)p_{\text{all}}},\tag{5.24}$$

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in X$.

Proof. It can be shown that π is the stationary distribution of state in $\Lambda_{i,j}$, where $i \neq j$ for $i, j \in \{1, 2\}$. The stationary distribution of state in $\Lambda_{i,j}$, where $i \neq j$ for $i, j \in \{1, 2\}$,

can be expressed by

$$\begin{aligned}\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]) &= (p_1 + p_2 + (1 - q)^2 p_3) \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]) \\ &\quad + qp_1 \pi([\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_2, \mathbf{a}_1]),\end{aligned}\tag{5.25}$$

$$\begin{aligned}\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_2]) &= (p_1 + p_2 + (1 - q)p_3) \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_2]) \\ &\quad + (1 - q)qp_3 \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]) \\ &\quad + q(1 - q)p_1 \pi([\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_3, \mathbf{a}_2]),\end{aligned}\tag{5.26}$$

$$\begin{aligned}\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_3]) &= (p_1 + p_2 + (1 - q)p_3) \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_3]) \\ &\quad + q^2 p_1 \pi([\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_3, \mathbf{a}_2]) + qp_1 \pi([\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_1, \mathbf{a}_3]),\end{aligned}\tag{5.27}$$

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in X$.

We adopt the same approach used to prove Lemma 4 to complete this proof. \square

Next, with Lemma 4 and Lemma 5, we compute the hit rate of the considering cache hierarchy. Define $\mu_{i,j}$ as a probability that the initial-state of the cache hierarchy, denoted by \mathbf{B}_0 , is in $\Lambda_{i,j}$, where $0 \leq \mu_{i,j} \leq 1$ for all $i, j \in \{1, 2\}$ and $\sum_{i,j} \mu_{i,j} = 1$. The hit rate of the cache hierarchy can be computed as

$$\begin{aligned}R_{FIFO}^H &= 1 - \mu_{1,1} \sum_{[\mathbf{a}_1, \mathbf{a}_2] \in \Phi_1} p_3 \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_2]) \\ &\quad - \mu_{2,2} \sum_{[\mathbf{a}_2, \mathbf{a}_1] \in \Phi_2} p_3 \pi([\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_2, \mathbf{a}_1]) \\ &\quad - \mu_{1,2} \sum_{[\mathbf{a}_1, \mathbf{a}_2] \in \Phi_1} p_3 \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]) \\ &\quad - \mu_{2,1} \sum_{[\mathbf{a}_2, \mathbf{a}_1] \in \Phi_2} p_3 \pi([\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_1, \mathbf{a}_2]) \\ &= 1 - (\mu_{1,1} + \mu_{2,2}) \frac{3p_1 p_2 p_3}{(4 - 3q)p_{\text{all}}} \\ &\quad - (\mu_{1,2} + \mu_{2,1}) \frac{3p_1 p_2 p_3}{(5 - 3q)p_{\text{all}}}\end{aligned}\tag{5.28}$$

\square

We next establish the relation between the performance of a cache hierarchy with FIFO and that with RR under a caching probability q .

Corollary 1. *A hit rate of a cache hierarchy with a combination of Prob(q) and FIFO is not greater than that with a combination of Prob(q) and RR, where $0 < q \leq 1$.*

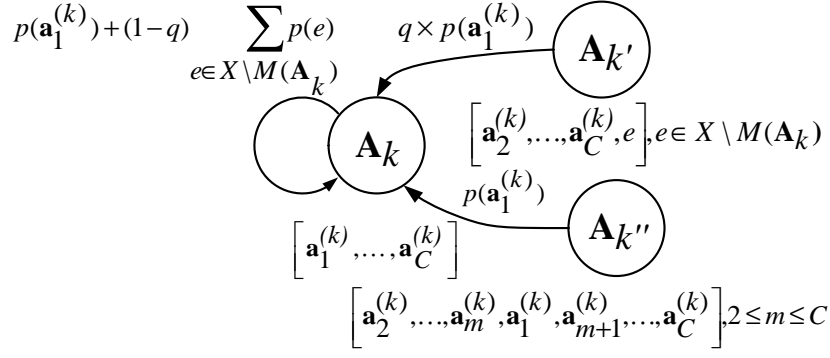


Figure 5.4: State transition probability in the Markov chain of a discrete cache whose cache management is a combination of $Prob(q)$ and LRU.

Proof. Consider (5.28). For all $0 \leq q \leq 1$,

$$\frac{3p_1p_2p_3}{(4-3q)p_{\text{all}}} \geq \frac{3p_1p_2p_3}{(5-3q)p_{\text{all}}}.$$

Because $\sum_{i,j} \mu_{i,j} = 1$, we finally obtain

$$(1 - \frac{3p_1p_2p_3}{(4-3q)p_{\text{all}}}) \leq R_{FIFO}^H \leq (1 - \frac{3p_1p_2p_3}{(5-3q)p_{\text{all}}}) = R_{RR}^H.$$

□

Based on Corollary 1, hit rate of the cache hierarchy with RR bounds that with FIFO for all caching probability q .

5.5 Performance Analysis of Probabilistic Caching Scheme and Least Recently Used Policy

In this section, we investigate the performance of a caching system with a combination of $Prob(q)$ and LRU. In addition to the insertion and eviction operations of FIFO, LRU has a *move-to-front* operation, where once an item currently in a cache is requested, it is moved to the first position of the cache.

Let us begin with hit rate of a discrete cache with LRU for a caching probability q . Figure 5.4 demonstrates a transition probability from other states to state \mathbf{A}_k in a Markov chain of a discrete cache with a combination of $Prob(q)$ and LRU. Define $\mathbf{S}''(\mathbf{A}_k)$ as a set of the cache states that store the same items as \mathbf{A}_k . These items are in particularly different order such that $\mathbf{S}''(\mathbf{A}_k) = \{\mathbf{A}_{k''} \in \Phi | \mathbf{A}_{k''} = [\mathbf{a}_2^{(k)}, \dots, \mathbf{a}_m^{(k)}, \mathbf{a}_1^{(k)}, \mathbf{a}_{m+1}^{(k)}, \dots, \mathbf{a}_C^{(k)}], \text{ where } 2 \leq m \leq C\}$.

Three states transiting to state \mathbf{A}_k consists of \mathbf{A}_k , $\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)$, and $\mathbf{A}_{k''} \in \mathbf{S}''(\mathbf{A}_k)$. The state transition probability from \mathbf{A}_k to itself is a probability of either one of the two events. First, it is an event that item $\mathbf{a}_1^{(k)}$ is requested. Second, it is an event that a request of an item not in \mathbf{A}_k arrives at the cache (a cache miss occurs) but the cache decides not to cache the corresponding item. This probability can be computed as

$$p(\mathbf{a}_1) + (1 - q) \sum_{e \in X \setminus M(\mathbf{A}_k)} p(e).$$

The state transition probability from state $\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)$ to \mathbf{A}_k is also dependent on a caching probability q . Because $\mathbf{a}_1^{(k)}$ is not in $\mathbf{A}_{k'}$, the state transition probability is thus equal to the probability that a request of item $\mathbf{a}_1^{(k)}$ arrives at a cache (which causes a cache miss) and a cache decides to store the item that comes in and corresponds to the request. This probability can be computed as

$$q \times p(\mathbf{a}_1^{(k)}).$$

In contrast, the state transition probability from state $\mathbf{A}_{k''} \in \mathbf{S}''(\mathbf{A}_k)$ to \mathbf{A}_k remains unchanged since the move-to-front operation is independent of caching probability q . This probability is equal to $p(\mathbf{a}_1^{(k)})$.

Based on the state transition probability, the stationary distribution of cache state $\pi(\mathbf{A}_k)$ can be expressed by

$$\begin{aligned} \pi(\mathbf{A}_k) = & \left(p(\mathbf{a}_1^{(k)}) + (1 - q) \sum_{e \in X \setminus M(\mathbf{A})} p(e) \right) \pi(\mathbf{A}_k) \\ & + q \times p(\mathbf{a}_1^{(k)}) \left(\sum_{\mathbf{A}_{k'} \in \mathbf{S}'(\mathbf{A}_k)} \pi(\mathbf{A}_{k'}) \right) \\ & + p(\mathbf{a}_1^{(k)}) \left(\sum_{\mathbf{A}_{k''} \in \mathbf{S}''(\mathbf{A}_k)} \pi(\mathbf{A}_{k''}) \right). \end{aligned} \quad (5.29)$$

We solve (5.29) for $\pi(\mathbf{A}_k)$ for all $\mathbf{A}_k \in \Phi$. Then, the hit rate of a caching system can be computed as

$$R_{LRU} = \sum_{a \in X} p(a) \sum_{\mathbf{A}_k \in \beta(a)} \pi(\mathbf{A}_k). \quad (5.30)$$

Next, we investigate the behavior of $Prob(q)$ when it is used with LRU by a cache hierarchy. Let us consider the network shown in Figure 5.2. The cache of each router in the network is controlled by LRU, whereas the other configurations are the same as

Section 5.4. Unlike a cache hierarchy with FIFO, the one with LRU is ergodic, i.e., different initial states all give the same steady-state of the cache hierarchy. The stationary distribution of each state in the Markov chain of the cache hierarchy with LRU can be expressed by

$$\begin{aligned}
\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_2]]) &= (p(\mathbf{a}_1) + (1 - q)^2 p(\mathbf{a}_3)) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_2]]) \\
&\quad + qp(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_1, \mathbf{a}_2]]) + qp(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_2, \mathbf{a}_1]]) \\
&\quad + q^2 p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_2, \mathbf{a}_3]]) \\
&\quad + p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_1, \mathbf{a}_2]]), \tag{5.31}
\end{aligned}$$

$$\begin{aligned}
\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_1]]) &= (p(\mathbf{a}_1) + (1 - q)p(\mathbf{a}_3)) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_1]]) \\
&\quad + (1 - q)qp(\mathbf{a}_3) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_2]]) \\
&\quad + (1 - q)p(\mathbf{a}_3) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_3]]) \\
&\quad + p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_3, \mathbf{a}_1]]), \tag{5.32}
\end{aligned}$$

$$\begin{aligned}
\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_3]]) &= p(\mathbf{a}_1) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_3]]) \\
&\quad + q(1 - q)p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_2, \mathbf{a}_3]]) \\
&\quad + p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_2, \mathbf{a}_3]]), \tag{5.33}
\end{aligned}$$

$$\begin{aligned}
\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]]) &= (p(\mathbf{a}_1) + (1 - q)^2 p(\mathbf{a}_3)) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]]) \\
&\quad + p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_2, \mathbf{a}_1]]), \tag{5.34}
\end{aligned}$$

$$\begin{aligned}
\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_2]]) &= (p(\mathbf{a}_1) + (1 - q)p(\mathbf{a}_3)) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_3, \mathbf{a}_2]]) \\
&\quad + (1 - q)p(\mathbf{a}_3) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_3]]) \\
&\quad + (1 - q)qp(\mathbf{a}_3) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1]]) \\
&\quad + q(1 - q)p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_3, \mathbf{a}_2]]) \\
&\quad + p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_3, \mathbf{a}_2]]), \tag{5.35}
\end{aligned}$$

$$\begin{aligned}
\pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_3]]) &= p(\mathbf{a}_1) \pi([[\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_3]]) + qp(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_3, \mathbf{a}_1]]) \\
&\quad + q^2 p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_3, \mathbf{a}_2]]) \\
&\quad + qp(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_3], [\mathbf{a}_1, \mathbf{a}_3]]) \\
&\quad + p(\mathbf{a}_1) \pi([[\mathbf{a}_2, \mathbf{a}_1], [\mathbf{a}_1, \mathbf{a}_3]]), \tag{5.36}
\end{aligned}$$

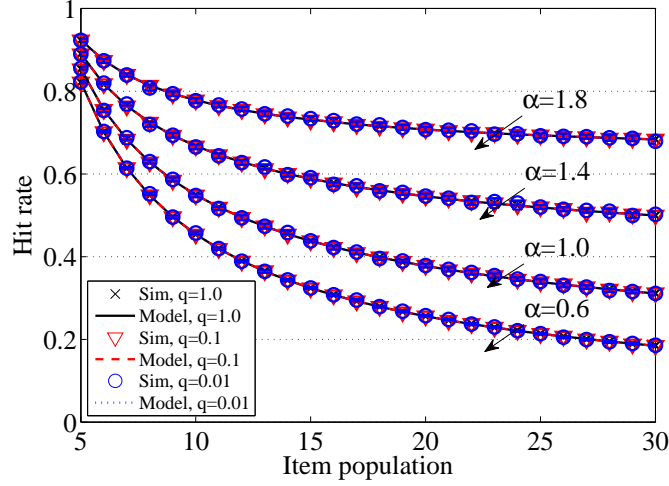


Figure 5.5: Hit rate versus item population for the discrete cache controlled by RR.

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in X$.

The stationary distribution of each state of the Markov chain is obtained by solving a system of linear equations that consists of (5.31), (5.32), (5.33), (5.34), (5.35), and (5.36).

Then, the hit rate of the considering cache hierarchy can be computed as

$$R_{LRU}^H = 1 - \sum_{[\mathbf{a}_1, \mathbf{a}_2] \in \Phi} p(\mathbf{a}_3) (\pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_1, \mathbf{a}_2]) + \pi([\mathbf{a}_1, \mathbf{a}_2], [\mathbf{a}_2, \mathbf{a}_1])). \quad (5.37)$$

5.6 Experimental Results

5.6.1 Model Validation and Insights

We begin the experiment with a discrete cache. A caching probability q is selected from a set $\{1.0, 0.1, 0.01\}$. Item population ranges from 5 to 30, where all items has the same size. A cache can store at most four items. We assume the access profile of requests follows the Zipf's law as in [15, 98]. Specifically, the probability of requesting item e_i is denoted by $p(e_i)$, where $p(e_i) = i^{-\alpha} \left(\sum_{j=1}^{|X|} j^{-\alpha} \right)^{-1}$, e_i denotes the item at the i^{th} rank of the item cardinality sorted by the item popularity in descending order, and $\alpha \in \{0.6, 1.0, 1.4, 1.8\}$. Note that our analytical model is applicable to any probability distribution of the access profile. We assume a constant rate of request generation, negligible delay, and abundant link bandwidth to suppress the impacts of link delay, request aggregation, and packet drop. We ignore the transient state and collect only the results in the steady-state of the caches. The simulation results are reported with 95% confidence interval.

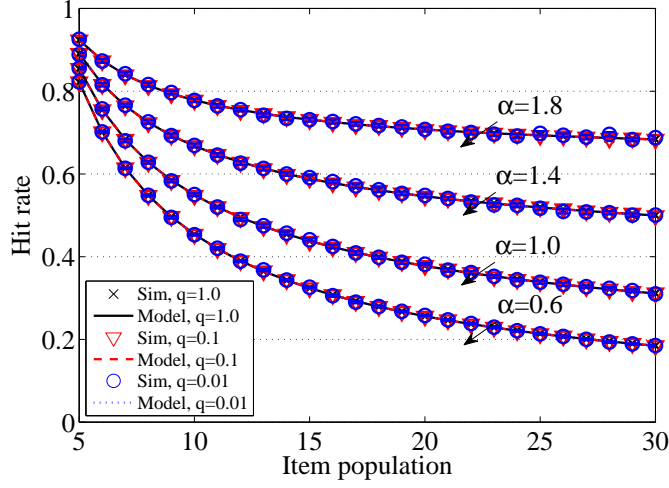


Figure 5.6: Hit rate versus item population for the discrete cache controlled by FIFO.

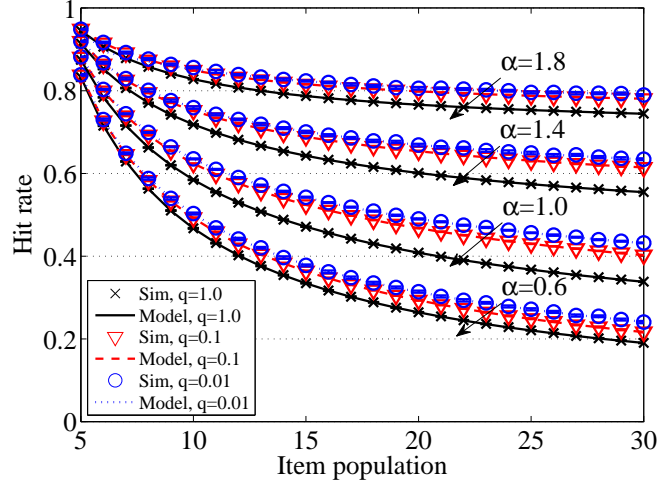


Figure 5.7: Hit rate versus item population of the discrete cache controlled by LRU.

Figures 5.5, 5.6, and 5.7 show the hit rate versus the item population in a discrete cache whose cache replacement policy is RR, FIFO, and LRU, respectively. The numerical results from our analytical model are identical to that from simulations. The hit rate provided by RR is identical to that by FIFO and does not vary with a caching probability. In other words, $Prob(q)$ does not have any impact to a discrete cache controlled by either RR or FIFO. In contrast, the hit rate varies inversely with a caching probability when a cache is controlled by LRU. The increment of the hit rate can be observed when a caching probability q decreases. The hit rate yielded by a combination of $Prob(q)$ and LRU is higher than that of RR and FIFO. We can see that the combination of $Prob(q)$ and LRU

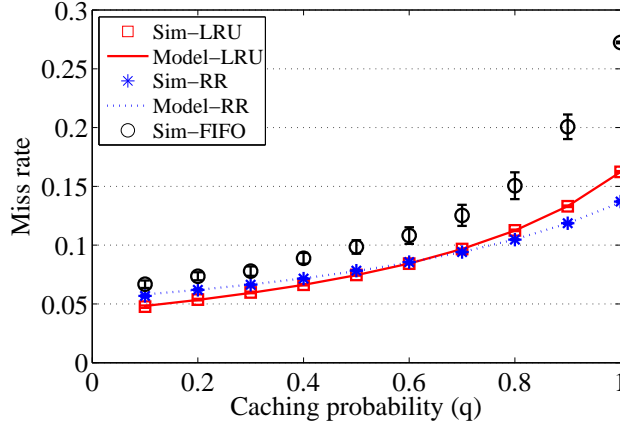


Figure 5.8: Miss rate of a cache network in Figure 5.2 versus a caching probability.

Table 5.2: Topological properties of the considered networks

Network	#node	#edge	Average degree	Topology
SINET4	50	55	2.2	mesh+star
GÈANT	40	59	2.95	mesh
Internet2	10	14	2.8	mesh

is an effective cache management for a discrete cache.

We next evaluate the accuracy of our analytical model of a cache hierarchy shown in Figure 5.2. Figure 5.8 shows the miss rate of the cache hierarchy when $\alpha = 1.0$. The numerical results from our model are identical to the simulation results both with RR and with LRU. We observe that the miss rate decreases when a caching probability is reduced for all cache replacement policies. The miss rate yielded by RR is lower than that by FIFO for all caching probabilities. The performance of FIFO approaches that of RR when a caching probability decreases. Moreover, RR yields lower miss rate than LRU when a caching probability is greater than 0.6. However, LRU provides lower miss rate than RR and FIFO when a caching probability approaches zero, thus becoming a more effective cache management scheme for a cache hierarchy.

5.6.2 Experiments with Real Internet Topologies

To validate the properties of $Prob(q)$ in more realistic networks, we run computer simulations on three academic Internet topologies: SINET4 [92], GÈANT [93], and Internet2 [94]. The topological properties of each network topology is summarized in Table 5.2.

We iterate ten simulations with the following configuration. Each network has 8,000

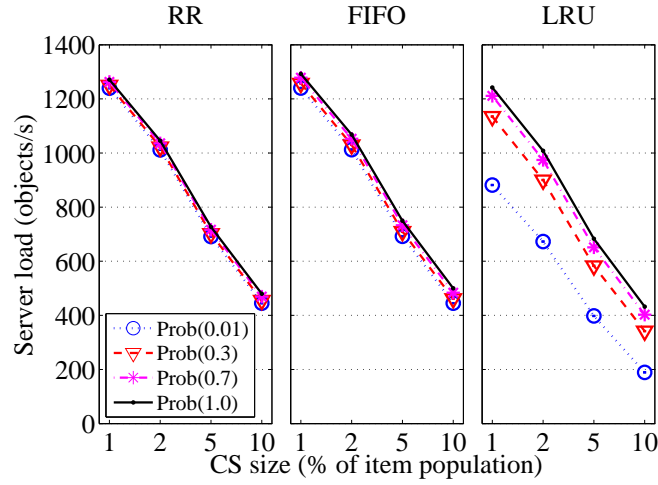


Figure 5.9: Server load versus the cache size in the SINET4 topology.

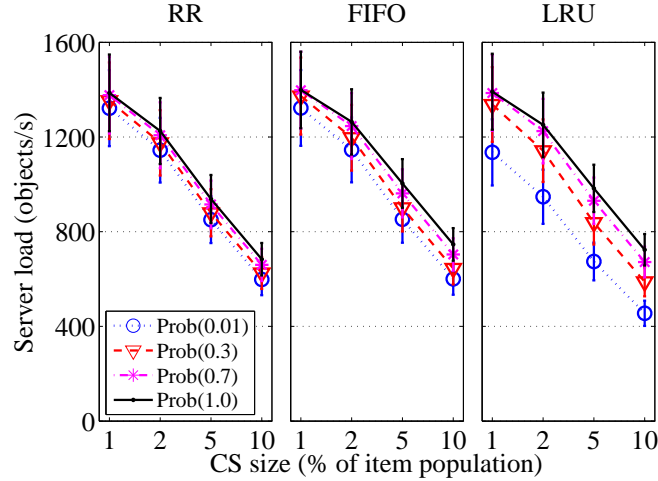


Figure 5.10: Server load versus the cache size in the GÉANT topology.

content objects, which are evenly distributed to eight servers (none of them store the same content object). The servers are connected to randomly selected routers. The cache size of each router ranges from 1% to 10% of the total content objects. The content requesters are connected to all routers and generate 50 requests per second, whereas the content popularity follows the Zipf's law with $\alpha = 1.0$. The delay and bandwidth of each link are set to be equal to 10 ms and 10 Gbps, respectively.

Figures 5.9, 5.10, and 5.11 show the server load versus the relative cache size of node in SINET4, GÉANT, and Internet2 topologies, respectively. Only the requests not satisfied by the in-network cache will reach the servers. The lower a server load, the higher

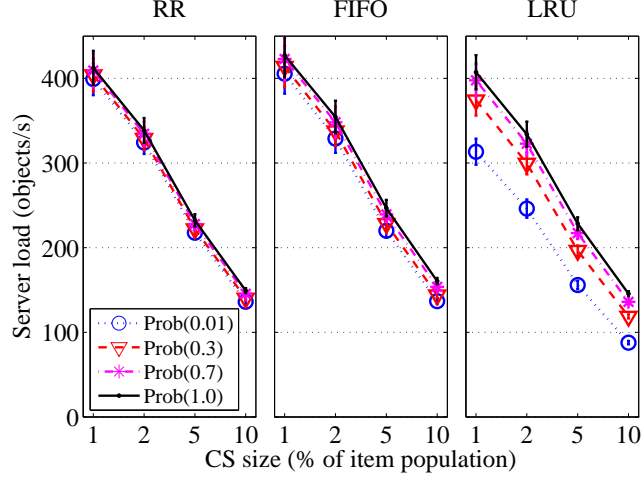


Figure 5.11: Server load versus the cache size in the Internet2 topology.

performance of the in-network cache. The results from three network topologies all agree with the conclusion obtained from our analytical model. The server load decreases when a caching probability is reduced for all cache replacement policies. The improvement is marginal both for RR and for FIFO. We observe that FIFO approaches RR when a caching probability decreases. In contrast, the improvement is significant for LRU. A combination of $Prob(q)$ and LRU gives an effective cache management for content-centric networks. However, if a CCN router cannot afford LRU because of its computational power limit, RR is preferred to FIFO.

5.7 Conclusion

We propose a Markov chain-based analytical model to analyze the performance of a probabilistic caching scheme with various cache replacement policies in content-centric networks. Based on the analytical model, we investigate several important properties of the probabilistic caching scheme. We demonstrate that the performance of a discrete cache controlled by either RR or FIFO is independent of a caching probability but that with LRU increases when a caching probability is reduced. In contrast, a caching probability impacts the performance of cache networks whose replacement policy is either RR, FIFO, or LRU. We also show that the performance of a probabilistic caching scheme paired with FIFO is upper-bounded by RR. A combination of a probabilistic caching scheme and LRU gives

an effective cache management scheme for content-centric networks as its performance is better than that of the other considered options. Nevertheless, if a CCN router cannot afford LRU because of its computational power limitation, RR is preferred to FIFO as it has better performance in the context of cache networks.

Chapter 6

Content Importance-based Routing Using Cooperative Network Coding

This chapter presents a novel QoS-aware routing scheme for reliable unicast transmission in multi-hop wireless networks. Cooperative network coding is exploited with special care to improve the utilization of wireless channels while maintaining QoS requirements of layered data. Simulation results show the effectiveness of this routing scheme under various users' demands and qualities of wireless links.

6.1 Introduction

In the second half of the dissertation, we focus on the data transmission in the edge area of the Internet. In a content retrieval path, contents are often delivered to end-users through the edge area of the Internet, where multi-hop wireless networks are increasingly becoming common. Links in multi-hop wireless networks are often unreliable and have limited capacity. In contrast, the quality of content at an end-user highly depends on the completeness and correctness of received data. Therefore, quality-of-service (QoS) is necessary for data transmission in multi-hop wireless networks.

In this chapter, we investigate a novel QoS-aware routing scheme for multi-hop wireless

networks. This routing scheme supports unicast transmission of layered data with QoS guarantee and improves channel utilization by applying CNC based on the local structure of the network. There are two main steps. In the first step, the scheme uses a linear optimization formulation to compute routes of all layered unicast flows. The constraints of this optimization problem, such as the transmission rate of each data layer and tolerable error rates in wireless transmissions, are derived to achieve QoS guarantee. In the second step, the proposed scheme decides whether or not CNC will be applied to different unicast flows at intermediate nodes to improve channel utilization. The decision criteria are determined by the local network structure and the corresponding QoS guarantee.

The rest of this chapter is organized as follows. The network model and assumptions made in this research are described in Section 6.2. The optimization formulation used to compute the optimal routing of layered data transmissions is derived in Section 6.3. A set of equations pertaining the reliability of CNC encoded flows is derived and the QoS-aware CNC decision is presented in Section 6.4. The performance of the proposed QoS-aware routing scheme is evaluated in Section 6.5 using numerical experiments under random network topologies and different traffic conditions. Concluding remarks are given in Section 6.6.

6.2 System Model and Problem Description

6.2.1 Network Model

We model a multi-hop wireless network as a directed graph $G(N, E)$, where N and E are the sets of nodes and bidirectional links in the network, respectively. There are several unicast sessions in the network. Each session is defined by a unique source-destination pair. Let s and d denote source and destination nodes of an arbitrary unicast session, respectively. Table 6.1 summarizes the notations used in this chapter.

The link conveying data from node a to node b is denoted by (a, b) . In general, a wireless link connecting any pair of nodes is bidirectional. However, we can represent a bidirectional link using two directed links having opposite flow directions. For example, a bidirectional link between node a and node b can be split to two links, namely (a, b) and (b, a) , which may have different loss characteristics.

Table 6.1: Summary of notations for QoS-aware routing in multi-hop wireless networks

$G(N, E)$: directed graph that represents a multi-hop wireless network
N	: set of nodes in the network
E	: set of links in the network
(a, b)	: link conveying data from node a to node b
$t(l)$: transmitter node of link l
$r(l)$: receiver node of link l
$T_O(n)$: set of outgoing links of node n
$T_I(n)$: set of incoming links of node n
Γ	: set of all pairs of source and destination nodes in the network
(s, d)	: pair of source node s and destination node d of an arbitrary unicast session
c_l	: normalized capacity of link l
p_l	: probability of packet loss of link l
$\overline{M}^{(s,d)}$: number of the original layers of data transmitted by (s, d)
$L_i^{(s,d)}$: i^{th} layer of (s, d)
$r_i^{(s,d)}$: transmission rate of i^{th} layer of (s, d)
$I_{\overline{M}^{(s,d)}}$: set of layer indices of (s, d) , where $I_{\overline{M}^{(s,d)}} = \{0, 1, 2, \dots, \overline{M}^{(s,d)} - 1\}$
\bar{t}	: normalized transmission rate
$M^{(s,d)}$: number of sublayers of (s, d) , where $M^{(s,d)} = \sum_{i \in I_{\overline{M}^{(s,d)}} r_i^{(s,d)} / \bar{t}$
M	: maximum number of sublayers that a source send to a destination in a network
$L_i^{(s,d)}$: the i^{th} sublayer of (s, d)
I_M	: set of sublayer indices, where $I_M = \{0, 1, 2, \dots, M - 1\}$
$P_i^{(s,d)}$: probability of a successful packet transmission for L_i of (s, d) , called the reliability
$f_{l,i}^{(s,d)}$: 0-1 variable that indicates whether or not link l is used to transmit a packet of L_i for (s, d)
$R_i^{(s,d)}$: set of links used to transmit packets of sublayer L_i from source s to destination d
$x_i^{(s,d)}$: 0-1 variable that indicates whether or not packets of sublayer L_i are transmitted from source s to destination d
$\kappa_i^{(s,d)}$: information value of $L_i^{(s,d)}$, used to prioritize data sublayers
$q_i^{(s,d)}$: QoS requirement of sublayer $L_i^{(s,d)}$
J	: set of indices for all independent sets
Z^j	: set of parameters indicating the links that can be activated at the same time according to the j^{th} independent set
z_l^j	: variable that indicates whether or not link l can be activated in the j^{th} independent set
a_j	: activation time fraction of the j^{th} independent set in each time slot
γ	: tuning parameter of the alternative objective function, where $0 < \gamma < 1$

Alternatively, for link $l \in E$, let $t(l)$ and $r(l)$ be the transmitter and receiver nodes of link l , respectively. For each node $n \in N$, let $T_O(n) = \{l \in E | n = t(l)\}$ and $T_I(n) = \{l \in E | n = r(l)\}$ be the sets of outgoing and incoming links of node n , respectively. Let Γ be the set of all source and destination pairs in the network. In other words, $(s, d) \in \Gamma$ denotes a unicast session.

Each link has a normalized positive integral capacity or transmission rate denoted by

c_l . A normalized unit capacity can be translated into bits per second. The probability of a packet loss of link l is denoted by p_l , where $0 \leq p_l \leq 1$. To obtain p_l , each node periodically broadcasts probes containing the number of received probes from each neighboring node. The number of received probes is calculated at the last T time interval in a sliding-window basis. Then, each node computes the p_l by using the delivery ratio of the probes sent on the forward and reverse directions. This approach is also used to obtain expected transmission count (ETX), which is a basic routing metric in wireless mesh networks [23]. There are also other techniques in literature that mentioned how to measure the packet loss rate [101, 102].

Each $(s, d) \in \Gamma$ transmits $\overline{M}^{(s,d)}$ original layers of data, where $L_{\bar{i}}^{(s,d)}$ is the \bar{i}^{th} layer with transmission rate $r_{\bar{i}}^{(s,d)}$. Let the set of layer indices of each (s, d) be $I_{\overline{M}^{(s,d)}}$, where

$$I_{\overline{M}^{(s,d)}} = \{0, 1, 2, \dots, \overline{M}^{(s,d)} - 1\}.$$

To generalize the layered scheme, we decompose each original layer $L_{\bar{i}}^{(s,d)}$ into several sublayers with the same transmission rate based on $r_{\bar{i}}^{(s,d)}$. Let \bar{t} be equal to one normalized unit. Then, (s, d) has $M^{(s,d)}$ sublayers, where

$$M^{(s,d)} = \sum_{\bar{i} \in I_{\overline{M}^{(s,d)}}} \frac{r_{\bar{i}}^{(s,d)}}{\bar{t}}.$$

Let M represent the maximum number of sublayers that a source sends to a destination in the network, *i.e.*,

$$M = \max_{(s,d) \in \Gamma} (M^{(s,d)}).$$

Therefore, each (s, d) has up to M sublayers, where $L_{\bar{i}}^{(s,d)}$ is the \bar{i}^{th} sublayer with the same common transmission rate \bar{t} , so that network coding can be applied across heterogeneous unicast sessions. Let the set of sublayer indices for all (s, d) be I_M , where

$$I_M = \{0, 1, 2, \dots, M - 1\}.$$

6.2.2 QoS Guarantee

Definition 1: The QoS guarantee for $(s, d) \in \Gamma$ and $i \in I_M$ is a lower bound of the probability that source s can transmit a packet of sublayer $L_i^{(s,d)}$ to destination d successfully.

The probability of a successful packet transmission for $L_i^{(s,d)}$, called the reliability and

denoted by $P_i^{(s,d)}$, can be expressed as

$$P_i^{(s,d)} = \prod_{l \in E} (1 - p_l)^{f_{l,i}^{(s,d)}}, \quad (6.1)$$

where $f_{l,i}^{(s,d)}$ indicates whether or not link $l \in E$ is used to transmit a packet of $L_i^{(s,d)}$. If it is used, $f_{l,i}^{(s,d)} = 1$. Otherwise, $f_{l,i}^{(s,d)} = 0$.

6.3 Optimal Path Selection for Layered Unicast

In this section, we describe a method for selecting an optimal set of paths to transmit layered data for all unicast sessions. The selection is constrained by the QoS guarantee of each data layer and by wireless link scheduling. The definition of an optimal set of paths is given below.

Definition 2: A path for $(s,d) \in \Gamma$ and $i \in I_M$ is a set of links, denoted by $R_i^{(s,d)}$, used to transmit packets of sublayer $L_i^{(s,d)}$. An optimal set of paths is such that $R_i^{(s,d)}$, $(s,d) \in \Gamma$, $i \in I_M$, maximize the objective function under a set of constraints.

We discuss the objective function as well as the set of constraints in the following subsections.

6.3.1 Objective Function

Let $x_i^{(s,d)}$ be the variable indicating whether or not packets of sublayer $L_i^{(s,d)}$ are transmitted with QoS guarantee. If the sublayer is transmitted with QoS guarantee, $x_i^{(s,d)} = 1$. Otherwise, $x_i^{(s,d)} = 0$. Moreover, to prioritize data sublayers, we define the information value of each sublayer as $\kappa_i^{(s,d)}$, where $\kappa_i^{(s,d)} \geq \kappa_j^{(s,d)}$, when $i < j$. This means that, the lower a data sublayer, the higher its priority. The throughput of sublayer $L_i^{(s,d)}$ is the product of the reliability $P_i^{(s,d)}$ and normalized transmission rate \bar{t} .

One of our objectives is to maximize the total throughput while taking the reliability into account. The information value of the sublayer $L_i^{(s,d)}$, $\kappa_i^{(s,d)}$, is used to provide priorities among different sublayers. Sublayers from the same original layer will have the same information value. Specifically, $\kappa_i^{(s,d)} = M - \bar{i}$, where the sublayer $L_i^{(s,d)}$ is from the \bar{i} -th layer. Consequently, $\kappa_i^{(s,d)} = \kappa_j^{(s,d)}$ if both $L_i^{(s,d)}$ and $L_j^{(s,d)}$ are from the same original layer. The concept of information value is demonstrated in Figure 6.1.

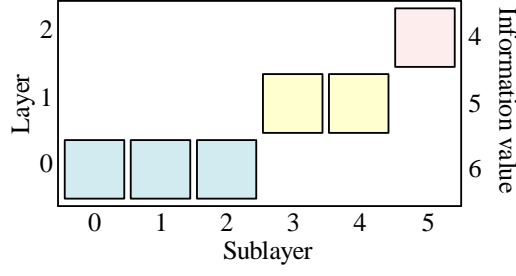


Figure 6.1: An example of defining the information value of each sublayer based on its original layer.

Furthermore, we attempt to reduce the channel use by minimizing the path length for each $R_i^{(s,d)}$ since a shorter path can result in a smaller number of transmissions used for each flow, leading to more efficient channel utilization and shorter delay in wireless networks. Based on the above discussion, we first select the following objective function:

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} x_i^{(s,d)} \log(P_i^{(s,d)} \bar{t}). \quad (6.2)$$

We use the logarithmic throughput in (6.2) since a sum of logarithmic utility functions ensures proportional fairness. To avoid nonlinear optimization which demands higher computational complexity, we can maximize the following equivalent function

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \{\log P_i^{(s,d)} + x_i^{(s,d)} \log \bar{t}\}, \quad (6.3)$$

which can be solved by linear optimization. The equivalence between these two objective functions is stated and proved as Theorem 1. The objective function in (6.3) can be rewritten as

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \{\log \prod_{l \in E} (1 - p_l)^{f_{l,i}^{(s,d)}} + x_i^{(s,d)} \log \bar{t}\}. \quad (6.4)$$

Thus, our objective function takes the following final form:

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \{\sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)} + x_i^{(s,d)} \log \bar{t}\}, \quad (6.5)$$

where $\bar{p}_l = \log(1 - p_l)$.

Relevant properties based on the objective function in (6.5) are summarized in Theorem 1.

Theorem 4. *The objective function in (6.5) has the following properties.*

1. *Maximizing the objective function*

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \{\log P_i^{(s,d)} + x_i^{(s,d)} \log \bar{t}\} \quad (6.6)$$

is equivalent to maximizing the objective function

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} x_i^{(s,d)} \log (P_i^{(s,d)} \bar{t}). \quad (6.7)$$

2. *In an optimal set of paths for each unicast session, for any two paths in this set, a path having a higher reliability transmits packets of either the same or higher information value.*
3. *Given a lossy network, the objective function yields a set of paths that has the minimum number of channel uses in the case of equal link loss probabilities.*

Proof. We prove each property of the objective function in (6.5) as follows.

1. Consider the nonlinear term in (6.7) of the form

$$x_i^{(s,d)} \log (P_i^{(s,d)} \bar{t}). \quad (6.8)$$

We can rewrite it as

$$x_i^{(s,d)} \log (P_i^{(s,d)} \bar{t}) = x_i^{(s,d)} \log P_i^{(s,d)} + x_i^{(s,d)} \log \bar{t}. \quad (6.9)$$

The term $x_i^{(s,d)} \log P_i^{(s,d)}$ can be replaced with $\log P_i^{(s,d)}$ since $\log P_i^{(s,d)} = 0$ when $x_i^{(s,d)} = 0$. More specifically, when $x_i^{(s,d)} = 0$, $f_{l,i}^{(s,d)} = 0$ for all links in $R_i^{(s,d)}$. Since $P_i^{(s,d)} = \prod_{l \in R_i^{(s,d)}} (1 - p_l)^{f_{l,i}^{(s,d)}}$, $P_i^{(s,d)} = 1$ and $\log P_i^{(s,d)} = 0$. Thus, the objective function can be written as

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \{\log P_i^{(s,d)} + x_i^{(s,d)} \log \bar{t}\}. \quad (6.10)$$

2. Consider an optimal solution with the optimal cost denoted by C_m . Suppose that $x_i^{(s,d)} = x_j^{(s,d)} = 1$ and $P_i^{(s,d)} > P_j^{(s,d)}$, but that $\kappa_i^{(s,d)} < \kappa_j^{(s,d)}$. Then, consider an alternative solution whose cost is C'_m and is obtained with the sublayers j and i interchanged. Then,

$$\begin{aligned} C_m - C'_m &= \kappa_i^{(s,d)} \log (P_i^{(s,d)} \bar{t}) + \kappa_j^{(s,d)} \log (P_j^{(s,d)} \bar{t}) \\ &\quad - \kappa_j^{(s,d)} \log (P_i^{(s,d)} \bar{t}) - \kappa_i^{(s,d)} \log (P_j^{(s,d)} \bar{t}) \end{aligned} \quad (6.11)$$

$$= (\kappa_i^{(s,d)} - \kappa_j^{(s,d)}) \times (\log (P_i^{(s,d)} \bar{t}) - \log (P_j^{(s,d)} \bar{t})). \quad (6.12)$$

Since $\kappa_i^{(s,d)} < \kappa_j^{(s,d)}$, the first term in (6.12) is negative. Since $P_i^{(s,d)} > P_j^{(s,d)}$, the second term in (6.12) is positive. It follows that $C_m - C'_m < 0$, contradicting the assumption that C_m is the optimal cost.

3. If a flow of the i^{th} sublayer for (s, d) can be transmitted, the term $x_i^{(s,d)} \log \bar{t}$ in the objective function (6.6) can be ignored. Since $x_i^{(s,d)} = 1$ and $\log \bar{t}$ are constant, the term is invariant with respect to the selected path. Now, consider the first term of (6.6)

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \log P_i^{(s,d)}. \quad (6.13)$$

It can be re-written as

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)}, \quad (6.14)$$

where $\bar{p} = \log(1 - p_l)$.

When \bar{p}_l for all $l \in E$ is equal to a fixed constant, since \bar{p}_l is a negative value, $f_{l,i}^{(s,d)}$ should be set equal to 0 as many times as possible to maximize (6.13). Since $f_{l,i}^{(s,d)} = 0$ refers to the unused link l for $R_i^{(s,d)}$. Thus, we conclude that the maximum objective function refers to the minimum number of channel uses.

□

6.3.2 Flow Conservation Constraint

The considered multi-hop wireless networks is modeled as a network with information flows. Consider information flows for each $(s, d) \in \Gamma$ of the i^{th} sublayer, where $i \in I_M$. The total flow into a particular intermediate node is equal to the total flow out of the node. The flow of sublayer $L_i^{(s,d)}$ from source node s to destination node d is equal to sublayer rate \bar{t} . Thus, the constraint on information flow conservation can be expressed mathematically as

$$\sum_{l \in T_O(n)} \bar{t} f_{l,i}^{(s,d)} - \sum_{l \in T_I(n)} \bar{t} f_{l,i}^{(s,d)} = \begin{cases} \bar{t} x_i^{(s,d)}, & n = s \\ -\bar{t} x_i^{(s,d)}, & n = d \\ 0, & \text{otherwise} \end{cases} \quad (6.15)$$

for all $i \in I_M$, $(s, d) \in \Gamma$, and $n \in N$. Note that, when $x_i^{(s,d)} = 0$, the total flow out of a source or into a destination must be zero.

6.3.3 Reliability Constraint

The QoS requirement of sublayer $L_i^{(s,d)}$ is denoted by $q_i^{(s,d)}$, where $0 \leq q_i^{(s,d)} \leq 1$. Based on (6.1), the constraint on packet transmissions of sublayer $L_i^{(s,d)}$ with QoS guarantee can be expressed as

$$P_i^{(s,d)} = \prod_{l \in E} (1 - p_l)^{f_{l,i}^{(s,d)}} \geq q_i^{(s,d)} \quad (6.16)$$

for all $i \in I_M$ and $(s, d) \in \Gamma$. By taking the logarithm on both sides of (6.16), we obtain

$$\sum_{l \in E} f_{l,i}^{(s,d)} \bar{p}_l \geq \bar{q}_i^{(s,d)}, \quad (6.17)$$

where $\bar{q}_i^{(s,d)} = \log q_i^{(s,d)}$. This constraint demands each path selected by the optimal routing to achieve the QoS requirement based on the reliability consideration.

The choice of the QoS requirement of each data layer, $q_i^{(s,d)}$, is based on its priority and type of application, and is obtained from experiences of end-users. For example, in Voice over IP (VoIP) traffic, the packet loss rate should not exceed 5 percent to not affect the quality significantly. When link qualities of a multi-hop wireless network are in hostile conditions, the original QoS requirements may not be feasible because the proposed optimization framework cannot find feasible transmitting paths guaranteeing the original QoS requirements of those data layers. This infeasibility is, however, common to communication networks. The problem can typically be handled through the process of call admission control (CAC), which we assume to exist but whose details are beyond the scope of our investigation. The ILP problem can be used for network resource allocation in conjunction with CAC.

6.3.4 Wireless Link Scheduling Constraint

Due to the broadcast nature of wireless communications, a transmission of a particular node can affect transmissions of other nodes in its coverage range. Since wireless channels are shared among multiple nodes, a node placed in the transmission and coverage ranges of other nodes may be interfered by simultaneous communications. In this work, we

assume that the wireless interference can be managed by an appropriate channel planning [103, 25]. A receiver node cannot simultaneously receive more than one packet whereas a transmitter node can send no more than one packet at a given time. Therefore, a wireless link scheduling technique is needed to coordinate wireless broadcasting.

A broadcast transmission scheduling technique using the independent set concept was proposed by [30]. An independent set consists of a set of links where no two links share a common end node. In our network model, any pair of links in an independent set must share neither a common transmitter node nor a common receiver node. It is also assumed that broadcasting is achieved using omnidirectional antenna, where the transmission of each packet goes into all directions. The number of all possible independent sets in a given network can be quite large since it grows exponentially with the number of links.

Instead of considering all independent sets, it suffices to consider a family of independent sets whose union can cover all links of the network. The problem of choosing such independent sets for a network can be formulated as a Set Covering Problem (SCP) [104] whose solution can be solved by using either integer linear optimization or greedy algorithm. To give an example as shown in Figure 6.2, instead of considering all feasible independent sets in the network, we can consider the following independent sets: $\{(1,2), (4,5), (3,6)\}$, $\{(1,2), (5,4), (3,6)\}$, $\{(1,2), (4,5), (6,3)\}$, $\{(1,2), (3,4)\}$, $\{(2,1), (3,4)\}$, $\{(2,3), (4,5)\}$, $\{(3,2), (4,5)\}$, $\{(1,2), (4,3)\}$, $\{(2,3), (5,1)\}$, and $\{(2,3), (1,5)\}$.

Let Z^j be a set of parameters indicating the links that can be activated at the same time according to the j^{th} independent set. In particular, $Z^j = \{z_l^j\}_{l \in E}$. If $z_l^j = 1$, link l can be activated; otherwise, link l cannot be activated in the j^{th} independent set. The set of indices for all Z^j is denoted by J . For example, Z^j for independent set $\{(1,2), (4,5), (3,6)\}$ of the network shown in Figure 6.2 is

$$\begin{aligned} Z^j &= \{z_{(1,2)}^j, z_{(2,3)}^j, z_{(3,4)}^j, z_{(4,5)}^j, z_{(5,1)}^j, z_{(1,5)}^j, \\ &\quad z_{(5,4)}^j, z_{(4,3)}^j, z_{(3,2)}^j, z_{(2,1)}^j, z_{(3,6)}^j, z_{(6,3)}^j\} \\ &= \{1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0\}. \end{aligned} \tag{6.18}$$

To achieve time sharing according to link capacities, each selected independent set Z^j will be activated for a time fraction a_j in each transmission time slot. The value of a_j is $0 \leq a_j \leq 1$ for $j \in J$, and $\sum_{j \in J} a_j = 1$. Then, the wireless link scheduling constraints can

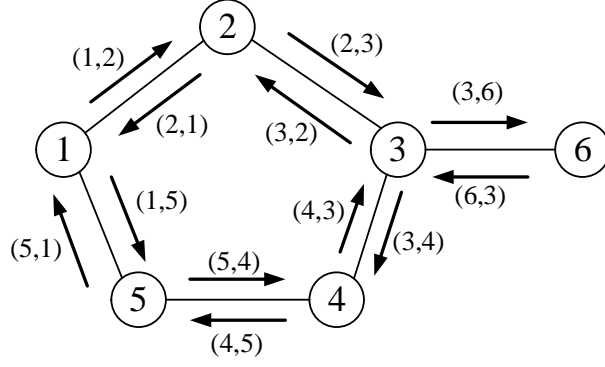


Figure 6.2: An exemplary wireless network.

be expressed mathematically as

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} f_{l,i}^{(s,d)} \leq \sum_{j \in J} c_l z_l^j a_j \quad (6.19)$$

for all $l \in E$, and

$$\sum_{j \in J} a_j = 1. \quad (6.20)$$

6.3.5 Problem Formulation: A Summary

Based on the above discussion, we can formulate the optimal path selection problem as a linear optimization using the objective function in (6.5) with constraints of the flow conservation in (6.15), the reliability in (6.16), and the wireless link scheduling in (6.19) and (6.20). The overall problem is summarized below.

Maximize

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \left\{ \sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)} + x_i^{(s,d)} \log \bar{t} \right\} \quad (6.21a)$$

subject to

$$\sum_{l \in T_O(n)} \bar{t} f_{l,i}^{(s,d)} - \sum_{l \in T_I(n)} \bar{t} f_{l,i}^{(s,d)} = \begin{cases} \bar{t} x_i^{(s,d)}, & n = s \\ -\bar{t} x_i^{(s,d)}, & n = d \\ 0, & \text{otherwise} \end{cases} \quad (6.21b)$$

$$\forall i \in I_M, \forall (s,d) \in \Gamma, \forall n \in N,$$

$$\sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)} \geq \bar{q}_i^{(s,d)}, \quad \forall i \in I_M, \forall (s,d) \in \Gamma, \quad (6.21c)$$

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} f_{l,i}^{(s,d)} \leq \sum_{j \in J} a_j c_l z_l^j, \quad \forall l \in E, \quad (6.21d)$$

$$\sum_{j \in J} a_j = 1, \quad (6.21e)$$

$$x_i^{(s,d)} \geq x_{i+1}^{(s,d)}, \forall i \in I_{M-1}, \forall (s,d) \in \Gamma, \quad (6.21f)$$

$$f_{l,i}^{(s,d)} \in \{0, 1\}, \forall i \in I_M, \forall (s,d) \in \Gamma, \forall l \in E, \quad (6.21g)$$

$$x_i^{(s,d)} \in \{0, 1\}, \forall i \in I_M, \forall (s,d) \in \Gamma, \quad (6.21h)$$

$$0 \leq a_j \leq 1, \forall j \in J, \quad (6.21i)$$

where $I_{M-1} = \{0, 1, \dots, M-2\}$.

Specifically, we can explain each constraint as follows. Constraint (6.21b) is the flow conservation constraint. Constraint (6.21c) is the reliability constraint. Constraint (6.21d) and (6.21e) are the wireless link scheduling constraints. Constraint (6.21f) is the layered data constraint. A transmission path of a higher sublayer will be chosen only if a transmission path of a lower sublayer has been selected. Constraints (6.21g), (6.21h), and (6.21i) are the feasible values of $f_{l,i}^{(s,d)}$, $x_i^{(s,d)}$, and a_j , respectively.

Searching for a set of paths that maximize the throughput of each layered unicast session requires high computational complexity because all feasible links must be considered. To reduce the complexity of the problem, the objective function in (6.21a) is modified as

$$\sum_{(s,d) \in \Gamma} \sum_{i \in I_M} \kappa_i^{(s,d)} \{x_i^{(s,d)} \log \bar{t} - \sum_{l \in E} \gamma f_{l,i}^{(s,d)}\}, \quad (6.22)$$

where γ is a tuning parameter between zero and one.

The objective function in (6.22) gives a sub-optimal solution with respect to the original objective function in (6.21a). The objective function in (6.22) may not satisfy the second property of the original objective function since it does not take into account the successful transmission probability of each link. However, the third property of the original objective function still holds, i.e., the objective function in (6.22) provides shortest paths in terms of hop distances satisfying both the transmission rate and QoS requirement of each sublayer, which can be proven by using the similar approach to the third property of Theorem 1.

The constrained linear optimization is solved to obtain an optimal set of paths $R_i^{(s,d)}$, $(s,d) \in \Gamma, i \in I_M$, as defined in Definition 2. An optimal solution to the problem can be obtained by various mathematical programming tools. We select CoinMP [105], which is a C-API library that supports most of the functionality of CLP (Coin LP), CBC (Coin

Branch-and-Cut), and CGL (Cut Generation Library) projects, to be the solver for linear programming. This is the first step of the proposed QoS-aware routing scheme. These obtained optimal paths are inputs to the second step of the proposed QoS-aware routing scheme as described in the next section.

6.4 QoS-Aware CNC for Layered Unicasts

The CNC Establishment (CNCE) protocol is presented in this section, and is used to decide whether or not CNC will be performed on different unicast pairs at intermediate nodes. The decision criterion is derived based on the QoS requirement of transmitted layered data.

6.4.1 Three Basic Local Structures

We consider three local structures for the application of CNC, called the A-B, Y, and X structures, as shown in Figure 6.5. The dashed and regular arrows shown in Figure 6.5 represent the overhearing and direct transmissions, respectively. These three local structures were partly used in [30, 57, 59]. They serve as the basis in typical networks.

For the A-B structure, CNC is employed at node C , which combines each pair of packets received from node A and node B , and then broadcasts the combined packet back to those nodes. Transmission delay and energy consumption in a shared network can be reduced at the cost of lower reliability of transmitted data. This is because, to receive the transmitted data correctly at nodes A and B , all data packets involved in the network coding operation must be successfully received by node C , while the network coded packets at node C must be successfully received by nodes A and B .

For the Y structure, there are two unicast flows: 1) from node A to node B and 2) from node B to node D . CNC is conducted at node C . In particular, node D receives a packet transmitted by node A via opportunistic listening. Node C encodes each pair of packets received from node A and node B , and then broadcasts the network coded packet to nodes B and D simultaneously.

For the X structure, there are two unicast flows: 1) from node A to node B and 2) from node E to node D . Network coding is operated at node C . The coded data packet

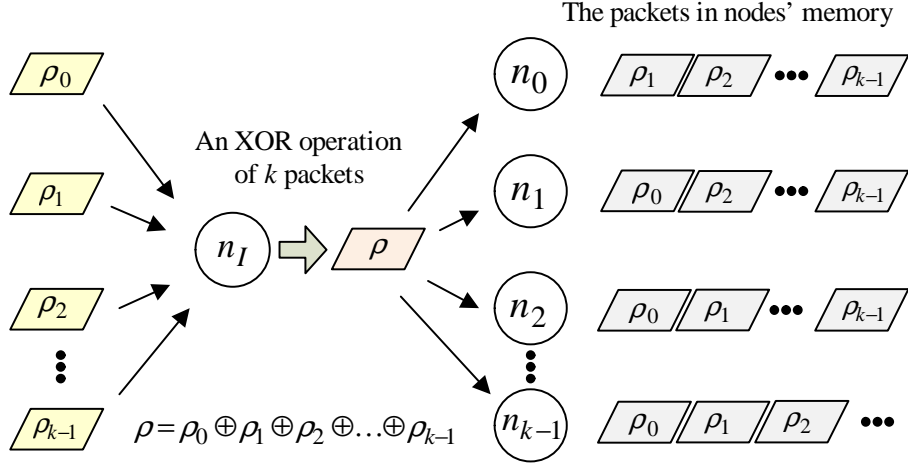


Figure 6.3: Coding rules with XOR (\oplus) operation.

is then broadcast to nodes D and B . Transmission delays and energy consumptions of these unicast flows are reduced since the number of channel uses is reduced due to CNC. However, the reliability in transmitted data deteriorates due to the dependency on required packets in data decoding at destination nodes.

These local structures are potentially embedded in general random topologies. We provide numerical results in terms of the transmission reliability for general random topologies that perform CNC using these three local structures in Section 6.4.3.

6.4.2 Coding Rules and Opportunities

In this subsection, we discuss the coding rules and opportunities of CNC. Consider k packets $\rho_0, \rho_1, \rho_2, \dots, \rho_{(k-1)}$ that are independent of one another and are on their own flows traversing a common intermediate node. The packets $\rho_0, \rho_1, \rho_2, \dots, \rho_{(k-1)}$ leave a common intermediate node and travel to nodes $n_0, n_1, n_2, \dots, n_{(k-1)}$, respectively. At the intermediate node, inter-flow coding using the XOR (\oplus) operation forms the coded packet $\rho = \rho_0 \oplus \rho_1 \oplus \rho_2 \oplus \dots \oplus \rho_{(k-1)}$. Next, the coded packet ρ is broadcast to nodes $n_0, n_1, n_2, \dots, n_{(k-1)}$. The coded packet is valid and can be decoded at each n_i only if n_i has received packets ρ_j for all $j \in \{0, 1, 2, \dots, k-1\}$ and $j \neq i$. These coding rules are demonstrated in Figure 6.3.

The next node n_i can have all mentioned packets ρ_j with the following two conditions:

1. Packet ρ_j belongs to a flow that has traveled through n_i , where n_i keeps the packet

in its memory for a period of time for the purpose of CNC. This situation, known as the non-opportunistic listening CNC operation, is applicable to the A-B structure

2. Node n_i receives packet ρ_j by overhearing the packet from a transmission of its adjacent node. For the network coding operation, node n_i keeps the packet for later decoding of an encoded packet. The operation, called the opportunistic listening CNC operation, is used by the X structure.

The Y structure conducts both non-opportunistic and opportunistic listening CNC operations. In what follows, we adopt these conditions as the coding rules and opportunities for the CNC establishment.

Note that the proposed coding rules may not be optimal in terms of the number of channel uses in some network topologies. Other COPE structures, which consist of more than two information flows and accordingly establish more complex encoding/decoding structures than ours, have different coding rules and potentially provide more reduction in the number of channel uses. However, these complex COPE structures are rarely seen in practical networks since they require overlapping transmission ranges of more nodes to form their structures compared to the basic local structures in our work.

6.4.3 Reliability Computation

In this section, the effects of performing CNC in lossy wireless networks on the reliability of layered data transmissions are investigated. A terminal node in each CNC structure can reproduce its desired packet if it has the coded packet and all the other involved non-coded packets. In addition, to encode a packet successfully for a unicast flow that passes node A and then node B , an intermediate node needs all required non-coded packets from other unicasts travelling along links that do not belong to path $R_i^{(A,B)}$.

Statement 1: For a flow on path $R_i^{(A,B)}$ that is associated with the CNC structure, the participating links to the flow on path $R_i^{(A,B)}$ are the links that are not on $R_i^{(A,B)}$ and carry either a non-coded packet to be used for encoding at an intermediate node or a non-coded packet to be used for decoding a coded packet at terminal node B . A participating link can be a link on the flow of another path cooperating with the flow on path $R_i^{(A,B)}$, or a link used in opportunistic listening.

We examine the participating links of the basic CNC structures below.

- For the A-B structure, the participating link to the flow on path $R_i^{(A,B)}$ is e_{BC} since node C needs a non-coded packet from node B to generate the coded packet, which is obtained by performing the XOR operation of a packet from node A and a packet from node B . Similarly, we can derive the participating link to the flow on path $R_i^{(B,A)}$.
- For the Y structure, the participating link to the flow on path $R_i^{(A,B)}$ is e_{BC} since node C needs a packet from node B to generate the coded packet. On the other hand, the participating links to the flow on path $R_i^{(B,D)}$ are e_{AC} and e_{AD} since both nodes C and D need packets from node A to generate and decode the coded packet, respectively. Note that node D can receive a packet from node A through opportunistic listening.
- For the X structure, e_{EC} and e_{EB} are the participating links to the flow on path $R_i^{(A,B)}$ since node C needs a packet from node E on e_{EC} , while node B needs a packet from node E on e_{EB} to generate the coded packet and to decode the coded packet, respectively. Similarly, one can derive the participating links to the flow on path $R_i^{(E,D)}$, which are e_{AC} and e_{AD} .

Statement 2: Let ξ be the set of participating links to the flow traveling along sub-path $R_i^{(A,B)}$ of $R_i^{(s,d)}$ with CNC performed, the reliability of the flow on $R_i^{(s,d)}$ can be expressed as

$$P_i^{(s,d)} = \left(\prod_{l \in R_i^{(s,d)}} (1 - p_l) \right) \cdot \left(\prod_{l \in \xi} (1 - p_l) \right). \quad (6.23)$$

Note that the probability of a successful packet transmission along path $R_i^{(s,d)}$ has previously been computed in (6.1). When CNC is applied, the reliabilities of the participating links affect the decodability of transmitted data at a terminal node. The expression in (6.23) reckons the probability of a successful packet transmission taking all reliabilities of links in $R_i^{(s,d)}$ and all participating links into account.

We use the A-B structure as an example. When the involved transmission links are lossy, the successful transmission probability of sublayer $L_i^{(s,d)}$ from node A to node B and

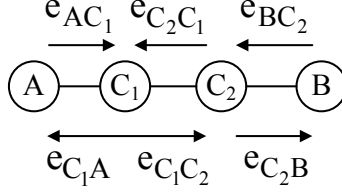


Figure 6.4: An extended A-B structure which has more than one intermediate node.

from node B to node A with CNC at node C can be expressed as

$$P_i^{(A,B)} = (1 - p_{e_{AC}})(1 - p_{e_{CB}})(1 - p_{e_{BC}}), \quad (6.24)$$

and

$$P_i^{(B,A)} = (1 - p_{e_{BC}})(1 - p_{e_{CA}})(1 - p_{e_{AC}}). \quad (6.25)$$

For the extended A-B structure that has two intermediate nodes, *i.e.*, nodes C_1 and C_2 , as shown in Figure 6.4, we can generalize (6.24) to

$$P_i^{(A,B)} = \prod_{\forall l \in R_i^{(A,B)}} (1 - p_l) \cdot (1 - p_{l_{C_n}}), \quad (6.26)$$

where node C_n is the node that performs CNC and l_{C_n} is the incoming link of node C_n in the direction opposite to the outgoing link of node C_n in $R_i^{(A,B)}$.

For the Y structure that has 5 transmission links as shown in Figure 6.5(b), the reliabilities of unicast flows traveling from node A to node B and from node B to node D with CNC at node C can be expressed as

$$P_i^{(A,B)} = (1 - p_{e_{AC}})(1 - p_{e_{CB}})(1 - p_{e_{BC}}), \quad (6.27)$$

and

$$P_i^{(B,D)} = (1 - p_{e_{BC}})(1 - p_{e_{CD}})(1 - p_{e_{AC}})(1 - p_{e_{AD}}), \quad (6.28)$$

respectively. They can be generalized as

$$P_i^{(A,B)} = \prod_{\forall l \in R_i^{(A,B)}} (1 - p_l) \cdot (1 - p_{e_{BC}}), \quad (6.29)$$

and

$$P_i^{(B,D)} = \prod_{\forall l \in R_i^{(B,D)}} (1 - p_l) \cdot (1 - p_{e_{AC}})(1 - p_{e_{AD}}), \quad (6.30)$$

where e_{AC} is the incoming link to node C on $R_i^{(A,B)}$, e_{BC} is the incoming link to node C on $R_i^{(B,D)}$, and e_{AD} is the incoming link to node D from any node upstream of node C on

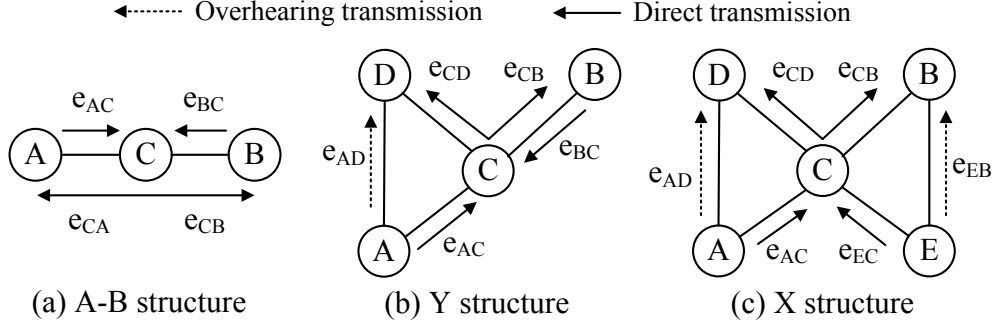


Figure 6.5: Three basic local structures for the CNCE protocol: (a) the A-B structure, (b) the Y structure and (c) the X structure. The dashed and regular arrows represent the overhearing and direct transmissions, respectively.

$$R_i^{(A,B)}.$$

For the X structure that has 6 links as shown in Figure 6.5(c), the reliabilities of unicast flows traveling from node A to node B and from node E to node D with CNC at node C are expressed as

$$P_i^{(A,B)} = (1 - p_{e_{AC}})(1 - p_{e_{CB}})(1 - p_{e_{EC}})(1 - p_{e_{EB}}), \quad (6.31)$$

and

$$P_i^{(E,D)} = (1 - p_{e_{EC}})(1 - p_{e_{CD}})(1 - p_{e_{AC}})(1 - p_{e_{AD}}). \quad (6.32)$$

We can generalize (6.31) and (6.32) for two unicast flows that join the X structure as

$$P_i^{(A,B)} = \prod_{\forall l \in R_i^{(A,B)}} (1 - p_l) \cdot (1 - p_{e_{EC}})(1 - p_{e_{EB}}), \quad (6.33)$$

and

$$P_i^{(E,D)} = \prod_{\forall l \in R_i^{(E,D)}} (1 - p_l) \cdot (1 - p_{e_{AC}})(1 - p_{e_{AD}}), \quad (6.34)$$

where e_{AC} and e_{EC} are the incoming links to node C on $R_i^{(A,B)}$ and $R_i^{(E,D)}$, e_{AD} is the incoming link to node D from any node upstream of A on $R_i^{(A,B)}$, and e_{EB} is the incoming link to node B from any node upstream of E on $R_i^{(E,D)}$.

6.4.4 CNCE Protocol

Different unicast flows can be combined to reduce the use of network resources when there are bottlenecks in the network. Our goal is to apply CNC as much as possible while guaranteeing the QoS of unicast flows of different data sublayers. However, if combining

unicast flows for CNC leads to a violation of the QoS requirement, CNC will not be performed and unicast flows will be separately transmitted.

The search for CNC structures is executed by the central controller. The optimal paths obtained from Section 6.3 are investigated over all links to find A-B, Y, and X structures. The central controller detects each CNC structure by examining whether a group of links match with the considered CNC structure. If a group of links match the underlying CNC structure, these links must convey two unicast flows having the same transmission rate.

The CNCE protocol can be executed step by step as follows.

Stage 1: CNCE for the A-B structure

Step 1: Find all A-B structures in $R_i^{(s,d)}$ for all $i \in I_M$ and for all $(s, d) \in \Gamma$.

Step 2: For each A-B structure identified in Step 1, we find two unicast flows from two pairs of $(s, d) \in \Gamma$ that go through this A-B structure.

Step 3: For each intermediate node, denoted by C_n , where $n = 1, 2, \dots, \Phi$ and Φ is the number of intermediate nodes in the A-B structure, we use (6.26) to compute the reliability of applying CNC at this node. Select an intermediate node C_n that satisfies the QoS requirements of two unicast flows obtained in Step 2. If there is more than one intermediate node that can satisfy the QoS requirements with CNC, choose the node with the best QoS. At the selected node, perform CNC on these two unicast flows. Otherwise, CNC will not be performed and these two unicast flows will be transmitted separately.

Stage 2: CNCE for the Y structure

Step 4: In $R_i^{(s,d)}$ for all $i \in I_M$ and for all $(s, d) \in \Gamma$, find all Y structures that have links not in the A-B structures identified in Step 1.

Step 5: For each Y structure in Step 4, use (6.29) and (6.30) to compute the reliabilities of two unicast flows. If the unicast flows transverse through the previous A-B structure, the reliabilities of (6.29) or (6.30) will be modified by multiplying the successful transmission probability of the link l_{C_n} from the A-B structure. This modification is needed since the reliability of the current CNC in the Y structure relies on the reliability of the CNC in the A-B structure.

Step 6: If the computed reliabilities from Step 5 of the Y structure satisfy the QoS requirements of these two unicast flows, perform CNC on two unicast flows. Otherwise,

these two unicast flows will be transmitted separately.

Stage 3: CNCE for the X structure

Step 7: In $R_i^{(s,d)}$ for all $i \in I_M$ and for all $(s, d) \in \Gamma$, find all X structures that have links not in the A-B structure and the Y structure as identified in Steps 1 and 4.

Step 8: For each X structure in Step 7, use (6.33) and (6.34) to compute the reliabilities of two unicast flows. If the unicast flows travel through the previous A-B structure, the reliabilities of (6.33) or (6.34) will be modified by multiplying the successful transmission probability of the link l_{C_n} from the A-B structure. If the unicast flows travel through the previous Y structure, the reliabilities of (6.33) or (6.34) will be modified by multiplying $(1 - p_{e_{BC}})$ or $(1 - p_{e_{AC}})(1 - p_{e_{AD}})$ in the Y structure, depending on the unicast flows. If the unicast flow travels through the A-B as well as the Y structures, both modifications are adopted.

Step 9: If the computed reliabilities from Step 8 of the X structure satisfy QoS requirements of these two unicast flows, perform CNC on the two unicast flows. Otherwise, they will be separately transmitted.

The computed reliabilities at the end of Step 9 yield the final reliability of sublayer $L_i^{(s,d)}$. We can infer from this reliability that all $L_i^{(s,d)}$ have the QoS guarantees since their end-to-end successful transmission probability are equal to or greater than their QoS requirements.

6.5 Experimental Results

In this section, we compare the performance of the following three routing schemes:

- Shortest Path Routing (SP-R), which was considered in [30, 57, 59, 54];
- QoS-aware layered unicast routing (QoSSP-R) as presented in Section 6.3;
- QoS-aware layered unicast routing with an alternative objective function (6.22) (QoS-R), the tuning parameter is set to 0.01 (i.e., $\gamma = 0.01$);

and their enhanced versions by incorporating our CNCE algorithm, which is presented in Section 6.4.4. Thus, we can evaluate how CNC affects these routing schemes.

Table 6.2: Distance thresholds for different transmission data rates.

Data rate (Mbps)	Normalized rate (Unit)	Received power (dBm)	Distance (m)
6	12	-82	100.0
12	24	-81	94.4
18	26	-79	84.1
24	48	-77	75.0
36	72	-74	63.0
48	96	-70	50.1
54	108	-66	39.8

6.5.1 Experimental Set-Up

We simulate the three routing schemes, SP-R, QoS-R, and QoSSP-R, and their modified schemes, SP-R w/ CNCE, QoS-R w/ CNCE, and QoSSP-R w/ CNCE, on random network topologies. We use the igraph library [95], which is a free software package to generate and simulate undirected and directed graphs of complex network research. The node positions are placed randomly in a square whose side length is 400 meters. The transmission range of each node is set to 100 meters.

The transmission rate is set depending on the received power threshold and the corresponding maximal distance based on the IEEE 802.11a standard [30]. We set the transmission rate from 6, 12, 18, 24, 36, 48, up to 54 Mbps. For the rate 6 Mbps, we set the maximal distance of 100 meters. Then, we calculate higher transmission rates for shorter distances corresponding to the path loss model $P_r = \alpha P_t / d^4$, where P_r , P_t , α , and d represent the received power, the transmitted power, the path loss coefficient used in the simulation, and the distance measured from the transmitter to the receiver, respectively. A normalized unit of link capacity is set to 512 kbps. The relationship between the transmission data rate and the received power is shown in Table 6.2.

While more accurate path loss models can be derived from complex analytical models or from measurements where system specifications such as the locations of access points must be known, a simplified path loss model is used because it can sufficiently capture the essence of signal propagation for the purpose of data delivering as well as interference consideration. Note that the proposed CNCE algorithm can also be applied when other path loss models are assumed.

We perform numerical experiments by adjusting one of the following three parameters:

- successful data transmission probabilities of links;
- node densities;
- and traffic demands.

Assume that there are no packet retransmissions. We evaluate the performance of each routing scheme by using three metrics: 1) total throughput of a network; 2) number of channel uses; and 3) throughput per channel use. The total throughput of a network is a sum of transmission rates of all sublayers $L_i^{(s,d)}$ that satisfy their QoS requirements. In our experiments, a sink node discards the sublayers that cannot satisfy their QoS requirements as well as their dependencies. The number of channel uses is a sum of links of all the paths used to transmit all layered unicast flows in each network. It reflects the wireless channel utilization of each routing scheme. Finally, the throughput per channel use is the ratio between the total throughput of a network and the number of channel uses. This metric measures the efficiency of wireless channels in data transmission.

A source-destination (s-d) pair transmits one base layer and two enhancement layers. We set \bar{t} equal to one normalized unit which is 512 kbps. The transmission rates of the base layer, the first enhancement layer, and the second enhancement layer are equal to 2, 1, and 1 units, respectively. We set the QoS requirements, which are successful transmission probabilities, to 0.90, 0.80, and 0.70 for the base layer, the first enhancement layer, and the second enhancement layer, respectively. Therefore, each s-d pair transmits four sublayers, $L_0^{(s,d)}$, $L_1^{(s,d)}$, $L_2^{(s,d)}$, and $L_3^{(s,d)}$. The information values of $L_0^{(s,d)}$, $L_1^{(s,d)}$, $L_2^{(s,d)}$, and $L_3^{(s,d)}$ are set to 4, 4, 3, and 2, respectively. The routing solution of each routing scheme is obtained from the Python programming language [106] together with the PuLP package [107] and the CoinMP solver [105].

6.5.2 Effects of Link Transmission Probabilities

Ten source-destination (s-d) pairs are randomly chosen in 50 randomly selected networks with 15 nodes. The successful data transmission probability of each link is randomly generated, where $Z \leq 1 - p_l \leq 1$ and $Z \in \{0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96\}$. Note that the x-axis of all the result plots specifies the value of Z .

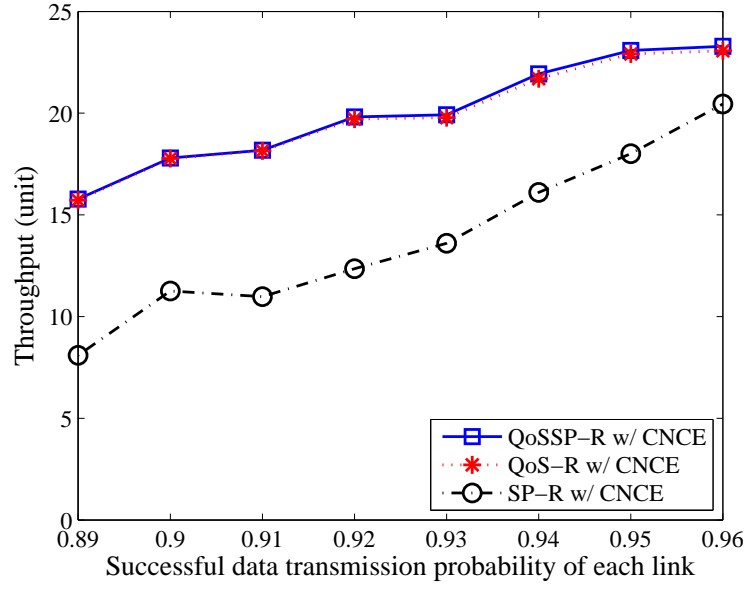


Figure 6.6: Comparison of the throughput for various link qualities in the networks having 15 nodes.

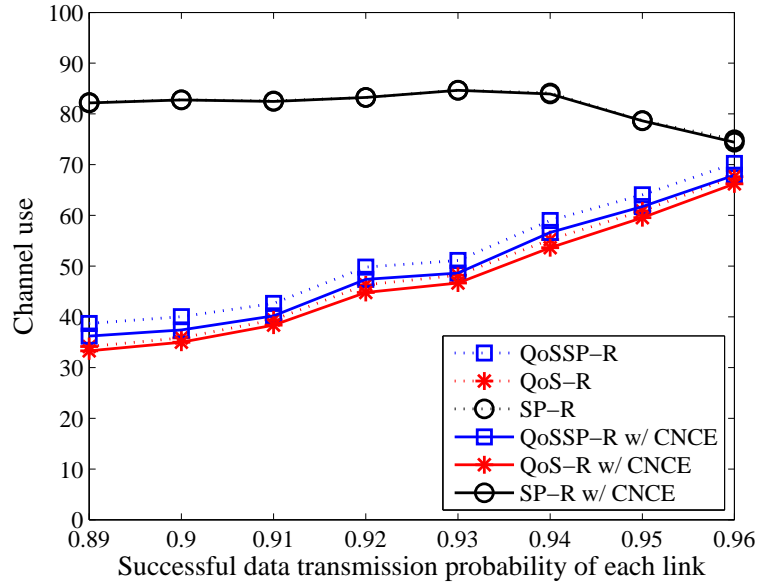


Figure 6.7: Comparison of the number of channel uses for various link qualities in the networks having 15 nodes.

Figure 6.6 shows the throughputs of the considered routing schemes with CNCE algorithm as a function of the successful packet transmission probability. From the results, QoS-SP-R w/ CNCE gives the highest throughput among all routing schemes. In addition, QoS-SP-R w/ CNCE gives significantly better results than SP-R at all cases of successful

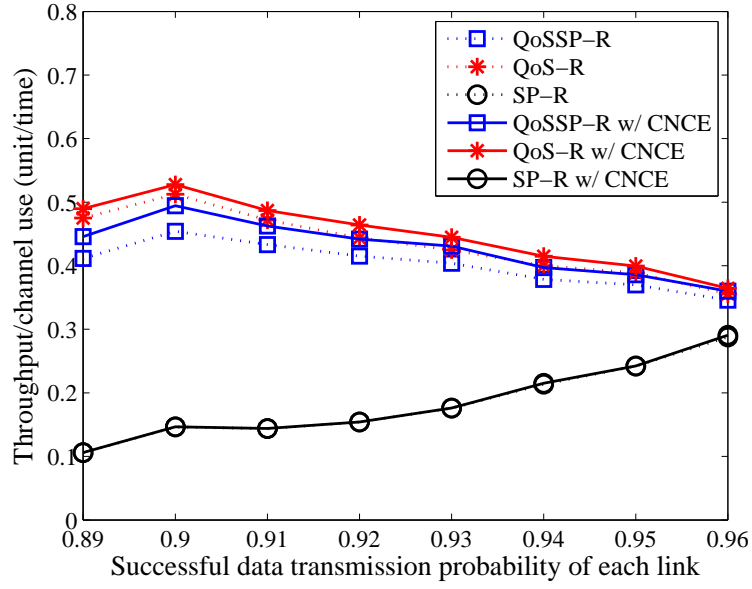


Figure 6.8: Comparison of the throughput per channel use for various link qualities in the networks having 15 nodes.

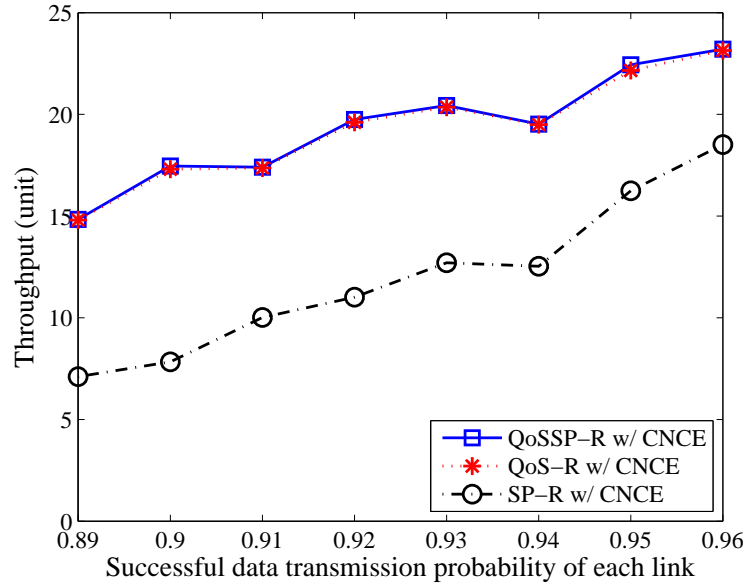


Figure 6.9: Comparison of the throughput for various link qualities in the networks having 20 nodes.

packet transmission probabilities. The throughput gain is more significant at low successful packet transmission probabilities because SP-R may select paths without guaranteeing QoS requirements. The throughput gain of QoSSP-R w/ CNCE over its suboptimal counterpart, QoS-R w/ CNCE, is modest. However, QoSSP-R w/ CNCE achieves a throughput

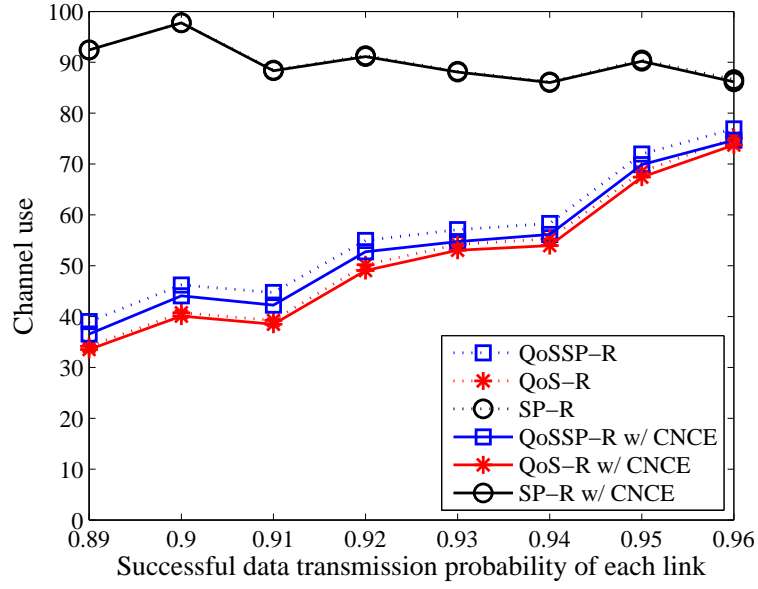


Figure 6.10: Comparison of the number of channel uses for various link qualities in the networks having 20 nodes.

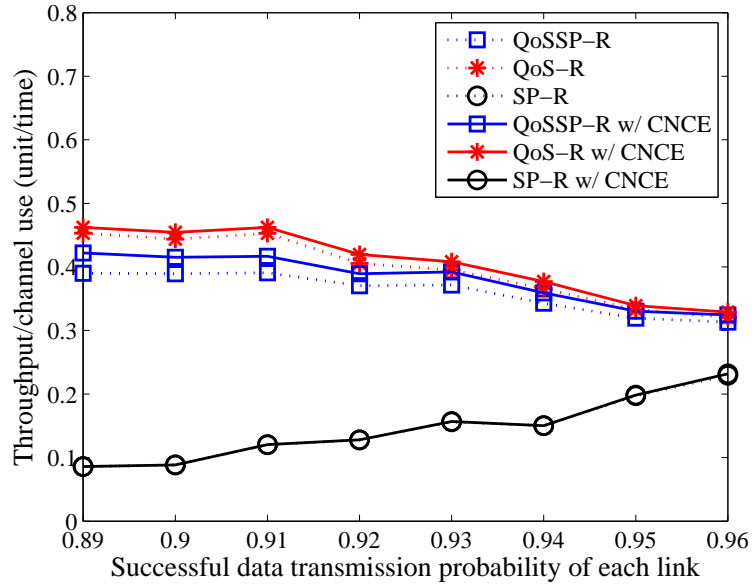


Figure 6.11: Comparison of the throughput per channel use for various link qualities in the networks having 20 nodes.

gain over the QoS-R w/ CNCE since QoSSP-R w/ CNCE selects paths with the highest end-to-end transmission reliability, whereas QoS-R w/ CNCE merely chooses paths that satisfy the QoS requirements. Obviously, transmission paths satisfying the QoS requirements may not give the highest reliability. The throughput of QoSSP-R w/ CNCE is close

to that of QoS-R w/ CNCE. We can conclude from the results that QoS-R w/ CNCE could be an effective alternative to QoSSP-R w/ CNCE if maximizing the throughput is our objective.

Next, Figure 6.7 evaluates the performances of routing schemes in terms of the number of channel uses. The numbers of channel uses of QoSSP-R and QoS-R are significantly less than that of SP-R at all link qualities. The number of channel uses from SP-R is the highest at all link qualities although its achievable throughput increases as a function of the successful transmission probability. In other words, SP-R has the lowest efficiency of channel utilization, especially at low link qualities. QoS-R has a lower number of channel uses than QoSSP-R both with and without CNCE algorithm. QoSSP-R selects paths with the highest transmission reliability regardless of the number of links used to transmit bitstreams whereas QoS-R chooses the shortest paths that satisfy the QoS requirements.

The importance of CNCE algorithm is also scrutinized with the considered routing schemes. First, there is not much difference in terms of the number of channel uses between SP-R and SP-R w/ CNCE. When SP-R is a routing scheme, the selected transmission paths of SP-R generally have low reliabilities. Applying CNCE algorithm will further deteriorate transmission reliabilities and QoS guarantees. Therefore, CNC structures are rarely formed to enhance channel utilization in this environment. However, the gain from using CNCE algorithm can be seen in both QoSSP-R and QoS-R. The number of channel uses of both routing schemes decreases due to CNCE algorithm. In addition, the CNCE algorithm can decrease the number of channel uses for QoSSP-R more than for QoS-R. This comes from the fact that QoSSP-R selects the optimal paths with higher reliabilities than QoS-R. Therefore, the CNCE algorithm has a better chance to establish more CNC structures without breaking QoS requirements.

Figure 6.8 shows the throughput per channel use of all routing schemes. QoS-R w/ CNCE achieves the best throughput per channel use among all routing schemes. Both QoSSP-R w/ CNCE and QoS-R w/ CNCE significantly achieve a better throughput per channel use than SP-R with and without CNCE algorithm in all network environments. Figures 6.9, 6.10, and 6.11 exhibit the throughput, number of channel uses, and throughput per channel use of all routing schemes, when the number of nodes in the simulated network is equal to 20. From the results, the performances of all routing schemes show the same

properties as those for the case of 15 nodes.

We can draw a conclusion from our experiments that QoS-R should be used in transmissions with QoS guarantees. QoS-R gives almost the same throughput as QoSS-R, whereas it provides better channel utilizations in all network environments. SP-R is not suitable to be used in wireless networks with poor link qualities since it cannot provide both QoS guarantees and high channel utilizations.

6.5.3 Effects of Node Densities

The influence of node densities over all routing schemes is studied in this section with 50 randomly selected networks. The number of nodes in the network is varied from 15 to 20 nodes. The successful transmission probability is random and uniform in the range of $0.90 \leq 1 - p_l \leq 1$. Figure 6.12 shows the throughputs of SP-R w/ CNCE, QoSSP-R w/ CNCE, and QoS-R w/ CNCE. There is no effect of the node density on the achievable throughput. Both QoSSP-R w/ CNCE and QoS-R w/ CNCE can achieve a throughput gain over SP-R w/ CNCE at all simulated node densities. The gains are more significant when we increase the number of nodes. Figure 6.13 illustrates the number of channel uses of routing schemes when we vary the node density. We found that the number of channel uses increases with the number of nodes in each network. In other words, the efficiency of channel utilization decreases because of the wireless link scheduling constraint. Transmitted packets have higher collision probabilities when nodes are denser. As a result, transmitted packets use more transmission channels from a source to a destination to avoid collision based on the definition of an independent set in Section 6.3.4. Figure 6.14 shows the throughput per channel use as a function of the number of nodes. Both QoSSP-R and QoS-R yield a better throughput per channel use than SP-R at all node densities. QoS-R w/ CNCE gives the best results.

6.5.4 Effects of Traffic Demands

We generate 50 random topologies in the experiment. A successful transmission probability of each link is randomly and uniformly generated, where $0.90 \leq 1 - p_l \leq 1$. The number of nodes in each network is set to 15. Traffic demands are determined by the number of s-d pairs. Source and destination nodes of these s-d pairs are randomly chosen from nodes

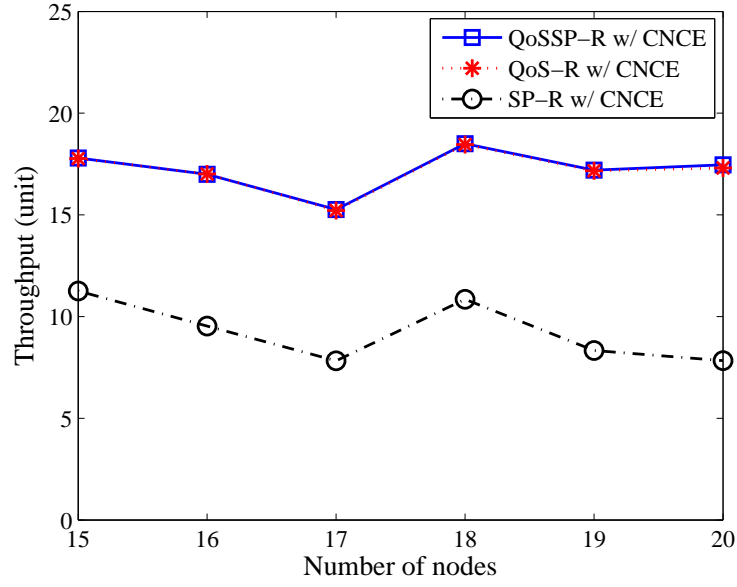


Figure 6.12: Comparison of the throughput for different routing schemes as a function of node density.

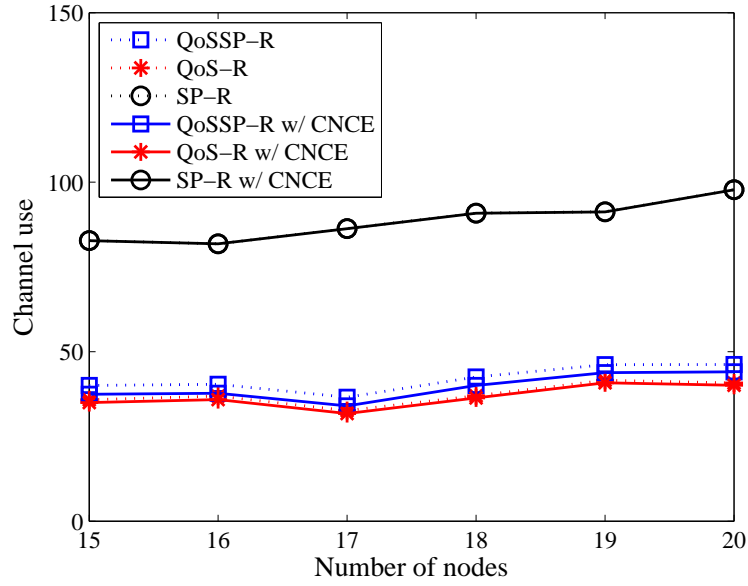


Figure 6.13: Comparison of the number of channel uses for different routing schemes as a function of node density.

in the network. The number of s-d pairs is varied from 10 to 20.

Figures 6.15, 6.16, and 6.17 show throughput, number of channel uses, and throughput per channel use of all routing schemes with different traffic demands, respectively. At all ranges of traffic demands, QoSSP-R w/ CNCE and QoS-R w/ CNCE give a bet-

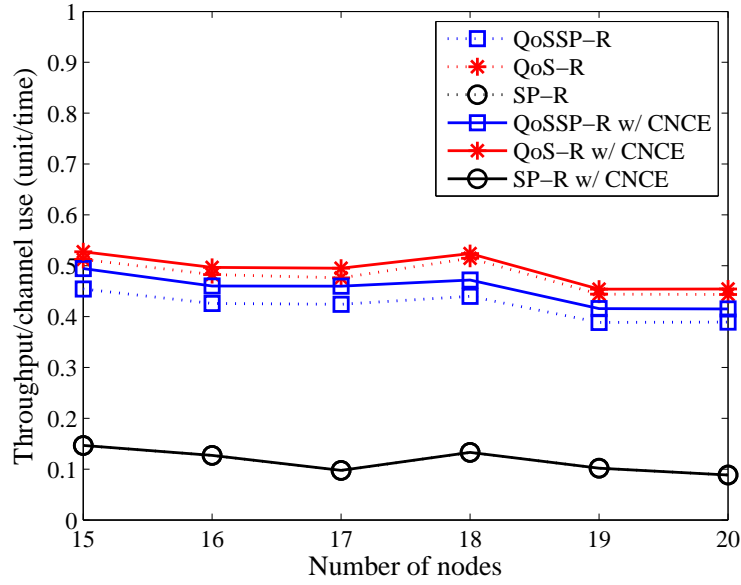


Figure 6.14: Comparison of the throughput per channel use for different routing schemes as a function of node density.

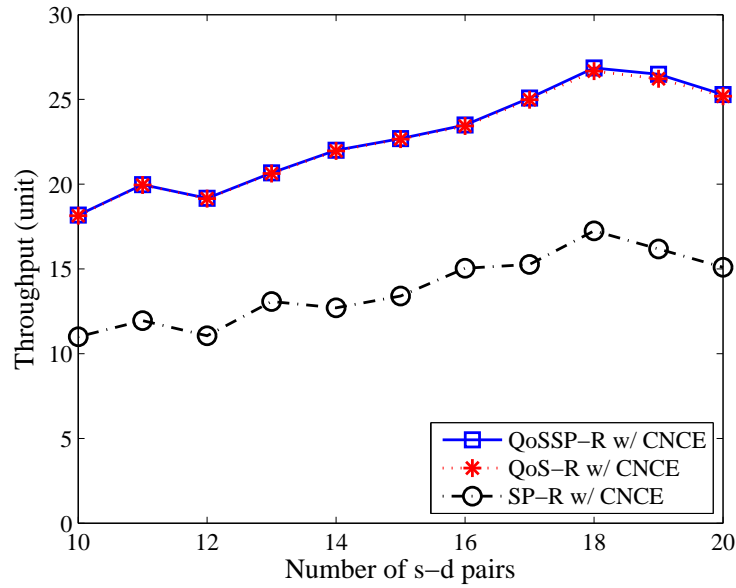


Figure 6.15: Comparison of the throughput for different routing schemes as a function of the number of s-d pairs in each network.

ter throughput than SP-R with and without CNCE algorithm. However, the throughput performance degrades as the amount of traffic demands increases since multiple s-d pairs compete for bandwidths and QoS guarantees. The performance gaps between the throughput from QoSSP-R and SP-R are more significant at high traffic demands. The throughput

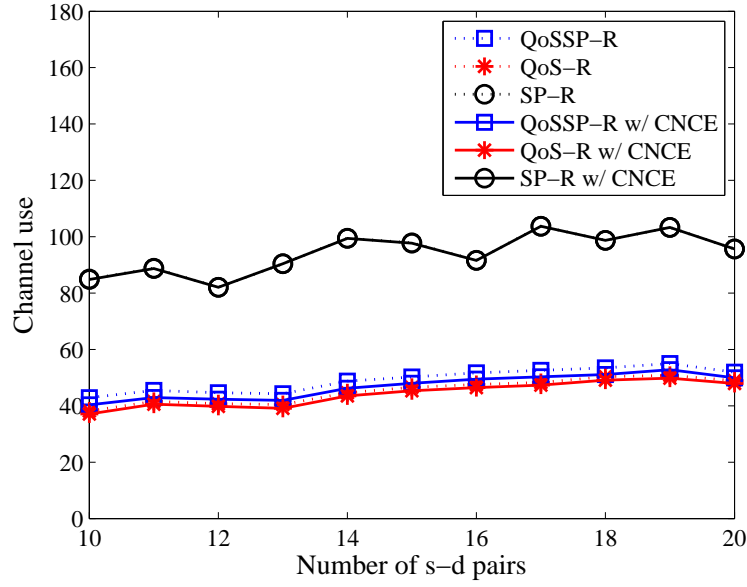


Figure 6.16: Comparison of the number of channel uses for different routing schemes as a function of the number of s-d pairs in each network.

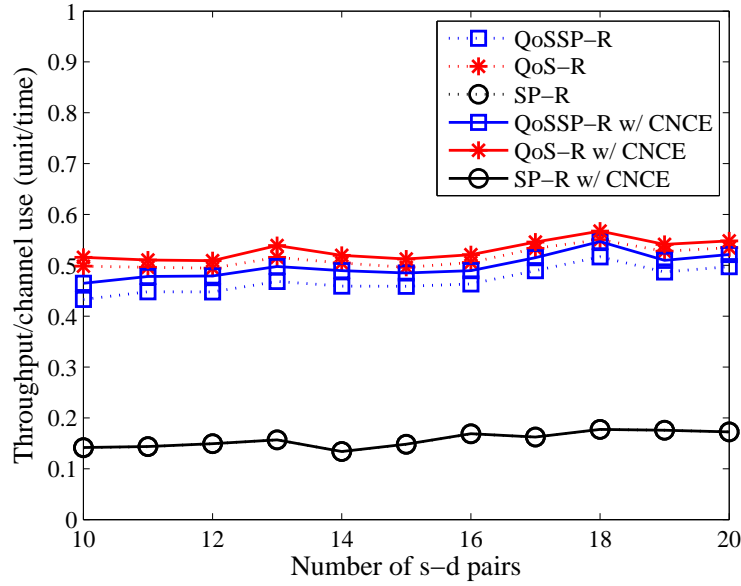


Figure 6.17: Comparison of the throughput per channel use for different routing schemes as a function of the number of s-d pairs in each network.

of QoSSP-R w/ CNCE is almost identical to QoS-R w/ CNCE, whereas the number of channel uses for QoSSP-R w/ CNCE is slightly higher than that of QoS-R w/ CNCE. Channel utilizations of both QoSSP-R and QoS-R surpass that of SP-R because our proposed routing schemes manage network resources more efficiently. Both QoSSP-R and

QoS-R put an emphasis on the reliability constraint so that they select paths based on the priorities of transmitted data and their QoS requirements. In contrast, SP-R selects the shortest paths from a source to a destination without considering QoS requirements and data priorities. For throughput per channel use, QoS-R w/ CNCE gives the best result. Both QoSSP-R and QoS-R overcome SP-R and SP-R w/ CNCE at all simulated traffic demands. However, CNCE algorithm cannot improve the throughput per channel use of SP-R as indicated by identical throughputs per channel use of SP-R and SP-R w/ CNCE.

6.5.5 Effects of Wireless Interference: A Case of Single Wireless Channel

In the earlier sections, we assume that the considering multi-hop wireless networks use multiple transmission channels together with careful channel planning such that interferences among active transmission links are minimized. However, when only one wireless channel is used, wireless interference becomes crucial. Here, we investigate the performance of our proposed scheme when there is only one wireless channel for transmission. We use the *protocol model* [108], which is a simplified version of wireless interference model, to define the conditions for a successful wireless transmission. In this model, each node n_i is equipped with a radio module with a transmission range R_i and a potentially larger interference range R'_i .

A transmission on link (n_i, n_j) with the physical distance of d_{ij} will be successful if two conditions are satisfied as follows.

1. Receiver node n_j is in the transmission range of transmitter node n_i ($d_{ij} \leq R_i$).
2. Receiver node n_j is not in the interference range of any transmitter node n_k that is using the wireless channel ($d_{kj} > R'_k$).

The maximal transmission range and interference range of each node n_i are set to 100 and 200 meters, respectively. The experimental settings are the same as in Section 6.5.2. Specifically, ten source-destination (s-d) pairs are randomly chosen in 50 random networks, where each network has 15 nodes. The successful packet transmission probability of each link is randomly generated, where $Z \leq 1 - p_l \leq 1$ and $Z \in \{0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96\}$.

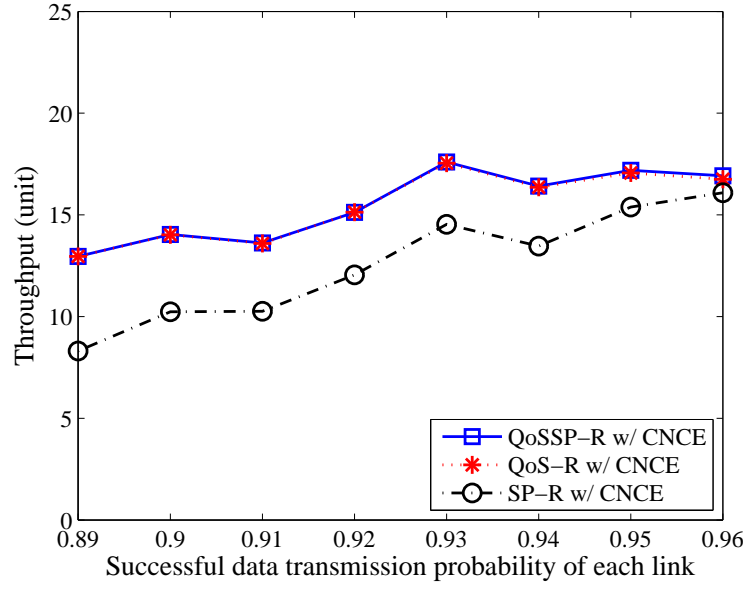


Figure 6.18: Comparison of the throughput for various link qualities in single-channel networks having 15 nodes.

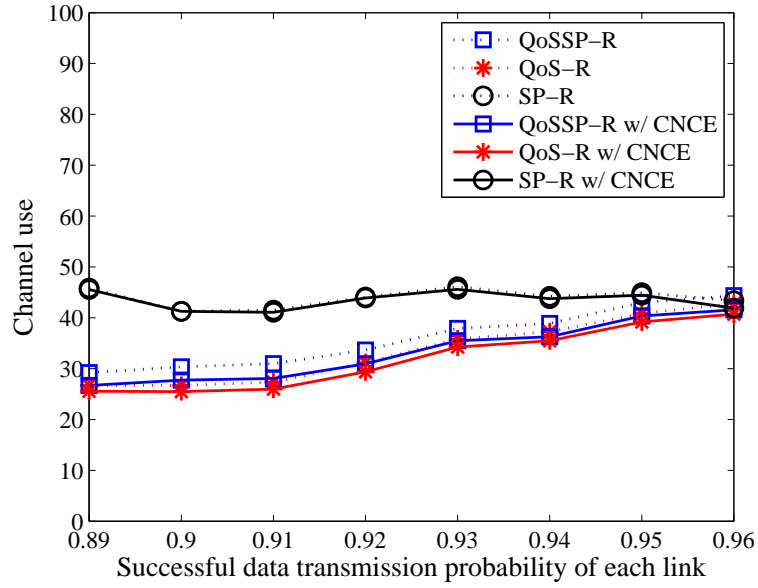


Figure 6.19: Comparison of the number of channel uses for various link qualities in single-channel networks having 15 nodes.

Figures 6.18, 6.19, and 6.20 show throughput, number of channel uses, and throughput per channel use of all routing schemes with different traffic demands, respectively. From the results, the comparative performances of all routing schemes show the same trend as in Section 6.5.2. Comparing with the multi-channel use, a single-channel use gives lower

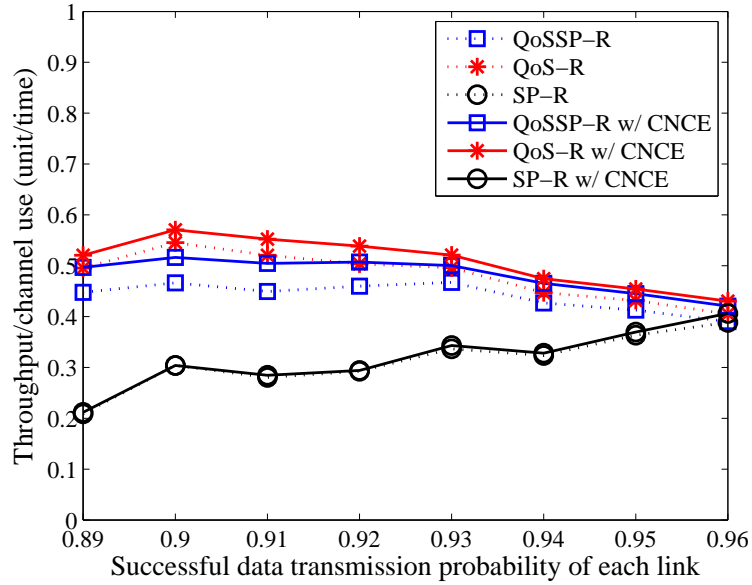


Figure 6.20: Comparison of the throughput per channel use for various link qualities in single-channel networks having 15 nodes.

throughput and lower efficiency of channel utilization.

The lower obtained throughputs can be explained as follows. By using the protocol model, there is one more condition added to the definition of an independent set. In particular, the receiver node of a link in an independent set must not be in the interference ranges of the transmitter nodes of the other links in the same independent set. Consequently, the size of an independent set is in general reduced, whereas the size of a family of independent sets whose union can cover all links of the network is in general increased. Given the same amount of traffic, the wireless link scheduling constraint, and the same link capacities, the number of channel uses available from independent sets to support traffic demands per unit time becomes smaller. Therefore, the capacity of multi-hop wireless networks decreases when there is a single wireless channel available.

6.6 Conclusion

A routing scheme that provides QoS guarantee for heterogeneous layered unicast transmissions in multi-rate lossy wireless networks with and without CNC was investigated and compared to its alternatives in this research. The path of each layered unicast flow is obtained by solving a constrained linear optimization problem subject to the QoS re-

quirement of each flow. The associated CNCE algorithm decides whether or not CNC will be performed at an intermediate node by considering the A-B, Y, and X structures in the network. It was demonstrated by computer simulations that the proposed QoS-aware routing scheme yields better throughput and higher channel use efficiency with the QoS guarantee on heterogeneous unicast flows.

Chapter 7

Content Importance-based Multicast Using Inter-source Network Coding

This chapter presents a QoS-aware routing scheme for robust multicast transmission in multi-hop wireless networks. This routing scheme employs a multi-source technique and a new inter-source network decoding method to improve the reliability of transmitted data. Simulation results demonstrate the benefits of this routing scheme over several existing schemes.

7.1 Introduction

In this chapter, we propose a new QoS-aware routing scheme for multicast transmission in multi-hop wireless networks. The contents that we particularly consider are scalable videos, which are a general target of multicast transmission. We propose the routing scheme with two objectives: 1) to support layered video multicast with QoS guarantee under lossy environments and 2) to improve transmission reliability and network resource usage by applying a new inter-source network decoding technique. To achieve the first objective, our proposed scheme uses an optimization formulation to find transmission paths of all layered video multicast flows for each source. QoS requirements, network resources and video transmission rates are the optimization constraints. To fulfill the

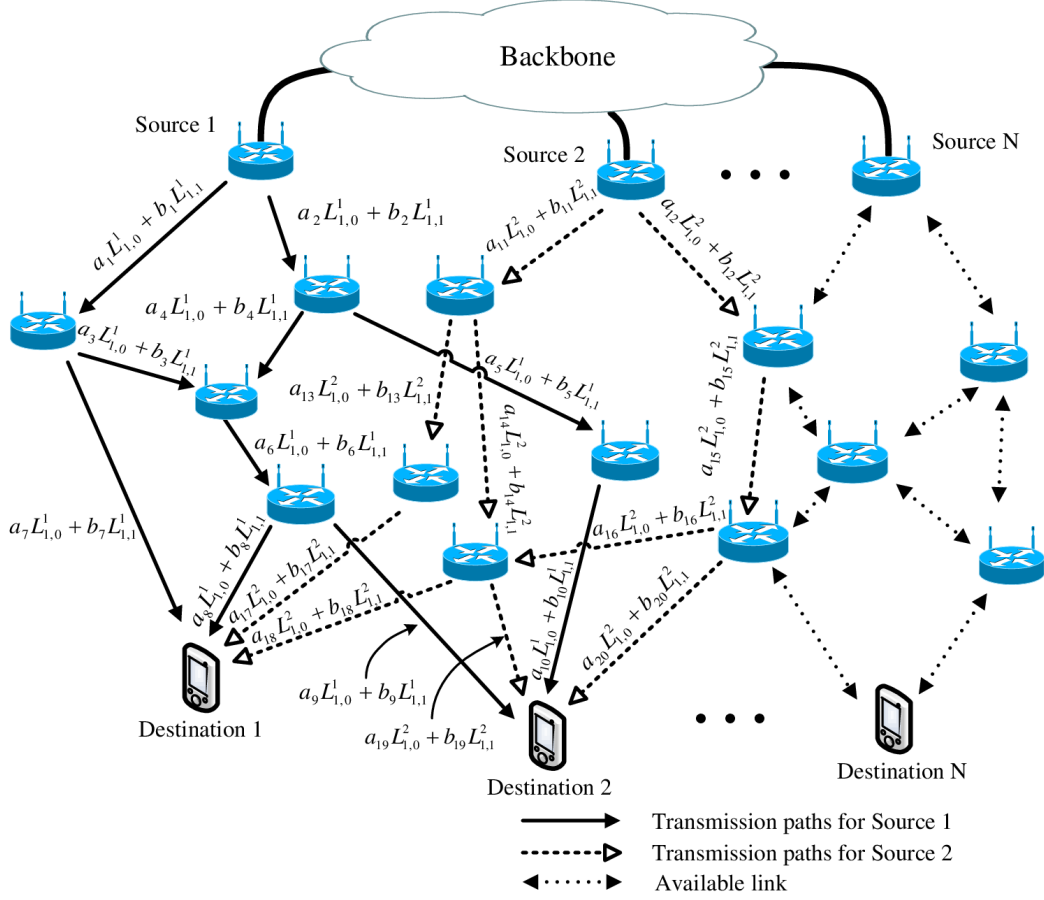


Figure 7.1: Layered multicast with multiple sources and inter-source network decoding.

second objective, the proposed scheme decides whether or not only the primary source can multicast video layers with each video layer's QoS requirement. If not, the secondary sources will multicast the same set of video layers to the destinations. The utilized number of secondary sources is increased until all QoS requirements of video layers are achieved or the network resources are depleted. Inter-source network decoding can be further adopted in the second step to enhance reliability by allowing destinations using packet data from different sources to decode transmitted data. The destinations need only to obtain a sufficient number of video packets to decode video data regardless of where they come from.

Figure 7.1 shows an example of layered video multicast transmission over lossy networks. In this figure, we consider a multi-hop wireless networks, which is a subset of lossy networks. The network topology is assumed to be known and link quality statistics such as the packet loss rate can be collected over time. The capacity of each link is set to one

unit, which can be translated to kilobits per second. The objective of the transmission is to multicast video layers to all destinations with QoS guarantee. The sources can search for the routes having the maximal reliability in the best-effort manner for multicasting layered video to destinations. In some cases, reliability of the obtained routes may not meet the QoS requirements of the destinations because of time-varying characteristics of wireless links. Moreover, if some sources or wireless network nodes are lost, video packets may not arrive at destinations.

To enhance reliability of video multicast, path diversity can be used by transmitting the same video layers from different source locations and different transmission paths. Destinations can have higher successful probabilities in receiving video packets. As shown in Figure 7.1, source i multicasts the same layered data L_1 with two data units, $L_{1,0}^i$ and $L_{1,1}^i$. To use network resource effectively, wireless nodes can apply network coding [28] to transmit data starting from the source node. A packet transmitted through each link is a linear combination of $L_{1,0}^i$ and $L_{1,1}^i$. For example, $a_1 L_{1,0}^1 + b_1 L_{1,1}^1$ is transmitted through the link connected with Source 1, where a_1 and b_1 are network coding coefficients that are randomly selected from a finite field [52].

The use of network coding can combat bottlenecks in the network. For instance, a bottleneck appearing in the routes connecting Source 1 to Destinations 1 and 2 can be solved by transmitting network coded data, *i.e.*, $a_6 L_{1,0}^1 + b_6 L_{1,1}^1$, instead of transmitting either $L_{1,0}^1$ or $L_{1,1}^1$ through the bottleneck link. At each destination in this example, it needs only two network coded packets to recover the transmitted data. For example, it requires $a_7 L_{1,0}^1 + b_7 L_{1,1}^1$ and $a_8 L_{1,0}^1 + b_8 L_{1,1}^1$ to obtain layered data L_1 . When path diversity with multiple sources is used, destinations can have more degrees of freedom in selecting network coding packets from other sources to recover layered data L_1 . This can be done because the network coded coefficients are randomly selected at different wireless nodes. With a sufficiently large field size, the linear combinations at wireless nodes will be independent among others with a high probability. Therefore, destinations need only two packets no matter which source they come from to decode layered data L_1 . For example, Destination 1 can recover L_1 by using two packets from $a_7 L_{1,0}^1 + b_7 L_{1,1}^1$, $a_8 L_{1,0}^1 + b_8 L_{1,1}^1$, $a_{17} L_{1,0}^2 + b_{17} L_{1,1}^2$, and $a_{18} L_{1,0}^2 + b_{18} L_{1,1}^2$. This approach further increases a successful decoding probability of layered data, when some packets are lost without a

concern on additional latency caused by packet retransmissions.

The rest of this chapter is organized as follows. Section 7.2 defines a network model and assumptions used in this chapter. Besides, the optimization formulation is set up to compute the optimal routes for layered video transmission. Section 7.3 describes the proposed QoS-aware multi-source routing scheme and the use of random network coding in multicast transmission. The concept of the inter-source network decoding is also described in this section. Section 7.4 evaluates the advantages of the proposed video routing scheme using objective and subjective qualities of reconstructed video obtained from computer simulations. Conclusion remarks are given in Section 7.5.

7.2 System Model and Problem Description

7.2.1 Scalable Video Coding

Scalable video coding (SVC) refers to an extension of the H.264/AVC standard [24, 109]. The SVC was introduced to provide a solution in adapting video bitstreams to various needs of end users, terminal capabilities, or network conditions. SVC supports key components of H.264/AVC standard, such as intra-picture predictive coding, motion compensation, and Network Abstract Layer Unit (NALU), to achieve high coding and transmission efficiency. The scalabilities in SVC includes the spatial, temporal, and quality or Signal-to-Noise ratio (SNR) scalabilities [24]. To reduce the data rate of videos, subsets of SVC bitstreams can be removed at the expense of losing resolution, frame rate, or quality of the videos. Figure 7.2 demonstrates the scalabilities of a video encoded with an SVC encoder. A SVC bitstream consists of a base layer and several enhancement (or successive refinement) layers. Conceptually, the base layer contributes the video quality more than enhancement layers. This is because SVC utilizes *hierarchical prediction structure* for high coding efficiency [24]. The decoding of an enhancement layer requires the data of based layer and the entire lower enhancement layers. When the qualities of transmission links are heterogeneous, it is reasonable to transmit the data of the base layer, rather than those of enhancement layers, in a more reliable transmission path.

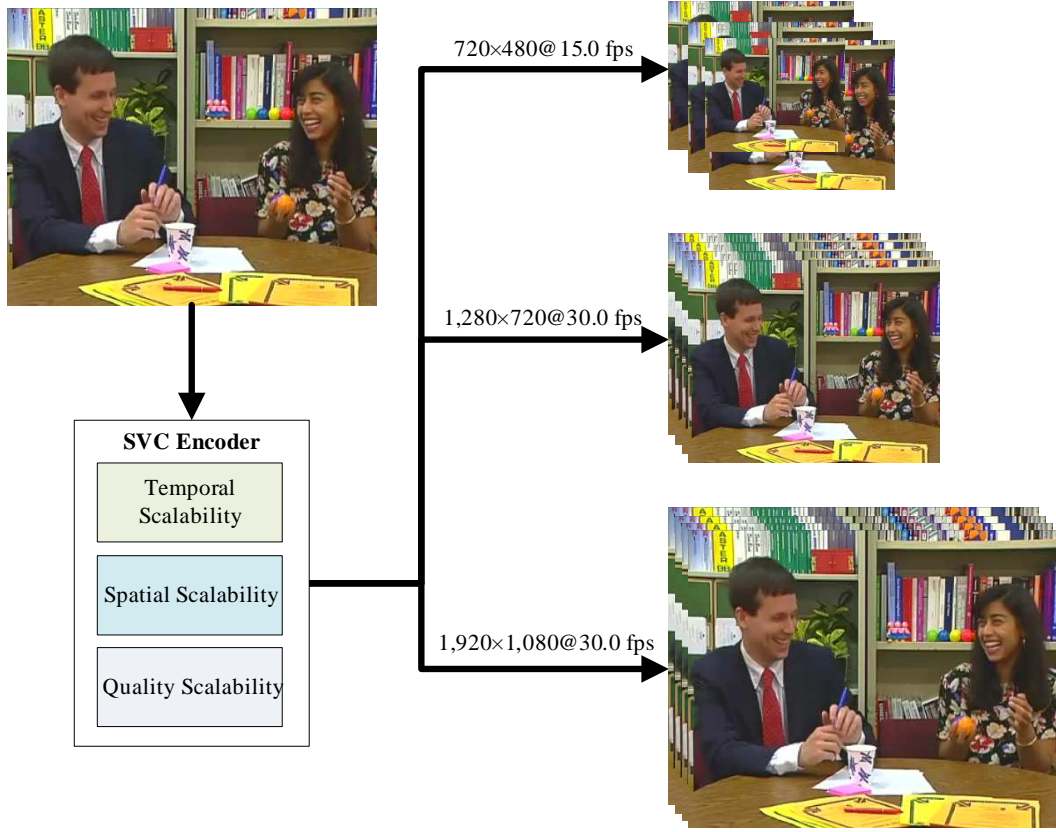


Figure 7.2: Different bitstreams enabling different spatial, temporal, and SNR scalabilities.

7.2.2 Network Model

We model a lossy network as a directed graph $G(N, E)$, where N and E are sets of nodes and links in the network, respectively. Let S and D be sets of source and destination nodes of a multicast transmission, respectively. For generality, we define the multicast transmission flows as a set of unicast flows (s, d) , where $s \in S$ and $d \in D$. Each s has the same set of data, which can be chosen to transmit data to nodes in D .

Define (a, b) as a link conveying data from node a to node b . For link $l \in E$, let $t(l)$ and $r(l)$ be the transmitter and receiver nodes of link l , respectively. For each node $n \in N$, let $T_O(n) = \{l \in E | n = t(l)\}$ and $T_I(n) = \{l \in E | n = r(l)\}$ be sets of outgoing and incoming links of node n , respectively. Let Γ be the set of all pairs of source and destination nodes in the network, where $(s, d) \in \Gamma$.

Each link has a normalized positive integral capacity or transmission rate denoted by c_l [63, 72, 73, 74]. One normalized unit of capacity can be translated to be bits per second. To give an example, if a considering link has transmission capacity 512 kbps, it can be

represented by $c_l = 2$ units, where each unit is equivalent to transmission rate 256 kbps. We assume that there is little or no interference occurring among wireless links, which may be technically achieved by using Orthogonal Frequency Division Multiplexing Access (OFDMA) [110]. For the link having a capacity c_l , where $c_l > 1$, we split the link to c_l links in parallel with a capacity equal to one for each link. The average probability of a packet loss of link l is p_l , where $0 \leq p_l \leq 1$. Each $(s, d) \in \Gamma$ transmits M layers of data, where L_i is the i^{th} layer with transmission rate r_i . Let the set of layer indices of each (s, d) be I_M , where $I_M = \{0, 1, 2, \dots, M-1\}$. In this chapter, video layer i will be more important than video layer j , when $i < j$.

7.2.3 QoS Guarantee

Definition 1: The QoS guarantee for $(s, d) \in \Gamma$ and $i \in I_M$ is a lower bound of the probability that source s can transmit a packet of L_i to destination d successfully.

The probability of a successful packet transmission for L_i , called the reliability and denoted by $P_i^{(s,d)}$, can be expressed as

$$P_i^{(s,d)} = \prod_{l \in E} (1 - p_l)^{f_{l,i}^{(s,d)}}, \quad (7.1)$$

where $f_{l,i}^{(s,d)}$ indicates whether or not link $l \in E$ is used to transmit a packet of L_i . If it is used, $f_{l,i}^{(s,d)} = 1$. Otherwise, $f_{l,i}^{(s,d)} = 0$.

7.2.4 Optimum Path Selection for Layered Multicast

Definition 2: A path for $(s, d) \in \Gamma$ and $i \in I_M$ is a set of links, denoted by $R_i^{(s,d)}$, used to transmit packets of layer L_i from source s to destination d . An optimal set of paths is such that $R_i^{(s,d)}$, $(s, d) \in \Gamma$, $i \in I_M$, maximizes the objective function under a set of constraints.

The optimal set of paths based on the optimization framework is formulated in this section. The objectives of the formulated framework are to maximize information values of transmitted layers and the transmission reliability.

We discuss the objective function as well as the set of constraints in the following subsections.

Table 7.1: Summary of notations for the optimal path selection for layered multicast

$G(N, E)$: directed graph that represents the network
N	: set of nodes in the network
E	: set of links in the network
S	: set of source nodes
D	: set of destination nodes
(s, d)	: pair of source node s and destination node d
(a, b)	: link conveying data from node a to node b
$t(l)$: transmitter node of link l
$r(l)$: receiver node of link l
$T_O(n)$: set of outgoing links of node n
$T_I(n)$: set of incoming links of node n
Γ	: set of all pairs of source and destination nodes in the network
c_l	: capacity of link l
p_l	: probability of packet loss of link l
M	: number of layers of data
L_i	: i^{th} layer
r_i	: transmission rate of the i^{th} layer
I_M	: set of layer indices, where $I_M = \{0, 1, 2, \dots, M - 1\}$
$P_i^{(s,d)}$: probability of a successful packet transmission for L_i of (s, d) , called the reliability
$f_{l,i}^{(s,d)}$: variable that indicates whether or not link l is used to transmit a packet of L_i for (s, d)
$R_i^{(s,d)}$: set of links used to transmit packets of layer L_i from source s to destination d
$x_i^{(s,d)}$: variable that indicates whether or not packets of layer L_i are transmitted from source s to destination d
κ_i	: information value used to prioritize data layers
$t_{l,i}$: total capacity of link l used by the i^{th} layer

7.2.5 Objective Function

Maximizing Information Values of Transmitted Layers

Define $x_i^{(s,d)}$ as a variable, where $x_i^{(s,d)} = 1$ indicates that packets of layer L_i are transmitted from source s to destination d . Else, $x_i^{(s,d)} = 0$. To prioritize data layers, we define the information value of each layer as κ_i , where $\kappa_i \geq \kappa_j$, when $i < j$. The information value physically represents the importance of each layer. Therefore, information values can be used as the estimated influence of layered data reflecting the reconstructed video quality. For example, the information value of video base layer is higher than those of video enhancement layers. Based on the defined variables, the first sub-objective function is defined for the optimal path selection of all destinations as

$$K_G = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i x_i^{(s,d)}. \quad (7.2)$$

K_G is the summation of the guaranteed information value from all traffics in the network. K_G varies proportionally to the number of transmitted layers of (s, d) .

Maximizing Reliability

When there is more than one path to agree with the first sub-objective function, this sub-objective function selects a path having the maximal reliability.

The reliability of each data layer can be maximized via the following sub-objective function.

$$K_R = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i P_i^{(s,d)}. \quad (7.3a)$$

To avoid non-linear optimization, which demands a higher computational complexity, we maximize a logarithmic version of reliability as the following linear function

$$\bar{K}_R = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i \log P_i^{(s,d)}. \quad (7.3b)$$

Theorem 5. *Maximizing the sub-objective function*

$$\bar{K}_R = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i \log P_i^{(s,d)} \quad (7.3c)$$

gives the same transmission paths as maximizing the sub-objective function

$$K_R = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i P_i^{(s,d)}. \quad (7.3d)$$

Proof. If a flow of the i^{th} layer for (s, d) can be transmitted, there can be $N \geq 1$ different paths for data transmission, denoted by $E_{i,j}^{(s,d)}$, where $j \in \{0, 1, 2, \dots, N\}$. Each $E_{i,j}^{(s,d)}$ has a reliability, denoted by $P_{i,j}^{(s,d)}$, where $0 < P_{i,j}^{(s,d)} \leq 1$. Maximizing (7.3d) is equivalent to select $E_{i,k}^{(s,d)}$ to be the selected path, where $P_{i,k}^{(s,d)} = \max(P_{i,j}^{(s,d)}, j \in \{0, 1, 2, \dots, N\})$.

Consider maximizing (7.3c), it is equivalent to maximize (7.3d) since the logarithmic function is a monotonically increasing function, and thus the order of data is preserved. In other words, if $P_{i,m}^{(s,d)} > P_{i,n}^{(s,d)}$, then $\log P_{i,m}^{(s,d)} > \log P_{i,n}^{(s,d)}$, where $m \neq n$. To obtain the maximal $P_{i,j}^{(s,d)}$, the $E_{i,k}^{(s,d)}$ must be selected as the transmission path. Thus, we conclude that maximizing the (7.3c) gives the same transmission path as maximizing (7.3d). \square

According to (7.1), we rewrite (7.3b) to

$$\overline{K}_R = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i \log \prod_{l \in E} (1 - p_l)^{f_{l,i}^{(s,d)}}, \quad (7.3e)$$

which is equivalent to

$$\overline{K}_R = \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i \sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)}, \quad (7.3f)$$

where $\bar{p}_l = \log(1 - p_l)$.

The κ_i is used to assign links with better channel conditions to more important layers of (s, d) . To maximize \overline{K}_R , transmissions of a lower layer with a larger κ_i will be placed on the link with higher \bar{p}_l or lower p_l .

While K_G is a sum of product between integer numbers, \overline{K}_R is a sum of product between integer and logarithmic fractional numbers having values less than one. For instance, there is a selected routing path for the layer $L_i^{(s,d)}$ with $0.1 < P_i^{(s,d)} < 1$ and a given κ_i , $K_G/\kappa_i = 1$, while $-1 < \overline{K}_R/\kappa_i < 0$ according to (7.2) and (7.3f). Hence, $|\frac{K_G}{\overline{K}_R}| > 1$, where $0.1 < P_i^{(s,d)} < 1$. Note that establishing data transmission with $P_i^{(s,d)} \leq 0.1$ is rarely seen in practical networks, the assumed scenario is therefore reasonable. The objective function gives priorities to information values of transmitted layers more than the reliability. The scheme maximizes reliability of transmitted layers in the best-effort manner, which is not limited by a QoS constraint.

We can combine (7.2) and (7.3f) to

$$\sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i x_i^{(s,d)} + \sum_{i \in I_M} \sum_{l \in E} \kappa_i \sum_{(s,d) \in \Gamma} \bar{p}_l f_{l,i}^{(s,d)} \quad (7.4a)$$

$$= \sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i \{x_i^{(s,d)} + \sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)}\}. \quad (7.4b)$$

7.2.6 Problem Formulation

To solve for the optimal set of paths of all $(s, d) \in \Gamma$, an integer linear optimization is formulated. The input parameters of the formulation, which are network resources and required loads, include a set of nodes in the network (N), a set of links in the network (E), link capacity of link l (c_l), probability of packet loss of link l (p_l), a set of source nodes (S), a set of destination nodes (D), a set of layer indices (I_M), and required transmission rate of the i^{th} layer (r_i). The linear optimization formulation can be shown as follows.

Maximize

$$\sum_{i \in I_M} \sum_{(s,d) \in \Gamma} \kappa_i \{x_i^{(s,d)} + \sum_{l \in E} \bar{p}_l f_{l,i}^{(s,d)}\} \quad (7.5a)$$

Subject to

$$\sum_{l \in T_O(n)} f_{l,i}^{(s,d)} - \sum_{l \in T_I(n)} f_{l,i}^{(s,d)} = \begin{cases} r_i x_i^{(s,d)}, & n = s \\ -r_i x_i^{(s,d)}, & n = d \\ 0, & \text{otherwise} \end{cases} \quad (7.5b)$$

$$\forall i \in I_M, \forall (s,d) \in \Gamma, \forall n \in N,$$

$$f_{l,i}^{(s,d)} \leq t_{l,i}, \forall (s,d) \in \Gamma, \forall i \in I, \forall l \in E, \quad (7.5c)$$

$$\sum_{i \in I} t_{l,i} \leq 1, \forall l \in E, \quad (7.5d)$$

$$x_i^{(s,d)} \geq x_{i+1}^{(s,d)}, \forall i \in I_{M-1}, \forall (s,d) \in \Gamma, \forall l \in E, \quad (7.5e)$$

$$f_{l,i}^{(s,d)} \in \{0, 1\}, \forall i \in I_M, \forall (s,d) \in \Gamma, \forall l \in E, \quad (7.5f)$$

$$t_{l,i} \in \{0, 1\}, \forall i \in I_M, \forall l \in E, \quad (7.5g)$$

$$x_i^{(s,d)} \in \{0, 1\}, \forall i \in I_M, \forall (s,d) \in \Gamma, \quad (7.5h)$$

where $I_{M-1} = \{0, 1, \dots, M-2\}$ and $t_{l,i}$ is the capacity of link l used by the i^{th} layer.

- Constraint (7.5b) is the flow conservation constraint.
- Constraints (7.5c) and (7.5d) are link capacity constraints under network coding condition. They allow the flows of different (s,d) with common L_i to share link capacity. Note that the network coding is executed separately layer-by-layer to maintain the QoS guarantee of each layer transmission.
- Constraint (7.5e) is the layered data constraint. A transmission path of a higher layer will be chosen only if a transmission path of a lower layer has been selected.
- Constraints (7.5f), (7.5g), and (7.5h) are the feasible values of $f_{l,i}^{(s,d)}$, $t_{l,i}$, and $x_i^{(s,d)}$, respectively.

The solution of the problem can be obtained by various mathematical approaches. We select the GNU Linear Programming Kit (GLPK) [111], which provides the well known

Branch-and-Bound algorithm, to be the solver for the multi-objective integer linear programming. The solution contains the sets of $x_i^{(s,d)}$ and $f_{l,i}^{(s,d)}$, which respectively represent the transmitted layers and selected links for each destination d . These parameters will be used in the next section.

7.3 Reliable Layered Multicast with Multiple Sources and Network Coding

7.3.1 Transmission Reliability of Layered Multicast with Multi-Source

Transmission reliability of layered data can be improved by using source and path diversities. Destinations can expect to receive layered data from alternative routes, when primary transmission paths experience severe packet loss or the QoS requirements of transmitted data cannot be fulfilled at destinations. Let q_i^d be a QoS requirement of transmitting data layer L_i to destination d , where $0 < q_i^d \leq 1$. $P_i^{(s_1,d)}$ is the transmission reliability of a path used to transmit L_i to destination d by a primary source. This reliability is an outcome from Section 7.2.6. When $P_i^{(s_1,d)} < q_i^d$, the QoS guarantee of L_i of destination d cannot be achieved by the primary source. Therefore, it may be helpful to increase the number of transmitting sources to improve reliability.

Let $P_i^{(s_j,d)}$ be the transmission reliability of paths used to transmit L_i to destination d by secondary source s_j , where $j \in J$, $j > 1$, and J is a set of positive integers. The set of paths to destination d of the primary source s_1 and every secondary source s_j are independent for all j . The reliability of L_i , when we use N transmitting sources, can be written as

$$\begin{aligned}\overline{P}_i^d(N) &= P\left\{\bigcup_{k=1}^N S_k\right\}, \\ &= 1 - \prod_{k=1}^N (1 - P_i^{(s_k,d)}),\end{aligned}\tag{7.6}$$

where $\overline{P}_i^d(N)$ is the transmission reliability of data L_i to destination d using N sources and S_k is the event that destination d receives data L_i successfully from source s_k . As we can see, the term $\prod_{k=1}^N (1 - P_i^{(s_k,d)})$ of (7.6) is the probability that destination d cannot obtain layer L_i from any source.

Table 7.2: Summary of notations for QoS-aware multicast with multiple sources and network coding

q_i^d	: QoS requirement of transmitting data layer L_i to destination d
$P_i^{(s_j,d)}$: transmission reliability of paths used to transmit L_i to destination d by source s_j
$\overline{P}_i^d(N)$: transmission reliability of data L_i to destination d using N sources
S_k	: event that destination d receives data L_i successfully from source s_k
Ω_s	: individual lower-bound achievable data rate obtained by single source routing from source s , called QoS parameter
Ψ	: set of achievable data rates for all sources
$P_{dec,i}^{(s_k,d)}$: decoding probability of L_i at destination d achieved by source s_k

7.3.2 QoS-aware Multicast with Multiple Sources and Network Coding

In this section, we present the algorithm in using multiple sources to reliably multicast data to destinations, when the required QoS guarantee cannot be achieved by a single source as in Section 7.2.6. However, the improved QoS guarantee comes with more network resource usage. Therefore, the multi-source multicast technique must be selectively used only with the data with high priorities. Network coding is used in this Section to combat with the bottlenecks in the network [52, 82].

When many flows are simultaneously transmitted in a network, some links may become bottlenecks for some destinations, *i.e.*, an aggregate of required data rate at the considering link exceeds the available link capacity. The network coding allows different flows of data to share link capacity instead of competing for individual capacity, thus relieves the bottlenecks of network with sacrificing some computational power. The QoS-aware multicast with multiple sources and network coding algorithm can be stated step-by-step as follows.

Step 1: The primary source is selected by choosing the source s giving the highest lower-bound of achievable data rate, which is the data rate destination can expect to receive from the transmission, obtained by the single source routing, denoted by Ω_s , where $\Omega_s = \sum_{i \in I} \sum_{d \in D} r_i P_i^{(s,d)} x_i^{(s,d)}$. The single source routing is performed by using our proposed optimization framework in Section 7.2.6. The optimal solution of the framework maximizes information values and the transmission reliability of transmitted layers, but may not

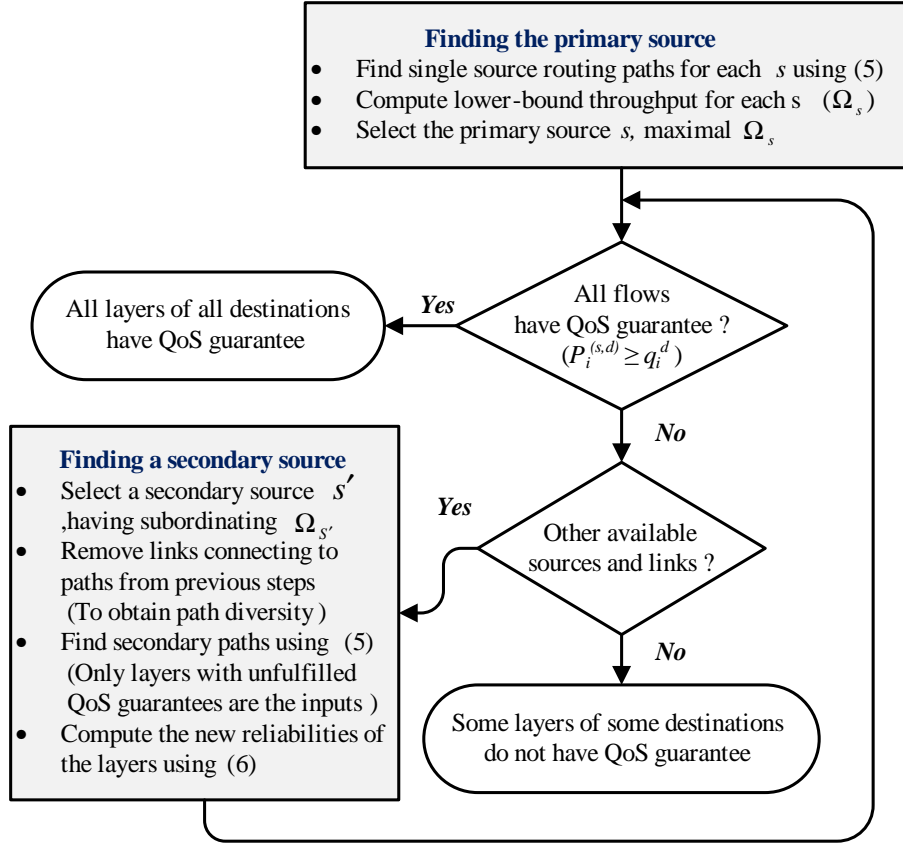


Figure 7.3: A flowchart demonstrating the QoS-aware multicast with multiple sources and network coding.

achieve the QoS guarantees to some L_i of some destinations. An example of the mentioned scenario is demonstrated in **Example 1**. The Ω_s for all $s \in S$ will be kept in a set of individual source achievable data rate, denoted by Ψ .

Step 2: Determine whether all L_i of (s, d) , $i \in I$, $d \in D$, can obtain its required QoS guarantee q_i^d or not. If all layers L_i of (s, d) , $i \in I$, have QoS guarantees, use the optimal routes to convey the layered data. Otherwise, the secondary source is needed and go to Step 3.

Step 3: Select the secondary source \hat{s} from the subordinate sources according to Ω_s from Ψ . To obtain path diversity, we modify the set of links E by removing all links having the end nodes in the set of paths from the previous steps, except the destination nodes.

Step 4: The secondary paths of the layers with the unfulfilled QoS guarantees are obtained by using the proposed optimization framework in Section 7.2.6. Note that only the layers with the unfulfilled QoS guarantees are considered.

Step 5: Compute the new reliability of the layers with unfulfilled QoS guarantees, which

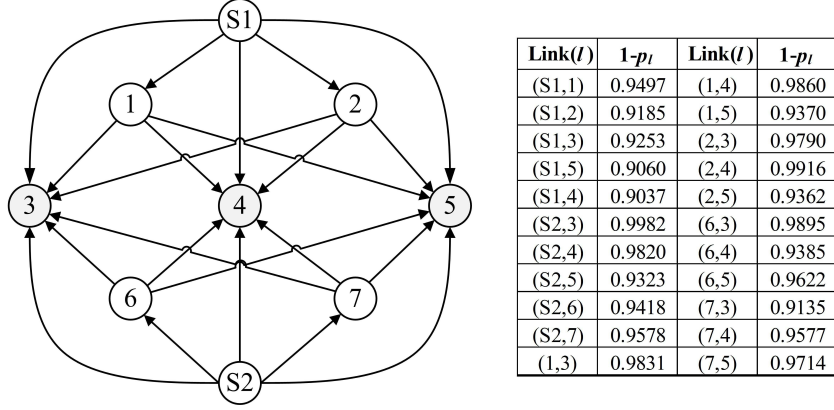


Figure 7.4: A network with two source nodes.

is computed from (7.6). The algorithm then determines the QoS guarantees for all L_i , $i \in I$, of all d . If there is any L_i of d that does not obtain QoS guarantee, repeat Steps 3, 4, and 5 until the QoS guarantees of all layers are achieved or all network resources and all sources are used.

Figure 7.3 shows a flow chart of the proposed algorithm. At the end of the algorithm, the set of optimal paths used to convey all layers L_i , $i \in I$, will be determined. These paths will be the input for network code assignment using random network coding to ensure that destinations can receive their max-flow rates. Network coding will be applied only in transmitting data from the same source and same layer. Therefore, decoding layer L_i successfully depends on whether or not the random network codes can be decoded at destinations and transmitting data packets arrive at destination successfully. Suppose that data layer L_i can be divided to be K transmitting packets, $L_{i,0}, \dots, L_{i,K-1}$. The decoding probability of layer L_i from source s_k can be expressed as

$$P_{dec,i}^{(s_k,d)} = \prod_{j=0}^{K-1} \left(1 - \frac{1}{|F|^{K-j}}\right) \times P_i^{(s_k,d)}, \quad (7.7)$$

where the first expression of (7.7) is the decoding probability of random network coding [83] with field size $|F|$, whereas the second expression is the reliability of transmission of L_i from source s_k to destination d over the chosen optimal paths. After replacing (7.7) to $P_i^{(s_k,d)}$ in (7.6), we obtain the decoding probability of data layer L_i from source s_k taking network coding into consideration.

Example 1: Consider the network with a set of average packet loss rates for each link

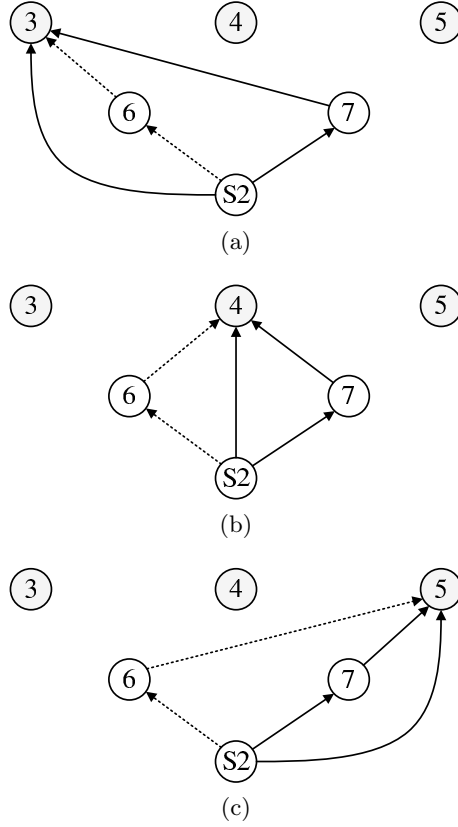


Figure 7.5: A set of paths for each destination obtained from the primary source node S2: (a) a set of paths for destination node 3, (b) a set of paths for destination node 4, and (c) a set of paths for destination node 5.

shown in Figure 7.4. A successful transmission probability of each link $l \in E$ is random and uniform in the range of $0.90 \leq 1 - p_l \leq 1$. The network has two source nodes having the same set of data, which are nodes $S1$ and $S2$. There are 3 destination nodes, *i.e.*, nodes 3, 4 and 5, $D = \{3, 4, 5\}$. Transmitting data consists of two layers denoted as L_0 and L_1 . Transmission rate and QoS requirement of L_0 are set to be $r_0 = 2$ units and $q_0^d = 0.90$, while transmission rate and QoS requirement of L_1 are set to be $r_1 = 1$ unit and $q_1^d = 0.80$, for all $d \in D$.

At the end of **Step 1**, we have $\Psi = \{\Omega_{S1}, \Omega_{S2}\}$, where $\Omega_{S1} = 7.7025$ and $\Omega_{S2} = 8.0046$. Therefore, we select $S2$ as the primary source. According to **Step 2**, a set of paths of each layer data for each destination are shown in Figure 7.5, where the sets of solid lines and dash lines are for L_0 and L_1 transmissions, respectively. The sets of paths for transmitting L_0 from $S2$ to destinations 3, 4, and 5 are $\{(S2, 7), (7, 3), (S2, 3)\}$, $\{(S2, 7), (7, 4), (S2, 4)\}$, and $\{(S2, 7), (7, 5), (S2, 5)\}$, respectively. The set of paths for transmitting L_1 to destina-

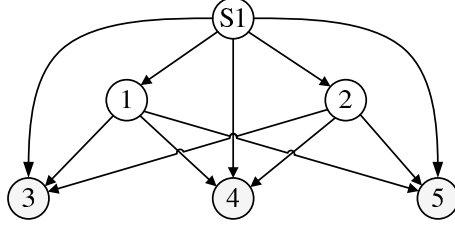


Figure 7.6: The network when a set of links has been removed in **Step 3**.

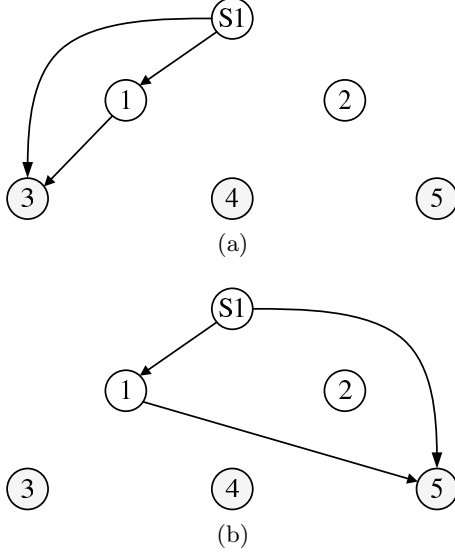


Figure 7.7: A set of paths for each destination obtained from the secondary source node S1: (a) a set of paths for destination node 3 and (b) a set of paths for destination node 5.

tions 3, 4, and 5 are $\{(S2, 6), (6, 3)\}$, $\{(S2, 6), (6, 4)\}$, and $\{(S2, 6), (6, 5)\}$, respectively.

Based on the imposed QoS requirements, the transmission of L_0 to destination nodes 3 and 5 cannot achieve the QoS requirement since $P_0^{(S2,3)}$ and $P_0^{(S2,5)}$ are equal to 0.8733 and 0.8673, respectively. The reliability is computed from (7.1), *e.g.*, $P_0^{(S2,3)} = (1 - p_{(S2,3)})(1 - p_{(S2,7)})(1 - p_{(7,3)}) = (0.9982)(0.9578)(0.9135) = 0.8733$. Therefore, we need **Step 3**. A set of links having the end nodes in the paths from the previous steps are removed. Therefore, the network has been changed to the network in Figure 7.6.

Next, we find a set of secondary paths of the layers with the unfulfilled QoS guarantees by using the proposed optimization framework in Section 7.2.6 at **Step 4**. After solving the optimization problem, the sets of secondary paths for destination nodes 3 and 5 are $\{(S1, 1), (1, 3), (S1, 3)\}$ and $\{(S1, 1), (1, 5), (S1, 5)\}$, respectively. The QoS of these secondary paths are equal to $P_0^{(S1,3)} = 0.8639$ and $P_0^{(S1,5)} = 0.8061$. These paths are depicted in Figure 7.7

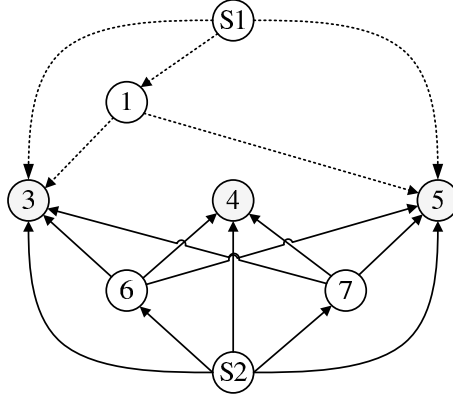


Figure 7.8: A set of paths from multiple sources.

Table 7.3: Comparison of the reliability obtained by single source and multi-source multicast.

Schemes	Layer 0	Layer 1
Source S1	(0.8639, 0.8463, 0.8061)	(0.8992, 0.8599, 0.9108)
Source S2	(0.8733, 0.9007, 0.8673)	(0.9319, 0.8839, 0.9062)
Multiple sources S1 and S2	(0.9828, 0.9007, 0.9743)	(0.9319, 0.8839, 0.9062)

At **Step 5**, according to (7.6), the new reliability of L_0 is $P_0^3 = 0.9828$ and $P_0^5 = 0.9743$ for destination nodes 3 and 5, respectively. The QoS guarantees for all L_i of all d are currently achieved, hence we do not need other secondary sources. The sets of routing paths for each destination is depicted in Figure 7.8, where the solid and the dash lines represent paths used by the primary source and the secondary source, respectively.

The reliability obtained by the proposed method compared with those provided by the two single-source multicast schemes are demonstrated in Table 7.3 layer-by-layer, where the first, second, and third components of tuples represent reliability of destination nodes 3, 4 and 5, respectively. The reliability of layered data using multi-source multicast is superior to that of single source multicast.

7.3.3 Inter-source Network Decoding

In this section, the inter-source network decoding is presented to further improve the reliability of transmitting layered data based on the multi-source multicast routing proposed in Section 7.3.2. The algorithm in Section 7.3.2 uses only the packet data from the same source for network decoding. If some packets are lost during transmissions, the other network coded data from the considering source are useless. Inter-source network decoding can alleviate this problem by allowing the use of network coded data of the same layer

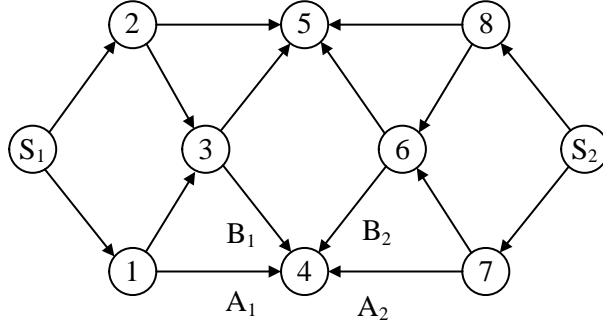


Figure 7.9: An example network using inter-source network coding.

from other sources to help recovering data at the considering destination. To recover the coded layer, each destination just needs to obtain a sufficient number of packets from any source so that it can successfully decode the data. Suppose that each source transmits layer L_i with k units or packets, $L_{i,0}, L_{i,1}, \dots, L_{i,k}$ and these packets may have to be coded at intermediate nodes. Obtaining k linearly network coded packets, the coded data can be recovered by solving linear equations. This is equivalent to receiving the global encoding kernels of random network coding for all k packets, which will form a matrix with rank k [112]. If there are N sources transmitting layer L_i , with inter-source network decoding, to recover all k units of layer L_i , each destination needs to obtain k network coded packets from total $N \times k$ to successfully decode L_i .

Let us consider the specific example of inter-source network decoding. Consider the network in Figure 7.9. The network has two source nodes S_1 and S_2 transmitting the same set of data. Each source separately multicasts layer L_i with two data packets, *i.e.*, $L_{i,0}$ and $L_{i,1}$, to destination nodes 4 and 5. At intermediate nodes, packets from the same source can be linearly coded, where the network coding coefficients are selected randomly and independently from other intermediate nodes. Based on Theorem 3 in [112], the random linear network coding can be applied to transmitting data starting from the source node.

The encoded data are transmitted to destination nodes 4 and 5 through the determined set of paths. At destination node 4, it obtains a set of encoded data as follows $A_1 = a_{1,1}L_{i,0} + a_{1,2}L_{i,1}$, $A_2 = a_{2,1}L_{i,0} + a_{2,2}L_{i,1}$, $B_1 = b_{1,1}L_{i,0} + b_{1,2}L_{i,1}$, and $B_2 = b_{2,1}L_{i,0} + b_{2,2}L_{i,1}$, where A_i and B_i are randomly and linearly coded data transmitted from S_j , where $j = 1, 2$.

The probability that destination node 4 successfully obtains $L_{i,0}$ and $L_{i,1}$ is equal to

the probability of the event that at least two encoded data either from S_1 or S_2 arrive at the destination. For instance, when destination node 4 receives A_1 and B_2 , we can formulate a matrix containing global encoding kernels as [112]

$$Y = \begin{bmatrix} a_{1,1} & a_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix}$$

and its linear equations can be written as

$$X = \begin{bmatrix} a_{1,1}L_{i,0} + a_{1,2}L_{i,1} \\ b_{2,1}L_{i,0} + b_{2,2}L_{i,1} \end{bmatrix}.$$

We can write the linear equations in the form of matrix of global encoding kernel as

$$Y \begin{bmatrix} L_{i,0} \\ L_{i,1} \end{bmatrix} = X.$$

When matrix Y is nonsingular, we can decode $L_{i,0}$ and $L_{i,1}$ by solving linear equations using the Gaussian elimination. As we can see from the example, the destination has more degree of freedom in choosing the packets from any source. It needs only two data packets to recover all transmitted data.

7.4 Experimental Results

To evaluate our proposed reliable layered video multicast scheme, we compare the simulation results with the following routing schemes:

Single source routing schemes

- The shortest path routing (QoS-oblivious w/o NC), which was partly used in [113, 114].
- QoS-aware routing for layered multicast transmission (QoS-aware w/ NC) [63].
- QoS-oblivious routing using network coding (QoS-oblivious w/ NC), which was partly used in [52, 75].

Multi-source routing schemes

- Peer-to-peer streaming of Scalable Video Coding (P2P streaming of SVC), which is based on [79, 80].
- QoS-aware routing using network coding and multiple sources (QoS-aware w/ NC and multiple sources).
- QoS-aware routing using inter-source network decoding and multiple sources (QoS-aware w/ inter-source ND and multiple sources).

We simulate these routing schemes on 20 randomly generated network topologies containing 30 nodes. Node positions are chosen randomly in a square whose the length of each side is 200 meters. Each link capacity is determined by a distance between a transmitter and a receiver at each end of the link. Each link capacity is shown in Table 7.4, where $1 \leq c_l \leq 3$ units [74]. One unit is equal to 512 kbps. The successful data transmission of each link is randomly generated, where $Z \leq 1 - p_l \leq 1$. Variable Z is a set of numbers labeled on x-axis of all graphs in this section, where $Z \in \{0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$. The Gilbert-Elliot [115] channel model is used to simulate the burst packet loss pattern. In the model, there are two states. The first state represents a good channel condition, where there is no probability of packet loss. The second state represents a bad channel condition, where the probability of packet loss is varied depended on the setting. The transitional probabilities from the first state to the second state as well as from the second state to the first state are 0.08 and 0.02, respectively. The achievable data rate, which is the data rate destination can expect to receive from the transmission, is calculated from a sum of products between the transmission rates and reliability of data layers. When some layers cannot be transmitted from the source due to limited network resources, achievable data rate of those layers will be zero.

The objectives of our experiments are to show achievable data rate gains and QoS guarantee improvements of multicast transmissions, when the proposed routing scheme is used. In addition, we simulate scalable video multicast and show the improvement both in subjective and objective qualities. Video codec is based on H.264 Joint Scalable Video Model (JSVM) [24]. The group of picture (GOP) size is eight. Here, we consider only SNR scalability. Each video sequence is encoded to be three layers, *i.e.*, a base layer and two enhancement layers. The information values of a base layer, the first enhancement

Table 7.4: Distance thresholds for different transmission data rates

Distance(m)	Link Capacity (unit)
63-100	1
40-63	2
0-39	3

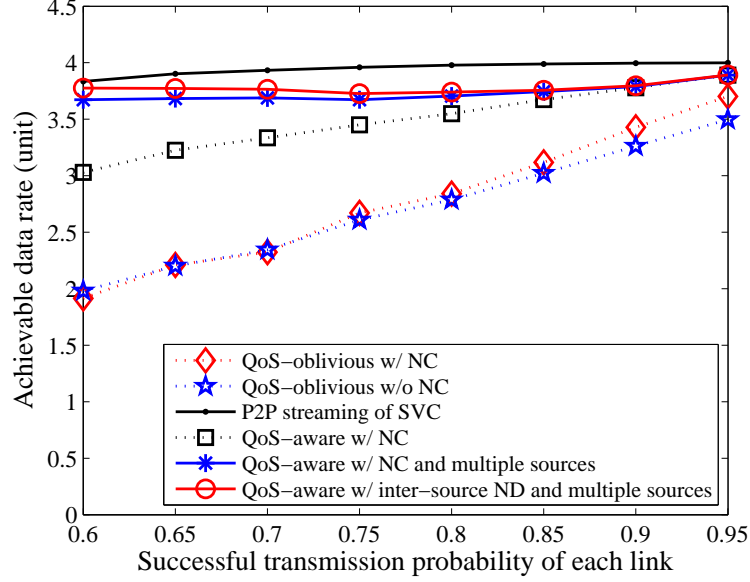


Figure 7.10: Comparison of achievable data rate of all layers for various link conditions.

layer, and the second enhancement layer are set to be three, two, and one, respectively. Note that the information value (κ) of more important layer will be higher than that of less important one. The frame rate and the resolution of the 4:2:0 YUV video sequences are 30 Hz and CIF (352×288 pixels), respectively. Error concealment mode of JSVM is enabled in our simulations. The simulation framework is partly based on the myEvalSVC framework [116]. There are four sources and three destinations in each network. Each destination requires three layers of data. The transmission rates of L_0 , L_1 , and L_2 are two, one, and one unit(s), respectively. The imposed QoS requirements of L_0 , L_1 , and L_2 for all destinations are 0.90, 0.85, and 0.80, respectively. Network coding is applied only in transmission of L_0 . The optimal route solution is computed using the Python programming language [106] together with PuLP package [107] and GLPK solver [111].

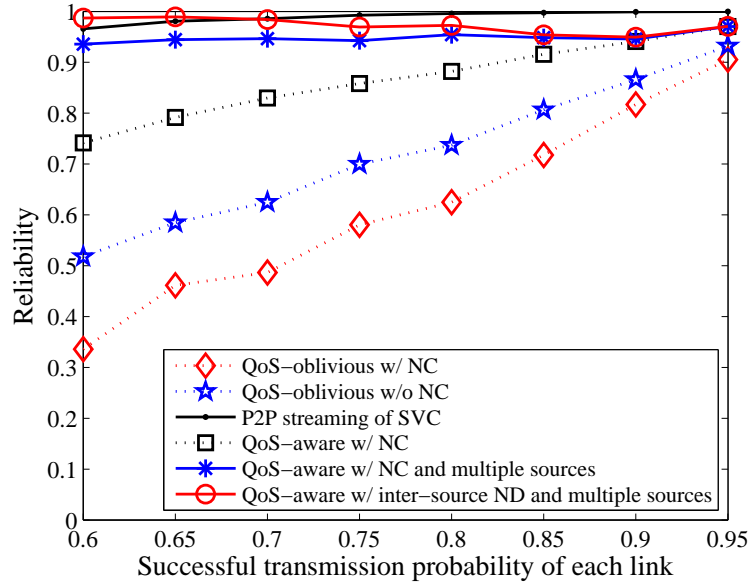


Figure 7.11: Comparison of reliability of data transmission of layer 0 for various link conditions.

7.4.1 Comparison of Achievable Data Rate and Reliability

Figure 7.10 shows the average achievable data rate with a successful transmission probability. The P2P streaming of SVC scheme gives the highest data rate to all destinations by utilizing all available network resources. An average data rate of the scheme slightly surpasses that of the QoS aware w/ inter-source ND and multiple sources by 5%. The performance gain is more obvious when we encounter hostile channel conditions. Our proposed routing scheme achieves an average gain of 8%, 36%, and 40% in average achievable data rate with respect to the QoS-aware w/ NC, QoS-oblivious w/ NC, and the QoS-oblivious w/o NC, respectively. Note that the achievable data rate gained by our multiple sources routing scheme slightly drops at some values of successful transmission probability. This is because we do not need to use multiple sources when QoS guarantee can be achieved by only one source.

Figs. 7.11, 7.12, and 7.13 demonstrate the average reliability in terms of successful transmission probability of data transmission of L_0 , L_1 , and L_2 from different routing schemes. The average reliability provided by the P2P streaming of SVC scheme is slightly higher than that given by our proposed scheme, *i.e.*, 2%, 6%, and 9% for L_0 , L_1 , and L_2 , respectively, whereas our proposed scheme gives better average reliability than those from

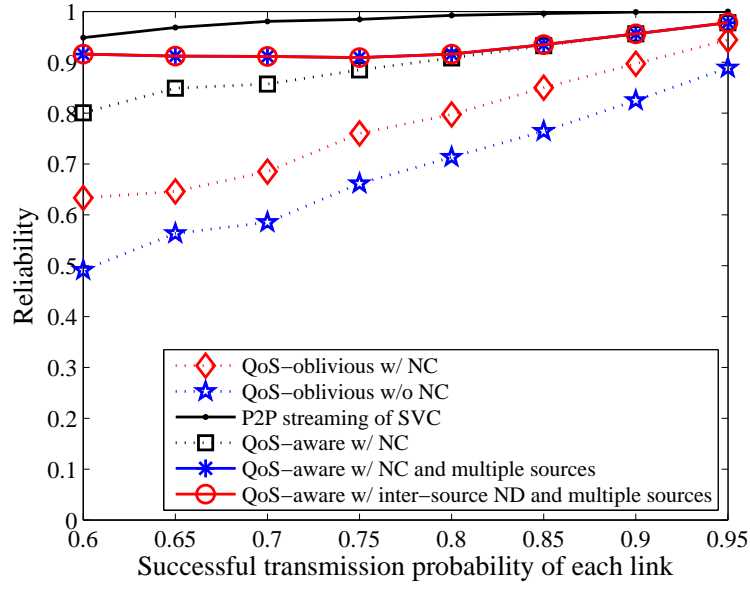


Figure 7.12: Comparison of reliability of data transmission of layer 1 for various link conditions.

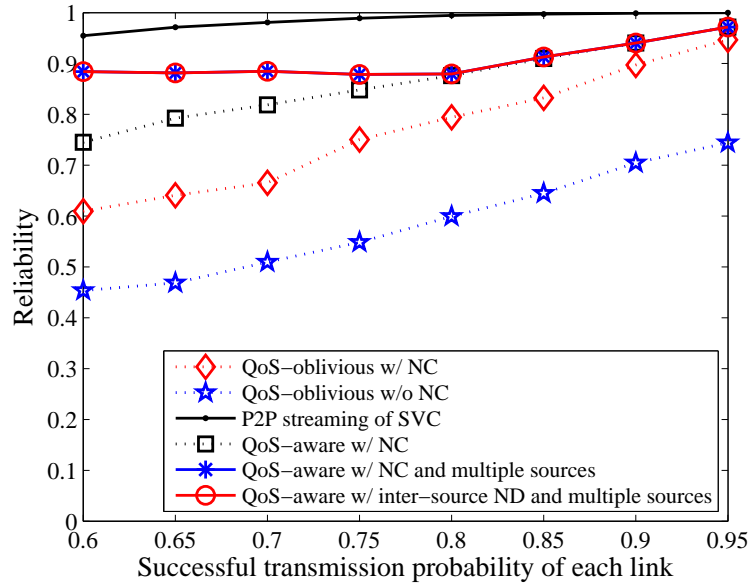


Figure 7.13: Comparison of reliability of data transmission of layer 2 for various link conditions.

other routing schemes. For transmission data layer 0, L_0 , the proposed routing scheme gives an average gain of 12%, 35%, and 58% over the QoS-aware w/ NC, QoS-oblivious w/o NC, and the QoS-oblivious w/ NC, respectively. For transmission data layer 1, L_1 , the proposed routing scheme gives an average gain of 4%, 20%, and 35% over the QoS-aware

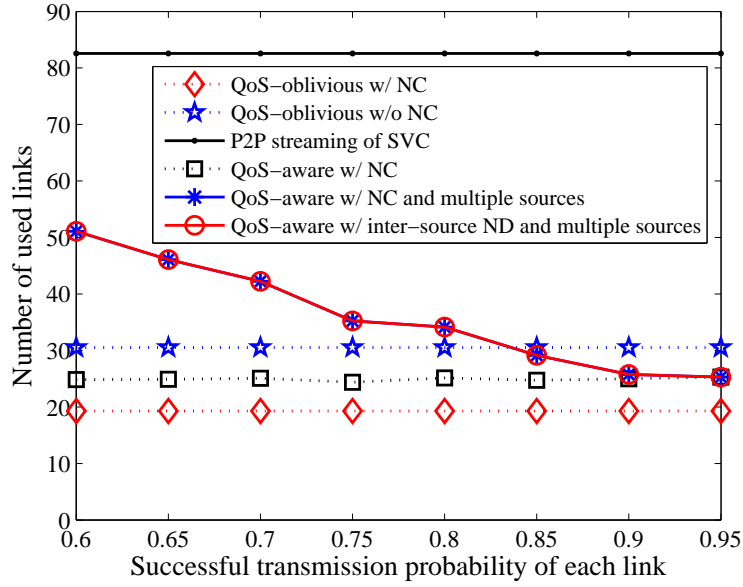


Figure 7.14: Comparison of the number of links used for each transmission scheme.

w/ NC, QoS-oblivious w/ NC, and the QoS-oblivious w/o NC, respectively. Finally, for transmission data layer 2, L_2 , the proposed routing scheme achieves an average gain of 5%, 18%, and 55% over the QoS-aware w/ NC, QoS-oblivious w/ NC, and the QoS-oblivious w/o NC, respectively. Note that the gain over the QoS-oblivious w/o NC is high because most data of L_2 cannot be transmitted due to the lack of available channels when the QoS-oblivious w/o NC is used.

Even though the P2P streaming SVC gives better results than our proposed QoS aware w/ inter-source and multiple sources, the advantages come with much higher network resource consumption. To be more specific, it consumes at least twice more channel usages than those used by other schemes. Figure 7.14 shows average numbers of used links from various routing schemes, which imply the network resource consumption in streaming video contents. Our proposed scheme uses much less network resource than that of the P2P streaming of SVC. When we consider the performance per channel use, our proposed scheme clearly outperforms the P2P streaming SVC. Moreover, the number of links used by our scheme decreases proportionally to packet loss rates of links. In other words, our scheme consumes less resource, when channel condition is in good states. Note that other single source routing schemes consume fewer channels than that of our proposed scheme but none of them can provide QoS guarantee to the scalable video streaming.

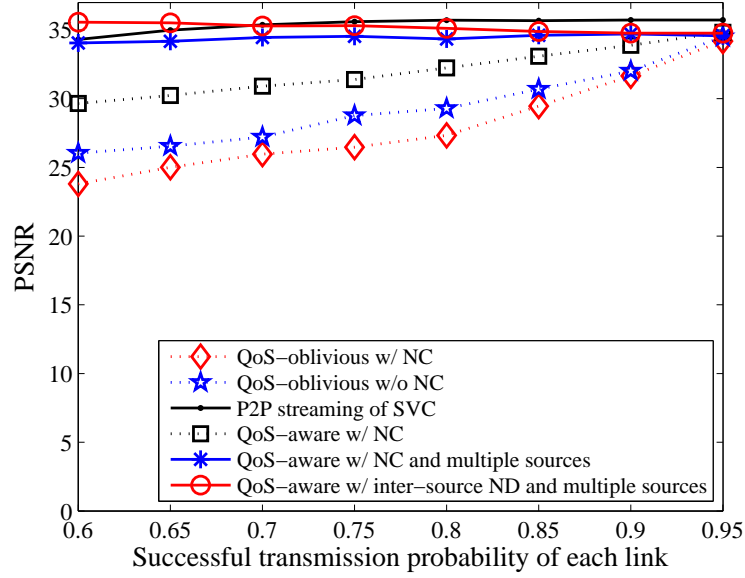


Figure 7.15: Comparison of the average PSNR of video sequence “Paris” for various link conditions.

7.4.2 Comparison of Objective and Subjective Qualities

In this section, we show the advantage in terms of peaked-signal-to-noise ratio (PSNR) in scalable video multicast with our proposed scheme. We use four video sequences consisting of “Paris”, “Bus”, “Football”, and “Mobile” in our simulation. These video sequences are available online at [117]. The packet loss rate of each video layer is calculated from the reliability in Section 7.4.1.

Figs. 7.15, 7.16, 7.17, and 7.18 show the PSNR comparison among different routing schemes of video sequences “Paris”, “Bus”, “Football”, and “Mobile”, respectively. As we can see from the results, the PSNR improvement using our proposed multicast scheme is significant, when the channel conditions are hostile, *e.g.*, the range of successful transmission probability is between 0.6 to 0.8. The average PSNR gain achieved by the P2P streaming of SVC is approximately 1-2 dB compared with that of our proposed scheme. The PSNR gain between our routing scheme and QoS-oblivious w/ NC is approximately 5-10 dB. This implies that the multiple sources and inter-source network decoding should be used, when the channel is hostile. When the channel is in good conditions, single source and network coding should be sufficient to give an acceptable transmitted video quality. Because multi-source multicast requires more network resource usages, it should

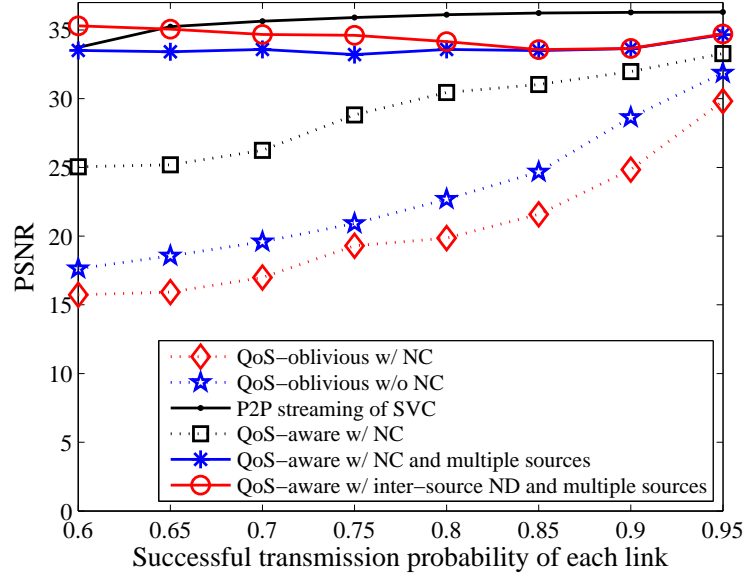


Figure 7.16: Comparison of the average PSNR of video sequence "Bus" for various link conditions.

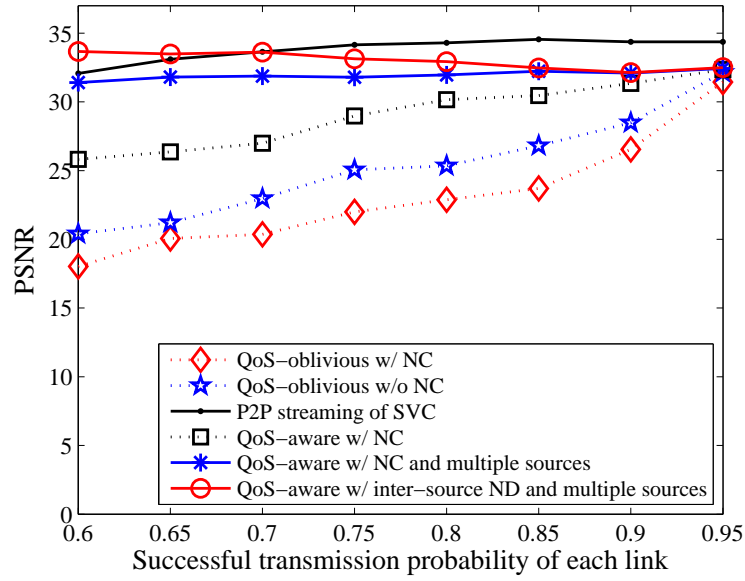


Figure 7.17: Comparison of the average PSNR of video sequence "Football" for various link conditions.

be selectively used.

Next, Figs. 7.19, 7.20, 7.21, and 7.22 show the subjective qualities of video frames from "Paris", "Bus", "Football" and "Mobile" video sequences, respectively. We compare video frames obtained from the proposed scheme and QoS-oblivious w/o NC. As we can

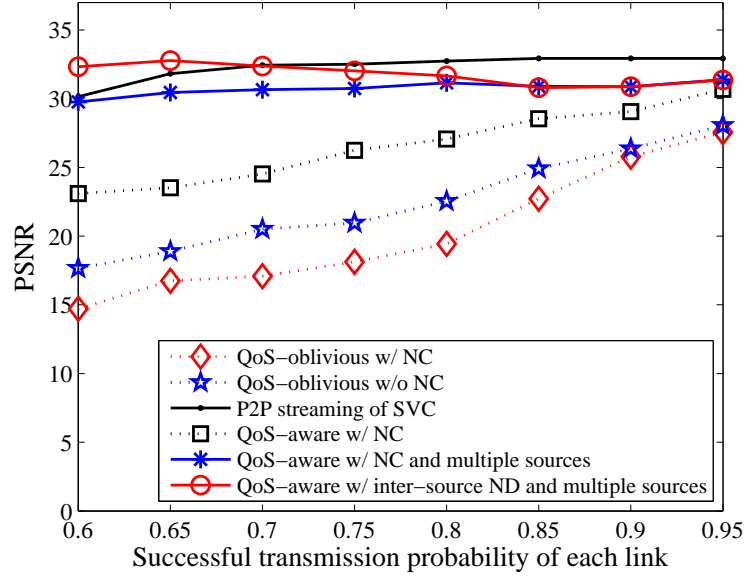


Figure 7.18: Comparison of the average PSNR of video sequence “Mobile” for various link conditions.



Figure 7.19: Visual example selected from video sequence “Paris” from various routing schemes : (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.

see from the results, our scheme gives superior visual qualities to the QoS-oblivious w/o NC, especially in high movement video sequences such as “Football” video sequence.

7.5 Conclusion

The reliable layered multicast using multiple sources and inter-source network decoding is proposed. The multi-source technique gives the path diversity that yields a better achievable data rate for layered transmissions. An optimization framework is formulated to find the optimal set of paths under several constraints. When the QoS guarantee of the

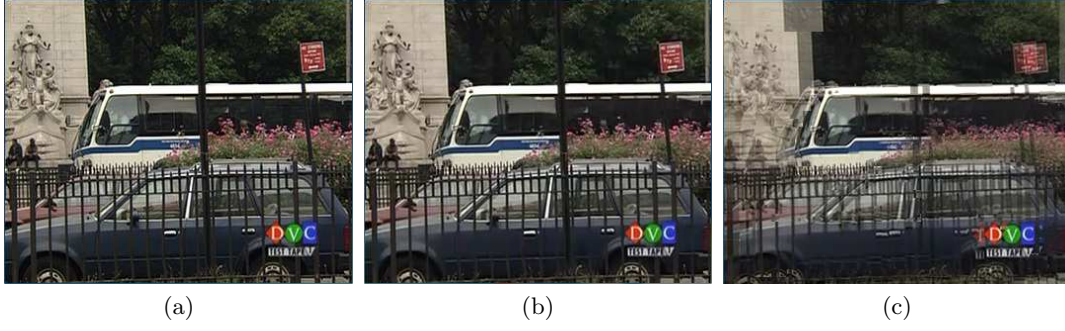


Figure 7.20: Visual example selected from video sequence “Bus” from various routing schemes: (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.



Figure 7.21: Visual example selected from video sequence “Football” from various routing schemes: (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.



Figure 7.22: Visual example selected from video sequence “Mobile” from various routing schemes: (a) Original, (b) QoS-aware w/ inter-source ND and multiple sources, and (c) QoS-oblivious w/o NC.

considering layered data cannot be achieved from the computed routes and some network resources are still available, a secondary source will find its optimal path and will transmit the same layered data to the considering destinations. This process will run repeatedly until the QoS guarantee is met or the network resources become unavailable. The network

coding is applied to the optimal set of paths. The inter-source network decoding is proposed to further improve the achievable data rate, whereas a sufficient number of network coded data are received not necessarily from the same source, the transmitted data can be recovered. Our experimental results show the improvement in the achievable data rate with QoS guarantee obtained from the proposed algorithm compared to previous works. The simulations of scalable video coding show the gains in both subjective and objective qualities under various video sequences and network topologies.

Chapter 8

Conclusion

This chapter gives a discussion about our presented approaches and the conclusion. Potential directions of future work are provided at the end of this chapter.

8.1 Discussion

The major part of link bandwidth in the Internet is occupied by content retrieval as a consequence of emerging content-oriented services and applications. In the process of content retrieval, a content delivery path often lies across multiple networks in the Internet, which can be categorized by location into the core and edge areas. In this dissertation, data transmission in both areas are facilitated by several novel content-oriented approaches to allow for efficient content delivery.

Data centers, which are major hosts of content servers, typically have direct connection to the core area of the Internet. The core area is therefore the first place where content starts a journey toward end-users. This area is prone to network congestion because it is loaded with a large amount of content retrieval traffic from end-users. Network caching has been an effective solution to network congestion. However, existing implementation of network caching, such as Web caches and Content Delivery Networks (CDNs), involves many unnecessarily complicated systems. Content-Centric Networking (CCN) architecture is considered to be a new solution to network congestion since it natively supports network caching in its protocol. Existing routing and caching schemes for CCN cannot

fully utilize in-network caches. We thus introduce a new cooperative routing scheme and a probabilistic caching scheme for CCN. The cooperative routing scheme optimizes data transmission paths based on the data similarity extracted from content retrieval statistics. The probabilistic caching scheme lets CCN routers randomly cache data with a caching probability to reduce redundant copies of the same content cached by CCN routers along a transmission path. Both schemes cooperatively improve the utilization of in-network caches, leading to more efficient data transmission in the core area of the Internet. The content delivery completes the first part of data transmission in the core area and proceeds in the edge area.

In the edge area, multi-hop wireless networks would be prevalent as a result of the increasing number of wireless devices, accelerated by advancement in modern electronic and computing technologies. Unreliable quality and limited bandwidth are common issues in the transmission links of these networks. On the other hand, streaming bandwidth-demanding content over such networks has become the norm. Moreover, the quality of reproduced content at an end-user is susceptible to data loss. Quality-of-service (QoS) is thus necessary for data transmission in multi-hop wireless networks. We propose two new QoS-aware routing schemes for multi-hop wireless networks, one is for unicast and the other is for multicast. In essence, our proposed routing schemes select optimal routes in given networks according to QoS requirements and the importance of data. Network coding techniques are exploited to further improve network resource utilization. Our routing schemes not only better guarantee QoS of data transmission in multi-hop wireless networks than several existing schemes, but also achieve efficient utilization of network resources.

We propose several content-oriented approaches for efficient data transmission in the Internet. Results from extensive evaluation show that our schemes can address data transmission more efficiently than several existing schemes which ignore the properties of data. Our proposed schemes may be orthogonal, considering that they are applicable to networks in different areas. Nevertheless, they can be jointly used to enable efficient content delivery over heterogeneous networks in the future Internet. For instance, a new framework for efficient content delivery can be described as follows. We can utilize CCN architecture along with the cooperative routing and probabilistic caching schemes to facilitate data

transmission in the core area. Much of redundant traffic could be eliminated by efficient in-network caches, thus network congestion would rarely happen. Upon receiving the data from the core area, networks in the edge area forward them to end-users. Some of data may be lost in the this area before reaching end-users as a result of unreliable wireless links and limited link bandwidth. In this case, our QoS-aware routing schemes can be used both to achieve a QoS guarantee and to prioritize data transmission. The cooperation among all of our proposed schemes would enable efficient content delivery in the future Internet.

It is worth to note that our work has some limitations. For example, our Optimal Cooperative Routing Protocol (OCRP) is driven by exchanges of signalling packets between CCN routers. Excessive signalling packets could affect the performance of particular CCN routers and routing stability. The number of signalling packets created depends on several parameters, such as the number of CCN routers, threshold of popularity index, and traffic sampling interval. These parameters are unique to each network and are not necessarily the same as those used by this work. In the performance analysis of a probabilistic caching scheme, we model the requests of content with Independent Reference Model (IRM). However, recent studies in [118] point out that IRM may reduce the accuracy of caching analysis because temporal locality of requests is ignored. We also assume Zero Download Delay (ZDD) in this work. In fact, download delay occurs for every cache-miss and may contribute to the discrepancy between the analytical results and the actual performance in real networks. Next, let us consider our QoS-aware routing schemes for multi-hop wireless networks. In our routing scheme for unicast transmission, the optimal route selection and CNC establishment are separate steps. An optimal route for the entire unicast sessions is calculated in the first step. Upon the obtained route, our Cooperative Network Coding Establishment (CNCE) algorithm decides whether CNC will be applied at which intermediate node. If this optimal route does not match any CNC structure, the network will not set up any CNC. To improve the opportunity of CNC, the optimal route calculation should be aware of CNC structures and QoS requirements. However, this would cost much more complexity than our proposed scheme. Finally, in our work on QoS-aware routing scheme for multicast transmission, we assume that only a multicast session exists in a multi-hop wireless network. In practice, more than one session may occur simultaneously and may increase contention for network resources. Hence, a multiple

source technique would be limitedly used in some networks.

8.2 Conclusion

Network congestion and lack of QoS guarantee are critical problems in the modern Internet, owing to an explosion in popularity of applications and services with content-oriented requirements. Many of existing solutions are either inefficient or unnecessarily complicated as they are based on a host-to-host communication model, which treats data as anonymous packets without considering properties of data. Aiming to find solutions that better fit these new requirements, we propose several content-oriented approaches for efficient data transmission in the core and edge areas of the future Internet.

We first address the problem of network congestion in the core area. We consider CCN to be a solution to network congestion as it natively supports network caching in its protocol. We introduce a cooperative routing for CCN and propose OCRP for CCN to improve the utilization of in-network caches in content-centric networks. OCRP makes use of content retrieval statistics collected from CCN routers for route optimization. An optimal route is obtained by solving an integer linear optimization problem, whose constraints are flow conservation constraint, cache contention mitigating constraint and path length constraint. Simulation results show that OCRP yields an improvement in the server load and round-trip time over the shortest path routing. Not only the routing but also the caching contributes to the utilization of in-network caches. We thus propose and analyse a probabilistic caching scheme, aiming to justify its feasibility in CCN. The behavior of the probabilistic caching scheme is investigated by means of computer simulation and mathematical analysis. Simulation results show that the behavior of a probabilistic caching scheme varies as a function of cache replacement policy. The hit rate and the duration of the initial state of a probabilistic caching system are inverse functions of a caching probability. In addition, we propose a Markov chain-based analytical model for performance analysis of the probabilistic caching scheme. The insights from the analytical model show that a combination of a probabilistic caching scheme and LRU gives an effective cache management scheme for CCN routers. However, if a CCN router cannot afford LRU due to its computational power limitation, RR is preferred to FIFO.

We proceed to address QoS guarantee for data transmission in multi-hop wireless networks, which would be common in the edge area. We propose a QoS-aware routing scheme for unicast transmission using cooperative network coding (CNC). We set up an integer linear optimization problem to obtain an optimal route of each unicast flow. We develop CNCE algorithm to decide whether or not CNC will be performed at an intermediate node. The criteria of this algorithm are based the A-B, Y, and X structures found in the network and QoS requirements of unicast sessions. Simulation results show that our routing scheme yields better throughput and higher channel use efficiency with the QoS guarantee of unicast flows than QoS-oblivious approaches. For multicast transmission, we propose a QoS-aware routing scheme with the help of a multi-source technique and inter-source network decoding. We formulate an integer linear optimization problem for calculating an optimal route based on QoS requirements and available network resources. We exploit a multi-source technique to improve the reliability of transmitted data through path diversity. Inter-source network decoding is exploited to improve the achievable data rate, where a data layer can be recovered by using a sufficient number of network coded data that are not necessarily obtained from the same source. Results from computer simulation show that our routing scheme yields a significant improvement both in subjective and in objective qualities of scalable videos compared to several existing schemes.

8.3 Future Work

8.3.1 Network Coding in Content-Centric Networks

Network coding could be used to improve the utilization of content-centric networks as well as to reduce content-retrieval time. An example of network coding in a content-centric network is shown in Fig. 8.1. We assume that each link in the network can carry only a chunk of data per a transmission slot. Node 4 is a content requester who demands content X that consists of two data chunks: x_1 and x_2 . Node 4 sends an Interest packet for content $X(x_1, x_2)$ to node 3. Node 3 does not have content X in its CS so it forwards the Interest packet to all potential content repositories. Both nodes 1 and 2 have data chunks x_1 and x_2 , thus becoming the potential content repositories for content X . Nodes 1 and 2 respond to received Interest packets with Data packets containing network-coded data $a_1x_1 + a_2x_2$

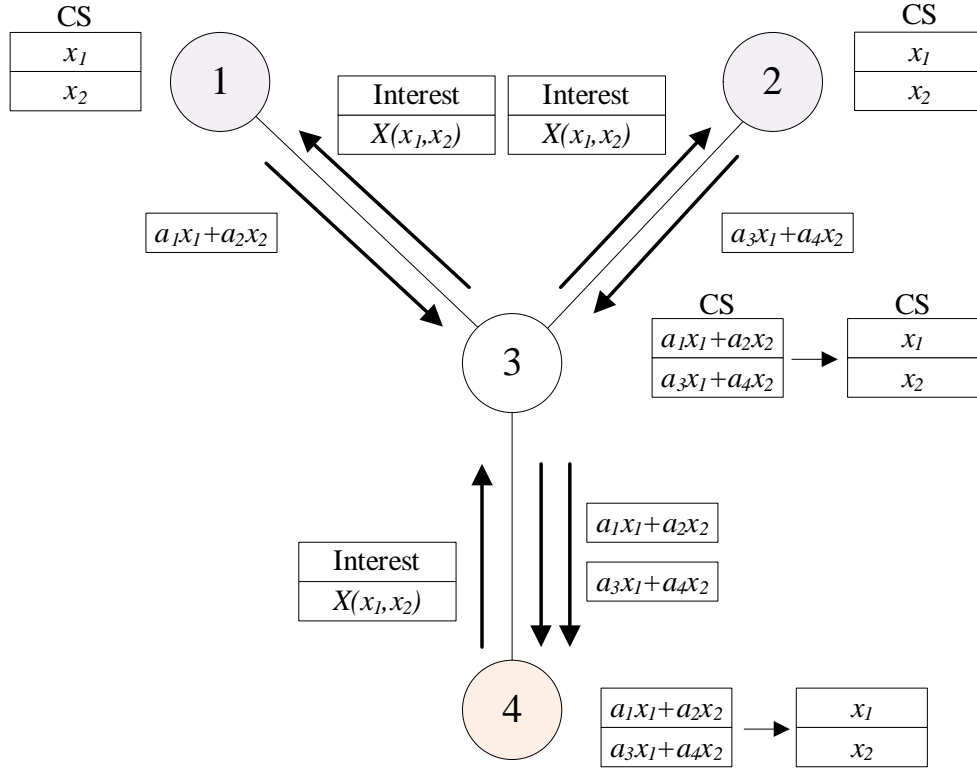


Figure 8.1: Example of network coding in a content-centric network.

and $a_3x_1 + a_4x_2$, respectively, where a_1 , a_2 , a_3 , and a_4 are coding coefficients. By randomly choosing these coding coefficients from a large enough field, the probability that coding vectors $[a_1, a_2]$ and $[a_3, a_4]$ are linearly independent is high. Node 3 forwards these Data packets to Node 4 and caches both network-coded data in its CS. Node 4 obtains content X by decoding the two network-coded data inside received Data packets. Node 3 can also decode the caches network-coded data to obtain data chunks x_1 and x_2 . It can optionally cache data chunks x_1 and x_2 in its CS to become a potential content repository of content X .

On the other hand, if network coding is not allowed, nodes 1 and 2 may respond to the Interest packet with Data packets containing the same data chunk. This event can happen because node 1 does not know which data chunk node 2 will send, and vice versa. When both nodes 1 and 2 send Data packets containing the same chunk to node 3, the link bandwidth is wasted as node 3 will forward only one of them to node 4. Moreover, node 4 needs to send another Interest packet for the data chunk that has not been received, resulting in a longer content-retrieval time. As we can see from this

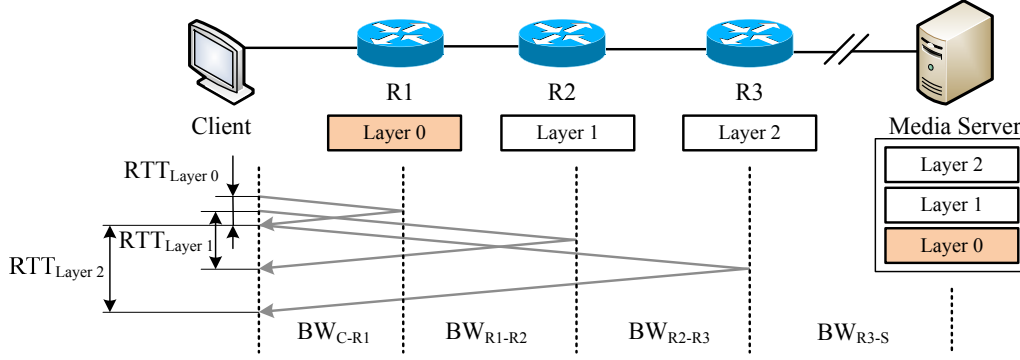


Figure 8.2: RTT variability of scalable video transmissions in a content-centric network.

example, network coding has potential to improve data transmission in content-centric networks. An investigation into the implementation of network coding in CCN should give an interesting future work.

8.3.2 Scalable Video Streaming in Content-Centric Networks

Scalable video streaming is one of the applications that are expected to highly benefit from the in-network caches. It holds several key features such as a high caching efficiency and a seamless quality adaptation [119, 120]. However, scalable video streaming in CCN would encounter several fundamental problems associated with bandwidth estimation. Figure 8.2 shows an example of the problems. Data layers of a scalable video bitstream may be cached by different CCN routers. As a result, the round-trip times (RTTs) and bandwidths for downloading chunks of different data layers could fluctuate in a unprecedentedly wide range.

Existing bandwidth estimations rely on an estimated RTT, which is a function of the RTTs of recent transmissions [121]. In content-centric networks, these approaches would be severely inaccurate due to a large variance of the estimated RTT. At a client, a quality adaptation logic of scalable video streaming utilizes the estimated bandwidth to determine the data rate, which affects the quality of video. The data rate is generally set to be as close as possible to the estimated bandwidth for the best video quality. Inaccurate bandwidth estimation leads to either underutilization of actual bandwidth or frequent video quality fluctuation, both of which degrade the experiences of the client. Therefore, an investigation into a framework for efficient scalable video streaming in content-centric networks is a challenging future work.

8.3.3 Cache Management Schemes for Video Workloads

According to the conclusion of the dissertation, a combination of a probabilistic caching scheme and LRU gives an effective cache management scheme for content-centric networks. Nevertheless, the conclusion is met under an assumption that all content objects are uncorrelated and have the same size. It might not hold for video workloads. In general, a video bitstream is divided into multiple chunks to be transmitted in heterogeneous networks. Upon receiving video requests, a video source (or a video server) sends video chunks to a client, one after another. The order of video chunks is deterministic, thus exhibiting strong temporal locality of traffic. A question of whether a combination of probabilistic caching scheme and LRU is suitable for video streaming workloads is worthwhile to consider. Furthermore, the video streaming traffic is expected to approximately account for 80% of all consumer Internet traffic in 2018 [5]. Therefore, it is reasonable to design cache management schemes based on the characteristics of video workloads.

Bibliography

- [1] Sandvine Incorporated ULC. Global internet phenomena report. <https://www.sandvine.com/trends/global-internet-phenomena/>. [Online; accessed 6-March-2015].
- [2] Akamai Technologies. The akamai state of the internet report. <http://www.akamai.com/stateoftheinternet/>. [Online; accessed 6-March-2015].
- [3] Bill Krogfoss, Marcus Weldon, and Lev Sofman. Internet architecture evolution and the complex economies of content peering. *Bell Labs Technical Journal*, 17(1):163–184, June 2012.
- [4] Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. Youtube around the world: Geographic popularity of videos. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12*, pages 241–250, New York, NY, USA, 2012. ACM.
- [5] Cisco. Cisco visual networking index: Forecast and methodology, 20132018. Technical report, Cisco, June 2014.
- [6] James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach*. Pearson Education Limited, England, 6nd edition, 2012.
- [7] 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, pages 1–12, New York, NY, USA, 2009. ACM.
- [8] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello. From content delivery today to information centric networking. *Computer Networks*, 57(16):3116 – 3127, 2013. Information Centric Networking.
- [9] NDN project. Named Data Networking. <http://named-data.net/>. [Online; accessed 10-June-2015].
- [10] G. Rossini, D. Rossi, M. Garetto, and E. Leonardi. Multi-terabyte and multi-gbps information centric routers. In *INFOCOM, 2014 Proceedings IEEE*, pages 181–189, April 2014.
- [11] Lan Wang, A K M Mahmudul Hoque, Cheng Yi, Adam Alyyan, and Beichuan Zhang. Ospfn: An ospf based routing protocol for named data networking. Technical report, NDN Technical Report NDN-0003, July 2012.

- [12] Huichen Dai, Jianyuan Lu, Yi Wang, and Bin Liu. A two-layer intra-domain routing scheme for named data networking. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 2815–2820, Dec 2012.
- [13] Kideok Cho, Munyoung Lee, Kunwoo Park, T.T. Kwon, Yanghee Choi, and Sangheon Pack. Wave: Popularity-based and collaborative in-network caching for content-oriented networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 316–321, March 2012.
- [14] 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, pages 55–60, New York, NY, USA, 2012. ACM.
- [15] D. Rossi and G Rossini. Caching performance of content centric networks under multi-path routing (and more). Technical report, Tech. Rep., Telecom ParisTech, July 2011.
- [16] Alec Wolman, M. Voelker, Nitin Sharma, Neal Cardwell, Anna Karlin, and Henry M. Levy. On the scale and performance of cooperative web proxy caching. *SIGOPS Oper. Syst. Rev.*, 33(5):16–31, December 1999.
- [17] 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*, HotNets-X, pages 1:1–1:6, New York, NY, USA, 2011. ACM.
- [18] C. Bernardini, T. Silverston, and O. Festor. Mpc: Popularity-based caching strategy for content centric networks. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3619–3623, June 2013.
- [19] Hao Wu, Jun Li, Tian Pan, and Bin Liu. A novel caching scheme for the backbone of named data networking. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3634–3638, June 2013.
- [20] S. Saha, A. Lukyanenko, and A. Yla-Jaaski. Cooperative caching through routing control in information-centric networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 100–104, April 2013.
- [21] 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, pages 61–66, New York, NY, USA, 2013. ACM.
- [22] Parth H. Pathak and Rudra Dutta. Centrality-based power control for hot-spot mitigation in multi-hop wireless networks. *Computer Communications*, 35(9):1074 – 1085, 2012. Special Issue: Wireless Sensor and Robot Networks: Algorithms and Experiments.
- [23] M.E.M. Campista, P.M. Esposito, I.M. Moraes, L.H.M.K. Costa, O.C.M.B. Duarte, D.G. Passos, C.V.N. de Albuquerque, D.C.M. Saade, and M.G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *Network, IEEE*, 22(1):6–12, Jan 2008.

- [24] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the h.264/avc standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, 2007.
- [25] Theodore S. Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall, 2 edition, January 2002.
- [26] Luigi Rizzo. Effective erasure codes for reliable computer communication protocols. *SIGCOMM Comput. Commun. Rev.*, 27(2):24–36, April 1997.
- [27] Kihong Park and Wei Wang. Afec: an adaptive forward error correction protocol for end-to-end transport of real-time traffic. In *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on*, pages 196–205, Oct 1998.
- [28] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.
- [29] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3):497–510, 2008.
- [30] Jin Zhang and Qian Zhang. Cooperative network coding-aware routing for multi-rate wireless networks. In *INFOCOM 2009*, pages 181–189, 2009.
- [31] E. Tuncel and K. Rose. Additive successive refinement. *IEEE Trans. Inf. Theor.*, 49(8):1983–1991, September 2006.
- [32] Toru Hasegawa. A survey of the research on future internet and network architectures. *IEICE Trans. Commun.*, E98-B(2):1385–1401, 2013.
- [33] J. Pan, S. Paul, and R. Jain. A survey of the research on future internet architectures. *Communications Magazine, IEEE*, 49(7):26–36, July 2011.
- [34] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, and G.C. Polyzos. A survey of information-centric networking research. *Communications Surveys Tutorials, IEEE*, 16(2):1024–1049, Second 2014.
- [35] M. Mangili, F. Martignon, and A. Capone. A comparative study of content-centric and content-distribution networks: Performance and bounds. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 1403–1409, Dec 2013.
- [36] Haowei Yuan, Tian Song, and P. Crowley. Scalable ndn forwarding: Concepts, issues and principles. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pages 1–9, July 2012.
- [37] Andrea Detti, Nicola Blefari Melazzi, Stefano Salsano, and Matteo Pomposini. Conet: A content centric inter-networking architecture. In *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, ICN ’11*, pages 50–55, New York, NY, USA, 2011. ACM.
- [38] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga. Potential based routing as a secondary best-effort routing for information centric networking (icn). *Computer Networks*, 57(16):3154 – 3164, 2013. Information Centric Networking.

- [39] T. Janaszka, D. Bursztynowski, and M. Dzida. On popularity-based load balancing in content networks. In *Teletraffic Congress (ITC 24), 2012 24th International*, pages 1–8, Sept 2012.
- [40] Wei Koong Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache ”less for more” in information-centric networks. In *Proceedings of the 11th International IFIP TC 6 Conference on Networking - Volume Part I*, IFIP’12, pages 27–40, Berlin, Heidelberg, 2012. Springer-Verlag.
- [41] III King, W. F. Analysis of paging algorithms. In *Proc. IFIP Cong.*, volume 3, pages 485–490, August 1971.
- [42] Erol Gelenbe. A unified approach to the evaluation of a class of replacement algorithms. *Computers, IEEE Transactions on*, C-22(6):611–618, June 1973.
- [43] Asit Dan and Don Towsley. An approximate analysis of the lru and fifo buffer replacement schemes. *SIGMETRICS Perform. Eval. Rev.*, 18(1):143–152, April 1990.
- [44] Jaime Garcia-Reinoso, Ivan Vidal, David Diez, Daniel Corujo, and Rui L. Aguiar. Analysis and enhancements to probabilistic caching in content-centric networking. *The Computer Journal*, 2015.
- [45] E.J. Rosensweig, J. Kurose, and D. Towsley. Approximate models for general cache networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.
- [46] Wang Guoqing, Huang Tao, Liu Jiang, Chen Jianya, and Liu Yunjie. Approximate models for ccn data transfer in general topology. *Communications, China*, 11(7):40–47, July 2014.
- [47] E.J. Rosensweig, D.S. Menasche, and J. Kurose. On the steady-state of cache networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 863–871, April 2013.
- [48] Hao Che, Ye Tung, and Z. Wang. Hierarchical web caching systems: modeling, design and experimental results. *Selected Areas in Communications, IEEE Journal on*, 20(7):1305–1314, Sep 2002.
- [49] Christine Fricker, Philippe Robert, and James Roberts. A versatile and accurate approximation for lru cache performance. In *Proceedings of the 24th International Teletraffic Congress, ITC ’12*, pages 8:1–8:8. International Teletraffic Congress, 2012.
- [50] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. In *INFOCOM, 2014 Proceedings IEEE*, pages 2040–2048, April 2014.
- [51] Giuseppe Bianchi, Andrea Detti, Alberto Caponi, and Nicola Blefari Melazzi. Check before storing: What is the performance price of content integrity verification in lru caching? *SIGCOMM Comput. Commun. Rev.*, 43(3):59–67, July 2013.
- [52] Ho Tracey, M. Medard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413–4430, Oct 2006.
- [53] Christina Fragouli, Jean-Yves Le Boudec, and Jörg Widmer. Network coding: an instant primer. *SIGCOMM Comput. Commun. Rev.*, 36:63–68, 2006.

- [54] D. Traskov, N. Ratnakar, D.S. Lun, R. Koetter, and M. Medard. Network coding for multiple unicasts: An approach based on linear optimization. In *IEEE International Symposium on Information Theory*, pages 1758–1762, 2006.
- [55] Zongpeng Li and Baochun Li. Network coding: The case of multiple unicast sessions. In *in Proceedings of the 42nd Allerton Annual Conference on Communication, Control, and Computing*, 2004.
- [56] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pages 1028–1036, 2007.
- [57] S. Sengupta, S. Rayanchu, and S. Banerjee. Network coding-aware routing in wireless networks. *IEEE/ACM Transactions on Networking*, 18(4):1158–1170, 2010.
- [58] Junning Liu, D. Goeckel, and D. Towsley. Bounds on the gain of network coding and broadcasting in wireless networks. In *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pages 724–732, 2007.
- [59] Xin Wei, Li Zhao, Ji Xi, and Qingyun Wang. Network coding aware routing protocol for lossy wireless networks. In *5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom’09)*, pages 1–4, 2009.
- [60] O. Alay, T. Korakis, Yao Wang, E. Erkip, and S.S. Panwar. Layered wireless video multicast using relays. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(8):1095–1109, 2010.
- [61] H. Seferoglu and A. Markopoulou. Opportunistic network coding for video streaming over wireless. In *Packet Video 2007*, pages 191–200, 2007.
- [62] Abinash Mahapatra, Kumar Anand, and Dharma P. Agrawal. Qos and energy aware routing for real-time traffic in wireless sensor networks. *Comput. Commun.*, 29(4):437–445, February 2006.
- [63] Sucha Supittayapornpong, Poompat Saengudomlert, and Wuttipong Kumwilaisak. A framework for reliability aware layered multi-cast in lossy networks with network coding. *Comput. Commun.*, 33(14):1651–1663, September 2010.
- [64] Wei Pu, Chong Luo, Feng Wu, and Chang Wen Chen. Qos-driven network coded wireless multicast. *IEEE Transactions on Wireless Communications*, 8(11):5662–5670, 2009.
- [65] C. Greco, I.D. Nemoianu, M. Cagnazzo, and B. Pesquet-Popescu. A network coding scheduling for multiple description video streaming over wireless networks. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO) 2012*, pages 1915–1919, 2012.
- [66] Hayoung Oh and Chong kwon Kim. Network coding-based mobile video streaming over unreliable wireless links. *IEEE Communications Letters*, 17(2):281–284, 2013.
- [67] Yuwang Yang, Chunshan Zhong, Yamin Sun, and Jingyu Yang. Network coding based reliable disjoint and braided multipath routing for sensor networks. *Journal of Network and Computer Applications*, 33(4):422 – 432, 2010.

- [68] Lei Wang, Yuwang Yang, and Wei Zhao. Network coding-based multipath routing for energy efficiency in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):115, 2012.
- [69] Zhaolong Ning, Qingyang Song, Lei Guo, and A. Jamalipour. Throughput improvement by network coding and spatial reuse in wireless mesh networks. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 4572–4577, Dec 2013.
- [70] M.F. Uddin, C. Rosenberg, Weihua Zhuang, P. Mitran, and A. Girard. Joint routing and medium access control in fixed random access wireless multihop networks. *Networking, IEEE/ACM Transactions on*, 22(1):80–93, Feb 2014.
- [71] Ying Zhu, Baochun Li, and Jiang Guo. Multicast with network coding in application-layer overlay networks. *Selected Areas in Communications, IEEE Journal on*, 22(1):107–120, Jan 2004.
- [72] Niveditha Sundaram, Parameswaran Ramanathan, and Suman Banerjee. Multirate media stream using network coding. In *in Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.
- [73] MinJi Kim, D. Lucani, Xiaomeng Shi, Fang Zhao, and M. Medard. Network coding for multi-resolution multicast. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.
- [74] Xu Chenguang, Xu Yinlong, Zhan Cheng, Wu Ruizhe, and Wang Qingshan. On network coding based multirate video streaming in directed networks. In *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationala*, pages 332–339, April 2007.
- [75] Jin Zhao, Fan Yang, Qian Zhang, Zhensheng Zhang, and Fuyan Zhang. Lion: Layered overlay multicast with network coding. *Multimedia, IEEE Transactions on*, 8(5):1021–1032, Oct 2006.
- [76] Yong Ding, Yang Yang, and Li Xiao. Multi-path routing and rate allocation for multi-source video on-demand streaming in wireless mesh networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2051–2059, April 2011.
- [77] Te Sun Han. Multicasting multiple correlated sources to multiple sinks over a noisy channel network. *Information Theory, IEEE Transactions on*, 57(1):4–13, Jan 2011.
- [78] Danjue Li, Chen-Nee Chuah, Gene Cheung, and S.J. Ben Yoo. Muvis: Multi-source video streaming service over wlans. *Communications and Networks, Journal of*, 7(2):144–156, June 2005.
- [79] N. Ramzan, E. Quacchio, T. Zgaljic, S. Asioli, L. Celetto, E. Izquierdo, and F. Rovati. Peer-to-peer streaming of scalable video in future internet applications. *Communications Magazine, IEEE*, 49(3):128–135, March 2011.
- [80] Xiaofeng Xu, Yao Wang, S.S. Panwar, and K.W. Ross. A peer-to-peer video-on-demand system using multiple description coding and server diversity. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1759–1762 Vol. 3, Oct 2004.

- [81] R.P. Selvam and V. Palanisamy. An optimized cluster based approach for multi-source multicast routing protocol in mobile ad hoc networks with differential evolution. In *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on*, pages 115–120, March 2012.
- [82] Tracey Ho, Muriel Mdard, Michelle Effros, and Ralf Koetter. Network coding for correlated sources. In *in CISS*, 2004.
- [83] O. Trullols-Cruces, J.M. Barcelo-Ordinas, and M. Fiore. Exact decoding probability under random linear network coding. *Communications Letters, IEEE*, 15(1):67–69, January 2011.
- [84] Xubo Zhao. Notes on "exact decoding probability under random linear network coding". *Communications Letters, IEEE*, 16(5):720–721, May 2012.
- [85] Yusung Kim and Ikjun Yeom. Performance analysis of in-network caching for content-centric networking. *Computer Networks*, 57(13):2465 – 2482, 2013.
- [86] D. Rossi and G. Rossini. On sizing ccn content stores by exploiting topological information. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 280–285, March 2012.
- [87] Cheng Yi, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. Adaptive forwarding in named data networking. *SIGCOMM Comput. Commun. Rev.*, 42(3):62–67, June 2012.
- [88] J. Reynolds and J. Postel. Assigned numbers. <http://tools.ietf.org/html/rfc1700>. [Online; accessed 6-March-2015].
- [89] International Business Machines Corporation (IBM). CPLEX Optimizer: High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming. <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>. [Online; accessed 13-March-2015].
- [90] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Yusheng Ji. Cooperative routing protocol for content-centric networking. In *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, pages 699–702, Oct 2013.
- [91] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.
- [92] National Institute of Informatics. SINET4: Science Information NETwork 4. <http://www.sinet.ad.jp/Network/>. [Online; accessed 6-March-2015].
- [93] European Commission Communication Networks, Content and Technology. GÉANT PROJECT Network Topology. http://www.geant.net/Network/The_Network/Pages/Network-Topology.aspx. [Online; accessed 6-March-2015].
- [94] Indiana University. Internet2 NOC Support - Engineering Map. <http://noc.net.internet2.edu/i2network/maps-documentation/maps.html>. [Online; accessed 6-March-2015].
- [95] The igraph core team. Get started with python-igraph. <http://igraph.org/python/>. [Online; accessed 6-March-2015].

- [96] Yuemei Xu, Yang Li, Tao Lin, Guoqiang Zhang, Zihou Wang, and Song Ci. A dominating-set-based collaborative caching with request routing in content centric networking. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3624–3628, June 2013.
- [97] Edward G. Jr. Coffman and Peter J. Denning. *Operating Systems Theory*. Prentice Hall Professional Technical Reference, 1973.
- [98] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 310–315, March 2012.
- [99] Nikolaos Laoutaris, Hao Che, and Ioannis Stavrakakis. The lcd interconnection of lru caches and its analysis. *Perform. Eval.*, 63(7):609–634, July 2006.
- [100] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Yusheng Ji. Performance of probabilistic caching and cache replacement policies for content-centric networks. In *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*, pages 99–106, Sept 2014.
- [101] Simone Basso, Michela Meo, Antonio Servetti, and Juan Carlos De Martin. Estimating packet loss rate in the access through application-level measurements. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Measurements Up the Stack, W-MUST '12*, pages 7–12, New York, NY, USA, 2012. ACM.
- [102] Angela Wang, Cheng Huang, Jin Li, and Keith W. Ross. Queen: Estimating packet loss rate between arbitrary internet hosts. In Sue B. Moon, Renata Teixeira, and Steve Uhlig, editors, *PAM*, volume 5448 of *Lecture Notes in Computer Science*, pages 57–66. Springer, 2009.
- [103] D. Benyamina, A. Hafid, and M. Gendreau. Wireless mesh networks design – a survey. *Communications Surveys Tutorials, IEEE*, 14(2):299–310, 2012.
- [104] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.
- [105] CoinMP. <https://projects.coin-or.org/CoinMP>. [Online; accessed 6-March-2015].
- [106] Python Programming Language. <http://www.python.org>. [Online; accessed 6-March-2015].
- [107] Optimization with PuLP. <http://www.coin-or.org/PuLP>. [Online; accessed 6-March-2015].
- [108] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom '03*, pages 66–80, New York, NY, USA, 2003. ACM.
- [109] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

- [110] Ki-Dong Lee and V.C.M. Leung. Fair allocation of subcarrier and power in an ofdma wireless mesh network. *Selected Areas in Communications, IEEE Journal on*, 24(11):2051–2060, Nov 2006.
- [111] GNU Linear Programing Lit (GLPK). <http://www.gnu.org/software/glpk/>. [Online; accessed 6-March-2015].
- [112] Saran Tarnoi, Wuttipong Kumwilaisak, and Poompat Saengudomlert. On the decodability of random linear network coding in acyclic networks. *IEICE Transactions*, 95-B(10):3120–3129, 2012.
- [113] Uyen Trang Nguyen and Jin Xu. Multicast routing in wireless mesh networks: Minimum cost trees or shortest path trees? *Communications Magazine, IEEE*, 45(11):72–77, November 2007.
- [114] Chang Wook Ahn and R.S. Ramakrishna. A genetic algorithm for shortest path routing problem and the sizing of populations. *Evolutionary Computation, IEEE Transactions on*, 6(6):566–579, Dec 2002.
- [115] Gerhard Hasslinger and Oliver Hohlfeld. The gilbert-elliott model for packet loss in real time services on the internet. In *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference -*, pages 1–15, March 2008.
- [116] C. H. Ke. myevalsvc: an integrated simulation framework for evaluation of h.264/svc transmission. *KSII Transactions on Internet and Information Systems*, 6(1):378–393, January 2012.
- [117] Video Test Media. <https://media.xiph.org/video/derf/>. [Online; accessed 14-May-2015].
- [118] Stefano Traverso, Mohamed Ahmed, Michele Garetto, Paolo Giaccone, Emilio Leonardi, and Saverio Niccolini. Temporal locality in today’s content caching: Why it matters and how to model it. *SIGCOMM Comput. Commun. Rev.*, 43(5):5–12, November 2013.
- [119] Jiangchuan Liu, Xiaowen Chu, and Jianliang Xu. Proxy cache management for fine-grained scalable video streaming. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1490–1500 vol.3, March 2004.
- [120] Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Loudec. Efficient http-based streaming using scalable video coding. *Signal Processing: Image Communication*, 27(4):329 – 342, 2012.
- [121] Guibin Tian and Yong Liu. Towards agile and smooth video adaptation in dynamic http streaming. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, CoNEXT ’12*, pages 109–120, New York, NY, USA, 2012. ACM.

Appendix A

Publication List

Articles in Journals

1. S. Tarnoi, W. Kumwilaisak, and Y. Ji, "Optimal cooperative routing protocol for efficient in-network cache management in content-centric networks," *IEICE Trans. Comm.*, vol.E97-B, no.12, pp.2627-2640, Dec. 2014.
2. S. Tarnoi, W. Kumwilaisak, P. Saengudomlert, Y. Ji, and C.-C. J. Kuo, "QoS-aware routing for heterogeneous layered unicast transmissions in wireless mesh networks with cooperative network coding," *EURASIP Journal on Wireless Communications and Networking*, 2014, 2014:81, doi:10.1186/1687-1499-2014-81.
3. S. Tarnoi, W. Kumwilaisak, and Y. Ji, and C.-C. J. Kuo, "Reliable layered multicast with source diversity and network coding in wireless mesh networks," *Journal of Visual Communication and Image Representation*, vol.24, no.5, pp.602-614, Jul. 2013.

Conference Proceedings

1. S. Tarnoi, V. Suppakitpaisarn, W. Kumwilaisak, and Y. Ji, "Performance analysis of probabilistic caching scheme using Markov chains", *IEEE Conference on Local Computer Networks (LCN)*, pp.255-263, Florida, USA, Oct. 2015.
2. S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Performance of probabilistic caching and cache replacement policies for Content-Centric Networks," *IEEE Conference on Local Computer Networks (LCN)*, pp.414-417, Edmonton, Canada, Sept. 2014.
3. S. Tarnoi, W. Kumwilaisak, and Y. Ji, "Optimal cooperative routing protocol based on prefix popularity for Content Centric Networking," *IEEE Conference on Local Computer Networks (LCN)*, pp.99-106, Edmonton, Canada, Sept. 2014.
4. S. Tarnoi, K. Suksomboon, and Y. Ji, "Cooperative routing protocol for content-centric networking," *IEEE Conference on Local Computer Networks (LCN)*, pp.699-702, Sydney, Australia, Oct. 2013.
5. K. Suksoomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, N. Motonori, M. Aoki, S. Urishidani, and S. Yamada, "PopCache: Cache more or less based on content popularity for information-centric networks," *IEEE Conference on Local Computer Networks (LCN)*, pp.236-243, Sydney, Australia, Oct. 2013.

6. S. Tarnoi, W. Kumwilaisak, and Y. Ji, “Reliable scalable video multi-cast with source diversity and inter-source network decoding in lossy networks,” *APSIPA Annual Summit and Conference 2012 (APSIPA ASC 2012)*, Hollywood, USA, Dec. 2012.

Others

1. S. Tarnoi, Vorapong Suppakitpaisarn, and Y. Ji, “Adaptive Probabilistic Caching Scheme for Information-Centric Networking,” to appear in *IEICE Tech. Rep.*, Set. 2015.
2. S. Tarnoi, Vorapong Suppakitpaisarn, and Y. Ji, “Analysis Model of Probabilistic Caching Scheme for Content-Centric Networking,” *IEICE Tech. Rep.*, vol. 114, no. 404, CQ2014-90, pp. 25-30, Jan. 2015.
3. S. Tarnoi, K. Suksomboon, and Y. Ji, “Optimizing video streaming services with cooperative routing and content-centric networks,” *IEICE Tech. Rep.*, vol. 113, no. 244, NS2013-114, pp.129-134, Oct. 2013.
4. K. Suksoomboon, S. Tarnoi, and Y. Ji, “Achieving fairness and low latency with content-centric networked games,” *IEICE Society Conference*, BS-7-47, Fukuoka, Sept. 2013.
5. S. Tarnoi, K. Suksoomboon, and Y. Ji, “Filtering effects in content-centric networks,” *IEICE Society Conference*, BS-7-1, Fukuoka, Sept. 2013.
6. S. Tarnoi, K. Suksomboon, and Y. Ji, “Caching aware routing for content-centric networking,” *IEICE General Conference*, BS-1-56, Gifu, Japan, Mar. 2013.