OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Virtual Content-Centric Networking for Realizing Efficient and Secure Content Retrieval and Distribution
Author(s)	塚本, 圭一郎
Citation	大阪大学, 2014, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/34570
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Virtual Content-Centric Networking for Realizing Efficient and Secure Content Retrieval and Distribution

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2014

Keiichiro TSUKAMOTO

List of Publications

Journal Papers

1. Keiichiro Tsukamoto, Masato Ohtani, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato and Junichi Murayama, "Virtual Content-Centric Networking," to appear in Journal of Networks (JNW), 2014.

2. Keiichiro Tsukamoto, Kaito Ohsugi, Hiroyuki Ohsaki, Toru Hasegawa and Masayuki Murata, "Cache Performance Analysis of Virtualized Router on Virtual Content Centric Networks," International Journal of Next Generation Networks (IJNGN), Vol.5, No.4, December 2013.

3. Keiichiro Tsukamoto, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato and Junichi Murayama, "Inferring Relevant Blocks on Hyperlinked Web Page based on Block-to-Block Similarity," to appear in International Journal of Knowledge and Web Intelligence (IJKWI), 2014.

Refereed Conference Papers

1. Masato Ohtani, Keiichiro Tsukamoto, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato, Junichi Murayama, and Makoto Imase, "VCCN: Virtual Content-Centric Networking for Realizing Group-based Communication," in *Proceedings of 12th IEEE International Conference on Communications (ICC 2013)*, pp. 2069-2073, June 2013.

2. Ryoichi Ishiyama, Keiichiro Tsukamoto, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato, Junichi Murayama and Makoto Imase,"On the Effectiveness of

Diffusive Content Caching in Content-Centric Networking," in *Proceedings of the 9th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT 2012)*, November 2012.

Non-Refereed Technical Papers

1. Keiichiro Tsukamoto, Kaito Ohsugi, Hiroyuki Ohsaki, Toru Hasegawa and Masayuki Murata, "On the Cache Performance of Virtualized CCN router on Virtual Content Centric Networks," IEICE technical report, CQ2013-60, November 2013 (in Japanese).

2. Ryoichi Ishiyama, Keiichiro Tsukamoto and Hiroyuki Ohsaki, "Selective Cache Information Diffusion in Content-Centric Networking," IEICE technical report, IA2013-34, October 2013.

3. Kaito Ohsugi, Keiichiro Tsukamoto and Hiroyuki Ohsaki, "Study on the Effect of CCN Router Virtualization on Content Delivery Time," IEICE Society Conference 2012, B-7-4, August 2012 (in Japanese).

4. Ryoichi Ishiyama, Keiichiro Tsukamoto, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato, Junichi Murayama and Makoto Imase, "A Proposal of Diffusive Content Caching in Content-Centric Networking," IEICE General Conference 2012, B-7-14, March 2012 (in Japanese).

5. Masato Ohtani, Keiichiro Tsukamoto, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato, Junichi Murayama, and Makoto Imase, "On Router Virtualization for Realizing Group-Based Communication in Content-Centric Networking," IEICE technical report, IN2011-59, July 2011 (in Japanese).

6. Masato Ohtani, Keiichiro Tsukamoto, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato, Junichi Murayama, and Makoto Imase, "A Router Virtualization for Realizing Group-Based Communication in Content Centric Network," IEICE General Conference 2011, B-7-82, February 2011 (in Japanese).

7. Keiichiro Tsukamoto, Sho Tsugawa, Yuki Koizumi, Hiroyuki Ohsaki, Kunio Hato, Junichi Murayama and Makoto Imase, "On Estimating Referring Block in Hyperlinked Web Page based on Block-to-Block Similarity," IEICE technical report, IN2010-118, January 2011 (in Japanese).

8. Keiichiro Tsukamoto, Sho Tsugawa, Hiroyuki Ohsaki, Makoto Imase, Takeshi Kuwahara, and Junichi Murayama, "On Inferring Referred Block Based on Block-to-Block Similarity of Hyperlinked Web Pages," IEICE General Conference 2009, D-13-4, March 2009 (in Japanese).

Preface

Content-centric networking (CCN) has recently emerged as a network architecture treating content rather than hosts as a primitive since the majority of Internet usage has been changed from utilizing channels between end hosts to retrieving and distributing content. CCN is designed to be layered over any previous Internet architecture, including Internet Protocol (IP) in order to resolve issues arising from the incompatibility between the Internet usage and the Internet architecture. Hence, CCN needs to be established as a network architecture that is general, practical and versatile in the future Internet as well as IP.

One of key factors to make CCN general, practical and versatile as the future Internet architecture is confidentiality. CCN is designed to be open as well as IP. However, a completely open content-centric network is not sufficient for real-world networking. For instances, since the Internet needed to realize private communication within a group based on IP as the Internet became large, it is expected that the same demand in a content-centric network will arise in the future. In addition, when the popularization of private communication within a group (e.g., Yammer, LINE, Google+) in the current Internet is taken into account, it is necessary to realize *group-based communication* in the future Internet. Specifically, group-based communication in CCN allows consumers to retrieve content only from authorized distributors and allows distributors to distribute content only to authorized consumers, maintaining advantages of CCN in terms of efficiency of content retrieval and distribution against both a user and a network.

Previous approaches utilize secure functions either at lower or upper layers without modifying the CCN layer: construction of secure channels between users

by encrypting packets in a layer upper than CCN, and construction of VPNs in a layer lower than CCN. The former approach uses public key mechanisms so that a consumer can control who distributes content and a distributor can control who retrieves content. The latter approach constructs a closed VPN within a group of users by logically slicing a backbone network and enables the users belonging to the group to transmit packets to the other in the VPN.

On the contrary, these approaches have drawbacks in terms of efficiency. The most serious drawback is that the approaches restrict in-network caching except for at end hosts or edge routers in CCN. This drawback wipes out in-network caching, one of the best benefits of CCN. In addition, these approaches disable either network-level multihoming or multicast in CCN. CCN is designed to be able to take maximum advantage of multiple simultaneous connectivity (e.g., ethernet, 3G, bluetooth and 802.11). However, since the VPN approach in a layer lower than CCN cannot make an arbitrary choice between lower protocols, it disables multihoming. Besides, CCN routers perform the aggregation of requests for the same content and they multicast the content to all the requesters. However, since the encryption approach in a layer upper than CCN has to identify who requested the content, it disables multicasting.

For realizing secure content retrieval and distribution without these drawbacks, VPNs should be constructed not on a layer lower than CCN but on a CCN layer. The VPN approach in a CCN layer does not restrict in-network caching and is independent of both locations and lower connectivities by keeping CCN routing mechanism. Moreover, the VPN approach in a CCN layer naturally inherits the advantage of that in a layer lower than CCN.

In this thesis, the author therefore designs and implements virtual-content centric networking (VCCN), which enables construction of virtual content-centric networks (VCCN slices) on a content-centric network, for realizing efficient and secure content retrieval and distribution. VCCN realizes group-based communication while maintaining in-network caching in CCN. The fundamental idea of VCCN is to operate a CCN router as logically independent multiple VCCN router instances by virtualization. Group-based communication is realized by

building VCCN slices, each of which is composed of multiple VCCN router instances. Through a preliminary performance evaluation of the VCCN implementation, the author shows that the introduction of VCCN has a positive or negative impact on CCN performance and that CCN router virtualization in VCCN incurs a little overhead to CCN in terms of the content delivery time.

However, two issues on efficiency of content retrieval and distribution that result from introduction of VCCN have not been addressed. The first issue is to balance the overall network performance and network performance for each VCCN slice in terms of efficiency of content distribution. The second issue is to quickly locate relevant parts in distributed content in terms of efficiency of content retrieval.

For resolving the first issue on efficiency of content distribution in VCCN, the author analytically and quantitatively investigates a trade-off among the network fairness for VCCN slices and overall network performance. The author investigates what resource allocation method provides the best balance between the network fairness for VCCN slices and overall network performance in conceivable three allocation methods (i.e., an exclusive method, a shared method and a hybrid method). Using several numerical examples, the author shows that when content request patterns are heterogeneous, a hybrid resource allocation method will provide the best balance between fairness and overall network performance.

For resolving the second issue on efficiency of content retrieval in VCCN, the author proposes an application-level approach for improving the efficiency Web browsing, which is the representative Internet usage in content retrieval of the current Internet. The approach called *HypErlink Referring Block estimation (HERB)* segments Web pages into blocks and infers the existence and location of all relevant content on hyperlinked Web pages based on a block-to-block similarity. Through experiments simulating ordinary Web browsing, the author shows that HERB can infer blocks relevant to a hyperlink with precision and recall that are as high or higher than those of existing methods on a block-based Web search. Furthermore, the two HERB-enabled implementations, namely, a Web proxy and Web browser, are also designed.

Acknowledgements

My research activities would have not been achieved without many individuals.

First of all, I would like to express my sincere appreciation to my supervisor, Professor Toru Hasegawa, Graduate School of Information Science and Technology, Osaka University, for his instructive guidance, valuable discussion and thoughtful support. For a year, he has always encouraged me in research. Without his ability to grasp the true nature of problems, this thesis will not have been completed.

I would like to express my deep appreciation to Professor Masayuki Murata, Professor Takashi Watanabe, Professor Teruo Higashino and Professor Morito Matsuoka, Graduate School of Information Science and Technology, Osaka University, for their valuable comments and reviewing this thesis. Thanks to their resourceful suggestion, this thesis became clearer and more valuable.

I would like to give my deep gratitude to Dr. Makoto Imase, Vice President of National Institute of Information and Communications Technology, and Professor Hiroyuki Ohsaki, Department of Informatics, School of Science and Technology, Kwansei Gakuin University, for their irreplaceable guidance and continuous support. They grounded me thoroughly in research. I'm proud to be their student.

I also wish to give my deep gratitude to Assistant Professor Yuki Koizumi, Graduate School of Information Science and Technology, Osaka University, for cordial guidance and much encouragement. His apt comments stimulated my daily research activities and helped me to make a breakthrough in research.

I am grateful to Professor Junichi Murayama, Department of Information and Communication Technology, Tokai University, Mr. Takeshi Kuwahara, Mr. Kunio Hato, Dr. Takeshi Yagi, Mr. Tsuyoshi Kondoh, Mr. Bo Hu and Mr. Yuichi Sudo

of NTT Corporation for fruitful discussion. Their supports made my researches practical.

I would like to thank Dr. Kohei Sugiyama of KDDI Corporation and Assistant Professor Sho Tsugawa, Faculty of Engineering, Information and Systems, University of Tsukuba, for their direction during my bachelor and master coursework. Their direction also promoted my daily research activities.

Special thanks are due to Masato Ohtani, Ryoichi Ishiyama and Kaito Ohsugi for their ideas and valuable discussions. They strengthened the base of my researches and reinforced the value of this thesis.

I would like to extend my appreciation to the members of the Information Sharing Platform Laboratory. I am thankful to Ms. Namiko Okada, Ms. Hiroko Hatagami and Ms. Yoshimi Fujita for their kind help. I thank Takamichi Nishijima, Kohei Watabe and all the other laboratory members for their support. I think that daily conversation with them gave me many inspirations.

Besides, I would like to express my thanks to my confidant, Shinpei Sasaki, for moral support. His recklessness suitably breaks my timidity in research and made me reborn.

Finally, I would like to express my heartfelt appreciation to my father, mother and little brother, Yasuyuki, Etsuko and Akihito, for their warm-hearted help and support. With the help of them, I could concentrate on this work and complete it with an easy mind.

Once more, I would like to express my hearty thanks to all of you for engaging with this work. Thank you.

Contents

List of Publications **i**

Preface **v**

Acknowledgements **ix**

1 Introduction **1**

1.1 Background 1

1.2 Proposed Solution 4

1.3 Outline of Thesis 8

2 Virtual Content-Centric Networking **13**

2.1 Introduction 13

2.2 Related Work 15

2.3 Virtual Content-Centric Networking (VCCN) 16

2.3.1 Overview 16

2.3.2 Extension of the Content Identifier 17

2.3.3 CCN Router Virtualization 18

2.3.4 Packet Transport between Virtualized CCN Routers 19

2.3.5 SNS Cooperative User/Group Identification 22

2.4 Implementation and Evaluation 25

2.4.1 VCCN Implementation 25

2.4.2 Performance Evaluation of the VCCN Implementation . . . 27

2.5 Open Issues 30

2.5.1 CCN Router Resource Management 30
2.5.2 VCCN Slice Mapping 32
2.5.3 Reliability 32
2.6 Summary 33

3 Cache Performance Analysis of Virtualized Router on Virtual Content-Centric Networks 35
3.1 Introduction 35
3.2 Related Work 37
3.3 CS Allocation to VCCN Router Instances 38
3.4 Virtualized CCN Router Model 39
3.4.1 Model Description and Notation 39
3.4.2 Determination of the Cache Hit Rate based on a Markov Chain Model 41
3.4.3 Determination of the Hit Rate using an Approximation Method 43
3.5 Numerical Example 45
3.5.1 Validation of the Model 45
3.5.2 Effects of Content Popularity Slopes in VCCN Slices 47
3.5.3 Effects of the Content Request Ratio for each VCCN Slice . 50
3.5.4 Effects of Content Request Patterns in VCCN Slices 51
3.6 Summary 54

4 Inferring Relevant Blocks on Hyperlinked Web Page based on Block-to-Block Similarity 55
4.1 Introduction 55
4.2 Taxonomy of Hyperlinks 59
4.3 HypErlink Referring Block estimation (HERB) 61
4.3.1 Overview 61
4.3.2 Web Page Segmentation 61
4.3.3 Feature Term Extraction from Each Block 63
4.3.4 Block-to-block Similarity Calculation 64
4.4 Experiments 64

4.4.1 Experimental Methods 64
4.4.2 Results: Evaluation using Relevance Scores 66
4.4.3 Results: Evaluation using Importance Scores 70
4.5 Design and Implementation 71
4.5.1 Design of HERB-enabled Web proxy 72
4.5.2 Design of HERB-enabled Web browser 73
4.5.3 Implementation of HERB-enabled Web proxy 75
4.6 Summary 76

5 Conclusion **79**

List of Figures

1.1 Comparison of approaches to realize secure content retrieval and distribution on a content-centric network. 11

2.1 Example of VCCN slices built on a content-centric network; two logically independent VCCN slices X and Y are built on the network of seven CCN routers, A through G. 16

2.2 Example of an extended content identifier; the first two components are used as the VCCN declaration and the VCCN identifier. 18

2.3 A virtualized CCN router; it is composed of a demultiplexer, VCCN router instances, and multiplexers. 18

2.4 Packet transport in a lower layer; if a lower layer protocol supports communication between an arbitrary pair of nodes (e.g., IP, UDP, TCP, and broadcast communication), any pair of CCN routers can communicate using the lower layer protocol. 20

2.5 Flooding in the CCN layer; in CCN, flooding can be realized simply by duplicating Interest packets and sending them through all faces of every CCN router. 21

2.6 Tunneling in CCN layer; as in the source routing option of IP, the CCN router forwards the packet to the face listed at the head of the source route. 22

2.7 Sequence diagram for requesting content in VCCN. 23

2.8 Sequence diagram when registering a content in VCCN. 24

2.9 Example of creating a group; four members are registered with the data-centric networking group on Facebook and every registered member can take part in group-based communication. 26
2.10 Processes for discarding three types of packet: (1) an Interest packet for X's content that a user who does not belong to any group requests through VCCN; (2) an Interest packet for X's content that a user belonging to Y requests through VCCN; and (3) an Interest packet for X's content that a user belonging to X requests through CCN. 27
2.11 Network topology used in the CNNx/VCCN comparison; four CCN routers are connected and two VCCN slices, X and Y, are created. 28
2.12 CDFs for content delivery delay when content is requested through a content-centric network and the VCCN slices. 29
2.13 Average content delivery delays against content request rate. . . . 30
2.14 Average content delivery delays against the number of VCCN slices in a content-centric network. 31

3.1 Examples of methods for the allocation of CS to VCCN routers (an exclusive method, a shared method and a hybrid method). 40
3.2 The model considered in this chapter. 40
3.3 Markov chain model. The state in which content c on VCCN slice S^n is placed in the kth segment of the CS is denoted by $s_{c,k}(0 \leq k \leq C(m))$. 42
3.4 Network used for the evaluation. 46
3.5 The cache hit rate against the size of CS for the exclusive method, the shared method and a hybrid method (Markov chain based analysis). 47
3.6 The cache hit rate against the size of CS for the exclusive method, the shared method and a hybrid method (approximate analysis). . 48
3.7 Average content delivery time and fairness index against the difference of Zipf exponent α between VCCN slices, for each allocation method. 49

3.8 Average content delivery time and fairness index against the ratio r of content requests for VCCN slice S^1 compared to all requests for each allocation method. 51

3.9 Average content delivery time and fairness index against content request ratio r of VCCN slice S^1 for each allocation method when there is a difference of Zipf exponent α between VCCN slices. . . . 52

3.10 Average content delivery time and fairness index against the size of CS allocated to R^n by hybrid(S^n). 53

4.1 Taxonomy of hyperlinks. 59

4.2 Examples of several types of hyperlinks. 60

4.3 Overview of HERB. 62

4.4 Histograms showing the distribution of block-to-block similarity scores assigned by HERB for sets of blocks with the same relevance score. 67

4.5 Box plots showing the relation between relevance scores and block-to-block similarity scores assigned by HERB. The dotted line is the regression line. 68

4.6 Precision and recall for a given T_h. Precision and recall measure how accurately and comprehensively relevant blocks can be extracted from the destination Web page, respectively. Lines in the plots correspond to cases where blocks with relevance scores greater than or equal to r are considered relevant. 69

4.7 Histogram showing the distribution of block-to-block similarity scores assigned by HERB for I and $\overline{I}$. 71

4.8 Precision, recall of the most important information for a given T_h. 72

4.9 HERB-enabled Web proxy. 73

4.10 HERB-enabled Web browser. 74

4.11 An example of using the HERB-enabled Web proxy. When a user selects a hyperlink on the source Web page, a destination Web page is served in which layout and style information is embedded for highlighting relevant blocks according to the similarity measures. Moreover, the text in the three blocks with the highest similarity to the block containing the selected hyperlink are displayed with intra-page links in order to facilitate in-page navigation. 76
4.12 Processing delay distribution of HERB-enabled Web Proxy. 77

List of Tables

1.1 Comparison of approaches for realizing group-based communication in CCN with respect to available mechanisms improving the efficiency of content retrieval and distribution. 4

Chapter 1

Introduction

1.1 Background

Content-centric networking (CCN) has recently emerged as a network architecture treating content rather than hosts as a primitive since the majority of Internet usage has been changed from utilizing channels between end hosts to retrieving and distributing content [1]. CCN is designed to be layered over any previous Internet architecture, including Internet Protocol (IP) [2], which is the most general network architecture in the current Internet, in order to resolve issues arising from the incompatibility between the Internet usage and the Internet architecture. CCN provides users with efficient content retrieval and distribution by content name based routing and in-network caching. CCN will be incrementally deployed as the future Internet architecture because of this advantage suitable for the current and future Internet usage. Hence, CCN needs to be established as a network architecture that is general, practical and versatile in the future Internet as well as IP.

One of key factors to make CCN general, practical and versatile as the future Internet architecture is confidentiality. CCN is designed to be open because ease of content reuse is one of the greatest advantages of CCN. However, a completely open content-centric network is not sufficient for real-world networking. For instance, IP was also designed to be open and this global openness of IP

contributed the rapid spread of IP. However, as the Internet became large, the Internet needed to realize private communication within a group (e.g., communication within fellow members belonging to a corporation, a organization, or a community) based on IP. Although this demand was satisfied by a framework for L3 Provider-Provisioned Virtual Private Networks (PPVPNs), especially L3 PE-based VPNs [3], it is expected that the same demand in a content-centric network will arise in the future. In addition, when the popularization of private communication within a group (e.g., Yammer [4], LINE [5], Google+ [6]) in the current Internet is taken into account, it is necessary to realize *group-based communication* in the future Internet (i.e., a content-centric network). Specifically, group-based communication in CCN allows consumers to retrieve content only from authorized distributors and allows distributors to distribute content only to authorized consumers, maintaining advantages of CCN in terms of efficiency of content retrieval and distribution against both a user and a network.

Previous approaches utilize secure functions either at lower or upper layers without modifying the CCN layer: construction of secure channels between users by encrypting packets in a layer upper than CCN, and construction of VPNs [3] in a layer lower than CCN. The former approach uses public key mechanisms so that a consumer can control who distributes content and a distributor can control who retrieves content. The former approach basically works as follows: (1) users register their own certificates to a trusted mean (e.g., a certificate authority (CA)), (2) users exchange the certificates through the trusted mean when they want to communicate with each other, (3) each user obtains a public key of the other from the exchanged certificate, and (4) each user sends packets encrypted by the public key or packets encrypted by exchanged common key using the public key to the other [7–10]. Since encrypted content can be decrypted only by their own secret key, secure channels between users can be constructed through the above process. One advantage of the former approach is that private communication with two users can be realized independent of network architecture in a lower layer without complicated network configuration [9].

In contrast, the latter approach constructs a closed VPN within a group of

users by logically slicing a backbone network and enables the users belonging to the group to transmit packets to the other in the VPN. This famous approach uses a framework for L3 Provider-Provisioned VPNs (PPVPNs), especially L3 PE-based VPNs [3]. In this framework, edge routers on a network provided by a service provider include a VPN forwarding instance (VFI) per VPN and each VFI has the router information base and forwarding information base for a VPN [3]. Packets belonging to a VPN are transmitted between VFIs in the same VPN through a VPN tunnel, which is a logical link between two edge routers realized by encapsulating packets according to the backbone network architecture [3]. Only specific customer edges communicate with the edge routers. VPNs can be constructed on IP network or Multi-Protocol Label Switching (MPLS) network lower than CCN by this frame work. Closed content-centric networks can be constructed by deploying CCN on IP VPN or MPLS VPN [11]. One advantage of the latter approach is that private communication within a group of users can be realized without complicated key management even if members of a group increase.

On the contrary, these approaches have drawbacks in terms of efficiency (Table 1.1). The most serious drawback is that the approaches restrict in-network caching except for at end hosts or edge routers in CCN. As shown in Fig. 1.1, the encryption approach in a layer upper than CCN makes channels between users confidential and the VPN approach in a layer lower than CCN makes channels between edge routers confidential. While every CCN router in a content-centric network has its buffer memory called ContentStore (CS) for later reuse in CCN, only CSs of edge CCN routers are utilized in these approaches. This drawback wipes out in-network caching, one of the best benefits of CCN. In addition, these approaches disable either network-level multihoming or multicast in CCN. CCN is designed to be able to take maximum advantage of multiple simultaneous connectivity (e.g., ethernet, 3G, bluetooth and 802.11) [1]. However, since the VPN approach in a layer lower than CCN cannot make an arbitrary choice between lower protocols, it disables multihoming. Besides, CCN routers perform the aggregation of requests for the same content and they multicast the content to all the requesters. However, since the encryption approach in a layer upper than CCN

Table 1.1: Comparison of approaches for realizing group-based communication in CCN with respect to available mechanisms improving the efficiency of content retrieval and distribution.

Approach	In-network caching	Anycast	Multihoming	Multicast
Encryption approach in a layer upper than CCN	×	◯	◯	×
VPN approach in a layer lower than CCN	×	◯	×	◯
Virtual Content Centric Networking	◯	◯	◯	◯

has to identify who requested the content, it disables multicasting.

For realizing secure content retrieval and distribution without these drawbacks, VPNs should be constructed not on a layer lower than CCN but on a CCN layer. The VPN approach in a CCN layer does not restrict in-network caching and is independent of both locations and lower connectivities by keeping CCN routing mechanism (see Fig. 1.1). Moreover, the VPN approach in a CCN layer naturally inherits the advantage of that in a layer lower than CCN.

1.2 Proposed Solution

The author thinks that confidentiality in CCN should be content-oriented due to the efficiency of content retrieval and distribution in CCN. The above three approaches realize user-oriented confidentiality, host-oriented confidentiality and content-oriented confidentiality, respectively. The encryption approach in a layer upper than CCN realizes user-oriented confidentiality (see Fig. 1.1 (1)). User-oriented confidentiality is defined as confidentiality that is achieved by identifying who retrieves/distributes content. In user-oriented confidentiality, each user authenticates communication partners using his/her certificate of a trust mean. By specifying trusted communication partners, each user prevents outsiders, which do not show their own certificates, from obtaining confidential content or dis-

tributing malicious content. On the contrary, content retrieval and distribution in user-oriented confidentiality is limited to communication between specific two users since a user gives his/her confidence to each user. The VPN approach in a layer lower than CCN realizes host-oriented confidentiality (see Fig. 1.1 (2)). Host-oriented confidentiality is defined as confidentiality that is achieved by identifying which host sends packets. In host-oriented confidentiality, a network provider authenticates hosts, which request/respond packets, in provider edges. By making users retrieve/distribute content through specific customer edges, a network provider prevents outsiders, who cannot use the customer edges, from obtaining confidential content or distributing malicious content. On the contrary, content retrieval and distribution in host-oriented confidentiality is limited to communication between specific two hosts. Finally, the VPN approach in a CCN layer realizes content-oriented confidentiality (see Fig. 1.1 (3)). Content-oriented confidentiality is defined as confidentiality that is achieved by identifying what is retrieved/distributed. In content-oriented confidentiality, a network provider checks what users retrieve/distribute in provider edges. By making users show the rights to access the content, a network provider prevents outsiders, who do not have the rights, from obtaining confidential content or distributing malicious content. Since the rights are associated with not users but content, content retrieval and distribution in content-oriented confidentiality can aggregate connections with multiple users who have the rights. This characteristic makes it possible to maintain mechanisms (e.g., in-network caching and multicast) to improve the efficiency of content retrieval and distribution in CCN.

In this thesis, the author therefore proposes virtual-content centric networking (VCCN), which enables construction of virtual content-centric networks (VCCN slices) on a content-centric network, for realizing efficient and secure content retrieval and distribution. VCCN realizes content-oriented confidentiality (see Fig. 1.1 (3)). In VCCN, edge routers in VCCN slices identify users and prevents unauthorized users from retrieving/distributing any content through the VCCN slice. In a VCCN slice, packet routing is based on content identifiers rather than neither user identifiers nor host identifiers. Namely, even if group-

based communication in CCN is realized, content retrieval and distribution in an inner network is performed in content-oriented form. Thus, users, which communicate through VCCN slices, cannot check who retrieves/distributes content, unlike previous approaches that realize either user-oriented or host-oriented confidentiality. The users are only guaranteed that content consumers/distributors are someone belonging to the same group by edge routers. However, VCCN intactly utilizes mechanisms improving the efficiency of content retrieval and distribution by maintaining content name based routing. Hence, VCCN realizes group-based communication while maintaining the efficiency of content retrieval and distribution in CCN. Moreover, VCCN provides dynamics in creating groups and changing users in the group, which is one of the important requirements to private communication, as users are identified personally rather than by the host on which they reside by SNS cooperative user/group identification. This has the advantage of preserving the location-independence of CCN since users can retrieve and distribute content through VCCN slices independently of hosts.

However, two issues on efficiency that result from introduction of VCCN in terms of content retrieval and distribution have not been addressed. The first issue is to balance the overall network performance and network performance for each VCCN slice in terms of efficiency of content distribution. Although the efficiency of content distribution (i.e., the efficiency of in-network caching in CCN) depends strongly on how a CCN router allocates its CS to VCCN router instances of which a VCCN slice is composed, a trade-off among the network fairness for VCCN slices and overall network performance is left out of consideration. The second issue is to quickly locate relevant parts in the content in terms of efficiency of content retrieval. Although the efficiency of content retrieval depends not only on how distributors quickly send contents to consumers but also on how consumers quickly locate relevant parts in the contents, the latter case is left out of consideration.

For resolving the first issue on efficiency of content distribution in VCCN, the author analytically and quantitatively investigates a trade-off among the network fairness for VCCN slices and overall network performance, which results from introduction of VCCN. When multiple VCCN slices are constructed, the perfor-

mance of each VCCN slice and that of the entire network are strongly affected by the CCN routers' CS allocation to VCCN router instances in VCCN slices. Several previous studies have shown clearly that, in CCN, the effectiveness of content caching depends strongly on the content request pattern experienced by the CS of a CCN router [12–14]. Hence, the performance of each VCCN slice and that of the entire network depend strongly on how a CCN router allocates its CS to VCCN router instances on VCCN slices that have different content request patterns. Three types of methods of allocating CS resources to VCCN router instances are conceivable : an exclusive method, a shared method and a hybrid method. In the exclusive method, each VCCN router instance within a CCN router monopolizes a given part of its CS. In the shared method, all VCCN router instances within a CCN router use its entire CS jointly. In the hybrid method, several VCCN router instances within a CCN router are assigned their own parts of its CS and other instances jointly use the remaining CS. The author investigates what resource allocation method provides the best balance between the network fairness for VCCN slices and overall network performance.

For resolving the second issue on efficiency of content retrieval in VCCN, the author proposes an generic approach to improve the efficiency of content retrieval in the form of Web browsing, which is the representative Internet usage in content retrieval of the current Internet. Lazonder *et al.* [15] showed that even users experienced at Web browsing spend almost the same amount of time as novices to locate sought-after information on specific Web sites (i.e., to browse related Web pages one-by-one simply by following hyperlinks). Conversely, Lazonder *et al.* also showed that the experienced users take on average one-third of the time that novices do to locate Web sites containing sought-after information using search engines. In Web browsing, both experienced and novice users must look through all the content of a destination Web page to determine whether it contains the sought-after information, resulting in the comparable Web browsing performance between them [15]. Therefore, it is more important to quickly locate all relevant content on destination Web pages for improving the efficiency of Web browsing. The author proposes the approach to quickly locate all relevant content on destination Web

pages after users click hyperlinks.

The main contributions of this thesis are the following. First, the author presents a general and practical network architecture (VCCN) for realizing group-based communication on a content-centric network. In VCCN, who retrieves content and who distributes the content are controled and content-centric private communication within a group is realized. Moreover, VCCN adapts to the change in a group and remains the advantages of CCN (i.e., availability, location-independence) by adopting the VPN approach in a CCN layer. Second, through several numerical examples, the author shows that when content request patterns are heterogeneous, a hybrid resource allocation method will provide the best balance between network fairness for VCCN slices and overall network performance. Although previous studies showed that an exclusive method is preferable for improving network fairness for applications or services [16, 17], the author shows that an exclusive method cannot keep up with the change of CS size required for each VCCN router instance and the fairness is degraded as the difference between the content popularity slopes increases. The author shows that a hybrid method, which assigns a part of the CS to a VCCN slice with low content popularity slope and low content request ratio, is preferable for improving network fairness for VCCN slices and is secondly preferable for maximizing the overall network performance. Third, the author presents an application-level approach (HERB) to identify all relevant fine-grained relevant blocks in destination Web pages, and HERB-enabled Web navigation system for improving the efficiency Web browsing. HERB enables users to quickly locate all fine-grained content in destination Web pages irrespective of the capability of Web browsing devices. Hence, using HERB improves the efficiency of content retrieval in VCCN from not only distributor but also user standpoints.

1.3 Outline of Thesis

The structure of this thesis is as follows.

In Chapter 2, the author designs and implements *Virtual Content-Centric Networking (VCCN)* for realizing secure content retrieval and distribution. VCCN

enables group-based communication on a content-centric network. The fundamental idea of VCCN is to operate a CCN router as logically independent multiple VCCN router instances by virtualization. Group-based communication is realized by building VCCN slices, each of which is composed of multiple VCCN router instances. In VCCN, a user communicates through an edge router that identifies the user and the relevant group memberships based on SNS cooperative user/group identification. Hence, an outsider cannot request any content of a group through VCCN. Through a preliminary performance evaluation of the VCCN implementation, the author shows that the introduction of VCCN has a positive or negative impact on CCN performance and that CCN router virtualization in VCCN incurs a little overhead to CCN in terms of the content delivery time. Precisely speaking, the overhead is defined as both allocation packets to VCCN slices and routing of packets on VCCN slices.

In Chapter 3, the author analyzes the performance of VCCN under heterogeneous content request patterns in VCCN slices. Specifically, the author analytically investigates the effect of CS allocation methods (i.e., an exclusive method, a shared method and a hybrid method) and content request patterns in VCCN slices in terms of the network fairness for VCCN slices and overall network performance. Previous studies of the effects of content caching on content-centric networks have focused only on the exclusive and shared methods [16–19]. However, when content request patterns are heterogeneous, these two methods can barely maintain a balance between network fairness for VCCN slices and overall network performance. Hence, the author conjectured that a hybrid method, which has the characteristics of both the exclusive and shared approaches, might be a useful CS allocation method on a content-centric network in which there are multiple content request patterns in VCCN slices. Using several numerical examples, the author shows that when content request patterns are heterogeneous, a hybrid resource allocation method will provide the best balance between fairness and overall network performance.

In Chapter 4, we presents an application-level approach for improving the efficiency Web browsing in the current Internet. The approach called *HypErlink*

Referring Block estimation (HERB) segments Web pages into blocks and infers the existence and location of all relevant content on hyperlinked Web pages based on a block-to-block similarity. Through experiments simulating ordinary Web browsing, the effectiveness of HERB is quantitatively investigated. The experiment results show that HERB can infer blocks relevant to a hyperlink with approximately 65% precision and 70% recall. These precision and recall are as high or higher than those of existing methods on a block-based Web search. Hence, the experiment results indicate that inference of relevant blocks by HERB will assist a user to search through relevant content of destination Web pages. Furthermore, the two HERB-enabled implementations, namely, a Web proxy and Web browser, are also designed.

Finally, Chapter 5 concludes this thesis and discusses future works.

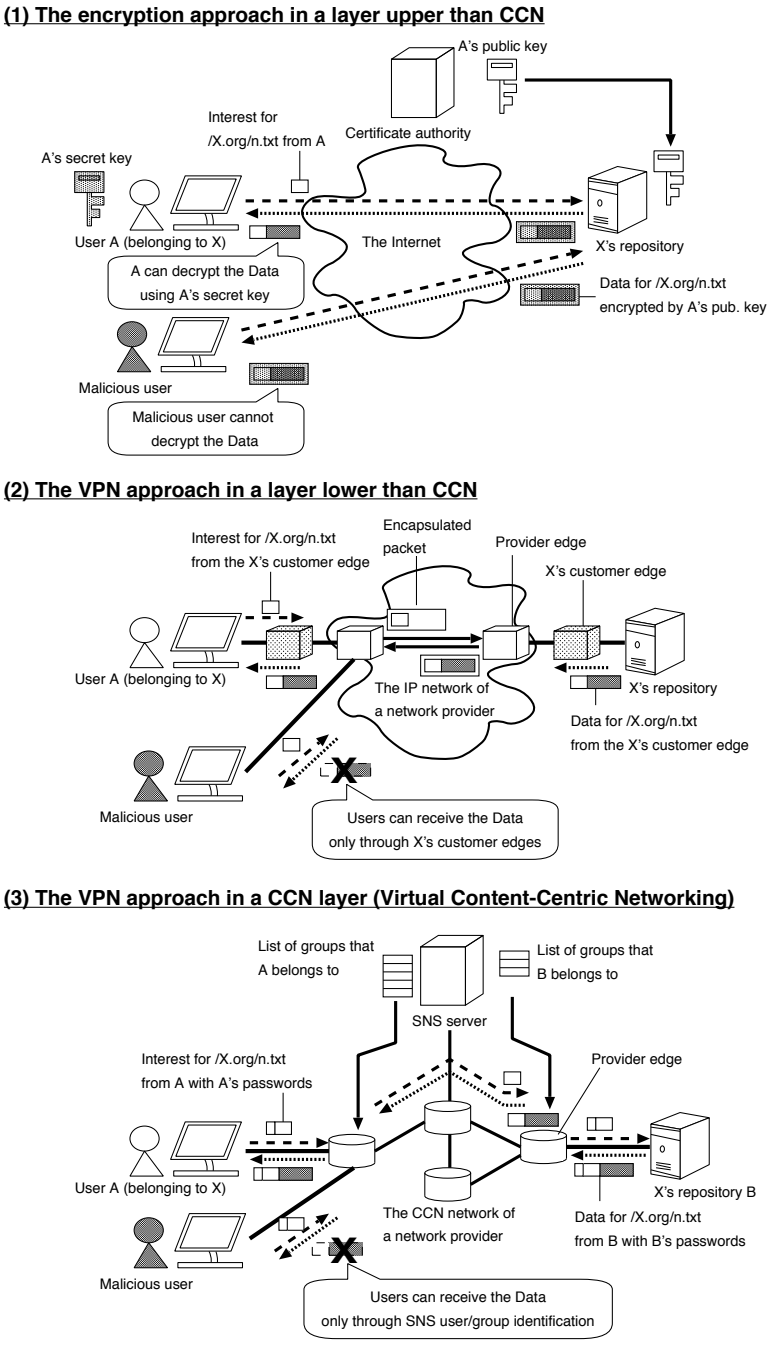


Figure 1.1: Comparison of approaches to realize secure content retrieval and distribution on a content-centric network.

Chapter 2

Virtual Content-Centric Networking

2.1 Introduction

Data-centric networking, which takes named data rather than hosts as being connected via the network as its central abstraction, has recently been gaining attention [20–23].

A representative design for data-centric networking is Content-Centric Networking (CCN) [1, 24], in which routers forward packets based on unique content identifiers. CCN adopts a *request-and-response* communication model. A request packet from a user, called an *Interest packet*, is routed between CCN routers according to the longest prefix matching the requesting content identifier. If the Interest packet is successfully delivered to the source, the content packet, called a *Data packet*, is sent back to the user by traversing the path of the Interest packet in reverse. CCN routers cache forwarded content in a buffer memory called the *contents store (CS)* for later reuse. When a CCN router receives an Interest packet for cached content, it returns the cached content as a Data packet so that the amount of traffic transferred over the network can be reduced.

Because ease of data reuse is one of the greatest advantages of data-centric networking [1], CCN is basically designed to be open: any user requesting some

content by specifying its identifier will receive it. CCN assumes that the primary means of controlling access to content is encryption in a layer higher than CCN [1, 25].

However, for real-world networking, a completely open data-centric network is not sufficient. For example, it is expected that security threats that abuse the global openness, such as spamming and phishing, will become more frequent on data-centric networks. However, advanced security measures to solve these problems may reduce the convenience of networks in many cases.

In this chaper, the author focuses on private communication within a closed group of users where only specific users can access content. In such *group-based communication* the above security issues are minimized.

The author proposes Virtual Content-Centric Networking (VCCN), which realizes group-based communication on a content-centric network. In VCCN every user can freely and dynamically create and change groups, as users are identified personally rather than by the host on which they reside. This has the advantage of preserving the location-independence of CCN [1].

The fundamental idea of VCCN is to operate a CCN router as logically independent multiple VCCN router instances by virtualization. Group-based communication is realized by building virtual content-centric networks (VCCN slices), each of which is composed of multiple VCCN router instances. In VCCN, a user communicates through an edge router that identifies the user and the relevant group memberships.

The main contributions of this chapter are the following. First, the author presents a general and practical network architecture (VCCN) for constructing virtual private networks on a content-centric network by CCN router virtualization. Second, the author shows that CCN router virtualization in VCCN incurs a little overhead to CCN in terms of the content delivery time, through a preliminary performance evaluation of our VCCN implementation. Precisely speaking, the overhead is defined as both allocation packets to VCCN slices and routing of packets on VCCN slices.

The organization of this chapter is as follows. Section 2.2 contains a summary

of related work. In Section 2.3 the author gives an overview of VCCN and its four building blocks. In Section 2.4, the author describes our VCCN implementation and the results of a preliminary performance evaluation. In Section 2.5, the author discusses open research issues in VCCN network construction. Finally, in Section 2.6 the author gives our conclusions.

2.2 Related Work

One attempt to realize group-based communication on data-centric networks is the Virtual Private Community (VPC) service [7,8]. VPC is a CCN-based service architecture designed to share content among users of a community. In VPC, a virtual private community is built hierarchically from three types of members: creator, owners, and members. If a user is invited by the creator or owner of a virtual private community, the user can join the community and share content with its members. VPC realizes group-based communication in a content-centric network, but controlling access to content among the users is done simply by content encryption in a layer higher than CCN.

The VCCN design proposed in this chapter was inspired by the Virtual Data-Oriented Network Architecture (VDONA) [26]. VDONA extends flat content identifiers in DONA by embedding group identifiers in the content identifiers, and changes the name resolution process in DONA into the two-stage (i.e., group and content) name resolution process. In VDONA, none but authorized users can resolve the group names. Thus, VDONA realizes group-based communication on data-centric-networks by closing the name resolution process. VCCN is similar to VDONA in the sense that a name space is split into multiple subspaces for enabling group-based communication. VCCN is, however, different from VDONA in terms of how a router is virtualized and how packet transport between virtualized routers is accomplished. In VCCN, content name based routing is performed, unlike VDONA. VCCN enables intact utilization of mechanisms improving the efficiency of content retrieval and distribution in CCN (e.g., multicast) by closing not only the name resolution process but also the packet transport process.

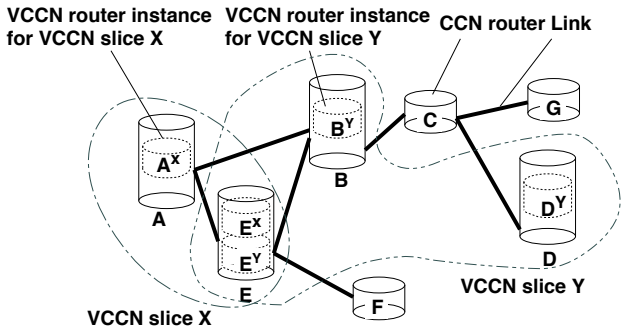


Figure 2.1: Example of VCCN slices built on a content-centric network; two logically independent VCCN slices X and Y are built on the network of seven CCN routers, A through G.

2.3 Virtual Content-Centric Networking (VCCN)

2.3.1 Overview

In VCCN, several VCCN router instances are created on a CCN router and a network is built by logically connecting VCCN router instances. An example of such a VCCN slice is shown in Fig. 2.1. Users are allowed to send Interest packets to VCCN slices that they belong to, and they can receive Data packets only from those networks. An Interest packet is routed within the VCCN slice by the logically connected VCCN router instances. If the Interest packet is successfully delivered, the corresponding Data packet is sent back to the user within the VCCN slice by traversing the path of the Interest packet in reverse.

The four building blocks of VCCN are as follows:

- **Extension of the content identifier**, which enables a virtualized CCN router to identify the VCCN slice to which every Interest/Data packet belongs.

- **CCN router virtualization**, which makes it possible to operate a single CCN router as multiple VCCN router instances.

- **Packet transport between virtualized CCN routers**, which enables packet delivery between virtualized CCN routers (i.e., CCN routers run-

ning VCCN router instances) which are not adjacent in the content-centric network.

- **SNS cooperative user/group identification**, which enables virtualized CCN routers to identify the sender and the receiver of Interest and Data packets for realizing group-based communication.

The first three building blocks—extension of the content identifier, CCN router virtualization, and packet transport between virtualized CCN routers—realize traffic separation for VCCN slices. The last building block, SNS cooperative user/group identification, prevents injection of unauthorized traffic into a VCCN slice by an outsider.

In the following, the author describes these building blocks in more detail.

2.3.2 Extension of the Content Identifier

Content identifiers in CCN are extended to enable a virtualized CCN router to identify the VCCN slice to which every Interest/Data packet belongs. Specifically, a VCCN identifier is embedded in a content identifier. Since content identifiers are variable-length bit strings, a VCCN identifier can be embedded in a content identifier in various ways.

An example of embedding a VCCN identifier in a content identifier is illustrated in Fig. 2.2. In this case, components of the content identifier are separated by slash delimiters. The first two components are used as the VCCN declaration and the VCCN identifier. Specifically, if the first component in a content identifier is `VCCN_ID`, a virtualized CCN router regards the packet as belonging to a VCCN slice and treats the second component as a VCCN identifier. If the first component is not `VCCN_ID`, the content identifier is interpreted as a standard CCN content identifier. Such a simple extension of the content identifier enables the isolation of name spaces, one of which is assigned to every VCCN slice.

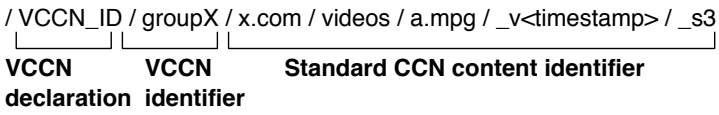


Figure 2.2: Example of an extended content identifier; the first two components are used as the VCCN declaration and the VCCN identifier.

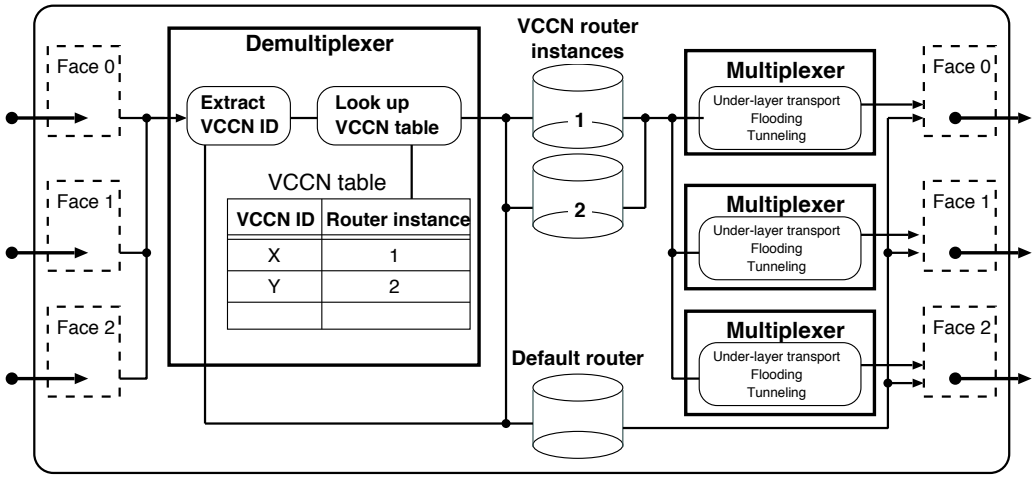


Figure 2.3: A virtualized CCN router; it is composed of a demultiplexer, VCCN router instances, and multiplexers.

2.3.3 CCN Router Virtualization

CCN router virtualization can be easily realized by switching three data structures used for packet routing in CCN: the forwarding information base (FIB), CS, and pending interest table (PIT) [1]. A CCN router can be equipped with multiple FIBs, CSs, and PITs and one of each of these tables is assigned to each VCCN slice. The CCN router selects an appropriate set of FIB, CS, and PIT according to the VCCN identifier embedded in a content identifier.

A virtualized CCN router is composed of a demultiplexer, VCCN router instances, and multiplexers (see Fig. 2.3). The author explains the operations of the demultiplexer, VCCN router instances, and multiplexers by describing the flow of packet processing.

An Interest/Data packet arriving at a face of a CCN router is first passed to

the demultiplexer. The demultiplexer tries to extract a VCCN identifier embedded in the content identifier of the packet. If the VCCN identifier can be extracted, the demultiplexer checks whether a VCCN router instance corresponding to the VCCN identifier exists in the CCN router. If the VCCN router instance exists, the packet is passed to that instance. If the VCCN identifier cannot be extracted from the content identifier or the VCCN router instance does not exist, the packet is passed to the default router, which routes and forwards packets as an ordinary CCN router.

A VCCN table manages the correspondence between a VCCN identifier and a VCCN router instance. Each entry of a VCCN table is a pair of a VCCN identifier and an identifier of the corresponding VCCN router instance.

A VCCN router instance routes packets received from the demultiplexer using its own data structures (i.e., FIB, CS, and PIT), and it determines one or more faces through which to send the packet out. Note that the VCCN router instance uses the remainder of the content identifier (i.e., a content identifier in a VCCN slice) rather than the entire content identifier. Finally, the CCN router emits the packet from one or more faces through multiplexers, which are responsible for realizing packet transport between virtualized CCN routers.

2.3.4 Packet Transport between Virtualized CCN Routers

A multiplexer emits the packet received from a VCCN router instance through faces of the virtualized CCN router. The multiplexer enables packet transport between virtualized CCN routers, which are commonly not adjacent in the content-centric network.

VCCN supports the following three types of packet transport between virtualized CCN routers.

- **Packet transport in a lower layer**

  The simplest and the most efficient approach is to use a protocol layer lower than CCN if that layer supports *any-to-any* communication (Fig. 2.4). CCN can operate on variety of lower layer protocols such as IP, UDP, TCP, broadcast communication, Ethernet, and P2P [27].

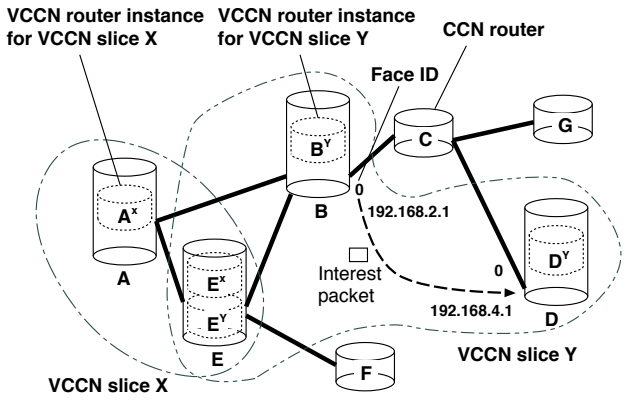


Figure 2.4: Packet transport in a lower layer; if a lower layer protocol supports communication between an arbitrary pair of nodes (e.g., IP, UDP, TCP, and broadcast communication), any pair of CCN routers can communicate using the lower layer protocol.

If a lower layer protocol supports communication between an arbitrary pair of nodes (e.g., IP, UDP, TCP, and broadcast communication), any pair of CCN routers can communicate using the lower layer protocol. Hence, packet transport between virtualized CCN routers can be easily realized.

- **Flooding in the CCN layer**

  If any-to-any communication is not supported in a lower layer protocol, then a simple approach is to flood the CCN layer (Fig. 2.5). In CCN, duplicate Interest packets are simply discarded. Hence, flooding can be realized simply by duplicating Interest packets and sending them through all faces of every CCN router. However, flooding is not efficient and might result in an excessive amount of traffic in a content-centric network. So flooding should not be permitted, especially when VCCN slices are sparsely constructed.

- **Tunneling in CCN layer**

  A complicated but more efficient approach than flooding is to tunnel packets through intermediate CCN routers. Even when any-to-any communication is not supported in a lower layer protocol than CCN and inefficiency caused

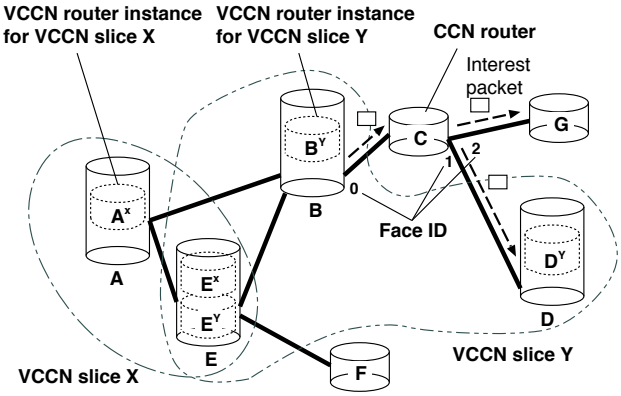


Figure 2.5: Flooding in the CCN layer; in CCN, flooding can be realized simply by duplicating Interest packets and sending them through all faces of every CCN router.

by the flooding in CCN layer is not acceptable, tunneling in CCN layer can transport packets between virtualized CCN routers (Fig. 2.6).

Since CCN is not a host-centric network architecture, tunneling in the CCN layer cannot be realized by a simple approach like IP-in-IP [28]. However, tunneling in the CCN layer is still realizable with source routing [2].

In CCN, a Data packet is sent back to the user by traversing the path of the Interest packet in reverse. Such path symmetry for Interest and Data packets is realized using the PIT as *bread crumbs* [1]. Hence, if a list of faces through which a packet should traverse is specified in any way, the locus of the packet can be controlled.

Based on this idea, Interest/Data packet headers are extended to store *source routing options* for realizing the tunneling in the CCN layer. Like the source routing option in IP [2], a CCN router forwards the packet to the face written at the head of the source route. Specifically, a multiplexer provides list of faces that the packet should traverse as a source routing option in the packet header. If a source routing option is specified in a packet, the demultiplexer in each CCN router pops the face from the head of the list, and transfers the packet through that face.

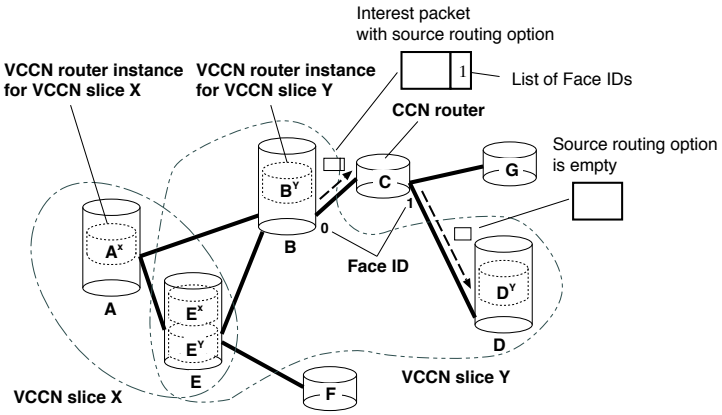


Figure 2.6: Tunneling in CCN layer; as in the source routing option of IP, the CCN router forwards the packet to the face listed at the head of the source route.

2.3.5 SNS Cooperative User/Group Identification

Social Networking Services (SNSs) such as Facebook and Google+ have become increasingly popular in the last decade. In those SNSs, users can dynamically create and modify groups, each of which generally corresponds to a set of friends and colleagues.

In VCCN, to significantly simplify user/group management, virtualized CCN routers utilize user/group information registered in an SNS for authenticating senders and receivers of Interest and Data packets. That is, VCCN and SNS work cooperatively to realize group-based communication. The author believes such a cross-layer cooperation between the network layer (i.e., VCCN) and the application layer (i.e., SNS) should dramatically ease the realization and management of user-aware communication services, such as group-based communication. Note that a similar idea has been proposed in SocialVPN [9].

In SNS cooperative user/group identification, virtualized CCN routers at the edge of a VCCN slice identify whether a user is allowed to access that VCCN slice. Access to a VCCN slice is checked only at these edge routers; once an Interest packet has been forwarded, the downstream virtualized CCN routers do not care about the source of the Interest packet.

Basically, every router at the edge of a VCCN slice is also a proxy for an SNS

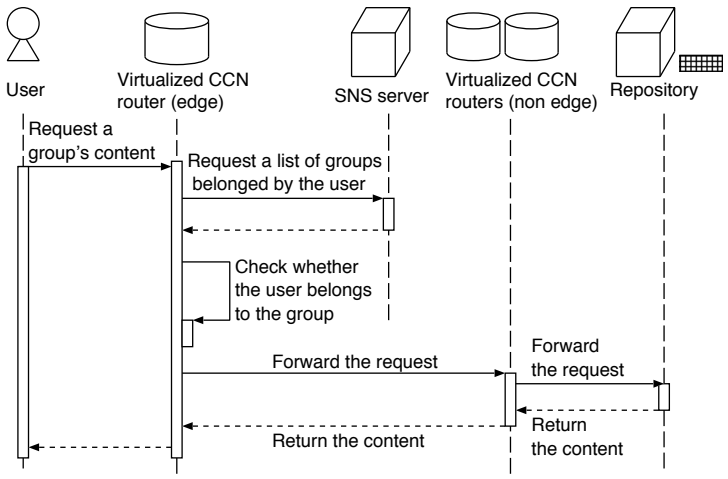


Figure 2.7: Sequence diagram for requesting content in VCCN.

authentication service (see Fig. 2.7). When users want to access a VCCN slice, they communicate with a router at the edge of the VCCN slice and sends their identification information (e.g., username and password in an SNS). The router forwards the identification to the SNS server to check its validity and determine whether the user belongs to the group corresponding to the VCCN slice. The user is allowed to send or receive packets only when both of these conditions are satisfied.

When a content is registered to a VCCN slice, every router at the edge of the VCCN slice is a proxy to a SNS authentication service, too (see Fig. 2.8). A repository communicates with a router at the edge of the VCCN slice before content registration. The repository sends its identification information (e.g., repository name and password in an SNS). As with access to a VCCN slice, the router forwards the identification to the SNS server, and checks to see whether the repository identification is valid and whether the repository belongs to the group corresponding the VCCN slice. When both conditions are satisfied, the router returns the VCCN identifier of the group to the repository. The repository embeds the received VCCN identifier in a content identifier and registers the content. The repository then performs the *Register* operation [1] in order to advertise the prefix of the registered content to VCCN routers on the VCCN slice. In a *Register* operation,

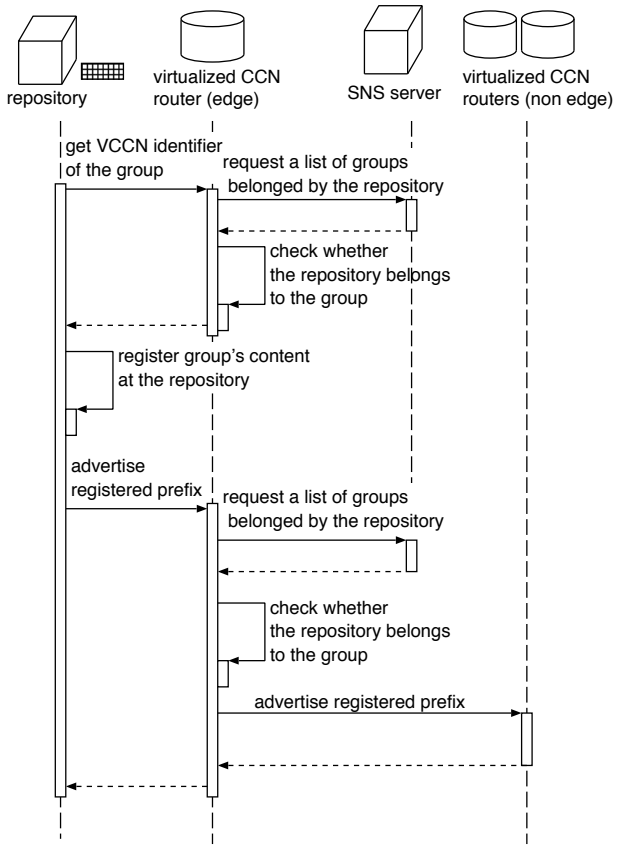


Figure 2.8: Sequence diagram when registering a content in VCCN.

the repository forwards its identification information and an Interest packet to advertise the prefix to the router at the edge of the VCCN slice. The repository is allowed to advertise the registered prefixes to routers in the VCCN slice only if both identification checks are successful again.

Although such cross-layer cooperation may debase the performance of a VCCN slice, there are some remedies. For example, after the diffusion of CCN an edge router will be able to communicate with an SNS server, using CCN rather than through an application layer protocol (e.g., HTTP). Moreover, since a CCN router can cache content in its CS, the CCN router can reuse the group information that it has received from an SNS server. Hence, requests for the group information in the procedures of content request and registration can be skipped and these procedures will be simpler than what was estimated.

2.4 Implementation and Evaluation

2.4.1 VCCN Implementation

The author implemented VCCN's basic features by extending the CCNx software [24], an open-source implementation of the CCN protocol. Our VCCN implementation is realized as wrapper programs for CCNx commands (e.g., `ccndstart`, `ccndstop`, `ccndc`, `ccndgetfile`, `ccndputfile`), and proxy software for SNS cooperative user/group identification. Our VCCN implementation allows users to initiate and terminate VCCN router instances, connect arbitrary VCCN router instances, and register and fetch content in a VCCN slice.

Our VCCN implementation realizes traffic separation for VCCN slices in the following way. An edge router of a VCCN slice embeds a user's VCCN identifier in the content identifier immediately after the user requests some content through the wrapper programs. In our VCCN implementation, a CCN router is virtualized by logically splitting the FIB for each VCCN slice: specifically, every FIB entry is tagged with a VCCN identifier. For simplicity, the CS and PIT are shared among all VCCN slices. Packet transport between virtualized CCN routers is realized with a lower layer protocol (UDP).

The author prevents the injection of unauthorized traffic from a user using Facebook's authentication mechanism. When a user/repository accesses a VCCN slice, an edge router with the proxy software checks for the relevant authorization using the provided identification information and an access token. Specifically, the edge router uses the Graph API of Facebook [29] to perform user/group identification. The Graph API can acquire a user's information from Facebook using an access token that is created at the time of the user's login (Fig.2.9). In the implemented identification, when a user/repository accesses to a VCCN slice, the user/repository passes an access token and a group name to the edge router of the VCCN slice. The edge router makes identification by checking whether there is the specified group in the group list to which the user/repository belongs obtained through Graph API. If the user/repository belongs to the specified group, the edge router replaces the group name with the identifier managed by Facebook

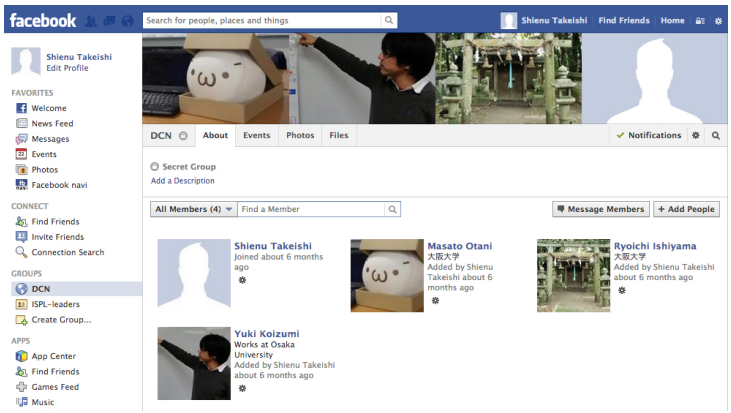


Figure 2.9: Example of creating a group; four members are registered with the data-centric networking group on Facebook and every registered member can take part in group-based communication.

and embeds that group identifier in a content identifier. The edge router then looks up the FIB corresponding to the group and forwards the extended Interest packet to relay routers.

In our VCCN implementation, an outsider cannot request any content of a group through VCCN. In particular, our VCCN implementation can discard several types of illegal Interest packets: (1) an Interest packet that a user who does not belong to any group requests through VCCN; (2) an Interest packet that a user belonging to another group requests through VCCN; and (3) an Interest packet that a user belonging to the group requests through CCN. Fig. 2.10 shows the processes for discarding these three types of packet. In case (1), an edge router judges the user to be unauthorized and discards the Interest packet during SNS cooperative user/group identification. In case (2), an edge router does not discard the Interest packet during SNS cooperative user/group identification. However, one of relay routers misses the longest-prefix matching of the Interest packet and discards it because it is transported in the VCCN slice of a different group. In case (3), an edge router does not discard the Interest packet during SNS cooperative user/group identification. However, one of relay routers misses the longest-prefix matching of the Interest packet and discards it because it is not correctly extended based on a group identifier and is being transported in a global content-centric

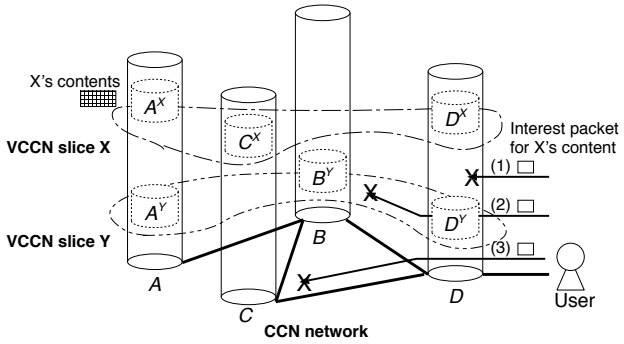


Figure 2.10: Processes for discarding three types of packet: (1) an Interest packet for X's content that a user who does not belong to any group requests through VCCN; (2) an Interest packet for X's content that a user belonging to Y requests through VCCN; and (3) an Interest packet for X's content that a user belonging to X requests through CCN.

network.

2.4.2 Performance Evaluation of the VCCN Implementation

The author conducted preliminary performance evaluations of our VCCN implementation. In the first experiment, content delivery delays in our VCCN implementation and the original CCNx are compared. In the second experiment, the author evaluates overhead of the CCN router virtualization using our VCCN implementation.

For the first experiment, the author used the network topology shown in Fig. 2.11—four CCN routers are connected, and two VCCN slices X and Y are built.

In the CCNx setup, 100 items of size 10 [Kbyte] are stored in CCN router A, and CCN router D randomly requests one of those items 3,000 times. Note that the hop count from the source (CCN router A) to the user (CCN router D) is always one.

In the VCCN setup, 50 items of size 10 [Kbyte] are stored in each of VCCN router instances A^X and A^Y. VCCN router instances D^X and D^Y randomly request one of those items in their VCCN slice 3,000 times. Note that the average

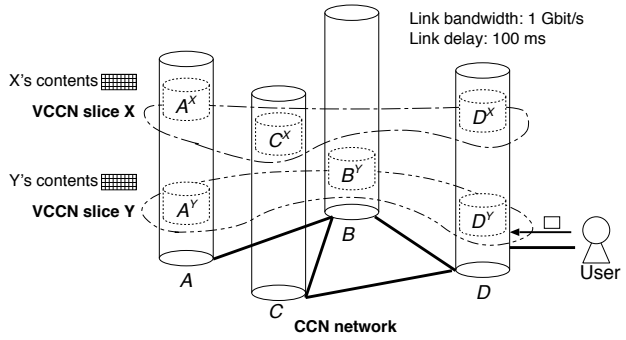


Figure 2.11: Network topology used in the CNNx/VCCN comparison; four CCN routers are connected and two VCCN slices, X and Y, are created.

hop count from the source (CCN router A) to the user (CCN router D) is 1.5 (i.e., one hop in VCCN slice Y and two hops in VCCN slice X).

The communication delays of all links are identically set to 100 [ms] using network emulators. The size of the CS ($CCND_CAP$) is set to 100 in all CCN routers except CCN routers A and D, whose packet caching is disabled. The author measured the content delivery delay disregarding the delays caused by identification processing.

Figure 2.12 shows the CDF (Cumulative Distribution Function) of content delivery delays in our VCCN implementation and in the original CCNx. Somewhat surprisingly, the content delivery delays in VCCN and CCNx are comparable even though VCCN has a larger hop count between the source and the consumer than CCNx: the average content delays were 2.79 [s] in VCCN and 2.53 [s] in CCNx. This similarity can be explained by the effect of content caching in CCN routers: CCNx utilizes the CS only in CCN router B, but VCCN utilizes the CSs in routers B and C. For instance, in our experiment, the average cache hit rate of CCN routers with VCCN was 51.8% whereas that without VCCN was 44.9%. VCCN router instances are dispersed in the network, so that VCCN can effectively utilize, at least in this experiment, the content stores in CCN routers.

It should be noted that efficiency of VCCN relies significantly on several factors, such as the CCN and VCCN network topologies, so the author does not claim that VCCN is more efficient than CCN. Instead, the author just addressed the

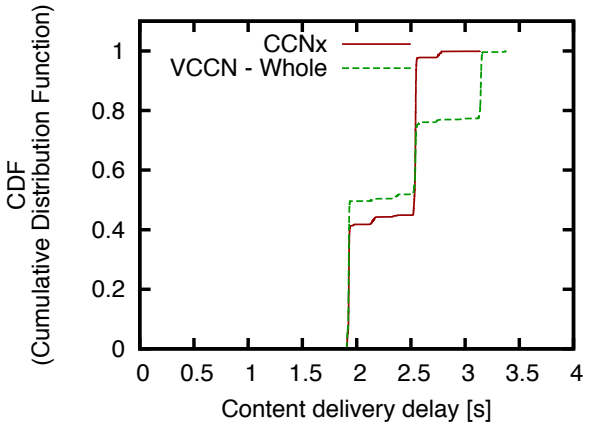


Figure 2.12: CDFs for content delivery delay when content is requested through a content-centric network and the VCCN slices.

question of whether the introduction of VCCN has a positive or negative impact on CCN performance. The author is planning to conduct more detailed experiments.

Secondly, since it is expected that the performance of VCCN slices will be debased by CCN router virtualization, the author evaluated overhead of the CCN router virtualization using our VCCN implementation. The author chooses to two types of metrics in affecting the overhead: content request rate and the number of VCCN slices. In the experiment, the result of setting the number of content items, which are stored in CCN router A or VCCN router instances A^X and A^Y, to 10,000 and setting $CCND_CAP$ to 1,000 were investigated.

Fig. 2.13 shows average content delivery delays against content request rate. This figure indicates that there is almost no change in average content delivery delay until the request rate reaches 8 [request/s]. Moreover, this figure indicates that CCN router virtualization does not affect the performance of a VCCN slice because the average content delivery delays in our VCCN implementation and in CCNx both increase at a rate of 8 [request/s].

Next the author considered how average content delivery delays vary with the number of VCCN slices in a content-centric network. When increasing the number of VCCN slices, the number of content items in the content-centric network is fixed and equal numbers of topologies X and Y for the VCCN slices are constructed.

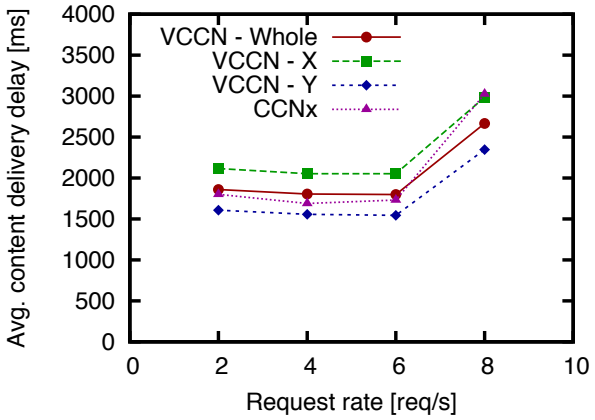


Figure 2.13: Average content delivery delays against content request rate.

For example, when the number of VCCN slices is 10, there are five X and five Y VCCN network topologies. In this experiment, the content request rate is set to 2 [request/s]. Fig. 2.14 indicates that the performance of virtualized CCN routers is not debased even if the number of VCCN slices is substantially increased. Moreover, the fact that the average content delivery delays are maintained does not depend on the network topology. The experiment results show that CCN router virtualization in VCCN incurs a little overhead to CCN in terms of the network performance.

2.5 Open Issues

In this section, the author discusses open research issues of VCCN slice construction based on knowledge acquired by designing, implementing and evaluating VCCN slices.

2.5.1 CCN Router Resource Management

One important issue for virtualizing a content-centric network is how resources (i.e., the FIB, CS, and PIT) of a CCN router are allocated to each VCCN router. Since a CCN router uses the three structures for routing a packet, the allocation of these resources affects the performance, confidentiality and robustness of a

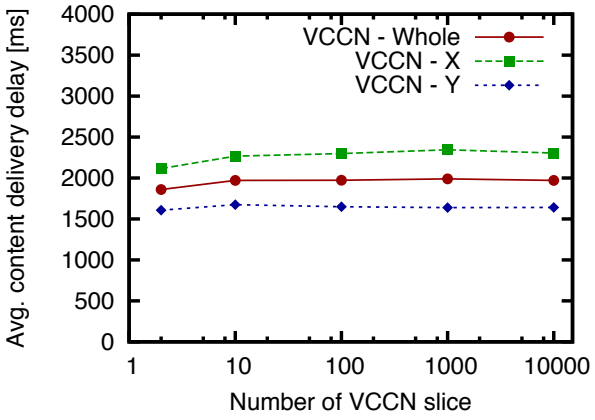


Figure 2.14: Average content delivery delays against the number of VCCN slices in a content-centric network.

network.

In content-centric network virtualization, the author will have to focus on the trade-off between overall performance and fairness. Sharing the resources of a CCN router among groups/applications is better than allocating the resources to each group/application in order to maximize overall network performance [17,18]. On the other hand, sharing the resources of a CCN router among groups may cause unfairness between groups. For instance, if the CS of a physical CCN router is shared between VCCN routers and the traffic of a certain group is especially large, the CS can be effectively occupied by the VCCN router of that group [17]. Then, while network performance for the group, whose VCCN router occupied the CS, may be very high, network performance for the other groups will be low. In a similar way, if the PIT is monopolized by a certain group, users of the other groups will not be able to communicate. This also means that, if a malicious user can gain access to any VCCN slice, that user can obstruct another VCCN slice by interest flooding [30].

To prevent resource occupation of a CCN network and improve overall network performance, the author should design a method to allocate the resources of a physical CCN router to each group. Some related methods have been proposed [16, 18] and our research group is planning to investigate analytically the effect of CS

allocation methods and content request patterns in VCCN slices on the average content delivery time of each separate VCCN slice and the entire network.

In regard to this issue, in the next chapter, the author analytically investigates the effect of CS allocation methods and content request patterns in VCCN slices in terms of the network fairness for VCCN slices and overall network performance.

2.5.2 VCCN Slice Mapping

The virtual network mapping/embedding problem, which means mapping virtual routers and links to specific nodes and links in the substrate network, has been investigated in previous studies of virtualization [31–33]. Since this mapping is an NP-hard problem, heuristics for providing efficient performance were proposed in these studies.

In content-centric network virtualization, existing virtual network mapping methods may not be applicable because these methods do not take data reuse into account. Mapping VCCN slices influences the effect of caching as well as performance and traffic. For example, in the experiment of Section 2.4, the content delivery delays in VCCN and CCNx are comparable due to a change of caching effect, despite the mapping increasing the average hop count from the user to the source. Furthermore, the efficiency of caching and network performance may be increased by increasing the number of relay VCCN routers in a VCCN slice. It is desirable to study this problem, taking the effect of caching into account.

2.5.3 Reliability

Although VCCN is a general and practical network architecture, there are some improvements required in order for a VCCN to operate as a reliable network architecture in various environments.

One necessary improvement is the decentralized management of a VCCN declarator and VCCN identifiers. VCCN realizes traffic separation between VCCN slices and a substrate content-centric network by checking if a VCCN declaration exists. Moreover, as in IP-VPN, VCCN uses label switching based on a VCCN identifier. Hence, in VCCN, it is necessary that an unauthorized user cannot spec-

ify a valid VCCN declaration and identifier. Our VCCN implementation solves this problem by defining a VCCN declaration `VCCN_ID` as a block phrase and getting Facebook to manage the VCCN identifiers of all groups. However, this solution places a lot of management load on Facebook. If the decentralized management of VCCN identifiers can be realized, VCCN will be more reliable network architecture. Moreover, if VCCN slices are constructed on a content-centric network composed of multiple autonomous systems, the decentralized management of VCCN declarators and identifiers must be performed reliably between the autonomous systems.

Another requirement is a lightweight and robust authentication mechanism, since routers at the edge of a VCCN slice authenticate users and consequently experience a huge load. On the other hand, countermeasures against the attacks of a malicious user should be implemented. For instance, a malicious user may attempt a denial-of-service attack on a VCCN slice by repeatedly accessing an edge router because of the load applied to the router in SNS cooperative user/group identification. This method may also be used to attack the authentication server itself. In regard to these attacks, the author will need not only to divide authentication processes and routing processes between a control plane and a forwarding plane but also implement a quick and lightweight authentication mechanism in order to prevent a content-centric network going down.

2.6 Summary

In this chapter, the author has proposed VCCN, which realizes group-based communication through CCN router virtualization. The fundamental idea is to operate a CCN router as multiple instances of VCCN routers, which run logically independently. Group-based communication is realized by building VCCN slices, which are composed of multiple VCCN router instances.

The author has implemented VCCN's basic features by extending the CCNx software and have conducted a preliminary performance evaluation of our implementation. The evaluation showed that virtualization has both positive and negative impacts on CCN performance and has the scalability of virtualized CCN

routers with respect to request rate and the number of VCCN slices. The author has also discussed open research issues in VCCN network construction based on knowledge acquired by designing, implementing and evaluating VCCN.

Chapter 3

Cache Performance Analysis of Virtualized Router on Virtual Content-Centric Networks

3.1 Introduction

Virtual content-centric networking (VCCN), which enables the construction of multiple virtual networks (called VCCN slices) on a content-centric network, has been recently proposed [34] as the author introduced in Chapter 2. VCCN slices are constructed by operating a CCN router as multiple, logically independent VCCN router instances and by logically connecting VCCN router instances that are not adjacent in the network.

When multiple VCCN slices are constructed, the performance of each VCCN slice and that of the entire network are strongly affected by the CCN routers' resource allocation to VCCN router instances in VCCN slices. Several previous studies have shown clearly that, in CCN, the effectiveness of content caching depends strongly on the content request pattern experienced by the CS of a CCN router [12–14]. Hence, the performance of each VCCN slice and that of the entire network depend strongly on how a CCN router allocates its CS to VCCN router instances on VCCN slices that have different content request patterns.

In this chapter, the author analytically investigates the effect of CS allocation methods and content request patterns in VCCN slices in terms of the network fairness for VCCN slices and overall network performance. On the assumption that a network provider provides groups with VCCN slices, the network provider should equally provide benefit of the resource allocation to the groups but would simultaneously want to maximize the efficiency of resource utilization. Hence, the author develops a mathematical model of virtualized CCN router to regulate the network fairness for VCCN slices and overall network performance, and quantitatively investigates the trade-off among those metrics. The author focuses on the effects of the content popularity slope and the content request ratio of each VCCN slice, which are the main features of a content request pattern and which significantly affect the effectiveness of content caching in particular.

In this chapter, the author focuses on three types of CS allocation methods: an exclusive method, a shared method and a hybrid method. In the exclusive method, each VCCN router instance within a CCN router monopolizes a given part of its CS. In the shared method, all VCCN router instances within a CCN router use its entire CS jointly. In the hybrid method, several VCCN router instances within a CCN router are assigned their own parts of its CS and other instances jointly use the remaining CS. Previous studies of the effects of content caching on content-centric networks have focused only on the exclusive and shared methods [16–19]. However, when content request patterns are heterogeneous, these two methods can barely maintain a balance between network fairness for VCCN slices and overall network performance. Hence, the author conjectured that a hybrid method, which has the characteristics of both the exclusive and shared approaches, might be a useful CS allocation method on a content-centric network in which there are multiple content request patterns in VCCN slices. In this chapter, the author quantitatively compares a hybrid method with the two existing methods in terms of the fairness for VCCN slices and the overall network performance.

The main contribution of this chapter is twofold. First, the author develops a mathematical model of virtualized CCN router for cache performance analysis under arbitrary content request patterns, and derive the cache hit rate for each

VCCN router instance and the aggregated cache hit rate of the virtualized CCN router. Second, through numerical examples, the author quantitatively shows that in diverse scenarios, the hybrid method can provide desirable trade-offs among the network fairness for VCCN slices and overall network performance.

The organization of this chapter is as follows. Section 3.2 contains a summary of related work. In Section 3.3, the author describes CCN router virtualization and CS allocation to VCCN router instances. In Section 3.4, our model of a virtualized CCN router which accommodates multiple VCCN router instances is described and analytical results are derived. In Section 3.5, through several numerical examples, the author analyzes the effects of CS allocation methods and content request patterns in VCCN slices on the network fairness for VCCN slices and the overall network performance. Finally, in Section 3.6, the author gives our conclusions and indicate the direction of future work.

3.2 Related Work

The effect of content caching on content-centric networks where multiple applications or services are running has been investigated in [16–19]. Carofiglio *et al.* [16,18] have clarified the role of the CS allocation method (an exclusive method) on the cache hit rate of multiple applications running on content-centric networks by means of experiments and simulations. Their results show that an exclusive method can guarantee application performance but it may decrease the overall performance of the entire network. Fricker *et al.* [19] have evaluated the cache hit rate of multiple services running concurrently on a content-centric network using an approximation proposed by Che *et al.* [35]. Their results show that allowing a service with a rapid content popularity slope to monopolize the CS raises the cache hit ratio rather than increasing the quantity of CS shared between all the services, when the size of the CS is large. Ohsugi *et al.* [17] investigated by means of a simulation the effect of CS allocation in CCN router virtualization (an exclusive method) in terms of the average content delivery time for an entire content-centric network on which multiple applications are running. Their results show that exclusive CS allocation in CCN router virtualization increases the average content

delivery time by about 20% in the worst case and improves network fairness for applications.

These studies show that while a shared method is preferable for maximizing the performance of the entire network, an exclusive method is preferable for improving network fairness for applications or services. However, a hybrid method, which will strike a balance between performance and fairness, has not yet been described or quantitatively evaluated.

The effect of a content request pattern in a content-centric network on caching performance has been investigated in [12, 13]. Rossini *et al.* [12, 13] evaluated by means of simulations the dependence of the cache hit rate on several aspects of network design, such as topology, content size, content popularity, the locality of user requests and the number of repositories. Their results show that the Zipf exponent α, representing content popularity, can have a dramatic impact on the performance of the entire network. However, in those studies, CCN router virtualization was not taken into consideration, and the relation between CS allocation methods and content request patterns in VCCN slices has not yet been clarified.

3.3 CS Allocation to VCCN Router Instances

In the following, the author describes three types of conceivable methods of allocating CS resources to VCCN router instances: an exclusive method, a shared method and a hybrid method (see Fig. 3.1).

- Exclusive method

  Each VCCN router instance monopolizes a given part of the CS of a CCN router. One advantage of the exclusive method is that the performance of content caching in a VCCN router instance is independent of that in other VCCN router instances. On the other hand, a disadvantage is that the cache miss rate may increase because a VCCN router instance monopolizes a given part of the CS regardless of the amount of traffic in the VCCN router instance.

- Shared method

All VCCN router instances jointly use the entire CS of a CCN router. One advantage of the shared method is that there is no loss of CS due to splitting of the CS and the cache hit rate may be increased because all VCCN router instances use the entire CS. On the other hand, a disadvantage is that a VCCN router instance with a large amount of traffic may monopolize most of the CS because each VCCN router instance affects the others within a given CCN router.

- Hybrid method

  Several VCCN router instances within a CCN router are assigned their own parts of its CS and other instances jointly use the remaining CS. Advantages of the hybrid method are that CS loss is reduced relative to the exclusive method and each VCCN router instance has a minimal effect on the others within a given CCN router. On the other hand, a disadvantage is that the management of the CS may be more complicated than in either the exclusive method or the shared method.

  Another approach, in which each VCCN router instance monopolizes a small part of the CS and all the instances jointly use the remaining CS, is also conceivable as a hybrid method. However, in this approach, each VCCN router instance needs to be assigned a certain amount of CS. As the number of VCCN router instances running on a CCN router increases, the required size of CS also increases. Hence, this hybrid approach is not considered in this chapter.

3.4 Virtualized CCN Router Model

3.4.1 Model Description and Notation

Our virtualized CCN router model is presented in Fig. 3.2. VCCN slices $S^n (1 \leq n \leq N)$ are constructed on a content-centric network, and VCCN router instances $R^n (1 \leq n \leq N)$ corresponding to the VCCN slices operate on a virtualized CCN router.

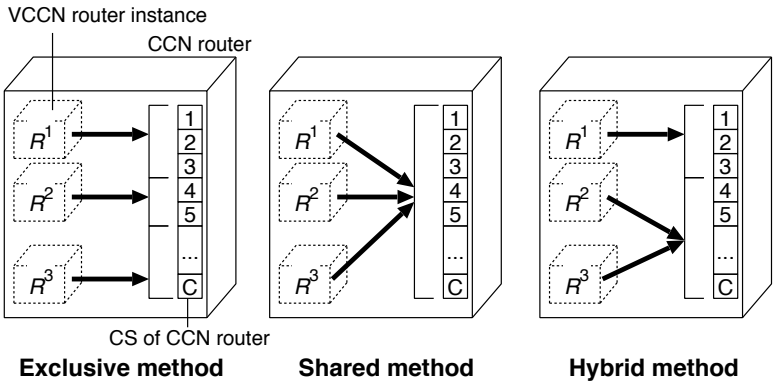


Figure 3.1: Examples of methods for the allocation of CS to VCCN routers (an exclusive method, a shared method and a hybrid method).

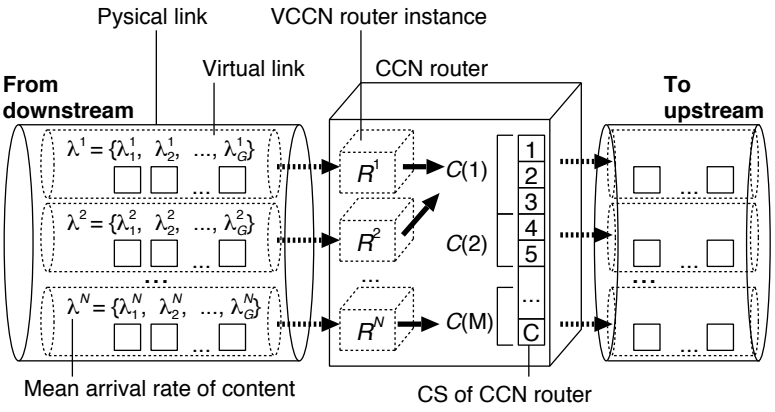


Figure 3.2: The model considered in this chapter.

The author assumes that the request arrival process for content c on VCCN router instances R^n is Poisson with mean arrival rate λ_c^n. Thus, neither the content popularity slope nor the content request ratio for each VCCN slice vary dynamically. G denotes the number of content items requested from all the VCCN router instances.

The CS of the virtualized CCN router is partitioned into M segments and segment $m(1 \leq m \leq M)$ of size $C(m)$ is jointly used by all VCCN router instances

belonging to a set $\Omega(m)$. The sets $\Omega(m)$ satisfy the following relations.

$$\Omega(m) \subset \{R^1, R^2, \cdots, R^N\} \tag{3.1}$$

$$\bigcup_{m=1}^{M} \Omega(m) = \{R^1, R^2, \cdots, R^N\} \tag{3.2}$$

$$\sum_{m=1}^{M} |\Omega(m)| = N \tag{3.3}$$

So a VCCN router instance does not use multiple segments. Each segment employs a least recently used (LRU) replacement policy.

In our model, each Interest and Data packet has size L in order to simplify the problem. If Data packets have different sizes, the author can use the methods of Fricker et al. [36].

Moreover, the author does not consider the aggregation of requests for the same content on the virtualized CCN router because request aggregation has no or little impact on the stationary average content delivery time [37]. When a CCN router receives an Interest packet for content that is already being requested, the CCN router prevents the dispatch of that Interest packet.

In addition, the author assumes that the processing times for managing the CS, writing Data packets into the CS and reading Data packets from the CS are negligible.

3.4.2 Determination of the Cache Hit Rate based on a Markov Chain Model

First, the author derives the cache hit rate p^n for each VCCN router instance $R^n (1 \leq n \leq N)$ and the aggregated cache hit rate p of the virtualized CCN router based on a Markov chain model. Although it is difficult to derive the performance of a large-scale content-centric network in this model because of the huge computational complexity, the state distribution for each successive storage of content in the CS is acquired. Hence, the author can estimate the size of CS required to hold a specific amount of content.

Here, the author focuses VCCN router instances $R^n \in \Omega(m)$ which jointly use segment m of the CS. The author denotes the state, in which content c on

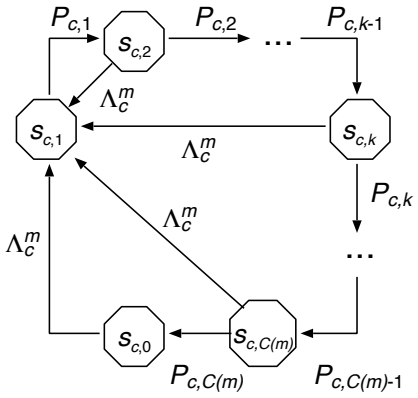


Figure 3.3: Markov chain model. The state in which content c on VCCN slice S^n is placed in the kth segment of the CS is denoted by $s_{c,k}(0 \leq k \leq C(m))$.

VCCN slice S^n is in the kth segment of the CS, by $s_{c,k}$ and the author considers the Markov chain composed of $s_{c,k}(0 \leq k \leq C(m))$ (see Fig. 3.3). The author assumes without loss of generality that the new content is written into the top of the CS segment. Note that the state in which content c on VCCN slice S^n is not in the CS segment is denoted by $s_{c,0}$ and transitions to the same state are omitted in Fig. 3.3.

Let $P_{c,k}$ and $P_{c,C(m)}$ be the transition rate from $s_{c,k}$ to $s_{c,k+1}$ and from $s_{c,C(m)}$ to $s_{c,0}$, respectively. $P_{c,k}$ is given by

$$P_{c,k} = \begin{cases} \sum_{i=1}^{G} \Lambda_i^m - \Lambda_c^m & k = 1 \\ \frac{\prod_{i=1}^{k} P_{c,i}}{\prod_{i=1}^{k-1} P_{c,i}} & 2 \leq k \leq C(m) \end{cases} \tag{3.4}$$

where Λ_c^m is the request arrival rate of content c at CS segment m which is given by $\sum_{R^n \in \Omega(m)} \lambda_c^n$

$$\begin{aligned} \prod_{i=1}^{k} P_{c,i} &= (k-1)! \sum_{\Xi \in \Theta_c^{k-1}} \prod_{\Lambda_j^i \in \Xi} \Lambda_j^i \\ &\quad \times \Big(\sum_{i=1}^{G} \Lambda_i^m - \sum_{\Lambda_j^i \in \Xi \cup \{\Lambda_c^m\}} \Lambda_j^i\Big) \end{aligned} \tag{3.5}$$

where Θ_c^k is the set composed of request rates for content in segment k, which satisfies $\Theta_c^k \subset \bigcup_{i=1, i \neq c}^{G} \Lambda_i^m$ and $|\Theta_c^k| = k$.

Let $\pi_{c,k}(0 \leq k \leq C(m))$ be the equilibrium probability of $s_{c,k}$. Then $\pi_{c,k}$ is given by

$$\pi_{c,k} = \begin{cases} \frac{\prod_{i=1}^{C(m)} P_{c,i}}{\prod_{i=1}^{C(m)} (\Lambda_c^m + P_{c,i})} & k = 0 \\ \frac{\Lambda_c^m}{\Lambda_c^m + P_{c,1}} & k = 1 \\ \frac{\Lambda_c^m \prod_{i=1}^{k-1} P_{c,i}}{\prod_{i=1}^{k} (\Lambda_c^m + P_{c,i})} & \text{otherwise} \end{cases} \tag{3.6}$$

The cache hit rate p_c^n of content c on VCCN slice S^n can be derived from $\pi_{c,k}$.

$$p_c^n = 1 - \frac{\prod_{i=1}^{C(m)} P_{c,i}}{\prod_{i=1}^{C(m)} (\Lambda_c^m + P_{c,i})} \tag{3.7}$$

Finally, the cache hit rate p^n for each VCCN router instance $R^n(1 \leq n \leq N)$ and the aggregated cache hit rate p of the virtualized CCN router are given by the following equations.

$$p^n = \frac{\sum_{c=1}^{G} \lambda_c^n p_c^n}{\sum_{c=1}^{G} \lambda_c^n} \tag{3.8}$$

$$p = \frac{\sum_{n=1}^{N} \sum_{c=1}^{G} \lambda_c^n p_c^n}{\sum_{n=1}^{N} \sum_{c=1}^{G} \lambda_c^n} \tag{3.9}$$

3.4.3 Determination of the Hit Rate using an Approximation Method

The author can derive an approximation to the cache hit rate p^n for each VCCN router instance $R^n(1 \leq n \leq N)$ and the aggregated cache hit rate p of the virtualized CCN router from the hierarchical Web caching model [35]. In this approximation, the network performance of a large-scale content-centric network can also be derived.

If the size of the cache memory is C, the cache replacement policy is LRU and the request arrival process for content c is Poisson with mean arrival rate λ_c, the cache hit rate of content c is given by

$$p_c \simeq 1 - e^{-\lambda_c t_c} \tag{3.10}$$

where t_c is called the *characteristic time* of content c and is defined as the maximum inter-arrival time between two adjacent requests for content c without a cache miss

at the cache [35]. The characteristic time can be calculated by solving

$$\sum_{i=1,i\neq c}^{G} F_i(t < t_c) = C \tag{3.11}$$

where $F_c(t < t_c)$ is the cumulative distribution $(1 - e^{-\lambda_c t_c})$ of the inter-arrival time for requests for content c at the cache. Without loss of generality, suppose that the caching of content c occurs at $t = 0$. Thus, the characteristic time t_c is the time at the CS, whose size is C, will be filled with content other than c. In addition, (3.10) and (3.11) can be simplified as follows [36].

$$p_c \simeq 1 - e^{-\lambda_c t_C} \tag{3.12}$$

where t_C is found by solving

$$\sum_{j=1}^{G}(1 - e^{-\lambda_j t_C}) = C \tag{3.13}$$

In our model (see Fig. 3.2), the CS of a virtualized CCN router is partitioned into M segments and each segment runs independently. Hence, the cache hit rate for each item of content can be obtained by applying the approximation [35] to each segment.

Thus, the cache hit rate p_c^n of content c on VCCN slice S^n is given by

$$p_c^n \simeq 1 - e^{-\Lambda_c^m t_c^m} \tag{3.14}$$

where t_c^m can be calculated by solving

$$\sum_{j=1,j\neq c}^{G} (1 - e^{-\sum_{R^i\in\Omega(m)} \lambda_j^i t_j^m}) = C(m) \tag{3.15}$$

These equations can also be simplified to

$$p_c^n \simeq 1 - e^{-\Lambda_c^m t^m} \tag{3.16}$$

where t^m is found by solving

$$\sum_{j=1}^{G}(1 - e^{-\sum_{R^i\in\Omega(m)} \lambda_j^i t^m}) = C(m) \tag{3.17}$$

From the above, the cache hit rate p^n of each VCCN router instance $R^n (1 \leq n \leq N)$ and the aggregated cache hit rate p of the virtualized CCN router are obtained:

$$p^n = \frac{\sum_{c=1}^{G} \lambda_c^n (1 - e^{-\Lambda_c^m t_c^m})}{\sum_{c=1}^{G} \lambda_c^n} \tag{3.18}$$

$$p = \frac{\sum_{n=1}^{N} \sum_{c=1}^{G} \lambda_c^n (1 - e^{-\Lambda_c^m t_c^m})}{\sum_{n=1}^{N} \sum_{c=1}^{G} \lambda_c^n} \tag{3.19}$$

If (3.12) and (3.13) are used, t_c^m is replaced with t^m.

3.5 Numerical Example

3.5.1 Validation of the Model

First, the author validated our model by comparing the analytic results of our model with the simulation results. In the determination of the cache hit rate using the approximation method, the author used (3.16) and (3.17). The network tested is shown in Fig. 3.4. Three ($N = 3$) VCCN slices S^1, S^2 and S^3 are constructed on a content-centric network so that the author can study three CS allocation methods (i.e., an exclusive method, a shared method and a hybrid method). The link delay between nodes is 10[ms] irrespective for all slices. In each slice, 10,000 content items ($L = 1$[Mbyte]) are stored in the repository (i.e., $G = 30,000$). Users generate content requests for each VCCN router instance according to a Poisson process of intensity $\lambda = 5$[req/s]. The distribution of content popularity is Zipf with parameter α_n for VCCN slice S^n. The content request ratios in all VCCN slices are equal. Here, α_1, α_2 and α_3 are set for 0.5, 0.75 and 1.5, respectively, by referring to the values of α on existing services [12,19]. Content requests in VCCN slices do not overlap; this is similar to the situation considered in [16, 19]. The author considers five CS allocation methods: an exclusive method that assigns an equal number of CS segments to each VCCN router instance, a shared method that assigns the entire CS to all VCCN router instances and hybrid methods (hybrid($S^n (1 \leq n \leq 3)$) that assign one-third of the CS segments to VCCN router instance R^n and assign the remaining segments to the others.

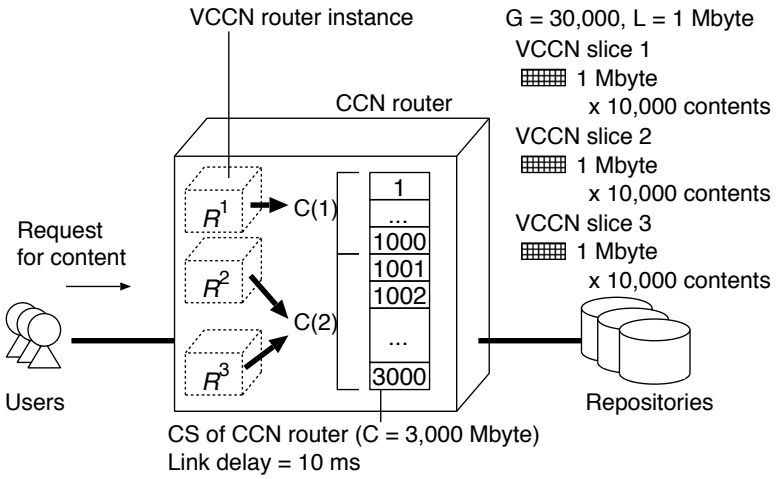
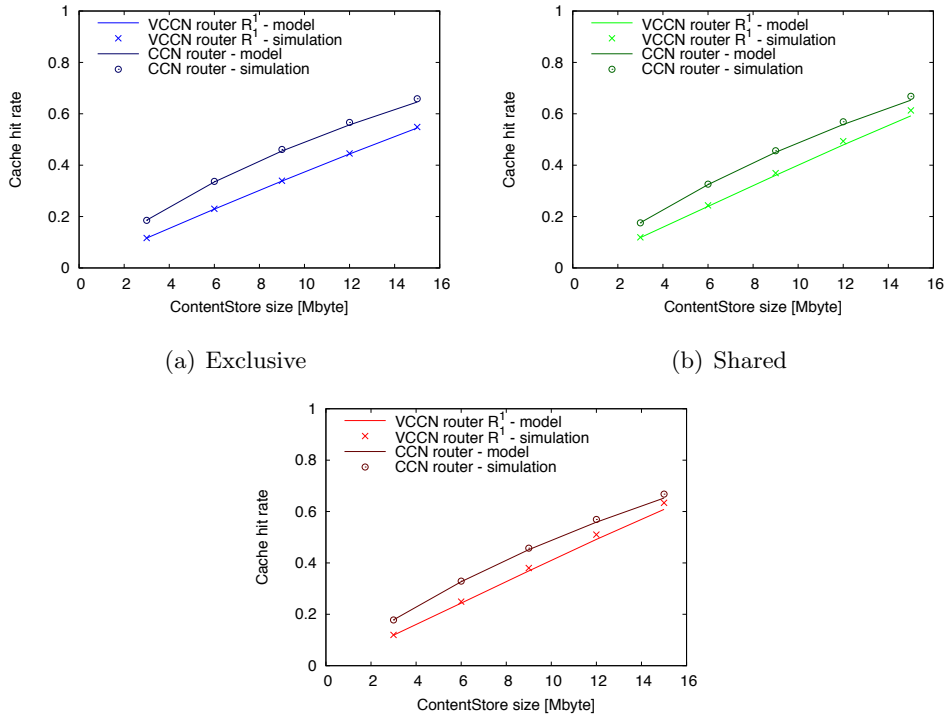


Figure 3.4: Network used for the evaluation.

The cache hit rates of VCCN router instance R^1 and the aggregated cache hit rates of the virtualized CCN router against the size of the CS are shown in Figs. 3.5 and 3.6. Fig. 3.5 shows the results based on the Markov chain model and the simulation results for the exclusive method, the shared method and the hybrid method. In Fig. 3.5, for convenience of computational complexity, G was set 30. The author confirmed that the tendency of content caching is not very different by this reduction of the number of contents. Fig. 3.6 shows the results from the approximate analysis and the simulation results for the exclusive method, the shared method and the hybrid method.

The differences between the analysis results and the simulation results are small in terms of both the VCCN slice's performance and the overall network performance, with a maximum error less than 2%. The results for the other VCCN router instances are similar but are omitted to save space. Moreover, it can be seen from Figs. 3.5 and 3.6 that the approximate analysis is as accurate as the Markov chain based analysis.

In all the following results, the author converts the cache hit rate into the average content delivery time so that it is easier to understand the direct impact on users. The author defines the average content delivery time between the users and the virtualized CCN router as τ_1 and that between the virtualized CCN router



(a) Exclusive

(b) Shared

(c) Hybrid(S^2)

Figure 3.5: The cache hit rate against the size of CS for the exclusive method, the shared method and a hybrid method (Markov chain based analysis).

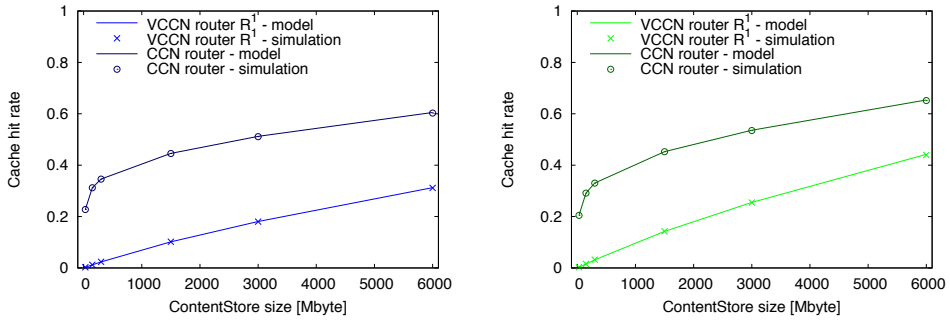
and the repositories as τ_2. The average content delivery times τ_1 and τ_2 include both a transmission delay and a processing delay. The average content delivery time D of the entire network is given by

$$D = \frac{\sum_{n=1}^{N} \sum_{c=1}^{G} 2\lambda_c^n (\tau_1 + \tau_2(1 - p_c^n))}{\sum_{n=1}^{N} \sum_{c=1}^{G} \lambda_c^n} \tag{3.20}$$

The analytic solutions for the average content delivery time D of the entire network are also highly accurate, with a maximum error less than 2%.

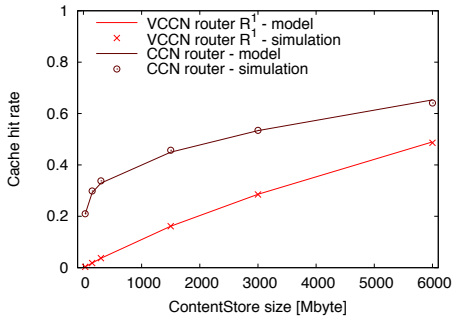
3.5.2 Effects of Content Popularity Slopes in VCCN Slices

Second, the author investigated the effects of content popularity slopes in VCCN slices on fairness and overall network performance. Figure 3.7 shows the average content delivery time of the entire network and the fairness index [38] for VCCN



(a) Exclusive

(b) Shared



(c) Hybrid(S^2)

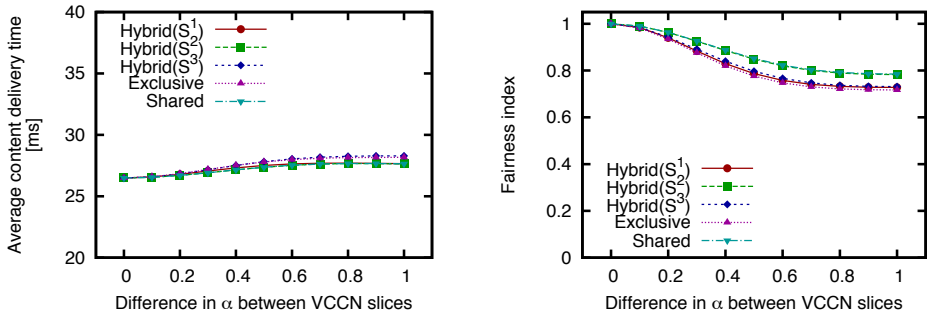
Figure 3.6: The cache hit rate against the size of CS for the exclusive method, the shared method and a hybrid method (approximate analysis).

slices against the difference of Zipf exponents α between VCCN slices. Specifically, the figure shows the result of approximate analysis when $\alpha_1 = 1 - d, \alpha_2 = 1$ and $\alpha_3 = 1 + d$ for $(0 \leq d \leq 1)$.

Jain's fairness index $f(x)$, which quantitatively measures the equality of user allocation between users i $(1 \leq i \leq N)$, is given by

$$f(x) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{3.21}$$

where x_i is an allocation metrics received by the i^{th} user [38]. The fairness index is bounded between 0 and 1. If the fairness index is 1, all users get the same amount (i.e., x_i's are all equal) and the resource allocation is 100% fair. In contrast, as the disparity increases, fairness decreases and only a few users are favored. For instance, if the fairness index is 0.2, the resource allocation is only 20% fair



(a) Average content delivery time of the entire network

(b) Fairness index for VCCN slices

Figure 3.7: Average content delivery time and fairness index against the difference of Zipf exponent α between VCCN slices, for each allocation method.

(i.e., 80% of users are not favored by the resource allocation). In this chapter, i corresponds to S^i and x_i corresponds to the cache hit rate p^i of each VCCN router instance R^i. The author regards fairness for VCCN slices as equality between average content delivery times for VCCN slices. However, if the average content delivery time is directly used as an allocation metrics, the standard of the fairness index is raised since 20[ms] is spent independently of however CS is allocated to VCCN router instances in this scenario. Hence, the author used the cache hit rate for each VCCN router instance, which ranges from 0 to 1, as an allocation metrics. Normally, the cache hit rate for each VCCN router instance is not the metrics directly reflecting the performance of each VCCN slice. However, the author configures this evaluation scenario so that the cache hit rate for VCCN routers instance are directly proportional to the average content delivery times for the VCCN slices. Hence, the fairness index in this scenario indicates how many slices are unfairly allocated the CS of the virtualized CCN router to in VCCN slice service from a network provider. In this scenario ($N = 3$), if the fairness index is less than 2/3, one of slices becomes certainly unfair and the VCCN slice service will be critically damaged.

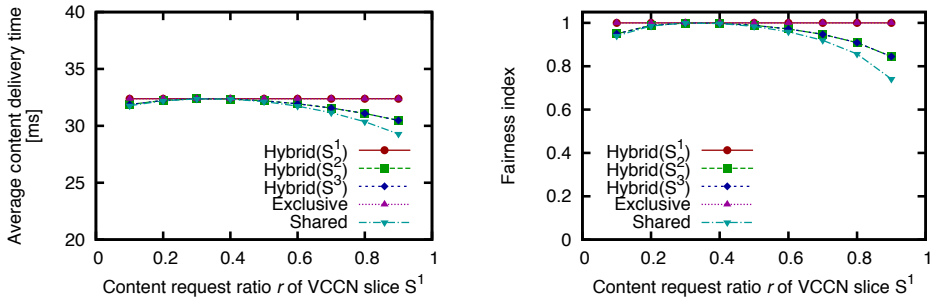
Figure 3.7 shows that the overall network performance hardly depends on CS allocation methods and that the shared method is preferable for improving fair-

ness, on a content-centric network in which there are slices with different content popularity slopes. As the difference between the content popularity slopes increases, the size of the CS that S^3 (with high α) requires to achieve a high cache hit rate decreases and the size of the CS that S^1 (with low α) requires increases. Hence, as the difference between the content popularity slopes increases, an exclusive method cannot keep up with the change of CS size required for each VCCN router instance and the fairness index is degraded. Since the fairness index for each allocation method is bounded after d is 0.8, one of slices will not critically become unfair. However, at the least, this result indicates that when the difference of α between any slices is 0.4 or more, the simple exclusive method is most unsuitable in terms of both fairness and overall network performance.

3.5.3 Effects of the Content Request Ratio for each VCCN Slice

Third, the author investigated the effects of content request ratios in VCCN slices on fairness and overall network performance. Figure 3.8 shows the average content delivery time for the entire network and the fairness index against the ratio $r(0 \leq r \leq 1)$ of content requests for VCCN slice S^1 compared with all requests. Specifically, the figure shows the results of the approximate analysis when the content request ratios of VCCN slices S^1, S^2, S^3 are respectively r, $(1-r)/2$ and $(1-r)/2$, and $\alpha = 0.75$ for all the VCCN slices.

Figure 3.8 shows that, on a content-centric network in which slices have very different content request ratios, while a shared method is preferable for maximizing the performance of the entire network, an exclusive method is preferable for improving fairness. This result is consistent with existing studies [16–18]. When the content request ratio of VCCN slice S^1 is higher than that of the other slices (i.e., $r \geq 0.4$), as r increases, the average content delivery time for the entire network decreases for the shared method and the fairness index increases for the exclusive method. Although the fairness index for the shared method is larger than 2/3 at $r = 0.9$, a shared method will get one of slices unfair when content request ratio of S^1 is more than ten times those of the other slices. On the other hand, Fig. 3.8 also shows that the hybrid methods (hybrid(S^2) and hybrid(S^3)), which



(a) Average content delivery time for the entire network

(b) Fairness index for VCCN slices

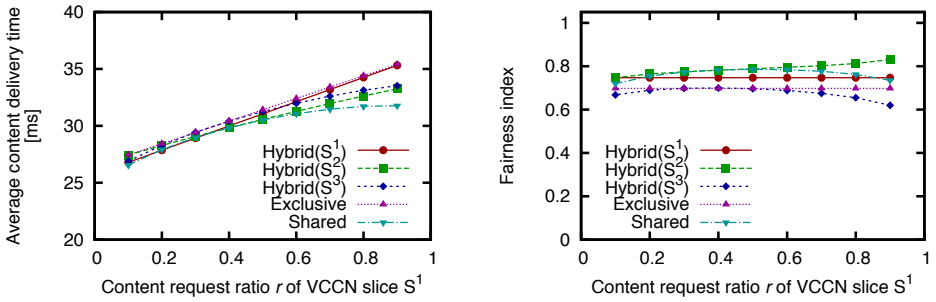
Figure 3.8: Average content delivery time and fairness index against the ratio r of content requests for VCCN slice S^1 compared to all requests for each allocation method.

assign a part of the CS to slices other than S^1, achieve moderate performance in terms of both metrics. At the least, this result indicates that when the content request ratio of a specific slice is higher than that of the other slices, a hybrid method which assigns a part of the CS to a slice with low content request ratio, is preferable for providing a balance between the two metrics. From the above results the author concludes that, when content request ratios in VCCN slices are different, the hybrid method is best suited for providing a balance between fairness and overall network performance.

3.5.4 Effects of Content Request Patterns in VCCN Slices

Finally, the author investigated the combined effects of content popularity slope and content request ratio in each VCCN slice on fairness and overall network performance. Figure 3.9 shows the average content delivery time for the entire network and the fairness index against content request ratio $r(0 \leq r \leq 1)$ for VCCN slice S^1. The figure shows the results of the approximate analysis when α_1, α_2 and α_3 are 0.5, 0.75 and 1.5, respectively.

Figure 3.9 shows that, on a content-centric network in which there are slices with widely different content popularity slopes and content request ratios the

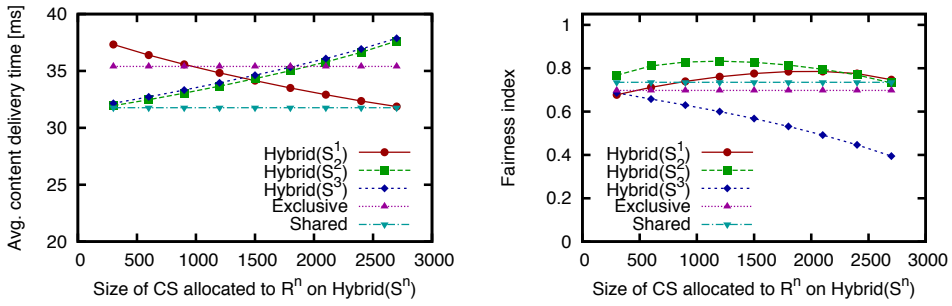


(a) Average content delivery time of the entire network

(b) Fairness index for VCCN slices

Figure 3.9: Average content delivery time and fairness index against content request ratio r of VCCN slice S^1 for each allocation method when there is a difference of Zipf exponent α between VCCN slices.

shared method is preferable for maximizing the performance of the entire network, while a hybrid method (hybrid(S^2)) is preferable for improving fairness. As the content request ratio of S^1 (which has low α) increases, the size of the CS required by R^1 is much larger than when only the content request ratio of S^1 differs. Hence, the fairness index for the shared method is degraded due to the occupancy of the CS by R^1, and that for the exclusive method is degraded for the reason given in Section 3.5.2. Since the fairness index for a shared method steadily decreases according to r and that for an exclusive method is near 2/3, these methods cannot maintain the fairness for slices. In this case, in order to improve fairness, R^1 and R^3, which require a small CS to achieve a high cache hit rate, should jointly use a part of CS, and R^2 should monopolize the remaining CS. At the least, this result indicates that when the difference of α between any slices is approximately 1.0 and the content request ratio of a slice with low α is higher than those of the other slices (i.e., $r \geq 0.4$), the hybrid method which assigns a part of CS to a slice with a low content popularity slope and low content request ratio, is preferable for improving fairness. From the above results, when both content popularity slopes and content request ratios in VCCN slices are different, the hybrid method is best suited for providing a balance between fairness and overall network performance.



(a) Average content delivery time for the entire network

(b) Fairness index for VCCN slices

Figure 3.10: Average content delivery time and fairness index against the size of CS allocated to R^n by hybrid(S^n).

For the hybrid methods, the author also investigated the effect of the size of the CS assigned to a VCCN slice on fairness and overall network performance. The author used three CS allocation methods: hybrid($S^n (1 \leq n \leq 3)$) that assign a CS segment whose size is V to VCCN router instance R^n and assign the remaining CS to the others. Figure 3.10 shows the average content delivery time for the entire network and the fairness index against V. Specifically, the figure shows the results of the approximate analysis when r, α_1, α_2 and α_3 are 0.9, 0.5, 0.75 and 1.5, respectively.

Figure 3.10 shows that several hybrid methods perform well over a wide range of V. If $V < 2000$ on hybrid(S^2) or $1000 < V$ on hybrid(S^1), the hybrid methods achieve higher performance than the exclusive method on both metrics. In particular, hybrid(S^2), which assigns a part of the CS to a slice with low content popularity slope and low content request ratio, can most efficiently balance both metrics when the part of the CS allocated to a slice is approximately one third (in general, $1/N$) of the whole CS. From this result, the author anticipates that even if the method used to allocate the CS to VCCN slices is very simple, a hybrid method will be effective in terms of both fairness and overall network performance.

Generally speaking, content request patterns in VCCN slices (i.e., the content popularity slope and content request ratio for each VCCN slice) will be different.

From the above results, the method for allocating CS resources to VCCN routers should be selected as follows. When the difference between content request patterns in VCCN slices is large, a hybrid method that assigns a part of the CS to a slice with low content popularity slope and low content request ratio will provide a balance between the two metrics. On the other hand, whenever fairness between VCCN slices is not important for any content request patterns in the VCCN slices, the shared method will maximize the performance of the entire network.

3.6 Summary

In this chapter, the author has analytically and quantitatively investigated the effects of CS allocation methods and content request patterns in VCCN slices on network fairness for VCCN slices and the overall network performance. The author developed a mathematical model of virtualized CCN router for cache performance analysis under arbitrary content request patterns, and derived the cache hit rate for each VCCN router instance and the aggregated cache hit rate of the virtualized CCN router. Furthermore, using several numerical examples, the author has shown that when content request patterns are heterogeneous, a hybrid resource allocation method will provide the best balance between fairness and overall network performance.

In the future the author will extend our model to investigate the effect of network topology in a network composed of multiple virtualized CCN routers. The author will also develop a dynamic CS allocation method, which operates efficiently in a distributed environment in which both the content popularity slope and the content request ratio of each VCCN slice are dynamically varying.

Chapter 4

Inferring Relevant Blocks on Hyperlinked Web Page based on Block-to-Block Similarity

4.1 Introduction

Since significant effort is spent in collecting information from the Web, improving the efficiency of Web browsing is one of the most important challenges in enhancing our daily activities. Many people use the Internet mainly for browsing Web pages [39]. It has been reported that on average people spend 103 minutes per day accessing the Internet, whereas they spend just 28 minutes per day reading newspapers [40].

Lazonder *et al.* [15] showed that even users experienced at Web browsing spend almost the same amount of time as novices to locate sought-after information on specific Web sites (i.e., to browse related Web pages one-by-one simply by following hyperlinks). Conversely, Lazonder *et al.* also showed that the experienced users take on average one-third of the time that novices do to locate Web sites containing sought-after information using search engines. In Web browsing, both experienced and novice users must look through all the contents of a destination Web page to determine whether it contains the sought-after information, resulting in the

comparable Web browsing performance between them [15]. Therefore, a key factor in improving the efficiency of Web browsing is rapid determination of whether the destination Web page contains sought-after information after the user selects a hyperlink.

Because most hyperlinks on the Web point to a page itself, rather than a part of the page, users encounter difficulty in rapidly determining whether the Web page contains relevant information. Although a hyperlink can point to a specific HTML tag in a destination Web page by means of a fragment identifier appended to a URL (e.g., `#article`) [41], the majority of hyperlinks are implemented without a fragment identifier. Hence, a user selecting a hyperlink must usually search through the contents of the destination Web page.

To improve the efficiency of Web browsing, several Web content filtering methods have been proposed for extracting the important parts of Web pages and for removing the unimportant parts (e.g., advertisements and navigation bars) [42–47]. Gupta *et al.* [42] proposed a Web content filtering method that removes link collections and advertisements from an HTML document represented as a document object model (DOM) tree [48]. Yi *et al.* [43] filtered noisy information (e.g., advertisements and navigation bars) from a Web page by utilizing the tendency for noisy information to be displayed at common locations on many Web pages. Pasternack *et al.* [45] developed a filtering method that extracts important parts from a Web page by using machine learning (i.e., naive Bayes local classifiers). Moreover, several publicly available services (e.g., Safari Reader [49], Capti Web Player [50], etc.) have been provided for extracting important parts (i.e., the article) from a blog page or a news page based on their Web content filtering methods.

However, many existing methods filter Web content without accounting for the user context in Web browsing (i.e., which hyperlink was selected on the previous Web page and what was the context around the selected hyperlink). When browsing the Web, a user generally visits a series of contextually related hyperlinked Web pages with the aim of finding sought-after information. Hence, the author believes that utilization of user context is a promising approach to improving the efficiency of Web browsing.

Borodin and co-workers have proposed two systems that utilize user context in Web browsing: CSurf [46] and CMo [47]. When a visually impaired user accesses a hyperlinked Web page with the aid of a screen reader, CSurf determines the starting position of the screen reader on the page from the user context. Specifically, CSurf infers the most relevant frame of the destination Web page on the basis of the similarity between the text on the source page and destination Web page. In this chapter, *frame* is defined as "the largest of the consistent frames on the path from a leaf to the root of a frame tree" [46], which corresponds to *block* in CSurf and CMo. CSurf segments the destination Web page into frames (e.g., header, footer, and side bar), and infers which is the most relevant based on the similarity between the text around the selected hyperlink and the text in each frame. Similarly, when a user accesses a Web page on a mobile device with a small screen (e.g., a PDA or smart phone), CMo utilizes user context to determine the starting point for rendering the page on the screen. As with CSurf, CMo determines the most relevant frame of the page.

As compared with CSurf and CMo, the author apply a similar approach to inferring the relevance of blocks, which are considerably smaller than frames, on a destination Web page. In [47], the CMo's authors were also interested in improving usability by using a finer granularity of block size, and allowing a user to navigate between the blocks in the future. However, they have not been showing the performance of a fine-grained relevant block inference and developing the context browsing system yet.

In this chapter, with the aim of improving Web browsing efficiency, the author proposes a method called *HypErlink Referring Block estimation (HERB)*, which infers the existence and location of all relevant contents on destination Web pages. HERB segments Web pages into blocks and then utilizes the hyperlink selected by the user and the context around the hyperlink to infer the blocks relevant to the hyperlink on the basis of a similarity between blocks. Moreover, through experiments simulating ordinary Web browsing, the author quantitatively investigates the effectiveness of HERB in improving browsing. The author designs two HERB implementations, namely, a Web proxy and Web browser, and discusses

their advantages and disadvantages.

The main contribution of this chapter, in particular compared with CSurf and CMo, is that the author has quantitatively shown that HERB realizes a fine-grained relevant block inference. HERB is significantly based on CSurf and CMo, but the main difference is the granularity of information chunks to be extracted in Web browsing. CSurf and CMo tried to determine the starting position of Web browsing for devices, which cannot show the whole structure of a page to a user in a destination Web page. Since those devices cannot show the whole structure of a page to a user, CSurf and CMo determine the starting position of Web browsing by inferring the most relevant frame (i.e., coarse-grained information chunk, which probably contains the main content). Hence, CSurf and CMo use a coarse-grained Web page segmentation method (i.e., Geometric Clustering algorithm). On the other hand, HERB tries to identify all fine-grained relevant blocks in a destination Web page irrespective of the capability of Web browsing devices. If a user browses a Web page using a conventional Web browser, a user would like to locate not only the most relevant frame but also all relevant blocks. Hence, HERB uses a finer-grained Web page segmentation method.

In this chapter, the author intentionally uses a simple algorithm for HERB to infer relevant blocks compared with CSurf and CMo. It is because, as the authors of CSurf explained in [46], individual contributions of SVM, word-stemming, and topic detection on the performance have not been understood. The author uses a simple algorithm for HERB to clearly investigate feasibility and effectiveness of a fine-grained relevant block inference using a conventional block segmentation method and common text mining techniques.

The rest of the chapter is organized as follows. In Section 4.2, the author classifies hyperlinks in Web pages in order to clarify the conditions under which blocks relevant to a hyperlink must be inferred. In Section 4.3, the author explains the HERB method, and the author shows the experimental results in Section 4.4. The author then designs two HERB implementations in Section 4.5 and discusses their advantages and disadvantages. Furthermore, the author gives an overview of the Web proxy prototype and an example use case. Finally, the author concludes

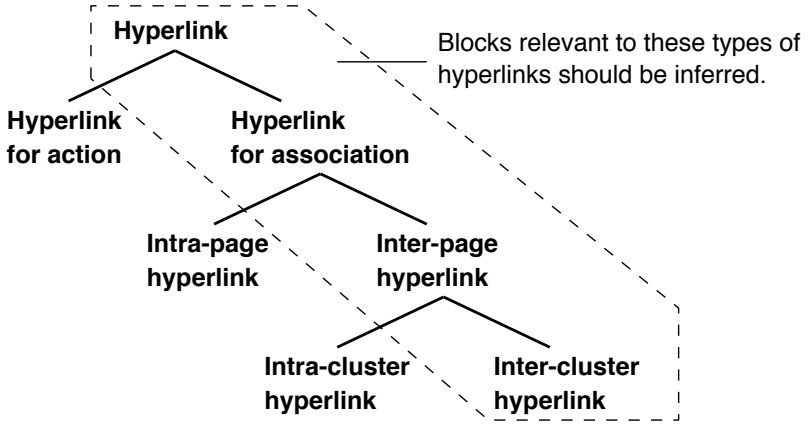


Figure 4.1: Taxonomy of hyperlinks.

this chapter and discuss future works in Section 4.6.

4.2 Taxonomy of Hyperlinks

Originally, hyperlinks were designed to relate a Web page with other Web pages [51]. However, with the advancement of Web technology, hyperlinks are now used for several purposes—to refer to a part of a Web page, to transmit information from a user in a query string, and to dynamically generate contents by using JavaScript. Hence, relevant blocks must be inferred for some hyperlinks, but not others.

Therefore, the author classifies hyperlinks in Web pages in order to clarify when blocks relevant to a hyperlink must be inferred (Figs. 4.1 and 4.2).

First, the author classifies hyperlinks based on *the direction of information* between the user and system: *hyperlinks for association*, which provide information to the user; and *hyperlinks for action*, which transmit information from the user. Note that hyperlinks for association and for action are not exclusive. A hyperlink can simultaneously serve for both association and action. Examples of hyperlinks for association include links to related content and to blogs or news articles. Examples of hyperlinks for action include links containing a query string (e.g., `?q=keyword`) [41] and a JavaScript function.

Second, the author classifies hyperlinks into *intra-page* and *inter-page* links,

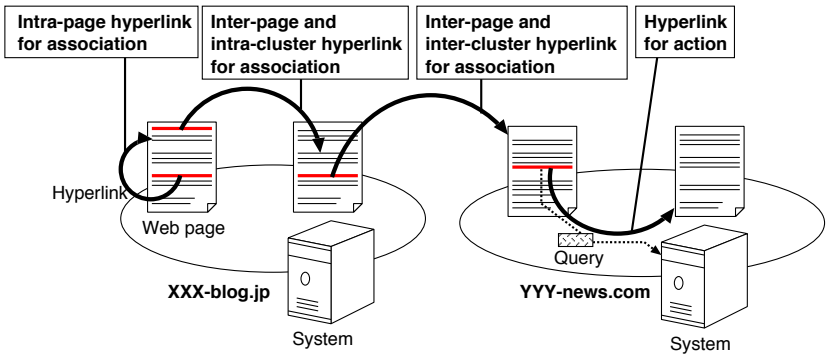


Figure 4.2: Examples of several types of hyperlinks.

based on whether the source page and destination page are the same. Inter-page links point to a destination Web page different from the source Web page (i.e., at a different URL). Intra-page links point to an HTML tag within the same Web page by means of a fragment identifier or anchor name. Intra-page links are often used for in-page navigation; following an intra-page link avoids cumbersome scrolling in the browser when the Web page is lengthy.

Third, the author classifies inter-page links into *intra-cluster* and *inter-cluster* links according to the credentials of the source and destination Web pages. The author defines a cluster of Web pages as a set of pages whose credentials are identical. For instance, Web pages owned, written, published, or copyrighted by the same entity can be treated as a cluster. Thus, intra-cluster links point to a Web page that has the same credentials. Navigation links (e.g., "next page" and "previous page" links) are examples of intra-cluster links used to navigate through a set of Web pages authored by the same entity. In contrast, inter-cluster links point to Web pages whose credentials are different from those of the source Web page.

Under this classification scheme, hyperlinks for which relevant blocks may need to be inferred in Web browsing are *for association, inter-page, and inter-cluster* links. Since hyperlinks for action transmit information to the system, they do not usually require identification of relevant blocks. Furthermore, intra-page links are used to navigate within a single Web page, and also are not typically

required to infer relevant blocks. For intra-page and intra-cluster hyperlinks, a user should be able to easily locate necessary information, because Web pages with the same credentials generally have high similarity in design or layout. As a result, the author assumes that *for association, inter-page, and inter-cluster* hyperlinks require the identification of relevant blocks during Web browsing.

4.3 HypErlink Referring Block estimation (HERB)

4.3.1 Overview

HERB infers blocks relevant to a hyperlink by utilizing the user context in Web browsing, in particular, the text around the selected hyperlink on the source Web page.

The main capabilities of HERB are (1) Web page segmentation, (2) feature terms extraction from each block, and (3) block-to-block similarity calculation (Fig. 4.3).

First, HERB segments the source and destination Web pages into blocks based on their structural and functional organization. To do so, HERB employs an existing Web page segmentation method.

HERB then extracts feature terms from the text in each block, and builds a feature vector composed of each term's weight based on its frequency in a block and a corpus. Explicitly, HERB calculates the term frequency-inverse document frequency (TF-IDF) [52] of each term and assigns these values as the weight elements of the feature vector.

Finally, HERB measures the similarity between the block containing the selected hyperlink and each block in the destination Web page by calculating the cosine similarity (i.e., the dot product) [53] between the feature vectors of the blocks. HERB infers that blocks with high block-to-block similarity are relevant.

4.3.2 Web Page Segmentation

HERB segments source and destination Web pages into blocks by using an existing Web page segmentation method. A block is a portion of the Web page and is

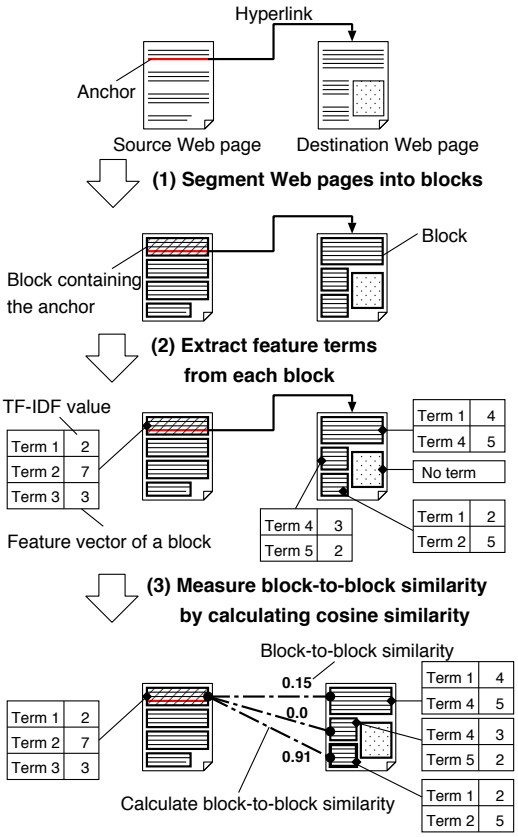


Figure 4.3: Overview of HERB.

obtained by segmenting the page based on its structural and functional organization. Let $c_1, \ldots, c_N$ be elements (HTML tags or text) in the HTML document of a Web page P, and let $B_k = \{c_i, \ldots, c_j | 1 \leq i, j \leq N\}$ $(1 \leq k \leq M)$ be the blocks obtained by segmenting P.

HERB can use any existing Web page segmentation method in order to achieve block segmentation [54–62]. For instance, Yu *et al.* [54, 55] proposed Web page segmentation based not only on the structure of an HTML document but also on spatial and visual cues (e.g., margin, font size, and background color). Lee *et al.* [56] developed a method called PARCELS that segments a Web page into blocks based on the structural blocks of HTML documents and classifies these elements (e.g., as advertisements, headlines, and navigation elements). Baluja [57] proposed

a 3 × 3 Web page segmentation method, in which blocks are associated with the numeric keypad on a mobile telephone. For mobile terminals, Hattori *et al.* [58] created a Web page segmentation method robust to common HTML syntax errors.

4.3.3 Feature Term Extraction from Each Block

HERB extracts feature terms from the text in block B^s containing the selected hyperlink, and blocks $B_k^d (1 \leq k \leq M)$ in the destination Web page. HERB then generates feature vectors corresponding to each of these blocks according to the frequency of each term.

HERB removes all HTML tags from block $B = \{c_i, \ldots, c_j\}$ and obtains a unique set of terms from the text in B. When text in the HTML document is separated by whitespace (e.g., as in English text), HERB divides the text into terms at whitespaces. If the text is not separated by whitespace (e.g., as in Japanese text), HERB performs a morphological analysis to extract terms. In this chapter, the author simply uses all terms as feature terms.

HERB builds feature vector $\mathbf{v}_B$ whose elements are the TF-IDF weights [52] of terms t_i $(1 \leq i \leq L)$ in B. Given t_i contained in corpus D for calculating TF-IDF weights, $\mathbf{v}_B$ is

$$\mathbf{v}_B = (\, TF(t_i, B)\ \ IDF(t_i)) \,. \tag{4.1}$$

Here $TF(t_i, B)$ is the frequency of t_i in B (i.e., the term frequency) [63] and is defined as

$$TF(t_i, B) = \frac{n_{i,B}}{\sum_{j=1}^{L} n_{j,B}}, \tag{4.2}$$

where $n_{i,B}$ is the number of times t_i appears in B. Moreover, $IDF(t_i)$ is the inverse document frequency of t_i in D [63],

$$IDF(t_i) = \log \frac{1 + |D|}{|D_{t_i}|}, \tag{4.3}$$

where $|D|$ is the number of documents contained in D, and $|D_{t_i}|$ is the number of documents that contain t_i in D.

4.3.4 Block-to-block Similarity Calculation

HERB measures the block-to-block similarity between B^s and $B_k^d (1 \leq k \leq M)$ by calculating the cosine similarity [53] between the blocks, which has been widely used in the field of information retrieval (IR). The cosine similarity $S(B_i, B_j)$ between B_i and B_j is defined as

$$S(B_i, B_j) = \frac{\mathbf{v}_{B_i} \cdot \mathbf{v}_{B_j}}{|\mathbf{v}_{B_i}| \, |\mathbf{v}_{B_j}|}. \tag{4.4}$$

HERB thus infers B_k^d as a relevant block if $S(B^s, B_k^d)$ is high.

4.4 Experiments

4.4.1 Experimental Methods

The author quantitatively investigates the effectiveness of HERB by comparing the subjective judgment of human assessors with the inference results of HERB. To the best of our knowledge, it has not been investigated how accurately and comprehensively blocks relevant with respect to the context around a hyperlink can be extracted from the destination Web page by the similarity between blocks. On the basis of the widely used evaluation methodology for Web search systems in the Text REtrieval Conference (TREC) Web Track Guidelines [64], the author analyzed the correspondence between relevance scores subjectively given by assessors and the similarity scores assigned by HERB.

To examine the fundamental characteristics of HERB, the author used a simple threshold-based algorithm for Web page segmentation. Specifically, the algorithm translates the HTML document of a Web page into a DOM tree [48], and splits this tree into non-overlapping subtrees such that the text length of subtree, L, lies between two parameters for controlling the size of blocks, T_{min} and T_{max}. Here, the text is a string interleaved with block elements, such as `p`, `div`, and `table` tags [65]. With existing Web page segmentation methods such as PARCLES [56], the mean and distribution of the block size vary according to the parameter settings and the structure of the Web page; however, our simple algorithm can control the block

size, which significantly simplifies the HERB experiments. In all experiments, $T_{min} = 200$ [character] and $T_{max} = 400$ [character] are used.

The author used 661 block pairs created from 32 Japanese Web pages randomly extracted from popular entries on a social bookmarking service between December 16, 2010 and January 13, 2011 [66]. These 32 pages were used as source Web pages, and one hyperlink contained in each source page was randomly selected to obtain the destination Web pages. To avoid using meaningless hyperlinks, the author excluded hyperlinks other than for-association, inter-page, and inter-cluster links (see Section 4.2). Specifically, a hyperlink is identified as for-association if the URL of the hyperlink refers to an HTML file and does not include a query string. A hyperlink is identified as inter-page if the URL of the hyperlink is different from that of the source Web page. A hyperlink is identified as inter-cluster if the host name of the hyperlink is different from that of the source Web page and the file path of the hyperlink is not a default file path. In addition, hyperlinks that refer to Web pages with less than 10 blocks were excluded because they are too short to infer relevant blocks.

The author also extracted all Web pages (9,313) from popular entries during the period December 15, 2009, to December 15, 2010, on the same social bookmarking service [66]. These pages were used as the corpus D (see Section 4.3.3) for calculating TF-IDF weights. Specifically, for each Web page contained in D, the author divided text interleaved with `body` tags into terms by using a Japanese morphological analyzer, MeCab [67]. The author thus utilized both the terms and the number of Web pages that contain each term in D.

Five graduate students in our laboratory individually labeled every block in each destination Web page with a binary value based on the block's relevance. Blocks judged as relevant to the context around the hyperlink in the source Web page were labeled 1, while the rest were labeled 0. In what follows, the author uses the *relevance score*, the total of the five scores as labeled by the assessors. Hence, each relevance score is an integer from 0 to 5. The five assessors were also requested to choose the most relevant block in the destination Web pages.

The numbers of blocks with relevance scores of 0, 1, 2, 3, 4, and 5 were 204,

105, 71, 100, 91, and 90, respectively. Moreover, among 661 block pairs, 78 blocks were judged as being the most relevant by at least one of the assessors.

4.4.2 Results: Evaluation using Relevance Scores

First, the author performs an analysis based on the relevance scores. The relation between relevance scores and block-to-block similarity scores assigned by HERB is shown in Fig. 4.4. Each histogram shows the distribution of block-to-block similarity scores for the set of blocks with the same relevance score, and indicates that a high block-to-block similarity was given to blocks that were subjectively judged as being *relevant*. Note that hardly any blocks with high relevance scores of 4 or 5 have low block-to-block similarity assigned by HERB. In other words, HERB has a low false negative ratio; a high block-to-block similarity may be incorrectly assigned to irrelevant blocks, but a low block-to-block similarity is unlikely to be given to a relevant block. In IR, realizing a low false negative ratio is vital [68] for exhaustive searches. Hence, these results indicate the effectiveness of HERB at least for exhaustive Web searches.

Box plots highlighting the relation between relevance scores and block-to-block similarity scores are shown in Fig. 4.5. Boxes indicate the value ranges between the first quartile (25th percentile) and third quartile (75th percentile), where the line within each box denotes the median value (50th percentile). Furthermore, the ends of the whiskers show the lowest value within the 1.5 interquartile range [69] (IQR) of the first quartile and the highest value within the 1.5 IQR of the third quartile. The dotted line in the figure is a regression line, and the correlation coefficient between relevance scores and block-to-block similarity scores is 0.39 ($R^2 = 0.16$). Hence, a moderate correlation exists between the relevance score and block-to-block similarity.

The author now investigates how accurately relevant information can be extracted from the destination Web page by using a block-to-block similarity threshold. Recall that block-to-block similarity scores assigned by HERB range between 0 and 1. If a large threshold is used, extracted blocks will have a high probability of being relevant, but many other relevant blocks might not be extracted. In con-

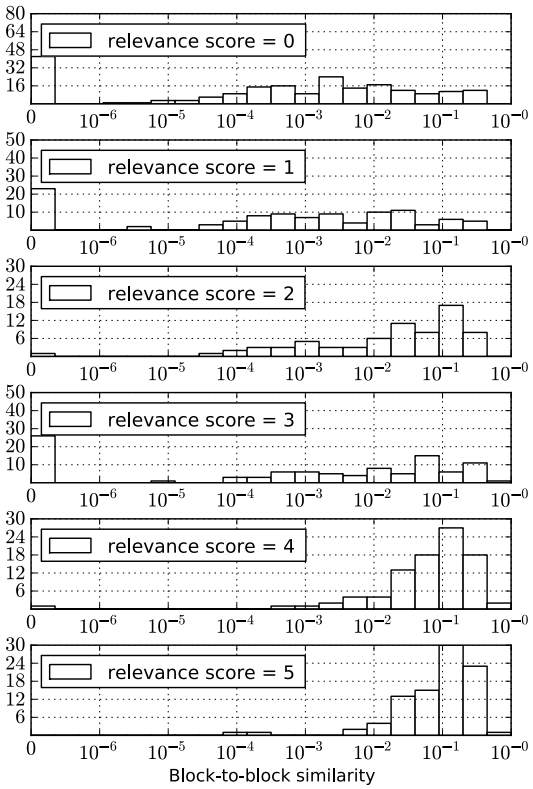


Figure 4.4: Histograms showing the distribution of block-to-block similarity scores assigned by HERB for sets of blocks with the same relevance score.

trast, if a small threshold is used, most relevant blocks will be extracted, but many non-relevant blocks might also be extracted. The author therefore determines the precision [63] and recall [63] of block extraction for a given block-to-block similarity threshold T_h. Here, precision and recall respectively measure how accurately and how comprehensively relevant blocks are extracted from the destination Web page.

Precision and recall values for different T_h values are shown in Figs. 4.6(a) and 4.6(b), respectively. The plots contain five lines, each corresponding to a value r such that when blocks have relevant scores greater than or equal to r they

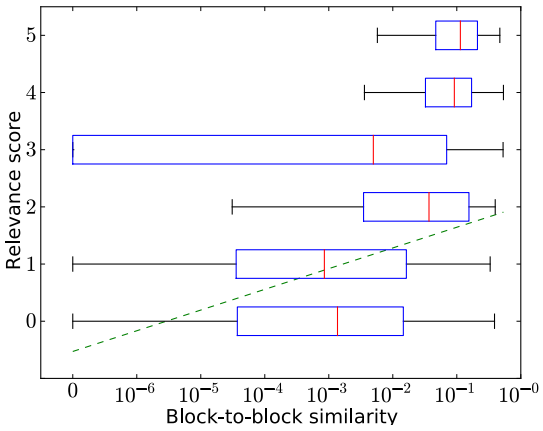
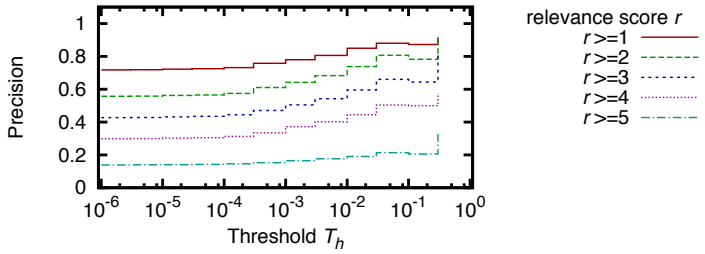


Figure 4.5: Box plots showing the relation between relevance scores and block-to-block similarity scores assigned by HERB. The dotted line is the regression line.

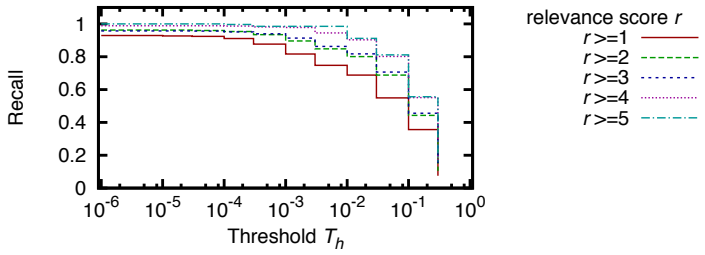
are considered to be relevant. The author also measured the maximum F-measure, which is the maximum of harmonic means of precision and recall, with varying threshold T_h for extracting blocks with relevance scores greater than or equal to r. Maximum F-measures for $r = 1, 2, 3, 4, 5$ were 0.81, 0.77, 0.69, 0.62 and 0.33, respectively.

For instance, Fig. 4.6(a) indicates that 65% of blocks extracted at $T_h \geq 0.03$ are relevant (i.e., they have a relevance score greater than or equal to 3). Furthermore, Fig. 4.6(b) shows that, in this case, 70% of the relevant blocks in the destination Web page are extracted.

With 266 (approximately 43%) of all blocks having relevance scores greater than or equal to 3, if p% of all blocks are extracted randomly from a destination Web page, the precision will be approximately 43% and the recall will be approximately p%. For instance, with $T_h = 0.03$, the precision (65%) is approximately 1.5-fold of that using random extraction (i.e., 43%) when the same number of blocks with HERB are randomly extracted, and the recall (70%) is approximately 1.8-fold of that with the random extraction (i.e., 40%). These precision and recall are as high or higher than those of existing method on block-based Web



(a) Precision



(b) Recall

Figure 4.6: Precision and recall for a given T_h. Precision and recall measure how accurately and comprehensively relevant blocks can be extracted from the destination Web page, respectively. Lines in the plots correspond to cases where blocks with relevance scores greater than or equal to r are considered relevant.

search [54, 70, 71]. Hence, these results indicate that inference of relevant blocks by HERB will assist a user to search through relevant contents of destination Web pages.

However, choosing an appropriate value of T_h is important for extracting blocks by means of HERB. Since precision and recall have a trade-off relation, the author must take account of this balance when determining the optimal block-to-block similarity threshold. Figures 4.6(a) and 4.6(b) suggest that T_h should be set between 10^{-3} and 10^{-1}; however, establishing the optimal value of T_h, which will be dependent on factors such as characteristics of the target Web page and the employed Web page segmentation algorithm, is beyond the scope of this chapter.

4.4.3 Results: Evaluation using Importance Scores

Next, the author performs an analysis based on importance score, which is the aggregate scores of the five assessors subjectively judging whether blocks on the destination Web pages are the most important with respect to the context of the source Web pages.

The author thus investigates the relation between the most relevant block and the block-to-block similarity scores assigned by HERB. Let I be the set of blocks that are judged to be the most relevant by any of the assessors, and let $\overline{I}$ be the complementary set of I. By this definition, there are one or more most relevant blocks for a given hyperlink. The histograms in Fig. 4.7 show the distributions of the block-to-block similarity scores assigned by HERB for I and $\overline{I}$. In the figure, the first, second, and third quartiles are denoted by dashed lines.

The results show that block-to-block similarity scores given to blocks in I are higher than similarity scores given to blocks in $\overline{I}$. Note that many blocks with block-to-block similarity scores equal to 0 are included in $\overline{I}$. The median of I is more than 10-fold that of $\overline{I}$, indicating that a block with high block-to-block similarity is judged most relevant with a high probability. Hence, when a user preferentially browses blocks with high block-to-block similarity, the author considers that the user can locate sought after information in a short time.

Finally, the author investigates how accurately the most relevant blocks, as judged by any of the assessors, are extracted from the destination Web page by using T_h. In this chapter, since there are one or more most relevant blocks for a given hyperlink, the performance of HERB can be measured by precision and recall. Precision and recall values for different T_h are given in Figs. 4.8(a) and 4.8(b), respectively. The maximum F-measure when varying threshold T_h was 0.47.

Figure 4.8(a) indicates that if blocks with $T_h \geq 0.03$ are extracted, 30% of those blocks are deemed the most relevant (i.e., they have an importance score greater than or equal to 1). In contrast, Fig. 4.8(b) shows that, in this case, 80% of the most relevant blocks in the destination Web page are extracted.

The number blocks judged as the most relevant by any of assessors was 78,

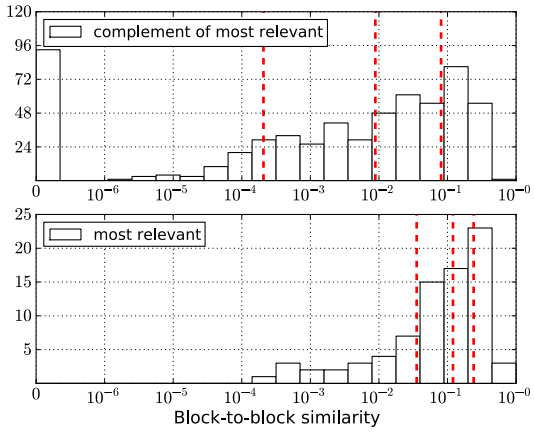
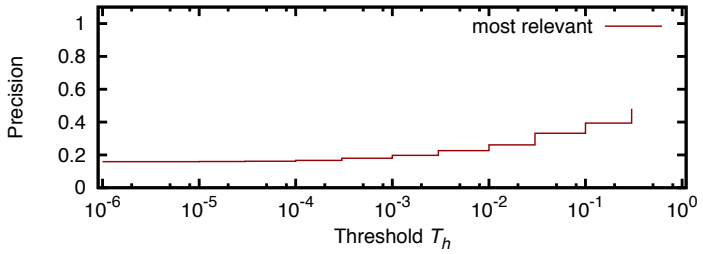


Figure 4.7: Histogram showing the distribution of block-to-block similarity scores assigned by HERB for I and $\overline{I}$.

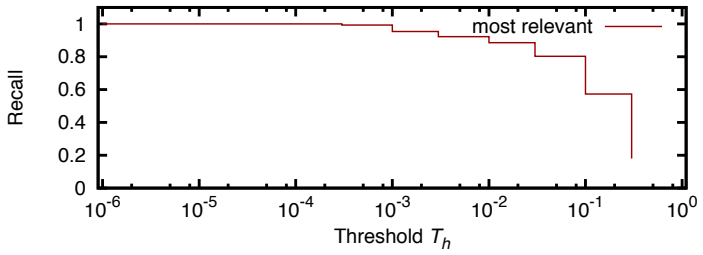
which is approximately 12% of total number of blocks. Hence, similar to Section 4.4.2, if p% of blocks are extracted randomly from a destination Web page, the precision of the blocks is approximately 12 % and the recall is approximately p%. For instance, with $T_h = 0.03$, the precision (30%) is approximately 1.5-fold of that using random extraction (i.e., 12%) when the same number of blocks with HERB are randomly extracted, and the recall (80%) is approximately 1.8-fold of that with the random extraction (i.e., 40%). These results indicate that HERB is valuable for inferring relevant blocks by extracting most relevant information with respect to the context of a source Web page.

4.5 Design and Implementation

In this section, the author outlines two HERB implementations, namely, a Web proxy and a Web browser, and discuss their advantages and disadvantages. Furthermore, the author shows an example use case for the implemented Web proxy prototype.



(a) Precision



(b) Recall

Figure 4.8: Precision, recall of the most important information for a given T_h.

4.5.1 Design of HERB-enabled Web proxy

Inferring relevant blocks by using HERB and presenting the results to a user can be realized by means of a HERB-enabled Web proxy (Fig. 4.9).

The HERB-enabled Web proxy typically fetches the source and destination Web pages based on the `Host` field (the URL of the destination Web page) and the `Referer` field (the URL of the source Web page) of an HTTP request [72]. Otherwise, the Web proxy serves the pages from its cache. The Web proxy then infers relevant blocks from these pages by means of HERB.

The HERB-enabled Web proxy transmits the content of a URL in response to an HTTP request and simultaneously presents the inferred results to the user by embedding layout and style information in the content. For example, the Web proxy can highlight relevant blocks by modifying the HTML document of the destination Web page [73], for example, embedding presentation elements (`b` tag, `strong` tag, etc.), calling a style sheet (`style` tag) [74], or using a ruled line (`hr`

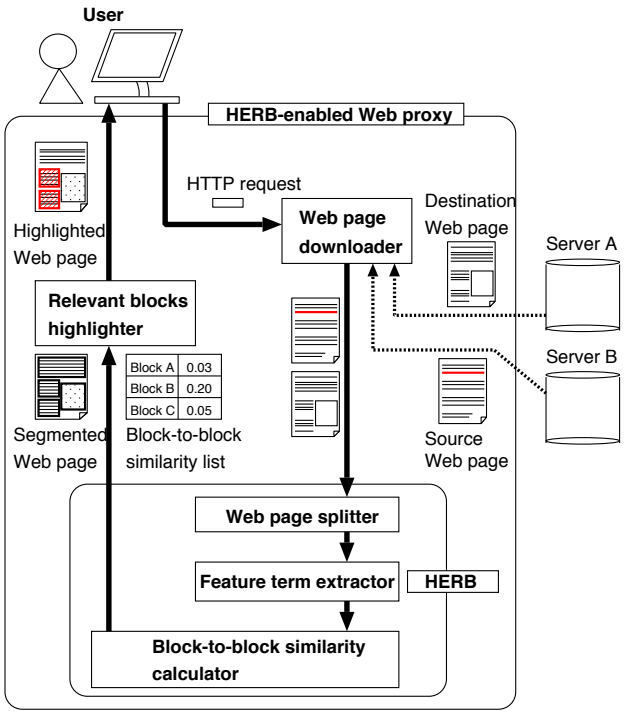


Figure 4.9: HERB-enabled Web proxy.

tag).

Advantages of using the HERB-enabled Web proxy are that the user is not required to install any software, and that the Web proxy is independent of the user environment (e.g., the Web browser and the computer processing speed). The primary disadvantage is lack of scalability; when many users access Web pages through the Web proxy, the load on the proxy increases, and can result in large processing delays.

4.5.2 Design of HERB-enabled Web browser

Inferring relevant blocks using HERB and presenting the results to a user can also be realized by means of a HERB-enabled Web browser (Fig. 4.10).

When a user follows a hyperlink, the Web browser saves the source Web page to memory, and the destination Web page is concurrently downloaded and also saved to memory. Similar to the HERB-enabled Web proxy, the Web browser then

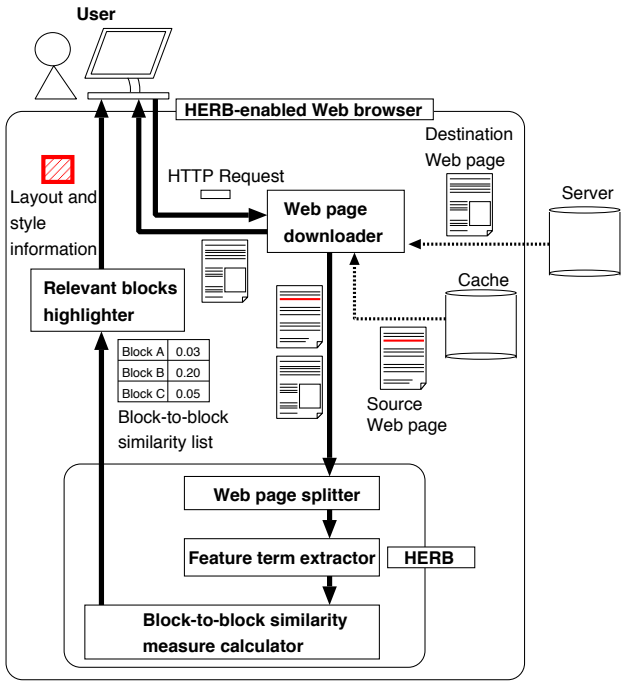


Figure 4.10: HERB-enabled Web browser.

infers relevant blocks from these pages by means of HERB.

Next, the Web browser renders the destination Web page and presents the relevant blocks to the user. For instance, relevant blocks inferred by HERB can be presented by embedding layout information in the HTML document of a destination Web page, by changing the associated style sheet [74] , or by displaying a figure or symbol indicating the relevant blocks.

Implementation of the HERB-enabled Web browser can be achieved either by modifying the browser itself or by implementing a browser plug-in.

Advantages of the HERB-enabled Web browser are that HERB parameters and the presentation method for relevant blocks can be independently set up by each user, and that processing delays are minimized because inference of relevant blocks and rendering of the destination Web page can be processed in parallel. A disadvantage is that users must install software suited to their computing environment.

4.5.3 Implementation of HERB-enabled Web proxy

An example use case of the implemented Web proxy prototype is now presented. Our Web proxy prototype assists a user to search through relevant contents of destination Web pages by highlighting relevant blocks according to block-to-block similarity measures and displaying the text in the three blocks with the highest similarity and the intra-page links to those blocks (Fig. 4.11).

Our Web proxy prototype was implemented in Python using the event-drive networking framework Twisted [75]. The system infers relevant blocks, and presents them to a user through the mechanism described in Section 4.5.1 (Fig. 4.11). To segment the Web page into blocks, the author employed both the simple threshold-based segmentation method explained in the previous section (Section 4.4.1) and PARCELS [56].

Presentation of relevant blocks inferred by HERB was accomplished by embedding layout information in the HTML document. Specifically, the prototype enclosed each block obtained by Web page segmentation within `block` tags, giving these tags one of N class attributes based on the similarity scores assigned by HERB. Furthermore, by adding a `style` tag in the header of the HTML document, the prototype called a style sheet specifying a background color attribute (`background-color`) for each class attribute of the block tags. In this way, the background of each block presented to the user is one of N colors based on the similarity scores.

The author evaluated the processing delay between receiving an HTTP request and finishing transmission of the HTML document to the user when applying our prototype system to source and destination Web pages extracted from popular entries on a social bookmarking service (see Section 4.4.1). The results are shown in Fig. 4.12. Here, HERB was implemented in Python 2.6.1 and Twisted 10.1.0 and executed on a PC running Mac OS X 10.6.7 (Darwin 10.7.0) with a 2.5 GHz Core 2 Duo processor and 4 GB of memory. The figure shows the relation between the HTML document size of the destination Web page and the processing delay of the system when using the simple threshold-based segmentation algorithm. The average processing delay of this system was approximately 1.0 s or less in all

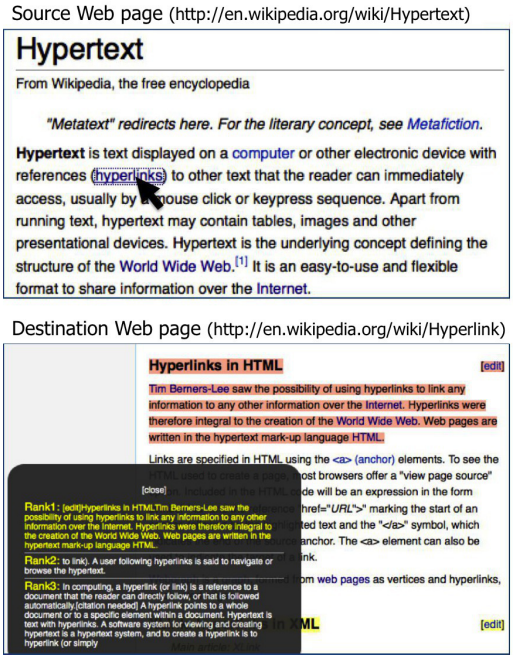


Figure 4.11: An example of using the HERB-enabled Web proxy. When a user selects a hyperlink on the source Web page, a destination Web page is served in which layout and style information is embedded for highlighting relevant blocks according to the similarity measures. Moreover, the text in the three blocks with the highest similarity to the block containing the selected hyperlink are displayed with intra-page links in order to facilitate in-page navigation.

cases. Therefore, although our prototype system is implemented in the interpreted language Python, the system still achieves sufficiently low processing delay.

4.6 Summary

In this chapter, the author aimed to improve Web browsing efficiency, by proposing the HERB method, which utilizes user context in order to infer the blocks relevant to a hyperlink. By classifying the hyperlinks in Web pages, the author was able to clarify the conditions under which the relevant blocks should be inferred. The author quantitatively evaluated the effectiveness of HERB through experi-

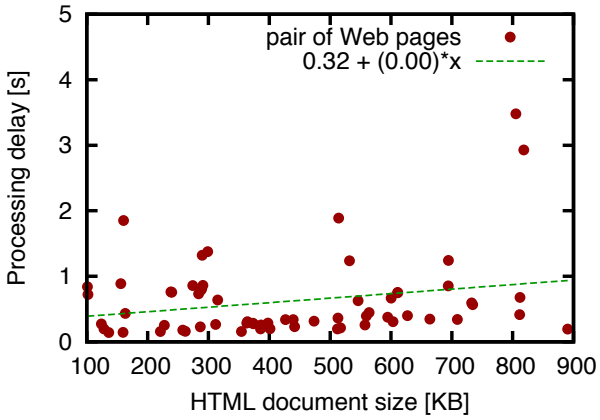


Figure 4.12: Processing delay distribution of HERB-enabled Web Proxy.

ments simulating ordinary Web browsing. Our experiments showed that HERB can infer blocks relevant to a hyperlink with approximately 65% precision and 70% recall, as well as infer the most relevant blocks in a page with approximately 30% precision and 80% recall. Furthermore, the author designed two HERB implementations, namely, a Web proxy and a Web browser, and the author discussed their advantages and disadvantages and also presented an overview of the Web proxy prototype and an example use case.

This chapter has demonstrated the feasibility and effectiveness of a fine-grained relevant block inference using a combination of Web page segmentation and text mining techniques (i.e., TF-IDF score and a cosine similarity measure). The primary objective of this chapter was to show that user context in Web browsing is helpful in identifying the relevant parts of a destination Web page, leading to significant improvement in Web browsing efficiency. However, several issues remain to be addressed in order to make HERB highly accurate, general, and versatile. For instance, it is important to develop to a superior block-to-block similarity measure, which is used various advanced natural language processing and machine learning techniques (e.g., text-formatting, Support Vector Machine and etc. [76]), than the simple combination of the TF-IDF score and cosine similarity measure which is currently used in HERB. As discussed in Section 4.4.1,

an arbitrary Web page segmentation method can be used in HERB, but the performance of HERB with different segmentation methods should be quantitatively evaluated. In addition, experiments of broader scope should be carried out using datasets composed of different types of Web pages.

Chapter 5

Conclusion

In this thesis, the author has attempted to establish virtual content-centric networking (VCCN) that realizes efficient and secure content retrieval and distribution on a content-centric network.

In the first part of this thesis (Chapter 2), the author has proposed VCCN, which realizes group-based communication in CCN. Specifically, group-based communication in CCN allows consumers to retrieve content only from authorized distributors and allows distributors to distribute content only to authorized consumers. The fundamental idea of VCCN is to operate a CCN router as multiple logically independent instances of VCCN routers. Group-based communication is realized by building VCCN slices, which are composed of multiple VCCN router instances. The author has implemented VCCN's basic features by extending the CCNx software and has conducted a preliminary performance evaluation of our implementation. Through a preliminary performance evaluation of the VCCN implementation, the author has showed that introduction of VCCN has both positive and negative impacts on CCN performance and that CCN router virtualization in VCCN incurs a little overhead to CCN in terms of the content delivery time. The author has also discussed open research issues in VCCN network construction based on knowledge acquired by designing, implementing and evaluating VCCN.

In the second part of this thesis, the author has tackled two issues on efficiency that result from introduction of VCCN in terms of content retrieval and

distribution.

First, in Chapter 3, the author analytically and quantitatively has investigated a trade-off among the network fairness for VCCN slices and overall network performance. Specifically, the author has investigated what resource allocation method provides the best balance between the network fairness for VCCN slices and overall network performance in three CS allocation methods (i.e., an exclusive method, a shared method and a hybrid method). The author has developed a mathematical model of virtualized CCN router for cache performance analysis under arbitrary content request patterns, and has derived the cache hit rate for each VCCN router instance and the aggregated cache hit rate of the virtualized CCN router. Using several numerical examples, the author has shown that when content request patterns are heterogeneous, a hybrid resource allocation method will provide the best balance between fairness and overall network performance.

Second, in Chapter 4, the author has proposed an application-level approach to improve the efficiency of Web browsing in order to further improve the efficiency of content retrieval in VCCN. The approach called HypErlink Referring Block estimation (HERB) segments Web pages into blocks and infers the existence and location of all relevant content on hyperlinked Web pages based on a block-to-block similarity. Through experiments simulating ordinary Web browsing, the effectiveness of HERB has been quantitatively investigated. The experiment results have showed that HERB can infer blocks relevant to a hyperlink with approximately 65% precision and 70% recall. These precision and recall are as high or higher than those of existing methods on a block-based Web search. Hence, the experiment results indicate that inference of relevant blocks by HERB will assist a user to search through relevant content of destination Web pages. Furthermore, the two HERB-enabled implementations, namely, a Web proxy and Web browser, have been also designed.

This thesis has presented a general and practical network architecture for realizing group-based communication on a content-centric network. However, several issues remain to be addressed in order to make VCCN reliable and flexible in the future. As one of issues, who names and manages VCCN identifiers and how

such tasks should be done should be considered. In addition, where VCCN router instances should be created or removed when a group is changed should also be considered.

Bibliography

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the fifth International Conference on emerging Networking EXperiments and Technologies (CoNEXT '09)*, pp. 1–12, Dec. 2009.

[2] J. Postel, "Internet protocol," *Request for Comments (RFC) 791*, Sept. 1981.

[3] R. Callon and M. Suzuki, "A framework for layer 3 provider-provisioned virtual private networks (PPVPNs)," *Request for Comments (RFC) 4110*, July 2005.

[4] Yammer Inc., "Yammer : The enterprise social network." `https://www.yammer.com`.

[5] LINE Corporation, "Line : Free calls & messages." `http://line.naver.jp/en/`.

[6] Google Inc., "Google+." `https://plus.google.com`.

[7] D. Y. Kim and J. Lee, "CCN-based virtual private community for extended home media service," *IEEE Transactions on Consumer Electronics*, vol. 57, pp. 532–540, May 2011.

[8] D. Y. Kim, M. wuk Jang, B.-J. Lee, and K. Kim, "Content-centric network-based virtual private community," in *Proceedings of the 29th International Conference on Consumer Electronics (ICCE 2011)*, pp. 843–844, Jan. 2011.

[9] P. S. Juste, D. Wolinsky, P. O. Boykin, M. J. Covington, and R. J. Figueiredo, "SocialVPN: Enabling wide-area collaboration with integrated social and overlay networks," *Computer Networks*, vol. 54, pp. 1926–1938, Aug. 2010.

[10] D. Smetters and V. Jacobson, "Securing network content," Tech. Rep. TR-2009-1, Xerox Palo Alto Research Center, 2009.

[11] E. Rosen and Y. Rekhter, "BGP/MPLS IP virtual private networks (VPNs)," *Request for Comments (RFC) 4364*, Feb. 2006.

[12] G. Rossini and D. Rossi, "A dive into the caching performance of content centric networking," tech. rep., Telecom ParisTech, 2011.

[13] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," tech. rep., Telecom ParisTech, 2011.

[14] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of CCN-caching trees," in *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I (Networking '11)*, pp. 78–91, May 2011.

[15] A. W. Lazonder, H. J. A. Biemans, and I. G. J. H. Wopereis, "Differences between novice and experienced users in searching information on the World Wide Web," *Journal of the American Society for Information Science*, vol. 51, pp. 576–581, Mar. 2000.

[16] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *Proceedings of tenth IEEE International Conference on Communications (ICC '11)*, pp. 1–6, June 2011.

[17] K. Ohsugi, K. Tsukamoto, and H. Ohsaki, "Study on the effect of CCN router virtualization on content delivery time (in Japanese)," *IEICE Society Conference 2012, B-7-4*, Aug. 2012.

[18] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Evaluating per-application storage management in content-centric networks," *Elsevier Computer Communications*, vol. 36, pp. 750–757, Apr. 2013.

[19] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *Proceedings of the First Workshop on Emerging Design Choices in Name-Oriented Networking at IEEE INFOCOM 2012 (NOMEN '12)*, pp. 310–315, Mar. 2012.

[20] S. Shenker, "The data-centric revolution in networking," in *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB 2003)*, p. 15, Sept. 2003.

[21] C. Esteve, F. L. Verdi, and M. F. Magalhães, "Towards a new generation of information-oriented internetworking architectures," in *Proceedings of the first Workshop on Re-Architecting the Internet (ReArch 2008)*, pp. 1–6, Dec. 2008.

[22] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *Communications Magazine, IEEE*, vol. 49, pp. 121–127, Mar. 2011.

[23] K. Cho, J. Choi, D. il Diko Ko, T. Kwon, and Y. Choi, "Content-oriented networking as a future internet infrastructure: Concepts, strengths, and application scenarios," in *Proceedings of the third International Conference on Future Internet Technologies (CFI 2008)*, June 2008.

[24] "Project CCNx." `http://www.ccnx.org/`.

[25] L. Zhang *et al.*, "Named Data Networking (NDN) project," Tech. Rep. NDN-0001, Palo Alto Research Center, Oct. 2010.

[26] K. Kanamori, "A data-oriented network architecture for group-based communication," Master's thesis, Graduate School of Information Science and Technology, Osaka University, Feb. 2010.

[27] W. A. Simpson, "The point-to-point protocol (PPP)," *Request for Comments (RFC) 1661*, July 1994.

[28] W. A. Simpson, "IP in IP tunneling," *Request for Comments (RFC) 1853*, Oct. 1995.

[29] "Facebook Graph API." `http://developers.facebook.com/docs/reference/api/`.

[30] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named-data networking," tech. rep., 2012.

[31] G. P. Alkmim, D. M. Batista, and N. L. da Fonseca, "Mapping virtual networks onto substrate networks," *Internet Services and Applications*, vol. 4, pp. 1–15, Jan. 2013.

[32] J. He, R. Zhang-Shen, Y. Li, C. yen Lee, J. Rexford, and M. Chiang, "DaVinci: dynamically adaptive virtual networks for a customized internet," in *Proceedings of the fifth International Conference on emerging Networking EXperiments and Technologies (CoNEXT 2008)*, p. 15, Dec. 2008.

[33] J. Lu and J. Turner, "Efficient mapping of virtual networks onto a shared substrate," tech. rep., WUCSE-2006-35, 2006.

[34] M. Ohtani, K. Tsukamoto, Y. Koizumi, H. Ohsaki, K. Hato, J. Murayama, and M. Imase, "VCCN: Virtual content-centric networking for realizing group-based communication," in *Proceedings oeedings of 12th IEEE International Conference on Communications (ICC '13)*, pp. 2069–2073, June 2013.

[35] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: Modeling, design and experimental results," *IEEE J. Selected Areas in Communications*, vol. 20, pp. 1305–1314, Sept. 2002.

[36] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *Proceedings of 24th International Teletraffic Congress (ITC '12)*, pp. 1–8, Sept. 2012.

[37] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proceedings of the 23rd International Teletraffic Congress (ITC '11)*, pp. 111–118, Sept. 2011.

[38] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Tech. Rep. DEC-TR-301, Digital Equipment Corporation, 1984.

[39] Ministry of Internal Affairs and Communications, Japan, "2013 white paper information and communications in japan (in Japanese).," Mar. 2013. `http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h25/pdf/25honpen.pdf`.

[40] Hakuhodo DY Media Partners Inc., "Media fixed point observation research 2010 (in Japanese).," June 2010. `http://www.media-kankyo.jp/upload/files/news_28/teiten2010.pdf`.

[41] T. Berners-Lee, L. Masinter, and M. McCahill, "Uniform Resource Locators (URL)," *Request for Comments (RFC) 1738*, Dec. 1994.

[42] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, "DOM-based content extraction of HTML documents," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 207–214, May 2003.

[43] L. Yi, B. Liu, and X. Li, "Eliminating noisy information in Web pages for data mining," in *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 296–305, Aug. 2003.

[44] S. Chakrabarti, "Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction," in *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, pp. 211–220, May 2001.

[45] J. Pasternack and D. Roth, "Extracting article text from the web with maximum subsequence segmentation," in *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, pp. 971–980, May 2009.

[46] J. Mahmud, Y. Borodin, and I. V. Ramakrishnan, "Csurf: A context-driven non-visual web-browser," in *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pp. 31–40, May 2007.

[47] Y. Borodin, J. Mahmud, and I. V. Ramakrishnan, "Context browsing with mobiles - when less is more," in *Proceedings of the Fifth International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 3–15, June 2007.

[48] W3C, "Document Object Model (DOM) Level 3 Core Specification.," Apr. 2004. `http://www.w3.org/DOM/`.

[49] Apple Inc., "Apple - safari - learn about the features available in safari.." `http://www.apple.com/safari/features.html`.

[50] Charmtech Labs LLC., "Charmtech labs home." `http://www.charmtechlabs.com/`.

[51] W3C, "HTML 4.01 specification.," Dec. 1999. `http://www.w3.org/TR/html4/struct/links.html`.

[52] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, pp. 513–523, Nov. 1988.

[53] S. Tata and J. M. Patel, "Estimating the selectivity of tf-idf based cosine similarity predicates," *ACM SIGMOD Record*, vol. 36, pp. 75–80, June 2007.

[54] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma, "Improving pseudo-relevance feedback in Web information retrieval using Web page segmentation," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 11–18, May 2003.

[55] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "VIPS: a vision-based page segmentation algorithm," Tech. Rep. MSR-TR-2003-79, Microsoft Research, Nov. 2003.

[56] C. H. Lee, M.-Y. Kan, and S. Lai, "Stylistic and lexical co-training for Web block classification," in *Proceedings of the Sixth Annual ACM International Workshop on Web Information and Data Management (WIDM '04)*, pp. 136–143, Nov. 2004.

[57] S. Baluja, "Browsing on small screens: recasting Web-page segmentation into an efficient machine learning framework," in *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pp. 33–42, May 2006.

[58] G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya, "Robust Web page segmentation for mobile terminal using content-distances and page layout information," in *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pp. 361–370, May 2007.

[59] J. Feng, P. Haffner, and M. Gilbert, "A learning approach to discovering Web page semantic structures," in *Proceedings of Eighth International Conference on Document Analysis and Recognition(ICDAR'05)*, pp. 1055–1059, Sept. 2005.

[60] D. Chakrabarti, R. Kumar, and K. Punera, "A graph-theoretic approach to Webpage segmentation," in *Proceedings of the 17th International Conference on World Wide Web(WWW '08)*, pp. 377–386, May 2008.

[61] C. Kohlschütter and W. Nejdl, "A densitometric approach to Web page segmentation," in *Proceeding of the 17th ACM Conference on Information and Knowledge Management(CIKM '08)*, pp. 1173–1182, Oct. 2008.

[62] S. Alcic and S. Conrad, "Page segmentation by Web content clustering," in *Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS '11)*, pp. 24:1–24:9, May 2011.

[63] Z. Markov and D. T. Larose, *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*. Wiley-Interscience, Apr. 2007.

[64] N. Craswell, C. Clarke, and I. Soboroff, "TREC 2010 Web track guidelines.," June 2010. `http://plg.uwaterloo.ca/~trecweb/2010.html`.

[65] W3C, "Document Object Model Core.," Apr. 2004. `http://www.w3.org/TR/DOM-Level-3-Core/core.html#ID-1312295772`.

[66] Hatena, "Hatena : Bookmark Hotentries." http://b.hatena.ne.jp/hotentry.

[67] MeCab, "MeCab: Yet Another Part-of-Speech and Morphological Analyzer.." http://mecab.sourceforge.net/.

[68] P. Morville, *Ambient Findability: What We Find Changes Who We Become*. O'Reilly Media, 1 ed., Sept. 2005.

[69] G. Upton and I. Cook, *Understanding Statistics*. Oxford University Press, Jan. 1997.

[70] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma, "Block-level link analysis," in *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval (SIGIR '04)*, pp. 440–447, July 2004.

[71] X. Wan, J. Yang, and J. Xiao, "Block-based similarity search on the Web using manifold-ranking," in *Proceedings of the Web Information Systems (WISE '06)*, vol. 4255, pp. 60–71, Oct. 2006.

[72] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol (HTTP/1.1)," *Request for Comments (RFC) 2616*, June 1999.

[73] W3C, "HTML5 a vocabulary and associated APIs for HTML and XHTML.," June 2010. http://www.w3.org/TR/html5/.

[74] W3C, "Web Style Sheets.." http://www.w3.org/Style/.

[75] Twisted Matrix Labs, "Twisted." http://twistedmatrix.com/trac/.

[76] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, Feb. 2007.